

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**JAVA PROGRAMLAMA DILI**

**00001.JPG**



**Dr. M. Turhan Çoban Nebraska Üniversitesi - Lincoln Makina Mühendisligi bölümü**

**N106 Walter Scott Engineering Center PO Box 880656**

**Lincoln, NE 68588-0656, U.S.A.**

**email :** [**MCOBAN2@netscape.net**](mailto:MCOBAN2@netscape.net)[**MCOBAN2@unl.edu**](mailto:MCOBAN2@unl.edu)

[**turhancoban@yahoo.com**](mailto:turhancoban@yahoo.com)

**web site:** [**www.cgimarket.com/~turhan/Java**](http://www.cgimarket.com/%7Eturhan/Java)

[ÖNSÖZ 6](#_TOC_250134)

[BÖLÜM 1 : JAVAYA GIRIS 8](#_TOC_250133)

* 1. [NIÇIN JAVA PROGRAMLAMA DILI 8](#_TOC_250132)
  2. [JAVA PROGRAMINI YAZMA VE ÇALISTIRMA 9](#_TOC_250131)
  3. [JAVA PROGRAMLAMA DILI TEMEL DEGISKEN TÜRLERI 20](#_TOC_250130)
     1. [Boolean degisken türü 20](#_TOC_250129)
     2. [char (harf) degisken türü 20](#_TOC_250128)
     3. [Tam sayi degisken türleri (byte, short,int,long) 21](#_TOC_250127)
     4. [Gerçek sayi degisken türleri (float, double,long double) 22](#_TOC_250126)
  4. [JAVA NESNESI OLARAK TANIMLANMIS TEMEL DEGISKENLER 22](#_TOC_250125)
     1. [String nesne tipi degiskeni 22](#_TOC_250124)
     2. [Integer nesne tipi degiskeni 23](#_TOC_250123)
     3. [Double nesne tipi degiskeni 23](#_TOC_250122)
     4. [diger nesne temelli degisken türleri 24](#_TOC_250121)
  5. [FINAL TERIMI VE SABITLER 24](#_TOC_250120)
  6. [ARITMETIK ISLEMLER 24](#_TOC_250119)
  7. [JAVADA MANTIKSAL ISLEMLER 27](#_TOC_250118)
  8. JAVA DILININ TEMEL KOMUTLARI 29
     1. if - elseif - else mantiksal karsilastirma yapisi 29
     2. while tekrarlama yapisi 32
     3. for tekrarlama yapisi 33
     4. switch - case yapisi 35
  9. ARITMETIK ISLEMLERDE DEGISKEN TÜRÜ DEGISTIRME (CASTING) OPERASYONU 39
  10. SAYI DEGIL VE SONSUZ SONUÇLARI 39
  11. ALISTIRMALAR 39

[BÖLÜM 2 : METODLAR VE OBJECT KULLANIMI 79](#_TOC_250117)

* 1. [JAVA API KÜTÜPHANESI 79](#_TOC_250116)
  2. [METOTLAR 82](#_TOC_250115)
  3. [NESNE (OBJECT) TANIMI VE METOTLARDA KULLANIMI 84](#_TOC_250114)
  4. [METOTLARIN KENDI KENDINI ÇAGIRMASI (RECURSION) 86](#_TOC_250113)
  5. [AYNI ADLI METOTLARIN BIR ARADA KULLANILMASI (OVERLOADING) 89](#_TOC_250112)
  6. [METOT (METHOD) VE SINIF(CLASS) DEGISKENLERI 91](#_TOC_250111)
  7. [ALISTIRMALAR 94](#_TOC_250110)

[BÖLÜM 3 . SINIF (CLASS) YAPILARINA GIRIS 124](#_TOC_250109)

* 1. STANDART KAVRAMLARIN TÜRKÇE KARSILIKLARI 124
  2. [SINIF YAPISININ GENEL TANIMI 124](#_TOC_250108)
  3. [KURUCU ( CONSTRUCTOR ) METOT 126](#_TOC_250107)
  4. [DIGER METOTLAR 127](#_TOC_250106)
  5. [THIS DEYIMININ KULLANIMI 129](#_TOC_250105)
  6. [SINIF DEGISKENLERININ DIS DÜNYADAN GIZLENMESI 129](#_TOC_250104)
  7. [SINIFLARDA KALITIM (INHERITANCE) 133](#_TOC_250103)
  8. [SINIFLARI BASKA BIR SINIFTA NESNE OLARAK ÇAGIRARAK BiR ARAYA GETiRME (COMPOSITION) 135](#_TOC_250102)
  9. [KALITIM YOLUYLA ALT SINIFA BAGLANMIS ÜST SINIF REFERANSI ÜZERINDEN ALT SINIFI ÇAGIRMA 136](#_TOC_250101)
  10. [NESNEYi DiNAMiK OLARAK SiLMEK FINALIZE() METOTU 137](#_TOC_250100)
  11. [ALISTIRMALAR 137](#_TOC_250099)

[BÖLÜM 4 : BOYUTLU DEGISKENLER VE NESNELER (ARRAYS) 202](#_TOC_250098)

* 1. [TEK BOYUTLU DEGiSKENLER 202](#_TOC_250097)
  2. [TEK BOYUTLU NESNE TiPi DEGiSKENLER 206](#_TOC_250096)
  3. [ÇOK BOYUTLU DEGISKENLER 206](#_TOC_250095)
  4. [BOYUTLU DEGISKENLERIN METOTLARA AKTARIMI 210](#_TOC_250094)
  5. [BOYUTLU DEGISKENLERDE BOYUT DEGISTIRME 210](#_TOC_250093)
  6. [ALISTIRMALAR 213](#_TOC_250092)

[BÖLÜM 5 : SINIFLARDA HIYERARSI, ABSTRACT SINIF VE INTERFACE 227](#_TOC_250091)

* 1. [ABSTRACT SINIF 227](#_TOC_250090)
  2. [INTERFACE 231](#_TOC_250089)
  3. [ALISTIRMALAR 235](#_TOC_250088)

[BÖLÜM 6 : GRAFIKLER, FONTLAR VE RENKLER 244](#_TOC_250087)

* 1. [GIRIS 244](#_TOC_250086)
  2. [GRAPHICS VE GRAPHICS2D SINIFLARI 244](#_TOC_250085)
  3. [RENK KONTROLÜ 247](#_TOC_250084)
  4. [YAZI KONTROLU 253](#_TOC_250083)
  5. [ÇiZGi ÇiZiMi 257](#_TOC_250082)
  6. [DiKDÖRTGEN ÇIZiMi 260](#_TOC_250081)
  7. [ÇIZILEN SEKILLERIN DEGISTIRILEREK ÇIZIMI (TRANSFORM) 265](#_TOC_250080)
  8. [OVAL VE AÇILI OVAL ÇIZIMI 267](#_TOC_250079)
  9. [ÇIZIMIN SEÇILEN BIR RESIMLE DOLDURULMASI 270](#_TOC_250078)
  10. [POLYGON ÇIZIMI 273](#_TOC_250077)
  11. [GENELLESTIRILMIS EGRI ÇIZIMI 276](#_TOC_250076)
  12. [RESIM GÖSTERIMI 280](#_TOC_250075)
  13. [ALISTIRMALAR 281](#_TOC_250074)

[BÖLÜM 7: GRAFIK APPLET VE ÇERÇEVE OLUSTURMA OLUSTURMA METOTLARI, GRAFIK KULLANICISI ARABIRIM PROGRAMLARI (GUI) , 308](#_TOC_250073)

* 1. GRAPHIC KULLANICISI ARABIRIM PROGRAMLARI, GUI, (GRAPHICS USER INTERFACE) 308
  2. [AWT LABEL (ETIKET) SINIFI , JAVAX JLABEL VE ICON SINIFLARI 309](#_TOC_250072)
  3. [BUTTON VE JBUTTON (DÜGME) SINIFLARI 315](#_TOC_250071)
  4. [TEXTFIELD (YAZIM ALANI) SINIFI 322](#_TOC_250070)
  5. [YAZIM ALANI AWT TEXTAREA VE SWING JTEXTAREA SINIFLARI 326](#_TOC_250069)
  6. [AWT CHOICE , SWING JCOMBOBOX SEÇIM SINIFLARI 329](#_TOC_250068)
  7. [AWT, CHECKBOX VE CHECKBOXGROUP VE SWING JCHECKBOX VE JRADIOBUTTON SINIFLARI 332](#_TOC_250067)
  8. [MOUSE(FARE) KONTROLU 339](#_TOC_250066)
  9. [AWT, LIST SINIFI VE JAVA SWING JLIST SINIFI , LISTEDEN SEÇIM 345](#_TOC_250065)
  10. [SWING JMENU 355](#_TOC_250064)
  11. [SWING JSLIDER SINIFI VE JPANEL TEMEL ÇIZIM ELEMANI (PANELI) 359](#_TOC_250063)
  12. [FLOWLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI 362](#_TOC_250062)
  13. [BORDERLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI 364](#_TOC_250061)
  14. [GRIDLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI 366](#_TOC_250060)

[7.17 JSPLITPANE SINIFI KULLANILARAK FORMATLAMA 369](#_TOC_250059)

* 1. [PANEL VE JPANEL SINIFI 371](#_TOC_250058)
  2. [JOPTIONPANE SINIFI 376](#_TOC_250057)
  3. [JFILECHOOSER SINIFI 379](#_TOC_250056)
  4. [JAVA JAR (JAVA ARCHIVES - JAVA ARSIVI) YAPILARININ KULLANIMI 381](#_TOC_250055)
  5. [ALISTIRMALAR 382](#_TOC_250054)

1. [JAVADA HATA ANALIZI VE YAKALANMASI 472](#_TOC_250053)
   1. [HATA ANALIZI 472](#_TOC_250052)
   2. [ALISTIRMALAR 476](#_TOC_250051)
2. [PARALEL KULLANIM (MULTITHREADING) , GERÇEK ZAMAN UYGULAMALARI, ANIMASYON 486](#_TOC_250050)
   1. [PARALEL KULLANIM(MULTITHREADING) VE GERÇEK ZAMAN PROGRAMLANMASI 486](#_TOC_250049)
   2. [PARALEL PROGRAM KULLANIMIDA HAFIZA SINKRONIZASYONU 491](#_TOC_250048)
   3. [ALISTIRMALAR 499](#_TOC_250047)

[BÖLÜM 10 JAVA GIRDI - ÇIKTI PROGRAMLANMASI 528](#_TOC_250046)

* 1. [JAVANIN I/O (GIRDI ÇIKTI ) KÜTÜPHANESINDE YER ALAN SINIFLAR VE INTERFACE’LER](#_TOC_250045)

........................................................................................................................................................................ 528

* 1. [FILE (DOSYA) SINIFI 530](#_TOC_250044)
  2. [ARDISIK (SEQUENTIAL) DOSYA YARATILMASI 532](#_TOC_250043)
  3. [RASLANTISAL ULASIM DOSYASI OKU (RANDOM ACCESS FILE) 555](#_TOC_250042)
  4. [DOSYA SIKISTIRILMASI (GZIP,GUNZIP,ZIP,UNZIP) 563](#_TOC_250041)
  5. [ALISTIRMALAR 571](#_TOC_250040)

[BÖLÜM 11 GELISMIS JAVA BILGI ISLEME YAPILARI 644](#_TOC_250039)

* 1. [STRINGTOKENIZER SINIFI 644](#_TOC_250038)
  2. [STRING BUFFER SINIFI 648](#_TOC_250037)
  3. [VECTOR SINIFI 650](#_TOC_250036)
  4. [LIST(LISTE) SINIFI 660](#_TOC_250035)
  5. [DIZI (STACK) SINIFI 667](#_TOC_250034)
  6. [SIRA (QUEUE) SINIFI 670](#_TOC_250033)
  7. [TREE(AGAÇ) SINIFI 672](#_TOC_250032)
  8. [DICTIONARY ve HASHTABLE SINIFLARI 675](#_TOC_250031)
  9. [ARRAYS SINIFI VE SIRALAMA 677](#_TOC_250030)
  10. [ALISTIRMALAR 678](#_TOC_250029)

[BÖLÜM 12 ÖRNEKLERLE SAYISAL ANALIZ 695](#_TOC_250028)

* 1. [SAYISAL ANALIZE GIRIS 695](#_TOC_250027)
  2. [MATRIX SINIFI 695](#_TOC_250026)
  3. [MATRIX SINIFI ÖRNEK PROGRAMLARI 725](#_TOC_250025)
  4. NUMERIC SINIFI (SAYISAL ANALIZ PAKETI) ÖRNEK PROBLEMLERI 753
  5. [ALISTIRMALAR 777](#_TOC_250024)

[BÖLÜM 13 DATABASE (VERI TABANI) PROGRAMLAMASINA GIRIS 778](#_TOC_250023)

* 1. [TEMEL KAVRAMLAR 778](#_TOC_250022)
  2. [JAVA VE SQL BAGLANTISI 781](#_TOC_250021)
  3. [ALISTIRMALAR 789](#_TOC_250020)

BÖLÜM 14 JAVA ANADIL (NATIVE LANGUAGE) PROGRAMLAMASINA GIRIS 815

C++ ve C PROGRAMLAMA DILLERININ JAVA ILE BIRLIKTE KULLANIMI 815

* 1. [TEMEL KAVRAMLAR 815](#_TOC_250019)
  2. [DEGISKENLERIN VE JAVA NESLERININ AKTARILMASI 816](#_TOC_250018)

[BÖLÜM 15 JAVA BEANS (JAVA FASULYELERI) PROGRAMLAMA 827](#_TOC_250017)

* 1. [TEMEL KAVRAMLAR 827](#_TOC_250016)
  2. [ALISTIRMALAR 843](#_TOC_250015)

1. [NETWORK PROGRAMLAMAYA GIRIS 847](#_TOC_250014)
   1. [TCP/IP PROTOKOLÜ 847](#_TOC_250013)
   2. [ALT SEVIYE ILETISIM : UDP KULLANIMI 848](#_TOC_250012)
   3. [TCP BILGI ILETIMI, SOCKET SINIFI 850](#_TOC_250011)
   4. [INTERNET SITELERIYLE BILGI ALISVERISI, URL SINIFI 853](#_TOC_250010)
2. [GÜVENLIK 860](#_TOC_250009)
   1. [APPLETLERDE GÜVENLIK 860](#_TOC_250008)
   2. [KONSOL PROGRAMLARINDA GÜVENLIK 866](#_TOC_250007)
   3. [GÜVENLIK SERTIFIKALARI 868](#_TOC_250006)
   4. [JAR DOSYALARININ GÜVENLIK KODUYLA IMZALANMASI 869](#_TOC_250005)
   5. [BROWSERLARDA GÜVENLIK 869](#_TOC_250004)

[EK A TEXT SINIFI 871](#_TOC_250003)

[EK B JAVA DERSI SINAV SORULARI 876](#_TOC_250002)

JAVA PROGRAMLAMA DILI GEBZE YÜKSEK TEKNOLOJI ENSTITÜSÜ GÜZ DÖNEMI 2000 BITIRME SINAVI SORULARI 876

[JAVA PROGRAMLAMA DILI GEBZE YÜKSEK TEKNOLOJI ENSTITÜSÜ 20 OCAK 2000 BITIRME SINAVI SORULARI 885](#_TOC_250001)

[18 REFERANS LISTESI 898](#_TOC_250000)

### ÖNSÖZ

Bilgisayar programlamasi günümüz teknolojisinin önemli bir ögesidir. Benim için bu yolculuk 1975 de aldigim ilk Fortran IV programlama dili kurduyla basladi diyebilirim. O dönemde genelde teorik bazda gördügümüz programlama dersinde yaptigimiz tek gerçek program örnegi kartlara delerek verdigimiz ve sonuçlari ertesi gün aldigimiz ikinci dereceden denklemin köklerini hesaplar bir programdi.Kullandigimiz ege üniversitesi bilgisayar merkezindeki IBM tabanli bir "mainframe" bilgisayardi. 1978 yilinin sonlarinda Unix tabanli siyah beyaz monitörler veya kagit yazicili monitorler araciligiyla çalisan IBM makinalarla çalismaya basladim. Tabii yil geregi Fortran 77 versiyonuna ulasmisti. 1982 yilinda Digital VAX bilgisayarlarinda programlamaya ve ayni zamanda sistem görevlisi olarak çalismaya basladim. Bu bilgisayarlar virtual (sanal) hafiza özellikleriyle çok büyük programlarin kullanilmasina izin veriyorlardi. Dil olarakta Fortranin yaninda Pascal da vardi. Üç boyutlu çizim ortami için özel olarak gerlistirilms evans-sutherland bilgisayari ile de bu dönemde tanisma firsatim oldu. Bu arada Commodore 64 bilgisayarlari piyasaya çikmisti. Evimde kullandigim bu küçük aletle asembler ve basic dillerini kullanarak birçok program gelistirdim. hatta assembler da yazilmis türkçe bir kelime islem programi bile vardi. 80 li yillarin sonu ve 90 li yillarin basinda Pc ler, dos ortami ve diliyle tanistim. sonra PC ortami Machintosh ve Amigadan çok sonra grafik programlama ortamini büyük bir bulus olarak ortaya atti ve Windows sistemlerini çikardi. Ve ikinci bilgisayarimi param oldukça aldigim parçalari birlestirerek kendim olusturdum. Bu bir 80386 Pc bilgisayari idi. Artik genelde c dilini kulaniyordum. C++ dilini gerçek anlamda kullanmaya baslamam, bilgisayar konusundaki lisans üstü kursunu alirken oldu. Bundan sonraki tüm programlama uygulamalarimda nesne kökenli olan bu dili kullanir oldum. Ayni kursta Lisp, Parallaxis gibi degisik dillerle de çalismistim. Bu arada evde de yine PC tabanli Pentium 100 bilgisayarina terfi ettik. Bana Java dilini esimin aldigi lisans üstü kursundaki Java kursu gösterdi. Java dili yeni çikmis, fakat yeni olmasina ragmen okullardaki bilgisayar bölümleri egitimde hizla bu dile yönelmislerdi. Bu dile baslar baslamaz da çok sevdim.

Her sey tüm diger dillere göre çok daha iyi düsünülmüs ve planlanmisti. Orijinal olarak C++ da yazilmis bir çok kodu fazla bir gayrete gerek duymadan java koduna çevirerek is ortamimda kullanmaya basladim. 1998 yilinda Dokuz eylül üniversitesinde mühendislik bilimleri fakultesinde ögretim üyeleri ve master ve doktora ögrencileri için "Java programlamlama dili" dersi açtim. Bu kitabin temelini bu ders için hazirladigim ders notlari olusturmustur. 1999 yilinda Gebze ileri teknoloji Enstitüsünde ayni dersi yinelerken ders notlari ilave problemlerle de zenginleserek biraz daha kitap halini aldi. Son bir gözden geçirmeden sonra elinizdeki haline geldi.

Java dili yapisi, kullanim olasiliklari ve kolay kullanimi, zengin kütüphaneleriyle gelecegin dili olacagina programcilarin çogunun inandigi bir dildir. C++ su an itibariyle çok daha fazla kullanilan bir dil olam özelligini sürdürmektedir, fakat C++ hatalara izin veren yapisiyla baslangiç seviyesi programcilara hitap eden bir dil degildir. Rahatlikla hata yapabilirsiniz, ve yaptiginiz bu hatalar rahatlikla gözden kaçabilir. Hemen sunu ilave edeyim, java daha yeni emeklemeye baslamis bir bebektir, ama gelecek on yil içinde çok iyi bir konuma yerlesmenin isaretlerini simdiden vermektedir.

Türkiyede de programcilik henüz yeni yeni olusmaya baslayan bir dal. Hem programci kapasitesi, hem de isteklere cevap vermek için yeterli boyutta degil henüz.Programlama teknolojisinde iyi bir boyuta gelebilmek paket kullanimindan degil programlamadan geçmektedir. Bu yüzden bu kitap eger yeni programlamacilarimiza bir seyler verebilir ve onlarin gelecekte daha iyi bir noktada olabilmelerini saglayabilirse ben emegimin karsiligini almis olurum.

Yeni baslayanlara bir tavsiye bilgisayar dillerini ögrenme biraz insan dillerini ögrenmeye benzer. Temel mantigini kapana kadar biraz zorlanabilirsiniz. Sakin ümitsizlige kapilip birakmayin. Bir kere temel mantigini anladiktan sonra ne kadar kolay oldugunu göreceksiniz. Iyi çalismalar.

Dr. Turhan Çoban TÜBITAK, MAM 21 Mart 2000, 21.36

[turhan@mam.gov.tr](mailto:turhan@mam.gov.tr)

### IKINCI BASKININ ÖNSÖZÜ

Bu Java kitabinin birinci baskisi umdugumun çok üstünde ilgi gördü. Saniyorum bunun temel sebebi türkçe kaynak eksikliginin oldukça fazla olmasi. Bu ilgi üzerine kendimi biraz daha ciddi bir çalistirma olusturmak için sizlere borçlu hissettim. Umuyorum bu ikinci baski çok daha fazla isinize yarayacak ve sadece bir ders notu olmanin disinda biraz daha bir referans kitabina yaklasacak bir eser olacak. Kitabin ilk baskisinda okuyuculardan oldukça yogun mektuplar aldim, bu mektuplarda begenilerinin yani sira benim de hakli buldugum elestiriler de

yer aliyordu. Gelen elestirilerden en yogunu kitabiniçindeki kodlarin bir CD olarak sunulmamis olmasiydi. Bu ikinci baskida bunu saglayabilmek için elimden geleni yapacagim, fakat basarili olamazsam kodlari [www.geocities.com/turhan\_coban/](http://www.geocities.com/turhan_coban/) adresi üzerinden yayinlamaya çalisacagim. Ikinci elestiri Text sinifini bulamamak veya çalistiramamak idi. Text sinifini ben özellikle eklerde de vermistim. Bu sinifin amaci javanin çok yogun kullandigi hata analizini asarak en azindan baslangiçta okuyuculara (ögrenicilere?) kolaylik saglamakti. Derslerde ögrenciler için bu kolayligi sagladi da, fakat sanirim kitapta kodlarin verilmeyisiyle birlikte kolaylik yerine zorluk getirdi. Bu ikinci baskida birinci bölümden itibaren girdi çiktida birden fazla alternatif bulacaksiniz. Bunlarin birisi de swing sinifindan olan JoptionPane sinifi, kullanilmasi gayet basit olan pencere tipi bir girdi çikti sinifi.Swing sinifini bu baskida bir öncekine göre oldukça yogun kullandim. Bazi programlarinda aradaki benzerlik ve farklari görebilmeniz amaciyla swing ve awt versiyonlarini arka arkaya koydum. Swing awt’ye göre çok daha kompleks bir yapi, yalniz olasiliklari arttirdigi için java dünyasina simdiden tamamen hakim olmus durumda. Bu baskida ilk baskida olmayan java güvenlik ve java anadil programlama bölümlerini bulacaksiniz. Ayrica daha öce ayri bir bölüm olarak verdigimiz swing ve 2D grafik programlama bölümünü de ilgili konulara yayarak kaldirdim. Bence ilk baskiya göre elinizde daha kullanisli bir kitap var. Elbette mükemmel degil, ama ilk baskinin en azindan dört kati bir emegin sonucu. Umarim sizlere java ögrenme yolunda bir ilk adim olarak faydali olur.

Bu yeni baskiyi gelistirmemde büyük paylari olan Gebze Yüksek Teknoloji Üniversitesi, java dersi ögrencilerine, ayrica bana iyi bir egitim vermek için ellerinden gelen çabayi harcayan babam Osman ve annem Hatice’ye, Kardeslerim Birsen, Nurhan ve Irfan’a ve bu kitabin yazilmasi sirasinda verdigi destek için esim Meral’e tesekkürlerimi bildirmek isterim.

Dr. Turhan Çoban TÜBITAK, MAM 11 Mart 2001, 23.34

[Turhan.Coban@posta.mam.gov.tr](mailto:Turhan.Coban@posta.mam.gov.tr)

### BÖLÜM 1 : JAVAYA GIRIS

### NIÇIN JAVA PROGRAMLAMA DILI

Java Programlama dili su anda dünyadaki en popüler programlama dillerinden biri haline gelmistir. Java SUN bilgisayar sirketince orijinal olarak elektrikli ev araçlarinin (mikrodalga firinlari, buzdolaplari , televizyonlar, uzaktan kumanda cihazlari vs.) birbiriyle haberlesmesini saglamayi amaçlayan bir proje içerisinde 1991 yilinda gelistirilmeye baslandi. Orijinal adi bu dilin yaraticilari James Gosling, Patrick Naughton, Chis Wartdh, Ed Frank ve Mike Sheridan tarafindan **Oak** olarak konulan programlama dili daha sonra bu isimde baska bir programlama dili oldugu kesfedilince o anda bir kahvehanede kahve içen programlama gurubu tarafindan kahve markasindan esinlenerek **Java** olarak degistirildi. Akilli elektronik ev araçlari pazari SUN gurubunun tahminlerinden çok daha yavas bir gelisme gösteriyordu. Bu yüzden Java dili projesi ticari bir gelistirme projesi olarak büyük olasilikla iptal edilecekti. 1993 Yilinda "World Wide Web" büyük bir atilim göstererek bütün dünyaya yayilmaya basladi. Javanin Dinamik Web sayfalari hazirlamadaki büyük potansiyelini gören SUN sirketi projeyi bu tarafa yönlendirdi ve bu javaya yeni bir canlilik ve yasama umudu sagladi.

Mayis 1995 de SUN javayi büyük bir konferansta tanitti. Program is dünyasi tarafindan derhal büyük bir ilgiyle karsilandi. Java Modern bilgisayar dünyasinin ses, grafik islem, haberlesme gibi ihtiyaçlarina cevap verebilen ve Ticari gayeler için hazirlanan bir Program dili olarak daha önceki bilgisayar dillerinin hiç birinin kapsayamadigi özellikleri içermekteydi. Bunun yani sira dil komut yapisi olarak C++ diline çok yakin olmasi da ögrenilmesini kolaylastiriyordu. SUN Javayi "World Wide Web" de kullanmak isteyen herkese ücretsiz olarak sundu. Java internette yayinlanmasinin ardindan çok büyük bir patlama yasadi. 1997 ye gelindiginde dünyadaki bütün bilgisayar okullarinda temel bilgisayar dili olarak gösterilmeye baslandi. Dünyada su anda hala en çok kullanilan bilgisayar dili olan C++ dilinin yapilan hatalari tam olarak denetlememesi programin çalisma hizini arttirma yönünden iyi bir özellik olsa da profesyonel programcilar disinda kullanilmasini sinirlandirici bir etki yapiyordu. Java ise bütün hatalari bildiren yapisi ve modern bilgisayarin bütün fonksiyonlarina ulasabilen kütüphaneleriyle programcilarin çok daha kolaylikla ögrenebilecegi bir dildir. Burada hemen sunu da belirtelim. C dili hizli çalisma amaci birinci planda tutularak yaratilmis bir dildir. Javada ise emniyet ilk planda yer almistir. Hiz açisindan düsünüldügünde java C (ve C++) diliyle rekabet edemez. zaten program derleyicisi de C++ dilinde yazilmistir.

Javanin diger önemli bir temel özelligi Nesne kökenli (object oriented) bir dil olmasidir. Nesne kökenli diller, nesneleri gerçek dünyadakine daha benzer bir yapida tanimlayarak anlasilmalarini kolaylastirirlar. Nesneleri gerçek dünyadaki gibi masa, sandalye, bilgisayar, gerçek gaz, isi esanjörü gibi tanimlayarak programlamak insan beyninin anlamasi açisindan çok daha kolaydir. Bundan önceki tam nesnel kökenli programlama dillerinden hiçbiri çok yayginlik kazanmamisti. Bu tür dillere Smalltalk,u Örnek verebiliriz. C++ nesnel kökenli programlama yapabilen bir dildir. Fakat yapisal (structural) birprogramlama dili olan C dilinin bir uzantisi olarak gelistirildiginden tam anlamiyla nesnel kökenli bir dil oldugu söylenemez.

Java dilini gelecegin dili yapan diger bir özelligi de çok kullanimli (multi-tasking) ve paralel kullanimli(multi- treading) bir dil olmasidir. Çok kullanimlilik birden fazla islemin ayni anda yapilabilmesinin tanimidir. Paralel kullanimlilik ise birden fazla programin ayni anda hafizayi beraber kullanarak kullanilabilmesidir. Örnek olarak World ve Excel programlarinin Windows NT ortaminda ayni anda kullanilmasidir. ADA gibi bazi eski Program dillerinde çok kullanimlilik programlanabiliyordu. Paralel kullanim olanaklarini sunan ilk bilgisayar dili ise javadir. Paralel kullanim paralel programlama kavramindan ayridir ve karistirilmamalidir. Paralel programlanmada birden fazla Bilgi islem Ünitesine (CPU ) ayri programlar veya bir programin ayri parçalari gönderilir. Paralel kullanimda ise bir CPU nun kullanim zamani küçük parçalara ayrilarak degisik Program veya Program parçaciklari bu zaman paketçiklerini paylasarak kullanirlar.

Javayi önemli bir Program dili haline getiren en önemli özelligi ise kullanilan bilgisayardan bagimsiz olmasidir. Javada yazilan bir Program Unix, Machintosh, Windows 95 veya Windows NT veya herhangi bir 32 bit makinada hiç degistirilmeden kullanilabilir. Java programlarinin grafikleri "Wold Wide Web" sayfalarinin programlama dili olan html (hypertext markup language) ile aktarilir. Bu yüzden html ve java programlarini birlikte kullanmak ve java programlarini gerçek zamanda www sayfalarinda göstermek mümkündür. Kendi web sayfanizi veya webde yazilmis kitabinizi bütün dünyaya aktarirken yaptiginiz analizleri de bu kitabin dinamik bir parçasi olarak sunabilirsiniz. Bu Türkiye gibi ülkelerin dünyaya kapilarini daha kolay açmalari açisindan oldukça önemli bir özelliktir.

Javanin HTML diliyle kullanilabilmesi, bazi kisilerde htmlin bir parçasi oldugu gibi bir kavram gelismesine yol açmistir. Programlamaya yeni baslayanlar sik sik html ile javayi birbirine karistirirlar. veya javanin HTML'nin bir uzantisi oldugunu düsünürler. Bu temelde oldukça yanlis bir varsayimdir. HTML internet belgelerini birbirine baglamak amaciyla gelistirilmis bir belge islem sistemidir ve bir programlama dili degildir. HTML ile javanin tek gerçek ilgisi HTML deki applet komutudur. bu komut yardimiyla java dilinde yazilmis programlarin sonuçlari HTML ortamina dinamik olarak aktarilabilir.

### JAVA PROGRAMINI YAZMA VE ÇALISTIRMA

Her programda oldugu gibi java programlarinda da ilk evre programi yazmaktir. Java programlari her editör programiyla yazilabilirler. Örnegin Unix sistemlerinde **vi** veya **emacs,windows 95** ve **windows NT** de **DOS Edit,** veya **Windows Notedefteri (notepad)** veya **WordPad** programlari kullanilabilir. Bildiginiz ve kullanmakta rahat oldugunuz herhangi bir editör programi varsa rahatlikla kullanabilirsiniz. **Windows Notedefteri (notepad)** editörüyle ilgili karsilasacabileceginiz bir programi burada hemen belirtmekte yarar var. Bu program tüm dosyalarin arkasini txt eki getirmeye çalisir. Programi direk içinde yazip çalistirabilen Microsoft Visual J++, Symantec Visual Café, Jpad, Borland JBuilder gibi Java gelistirme paketleri de mevcuttur. Bu kitapta bu paketler yardimiyla program yapilmasina direk olarak hiç deginilmemistir. Kisisel olarak bu tür paketler java dilini ögrenme açisindan bilhassa yeni programlama ögrenenlere çok fazla yardimi olacagini sanmiyorum, üstelik sadece o paketi kullanmayai ögreneceginiz için bir tür bagimliliga sebep olabilirler. Fakat bu paketler uzman hale geldikten sonra isinizi kolaylastirmak açisindan çok yararli olabilir.

Süphesiz programi yazmadan önce Java compilerinin bir kopyasini edinmeniz de gerekir. Java gelistirme seti (Java Developers kit - JDK) SUN bilgisayar sirketinin

[http://www.java.sun.com](http://www.java.sun.com/)

veya [http://www.javasoft.com](http://www.javasoft.com/)

internet adreslerinden bedava çekilebilir. Çekme zahmeti istemiyorsaniz SUN bu programlari CD formatinda da satmaktadir. Sun programin kitaplar ve üçüncü el kaynaklarda verilmesine müsade etmemektedir. Ayrica derleyici sürekli olarak yenilendiginden orijinal kaynagindan her zaman en yeni versiyonunu elde edebilirsiniz. Java ile yaptiginiz lisans anlasmasina göre program dilini bedava olarak kullanabilirsiniz, ancak satacaginiz profesyonel bir program gelistirdiginizde sun sirketine kazanciniz üzerinden belirli bir yüzde ödemeniz gerekecektir.

Java programi **isim.java** seklinde editörle yazilir. Java programiniz yazildiktan sonra ilk step java derleyici (compiler) programiyla **isim.class** ara bilgisayar kodunu olusturmaktir.

Java programini yazarken dikkat etmeniz gereken diger bir özellik de javanin nesne sinifinin ismiyle dosyadaki isminin ayni olmasidir. Bilgisayar konsol ekraninda (DOS veya UNIX Console)

### javac isim.java

Komutunu bilgisayara gönderdiginizde programinizda hata yoksa java derleyicisi **isim.class** dosyasini yaratir. Eger programinizda hata varsa hatalarin listesini verir. Programlamanin dili ingilizce oldugundan bu hata açiklamalari da ingilizcedir. Editör programina geri dönüp hatalari düzeltmeniz, programi tekrar kaydetmeniz ve sonra bu komutu tekrar denemeniz gerekecektir. Java programlari iki ortam için yaratilabilir. Birincisi Konsol ortamidir (DOS, Unix console gibi). Konsol ortamini kullanirken java dilinin konsol komutu java kullanilarak isim.class dosyasi yerel bilgisayarin makine diline program çalisirken dönüstürülür. Bu temelde bir tercüme (interpretation) islemidir. Sonuç olarak java dilinde iki kademeli bir proses kullanilir, birinci islemde java dosyasi class dosyasina çevirilir (derlenir), ikinci islemde ise class dosyasi yerel bilgisayarin makine diline dönüstürülür. konsol ortaminda çalistirilan program sonuçlari yine konsol ortaminda alinacagi gibi pop-up pencere grafik ortaminda da alinabilir. Ikincisi ise internet browser penceresi (Applet) ortamidir. Eger Program konsol ortaminda yazilmissa

### java isim

komutu kullanilarak sonuçlar ekranda veya açilan bir pencerede görülebilir. Eger Program applet ortaminda yazilmissa java programina ilave olarak html dilinde **isim.html** dosyasi açilir. En basit sekilde **html** dosyasi su

sekilde tanimlanabilir :

### <html >

**<applet code="isim.class" width=200 height=50** >

**</applet** >

**</html>**

Buradaki width pencerenin enini pixel olarak, height’da yüksekligini yine pixel(görüntü noktasi) olarak vermektedir. daha sonra bu dosya konsoldan

### appletviewer isim.html

komutuyla çalistirilabilir. Buradaki appletviewer komutu java program paketindeki basitlestirilmis bir browser programidir. Bu komut yerine isim.html dosyasi herhangi bir browser programina yüklenerek görülebilir.

Burada hemen sunu not etmeliyiz: braowser program yazicilari genelde java versiyonlarini kendi programlarina aktarmakta biraz yavas kalmaktadirlar bu yüzden en yeni java derleyicisiyle derlediginiz bir programi browser’iniz göstermeyebilir. Buna karsilik java “plug-in” denilen küçük programlar gelistirmistir. Diger bir olasilikta elbette eski java derleyicinizi hemen atmayip bir süre daha (yeni versiyonu browser programlari tarafindan tam olarak taninincaya kadar) profesyonel applet programlari gelistirirken kullanmanizdir.

Programcilikta adeta adet haline gelen ilk uygulama daima o dilde ekrana Hosgeldiniz çiktisi almaktir. Bizde bu kuralin disina çikmayalim. Asagidaki listede ilk konsol programimiz **Hosgeldiniz.java** görülmektedir.

### Program 1.1 Hosgeldiniz.java konsol programi

import java.io.\*; //java girdi cikti sinifini cagir class Hosgeldiniz

{

public static void main(String args[])

{

System.out.println("Java Sinifina Hos Geldiniz!");

}

}

bu programi çalistirmak için

### javac Hosgeldiniz.java java Hosgeldiniz

komutlarini konsolda vermemiz yeterlidir konsol çiktisi olarak:

### Java Sinifina Hos Geldiniz!

Yazisini aliriz.

Simdi bu programdaki satirlara tek göz atalim.

**import java.io.\*;** Bu deyim java konsol girdi çikti kütüphanesindeki metotlari programimiza aktarir.

### //java girdi cikti sinifini cagir

**//** isaretiyle baslayan yazilar bilgisayar tarafindan kullanilamaz sadece programi okuyani bilgilendirmek amaci tasir /\* ..... \*/ isaretleriyle de verilebilir. Örnegin :

### /\* Bu kisimda yazilanlarla bilgisayar ilgilenmez Sadece insanlarin programi anlamasi içindir

**\*/**

**class Hosgeldiniz** Class (sinif) nesne kökenli programlama yapisinin temel tasidir. Her Class'in kendine özgü bir ismi olmasi gerekir. Daha önce yazilmis Class'lar **import** deyimiyle programa çagirilabilirler. Yeniden

yazilmalari gerekmez. Class yapisi ve tanimlamalari üzerinde daha sonraki bölümlerde çok daha detayli bilgi verilecektir.

**{ }** baslatma ve bitirme : Sinif tanimlamasi da dahil olmak üzere Tüm Program parçaciklari { isaretiyle baslar ve

} isaretiyle son bulur. {} isaretleri arasinda yazilan program parçaciklarini normal metin yazilmasindaki paragraf kavramina benzetebiliriz. Ayrica program koduna baktigimizda "**;** " isaretine sik sik rastliyoruz. Bu isaret java komutunun bittigini belirtir. Cümle yagisinda kullanilan nokta "**.**" isareti gibi islev görür. Java programlarini istersek "**;**" ve "**{}**" ayraçlarini kullanarak tek bir satir olarak yazabiliriz. Fakat bu programin bizim tarafimizdan okunurlugunu zorlastirir. Program kodunu yazarken bosluk ve satir araliklari birakarak insanlar tarafindan programin okumasini kolaylastirabiliriz. Modern programciliktaki en büyük sorunlardan biri bigisayar kodlarinin bakim ve onarimi sorunudur. **Ticari olarak yazilan her programin belli bir ömrü oldugunu ve daha sonra baska bir programci tarafindan üzerinde tekrar çalisilacagini sürekli olarak hatirlamak zorundayiz.**

**public static void main(String args[])** diger dillerde fonksiyon (function), subroutine, procedure gibi adlarla anilan Program alt parçaciginin basligi. Bu Program parçaciklari javada Metot (Method) adiyla anilir. Metotlarin girdileri ve çiktilari olabilir. Bu metotun adi **main** (ana Program) dir. String degisken türü args[] degiskeni girdi olarak girebilir. Bu metotta **void** çikti türü kullanilmistir. Void kelimesi bu metotdan hiçbir çikti alinmayacagini belirtir. **public** metota disaridan ulasilabilme izni verir.

**System.out.println("Java Sinifina Hos Geldiniz!");** Bu satir System.out sinifindan println metotunu çalistirir. Bu metot method girisinde gönderilen String tipi degiskeni konsol çiktisina aktarir. Fortran dilindeki write(\*,\*) veya print\*, c dilindeki printf, c++ dilindeki cout<< terimlerinin javadaki tam karsiligidir.

**"Java Sinifina Hos Geldiniz!"** stringi aynen aktarilirken içinde c dilinde kullanilan temel kontrol terimleri yer alabilir ve aynen c dilindeki anlamlariyla kullanilabilir. Bu kontrol terimlerinden en yaygin olarak kullanilan bazilari:

\n alt satira geç

\t bir sonraki tab’a atla

\r gir tusu bas

\\ \ isaretini yaz

\” “ isaretini yaz seklinde gösterilebilir.

Ikinci konsol programimiz HosgeldinizSW.java Program 1.2 de verilmistir. Bu program SWING kütüphanesinin alt kütüphanesi olan JoptionPane kütüphanesi kullanilarak yaratilmistir. Çikti yazisi ekranda grafik pencere olarak belirir ve Java sinifina Hos geldiniz! mesajini verir. Tüm bu kütüphanelerin detaylari ilerideki bölümlerde verilecektir. Simdilik nasil kullanildiklarini ögrenmekle ve programlarimizi örnek programlara benzeterek yazmaya çalisacagiz. Uzman programcilar isterlerse ilgili bölümlerden komut detaylarina derhal girebilirler.

**Program 1.2** HosgeldinizSW.java ***programi***

import javax.swing.JOptionPane; class HosgeldinizSW {

public static void main (String args[])

{

JOptionPane.showMessageDialog(null,"Java sinifina hos geldiniz!", "HosgeldinizSW programi",JOptionPane.INFORMATION\_MESSAGE); System.exit(0);

}

}

Bu programi

javac HosgeldinizSW.java ve

java HosgeldinizSW

komutlarini kullanarak çalistirdigimizda

### 01001.JPG



**Sekil 1.1 HosgeldinizSW dosyasinin konsol SWING pop-up grafik çiktisi kullanarak görülmesi**

Bu çiktiyi alabilmek için java SWING kütüphanesi içerisinde yer alan **JOptionPane.showMessageDialog(....)** metodunu kullandik. Bu metodun görevi System.out.println metoduyla aynidir. Farki ise grafik ortamda çikti vermesidir. Metot içindeki ikinci sirada verilen String tipi sabit, "Java sinifina hos geldiniz!" grafik çiktiya aktarilmistir. Üçüncü sirada verilen string tipi "HosgeldinizSW programi" sabiti de grafik kutusunun adini belirtmistir. JoptionPane.INFORMATION\_MESSAGE komutu ise bilgi (information) mesaji penceresi açilacagini belirtmistir. (grafikteki i harfi – information – bilgi kelimesinin bas harfidir). Bu alanda kullanabilecegimiz diger alternatifler sunlardir :

JOptionPane.ERROR\_MESSAGE : hata mesaji JoptionPane.WARNING\_MESSAGE : uyari mesaji JoptionPane.QUESTION\_MESSAGE : soru mesaji JoptionPane.PLAIN\_MESSAGE : sade (ikonsuz) mesaj

System.exit(0); metodu çikti penceresinde görülen OK dügmesine basildiginda pencerenin kapanmasini saglar.

**import javax.swing.JoptionPane;** deyimi swing kütüphanesinde yer alan JoptionPane sinifini çagirir. Programdan da görülecegi gibi sinifimizin (class) adi HosgeldinizSW dir. Bir önceki programdaki gibi bu programda kullandigimiz komutlar **public static void main(String args[])** komutuyla tanimlanan ana metodun içinde yazilmistir.

Üçüncü konsol program örnegimizde iki java programini birlestirecegiz Ilk programimiz BasicWindowAdaptor.java Program 1.3 de verilmistir. Program 1.4 ise HosgeldinizSWF.java programini kapsamaktadir. BasicWindowMonitor programini bundan sonraki programlarimizda sik sik kullanacagiz.

**Program 1.**3 **BasicWindowMonitor.java *programi***

import java.awt.event.\*; import java.awt.Window;

public class BasicWindowMonitor extends WindowAdapter

{

public void windowClosing(WindowEvent e)

{

Window w=e.getWindow(); w.setVisible(false); w.dispose();

System.exit(0);

}

}

**Program 1.4** HosgeldinizSWF.java ***programi***

import javax.swing.\*;

import BasicWindowMonitor;

public class HosgeldinizSWF

{

public static void main(String[] args)

{

JLabel etiket=new JLabel("Java Programina Hos geldiniz!"); JFrame cerceve=new JFrame("HosgeldinizSWF"); cerceve.addWindowListener(new BasicWindowMonitor());

cerceve.getContentPane().add(etiket); cerceve.pack(); cerceve.setVisible(true);

}

}

Burada iki ayri program bulunmaktadir. Bu iki programlik seti çalistirmak için önce iki programi da derlemek gerekir.

### javac BasicWindowMonitor.java javac HosgeldinizSWF.java

sonra ikinci prgrami

### java HosgeldinizSWF

komutunu kullanarak çalistirabiliriz. Burada birinci programin görevi hosgeldinizSWF penceresini kösedeki X isaretine basdigimizda pencereyi kapatmaktir. Bu programi bundan sonraki birçok uygulamamizda kullanacagiz. Programimiz JFrame denilen grafik pencere ortaminda yazilmistir. Çikti penceresi :

### 01002.JPG



**Sekil 1.2 HosgeldinizSWF dosyasinin konsol SWING JFrame pop-up grafik çiktisi kullanarak görülmesi**

Görüldügü gibi burada yazma islemini yaptirmak için tek bir komut yerine bir komutlar ve metotlar zinciri kullanildi. Bu program swing grafik kütüphanesiyle grafik formatinda çikti almanin çok daha standart bir yoludur. Temel olarak SWING grafik penceresine JLabel sinif tipi etiket eklenerek yazi grafik ekrana aktarilmistir.

Program 1.5 de yine swing metodu kullanilmistir. Buradaki küçük farklilik bir önceki HosgeldinizSWF programinda kullanilan Jlabel nesnesi üzerinden yazdirma yerine paint metodunun kullanilmis olmasidir. Burada da BasicWindowMonitor programi kullanilmistir. fakat bu programi daha önce derlemis oldugumuzdan tekrar derlememiz gerekmez. Sadece

**javac Hosgeldiniz1SWF.java** komutuyla bu programi derlememiz ve **java Hosgeldiniz1SWF**

komutuyla çalistirmamiz kafidir.

**Program 1.5** Hosgeldiniz1SWF.java ***programi***

import javax.swing.\*; import java.awt.\*;

import BasicWindowMonitor;

public class Hosgeldiniz1SWF extends Jframe

{

public void paint(Graphics g)

{

g.drawString("Java sinifina hosgeldiniz!",10,40);

}

public static void main(String[] args)

{

Hosgeldiniz1SWF cerceve=new Hosgeldiniz1SWF(); cerceve.addWindowListener(new BasicWindowMonitor()); cerceve.setSize(160,50);

cerceve.setVisible(true);

}

}

Program çiktisi :

### 01003.JPG

**Sekil 1.3 Hosgeldiniz1SWF.java dosyasinin konsol SWING pop-up grafik çiktisi kullanarak görülmesi**

Burada yaziyi temel olarak bir grafik metodu olan g.drawString("Java sinifina hosgeldiniz!",10,40);

metodunu kullanarak pencere ortamina aktardik. bu metodda kullanilan 10,40 deyimi yazinin grafik ortamindaki x ve y koordinatlarini göstermektedir. y koordinati yukardan asagi dogru tanimlanmistir. cerceve.setSize(160,50);

deyimi açilan pencerenin boyutunun 160 genisliginde ve 50 yüksekliginde olacagini belirtmektedir. cerceve.setVisible(true);

terimi ise pencerenin görünmesini (çizilmesini) saglamaktadir.

Buraya kadar kullandigimiz programlar konsol programlariydi. Bu programlarda da gördügünüz gibi grafik ortamini yogun olarak kullanabiliyoruz. Yalniz programlarin çalistirilmasi konsoldan verilen **java isim** komutuyla gerçeklestirilmektedir.

Simdi de applet grafik ortaminda çikti almaya örnek verelim. Ilk applet programimiz **HosgeldinizApplet.java**

listesi Program 1.6 da verilmistir

### Program 1.6 HosgeldinizApplet.java *programi*

**import java.applet.Applet; import java.awt.\*;**

**public class HosgeldinizApplet extends Applet**

**{**

**public void paint(Graphics g)**

**{**

**g.drawString("Java sinifina hos geldiniz!",25,35);**

**}**

**}**

bu programi çalistirmak için önce HosgeldinizApplet.html HTML dili programini yazmamiz gerekir

### Program 1.7 HosgeldinizApplet.html *HTML kodu*

<html >

<applet code="HosgeldinizApplet.class" width=200 height=50 >

</applet >

</html>

bu html kodunda

### <applet code="HosgeldinizApplet.class" width=200 height=50 >

komutu html'e compile edilmis olan **HosgeldinizApplet.class** dosyasinin gösterilmesini istemektedir. Süphesiz ayni komut çok daha kompleks bir html dosyasinin içinde de ayni görevi görmek amaciyla kullanilabilirdi.

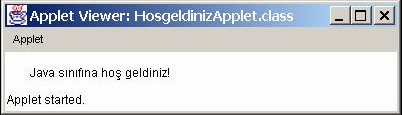
Programi çalistirmak için

### javac HosgeldinizApplet.java appletviewer HosgeldinizApplet.html

komutlarini konsolda vermemiz yeterlidir. Elbette HosgeldinizApplet.html dosyasinin yukarida belirtildigi gibi hazirlanmis olmasi da gerekir. Fakat Konsol programlarindan farkli olarak sonucu konsol yerine Applette (browser programinda) görürüz. Ayni sonucu **appletviewer** deyimini kullanmadan herhangi bir browser

programi çagirarak da elde edebiliriz. Browser programinda sonucu görebilmek için önce programi açmamiz sonra da HosgeldinizApplet.html dosyasini bularak açmamiz gerekir. Çikti görüntüsü asagidaki gibi olacaktir.

### 01004.JPG



**Sekil 1.4 HosgeldinizApplet.html dosyasinin appletviewer programi kullanarak görülmesi**

Bu programinda temel olarak bazi farkliliklar disinda bir önceki programlara benzedigini gözlemleyebiliriz. Bu programin temel farki internet ortamindan herhangi bir bilgisayardan da (gerekli ayarlama ve kopyelemelerden sonra) görülebilir olmasidir.

### import java.applet.Applet; import java.awt.\*;

deyimleri java kütüphanesindeki Applet ve pencere(awt) sinif grafik programlarini programimiza aktarir.

**public class HosgeldinizApplet extends Applet** deyimi HosgeldinizApplet isimli sinifi (class) yaratir. Bu sinif tüm Applet sinifinin bir devami olarak tanimlanmaktadir (**extends Applet**) diger bir deyimle Applet sinifinin bütün metotlari extends deyimiyle HosgeldinizApplet sinifimizin içine kopyalanmistir.

**public void paint(Graphics g)** deyimi appletin grafik olarak çizilmesini saglar. Bu Metot **Graphics** sinifi **g**

nesnesini girdi olarak kullanir. Metotun çiktisi yoktur.

**g.drawString("Java sinifina hos geldiniz!",25,35);** Bu terim Appletin x = 25, y=35 noktasina **Java sinifina hos geldiniz!** Yazisini çizer. DrawString Grafik sinifi g nesnesine ait bir metotdur.

Bu programda temel grafik kütüphanesi olarak abstract window toolkit (awt) kullanilmistir. ayrica konsol programlarindan farkli olarak bu programda main metodunun kullanilmadigina da dikkat ediniz. bir program hem applet hem de konsol olarak da olusturulabilir, yani hem main metodu olabilir hem de applet (html) ortaminda çalisabilir.

Ayni appleti oldukça benzer bir sekilde SWING kütüphanesini kullanarak da olusturabilirdik.

### Program 1.8 HosgeldinizSWA.java *programi*

import javax.swing.Japplet; import java.awt.Graphics;

public class HosgeldinizSWA extends JApplet

{

public void paint(Graphics g)

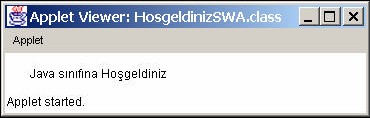
{

g.drawString("Java sinifina Hosgeldiniz",25,25);

}

}

### 01005.JPG



**Sekil 1.5 HosgeldinizSWA.java dosyasinin applet SWING pop-up grafik çiktisi kullanarak görülmesi**

Bu programdaki

### import javax.swing.Japplet; import java.awt.Graphics;

deyimleri java swing kütüphanesindeki JApplet ve pencere(awt) sinifindaki Graphics alt programlarini (siniflarini) programimiza aktarir.

### public class HosgeldinizSWA extends JApplet

deyimi HosgeldinizAppletSWA isimli sinifi (class) yaratir. Bu sinif swing sinifindaki JApplet sinifinin bir devami olarak tanimlanmaktadir (**extends JApplet**). Diger bir deyimle JApplet sinifinin bütün metotlari extends deyimiyle HosgeldinizAppletSWA sinifimizin içine kopyalanmistir.

Ayni isi yapmak için bir çok degisik program yazdik. Bazilarimiz buna ne gerek vardi, tek bir sekilde bu isi halledemezmi idik diyebilirler. Haklidirlarda. Fakat modern bilgisayar dillerinde kullanicilari da göz önüne almak durumundayiz. Ayni isi daha renkli ve çesitli sekillerde yapmamiz temel olarak program kullanicilarina daha çesitli alternatifler sunabilir. Ayni zamanda diger programlama alternatiflerine göre daha üstün bir çalisma ortami yaratabilir.

Simdi de programimiza bir basamak daha ekleyelim ve program kullanicilara isimlerini sorsun, yani ekrandan bir veri girelim. Bunu yapmak için ilk tanimlayacagimiz program konsol ortaminda çalisan **Hosgeldiniz2.java** programi olacaktir. Bu programin çiktisi dos konsol ortaminda sekil 1.6 da görülmektedir. Program ekranda isminizi giriniz yazacak, ve ekranda isim girildikten sonra

### java sinifina hosgeldiniz! isim

çiktisini verecektir.

**Program 1.9 Hosgeldiniz2.java *programi***

import java.io.\*; class Hosgeldiniz2 {

public static void main (String args[]) throws IOException

{

BufferedReader cin=new BufferedReader(new InputStreamReader(System.in)); System.out.println("isminizi giriniz : ");

String isim=new String(cin.readLine()); System.out.println("Java Sinifina Hos Geldiniz! "+isim); cin.close();

}

}

=============================

C:\co\java\prog>java Hosgeldiniz2 isminizi giriniz :

Turhan

Java Sinifina Hos Geldiniz! Turhan C:\co\java\prog>

=============================

### Sekil 1.6 Hosgeldiniz2.java dosyasinin sonuçlarinin dos çiktisi kullanarak görülmesi

Bu programda okuma islemini yapmakiçin önce

BufferedReader cin=new BufferedReader(new InputStreamReader(System.in)); Deyimini kullanarak bir okuma kanali açtik, sonra

System.out.println("isminizi giriniz : "); String isim=new String(cin.readLine());

Deyimindeki **cin.readLine()** komutuyla ismi cin kanalindan okuyarak isim degiskenine aktardik.

Ayni islemi yapan fakat girdi çikti sinif tanimlarinda Ek 1 de verilen Text sinifini kullanan Hosgeldiniz3.java programi sekil 1.10 da verilmistir.

**Program 1.**10 Hosgeldiniz3.java ***programi***

import java.io.\*; import Text;

class Hosgeldiniz3 {

public static void main (String args[]) throws IOException

{

Text cin=new Text(); System.out.print("isminizi giriniz : "); String isim=cin.readString();

System.out.println("Java Sinifina Hos Geldiniz! "+isim);

}

}

Hosgeldiniz2.java programinin çiktisi da Hosgeldiniz2.java programiyla ayni olacaktir.

Simdi de swing kütüphanesindeki JoptionPane.showInputDialog() kullanarak girdi penceresi açan ve çikti penceresine JoptionPane.showMessageDialog() kullanarak mesaji yazan Hosgeldiniz1SW.java programini görecegiz.

### Program 1.11 Hosgeldiniz1SW.java *programi*

import javax.swing.JoptionPane; class Hosgeldiniz1SW {

public static void main (String args[])

{

String isim=JoptionPane.showInputDialog("Lutfen isminizi giriniz :"); JoptionPane.showMessageDialog(null,"Java Sinifina Hos Geldiniz! "+isim, "Hosgeldiniz1SW programi",JOptionPane.INFORMATION\_MESSAGE); System.exit(0);

}

}

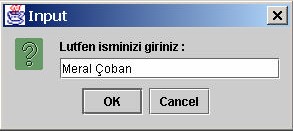
Bu programi çalistirmak içinde javac Hosgeldiniz1SW.java

komutuyla bu programi derlememiz ve

java Hosgeldiniz1SW

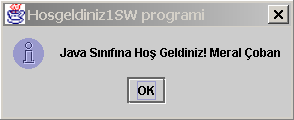
komutuyla çalistirmamiz gerekecektir. Program girdi ve çikti pencereleri sekil 1.7 ve Sekil 1.8 de verilmistir.

### 01007.JPG



**Sekil 1.7 Hosgeldiniz1SW.java dosyasinin pencere popup girdi penceresi**

**01008.JPG**



**Sekil 1.8 Hosgeldiniz1SW.java dosyasinin pencere popup çikti penceresi**

Simdi ayni girdi ve çikti programlamasini JApplet sinifini kullanarak yapalim. Input (girdi) için yine JoptionPane sinifindan yararlanacagiz.

**Program 1.12** Hosgeldiniz1SWA.java ***programi***

import javax.swing.\*; import java.awt.Graphics;

public class Hosgeldiniz1SWA extends JApplet

{

String isim; public void init()

{

isim=JoptionPane.showInputDialog("isminizi giriniz : ");

}

public void paint(Graphics g)

{

g.drawString("Java sinifina Hosgeldiniz "+isim,25,25);

}

}

Bu programi çalistirmak için compile prosesinde yine

**javac Hosgeldiniz1SWA.java** komutunu kullanacagiz. sonra Hosgeldiniz1SWA.html dosyasini olusturacagiz.

### Program 1.13 Hosgeldiniz1SWA.html *programi*

<html>

<applet code="Hosgeldiniz1SWA.class" width=300 height=30>

</applet>

</html>

Hosgeldiniz1SWA.html dosyasini herhangi bir browser programinda görebilecegimiz gibi javanin kendi basit browser programi appletviewer da da

### appletviewer Hosgeldiniz1SWA.html

deyimiyle gösterebiliriz. Program girdi ve çiktisi :

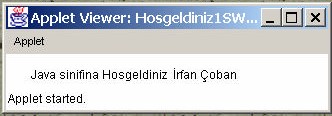
### Resim 01009.JPG



**Sekil 1.9 Hosgeldiniz1SWA.java dosyasinin applet popup girdi penceresi**

Program çiktisi appletviewer browserinda görülecegi sekliyle :

### Resim 01010.JPG



**Sekil 1.10 Hosgeldiniz1SWA.java dosyasinin applet çikti penceresi**

olacaktir. Son olarak girdi ve çikti islemlerini ayni pencerede yapildigi biraz daha kompleks olan Hosgeldiniz5SWA.java programina göz atalim.

### Program 1.13 Hosgeldiniz5SWA.java *programi*

import javax.swing.\*; // java JApplet sinifini çagir

import java.awt.\*; // java pencere kullanma siniflarini (awt kütüphanesi) çagir

import java.awt.event.\*; // java pencereyi dinleme siniflarini (awt kütüphanesi) çagir

public class Hosgeldiniz5SWA extends JApplet implements ActionListener

{

//sinif degiskenleri

JTextArea ciktiAlani; //JTextArea sinifi çikti alani degiskeni JLabel kutubasligi; //JLabel sinifi degiskeni (nesnesi) kutubasligi

JTextField kutugirdisi;//JTextfield sinifi degiskeni (nesnesi) kutugirdisi String isim;

public void init()

{

//girdi çikti programlarinin içine yazilacagi ana pencereyi tanimla Container c=getContentPane();

c.setLayout(new FlowLayout()); isim="Mürside Gökagaç";

kutubasligi=new JLabel("isminizi giriniz :"); c.add(kutubasligi);//kutubasligini pencereye yaz kutugirdisi=new JTextField(10); c.add(kutugirdisi);//kutuyu pencereye yerlestir ciktiAlani=new JTextArea();

Color co=c.getBackground(); ciktiAlani.setBackground(co); c.add(ciktiAlani);

ciktiAlani.setText("Java sinifina Hosgeldiniz ! "+ isim);

// kutuya yeni ilave edilecek komutlari bekle

// her yeni komutta actionPerformed metodunu calistir. kutugirdisi.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

//pencereden ismi oku isim=e.getActionCommand();

ciktiAlani.setText("Java sinifina hosgeldiniz ! "+ isim); repaint();

}

}

bu programi da yine

javac Hosgeldiniz5SWA.java deyimiyle compile edip,

appletviewer Hosgeldiniz5SWA.html komutuyla çalistiririz. Hosgeldiniz5SWA.html dosyasinin yukarida anlatildigi gibi hazirlanmasi gerekir. Program çiktisi sekil 1.11 de gösterilmistir.

### Resim 01011.JPG



**Sekil 1.11 Hosgeldiniz5SWA.java dosyasinin applet çikti penceresi**

Bu programda verilen komut detaylarini su anda bilmek zorunda degilsiniz. Benzer islemleri yapmak için ayni formattan yararlanabilirsiniz. Ilerki bölümlerde grafik programlarinin alt metodlari detayli olarak incelenecektir.

### JAVA PROGRAMLAMA DILI TEMEL DEGISKEN TÜRLERI

Programlama dillerinde rakamlar bilgisayar belleginin temel depolama birimlerine yazilirlar. Temel bilgisayar bellek birimi bit olarak adlandirilir. Bir bilgisayar bellegindeki tek bir transistörden olusmustur. Bu transistörden akim geçiyorsa tansistörün veya bitin bellek degeri 1 (bir) veya true (dogru) olarak alinir. Eger akim geçmiyor veya düsük düzeyde bir akim geçiyorsa transistörün bellek degeri 0 (sifir) veya false(yanlis) olarak alinir.

Bilgisayar bit birimleri bir araya gelerek bilgisayar temel degisken türlerini yazabilecegimiz bir sistem olustururlar. Degisken türü bilgisayar tarafindan bilinmelidir, çünkü ayni bit toplulugu bir harfi simgeleyebilecegi gibi bir rakami da simgeleyebilir. Java dilinde Tablo 1.1 de görülen temel degisken türleri mevcuttur.

### Tablo 1.1 Java Temel Degisken türleri

|  |  |  |  |
| --- | --- | --- | --- |
| **Degisken Türü** | **Türkçe karsiligi** | **Bit büyüklügü** | **Sinir Degerleri** |
| boolean | Mantik degiskeni | 1 | true(dogru) , false(yanlis) |
| char | harf degiskeni | 16 | '\u0000' den '\uFFFF' |
| byte | tam sayi degiskeni | 8 | -128 den 127 e |
| short | tam sayi degiskeni | 16 | -32768 den 32767 e |
| int | tam sayi degiskeni | 32 | -2157483648 den 2147483647 e |
| long | tam sayi degiskeni | 64 | -9223372036854775808 den  9223372036854775808 e |
| float | Gerçek sayi degiskeni | 32 | -3.40292347e+38 den 3.40292347e+38 e |
| double | Gerçek sayi degiskeni | 64 | -1.7976931348623157e+308 den 1.7976931348623157e+308 e |

Simdi bu temel degisken türlerinin bazilarini daha detayli inceleyelim:

### boolean degisken türü

boolean degisken türü mantik islemlerinde kullanilir. Sadece true veya false degerleri alir. true dogru false yanlis anlami tasir. Mantik degiskenlerine dogrudan true veya false degerleri yüklenebilecegi gibi diger degiskenleri mantik islemlerini kullanarak karsilastirarakta degerleri programlarin içinde hesaplanabilir.

Asagidaki küçük program parçasi mantik degiskeni ilk mantik degiskenini tanimlamakta ve dogru degerini bu degiskene yüklemektedir.

### boolean ilkmantikdegiskeni; ilkmantikdegiskeni=true;

### char (harf) degisken türü

char degisken türü harflerin tanimlanmasinda kullanilir. Harfler java dilinde ISO Unicode kodu ile bilgisayara aktarilir. Unicode 4 hekzagonal (16 tabanli) sayinin bir araya gelmesiyle olusur. Hekzagonal sayi sisteminin onlu ve ikili sayi sistemiyle esitligi söyledir :

### Tablo 1.2 Haksagonal(onaltili), onlu ve ikili sayi sistemleri esitlikleri

|  |  |  |
| --- | --- | --- |
| **Heksagonal** | **Onlu** | **Ikili** |
| **0** | **0** | **0000** |
| **1** | **1** | **0001** |
| **2** | **2** | **0010** |
| **3** | **3** | **0011** |
| **4** | **4** | **0100** |
| **5** | **5** | **0101** |
| **6** | **6** | **0110** |
| **7** | **7** | **0111** |
| **8** | **8** | **1000** |
| **9** | **9** | **1001** |
| **A** | **10** | **1010** |
| **B** | **11** | **1011** |
| **C** | **12** | **1100** |
| **D** | **13** | **1101** |
| **E** | **14** | **1110** |
| **F** | **15** | **1111** |

ISO Unicode da tanimlanan '\u0041' kodu '**A**' harfi anlamina gelir. Veya '\u03E1' kodu '**** ' harfini tanimlar. ISO Unicode karakter setinin ilk iki rakami sifir oldugunda ASCII karakter kodunu tanimlar. Unicode hakkinda daha ayrintili bilgi için

### http:\\unicode.org

internet adresini ziyaret edebilirsiniz. Asagidaki küçük program parçacigi **char** tipi A1, A2 ve alpha1,alpha2 degiskenlerine A ve **** harflerini yüklemektedir.

### char A1,A2;

**char alpha1,alpha2;**

**A1='\u0041';**

**A2='A';**

**alpha1='\u03E1'; alpha2=' ';**

Burada sunu kaydedelim alpha1 degiskenini her ortamda dogru olarak görebiliriz, fakat alpha2 degiskeni bazi ortamlarda dogru olarak görülmeyebilir.

### Tam sayi degisken türleri (byte, short,int,long)

Tam sayi degiskenler hafizada isgal ettikleri yere göre byte(8 bit), short(16 bit), int(32 bit) ve long(64 bit) adini alirlar. Bir bitlik hafizaya sadece iki rakamin (0 veya 1) yazilabilecegi göz önüne alinirsa örnegin sekiz bitlik byte türü tamsayi degiskenine 256 sayi (ikili sayi esiti 11111111) yazilabilecegi ortaya çikar. Bitlerden biri + veya - isareti için kullanildigindan byte degiskeninin sinir degerleri -128den 127 e kadardir (toplam 256 sayi). Eger bir tamsayi degiskenin sadece arti degerlerini kullanmak istersek unsigned terimini kullaniriz. Örnegin **unsigned byte** tipi tamsayi degiskenin sinir degerleri 0 dan 256 ya kadardir. Tamsayi degisken türleri içinde en fazla kullanilan int türüdür. Asagidaki küçük program parçaciginda int türü **ilktamsayi** degiskenine 32 rakami yüklenecektir.

### int ilktamsayi; ilktamsayi=32;

bu iki satirlik program parçacigi tek bir satir olarak

### int ilktamsayi=32;

seklinde de yazilabilirdi.

Javadaki tamsayi degisken türleri + ve – degeri alabilen türlerdir. Örnegin byte degisken türü –128 den +127 e kadar toplam 256 sayi kullanabilir. Eger – bölgeyi kullanmayacaksak tamsayi degiskenlerin kullanim bölgesini **unsigned** deyimini kullanarak tamamen arti bölgeye çekebiliriz. Örnegin

### unsigned byte artibolgetamsayisi;

islemiyle tanimlanan **artibolgetamsayisi** degiskeni 0 ile 256 arasinda degerler alabilir.

### Gerçek sayi degisken türleri (float, double,long double)

Gerçek sayi sistemleri de 0 ve 1 bitleri kullanilarak olusturulabilir. Gerçek sayilarin tamsayidan küçük kisimlari 2 li tabanda eksi üstler kabul edilerek olusturulur. Örnek olarak 16 bitlik gerçek bir sayiyi ikili sistemdeki bilgisayar belleginde su sekilde gösterebiliriz :

### Tablo 1.3 gerçek sayi 22.625e-17 yazilmasinda kullanilan onlu ve ikili sayilarin bellekte gerçek sayi organizasyonu örnegi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **+/-** | **24** | **23** | **22** | **21** | **20** | **2-1** | **2-2** | **2-3** | **2-4** | **+\-** | **24** | **23** | **22** | **21** | **20** |
| **+** | **16** | **8** | **4** | **2** | **1** | **0.5** | **0.25** | **0.125** | **0.0625** | **-** | **16** | **8** | **4** | **2** | **1** |
| **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **1** |

Yukarida 16 bitle temsil edilen sayi : +16+4+2+0.5+0.125)\*10-(16+1) = 22.625e-17 rakamidir. Gerçek sayi degiskende yeterli hassasiyeti saglayabilmek için genelde 64 bit uzunlugundaki **double** degisken türü kullanilir. Javadaki Matematik kütüphaneleri de double degisken türü için tanimlanmistir. Asagidaki küçük program parçaciginda **double** türü **ilkgercekdegisken** degiskenine 22.625e-7 sayisini yüklüyoruz.

### double ilkgerçekdegisken; ilkgerceldegisken=22.625e-17;

eger 64 bit double sayi hassasligi yetmezse **long double** kullanilabilir. Bu degisken türü 128 bit boyutundadir.

### JAVA NESNESI OLARAK TANIMLANMIS TEMEL DEGISKENLER

Java nesne temelli bir dildir. Javadaki tüm degisken türlerinin nesne kökenli esdegerleri mevcuttur. Bunun yaninda temel degisken türleri arasinda yer almiyan String gibi sadece nesne kökenli olarak tanimlanmis degisken türleri de mevcuttur. Simdi bunlarin bir kismini biraz daha detayli inceleyelim.

### String nesne tipi degiskeni

String degiskeni yazi yazdirma isleri için kullanilir. Nesne türü degiskendir. Java dilinde char degisken türü kullanilarak tanimlanmis bir nesne tipi degiskendir. Asagidaki küçük program parçaciginda bu degisken türünün kullanilmasini görüyoruz.

### String a="ali";

**String c="veli";**

**String d;**

**d=a+b; // d nin degeri "ali veli" System.out.println(d); // bu satir ali veli çiktisi verir**

**String türü** sabitler her zaman " isaretleri arasina yazilirlar ve + isaretiyle bir araya getirilebilirler. String türü degiskenleri tanimlamak için ayni zamanda

### String a=new String("ali");

**String c=new String("veli");**

**String d=new String();**

**d=a+b; // d nin degeri "ali veli" System.out.println(d); // bu satir ali veli çiktisi verir**

Formunu da kullanabiliriz. Ikinci verdigimiz form islem olarak ilk verilenin tamamen aynidir. String türü bir degiskenin toplam boyutu length() alt metoduyla belirlenebilir. Örnegin

### String a=new String("ali");

**int x=a.length();**

x degiskenine 3 degerini yükleyecektir.

Stringin alt metodu equals() karsilastirma yapar. Örnegin

### String a=new String("ali"); boolean y=a.equals(“ali”); boolean z=a.equals(“veli”);

y boolean degiskenine true degerini, z boolean degiskenine false degerini yükleyecektir.

### Integer nesne tipi degiskeni

Integer degiskeni, string degiskeni gibi

### Integer i;

**i=new Integer(3);**

veya

### Integer i=new Integer(3);

Seklinde tanimlanabilir. Integer tanimi String degiskeni üzerinden de tanimlanabilir. Örnegin

### String s=”15”;

**Integer i=new Integer(s);**

Tanimi 15 degerini yükler. Integer degerini int degerine (temel degisken) çevirebiliriz.

### int x;

**Integer y=new Integer(3); x=Integer.IntegerValue(y);**

String degerini int degerine direkt olarak çevirmek içinse

### String s=”15”;

**Int x=Integer.parseInt(s);**

kullanilir.

int tipi (temel) degiskeni String degiskenine degistirmek için

### int x=3;

**String s=Integer.toString(x);**

kullanilir.

### Double nesne tipi degiskeni

Double degiskeni kullanim olarak Integer degiskeninden bir farki yoktur. Ayni tanimlamalar Integer yerine Double kullanilarak yapilabilir. Degiskenler

### Double x;

**X=new Double(3.66e5);**

veya

### Double x=new Double(3.66e5);

Seklinde tanimlanabilir. Double tanimi String degiskeni üzerinden de tanimlanabilir. Örnegin

### String s=”15.66e-3”; Double x=new Double(s);

Tanimi 15.66\*10-3 degerini yükler. Double degerini double degerine (temel degisken) çevirebiliriz.

### double x;

**Double y=new Double(15.66e-3); X=Double.doubleValue(y);**

**int x;**

**Double y=new Double(15.66e3); x=Double.intValue(y);**

String degerini double (temel degisken) degerine direkt olarak çevirmek içinse

### String s=”15.25”;

**double x=Double.parseDouble(s);**

Islemi kullanilabilir.

double tipi (temel) degiskeni String degiskenine degistirmek için

### double x=3.75;

**String s=Double.toString(x);**

kullanilir.

### diger nesne temelli degisken türleri

Object, Long, Float, Boolean, Character, Vector gibi diger nesne tipi degisken türleri de mevcuttur. Object nesne tipi tüm bu degisken nesne tiplerini içinde barindiran genel bir tiptir. Tüm nesne tipi degiskenlerin bizim için temel avantaji alt metodlarini kullanarak islemler gerçeklestime olasiliklaridir. Fakat bu tür degiskenler temel degisken türlerine göre daha çok hafiza yeri isgal ettiklerinden mecbur kalinmadikça da kullanilmamalidirlar.

Ayrica Yine nesne tipi BigDecimal ve BigInteger türleri de hassasiyeti kullanici tarafindan belirlenen nesne tipi degiskenler olarak kullanilabilirler.

### FINAL TERIMI VE SABITLER

Java dilinde degisken yerine sabit kullanmak istersek tanimin basina **final** sözcügünü getiririz. Final olarak tanimlanan sabitlerin bir kere degerleri verildikten sonra degistirilemez.

Örnegin

### final double pi=3.14159;

pi sabitini tanimlar.

### ARITMETIK ISLEMLER

**Javadaki temel aritmetik islemler sunlardir :**

+ toplama

- çikarma

\* çarpma

/ bölme

% tamsayilar için bölmede kalan (modül)

Program 1.15 deki program örnegi int tamsayi kullanarak bu aritmetik islemlerin yapilmasini göstermektedir.

### Program 1.15 : Aritmetik.java konsol programi

**import java.io.\*; //java girdi cikti sinifini cagir class Aritmetik**

**{**

**public static void main(String args[])**

**{**

**int sayi1,sayi2;**

**int toplama,cikarma,carpma,bolme,kalan; sayi1=2;**

**sayi2=3;**

**toplama = sayi1+sayi12; cikarma = sayi1-sayi2; carpma = sayi1\*sayi2;**

**bolme=sayi1/sayi2; kalan=sayi1%sayi2;**

**System.out.println("sayi1 + sayi2 ="+toplama); System.out.println("sayi1 - sayi2 ="+cikarma); System.out.println("sayi1 \* sayi2 ="+carpma); System.out.println("sayi1 / sayi2 ="+bolme); System.out.println("sayi1 % sayi2 ="+kalan);**

**}**

**}**

Aritmetik.java konsol programi çiktisi : sayi1 + sayi2 =5

sayi1 - sayi2 =-1

sayi1 \* sayi2 =6 sayi1 / sayi2 =0 sayi1 % sayi2 =2

seklinde olacaktir. Swing grafik kütüphanesi çiktisi pratigi yapmak amaciyla ayni programi swing JoptionPane çiktisi ile yazarsak :

### Program 1.16 : AritmetikSW.java konsol programi

import javax.swing.JOptionPane; //java swing cikti sinifini cagir

class AritmetikSW

{

public static void main(String args[])

{

String s;

int sayi1,sayi2;

int toplama,cikarma,carpma,bolme,kalan; sayi1=Integer.parseInt( JOptionPane.showInputDialog("birinci sayiyi giriniz : ")

);

sayi2=Integer.parseInt( JOptionPane.showInputDialog("ikinci sayiyi giriniz : ")

);

toplama = sayi1+sayi2; cikarma = sayi1-sayi2; carpma = sayi1\*sayi2; bolme = sayi1/sayi2; kalan = sayi1%sayi2; s="";

s=s+sayi1+" + "+sayi2+" = "+toplama+"\n"; s=s+sayi1+" - "+sayi2+" = "+cikarma+"\n"; s=s+sayi1+" / "+sayi2+" = "+carpma +"\n"; s=s+sayi1+" \* "+sayi2+" = "+bolme +"\n"; s=s+sayi1+" % "+sayi2+" = "+kalan +"\n"; JOptionPane.showMessageDialog(null,s,

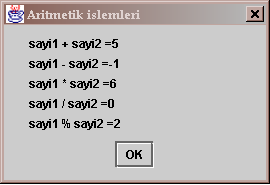
"Aritmetik islemleri",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Bu programin çiktisi :

### 01012.JPG



**Sekil 1.12 AritmetikSW.java dosyasinin JOptionPane çikti penceresinde görünümü**

Java ve C dillerinde temel aritmetik islemlerinin yani sira bu islemlerin kisaltilmis sembolleri de oldukça sik kullanilir. Bu islemler :

**++** bir ilave et

**--** bir çikar

+= sag taraftaki rakami ilave et

**-=** sag taraftaki rakami çikar

**\*=** sag taraftaki rakamla çarp

**/=** sag taraftaki rakama böl

%= sag taraftaki rakama bölüp kalanini al islemleridir.

**++** ve **--** islemleri degiskenden önce veya sonra gelebilir. Bu iki seklin degisken tek baslarina kullanildiginda bir farkliliklari yoktur, fakat ikinci bir degiskenle birlikte tek bir deyimde kullanildiklarinda ikinci degiskenin degeri degisir. Asagidaki iki program parçacigina göz atalim.

***++ islemcisinin degiskenden sonra kullanilmasi***

**int a=2; int c;**

**c=a++; // a nin degeri 3 c nin degeri 2**

***++ islemcisinin degiskenden önce kullanilmasi***

**int a=2; int c;**

**c=++a; // a nin degeri 3 c nin degeri 3**

a degiskeninin degeri baslangiçta 2 ise görüldügü gibi hem ++a isleminde hem de a++ isleminde degeri 3 e çikmaktadir. Ancak c degiskeninin degeri c=a++ isleminde 2 olarak kalirken (önce degeri yükle sonra arttir), c=++a isleminde 3 olmaktadir(önce arttir, sonra degeri yükle). ++ ve -- islemleri genelde tam sayi degiskenler için kullanilir.

Program 1.17 da aritmetik islemleri gösteren bir program verilmistir. Bu programin sonuçlari da altta verilmektedir. Programi basamak basamak takip ederek sonuçlarin dogrulugunu elde hesaplayarak kontrol edebilirsiniz.

**Program 1.17 : Aritmetik1.java *aritmetik islem örnegi***

### import java.io.\*; public class Aritmetik1

**{ public static void main(String[] args)**

**{**

**int sayi1=1; int sayi2=2; sayi1+=sayi2;**

**System.out.println("sayi1+= : "+ sayi1); sayi1-=sayi2;**

**System.out.println("sayi1-= : "+ sayi1); sayi1\*=sayi2; System.out.println("sayi1\*= : "+ sayi1); sayi1/=sayi2; System.out.println("sayi1/= : "+ sayi1);**

**}**

**}**

***Aritmetik1 sinifinin sonuçlari***

**sayi1+= : 3**

**sayi1-= : 1**

**sayi1\*= : 2**

**sayi1/= : 1**

### JAVADA MANTIKSAL ISLEMLER

Javadaki temel degisken türlerine bakarken boolean türü degiskenleri tanimlamistik. Tam sayi ve gerçek degiskenlerin aritmetik islemlerde kullanilabildigi gibi boolean degiskenler de mantiksal islemlerde kullanilabilirler. Mantiksal islem islemcileri sunlardir :

### Tablo 1.3 Java dilindeki mantiksal islemler ve anlamlari

|  |  |
| --- | --- |
| **Mantiksal Islemci** | **Anlami** |
| **&&** | **ve** |
| || | **veya** |
| > | **büyüktür** |
| **<** | **küçüktür** |
| **==** | **esittir** |
| **>=** | **büyük ve esittir** |
| **<=** | **küçük ve esittir** |
| **!** | **degildir** |
| **!=** | **esit degildir** |

Mantiksal islemleri pekistirmek amaciyla Program 1.18 de bir örnek verilmistir.

**Program 1.18 : *Java mantiksal islemleri örnek problemi* booleanvar.java**

import java.io.\*;

public class booleanvar

{ public static void main(String[] args)

{

boolean b1,b2; b1=true; b2=false;

System.out.println("b1 : "+ b1+" b2 : "+b2); System.out.println("b1 && b2 "+ (b1&&b2) ); System.out.println("b1 || b2 "+ (b1||b2) );

int sayi1,sayi2; sayi1=1; sayi2=2;

System.out.println("sayi 1 : "+sayi1); System.out.println("sayi 2 : "+sayi2); b1=(sayi1 > sayi2);

System.out.println(sayi1+" > "+sayi2+" "+ b1 ); b1=(sayi1 < sayi2);

System.out.println(sayi1+" < "+sayi2+" "+ b1 ); b1=(sayi1 == sayi2);

System.out.println(sayi1+" == "+sayi2+" "+ b1 ); b1=(sayi1 >= sayi2);

System.out.println(sayi1+" >= "+sayi2+" "+ b1 ); b1=(sayi1 <= sayi2);

System.out.println(sayi1+" <= "+sayi2+" "+ b1 ); b1=(sayi1 != sayi2);

System.out.println(sayi1+" != "+sayi2+" "+ b1 );

}

}

Asagida bu örnek problemin (Program 1.5) sonucunu verilmektedir. Satir satir programi takip ederek sonuculari kontrol ediniz.

b1 : true b2 : false b1 && b2 false b1 || b2 true

sayi 1 : 1

sayi 2 : 2

1 > 2 false

1 < 2 true

1 == 2 false

1 >= 2 false

1 <= 2 true

1 != 2 true

### JAVA DILINDE BIT KOMUTLARI VE ISLEMLERI

Bit islemi dogrudan dogruya bitler üzerinde yapilan islemlerdir. Bit islemleri genellikle veri trasferi (bilgisayardan bilgisayara veya çesitli kontrol sistemlerinde) islemlerinde kullanilir. Javadaki bit islemleri Tablo

1.4 de verilmistir.

### Tablo 1.4 Java dilindeki bit islemleri ve anlamlari

|  |  |
| --- | --- |
| **Bit Islemci** | **Anlami** |
| **&** | **AND ve** |
| | | **OR veya** |
| ^ | **XOR** |
| **~** | **Ters bit (NOT)** |
| **>>** | **Bir bit saga kay** |
| **<<** | **Bir bit sola kay** |

Bit isleminde ne olduguna bir örnekle bakalim. int x=7;

x=x<<1;

bit isleminde x degiskeninin bit yapisi sola dogru bir bit kaydirilmistir. 7 degerinin bit esdegeri 00000000000000000000000000000111

seklindedir. Sola dogru bir bit kaydigimizda 00000000000000000000000000001110

bit yapisi olusacaktir. Bu yapinin int karsiligi 14 dür.

int x=7; x=x<<1;

x=~x; islemi

11111111111111111111111111110001

degerini verecektir.

Küçük bir programda bit islemlerinin bazilarini kullanalim.

**Program 1.19 : *Java bit islemleri örnek problemi* bitislemi.java**

class bitislemi{

public static void main (String args[])

{

int x=7; System.out.println("x = "+x); x= x<<1;

System.out.println("x = "+x);

x= x<<1;

System.out.println("x = "+x); x= x>>2;

System.out.println("x = "+x); x= ~x;

System.out.println("x = "+x);

}

}

bu programin sonucu x = 7

x = 14

x = 28

x = 7

x = -8

seklinde olacaktir. Çikan sonuçlarin bit esdegerlerini kontrol ediniz. Bit islemleri aslinda bilgisayarin ve digital elektronik devrelerin temel islemleridir. Bu islemler assembler dillerinin de temel islemlerini teskil ederler.

### JAVA DILININ TEMEL KOMUTLARI

Bilgisayar dillerinde degiskenlerden ve islemlerden sonraki en temel yapi temel dil komutlaridir. Simdi bu komut yapilarina tek tek göz atalim.

### if - elseif - else mantiksal karsilastirma yapisi

Bu yapi programda karar mekanizmalarini olusturmak için kullanilir. Temel yapisi :

### if( boolean degisken yada sabit)

**{ boolean degisken veya sabitin degeri true oldugunda yapilan islemler} else if( boolean degisken yada sabit)**

**{ boolean degisken veya sabitin degeri true oldugunda yapilan islemler}**

**else**

**{ yukardaki boolean degisken vede sabitlerin hiçbiri true degerine esit degilse yapilacak olan**

**islemler}**

else if yapisi probleme göre istenilen sayida olabilir, if ve else yapisi sadece bir tane olarak kullanilir. if yapisi java ve C dillerinde kisaltilmis bir sekilde de yazilabilir bu form

### degisken adi = ( boolean degisken yada sabit ? boolean true ise degisken degeri : boolean false ise degisken degeri );

seklindedir. Küçük bir örnek programla if-elseif-else yapisinin çalismasini inceleyebiliriz.

**Program 1.20 : *if yapisini gösteren* ifyapisi.java *programi***

### import java.io.\*; // giris çikis import Text; // okuma sinifi class ifyapisi

**{**

**public static void main (String args[]) throws IOException**

**{**

**double not;**

**//**

**Text cin=new Text(); not=cin.readDouble();**

**if( not ==90)**

**{ System.out.println("A"); } else if(not ==75)**

**{ System.out.println("B"); }**

**else if(not ==60)**

**{ System.out.println("C"); } else if(not ==50)**

**{ System.out.println("D"); } else if(not ==40)**

**{ System.out.println("E"); } else**

**{ System.out.println("F"); }**

**}**

**}**

***If yapisi programinin çiktisi :***

**Ogrencinin notunu giriniz : 53**

**D**

Bu programda ögrenci notu ekrandan girdi olarak girilmektedir. Bunu saglayabilmek için önce

### Text cin=new Text();

komutunu kullanarak ekrandan girdi girme kanali olan cin degiskeni tanimlanmakta, sonra bu kanal üzerinden

### not=cin.readDouble();

deyimiyle ekrandan girilen deger not degiskenine aktarilmaktadir. Okuma islemi normalde daha detayli tanimlar gerektirir. Bu tanimlar simdilik Text sinifinin arkasinda verilmistir. Text sinifinin detayli Yeri geldiginde okuma islemi daha detayli olarak incelenecektir.

not degiskeni bir ekran girdisi oldugundan girilen nota göre bilgisayar A,B,C,D,E veya F notlarindan birini verecektir. Bu ilk programimizda if ve else if parantezlerinin içinde direk olarak boolean sabit kullanilmistir. Parantez içi boolean degisken de olabilir.

Ayni program kodunu java swing kütüphanesi JoptionPane alt sinifini kullanarak yazacak olursak :

**Program 1.21 : *if yapisini gösteren* ifyapisiSW.java *programi***

import javax.swing.JOptionPane; class ifyapisiSW

{

public static void main (String args[])

{

double not;

//JOptionPane her zaman String turu çikti verir. Bunu Double.parseDouble()

//kullanarak double degerine çevirebiliriz. not=Double.parseDouble(JOptionPane.showInputDialog("Ogrencinin notunu giriniz (0-100) : "));

if( not >= 90)

{ JOptionPane.showMessageDialog(null,"A","Ogrenci notu",JOptionPane.PLAIN\_MESSAGE);} else if(not >=75)

{ JOptionPane.showMessageDialog(null,"B","Ogrenci notu",JOptionPane.PLAIN\_MESSAGE);} else if(not >=60)

{ JOptionPane.showMessageDialog(null,"C","Ogrenci notu",JOptionPane.PLAIN\_MESSAGE);} else if(not >=50)

{ JOptionPane.showMessageDialog(null,"D","Ogrenci notu",JOptionPane.PLAIN\_MESSAGE);} else if(not >=40)

{ JOptionPane.showMessageDialog(null,"E","Ogrenci notu",JOptionPane.PLAIN\_MESSAGE);} else

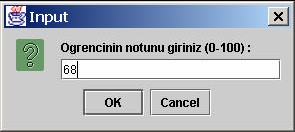
{ JOptionPane.showMessageDialog(null,"F","Ogrenci notu",JOptionPane.PLAIN\_MESSAGE);} System.exit(0);

}

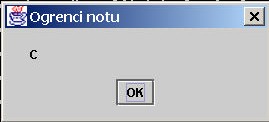
}

Program çiktisi :

### 01013.JPG



**01014.JPG**



Asagidaki program parçasinda a degiskeninin degeri : true yazisini yazdiracaktir.

### boolean a=true; if(a)

**{System.out.println("a degiskeninin degeri : true");} else**

**{System.out.println("a degiskeninin degeri : false");}**

if - elseif - else yapilari iç içe girmis yapilar veya basit mantiklarin **&&** (ve), **||** (veya) islemleriyle baglanmasiyla daha kompleks mantiksal sartlar olusturulabilir.

Iç içe geçmis if yapisi kullanilarak mantiksal kontrol prosesi alttaki program parçasinda açiklanmistir :

***Iç içe geçmis if yapisi***

double x = 7.0; double y = 3.0; if(x > 5)

{

if(y > 5) { System.out.println("x ve y besten buyuktur.");}

elseif(y==5) { System.out.println("x besten buyuktur ve y bese esittir.");} else { System.out.println("x besten buyuktur ve y besten kucuktur.");}

}

else

{

if(y > 5) { System.out.println("x besten kucuktur veya esittir ve y 5 den buyuktur.");} elseif(y==5) { System.out.println("x besten kucuktur veya esittir ve y bese esittir.");} else { System.out.println("x besten kucuktur veya esittir ve y besten kucuktur.");

}

}

Burada ise && terimiyle baglanmis bir program parçacigi görüyorsunuz.

***&& (ve) deyimiyle baglanmis if yapisi***

double x = 7.0; double y = 3.0;

if(x 5 && y 5) { System.out.println("x ve y besten buyuktur.");}

else if(x>5 && y==5) { System.out.println("x besten buyuktur ve y bese esittir.");} else if(x>5 && y<5) { System.out.println("x besten buyuktur ve y besten kucuktur.");} else if(x<5 && y>5) { System.out.println("x besten kucuktur ve y 5 den buyuktur.");} else if(x < 5 && y==5) { System.out.println("x besten kucuktur ve y bese esittir.");}

}

}

Yukaridaki program parçaciklarindan da görebildigimiz gibi her türlü kompleks mantigi bilgisayar programlarina aktarmamiz mümkündür. If - else if - else yapilari programlamada en çok kullanilan mantik aktarimi yapi taslaridir.

### while tekrarlama yapisi

While en çok kullanilan tekrarlama yapilarindan biridir. ( ) içindeki boolean terim true (dogru) oldugu sürece yeniden islemi döndürür. Eger programda boolean islemini sonuçlandiracak bir ifade yoksa sonsuza dek veya program baska bir yöntemle durdurulana dek devam eder.

### while( boolean degisken yada sabit)

**{ boolean degisken veya sabitin degeri true oldugunda yapilan islemler}**

örnegin :

### while(true)

**{ System.out.println("Bu yazi surekli olarak tekrar edecektir");}**

yapisi sonsuz bir döngü olusturur. Asagidaki program yapisi ise sayi 1000 den büyük veya 1000 e esit oldugunda durur.

### int sayi=2; while(sayi <= 1000)

**{ sayi \*= 2; }**

While tekrarlama yapisi bazen biraz daha farkli bir yapi içinde verilebilir. Asagida gösterilen bu yeni yapi, bir önceki while yapisina göre daha ender kullanilar.**do..while** yapisinin while yapisindan temel farki, döngünün içine en az bir kere girilme zorunlugunun olmasidir.

### do

**{**

**boolean degisken veya sabitin degeri true oldugunda yapilan islemler.**

**Bu paranteze boolean degerinin ne olduguna bakilmaksizin en az bir kere girilir**

**}**

**while( boolean degisken yada sabit)**

Program 1.22 de dowhile döngüsünün kullanildigi **dowhileApplet.java** programi görülmektedir

### Program 1.22 : do..while döngüsü örnegi dowhileApplet.java programi import java.awt.Graphics;

**import java.applet.Applet;**

**public class dowhileApplet extends Applet**

**{**

**public void paint(Graphics g)**

**{**

**int saydirici=1; int x=25;**

**do {**

**g.drawString(Integer.toString(saydirici),x,25); x += 15;**

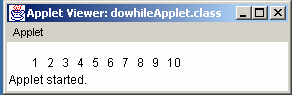
**} while(++saydirici <= 10);**

**}**

**}**

Bu program calistiginda appletviewer Sekil 1.15 deki sonucu verecektir.

### 01015.JPG



**Sekil 1.15 *dowhileApplet.class* dosyasinin appletviewer programi kullanarak görülmesi**

Ayni programi Swing JoptionPane çiktisi kullanarak yazalim :

### Program 1.23 : do..while döngüsü örnegi dowhileSW.java programi

import javax.swing.\*;

public class dowhileSW

{

public static void main(String args[])

{

int saydirici=1;

String s="";

do {

s=s+Integer.toString(saydirici)+" ";

} while(++saydirici <= 10); JOptionPane.showMessageDialog(null,s,

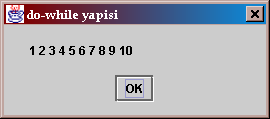
"do-while yapisi",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Program çiktisi :

### 01016.JPG



**Sekil 1.16 *dowhileSW.class* dosyasinin JoptionPane.showMassageDialog metodu kullanarak görülmesi**

* + 1. **for tekrarlama yapisi**

For tekrarlama yapisi genellikle sayilari belli bir düzen içinde arttirmak için kullanilir. Genel yapisi asagida gösterilmistir.programlamadaki sayi saydirmaya dönük uygulamalarda en fazla kullanilan yapidir.

***for tekrarlama yapisi***

**for(baslangiç degerleri ; boolean degisken veya sabiti ; tekrarlama degiskeni artisi)**

**{ boolean degisken veya sabiti true (dogru) oldugunda yapilan islemler }**

Program 1.23 de 1 den yüze kadar sayilarin toplamini hesaplayan bir konsol programi verilmistir.

### Program 1.24 : birden 100 e kadar sayilarin toplamini hesaplayan foryapisi.java konsol programi import java.io.\*;

**class foryapisi**

**{**

**public static void main(String args[])**

**{**

**int toplam=0;**

**for(int sayi=1;sayi<=100;sayi++)**

**{ toplam+=sayi;}**

**System.out.println("1 den 100 e sayilarin toplami : "+toplam);**

**}**

**}**

**foryapisi.java** kompile edildiginde

### 1 den 100 e sayilarin toplami : 5050

sonucunu verecektir.

### Program 1.25 : faizOraniSW.java *applet programi* import java.applet.Applet; // java applet sinifi import java.awt.Graphics; // Java grafik sinifi public class faizOraniApplet extends Applet

**{**

**public void paint( Graphics g)**

**{**

**double ilkYatanPara=100e6;//yuz milyon lira double paraMiktari=ilkYatanPara;**

**double faizOrani=0.85;//yuzde seksenbes int y=40;**

**g.drawString("Yil",25,25); g.drawString("paraMiktari",100,25); for(int yil=1; yil<=10 ; yil++)**

**{**

**paraMiktari\*=(1.0+faizOrani); g.drawString(" "+yil,25,y); g.drawString(" "+paraMiktari,100,y);**

**// not: drawString terimi daima String türü**

**// degiskenle baslamali veya string e dönüstürülmelidir. y+=15;**

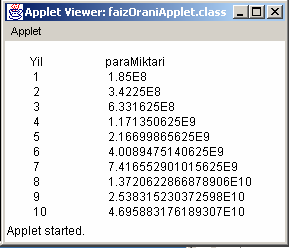
**} // for döngüsünün sonu**

**} // paint metotunun sonu**

**} // faizOraniApplet sinifinin son**

Program 1.24 de yine for döngüsünün uygulamasi konusunda bir örnek problem verilmistir. Programi nasil çalistigini inceleyiniz. Bu programda yillik bilesik faizler on yil için hesaplanmaktadir. Ilk iki faiz oranini elde hesaplayiniz. Programi çalistirarak sonuçlari kontrol ediniz. Program sonuçlari Sekil 1.14 de listelenmistir.

### 01017.JPG



**Sekil 1.17 faizOraniApplet.html appletinin appletviewer browser programinda görünümü Program 1.26 : faizOraniSW.java applet programi**

import javax.swing.JOptionPane; //JOptionPane sinifini çagir

public class faizOraniSW

{

public static void main( String args[])

{

double ilkYatanPara=100e6;//yuz milyon lira double paraMiktari=ilkYatanPara;

double faizOrani=0.85;//yuzde seksenbes int y=40;

String s="Yil Para Miktari\n"; for(int yil=1; yil<=10 ; yil++)

{

paraMiktari\*=(1.0+faizOrani); s=s+yil+" "+paraMiktari+"\n";

} // for döngüsünün sonu JOptionPane.showMessageDialog(null,s,"faiz orani", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

} // main metodunun sonu

} // faizOraniSW sinifinin sonu

### 01018.JPG



**Sekil 1.18 faizOraniSW.java program çiktisinin swing JoptionPane**. **showMessageDialog metodu kullanilarak görüntülenmesi**

* + 1. **switch - case yapisi**

Switch case yapisi çesitli alternatifler içeren bir tablodan seçme yapma türü islerde kullanilir. Temel yapi: switch - case yapisi:

### switch( degisken ismi)

**{**

**case *degiskenin alacagi deger :***

**bu deger olustugunda yapilacak islemler break; // bu deyim döngünün bitmesini saglar.**

**case degiskenin alacagi baska bir deger : bu deger olustugunda yapilacak islemler**

**break; // bu deyim döngünün bitmesini saglar.**

**................**

**default :**

**eger yukaridaki case kutularindan hiçbirine girilemediyse yapilacak islemler**

**break;**

**}**

seklindedir. Switch yapisindan sonra degisken ismi gelir. Her case deyiminden sonra ise degiskenin alacagi yeni bir deger alternatifi verilir. Eger switch yapisinda verilen alternatif hiçbir case blogunda tanimlanmamissa default bloguna gider ve buradaki islemler gerçeklestirilir. Case ve default bloklari break deyimi ile sona erer. break deyimi herhangi bir döngü yapisindan döngü bitmeden çikma amaciyla da kullanilabilir. break yapisiyla döngünün disina çikma amaciyla diger döngü yapilarinda da kullanilabilir(if, for while gibi), fakat bu tür break kullanilarak döngü disina çikilmasi yapisal programlama teknikleri tarafindan tavsiye edilmez. Bilgisayar listelerini yapisalliktan uzaklastirip, spagetti programlama dedigimiz karmakarisik programlamaya yol açabilir. Döngü yapilarindan standart kontrollari yoluyla çikilmasi tavsiye edilir.

### int i=0; while(true)

**{ i++;**

**if(i==3) break;**

**System.out.println("Bu yazi uc kere tekrar edecektir");}**

Program 1.27 de verilen switchApplet.java programi switch deyiminin nasil kullanildigini göstemektedir. Programcilara göre kullanilma sikligi degisebildigi varsayilsa bile, switch yapisi su ana kadar bahsedilen yapilar içinde en az kulanilanidir.

**Program 1.27 *switch - case yapisi örnegi* switchApplet.java *applet programi*** . **Bu program ögrenci notlarini appletden girer ve her nottan kaç tane girildigini listeler.**

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir public class switchApplet extends Applet implements ActionListener

{

//sinif degiskenleri

Label kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi TextField kutugirdisi;//Textfield sinifi degiskeni (nesnesi) kutugirdisi char not;

int Asayisi=0,Bsayisi=0,Csayisi=0,Dsayisi=0,Esayisi=0,Fsayisi=0;

// pencereyi baslatma metotu public void init()

{

kutubasligi=new Label("Ogrencinin notunu giriniz (A B C.. : "); add(kutubasligi);//kutubasligini pencereye yaz

kutugirdisi=new TextField(5); add(kutugirdisi);//kutuyu pencereye yerlestir

// kutuya yeni ilave edilecek komutlari bekle

// her yeni komutta actionPerformed metotunu calistir. kutugirdisi.addActionListener(this);

}

public void paint(Graphics g)

{

g.drawString("Her harf notun toplami : ",25,40); g.drawString("A : "+Asayisi,25,55); g.drawString("B : "+Bsayisi,25,70);

g.drawString("C : "+Csayisi,25,85); g.drawString("D : "+Dsayisi,25,100); g.drawString("E : "+Esayisi,25,115); g.drawString("F : "+Fsayisi,25,130);

}

// girdi alanindaki olan olaylari dinleme metotu public void actionPerformed(ActionEvent e)

{

//ogrencinin notunu pencereden oku String not1=e.getActionCommand(); not=not1.charAt(0);

showStatus(""); // sonuç bölgesindeki yaziyi sil kutugirdisi.setText(""); //kutudaki harfi sil switch(not)

{

case 'A': case 'a':

++Asayisi; break;

case 'B': case 'b':

++Bsayisi; break;

case 'C': case 'c':

++Csayisi; break;

case 'D': case 'd':

++Dsayisi; break;

case 'E': case 'e':

++Esayisi; break;

case 'F': case 'f':

++Fsayisi; break; default:

showStatus("yanlis not tanimlandi yani bir not giriniz."); break;

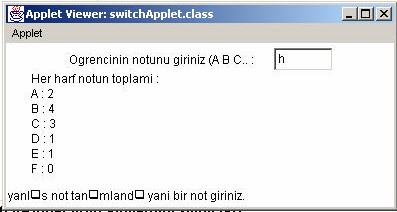
} // switch deyiminin sonu

repaint();//pencereyi yeniden paint metotuna gore ciz

}

}

### 01019.JPG



**Sekil 1.19 switch - case yapisi örnegi switchApplet.html appletinin sonuçlari**

**Program 1.28 *switch - case yapisi örnegi* switchSWA.java swing *applet programi*** . **Bu program ögrenci notlarini appletden girer ve her nottan kaç tane girildigini listeler.**

Import javax.swing.\*; // java swing JApplet sinifini cagir import java.awt.\*; // java pencere awt sinifini cagir

import java.awt.event.\*; // java pencereyi dinleme sinifini cagir

public class switchSWA extends JApplet implements ActionListener

{

//sinif degiskenleri

JLabel kutubasligi; // Label sinifi degiskeni (nesnesi) kutubasligi JTextField kutugirdisi; // Textfield sinifi degiskeni (nesnesi) kutugirdisi JTextArea ciktiAlani; // çiktiyi tanimlayacak alani yaratir

char not;

int Asayisi=0,Bsayisi=0,Csayisi=0,Dsayisi=0,Esayisi=0,Fsayisi=0;

// pencereyi baslatma metodu

public void init()

{

Container c=getContentPane();

kutubasligi=new JLabel("Ogrencinin notunu giriniz (A B C.. : "); c.add(kutubasligi,BorderLayout.WEST);//kutubasligini pencereye yaz kutugirdisi=new JTextField(5); c.add(kutugirdisi,BorderLayout.EAST);//kutuyu pencereye yerlestir ciktiAlani=new JTextArea("\n\n\n "); c.add(ciktiAlani,BorderLayout.SOUTH);

ciktiAlani.setText(" ");

// kutuya yeni ilave edilecek komutlari bekle

// her yeni komutta actionPerformed metodunu calistir. kutugirdisi.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

//ogrencinin notunu pencereden oku String cikti="\n";

String not1=e.getActionCommand(); not=not1.charAt(0);

showStatus(""); // sonu‡ b”lgesindeki yaz• y• sil kutugirdisi.setText(""); //kutudaki harfi sil switch(not)

{

case 'A': case 'a':

++Asayisi; break;

case 'B': case 'b':

++Bsayisi; break;

case 'C': case 'c':

++Csayisi; break;

case 'D': case 'd':

++Dsayisi; break;

case 'E': case 'e':

++Esayisi; break;

case 'F': case 'f':

++Fsayisi; break; default:

showStatus("yanlis not tanimlandi yani bir not giriniz."); break;

} // switch deyiminin sonu

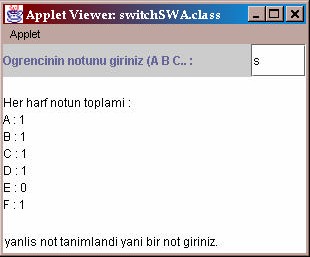
cikti=cikti+"Her harf notun toplami : \n"; cikti=cikti+"A : "+Asayisi+"\n"; cikti=cikti+"B : "+Bsayisi+"\n"; cikti=cikti+"C : "+Csayisi+"\n"; cikti=cikti+"D : "+Dsayisi+"\n"; cikti=cikti+"E : "+Esayisi+"\n"; cikti=cikti+"F : "+Fsayisi+"\n"; ciktiAlani.setText(cikti);

repaint();//pencereyi yeniden paint metoduna gore ciz

}

}

### 01020.JPG



**Sekil 1.20 switch - case yapisi örnegi switchSWA.html swing appletinin sonuçlari**

* 1. **ARITMETIK ISLEMLERDE DEGISKEN TÜRÜ DEGISTIRME (CASTING) OPERASYONU**

C veya C++ dilinde bir degisken türünden degiskeni diger degisken türünden degiskene direk olarak atama mümkündür. örnegin

### char a='x'; int b=a;

islemi c++ da geçerli bir islemdir. Javada ise bu islem hata verir. fakat ayni islem

### char a='x'; int b=(int)a;

seklinde gerçeklestirilebilir **(int)** deyimi orijinal olarak char olarak tanimlanan a degiskeninin degerini int türüne dönüstürmüs sonra da in türünden b degiskenine yüklemistir. Bu dönüsüm islemlerini tüm degisken türleri için kullanabiliriz, yalniz burada programcinin çok dikkatli olmasi ve degisken türlerini, ve sinirlarini iyi tanimasi gerekmektedir. Degisken türlerini degistirirken bazi bilgiler kaybolacagi gibi hiç istemedigimiz bir sonuca ulasma olasiligimiz da mevcuttur.

### SAYI DEGIL VE SONSUZ SONUÇLARI

Java gerçek sayi degisken türleri sayi degil (NaN) ve arti sonsuz (Pozitif infinity) ve eksi sonsuz(negative infinity) sonuçlari verebilirler. sayi degil (NaN) sonucu sifir bölü sifir, sifir çarpi sonsuz gibi islemlerden çikar. sonsuz islemi ise sayi bölü sifir gibi islemlerden çikar. Bu islemler sadece float ve double degiskenleri için tanimlidir. Tamsayi degiskenler sifira bölünemezler.

### ALISTIRMALAR

1. Notdefteri programini veya diger herhangibir editörü kullanarak Hosgeldiniz.java programini incele. Konsole (MS DOS veya UNIX console) ortaminda asagidaki komutlari kullanarak calistir :

### javac Hosgeldiniz.java java Hosgeldiniz

**Program 1.1** : **Hosgeldiniz.java *konsol programi***

import java.io.\*; class Hosgeldiniz

{

public static void main (String args[])

{

System.out.println("Java Sinifina Hos Geldiniz!")

}

}

1. Notdefteri programini kullanarak HosgeldinizApplet.java programini incele. Yine not defteriyle HosgeldinizApplet.html programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir:

### javac HosgeldinizApplet.java appletviewer HosgeldinizApplet.html

**Program 1.2 : HosgeldinizApplet.java**

import java.applet.Applet; import java.awt.\*;

public class HosgeldinizApplet extends Applet

{

public void paint(Graphics g)

{ g.drawString("Java sinifina hos geldiniz!",25,25); }

}

1. Notdefteri programini kullanarak ikisayitopla.java programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir :

### javac ikitamsayitopla.java java ikitamsayitopla

**Program 1.29 : ikitamsayitopla.java**

import java.io.\*; //giris çikis import Text;

class ikitamsayitopla

{

public static void main (String args[]) throws IOException

{

int sayi1,sayi2; int toplam=0;

//

Text m=new Text(); System.out.println("Bir tam sayi giriniz : "); sayi1=m.readInt();

System.out.println("Ikinci bir tam sayi girermisiniz : "); sayi2=m.readInt();

toplam=sayi1+sayi2;

System.out.println("Iki tam sayinin toplami : "+toplam);

}

}

### 5. ikitamsayitoplaSWA.java

Notdefteri programini kullanarak ikisayitoplaSWA.java ve ikisayitoplaSWA.html programini ve incele. Konsole (MS DOS) ortaminda asagidaki komutlari kullanarak calistir :

### javac ikitamsayitoplaSWA.java appletviewer ikitamsayitoplaSWA.html

**Program 1.30 : ikitamsayitoplaSWA.java**

import java.awt.\*; //giris cikis import javax.swing.\*;

public class ikitamsayitoplaSWA extends JApplet

{

int toplam;

public void init()

{

int sayi1,sayi2; toplam=0;

//

sayi1=Integer.parseInt(JOptionPane.showInputDialog(" Bir tam sayi giriniz : ")); sayi2=Integer.parseInt(JOptionPane.showInputDialog(" Ikinci bir tam sayi giriniz : ")); toplam=sayi1+sayi2;

}

public void paint(Graphics g)

{

g.drawRect(15,10,270,20);

g.drawString("iki sayinin toplami : "+toplam,25,25);

}

}

1. Notdefteri programini kullanarak **tamsayitoplaApplet.java** programini incele. Yine not defteriyle tamsayitoplaApplet.html programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir

:

### javac tamsayitoplaApplet .java

appletviewer tamsayitoplaApplet.html

### Program 1.31 tamsayitoplaApplet.java programi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir

import java.awt.event.\*; // java pencereyi dinleme sinifini cagir

public class tamsayitoplaApplet extends Applet implements ActionListener

{

//sinif degiskenleri

Label kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi TextField kutugirdisi;//Textfield sinifi degiskeni (nesnesi) kutugirdisi int sayi;

int toplam;

// pencereyi baslatma metotu public void init()

{

kutubasligi=new Label("Bir tamsayi giriniz : "); add(kutubasligi);//kutubasligini pencereye yaz kutugirdisi=new TextField(10); add(kutugirdisi);//kutuyu pencereye yerlestir toplam=0;

// kutuya yeni ilave edilecek komutlari bekle

// her yeni komutta actionPerformed metotunu calistir.

kutugirdisi.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metotu public void actionPerformed(ActionEvent e)

{

//sayiyi pencereden oku

Integer sayi1=new Integer(e.getActionCommand() ); sayi=sayi1.intValue();

// sayiyi okutma islemini asagidaki deyimle de yapabiliriz

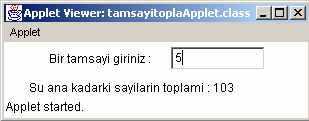
// sayi=Integer.parseInt( e.getActionCommand() ); toplam+=sayi; // veya toplam=toplam+sayi; repaint();//pencereyi yeniden paint metotuna göre ciz

}

public void paint(Graphics g)

{ g.drawString("Su ana kadarki sayilarin toplami : "+toplam,25,50); }

### 01021.JPG



**Sekil 1.21 tamsayitoplaApplet.html appletinin sonuçlari**

1. **tamsayitoplaSWA.java**

Notdefteri programini kullanarak tamsayitoplaSWA.java ve ikisayitoplaSWA.html programini ve incele. Konsole (MS DOS) ortaminda asagidaki komutlari kullanarak calistir :

javac tamsayitoplaSW.java appletviewer tamsayitoplaSW.html

### Program 1.32 tamsayitoplaSWA.java programi :

import java.awt.\*; //giris cikis import javax.swing.\*;

public class ikitamsayitoplaSWA extends JApplet

{

int toplam;

public void init()

{

int sayi1,sayi2; toplam=0;

//

sayi1=Integer.parseInt(JOptionPane.showInputDialog(" Bir tam sayi giriniz : ")); sayi2=Integer.parseInt(JOptionPane.showInputDialog(" Ikinci bir tam sayi giriniz : ")); toplam=sayi1+sayi2;

}

public void paint(Graphics g)

{

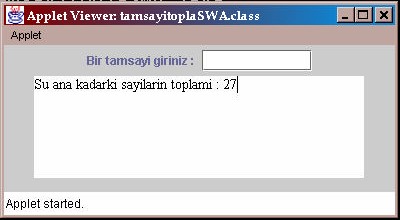
g.drawRect(15,10,270,20);

g.drawString("iki sayinin toplami : "+toplam,25,25);

}

}

### 01022.JPG



**Sekil 1.22 tamsayitoplaSWA.html appletinin sonuçlari**

**7. tamsayitoplaSWFrame.java**

Not defteri programini kullanarak tamsayitoplaSW.java programini ve incele. Konsole (MS DOS) ortaminda asagidaki komutlari kullanarak calistir :

javac tamsayitoplaSWFrame.java java tamsayitoplaSWFrame

Program 1.33 tamsayitoplaSWFrame.java programi import javax.swing.\*; //java swing sinifini çagirir import java.awt.\*; // Java grafik sinifini çagirir

import java.awt.event.\*; // java pencereyi dinleme sinifini çagirir

public class tamsayitoplaSWFrame extends JFrame implements ActionListener

{

//Layout container tanimi

//sinif degiskenleri

JLabel kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi JTextField kutugirdisi;//Textfield sinifi degiskeni (nesnesi) kutugirdisi JTextArea cikti;

int sayi; int toplam;

// pencereyi baslatma metodu public tamsayitoplaSWFrame()

{

super("tam sayilarin toplami"); Container cerceve=getContentPane(); cerceve.setLayout( new FlowLayout() );

kutubasligi=new JLabel("Bir tamsayi giriniz : "); cerceve.add(kutubasligi);//kutubasligini pencereye yaz kutugirdisi=new JTextField(10); cerceve.add(kutugirdisi);//kutuyu pencereye yerlestir cikti=new JTextArea(6,30);

cikti.setFont(new Font("Times New Roman",Font.PLAIN,14)); cerceve.add(cikti);

toplam=0;

// kutuya yeni ilave edilecek komutlari bekle

// her yeni komutta actionPerformed metodunu calistir. kutugirdisi.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

//sayiyi pencereden oku

Integer sayi1=new Integer(kutugirdisi.getText() ); sayi=sayi1.intValue();

// sayiyi okutma islemini asagidaki deyimle de yapabiliriz

// sayi=Integer.parseInt( e.getActionCommand() ); toplam+=sayi; // veya toplam=toplam+sayi; kutugirdisi.setText("");

cikti.setText("Su ana kadarki sayilarin toplami : "+toplam);

}

public static void main( String[] argv ) { tamsayitoplaSWFrame frame = new tamsayitoplaSWFrame(); ; frame.addWindowListener( new WindowAdapter()

{

public void windowClosing( WindowEvent e ) { System.exit( 0 );}

});

frame.setSize( 400, 300 ); frame.show();

}

}

1. Notdefteri programini kullanarak dortArtiIslem.java programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir :

javac dortArtiIslem.java java dortArtiIslem

### Program 1.34 dortArtiIslem.java programi

import java.io.\*; //giris çikis import Text;

class dortArtiIslem

{

public static void main (String args[]) throws IOException

{

int sayi1,sayi2;

//

DataInputStream cin=new DataInputStream(System.in); System.out.println("Bir tam sayi giriniz : "); sayi1=Text.readInt(cin);

System.out.println("Ikinci bir tam sayi giriniz : "); sayi2=Text.readInt(cin);

System.out.println("Iki tam sayinin toplami : "+(sayi1+sayi2)); System.out.println("Iki tam sayinin farki : " +(sayi1-sayi2)); System.out.println("Iki tam sayinin carpimi : "+(sayi1\*sayi2)); System.out.println("Iki tam sayinin bolumu : "+(sayi1/sayi2)); System.out.println("Iki tam sayinin kalani : "+(sayi1%sayi2));

// Matematik tipi degisken islemleri :

// + : topla

// - : cikar

// \* : carp

// / : bol

// % : bolumun kalani

// += : kendisi ve sag tarafindaki ile topla

// sayi1+=3 ve sayi1=sayi1+3 ayni islemdir

// -= : kendisinden sag tarafindakini cikar

// sayi1-=3 ve sayi1=sayi1-3 ayni islemdir

// \*= : kendisi ve sag tarafindakini carp

// sayi1\*=3 ve sayi1=sayi1\*3 ayni islemdir

// /= : kendisini sag tarafindakine bol

// sayi1/=3 ve sayi1=sayi1/3 ayni islemdir

// %= : kendisini sag tarafindakine bol kalanini ver

// sayi1%=3 ve sayi1=sayi1%3 ayni islemdir

// ++ : kendisine bir ekle

// sayi1++ ve sayi1=sayi1+1 ayni islemdir

// ++ : kendisine bir ekle

// ++sayi1 ve sayi1=sayi1+1 ayni islemdir

// -- : kendisinden bir cikar

// sayi1-- ve sayi1=sayi1-1 ayni islemdir

// -- : kendisinden bir cikar

// --sayi1 ve sayi1=sayi1-1 ayni islemdir boolean mantik1,mantik2;

mantik1=true; // dogru mantik2=false;//yanlis

System.out.println("dogru ve yanlis : "+(mantik1 && mantik2)); System.out.println("dogru veya yanlis : "+(mantik1 || mantik2)); mantik1=( sayi1 sayi2 );

mantik2=( sayi1 < sayi2 );

// Boolean (mantik) tipi degisken islemleri :

// : buyuktur

// < : kucuktur

// == : esittir

// = : buyuk veya esittir

// <= : kucuk veya esittir

// ! : mantigi tersine cevirme

// != : esit degildir

// && : ve

// || : veya mantik1=( sayi1 sayi2 );

mantik2=( sayi1 < sayi2 );

System.out.println("sayi1 buyuktur sayi2 : "+mantik1); System.out.println("sayi 1 kucuktur sayi2 : "+mantik2);

}

}

### Bir tam sayi giriniz : 3 Ikinci bir tam sayi giriniz : 2 Iki tam sayinin toplami : 5 Iki tam sayinin farki : 1

**Iki tam sayinin carpimi : 6 Iki tam sayinin bolumu : 1 Iki tam sayinin kalani : 1 dogru ve yanlis : false dogru veya yanlis : true sayi1 buyuktur sayi2 : true sayi 1 kucuktur sayi2 : false**

1. Notdefteri programini kullanarak gercekSayiOrtalamaApplet.java programini incele. Yine not defteriyle gercekSayiOrtalamaApplet.html programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir :

javac gercekSayiOrtalamaApplet .java appletviewer gercekSayiOrtalamaApplet.html

### Program 1.34 : gercekSayiOrtalamaApplet.java programi

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir

public class gercekSayiOrtalamaApplet extends Applet implements ActionListener

{

//sinif degiskenleri

Label kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi TextField kutugirdisi;// Textfield sinifi degiskeni (nesnesi) kutugirdisi double sayi; // Gerçel degisken sayi

double toplam; // Gerçel degisken toplam

double carpim; // Gerçel degisken carpim

double aritmetikOrtalama; // Gerçel degisken aritmetik ortalama double geometrikOrtalama; // Gerçel degisken geometrik ortalama int sayici; // tam sayi degisken toplam rakam sayisini sayici

// pencereyi baslatma metotu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

kutubasligi=new Label("Bir sayi giriniz : "); add(kutubasligi);

kutugirdisi=new TextField(10); add(kutugirdisi);

toplam=0; carpim=1.0; aritmetikOrtalama=0; geometrikOrtalama=0; sayici=0;

kutugirdisi.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metotu

// Her yeni rakam girildiginde bu metot çagirilir public void actionPerformed(ActionEvent e)

{

sayici++;

Double sayi1=new Double(e.getActionCommand() ); sayi=sayi1.doubleValue();

toplam+=sayi; // veya toplam=toplam+sayi; aritmetikOrtalama=toplam/sayici; carpim\*=sayi; // veya carpim=carpim\*sayi;

geometrikOrtalama=Math.pow(carpim,(1.0/sayici));//carpimin koku kutugirdisi.setText("");//kutudaki rakami sil

repaint();//pencereyi yeniden paint metotuna göre ciz

}

public void paint(Graphics g)

{

// drawString Metotundaki eksen sistemi

// ----------------------- x ekseni

// |

// |

// |

// V y ekseni

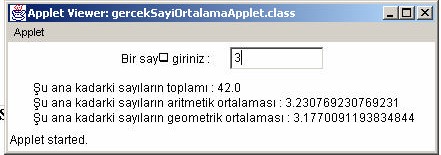
g.drawString("Su ana kadarki sayilarin toplami : "+toplam,25,50);

g.drawString("Su ana kadarki sayilarin aritmetik ortalamasi : "+aritmetikOrtalama,25,65); g.drawString("Su ana kadarki sayilarin geometrik ortalamasi : "+geometrikOrtalama,25,80);

}

}

### 01023.JPG



**Sekil 1.23 gercekSayiOrtalamaApplet.html çiktisi**

1. Notdefteri programini kullanarak bessayitopla.java programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir :

### javac bessayitopla.java java bessayitopla

**Program 1.35 : bessayitopla.java programi, bes gerçek sayiyi toplar. import java.io.\*; // giris çikis**

**import Text; // okuma sinifi class bessayitopla**

**{**

**public static void main (String args[]) throws IOException**

**{**

**double sayi; double toplam=0; int saydirici=0;**

**//**

**Text cin=new Text(); while( saydirici < 5)**

**{**

**System.out.println("Bir tam sayi giriniz : "); sayi=cin.readDouble();**

**toplam+=sayi; saydirici++;**

**}**

**System.out.println("sayilarin toplami : "+toplam);**

**}**

**}**

**Bir tam sayi giriniz : 1 Bir tam sayi giriniz : 2 Bir tam sayi giriniz : 3 Bir tam sayi giriniz : 4 Bir tam sayi giriniz : 5 sayilarin toplami : 15.0**

1. Notdefteri programini veya herhangi bir editör programi kullanarak StringDegisken.java programini incele. Konsole (MS Dos) ortaminda asagidaki komutlari kullanarak calistir :

### javac StringDegisken.java java StringDegisken

**Program 1.36 : StringDegisken.java programi, String degiskenlerini + operatörüyle ekleme islemi**

import java.io.\*; //giris çikis sinifini cagir class StringDegisken

{

public static void main (String args[]) throws IOException

{

String s1,s2,s3,s4; int i1;

String s; s1="Ali "; s2="Veli "; s3="49";

s4=" Elli"; s=s1+s2+s3+s4; System.out.println(s); i1=49;

s=s1+s2+i1+s4;

System.out.println(s);

}

}

### Ali Veli 49 Elli Ali Veli 49 Elli

1. alistirma 7 deki bessayi topla.java programini iyice anladiktan sonra birden yüze (100) kadar sayilarin ortalamasini hesaplayan birdenyüzeOrtalama.java program ve sinifini (Konsole) yaz ve calistir.

### Program 1.37 : Birdenyüzeortalama.java programi, birden yüze kadar sayilarin ortalamasini hesaplar.

import java.io.\*; // giris cikis import Text; // okuma sinifi class birdenyuzeortalama

{

public static void main (String args[])

{

double sayi=1; double toplam=0; while( sayi <= 100)

{

toplam+=sayi++;

}

System.out.println("sayilarin toplami : "+toplam); System.out.println("sayilarin ortalamasi : "+toplam/100.0);

}

}

### sayilarin toplami : 5050.0 sayilarin ortalamasi : 50.5

1. If yapisini kullanma programi, karsilastirmaApplet.java , Iki String türü degiskeni Applet TextField girdi alanindan girer ve birbiriyle karsilastirir. Sonuçlari Applet ortaminda yazdirir. Bu program applet olarak dizayn edilmistir. programi çalistiriniz ve sonuçlarini applet ortaminda aliniz.

### Program 1.38 : karsilastirmaApplet.java programi

//if deyiminin kullanimi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir

import java.awt.event.\*; // java pencereyi dinleme sinifini cagir

public class karsilastirmaApplet extends Applet implements ActionListener

{

Label soru1; //kullaniciya birinci degeri girme sorusu sorar TextField girdi1; //birinci degeri girme kutusu

Label soru2; //kullaniciya ikinci degeri girme sorusunu sorar TextField girdi2; //ikinci degeri girme kutusu

int sayi1,sayi2; //girilen iki sayi

//graphic kullanici baglantisindaki ilk degerleri tanimla public void init()

{

soru1=new Label("Bir sayi giriniz "); girdi1=new TextField(10);

soru2=new Label("Ikinci bir sayi giriniz"); girdi2=new TextField(10);

add(soru1); //soru1 yazisini applete cizer

add(girdi1);//girdi1 kutusunu applete cizer add(soru2); //soru2 yazisini applete cizer girdi2.addActionListener(this);

// soru2 den sonra girdi (return) tusunu bekler

// girdi(return) tusundan sonra actionPerformed

// metotuna gider

add(girdi2); //girdi2 kutusunu applete cizer

}

// sonuclari applete yazdir public void paint(Graphics g)

{

g.drawString("Karsilastirma sonuclari : ",70,75); if(sayi1 == sayi2)

g.drawString(sayi1+" == "+sayi2,100,90); if(sayi1 != sayi2)

g.drawString(sayi1+" != "+sayi2,100,105); if(sayi1 < sayi2)

g.drawString(sayi1+" < "+sayi2,100,120); if(sayi1 sayi2)

g.drawString(sayi1+" "+sayi2,100,135); if(sayi1 <= sayi2)

g.drawString(sayi1+" <= "+sayi2,100,150); if(sayi1 = sayi2)

g.drawString(sayi1+" = "+sayi2,100,165);

}

// eger girdi(return) tusuna basildiysa bu

// metotu cagir

public void actionPerformed( ActionEvent e)

{

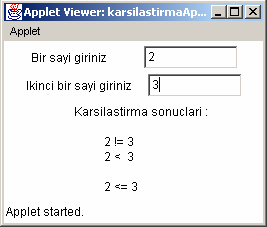
// sayi 1 i girdi kutusundan oku sayi1=Integer.parseInt(girdi1.getText());

// sayi 2 i girdi kutusundan oku sayi2=Integer.parseInt(girdi2.getText()); repaint();

}

}

### 01024.JPG



**Sekil 1.24 karsilastirmaApplet.html çiktisi**

1. **whileyapisi.java** programini incele, çalistir ve sonuçlarini konsol ortaminda al.

### Program 1.39 :whileyapisi.java programi. Bu program while yapisini kullanarak sayilarin iki ile çarpimini hesaplar.

import java.io.\*; //java girdi cikti sinifini cagir class whileyapisi

{

public static void main(String args[])

{

int sayi=2; while(sayi<=1000)

{

sayi\*=2;

System.out.println("sayi = "+sayi);

}

}

sayi = 4

sayi = 8

sayi = 16

sayi = 32

sayi = 64

sayi = 128

sayi = 256

sayi = 512

sayi = 1024

**dowhileApplet.java** programini incele ve çalistir. Bu program do-while döngü yapisi kullanarak bir sayi dizisini applet ortaminda yazdirmaktadir.

**Program 1.40 :**dowhileApplet.java **programi**

import java.awt.Graphics; import java.applet.Applet;

public class dowhileApplet extends Applet

{

public void paint(Graphics g)

{

int saydirici=1; int x=25;

do {

g.drawString(Integer.toString(saydirici),x,25);

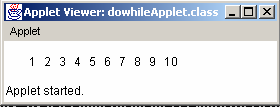
x += 15;

} while(++saydirici <= 10);

}

}

### 01025.JPG



**Sekil 1.25 dowhileApplet.html çiktisi**

1. foryapisi.java programini incele ve çalistir. Bu program for döngü yapisi kullanarak bir sayi dizisini applet ortaminda yazdirmaktadir.

**Program 1.41 :** foryapisi.java **programi**

import java.io.\*; class foryapisi

{ public static void main(String args[])

{

int toplam=0;

for(int sayi=1;sayi<=100;sayi++)

{ toplam+=sayi;}

System.out.println("1 den 100 e sayilarin toplami : "+toplam);

}

}

foryapisi.java **programi çiktisi :**

1 den 100 e sayilarin toplami : 5050

1. faizOraniApplet.java programini incele ve çalistir. Bu program döngü yapisi kullanarak faiz oranlarini hesaplamaktadir.
2. switchApplet.java programini incele ve çalistir. Bu program **switch** yapisini kullanmaktadir..
3. Bir java konsol programi (kilo.java) yazarak On kisilik bir siniftaki kisilerin agirliklarini girdi olarak giriniz.

 Agirliklari 40 kg dan az olanlari zayif olarak,

 Agirliklari 40 kilo ile 80 kilo arasinda olanlari normal,

 Agirliklari 80 kg in üstünde olanlari sisman katagorisinde siniflarsak

her sinifta kaç kisi oldugunu yazdiriniz. Dosyanizin adi **kilo.java**, sinifinizin adi : **kilo** olacaktir. Bilmediginiz yapilari örnek problemleri inceleyerek ögrenebilirsiniz.

**Program 1.42 :** kilo.java **programi.**

import java.io.\*; // giris çikis import Text; // okuma sinifi class kilo

{

public static void main (String args[]) throws IOException

{

double kilo; // Text cin=new Text();

System.out.println("kilonuzu giriniz : "); kilo=cin.readDouble();

if( kilo <= 40)

{ System.out.println("zayif"); }

else if( kilo40 && kilo<80 )

{ System.out.println("normal"); }

else

{ System.out.println("sisman"); }

}

}

program çiktisi :

kilonuzu giriniz : 45 normal

kilonuzu giriniz : 95 sisman

1. **yildiz3forApplet.java programi** altta verilmistir. bu programi inceleyiniz ve Applet çiktisini olusturunuz.

### Program 1.21 : yildiz3forApplet.java programi

import java.awt.Graphics; import java.applet.Applet;

public class yildiz3forApplet extends Applet

{

public void paint(Graphics g)

{

int x,y;

for(x=15;x <= 150;x+=15)

{ y=15;

g.drawString("\*",x,y);

g.drawString("\*",x,y+45);

}

for(y=15;y <= 65;y+=15)

{ x=15;

g.drawString("\*",x,y);

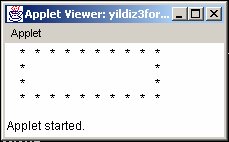
g.drawString("\*",x+135,y);

}

}

}

### 01026.JPG



**Sekil 1.26 yildiz3forApplet.html çiktisi**

1. **yildiz1whileApplet.java** programi altta verilmistir. Bu programi inceleyiniz ve applet çiktisini olusturunuz.

### Program 1.22 : yildiz1whileApplet.java programi

import java.awt.Graphics;import java.applet.Applet; public class yildiz1whileApplet extends Applet

{

public void paint(Graphics g)

{

int i=1; int x=15;

while(i++ <= 11) //i degeri birden 11 a kadar degisir

{

g.drawString("\*",x,x);

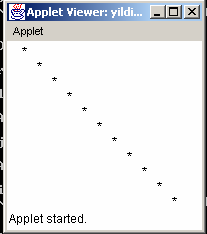
x += 15; // x degeri her i icin 15 artiyor

}

}

}

### 01027.JPG



**Sekil 1.27 yildiz1whileApplet.html program çiktisi**

1. **yildiz2forApplet.java** programi altta verilmistir. Bu programi inceleyiniz ve applet çiktisini olusturunuz.

### Program 1.23 : yildiz2forApplet.java import java.awt.Graphics;

**import java.applet.Applet;**

**public class yildiz2forApplet extends Applet{ public void paint(Graphics g)**

**{ int i;**

**int x=0;**

**for(i=1;i <= 11;i++) //i degeri birden 11 a kadar degisir**

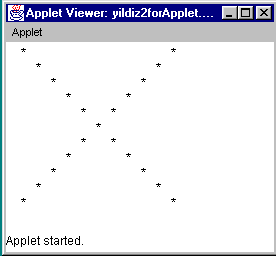
**{**

**x += 15; // x degeri her i icin 15 artiyor g.drawString("\*",x,x);**

**g.drawString("\*",x,(180-x));**

**}**

**}**

**} 01028.JPG**

**Sekil 1.28 yildiz2forApplet.html çiktisi**

1. **yildiz2whileApplet.java** programi altta verilmistir. Bu programi inceleyiniz ve applet çiktisini olusturunuz.

### Program 1.24 : yildiz2whileApplet.java programi

import java.awt.Graphics; import java.applet.Applet;

public class yildiz2whileApplet extends Applet

{

public void paint(Graphics g)

{

int i=1; int x=0;

while(i++ <= 11) //i degeri birden 11 a kadar degisir

{

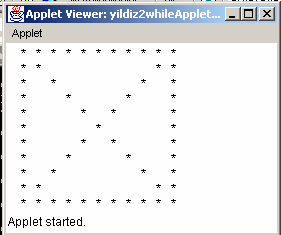
x += 15; // x degeri her i icin 15 artiyor g.drawString("\*",x,15); // x lineer degisken y 15 g.drawString("\*",15,x); // x 15 y lineer degisken g.drawString("\*",x,165); // x lineer degisken y 165 g.drawString("\*",165,x); // x 165 y lineer degisken g.drawString("\*",x,x); // çiz g.drawString("\*",x,(180-x)); // çiz

}

}

}

### 01029.JPG



**Sekil 1.29 yildiz2whileApplet.html çiktisi**

1. **yildiz1forApplet.java** programi altta verilmistir. Bu programi inceleyiniz ve applet çiktisini olusturunuz.

### Program 1.24 yildiz1forApplet.java programi

import java.awt.Graphics; import java.applet.Applet;

public class yildiz1forApplet extends Applet

{

public void paint(Graphics g)

{

int i;

int x=25;

for(i=1;i <= 10;i++) //i degeri birden 10 a kadar degisir

{

g.drawString("\*",x,x);

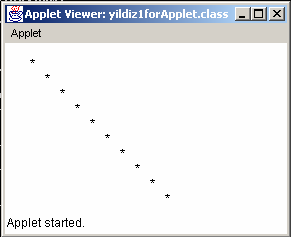
x += 15; // x degeri her i icin 15 artiyor

}

}

}

### 01030.JPG



**Sekil 1.30 yildiz1forApplet.html çiktisi**

1. **piApplet.java** programi altta verilmistir. Bu programi inceleyiniz ve applet çiktisini olusturunuz. Burada dizi formülü ve for döngüsünden yararlanilarak **** sayisi hesaplanmaktadir. Kullanilan seri formülünün dönüsme hizinin yüksek olmadigindan seri 100000 terim için hesaplanmistir. Bu serinin açik yazilimi :

###  = 4 - 4/3 + 4/5 -4/7+4/9-4/11+......

seklindedir.

**Program 1.25 piApplet.java** programi. import java.awt.Graphics;

import java.applet.Applet;

public class piApplet extends Applet

{

public void paint(Graphics g)

{

int i;

double pi=0; int artieksi=-1;

for(i=1;i < 100000;i++)

{

artieksi\*=-1;

pi+=4/(2.0\*i-1.0)\*artieksi;

}

g.drawString("pi : "+pi,25,25);

}

}

1. **Pisagor üçlüsü :** bir dik üçgende üç kenarin da tam sayi olmasi mümkündür. Bu tür sayi setine pisagor üçlüsü adi verilir. Pisagor üçlüsünde ilk iki sayinin karelerinin toplami üçüncü sayinin karesine esit olmalidir. Birincidikkenar, ikincidikkenar ve hipotenüs tam sayi pisagor üçlülerini üçlü for döngüsü kulanarak bulan bir program yaziniz.
2. Java applet programi kullanarak asagidaki sekli applet de çizdiriniz :

\* \* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Program 1.26 H3A1.java** programi. Bu program üçgen bir yildizi alani çizdirir. import java.awt.Graphics;

import java.applet.Applet;

public class H3A1 extends Applet

{

public void paint(Graphics g)

{

int x,y;

for(x=15;x <= 150;x+=15)

{

for(y=15;y <= x;y+=15)

{

g.drawString("\*",x,y);

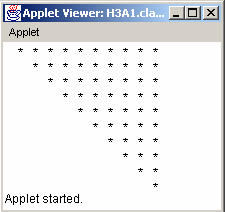
}

}

}

}

### 01031.JPG



**Sekil 1.31 yildiz üçgeni çizen H3A1.html çiktisi**

1. Java applet programi kullanarak asagidaki sekli appletde çizdiriniz

\* \* \* \* \* \* \* \* \* \*

\* \*

\* \*

\* \* \* \* \* \* \* \* \* \*

### Program 1.27 : H3A2.java, Yildizlardan olusan dikdörtgen çizer

import java.awt.Graphics; import java.applet.Applet;

public class H3A2 extends Applet

{

public void paint(Graphics g)

{

int x,y;

for(x=15;x <= 150;x+=15)

{

g.drawString("\*",x,15);

g.drawString("\*",x,60);

}

for(y=15;y <= 60;y+=15)

{

g.drawString("\*",15,y);

g.drawString("\*",150,y);

}

}

}

1. Java konsol programi kullanarak asagidaki sekli konsolda çizdiriniz. x ekseni ve y eksenindeki toplam yildiz sayisini ekrandan giriniz.

\* \* \* \* \* \* \* \* \* \*

\* \*

\* \*

\* \* \* \* \* \* \* \* \* \*

1. Java konsol programi **faktoriyel.java** programini olusturunuz ve bir sayinin faktoriyelini hesaplayiniz.

### n!=n\*(n-1)\*(n-2)….4.3.2.1 4!= 4\*3\*2\*1 3!=3\*2\*1

**Program 1.28 faktoriyel.java programi. Sayinin faktoriyelini döngü kullanarak hesaplar.**

import java.io.\*; import Text; class faktoriyel

{

public static void main(String args[]) throws IOException

{

int faktoriyel=1; Text cin=new Text(); int faktoriyelsayi;

System.out.println("Bir tamsayi giriniz:"); faktoriyelsayi=cin.readInt();

for(int sayi=1;sayi<=faktoriyelsayi;sayi++)

{ faktoriyel\*=sayi;}

System.out.println(faktoriyelsayi+" faktoriyel: "+faktoriyel);

}

}

### Bir tamsayi giriniz: 4 4 faktoriyel: 24

1. Java konsol programi kullanarak asagidaki formulü hesaplayiniz:

### ex = 1 + x /1! + x2 / 2! + x3 / 3! + x4 / 4! + .. not : e=2.7182818

**Program 1.29 H3A4.java programi. For döngüsü kullanarak exponent hesaplar.**

import java.io.\*; import Text; class H3A4

{

public static void main(String args[]) throws IOException

{

double faktoriyel=1; double x;

double us=1; double exponent=1;

Text cin=new Text();

System.out.println("Bir sayi giriniz:"); x=cin.readDouble();

for(double sayi=1;sayi<=200;sayi++)

{

faktoriyel\*=sayi; us\*=x;

exponent+=us/faktoriyel;

}

System.out.println("exp("+x+") = "+exponent); System.out.println("gercek exp("+x+") = "+Math.exp(x));

}

}

Bir sayi giriniz: 3

exp(3.0) = 20.08553692318766 gercek exp(3.0) = 20.08553692318767

1. Alistirma 28 deki exponent hesabini for döngüsü yerine while döngüsü kullanarak gerçeklestiriniz.
2. Hata fonksiyonu erf(x) asagidaki formülle tanimlanabilir.

1  *n x* 2 *n*  1

## *erf*

( *x* ) 

 (  1 )

*n*  0

**

*n* ! ( 2 *n*

#  1 )

Bir konsol programi yaziniz. Text sinifi kullanarak disaridan x degerini okuyunuz. For döngüsü kullanarak formülü hesaplayiniz ve sonucunu yazdiriniz.

1. gerçek sayi faktoriyel fonksiyonu **?** 'nin 1.05 noktasindaki degeri **?** (1.05) in degeri 0.9735 olarak bilinmektedir. **? ?f**onksiyonu da faktoriyel gibi

### ? ?x+1??? ?x \* ? ?x?

ifadesiyle tanimlanabildigine göre **?** (13.05) degerini hesaplayiniz.

### H1OD1.java

**Program 1.30 H1OD1.java**

import java.io.\*; class H1OD1

{

public static void main(String args[])

{

// Bu program konsolda(MS DOS)

// "Java Programlama dili"

// cok zevkli bir ders

// yazdirir

System.out.println("\"Java Programlama dili\"\ncok zevkli bir ders");

}

}

* programini herhangi bir editör kullanarak yaziniz.
* H1OD1.java adiyla kaydediniz.
* javac H1OD1.java terimiyle compile ediniz
* yanlislarinizi düzeltiniz
* Java H1OD1 terimiyle isletiniz

### H1OD2.java

**Program 1.31 H1OD2.java**

import java.applet.Applet; import java.awt.\*;

public class H1OD2 extends Applet

{

public void paint(Graphics g)

{

// Bu program apletde(pencere)

// "Java Programlama dili"

// cok zevkli bir ders

// yazdirir

g.drawString(("\"Java Programlama dili \"",25,25); g.drawString("cok zevkli bir ders");",25,40);

}

}

 programini herhangi bir editör kullanarak yaziniz.

 H1OD2.java adiyla kaydediniz.

 javac H1OD2.java terimiyle compile ediniz.

 Yanlislarinizi düzeltiniz

 Herhangibir editor yardimiyla H1OD2.html dosyasini olusturunuz.

 **appletviewer H1OD2.html** komutuyla programi çalistiriniz.

 Bir browser programinda H1OD2.html dosyasini açiniz

### H1OD3.java

Iki gerçek sayinin(double) (bu sayilar 2 ve 5 olsun) farklarini Konsol ortaminda (MS DOS) hesaplayacak java programini yazip isletiniz

### H1OD4.java

Iki gerçek sayinin(double) bölümlerini Konsol ortaminda (MS DOS) hesaplayacak java programini yazip isletiniz. Sayilar ekrandan Text sinifini kullanarak girilecektir (okunacaktir)

### H1OD5.java

Iki gerçek sayinin(double) çarpimlarini Applet(window) ortaminda hesaplayacak java programini yazip isletiniz.

### H1OD6.java

Bes gerçek sayinin(double) çarpimlarini Konsol ortaminda (MS DOS) hesaplayacak java programini yazip isletiniz. Sayilar ekrandan Text sinifini kullanarak girilecektir (okunacaktir)

1. **H2AL1.java** : if -elseif -else yapisi ve boolean degisken örnegi, konsol programi. Programi incele ve çalistir.

### Program 1.32 . H2AL1.java programi boolean double ve String degisken karsilastirma örnegi

import java.io.\*;

class H2AL1

{

public static void main(String args[]) throws IOException

{

// Bu program konsolda(MS DOS)

// if metotu ve boolean kullanarak

// karsilatirmalar yapar

// ve sonucu yazdirir boolean b1;

boolean b2; boolean b3; double sayi1; double sayi2; String s1,s2;

Text cin=new Text();

b1=true; b2=false;

b3=b1&&b2; System.out.println(b1+"&&"+b2+" = "+b3);

if(b3)

{

System.out.println("if yapisinin ici b3 = "+b3);

}

else if(!b3)

{

System.out.println("else if yapisinin ici b3 = "+b3);

}

// iki gercek sayiyi karsilastir

System.out.print("Birinci gercek sayiyi oku : "); sayi1=cin.readDouble(); System.out.print("Ikinci gercek sayiyi oku : "); sayi2=cin.readDouble();

b3=(sayi1>sayi2); if(sayi1>sayi2)

{

System.out.println(sayi1+">"+sayi2+" = "+b3);

}

else if(sayi1<sayi2)

{

System.out.println(sayi1+"<"+sayi2+" = "+(sayi1<sayi2));

}

else

{

System.out.println(sayi1+"=="+sayi2+" = "+(sayi1<sayi2));

}

// iki String karsilastir

System.out.print("Birinci String'i oku : "); s1=cin.readString(); System.out.print("Ikinci String'i oku : "); s2=cin.readString();

b3=s1.equals(s2); if(s1.equals(s2))

{

System.out.println(s1+" esittir "+s2+" b3 = "+b3);

}

else

{

System.out.println(s1+" esit degildir "+s2+" b3= "+b3);

}

}//main metotunun sonu

} //H2AL1 sinifinin sonu

1. H2AL2 : while döngüsü örnegi applet programi, 10 dan 30 a kadar sayilari applete yazdirir. Programi incele ve çalistir.

### Program 1.33 : H2Al2.java, while döngüsü örnegi applet programi, 10 dan 30 a kadar sayilari applete yazdirir.

import java.applet.Applet; import java.awt.\*;

public class H2AL2 extends Applet

{

public void paint(Graphics g)

{

// Bu program apletde(pencere)

// 10 dan 30 a kadar sayilari yazdirir int i=0;

double x=10.0; while(x<=20.0)

{

g.drawString(" "+(x+=1.0),i+=30,25);

}

}

}

1. H2AL3 : do - while döngüsü örnegi applet programi. Programi incele ve çalistir.

### Problem 1.34 H2AL3.java : do-while döngüsü. Bu problem 10 dan 30 a kadar sayilari 2 ser 2 ser yazdirir.

import java.applet.Applet; import java.awt.\*;

public class H2AL3 extends Applet

{

public void paint(Graphics g)

{

// Bu program apletde(pencere)

// 10 dan 30 a kadar sayilari ikiser ikiser yazdirir int i=0;

double x=10.0; do

{

g.drawString(" "+(x+=2.0),i+=30,25);

} while(x<=20.0)

}

}

1. H2AL4 : for döngüsü örnegi, applet programi . Programi incele ve çalistir.

### Problem 1.35 H2AL4.java : for döngüsü. Bu problem 10 dan 30 a kadar sayilari 3 er 3 er yazdirir.

import java.applet.Applet; import java.awt.\*;

public class H2AL4 extends Applet

{

public void paint(Graphics g)

{

// Bu program apletde(pencere) 10 dan 30 a

// kadar sayilari 3 er 3 er yazdirir int i=0;

for(double x=10.0;x<=20.0;x+=3.0)

{

g.drawString(" "+x,i+=30,25);

}

}

}

1. H2AL6: iç içe iki döngü, applet programi . Programi incele ve çalistir.

### Problem 1.36 H2AL6.java : for döngüsü. Bu problem iki applet ortaminda boyutlu bir matrix yazdirir.

import java.applet.Applet; import java.awt.\*;

public class H2AL6 extends Applet

{

public void paint(Graphics g)

{

// Bu program apletde(pencere)

// 11 12 13 14

// 21 22 23 24

// 31 32 33 34

// 41 42 43 44 yazdirir for(int i=1;i<5;i++)

{

for(int j=1;j<5;j++)

{

g.drawString(""+i+j,i\*25,j\*25);

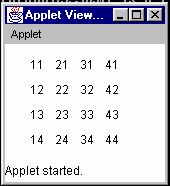
} // for(int j ...dongusunun kapanisi

} //for(int i ..dongusunun kapanisi

} //paint metotunun kapanisi

} //H2AL6 sinifinin kapanisi

### 01032.JPG



**Sekil 1.32 rakam matrisi çizen H2AL6.html çiktisi**

1. H2AL7: sayi1 den sayi2 e n er ner sayilarin toplami, applet programi

### Problem 1.37 H2AL7.java : for döngüsü. Bu problem sayi1 den sayi2 e n er ner sayilarin toplamini hesaplar.

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class H2AL7 extends Applet implements ActionListener

{

int sayi1,sayi2,sayi3; int toplam;

Label s1,s2,s3; //yazi alanlari sinifi TextField t1,t2,t3; //girdi alanlari (kutulari)

//pencereyi baslatma metotu public void init()

{

s1=new Label("den "); s2=new Label("e kadar ");

s3=new Label("er sayilarin toplami : "); t1=new TextField(5);

t2=new TextField(5); t3=new TextField(5);

//pencereye ekle add(t1);

add(s1);

add(t2);

add(s2);

add(t3);

add(s3); t3.addActionListener(this);

}

public void paint(Graphics g)

{

g.drawString(""+sayi1+"den "+sayi2+"e kadar sayilarin "+ sayi3+"er "+sayi3+"er toplami :"+toplam,15,100);

} //paint metotunun kapanisi

//girdi alanindan sinyal bekle ve sinyal gelince su islemleri yap public void actionPerformed(ActionEvent e)

{

toplam=0;

//sayi1, sayi2 ve sayi3 kutulardan oku Integer Sayi1=new Integer(t1.getText()); sayi1=Sayi1.intValue();

Integer Sayi2=new Integer(t2.getText()); sayi2=Sayi2.intValue();

Integer Sayi3=new Integer(t3.getText()); sayi3=Sayi3.intValue();

for(int i=sayi1;i<=sayi2;i+=sayi3)

{

toplam+=i;

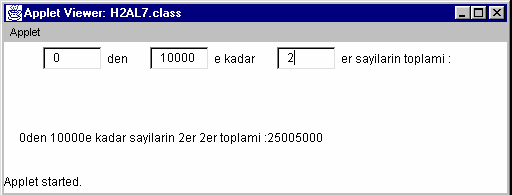
}

repaint();

}//ActionPerformed metotunun kapanisi

} //H2AL7 sinifinin kapanisi

### 01033.JPG



**Sekil 1.33 degisken sinirlarla toplama yapan H2AL7.java programinin applet çiktisi**

1. H2OD1

while döngüsü ve konsol programi kullanarak 1 den n e kadar olan sayilarin aritmetik ortalamasini hesaplayiniz.

n sayisi konsoldan girilecektir.

Aritmetik ortalama (1+2+3+..+n)/n formülüyle hesaplanir.

### Problem 1.38 H2OD1.java : while döngüsü. Bu problem 1 den n e kadar sayilarin aritmetik ortalamasini hesaplar.

import java.io.\*; class H2OD1

{

public static void main (String args[])

{

//bu program 1 den n ye kadar olan

//sayilarin aritmatik ortalamasini hesaplar int n;

int sayi=1; double ort=0;

Text cin = new Text (); System.out.println("n : "); n = cin.readInt (); while(sayi<=n)

{

ort=(ort + sayi)/sayi; sayi++;

}

System.out.println("aritmatik ortalama : "+ort);

}

}

1. H2OD2

For döngüsü ve "\*" karekterini kullanarak 5\*4 yildiz dikdortgenini applet olarak çizdiriniz

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

### Problem 1.39 H2OD2.java : for döngüsü. Bu problem 5\*4 yildiz dikdortgenini applet olarak çizdir

import java.awt.Graphics; import java.applet.Applet;

public class H2OD2 extends Applet

{

public void paint(Graphics g)

{

int x,y; for(y=1;y<6;y++)

{

for(x=1;x<5;x++)

{

g.drawString ("\*",20\*x,20\*y);}

}

}

}

1. H2OD3

do- while döngüsü kullanarak 1 den n e kadar olan sayilarin küplerinin toplamini ve aritmetik ortalamasini applet programi olarak hesaplatiniz. n sayisi applet kutu girdisinden girilecektir.

### Problem 1.40 H2OD3.java : 1 den n e kadar sayilarin küplerinin toplamini applet olarak çizdir

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class H2OD3 extends Applet implements ActionListener

{

int sayi1; int toplam; double ort;

Label s1,s2; //yazi alanlari sinifi TextField t1; //girdi alanlari (kutulari)

//pencereyi baslatma metotu public void init()

{

s1=new Label("kup ortalamasi 1 den "); s2=new Label("e kadar ");

t1=new TextField(5);

//pencereye ekle add(s1);

add(t1);

add(s2); t1.addActionListener(this);

}

public void paint(Graphics g)

{

g.drawString(""+"1den "+sayi1+" e kadar kup ortalamasi :"+ort,15,100);

} //paint metotunun kapanisi

//girdi alanindan sinyal bekle ve sinyal gelince su islemleri yap public void actionPerformed(ActionEvent e)

{

toplam=0;

//sayi1 i kutulardan oku

Integer Sayi1=new Integer(t1.getText()); sayi1=Sayi1.intValue();

int i=1; do

{

toplam+=i\*i\*i;

}while(++i<=sayi1); ort=(double)toplam/(double)sayi1; repaint();

}//ActionPerformed metotunun kapanisi

} //H2OD3 sinifinin kapanisi

**46.** H2OD3\_2000

do- while döngüsü kullanarak 1 den n e kadar olan sayilarin küplerinin toplamini ve aritmetik ortalamasini swing applet (JApplet) programi olarak hesaplatiniz. n sayisi JApplet kutu girdisinden girilecektir.

### Problem 1.40 H2OD3a\_2000.java : 1 den n e kadar sayilarin küplerinin toplamini swing applet olarak çizdir

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class H2OD3a\_2000 extends Japplet implements ActionListener

{ int sayi1; int toplam; double ort;

JLabel s1,s2; //yazi alanlari sinifi JTextField t1; //girdi alanlari (kutulari) JTextArea ciktiAlani;

//pencereyi baslatma metotu public void init()

{

Container c=getContentPane(); c.setLayout(new FlowLayout()); Color renk=c.getBackground();

s1=new JLabel("küp ortalamasi 1 den "); s2=new JLabel("e kadar ");

t1=new JtextField(5); //pencereye ekle ciktiAlani=new JTextArea(); ciktiAlani.setBackground(renk); c.add(s1);

c.add(t1);

c.add(s2); c.add(ciktiAlani); t1.addActionListener(this);

}

//girdi alanindan sinyal bekle ve sinyal gelince su islemleri yap public void actionPerformed(ActionEvent e)

{

toplam=0; //sayi1 i kutulardan oku sayi1=Integer.parseInt(t1.getText()); int i=1;

do { toplam+=i\*i\*i;

}

while(++i<=sayi1); ort=(double)toplam/(double)sayi1;

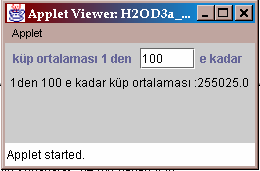
ciktiAlani.setText(""+"1den "+sayi1+" e kadar küp ortalamasi :"+ort); repaint();

}

//ActionPerformed metotunun kapanisi

} //H2OD3\_2000 sinifinin kapanisi

### 01034.JPG



**Sekil 1.34 1 den n e kadar sayilarin küplerinin ortalamasini hesaplayan H2OD3a.java programinin JApplet çiktisi**

H2OD3\_2000 Ayni problemin ikinci bir çözümü

do- while döngüsü kullanarak 1 den n e kadar olan sayilarin küplerinin toplamini ve aritmetik ortalamasini swing applet (JApplet) programi olarak hesaplatiniz. n sayisi JOptionPane.showInputDialog() girdisinden girilecektir.

import java.awt.\*; import javax.swing.\*;

public class H2OD3\_2000 extends JApplet

{

int n, i = 1;

int toplam = 0; long toplam3 = 0; double ortalama;

public void init()

{

n=Integer.parseInt(JOptionPane.showInputDialog(" Bir tam sayi giriniz : ")); while (i <= n){

toplam3 += i\*i\*i; toplam += i;

i++;

}

ortalama = (double)toplam / n;

}

public void paint(Graphics g)

{

g.drawRect(15,10,370,20); g.drawRect(15,50,370,20);

g.drawString(n + "'ye kadar olan sayilarin kuplerinin toplami : " + toplam3,25,25); g.drawString(n + "'ye kadar olan sayilarin aritmetik ortalamasi : " + ortalama,25,65);

}

}

1. H2OD4

herhangi bir döngü yapisi ve konsol programi kullanarak, n=200 degeri için

1-x/1!+x2/2!-x3/3!+x4/4!-x5/6!+..+xn-1/(n-1)!-xn/n! formülünü hesaplayiniz.

not: ! (faktoriyel) sayinin birer eksilerek bire kadar çarpildigi diziye denir. Örnegin 4!=4\*3\*2\*1

### Problem 1.41 H2OD4.java : 1-x/1!+x2/2!-x3/3!+x4/4!-x5/6!+..+xn-1/(n-1)!-xn/n! formülünün hesabi.

import java.io.\*; import Text; class H2OD4

{

public static void main(String args[]) throws IOException

{

double faktoriyel=1; double x;

double us=1; double exponent=1;

Text cin=new Text(); System.out.println("Bir sayi giriniz:"); x=cin.readDouble();

for(double sayi=1;sayi<=200;sayi++)

{

faktoriyel\*=sayi; us\*=(-1)\*x;

exponent+=us/faktoriyel;

}

System.out.println("exp(-"+x+") = "+exponent);

}

}

c:\co\java\prog>Bir sayi giriniz: 1.0 c:\co\java\prog>exp(-1.0) = 0.36787944117144245

1. H2OD5

Bir bilgisayar konsol bilgisayar programi yaziniz. bu program kullanicinin ismini ve boyunu sorsun. eger boyu

1.50 cm nin altindaysa "o **kullaniciismi** siz pek kisaymissiniz" yazsin,eger kullanici boyu 1.90 cm nn üzerindeyse "o **kullaniciismi** boyunuzda ne kadar uzunmus" yazsin. kullanici boyu bu degerlerin arasindaysa "merhaba kullaniciismi sistemimize hosgeldiniz" yazsin.

### Problem 1.42 H2OD5.java : if-elseif blogu

import java.io.\*; import Text; class H2OD5

{

public static void main(String args[]) throws IOException

{

int boy;

String isim;

Text cin=new Text(); System.out.println("isminiz nedir : "); isim=cin.readString();

System.out.println(""+isim+" lutfen boyunuzu giriniz:"); boy=cin.readInt();

if(boy<=150)

{System.out.println(isim+" siz de pek kisaymisiniz.");} else if(boy>=190)

{System.out.println(isim+" boyunuz da ne kadar uzunmus.");} else

{System.out.println("merhaba "+isim+" sistemimize hos geldiniz.");}

}

}

### 49. H2OD5\_2000

Bir bilgisayar konsol bilgisayar programi yaziniz. bu program kullanicinin ismini ve boyunu sorsun. eger boyu

1.50 cm nin altindaysa "o **kullaniciismi** siz pek kisaymissiniz" yazsin,eger kullanici boyu 1.90 cm nn üzerindeyse "o **kullaniciismi** boyunuzda ne kadar uzunmus" yazsin. kullanici boyu bu degerlerin arasindaysa "merhaba kullaniciismi sistemimize hosgeldiniz" yazsin. Çikti için java swing JOPtionPane kullaniniz.

### Problem 1.42 H2OD5\_2000.java : if-elseif blogu

import javax.swing.JOptionPane; class H2OD5\_2000

{

public static void main(String args[])

{

int boy;

String isim;

isim=JoptionPane.showInputDialog("isminiz nedir : "); System.out.println(""+isim+" lutfen boyunuzu giriniz:"); boy=Integer.parseInt(

JoptionPane.showInputDialog(" lutfen boyunuzu giriniz:")); if(boy<=150)

{

JoptionPane.showMessageDialog(null,"sizde pek kisaymissiniz", "boy : "+boy,JOptionPane.PLAIN\_MESSAGE);

}

else if(boy>=190)

{

JoptionPane.showMessageDialog(null," boyunuz da ne kadar uzunmus", "boy : "+boy,JOptionPane.PLAIN\_MESSAGE);

}

else

{

JoptionPane.showMessageDialog(null, "merhaba "+isim+" sistemimize hos geldiniz.", "boy : "+boy,JOptionPane.PLAIN\_MESSAGE);

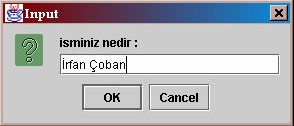
}

System.exit(0);

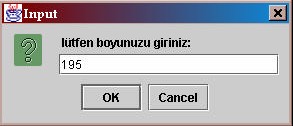
}

}

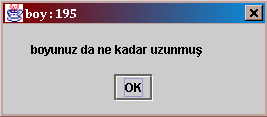
### 01035.JPG



**01036.JPG**



**01037.JPG**



**Sekil 1.35,1.36,1.37 if yapisini test eden H2OD5\_2000.java programinin swing JoptionPane çiktisi 50** H2AL1\_2000 : if -elseif -else yapisi ve boolean degisken örnegi, JoptionPane konsol programi Import javax.swing.JOptionPane;

class H2AL1\_2000

{

public static void main(String args[])

{

// Bu program konsolda(MS DOS)

// if metodu ve boolean kullanarak

// karsilastirmalar yapar

// ve sonucu yazdirir boolean b1;

boolean b2; boolean b3; double sayi1; double sayi2; String s1,s2; b1=true; b2=false; b3=b1&&b2;

JoptionPane.showMessageDialog(null,""+b1+"&&"+b2+" = "+b3, "if yapisinin disi",JOptionPane.PLAIN\_MESSAGE);

if(b3)

{

JoptionPane.showMessageDialog(null,"if yapisinin ici b3 = "+b3, "if yapisinin ici",JOptionPane.PLAIN\_MESSAGE);

}

else if(!b3)

{

JoptionPane.showMessageDialog(null,"else if yapisinin ici b3 = "+b3, "if yapisinin ici",JOptionPane.PLAIN\_MESSAGE);

}

// iki gercek sayiyi karsilastir sayi1=Double.parseDouble(JOptionPane.showInputDialog("birinci gercek sayiyi oku : ")); sayi2=Double.parseDouble(JOptionPane.showInputDialog("ikinci gercek sayiyi oku : ")); b3=(sayi1>sayi2);

if(sayi1>sayi2)

{

JoptionPane.showMessageDialog(null,""+sayi1+">"+sayi2+" = "+b3, "iki gercek sayi karsilastir",JOptionPane.PLAIN\_MESSAGE);

}

else if(sayi1<sayi2)

{

JoptionPane.showMessageDialog(null, ""+sayi1+"<"+sayi2+" = "+(sayi1<sayi2),

"iki gercek sayi karsilastir",JOptionPane.PLAIN\_MESSAGE);

}

else

{

JoptionPane.showMessageDialog(null,""+sayi1+"=="+sayi2+" = "+(sayi1==sayi2), "iki gercek sayi karsilastir",JOptionPane.PLAIN\_MESSAGE); System.out.println();

}

// iki String karsilastir

s1=JoptionPane.showInputDialog("birinci String'i oku : "); s2=JoptionPane.showInputDialog("ikinci String'i oku : "); b3=s1.equals(s2);

if(s1.equals(s2))

{

JoptionPane.showMessageDialog(null,s1+" esittir "+s2+" b3 = "+b3, "iki gercek sayi karsilastir",JOptionPane.PLAIN\_MESSAGE);

}

else

{

JoptionPane.showMessageDialog(null,s1+" esit degildir "+s2+" b3= "+b3, "iki gercek sayi karsilastir",JOptionPane.PLAIN\_MESSAGE);

}

System.exit(0);

}//main metodunun sonu

} //H2AL1\_2000 sinifinin sonu

1. H2AL2\_2000 : while döngüsü örnegi JoptionPane konsol programi import javax.swing.JOptionPane;

public class H2AL2\_2000

{

public static void main(String args[])

{

// Bu program

// 10 dan 20'ye kadar sayilari yazdirir double x=10.0;

String s=""; do

{

s=s+" "+(x+=1.0);

}

while(x<=20.0); JoptionPane.showMessageDialog(null,s,

"10 dan 30 a sayilar",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 01038.JPG



1. H2AL4\_2000 : for döngüsü örnegi, JoptionPane konsol programi import javax.swing.JOptionPane;

public class H2AL4\_2000

{

public static void main(String args[])

{

// Bu program

// 10 dan 20 ye kadar sayilari yazdirir String s="";

for(double x=10.0;x<=30.0;x+=1.0)

{

s=s+" "+x;

}

JoptionPane.showMessageDialog(null,s,

"10 dan 30 a sayilar",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

1. H2AL5\_2000 : switch- case yapisi, JoptionPane konsol programi import javax.swing.JOptionPane;

class H2AL5\_2000

{

public static void main(String args[])

{

// Bu program konsolda(MS DOS)

// switch case metodunu kullan• r char not;

Text cin=new Text(); int A=0;

int B=0; int C=0; int D=0; int E=0; int F=0; while(true)

{

String s="";

not=JOptionPane.showInputDialog("notu oku : (z cikis) ").toCharArray()[0]; if(not=='z') {System.exit(0);}

switch(not)

{

case 'A':case 'a':

{

++A;

break;

}

case 'B':case 'b':

{

++B;

break;

}

case 'C':case 'c':

{

++C;

break;

}

case 'D':case 'd':

{

++D;

break;

}

case 'E':case 'e':

{

++E;

break;

}

case 'F':case 'f':

{

++A;

break;

}

default:

{

JoptionPane.showMessageDialog(null,"yanlis not tanimlandi yeni bir not giriniz", "switch - case örnegi",JOptionPane.ERROR\_MESSAGE);

break;

}

} // switch deyiminin sonu s=s+"A larin sayisi :"+A+"\n"; s=s+"B lerin sayisi :"+B+"\n"; s=s+"C lerin sayisi :"+C+"\n"; s=s+"D lerin sayisi :"+D+"\n"; s=s+"E lerin sayisi :"+E+"\n"; s=s+"F lerin sayisi :"+F+"\n";

JoptionPane.showMessageDialog(null,s,

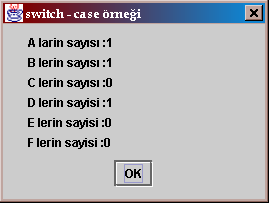
"switch - case örnegi",JOptionPane.PLAIN\_MESSAGE);

}

}//main metodunun sonu

} //H2AL5\_2000 sinifinin sonu

### 01039.JPG



**01040.JPG**



1. H2AL6\_2000: iç içe iki döngü, JoptionPane konsol programi import javax.swing.JOptionPane;

public class H2AL6\_2000

{

public static void main(String args[])

{

// Bu program

// 11 12 13 14

// 21 22 23 24

// 31 32 33 34

// 41 42 43 44 yazdirir String s="";

for(int i=1;i<5;i++)

{

for(int j=1;j<5;j++)

{

s=s+" "+i+j;

} // for(int j ...dongusunun kapanisi s=s+"\n";

} //for(int i ..dongusunun kapanisi

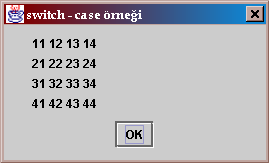
JOptionPane.showMessageDialog(null,s,

"switch - case örnegi",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

} //main metodunun kapanisi

} //H2AL6\_2000 sinifinin kapanisi

### 01041.JPG



1. H2AL7\_2000: sayi1 den sayi2 e n er ner sayilarin toplami, swing Japplet programi, JTextArea çiktisi ile Import javax.swing.\*;

import java.awt.\*; import java.awt.event.\*;

public class H2AL7\_2000 extends JApplet implements ActionListener

{ int sayi1,sayi2,sayi3; int toplam;

double ort;

JLabel s1,s2,s3; //yazi alanlari sinifi JTextField t1,t2,t3; //girdi alanlari (kutulari) JTextArea ciktiAlani;

//pencereyi baslatma metotu public void init()

{

Container c=getContentPane(); c.setLayout(new FlowLayout()); Color renk=c.getBackground(); s1=new JLabel("den");

s2=new JLabel("e kadar ");

s3=new JLabel("er sayilarin toplami : "); Font f=s1.getFont();

Color r1=s1.getForeground();

t1=new JTextField(5); //pencereye ekle t2=new JTextField(5); //pencereye ekle t3=new JTextField(5); //pencereye ekle

ciktiAlani=new JTextArea(); t1.setBackground(renk); t1.setForeground(r1); t2.setBackground(renk); t2.setForeground(r1); t3.setBackground(renk); t3.setForeground(r1); t1.setFont(f);

t2.setFont(f); t3.setFont(f);

ciktiAlani.setForeground(r1); ciktiAlani.setBackground(renk); ciktiAlani.setFont(f);

c.add(t1);

c.add(s1);

c.add(t2);

c.add(s2);

c.add(t3);

c.add(s3); c.add(ciktiAlani); t3.addActionListener(this);

}

//girdi alanindan sinyal bekle ve sinyal gelince su islemleri yap public void actionPerformed(ActionEvent e)

{

toplam=0; //sayi1 i kutulardan oku sayi1=Integer.parseInt(t1.getText()); sayi2=Integer.parseInt(t2.getText()); sayi3=Integer.parseInt(t3.getText()); int i;

for(i=sayi1;i<=sayi2;i+=sayi3)

{

toplam+=i;

}

ciktiAlani.setText(""+sayi1+"den "+sayi2+"e kadar sayilarin "+ sayi3+"er "+sayi3+"er toplami :"+toplam);

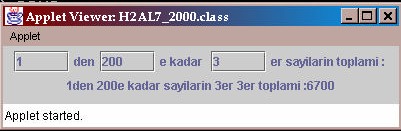
repaint();

}

//ActionPerformed metotunun kapanisi

} //H2OD3 sinifinin kapanisi

### 01042.JPG



1. H2AL8\_2000: sayinin e inci kuvvetinin hesaplanmasi (for döngüsü örnegi) JoptionPane konsol programi import javax.swing.JOptionPane;

class H2AL8\_2000

{

public static void main(String args[]) throws IOException

{

double faktoriyel=1; double x;

double us=1; double exponent=1;

x=Double.parseDouble(JOptionPane.showInputDialog("Bir sayi giriniz : ")); for(double sayi=1;sayi<=200;sayi++)

{

faktoriyel\*=sayi; us\*=x;

exponent+=us/faktoriyel;

}

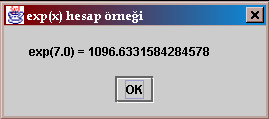
JOptionPane.showMessageDialog(null,"exp("+x+") = "+exponent, "exp(x) hesap örnegi",JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

### 01043.JPG



1. H2AL9\_2000: pi sayisini hesaplayan bir seri formülü. JoptionPane konsol programi import javax.swing.JOptionPane;

class H2AL9\_2000

{

public static void main(String args[])

{

int i;

double pi=0; int artieksi=-1;

for(i=1;i < 100000;i++)

{

artieksi\*=-1;

pi+=4/(2.0\*i-1.0)\*artieksi;

}

String s="pi : "+pi+"\n";

s=s+"pi nin gercek degeri: "+Math.PI; JOptionPane.showMessageDialog(null,s,"pi sayisi hesabi", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

### 01044.JPG



**58 H2OD1\_2000**

for döngüsü ve konsol programi kullanarak 1 den n e kadar olan sayilarin aritmetik ortalamasini hesaplayiniz. n sayisi swing JOptionPane olarak girilecektir.

Aritmetik ortalama (1+2+3+..+n)/n formülüyle hesaplanir.

import java.io.\*;

import javax.swing.JOptionPane;

class H2OD1\_2000

{

public static void main (String args[]) throws IOException

{

int n;

int toplam=0; double ortalama;

n=Integer.parseInt(JOptionPane.showInputDialog("Aritmetik ortalama kaca kadar yapilacak?

\nLutfen n sayisini giriniz :")); for (int i = 1; i <= n; i++){ toplam += i;

}

ortalama = (double)toplam / n;

JOptionPane.showMessageDialog(null,n + "'ye kadar olan sayilarin aritmetik ortalamasi : " + ortalama + " dir", "Aritmetik Ortalama",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 59 H2OD2\_2000

while döngüsü ve "\*" karekterini kullanarak 7\*4 yildiz dikdortgenini swing JOptionPane veya swing JApplet kullanarak çizdiriniz

\* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \*

import java.io.\*;

import javax.swing.JOptionPane;

class H2OD2T\_2000

{

public static void main (String args[]) throws IOException

{

int i = 0,j; String s = "";

while (i++ < 4)

{

j=0;

while(j++ < 7)

{

s += "\* ";

}

s += "\n";

}

JOptionPane.showMessageDialog(null,s, "Yildiz Dikdortgen"

,JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 01045.JPG



60 H2OD6\_2000

for döngüsü ve "\*" karekterini kullanarak grafik ortaminda swing JApplet ve paint metodunu kullanarak asagidaki sekli çizdiriniz

\*

\*

\* \* \* \* \* \* \*

\*

\*

import javax.swing.\*; import java.awt.\*;

public class H2OD6\_2000 extends JApplet

{

public void paint(Graphics g)

{

int yer = 0;

for(int i=0; i < 7; i++)

{

g.drawString("\*",80,i\*20+20);

}

for(int j=0; j < 7; j++)

{

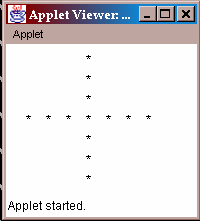
g.drawString("\*",j\*20+20,80);

}

}

}

### 01046.JPG



### BÖLÜM 2 : METODLAR VE OBJECT KULLANIMI

### JAVA API KÜTÜPHANESI

Gerçek problemlerin boyutlari genelde birinci bölümde gördügümüz problemlere göre çok daha büyüktür. Bu yüzden genelde programlar küçük parçalara bölünerek daha kolaylikla anlasilir sekle getirilmeye çalisilir. Buna *böl parçala ve yönet teknigi* de diyebiliriz. Bu küçük parçalara javada metot adi verilir. Java programlari yeni metotlari içeren java siniflariyla(class) java sinif kütüphanelerinde yer alan metotlarin birlesmesinden olusur. Java kütüphanesinde matematik islemlerinden grafik, ses, giris çikis islemleri, yanlis kontrolu gibi birçok konularda yazilmis metotlar mevcuttur. Java metotlar kolleksiyonu **Java API** Adini alir. Java sinif kütüphanesi adi da verilir. Süphesiz bunun disinda java metodlar kolleksiyonlarinin oldugu çesitli diger kütüphaneler de mevcutur.

Java API kütüphanelerinin bazilari sunlardir :

**java.applet : (java applet paketi)** Java applet paketi applet grafik ortamini saglar (grafik programlarinin internet browser programlari araciligiyla görüntülenebilmesinin temellerini olusturur.). Bu sinifa paralel olarak Java 1.2 de grafik applet arayüz birimi **javax.swing.JApplet** gelistirilmistir.

**java.awt** : (**java window araçlari paketi)** Bu sinif grafik arayüzleri tanimlamaya yara. Java 1.2 de paralel sinif

**javax.swing** gelistirilmistir.

**java.awt.color:** renk tanimlar

**java.awt.datatransfer:** bilgisayarin geçic hafizasiyla java programi arasinda bilgi akisini saglar.

**java.awt.dnd :** ekle ve çalistir türü programlar arasinda geçis yapabilen paketlerden olusur.

**java.awt.event :** grafik programlama ortamiyla kullanici arasindakiiliskiyi saglar. Örnegin grafik penceresinde return tusuna basildiginda veya farenin dügmesine basildiginda ne yapilmasi gerektigini belirler. **Java.awt** ve **javax.swing** siniflarinin ikisiyle de kullanilir.

**java.awt.font :** yazi fontlari ve bu fontlarin degistirilmesiyle ilgili programalri içerir.

**java.awt.geom :** iki boyutlu geometrik sekilleri tanimlar.

**java.awt.im :** java programlarina çince, japonca ve kore dilinde girdi girebilmek için tanimlanmis bir arabirimdir.

**java.awt.image :** java görüntü isleme paketi

**java.awt.image.renderable :** bir program içindeki resimleri depolama ve degistirme için gerekli olan programlari içerir.

**java.awt.print :** java programlarindan yazicilara yazim için bilgi aktaran paketleri içerir.

**java.beans :** java fasulyeleri paketi, bu paketin detaylari için ilgili bölümü inceleyiniz.

**java.beans.beancontext :** tekrar kullanilabilen ve otomatik olarak birlestirilebilen program parçaciklari tanimlar.

**java.io :** dosya ve ekrandan girdi ve çikti yapmak için gerekli program paketleri tanimlanir.

**java.lang :** bu paket birçok temel program içerir Bunlardan biri olan Math paketi asagida açiklanacaktir. Lang paketi java programlarinca otomatik olarak kullanilir. Import deyimiyle iave edilmesi gerekmez.

**java.lang.ref** : bu paket java programiyla garbage collector (çöp toplayicisi) denilen otomatik kullanilmayan hafiza temizleme programiyla iliskiyi saglar.

**java.lang.reflect :** bu paketteki programlar java programinin çagrildiginda içinde bulunan degisken ve metotlari dinamik olarak tespit etmesinive kullanma izni almasini saglar.

**java.math :** bu sinif degisebilir hassasiyette arimetik islemler yapmak için gereken tanimlari ve siniflari içerir.

**java.net :** ag islemlerinde kullanilan çesitli paketleri tanimlar

**java.rmi , java.rmi.activation, java.rmi.dgc, java.rmi.registry, java.rmi.server:** programlarin ve metodlarin uzaktan çalisabilmeleri için gerekli tanimlamalari içerir.

**java.security, java.security.acl, java.security.cert, java.security.interfaces, java.security.spec :** güvenlikle ilgili programlar demetini içerir.

**java.sql** : database programiyla java programlarini baglantisini saglar.

**java.text :** sayi karekter ve string tarih gibi islemlerle ilgili programlar demeti içerir. Örnegin çok dilli programlar yazmayi saglayan **internalisation** paketi bu paket içinde yer alir.

**java.util :** bu sinif tarih ve zamanla ilgili fonksiyonlar, tesadüfi sayi üreticileri, **StringTokenizer** gibi programlari barindirir.

**java.util.jar** bu paket java programlarini paketlemeye yariyan jar yapilarini tanimlamada gerekli olan programlari tanimlar.

**java.util.zip :** bu paket programlari sikistirmaya yarayan programlari tanimlar.

**Java.accessibility :** bu paket özürlülerin kullanabilecegi ara üniteler tanimlar. Ekran okuyucular ve ekran büyüteçleri gibi.

**javax.swing :** grafik kullanici arabirimi tanimlar. Bu siniftakitanimlar araciligiyla ekrandan grafik ortaminda veri transferi kolaylikla yapilabilir.

**javax.swing.borders :** grafik arabirimi sinir yaratiminda çesitli sinir sekilleri yaratmakta kullanilir.

**Javax.swing.colorchooser :** renk seçimi için tanimlanmis grafik kullanici arabirimini tanimlar.

Java API dosyalarindaki metotlarin sadece isimlerinin listesi yaklasik 200 sayfa boyutunda bir liste tuttugundan burada sadece yeri geldigine programlari kullanirken örneklerde gerekli API isimlerini görecegiz. Su ana kadar kullandigimiz java programlarinda java.applet, java.awt, java.io javax.swing, javax.swing.JApplet ve javax.awt.event API paketlerini kullandik. Java.applet paketi appletleri olusturma, appletlere giris çikislari ve dosyalari applet içerisinde çalistirmayi saglar. java.awt pencere(window) ortamindaki tüm programlama siniflarini ve metotlarini içerir. Java swing (javax.swing) paketi yine ayni tür pencere(window) ortaminda programlama için gerekli girdi çikti grafik ortamlarini içerir. awt grafik ortamina göre daha zengin ve gelismis bir kolleksiyondur. Bu metotlari ilerideki bölümlerde daha detayli inceleyecegiz. java.io paketi javaya disaridan(dosya,ekran v.b) bilgi giris çikisini ayarlar.

Java metodlarinin tamaminin listesini (Ingilizce olarak) Java doküman kütüphanesindeki api alt gurubunda bulabilirsiniz. Java döküman kütühanesi [http://java.sun.com](http://java.sun.com/) adresinden çekilebilir.

Java dilinde en çok kullanilan API siniflarindan birisi Math sinifidir. Bu sinif(class) java.lang paketinde yer alir. Java.lang paketi java programi açilirken otomatik olarak çagirilir. Bütün diger paketler **import** deyimi kullanilarak programa ilave edilirler. Math sinifinda tanimlanan metotlarin bazilari Sekil 2.1 de listelenmistir.

Tablodaki x veya y degiskenleri **double** degisken türündendir. Math sinifinda(class) iki tane de sabit tanimlanmistir. Bu sabitlerden birisi **Math.PI** 3.14159265358979323846 (pi) sayisina esittir. Digeri **Math.E** 2.7182818284590452354 (e) sayisina esittir.

### Tablo 2.1 Math sinifinda(class) çalisan metotlardan bazilari

|  |  |  |
| --- | --- | --- |
| **Metot** | **Tanim** | **Örnek** |
| abs(x) | x degisekeninin mutlak degeri türkçesi : Mutlak | Math.abs(9.2) = 9.2  Math.abs(-9.2) = 9.2 |
| ceil(x) | x degiskenini bir üst tamsayiya dönüstürür (türkçesi : tavan) | Math.ceil(9.2) = 10  Math.ceil(-9.8) = -9 |
| floor(x) | x degiskenini bir alt tamsayiya dönüstürür. (türkçesi : taban) | Math.floor(9.2) = 9  Math.floor(-9.8) = -10 |
| cos(x) | x in trigonometrik cosünisü (x radyan cinsinden tanimlanmistir.) | Math.cos(1.0)=0.54030230568 |
| sin(x) | x in trigonometrik sinüsü (x radyan cinsinden tanimlanmistir.) | Math.sin(1.0)=0.8414709840709 |
| tan(x) | x in trigonometrik tanjanti (x radyan cinsinden tanimlanmistir.) | Math.tan(1.0)=1.557040724655 |
| acos(x) | x in trigonometrik cosünisünün tersi  (sonuç radyan cinsindendir) | Math.acos(0.54030230568)=1.0 |
| asin(x) | x in trigonometrik sinüsü (x radyan cinsinden tanimlanmistir.) | Math.asin(0.8414709840709)=1.0 |
| atan(x) atan2(x,y) | x in trigonometrik tanjanti (x radyan cinsinden tanimlanmistir.)  x,y noktanin x,ykoordinatlaridir | 4.0\*Math.atan(1.0)=3.14159265359  4.0\*Math.atan2(1.0,1.0)=3.14159265359 |
| exp(x) | e(2.7182818284590452354) nin x | Math.exp(1.0) =2.718281828459 |

Program 2.1 de Math sinifindaki metotlari kullanan Matematik1 sinifini görüyoruz.

|  |  |  |
| --- | --- | --- |
|  | inci kuvveti | Math.exp(2.0) =7.389056098931 |
| log(x) | e tabanina göre logaritma | Math.log(2.718281828459)=1.0 |
| pow(x,y) | x in y inci kuvveti | Math.pow(3,2)=9.0 |
| max(x,y) | x ve y nin büyük olani | Math.max(2.3,12.7)=12.7 |
| min(x,y) | x ve y nin küçük olani | Math.min(2.3,12.7)=2.3 |
| random() | Raslantisal sayi (0 ile 1 arasinda) | Math.random() = 0 ila bir arasinda her sayi çikabilir |

### Program 2.1 Math sinifinda(class) çalisan metotlardan bazilarini gösteren Matematik1.java programi

import javax.swing.JOptionPane; // giris çiki class matematik1SW

{

public static void main (String args[])

{

double x,y;

String s="";

x=Double.parseDouble(JOptionPane.showInputDialog ("Bir gerçek sayi giriniz : ")); y=Double.parseDouble(JOptionPane.showInputDialog ("Ikinci bir gerçek sayi giriniz : "));

s +="Statik Matematik kütüphanesi Math de \n"; s+="en cok kullanilan metodlar :\n" ;

// Math.abs(3.2)=3.2 Math.abs(-3.2)=3.2 s+="Math.abs("+x+") = "+Math.abs(x)+"\n";

// Math.ceil(9.2)=10 Math.ceil(-9.8)=-9

// Math.ceil(x) x den buyuk en kucuk tam say• ya yuvarlar s+="Math.ceil("+x+") = "+Math.ceil(x)+"\n";

// Math.floor(9.2)=9 Math.floor(-9.8)=-10

// Math.floor(x) x den kucuk en buyuk tam say• ya yuvarlar s+="Math.floor("+x+") = "+Math.floor(x)+"\n";

// Trigonometrik fonksiyonlar sin(x) cos(x) tan(x) x:radyan

// Math.PI pi sayisi

s += "Math.PI = "+Math.PI+"\n";

s += "Math.sin("+x+") = "+Math.cos(x)+"\n";

s += "Math.cos("+x+") = "+Math.sin(x)+"\n";

s += "Math.tan("+x+") = "+Math.tan(x)+"\n";

//Trigonometrik fonksiyonlar asin(y) acos(y) atan(y)

// asin,acos,atan : radyan

s += "Math.asin("+y+") = "+Math.acos(y)+"\n";

s += "Math.acos("+y+") = "+Math.asin(y)+"\n";

s += "Math.atan("+y+") = "+Math.atan(y)+"\n";

// Math.log(x) dogal logaritma (e) taban• na gore

// Math.E e sayisi = 2.718281828... s += "Math.E = "+Math.E+"\n";

s += "Math.log("+x+") = "+Math.log(x)+"\n";

// Math.pow(x,y) x in y inci kuvveti

s += "Math.pow("+x+","+y+" ) = "+Math.pow(x,y)+"\n";

// Math.exp(x) Math.E=e=2.718281828.. in x inci kuvveti s += "Math.exp("+x+" ) = "+Math.exp(x)+"\n";

// Math.sqrt(x) x in kare koku

s += "Math.sqrt("+x+" ) = "+Math.sqrt(x)+"\n";

// Maximum - minimum functions

// Math.max(x,y) Math.min(x,y)

s += "Math.max("+x+","+y+" ) = "+Math.max(x,y)+"\n";

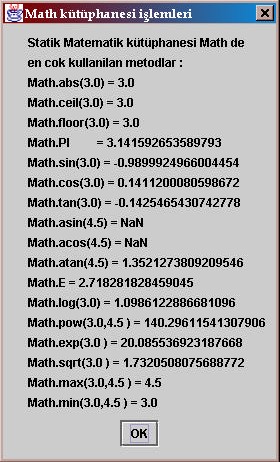
s += "Math.min("+x+","+y+" ) = "+Math.min(x,y)+"\n"; JOptionPane.showMessageDialog(null,s,

"Math kütüphanesi islemleri",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 02001.JPG



**Sekil 2.1 matematik1SW.java programinin çiktisi**

### METOTLAR

Metotlar java programlarinin ana parçalaridir. Metotlar siniflarin(class) içinde yer alan küçük program parçaciklaridir. Metotlarin çogunda degisken parametreler metotlar ve siniflar arasinda iletisimi saglarlar. Ayrica her metotun kendine özgü degiskenleri de vardir. Metot yapisinin ana sebebi programlari modüler hale getirmektir. Ayni zamanda ayni program parçaciginin tekrarlanmasini önlemeyi de saglar. Her metot çagrildigi program parçacigina belli bir degiskeni aktarabilir. Metotlarin tanimlarinda aktardiklari degisken türü tanimlanir. Eger metot hiçbir degisken aktarmiyorsa **void** sözcügüyle tanimlanir. Metotlarin genel tanimi asagidaki gibidir. Parantez içindeki terimler kullanilmiyabilir.

### Genel Metot tanimi

(public) (static) sinif degisken türü sinif ismi( sinif degisken girdi listesi)

{

sinif içinde geçerli degisken tanimlari Metotun ana gövdesi

return **metot çikti degiskeni**

}

Örnek olarak asagidaki metotu verebiliriz :

### Public static double metotornegi(int x,double y)

**{**

**double z = 3.5; double f ;**

**f = z\*x\*x+y; return f;**

**}**

Bu metotdaki x ve y gerçek (double) degiskenleri metotun girdi degiskenleridir. z ve f degiskenleri metotun yerel degiskenleridir ve bu metot disinda tanimlari yoktur. Metot f degiskeninin degerini çikti olarak metotun disina aktarmaktadir. Eger yukaridaki metot ayni sinifin içindeki baska bir metotda veya main metotunda asagidaki gibi bir örnekte kullanilirsa,

### double z = 2.5; double r ;

**int i = 3;**

**r = metotornegi(i,z); System.out.println(“r = “+r);**

**r = 34** sonucuna ulasilir. Simdi ayni metotun statik metot olarak bir appletin içinde kullanilmasini görelim.

### Program 2.2 : metotornegi1.java programinda kare statik metotunun kullanilmasi

import java.io.\*;

public class metotornegi1

{

// sayinin karesi static metotu

### public static double kare(double x)

**{**

**return x\*x;**

**}**

public static void main(String[] args) throws IOException

{

double sayi;

Text cin=new Text(); System.out.println("Bir tam sayi giriniz : "); sayi=cin.readDouble();

System.out.println("girilen sayinin karesi : "+**kare(sayi)**);

}

}

### Program 2.3 : metotornegi2.java programinda kare metotunun kullanilmasi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir

public class metotornegi2 extends Applet

{

double sayi=5; // Gercek degisken sayi

// sayinin karesi dinamik metotu

### public double kare(double x)

**{**

**return x\*x;**

**}**

public void paint(Graphics g)

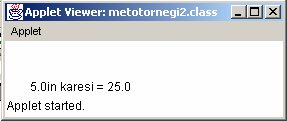
{

g.drawString(sayi+"nin karesi = "+**kare(sayi)**,25,50);

}

}

### 02002.JPG



**Sekil 2.2 metotornegi2.java programinda kare metotunun kullanilmasi**

Sekil 2.2.3 de tanimlanan **public double kare(double x)** metotu**, public void paint(Graphics g)** metotundaki

### g.drawString(sayi+"nin karesi = "+kare(sayi),25,50);

isleminde çagirilmistir. Bilgisayar **kare(sayi)** ifadesini gördügünde metotun içine gider, orada tanimlanan tüm islemleri yaptiktan sonra **return** kelimesinin karsisindaki islemleri çagirildigi yere geri iletir. Metotornegi1 java programinda çagirilan kare(sayi) metotunda sadece sayinin kendisiyle çarpimi return kelimesinin karsisina yazildigindan, **kare(sayi**) ifadesi **sayi\*sayi** ifadesiyle esdegerdir. Burada kare metotundandaki degisken adinin **sayi** degil **x** olarak verildigini de burada not edelim. Sayi degiskeni metotun girisinde x degiskenine yüklenir.

Gerekli islemler metotda yapildiktan sonra sadece metotun sonuçlari return deyimiyle metodun degisken türü olarak (bu metod için double ) geri döner.

### NESNE (OBJECT) TANIMI VE METOTLARDA KULLANIMI

Birinci bölümde temel degisken türlerini (double,int,boolean v.b.) nasil tanimladigimizi görmüstük. Nesne tipi programlamanin en önemli özelligi kendi degisken türlerimizi yaratabilmemizdir. Ilerideki bölümlerde kendi degisken türlerimizi sinif (class) yapisini kullanarak nasil yaratacagimizi daha detayli olarak inceleyecegiz. Bu bölümde nesnelerin program içindeki tanimlanmasina göz atalim. Daha önce tanimladigimiz bazi applet programlarinda nesne tanimlari zaten geçmisti. Örnegin daha önceki programlarimizda bilgi okutmak için kullandigimiz

### Text cin=new Text();

Terimi Text sinifindan(class) cin nesnesini tanimlar. Nesne(object) tanimi yaparken ayni metotlarda oldugu gibi degisken veya nesneleri girdi olarak tanimlamamiz mümkündür.

Diger bir örnek olarak yine daha önceki örnek problemlerde kullandigimiz Label sinifindan(class) kutubasligi nesnesinin tanimini ve TextField sinifindan(class) kutugirdisi nesnesini gösterebiliriz.

### Label kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi kutubasligi=new Label("Ogrencinin notunu giriniz (A B C.. : ");

**TextField kutugirdisi;//Textfield sinifi degiskeni (nesnesi) kutugirdisi kutugirdisi=new TextField(5);**

Bu örneklerde de görüldügü gibi Nesne tanimi su sekilde yapilmaktadir:

### Sinif (class) isminesne(object) ismi;

**nesne(object) ismi = new Sinif (class) ismi(degisken veya nesne girdi listesi)**

Nesne tanimini daha iyi anlamak için metotornegi2 sinifini yazdigimiz programi biraz degistirerek metotornegi3 ve metotornegi3a siniflarini olusturalim ve kare metotumuzu metotornegi3a sinifina yerlestirelim.

### Program 2.4 : metotornegi.java programinda kare metotunun kullanilmasi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir class **metotornegi3a**

{

// sayinin karesi dinamik metotu

### public double kare(double x)

**{**

**return x\*x;**

**}**

}

public class metotornegi3 extends Applet

{

double sayi=5; // Gercek degisken sayi **metotornegi3a nesne1=new metotornegi3a();** public void paint(Graphics g)

{

g.drawString(sayi+"nin karesi = "+**nesne1.kare(sayi)**,25,50);

}

}

Metotornegi3.java programinda iki tane sinif(class) yaratilmistir. Ilk sinif metotornegi3a da kare metotu tanimlanmistir. Ikinci sinif(class) metotornegi3 de ise metotornegi3a sinifi nesne1 nesnesi tanimlanmis, ve kare metotu nesne1 nesnesi üzerinden **nesne1.kare(sayi)** olarak çagirilmistir. Bu metot nesne1 nesnesi için bilgisayar belleginde bir yer ayirir. Kare metotunu çagirirken bu yerin adresini kullanir. Metotornegi2.java programinda kare(sayi) dogrudan kullanilmisti. Çunku kare metotu ayni sinifin(class) içinde tanimlanmisti. Metotornegi3.java programinda ise kare metotu ayri bir sinif(class) ta tanimlandigindan yerinin tanimlanmasi gerekir. Bu bizim kendi odamizin yatak odasini tanimlarken sadece yatak odasi dememizle baska bir evin yatak odasini tanimlarken Ahmet beyin evinin yatak odasi dememiz gibidir. Metotlar statik olarak da tanimlanabilirler. Statik olarak tanimlanan metotlar nesne kullanilmadan direk olarak sinif(class) adlari kullanilarak çagirilabilirler.

### Program 2.5 : metotornegi4.java programinda static kare metotunun kullanilmasi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir class metotornegi4a

{

// sayinin karesi dinamik metotu public static double kare(double x)

{

return x\*x;

}

}

public class metotornegi4 extends Applet

{

double sayi=5; // Gercek degisken sayi public void paint(Graphics g)

{

g.drawString(sayi+"nin karesi = "+ metotornegi4a.kare(sayi),25,50);

}

}

Görüldügü gibi metotornegi4 sinifinda(class) kare metotu metotornegi4a.kare(sayi) olarak tanimlanmis ve ek bir nesne adresi kullanilmamistir. Static olarak tanimlanan metotlarin kendikendine yeterli metotlar olamasi gerekir. Yani içinde bulundugu sinifla veya baska siniflarla direk olarak veri alisverisi olamaz. Girdileri sadece girdi parantezleri () arasinda verilen degiskenlerdir.dinamik metotlar ise kendi siniflari ve diger siniflarla degisik yollardan bilgi ve degisken aktarimi yapabilirler

Metotornegi3.java ve Metotornegi4.java programlarinda iki sinif(class) ayni dosyada yer almisti. Her sinifi ayri bir dosyada da tanimlamak mumkündür. Bu yapildiginda diger dosya import(ithal) terimiyle diger programa aktarilir. Sekil 2.6 ve 2.7 de Sekil 2.5 de verilen programin iki ayri dosyada yazilimini görüyoruz.

### Program 2.6 : metotornegi6.java programinda kare metotunun tanimlanmasi

class **metotornegi6**

{

// sayinin karesi dinamik metotu

### public double kare(double x)

**{**

**return x\*x;**

**}**

}

### Program 2.7 : metotornegi7.java programinda kare metotunun metotornegi6 sinifi x nesnesi üzerinden kullanilisi.

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir import metotornegi6; // metotornegi6 sinifini çagir

public class metotornegi7 extends Applet

{

double sayi=5; // Gercek degisken sayi metotornegi6 x=new metotornegi6(); public void paint(Graphics g)

{

g.drawString(sayi+"nin karesi = "+ x.kare(sayi),25,50);

}

}

### METOTLARIN KENDI KENDINI ÇAGIRMASI (RECURSION)

Bazi problemlerin çözümünde bir metotun kendi kendini çagirmasi yararli olabilir. Java bir metotun kendi kendini çagirmasina izin verir. Burada dikkatli olunmasi gereken nokta bu kendi kendine çagirilma döngüsünün sonsuza kadar sürmesinin bir kontrol yapisi kullanilarak engellenmesidir. Kendi kendini çagiran programlarda mutlaka bir döngü çikis sarti tanimlanmalidir.

Daha önceki alistirmalarimizda faktoriyel sinifi tanimlanmisti. Önce bu programdaki faktoriyel hesabini ayri metot haline dönüstürelim. Program 2.8 de daha önce de gördügümüz faktoriyeli ana metotumuz olan main içinde hesaplayan faktoriyel programini görmekteyiz. Program 2.9 da faktoriyel hesabi for döngüsü kullanilarak faktoriyel metotonda gerçeklestirilmektedir.

### Program 2.8 : Metot kullanilmadan (main metotunun içinde) faktoriyel hesaplayan faktoriyel.java programi

import java.io.\*; import Text; class faktoriyel

{

public static void main(String args[]) throws IOException

{

int faktoriyel=1; Text cin=new Text(); int faktoriyelsayi;

System.out.print("Bir tamsayi giriniz : "); faktoriyelsayi=cin.readInt();

for(int sayi=1;sayi<=faktoriyelsayi;sayi++)

{ faktoriyel\*=sayi;}

System.out.println(faktoriyelsayi + " faktoriyel: " + faktoriyel);

}

}

degisik giris çikis opsiyonlarini biraz daha anlamak için ayni programin JOPtionPane kullanilarak yazilmis esdegerini de verelim :

### Program 2.9 : Metot kullanilmadan (main metotunun içinde) faktoriyel hesaplayan faktoriyeSW.java programi

import javax.swing.JOptionPane; class faktoriyelSW

{

public static void main(String args[])

{

int faktoriyel=1; int faktoriyelsayi;

faktoriyelsayi=Integer.parseInt(JOptionPane.showInputDialog("Bir tam sayi giriniz : ")); for(int sayi=1;sayi<=faktoriyelsayi;sayi++)

{ faktoriyel\*=sayi;}

String s=faktoriyelsayi+" faktoriyel: "+faktoriyel; JOptionPane.showMessageDialog(null,s, "Metod örnegi faktoriyelSW.java ", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

Simdi ayni programin faktoriyel metodu ayri yazilmis fakat yine for döngüsü kullanan sekillerine göz atalim. Program 2.10 ve 2.11 de birbirinin ayni olu, program 2.10 da çikti System.out.println deyimiyle alinirken 2.11 de çikti JOptionPane.showMessageDialog metodu kullanilarak aktarilmistir.

### Program 2.10 : for döngülü faktoriyel metotunu kullarak faktoriyel hesaplayan faktoriyel1.java programi

import java.io.\*; import Text; class faktoriyel1

{

public static long faktoriyel(int x)

{

long faktoriyel=1;

for(int sayi=1;sayi<=x;sayi++)

{ faktoriyel\*=sayi;} return faktoriyel;

}

public static void main(String args[]) throws IOException

{

Text cin=new Text(); int faktoriyelsayi;

System.out.println("Bir tamsayi giriniz:"); faktoriyelsayi=cin.readInt();

System.out.println(faktoriyelsayi+" faktoriyel:" + faktoriyel(faktoriyelsayi));

}

### Program 2.11 : for döngülü faktoriyel metotunu kullarak faktoriyel hesaplayan faktoriyel1SW.java programi

**(JoptionPane girdi çikti sistemi kullanarak hesapliyor)**

import javax.swing.JOptionPane;

class faktoriyel1SW

{

public static long faktoriyel(int x)

{ long faktoriyel=1;

for(int sayi=1;sayi<=x;sayi++)

{ faktoriyel\*=sayi;} return faktoriyel;

}

public static void main(String args[])

{ int faktoriyelsayi; faktoriyelsayi=Integer.parseInt(

JOptionPane.showInputDialog("Bir tam sayi giriniz : "));

String s=faktoriyelsayi+" faktoriyel: "+faktoriyel(faktoriyelsayi);

JOptionPane.showMessageDialog(null,s,"Metod örnegi faktoriyel1SW.java ",JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

Program 2.12 ve 2.13 de faktoriyeli kendi kendini çagiran(recursive) faktoriyel metotunu kullanarak hesaplayalim.

### Program 2.12 : kendi kendini çagiran (recursive) faktoriyel metotunu kullarak faktoriyel hesaplayan faktoriyel2.java programi

import java.io.\*; import Text; **class faktoriyel2**

### {

**public static long faktoriyel(int x)**

**{**

**if( x <= 1 ) return 1; else**

**return x \* faktoriyel( x – 1);**

**}**

public static void main(String args[]) throws IOException

{

DataInputStream cin=new DataInputStream(System.in); int faktoriyelsayi;

System.out.println("Bir tamsayi giriniz:"); faktoriyelsayi=Text.readInt(cin);

System.out.println(faktoriyelsayi+" faktoriyel:" + faktoriyel(faktoriyelsayi));

}

}

Bir tamsayi giriniz: 4 4 faktoriyel:24

### Program 2.13 : kendi kendini çagiran (recursive) faktoriyel metotunu kullarak faktoriyel hesaplayan Faktoriyel2SW.java programi (JOPtionPage girdi/çikti)

import javax.swing.JOptionPane; class faktoriyel2SW

{

public static long faktoriyel(int x)

{

if( x <= 1 ) return 1;

else

return x \* faktoriyel( x - 1);

}

public static void main(String args[])

{ int faktoriyelsayi;

faktoriyelsayi=Integer.parseInt(JOptionPane.showInputDialog("Bir tam sayi giriniz : ")); String s=faktoriyelsayi+" faktoriyel: "+faktoriyel(faktoriyelsayi);

JoptionPane.showMessageDialog(null,s, "kendi kendini çagiran (recursive) Metod örnegi faktoriyel2SW.java

",

JoptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 02003.JPG



**Sekil 2.3 : kendi kendini çagiran (recursive) faktoriyel metotunu kullarak faktoriyel hesaplayan Faktoriyel2SW.java programinin sonuçlarinin JOPtionPage çiktisi olarak görülmesi**

Program 2.132veya 2.13’ü çalistirdigimizda faktoriyel hesabi su sekilde yapilacaktir : 5! = 5 \* 4! = 4 \* 3! = 3 \* 2! = 2 \* 1! = 1

faktoriyel metotu 1 e ulastiginda tekrar kendisini çagirmiyacagindan geriye dogru yaptigi hesaplari göndermeye basliyacaktir.

1 = (2\*1!(=1)) = 2 = (3 \* 2!(=2)) = 6= (4\*3!(=6))=24= (5\*4!(24))=120= 5!(=120)

sonuç 120 olarak ana programa gönderilecektir. Kendi kendini çagiran metodlar mutlak bir gereksinim olmadikça tercih edilmez. Bu tür programlamada hem sonsuz döngülerin görülmesi daha güç olabilir, hemde for, while gibi döngüler kullanilarak programlamaya göre programlar daha yavas çalisir. Gerçek uygulamalarda kendi kendini çagiran programlar genellikle bir döngü yapisi kurmadan çok tek kerelik çagirmalari gerçeklestirmek için kullanilirlar.

### AYNI ADLI METOTLARIN BIR ARADA KULLANILMASI (OVERLOADING)

Java ayni adli metotlarin ayni sinif içerisinda kullanilmasina izin verir. Ayni sinifta(class) kullanilan metotlarin girdi degiskenlerinin degisken türlerinin veya degisken sayilarinin farkli olmasi gerekir. Java ayni isimli ve ayni degisken türlü iki metotu birbirinden ayiramaz. Örnegin

### Public double Bmetodu(double Adegiskeni)

Ve

### Public double Bmetodu(double Bdegiskeni)

Veya

### Public int Bmetodu(double Cdegiskeni)

Java tarafindan ayirt edilemez , Fakat

### Public double Cmetodu(double Adegiskeni)

Ve

### Public double Cmetotu(int Bdegiskeni)

veya

### Public double Cmetotu(double Adegiskeni, double Bdegiskeni)

Java tarafindan ayirt edilir ve birlikte ayni sinifin (class) içinde yer alabilir. Eger birbirinin tam olarak esiti metotlar degisik siniflarda yer aliyorsa bunun herhangi bir sakincasi yoktur. Program 2.11'de ayniisimliikimetot.java programi görülmektedir. Bu programda daha önceki programlarda da kullandigimiz kare metotunu int ve double girdi degiskenleriyle iki kere ayni sinifin içerisinde tanimlayacagiz.

### Program 2.14 : ayniisimliikimetot.java programi

import java.awt.Graphics; import java.applet.Applet;

public class ayniisimliikimetot extends Applet

{

// void tipi paint metotu public void paint( Graphics g)

{

g.drawString(“ 7 nin karesi = “+kare(7),25,25); g.drawString( 7.5 un karesi = “+kare(7.5),25,40);

}

### // int tipi kare metotu int kare(int x)

**{**

**return x\*x;**

**}**

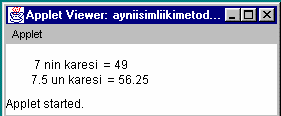
/**/ double tipi kare metotu double kare(double x)**

**{**

**return x\*x;**

**}**

**}**



**02004.JPG**

**Sekil 2.4 ayniisimliikimetot.html programiyla ayniisimliikimetot.class in gösterilmesi**

Bu programda parantez içerisinde 7 (tamsayi) verildiginde java tamsayi metotu kare(int x), 7.5(gerçek sayi) verildiginde gerçeksayi metotu kare(double x) i çagirir.

Benzer bir örnegi swing applet olarak verelim :

### Program 2.15 : H3AL9\_2000 programi

import java.awt.Container; import javax.swing.\*;

public class H3AL9\_2000 extends JApplet

{

JTextArea ciktiAlani;

public void init()

{

ciktiAlani=new JTextArea(); Container c=getContentPane(); c.add(ciktiAlani);

int x1=7; double x2=7.0;

ciktiAlani.setText(" exp ("+x1+") (int metod girdisi ) = "+exp(x1)+"\n"); ciktiAlani.append(" exp ("+x2+") (double metod girdisi) = "+exp(x2)+"\n"); x1=5;

x2=5.0;

ciktiAlani.append(" exp ("+x1+") (int metod girdisi ) = "+exp(x1)+"\n"); ciktiAlani.append(" exp ("+x2+") (double metod girdisi) = "+exp(x2)+"\n");

}

// double girisli exp metodu public double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1; for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

// int girisli exp metodu public double exp(int x)

{

double exponent=1.0; for(int i=1;i<=x;i++)

{

exponent\*=Math.E;

}

return exponent;

}

}

### 02005.JPG



**Sekil 2.5 Gerçek ve tamsayi degiskenleri çagiran ayni adli metodlarin çagirilmasi örnegi**

### METOT (METHOD) VE SINIF(CLASS) DEGISKENLERI

Metotlarin içinde kullanilan degiskenler sadece metotlara aittir. O metotun disinda tanimlari yoktur. Eger degiskenler metotlarin disinda sinif(class) degiskenleri olarak tanimlanirsa metota da aynen aktarilirlar. Sinif degiskeniyle ayni isimde bir degisken mettotta da tanimlanmissa bu degisken metotun içinde kullanilan metotun degiskenidir

### Program 2.16 : metotvesinifdeg.java programinda metot degiskenlerinin sinifa aktarilmasi

import java.awt.Graphics; import java.applet.Applet;

public class metotvesinifdeg extends Applet

{

//buradaki degiskenler tüm sinifa aittir double x ;

int y ;

void metot1( Graphics g) //metota hiç bir degisken girmiyor

{ x=5.5; y=6;

g.drawString("metot1 in ici : x = "+x+" y = "+y,25,40);

}

public void paint(Graphics g)

{

x = 2.5;

y = 3;

g.drawString("metot1 in disi : x = "+x+" y = "+y,25,25);

### metot1(g);

g.drawString("metot1 in disi : x = "+x+" y = "+y,25,55);

}

}

### metotvesinifdeg.java programinin sonuçlari

Metot1 in disi : x=2.5 y=3 Metot1 in ici : x=5.5 y=6 Metot1 in disi : x=5.5 y=6

### Program 2.17 : metotvesinifdeg1.java programinda metot degiskenlerinin sinifa aktarilmasi

import java.awt.Graphics; import java.applet.Applet;

public class metotvesinifdeg1 extends Applet

{

//buradaki degiskenler tüm sinifa aittir double x ;

int y ;

void metot1( Graphics g) //metota hi‡ bir de§iŸken girmiyor

{

double x; int y; x=5.5; y=6;

g.drawString("metot1 in ici : x = "+x+" y = "+y,25,40);

}

public void paint(Graphics g)

{

x = 2.5;

y = 3;

g.drawString("metot1 in disi : x = "+x+" y = "+y,25,25); metot1(g);

g.drawString("metot1 in disi : x = "+x+" y = "+y,25,55);

}

}

### metotvesinifdeg1.java programinin sonuçlari

Metot1 in disi : x=2.5 y=3 Metot1 in ici : x=5.5 y=6 Metot1 in disi : x=2.5 y=3

Program 2.16'daki metotvesinifdeg sinifindaki metot1 metotunun içinde degistirilen x ve y degiskenleri tüm tüm sinifa(class) aittir. Bu yüzden paint metotundaki drawString metotu tekrar çagirildiginda x ve y nin degerlerinin degistigini görürüz.

Program 2.17 deki metotvesinifdeg1 sinifindaki metot1 metotunun içinde ise yerel degiskenler x ve y tanimlanmistir. Bu yüzden metot1 in içindeki x ve y degiskenlerinin degismesi paint metotundaki drawString metotu tekrar çagirildiginda x ve y nin degerlerinin degismedigi görülür.

Nesne tanimi yapildiginda Bilgisayar bu nesne için bir adres yaratir. Yeni adres yarat komutu **new** komutudur.

### Pogram 2.18 : Point1.java programda nokta1 ve nokta2 nesnelerinin yaratilmasi ve adres paylasimi.

// java kutuphanelerinde tanimli bir sinifin cagirilip

// nesne olarak tanimlanmasi ve adres paylasimi import java.io.\*; //java girdi cikti sinifini cagir import java.awt.Point; //Point sinifini cagir

class Point1

{

public static void main(String args[])

{

//Point sinifi x,y koordinatli bir nokta tanimlar

### Point nokta1,nokta2; nokta1=new Point(100,100); nokta2=nokta1;

// nokta1 ve nokta2 bilgisayarda ayni

// adres kutusunu paylasiyor

System.out.println("nokta1 ve nokta2 ayni degerleri tasiyor"); System.out.println("ve ayni bilgisayar adresini paylasiyor"); System.out.println("nokta 1 : "+nokta1.x+" , "+nokta1.y); System.out.println("nokta 2 : "+nokta2.x+" , "+nokta2.y); **nokta1.x=200;**

### nokta1.y=200;

System.out.println("nokta1 in degeri degistirildi"); System.out.println("nokta 1 : "+nokta1.x+" , "**+nokta1.y**); System.out.println("nokta 2 : "+nokta2.x+" , "+**nokta2.y**);

// nokta1 ve nokta2 nin degerleri ayni fakat bilgisayarda

// degisik adreslerde tanimlanmis **nokta1=new Point(100,100); nokta2=new Point(100,100);**

System.out.println("nokta1 ve nokta2 ayni degerleri tasiyor"); System.out.println("fakat ayni bilgisayar adresini paylasmiyor"); System.out.println("nokta 1 : "+nokta1.x+" , "+**nokta1.y**); System.out.println("nokta 2 : "+nokta2.x+" , "+**nokta2.y**); **nokta1.x=200;**

### nokta1.y=200;

System.out.println("nokta1 in degeri degistirildi"); System.out.println("nokta 1 : "+nokta1.x+" , "+nokta1.y); System.out.println("nokta 2 : "+nokta2.x+" , "+nokta2.y);

}

}

Point1.java da verilen

### Point nokta1,nokta2; nokta1=new Point(100,100); nokta2=nokta1;

taniminda Point sinifindan(class) nokta1 ve nokta2 degiskenleri tanimlanmis, nokta1 degiskeni için yeni(new) adres tanimlanmis, nokta2 degiskenine de ayni adresi (nokta1'in adresini) kullanmasi soylenmistir. Bu yüzden programda nokta1 e yeni deger verildiginde nokta2 nin degeri de ayni degeri alir.

daha sonra kullanilan

### nokta1=new Point(100,100); nokta2=new Point(100,100);

deyimi ile ise nokta1 ve nokta2 için iki ayri adres yaratilmistir. Bu yüzden nokta1 degistiginde nokta2 eski degerinde kalir.

***Ayni adresin iki degisken tarafindan kullanilmasi ancak gerektigi zaman ve çok dikkat sarfederek yapilmalidir. Gereksiz yere kullanilmasindan kesin olarak kaçinilmalidir. Program güvenlik ve kontrol problemleri yaratabildigi gibi, istemedigimiz degisken degerlerinin silinmesi sonucunu da verebilir.***

### ALISTIRMALAR

1. Math kütüphanesindeki Math.random() metotunu kullanarak **static zar** metotunu yaziniz. Bu metotu H4A1a sinifinda(class) tanimlayiniz.

Not : 1-den 6 ya kadar tesadüfi (random) sayilari 1+(int)(Math.random()\*6) formülü ile hesaplayabiliriz.

### Program 2.19 H4A1a.java ve static int zar metodu

public class H4A1a

{

public static int zar()

{

return 1+(int)(Math.random()\*6);

}

}

H4A1b sinifinda(class) java konsol ana metotu (public void main) yazarak zar degerini yazdiriniz.

### Program 2.20 H4A1b.java

import java.io.\*; // giris cikis import H4A1a;

class H4A1b

{

public static void main (String args[]) throws IOException

{

System.out.println("zar degeri : "+H4A1a.zar());

}

}

H4A1c sinifinda java konsol ana metotunda (public void main) metotunda yüz(100) kere zar attirarak ortalamasini hesaplayiniz.

Not : for veya while döngüsü kullanabilirsiniz.

### Program 2.21 H4A1c.java

import java.io.\*; // giris cikis import H4A1a;

class H4A1c

{

//yuz zarin ortalamasi

public static void main (String args[]) throws IOException

{

double ortalama; double toplam=0;

for(int sayi=0;sayi<100;sayi++)

{

toplam+=H4A1a.zar();

}

ortalama=toplam/100.0;

System.out.println("100 zarin ortalamasi = "+ortalama);

}

}

H4A1d sinifinda java konsol ana metotunda (public void main) bin(1000) kere zar attirarak her rakamin kac kere geldigini bulunuz ve yazdiriniz.

Not : switch – case deyimi bu tür programlar için idealdir

### Program 2.22 H4A1d.java

import java.io.\*; // giris cikis import H4A1a;

public class H4A1d

{

//bin zar atiminda her zar kac kere gelir

public static void main (String args[]) throws IOException

{

int zardegeri;

int zar1,zar2,zar3,zar4,zar5,zar6; int zaratimsayisi=0;

zar1=0; zar2=0; zar3=0; zar4=0; zar5=0; zar6=0;

while(zaratimsayisi<1000)

{

zardegeri=H4A1a.zar(); switch(zardegeri)

{

case 1: zar1++; break; case 2: zar2++; break; case 3: zar3++; break; case 4: zar4++; break; case 5: zar5++; break; case 6: zar6++; break;

}

zaratimsayisi++;

}

System.out.println("toplam zar 1 sayisi : "+zar1); System.out.println("toplam zar 2 sayisi : "+zar2); System.out.println("toplam zar 3 sayisi : "+zar3); System.out.println("toplam zar 4 sayisi : "+zar4); System.out.println("toplam zar 5 sayisi : "+zar5);

System.out.println("toplam zar 6 sayisi : "+zar6);

}

}

toplam zar 1 sayisi : 169

toplam zar 2 sayisi : 165

toplam zar 3 sayisi : 165

toplam zar 4 sayisi : 146

toplam zar 5 sayisi : 178

toplam zar 6 sayisi : 177

1. Math kütüphanesindeki Math.random() metotunu kullanarak **static zar** metotunu yazilmistir. Bu metot H4A1a sinifinda(class) Problem 1 de tanimlanmistir.

H4A1bSW sinifinda(class) java konsol ana metotu (public void main) yazarak zar degerini yazdiriniz. Yazdirma islemini java swing sinifindan JoptionPane sinifini kullanarak yaptiriniz.

### Program 2.23 H4A1bSW.java

import javax.swing.JOptionPane; // giris cikis import H4A1a;

class H4A1bSW

{

public static void main (String args[])

{

JOptionPane.showMessageDialog(null,"atilan zar = "+H4A1a.zar()

,"sinif örnegi, zar atimi ",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

H4A1cSW sinifinda java konsol ana metotunda (public void main) metotunda yüz(100) kere zar attirarak ortalamasini hesaplayiniz. Yazdirma islemini java swing sinifindan JoptionPane sinifini kullanarak yaptiriniz. Not : for veya while döngüsü kullanabilirsiniz.

### Program 2.24 H4A1cSW.java

import javax.swing.JOptionPane; // giris cikis import H4A1a;

class H4A1cSW

{

//yüz zarin ortalamasi

public static void main (String args[])

{

double ortalama; double toplam=0;

for(int sayi=0;sayi<100;sayi++)

{

toplam+=H4A1a.zar();

}

ortalama=toplam/100.0;

JOptionPane.showMessageDialog(null,"100 zarin ortalamasi = "+ortalama

,"sinif örnegi, 100 kez zar atimi ortalamasi H4A1c", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

H4A1dSW sinifinda java konsol ana metotunda (public void main) bin(1000) kere zar attirarak her rakamin kac kere geldigini bulunuz ve yazdiriniz. Yazdirma islemini java swing sinifindan JoptionPane sinifini kullanarak yaptiriniz.

Not : switch – case deyimi bu tür programlar için idealdir

### Program 2.25 H4A1dSW.java

import javax.swing.JOptionPane; // giris cikis import H4A1a;

public class H4A1dSW

{

//bin zar atiminda her zar kac kere gelir public static void main (String args[])

{

int zardegeri;

int zar1,zar2,zar3,zar4,zar5,zar6; int zaratimsayisi=0;

zar1=0; zar2=0; zar3=0; zar4=0; zar5=0; zar6=0;

while(zaratimsayisi<1000)

{

zardegeri=H4A1a.zar(); switch(zardegeri)

{

case 1: zar1++; break; case 2: zar2++; break; case 3: zar3++; break; case 4: zar4++; break; case 5: zar5++; break; case 6: zar6++; break;

}

zaratimsayisi++;

}

String s="";

s+="toplam zar 1 sayisi : "+zar1+"\n"; s+="toplam zar 2 sayisi : "+zar2+"\n"; s+="toplam zar 3 sayisi : "+zar3+"\n"; s+="toplam zar 4 sayisi : "+zar4+"\n"; s+="toplam zar 5 sayisi : "+zar5+"\n"; s+="toplam zar 6 sayisi : "+zar6+"\n"; JOptionPane.showMessageDialog(null,s

,"sinif örnegi, 1000 kez zar atimi gelen zarlar H4A1cdSW", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

### 02006.JPG



**3.**

 zarApplet.java programini incele ve calistir.

 ZarApplet.java programindaki zar metotunu H4A2b.java sinifina yerlestir. ZarApplet.java programindaki zar metotunu sil ve import H4A2b; deyimini ekle.

 ZarApplet.java programini H4A2c olarak degistir. sinifindan yeni nesne z yi H42c.java programina ekle ve Paint methodunda çagirilan zar metotunda gerekli degisiklikleri yap. Programi yeni sekliyle calistir.

### Program 2.26 : ZarApplet.java, zar atma metot örnegi

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class zarApplet extends Applet implements ActionListener

{

int toplamzaratisi=0; int toplam=0;

TextField ilkzar,ikincizar; Button salla; //zar atma dügmesi int zar1,zar2;

public static int zar()

{

return 1+(int)(Math.random()\*6);

}

public void init()

{

//programi baslat

ilkzar=new TextField(10); // Textfield sinifi ilkzar nesnesini yarat add(ilkzar); // ilk zar nesnesini pencereye ekle

ikincizar=new TextField(10);// Textfield sinifi ikincizar nesnesini yarat add(ikincizar); // ikinci zar nesnesini pencereye ekle

salla=new Button("Zari salla ve at"); add(salla); salla.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

zar1=zar(); zar2=zar();

ilkzar.setText(Integer.toString(zar1)); ikincizar.setText(Integer.toString(zar2)); toplam+=(zar1+zar2);

toplamzaratisi++; repaint();

}

public void paint( Graphics g)

{

g.drawString("toplam = "+toplam+" Atilan zar sayisi = "+toplamzaratisi,25,50);

}

}

public class H4A2b

{

public static int zar()

{

return 1+(int)(Math.random()\*6);

}

}

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*; import H4A2b;

public class H4A2c extends Applet implements ActionListener

{

int toplamzaratisi=0; int toplam=0;

TextField ilkzar,ikincizar; Button salla; //zar atma d• gmesi int zar1,zar2;

H4A2b z=new H4A2b(); public void init()

{

//programi baslat

ilkzar=new TextField(10); // Textfield sinifi ilkzar nesnesini yarat add(ilkzar); // ilk zar nesnesini pencereye ekle

ikincizar=new TextField(10);// Textfield sinifi ikincizar nesnesini yarat add(ikincizar); // ikinci zar nesnesini pencereye ekle

salla=new Button("Zari salla ve at"); add(salla); salla.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

zar1=z.zar();

zar2=z.zar(); ilkzar.setText(Integer.toString(zar1)); ikincizar.setText(Integer.toString(zar2)); toplam+=(zar1+zar2);

toplamzaratisi++; repaint();

}

public void paint( Graphics g)

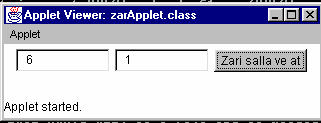
{

g.drawString("toplam = "+toplam+" Atilan zar sayisi = "+toplamzaratisi,25,50);

}

}

### 02007.JPG



**Sekil 2.7 ZarApplet.html çiktisi**

1. **faktoriyeltesti1.java** programini inceleyiniz, ve applet ortaminda çalistiriniz.

### Program 2.27 FaktoriyelTesti

import java.awt.Graphics; import java.applet.Applet;

public class faktoriyeltesti extends Applet

{

public void paint(Graphics g)

{

int y=25;

for(long i=0;i<=10;i++)

{

g.drawString(i+"! = "+faktoriyel(i),25,y); y+=15;

}

}

### public long faktoriyel(long sayi)

**{**

**if(sayi <=1) return 1;**

**else return sayi\*faktoriyel(sayi - 1);**

**}**

}

1. **fibonachitesti1.java** programini inceleyiniz ve applet ortaminda çalistiriniz.

### Program 2.28 Fibonachitesti1.java

import java.awt.Graphics; import java.applet.Applet;

public class fibonachitesti1 extends Applet

{

public void paint(Graphics g)

{

int y=25;

for(long i=0;i<=15;i++)

{

g.drawString("Fibonachi("+i+") = "+fibonachi(i),25,y); y+=15;

}

}

### public long fibonachi(long sayi)

**{**

**if(sayi <=0) return 0;**

**else if(sayi ==1) return 1;**

**else return fibonachi(sayi - 1)+fibonachi(sayi - 2);**

**}**

}

### fibonnachitesti1SWA.java swing applet programini inceleyiniz ve appletviewer ile çalistiriniz Program 2.29 Fibobachi testi, swing applet programi fibonnachitesti1SWA.java

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class fibonnachitesti1SWA extends JApplet

{

private JTextArea cikti;

public void init()

{

Container c=getContentPane(); c.setLayout(new FlowLayout()); cikti=new JTextArea(" "); c.add(cikti);

cikti.setBackground(c.getBackground()); String s="";

for(long i=0;i<=15;i++)

{

s+="Fibonachi("+i+") = "+fibonachi(i)+"\n";

}

cikti.setText(s);

}

### public long fibonachi(long sayi)

**{**

**if(sayi <=0) return 0;**

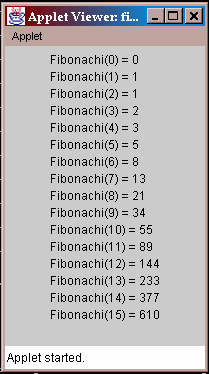
**else if(sayi ==1) return 1;**

**else return fibonachi(sayi - 1)+fibonachi(sayi - 2);**

**}**

}

### 02008.JPG



**Sekil 2.8 fibonnachitesti1SWA swing applet çiktisi**

1. **fibonnachitesti1SWF.java swing** Frame programini inceleyiniz ve java komutu ile çalis**tiriniz Program 2.30 Fibobachi testi, swing applet programi fibonnachitesti1SWA.java**

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class fibonnachitesti1SWF extends JFrame

{

private JTextArea cikti;

public fibonnachitesti1SWF()

{

super("Fibonacchi Testi 1 Swing Frame"); Container c=getContentPane(); c.setLayout(new FlowLayout()); cikti=new JTextArea(" ");

c.add(cikti); cikti.setBackground(c.getBackground()); String s="";

for(long i=0;i<=15;i++)

{

s+="Fibonachi("+i+") = "+fibonachi(i)+"\n";

}

cikti.setText(s);

}

public long fibonachi(long sayi)

{

if(sayi <=0) return 0;

else if(sayi ==1) return 1;

else return fibonachi(sayi - 1)+fibonachi(sayi - 2);

}

public static void main(String[] args)

{

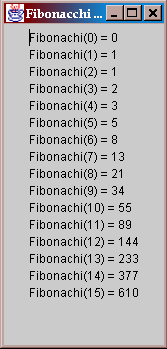
fibonnachitesti1SWF pencere= new fibonnachitesti1SWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(150,350);

pencere.setVisible(true);

}

}

### 02009.JPG



**Sekil 2.9 fibonnachitesti1SWF çiktisi**

1. **fibonachitesti2.java** programini inceleyiniz ve applet ortaminda çalistiriniz.

### Problem 2.31 fibonachitesti2.java programi

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class fibonachitesti2 extends Applet implements ActionListener

{

Label rakametiketi,sonucetiketi; TextField rakam,sonuc;

public void init()

{

rakametiketi=new Label("Bir tamsayi giriniz : "); rakam=new TextField(10); rakam.addActionListener(this); sonucetiketi=new Label("Fibonachi degeri : "); sonuc=new TextField(10); sonuc.setEditable(false);

add(rakametiketi); add(rakam); add(sonucetiketi); add(sonuc);

}

public void actionPerformed( ActionEvent e)

{

long sayi;

long fibonachisayisi; sayi=Long.parseLong(rakam.getText()); showStatus("Fibonachi hesaplan• yor...."); fibonachisayisi=fibonachi(sayi); showStatus("Fibonachi hesaplandi...."); sonuc.setText(Long.toString(fibonachisayisi));

}

public long fibonachi(long sayi)

{

if(sayi <=0) return 0;

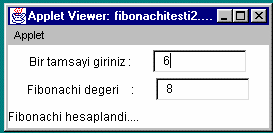
else if(sayi ==1) return 1;

else return fibonachi(sayi - 1)+fibonachi(sayi - 2);

}

}

### 02010.JPG



**Sekil 2.10 Fibonachi testi 2 çiktisi (AWT Aplet)**

1. **fibonachitesti2SWA.java** programinda bir önceki programda yazilan fibonachi testi swing appletiortaminda yazilmistir. Programi inceleyiniz ve applet ortaminda çalistiriniz.

### Problem 2.32 fibonachitesti2SWA.java programi

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

public class fibonachitesti2SWA extends JApplet implements ActionListener

{

JLabel rakametiketi,sonucetiketi; JTextField rakam,sonuc;

public void init()

{

Container c=getContentPane(); c.setLayout(new FlowLayout());

rakametiketi=new JLabel("Bir tamsayi giriniz : "); rakam=new JTextField(10); rakam.addActionListener(this);

sonucetiketi=new JLabel("Fibonachi degeri : "); sonuc=new JTextField(10); sonuc.setEditable(false);

c.add(rakametiketi); c.add(rakam); c.add(sonucetiketi); c.add(sonuc);

}

public void actionPerformed( ActionEvent e)

{

long sayi;

long fibonachisayisi; sayi=Long.parseLong(rakam.getText()); showStatus("Fibonachi hesaplaniyor...."); fibonachisayisi=fibonachi(sayi); showStatus("Fibonachi hesaplandi...."); sonuc.setText(Long.toString(fibonachisayisi));

}

public long fibonachi(long sayi)

{

if(sayi <=0) return 0;

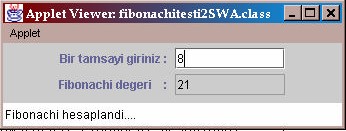
else if(sayi ==1) return 1;

else return fibonachi(sayi - 1)+fibonachi(sayi - 2);

}

}

### 02011.JPG



**Sekil 2.11Fibonachi testi 2 çiktisi (swing applet)**

1. Bir kürenin hacmini hesaplayacak bir applet programi yaziniz. Kürenin hacmi küreHacmi isimli metot tarafindan hesaplansin. Applet bir girdi alanina kürenin yariçapini sorsun ve yariçap girildiginde yeni küre hacmi ekranda gösterilsin.

Not : Kürenin hacmi : (4.0/3.0)\*Math.PI\*Math.pow(r,3);

### Problem 2.33 : Bir kürenin hacmini hesaplayan H5A1.java programi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir

import java.awt.event.\*; // java pencereyi dinleme sinifini cagir public class H5A1 extends Applet implements ActionListener

{

// bu sinif kurenin hacmini hesaplar

//=====================================================

// sinif degiskenleri

// bu degiskenler tum s• n• fa aittir

Label kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi TextField kutugirdisi;// Textfield sinifi degiskeni (nesnesi) kutugirdisi double yaricap; // Gercek degisken yar• cap

double hacim; // Gercek degisken hacim

//=====================================================

// pencereyi baslatma metotu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

kutubasligi=new Label("Kurenin yaricapini (m)giriniz : "); add(kutubasligi);

kutugirdisi=new TextField(10); add(kutugirdisi);

yaricap=0; hacim=0;

kutugirdisi.addActionListener(this);

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metotu

// Her yeni rakam girildiginde bu metot cagirilir public void actionPerformed(ActionEvent e)

{

Double sayi1=new Double(e.getActionCommand() ); yaricap=sayi1.doubleValue(); hacim=kureninhacmi(yaricap);

kutugirdisi.setText(" ");//kutudaki rakami sil repaint();//pencereyi yeniden paint metotuna gore ciz

}

//=====================================================

// paint (boya=pencereye ciz metotu) public void paint(Graphics g)

{

g.drawString(yaricap+" m capindaki kurenin hacmi : "+hacim+" m^3",25,50);

}

//=====================================================

//kurenin hacmini hesaplama metotu public double kureninhacmi(double r)

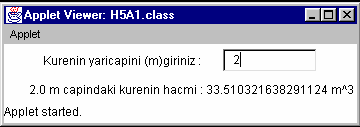
{

return (4.0/3.0)\*Math.PI\*r\*r\*r;

}

}

### 02012.JPG



**Sekil 2.12 Kürenin hacmini hesaplayan H5A1.html applet çiktisi**

1. Bir dik üçgenin hipotenüsünü hesaplayan bir applet programi yaziniz. Program dik üçgenin iki dik kenarini applet de girsin ve hipotenus metotuyla hesaplasin ve sonucu applete yazdirsin.

Not : Dik üçgenin hipotenüsü :

Hipotenüs = Math.sqrt(dikkkenar1\*dikkenar1+dikkenar2\*dikkenar2);

### Problem 2.34 : Bir dik üçgenin hipotenüsünü hesaplayan H5A2.java programi

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir

import java.awt.event.\*; // java pencereyi dinleme sinifini cagir public class H5A2 extends Applet implements ActionListener

{

// bu sinif dik ucgenin hipotenusunu hesaplar

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

Label kutubasligi1; //Label sinifi degiskeni (nesnesi) kutubasligi1 TextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 Label kutubasligi2; //Label sinifi degiskeni (nesnesi) kutubasligi2 TextField kutugirdisi2; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 double dikkenar1; // Gercek degisken dikkenar1

double dikkenar2; // Gercek degisken dikkenar1

//=====================================================

// pencereyi baslatma metotu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

kutubasligi1=new Label("Birinci dik kenari giriniz : ");

add(kutubasligi1); kutugirdisi1=new TextField(10); add(kutugirdisi1);

kutubasligi2=new Label("Ikinci dik kenari giriniz : "); add(kutubasligi2);

kutugirdisi2=new TextField(10); add(kutugirdisi2);

dikkenar1=0; dikkenar2=0;

kutugirdisi2.addActionListener(this);

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metotu

// Her yeni rakam girildiginde bu metot cagirilir public void actionPerformed(ActionEvent e)

{

Double sayi1=new Double(kutugirdisi1.getText() ); dikkenar1=sayi1.doubleValue();

Double sayi2=new Double(kutugirdisi2.getText() ); dikkenar2=sayi2.doubleValue();

//kutugirdisi.setText(" ");//kutudaki rakami sil repaint();//pencereyi yeniden paint metotuna gore ciz

}

//=====================================================

// paint (boya=pencereye ciz metotu) public void paint(Graphics g)

{

g.drawString(" hipotenus = "+hipotenus(dikkenar1,dikkenar2),25,80);

}

//=====================================================

//ucgenin hipotenusunu hesaplama metotu public double hipotenus(double dk1,double dk2)

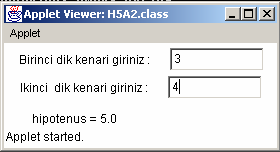
{

return Math.sqrt(dk1\*dk1+dk2\*dk2);

}

}

### 02013.JPG



**Sekil 2.13 Bir dik üçkenin hipotenüsünü hesaplayan H5A2.html applet çiktisi 12. H3OD5\_2000**

iki kenar ve aralarindaki açi (derece cinsinden ) verildiginde üçgenin üçüncü kenarini hesaplayacak

### public static double ucgenkarsikenar(double birinciyan, double ikinciyan, double aciderece)

metodunu yaziniz. not:karsikenar=Math.sqrt(birinciyan\*birinciyan+ikinciyan\*ikinciyan- 2\*birinciyan\*ikinciyan\*Mathd.cosd(aradakiaçi derece));

bu metodu H3OD5\_2000 sinifi **JoptionPane** konsol programi main metodunda kullaniniz (iki yan kenar **ve aradaki açi derece cinsinden verilecek, üçüncü kenar hesaplanarak yazilacak)**

### Problem 2.35 H3OD5\_2000.java programi diküçgenin hipotenüsünü hesaplar

import javax.swing.JOptionPane; // giris çikisi import Mathd;

public class H3OD5\_2000

{

// ucgenin kenarinin bulunmasi metodu

public static double kkenar(double a,double b,double t)

{

double kenar;

kenar=Math.sqrt(a\*a+b\*b-2\*a\*b\*Mathd.cosd(t)); return kenar;

}

public static void main(String[] args)

{

Text cin=new Text(); double sayi1;

double sayi2; double sayi3;

sayi1=Double.parseDouble(JOptionPane.showInputDialog ("birinci kenari giriniz : "));

sayi2=Double.parseDouble(JOptionPane.showInputDialog ("ikinci kenari giriniz : "));

sayi3=Double.parseDouble(JOptionPane.showInputDialog (" iki kenar arasindaki açiyi derece cinsinden giriniz : "));

JoptionPane.showMessageDialog(null,

"üçgenin üçüncü kenari : "+kkenar(sayi1,sayi2,sayi3), "üçgen kenar hesabi", JoptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

}

### 02014.JPG



**Sekil 2.14 Bir dik üçkenin hipotenüsünü hesaplayan H3OD5\_2000.java programinin JoptionPane çiktisinda görülmesi**

1. **H3AL2** sinifinda **static double exp(double x)** metodunu yaz ve ayni sinif içinde çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.36 : H3AL2.java** exp metotunun çagrilmasi import java.io.\*;

public class H3AL2

{

// sayinin exponenti static metodu public static double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1; for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

public static void main(String[] args) throws IOException

{

double sayi;

Text cin=new Text(); System.out.println("Bir sayi giriniz : "); sayi=cin.readDouble();

System.out.println("girilen sayinin exponenti : "+exp(sayi));

}

}

Bir sayi giriniz : girilen sayinin exponenti : 2.7182818284590455

1. **H3AL2\_2000** sinifinda **static double exp(double x)** metodunu yaz ve ayni sinif içinde çagirarak ekrandan

**JoptionPane** kullanilarak girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.37 : H3AL2\_2000.java** exp metotunun çagrilmasi import javax.swing.JOptionPane; // giris çiki

public class H3AL2\_2000

{

// sayinin exponenti static metodu

public static double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1; for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

public static void main(String[] args)

{

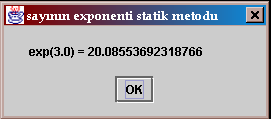
double sayi; sayi=Double.parseDouble(JOptionPane.showInputDialog ("Bir gerçek sayi giriniz : "));

JoptionPane.showMessageDialog(null,"exp("+sayi+") = "+exp(sayi), "sayinin exponenti statik metodu",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 02015.JPG



**Sekil 2.15 Sayinin exponentini statik metodla hesaplayan H3AL2\_2000.java programini JoptionPane kullanan çiktisi**

1. **H3AL2** sinifindaki **static double exp(double x)** metodunu **H3AL3** sinifinda çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.38 : H3AL3.java** exp metotunun çagrilmasi import java.io.\*;

import H3AL2; public class H3AL3

{

public static void main(String[] args) throws IOException

{

double sayi;

Text cin=new Text(); System.out.println("Bir sayi giriniz : "); sayi=cin.readDouble();

System.out.println("girilen sayinin exponenti : "+H3AL2.exp(sayi));

}

}

1. **H3AL2\_2000** sinifindaki **static double exp(double x)** metodunu **H3AL3\_2000** sinifinda çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.39 : H3AL3.java** exp metotunun çagrilmasi import javax.swing.JOptionPane; // giris çiki

import H3AL2\_2000;

public class H3AL3\_2000

{

public static void main(String[] args)

{

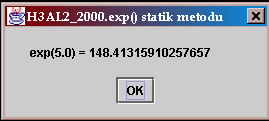
double sayi; sayi=Double.parseDouble(JOptionPane.showInputDialog ("Bir gerçek sayi giriniz : "));

JoptionPane.showMessageDialog(null,"exp("+sayi+") = "+H3AL2\_2000.exp(sayi), "H3AL2\_2000.exp() statik metodu",JoptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 02016.JPG



**Sekil 2.16 Sayinin exponentini statik metodla hesaplayan H3AL3\_2000.java programini JoptionPane kullanan çiktisi**

1. **H3AL4** sinifinda yazilan **double exp(double x)** metodunu ayni sinifda çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.40 : H3AL4.java** exp metotunun çagrilmasi import java.io.\*;

public class H3AL4

{

// sayinin exponenti dinamik metodu

public double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1; for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

public static void main(String[] args) throws IOException

{

double sayi;

Text cin=new Text();

H3AL4 y=new H3AL4();

System.out.println("Bir sayi giriniz : "); sayi=cin.readDouble();

System.out.println("girilen sayinin exponenti : "+y.exp(sayi));

}

}

1. **H3AL4\_2000** sinifinda yazilan **double exp(double x)** metodunu ayni sinifda çagirarak ekrandan girilen bir x degeri için sonucu JOPtionPage kullanarak konsol ortaminda hesaplattir.

**Problem 2.41 : H3AL4\_2000.java** exp metotunun çagrilmasi import javax.swing.JOptionPane; // giris çiki

public class H3AL4\_2000

{

// sayinin exponenti dinamik metodu

public double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1;

for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

public static void main(String[] args)

{

double sayi; sayi=Double.parseDouble(JOptionPane.showInputDialog ("Bir gerçek sayi giriniz : "));

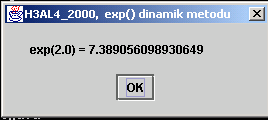
H3AL4 y=new H3AL4();

JoptionPane.showMessageDialog(null,"exp("+sayi+") = "+y.exp(sayi), "H3AL4\_2000, exp() dinamik metodu",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 02017.JPG



**Sekil 2.17 Sayinin exponentini dinamik metodla hesaplayan H3AL3\_2000.java programini JoptionPane kullanan çiktisi**

1. **H3AL5a** sinifinda yazilan **double exp(double x)** metodunu **H3AL5** sinifinda çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.42 :** H3AL5.java, exp metotunun çagrilmasi import java.io.\*;

class H3AL5a

{

// sayinin exponenti dinamik metodu public double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1; for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

}

public class H3AL5

{

public static void main(String[] args) throws IOException

{

double sayi;

Text cin=new Text();

H3AL5a y=new H3AL5a();

System.out.println("Bir sayi giriniz : "); sayi=cin.readDouble();

System.out.println("girilen sayinin exponenti : "+y.exp(sayi));

}

}

1. **H3AL5a** sinifinda yazilan **double exp(double x)** metodunu **H3AL5\_2000** sinifinda çagirarak ekrandan girilen bir x degeri için sonucu swing JOPtionPane kullanarak konsol ortaminda hesaplattir.

**Problem 2.43 : H3AL5.java**, exp metotunun çagrilmasi import javax.swing.JOptionPane; // giris çiki

class H3AL5a

{

// sayinin exponenti dinamik metodu public double exp(double x)

{

double faktoriyel=1; double us=1;

double exponent=1; for(double i=1;i<=300;i++)

{

faktoriyel\*=i; us\*=x;

exponent+=us/faktoriyel;

}

return exponent;

}

}

public class H3AL5\_2000

{

public static void main(String[] args)

{

double sayi; sayi=Double.parseDouble(JOptionPane.showInputDialog ("Bir gerçek sayi giriniz : "));

H3AL5a y=new H3AL5a(); JOptionPane.showMessageDialog(null,"exp("+sayi+") = "+y.exp(sayi), "H3AL5a, exp() dinamik metodu",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

1. **H3AL5a** sinifinda yazilan **double exp(double x)** metodunu **H3AL6 sinifinda ve H3AL6.java dosyasindan** çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.44 :** H3AL6.java, exp metotunun çagrilmasi import java.io.\*;

import H3AL5a;

public class H3AL6

{

public static void main(String[] args) throws IOException

{

double sayi;

Text cin=new Text(); H3AL5a y=new H3AL5a();

System.out.println("Bir sayi giriniz : "); sayi=cin.readDouble();

System.out.println("girilen sayinin exponenti : "+y.exp(sayi));

}

}

1. **H3AL5a** sinifinda yazilan **double exp(double x)** metodunu **H3AL6\_2000 sinifinda ve H3AL6\_2000.java dosyasindan** çagirarak ekrandan girilen bir x degeri için sonucu konsol ortaminda hesaplattir.

**Problem 2.45 : H3AL6\_2000**, exp metotunun çagrilmasi import javax.swing.JOptionPane; // giris çiki

import H3AL5a;

public class H3AL6\_2000

{

public static void main(String[] args)

{

double sayi; sayi=Double.parseDouble(JOptionPane.showInputDialog ("Bir gerçek sayi giriniz : "));

H3AL5a y=new H3AL5a(); JOptionPane.showMessageDialog(null,"exp("+sayi+") = "+y.exp(sayi), "H3AL5a, exp() dinamik metodu",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 23 . H3OD1

Math kütüphanesinde asagidaki metotlar tanimlanmistir. public static native double sin(double a)

a radyan cinsinden

public static native double cos(double a) a radyan cinsinden

public static native double tan(double a) a radyan cinsinden

public static native double asin(double a) sonuç degeri : -pi/2 den pi/2 ye kadar. public static native double acos(double a) sonuç degeri 0 dan pi’ ye kadar

public static native double atan(double a) sonuç degeri : -pi/2 den pi/2 ye kadar. public static native double atan2(double a,

double b) sonuç degeri : -pi den pi ye kadar

bu metotlarin hepsi radyan cinsinden tanimlidir. Bu metotlarin derece esitlerini tanimlamak istiyoruz. Mathd sinifi yaratarak, bu sinifta

public static double sind(double a) a derece cinsinden

public static double cosd(double a) a derece cinsinden

public static double tand(double a)

a derece cinsinden

public static double asind(double a)

sonuç degeri : -90 dan 90 dereceye kadar. public static double acosd(double a) sonuç degeri 0 dan 180 dereceye kadar public static double atand(double a) sonuç degeri : -90 dan 90 a kadar.

public static double atan2d(double a,

double b)

sonuç degeri : -180 den 180 dereceye kadar

statik metotlarini olusturunuz. H3OD1.java applet programini yaziniz. Bu programa derece cinsinden bir girdi giriniz ve hesaplanmis sin,cos,tan degerlerini apletde yazdiriniz.

not : derece=radyan/Math.PI\*180;

### Problem 2.46 : Mathd.java

class Mathd

{

public static double sind(double a)

{

double pi=Math.PI; double r=(a/180.0)\*pi; double sind=Math.sin(r); return sind;

}

public static double cosd(double a)

{

double pi=Math.PI; double r=(a/180.0)\*pi; double cosd=Math.cos(r); return cosd;

}

public static double tand(double a)

{

double pi=Math.PI; double r=(a/180.0)\*pi; double tand=Math.tan(r); return tand;

}

public static double asind(double b)

{

double asind=Math.asin(b)\*180.0/Math.PI; return asind;

}

public static double acosd(double b)

{

double acosd=Math.acos(b)\*180/(Math.PI); return acosd;

}

public static double atand(double b)

{

double atand=Math.atan(b)\*180/(Math.PI); return atand;

}

public static double atan2d(double a,double b)

{

double atand=Math.atan2(a,b)\*180.0/(Math.PI); return atand;

}

}

### Problem 2.47 : H3OD1.java

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import Mathd;

public class H3OD1 extends Applet implements ActionListener

{

double a; double b; Label s1; TextField t1; Label s2; TextField t2;

public void init()

{

s1=new Label("Derece cinsinden aciyi giriniz : "); t1=new TextField(15);

s2=new Label("0 ile 1 arasinda arc giriniz : "); t2=new TextField(15);

add(s1);

add(t1);

add(s2);

add(t2); t1.addActionListener(this); t2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

Double sayi1=new Double(t1.getText() ); a=sayi1.doubleValue();

Double sayi2=new Double(t2.getText() ); b=sayi2.doubleValue();

t1.setText("");

t2.setText(""); repaint();

}

// trigonometrik fonksiyonlarin derece cinsinden yazilmis hali public void paint(Graphics g)

{

g.drawString("sin("+a+") : " + Mathd.sind(a),35,70);

g.drawString("cos("+a+") : " + Mathd.cosd(a),35,90);

g.drawString("tan("+a+") : " + Mathd.tand(a),35,110);

g.drawString("asin("+b+"): " + Mathd.asind(b),35,130);

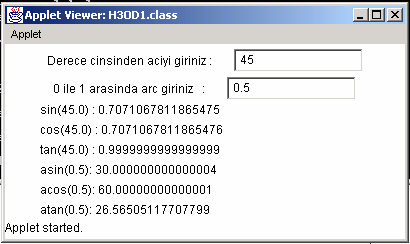
g.drawString("acos("+b+"): " + Mathd.acosd(b),35,150);

g.drawString("atan("+b+"): " + Mathd.atand(b),35,170);

}

}

### 02018.JPG



**Sekil 2.18 Bir derece cinsinden trigonometrik fonksiyonlari hesaplayan H3OD1.html çiktisi**

1. **H3OD2**

Bir konsol programi olusturunuz. Bu programa 0-1 arasinda bir girdi giriniz. Mathd sinifindan atand,acosd ve asind statik metotlarini kullanarak bu metotlarin sonuçlarini konsol ortaminda çiktiya yazdiriniz.

**Problem 2.48 : H3OD2.java**, **import** java.io.\*;

import Text;

import Mathd; public class H3OD2

{

// trigonometrik fonksiyonlarin derece cinsinden yazilmis hali public static void main(String args[]) throws IOException

{

double a;

Text cin=new Text();

System.out.println("0-1 arasinda bir sayi giriniz : "); a=cin.readDouble();

System.out.println("asin("+a+"): " + Mathd.asind(a)+" derece"); System.out.println("acos("+a+"): " + Mathd.acosd(a)+" derece"); System.out.println("atan("+a+"): " + Mathd.atand(a)+" derece");

}

}

### H3OD3

kendi kendini çagiran (recursive), tabanin kuvvetinci kuvvetini hesaplayacak ( taban^kuvvet, 2^3=8 gibi)

### public static pow(double taban, double kuvvet)

metotunu H3OD3 sinifinda, konsol programi olarak yaziniz. main metotunda disaridan bir sayi okuyarak bu metot yardimiyla sonucu hesaplayiniz.

not: H3OD3.pow(taban,1) çagirimi taban degerini geri verecektir.

**Problem 2.49 : H3OD3.java**, import java.io.\*;

public class H3OD3

{

// sayinin exponenti dinamik metodu

public double kuvvet(double x,double y)

{

// y nin pozitif oldugu kabul edilmistir. if(y<0) return 0;

if(y ==0) return 1;

else return x\*kuvvet(x,y-1);

}

public static void main(String[] args) throws IOException

{

Text cin=new Text(); double sayi1;

double sayi2;

H3OD2 x=new H3OD2();

System.out.println(" sayiyi giriniz : "); sayi1=cin.readDouble(); System.out.println(" kuvveti giriniz : "); sayi2=cin.readDouble();

System.out.println(""+sayi1+" nin "+sayi2+" inci kuvveti : "+x.kuvvet(sayi1,sayi2));

}

}

### H3OD4

döngü yapisiyla hesap yapan, tabanin kuvvetinci kuvvetini hesaplayacak ( taban^kuvvet, 2^3=8 gibi) public pow(double taban, double kuvvet)

H3OD4a sinifinda yaziniz. H3OD4 sinifinda bir applet programi olusturarak H3OD4a sinfindaki pow metotu yardimiyla verilen bir sayinin verilen bir kuvvetini hesaplayiniz. ve appletde çikti olarak yazdiriniz.

### Problem 2.50 : H3OD4.java,

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

class **H3OD4a**

{

public static double kuvvet(double a,double b)

{

double k=1.0; if(b<0)

{

for(int i=0;i<(int)(-b);i++)

{k\*=a;} k=1.0/k;

}

else if(b==0)

{

k=1.0;

}

else if(b>0)

{

for(int i=1;i<=(int)b;i++)

{k\*=a;}

}

return k;

}

}

public class **H3OD4** extends Applet implements ActionListener

{

double a; double b; Label s1; TextField t1; Label s2;

TextField t2;

public void init()

{

s1=new Label("tabani giriniz : "); t1=new TextField(10);

s2=new Label("kuvveti giriniz : "); t2=new TextField(10);

add(s1);

add(t1);

add(s2);

add(t2); t1.addActionListener(this); t2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

Double sayi1=new Double(t1.getText() ); a=sayi1.doubleValue();

Double sayi2=new Double(t2.getText() ); b=sayi2.doubleValue();

t1.setText("");

t2.setText(""); repaint();

}

// trigonometrik fonksiyonlarin derece cinsinden yazilmis hali public void paint(Graphics g)

{

g.drawString("taban "+a+" nin "+b+" inci kuvveti : "+ H3OD4a.kuvvet(a,b),35,70);

}

}

### 27. H3OD4\_2000

döngü yapisiyla hesap yapan, tabanin kuvvetinci kuvvetini hesaplayacak ( taban^kuvvet, 2^3=8 gibi) public pow(double taban, double kuvvet)

H3OD4a sinifinda yaziniz. H3OD4\_2000 sinifinda bir siwng JApplet programi olusturarak H3OD4a sinfindaki pow metotu yardimiyla verilen bir sayinin verilen bir kuvvetini hesaplayiniz. ve appletde çikti olarak yazdiriniz.

**Problem 2.51 : H3OD4.java**, import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class H3OD4a

{

public static double kuvvet(double a,double b)

{

double k=1.0; if(b<0)

{

for(int i=0;i<(int)(-b);i++)

{k\*=a;} k=1.0/k;

}

else if(b==0)

{

k=1.0;

}

else if(b>0)

{

for(int i=1;i<=(int)b;i++)

{k\*=a;}

}

return k;

}

}

public class H3OD4\_2000 extends JApplet implements ActionListener

{

double a; double b; JLabel s1; JTextField t1; JLabel s2; JTextField t2; JTextArea cikti;

public void init()

{

cikti=new JTextArea(); Container c=getContentPane(); c.setLayout(new FlowLayout()); s1=new JLabel("tabani giriniz : ");

t1=new JTextField(10);

s2=new JLabel("kuvveti giriniz : "); t2=new JTextField(10);

cikti=new JTextArea("ilk deðer"); c.add(s1);

c.add(t1);

c.add(s2);

c.add(t2);

c.add(cikti); t1.addActionListener(this); t2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

a=Double.parseDouble(t1.getText() ); b=Double.parseDouble(t2.getText() ); t1.setText("");

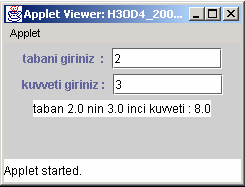
t2.setText("");

cikti.setText("taban "+a+" nin "+b+" inci kuvveti : "+ H3OD4a.kuvvet(a,b)); repaint();

}

}

### 02019.JPG



**Sekil 2.19 H3OD1.html çiktisi**

1. **H3OD5**

iki kenar ve aralarindaki açi (derece cinsinden ) verildiginde üçgenin üçüncü kenarini hesaplayacak public static double ucgenkarsikenar(double birinciyan, double ikinciyan,

double aciderece) metotunu yaziniz.

not : karsikenar=Math.sqrt(birinciyan\*birinciyan+ikinciyan\*ikinciyan- 2\*birinciyan\*ikinciyan\*Mathd.cosd(aradakiaçi derece));

bu metotu H3OD5 sinifi konsol programi main metotunda kullaniniz (iki yan kenar ve aradaki açi derece cinsinden verilecek, üçüncü kenar hesaplanarak yazilacak)

**Problem 2.52 : H3OD5.java**, import java.io.\*;

import Mathd;

public class H3OD5

{

// ucgenin kenarinin bulunmasi metodu

public static double kkenar(double a,double b,double t)

{

double kenar;

kenar=Math.sqrt(a\*a+b\*b-2\*a\*b\*Mathd.cosd(t)); return kenar;

}

public static void main(String[] args) throws IOException

{

Text cin=new Text(); double sayi1;

double sayi2; double sayi3;

System.out.println(" birinci kenari giriniz : "); sayi1=cin.readDouble();

System.out.println(" ikinci kenari giriniz : "); sayi2=cin.readDouble();

System.out.println(" iki kenar arasindaki aciyi derece cinsinden giriniz : "); sayi3=cin.readDouble();

System.out.println("ucgenin ucuncu kenari : "+kkenar(sayi1,sayi2,sayi3));

}

}

1. Hata fonksiyonu erf(x) asagidaki formülle tanimlanabilir.

1  *n x* 2 *n*  1

*erf*

( *x* ) 

 (  1 )

*n*  0

**

*n* ! ( 2 *n*

#  1 )

Bu formülü hesaplayacak olan statik erf metodunu yaziniz ve java swing çiktisi verecek bir örnekte deneyiniz.

1. Bir konsol programi yaziniz. Text sinifi kullanarak disaridan x degerini,**celcius,** fahrenheight cinsinden girilen degerin celcius esitini hesaplasin. Bunun için

### C=5/9\*(F-32)

formülünü kullanabilirsiniz.

Ikinci metot, **fahrenheight,** derece celcius cinsinden girilen degerin fahrenheight esdegerini hesaplasin.

1. Bir applet programi yaziniz. Bu programin içindeki kubunhacmi metodu kübün hacmini hesaplasin. Applet kullaniciya kübün kenar uzunlugunu sorsun ve hacmini versin. okuyunuz. erf statik metodu yaratarak formülü hesaplayiniz. Ana programda erf metodunu çagiriniz ve sonucunu yazdiriniz. Hata analizi yaparak döngüyü kaç terim için hesaplamaniz gerektigine karar veriniz.
2. Sicaklik.java programini denemek için bir swing JApplet programi yaziniz.
3. **mesafe** metodunu yaziniz. Bu metod iki noktanin (x1,y1) ve (x2,y2) arasindaki mesafeyi hesaplasin. Bu metodu bir konsol test programinda deneyin.
4. Bir tam sayi eger kendisinden baska bir tam sayiyla bölünemiyorsa, asal sayi adini alir. Örnegin 2,3,5,7,11..

Asal sayilari bulan bir metot yaziniz. Bu metotu bir applette kullanarak 1 den 1000 e kadar olan sayilar arasindaki asal sayilari belirleyiniz.

1. Üç (gerçek veya tamsayi) sayinin arasindan minimum olani bulan **minimum3** metodunu yaziniz ve bir konsol programinda test ediniz
2. Üç sayinin (gerçek veya tamsayi) arasindan maksimum olani bulan **maksimum3** metodunu yaziniz ve bir konsol programinda test ediniz
3. **tekmi** isimli boolean bir metot yaziniz. bu metot eger girilen sayi tekse true degeri, sayi çiftse false degeri versin. Bu metodu bir applet programinda test ediniz.
4. kürenin hacmini hesaplayan hacim(double x) metodunu içeren kureApplet applet programini olusturunuz. Bu program kürenin yariçapini sorsun, ve sonuç olarak hacim metodunda hesaplanan kürenin hacmini versin.

### Problem 2.53 kürenin hacmini hesaplayan küreapplet.java programi

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class kureapplet extends Applet implements ActionListener

{

Label rakametiketi,sonucetiketi; TextField rakam,sonuc;

public void init()

{

rakametiketi=new Label("Bir tamsayi giriniz : "); rakam=new TextField(10); rakam.addActionListener(this); sonucetiketi=new Label("Kurenin Hacmi : "); sonuc=new TextField(10); sonuc.setEditable(false);

add(rakametiketi); add(rakam); add(sonucetiketi); add(sonuc);

}

public void actionPerformed( ActionEvent e)

{

double sayi; double sayisi;

Double sayi1=new Double(e.getActionCommand() ); sayi=sayi1.doubleValue();

sayisi=hacim(sayi); showStatus("hesaplandi...."); sonuc.setText(Double.toString(sayisi));

Double sayi2=new Double(e.getActionCommand() ); sayisi=sayi2.doubleValue();

}

public static double hacim(double x)

{

return (4.0/3.0)\*Math.PI\*Math.pow(x,3);

}

}

### BÖLÜM 3 . SINIF (CLASS) YAPILARINA GIRIS

* 1. **STANDART KAVRAMLARIN TÜRKÇE KARSILIKLARI Class : sinif** ana nesne kökenli programlama elemani.

**Object : nesne** sinif (Class) yapilari bilgisayar belleginde bir islem için kullanildiginda aldiklari (özel) isim. **new :** siniflarin kurucu metotunu çagirip nesne tanimini ve bilgisayar adreslerini olusturan deyim **Method: metot** siniflarin içindeki islevsel program parçaciklari

**Constructor : Kurucu metot** nesne ilk defa olusturulurken hangi sinif degiskenlerinin hangi degerleri alacagini belirleyen metot

### SINIF YAPISININ GENEL TANIMI

Su na kadarki bütün programlarimizda sinif(class) yapisini kullandik. Fakat tam olarak siniflarin ne oldugunu ve neler yapabileceklerini anlatmadik. Sinif yapisinin en onemli ozelligi yeni bir degisken türü tanimliyor olmasidir. Su ana kadar programlarimizi tanimlamak amaciyla çesitli siniflar tanimladik. Bu bölümde siniflarin degisken türü olarak tanimlamasini ve kullanmasini görecegiz.

Sinif (Class) deyince herhangi bir nesnenin genel tanimlanmasini anliyoruz. Bu siniflardan degiskenler tanimlandiklarinda nesneleri olustururlar. Örnegin kalem dedigimizde genel bir sinifi anlatiyoruz. Alinin cebindeki kirmizi kalem ise belirli bir nesnedir. Sinifin genel tanimi su sekilde yapilabilir :

### class sinifin\_ismi

**{**

**//sinifa ait degisken listesi degisken\_türü degisken\_ismi; degisken\_türü degisken\_ismi;**

**.........**

**//sinifa ait metot listesi**

**degisken türü metotadi(metot girdi degisken listesi)**

**{**

**//metotun içindeki hesaplanan islemler return çikti\_degiskeni;**

**}**

**..............**

**}**

Sinif (class) taniminda metotlarin (veya degiskenlerin) bulunmasi sart degildir. Bulunduklari takdirde bunlar o sinifa ait veya sinifin üyesi metotlar olarak düsünülür. Ilk sinif örnegi olarak içinde hiçbir metotun yer almadigi bir kutu sinifi tanimliyalim. Bir kutuyu üç geometrik boyutu (eni,boyu,yüksekligi) cinsinden tanimlayabiliriz.

### class Kutu

**{**

**double en; double boy; double yukseklik;**

**}**

Üstte bahsettigimiz gibi sinif yeni bir degisken türü yaratir. Yukardaki sinifin yarattigi degisken türünün adi Kutu’dur. Kutu degiskenini baska bir programda tanimlarken budegisken türünden bir nesne yaratilir. Örnegin

### Kutu kutu1= new Kutu();

kutu1, kutu türünde tanilanmis bir nesnedir. Bu nesnenin kendisinin de en boy ve yükseklik olmak üzere üç alt degiskeni mevcuttur. Eger bir programda kutu1 degiskeni tanimlanmissa, bu degiskenin alt degiskenlerine su sekilde ulasilabilir :

### kutu1.en=1.0;

Simdi Kutu sinifindan nesneyi kullanacak ufak bir kututesti sinifi olusturalim ve Kutu sinifini bu sinifta kullanalim.

import javax.swing.JOptionPane; // giris çikis

class Kutu

{

double en; double boy; double yukseklik;

}

class kututesti

{

public static void main(String args[])

{

double hacim;

String s="";

Kutu kutu1=new Kutu(); kutu1.en=10.0; kutu1.boy=20.0; kutu1.yukseklik=30.0;

hacim=kutu1.en\*kutu1.boy\*kutu1.yukseklik; s=s+"Kutunun hacmi : "+hacim+"\n"; JOptionPane.showMessageDialog(null,s,

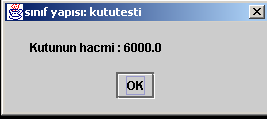
"sinif yapisi: kututesti1",JOptionPane.PLAIN\_MESSAGE); System.exit(0);}

}

Bu programin adinin kututesti.java olmasi zorunludur. Çünki ana metot(main) kututesti sinifinda tanimlanmistir. Fakat programi javac terimiyle bilgisayar diline çevirdigimizde iki tane sinif(class) dosyasinin olustugunu görürüz. Birisi **Kutu.class** digeride **kututesti.class**. Ana program kututesti oldugu için bu programin isletimi **java kututesti**

deyimini kullanarak yapilir. Program isletildiginde

### 03001.JPG



**Sekil 3.1 kututesti.java programinin JoptionPage grafik çiktisindaki görünümü**

Sonucunu verir. kutu1 nesnesi yaratildiginda bilgisayar belleginde kutu1 için bir adres tanimlanir. New Kutu() deyimi verildiginde ise bilgisayrin belleginde en,boy ve yükseklik degiskenleri için yer ayrilir ve bu yerlerin adresleriyle kutu1 nesnesinin adresi arasinda baglanti saglanir. Bunu grafik olarak anlatalim :

Tablo 3.1 Nesne adreslerinin bilgisayar bellegindeki yerleri ve yerlesmesi

|  |  |  |
| --- | --- | --- |
| **Deyim** | **Bilgisayar ana nesne bellegi** | **Nesne alt hafiza bellegi** |
| Kutu kutu1; | null | Mevcut degil |
| kutu1=new Kutu(); | Kutu1  kutu1 adresi (örnegin: 23436A3F) | en Boy  Yükseklik |

Not: null terimi bilgisayar beleginin bos oldugunu gösterir. Yukardaki grafikte ilk olarak

### Kutu kutu1;

tanimi yapildiginda bilgisayar belleginde 1 byte boyutunda bir yer tanimlar. Ve bu yere bos anlamina gelen null degerini yükler. Ikinci durumda ise (**kutu1=new Kutu()** taniminda) kutu1 nesne adresindeki kutuda en,boy,yükseklik degiskenlerinin adresleri kaydedilmistir. Bu adres araciligiyla en boy ve yükseklik degiskenlerine benimkutum degiskeni üzerinden ulasilabilir. Diger deyimle en,boy,yükseklik kutu1 nesnesinin alt degiskenleridir.

### KURUCU ( CONSTRUCTOR ) METOT

Ikinci bir programda Kutu sinifi iki nesne tanimlama geregi duydugumuzu varsayalim. Buna örnek olarak Sekil

3.2.1 de kututesti1.java programi tanimlanmistir.

### Program 3.1 : Kurucu metot örnegi, Kutu sinifi iki nesne tanimlayan kututesti1.java programi

import javax.swing.JOptionPane; // giris çikis class Kutu

{

double en; double boy; double yukseklik;

}

class kututesti1

{

public static void main(String args[])

{

double hacim;

Kutu kutu1=new Kutu(); String s=""; kutu1.en=10.0; kutu1.boy=20.0; kutu1.yukseklik=30.0;

hacim=kutu1.en\*kutu1.boy\*kutu1.yukseklik; s+="Birinci kutunun hacmi : "+hacim+"\n"; Kutu kutu2=new Kutu();

kutu2.en=5.0; kutu2.boy=10.0; kutu2.yukseklik=15.0;

hacim=kutu2.en\*kutu2.boy\*kutu2.yukseklik; s+="Ikinci kutunun hacmi : "+hacim+"\n"; JOptionPane.showMessageDialog(null,s,

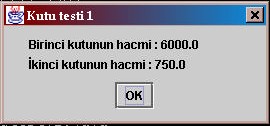
"Kutu testi 1",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Bu programin çiktisi :

### 03002.JPG



**Sekil 3.2 kututesti1.java**

Bu programda Kutu sinifindan **kutu1** ve **kutu2** degiskenleri tanimlanmistir. Bu basit programda bile en,boy ve yükseklik degikenlerini iki defa çagirmak program listesinin uzamasina sebep olmustur. Gerçek uygulamalarda ise çok daha yüksek sayida degisken içeren siniflar olusturulabilir ve bu siniflarda olusturulan toplam nesne

sayisi da çok fazla olabilir. Bu yüzden nesneyi ilk tanimlarken ayni zamanda degiskenlerinin degerlerini de tanimlayabilirsek, çok daha okunakli ve kisa bir bilgisayar listesiyle ayni isi yapabiliriz. Bu is için kurucu (constructor) adi verilen metotlar kullanilir. Kurucu metotlari diger metotlardan biraz farklidir. Once metot önüne metot degisken türü gelmez ve metot disina hiçbir degisken göndermezler. Isimleri de herzaman sinif(class) ismiyle ayni olur. Program 3.1 deki programi kurucu (constructor) metotu kullanarak tekrar yazalim.

### Program 3.2 : Kurucu metot örnegi, Kutu sinifi iki nesneyi kurucu(constructor) metotu yardimiyla tanimlayan kututesti2.java programi

import javax.swing.JOptionPane; // giris çikis class Kutu

{

double en; double boy; double yukseklik;

//kurucu metod kutu

Kutu(double e,double b,double y)

{

en=e; boy=b; yukseklik=y;

}

}

class kututesti2

{

public static void main(String args[])

{

double hacim;

String s="";

Kutu kutu1=new Kutu(10.0,20.0,30.0); hacim=kutu1.en\*kutu1.boy\*kutu1.yukseklik; s+="kutu1\'in hacmi : "+hacim+"\n";

Kutu kutu2=new Kutu(5.0,10.0,15.0); hacim=kutu2.en\*kutu2.boy\*kutu2.yukseklik; s+="kutu2\'nin hacmi : "+hacim;

JOptionPane.showMessageDialog(null,s,

"Kutu testi 2",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Bu program bir öncekinin ayni sonucu verir. Fakat burada bir önceki programda yapilan

### Kutu kutu1=new Kutu();

Tanimi yerini

### Kutu kutu1=new Kutu(10.0,20.0,30.0);

Tanimina birakmis ve kututesti2 sinifi yazilirken en ,boy yükseklik degiskenlerinin yeniden tanimlanmasi ihtiyacini ortadan kaldirmistir.

### DIGER METOTLAR

Tekrar yukarida inceledigimiz kutu sinifini göz önüne alalim. Hacim bir kutunun özellikleri arasinda yer alir, ve her zaman kutunun hacminin hesaplanmasi gerekebilir. Bu yüzden hacim islemini Kutu sinifinin içinde

tanimlamak daha ideal olacaktir. Sekil 3.3.1 deki kutuornegi3.java programinda hacim hesaplarini yapan hacim metotu Kutu sinifinin içinde tanimlanmistir.

### Program 3.3 : Kurucu metot örnegi, Kutu sinifi iki nesneyi kurucu(constructor) metotu yardimiyla tanimlayan ve hacmi hacim() metotuyla hesaplayan kututesti3.java programi

import javax.swing.JOptionPane; // giris çikis class Kutu

{

double en; double boy; double yukseklik;

//kurucu metod kutu

Kutu(double e,double b,double y)

{

en=e; boy=b; yukseklik=y;

}

### //bu metot kutunun hacmini hesaplar double hacim()

**{**

**return en\*boy\*yukseklik;**

**}**

}

class kututesti3

{

public static void main(String args[])

{

double hacim;

String s="";

Kutu kutu1=new Kutu(10.0,20.0,30.0); s+="kutu1\'in hacmi : "+kutu1.hacim()+"\n"; Kutu kutu2=new Kutu(5.0,10.0,15.0); s+="kutu2\'nin hacmi : "+kutu2.hacim();

JOptionPane.showMessageDialog(null,s,

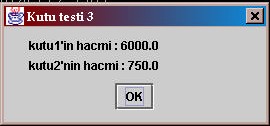
"Kutu testi 3",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Program çiktisi yine ayni olacaktir.

### 03003.JPG



**Sekil 3.3 kututesti3.java**

Bir önceki bölümde metot yazimi konusunda anlattiklarimizin hepsini buraya ilave edebiliriz.

### THIS DEYIMININ KULLANIMI

Bazen bir metotun içinde o metotun ait oldugu siniftan yaratilacak nesneyi veya o nesnenin bir alt degiskenini tanimlamak gerekir. Nesne daha tanimlanmadigindan direk olarak nesne ismini kullanamayiz. Bunun yerina java **this** deyimini kullanir. This deyimi bilhassa sinifa ait degisken isimlerinin aynisi metotda kullanilmissa da ise yarar. Bu durumda tüm sinifa ait degiskenler **this.degisken\_ismi** komutuyla çagirilabilir. geçen bölümde sinif ve metota ait degiskenleri izlerken metotvesinifdeg.java ve metotvesinifdeg1.java isimlerinde iki örnek problemi incelemistik.

Simdi de aynigruptan metotvesinifdeg3.java programini inceleyelim.

### Program 3.4 : metotvesinifdeg3.java programiyla ve this deyimi yardimiyla metotun içinde yerel ve sinifa ait degiskenlere ulasim.

import java.awt.Graphics; import java.applet.Applet;

public class metotvesinifdeg3 extends Applet

{

//buradaki degiskenler tum sinifa aittir

### double x ; int y ;

**void metot1( Graphics g) //metota hiç bir degisken girmiyor**

**{**

**double x; int y; x=5.5; y=6;**

**//tum sinifa at degiskenler this kelimesi ile birlikte kullanilabilirler**

**g.drawString("metot1 in dis degiskenleri this ile ulasim : x = "+this.x+" y = "+this.y,25,25); g.drawString("metot1 in ic degiskenleri : x = "+x+" y = "+y,25,40);**

**}**

public void paint(Graphics g)

{

x = 2.5;

y = 3;

metot1(g);

}

}

Sinifa ait her degiskeni basina **this**. getirerek kullanabiliriz. Fakat this kullanimi mutlaka gerekli degilse kullanilmamasi tavsiye edilir.

Örnegin yukaridaki kurucu metot kutu

### //kurucu metot kutu

**kutu(double en,double boy,double yukseklik)**

**{**

**this.en=en; this.boy=boy;**

**this.yukseklik=yukseklik;**

**}**

seklinde this degiskenini kullanarak da yazilabilirdi. Burada girdi degiskeni ile sinif degiskeninin adi ayni verilmistir. This sol taraftaki degiskenin sinif degiskeni oldugunu bilgisayara bildirmektedir.

### SINIF DEGISKENLERININ DIS DÜNYADAN GIZLENMESI

Yukarda inceledigimiz kutu metodu. Program 3.5 de daha detayli yazilmistir. Burada en boy ve yükseklik degiskenleri private olarak tanimlanmistir. Private olarak tanimlanan degiskenlere disaridan direk olarak

Kutu x=new Kutu(3.0,5.0,7.0); x.en=8.0;

gibi direk ulasim mevcut degildir. Bu degiskenlere ulasim (degerlerini degistirme veya çagirma) ancak ulasim metodlari adini verdigimiz metodlar araciligiyla yapilabilir.

### Program 3.5 : Kurucu metot örnegi, Kutu sinifi iki nesneyi kurucu(constructor) metotu yardimiyla tanimlayan ve hacmi hacim() metotuyla hesaplayan ve diger metod tanimlarinin da bulundugu kututesti4.java programi.

import javax.swing.JOptionPane; // giris çikis class Kutu

{

private double en; private double boy; private double yukseklik;

//kurucu metod kutu

Kutu(double e,double b,double y)

{

en=e; boy=b; yukseklik=y;

}

//sinif degiskenlerini degistirme set metodu public void setKutu(double e,double b,double y)

{

en=e; boy=b; yukseklik=y;

}

//sinif degiskenlerine ulasma (get) metotlari public double getEn()

{return en;}

public double getBoy()

{return boy;}

public double getTukseklik()

{return yukseklik;}

//bu metot kutunun hacmini hesaplar (islem- hesap) metodu public double hacim()

{

return en\*boy\*yukseklik;

}

//string çikti metodu public String toString()

{

return("en = "+en+" boy = "+boy+" yükseklik = "+yukseklik+"\n"+ "Hacim = "+hacim()+"\n");

}

}

class kututesti4

{

public static void main(String args[])

{

double hacim;

String s="";

Kutu kutu1=new Kutu(10.0,20.0,30.0); s+=kutu1.toString();

Kutu kutu2=new Kutu(5.0,10.0,15.0); s+=kutu2.toString();

JOptionPane.showMessageDialog(null,s,

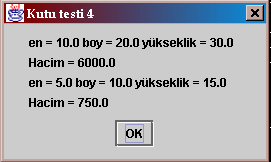
"Kutu testi 4",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Bu programin çiktisi :

### 03004.JPG



**Sekil 3.4 kututesti4.java**

Ikinci bir örnek olarak dörtislemhesapmakinasi sinifina bakalim. Burada da sinif degiskenleri protected olarak tanimlanmistir. Asagida dort islem (toplama cikarma carpma bolme yapan dortislemhesapmakinasi sinifini görüyorsunuz.

### Program 3.5 : dörtislemhesapmakinasi.java programi ve sinif degiskenlerinin degerlerinin metot kullanilarak degistirilmesi

public class dortislemhesapmakinasi

{

//sinif degiskenleri x sonuc hafiza

### protected double x,sonuc,hafiza;

// kurucu metot dortislemhesapmakinesi public dortislemhesapmakinasi(double y)

{

//constructor sonuc=y; hafiza=0;

}

// ikinci bir kurucu metot dortislemhesapmakinesi public dortislemhesapmakinasi()

{

sonuc=0; hafiza=0;

}

public double topla(double x)

{

sonuc+=x;

return sonuc;

}

public double cikar(double x)

{

sonuc-=x; return sonuc;

}

public double carp(double x)

{

sonuc\*=x; return sonuc;

}

public double bol(double x)

{

sonuc/=x; return sonuc;

}

//bu metotla sonuc degeri degistirilir public void gir(double x)

{

sonuc=x;

}

public double oku()

{

return sonuc;

}

public void C()

{

sonuc=0;

}

public void M(double x)

{

//hafizaya al hafiza=x;

}

public double MR()

{

//hafizayi çagir return hafiza;

}

public void MTopla(double x)

{

hafiza+=x;

}

public void Mcikar(double x)

{

hafiza-=x;

}

public double isaretdegistir()

{

sonuc=-sonuc; return sonuc;

}

public String toString()

{

return ""+sonuc;

}

public String MtoString()

{

return ""+hafiza;

}

}

Bu sinifta (class) sinif degiskeni **sonuc** degerlerinin degistirilmesi için **gir** metotu yazilmistir. Bunun sebebi programlarda sinifa ait degiskenlere disaridan direk ulasimin önlenmesini saglamaktir. Iyi programlama sartlarindan birisi sinifin degiskenlerine sadece o sinifin metotlari üzerinden ulasmaktir. Herkese serbest ulasim verilmez. Hatta bunu saglamak için degiskenlerin basina sadece bu sinifa aittir disaridan kullanilamaz anlamindaki **private** sözcügü getirilebilir**. private** sözcügü yerine ancak disarda bu siniftan yaratilana yeni siniflar (akraba siniflar) kullanabilir, yabanci siniflar kullanamaz anlamindaki **protected** sözcügü de sik olarak kullanilir. Protected sözcügü ayni directorydeki diger dosyalari da (public) gibi kabula eder, fakat baska direktorylerdeki dosyalar için private olarak davranir.

**dortislemhesapmakinasi.java** programinin x, sonuc ve hafiza degiskenleri için **protected** sözcügü kullandigimiz için

### dortislemhesapmakinasi h = new dortislemhesapmakinasi(); h.sonuc = 2.0;

seklinde ulasilamaz. Ayni islemi yapmak için asagidaki üç yoldan birini kullanabiliriz.

### dortislemhesapmakinasi h = new dortislemhesapmakinasi(); h.gir(2.0);

veya

### dortislemhesapmakinasi h = new dortislemhesapmakinasi(2.0);

seklinde sonuç degiskenine 2.0 degeri atanir.

Hafiza degiskeninin degerinin degistirilmesi için ise :

### h.M(3.0);

metotu çagirilir.

Sinif degiskenlerinin dis dünyadan gizlenmesi niçin gereklidir? Bunun en önemli sebebi program güvenliginin saglanmasidir. Degiskenlere disaridan direk müdahale edilebilirse, degerleri de degistirilebilir. Örnegin degiskeni bir banka hesabi veya sirketin özel hesabi olarak düsünebiliriz. Dis dünyadan gizlenmemis bir degiskenin degeri degistirilebilir. Private veya protected degiskenler ise ancak metotlar üzerinden degistirilebilir, burada yeterli güvenlik önlemlerini almak mümkündür.

### SINIFLARDA KALITIM (INHERITANCE)

Bir sinifin kurulmasinda o sinifla direk olarak iliskisi olan bir siniftan onun tüm metotlari ve degiskenleri **kalitim (inheritance)** yoluyla yeni kurulan bir sinifa aktarilabilir. Kalitim için sinif(class) taniminda **extends** kelimesi kullanilir. Eger bir siniftan daha alt siniflara kalitim yoluyla aktarma yapilacaksa **private** degisken türü yerini **protected** degisken türüne birakir. Protected private deyiminin aynidir. Fakat kalitimla degiskenlerin aktarildigi alt siniflara degiskenleri görme izni verir. Private terimi ise kalitimla aktarilan alt siniflardan bile o degiskenleri saklar.

Program 3.6 da nokta sinifi tanimlanmistir. Program 3.7 de ise nokta sinifindan kalitim yoluyla yaratilan daire sinifini görüyoruz. Programda da görülecegi gibi daire sinifinin tanimi

### public class daire extends nokta

terimiyle yapilmistir. Ayrica daire sinifinin kurucusu yazilirken nokta sinifinin kurucusu direk olarak

**super(a,b);** terimiyle çagirilmistir.

Burada önemli bir noktayi hatirlatalim :

### "Java dilindeki bir sinif sadece bir tane baska sinifin metot ve degiskenlerini kalitim yoluyla aktarabilir. "

**Program 3.6 : nokta.java programi ve nokta sinifinin tanimi.**

public class nokta

{

protected double x, y;

public nokta(double a, double b)

{

noktagir(a,b);

}

public void noktagir(double a, double b)

{

x=a; y=b;

}

public double Xoku()

{

return x;

}

public double Yoku()

{

return y;

}

public String toString()

{

return "["+x+","+y+"]";

}

}

### Program 3.7 : daire.java programi ve daire sinifinin tanimi. Daire sinifi nokta sinifindan Kalitim (inheritance) yoluyla türetilmistir.

import nokta;

public class daire extends nokta

{

protected double yaricap;

public daire()

{

//daire kalitim yaptigi nokta sinifinin kurucu metotunu

// super deyimi ile cagirabilir.

### super(0,0); yaricapgir(0);

}

public daire(double r, double a, double b)

{

### super(a,b); yaricapgir(r);

}

public void yaricapgir(double r)

{

if(r = 0.0) yaricap=r;

else

yaricap=0.0;

}

public double yaricapoku()

{

return yaricap;

}

public double alan()

{

return 3.14159\*yaricap\*yaricap;

}

public String toString()

{

return "Merkez = "+"["+x+","+y+"]"+ "; Yaricap="+yaricap;

}

}

Kalitim yoluyla aktarilan metotlar ve degiskenler yeni programda o programin içinde yazilmis gibi aynen kullanilabilir.

### SINIFLARI BASKA BIR SINIFTA NESNE OLARAK ÇAGIRARAK BiR ARAYA GETiRME (COMPOSITION)

Siniftaki degisken ve metotlari kalitim yoluyla aktarabildigimiz gibi, sadece üst sinifa ait bir nesne (object) tanimini alt sinifta tanimliyarak ta yapabiliriz. Buna nesne yoluyla aktarim (ingilizce composition) diyoruz.

### Program 3.8 : 1 daire1.java programi ve daire sinifinin tanimi. Nokta sinifi daire1 sinifinda merkez adli nesne olarak tanimlanmistir.

import nokta; public class daire1

{

protected double yaricap;

### protected nokta merkez;

public daire1()

{

### merkez=new nokta(0,0); yaricapgir(0);

}

public daire1(double r, double a, double b)

{

### merkez=new nokta(a,b); yaricapgir(r);

}

public void yaricapgir(double r)

{

if(r = 0.0) yaricap=r;

else

yaricap=0.0;

}

public double yaricapoku()

{

return yaricap;

}

public double alan()

{

return 3.14159\*yaricap\*yaricap;

}

public String toString()

{

return "Merkez = "+"["+merkez.x+","+merkez.y+"]"+ "; Yaricap="+yaricap;

}

}

daire1 programinda

### KALITIM YOLUYLA ALT SINIFA BAGLANMIS ÜST SINIF REFERANSI ÜZERINDEN ALT SINIFI ÇAGIRMA

Yukarida nokta ve daire siniflarini tanimlamistik. Baska bir siniftan kalitim yoluyla türeyen sinifin(dairenin noktadan kalitim yoluyla türemesi gibi), üst sinifin adresi(tanimi) üzerinden programin içinde aktarimi mümkündür.

### Program 3.9 : noktatesti.java sinifinda üst sinif üzerinden adres referansi kullanimi

import java.awt.Graphics; import java.applet.Applet;

public class noktatesti extends Applet

{

private nokta noktaref,n; private daire daireref,d; public void init()

{

n=new nokta(30,50); d=new daire(2.7,120,89);

}

public void paint(Graphics g)

{

//once noktamizi ve dairemizi yazdiralim g.drawString("Nokta n : "+n.toString(),25,25); g.drawString("Daire d : "+d.toString(),25,40);

//c dairesinin adresini nokta olarak tanimlanan noktaref e aktaralim noktaref=d;

g.drawString("Daire(noktaref uzerinden) :"+noktaref.toString(),25,70);

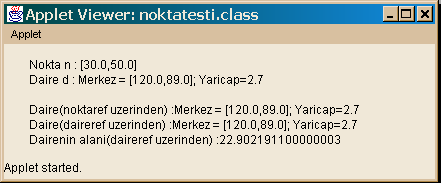
### //noktaref nesnesini daireref degiskenine(nesnesine) aktaralim daireref=(daire)noktaref;

g.drawString("Daire(daireref uzerinden) :"+daireref.toString(),25,85); g.drawString("Dairenin alani(daireref uzerinden) :"+daireref.alan(),25,100);

}

}

### 03005.JPG



**Sekil 3.5 Noktatesti.html**

Program 3.9 da görülen noktatesti.java sinifinda, n nokta sinifi nesnesi, ve noktaref noktasinifi nesnesi, ve d dairesinifi nesnesi, ve daireref daire sinifi nesnesi tanimlanmistir. Init metotunda p ve c nesnelerinin adresleri

kurucu metotu new deyimiyle tasiyarak tanimlanmistir. Dahasonra noktaref=d; deyimiyle aslinda daire sinifindan olan d nesnesi nokta sinifindan tanimlanan noktaref nesnesinin adresine yüklenmistir. Programda da görüldügü gibi noktaref üzerinden daireyi kullanmak mümkündür çünkü **nokta dairenin kalitim yoluyla üst sinifidir.** Ayni zamanda **cast** islemcisi **(daire)** terimini kullanarak noktaref’i daireref’e aktarmak da mümkün olmustur. Bu aktarma nokta dairenin super sinifi olmasa basarilamazdi. Bazen sadece bu süper sinif aktarimini saglamak amaciyla sinif hiyerarsisinin en tepesine aslinda belli bir görevi olmiyan **abstract** siniflar yerlestirilir. Bu sinifin alt siniflari olarak tanimlanan tüm siniflar bu sinif adresinden yararlanilarak aktarilabilir. Abstract siniflarin taniminda **abstract** sözcügü kullanilir. Abstract tür metotlarin içinde metotlarin sadece basliklari verilip tanimlari yapilmiyabilir. (metotun ana gövdesi tanimlanmiyabilir). Abstract super sinifinin kullanilmasina söyle bir örnek verebiliriz : sekil isimli bir abstract sinifin alt siniflari olarak daire,üçgen,dikdörtgen ve kare siniflarini olusturdugumuzu ve bu siniflarin herbirinin kendi seklini çizecek ciz metotunu içerdigini düsünelim. Programi yazarken abstract sinif sekilde de alan metotu tanimi yapariz. Ve sekil sinifindan bir nesne tanimlariz. Program çalisirken sekil üzerinden istedigimiz alt seklin çizimi dinamik olarak her biri için ayri bir adres tanimlamadan mümkün olur. Ayni sekilde bir nümerik analiz programi yazarken, örnegin her degisik fonksiyon için ayri ayri integral metotu yazma yerine abstract tür bir fonksiyon için integral programini yazar ve gerçek foksiyonumuzu tanimladigimiz sinifi bu abstract sinifin alt sinifi olarak tanimlariz.

Bu kavrami daha sonraki bölümlerimizde daha genis örneklerle açiklamaya çalisacagiz.

### NESNEYi DiNAMiK OLARAK SiLMEK FINALIZE() METOTU

Java dili kullanilmayan nesneleri silmek için arka planda çalisan bir sisteme sahiptir. Bu yüzden C++ gibi dillerde dinamik nesne tanimlarinda mutlaka yer almasi gereken nesneyi silme metotlari javada o kadar gerekli degildir. Fakat eger kendiniz metlaka bir nesneyi acil olarak silmek isteseniz **finalize()** isimli bir metotu sinifinizda tanimlayip kullanabilirsiniz. Örnek olarak asagidaki metotu verebiliriz.

protected void finalize() throws Throwable

{

super.finalize();

}

### ALISTIRMALAR

1. kututesti1.java, kututesti2.java,kututesti3.java ve kututesti4.java programlarini karsilastiriniz. Aralarindaki fark ve benzerlikleri açiklayiniz.
2. dairetest.java programini inceleyiniz

### Program 3.10 dairetesti.java, bu program nokta sinifindan (Program 3.5) kalitim yoluyla türüyen daire sinifinin(Program 3.6) kullanilmasini göstermektedir.

import java.io.\*; import daire;

class dairetest

{

public static void main(String args[]) throws IOException

{

nokta merkez=new nokta(1.0,2.0); daire d1=new daire(2.5,1.0,2.0);

System.out.println("nokta : "+merkez.toString() ); System.out.println("daire : "+d1.toString() ); System.out.println("dairenin alan• : "+d1.alan());

//not : noktagir metodu kalitimla nokta sinifindan aktarilmistir d1.noktagir(1.5,2.5);

System.out.println("daire(merkez degisti) : "+d1.toString() ); d1.yaricapgir(1.0);

System.out.println("daire(yaricap degisti) : "+d1.toString() ); System.out.println("dairenin alan• : "+d1.alan());

}

}

nokta : [1.0,2.0]

daire : Merkez = [1.0,2.0]; Yaricap=2.5 dairenin alani : 19.6349375

daire(merkez degisti) : Merkez = [1.5,2.5]; Yaricap=2.5 daire(yaricap degisti) : Merkez = [1.5,2.5]; Yaricap=1.0 dairenin alani : 3.14159

1. dairetest\_2000.java programini inceleyiniz

### Program 3.11 dairetest\_2000.java, bu program nokta sinifindan (Program 3.5) kalitim yoluyla türüyen daire sinifinin(Program 3.6) kullanilmasini göstermektedir.

import javax.swing.JOptionPane; // giris - çikis import daire;

class dairetest\_2000

{

public static void main(String args[])

{

nokta merkez=new nokta(1.0,2.0); daire d1=new daire(2.5,1.0,2.0); String s="";

s+="nokta : "+merkez.toString()+"\n"; s+="daire : "+d1.toString()+"\n"; s+="dairenin alaný : "+d1.alan()+"\n";

//not : noktagir metodu kalýtýmla nokta sýnýfýndan aktarýlmýþtýr d1.noktagir(1.5,2.5);

s+="daire(merkez degisti) : "+d1.toString()+"\n"; d1.yaricapgir(1.0);

s+="daire(yaricap degisti) : "+d1.toString()+"\n"; s+="dairenin alaný : "+d1.alan()+"\n"; JOptionPane.showMessageDialog(null,s,

"daire testi 2000",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 03006.JPG



**Sekil 3.6 daire testi 2000 JoptionPage çiktisi**

1. Nokta sinifini kullanarak dogru sinifini yarat. Dogru iki noktayla tanimlanir. Dogru sinifinin kurucu metotunu, boyunu hesaplayan boy metotunu, uç noktalarini String degiskeni olarak tanimlayan toString metotunu ve gerekli olan diger metotlari tanimla. Konsol programi olarak dogrutesti sinifi olusturarak bu sinifin

içinde **cokdogrubirdogru** isimli bir nesne olustur be bu nesnenin baslangiç ve bitis noktalariyla boyunu yazdir. Girdi degerlerini program içinde sabit olarak tanimla.

### Program 3.12 dogru.java

import nokta; public class dogru

{

protected nokta n1,n2;

//kurucu metot metot girdisi x1,y1,x2,y2

public dogru(double x1, double y1,double x2,double y2)

{

n1=new nokta(x1,y1); n2=new nokta(x2,y2);

}

//kurucu metot, method girdisi iki nokta public dogru(nokta nokta1,nokta nokta2)

{

n1=new nokta(nokta1.x,nokta1.y); n2=new nokta(nokta2.x,nokta2.y);

}

public void dogrugir(double x1, double y1,double x2,double y2)

{

n1.noktagir(x1,y1); n2.noktagir(x2,y2);

}

public String toString()

{

return "("+n1.toString()+","+n2.toString()+")";

}

public double boy()

{

return Math.sqrt((n1.Xoku()-n2.Xoku())\*(n1.Xoku()-n2.Xoku()) +

(n1.Yoku()-n2.Yoku())\*(n1.Yoku()-n2.Yoku()) );

}

}

### Program 3.13 dogrutesti.java

import java.io.\*; import nokta;

public class dogrutesti

{

public static void main(String args[]) throws IOException

{

dogru cokdogrubirdogru=new dogru(1.0,1.0,2.0,2.0); System.out.println(cokdogrubirdogru.toString()); System.out.println("dogru boyu : "+cokdogrubirdogru.boy());

}

}

([1.0,1.0],[2.0,2.0])

dogru boyu : 1.4142135623730951

1. Nokta sinifini kullanarak dikdortgen sinifini yarat. Dikdörtgen iki noktayla tanimlanir. Dikdortgen sinifinin kurucu metotunu, çevresini hesaplayan çevre metotunu, Alanini hesaplayan alan metotonu, dikdortgeni tanimlayan iki noktayi String degiskeni olarak tanimlayan toString metotunu ve gerekli olan diger metotlari tanimla. Applet programi olarak dikdortgentesti sinifi olusturarak bu sinifin içinde **cokguzelbirdikdortgen**

isimli bir nesne olustur be bu nesnenin baslangiç ve bitis noktalariyla boyunu yazdir. Girdi degerlerini program içinde sabit olarak tanimla.

Program 3.14 dikdortgenTesti.java import java.awt.Graphics;

import java.applet.Applet; import nokta;

class dikDortgen

{

protected nokta n1,n2;

//kurucu metot metot girdisi x1,y1,x2,y2

public dikDortgen(double x1, double y1,double x2,double y2)

{

n1=new nokta(x1,y1); n2=new nokta(x2,y2);

}

//kurucu metot, method girdisi iki nokta public dikDortgen(nokta nokta1,nokta nokta2)

{

n1=new nokta(nokta1.x,nokta1.y); n2=new nokta(nokta2.x,nokta2.y);

}

public void dikdortgengir(double x1, double y1,double x2,double y2)

{

n1.noktagir(x1,y1); n2.noktagir(x2,y2);

}

public String toString()

{

return "dikdortgen : ("+n1.toString()+","+n2.toString()+")";

}

public double alan()

{

return (n1.Xoku()-n2.Xoku())\*(n1.Yoku()-n2.Yoku());

}

public double cevre()

{

return 2\*(n1.Xoku()-n2.Xoku())+ 2\*(n1.Yoku()-n2.Yoku());

}

}

public class dikdortgentesti extends Applet

{

public void paint(Graphics g)

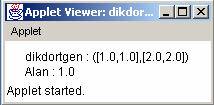
{

dikDortgen cokguzelbirdikdortgen=new dikDortgen(1,1,2,2); g.drawString(cokguzelbirdikdortgen.toString(),20,20); g.drawString("Alan : "+ cokguzelbirdikdortgen.alan(),20,35);

}

}

### 03007.JPG



**Sekil 3.7 dikdortgentesti.html çiktisi**

1. dogrutesti1 sinifini yaz. Bu sinifta dogruyu tanimliyan noktalari (toplam iki nokta) swing JoptionPane konsol programina ekrandan degisken olarak gir. Dogrunun iki noktasini ve dogru boyunu ekrana çikti olarak yazdir.
2. Dikdortgentesti1 sinifini yaz. Bu sinifta dogruyu tanimliyan uç noktalari JApplet programina disardan degisken olarak gir.
3. Hesapmakinesi sinifini yarat. Bu sinif iki gerçek degiskeni sinif degiskeni olarak tanimlasin. Topla,cikar,carp ve bol metotlarini yaz ve bu islemleri metotlarin içinde tanimla.

bir test programini yaz ve bu sinif hesapmakinesi sinifini kullanarak disaridan girilen iki gercek sayinin dört islemini hesapla**. dortislemhesapmakinasi** programi Program 3.5 de verilmistir.

Program 3.15 Dörtislem hesap makinasi appleti tanimlayan H4AL1.java programi import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir

import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import dortislemhesapmakinasi;

public class H4AL1 extends Applet implements ActionListener

{

// Dört islem hesap makinasi

TextField sonucgirdisi;

TextField kutugirdisi;

Button Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;

//Button sinifi degiskenleri double sayi; dortislemhesapmakinasi D;

// pencereyi baslatma metotu public void init()

{

D=new dortislemhesapmakinasi(); sonucgirdisi=new TextField(30); sonucgirdisi.setEditable(false); add(sonucgirdisi); kutugirdisi=new TextField(30);

add(kutugirdisi);//kutuyu pencereye yerlestir Gir=new Button("gir");

Arti=new Button("+"); Eksi=new Button("-"); Carpi=new Button("\*"); Bolu=new Button("/"); M=new Button("M"); MR=new Button("MR"); MArti=new Button("M+"); MEksi=new Button("M-"); Isaret=new Button("+/-"); C=new Button("C"); add(Gir);

add(Arti); add(Eksi); add(Carpi); add(Bolu); add(M);

add(MR); add(MArti); add(MEksi); add(Isaret); add(C);

Gir.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this); Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this); MArti.addActionListener(this); MEksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metotu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi); else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(); else if(e.getSource()==C) D.C(); sonucgirdisi.setText(D.toString());

if(e.getSource()==MR|| e.getSource()==M ||e.getSource()==MArti ||e.getSource()==MEksi) kutugirdisi.setText(D.MtoString());

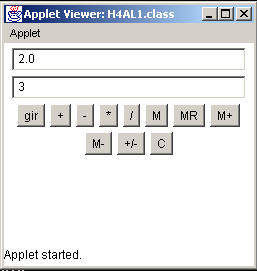
else

kutugirdisi.setText("");

}

}

### 03008.JPG



**Sekil 3.8 Dörtislem hesap makinasi çikti appleti H4AL1.html**

1. Hesapmakinesi sinifini için. bir JApplet test programi yaz ve bu sinif hesapmakinesi sinifini kullanarak disaridan girilen iki gercek sayinin dört islemini hesapla**. dortislemhesapmakinasi** programi yukarida verilmistir.

Program 3.16 Dörtislem hesap makinasi swing JAppleti tanimlayan H4AL1\_2000.java programi

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import dortislemhesapmakinasi;

public class H4AL1\_2000 extends JApplet implements ActionListener

{

// Dört islem hesap makinasi JTextField sonucgirdisi; JTextField kutugirdisi;

JButton Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;

//JButton sinifi degiskenleri double sayi; dortislemhesapmakinasi D;

// pencereyi baslatma metodu public void init()

{

D=new dortislemhesapmakinasi();

Container c=getContentPane(); c.setLayout(new FlowLayout()); sonucgirdisi=new JTextField(30); sonucgirdisi.setEditable(false); c.add(sonucgirdisi); kutugirdisi=new JTextField(30);

c.add(kutugirdisi);//kutuyu pencereye yerlestir Gir=new JButton("gir");

Arti=new JButton("+"); Eksi=new JButton("-"); Carpi=new JButton("\*"); Bolu=new JButton("/"); M=new JButton("M"); MR=new JButton("MR"); MArti=new JButton("M+"); MEksi=new JButton("M-"); Isaret=new JButton("+/-"); C=new JButton("C"); c.add(Gir);

c.add(Arti);

c.add(Eksi);

c.add(Carpi);

c.add(Bolu);

c.add(M);

c.add(MR);

c.add(MArti);

c.add(MEksi); c.add(Isaret); c.add(C);

Gir.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this); Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this);

MArti.addActionListener(this); MEksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi); else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(); else if(e.getSource()==C) D.C(); sonucgirdisi.setText(D.toString());

if(e.getSource()==MR|| e.getSource()==M ||e.getSource()==MArti ||e.getSource()==MEksi) kutugirdisi.setText(D.MtoString());

else

kutugirdisi.setText("");

}

}

1. Hesapmakinesi sinifini için. bir **JFrame** test programi yaz ve bu sinif hesapmakinesi sinifini kullanarak disaridan girilen iki gercek sayinin dört islemini hesapla**. dortislemhesapmakinasi** programi yukarida verilmistir.

Program 3.17 Dörtislem hesap makinasi appleti tanimlayan H4AL1SWF\_2000.java programi (java komutuyla çalistiracagiz)

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import dortislemhesapmakinasi;

import BasicWindowMonitor;

//dörtislem hesap makinasi sinifini çagirir

//swing frame versiyonu

public class H4AL1SWF\_2000 extends JFrame implements ActionListener

{

// Dört islem hesap makinasi JTextField sonucgirdisi; JTextField kutugirdisi;

JButton Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;

//JButton sinifi degiskenleri double sayi; dortislemhesapmakinasi D;

// pencereyi baslatma metodu public H4AL1SWF\_2000()

{

D=new dortislemhesapmakinasi();

Container c=getContentPane(); c.setLayout(new FlowLayout()); sonucgirdisi=new JTextField(30); sonucgirdisi.setEditable(false); c.add(sonucgirdisi); kutugirdisi=new JTextField(30);

c.add(kutugirdisi);//kutuyu pencereye yerlestir Gir=new JButton("gir");

Arti=new JButton("+"); Eksi=new JButton("-"); Carpi=new JButton("\*"); Bolu=new JButton("/"); M=new JButton("M"); MR=new JButton("MR"); MArti=new JButton("M+"); MEksi=new JButton("M-"); Isaret=new JButton("+/-"); C=new JButton("C"); c.add(Gir);

c.add(Arti);

c.add(Eksi);

c.add(Carpi);

c.add(Bolu);

c.add(M);

c.add(MR);

c.add(MArti);

c.add(MEksi); c.add(Isaret); c.add(C);

Gir.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this); Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this); MArti.addActionListener(this); MEksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi);

else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(); else if(e.getSource()==C) D.C(); sonucgirdisi.setText(D.toString());

if(e.getSource()==MR|| e.getSource()==M ||e.getSource()==MArti ||e.getSource()==MEksi) kutugirdisi.setText(D.MtoString());

else

kutugirdisi.setText("");

}

//=====================================================

public static void main(String[] args)

{

H4AL1SWF\_2000 pencere= new H4AL1SWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

}

1. Bayagi kesirler sanirim ilk okulda sizleri epey zorlamistir. Daha fazla aklimizi karistirmalarini önlemek amaciyla **bayagikesir** sinifini tanimlayalim. Bu sinif bayagi kesirlerin dört islemini ve sadelestirilmelerini yapabilme kapasitesinde olsun. Ayni zamanda bayagi kesirleri yazdiran toString metotunu da tanimlayalim. Bu metot [2/3] veya 3[5/4] seklinde bayagi kesirleri yazdirabilsin.

Bayagi kesirleri sadelestirirken en büyük ortak bölen hesabi gerekebilir. Bu amaçla asagidaki metot verilmistir.

public int enbuyukortakbolen(int n, int m)

{

// iki tam sayinin en buyuk ortak bolenini hesaplar if(n==0)

return m; if(m==0) return n; while(m != n)

{

if(nm)

n=n-m; else m=m-n;

}

return n;

}

not : sadelestirme islemi pay ve paydayi en büyük ortak bölene bölerek yapilir.

Bayagi kesir sinifinin calismasini kontrol amaciyla bayagikesirtest.java applet programini yaz. Bu programa iki bayagi kesir gir ve dort islemini hesapla.

Program 3.18 : bayagikesir.java,bayagikesir sinifi tanimi

//===========================================

// bayagikesir sinifi tanimi

// Dr. Turhan Coban

//===========================================

import java.io.\*;

// sinif bayagikesir

// bu sinif bayagikesir sayilarin matematik

// islemlerini tanimlar

//

class bayagikesir {

// sinif degiskenleri int pay;

int payda;

// kurucu metotlar

public bayagikesir()

{

pay=0; payda=1;

}

public bayagikesir(int npay,int npayda)

{

pay=npay; payda=npayda;

}

public bayagikesir(int numer)

{

pay=numer; payda=1;

}

public bayagikesir(bayagikesir c )

{

pay=c.Pay(); payda=c.Payda();

}

// giris- çikis metodlari

public int enbuyukortakbolen()

{

int n=pay; int m=payda;

// iki tam sayinin en buyuk ortak bolenini hesaplar if(n==0)

return m; if(m==0)

return n; while(m != n)

{

if(nm)

n=n-m; else

m=m-n;

}

return n;

}

public void sadelestir()

{

//sadelestir int isaret=1; if(pay<0)

{

isaret=-isaret;

pay=-pay;

}

if(payda<0)

{

isaret=-isaret; payda=-payda;

}

if(payda==0) {pay=0;payda=1;} int ebob=enbuyukortakbolen(); ebob=Math.abs(ebob); pay=isaret\*pay/ebob; payda=payda/ebob;

}

public int Pay()

{

return pay;

}

public int Payda()

{

return payda;

}

public void payGir(int r)

{

pay=r;

}

public void paydaGir(int i)

{

payda=i;

}

public void bayagikesirGir(bayagikesir sag)

{

pay=sag.Pay(); payda=sag.Payda();

}

public void bayagikesirGir(int nr,int ni)

{

pay=nr; payda=ni;

}

public void bayagikesirGir(int nr)

{

pay=nr; payda=1;

}

public void bayagikesirGir(double d) throws bayagikesirException

{

// tam sayinin bayagikesir esitini y• kle

// eh birazc• k yaklas• m var tabi bayagikesirGir((int)d\*10000,10000); sadelestir();

}

public double toDouble()

{

//bayagikesir sayinin gercek sayi esidi return ((double)pay/(double)payda);

}

public static bayagikesir BayagikesireCevir(double d) throws bayagikesirException

{

// tam sayinin bayagikesir esiti

// eh birazc• k yaklas• m var tabi bayagikesir b=new bayagikesir(); b.bayagikesirGir((int)d\*10000,10000); b.sadelestir();

return b;

}

public void topla(bayagikesir sag) throws bayagikesirException

{

//ikinci bir bayagikesirle topla

pay = pay\*sag.Payda() + sag.Pay()\*payda; payda = payda\*sag.Payda();

sadelestir();

}

public void topla(int sag) throws bayagikesirException

{

//bir gercek say• yla topla pay = pay + sag\*payda; sadelestir();

}

public void cikar(bayagikesir sag) throws bayagikesirException

{

//ikinci bir bayagikesirle topla

pay = pay\*sag.Payda() - sag.Pay()\*payda; payda = payda\*sag.Payda();

sadelestir();

}

public void cikar(int sag) throws bayagikesirException

{

//bir gercek say• yla topla pay = pay - sag\*payda; sadelestir();

}

public void carp(bayagikesir sag ) throws bayagikesirException

{

//bir bayagikesir sayiyla carp pay = pay\*sag.Pay();

payda = payda\*sag.Payda(); sadelestir();

}

public void carp(int sag ) throws bayagikesirException

{

//bir gercek sayiyla carp pay = pay\*sag; sadelestir();

}

public void bol(bayagikesir sag ) throws bayagikesirException

{

//bir bayagikesir say• yla bol pay = pay\*sag.Payda(); payda = payda\*sag.Pay(); sadelestir();

}

public void bol(int sag ) throws bayagikesirException

{

//bir Pay say• yla bol payda = payda\*sag; sadelestir();

}

public static bayagikesir topla(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{

// iki bayagikesir sayinin toplamini return deyimiyle bayagikesir olarak

// aktarir

int r1= sol.Pay() \* sag.Payda() + sag.Pay() \* sol.Payda(); int i1= sol.Payda()\* sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir topla(bayagikesir sol, int sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktarir

int r1=sol.Pay() + sag\*sol.Payda(); int i1=sol.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir topla(int sol, bayagikesir sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1=sag.Pay() + sol\*sag.Payda(); int i1=sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

//===================================

public static bayagikesir cikar(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{

// iki bayagikesir sayinin farkini return deyimiyle bayagikesir olarak

// aktarir

int r1= sol.Pay() \* sag.Payda() - sag.Pay() \* sol.Payda(); int i1= sol.Payda()\* sag.Payda();

bayagikesir sonuc;

sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir cikar(bayagikesir sol, int sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay sayinin toplamini return deyimiyle bayagikesir olarak

// aktarir

int r1=sol.Pay() - sag\*sol.Payda(); int i1=sol.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir cikar(int sol, bayagikesir sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay sayinin toplamini return deyimiyle bayagikesir olarak

// aktarir

int r1=sag.Pay() - sol\*sag.Payda(); int i1=sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

//===================================

public static bayagikesir carp(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{ // iki bayagikesir sayinin carpimini aktar bayagikesir sonuc;

sonuc=new bayagikesir(sol.Pay()\*sag.Pay(),sol.Payda()\*sag.Payda()); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir carp(bayagikesir sol, int sag) throws bayagikesirException

{ // bir bayagikesir ve bir double sayinin carpimini aktar bayagikesir sonuc;

sonuc=new bayagikesir(sol.Pay()\*sag,sol.Payda()); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir carp(int sol, bayagikesir sag) throws bayagikesirException

{ // bir bayagikesir ve bir double sayinin carpimini aktar bayagikesir sonuc;

sonuc=new bayagikesir(sag.Pay()\*sol,sag.Payda()); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir bol(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{ // iki bayagikesir sayinin b”l• mlerini aktar

double a=sag.Pay()\*sag.Pay()+sag.Payda()\*sag.Payda(); bayagikesir sonuc;

sonuc=new bayagikesir((int)(sol.Pay()\*sag.Payda()),(int)(sol.Payda()\*sag.Pay())); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir bol(bayagikesir sol, int sag) throws bayagikesirException

{ // bayagikesir sayiyi double sayiya bol bayagikesir sonuc;

sonuc=new bayagikesir(sol.Pay(),(sol.Payda()\*sag)); sonuc.sadelestir();

return sonuc;

}

public boolean kucuktur(bayagikesir sol,bayagikesir sag)

{

// less then comparison of two bayagikesir numbers return (sol.toDouble() < sag.toDouble());

}

public boolean kucuktur\_esittir(bayagikesir sol,bayagikesir sag)

{

// less then and esittir comparison of two bayagikesir numbers return (sol.toDouble() <= sag.toDouble());

}

public boolean buyuktur(bayagikesir sol,bayagikesir sag)

{

// buyuktur then comparison of two bayagikesir numbers return sol.toDouble() sag.toDouble();

}

public boolean buyuktur\_esittir(bayagikesir sol,bayagikesir sag)

{

// buyuktur then and esittir comparison of two bayagikesir numbers return sol.toDouble() = sag.toDouble();

}

public boolean esittir(bayagikesir sol,bayagikesir sag)

{

// esittir comparison of two bayagikesir numbers return sol.toDouble() == sag.toDouble();

}

public boolean esit\_degildir(bayagikesir sol,bayagikesir sag)

{

// not esittir comparison of two bayagikesir numbers return sol.toDouble() != sag.toDouble();

}

public static String toString(bayagikesir value)

{

String b=""; if(Math.abs(value.Payda())!=1)

{

b=b+"( "+value.Pay()+" / "+value.Payda()+" )";

}

else

{

b=b+value.Pay()+" ";

}

return b;

}

public String toString()

{

// yazima haz• r bayagikesir formda String de§iskeni iletir. String b="";

if(Math.abs(Payda())!=1)

{

b=b+"( "+pay+" / "+Payda()+" )";

}

else

{

b=b+Pay()+" ";

}

return b;

}

};

//bayagikesir sinifinin taniminin sonu

Program 3.19 :Bayagikesir sinifi test programi bkTesti.java

// Dr. Turhan Çoban

// Bayagi kesir testi

//--------------------------------------------------------------------------------------------

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import bayagikesir;

public class bkTesti extends Applet implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

Label kutubasligi1; //Label sinifi degiskeni (nesnesi) kutubasligi1 TextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 Label kutubasligi2; //Label sinifi degiskeni (nesnesi) kutubasligi2 TextField kutugirdisi2; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 TextField kutugirdisi3; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 TextField kutugirdisi4; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 bayagikesir k1; // bayagikesir degisken k1

bayagikesir k2; // bayagikesir degisken k2

//=====================================================

// pencereyi baslatma metotu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

k1=new bayagikesir(1,2); k2=new bayagikesir(2,4);

kutubasligi1=new Label("Birinci bayagikesir sayiyi giriniz : "); add(kutubasligi1);

kutugirdisi1=new TextField(5); kutugirdisi1.addActionListener(this); add(kutugirdisi1);

kutugirdisi2=new TextField(5); kutugirdisi2.addActionListener(this); add(kutugirdisi2);

kutubasligi2=new Label("Ikinci bayagikesir sayiyi giriniz : "); add(kutubasligi2);

kutugirdisi3=new TextField(5);

kutugirdisi3.addActionListener(this); add(kutugirdisi3);

kutugirdisi4=new TextField(5); kutugirdisi4.addActionListener(this); add(kutugirdisi4);

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metotu

// Her yeni rakam girildiginde bu metot cagirilir public void actionPerformed(ActionEvent e)

{

Integer sayi1,sayi2,sayi3,sayi4;

sayi1=new Integer(kutugirdisi1.getText() ); sayi2=new Integer(kutugirdisi2.getText() ); sayi3=new Integer(kutugirdisi3.getText() ); sayi4=new Integer(kutugirdisi4.getText() ); showStatus("Hesaplaniyor...."); k1.bayagikesirGir(sayi1.intValue(),sayi2.intValue()); k2.bayagikesirGir(sayi3.intValue(),sayi4.intValue()); showStatus("Hesaplandi");

repaint();//pencereyi yeniden paint metotuna gore ciz

}

//=====================================================

// paint (boya=pencereye ciz metotu) public void paint(Graphics g)

{

g.drawString("ilk sayi : "+k1.toString(),25,80); g.drawString("ikinci sayi : "+k2.toString(),25,95); try{

g.drawString("iki sayinin toplami : "+bayagikesir.toString(bayagikesir.topla(k1,k2)),25,110); g.drawString("iki sayinin farki : "+bayagikesir.toString(bayagikesir.cikar(k1,k2)),25,125); g.drawString("iki sayinin carpimi : "+bayagikesir.toString(bayagikesir.carp(k1,k2)),25,140 ); g.drawString("iki sayinin bolumu : "+bayagikesir.toString(bayagikesir.bol(k1,k2)),25,155 );

}

catch (bayagikesirException e)

{

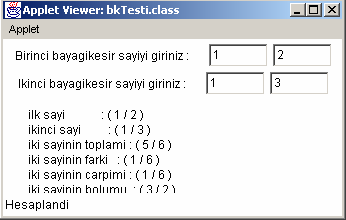
showStatus(e.getMessage());

}

}

}

### 03009.JPG



**Sekil 3.9 Bayagi kesir test programi bkTest.html applet çiktisi**

Program 3.20 :Bayagikesir sinifi Java swing applet test programi bkTesti\_2000.java

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import bayagikesir;

public class bkTesti\_2000 extends JApplet implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

JLabel kutubasligi1; //JLabel sinifi degiskeni (nesnesi) kutubasligi1 JTextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 JLabel kutubasligi2; //JLabel sinifi degiskeni (nesnesi) kutubasligi2 JTextField kutugirdisi2; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi3; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi4; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextArea cikti; // JTextArea sinifi degiskeni (nesnesi) cikti bayagikesir k1; // bayagikesir degisken k1

bayagikesir k2; // bayagikesir degisken k2

//=====================================================

// pencereyi baslatma metodu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

k1=new bayagikesir(1,2); k2=new bayagikesir(2,4); Container c=getContentPane(); c.setLayout(new FlowLayout());

kutubasligi1=new JLabel("Birinci bayagikesir sayiyi giriniz : "); c.add(kutubasligi1);

kutugirdisi1=new JTextField(5); kutugirdisi1.addActionListener(this); c.add(kutugirdisi1); kutugirdisi2=new JTextField(5); kutugirdisi2.addActionListener(this); c.add(kutugirdisi2);

kutubasligi2=new JLabel("Ikinci bayagikesir sayiyi giriniz : "); c.add(kutubasligi2);

kutugirdisi3=new JTextField(5); kutugirdisi3.addActionListener(this); c.add(kutugirdisi3); kutugirdisi4=new JTextField(5); kutugirdisi4.addActionListener(this); c.add(kutugirdisi4);

cikti=new JTextField(" ");

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metodu

// Her yeni rakam girildiginde bu metod cagirilir public void actionPerformed(ActionEvent e)

{

Integer sayi1,sayi2,sayi3,sayi4;

sayi1=new Integer(kutugirdisi1.getText() ); sayi2=new Integer(kutugirdisi2.getText() ); sayi3=new Integer(kutugirdisi3.getText() ); sayi4=new Integer(kutugirdisi4.getText() );

showStatus("Hesaplaniyor...."); k1.bayagikesirGir(sayi1.intValue(),sayi2.intValue()); k2.bayagikesirGir(sayi3.intValue(),sayi4.intValue()); showStatus("Hesaplandi");

String s="";

s+="ilk sayi : "+k1.toString()+"\n"; s+="ikinci sayi : "+k2.toString();

s+="iki sayinin toplami : "+bayagikesir.toString(bayagikesir.topla(k1,k2))+"\n"; s+="iki sayinin farki : "+bayagikesir.toString(bayagikesir.cikar(k1,k2))+"\n"; s+="iki sayinin carpimi : "+bayagikesir.toString(bayagikesir.carp(k1,k2))+"\n"; s+="iki sayinin bolumu : "+bayagikesir.toString(bayagikesir.bol(k1,k2))+"\n"; cikti.setText(s);

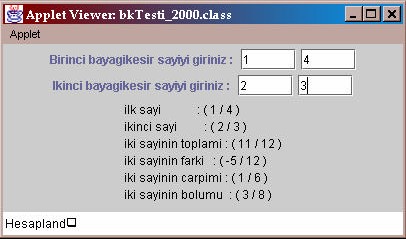
repaint();//pencereyi yeniden paint metoduna gore ciz

}

//=====================================================

}

### 03010.JPG



Sekil 3.10 bkTesti\_2000 swing JApplet çiktisi

Program 3.21 :Bayagikesir sinifi test programi bkTestiSWF\_2000.java swing Frame test programi import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import bayagikesir;

import BasicWindowMonitor;

public class bkTestiSWF\_2000 extends JFrame implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

JLabel kutubasligi1; //JLabel sinifi degiskeni (nesnesi) kutubasligi1 JTextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 JLabel kutubasligi2; //JLabel sinifi degiskeni (nesnesi) kutubasligi2 JTextField kutugirdisi2; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi3; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi4; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextArea cikti; // JTextArea sinifi degiskeni (nesnesi) cikti bayagikesir k1; // bayagikesir degisken k1

bayagikesir k2; // bayagikesir degisken k2

//=====================================================

// pencereyi baslatma metodu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public bkTestiSWF\_2000()

{

super("bayagi kesir Testi bkTestiSWF\_2000, frame format"); k1=new bayagikesir(1,2);

k2=new bayagikesir(2,4); Container c=getContentPane(); c.setLayout(new FlowLayout());

kutubasligi1=new JLabel("Birinci bayagikesir sayiyi giriniz : "); c.add(kutubasligi1);

kutugirdisi1=new JTextField(5); kutugirdisi1.addActionListener(this); c.add(kutugirdisi1); kutugirdisi2=new JTextField(5); kutugirdisi2.addActionListener(this); c.add(kutugirdisi2);

kutubasligi2=new JLabel("Ikinci bayagikesir sayiyi giriniz : "); c.add(kutubasligi2);

kutugirdisi3=new JTextField(5); kutugirdisi3.addActionListener(this); c.add(kutugirdisi3); kutugirdisi4=new JTextField(5); kutugirdisi4.addActionListener(this); c.add(kutugirdisi4);

cikti=new JTextArea(toString()); cikti.setBackground(c.getBackground()); c.add(cikti);

}

//=====================================================

public String toString()

{

String s="";

s+="ilk sayi : "+k1.toString()+"\n"; s+="ikinci sayi : "+k2.toString()+"\n";

s+="iki sayinin toplami : "+bayagikesir.toString(bayagikesir.topla(k1,k2))+"\n"; s+="iki sayinin farki : "+bayagikesir.toString(bayagikesir.cikar(k1,k2))+"\n"; s+="iki sayinin carpimi : "+bayagikesir.toString(bayagikesir.carp(k1,k2))+"\n"; s+="iki sayinin bolumu : "+bayagikesir.toString(bayagikesir.bol(k1,k2))+"\n"; return s;

}

// girdi alanindaki olan olaylari dinleme metodu

// Her yeni rakam girildiginde bu metod cagirilir

public void actionPerformed(ActionEvent e)

{

Integer sayi1,sayi2,sayi3,sayi4;

sayi1=new Integer(kutugirdisi1.getText() ); sayi2=new Integer(kutugirdisi2.getText() ); sayi3=new Integer(kutugirdisi3.getText() ); sayi4=new Integer(kutugirdisi4.getText() ); k1.bayagikesirGir(sayi1.intValue(),sayi2.intValue()); k2.bayagikesirGir(sayi3.intValue(),sayi4.intValue()); cikti.setText(toString());

}

//=====================================================

public static void main(String[] args)

{

bkTestiSWF\_2000 pencere= new bkTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 03011.JPG



**Sekil 3.11 bkTestiSWF\_2000 swing JFrame çiktisi**

Kompleks sinifi örnek olarak burada verilmistir. Problem 11 ve 12 ile ilgili önemli bir nokta bunlarin gerçek hayatta kullanilabilecek siniflar olmalaridir. Sadece bir eksersiz olarak kullanilmak amaciyla degil, bu siniflara ihtiyacimiz oldugunda bize hizmet vermek amaciyla ve tüm detay metodlari düsünülerek yazilmislardir. Bu siniflar her zaman kütüphanenizde kalabilecek ve gerektiginde direk olarak gerçek programlarinizda kullanilabilecek özelliklerde siniflardir:

Program 3.22 : kompleks sinifinin tanimi : kompleks.java

/===========================================

// kompleks sinifi tanimi

// Dr. Turhan Coban

//===========================================

import java.io.\*;

// sinif kompleks

// bu sinif kompleks sayilarin matematik

// islemlerini tanimlar

//

class kompleks {

// sinif degiskenleri

double ger; // kompleks sayinin gerçek kismi double imaj; // kompleks sayinin imajineri kismi

// kurucu metotlar public kompleks()

{

ger=0; imaj=0;

}

public kompleks(double nger,double nimaj)

{

ger=nger; imaj=nimaj;

}

public kompleks(double numer)

{

ger=numer; imaj=0;

}

public kompleks(kompleks c )

{

ger=c.gercek(); imaj=c.imajineri();

}

// giris - çikis metotlari public double gercek()

{

return ger;

}

public double imajineri()

{

return imaj;

}

public void gercekgir(double r)

{

ger=r;

}

public void imajinerigir(double i)

{

imaj=i;

}

public void kompleksgir(kompleks sag)

{

ger=sag.gercek(); imaj=sag.imajineri();

}

public void kompleksgir(double nr,double ni)

{

ger=nr; imaj=ni;

}

public double R()

{

//kompleks sayinin polar koordinat cap esidi return Math.sqrt(ger\*ger+imaj\*imaj);

}

public double theta()

{

//kompleks sayinin polar koordinat radyan aci esidi return Math.atan2(imaj,ger);

}

public double dtheta()

{

//kompleks sayinin polar koordinat derece aci esidi return Math.atan2(imaj,ger)\*45.0/Math.atan(1.0);

}

public void topla(kompleks sag)

{

//ikinci bir kompleks say• yla topla imaj = imaj + sag.imajineri();

ger = ger + sag.gercek();

}

public void topla(double sag)

{

//bir gercek say• yla topla ger = ger + sag;

}

public void cikar(kompleks sag)

{

imaj = imaj - sag.imajineri(); ger = ger - sag.gercek();

}

public void cikar(double sag)

{

//bir gercek say• y• cikar ger = ger - sag;

}

public void carp(kompleks sag )

{

//bir kompleks sayiyla carp

ger = ger\*sag.gercek() - imaj\*sag.imajineri(); imaj = ger\*sag.imajineri() + imaj\*sag.gercek();

}

public void carp(double sag )

{

//bir gercek sayiyla carp ger = ger\*sag;

imaj = imaj\*sag;

}

public void bol(kompleks sag )

{

//bir kompleks say• yla bol double a=ger\*ger+imaj\*imaj;

ger = ( ger\*sag.gercek() + imaj\*sag.imajineri())/a; imaj = (-ger\*sag.imajineri() + imaj\*sag.gercek())/a;

}

public void bol(double sag )

{

//bir gercek say• yla bol ger = ger/sag;

imaj = imaj/sag;

}

public static kompleks topla(kompleks sol, kompleks sag)

{

// iki kompleks say• n• n toplam• n• return deyimiyle kompleks olarak

// aktar• r

double r1=(sol.gercek() + sag.gercek()); double i1=(sol.imajineri() + sag.imajineri()); kompleks sonuc;

sonuc=new kompleks(r1,i1); return sonuc;

}

public static kompleks topla(kompleks sol, double sag)

{

// bir kompleks ve bir gercek say• n• n toplam• n• return deyimiyle kompleks olarak

// aktar• r

double r1=(sol.gercek() + sag); double i1=sol.imajineri(); kompleks sonuc;

sonuc=new kompleks(r1,i1); return sonuc;

}

public static kompleks topla(double sol, kompleks sag)

{

// bir kompleks ve bir gercek say• n• n toplam• n• return deyimiyle kompleks olarak

// aktar• r

double r1=(sag.gercek() + sol); double i1=sag.imajineri(); kompleks sonuc;

sonuc=new kompleks(r1,i1); return sonuc;

}

public static kompleks cikar(kompleks sol, kompleks sag)

{ // iki kompleks sayinin sonuclarini aktar kompleks sonuc;

sonuc=new kompleks((sol.gercek() - sag.gercek()), (sol.imajineri() - sag.imajineri()));

return sonuc;

}

public static kompleks cikar(kompleks sol, double sag)

{ // bir kompleks sayidan bir gercek sayiyi cikar sonuclarini aktar kompleks sonuc;

sonuc=new kompleks((sol.gercek() - sag),sol.imajineri()); return sonuc;

}

public static kompleks cikar(double sol, kompleks sag)

{ // bir double sayidan bir kompleks sayiyi cikar sonuclarini aktar kompleks sonuc;

sonuc=new kompleks((sol - sag.gercek()),-sag.imajineri()); return sonuc;

}

public static kompleks carp(kompleks sol, kompleks sag)

{ // iki kompleks sayinin carpimini aktar kompleks sonuc;

sonuc=new kompleks

((sol.gercek()\*sag.gercek() - sol.imajineri()\*sag.imajineri()), (sol.gercek()\*sag.imajineri() + sol.imajineri()\*sag.gercek())); return sonuc;

}

public static kompleks carp(kompleks sol, double sag)

{ // bir kompleks ve bir double sayinin carpimini aktar kompleks sonuc;

sonuc=new kompleks(sol.gercek()\*sag,sol.imajineri()\*sag); return sonuc;

}

public static kompleks carp(double sol, kompleks sag)

{ // bir kompleks ve bir double sayinin carpimini aktar kompleks sonuc;

sonuc=new kompleks(sag.gercek()\*sol,sag.imajineri()\*sol); return sonuc;

}

public static kompleks bol(kompleks sol, kompleks sag)

{ // iki kompleks sayinin b”l• mlerini aktar

double a=sag.gercek()\*sag.gercek()+sag.imajineri()\*sag.imajineri(); kompleks sonuc;

sonuc=new kompleks

((sol.gercek()\*sag.gercek() + sol.imajineri()\*sag.imajineri())/a, (-sol.gercek()\*sag.imajineri() + sol.imajineri()\*sag.gercek())/a); return sonuc;

}

public static kompleks bol(kompleks sol, double sag)

{ // kompleks say• y• gercek say• ya bol kompleks sonuc;

sonuc=new kompleks(sol.gercek()/sag,sol.imajineri()/sag); return sonuc;

}

public static kompleks pow(kompleks sol, double sag)

{ // kompleks bir say• n• n gercek kuvvetini hesaplar. double Rad,th;

Rad=Math.pow(sol.R(),sag); th=sag\*sol.theta(); kompleks sonuc;

sonuc =new kompleks((Rad\*Math.cos(th) ), (Rad\*Math.sin(th) ) );

return sonuc;

}

public boolean kucuktur(kompleks sol,kompleks sag)

{

// less then comparison of two kompleks numbers return (sol.R() < sag.R());

}

public boolean kucuktur\_esittir(kompleks sol,kompleks sag)

{

// less then and esittir comparison of two kompleks numbers return (sol.R() <= sag.R());

}

public boolean buyuktur(kompleks sol,kompleks sag)

{

// buyuktur then comparison of two kompleks numbers return sol.R() sag.R();

}

public boolean buyuktur\_esittir(kompleks sol,kompleks sag)

{

// buyuktur then and esittir comparison of two kompleks numbers return sol.R() = sag.R();

}

public boolean esittir(kompleks sol,kompleks sag)

{

// esittir comparison of two kompleks numbers return sol.R() == sag.R();

}

public boolean esit\_degildir(kompleks sol,kompleks sag)

{

// not esittir comparison of two kompleks numbers return sol.R() != sag.R();

}

public static String toString(kompleks value)

{

String b=""; if(Math.abs(value.imajineri())!=1)

{

if(value.imajineri() = 0)

b=b+"("+value.gercek()+" + "+value.imajineri()+"i )"; else

b=b+"("+value.gercek()+" - "+(-value.imajineri())+"i )";

}

else

{

if(value.imajineri() = 0) b=b+"("+value.gercek()+" + i )"; else

b=b+"("+value.gercek()+" - i )";

}

return b;

}

public String toString()

{

// yazima haz• r kompleks formda String de§iskeni iletir. String b="";

if(Math.abs(imajineri())!=1)

{

if(imajineri() 0)

b=b+"("+gercek()+" + "+imajineri()+"i )"; else if(imajineri() <0 )

b=b+"("+gercek()+" - "+(-imajineri())+"i )"; else

b=b+gercek()+ " ";

}

else

{

if(imajineri() 0) b=b+"("+gercek()+" + i )"; else if(imajineri() < 0) b=b+"("+gercek()+" - i )"; else

b=b+gercek()+" ";

}

return b;

}

};

//kompleks sinifinin taniminin sonu

### Program 3.21 : kompleks sinifini test eden H8A1.java programi

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import kompleks;

public class H8A1 extends Applet implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum s• n• fa aittir

Label kutubasligi1; //Label sinifi degiskeni (nesnesi) kutubasligi1 TextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 Label kutubasligi2; //Label sinifi degiskeni (nesnesi) kutubasligi2 TextField kutugirdisi2; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 TextField kutugirdisi3; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 TextField kutugirdisi4; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 kompleks k1; // kompleks degisken k1

kompleks k2; // kompleks degisken k2

//=====================================================

// pencereyi baslatma metotu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

k1=new kompleks(); k2=new kompleks();

kutubasligi1=new Label("Birinci kompleks sayiyi giriniz : "); add(kutubasligi1);

kutugirdisi1=new TextField(5); kutugirdisi1.addActionListener(this); add(kutugirdisi1);

kutugirdisi2=new TextField(5); kutugirdisi2.addActionListener(this); add(kutugirdisi2);

kutubasligi2=new Label("Ikinci kompleks sayiyi giriniz : "); add(kutubasligi2);

kutugirdisi3=new TextField(5); kutugirdisi3.addActionListener(this); add(kutugirdisi3);

kutugirdisi4=new TextField(5); kutugirdisi4.addActionListener(this); add(kutugirdisi4);

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metotu

// Her yeni rakam girildiginde bu metot cagirilir public void actionPerformed(ActionEvent e)

{

Double sayi1=new Double(kutugirdisi1.getText() ); Double sayi2=new Double(kutugirdisi2.getText() ); Double sayi3=new Double(kutugirdisi3.getText() ); Double sayi4=new Double(kutugirdisi4.getText() ); k1.kompleksgir(sayi1.doubleValue(),sayi2.doubleValue()); k2.kompleksgir(sayi3.doubleValue(),sayi4.doubleValue()); repaint();//pencereyi yeniden paint metotuna gore ciz

}

//=====================================================

// paint (boya=pencereye ciz metotu) public void paint(Graphics g)

{

g.drawString("ilk sayi : "+k1.toString(),25,80); g.drawString("ikinci sayi : "+k2.toString(),25,95);

g.drawString("iki sayinin toplami : "+kompleks.toString(kompleks.topla(k1,k2)),25,110); g.drawString("iki sayinin farki : "+kompleks.toString(kompleks.cikar(k1,k2)),25,125); g.drawString("iki sayinin carpimi : "+kompleks.toString(kompleks.carp(k1,k2)),25,140 ); g.drawString("iki sayinin bolumu : "+kompleks.toString(kompleks.bol(k1,k2)),25,155 );

}

}

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import kompleks;

public class H8A1 extends Applet implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum s• n• fa aittir

Label kutubasligi1; //Label sinifi degiskeni (nesnesi) kutubasligi1

TextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 Label kutubasligi2; //Label sinifi degiskeni (nesnesi) kutubasligi2 TextField kutugirdisi2; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 TextField kutugirdisi3; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 TextField kutugirdisi4; // Textfield sinifi degiskeni (nesnesi) kutugirdisi2 kompleks k1; // kompleks degisken k1

kompleks k2; // kompleks degisken k2

//=====================================================

// pencereyi baslatma metotu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

k1=new kompleks(); k2=new kompleks();

kutubasligi1=new Label("Birinci kompleks sayiyi giriniz : "); add(kutubasligi1);

kutugirdisi1=new TextField(5); kutugirdisi1.addActionListener(this); add(kutugirdisi1);

kutugirdisi2=new TextField(5); kutugirdisi2.addActionListener(this); add(kutugirdisi2);

kutubasligi2=new Label("Ikinci kompleks sayiyi giriniz : "); add(kutubasligi2);

kutugirdisi3=new TextField(5); kutugirdisi3.addActionListener(this); add(kutugirdisi3);

kutugirdisi4=new TextField(5); kutugirdisi4.addActionListener(this); add(kutugirdisi4);

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metotu

// Her yeni rakam girildiginde bu metot cagirilir public void actionPerformed(ActionEvent e)

{

Double sayi1=new Double(kutugirdisi1.getText() ); Double sayi2=new Double(kutugirdisi2.getText() ); Double sayi3=new Double(kutugirdisi3.getText() ); Double sayi4=new Double(kutugirdisi4.getText() ); k1.kompleksgir(sayi1.doubleValue(),sayi2.doubleValue()); k2.kompleksgir(sayi3.doubleValue(),sayi4.doubleValue()); repaint();//pencereyi yeniden paint metotuna gore ciz

}

//=====================================================

// paint (boya=pencereye ciz metotu) public void paint(Graphics g)

{

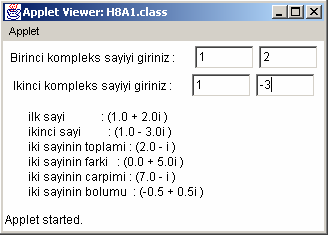
g.drawString("ilk sayi : "+k1.toString(),25,80); g.drawString("ikinci sayi : "+k2.toString(),25,95);

g.drawString("iki sayinin toplami : "+kompleks.toString(kompleks.topla(k1,k2)),25,110); g.drawString("iki sayinin farki : "+kompleks.toString(kompleks.cikar(k1,k2)),25,125); g.drawString("iki sayinin carpimi : "+kompleks.toString(kompleks.carp(k1,k2)),25,140 ); g.drawString("iki sayinin bolumu : "+kompleks.toString(kompleks.bol(k1,k2)),25,155 );

}

}

03012.JPG



### Sekil 3.12 Kompleks sinifi testi H8A1.html applet çiktisi

**Program 3.23 : kompleks sinifini test eden java swing applet programi kompleksTesti\_2000.java**

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import kompleks;

public class kompleksTesti\_2000 extends JApplet implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

JLabel kutubasligi1; //JLabel sinifi degiskeni (nesnesi) kutubasligi1 JTextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 JLabel kutubasligi2; //JLabel sinifi degiskeni (nesnesi) kutubasligi2 JTextField kutugirdisi2; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi3; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi4; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextArea cikti; // JTextArea sinifi degiskeni (nesnesi) cikti kompleks k1; // kompleks degisken k1

kompleks k2; // kompleks degisken k2

//=====================================================

// pencereyi baslatma metodu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public void init()

{

k1=new kompleks(1,2); k2=new kompleks(2,4); Container c=getContentPane();

c.setLayout(new FlowLayout());

kutubasligi1=new JLabel("Birinci kompleks sayiyi giriniz : "); c.add(kutubasligi1);

kutugirdisi1=new JTextField(5); kutugirdisi1.addActionListener(this); c.add(kutugirdisi1); kutugirdisi2=new JTextField(5); kutugirdisi2.addActionListener(this); c.add(kutugirdisi2);

kutubasligi2=new JLabel("Ikinci kompleks sayiyi giriniz : "); c.add(kutubasligi2);

kutugirdisi3=new JTextField(5); kutugirdisi3.addActionListener(this); c.add(kutugirdisi3); kutugirdisi4=new JTextField(5); kutugirdisi4.addActionListener(this); c.add(kutugirdisi4);

cikti=new JTextArea( toString()); cikti.setBackground(c.getBackground()); c.add(cikti);

}

public String toString()

{

String s="";

s+="ilk sayi : "+k1.toString()+"\n"; s+="ikinci sayi : "+k2.toString()+"\n";

s+="iki sayinin toplami : "+kompleks.toString(kompleks.topla(k1,k2))+"\n"; s+="iki sayinin farki : "+kompleks.toString(kompleks.cikar(k1,k2))+"\n"; s+="iki sayinin carpimi : "+kompleks.toString(kompleks.carp(k1,k2))+"\n"; s+="iki sayinin bolumu : "+kompleks.toString(kompleks.bol(k1,k2))+"\n"; return s;

}

//=====================================================

// girdi alanindaki olan olaylari dinleme metodu

// Her yeni rakam girildiginde bu metod cagirilir public void actionPerformed(ActionEvent e)

{

Double sayi1,sayi2,sayi3,sayi4;

sayi1=new Double(kutugirdisi1.getText() ); sayi2=new Double(kutugirdisi2.getText() ); sayi3=new Double(kutugirdisi3.getText() ); sayi4=new Double(kutugirdisi4.getText() ); showStatus("Hesaplaniyor....");

k1.kompleksgir(sayi1.doubleValue(),sayi2.doubleValue()); k2.kompleksgir(sayi3.doubleValue(),sayi4.doubleValue()); showStatus("Hesaplandi");

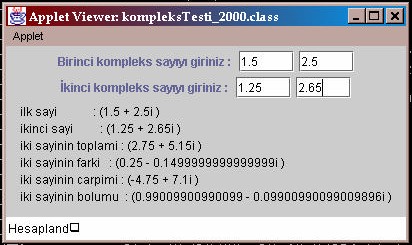
cikti.setText(toString()); repaint();//pencereyi yeniden çiz

}

//=====================================================

}

03013.JPG



### Sekil 3.13 Kompleks sinifi testi kompleksTesti\_2000.html JApplet çiktisi

**Program 3.24 : kompleks sinifini test eden kompleksTestiSWF\_2000.java JFrame programi**

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import kompleks;

import BasicWindowMonitor;

public class kompleksTestiSWF\_2000 extends JFrame implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

JLabel kutubasligi1; //JLabel sinifi degiskeni (nesnesi) kutubasligi1 JTextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 JLabel kutubasligi2; //JLabel sinifi degiskeni (nesnesi) kutubasligi2 JTextField kutugirdisi2; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi3; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextField kutugirdisi4; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextArea cikti; // JTextArea sinifi degiskeni (nesnesi) cikti kompleks k1; // kompleks degisken k1

kompleks k2; // kompleks degisken k2

//=====================================================

// pencereyi baslatma metodu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public kompleksTestiSWF\_2000()

{

super("kompleks sayi Testi kompleksTestiSWF\_2000, frame format"); k1=new kompleks(1,2);

k2=new kompleks(2,4); Container c=getContentPane(); c.setLayout(new FlowLayout());

kutubasligi1=new JLabel("Birinci kompleks sayiyi giriniz : "); c.add(kutubasligi1);

kutugirdisi1=new JTextField(5); kutugirdisi1.addActionListener(this); c.add(kutugirdisi1); kutugirdisi2=new JTextField(5); kutugirdisi2.addActionListener(this); c.add(kutugirdisi2);

kutubasligi2=new JLabel("Ikinci kompleks sayiyi giriniz : "); c.add(kutubasligi2);

kutugirdisi3=new JTextField(5); kutugirdisi3.addActionListener(this); c.add(kutugirdisi3); kutugirdisi4=new JTextField(5); kutugirdisi4.addActionListener(this); c.add(kutugirdisi4);

cikti=new JTextArea(toString()); cikti.setBackground(c.getBackground()); c.add(cikti);

}

//=====================================================

public String toString()

{

String s="";

s+="ilk sayi : "+k1.toString()+"\n"; s+="ikinci sayi : "+k2.toString()+"\n";

s+="iki sayinin toplami : "+kompleks.toString(kompleks.topla(k1,k2))+"\n"; s+="iki sayinin farki : "+kompleks.toString(kompleks.cikar(k1,k2))+"\n"; s+="iki sayinin carpimi : "+kompleks.toString(kompleks.carp(k1,k2))+"\n"; s+="iki sayinin bolumu : "+kompleks.toString(kompleks.bol(k1,k2))+"\n"; return s;

}

// girdi alanindaki olan olaylari dinleme metodu

// Her yeni rakam girildiginde bu metod cagirilir

public void actionPerformed(ActionEvent e)

{

Double sayi1,sayi2,sayi3,sayi4;

sayi1=new Double(kutugirdisi1.getText() ); sayi2=new Double(kutugirdisi2.getText() ); sayi3=new Double(kutugirdisi3.getText() ); sayi4=new Double(kutugirdisi4.getText() );

k1.kompleksgir(sayi1.doubleValue(),sayi2.doubleValue()); k2.kompleksgir(sayi3.doubleValue(),sayi4.doubleValue()); cikti.setText(toString());

}

//=====================================================

public static void main(String[] args)

{

kompleksTestiSWF\_2000 pencere= new kompleksTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 03014.JPG



**Sekil 3.14 Kompleks sinifi testi kompleksTestiSWF\_2000.html JFrame konsol çiktisi**

**13.** Kalitim yoluyla dortislemhesapmakinasi sinifindan türetilen bilimselhesapmakinasi1 sinifi verilmistir. Bu sinifi inceleyiniz.

### Program 3.25 : bilimselhesapmakinasi1.java programi

import dortislemhesapmakinasi;

public class bilimselhesapmakinasi1 extends dortislemhesapmakinasi

{

//bu metot miras (inheritance) yoluyla aktarma yapmaktadir.

// kurucu metot dortislemhesapmakinesi double arasonuc;

double arasonuc2;

public bilimselhesapmakinasi1(double y)

{

//constructor super(y); arasonuc=0; arasonuc2=0;

}

public bilimselhesapmakinasi1()

{

super(); arasonuc=0; arasonuc2=0;

}

public double topla(double x)

{

if(arasonuc2==0) {sonuc+=x;}

else {sonuc+=Math.pow(arasonuc2,x);arasonuc2=0;} return sonuc;

}

public double cikar(double x)

{

if(arasonuc2==0) {sonuc-=x;}

else {sonuc-=Math.pow(arasonuc2,x);arasonuc2=0;} return sonuc;

}

public double carp(double x)

{

if(arasonuc2==0) {sonuc\*=x;}

else {sonuc\*=Math.pow(arasonuc2,x);arasonuc2=0;} return sonuc;

}

public double bol(double x)

{

if(arasonuc2==0) {sonuc/=x;}

else {sonuc/=Math.pow(arasonuc2,x);arasonuc2=0;} return sonuc;

}

public void gir(double x)

{

if(arasonuc2==0) {sonuc=x;}

else {sonuc=Math.pow(arasonuc2,x);arasonuc2=0;}

}

public double oku()

{

arasonuc=sonuc; return arasonuc;

}

public double pow(double x)

{

// sonucun x inci kuvveti

arasonuc=x; arasonuc2=x; return arasonuc;

}

public double kok(double x)

{

arasonuc=Math.sqrt(x); return arasonuc;

}

public double kare(double x)

{

arasonuc=x\*x; return arasonuc;

}

public double bir\_bolu\_x(double x)

{

arasonuc=1.0/x; return arasonuc;

}

public double cos(double x)

{

arasonuc=Math.cos(x); return arasonuc;

}

public double sin(double x)

{

arasonuc=Math.sin(x); return arasonuc;

}

public double tan(double x)

{

arasonuc=Math.cos(x); return arasonuc;

}

public double RtoD(double x)

{

arasonuc=180.0/Math.PI\*x; return arasonuc;

}

public double DtoR(double x)

{

arasonuc=Math.PI/180.0\*x; return arasonuc;

}

public double acos(double x)

{

arasonuc=Math.acos(x); return arasonuc;

}

public double asin(double x)

{

arasonuc=Math.asin(x); return arasonuc;

}

public double atan(double x)

{

arasonuc=Math.atan(x); return arasonuc;

}

public double exp(double x)

{

arasonuc=Math.exp(x); return arasonuc;

}

public double ln(double x)

{

arasonuc=Math.log(x); return arasonuc;

}

public double pi()

{

arasonuc=Math.PI; return arasonuc;

}

public double log10(double x)

{

arasonuc=Math.log(x)/Math.log(10.0); return arasonuc;

}

public double isaretdegistir(double x)

{

arasonuc=-x; return arasonuc;

}

public String aratoString()

{

return ""+arasonuc;

}

}

Object kullanarak dortislemhesapmakinasi sinifindan türetilen bilimselhesapmakinasi2 sinifi verilmistir. Bu sinifi inceleyiniz. Alistirma 10’da verilen bilimselhesapmakinasi1 programindan farklarini açiklayiniz.

### Program 3.26 : bilimselmhesapmakinasi2.java programi

import dortislemhesapmakinasi; public class bilimselhesapmakinasi2

{

//bu metot nesne (object) yoluyla aktarma yapmaktad• r.

// kurucu metot dortislemhesapmakinesi

dortislemhesapmakinasi DH; double arasonuc;

double arasonuc2;

public bilimselhesapmakinasi2(double y)

{

//constructor

DH=new dortislemhesapmakinasi(y); arasonuc=0;

arasonuc2=0;

}

public bilimselhesapmakinasi2()

{

DH=new dortislemhesapmakinasi(); arasonuc=0;

arasonuc2=0;

}

public double topla(double x)

{

if(arasonuc2==0) {DH.setSonuc(DH.getSonuc()+x);}

else {DH.setSonuc(DH.getSonuc()+Math.pow(arasonuc2,x));arasonuc2=0;} return DH.getSonuc();

}

public double cikar(double x)

{

if(arasonuc2==0) {DH.setSonuc(DH.getSonuc()-x);}

else {DH.setSonuc(DH.getSonuc()-Math.pow(arasonuc2,x));arasonuc2=0;} return DH.getSonuc();

}

public double carp(double x)

{

if(arasonuc2==0) {DH.setSonuc(DH.getSonuc()\*x);}

else {DH.setSonuc(DH.getSonuc()\*Math.pow(arasonuc2,x));arasonuc2=0;} return DH.getSonuc();

}

public double bol(double x)

{

if(arasonuc2==0) {DH.setSonuc(DH.getSonuc()/x);}

else {DH.setSonuc(DH.getSonuc()/Math.pow(arasonuc2,x));arasonuc2=0;} return DH.getSonuc();

}

public double gir(double x)

{

if(arasonuc2==0) {DH.setSonuc(x);}

else {DH.setSonuc(Math.pow(arasonuc2,x));arasonuc2=0;} return DH.getSonuc();

}

public double oku()

{

arasonuc=DH.getSonuc(); return arasonuc;

}

public double pow(double x)

{

// sonucun x inci kuvveti arasonuc=x; arasonuc2=x;

return arasonuc;

}

public double kok(double x)

{

arasonuc=Math.sqrt(x); return arasonuc;

}

public double kare(double x)

{

arasonuc=x\*x; return arasonuc;

}

public double bir\_bolu\_x(double x)

{

arasonuc=1.0/x; return arasonuc;

}

public double cos(double x)

{

arasonuc=Math.cos(x); return arasonuc;

}

public double sin(double x)

{birde acele arasonuc=Math.sin(x); return arasonuc;

}

public double tan(double x)

{

arasonuc=Math.cos(x); return arasonuc;

}

public double RtoD(double x)

{

arasonuc=180.0/Math.PI\*x; return arasonuc;

}

public double DtoR(double x)

{

arasonuc=Math.PI/180.0\*x;

return arasonuc;

}

public double acos(double x)

{

arasonuc=Math.acos(x); return arasonuc;

}

public double asin(double x)

{

arasonuc=Math.asin(x); return arasonuc;

}

public double atan(double x)

{

arasonuc=Math.atan(x); return arasonuc;

}

public double exp(double x)

{

arasonuc=Math.exp(x); return arasonuc;

}

public double ln(double x)

{

arasonuc=Math.log(x); return arasonuc;

}

public double pi()

{

arasonuc=Math.PI; return arasonuc;

}

public double log10(double x)

{

arasonuc=Math.log(x)/Math.log(10.0); return arasonuc;

}

public double isaretdegistir(double x)

{

arasonuc=-x; return arasonuc;

}

public String aratoString()

{

return ""+arasonuc;

}

}

**13.** bilimsel hesapmakinasi1 sinifini kullanan hesap makinasi simulasyon programi H4O1.java tanimlanmistir. Bu programi inceleyiniz. Ayni islemi bilimselhesapmakinasi2 sinifini kullanarak yapan H4O1b.java programini yaziniz. Bilimselhesapmakinasi1.java ve bilimselhesapmakinasi2.java programlarinin temel farklarini (miras- inheritance ve nesne-object yoluyla aktarma-composition) izah ediniz

### Program 3.27 : H4O1.java programi

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import bilimselhesapmakinasi1;

public class H4O1 extends Applet implements ActionListener

{

// bilimsel hesap makinasi

TextField sonucgirdisi;

TextField kutugirdisi;

Button Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;

Button Pow,Cos,Sin,Tan,RtoD,DtoR,Acos,Asin,Atan,Exp,Kok,Kare,birBoluX; Button Ln,Log10,Oku,PI;

//Button sinifi degiskenleri double sayi; bilimselhesapmakinasi1 D;

// pencereyi baslatma metodu public void init()

{

D=new bilimselhesapmakinasi1(); sonucgirdisi=new TextField(25); sonucgirdisi.setEditable(false); add(sonucgirdisi); kutugirdisi=new TextField(25);

add(kutugirdisi);//kutuyu pencereye yerlestir Gir=new Button(" Gir ");

Oku=new Button(" Oku "); Arti=new Button(" + "); Eksi=new Button(" - "); Carpi=new Button(" \* "); Bolu=new Button(" / "); M=new Button("M"); MR=new Button("MR"); MArti=new Button("M+"); MEksi=new Button("M-"); Isaret=new Button("+/-"); C=new Button("C"); Pow=new Button("x^y"); Cos=new Button("cos"); Sin=new Button("sin"); Tan=new Button("tan"); RtoD=new Button("R->D"); DtoR=new Button("D->R"); Acos=new Button("acos"); Asin=new Button("asin"); Atan=new Button("atan"); Exp=new Button("exp"); Ln=new Button("ln"); Log10=new Button("log10"); Kok=new Button("x^0.5"); Kare=new Button("x^2"); birBoluX = new Button("1/x"); PI=new Button("pi"); add(Gir);

add(Oku); add(Arti); add(Eksi); add(Carpi);

add(Bolu); add(M);

add(MR); add(MArti); add(MEksi); add(Isaret); add(C);

add(Pow);

add(Cos);

add(Sin);

add(Tan); add(DtoR); add(Acos); add(Asin); add(Atan); add(RtoD); add(birBoluX); add(Exp);

add(Ln); add(Log10); add(Kok); add(Kare); add(PI);

Pow.addActionListener(this); Cos.addActionListener(this); Sin.addActionListener(this); Tan.addActionListener(this); DtoR.addActionListener(this); RtoD.addActionListener(this); Acos.addActionListener(this); Asin.addActionListener(this); Atan.addActionListener(this); Exp.addActionListener(this); Ln.addActionListener(this); Log10.addActionListener(this); Gir.addActionListener(this); Oku.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this); Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this); MArti.addActionListener(this); MEksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this); Kok.addActionListener(this); Kare.addActionListener(this); birBoluX.addActionListener(this); PI.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); if(e.getSource()==Oku) D.oku(); if(e.getSource()==PI) D.pi();

else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi); else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(sayi); else if(e.getSource()==C) D.C();

else if(e.getSource()==Pow) D.pow(sayi); else if(e.getSource()==Cos) D.cos(sayi); else if(e.getSource()==Sin) D.sin(sayi); else if(e.getSource()==Tan) D.tan(sayi); else if(e.getSource()==RtoD) D.RtoD(sayi); else if(e.getSource()==DtoR) D.DtoR(sayi); else if(e.getSource()==Acos) D.acos(sayi); else if(e.getSource()==Asin) D.asin(sayi); else if(e.getSource()==Atan) D.atan(sayi); else if(e.getSource()==Exp) D.exp(sayi); else if(e.getSource()==Ln) D.ln(sayi);

else if(e.getSource()==Log10) D.log10(sayi); else if(e.getSource()==Kok) D.kok(sayi); else if(e.getSource()==Kare) D.kare(sayi);

else if(e.getSource()==birBoluX) D.bir\_bolu\_x(sayi); sonucgirdisi.setText(D.toString()); if(e.getSource()==MR || e.getSource()==M || e.getSource()==MArti||e.getSource()==MEksi)

{

kutugirdisi.setText(D.MtoString());

}

else if(e.getSource()==Cos || e.getSource()==Sin||e.getSource()==Tan||e.getSource()==RtoD || e.getSource()==DtoR || e.getSource()==Acos || e.getSource()==Ln || e.getSource()==Asin||e.getSource()==Atan||e.getSource()==Exp || e.getSource()==Log10||e.getSource()==Isaret ||e.getSource()==Kok || e.getSource()==Kare || e.getSource()==birBoluX || e.getSource()==Oku || e.getSource()==PI)

{

kutugirdisi.setText(D.aratoString());

}

else

{

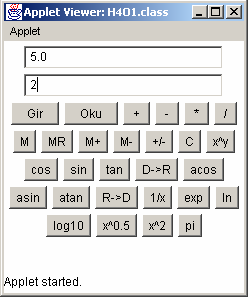
kutugirdisi.setText("");

}

}

}

### 03015.JPG



**Sekil 3.15 H4O1.html bilimsel hesap makinasi simulasyonu**

**Program 3.28 : H4O1b.java programi, Bilimsel hesap makinasi, bu programda hesap makinasi dügmeleri formatlanmistir.**

import java.applet.Applet; // java applet sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import bilimselhesapmakinasi1;

public class H4O1b extends Applet implements ActionListener

{

// bilimsel hesap makinasi

TextField sonucgirdisi;

TextField kutugirdisi;

Button Gir,Arti,Eksi,Carpi,Bolu,M,MR,Marti,MEksi,Isaret,C;

Button Pow,Cos,Sin,Tan,RtoD,DtoR,Acos,Asin,Atan,Exp,Kok,Kare,birBoluX; Button Ln,Log10,Oku,PI;

//Button sinifi degiskenleri double sayi; bilimselhesapmakinasi1 D;

// pencereyi baslatma metodu public void init()

{

D=new bilimselhesapmakinasi1(); sonucgirdisi=new TextField(40); sonucgirdisi.setEditable(false); add(sonucgirdisi); kutugirdisi=new TextField(40);

add(kutugirdisi);//kutuyu pencereye yerlestir Gir=new Button(" Gir ");

Oku=new Button(" Oku "); Arti=new Button(" + "); Eksi=new Button(" - "); Carpi=new Button(" \* "); Bolu=new Button(" / "); M=new Button("M"); MR=new Button("MR"); Marti=new Button("M+"); Meksi=new Button("M-"); Isaret=new Button("+/-"); C=new Button("C"); Pow=new Button("x^y"); Cos=new Button("cos"); Sin=new Button("sin");

Tan=new Button("tan"); RtoD=new Button("R->D"); DtoR=new Button("D->R"); Acos=new Button("acos"); Asin=new Button("asin"); Atan=new Button("atan"); Exp=new Button("exp"); Ln=new Button("ln"); Log10=new Button("log10"); Kok=new Button("x^0.5"); Kare=new Button("x^2"); birBoluX = new Button("1/x"); PI=new Button("pi");

Panel m=new Panel(); m.setLayout(new GridLayout(5,8)); m.add(Gir);

m.add(Oku);

m.add(Arti);

m.add(Eksi);

m.add(Carpi);

m.add(Bolu);

m.add(M);

m.add(MR);

m.add(MArti);

m.add(MEksi); m.add(Isaret); m.add(C);

m.add(Pow);

m.add(Cos);

m.add(Sin);

m.add(Tan);

m.add(DtoR);

m.add(RtoD); m.add(birBoluX); m.add(Acos);

m.add(Asin);

m.add(Atan);

m.add(Kok);

m.add(Kare);

m.add(Exp);

m.add(Ln);

m.add(Log10);

m.add(PI);

add(m); Pow.addActionListener(this); Cos.addActionListener(this); Sin.addActionListener(this); Tan.addActionListener(this); DtoR.addActionListener(this); RtoD.addActionListener(this); Acos.addActionListener(this); Asin.addActionListener(this); Atan.addActionListener(this); Exp.addActionListener(this); Ln.addActionListener(this); Log10.addActionListener(this); Gir.addActionListener(this); Oku.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this);

Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this); Marti.addActionListener(this); Meksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this); Kok.addActionListener(this); Kare.addActionListener(this); birBoluX.addActionListener(this); PI.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); if(e.getSource()==Oku) D.oku(); if(e.getSource()==PI) D.pi();

else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi); else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(sayi); else if(e.getSource()==C) D.C();

else if(e.getSource()==Pow) D.pow(sayi); else if(e.getSource()==Cos) D.cos(sayi); else if(e.getSource()==Sin) D.sin(sayi); else if(e.getSource()==Tan) D.tan(sayi); else if(e.getSource()==RtoD) D.RtoD(sayi); else if(e.getSource()==DtoR) D.DtoR(sayi); else if(e.getSource()==Acos) D.acos(sayi); else if(e.getSource()==Asin) D.asin(sayi); else if(e.getSource()==Atan) D.atan(sayi); else if(e.getSource()==Exp) D.exp(sayi); else if(e.getSource()==Ln) D.ln(sayi);

else if(e.getSource()==Log10) D.log10(sayi); else if(e.getSource()==Kok) D.kok(sayi); else if(e.getSource()==Kare) D.kare(sayi);

else if(e.getSource()==birBoluX) D.bir\_bolu\_x(sayi); sonucgirdisi.setText(D.toString()); if(e.getSource()==MR || e.getSource()==M || e.getSource()==MArti||e.getSource()==MEksi)

{

kutugirdisi.setText(D.MtoString());

}

else if(e.getSource()==Cos || e.getSource()==Sin||e.getSource()==Tan||e.getSource()==RtoD ||

e.getSource()==DtoR || e.getSource()==Acos || e.getSource()==Ln || e.getSource()==Asin||e.getSource()==Atan||e.getSource()==Exp || e.getSource()==Log10||e.getSource()==Isaret ||e.getSource()==Kok || e.getSource()==Kare || e.getSource()==birBoluX || e.getSource()==Oku || e.getSource()==PI)

{

kutugirdisi.setText(D.aratoString());

}

else

{

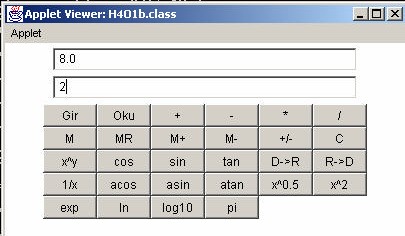
kutugirdisi.setText("");

}

}

}

### 03016.JPG



**Sekil 3.16 H4O1b.html bilimsel hesap makinasi simulasyonu. Bu program üsttekinin aynidir,yalniz dügmeleri formatli bir sekilde yerlestirilmistir. Bu format (layout) islemini daha sonraki bölümlerde detayli olarak inceleyecegiz.**

**Program 3.29 : H4O1bSW.java programi, Bilimsel hesap makinasi, bu programda hesap makinasi dügmeleri formatlanmistir ve temel olarak swing applet sinifi kullanilmistir.**

import javax.swing.\*; // java swing sinifini cagir import java.awt.\*;

import java.awt.event.\*;

import bilimselhesapmakinasi1;

public class H4O1bSW extends JApplet implements ActionListener

{

// bilimsel hesap makinasi

JTextField sonucgirdisi;

JTextField kutugirdisi;

JButton Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;

JButton Pow,Cos,Sin,Tan,RtoD,DtoR,Acos,Asin,Atan,Exp,Kok,Kare,birBoluX; JButton Ln,Log10,Oku,PI;

//Button sinifi degiskenleri double sayi; bilimselhesapmakinasi1 D;

// pencereyi baslatma metodu

public void init()

{

D=new bilimselhesapmakinasi1(); Container c=getContentPane();

c.setLayout(new BorderLayout()); sonucgirdisi=new JTextField(); sonucgirdisi.setEditable(false); sonucgirdisi.setBackground(Color.green);

sonucgirdisi.setFont(new Font("SansSerif",Font.BOLD,14)); JPanel e=new JPanel();

e.setLayout(new GridLayout(2,1)); kutugirdisi=new JTextField(40);

kutugirdisi.setFont(new Font("SansSerif",Font.BOLD,14)); e.add(sonucgirdisi);

e.add(kutugirdisi); c.add(e,BorderLayout.NORTH); Gir=new JButton(" Gir "); Oku=new JButton(" Oku "); Arti=new JButton(" + "); Eksi=new JButton(" - "); Carpi=new JButton(" \* "); Bolu=new JButton(" / "); M=new JButton("M"); MR=new JButton("MR"); MArti=new JButton("M+"); MEksi=new JButton("M-"); Isaret=new JButton("+/-"); C=new JButton("C");

Pow=new JButton("x^y"); Cos=new JButton("cos"); Sin=new JButton("sin"); Tan=new JButton("tan"); RtoD=new JButton("R->D"); DtoR=new JButton("D->R"); Acos=new JButton("acos"); Asin=new JButton("asin"); Atan=new JButton("atan"); Exp=new JButton("exp"); Ln=new JButton("ln"); Log10=new JButton("log10"); Kok=new JButton("x^0.5"); Kare=new JButton("x^2"); birBoluX = new JButton("1/x"); PI=new JButton("pi");

JPanel m=new JPanel(); m.setLayout(new GridLayout(5,8)); m.add(Gir);

m.add(Oku);

m.add(Arti);

m.add(Eksi);

m.add(Carpi);

m.add(Bolu);

m.add(M);

m.add(MR);

m.add(MArti);

m.add(MEksi); m.add(Isaret); m.add(C);

m.add(Pow);

m.add(Cos);

m.add(Sin);

m.add(Tan);

m.add(DtoR);

m.add(RtoD);

m.add(birBoluX); m.add(Acos);

m.add(Asin);

m.add(Atan);

m.add(Kok);

m.add(Kare);

m.add(Exp);

m.add(Ln);

m.add(Log10);

m.add(PI); c.add(m,BorderLayout.SOUTH); Pow.addActionListener(this); Cos.addActionListener(this); Sin.addActionListener(this); Tan.addActionListener(this); DtoR.addActionListener(this); RtoD.addActionListener(this); Acos.addActionListener(this); Asin.addActionListener(this); Atan.addActionListener(this); Exp.addActionListener(this); Ln.addActionListener(this); Log10.addActionListener(this); Gir.addActionListener(this); Oku.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this); Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this); MArti.addActionListener(this); MEksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this); Kok.addActionListener(this); Kare.addActionListener(this); birBoluX.addActionListener(this); PI.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); if(e.getSource()==Oku) D.oku(); if(e.getSource()==PI) D.pi();

else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi); else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(sayi); else if(e.getSource()==C) D.C();

else if(e.getSource()==Pow) D.pow(sayi); else if(e.getSource()==Cos) D.cos(sayi); else if(e.getSource()==Sin) D.sin(sayi); else if(e.getSource()==Tan) D.tan(sayi); else if(e.getSource()==RtoD) D.RtoD(sayi); else if(e.getSource()==DtoR) D.DtoR(sayi); else if(e.getSource()==Acos) D.acos(sayi); else if(e.getSource()==Asin) D.asin(sayi); else if(e.getSource()==Atan) D.atan(sayi); else if(e.getSource()==Exp) D.exp(sayi); else if(e.getSource()==Ln) D.ln(sayi);

else if(e.getSource()==Log10) D.log10(sayi); else if(e.getSource()==Kok) D.kok(sayi); else if(e.getSource()==Kare) D.kare(sayi);

else if(e.getSource()==birBoluX) D.bir\_bolu\_x(sayi); sonucgirdisi.setText(D.toString()); if(e.getSource()==MR || e.getSource()==M || e.getSource()==MArti||e.getSource()==MEksi)

{

kutugirdisi.setText(D.MtoString());

}

else if(e.getSource()==Cos || e.getSource()==Sin||e.getSource()==Tan||e.getSource()==RtoD || e.getSource()==DtoR || e.getSource()==Acos || e.getSource()==Ln || e.getSource()==Asin||e.getSource()==Atan||e.getSource()==Exp || e.getSource()==Log10||e.getSource()==Isaret ||e.getSource()==Kok || e.getSource()==Kare || e.getSource()==birBoluX || e.getSource()==Oku || e.getSource()==PI)

{

kutugirdisi.setText(D.aratoString());

}

else

{

kutugirdisi.setText("");

}

}

public static void main(String s[]) {

JFrame f = new JFrame("Hesap Makinasi Java Programlama dili"); f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

JApplet applet = new H4O1bSW(); f.getContentPane().add("Center", applet); applet.init();

f.pack();

f.setSize(new Dimension(450,220)); f.show();

}

}

### 03017.JPG



**Sekil 3.17 H4O1bSW.html bilimsel hesap makinasi simulasyonu. Bu program üsttekinin aynidir,yalniz Applet yerine Swing Applet Japplet kullanilmistir.**

**14.** H4O2

nokta3D.java ve dogru3D.java programlarini inceleyiniz. Bu programlardan yararlanarak vektor3D.java programini (vektor sinifini) olusturunuz. Testprogrami H4O2.java’da ayni baslangiç noktali iki vektörü toplatiniz.

Not : vektorler dogrular gibi iki nokta ve bir fiziksel boyut (uzay veya baska bir fiziksel boyut olabilir) ile tanimlanirlar. bir yonleri mevcuttur (daima P0 dan P1 e dogru).

### Program 3.30 : H4O2.java programi

import java.io.\*; import vektor3D; import Text;

class **H4O2**

{

public static void main(String arg[]) throws IOException

{

double xi,xj,xk; double Fi,Fj,Fk; nokta3D n1; vektor3D v1; vektor3D v2;

Text cin=new Text();

System.out.print("vektor etki noktasini gir : "); xi=cin.readDouble();

xj=cin.readDouble(); xk=cin.readDouble(); n1=new nokta3D(xi,xj,xk);

System.out.print("vektor de§erini gir : "); Fi=cin.readDouble(); Fj=cin.readDouble(); Fk=cin.readDouble();

v1=new vektor3D(n1,Fi,Fj,Fk); System.out.print("vektor de§erini gir : "); Fi=cin.readDouble(); Fj=cin.readDouble(); Fk=cin.readDouble();

v2=new vektor3D(xi,xj,xk,Fi,Fj,Fk); v1.topla(v2);

System.out.println("toplam = \n"+v1.toString()); System.out.println("toplamin mutlak degeri . "+v1.F());

}

}

vektor etki noktasini gir : 1 1 1

vektor degerini gir : 1 1 1

vektor degerini gir : 2 2 2 toplam =

baslangic noktasi : 1.0 i + 1.0 j + 1.0 k Vektor : 3.0 i + 3.0 j + 3.0 k

toplamin mutlak degeri : 5.196152422706632

### Program 3.31 : nokta3D.java programi

public class nokta3D

{

protected double xi,xj,xk;

public nokta3D()

{

xi=0; xj=0; xk=0;

}

public nokta3D(double yi,double yj,double yk)

{

xi=yi; xj=yj; xk=yk;

}

public nokta3D(nokta3D y)

{

xi=y.xi; xj=y.xj; xk=y.xk;

}

public void xiGir(double yi)

{

xi=yi;

}

public void xjGir(double yj)

{

xj=yj;

}

public void xzGir(double yk)

{

xk=yk;

}

public void noktaGir(double yi,double yj,double yk)

{

xi=yi; xj=yj; xk=yk;

}

public void noktaGir(nokta3D y)

{

xi=y.xi; xj=y.xj; xk=y.xk;

}

public double xiOku()

{

return xi;

}

public double xjOku()

{

return xj;

}

public double xkOku()

{

return xk;

}

public void topla(nokta3D y)

{

xi+=y.xi; xj+=y.xj; xk+=y.xk;

}

public static nokta3D topla(nokta3D y1,nokta3D y2)

{

nokta3D x=new nokta3D(y1); x.topla(y2);

return x;

}

public void fark(nokta3D y)

{

xi-=y.xi;

xj-=y.xj;

xk-=y.xk;

}

public static nokta3D fark(nokta3D y1,nokta3D y2)

{

nokta3D x=new nokta3D(y1); x.fark(y2);

return x;

}

public nokta3D noktaOku()

{

return this;

}

public double R()

{

//koordinat merkezi ile nokta aras• ndali mesafe return Math.sqrt(xi\*xi+xj\*xj+xk\*xk);

}

public double cosai()

{

//noktan• n i ekseniyle yapt• §• a‡• n• n cosin• s• return xi/R();

}

public double cosaj()

{

//noktan• n j ekseniyle yapt• §• a‡• n• n cosin• s• return xj/R();

}

public double cosak()

{

//noktan• n k ekseniyle yapt• §• a‡• n• n cosin• s• return xk/R();

}

public boolean esittir(nokta3D v)

{

boolean b=((xi==v.xi)&&(xj==v.xj)&&(xk==v.xk)); return b;

}

public boolean buyuktur(nokta3D v)

{

return (this.R()>v.R());

}

public boolean kucuktur(nokta3D v)

{

return (this.R()<v.R());

}

public String toString()

{

String s=""; if(xi!=0) s+=xi+" i "; if(xj>0 )

s+="+ "+xj+" j";

else if(xj<0)

s+="- "+Math.abs(xj)+" j"; if(xk>0 )

s+="+ "+xk+" k";

else if(xk<0)

s+="- "+Math.abs(xk)+" k"; return s;

}

}

### Program 3.32 : dogru3D.java programi

import nokta3D; public class dogru3D

{

public nokta3D P0,P1; double xi,xj,xk;

//\*\*\*\*\* kurucu metotlar dogru3D

public dogru3D()

{

P0=new nokta3D(); P1=new nokta3D(); xi=0;

xj=0; xk=0;

}

public dogru3D(double yi0,double yj0,double yk0,double yi1,double yj1,double yk1)

{

P0=new nokta3D(yi0,yj0,yk0); P1=new nokta3D(yi1,yj1,yk1); nokta3D P=new nokta3D(P0); P.fark(P1);

xi=P.xiOku();

xj=P.xjOku();

xk=P.xkOku();

}

public dogru3D(nokta3D y0,nokta3D y1)

{

P0=new nokta3D(y0); P1=new nokta3D(y1); nokta3D P=new nokta3D(P0); P.fark(P1);

xi=P.xiOku();

xj=P.xjOku();

xk=P.xkOku();

}

//\*\*\*\*\* s• n• f de§iŸken giriŸi ..Gir (veya set..) metotlar•

public void P0Gir(double yi,double yj,double yk)

{

P0.noktaGir(yi,yj,yk);

}

public void P0Gir(nokta3D y)

{

P0.noktaGir(y);

}

public void P1Gir(double yi,double yj,double yk)

{

P1.noktaGir(yi,yj,yk);

}

public void P1Gir(nokta3D y)

{

P1.noktaGir(y);

}

//\*\*\*\*\* s• n• f de§iŸken ‡• k• Ÿ• ..Oku veya get.. metotlar• public double xiOku(int i)

{

if(i==0) return P0.xiOku();

else return P1.xiOku();

}

public double xjOku(int i)

{

if(i==0) return P0.xjOku(); else return P1.xjOku();

}

public double xkOku(int i)

{

if(i==0) return P0.xkOku(); else return P1.xkOku();

}

public double xiOku()

{

return xi;

}

public double xjOku()

{

return xj;

}

public double xkOku()

{

return xk;

}

//\*\*\*\*sinif islem metotlari

public double R()

{

//iki nokta arasindaki mesafe

return Math.sqrt(xi\*xi+xj\*xj+xk\*xk);;

}

public double cosai()

{

//noktan• n i ekseniyle yapt• §• a‡• n• n cosin• s• return xi/R();

}

public double cosaj()

{

//noktanin j ekseniyle yaptigi açinin kosinüsü return xj/R();

}

public double cosak()

{

//noktan• n k ekseniyle yaptigi açinin kosinüsü return xk/R();

}

//\*\*\*\*\*sinif mantiksal karsilastirma metodlari public boolean esittir(dogru3D v)

{

boolean b=((xi==v.xi)&&(xj==v.xj)&&(xk==v.xk));

return b;

}

public boolean buyuktur(dogru3D v)

{

return (this.R()>v.R());

}

public boolean kucuktur(dogru3D v)

{

return (this.R()<v.R());

}

//\*\*\*\*\*\* sinif string çikti metotu toString public String toString()

{

String s="P0 = "+P0.toString()+"\n"; s=s+"P1 = "+P1.toString()+"\n"; s=s+""+xi+" i + "+xj+" j + "+xk+" k"; return s;

}

}

### Program 3.33 : vektor3D.java programi

import dogru3D; import nokta3D;

class vektor3D extends nokta3D

{

protected double Fi,Fj,Fk; public vektor3D()

{

super();

Fi=0;

Fj=0;

Fk=0;

}

public vektor3D(double yi0,double yj0,double yk0, double Fi0,double Fj0,double Fk0)

{

// (yi0,yj0,yk0) noktasindan baslayan vektor super(yi0,yj0,yk0);

Fi=Fi0; Fj=Fj0; Fk=Fk0;

}

public vektor3D(double yi0,double yj0,double yk0,double yi1,double yj1,double yk1,double F)

{

//iki dogrudan gecen vektorun tanimi

dogru3D D=new dogru3D(yi0,yj0,yk0,yi1,yj1,yk1); Fi=F\*D.cosai();

Fj=F\*D.cosaj();

Fk=F\*D.cosak();

}

public vektor3D(double Fi0,double Fj0,double Fk0)

{

//0,0,0 noktas• ndan baslayan vektor tan• m• super();

Fi=Fi0; Fj=Fj0; Fk=Fk0;

}

public vektor3D(vektor3D y)

{

xi=y.xi; xj=y.xj; xk=y.xk; Fi=y.Fi; Fj=y.Fj; Fk=y.Fk;

}

public vektor3D(nokta3D y0,double Fi0,double Fj0,double Fk0)

{

// y0 noktas• ndan baslayan vektor super(y0);

Fi=Fi0; Fj=Fj0; Fk=Fk0;

}

public vektor3D(nokta3D y0,nokta3D y1,double F)

{

//iki noktadan gecen vektorun tan• m• dogru3D D=new dogru3D(y0,y1); Fi=F\*D.cosai();

Fj=F\*D.cosaj();

Fk=F\*D.cosak();

}

public void FiGir(double yi)

{

xi=yi;

}

public void FjGir(double yj)

{

xj=yj;

}

public void FzGir(double yk)

{

xk=yk;

}

public void FGir(double Fi0,double Fj0,double Fk0)

{ Fi=Fi0; Fj=Fj0; Fk=Fk0;

}

public void vektorGir(vektor3D y)

{

xi=y.xi; xj=y.xj; xk=y.xk; Fi=y.Fi; Fj=y.Fj; Fk=y.Fk;

}

public double FiOku()

{

return Fi;

}

public double FjOku(){ return Fj;

}

public double FkOku()

{

return Fk;

}

public nokta3D vektorOku()

{

return this;

}

public double F()

{

//vektorun absolute büyüklügü

return Math.sqrt(Fi\*Fi+Fj\*Fj+Fk\*Fk);

}

//not R() metodu, vektorun merkezden uzakligi nokta3D de tanimlanmistir.

// tekrar tanimi gerekmez.

/\*

public double R()

{

//koordinat merkezi ile nokta aras• ndali mesafe return Math.sqrt(xi\*xi+xj\*xj+xk\*xk);

}

\*/

public double cosFi()

{

//vektorun i ekseniyle yaptigi açinin kosinüsü return Fi/F();

}

public double cosFj()

{

//vektorun j ekseniyle yaptigi açinin kosinüsü return Fj/F();

}

public double cosFk()

{

//vektorun k ekseniyle yaptigi açinin kosinüsü return Fk/F();

}

public void topla(vektor3D v)

{

if(noktaEsittir(v))

{

Fi+=v.Fi; Fj+=v.Fj; Fk+=v.Fk;

}

}

public static vektor3D topla(vektor3D v1,vektor3D v2)

{

vektor3D x;

x=new vektor3D(v1); x.topla(v2);

return x;

}

public void cikar(vektor3D v)

{

if(noktaEsittir(v))

{

Fi-=v.Fi;

Fj-=v.Fj;

Fk-=v.Fk;

}

}

public static vektor3D cikar(vektor3D v1,vektor3D v2)

{

vektor3D x=new vektor3D(v1); x.cikar(v2);

return x;

}

public void vektorelcarp(vektor3D v)

{

if(noktaEsittir(v))

{

double Fi1=Fi; double Fj1=Fj; double Fk1=Fk;

Fi=(Fj1\*v.Fk-Fk1\*v.Fj);

Fj=-(Fi1\*v.Fk-Fk1\*v.Fi);

Fk=(Fi1\*v.Fj-Fj1\*v.Fi);

}

}

public double scalarcarp(vektor3D v)

{

double s=0.0; if(noktaEsittir(v))

{

s=Fi\*v.Fi+Fj\*v.Fj+Fk\*v.Fk;

}

return s;

}

public boolean esittir(vektor3D v)

{

boolean b=((xi==v.xi)&&(xj==v.xj)&&(xk==v.xk)); b=b && ((Fi==v.Fi)&&(Fj==v.Fj)&&(Fk==v.Fk)); return b;

}

public boolean noktaEsittir(vektor3D v)

{

boolean b=((xi==v.xi)&&(xj==v.xj)&&(xk==v.xk)); return b;

}

public boolean buyuktur(vektor3D v)

{

return (this.F()>v.F());

}

public boolean kucuktur(vektor3D v)

{

return (this.F()<v.F());

}

public String toString()

{

String s="baslangic noktas• : "+xi+" i + "+xj+" j + "+xk+" k\n"; s=s+"Vektor : "+Fi+" i + "+Fj+" j + "+Fk+" k\n";

return s;

}

}

### 14. H4OD3\_2000

**nokta3D.java** ve **dogru3D.java** programlarini inceleyiniz. Bu programlardan yararlanarak **vektor3D.java** programini (vektor sinifini) olusturunuz. Java swing JFrame Test programi H4OD3\_2000.java’da ayni baslangiç noktali iki vektörü toplatiniz.

Not : vektorler dogrular gibi iki nokta ve bir fiziksel boyut (uzay veya baska bir fiziksel boyut olabilir) ile tanimlanirlar. bir yonleri mevcuttur (daima P0 dan P1 e dogru).

### Program 3.34 : vektor3D sinifini test eden swing JFrame, H4OD3\_2000.java programi

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import vektor3D;

import vektorpanel;

import BasicWindowMonitor;

public class H4OD3\_2000 extends JFrame implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

JTextArea cikti; // JTextArea sinifi degiskeni (nesnesi) cikti vektor3D k[]=new vektor3D[2]; // vector3D boyutlu degisken vektorpanel p[]=new vektorpanel[2]; //vektorPanel degisken p1

//=====================================================

// pencereyi baslatma metodu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public H4OD3\_2000()

{

super("vektor testi H4OD3\_2000, frame format"); k[0]=new vektor3D(1.0,1.0,1.0,2.0,2.0,2.0);

k[1]=new vektor3D(1.0,1.0,1.0,3.0,3.0,3.0);

Container c=getContentPane(); c.setLayout(new FlowLayout()); JPanel p1=new JPanel(); p1.setLayout(new GridLayout(2,1));

p[0]=new vektorpanel("V1 xi,xj,xk:","V1 Fi,Fj,Fk:");

p[1]=new vektorpanel("V2 xi,xj,xk:","V2 Fi,Fj,Fk:"); p1.add(p[0],BorderLayout.NORTH); p1.add(p[1],BorderLayout.SOUTH);

c.add(p1);

cikti=new JTextArea(toString()); cikti.setBackground(c.getBackground()); c.add(cikti); p[1].F[2].addActionListener(this);

}

//=====================================================

public String toString()

{

String s="";

for(int i=0;i<2;i++)

{

s+="Vektör "+(i+1)+"\n";

s+=k[i].toString();

}

s+="Toplam vektör : \n"; k[0].topla(k[1]);

s+=k[0].toString(); return s;

}

// girdi alanindaki olan olaylari dinleme metodu

// Her yeni rakam girildiginde bu metod cagirilir

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<2;i++) k[i].vektorGir(Double.parseDouble(p[i].x[0].getText()),

Double.parseDouble(p[i].x[1].getText()), Double.parseDouble(p[i].x[2].getText()), Double.parseDouble(p[i].F[0].getText()), Double.parseDouble(p[i].F[1].getText()), Double.parseDouble(p[i].F[2].getText()));

cikti.setText(toString());

}

//=====================================================

public static void main(String[] args)

{

H4OD3\_2000 pencere= new H4OD3\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(300,200);

pencere.setVisible(true);

}

}

### 03018.JPG

**Sekil 3.18 vektor3D sinifini test eden H4OD3\_2000 java frame programi yukaridaki programda kullanilan alt sinif vektorpanel.java**

**Program 3.35 : vektor3D sinifini test eden swing JFrame, H4OD3\_2000.java programinda kullanilan vektorpanel alt sinifi (vektorpanel.java programi)**

import java.awt.\*; import javax.swing.\*;

public class vektorpanel extends JPanel

{

public JLabel xisim;

public JTextField x[]=new JTextField[3]; public JLabel Fisim;

public JTextField F[]=new JTextField[3];

public vektorpanel()

{

this.setLayout(new GridLayout(2,4)); xisim=new JLabel("xi,xj,xk"); x[0]=new JTextField(5);

x[1]=new JTextField(5); x[2]=new JTextField(5); Fisim=new JLabel("Fi,Fj,Fk"); F[0]=new JTextField(5); F[1]=new JTextField(5); F[2]=new JTextField(5); this.add(xisim);

for(int i=0;i<3;i++) this.add(x[i]);

this.add(Fisim); for(int i=0;i<3;i++)

this.add(F[i]);

}

public vektorpanel(String isim1,String isim2)

{

this.setLayout(new GridLayout(2,4)); xisim=new JLabel(isim1);

x[0]=new JTextField(5); x[1]=new JTextField(5); x[2]=new JTextField(5); Fisim=new JLabel(isim2); F[0]=new JTextField(5);

F[1]=new JTextField(5); F[2]=new JTextField(5); this.add(xisim);

for(int i=0;i<3;i++) this.add(x[i]);

this.add(Fisim); for(int i=0;i<3;i++)

this.add(F[i]);

}

public void setvectorPanel(double xi[],double Fi[])

{

for(int i=0;i<3;i++) this.x[i].setText(""+xi[i]);

for(int i=0;i<3;i++) this.F[i].setText(""+Fi[i]);

}

}

### 15. H4O3

Üç noktadan bir düzlem geçtigini biliyoruz. düzlemin formülü z=a\*x+b\*y+c dir (a,b,c sabit). 3 noktadan geçen düzlemi tanimlayan duzlem.java programini ve duzlem sinifini yaziniz.

not: birim vektorler xi, x dogrultusunda, xj y dogrultusunda ve xk z dogrultusundadir. H2O3.java konsol türü örnek test programinda yarattiginiz bir düzlem nesnesinin(object) baslangiç kordinatlarina uzakligini (baslangiç noktasindan geçen ve düzleme dik olan bir dogrunun boyunu) hesaplayiniz.

### Program 3.36 : vektor3D.java, nokta3D.java test programi H4O3.java

import duzlem; import vektor3D; import nokta3D;

class H4O3

{

public static void main(String args[])

{

//bu test d• zlem ile nokta aras• ndaki mesafeyi hesaplar. double a,b,c;

a = (-7.0/3.0);

b = (5.0/3.0);

c = (2.0/3.0);

duzlem d=new duzlem(a,b,c);

nokta3D P1=new nokta3D(1,1,d.z(1.0,1.0)); nokta3D P0=new nokta3D(2,-1,3); vektor3D v=new vektor3D(P1,b,c,1.0); nokta3D P=new nokta3D(P0);

P.fark(P1);

vektor3D v1=new vektor3D(P1,P.xiOku(),P.xjOku(),P.xkOku()); double sonuc=v.scalarcarp(v1)/v.F(); System.out.println("duzlem d = "+d.toString()); System.out.println("nokta P0 = "+P0.toString());

System.out.println("duzlem(d) ile nokta arasindaki mesafe:"+sonuc);

}

}

duzlem d = z = -2.3333333333333335 + 1.6666666666666667x + 0.6666666666666666y

nokta P0 = 2.0 i - 1.0 j+ 3.0 k

duzlem(d) ile nokta arasindaki mesafe:1.6222142113076254

###  H4OD4\_2000

kompleks sinifina kompleks sayinin karekokünü hesaplayacak

### public static kompleks karekok(kompleks sol)

metodunu ekleyiniz. Bir test programi yazarak (**H4OD4\_2000.java)** girdiginiz kompleks sayinin karesini hesaplatiniz.

not : **public static kompleks pow(kompleks sol)** metodu kompleks sayinin üssünü hesaplar.

public static kompleks karekok(kompleks sol)

{

return pow(sol,0.5);

}

### Program 3.37 : vektor3D.java, nokta3D.java test programi H4O3.java

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import kompleks;

import BasicWindowMonitor;

public class H4OD4\_2000 extends JFrame implements ActionListener

{

//

//=====================================================

// sinif degiskenleri

// bu degiskenler tum sinifa aittir

JLabel kutubasligi1; //JLabel sinifi degiskeni (nesnesi) kutubasligi1 JTextField kutugirdisi1; // Textfield sinifi degiskeni (nesnesi) kutugirdisi1 JTextField kutugirdisi2; // JTextfield sinifi degiskeni (nesnesi) kutugirdisi2 JTextArea cikti; // JTextArea sinifi degiskeni (nesnesi) cikti kompleks k1; // kompleks degisken k1

//=====================================================

// pencereyi baslatma metodu

// pencere ilk basladiginda

// degiskenler buradaki degerleri alirlar public H4OD4\_2000()

{

super("kompleks sayi karekök testi H4OD4\_2000, frame format"); k1=new kompleks(1.0,1.0);

Container c=getContentPane(); c.setLayout(new FlowLayout());

kutubasligi1=new JLabel("kompleks sayiyi giriniz : "); c.add(kutubasligi1);

kutugirdisi1=new JTextField(5); kutugirdisi1.addActionListener(this); c.add(kutugirdisi1); kutugirdisi2=new JTextField(5); kutugirdisi2.addActionListener(this); c.add(kutugirdisi2);

cikti=new JTextArea(toString()); cikti.setBackground(c.getBackground()); c.add(cikti);

}

//=====================================================

public String toString()

{

String s="";

s+="kompleks sayi : "+k1.toString()+"\n";

s+="kompleks sayinin karekökü : "+kompleks.toString(kompleks.karekok(k1))+"\n"; s+="kompleks sayinin karesi : "+kompleks.toString(kompleks.kare(k1))+"\n";

return s;

}

// girdi alanindaki olan olaylari dinleme metodu

// Her yeni rakam girildiginde bu metod cagirilir

public void actionPerformed(ActionEvent e)

{

k1.kompleksgir(Double.parseDouble(kutugirdisi1.getText()), Double.parseDouble(kutugirdisi2.getText()));

cikti.setText(toString());

}

//=====================================================

public static void main(String[] args)

{

H4OD4\_2000 pencere= new H4OD4\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(500,200);

pencere.setVisible(true);

}

}

### BÖLÜM 4 : BOYUTLU DEGISKENLER VE NESNELER (ARRAYS)

### 4.1 TEK BOYUTLU DEGiSKENLER

Boyutlu degiskenler ayni degisken türü veya sinif türünden birden fazla nesne veya degiskenin tek bir isimle tanimlanmasidir. Boyutlu degiskenlere referans indeks numarasi kullanilarak ulasilir. Genel olarak bir boyutlu degiskenler

### Degisken\_türü degisken\_ismi[];

veya

### Degisken\_türü[] degisken\_ismi;

Seklinde tanimlanir. Birinci tanimin kullanilmasi daha yaygindir. Iki tanim arasinda bilgisayar açisindan bir fark mevcut degildir. Bir örnek verecek olursak

### int ayin\_gunleri[]; veya int[] ayin\_gunleri;

Bu tanimlar her ne kadar haftanin\_gunleri nin boyutlu bir degisken oldugunu belirtiyorsa da kesin boyutunu bildirmemektedir. Boyutlu degiskenin tam boyutunu tanimlamak ve bilgisayar hafizasindaki yerini tam olarak saptamak için

Degisken\_ismi=new Degisken\_türü[degisken\_boyutu]; Terimi kullanilir. Bunu bir önceki örnege uygularsak : **Ayin\_gunleri=new int[12];**

Boyutlu degiskeni bir kere tanimladiktan sonra onun içindeki her alt degere indeksi üzerinden ulasmak mümkündür.

Örnegin: ayin\_gunleri[0] = 31;

ayin\_gunleri[1] = 28;

gibi. . Indeks degiskeni herzaman sifirdan baslar.

Küçük bir örnek sinifta bu kavrami daha açik olarak vermeye çalisalim.

### Program 4.1 Boyut.java

import java.io.\*; class boyut

{

public static void main(String args[] )

{

int ayin\_gunleri[]; ayin\_gunleri=new int[12]; ayin\_gunleri[0]=31; ayin\_gunleri[1]=28; ayin\_gunleri[2]=31; ayin\_gunleri[3]=30; ayin\_gunleri[4]=31; ayin\_gunleri[5]=30; ayin\_gunleri[6]=31; ayin\_gunleri[7]=31; ayin\_gunleri[8]=30; ayin\_gunleri[9]=31;

ayin\_gunleri[10]=30; ayin\_gunleri[11]=31;

System.out.println("Nisan ayi "+ayin\_gunleri[3]+" gun ceker");

}

}

Bu programi daha kisa olarak yazmak için Sekil 4.1.2 de görülen sekilde bütün degiskenlerin degerlerini ayni anda tanimliyabiliriz.

### Program 4.2 : Boyut1.java

import java.io.\*; class boyut1

{

public static void main(String args[] )

{

int ayin\_gunleri[]={31,28,31,30,31,30,31,31,30,31,30,31};

System.out.println("Nisan ayi "+ayin\_gunleri[3]+" gun ceker");

}

}

Program 4.3’de boyut kavraminin for döngüsüyle birlikte kullanisini görüyorsunuz. Boyutlu degiskenin toplam boyutuna da **sayi.length** degiskeniyle ulasiyoruz. Daha önceki birçokprogramlama dilinde bu mümkün degildir ve boyutlu degiskenin boyutunun bildirilmesi gerekir. Program 4.3 ün çiktisini step step takip ederek ve bir hesap makinasi kullanarak hesaplayiniz.

### Program 4.3 : Aritmetik2.java

import java.io.\*; //java girdi cikti sinifini cagir class Aritmetik2

{

public static void main(String args[])

{

int sayi[]={10,5,7,9,11,13,14,18};

int toplam=0; int i;

for(i=0;i<**sayi.length**;i++)

{

toplam+=sayi[i];

}

System.out.println("Ortalama = "+toplam/sayi.length);

}

}

Simdi de bir applet örneginde tek boyutlu degiskenleri kullanalim. Daha önce zarApplet.java programini incelemistik. Simdi bu programa her bir zar yüzeyinin atilis frekanslarini ekleyelim. Toplam alti yüz oldugundan birden altiya kadar her sayinin kaç kere geldigini saydirmamiz gerekir. Bunun için bir boyutlu **zarfrekansi** degiskenini tanimlayacagiz.

Program listesi ve sonuç appleti Program 4.4 ve sekil 4.1 de görülmektedir.

### Program 4.4 : zarfrekansiApplet

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class zarfrekansiApplet extends Applet implements ActionListener

{

int toplamzaratisi=0; int toplam=0;

TextField ilkzar,ikincizar; Button salla; //zar atma dügmesi

int zar1,zar2;

int zarfrekansi[];

public static int zar()

{

return 1+(int)(Math.random()\*6);

}

public void init()

{

//programi baslat zarfrekansi=new int[6];

ilkzar=new TextField(10); // Textfield sinifi ilkzar nesnesini yarat add(ilkzar); // ilk zar nesnesini pencereye ekle

ikincizar=new TextField(10);// Textfield sinifi ikincizar nesnesini yarat add(ikincizar); // ikinci zar nesnesini pencereye ekle salla=new Button("Zari salla ve at");

add(salla); salla.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

zar1=zar(); zar2=zar();

ilkzar.setText(Integer.toString(zar1)); ikincizar.setText(Integer.toString(zar2)); toplam+=(zar1+zar2);

toplamzaratisi++;

// not boyutlu degisken indeksi 0 dan basliyor. zarfrekansi[zar1-1]++;

zarfrekansi[zar2-1]++; repaint();

}

public void paint( Graphics g)

{

g.drawString("toplam = "+toplam+" Atilan zar sayisi = "+toplamzaratisi,25,50); for(int i=0;i<6;i++)

{

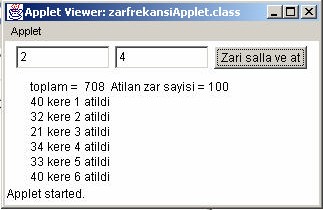
g.drawString(zarfrekansi[i]+" kere "+(i+1)+" atildi ",25,(65+15\*i));

}

}

}

### 04001.JPG



**Sekil 4.1 ZarfrekansiApplet.java appletinin görünümü**

zarfrekansiApplet.java programinin bir de swing esdegeri zarfrekansiSWF.java swing Frame programini verelim

:

### Program 4.5 : zarfrekansiSWF, java swing JFrame programi

import javax.swing.\*; import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*; import BasicWindowMonitor;

public class zarfrekansiSWF extends JFrame implements ActionListener

{

int toplamzaratisi=0; int toplam=0;

JTextField ilkzar,ikincizar; JButton salla; //zar atma dügmesi JTextArea cikti;

int zar1,zar2;

int zarfrekansi[];

public static int zar()

{

return 1+(int)(Math.random()\*6);

}

public zarfrekansiSWF()

{

//programi baslat

super("Zar frekansi swing JFrame"); zarfrekansi=new int[6];

Container c=getContentPane(); c.setLayout(new FlowLayout());

ilkzar=new JTextField(10); // JTextfield sinifi ilkzar nesnesini yarat c.add(ilkzar); // ilk zar nesnesini pencereye ekle

ikincizar=new JTextField(10);// Textfield sinifi ikincizar nesnesini yarat c.add(ikincizar); // ikinci zar nesnesini pencereye ekle salla=new JButton("Zari salla ve at");

c.add(salla); salla.addActionListener(this); cikti=new JTextArea();

cikti.setBackground(c.getBackground()); c.add(cikti);

}

public void actionPerformed(ActionEvent e)

{

String s=""; zar1=zar(); zar2=zar();

ilkzar.setText(Integer.toString(zar1)); ikincizar.setText(Integer.toString(zar2)); toplam+=(zar1+zar2);

toplamzaratisi++;

// not boyutlu degisken indeksi 0 dan basliyor. zarfrekansi[zar1-1]++;

zarfrekansi[zar2-1]++;

s+="toplam = "+toplam+" Atilan zar sayisi = "+toplamzaratisi+"\n"; for(int i=0;i<6;i++)

{

s+=zarfrekansi[i]+" kere "+(i+1)+" atildi \n";

}

cikti.setText(s); repaint();

}

//=====================================================

public static void main(String[] args)

{

zarfrekansiSWF pencere= new zarfrekansiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(450,200);

pencere.setVisible(true);

}

}

### 04002.JPG



**Sekil 4.2 ZarfrekansiAppletSWF.java appletinin görünümü**

### TEK BOYUTLU NESNE TIPI DEGISKENLER

Nesne tipi degiskenler için de ayni normal basit degiskenlerde oldugu gibi boyut tanimi yapilir. Normal boyut tanimi yapildiktan sonra, boyuttaki her nesne teker teker tanimlanir. Örnegin :

kompleks sayi[]=new kompleks[5]; sayi[0]=new kompleks(1,2); sayi[1]=new kompleks(1,-1); sayi[2]=new kompleks(2,0); sayi[3]=new kompleks(1.1,-0.5); sayi[4]=new kompleks(1,1.5);

Dogru3 n[]; n=new dogru3[3];

n[0]=new dogru3(1,1,1,2,2,2);

n[1]=new dogru3(2,2,2,3,3,3);

n[2]=new dogru3(3,3,3,4,4,4); gibi.

### ÇOK BOYUTLU DEGISKENLER

Iki ve daha fazla boyutlu degiskenler de tanimlamak mümkündür. Iki boyutlu degiskenler en fazla tablo veya matris gibi satir ve sütun olmak üzere iki boyutta gösterilmesi gereken büyüklükleri olusturmak için kullanilir.Java iki boyutlu degiskenleri direk olarak açamaz. Tek boyutlu degiskenlerin yine tek boyutlu degiskenini açar. Sonuç olarak iki boyut saglanmis olur. Örnek verecek olursak

Int b[][]=new int[3][4]. Genel olarak ilk parantes satir sayisi, ikinci parantez sütun sayisi olarak kabul edilir. Buna göre b degiskenini söyle düsünebiliriz :

|  |  |  |
| --- | --- | --- |
| b[0][0] | b[0][1] | b[0][2] b[0][3] |
| b[1][0] | b[1][1] | b[1][2] b[1][3] |
| b[2][0] | b[2][1] | b[2][2] b[2][3] |

Iki boyutlu degisken de bir boyutlu degiskenlerde oldugu gibi degerleri direk olarak yüklüyebiliriz, ve yükleme sirasinda boyutlari da tayin edebiliriz.

Örnegin :

Int b[][] = {{1,2},{3,4}};

Terimi bize asagidaki tabloyu tanimlar: 1 2

3 4

b[0][0] b[0][1]

b[1][0] b[1][1]

Iki boyutlu degiskenlerin kullanilmasini göstermek amaciyla bir örnek problem verelim:

### Program 4.6 : Iki boyutlu degiskenler örnegi : ikiboyut.java

import java.awt.Graphics; import java.applet.Applet;

public class ikiboyut extends Applet

{

// void tipi paint metotu public void paint( Graphics g)

{

int a1[][]={{1,2,3},{4,5,6}};

int a2[][]={{1,2},{4}};

g.drawString(" a1 : ",25,25); ikiboyutyazdir(a1,g,40); g.drawString(" a2 : ",25,70); ikiboyutyazdir(a2,g,85);

}

public void ikiboyutyazdir(int a[][], Graphics g,int y)

{

int x=25;

for(int i=0;i<**a.length**;i++)

{

for(int j=0;j<**a[i].length**;j++)

{

g.drawString(String.valueOf(a[i][j]),x,y); x+=15;

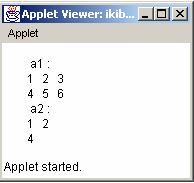
} x=25; y+=15;

}

}

}

### 04003.JPG



**Sekil 4.3 Iki boyutlu degiskenler örnegi ikiboyut.html doyasinin appletviewer ile görünümü**

Ayni programi swing JoptionPane çiktisi olarak da tanimlayalim :

### Program 4.7 : Iki boyutlu degiskenler örnegi : ikiboyut\_2000.java

import javax.swing.JOptionPane;

public class ikiboyut\_2000

{

public static void main( String args[])

{

int a1[][]={{1,2,3},{4,5,6}};

int a2[][]={{1,2},{4}};

String s=" a1 : \n"; s=s+ikiboyutyazdir(a1); s=s+" a2 : \n"; s=s+ikiboyutyazdir(a2);

JOptionPane.showMessageDialog(null,s); System.exit(0);

}

public static String ikiboyutyazdir(int a[][])

{

String s="";

for(int i=0;i<a.length;i++)

{

for(int j=0;j<a[i].length;j++)

{

s+=String.valueOf(a[i][j])+" ";

}

s+="\n";

}

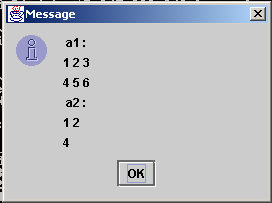
return s;

}

}

JoptionPane çiktisi :

### 04004.JPG



**Sekil 4.4 Iki boyutlu degiskenler örnegi ikiboyut\_2000.html doyasinin JoptionPane çiktisi olarak görünümü**

Ayni programin swing JFrame versiyonunu da inceleyelim :

### Program 4.8 : Iki boyutlu degiskenler örnegi : ikiboyutSWF\_2000.java

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

import BasicWindowMonitor;

public class ikiboyutSWF\_2000 extends JFrame

{

JTextArea cikti;

public ikiboyutSWF\_2000()

{

super("iki boyutlu degisken örnegi"); Container c=getContentPane(); c.setLayout(new FlowLayout());

int a1[][]={{1,2,3},{4,5,6}};

int a2[][]={{1,2},{4}};

String s=" a1 : \n"; s=s+ikiboyutyazdir(a1); s=s+" a2 : \n"; s=s+ikiboyutyazdir(a2); cikti=new JTextArea(s);

cikti.setBackground(c.getBackground()); c.add(cikti);

}

public static String ikiboyutyazdir(int a[][])

{

String s="";

for(int i=0;i<a.length;i++)

{

for(int j=0;j<a[i].length;j++)

{s+=String.valueOf(a[i][j])+" ";} s+="\n";

}

return s;

}

//=====================================================

public static void main(String[] args)

{

ikiboyutSWF\_2000 pencere= new ikiboyutSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor());

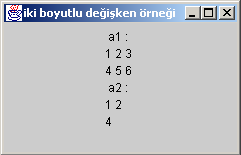
pencere.setSize(200,200); pencere.setVisible(true);

}

}

Swing frame çiktisi :

### 04005.JPG



**Sekil 4.5 Iki boyutlu degiskenler örnegi ikiboyutSWF\_2000.html doyasinin JFrame çiktisi olarak görünümü**

Boyutlara ulasmak için yukaridaki programlardan da görülecegi gibi for döngüsü kullandik. Boyutlu degiskenlere ulasim. for döngülerinin en çok kullanildigi yerlerden biridir.

Satirlarin boyutuna **a.length**, sütunlarin boyutuna ise **a[satir].length** terimiyle ulastik. Bu tanim bize toplam satir ve sütun boyutunu direk tanimlar, ve kullanimda yeterince esneklik saglar.

### BOYUTLU DEGISKENLERIN METOTLARA AKTARIMI

Boyutlu degiskenler aynen boyutsuz degiskenler gibi metotlara aktarilirlar. Örnegin eger

**int sicaklik[] = new int[24];** deyimiyle tanimlanan sicaklik; **sicakligidegistir(sicaklik);**

terimiyle sicakligidegistir metotuna aktarilabilir. Metot tanimlanirken, sicakligin boyutlu degisken oldugu tanimlanmalidir:

### double void sicakligidegistir(int sicaklik[])

**{**

**..........................**

**}**

gibi.

Metotlarin çikti degiskenleri de çok boyutlu olarak tanimlanabilir ve kullanilabilir. Örnegin :

### public static double[][] inversematris(double[][] a) gibi.

### BOYUTLU DEGISKENLERDE BOYUT DEGISTIRME

Boyutlu degiskenlerin boyutlari ilk tanimlamalarinda belirtildiginden normal olarak degistirilemez. Ancak indirek yollarla boyutun degistirilmesi mümkündür. Bu yol önce yeni boyutta bir boyutlu degisken tanimlamak, sonra boyutlu degiskenin içerisindeki degerleri bu yeni degiskene aktarmak ve sonra orijinal boyutlu degiskenin adresini yeni olusturulan boyutlu degisken adresiyle degistirmek seklinde yapilir.

### double a[]={3.0,5.0,7.0} double b[]=new double[4]; for(int i=0;i<a.length;i++)

**{b[i]=a[i];} a=b;**

program parçaciginda a degiskeninin boyutu üçten 4 e degistirilmistir. Daha detayli bir örnekle boyut degistirmeyi inceleyelim.

### Program 4.9 : boyutlu degistirme örnegi : doubleBoyut.java

public class doubleBoyut

{

//sinif degiskenleri public double a[]; public int length;

public doubleBoyut(double x[] )

{

length=x.length; a=new double[length];

for(int i=0;i<length;i++) a[i]=x[i];

}

public doubleBoyut(int n)

{

a=new double[n]; length=a.length;

}

public void boyutEkle(int n)

{

if(n>0)

{

int z=a.length+n;

double[] b=new double[z]; for(int i=0;i<a.length;i++)

{b[i]=a[i];} a=b; length=a.length;

}

}

public void boyutEkle()

{

boyutEkle(1);

}

public void boyutAzalt(int n)

{

int z=a.length-n; if(z>1)

{

double[] b=new double[a.length-n]; for(int i=0;i<z;i++)

b[i]=a[i]; a=b;

}

else

{

double[] b=new double[1]; b[0]=a[0];

a=b;

}

length=a.length;

}

public void boyutAzalt()

{

boyutAzalt(1);

}

public double getValue(int i)

{

return a[i];

}

public double[] getValue()

{

return a;

}

public void setValue(double x,int i)

{

a[i]=x;

}

public void setValue(double[] x )

{

length=x.length; a=new double[length];

for(int i=0;i<length;i++)

{

a[i]=x[i];

}

}

public String toString(int i)

{

return ""+a[i];

}

public String toString()

{

String s="";

for(int i=0;i<length;i++)

{

s+=a[i]+" ";

}

s+="\n"; return s;

}

}

DoubleBoyut sinifinda boyutu degistirilebilen degisken (nesne) tanimladik. Bu degiskeni boyutDegistir sinifinda test edelim :

### Program 4.10 : boyutlu degistirme testi : doubleBoyut sinifini kullanan boyutDegistir.java

import javax.swing.JOptionPane; import doubleBoyut;

public class boyutDegistir

{

public static void main(String args[])

{

double x[]={3.1,5.3,7.0,9.7,11.0,11.5,12.3};

doubleBoyut y=new doubleBoyut(x);

String s="Orijinal boyutlu ("+y.length+") double : \n"; s+=y.toString();

y.boyutEkle(3);

s+="3 eklenmis boyutlu ("+y.length+")double : \n"; s+=y.toString();

y.boyutAzalt(5);

s+="5 azaltilmis boyutlu ("+y.length+")double : \n"; s+=y.toString(); JOptionPane.showMessageDialog(null,s,

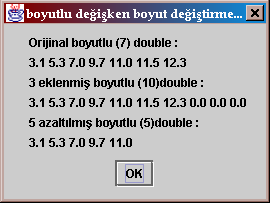
"boyutlu degisken boyut degistirme eksersizi", JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Program çiktisi :

### 04006.JPG



**Sekil 4.6 boyut degistirme örnegi boyutu degistirilebilen doubleBoyut sinifinin boyutDegistir sinifinda JoptionPane çiktisi olarak görünümü**

### ALISTIRMALAR

1. nokta3 ve dogru3 siniflarinin tanimlari verilmistir. Bu tanimlari kullanarak konsol ortaminda 5 adet dogru3 tanimlayiniz. Dogrularin baslangiç ve bitis noktalarini konsol ortamindan giriniz. ve dogrularin boylarinin ortalamasini konsol ortaminda ekrana yazdiriniz.

### Program 4.11 : nokta3.java

public class nokta3

{

protected double x, y, z;

//kurucu metotlar public nokta3()

{

nokta3gir(0,0,0);

}

public nokta3(double a, double b,double c)

{

nokta3gir(a,b,c);

}

public nokta3(nokta3 n)

{

nokta3gir(n.Xoku(),n.Yoku(),n.Zoku());

}

// giriŸ c• k• s metotlar•

public void nokta3gir(double a, double b,double c)

{

x = a;

y = b; z = c;

}

public void nokta3gir(nokta3 in1)

{

x = in1.x; y = in1.y; z = in1.z;

}

public void Xgir(double a)

{

x = a;

}

public void Ygir(double b)

{

y = b;

}

public void Zgir(double c)

{

x = c;

}

public double Xoku()

{

return x;

}

public double Yoku()

{

return y;

}

public double Zoku()

{

return z;

}

public String toString()

{

return "["+x+","+y+","+z+"]";

}

}

public class nokta3

{

protected double x, y, z;

//kurucu metotlar public nokta3()

{

nokta3gir(0,0,0);

}

public nokta3(double a, double b,double c)

{

nokta3gir(a,b,c);

}

public nokta3(nokta3 n)

{

nokta3gir(n.Xoku(),n.Yoku(),n.Zoku());

}

// giriŸ c• k• s metotlar•

public void nokta3gir(double a, double b,double c)

{

x = a; y = b;

z = c;

}

public void nokta3gir(nokta3 in1)

{

x = in1.x; y = in1.y; z = in1.z;

}

public void Xgir(double a)

{

x = a;

}

public void Ygir(double b)

{

y = b;

}

public void Zgir(double c)

{

x = c;

}

public double Xoku()

{

return x;

}

public double Yoku()

{

return y;

}

public double Zoku()

{

return z;

}

public String toString()

{

return "["+x+","+y+","+z+"]";

}

}

### Program 4.12 : dogru3.java

public class dogru3

{

// 3 boyutlu dogru nokta3 n1; nokta3 n2;

public dogru3()

{

n1=new nokta3(); n2=new nokta3();

}

public dogru3(double n1x, double n1y,double n1z, double n2x, double n2y,double n2z)

{

n1=new nokta3(n1x,n1y,n1z); n2=new nokta3(n2x,n2y,n2z);

}

public dogru3(nokta3 na,nokta3 nb)

{

n1=new nokta3(na.Xoku(),na.Yoku(),na.Zoku());

n2=new nokta3(nb.Xoku(),nb.Yoku(),nb.Zoku());

}

public void dogru3gir(double x1, double y1,double z1, double x2, double y2,double z2)

{

n1.nokta3gir(x1,y1,z1); n2.nokta3gir(x2,y2,z2);

}

public void dogru3gir(nokta3 in1, nokta3 in2 )

{

n1.nokta3gir(in1.x,in1.y,in1.z); n2.nokta3gir(in2.x,in2.y,in2.z);

}

public void n1gir(double x1,double y1,double z1)

{

n1.nokta3gir(x1,y1,z1);

}

public void n1gir(nokta3 in1)

{

n1.nokta3gir(in1.x,in1.y,in1.z);

}

public void n2gir(double x2,double y2,double z2)

{

n2.nokta3gir(x2,y2,z2);

}

public void n2gir(nokta3 in2)

{

n2.nokta3gir(in2.x,in2.y,in2.z);

}

public nokta3 n1oku()

{

return n1;

}

public nokta3 n2oku()

{

return n2;

}

public double X()

{

//dogrunun x koordinat uzunlugu return (n2.Xoku()-n1.Xoku());

}

public double Y()

{

//dogrunun y koordinat uzunlugu return (n2.Yoku()-n1.Yoku());

}

public double Z()

{

//dogrunun z coordinat uzunlugu return (n2.Zoku()-n1.Zoku());

}

public double ac• \_n1\_xy()

{

//dogrunun n1 noktas• ndan x y duzlemindeki ac• s• return Math.atan2(Y(),X());

}

public double ac• \_n1\_xz()

{

// dogrunun n1 noktasindan x z duzlemindeki açisi

// radyan

return Math.atan2(Z(),X());

}

public double ac• \_n1\_yz()

{

//dogrunun n1 noktas• ndan y z duzlemindeki ac• s•

//radyan

return Math.atan2(Z(),Y());

}

public double R()

{

//dogrunun boyu

return Math.sqrt(X()\*X()+Y()\*Y()+Z()\*Z());

}

public String toString()

{

return "n1: "+n1.toString()+"n2: "+n2.toString();

}

}

1. Sizin için kompleks sinifi yaratilmis ve bu sinifin nasil kullanildigini açiklayan H8A1 applet programi verilmistir. Bu iki programi incele. Ayrica Aritmetik2 programini incele ve boyutlu degiskenlerin kullanimini ögren. Sonra 5 kompleks sayinin toplamini hesaplayan bir konsol programi yaz.

### Program 4.13 H8A1.java programi

import java.io.\*; //java girdi cikti sinifini cagir import kompleks;

class **H8A1**

{

public static void main(String args[])

{

kompleks sayi[]=new kompleks[5]; sayi[0]=new kompleks(1,2); sayi[1]=new kompleks(1,-1); sayi[2]=new kompleks(2,0); sayi[3]=new kompleks(1.1,-0.5); sayi[4]=new kompleks(1,1.5); kompleks toplam=new kompleks(); int i;

for(i=0;i<sayi.length;i++)

{

toplam.topla(sayi[i]);

}

toplam.bol((double)sayi.length); System.out.println("Ortalama = "+toplam.toString());

}

}

Ortalama = (1.22 + 0.4i )

### Program 4.14 H8A1a.java programi

import java.io.\*; //java girdi cikti sinifini cagir import kompleks;

import Text; class H8A1a

{

public static void main(String args[]) throws IOException

{

kompleks sayi[]=new kompleks[5]; double n1,n2;

kompleks toplam=new kompleks(); int i;

Text cin=new Text(); for(i=0;i<sayi.length;i++)

{

System.out.println("bir kompleks sayi giriniz : "); n1=Text.readDouble(cin); n2=Text.readDouble(cin);

sayi[i]=new kompleks(n1,n2); toplam.topla(sayi[i]);

}

toplam.bol((double)sayi.length); System.out.println("Ortalama = "+toplam.toString());

}

}

bir kompleks sayi giriniz : 1 1 bir kompleks sayi giriniz : 2 2 bir kompleks sayi giriniz : 3 3 bir kompleks sayi giriniz : 4 4 bir kompleks sayi giriniz : 5 5 Ortalama = (3.0 + 3.0i )

1. Üç boyutlu nokta ve dogru tanimlari nokta3 ve dogru3 siniflarinda verilmistir. 3 tane dogruyu boyutlu degisken olarak **H8A2** sinifinda tanimlayin ve dogrularin toplam boylarini ve baslangiç noktasinin xy xz ve yz düzlemlerinde yaptigi açilari derece cinsinden yazdirin.

### Program 4.15 H8A1.java programi

import java.io.\*; //java girdi cikti sinifini cagir import kompleks;

class H8A2

{

public static void main(String args[])

{

dogru3 n[]=new dogru3[3]; n[0]=new dogru3(1,1,1,2,2,2);

n[1]=new dogru3(2,2,2,3,3,3);

n[2]=new dogru3(3,3,3,4,4,4); int i;

for(i=0;i<n.length;i++)

{

System.out.println(n[i].toString()+"boy : "+n[i].R()); System.out.println("aci xy = "+n[i].aci\_n1\_xy()); System.out.println("aci xz = "+n[i].aci\_n1\_xz()); System.out.println("aci yz = "+n[i].aci\_n1\_yz());

}

}

}

n1: [1.0,1.0,1.0]n2: [2.0,2.0,2.0]boy : 1.7320508075688772 aci xy = 0.7853981633974483

aci xz = 0.7853981633974483 aci yz = 0.7853981633974483

n1: [2.0,2.0,2.0]n2: [3.0,3.0,3.0]boy : 1.7320508075688772 aci xy = 0.7853981633974483

aci xz = 0.7853981633974483 aci yz = 0.7853981633974483

n1: [3.0,3.0,3.0]n2: [4.0,4.0,4.0]boy : 1.7320508075688772

aci xy = 0.7853981633974483 aci xz = 0.7853981633974483 aci yz = 0.7853981633974483

* 1. **H5AL3.java** programi. . Programi inceleyiniz.

### Program 4.16 H5AL3.java programi

//bu sinif sinif degiskeni kompleksi boyutlu olarak çagirir import java.io.\*; //java girdi cikti sinifini cagir

import kompleks;

class H5AL3

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); double n1,n2;

System.out.print("kompleks sayilarin miktari, n = "); int n=cin.readInt();

kompleks sayi[]=new kompleks[n]; kompleks toplam=new kompleks(); int i;

for(i=0;i<sayi.length;i++)

{

System.out.print("bir kompleks sayi giriniz : "); n1=cin.readDouble();

n2=cin.readDouble(); sayi[i]=new kompleks(n1,n2); toplam.topla(sayi[i]);

}

toplam.bol((double)sayi.length); System.out.println("Ortalama = "+toplam.toString());

}

}

* 1. **H5O1** : java konsol programinda toplam bayagikesir sayisini sorduktan sonra ekrandan verilen sayida bayagi kesiri giriniz ve ortalamalarini hesaplayiniz.

### Program 4.17 H5O1.java programi

//bu sinif sinif degiskeni kompleksi boyutlu olarak çagirir import java.io.\*; //java girdi cikti sinifini cagir

import bayagikesir;

class H5O1

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); int n1,n2;

System.out.print("toplam bayagi kesir sayisi, n = "); int n=cin.readInt();

bayagikesir sayi[]=new bayagikesir[n]; bayagikesir toplam=new bayagikesir(); int i;

for(i=0;i<sayi.length;i++)

{

System.out.print("bir bayagikesir giriniz : "); n1=cin.readInt();

n2=cin.readInt();

sayi[i]=new bayagikesir(n1,n2);

}

for(i=0;i<sayi.length;i++)

{

try{

toplam.topla(sayi[i]);

}

catch (bayagikesirException e)

{

System.err.println(e.getMessage());

}

}

System.out.println("Toplam = "+toplam.toString());

try{

toplam.bol(sayi.length);

}

catch (bayagikesirException e)

{

System.err.println(e.getMessage());

}

System.out.println("Ortalama = "+toplam.toString());

}

}

### H5O3

Üç boyutlu n kompleks sayiyi java konsol programinda toplam kompleks sayilarin sayisini sorduktan sonra ekrandan verilen sayida kompleks sayiyi giriniz. Kompleks sayilar girildikten sonra, kompleks sayilarin ortalamasini hesaplayiniz.

### Program 4.18 H5O3.java programi

//bu sinif sinif degiskeni kompleksi boyutlu olarak çagirir import java.io.\*; //java girdi cikti sinifini cagir

import kompleks;

class H5O3

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); double n1,n2;

System.out.print("toplam kompleks sayi, n = "); int n=cin.readInt();

kompleks sayi[]=new kompleks[n]; kompleks toplam=new kompleks(); int i;

for(i=0;i<sayi.length;i++)

{

System.out.print("kompleks sayiyi giriniz : "); n1=cin.readDouble();

n2=cin.readDouble(); sayi[i]=new kompleks(n1,n2);

}

for(i=0;i<sayi.length;i++)

{

toplam.topla(sayi[i]);

}

System.out.println("Toplam = "+toplam.toString()); toplam.bol(sayi.length);

System.out.println("Ortalama = "+toplam.toString());

}

}

### 15 H5OD1\_2000

Problem 15 de verilen H5O3.java örnek programini inceleyiniz. Java konsol programinda (Text sinifini kullanabilirsiniz) toplam kompleks sayisini sorduktan sonra ekrandan verilen sayida kompleks sayiyi giriniz ve kareköklerinin ortalamalarini hesaplayiniz.

### Program 4.19 H5OD1\_2000.java programi

//bu sinif sinif degiskeni kompleksi boyutlu olarak çagirir import java.io.\*; //java girdi cikti sinifini cagir

import kompleks;

class H5OD1\_2000

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); int n1,n2;

System.out.print("toplam kompleks sayi, n = "); int n=cin.readInt();

kompleks sayi[]=new kompleks[n]; kompleks toplam=new kompleks(); int i;

for(i=0;i<sayi.length;i++)

{

System.out.print("bir kompleks sayi giriniz : "); n1=cin.readInt();

n2=cin.readInt();

sayi[i]=new kompleks(n1,n2);

}

for(i=0;i<sayi.length;i++)

{

toplam.topla(kompleks.karekok(sayi[i]));

}

System.out.println("kompleks sayilarin kareköklerinin toplami = "+toplam.toString()); toplam.bol(sayi.length);

System.out.println("kompleks sayilarin karekoklerinin ortalamasi = "+toplam.toString());

}

}

### 15. H5OD2\_2000

Double boyutlu sayilarin boyutlarini degistirerek kullanabilen doubleBoyut sinifi yukarida tanimlanmistir. Integer (tamsayi) degiskenlerin boyutlarini degistirerek kullanabilecegimiz intBoyut sinifini olusturunuz. Bir test programi yaziniz. Bu programda bir degiskene 5 boyutlu bir integer degisken gurubu yükleyiniz, sonra boyutu 6 ya çikariniz ve sonrada boyutu 4 e düsürünüz. Sonuçlari yazdiriniz.

### Program 4.20 intBoyut.java programi

public class intBoyut

{

//sinif degiskenleri public int a[]; public int length;

public intBoyut(int x[] )

{

length=x.length; a=new int[length];

for(int i=0;i<length;i++) a[i]=x[i];

}

public intBoyut(int n)

{

a=new int[n]; length=a.length;

}

public void boyutEkle(int n)

{

if(n>0)

{

int z=a.length+n; int[] b=new int[z];

for(int i=0;i<a.length;i++)

{b[i]=a[i];} a=b; length=a.length;

}

}

public void boyutEkle()

{

boyutEkle(1);

}

public void boyutAzalt(int n)

{

int z=a.length-n; if(z>1)

{

int[] b=new int[a.length-n]; for(int i=0;i<z;i++)

b[i]=a[i]; a=b;

}

else

{

int[] b=new int[1]; b[0]=a[0];

a=b;

}

length=a.length;

}

public void boyutAzalt()

{

boyutAzalt(1);

}

public int getValue(int i)

{

return a[i];

}

public int[] getValue()

{

return a;

}

public void setValue(int x,int i)

{

a[i]=x;

}

public void setValue(int[] x )

{

length=x.length; a=new int[length];

for(int i=0;i<length;i++)

{

a[i]=x[i];

}

}

public String toString(int i)

{

return ""+a[i];

}

public String toString()

{

String s="";

for(int i=0;i<length;i++)

{

s+=a[i]+" ";

}

s+="\n"; return s;

}

}

### Program 4.21 H5OD2\_2000.java programi

import javax.swing.JOptionPane; import doubleBoyut;

public class H5OD2\_2000

{

public static void main(String args[])

{

int x[]={3,5,7,9,11};

intBoyut y=new intBoyut(x);

String s="Orijinal boyutlu ("+y.length+") integer : \n"; s+=y.toString();

y.boyutEkle();

s+="1 eklenmis boyutlu ("+y.length+") integer : \n"; s+=y.toString();

y.boyutAzalt(2);

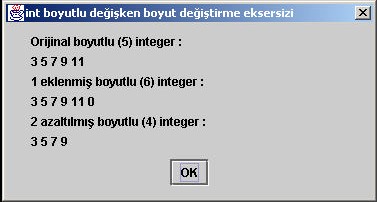
s+="2 azaltilmis boyutlu ("+y.length+") integer : \n"; s+=y.toString(); JOptionPane.showMessageDialog(null,s,

"int boyutlu degisken boyut degistirme eksersizi", JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 04007.JPG



**Sekil 4.7 intBoyut, boyutlu tam sayi boyut degistirme eksersizi, H5OD2\_2000.java, JoptionPane çiktisi 16. H5OD3\_2000**

**String isim; String Soysisim; int not;**

sinif degiskenleri ve gerekli metodlari kapsayan “ogrenci” sinifini yaratiniz.Siniftaki ögrenci sayisini sorup, daha sonra tümögrenciler için ögrenci ismi, soyismi, ve notunu giriniz (sinifinizda minimum üç ögrenci olsun). Çokti alaninda ögrencilerin isim soyisim ve not listesini siraladiktan sonra sinif ortalamasini da verin.

### Program 4.22 ogrenci.java programi

class ogrenci

{

public String isim; public String soyisim; public int not;

public ogrenci(String isimGir,String soyisimGir,int notGir)

{

isim=isimGir; soyisim=soyisimGir; not=notGir;

}

public void isimGir(String isim)

{

this.isim=isim;

}

public void soyisimGir(String soyisim)

{

this.soyisim=soyisim;

}

public void notGir(int not)

{

this.not=not;

}

public void ogrenciGir(String isimGir,String soyisimGir,int notGir)

{

isim=isimGir; soyisim=soyisimGir; not=notGir;

}

public String isimOku()

{

return isim;

}

public String soyisimOku()

{

return soyisim;

}

public int notOku()

{

return not;

}

public String toString()

{

return isim+" "+soyisim+" "+not;

}

}

### Program 4.23 H5OD3\_2000.java programi

import java.io.\*; //java girdi cikti sinifini cagir import ogrenci;

import ogrenciFrame;

class H5OD3\_2000

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); int i,n;

int not;

double toplam = 0; String isim, soyisim; ogrenci temp;

System.out.print("Lutfen ogrenci sayisini giriniz = "); n=cin.readInt();

ogrenci dizi[]=new ogrenci[n]; for(i=0;i<n;i++)

{

//System.out.print("Ogrenci ismi = "); String s[]=new String[3];

System.out.print("isim : "); isim = cin.readString(); System.out.print("soyisim : "); soyisim=cin.readString(); System.out.print("not : "); not=cin.readInt();

dizi[i]=new ogrenci(isim, soyisim, not); toplam += not;

}

toplam /= (double)n; System.out.println("Ogrenci Listesi = \n "); for(i=0;i<dizi.length;i++)

System.out.println(dizi[i].toString()); System.out.println("sinif Ortalamasi = " + toplam);

}

}

### H5OD5\_2000.java programinin çiktisi : Turhan Çoban 25

**Meral Çoban 75**

**Ali Velizade 99**

**sinif Ortalamasi = 66.33333333333333**

### BÖLÜM 5 : SINIFLARDA HIYERARSI, ABSTRACT SINIF VE INTERFACE

### ABSTRACT SINIF

Üçüncü bölümde siniflarin diger siniflardan kalitim (Inheritance) yoluyla türetilmesini görmüstük. Siniflari kalitim yoluyla birbirinden türetirken eger ortak özellikli siniflar yaratiyorsak, bütün bu sinif gurubunun en tepesine abstract bir sinif koyabiliriz. Abstract sinif diger siniflarin kullanilmasinda extends yoluyla tepe siniflik etmek ve bütün alt siniflara ortak bir adres çikis noktasi saglamak disinda bir görevi yoktur. Diger bir deyimle bu siniflar hiçbir zaman dogrudan kullanilmazlar. Ancak alt siniflarindan birini referans olarak göstermek ve onlara dolayli yoldan ulasmak amaciyla kullanilirlar.

Kavrami daha iyi verebilmek için bir örnek problem olusturalim. Daha önce sinif kavramini açiklarken kullandigimiz nokta,daire siniflarini hatirliyacaksiniz. Simdi bu siniflarin üzerine sekil isimli bir abstract sinif ekleyerek tekrar olusturalim.

### Program 5.1 Abstract sinif sekil (sekilX.java dosyasinda yer aliyor)

public abstract class sekilX

{

//not burada baska degiskenlerde olabilir public double alan(){return 0.0;}

public double hacim() {return 0.0;} public abstract String isim();

}

### Program 5.2: sekil sinifindan kalitim yoluyla türetilen noktaX sinifi

import sekilX;

public class noktaX extends sekilX

{

protected double x, y;

public noktaX(double a, double b)

{

x=a; y=b;

}

public void noktagir(double a, double b)

{

x=a; y=b;

}

public double Xoku()

{

return x;

}

public double Yoku()

{

return y;

}

public String toString()

{

return "["+x+","+y+"]";

}

public String isim() {return "nokta";}

}

### Program 5.3: noktaX sinifindan kalitim yoluyla türetilen daireX sinifi

import noktaX;

public class daireX extends noktaX

{

protected double yaricap;

public daireX()

{

//daire kalitim yaptigi nokta sinifinin kurucu metodunu

// super deyimi ile cagirabilir. super(0,0);

yaricapgir(0);

}

public daireX(double r, double a, double b)

{

super(a,b); yaricapgir(r);

}

public void yaricapgir(double r)

{

if(r >= 0.0) yaricap=r;

else

yaricap=0.0;

}

public double yaricapoku()

{

return yaricap;

}

public double alan()

{

return 3.14159\*yaricap\*yaricap;

}

public String toString()

{

return "Merkez = "+"["+x+","+y+"]"+ "; Yaricap="+yaricap;

}

public String isim() {return "daire";}

}

### Program 5.4: daireX sinifindan kalitim yoluyla türetilen silindirX sinifi

import daireX;

public class silindirX extends daireX

{

protected double yukseklik;

public silindirX()

{

yukseklikoku(0);

}

public silindirX( double h, double r, int a, int b)

{

super(r, a, b ); yukseklikoku( h);

}

public void yukseklikoku (double h )

{

if(h>=0) yukseklik=h; else yukseklik=0;

}

public double yukseklikoku(){return yukseklik;} public double alan()

{

return 2\* super.alan()+ 2\*Math.PI\*yaricap\*yukseklik;

}

public double hacim(){return super.alan()\*yukseklik;} public String toString()

{return super.toString()+";yukseklik="+yukseklik;}

public String isim(){return "silindir";}

}

### Program 5.4 : Abstract sinif ve indirek referanslamayi test eden abstracttesti sinifi

import java.awt.Graphics; import java.applet.Applet;

public class abstracttesti extends Applet

{

private noktaX n; private daireX d; private silindirX s; private sekilX a[]; public void init()

{

n = new noktaX(7,11);

d = new daireX(3.5,22,8);

s = new silindirX(10,3.3,10,10); a = new sekilX[3];

a[0]=n;

a[1]=d;

a[2]=s;

}

public void paint(Graphics g)

{

//direk cagirma :

g.drawString(n.isim()+": "+n.toString(),25,25);

g.drawString(d.isim()+": "+d.toString(),25,40);

g.drawString(s.isim()+": "+s.toString(),25,55);

//indirek cag• rma int y=85;

for(int i=0;i<a.length;i++)

{

g.drawString(a[i].isim()+": "+a[i].toString(),25,y); y+=15;

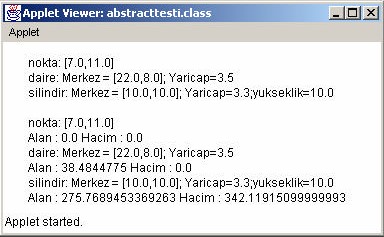
g.drawString("Alan : "+a[i].alan()+" Hacim : "+a[i].hacim(),25,y); y+=15;

}

}

}

### 05001.JPG



**Sekil 5.1 Abstract sinif ve indirek referanslamayi test eden abstractTesti sinifinin sonuclarinin applet de görülmesi**

**Program 5.5 : Abstract sinif ve indirek referanslamayi test eden abstractTesti\_2000.java**

import javax.swing.JOptionPane; import silindirX;

import noktaX; import daireX;

public class abstractTesti\_2000

{

public static void main(String args[])

{

noktaX n; daireX d; silindirX s; sekilX a[];

n = new noktaX(7,11);

d = new daireX(3.5,22,8);

s = new silindirX(10,3.3,10,10); a = new sekilX[3];

a[0]=n;

a[1]=d;

a[2]=s;

String st="";

st+=n.isim()+": "+n.toString()+"\n";

st+=d.isim()+": "+d.toString()+"\n";

st+=s.isim()+": "+s.toString()+"\n";

//indirek cagirma int y=85;

for(int i=0;i<a.length;i++)

{

st+=a[i].isim()+": "+a[i].toString()+"\n";

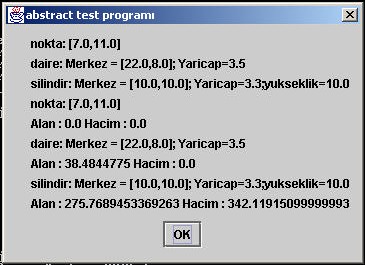
st+="Alan : "+a[i].alan()+" Hacim : "+a[i].hacim()+"\n";

}

JOptionPane.showMessageDialog(null,st,"abstract test programi", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

} 05002.JPG

### Sekil 5.2 Abstract sinif ve indirek referanslamayi test eden abstractTesti\_2000 sinifinin sonuclarinin JoptionPane çiktisi olarak görülmesi

Örnekten de görüldügü gibi nokta, daire ve silindir abstract sekil sinifinin alt siniflari olarak olusturulmus, ve ayni zamanda bu siniflara indirek referans olarak kullanilmistir. Diger bir deyimle nokta daire ve silindirin adresleri sekil sinifi nesneye aktarilmis ve bu nesne üzerinden üç sinifta indirek olarak kullanilmistir. Bu indirek kullanimin bize temel faydasi, Programimiza yeni siniflar ekledigimizde daha kolay ve az program degistirerek uyumu saglayabilme yetenegidir. Ornegin yukaridaki program siniflari gurubuna koni eklemek istersek bütün yapacagimiz koni sinifini tanimladiktan sonra, sekil nesnesi a nin boyutunu bir arttirarak ayni hesaplamaya koniyide ilave etmekten ibaret olacaktir. Bu tür uygulamalar bilgisayar kodunun degisim gereksinimini minimuma indirgediginden nesne kokenli bilgisayar programciliginin onemli uygulamalarindan biridir.

### INTERFACE

Abstract siniflarin dezavantaji sadece bir sinifin kalitim yoluyla ve extends sozcügü kullanilarak baglanabilir olmasidir. Ve eger baska bir sinif halihazirda o sinifa kalitim yoluyla baglanmissa, bizim yeni bir sinifi üst sinif olarak kullanma olasiligimiz yok olmus demektir. Fakat bazi uygulamalar için bu gerekebilir.Bu zaman **interface** kullaniriz. Interface de abstract sinif gibi kendi içinde bir islem yapmaz. Sadece öbür siniflarin baglanmasini teskil eden bir adres noktasi olarak is görür. Interface’i bir baska sinifa baglamak gerektiginde extends yerine **implements** sözcügü kullanilir. Abstract siniflar kendi basina bir bütün teskil edebilmelerine ragmen interface’in sadece kendi basina bir anlami yoktur. Interface sinifinda yer alan tüm metod ve degiskenlerin alt siniflarda tanimlanmasi gerekir.

Örnek olarak bir önceki problemi interface ile tanimliyalim.

### Program 5.5 interface sekil (sekilY.java dosyasinda yer aliyor)

public interface sekilY

{

//burada de§iŸken veya nesnelerde olabilir

public abstract double alan(); public abstract double hacim(); public abstract String isim();

}

### Program 5.6 : interface sekilden implements kelimesiyle türeyen noktaY sinifi (noktaY.java dosyasinda yer aliyor)

import sekilY;

public class noktaY implements sekil

{

protected double x, y;

public noktaY(double a, double b)

{

noktagir(a,b);

}

public void noktagir(double a, double b)

{

x=a; y=b;

}

public double Xoku()

{

return x;

}

public double Yoku()

{

return y;

}

public double alan() {return 0.0;} public double hacim() {return 0.0;} public String toString()

{

return "["+x+","+y+"]";

}

public String isimOku()

{

return "Nokta";

}

}

Görüldügü gibi burada nokta sinifinin bir öncekinden farki implements kelimesinin kullanilmis olamasi ve bos alan ve hacim metotlarinin bu dosyada yazilmasinin gerekmesinden ibarettir. Interface programla yapilan ve alt siniflarda bu metodlarin (veya degiskenlerin) tanimlanacagina dair yapilan bir kontrattir.DaireY ve silindirY siniflarinda bir fark mevcut degildir. Iki seklin islemsel sonuçlari arasinda da bir fark mevcut degildir.

Yukardaki örnege parelel olarak burada da daireY,SilindirY tanimlarini yapalim ve interfaceTesti\_2000.java programinda test edelim.

### Program 5.6 : interface sekilY’den implements kelimesiyle türeyen noktaY sinifindan extends yoluyla türeyen daireY sinifi(daireY.java dosyasinda yer aliyor)

import noktaY;

public class daireY extends noktaY

{

protected double yaricap;

public daireY()

{

//daire kalitim yaptigi nokta sinifinin kurucu metodunu

// super deyimi ile cagirabilir. super(0,0);

yaricapgir(0);

}

public daireY(double r, double a, double b)

{

super(a,b); yaricapgir(r);

}

public void yaricapgir(double r)

{

if(r >= 0.0) yaricap=r;

else

yaricap=0.0;

}

public double yaricapoku()

{

return yaricap;

}

public double alan()

{

return 3.14159\*yaricap\*yaricap;

}

public String toString()

{

return "Merkez = "+"["+x+","+y+"]"+ "; Yaricap="+yaricap;

}

public String isim() {return "daire";}

}

### Program 5.7 : interface sekilY’den implements kelimesiyle türeyen noktaY sinifindan extends yoluyla türeyen daireY sinifindan extends yoluyla türeyen silindirY sinifi(silindirY.java dosyasinda yer aliyor)

import daireY;

public class silindirY extends daireY

{

protected double yukseklik;

public silindirY()

{

super(0, 0, 0 ); yukseklikoku(0);

}

public silindirY( double h, double r, int a, int b)

{

super(r, a, b ); yukseklikoku( h);

}

public void yukseklikoku (double h )

{

if(h>=0) yukseklik=h; else yukseklik=0;

}

public double yukseklikoku(){return yukseklik;} public double alan()

{

return 2\* super.alan()+ 2\*Math.PI\*yaricap\*yukseklik;

}

public double hacim(){return super.alan()\*yukseklik;} public String toString()

{return super.toString()+";yukseklik="+yukseklik;}

public String isim(){return "silindir";}

}

### Program 5.8 : interface sekilY’den türeyen siniflari test eden ve indirek olarak referansliyan interfaceTesti\_2000.java programi

import javax.swing.JOptionPane; import silindirY;

import noktaY; import daireY;

public class interfaceTesti\_2000

{

public static void main(String args[])

{

noktaY n; daireY d; silindirY s; sekilY a[];

n = new noktaY(7,11);

d = new daireY(3.5,22,8);

s = new silindirY(10,3.3,10,10); a = new sekilY[3];

a[0]=n;

a[1]=d;

a[2]=s;

String st="";

st+=n.isim()+": "+n.toString()+"\n";

st+=d.isim()+": "+d.toString()+"\n";

st+=s.isim()+": "+s.toString()+"\n";

//indirek cagirma int y=85;

for(int i=0;i<a.length;i++)

{

st+=a[i].isim()+": "+a[i].toString()+"\n";

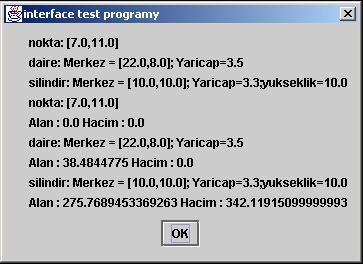
st+="Alan : "+a[i].alan()+" Hacim : "+a[i].hacim()+"\n";

}

JOptionPane.showMessageDialog(null,st,"interface test programi", JOptionPane.PLAIN\_MESSAGE);

System.exit(0);

}

} 05003.JPG

### Sekil 5.2 interface sinif ve indirek referanslamayi test eden interfaceTesti\_2000 sinifinin sonuclarinin JoptionPane çiktisi olarak görülmesi

### ALISTIRMALAR

1. Derste gördügümüz **abstract class sekilX,**noktaX,daireX, silindirX siniflarina ve ilave olarak koniX sinifini yaratiniz. Bu siniflarin sekiltestiX.java test sinifinda alanlarini ve hacimlerini sekilX referansini kullanarak yazdiriniz.
2. Derste gördügümüz **interface sekilY,** noktaY,daireY, silindirY siniflarina ve ilave olarak küreY sinifini yaratiniz. Bu siniflarin sekiltesti.java testY sinifinda alanlarini ve hacimlerini sekilY referansini kullanarak yazdiriniz.
3. **H5AL1.java** programini inceleyiniz. Bu program bir boyutlu abstract sinif fx ve bu sinifta tanimlanan abstract metot f(double x) i kullanan turev ve integral metodlarini içerir. Integral ve türev metodlarini niçin abstact kavrami kullanarak yazdigimizi açiklayiniz.

Program 5.7 H5AL1.java, fx ve H5AL1 siniflarinin tanimi

//abstract sinif ve boyutlu degiskenler alistirmasi abstract class fx

{

abstract double f(double x);

}

public class H5AL1

{

public static double turev(fx fi,double x)

{

// bu metod birinci dereceden bir fonksiyonun t• revini hesaplar

// metodun hata miktari h0 and n degiskenlerinin degisimi ile

// degistirilebilir. fonksiyon girisi fx abstract s• n• f• • zerinden

// yapilmalidir.

// Bu problem boyutlu de§iskenerin kullan• lmasi ile ilgili de bir

// ornektir.

double h0=0.0256; int i,m;

int n=7;

//basit bir fonksiyonun t• revi double T[][];

T=new double[n][n]; double h[];

h=new double[n]; for(i=0;i<n;i++)

{

h[i]=0;

for(int j=0;j<n;j++) T[i][j]=0;

}

h[0]=h0;

double r=0.5; for( i=1;i<n;i++)

{

h[i]=h0\*Math.pow(r,i);

}

for(i=0;i<n;i++)

{

T[i][0]=( fi.f(x + h[i]) - fi.f( x - h[i]))/(2.0\*h[i]);

}

for(m=1;m<n;m++)

{

for(i=0;i<n-m;i++)

{

T[i][m]=(h[i]\*h[i]\*T[i+1][m-1] - h[i+m]\*h[i+m]\*T[i][m-1])/ (h[i]\*h[i]- h[i+m]\*h[i+m]);

}

}

double xx=T[0][n-1]; return xx;

}

public static double integral(fx fi,double a,double b)

{

//gauss-legendre denklemini kullanarak integral hesaplama

//

double s[],w[]; int i;

s=new double[30]; w=new double[30];

s[ 0] = .15532579626752470000E-02; s[ 1] = .81659383601264120000E-02; s[ 2] = .19989067515846230000E-01; s[ 3] = .36899976285362850000E-01; s[ 4] = .58719732103973630000E-01; s[ 5] = .85217118808615820000E-01; s[ 6] = .11611128394758690000E+00; s[ 7] = .15107475260334210000E+00; s[ 8] = .18973690850537860000E+00; s[ 9] = .23168792592899010000E+00; s[10] = .27648311523095540000E+00; s[11] = .32364763723456090000E+00; s[12] = .37268153691605510000E+00; s[13] = .42306504319570830000E+00; s[14] = .47426407872234120000E+00; s[15] = .52573592127765890000E+00; s[16] = .57693495680429170000E+00; s[17] = .62731846308394490000E+00; s[18] = .67635236276543910000E+00; s[19] = .72351688476904450000E+00;

s[20] = .76831207407100990000E+00; s[21] = .81026309149462140000E+00; s[22] = .84892524739665800000E+00; s[23] = .88388871605241310000E+00; s[24] = .91478288119138420000E+00; s[25] = .94128026789602640000E+00; s[26] = .96310002371463720000E+00; s[27] = .98001093248415370000E+00; s[28] = .99183406163987350000E+00; s[29] = .99844674203732480000E+00; w[ 0] = .39840962480827790000E-02; w[ 1] = .92332341555455000000E-02; w[ 2] = .14392353941661670000E-01; w[ 3] = .19399596284813530000E-01; w[ 4] = .24201336415292590000E-01; w[ 5] = .28746578108808720000E-01; w[ 6] = .32987114941090080000E-01; w[ 7] = .36877987368852570000E-01; w[ 8] = .40377947614710090000E-01; w[ 9] = .43449893600541500000E-01; w[10] = .46061261118893050000E-01; w[11] = .48184368587322120000E-01; w[12] = .49796710293397640000E-01; w[13] = .50881194874202750000E-01; w[14] = .51426326446779420000E-01; w[15] = .51426326446779420000E-01; w[16] = .50881194874202750000E-01; w[17] = .49796710293397640000E-01; w[18] = .48184368587322120000E-01; w[19] = .46061261118893050000E-01; w[20] = .43449893600541500000E-01; w[21] = .40377947614710090000E-01; w[22] = .36877987368852570000E-01; w[23] = .32987114941090080000E-01; w[24] = .28746578108808720000E-01; w[25] = .24201336415292590000E-01; w[26] = .19399596284813530000E-01; w[27] = .14392353941661670000E-01; w[28] = .92332341555455000000E-02; w[29] = .39840962480827790000E-02;

int n=30; double z=0; double x,y; for(i=0;i<n;i++)

{

x=(b+a)/2.0+(b-a)/2.0\*s[i]; y=fi.f(x);

z+=(b-a)/2\*w[i]\*y;

}

for(i=0;i<n;i++)

{

x=(b+a)/2.0+(b-a)/2.0\*(-s[i]); y=fi.f(x);

z+=(b-a)/2.0\*w[i]\*y;

}

return z;

}

}

1. **H5AL2.java** programi alistirma 3 de verilen turev ve integral siniflarini konsol ortaminda kullanmaktadir. Programin inceleyiniz.

### Program 5.8 H5AL2.java

import java.io.\*; import H5AL1;

class f1 extends fx

{

double f(double x)

{

return x\*x;

}

}

class f2 extends fx

{

double f(double x)

{

return Math.sin(x);

}

}

public class H5AL2

{

public static void main(String args[]) throws IOException

{

f1 b1; b1=new f1();

System.out.println("Integral : "+H5AL1.integral(b1,0.0,1.0)); System.out.println("Turev : "+H5AL1.turev(b1,1.0));

f2 b2; b2=new f2();

System.out.println("Integral : "+H5AL1.integral(b2,0.0,Math.PI)); System.out.println("Turev : "+H5AL1.turev(b2,Math.PI));

}

}

**5.** H5O2 :

abstract class fxi

{

abstract double f(double x[]);

}

tanimini ve H5AL1 sinifindaki turev metotuna H5O2 sinifinda public static double turev(fx fi,double x)

gerekli degisiklikleri yaparak

public static double turev(fxi fi,double x[],int i)

metotunu olusturunuz ve i'inci x degeri için denklemin türevini hesaplayiniz. ( dfi(x[])/dxi türevini hesaplayiniz. )

### Program 5.9 H5O2.java

//abstract sinif ve boyutlu degiskenler odevi abstract class fxi

{

abstract double f(double x[]);

}

public class H5O2

{

public static double turev(fxi fi,double x[],int k)

{

// bu metot ninci dereceden bir fonksiyonun t• revini hesaplar

// metotun hata miktari h0 and n degiskenlerinin degisimi ile

// degistirilebilir. fonksiyon girisi fxi abstract s• n• f• • zerinden

// yapilmalidir.

// Bu problem boyutlu de§iskenerin kullan• lmasi ile ilgili de bir

// ornektir.

double h0=0.0256; int i,m;

int n=7;

//basit bir fonksiyonun t• revi double T[][];

T=new double[n][n]; double h[];

h=new double[n]; for(i=0;i<n;i++)

{

h[i]=0;

for(int j=0;j<n;j++) T[i][j]=0;

}

h[0]=h0;

double r=0.5; for( i=1;i<n;i++)

{

h[i]=h0\*Math.pow(r,i);

}

for(i=0;i<n;i++)

{

double temp=x[k]; x[k]=x[k] + h[i]; double fiP=fi.f(x); x[k]=temp - h[i]; double fiM=fi.f(x); x[k]=temp;

T[i][0]=( fiP - fiM)/(2.0\*h[i]);

}

for(m=1;m<n;m++)

{

for(i=0;i<n-m;i++)

{

T[i][m]=(h[i]\*h[i]\*T[i+1][m-1] - h[i+m]\*h[i+m]\*T[i][m-1])/ (h[i]\*h[i]- h[i+m]\*h[i+m]);

}

}

double xx=T[0][n-1]; return xx;

}

public static double integral(fxi fi,double x[],int k, double a,double b)

{

//gauss-legendre denklemini kullanarak integral hesaplama

//integral(x[k]=a dan x[k]=b ye kadar fonksiyon fi(x), x=x[0]..x[k]..x[n]

//

double s[],w[]; double tempxk=x[k]; int i;

s=new double[30]; w=new double[30];

s[ 0] = .15532579626752470000E-02; s[ 1] = .81659383601264120000E-02; s[ 2] = .19989067515846230000E-01; s[ 3] = .36899976285362850000E-01; s[ 4] = .58719732103973630000E-01; s[ 5] = .85217118808615820000E-01; s[ 6] = .11611128394758690000E+00; s[ 7] = .15107475260334210000E+00; s[ 8] = .18973690850537860000E+00; s[ 9] = .23168792592899010000E+00; s[10] = .27648311523095540000E+00; s[11] = .32364763723456090000E+00; s[12] = .37268153691605510000E+00; s[13] = .42306504319570830000E+00; s[14] = .47426407872234120000E+00; s[15] = .52573592127765890000E+00; s[16] = .57693495680429170000E+00; s[17] = .62731846308394490000E+00; s[18] = .67635236276543910000E+00; s[19] = .72351688476904450000E+00; s[20] = .76831207407100990000E+00; s[21] = .81026309149462140000E+00; s[22] = .84892524739665800000E+00; s[23] = .88388871605241310000E+00; s[24] = .91478288119138420000E+00; s[25] = .94128026789602640000E+00; s[26] = .96310002371463720000E+00; s[27] = .98001093248415370000E+00; s[28] = .99183406163987350000E+00; s[29] = .99844674203732480000E+00; w[ 0] = .39840962480827790000E-02; w[ 1] = .92332341555455000000E-02; w[ 2] = .14392353941661670000E-01; w[ 3] = .19399596284813530000E-01; w[ 4] = .24201336415292590000E-01; w[ 5] = .28746578108808720000E-01; w[ 6] = .32987114941090080000E-01; w[ 7] = .36877987368852570000E-01; w[ 8] = .40377947614710090000E-01; w[ 9] = .43449893600541500000E-01; w[10] = .46061261118893050000E-01; w[11] = .48184368587322120000E-01; w[12] = .49796710293397640000E-01; w[13] = .50881194874202750000E-01; w[14] = .51426326446779420000E-01; w[15] = .51426326446779420000E-01; w[16] = .50881194874202750000E-01; w[17] = .49796710293397640000E-01; w[18] = .48184368587322120000E-01; w[19] = .46061261118893050000E-01; w[20] = .43449893600541500000E-01; w[21] = .40377947614710090000E-01; w[22] = .36877987368852570000E-01; w[23] = .32987114941090080000E-01;

w[24] = .28746578108808720000E-01; w[25] = .24201336415292590000E-01; w[26] = .19399596284813530000E-01; w[27] = .14392353941661670000E-01; w[28] = .92332341555455000000E-02; w[29] = .39840962480827790000E-02;

int n=30; double z=0; double y; for(i=0;i<n;i++)

{

x[k]=(b+a)/2.0+(b-a)/2.0\*s[i]; y=fi.f(x);

z+=(b-a)/2\*w[i]\*y;

}

for(i=0;i<n;i++)

{

x[k]=(b+a)/2.0+(b-a)/2.0\*(-s[i]); y=fi.f(x);

z+=(b-a)/2.0\*w[i]\*y;

}

x[k]=tempxk; return z;

}

}

1. Program 5.10 da verilen polar sinifini inceleyiniz. Polar sinifinin kompleks sinifiyla oldukça benzer oldugunu gözlemleyebilirsiniz. Bu iki sinifin temel özelliklerinden yararlanan abstact bir sinif olusturunuz, bu sinifin alt siniflari olarak polar ve kompleks siniflarini yeniden tanimlayiniz. Bir örnek programda abstract sinif üzerinden polar ve kompleks siniflari çagirarak kullaniniz.

### Program 5.10 Polar sinifi tanimi Polar.java

public class polar

{

//duzlemde tanimlanmis polar koordinat sistemi protected double R,teta;

public polar()

{ R=0;

teta=0;

}

public polar(double Ri,double tetai)

{ R=Ri;

teta=tetai;

}

public polar(polar y)

{ R=y.ROku();

teta=y.tetaOku();

}

public void RGir(double Ri)

{ R=Ri;

}

public void tetaGir(double tetai)

{

teta=tetai;

}

public void polarGir(double Ri,double tetai)

{ R=Ri;

teta=tetai;

}

public void kartesienGir(double xi,double xj)

{

R=Math.sqrt(xi\*xi+xj\*xj); teta=Math.atan2(xj,xi);

}

public double ROku()

{

return R;

}

public double tetaOku()

{

return teta;

}

public double xi()

{

return R\*Math.cos(teta);

}

public double xj()

{

return R\*Math.sin(teta);

}

public void topla(polar y)

{

kartesienGir((xi()+y.xi()),(xj()+y.xj()));

}

public void fark(polar y)

{

kartesienGir((xi()-y.xi()),(xj()-y.xj()));

}

public polar polarOku()

{

return this;

}

public boolean esittir(polar v)

{

boolean b=((R==v.ROku())&&(teta==v.tetaOku())); return b;

}

public boolean buyuktur(polar v)

{

return (this.R>v.ROku());

}

public boolean kucuktur(polar v)

{

return (this.R<v.ROku());

}

public String toString()

{

return ""+R+"\*exp("+teta+"i) ";

}

}

### BÖLÜM 6 : GRAFIKLER, FONTLAR VE RENKLER

### GIRIS

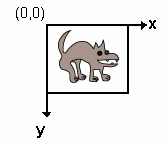
Daha önce applet çizerken **java.awt** kütüphanesindeki grafik (Graphics) sinifini kullanmistik. Ve grafik (Graphics) sinifinina ait drawString metotunu kullanarak applete yazi yazdirmistik. Simdi bu sinifin diger özelliklerini ve metotlarini ögrenecegiz.

Graphics sinifi java.lang.Object sinifinin altinda yer alir. Bu gurupta ayni zamanda grafik çiziminin çesitli fazlarinda kullanilan java.awt.Color, java.awt:Font, java.awt.FontMetrics, java.awt, java.awt.Graphics, java.awt.Polygon yer alir. Daha detayli grafik çizme islemleri tanimlayabilen java.awt. Graphics2D paketi java.awt.Graphics paketinden türetilmistir. Ayrica java.awt.BasicStroke, java.awt.GradientPaint, java.awt.TexturePaint gibi temel grafik alt programlari da java.lang.Object altinda tanimlanmistir. Temel sekilleri çizdirmeye yarayan java.awt.geom.GeneralPath, java.awt.geom.Line2D, java.awt.geom.RectangularShape de java.lang.Object altinda tanimlanir. java.awt.geom.Arc2D, . java.awt.geom.Ellipse2D, . java.awt.geom.Rectangle2D, . java.awt.geom.RoundRectangle2D gibi çizim paketleri de java.awt.geom.RectangularShape paketinin alt paketleri olarak tanimlanmistir.

Graphics sinifi, yazi yazma, çizgi çizme, dikdörtgen, oval çizme gibi bir dizi metotu barindirir. Bunun disinda awt kütüphanesinin önemli bir sinifi da Color sinifidir. Color sinifi rekleri tanimlar ve degisik renklerin kullanimina imkan verir. Font sinifi grafikte kullanilan yazilarin fontlarinin (yazi tipinin ve boyutunun) belirlenmesi amaciyla kullanilir. FontMetrics sinifi fontlarin boyutlarinin belirlenmesiyle ilgili metotlari içerir. Polygon sinifi polygon çizimiyle ilgili metotlari barindirir. Bu metodlarin bir kismini burada görecegiz. Temel olarak Graphics ve Graphics2D siniflarini grafik çiziminde kullanacagiz, Bu islemi yaparken de diger tüm yardimci grafik siniflarindan yararlanacagiz. Graphics2D sinifi daha yeni olarak tanimlanmis ve Graphics sinifindan türetilmis bir siniftir. Fakat çok daha komleks grafik çizim kapasitelerini barindirir.

Java appletlerinde çizim yaparken ilk hatirlamamiz gereken nokta, koordinat merkezinin ekranin sol üst kösesi oldugudur. X ekseni sola dogru, Y ekseni asagiya dogru gider.

### 06001.JPG



**Sekil 6.1 Java grafik koordinat sistemi, x ve y eksenleri**

Y ekseninin alta dogru gitmesi konsol programlarinda (ve ilk konsol programlama ortami olan satir yazicilarinda) satirlarin alta dogru ilerlemesi gibi tarihi bir nedene dayanmaktadir. Bütün bilgisayar diileri ve ekrani grafik programlamalarinda standartdir.

### GRAPHICS VE GRAPHICS2D SINIFLARI

Graphics sinifi grafik çizimi için gerekli olan bir çok metotu içinde barindirir. Grafik sinifi java.lang.Object sinifinin alt siniflarindandir. Grafik çizme amaciyla genellikle Graphics sinifindan bir nesne **paint veya paintComponent** metotunun içinde çagirilir. Paint metotunun tanimi söyledir

### public void paint (Graphics g)

Bu tanim Applet, Japplet,JFrame extend etmis programlardan çagirilabilir. Paint metodu tüm bu üç sinifin türetildigi Component sinifinda tanimlanmistir. javax.swing sinifi altinda tanimlanan JComponent sinifinda ise paintComponent metodu tanimlanmistir. JComponentin alt metodu JPanel bu metodu kullanir. Tanimi paint metoduna benzer

### public void paintComponent (Graphics g)

paint veya paintComponent metodlari Applet, Japplet, Frame, JFrame, Panel, JPanel siniflari tarafindan direk olarak program açildiginda çagirilir. Kullanici tarafindan çagirilmaz. Tekrar kullanici tarafindan çagirilmasi gerektiginde ise **repaint** metotu kullanilir ve repaint metotu üzerinden indirek olarak paint metotu çagirilir. Bu metodun tanimi :

### public void repaint ()

seklindedir.

Ikinci ilginç metod update metodudur. Bu metod

### public void update (Graphics g)

seklinde tanimlanmistir. Update metodu çagirildiginda Graphics metodu sistem tarafindan otomatik olarak gönderilir. Graphics2D sinifi Graphics sinifinin bir alt sinifi olarak tanimlanmistir. Bu sinif Graphicss sinifina göre çok daha kompleks çizimler yapabilen metodlara sahiptir. Burada graphics ve Graphics2D de yer alan bazi sinif ve çizim tekniklerini yakindan inceleyecegiz.

### 6.1.1 drawString, drawChars ve drawBytes metotlari

Graphics sinifinin altmetodlari olan bu metotlarin tanimlari su sekilde verilmistir.

### public abstract void drawString( String stringYazi, int x,int y)

**public void drawChars( char charYazi[], int baslangiçindeksi,int yazilacakharfsayisi, int x,int y) public void drawBytes( byte byteYazi[],**

**int baslangiçindeksi,int yazilacakharfsayisi, int x,int y)**

**drawString** metotu bir string degiskenini verilen x ve y koordinatlarindan baslayarak çizer.

DrawChars bir boyutlu Char tipi degisken dizisini baslangiçindeksi indeksinden baslayarak yazilacakharfsayisi kadar kismini x ve y koordinatlarindan basliyarak çizer. DrawByte metotu da drawChars metotu gibidir tek degisikligi Byte türü boyutlu degisken kullanmasidir.

Program 6.1. de bu metotlarin kullanilmasini açiklayan bir program verilmistir.

### Program 6.1 : ciz.java programi : drawString,drawChars,drawBytes metotlarinin kullanilmasi

import java.applet.Applet; import java.awt.\*;

public class ciz extends Applet

{

private String s ="Bunu drawString Metoduyla yazdir"; private char c[]={'c','h','a','r','s',' ','8'};

private byte b[]={'b','y','t','e',25,26,27};

public void paint(Graphics g)

{

g.setColor(new Color(0,0,100));

g.setFont(new Font("SansSerif",Font.PLAIN,24)); g.drawString(s,100,25); g.drawChars(c,2,3,100,50);

g.drawBytes(b,0,5,100,75);

}

}

### 06002.JPG



**Sekil 6.2 drawString drawChars drawBytes metotlarini kullanan ciz programinin Applet çiktisi**

Simdi ayni islemi ve ayni metodlari Graphics2D üzerinden yaptiralim :

### Program 6.2 : ciz2DSWF\_2000.java programi : drawString,drawChars,drawBytes metotlarinin swing JFrame ve Graphics2D ile birlikte kullanilmasi

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*; import renk;

public class ciz2DSWF\_2000 extends JFrame { final static Color bg = Color.white;

final static Color fg = Color.black;

final static Color red = Color.red; final static Color white = Color.white;

private String s ="Bunu drawString Metoduyla yazdir"; private char c[]={'k','a','r','e','k','t','e','r','l','e','r'};

public ciz2DSWF\_2000()

{

//Initialize drawing colors super("String character çizimi"); setBackground(bg); setForeground(fg);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g;

g2.setFont(new Font("TimesRoman",Font.BOLD,14)); g2.drawString(s,100,50);

//ikinci charecterden baslayarak 5 character yaz. g2.drawChars(c,2,5,100,100);

}

//=====================================================

public static void main(String[] args)

{

ciz2DSWF\_2000 pencere= new ciz2DSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

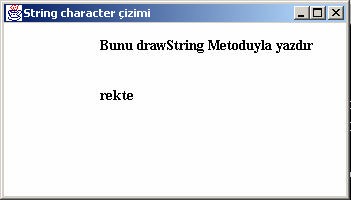
pencere.setVisible(true);

}

}

Burada hemen belirtelimki swing siniflarini kullanirken Graphics2D siniflari kullanmak zorunda degiliz, ama burada genel olarak swing ve Graphics2D kullanimlarini birlikte sunacagiz. Bu programin çiktisi da bir önceki programin çiktisinda oldugu gibidir :

### 06003.JPG



**Sekil 6.3 drawString drawChars metotlarini kullanan ciz2DSWF\_2000 programinin swing JFrame çiktisi**

Programdan da görülecegi gibi paint metodu yine Graphics cinsinden çagirilmakta, daha sonra

### Graphics2D g2 = (Graphics2D)g;

Kullanilarak adres Graphics2D sinifindan nesneye yüklenmektedir.

### RENK KONTROLÜ

Appletlerde renk controlu **Color** sinifini kullanarak yapilir. Renk olusumunda monitör ve Tv de kullanilan RGB (Red-Green-Blue=Kirmizi-Yesil-Mavi) sistemi temel alinir. Bu üç renk monitorde 0-255 arasi degerler alabilir. Üç rengin karisimi toplam 256\*256\*256 renk tanimlar.

Color sinifi kurucu metotlari su sekilden tanimlanir.

**public Color( int kirmizi,int yesil,int mavi)** //her renk 0-255 arasi **public Color(float kirmizi,float yesil,float mavi)** //her renk 0.0-1.0 arasi Color sinifinda ayrica renk degerlerini okuyabildigimiz

**public int getRed() //** Kirmizi tonunu oku **public int getGreen()//**Yesil tonunu oku **public int getBlue()//**Mavi tonunu oku **public abstract Color getColor() //**rengi oku metotlari ve rengi degistirebildigimiz

### public abstract Color setColor(Color c)

metotu mevcuttur. Color sinifinda sabit olarak tanimlanmis :

### Tablo 6.1 Color sinifinda tanimlanmis sabir renkler:

|  |  |  |
| --- | --- | --- |
| **Renk sabiti** | **Renk** | **RGB degeri** |
| Public final static Color orange | **portakal** | **255,200,0** |
| **Public final static Color pink** | **Pembe** | **255,175,175** |
| **public final static Color cyan** | **camgöbegi** | **0,255,255** |
| **public final static Color magenta** | **Mor** | **255,0,255** |
| **public final static Color yellow** | **Sari** | **255,255,0** |
| **public final static Color black** | **Siyah** | **0,0,0** |
| **public final static Color white** | **Beyaz** | **255,255,255** |
| **public final static Color gray** | **Gri** | **128,128,128** |
| **public final static Color lightGray** | **Açik gri** | **192,192,192** |
| **public final static Color darkGray** | **Koyu Gri** | **64,64,64** |
| **public final static Color red** | **kirmizi** | **255,0,0** |
| **public final static Color green** | **Yesil** | **0,255,0** |
| **public final static Color blue** | **mavi** | **0,0,255** |

Color sinifinin daha iyi anlasilabilmesi amaciyla Color sinifinin bir alt sinifi olan renk sinifini tanimladik.

### Program 6.3 : Color sinifinin alt sinifi renk sinifini tanimlayan renk.java programi :

import java.io.\*; import java.awt.\*;

public class renk extends Color

{

public final static renk kirmizi=new renk(255,0,0); public final static renk mavi=new renk(0,0,255); public final static renk siyah=new renk(0,0,0);

public final static renk camgobegi=new renk(0,255,255); public final static renk koyugri=new renk(64,64,64); public final static renk gri=new renk(128,138,128); public final static renk yesil=new renk(0,255,0);

public final static renk acikgri=new renk(192,192,192); public final static renk mor=new renk(255,0,255); public final static renk portakal=new renk(255,200,0); public final static renk pembe=new renk(255,175,175); public final static renk beyaz=new renk(255,255,255); public final static renk sari=new renk(255,255,0);

public final static renk acikmavi=new renk(150,206,237); public final static renk lacivert=new renk(0,0,128);

//burada kendi yeni renginizi tanimlayabilirsiniz.

public renk(float kirmizi,float yesil, float mavi)

{

super(kirmizi,yesil,mavi);

}

public renk(double kirmizi,double yesil, double mavi)

{

super((float)kirmizi,(float)yesil,(float)mavi);

}

public renk(int kirmizi,int yesil, int mavi)

{

super(kirmizi,yesil,mavi);

}

public renk(int RGB)

{

super(RGB);

}

public renk(renk r)

{

super(r.getKirmizi(),r.getYesil(),r.getMavi());

}

public int getKirmizi()

{

return getRed();

}

public int getYesil()

{

return getGreen();

}

public int getMavi()

{

return getBlue();

}

public int getKYM()

{

return getRGB();

}

public Color toColor()

{

return (Color)this;

}

public static renk toRenk(Color r)

{return (renk)r;}

public String toString()

{

return "renk[ kirmizi "+getKirmizi()+" mavi "+getMavi()+" yesil "+getYesil()+"]";

}

//metodlar

//static int HSBtoRGB(float hue,float saturation,float brightness)

//getRed(),getGreen(),getBlue(),getRGB()

}

Renk sinifini bir çok yerde Color sinifi yerine kullanabiliriz. Kullanamadigimiz durumlarda cast islemcisi (Color) kullanilabilir.

Applette renk degistirdigimiz küçük bir örnek program yazalim :

### Program 6.4 : renkTest.java

import java.awt.\*;

import java.applet.Applet; import renk;

public class RenkTest extends Applet

{

private int kirmizi,yesil,mavi;

public void init()

{

kirmizi=150; yesil=255; mavi=125;

// arka palanin rengi mavi olarak veriliyor setBackground(renk.mavi);

}

public void paint(Graphics g)

{

// yazýnýn rengi 100,255,125 olarak alýndý

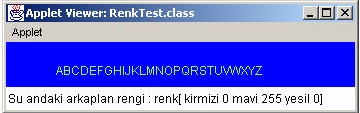
g.setColor(new renk(kirmizi,yesil,mavi)); g.drawString("ABCDEFGHIJKLMNOPQRSTUVWXYZ",50,33);

showStatus("Su andaki arkaplan rengi : "+getBackground());

}

}

### 06004.JPG



**Sekil 6.4 Renk.html çiktisi**

Bu programda kirmizi degeri 100, yesil degeri 255 ve mavi degeri de 125 alinmis ve setColor deyimiyle yeni renk tanimlanmistir. Arka plan rengi ise **public void setBackground(Color c)**

Metotunu kullanarak maviye dönüstürülmüstür.

Ayni programin JApplet esidi olan, ayni zamanda java komutuyla kendi çerçevesini de olusturabilen bir esdegerini verelim. Renk2D sinifinda Graphics2D sinifi kullanilmaktadir. Grafik kalitesini karsilastirabilirsiniz.

### Program 6.5 : Renk2D.java programi

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class Renk2D extends JApplet { private int kirmizi,yesil,mavi;

public void init()

{

kirmizi=100; yesil=255; mavi=125;

setBackground(new Color(0,0,255)); setForeground(new Color(kirmizi,yesil,mavi));

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D) g;

g2.setPaint(new Color(kirmizi,yesil,mavi)); g2.drawString("ABCDEFGHIJKLMNOPQRSTUVWXYZ",50,33);

g2.drawString("Su andaki renk : "+g2.getColor(),50,53);

}

public static void main(String s[]) { JFrame f = new JFrame("Renk2D");

f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

JApplet applet = new Renk2D(); f.getContentPane().add("Center", applet); applet.init();

f.pack();

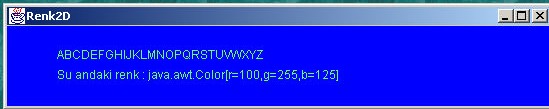
f.setSize(new Dimension(550,100)); f.show();

}

}

Renk2D sinifinin çiktisi :

### 06005.JPG



**Sekil 6.4 Renk2D.java çiktisi**

Swing sinifinda renk seçimini yapabilmek amaciyla JColorChooser adli bir program tanimlanmistir. Bu programi kullanarak renk grafik kutusunun rengini degistirebildigimiz renkSeciciSWF swing JFrame programi asagida verilmistir.

### Program 6.6 : renkSeciciSWF\_2000.java

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import BasicWindowMonitor;

public class renkSeciciSWF\_2000 extends JFrame implements ActionListener

{

// Renk secme ornegi JButton renkdegistir; Color r=Color.lightGray; Container c;

// pencereyi baslatma metodu public renkSeciciSWF\_2000()

{

super("JColorChoser "); c=getContentPane(); c.setLayout(new FlowLayout()) ;

renkdegistir=new JButton("renk degistirmek için dügmeye basiniz"); renkdegistir.addActionListener(this);

c.add(renkdegistir);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{ r=JColorChooser.showDialog(this,"Renk seçiniz :",r); if(r==null)

r=renk.acikgri; c.setBackground(r); repaint();

}

//=====================================================

public static void main(String[] args)

{

//ana program

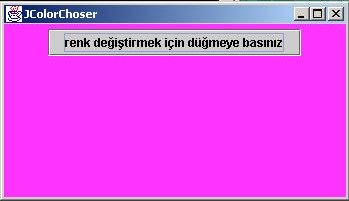
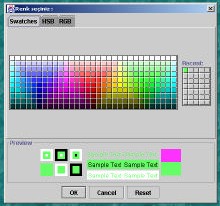
renkSeciciSWF\_2000 pencere= new renkSeciciSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

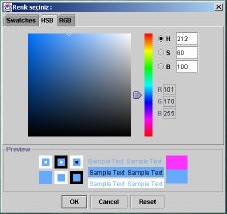
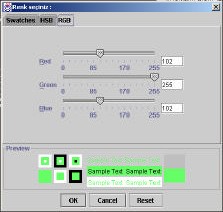
}

}

### 06006.JPG 06007.JPG

### 06008.JPG 06009.JPG

### Sekil 6.6-6.9 renkSeciciSWF\_2000.java java swing frame programi ve JcolorChooser renk seçme programi

Renk seciciSWF\_2000.java programinda temel olarak JFrame grafik çiktisi kullanilmistir. Bu programin içerisinde çagirilan

### r=JColorChooser.showDialog(this,"Renk seçiniz :",r);

metodu java kütüphanesinde yer almaktadir. ShowDialog metodu bize Color sinifindaki renk degiskenini aktarmakta ve

### c.setBackground(r);

komutuylada arka plan renklerini degistirmektedir.

JFrame ve JApplet arasindaki farki anlamaya biraz daha katkida bulunmasi amaciyla ayni programi bir de JApplet olarak verelim :

### Program 6.7 : renkSeciciSWA\_2000.java

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import BasicWindowMonitor;

public class renkSeciciSWA\_2000 extends JApplet implements ActionListener

{

// Renk secme ornegi JButton renkdegistir; Color r=Color.lightGray; Container c;

// pencereyi baslatma metodu public void init()

{

c=getContentPane();

c.setLayout(new FlowLayout()) ;

renkdegistir=new JButton("renk degistirmek için dügmeye basiniz"); renkdegistir.addActionListener(this);

c.add(renkdegistir);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

r=JColorChooser.showDialog(this,"Renk seçiniz :",r); if(r==null)

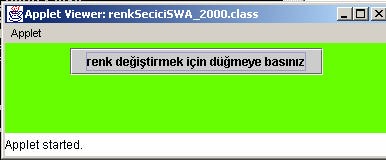
r=renk.acikgri; c.setBackground(r); repaint();

}

}

JFrame ve JApplet programlarinin en büyük farki JFrame ana programdan (main) çalistigi halde JApplet’in bir browser programindan çalistiriliyor olmasidir. Burada da en dis çerçeve disinda JFrame ile ayni görüntüyü elde ederiz.

### 06010.JPG



**Sekil 6.10 renkSeciciSWA\_2000.java programinin Appletviewer’daki çiktisi**

### YAZI KONTROLU

Appletlerde yazi kontrolu **Font** sinifini kullanarak yapilir. Sekil 5.3.1 de java 1.1.5 de kullanilabilecek tüm yazi stillerinin (font) listesini veren stil.java programini görüyoruz.

### Program 6.8 : stil.java programi, java kütüphanesinde mevcut olan yazi sitillerini gösterir

import java.applet.\*; import java.awt.\*;

public class stil extends Applet

{

public void paint(Graphics g)

{

String stilListesi[]; stilListesi=getToolkit().getFontList(); int y=15;

for(int i=0;i<stilListesi.length;i++)

{

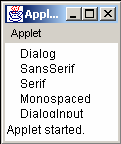
g.drawString(stilListesi[i],15,y); y+=15;

}

}

}

### 06011.JPG



**Sekil 6.11 stil.java programi, java kütüphanesinde mevcut olan yazi sitillerini gösterir applet programinin sonuçlari.**

Yine ayni programin swing JApplet esitini verelim :

### Program 6.9 : stil2D.java programi, java kütüphanesinde mevcut olan yazi sitillerini gösterir

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class stil2D extends JApplet { public void init()

{

setBackground(Color.white); setForeground(Color.black);

}

public void paint(Graphics g)

{

Graphics2D g2 = (Graphics2D) g; String stilListesi[]; stilListesi=getToolkit().getFontList(); int y=15;

for(int i=0;i<stilListesi.length;i++)

{

g2.drawString(stilListesi[i],15,y); y+=15;

}

}

public static void main(String s[]) { JFrame f = new JFrame("stil2D");

f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

JApplet applet = new stil2D(); f.getContentPane().add("Center", applet); applet.init();

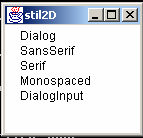
f.pack();

f.setSize(new Dimension(550,300)); f.show();

}

}

### 06012.JPG



**Sekil 6.12 : stil2D.java program çiktisi**

Bu yazi sitillerinden birisi asagida verilen yazi tiplerinden birisiyle beraber ve verilen belli bir yazi boyutunda gösterilebilir.

**public final static int PLAIN :** düz yazi sitili **public final static int BOLD** : Kalin yazi sitili **public static int ITALIC :** Italik (yatik) yazi sitili

bu tanimlardaki **final** sözcügü bu ifadelerin degisken degil **sabit** oldugunu gösterir.Font sinifinin kurucu metotu su sekildedir.

### Public Font( String s, int yazi\_sitili, int harf\_boyutu);

Bir appletin yazi sitili Graphics sinifindaki setFont metotuyla degistirilebilir.

### Public abstract void setFont(Font f)

Örnek olarak asagidaki deyimi verebiliriz :

### g.setFont(new Font(“Serif”,Font.BOLD,12);

Simdi setFont metotunu küçük bir örnekte kullanarak gösterelim :

### Program 6.10 : RenkFont.java programi, java kütüphanesindeki Font uygulamasini gösterir

import java.awt.\*;

import java.applet.Applet;

public class RenkFont extends Applet

{

private int kirmizi,yesil,mavi; public void init()

{

kirmizi=100; yesil=255; mavi=125;

}

public void paint(Graphics g)

{

g.setColor(new Color(kirmizi,yesil,mavi)); setBackground(new Color(0,0,255));

g.setFont(new Font("TimesRoman",Font.BOLD,36)); g.drawString("ABCDEFGHIJKLMNOPQRSTUVWXYZ",50,33);

showStatus("Su andaki font : "+g.getFont());

}

}

### 06013.JPG



**Sekil 6.13 RenkFont.html programi, java kütüphanesindeki Font uygulamasini gösterir**

bu programin da Japplet esdegerini verelim:

### Program 6.11 : RenkFont2D.java swing JApplet programi, java kütüphanesindeki Font uygulamasini gösterir

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class RenkFont2D extends JApplet { private int kirmizi,yesil,mavi;

public void init()

{

kirmizi=100; yesil=255; mavi=125;

setBackground(new Color(0,0,255)); setForeground(new Color(kirmizi,yesil,mavi));

}

public void paint(Graphics g)

{

Graphics2D g2 = (Graphics2D) g; g2.setColor(new Color(kirmizi,yesil,mavi)); setBackground(new Color(0,0,255));

Font font=g2.getFont();

g2.setFont(new Font(font.getName(),Font.BOLD,24)); g2.drawString("ABCDEFGHIJKLMNOPQRSTUVWXYZ",50,33);

g2.drawString("Su andaki font : "+g.getFont(),50,70); g2.setFont(new Font(font.getName(),Font.PLAIN,12)); g2.drawString("Su andaki font : "+g.getFont(),50,85);

}

public static void main(String s[]) { JFrame f = new JFrame("RenkFont2D");

f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

JApplet applet = new RenkFont2D(); f.getContentPane().add("Center", applet); applet.init();

f.pack();

f.setSize(new Dimension(550,200)); f.show();

}

}

### 06014.JPG



**Sekil 6.14 RenkFont2D.java JFrame programi, java kütüphanesindeki Font uygulamasini gösterir**

### ÇiZGi ÇiZiMi

Javada çizgi çizmek için kullanilan temel metot Graphics sinifinda **drawLine** metotudur. Bu metot Graphics sinifinda tanimlanmistir. Kurucu metotu :

### Public abstract void drawLine( int x1, // ilk noktanin x koordinati int y1, //ilk noktanin y koordinati

**int x2, // ikinci noktanin x koordinati int y2) //ikinci noktanin y koordinati**

Graphics2D sinifinda ise çizgi çiziminde java.awt.geom kütüphanesinde tanimlanan Line2D sinifi, Graphics2D sinifindaki draw metoduyla birlikte kullanilir.

Bir küçük örnek problem **cizgiciz.java'**da drawLine metotunun kullanimini görelim:

### Program 6.12 : Çizgi çizim örnegi cizgiciz.java

import java.applet.\*; import java.applet.Applet; import java.awt.\*;

public class cizgiciz extends Applet

{

public void paint(Graphics g)

{

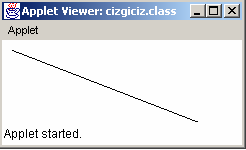
g.setColor(new Color(0,0,0));

g.setFont(new Font("SansSerif",Font.PLAIN,24)); g.drawLine(10,10,230,95);

}

}

### 06015.JPG



**Sekil 6.615 ve çizginin appletde görünümü**

Simdi ayni problemin yaklasik esidini cizgiSWF\_2000.java programinda görelim :

### Program 6.13 : Çizgi çizim örnegi cizgiSWF\_2000.java

import java.awt.\*; import java.awt.event.\*;

import java.awt.geom.\*; import javax.swing.\*; import renk;

public class cizgiSWF\_2000 extends JFrame { final static Color bg = Color.white;

final static Color fg = Color.black; final static Color red = Color.red; final static Color white = Color.white;

public cizgiSWF\_2000()

{

//Initialize drawing colors super("çizgi çizimi"); setBackground(bg); setForeground(fg);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.black);

Line2D l=new Line2D.Double(50,50,100,75); g2.draw(l);

g2.setStroke(new BasicStroke(1f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f));

Line2D l1=new Line2D.Double(50,150,100,175); g2.draw(l1);

}

//=====================================================

public static void main(String[] args)

{

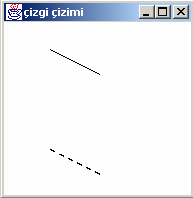
cizgiSWF\_2000 pencere= new cizgiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

}

### 06016.JPG



**Sekil 6.16 ve çizginin cizgiSWF\_2000 JFrame çiktisinda görünümü**

Bu programdaki temel çizim metodlarina göz atalim :

### Graphics2D g2 = (Graphics2D)g;

Graphics sinifi g nesnesini Graphics2D sinifi g2 nesnesine aktarir.

### g2.setPaint(Color.black);

Grafik çizim rengini siyah olarak tanimlar.

### Line2D l=new Line2D.Double(50,50,100,75);

Ilk devamli çizginin koordinatlarini tanimlar

### g2.draw(l);

Ilk devamli çizgiyi çizer

### g2.setStroke(new BasicStroke(1f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f));

Çizgiyi noktali çizme komutunu tanimlar

### Line2D l1=new Line2D.Double(50,150,100,175);

Noktali çizginin koordinatlarini tanimlar

### g2.draw(l1);

Noktali çizgiyi çizer.

Burada bizim için yeni bir kavram olan noktali çizgi çizmenin detaylarina biraz göz atalim. Noktaliçizgi olusturmak için temel olarak setStroke metodunu kullaniyoruz. Bu metodun içinde Basic Stroke metodu bir nesne yaratiyoruz. BasicStroke türü nesne çizgi kalinligi, çizgi baglama sitili,çizgi ucu bitis sitili ve çizgi-bosluk bilgisi içerir. Çizgi kalinligi çizgi istikametine dik olarak ölçülür. Çizgi kalinligi 1f (buradaki f sayinin float temel degisken türü oldugunu belitir) 0.35277 milimetreye esittir. Yalniz transformasyon kullanildiginda bu degisebilir. Transformasyonlara biraz sonra deginecegiz.

Çizgi baglama sitili iki çizginin birbirine baglandigi noktadaki seklini belirler. BasicStroke sinifinda 3 baglanma sitili mevcuttur. Bunlar :

JOIN\_BEVEL

JOIN\_MITER

JOIN\_ROUND

sitilleridir. Çizgi ucu bitis sitilleri bos çizgi ucunun seklini belirtir. Yine temel olarak 3 BasicStroke çizgi ucu bitis sitili mevcuttur. Bunlar :

CAP\_BUTT

CAP\_ROUND

CAP\_SQUARE

sitilleridir. Çizgi bosluk bilgisi birbirini takip eden seffaf ve görünür çizgi parçaciklari ve bunlarin boylarini tanimlar. Bir ara bosluk degiskeni ve çizgi bosluk çizgi bosluk olmak üzere bir seriden olusur. Son olarak son rakam serinin hangi elemanindan çizime baslanacagini gösterir Örnegin:

g2.setStroke(new BasicStroke(1,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,0,new float[] {10,5,2,5},0));

10 boyutu bir çizgi 5 bosluk 2 boyutu çizgi 5 bosluk belirtir. g2.setStroke(new BasicStroke(1,BasicStroke.CAP\_ROUND,

BasicStroke.JOIN\_ROUND,5,new float[] {5},0));

5 bosluk ve 5 çizgi belirtir. cizgiSWF\_2000.java programinda çesitli stoke tiplerini deneyebilirsiniz.

### DiKDÖRTGEN ÇIZIMI

Java Graphics sinifinda dikdörtgen çizmek için kullanilan temel metot **drawRect** metotudur. Bu metot Graphics sinifinda tanimlanmistir. Bu metotlarin tanimi :

### Public abstract void drawRect(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi**

**int yukseklik) // dikdörtgenin yüksekligi**

Bu metota ilave olarak dikdörtgenin içini verilen renkte boyayan

### Public abstract void fillRect(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi**

**int yukseklik) // dikdörtgenin yüksekligi**

ve dikdörtgeni arkaplan renginde boyayan (silen)

### Public abstract void clearRect(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi**

**int yukseklik) // dikdörtgenin yüksekligi**

metotlari mevcuttur. DrawRect ve fillRec metotunun kullanimi Program 6.6 da verilmistir.

### Program 6.14 : dikdortgenCiz.java

import java.applet.Applet; import java.awt.\*;

public class dikdortgenCiz extends Applet

{

public void paint(Graphics g)

{

g.setColor(new Color(255,0,0)); g.drawRect(10,10,100,50); g.fillRect(129,10,100,50);

}

}

### 06017.JPG



**Sekil 6.17 dikdortgenCiz.java programi ve dikdortgenCiz.html appleti**

Graphics2D metodunda dikdörtgen çizmek için

### g2.draw(new Rectangle2D.Double(x, y,dikdortgenEn,dikdortgenYukseklik));

kullanilir.

Dikdörtgenin içini maviye boyamak için

### g2.setPaint(Color.blue);

**Rectangle2D dikdortgen2=new Rectangle2D.Double(50,50,100,40); g2.fill(dikdortgen2);**

kullanilabilir. Graphics2D de tek renk yerine degisen bir renk profiliyle boyamak da mümkündür. Örnegin :

### final static Color beyaz=Color.white; final static Color siyah=Color.black;

**GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);**

**Rectangle2D dikdortgen4=new Rectangle2D.Double(250,50,100,40); g2.fill(dikdortgen4);**

kodu dikdörtgeni kirmizidan beyaza dönüsen bir spekturumla boyar. Tüm bu kodu bir örnek problemde gösterelim :

### Program 6.15 : dikdortgenSWF-2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class dikdortgenSWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white; final static Color siyah=Color.black;

public dikdortgenSWF\_2000()

{

super("Dikdörtgen çizimi"); setBackground(Color.lightGray); setForeground(siyah);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.blue);

Rectangle2D dikdortgen1=new Rectangle2D.Double(55,55,90,30); g2.fill(dikdortgen1);

g2.setPaint(siyah);

Rectangle2D dikdortgen2=new Rectangle2D.Double(50,50,100,40); g2.draw(dikdortgen2);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,3f,new float[] {10f},0f)); g2.setPaint(Color.black);

Rectangle2D dikdortgen3=new Rectangle2D.Double(50,150,100,40); g2.draw(dikdortgen3);

GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);

Rectangle2D dikdortgen4=new Rectangle2D.Double(250,50,100,40); g2.fill(dikdortgen4);

GradientPaint kirmizidanmaviye=new GradientPaint(250,150,kirmizi,350,190,Color.blue); g2.setPaint(kirmizidanmaviye);

Rectangle2D dikdortgen5=new Rectangle2D.Double(250,150,100,40); g2.fill(dikdortgen5);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f)); g2.setPaint(Color.black);

g2.draw(dikdortgen5);

}

//=====================================================

public static void main(String[] args)

{

dikdortgenSWF\_2000 pencere= new dikdortgenSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

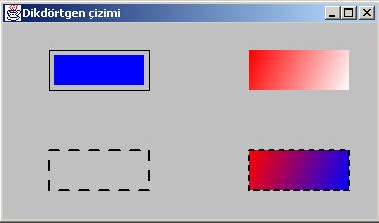
pencere.setVisible(true);

}

}

Bu programin çiktisi :

### 06018.JPG



**Sekil 6.18 çesitli dikdörtgen çizimlerinin dikdortgenSWF\_2000 JFrame çiktisinda görünümü**

Graphics sinifinda **Köseleri yuvarlatilmis dikdörtgen** çizmek içinse **drawRoundRect** metotu kullanilir

### Public abstract void drawRoundRect( int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi

**int yukseklik, // dikdörtgenin yüksekligi int koseeni, // yuvarlak kosenin eni**

**int koseyuksekligi )// yuvarlak kosenin yuksekligi**

Bu metota ilave olarak dikdörtgenin içini verilen renkte boyayan

### Public abstract void fillRoundRect(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi**

**int yukseklik, // dikdörtgenin yüksekligi int koseeni, // yuvarlak kosenin eni**

**int koseyuksekligi )// yuvarlak kosenin yuksekligi**

metotu da paralel olarak tanimlanmistir. Yine bir örnek problemde bu tanimi kullanacak olursak :

### Program 6.16 : yuvarlakKoseliDikdortgenCiz.java programi

import java.applet.Applet; import java.awt.\*;

public class yuvarlakKoseliDikdortgenCiz extends Applet

{

public void paint(Graphics g)

{

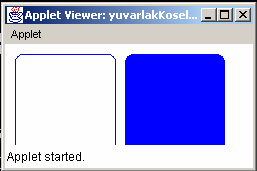
g.setColor(new Color(0,0,255));

g.setFont(new Font("SansSerif",Font.PLAIN,12)); g.drawRoundRect(10,10,100,100,15,15); g.fillRoundRect(120,10,100,100,15,15);

}

}

### 06019.JPG



**Sekil 6.19 yuvarlakKoseliDikdortgenCiz.html appleti**

Graphics2D sinifinda koseleri yuvarlatilmis dikdörtgen çizmek için : **g2.draw(new Rectangle2D.Double(x, y,dikdortgenEn,dikdortgenYukseklik,xKöseYariçapi,yKöseYariçapi));** metodlari kullanilir. Rakamsal örnek verirsek :

### g2.setPaint(Color.blue);

**RoundRectangle2D dikdortgen2=new RoundRectangle2D.Double(50,50,100,40,15,15); g2.draw(dikdortgen2);**

kullanilabilir. Dikdörtgenin içini boyamak içinse:

### g2.setPaint(Color.blue);

**RoundRectangle2D dikdortgen2=new RoundRectangle2D.Double(50,50,100,40,15,15); g2.fill(dikdortgen2);**

kodu kullanilabilir. Elbette bir önceki örnekte oldugu gibi spektrumlu boyama da mümkündür. Tüm bu kodu bir örnek problemde gösterelim

### Program 6.17 : yuvarlakKoseliDikdortgenSWF\_2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class yuvarlakKoseliDikdortgenSWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white;

final static Color siyah=Color.black;

public yuvarlakKoseliDikdortgenSWF\_2000()

{

super("Yuvarlak Köseli Dikdörtgen Çizimi"); setBackground(Color.lightGray); setForeground(siyah);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.blue);

RoundRectangle2D dikdortgen1=new RoundRectangle2D.Double(55,55,90,30,15,15);

g2.fill(dikdortgen1); g2.setPaint(siyah);

RoundRectangle2D dikdortgen2=new RoundRectangle2D.Double(50,50,100,40,15,15); g2.draw(dikdortgen2);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,3f,new float[] {10f},0f)); g2.setPaint(Color.black);

RoundRectangle2D dikdortgen3=new RoundRectangle2D.Double(50,150,100,40,15,15); g2.draw(dikdortgen3);

GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);

RoundRectangle2D dikdortgen4=new RoundRectangle2D.Double(250,50,100,40,15,15); g2.fill(dikdortgen4);

GradientPaint kirmizidanmaviye=new GradientPaint(250,150,kirmizi,350,190,Color.blue); g2.setPaint(kirmizidanmaviye);

RoundRectangle2D dikdortgen5=new RoundRectangle2D.Double(250,150,100,40,15,15); g2.fill(dikdortgen5);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f)); g2.setPaint(Color.black);

g2.draw(dikdortgen5);

}

//=====================================================

public static void main(String[] args)

{

yuvarlakKoseliDikdortgenSWF\_2000 pencere= new yuvarlakKoseliDikdortgenSWF\_2000();

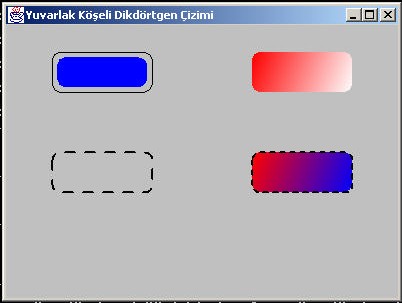
pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06020.JPG



**Sekil 6.20 çesitli kösesi yuvarlatilmis dikdörtgen çizimlerinin yuvarlakKoselidikdortgenSWF\_2000 JFrame çiktisinda görünümü**

Graphics’de 3 boyutlu dikdortgen çizen **draw3DRect** ve **fill3Drect** metotlari mevcutttur. Bu metotlarin tanimlari su sekilde verilmistir.

### Public abstract void draw3DRect(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi**

**int yukseklik, // dikdörtgenin yüksekligi**

**boolean yukarikaldirma) // yukari kalkik 3üncü boyut**

**Public abstract void fill3DRect(**

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // dikdörgenin genisligi**

**int yukseklik, // dikdörtgenin yüksekligi**

**boolean yukarikaldirma) // yukari kalkik 3üncü boyut Program 61.8 : ucBoyutluDikdortgenCiz.java programi**

import java.applet.Applet; import java.awt.\*;

public class ucBoyutluDikdortgenCiz extends Applet

{

public void paint(Graphics g)

{

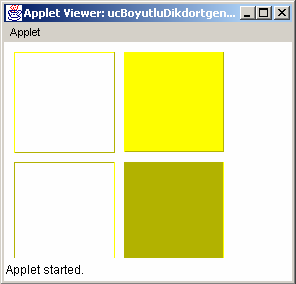
g.setColor(Color.yellow);

g.setFont(new Font("SansSerif",Font.PLAIN,12)); g.draw3DRect(10,10,100,100,true); g.fill3DRect(120,10,100,100,true); g.draw3DRect(10,120,100,100,false); g.fill3DRect(120,120,100,100,false);

}

}

### 06021.JPG



**Sekil 6.21 ucBoyutluDikdortgenCiz.html appleti**

### ÇIZILEN SEKILLERIN DEGISTIRILEREK ÇIZIMI (TRANSFORM)

Graphics2D sinifinda çizilen sekiller dinamik olarak degistirilebilirler. Bu degisimler döndürme, kösedençekip uzatma gibi degisik islemler olabilir. Bu degistirme prosesini önce bir örnekle gösterelim, sonrada izah etmeye çalisalim.

### Program 6.19 : donmusDikdortgenSWF-2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class donmusDikdortgenSWF\_2000 extends JFrame {

final static Color kirmizi=Color.red; final static Color beyaz=Color.white; final static Color siyah=Color.black;

public donmusDikdortgenSWF\_2000()

{

super("Dödürülmüs - çekilmis Dikdörtgen çizimleri"); setBackground(Color.lightGray); setForeground(siyah);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.blue);

Rectangle2D dikdortgen1=new Rectangle2D.Double(55,55,90,30); AffineTransform at= AffineTransform.getRotateInstance(45,50,50); g2.setTransform(at);

g2.fill(dikdortgen1); g2.setPaint(siyah);

Rectangle2D dikdortgen2=new Rectangle2D.Double(50,50,100,40); g2.draw(dikdortgen2);

g2.setTransform(new AffineTransform());

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,3f,new float[] {10f},0f)); g2.setPaint(Color.black);

Rectangle2D dikdortgen3=new Rectangle2D.Double(50,150,100,40); AffineTransform at1= AffineTransform.getScaleInstance(1.5,1.5); g2.setTransform(at1);

g2.draw(dikdortgen3); g2.setTransform(new AffineTransform());

GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);

Rectangle2D dikdortgen4=new Rectangle2D.Double(250,50,100,40); AffineTransform at2= AffineTransform.getShearInstance(0.4,0); g2.setTransform(at2);

g2.fill(dikdortgen4); g2.setTransform(new AffineTransform()); GradientPaint kirmizidanmaviye=

new GradientPaint(250,150,kirmizi,350,190,Color.blue); g2.setPaint(kirmizidanmaviye);

Rectangle2D dikdortgen5=new Rectangle2D.Double(250,150,100,40); g2.fill(dikdortgen5);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f)); g2.setPaint(Color.black);

AffineTransform at3= AffineTransform.getTranslateInstance(0.0,10.0); g2.setTransform(at3);

g2.draw(dikdortgen5); g2.setTransform(new AffineTransform());

}

//=====================================================

public static void main(String[] args)

{

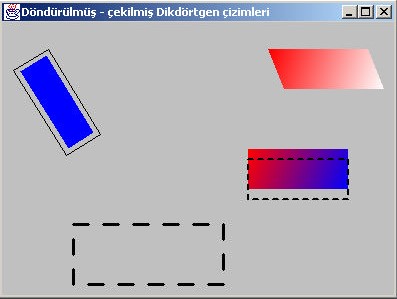
donmusDikdortgenSWF\_2000 pencere= new donmusDikdortgenSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06022.JPG



**Sekil 6.22 çesitli sekli degistirilmis dikdörtgen çizimlerinin donmusDikdortgenSWF-2000 JFrame çiktisinda görünümü**

**AffineTransform at= AffineTransform.getRotateInstance(x,y,açiDerece);**

Sekli x,y noktasi etrafinda açiDerece kadar saat akraninin tersi yönde döndürür.

### AffineTransform at2= AffineTransform.getShearInstance(xÇekisi,yÇekisi);

Sekli x ekseninde xÇelisi kadar, y ekseninde yÇekisi kadar uzatir (orijinal sekilden elastik ve üst kösesi tutturulmus sekilde)

### AffineTransform at1= AffineTransform.getScaleInstance(xBoyutFaktörü,yBoyutFaktörü);

Seklin boyutunu **xBoyutFaktörü,yBoyutFaktörü** faktörleriyle çarparak büyütür.

### AffineTransform at3= AffineTransform.getTranslateInstance(xTasimaFaktörü,yTasimaFaktörü);

Sekli yerinden hareket ettirir. Orijinal transforma geri dönmek için :

### g2.setTransform(new AffineTransform());

metodu çagirilabilir.

### OVAL VE AÇILI OVAL ÇIZIMI

Graphics sinifiOval çizimi için drawOval ve fillOval metotlari kullanilir bu metotlarin tanimi söyledir :

### Public abstract void drawOval(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // ovalin genisligi**

**int yukseklik) // ovalin yüksekligi**

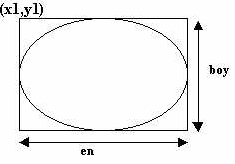
**Public abstract void fillOval(**

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // ovalin genisligi**

**int yukseklik) // ovalin yüksekligi**

(x1,y1) koordinatinin ovalle iliskisini daha iyi anlayabilmek için asagidaki grafigi verebiliriz :

### 06023.JPG



**Sekil 6.23 Oval tanimi koordinat sistemi**

Oval programini küçük bir örnek programda kullanalim :

### Program 6.20 : daireCiz.java programi

import java.applet.Applet; import java.awt.\*;

public class daireCiz extends Applet

{

public void paint(Graphics g)

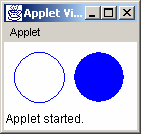
{

g.setColor(new Color(0,0,255)); g.drawOval(10,10,50,50); g.fillOval(70,10,50,50);

}

}

### 06024.JPG



**Sekil 6.24 daireCiz.html apleti**

Graphics ile bir ovalin tamamini degilde sadece bir bölümünü çizmek dilenilirse, **drawArc ve fillArc** metotlari kullanilabilir. Bu metotlarin tanimi :

### Public abstract void drawArc(

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // ovalin genisligi**

**int yukseklik, // ovalin yüksekligi**

**int baslangiçaçisi, //arkin baslangiç açisi, derece int çizimaçisi) // arkin çizim açisi, derece**

**Public abstract void fillArc(**

**int x1, // üst tepe noktanin x koordinati int y1, // üst tepe noktanin y koordinati int en, // ovalin genisligi**

**int yukseklik, // ovalin yüksekligi**

**int baslangiçaçisi, //arkin baslangiç açisi, derece int çizimaçisi) // arkin çizim açisi, derece**

Simdi de bu metotu kullanan bir bilgisayar programi yazalim :

### Program 6.21 : arcCiz.java, arc çizimi programi

import java.applet.Applet; import java.awt.\*;

public class arcCiz extends Applet

{

public void paint(Graphics g)

{

g.setColor(new Color(0,0,255));

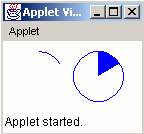
g.setFont(new Font("SansSerif",Font.PLAIN,12)); g.drawOval(70,10,50,50); g.drawArc(10,10,50,50,30,60);

g.fillArc(70,10,50,50,30,60);

}

}

### 06025.JPG



**Sekil 6.25 arcCiz.html apleti**

Graphics2D sinifinda oval (elips) çizmek için : **Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.blue);**

### Ellipse2D elips1=new Ellipse2D.Double(x,y,en,yükseklik); g2.draw(elips1);

kullanabiliriz. Içini boyamak için ise :

### g2.fill(elips1);

metodunu kullaniriz.

Yine örnegimizde kullanacak olursak :

### Program 6.21 : elipsSWF\_2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class elipsSWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white; final static Color siyah=Color.black;

public elipsSWF\_2000()

{

super("Elips çizimi"); setBackground(Color.lightGray); setForeground(siyah);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.blue);

Ellipse2D elips1=new Ellipse2D.Double(55,55,90,30); g2.fill(elips1);

g2.setPaint(siyah);

Ellipse2D elips2=new Ellipse2D.Double(50,50,100,40); g2.draw(elips2);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,3f,new float[] {10f},0f)); g2.setPaint(Color.black);

Ellipse2D elips3=new Ellipse2D.Double(50,150,100,40); g2.draw(elips3);

GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);

Ellipse2D elips4=new Ellipse2D.Double(250,50,100,40); g2.fill(elips4);

GradientPaint kirmizidanmaviye=new GradientPaint(250,150,kirmizi,350,190,Color.blue); g2.setPaint(kirmizidanmaviye);

Ellipse2D elips5=new Ellipse2D.Double(250,150,100,40); g2.fill(elips5);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f)); g2.setPaint(Color.black);

g2.draw(elips5);

}

//=====================================================

public static void main(String[] args)

{

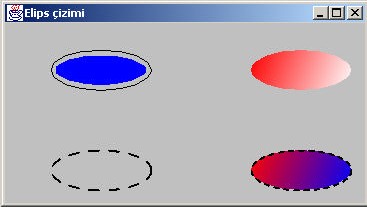
elipsSWF\_2000 pencere= new elipsSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06026.JPG



**Sekil 6.26 çesitli oval (elips) çizimlerinin elipsSWF\_2000JFrame çiktisinda görünümü**

### ÇIZIMIN SEÇILEN BIR RESIMLE DOLDURULMASI

Graphics2D sinifinda bir grafigi boyarken bir resim dosyasinin içinde tanimlanan bir motifi de kullanabiliriz.

### Program 6.22 :textureElipsSWF\_2000.java

import javax.swing.\*; import java.awt.geom.\*;

import java.awt.image.\*; import java.net.URL; import java.awt.\*;

public class textureElipsSWF\_2000 extends JFrame {

TexturePaint tp = getImageTexture("sybex.gif"); final static Color kirmizi=Color.red;

final static Color beyaz=Color.white; final static Color siyah=Color.black;

public textureElipsSWF\_2000()

{

super("Elips çizimi"); setBackground(beyaz); setForeground(siyah);

}

public TexturePaint getImageTexture(String imageFile)

{

URL url = getClass().getResource(imageFile); Image img = getToolkit().getImage(url);

try {

MediaTracker tracker = new MediaTracker(this); tracker.addImage(img, 0);

tracker.waitForID(0);

} catch (Exception e) {}

int width = img.getWidth(this); int height = img.getHeight(this);

BufferedImage buffImg = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_ARGB); Graphics g = buffImg.getGraphics();

g.drawImage(img, 0, 0, this);

Rectangle2D rect = new Rectangle(0, 0, width, height); return new TexturePaint(buffImg, rect);

}

//\*\*\*\*\*\*\*\*\*\*\*\*

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(tp);

Ellipse2D elips1=new Ellipse2D.Double(55,55,90,30); g2.fill(elips1);

g2.setPaint(siyah);

Ellipse2D elips2=new Ellipse2D.Double(50,50,100,40); g2.draw(elips2);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,3f,new float[] {10f},0f)); g2.setPaint(Color.black);

Ellipse2D elips3=new Ellipse2D.Double(50,150,100,40); g2.draw(elips3);

GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);

Ellipse2D elips4=new Ellipse2D.Double(250,50,100,40); g2.fill(elips4);

g2.setPaint(tp);

Ellipse2D elips5=new Ellipse2D.Double(250,150,100,40); g2.fill(elips5);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f)); g2.setPaint(Color.black);

g2.draw(elips5);

}

//=====================================================

public static void main(String[] args)

{

textureElipsSWF\_2000 pencere= new textureElipsSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(300,300);

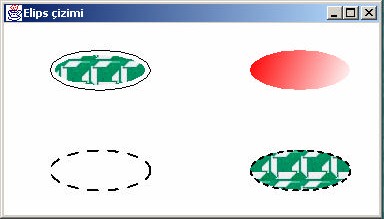
pencere.setVisible(true);

}

//\*\*\*\*\*\*\*\*\*\*\*\*

}

### 06027.JPG



**Sekil 6.27 çesitli oval (elips) çizimlerinin ve arkaplan resimli boyama isleminin textureElipsSWF\_2000JFrame çiktisinda görünümü**

Burada texture sybex.gif dosyasindan

### public TexturePaint getImageTexture(String imageFile)

metodu kullanilarak aktarilmis ve aynen yeni bir renk gibi çizim ortaminda kullanilmistir.

Simdi de Graphics2D metoduyla arc çizdirmeye bakalim arc yukarda da bahsettigimiz gibi tamamlanmamis bir elipstir.

### Program 6.23 : arcSWF\_2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class arcSWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white;

final static Color siyah=Color.black; public arcSWF\_2000()

{

super("arc çizimi"); setBackground(Color.lightGray); setForeground(siyah);

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g; g2.setPaint(Color.blue);

Arc2D arc1=new Arc2D.Double(55,55,90,30,90,270,Arc2D.PIE); g2.fill(arc1);

g2.setPaint(siyah);

Arc2D arc2=new Arc2D.Double(50,50,100,40,90,270,Arc2D.PIE);

g2.draw(arc2);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,3f,new float[] {10f},0f)); g2.setPaint(Color.black);

Arc2D arc3=new Arc2D.Double(50,150,100,40,90,270,Arc2D.PIE);

g2.draw(arc3);

GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,kirmizi,350,90,beyaz); g2.setPaint(kirmizidanbeyaza);

Arc2D arc4=new Arc2D.Double(250,50,100,40,90,270,Arc2D.PIE);

g2.fill(arc4);

GradientPaint kirmizidanmaviye=new GradientPaint(250,150,kirmizi,350,190,Color.blue); g2.setPaint(kirmizidanmaviye);

Arc2D arc5=new Arc2D.Double(250,150,100,40,90,270,Arc2D.PIE);

g2.fill(arc5);

g2.setStroke(new BasicStroke(2f,BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND,1f,new float[] {5f},0f)); g2.setPaint(Color.black);

g2.draw(arc5);

}

//=====================================================

public static void main(String[] args)

{

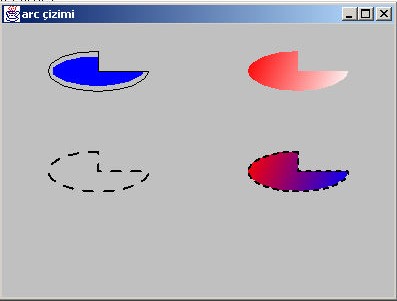
arcSWF\_2000 pencere= new arcSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06028.JPG



**Sekil 6.28 çesitli tamamlanmamis oval (ark) çizimlerinin arcSWF\_20000 JFrame çiktisinda görünümü**

### POLYGON ÇIZIMI

Graphics sinifinda Açik ve kapali polygon sekilleri çizmek için **drawPolygon** ve **fillPolygon** metotlari kullanilir. Metotlarin tanimi :

### Program 6.24 : drawPolygon.java

public abstract void drawPolygon( int x[], // x koordinat vektörü

int y[], // y koordinati vektörü

int nokta sayisi) // nokta sayisi

public abstract void drawPolyline( int x[], // x koordinat vektörü

int y[], // y koordinati vektörü int nokta sayisi) // nokta sayisi

public abstract void drawPolygon( Polygon p) //Graphics sinifi metotu public Polygon() //Polygon sinifi

public Polygon(

int x[], // x koordinat vektörü int y[], // y koordinati vektörü int nokta sayisi) // nokta sayisi

Metot tanimlarindan da görüldügü gibi polygon int türü vektörle çizilebilecegi gibi bu vektörü içinde barindiran Polygon sinifi bir nesne kullanilarak ta çizilebilir.

Örnek olarak üçgen çizen bir program verelim :

### Program 6.25 : ucgenCiz.java, poligon çizim programi

import java.applet.Applet; import java.awt.\*;

public class ucgenCiz extends Applet

{

public void paint(Graphics g)

{

int x[]={10,(int)(10+100\*Math.cos(Math.PI/3.0)),110,10};

int y[]={100,(int)(100-100\*Math.sin(Math.PI/3.0)),100,100}; int x1[]={130,(int)(130+100\*Math.cos(Math.PI/3.0)),230,130}; int y1[];

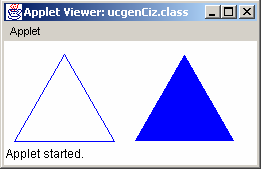
y1=y;

g.setColor(new Color(0,0,255)); g.drawPolygon(x,y,4); g.fillPolygon(x1,y1,4);

}

}

### 06029.JPG



**Sekil 6.29 ucgenCiz.java programi ve apleti**

Graphics2D programinda grafik çizdirmek için kullanilan metodlar Graphics sinifina göre oldukça degisiktir. Burada önce örnek programimizi verelim:

### Program 6.26 : polygonSWF.java

import java.awt.\*;

import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class polygonSWF extends JFrame

{

public polygonSWF()

{

super("Polygon çizimi");

}

public void paint(Graphics g)

{

int x[]={55,67,109,73,83,55,27,37,1,43};

int y[]={0,36,36,54,96,72,96,54,36,36};

Graphics2D g2=(Graphics2D)g; GeneralPath yildiz=new GeneralPath(); yildiz.moveTo(x[0],y[0]);

for(int i=1;i<x.length;i++)

{yildiz.lineTo(x[i],y[i]);} yildiz.closePath(); g2.translate(200,200); for(int i=0;i<20;i++)

{

g2.rotate(Math.PI/10.0);

g2.setColor(new Color((int)(Math.random()\*256),(int)(Math.random()\*256), (int)(Math.random()\*256)));

g2.fill(yildiz);

g2.setStroke(new BasicStroke(2)); g2.setColor(Color.black); g2.draw(yildiz);

}

}

public static void main(String[] args)

{

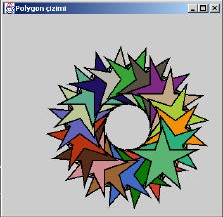
polygonSWF pencere= new polygonSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06030.JPG



**Sekil 6.30 çesitli yildizlarin (polygonun) çizim ve döndürme islemi polygonSWF JFrame çiktisinda görünümü**

bu programda önce

### int x[]={55,67,109,73,83,55,27,37,1,43};

**int y[]={0,36,36,54,96,72,96,54,36,36};**

**GeneralPath yildiz=new GeneralPath(); yildiz.moveTo(x[0],y[0]);**

**for(int i=1;i<x.length;i++)**

**{yildiz.lineTo(x[i],y[i]);} yildiz.closePath();**

ile yildiz tanimlanmis, ve

**g2.draw(yildiz);** ile çizilmis, ve **g2.fill(yildiz);**

ile içi boyanmistir. Ayrica

**g2.translate(200,200);** ile çizim baslama noktasi 200,200 noktasina kaydirilmis ve

### g2.rotate(Math.PI/10.0);

komutu ile döndürülmüstür. Programda setPAint kullanarak tesadüfi renkler seçilmis ve yildiz çizimi doldurulmustur.

Programda kullanilan

### GeneralPath yildiz=new GeneralPath();

Terimine özellikle dikkatinizi çekmek isteriz. Bu terim GeneralPath nesnesi yildizi tanimlar.

### GENELLESTIRILMIS EGRI ÇIZIMI

GeneralPath nesnesi kullanarak polinomdan çok daha kompleks çekiller de tanimlayabiliriz.

### Program 6.28 : generalPathSWF\_2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class generalPathSWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white;

final static Color siyah=Color.black;

public generalPathSWF\_2000()

{

super("ikinci ve üçüncü dereceden genel egri çizimi");

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON); setBackground(Color.lightGray);

setForeground(siyah); Dimension boyut=getSize(); int dx=boyut.width;

int dy=boyut.height; g2.setPaint(siyah); g2.setStroke(new BasicStroke(3)); g2.draw3DRect(0,0,dx-3,dy-3,true);

g2.draw3DRect(3,3,dx-7,dy-7,false);

GeneralPath sekil=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD); sekil.moveTo(20,20);

//quadratik ekleme sekil.quadTo(160,120,245,45);

//kübik ekleme sekil.curveTo(195,95,295,145,245,195); sekil.curveTo(-80,110,345,110,20,195); sekil.curveTo(400,250,200,250,20,20);

g2.draw(sekil);

}

//=====================================================

public static void main(String[] args)

{

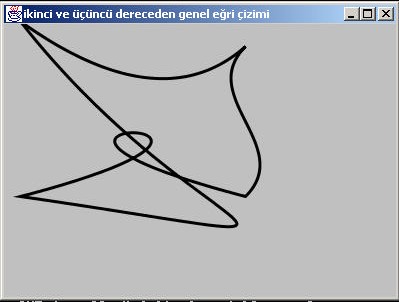
generalPathSWF\_2000 pencere= new generalPathSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06031.JPG



**Sekil 6.31 ikinci ve üçüncü dereceden genel egri çizimi, generalPathSWF\_2000, JFrame çiktisinda görünümü**

Bu çizimdeki çizginin oldukça düzgün olduguna dikkatinizi çekeriz. Bunu saglayan temel komut : **g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);**

Komutudur. Bu komut çizgileri daha düzgün hale getirir, fakat belli bir hesap ve hafiza kapasitesi harcar. Programi bu komutu çikararak da çalistiriniz ve aradaki farklari inceleyiniz.

Sekli olusturan temel komutlar:

### GeneralPath sekil=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD); sekil.moveTo(20,20);

**//quadratik ekleme sekil.quadTo(160,120,245,45);**

**//kübik ekleme sekil.curveTo(195,95,295,145,245,195);**

komutlaridir. Daha sonra bu sekiller

### g2.draw(sekil);

komutuyla çizdirilmistir.

### g2.fill(sekil);

komutu burada kullanilmamakla birlikte, geçerli bir komuttur. Programda deneyiniz. Bu programda çizdirilen ikinci dereceden (quadratik egriler) quadTo ve curveTo metodlari kullanilarak olusturulmustur. Simdi de ikinci derece polynom uydurmasini QuadCurve2D sinifini kullanarak yapan bir programi inceleyelim :

### Program 6.29 : curve2SWF\_2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class curve2SWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white;

final static Color siyah=Color.black;

public curve2SWF\_2000()

{

super("ikinci dereceden egri çizimi");

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON); setBackground(Color.lightGray);

setForeground(siyah); Dimension boyut=getSize(); int dx=boyut.width;

int dy=boyut.height; g2.setPaint(siyah); g2.draw3DRect(0,0,dx-3,dy-3,true); g2.draw3DRect(3,3,dx-7,dy-7,false); g2.setPaint(siyah);

QuadCurve2D qc2=new QuadCurve2D.Double(0,125,140,225,225,150); g2.draw(qc2);

QuadCurve2D qc2\_1=new QuadCurve2D.Double(0,200,155,225,225,170); g2.setPaint(Color.blue);

g2.fill(qc2\_1);

}

//=====================================================

public static void main(String[] args)

{

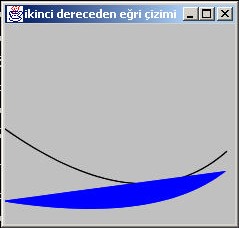
curve2SWF\_2000 pencere= new curve2SWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06032.JPG



**Sekil 6.32 ikinci dereceden genel egri çizimi, curve2SWF\_2000, JFrame çiktisinda görünümü**

Simdide üçüncü dereceden egri uyduran bir program koduna göz atalim :

### Program 6.30 : curve3SWF\_2000.java

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class curve3SWF\_2000 extends JFrame { final static Color kirmizi=Color.red;

final static Color beyaz=Color.white; final static Color siyah=Color.black;

public curve3SWF\_2000()

{

super("üçüncü dereceden egri çizimi");

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON); setBackground(Color.lightGray);

setForeground(siyah); Dimension boyut=getSize(); int dx=boyut.width;

int dy=boyut.height; g2.setPaint(siyah); g2.draw3DRect(0,0,dx-3,dy-3,true); g2.draw3DRect(3,3,dx-7,dy-7,false); g2.setPaint(siyah);

CubicCurve2D cc3=new CubicCurve2D.Double(50,175,140,90,10,90,100,175); g2.draw(cc3);

CubicCurve2D cc3\_1=new CubicCurve2D.Double(50,275,140,190,10,190,100,275); g2.setPaint(Color.blue);

g2.fill(cc3\_1);

}

//=====================================================

public static void main(String[] args)

{

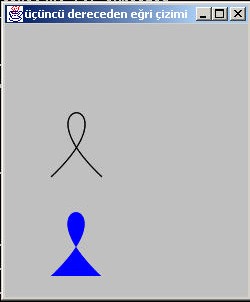
curve3SWF\_2000 pencere= new curve3SWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,300);

pencere.setVisible(true);

}

}

### 06033.JPG



**Sekil 6.33 üçüncü dereceden genel egri çizimi, curve3SWF\_2000, JFrame çiktisinda görünümü**

### RESIM GÖSTERIMI

Java dili sadece .gif ve .jpg standartlarindaki resimleri gösterebilir. Resim önce **getImage** metotuyla bir Image sinifi nesneye yuklenir, sonra **drawImage** metotuyla gösterilir.

Ornegin Image **resim=getImage(getDocumentBase(),”resim.gif”);**

Ve resim **g.drawImage(resim,1,1,this);** deyimiyle çizilir. Burada ikinci ve üçüncü noktalar baslangiç x ve y koordinatlaridir.Son olarak **this** resmin çizilecegi appletiniçinde bulunulan sinifin appleti oldugunu belirtmektedir.

Bir örnek problemle resim çizimini biraz daha belirgin hale getitelim :

### Program 6.31: resim.java resim aktarim programi

import java.applet.Applet; import java.awt.\*;

public class resim extends Applet

{

private Image res; public void init()

{

res=getImage(getDocumentBase(),"ballarat.gif");

}

public void paint(Graphics g)

{

g.drawImage(res,1,1,this);

/\*

// resmi orijinalinin iki kati boyunda ciziniz int en=res.getWidth(this);

int yukseklik=res.getWidth(this); g.drawImage(res,1,90,en\*2,yukseklik\*2,this);

\*/

}

}

### 06034.JPG



**Sekil 6.34 : Resmin java üzerinden aktarilmasi, resim.html**

Java grafik çizim olasiliklari burada verilenden çok aha genis boyutludur. Bu bölümde veremedigimiz bir çok detayi javanin parasiz olarak çekebileceginiz döküman kütüphanesinde bulabilirsiniz.

### ALISTIRMALAR

1. ciz1.java programini inceleyiniz.

### Program 6.13 ciz1.java

import java.applet.Applet; import java.awt.\*;

public class ciz1 extends Applet

{

private String s ="Bunu drawString Metotuyla yazdir"; private char c[]={'c','h','a','r','s',' ','8'};

private byte b[]={'b','y','t','e',75,76,77}; public void init()

{

Color c=new Color(0,0,255); setBackground(c);

}

public void paint(Graphics g)

{

g.setColor(new Color(100,255,125)); g.setFont(new Font("SansSerif",Font.PLAIN,24)); g.drawString(s,100,25); g.drawChars(c,2,3,100,50);

g.drawBytes(b,0,6,100,75);

}

}

### 06015.JPG



**Sekil 6.15 ciz1.html appleti**

1. Iki boyutlu plot çizimi amaciyla bir program paketi hazirlanmistir. Bu paket temel olarak üç java programindan olusmaktadir. Birinci program dosyadan plot bilgisini okuyan Plot.java programidir. Çizilecek Plotla ilgili en temel bilgi Plot.txt dosyasinda bulunmaktadir. Bu dosyadaki ilk satir Pot basligini, ikinci satir x ekseni basligini, üçüncü satir y ekseni basligini, dördüncü satir çizilecek iki boyutlu plot sayisini (okunacak dosya sayisini), besinci ve sonraki satirlar plot bilgisinin oldugu dosya ismi, çizgi çizim türü, ve integer RGB renk degerlerini vermektedir. Çizgi çizim türlerini bazilari

// plottype = 0 sürekli çizgi

// plottype = 10 kullanici tarafindan tanimlanan karakter

// plottype = 20 dikdörtgen

// plottype = 21 daire

// plottype = 22 üçgen

// plottype = 23 eskenar dörtgen

// plottype = 24 besgen

// plottype = 25 altigen

eger çizgi türü 10 olarak tanimlanmissa kullanilacak olan karakterinde verilmesi gerekir a.dat 10 \* 0 0 255 gibi. Asagida örnek bir Plot.txt dosyasi degeri verilmektedir.

türkçe yazi x ekseni

y ekseni 2

a.dat 26 0 0 255

a.dat 5 0 0 255

Plot bilgi dosyasinda iki kolon olarak x,y plot bilgisi verilir. Küçük bir örnek dosyasi verecek olursak : 1 1

2 4

3 9

4 16

5 25

6 36

7 49

Burada bilgi dosyadan okunmaktadir. Dosya okuma yazma komutlarini daha detayli olarak 10. bölümde inceleyecegiz.

### Program 6.14 : Plot.java

//======================================================

// Numerical Analysis package in java

// Plot class to plot 2D data

// Dr. Turhan Coban

// =====================================================

import java.io.\*; public class Plot

{

public String label; // Plot Label

public String xlabel; // X axis Label

public String ylabel; // Y axis Label

public String filename; // file name

public double xmax,xmin,ymax,ymin; // real max,min scale

public int xabsmin,yabsmin,abswidth,absheight; // absolute max,min scale public int nmax; // max number of data points

public int nline; // max number of data sets public double x[][]; // x data

public double y[][]; // y=f(x) data

public int n[]; // n : number of data in each line public int red[],green[],blue[]; // color code 0-255

public int plottype[]; public int xgridon; public int ygridon;

// plottype = 0 continuous line

// plottype = 10 user defined character plot

// plottype = 20 rectangle

// plottype = 21 circle

// plottype = 22 triangle

// plottype = 23 diamond

// plottype = 24 pentagon

// plottype = 25 hexagon

public char ch[]; // plot character (used with plottype 10) public int xntic; // number of tics in x axis

public int yntic; // number of tics in y axis BufferedReader fin;

BufferedReader ffile;

File plotFile;

File lineDataFile;

public void set\_plotwindow(int width,int height,double xip,double yip,double dxp,double dyp)

{

//sets rectangular plot window xabsmin = (int)(width\*(xip+0.2)); yabsmin = (int)(height\*(yip+0.2)); abswidth = (int)(width\*dxp); absheight = (int)(height\*dyp);

}

public Plot(String pl,String xl,String yl,int xnt,int ynt

,int xgo,int ygo,String ifn[],int ipt[],int ir[],int ig[]

,int ib[]) throws IOException

{

xmin=9.99e50; xmax=-9.99e50; ymin=9.99e50; ymax=-9.99e50;

xntic=8; yntic=8;

double xtemp,ytemp; label=pl;

xlabel=xl; ylabel=yl; nline=ifn.length; xntic=xnt; yntic=ynt; xgridon=xgo; ygridon=ygo; n=new int[nline]; red=new int[nline];

green=new int[nline]; blue=new int[nline]; plottype=new int[nline]; ch=new char[nline];

//read all data to determine limit values String fn[];

fn=new String[nline]; int i,j; for(i=0;i<nline;i++)

{

fn[i]=ifn[i]; plottype[i]=ipt[i]; if(plottype[i] == 10) ch[i]='\*'; red[i]=ir[i];

green[i]=ig[i];

blue[i]=ib[i]; j=0;

n[i]=0;

//open ffile try

{

ffile=new BufferedReader(new FileReader(fn[i]));

} catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

try

{

while(ffile!=null)

{

xtemp=Text.readDouble(ffile); ytemp=Text.readDouble(ffile); if(xtemp<xmin) xmin=xtemp; if(xtempxmax) xmax=xtemp; if(ytemp<ymin) ymin=ytemp; if(ytempymax) ymax=ytemp; n[i]++;

}

} catch(EOFException e\_eof) { ffile.close();} if(i==0) nmax=n[i];

else

{

if(n[i]nmax) nmax=n[i];

}

x=new double[nline][nmax]; y=new double[nline][nmax];

for(i=0;i<nline;i++)

{

//open ffile try{

ffile=new BufferedReader(new FileReader(lineDataFile));

} catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

for(j=0;j<n[i];j++)

{

x[i][j]=Text.readDouble(ffile); y[i][j]=Text.readDouble(ffile);

}

}

}

set\_plotwindow(400,400,0.1,0.1,0.9,0.9);

}

public Plot() throws IOException

{

//adding max-min prompts and input fields

//initial values of max and mins

// Structure of input file

// filename

// xlabel

// ylabel

// nline xntic yntic xgridon ygridon

// datafilename plottype redcolor greencolor bluecolor

// ........

// datafilename plottype redcolor greencolor bluecolor

//

// A sample data file input :

//

// 4 10 10 1 1

// b.dat 22 0 0 0

// a.dat 21 0 255 0

// a.dat 0 0 255 0

// b.dat 0 0 0 255

//

// see above for meaning of plot types xmin=9.99e50;

xmax=-9.99e50; ymin=9.99e50; ymax=-9.99e50; xntic=8; yntic=8;

double xtemp,ytemp; try{

fin=new BufferedReader(new FileReader("Plot.txt"));

} catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

label=Text.readStringLine(fin); xlabel=Text.readStringLine(fin); ylabel=Text.readStringLine(fin); nline=Text.readInt(fin); xntic=Text.readInt(fin);

yntic=Text.readInt(fin); xgridon=Text.readInt(fin); ygridon=Text.readInt(fin); n=new int[30];

red=new int[30]; green=new int[30]; blue=new int[30]; plottype=new int[30]; ch=new char[30];

//read all data to determine limit values String fn[];

fn=new String[nline]; int i,j; for(i=0;i<nline;i++)

{

fn[i]=Text.readString(fin); plottype[i]=Text.readInt(fin);

if(plottype[i] == 10) ch[i]=Text.readChar(fin); red[i]=Text.readInt(fin); green[i]=Text.readInt(fin); blue[i]=Text.readInt(fin);

j=0; n[i]=0;

//open ffile try{

ffile=new BufferedReader(new FileReader(fn[i]));

} catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

try{ while(ffile!=null)

{

xtemp=Text.readDouble(ffile); ytemp=Text.readDouble(ffile); if(xtemp<xmin) xmin=xtemp; if(xtempxmax) xmax=xtemp; if(ytemp<ymin) ymin=ytemp; if(ytempymax) ymax=ytemp; n[i]++;

}

} catch(EOFException e\_eof)

{

//close ffile try{ ffile.close();

} catch(IOException e)

{

System.err.println("Error Closing File\n"+e.toString()); System.exit(1);

}

} //End of EOFException if(i==0) nmax=n[i];

else

{

if(n[i]nmax) nmax=n[i];

}

}

//close ffile

try{

ffile.close();

} catch(IOException e)

{

System.err.println("Error Closing File\n"+e.toString()); System.exit(1);

}

x=new double[nline][nmax]; y=new double[nline][nmax];

//re-read the data again to load it for(i=0;i<nline;i++)

{

//open ffile try{

ffile=new BufferedReader(new FileReader(fn[i]));

} catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

for(j=0;j<n[i];j++)

{

x[i][j]=Text.readDouble(ffile); y[i][j]=Text.readDouble(ffile);

}

//close ffile try{

ffile.close();

} catch(IOException e)

{

System.err.println("Error Closing File\n"+e.toString()); System.exit(1);

}

}

set\_plotwindow(400,400,0.1,0.1,0.9,0.9);

//close fin try{

fin.close();

} catch(IOException e)

{

System.err.println("Error Closing File\n"+e.toString()); System.exit(1);

}

}

}

### Program 6.15 : PlotShapes.java

//======================================================

// Numerical Analysis package in java

// PlotShapes class

// This class convert graphic draw methods to

// plot coordinates and gives additional plotting methods

// Dr. Turhan Coban

// =====================================================

import java.io.\*; import java.awt.\*;

import java.applet.Applet;

public class PlotShapes

{

Graphics g;

int xabsmin,yabsmin; int absheight,abswidth;

double xmin,xmax,ymin,ymax;

public PlotShapes(Graphics gi,int xabsmini ,int yabsmini,

int absheighti,int abswidthi, double xmini,double xmaxi, double ymini,double ymaxi)

{

// xabsmin : absulute stating point x axis

// yabsmin : absolute starting point y axis

// absheight : absoulute height of plotting window

// abswidth : absolute width of plotting window

// xmin : minimum x value (real number)

// xmax : maximum x value (real number)

// ymin : minimum y value (real number)

// ymax : maximum y value (real number)

// g : graphic object that actual drawing is done through g=gi;

xabsmin=xabsmini; yabsmin=yabsmini; absheight=absheighti; abswidth=abswidthi; xmin=xmini; xmax=xmaxi; ymin=ymini; ymax=ymaxi;

}

public void drawLine(int plottype,double x1,double y1,double x2,double y2)

{

// draw a line from (x1,y1) to (x2,y2) if(plottype==0) //draw a continuous line

{

g.drawLine( (int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight), (int)(xabsmin+(x2-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y2-ymin)/(ymax-ymin)\*absheight));

}

} //end of drawLine

public void drawChar(char ch,double x1,double y1)

{

// draws a single character at (x1,y1) char ch1[]=new char[1]; ch1[0]=ch;

int h=g.getFontMetrics().getHeight(); int w=h-2;

g.drawChars(ch1,0,1,

(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth)-2, (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+h/2);

} //end of PlotShapes.drawChar

public void drawChars(char ch[],int firstspace,int numberofchars,

double x1,double y1)

{

// draws a character array from space firstspace to (firstspace+numberofchars)

// starting at (x1,y1)

int h=g.getFontMetrics().getHeight(); int w=h-2;

g.drawChars(ch,firstspace,numberofchars, (int)((xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth)+5),

(int)((yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+h/2));

} //end of PlotShapes.drawChars

public void drawString(String s,double x1,double y1)

{

//draws a String at (x1,y1)

int h=g.getFontMetrics().getHeight(); int w=h-2;

g.drawString(s,

(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight));

} //end of PlotShapes.drawStrings

public void drawRect(double x1,double y1,int rectwidth,int rectheight)

{

// draw a rectangle starting at (x1,y1)

// with dimensions of (rectwidth,rectheight) g.drawRect(

(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-rectwidth/2), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight-rectheight/2),

rectwidth,rectheight);

}//end of drawRect

public void drawOval(double x1,double y1,int width,int height)

{

// draw an oval with the centre of (x1,y1)

// with dimension of (width,height) g.drawOval(

(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-width/2), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight-height/2), width,height);

}//end of drawRect

public void drawPolygon(double x1,double y1,int radius,int side)

{

//draw a polygon of n sides n=3(trinagle),n=4(dimond).... double Pi=4.0\*Math.atan(1.0);

int xvalues[]=new int[side+1]; int yvalues[]=new int[side+1]; double angle\_increase;

double angle; angle\_increase=2.0\*Pi/side; angle=Pi/2.0;

for(int i=0;i<side;i++)

{

xvalues[i]= (int)(Math.floor(xabsmin+(x1-xmin)/ (xmax-xmin)\*abswidth+radius\*Math.cos(angle)));

yvalues[i]= (int)(Math.floor(yabsmin+absheight-(y1-ymin)/ (ymax-ymin)\*absheight-radius\*Math.sin(angle)));

if(i==0) {xvalues[side]=xvalues[i];yvalues[side]=yvalues[i];} angle+=angle\_increase;

}

g.drawPolygon(xvalues,yvalues,side+1);

}//end of drawPolygon

public void drawXTic(int ticNumber,double ticHeight,int ticSide)

{

//draw a series of x axis tics

double dtic=(xmax-xmin)/ticNumber; double x1=xmin;

double y1=ymin; if(ticSide==0)

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,x1,(y1-ticHeight)); x1=x1+dtic;

}

}

else

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,x1,(y1+ticHeight)); x1=x1+dtic;

}

}

}

public void drawYTic(int ticNumber,double ticWidth,int ticSide)

{

//draw a series of y axis tics

double dtic=(ymax-ymin)/ticNumber; double x1=xmin;

double y1=ymin; if(ticSide==0)

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,(x1-ticWidth),y1); y1=y1+dtic;

}

}

else

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,(x1+ticWidth),y1); y1=y1+dtic;

}

}

}

public void drawXGrid(int ticNumber)

{

//draw x gridlies

double dtic=(xmax-xmin)/ticNumber; double x1=xmin;

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,ymin,x1,ymax); x1=x1+dtic;

}

}

public void drawYGrid(int ticNumber)

{

//draw y gridlines =====

double dtic=(ymax-ymin)/ticNumber; double y1=ymin;

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,xmin,y1,xmax,y1); y1=y1+dtic;

}

}

public void drawXNumbers(int ticNumber)

{

//draw x numbers

double dtic=(xmax-xmin)/ticNumber; double x1=xmin;

double y1=ymin;

String s;

s=" ";

for(int i=0;i<=ticNumber;i++)

{

s=Double.toString(Math.floor(x1\*100.0)/100.00); g.drawString(s,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+20); x1=x1+dtic;

}

}

public void drawYNumbers(int ticNumber)

{

// draw y numbers

double dtic=(ymax-ymin)/ticNumber; double x1=xmin;

double y1=ymin;

String s;

s=" ";

for(int i=0;i<=ticNumber;i++)

{

s=Double.toString(Math.floor(y1\*100)/100.0)+" "; g.drawString(s,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-40), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)); y1=y1+dtic;

}

}

public void drawXLabel(String xLabel)

{

// draw x labels

double x1=xmin+(xmax-xmin)/2.0; double y1=ymin;

g.drawString(xLabel,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+30);

}

public void drawYLabel(String yLabel)

{

// draw y labels

double x1=xmin-(xmax-xmin)/6.0; double y1=ymax-(ymax-ymin)/3.0; int n=yLabel.length();

char ch[]=new char[n]; yLabel.getChars(0,n,ch,0);

// g.DrawString(yLabel,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-60),

// (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)); for(int i=0;i<n;i++)

{

drawChar(ch[i],x1,y1); y1=y1-(ymax-ymin)/30;

}

}

public void drawLabel(String Label)

{

//draw graphic label

double x1=xmin+(xmax-xmin)/2.0; double y1=ymax;

g.drawString(Label,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)-10);

}

public void

drawPlotLines(int i,int plottype[],double x[][],double y[][],int n[],char ch[])

{

int j;

//draw lines if(plottype[i]==0)

{

for(j=0;j<n[i]-1;j++)

{

if((x[i][j]=xmin && x[i][j]<=xmax )

&& (y[i][j]=ymin && y[i][j]<=ymax))

{

if((x[i][j+1]=xmin && x[i][j+1]<=xmax) && (y[i][j+1]=ymin && y[i][j+1]<=ymax))

{

drawLine(0,x[i][j],y[i][j],x[i][j+1],y[i][j+1]);

}

else if(x[i][j+1]xmax)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j]-b\*x[i][j]; drawLine(0,x[i][j],y[i][j],xmax,(a+b\*xmax));

}

else if(y[i][j+1]ymax)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j]-b\*x[i][j]; drawLine(0,x[i][j],y[i][j],(ymax-a)/b,ymax);

}

else if(x[i][j+1]xmax && y[i][j+1]ymax)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j]-b\*x[i][j]; drawLine(0,x[i][j],y[i][j],(ymax-a)/b,(a+b\*xmax));

}

}

else if((x[i][j+1]=xmin && x[i][j+1]<=xmax ) && (y[i][j+1]=ymin && y[i][j+1]<=ymax))

{

if(x[i][j]<xmin)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j+1]-b\*x[i][j+1]; drawLine(0,xmin,(a+b\*xmin),x[i][j+1],y[i][j+1]);

}

if(y[i][j]<ymin)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j+1]-b\*x[i][j+1]; drawLine(0,(ymin-a)/b,ymin,x[i][j+1],y[i][j+1]);

}

}

}//end of for(j=0

}//end of if(plottype

//draw characters

else if(plottype[i]==10)

{

for(j=0;j<n[i];j++)

{

if((x[i][j]=xmin && x[i][j]<=xmax )

&& (y[i][j]=ymin && y[i][j]<=ymax))

{ drawChar(ch[i],x[i][j],y[i][j]); }

}//end of for(j=0;

}//end else if(plottype[i]==10)

//draw rectangles

else if(plottype[i]==20)

{

for(j=0;j<n[i];j++)

{

if((x[i][j]=xmin && x[i][j]<=xmax )

&& (y[i][j]=ymin && y[i][j]<=ymax))

{ drawRect(x[i][j],y[i][j],4,4); }

}//end of for(j=0;

}//end else if(plottype[i]==20)

//draw circle

else if(plottype[i]==21)

{

for(j=0;j<n[i];j++)

{

if((x[i][j]=xmin && x[i][j]<=xmax )

&& (y[i][j]=ymin && y[i][j]<=ymax))

{ drawOval(x[i][j],y[i][j],4,4); }

}//end of for(j=0;

}//end else if(plottype[i]==21)

else if(plottype[i]=22 && plottype[i]<=27)

{

for(j=0;j<n[i];j++)

{ if((x[i][j]=xmin && x[i][j]<=xmax )

&& (y[i][j]=ymin && y[i][j]<=ymax))

{drawPolygon(x[i][j],y[i][j],5,(plottype[i]-19));}

}//end of for(j=0;

}//end else if(plottype[i]==21..27)

}

}

### Program 6.16 : PlotApplet.java

//======================================================

// Numerical Analysis package in java

// PlotApplet class to plot 2D data

// Dr. Turhan Coban

// =====================================================

import java.io.\*; import java.applet.\*; import java.awt.\*;

import java.awt.event.\*; import PlotShapes; import Plot;

public class PlotApplet extends Applet implements ActionListener

{

Label promptXmin; // Label prompt in Xmin field Label promptXmax; // Label prompt in Xmax field Label promptYmin; // Label prompt in Ymin field Label promptYmax; // Label prompt in Ymax field

TextField inputXmin; // input field Xmin TextField inputXmax; // input field Xmax TextField inputYmin; // input field Ymin TextField inputYmax; // input field Ymax Plot p1;

public void init()

{

//adding max-min prompts and input fields

//label

//xaxis

//yaxis

// 4 10 10 1 1

// b.dat 22 0 0 0

// a.dat 21 0 255 0

// a.dat 0 0 255 0

// b.dat 0 0 0 255 int xnt=10;

int ynt=10; int xgo=1; int ygo=1;

String il=new String("label"); String xl=new String("x axis"); String yl=new String("y axis"); String ifn[]=new String[4]; ifn[0]="b.dat";

ifn[1]="a.dat";

ifn[2]="a.dat";

ifn[3]="b.dat";

int ipt[]=new int[4]; int ib[]=new int[4]; int ir[]=new int[4]; int ig[]=new int[4]; ipt[0]=22; ipt[1]=21;

ipt[2]=0;

ipt[3]=0;

ir[0]=0;

ig[0]=0;

ib[0]=0;

ir[1]=0; ig[1]=255; ib[1]=0;

ir[2]=0; ig[2]=255; ib[2]=0;

ir[3]=0; ig[3]=255; ib[3]=0;

try{

//p1=new Plot(il,xl,yl,xnt,ynt,xgo,ygo,ifn,ipt,ir,ig,ib); p1=new Plot();

} catch(IOException ioe) {System.err.println("IOException in opening Plot");} promptXmin=new Label("Xmin ");

inputXmin=new TextField(5); promptXmax=new Label("Xmax "); inputXmax=new TextField(5); promptYmin=new Label("Ymin "); inputYmin=new TextField(5); promptYmax=new Label("Ymax "); inputYmax=new TextField(5);

inputXmin.addActionListener(this); inputXmax.addActionListener(this); inputYmin.addActionListener(this); inputYmax.addActionListener(this); add(promptXmin);

add(inputXmin); add(promptXmax); add(inputXmax); add(promptYmin); add(inputYmin); add(promptYmax); add(inputYmax);

}

public void paint(Graphics g)

{

inputXmin.setText(Double.toString(p1.xmin)); inputXmax.setText(Double.toString(p1.xmax)); inputYmin.setText(Double.toString(p1.ymin)); inputYmax.setText(Double.toString(p1.ymax)); PlotShapes ps=new PlotShapes(g,p1.xabsmin,p1.yabsmin,

p1.absheight,p1.abswidth,p1.xmin,p1.xmax,p1.ymin,p1.ymax); g.drawRect(p1.xabsmin,p1.yabsmin,p1.abswidth,p1.absheight); ps.drawXTic(p1.xntic,(p1.ymax-p1.ymin)/80.0,0); ps.drawYTic(p1.yntic,(p1.xmax-p1.xmin)/80.0,0); ps.drawXNumbers(p1.xntic);

ps.drawYNumbers(p1.xntic); ps.drawLabel(p1.label); ps.drawXLabel(p1.xlabel); ps.drawYLabel(p1.ylabel); if(p1.xgridon!=0) ps.drawXGrid(p1.xntic); if(p1.ygridon!=0) ps.drawYGrid(p1.yntic);

int i,j; for(i=0;i<p1.nline;i++)

{

// Select plot colors

g.setColor(new Color(p1.red[i],p1.green[i],p1.blue[i])); ps.drawPlotLines(i,p1.plottype,p1.x,p1.y,p1.n,p1.ch);

}//end of for(i=0

}//end of method

public void actionPerformed( ActionEvent e)

{

Double valXmin=new Double(inputXmin.getText()); p1.xmin=valXmin.doubleValue();

Double valXmax=new Double(inputXmax.getText()); p1.xmax=valXmax.doubleValue();

Double valYmin=new Double(inputYmin.getText()); p1.ymin=valYmin.doubleValue();

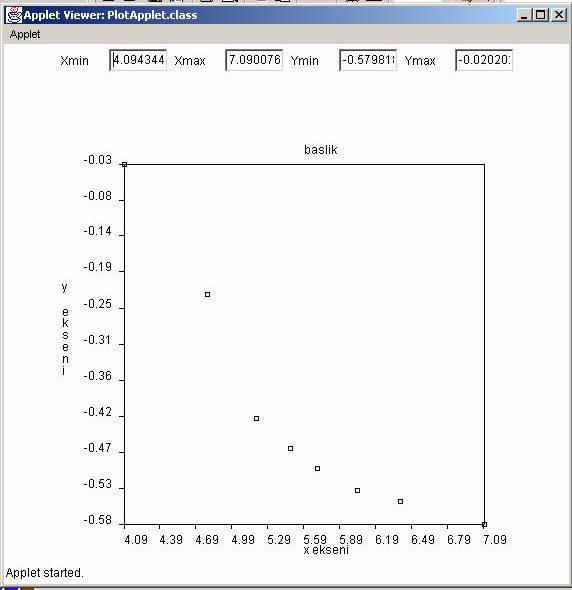
Double valYmax=new Double(inputYmax.getText()); p1.ymax=valYmax.doubleValue();

repaint();

}

}

### 06016.JPG



**Sekil 6.16 : PlotApplet.html çiktisi**

1. drawRect ve fillRect metotlarini kullanarak biri dolu digeri sadece bir çizgi olan iki dikdörtgen çiz.
2. Oval ve fillOval metotlarini kullanarak biri dolu digeri sadece bir çizgi olan iki daire çiz.
3. Paint2D metodlarindan yararlanarak biri dolu digeri sadece bir çizgi olan iki eskenar dörtgen çizdir.
4. resim.java sinifinda çizilen resmi iki kat büyüklügnde çizdir.

### Program 6.17 resim1.java

import java.applet.Applet; import java.awt.\*;

public class resim1 extends Applet

{

private Image bebek; public void init()

{

bebek=getImage(getDocumentBase(),"turhan.gif");

}

public void paint(Graphics g)

{

g.drawImage(bebek,1,1,this);

}

}

1. drawPolygon ve fillPolygon metotlarini kullanarak biri dolu digeri sadece bir çizgi olan iki eskenar üçgen çiz.
2. Türkçe renkleri tanimliyan renk sinifini yaz (renk.java).

### Program 6.18 . renk.java, türkçe renk tanimi sinifi

import java.io.\*; import java.awt.\*;

import java.applet.Applet; public class renk extends Color

{

public final static renk kirmizi=new renk(255,0,0); public final static renk mavi=new renk(0,0,255); public final static renk siyah=new renk(0,0,0);

public final static renk camgobegi=new renk(0,255,255); public final static renk koyugri=new renk(64,64,64); public final static renk gri=new renk(128,138,128); public final static renk yesil=new renk(0,255,0);

public final static renk acikgri=new renk(192,192,192); public final static renk mor=new renk(255,0,255); public final static renk portakal=new renk(255,200,0); public final static renk pembe=new renk(255,175,175); public final static renk beyaz=new renk(255,255,255); public final static renk sari=new renk(255,255,0);

public final static renk acikmavi=new renk(150,206,237); public final static renk lacivert=new renk(0,0,128);

//burada kendi yeni renginizi tanimlayabilirsiniz. public renk(float kirmizi,float yesil, float mavi)

{

super(kirmizi,yesil,mavi);

}

public renk(double kirmizi,double yesil, double mavi)

{

super((float)kirmizi,(float)yesil,(float)mavi);

}

public renk(int kirmizi,int yesil, int mavi)

{

super(kirmizi,yesil,mavi);

}

public renk(int RGB)

{

super(RGB);

}

public renk(renk r)

{

super(r.getKirmizi(),r.getYesil(),r.getMavi());

}

public int getKirmizi()

{

return getRed();

}

public int getYesil()

{

return getGreen();

}

public int getMavi()

{

return getBlue();

}

public int getKYM()

{

return getRGB();

}

public String toString()

{

return "renk[ kirmizi "+getKirmizi()+" mavi "+getMavi()+" yesil "+getYesil()+"]";

}

//metotlar

//static int HSBtoRGB(float hue,float saturation,float brightness)

//getRed(),getGreen(),getBlue(),getRGB()

}

1. Problem 5.10 da verilen Polar.java, yildiz.java ve yildizCiz.java programlarini inceleyiniz.

### Problem 6.19 : Yildiz.java

import java.io.\*; import java.awt.\*;

import java.applet.Applet; import polar;

public class yildiz

{

public static void drawYildiz1(Graphics g,int xi,int yi, int n,int yildizboyu,double aci )

{

// bu yildiz cizime teta=pi/2+aci radyandan baslar double teta=2.0\*Math.PI/n;

double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar(); for(int i=0;i<n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2);

g.drawLine((xi+(int)P1.xi()),(yi-(int)P1.xj()),

(xi+(int)P2.xi()),(yi-(int)P2.xj())); double teta3=teta\*(i+1)+Math.PI/2.0+aci; P3.polarGir(R,teta3); g.drawLine((xi+(int)P2.xi()),(yi-(int)P2.xj()),

(xi+(int)P3.xi()),(yi-(int)P3.xj())); g.drawLine(xi,yi,

(xi+(int)P1.xi()),(yi-(int)P1.xj())); g.drawLine(xi,yi,

(xi+(int)P2.xi()),(yi-(int)P2.xj()));

}

} //drawYildiz1 metodu sonu

public static void drawYildiz1(Graphics g,int xi,int yi, int n,int yildizboyu )

{

drawYildiz1(g,xi,yi,n,yildizboyu,0);

} //drawYildiz1 metodu sonu

public static void drawYildiz(Graphics g,int xi,int yi, int n,int yildizboyu,double aci )

{

// bu y• ld• z cizime teta=pi/2+aci radyandan baslar double teta=2.0\*Math.PI/n;

double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2]; for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2.0+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2); x[2\*i]=xi+(int)P1.xi();

y[2\*i]=yi-(int)P1.xj();

x[2\*i+1]=xi+(int)P2.xi();

y[2\*i+1]=yi-(int)P2.xj(); if(i==n)

{

x[2\*i]=x[0];

y[2\*i]=y[0];

}

}

g.drawPolygon(x,y,(2\*n+2));

} //drawYildiz1 metodu sonu

public static void fillYildiz(Graphics g,int xi,int yi, int n,int yildizboyu,double aci )

{

// bu y• ld• z cizime teta=pi/2+aci radyandan baslar

// yild• zin i‡ini boyar double teta=2.0\*Math.PI/n; double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2]; for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2.0+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2); x[2\*i]=xi+(int)P1.xi();

y[2\*i]=yi-(int)P1.xj();

x[2\*i+1]=xi+(int)P2.xi();

y[2\*i+1]=yi-(int)P2.xj(); if(i==n)

{

x[2\*i]=x[0];

y[2\*i]=y[0];

}

}

g.fillPolygon(x,y,(2\*n+2));

} //fillYildiz metodu sonu

public static void fillYildiz1(Graphics g,int xi,int yi, int n,int yildizboyu,double aci )

{

// bu y• ld• z cizime teta=pi/2+aci radyandan baslar

// yild• zin i‡ini boyar double teta=2.0\*Math.PI/n; double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2]; for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2.0+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2); x[2\*i]=xi+(int)P1.xi();

y[2\*i]=yi-(int)P1.xj();

x[2\*i+1]=xi+(int)P2.xi();

y[2\*i+1]=yi-(int)P2.xj(); if(i==n)

{

x[2\*i]=x[0];

y[2\*i]=y[0];

}

}

g.fillPolygon(x,y,(2\*n+2));

} //fillYildiz metodu sonu

}

### Program 6.20 : YildizCiz.java

import java.applet.Applet; import java.awt.\*;

import yildiz; import renk;

public class yildizCiz extends Applet

{

public void paint(Graphics g)

{

//g.setColor(Color.yellow); g.setColor(renk.sari); setBackground(renk.mavi); yildiz.fillYildiz(g,100,100,3,100,0); g.setColor(renk.siyah); yildiz.drawYildiz1(g,100,100,3,100,0);

}

}

**10. yildizCiz1.java** programini inceleyiniz.

### Program 6.21 YildizCiz1.java

import java.applet.Applet; import java.awt.\*;

import yildiz;

public class yildizCiz1 extends Applet

{

public void paint(Graphics g)

{

double aci=0;

for(long j=0;j<5000;j++)

{

g.setColor(Color.black); yildiz.drawYildiz(g,100,100,3,100,aci); for(long i=0;i<10000;i++) {}; g.setColor(Color.white); yildiz.drawYildiz(g,100,100,3,100,aci); aci+=.001;

}

g.setColor(Color.black); yildiz.drawYildiz(g,100,100,3,100,0);

}

}

**7.** Program 6.18 de verilen renk sinifini test eden renkTest.java programini inceleyiniz. Ve applet çiktisini çalistiriniz.

### Program 6.21 renkTest.java

import java.awt.\*;

import java.applet.Applet; import renk;

public class RenkTest extends Applet

{

private int kirmizi,yesil,mavi;

public void init()

{

kirmizi=150; yesil=255; mavi=125;

// arka palan• n rengi mavi olarak veriliyor setBackground(renk.mavi);

}

public void paint(Graphics g)

{

// yazinin rengi 100,255,125 olarak alindi

g.setColor(new renk(kirmizi,yesil,mavi)); g.drawString("ABCDEFGHIJKLMNOPQRSTUVWXYZ",50,33);

showStatus("Su andaki arkaplan rengi : "+getBackground());

}

### 06017



**Sekil 6.17 RenkTest.html appletinin görünümü**

**11. H6O2**

polar.java, yildiz.java, yildizCiz.java programlarinin inceleyiniz. ayni programlama prensiplerini kullanarak n eskenar kenarli çokken çizecek eskenarcokgen.java sinifini yaratiniz. Bu sinif

### public static void drawEskenarCokgen(Graphics g,int xi,int yi, int n,int esgenarcokkencapi,double aci ) public static void fillEskenarCokgen(Graphics g,int xi,int yi, int n,int yildizboyu,double aci )

metotlarini kullanarak eskenar cokgen çizsin ve icini boyasin. not xi,yi : çokken merkezinin koordinatlari

n : çokken köse sayisi

eskenarcokkencapi: çokken merkezinden cokken kösesine olan mesafe

aci : 90 derece açi çokkenin baslama açisi olarak tanimlanmistir. bu açidan sapis radyan olarak.

### Program 6.22 eskenarcokgen.java

import java.io.\*; import java.awt.\*;

import java.applet.Applet; import polar;

public class eskenarcokgen

{

public static void drawEskenarcokgen(Graphics g,int xi,int yi, int n,int eskenarcokgenboyu,double aci )

{

// xi,yi eskenar cokgen merkez koordinatlari

// n : eskenar dortgen kose sayisi

// eskenarcokkenboyu : eskenar cokgen merkezinden

// kenarina olan mesafe

// bu eskenarcokken cizime teta=pi/2+aci radyandan baslar

double teta=2.0\*Math.PI/n; double R=eskenarcokgenboyu; polar P1=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2]; for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; P1.polarGir(R,teta1); x[i]=xi+(int)P1.xi();

y[i]=yi-(int)P1.xj(); if(i==n)

{ x[i]=x[0];

y[i]=y[0];

}

}

g.drawPolygon(x,y,(n+1));

} //draweskenarcokgen metotu sonu

public static void fillEskenarcokgen(Graphics g,int xi,int yi, int n,int eskenarcokgenboyu,double aci )

{

// bu eskenarcokgen cizime teta=pi/2+aci radyandan baslar double teta=2.0\*Math.PI/n;

double R=eskenarcokgenboyu; polar P1=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2]; for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; P1.polarGir(R,teta1); x[i]=xi+(int)P1.xi();

y[i]=yi-(int)P1.xj(); if(i==n)

{ x[i]=x[0];

y[i]=y[0];

}

}

g.fillPolygon(x,y,(n+1));

} //filleskenarcokgen metotu sonu

}

1. eskenarcokgen sinifini test etmek amaciyla bir applet programi yaziniz ve ekrana besken çizdiriniz.
2. eskenarcokgen sinifini test etmek amaciyla bir JFrame programi yaziniz ve ekrana üçken çizdiriniz.

### 13.

**H6O3**

grafik sinifindaki:

### public abstract void drawRect(int x1,int y1, int en,int yükseklik)

metotundan yararlanarak H6O3 sinifinda

### public static void drawAciliDortgen(Graphics g, int x1,int y1, int en,int yükseklik,double aci)

metotunu yaziniz. Bu metot dikdortgeni verilen açi kadar bir açiyla cizdirsin. x1,y1 : dörtgenin sag üst köse koordinatlari

en,yükseklik : dörtgenin eni ve yüksekligi

aci: dörgenin radyan cinsinden x1,y1 noktasi etrafinda dönme miktari. not :polar sinifi ve drawPolygon metotlarindan yararlanabilirsiniz.

### Program 6.23 açili dikdortgen çizdiren dikdortgen sinifi

Import java.io.\*; import java.awt.\*;

import java.applet.Applet; import polar;

public class dikdortgen

{

public static void drawdikdortgen(Graphics g,int xi,int yi,int en,int boy,double aci )

{

double R=boy; double r=en;

polar P1=new polar();

polar P2=new polar(); polar P3=new polar(); polar P4=new polar();

int x[]=new int[5]; int y[]=new int[5];

int k=0;

//P1.polarGir(R,teta1); x[0]=xi;

y[0]=yi;

double teta1=aci;

P2.polarGir(r,teta1); x[1]=x[0]+(int)P2.xi();

y[1]=y[0]-(int)P2.xj();

teta1+=Math.PI/2; P3.polarGir(R,teta1); x[2]=x[1]+(int)P3.xi();

y[2]=y[1]-(int)P3.xj();

teta1+=Math.PI/2; P4.polarGir(r,teta1); x[3]=x[2]+(int)P4.xi();

y[3]=y[2]-(int)P4.xj();

x[4]=xi;

y[4]=yi; g.drawPolygon(x,y,4);

}

public static void filldikdortgen(Graphics g,int xi,int yi,int en,int boy,double aci )

{

double R=boy; double r=en;

polar P1=new polar(); polar P2=new polar(); polar P3=new polar(); polar P4=new polar();

int x[]=new int[5]; int y[]=new int[5];

int k=0; x[0]=xi;

y[0]=yi;

double teta1=aci;

P2.polarGir(r,teta1); x[1]=x[0]+(int)P2.xi();

y[1]=y[0]-(int)P2.xj();

teta1+=Math.PI/2; P3.polarGir(R,teta1); x[2]=x[1]+(int)P3.xi();

y[2]=y[1]-(int)P3.xj();

teta1+=Math.PI/2; P4.polarGir(r,teta1); x[3]=x[2]+(int)P4.xi();

y[3]=y[2]-(int)P4.xj();

x[4]=xi;

y[4]=yi; g.fillPolygon(x,y,4);

}

}

### Program 6.24 : dikdortgen sinifini kullanan H6O2.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import dikdortgen; import renk;

public class H6O2 extends Applet implements ActionListener

{

int en,boy,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk

Label kutu1; Label kutu2; Label kutu3; Label kutu4; Label kutu5;

Label bos1;

Label bos2;

Label bos3;

Label renk1;

Label kirmizi;

Label mavi;

Label yesil;

TextField kutugir1;

TextField kutugir2;

TextField kutugir3;

TextField kutugir4;

TextField kutugir5;

TextField kutukirmizi;

TextField kutumavi;

TextField kutuyesil;

public void init(){

//not : init metotunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. Panel p=new Panel();

p.setLayout(new GridLayout(5,4));

//5\*4 tablo p paneli haz• rland•

kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("eni giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yuksekligi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

bos1=new Label(" "); bos2=new Label(" "); bos3=new Label(" ");

renk1=new Label("renk"); kirmizi=new Label("kirmizi"); mavi=new Label("mavi"); yesil=new Label("yesil"); kutukirmizi=new TextField(4); kutumavi=new TextField(4); kutuyesil=new TextField(4);

xi1=300; yi1=300; en=150;

boy=100; yaricap1=(int)Math.sqrt(100\*100+150\*150); aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+en); kutugir4.setText(""+boy); kutugir5.setText(""+aci1); kutukirmizi.setText("255"); kutumavi.setText("200"); kutuyesil.setText("0"); renk=new renk(255,200,0); p.add(bos1);

p.add(bos2);

p.add(bos3); p.add(kirmizi); p.add(yesil);

p.add(mavi);

p.add(renk1); p.add(kutukirmizi);

p.add(kutumavi); p.add(kutuyesil);

//elemanlar p paneline p paneli de aplete eklendi add(p,BorderLayout.NORTH);

kutukirmizi.addActionListener(this); kutumavi.addActionListener(this); kutuyesil.addActionListener(this); kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer eni=new Integer(kutugir3.getText()); en=eni.intValue();

Integer boyi=new Integer(kutugir4.getText()); boy=boyi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

Integer Kirmizii=new Integer(kutukirmizi.getText()); int kirmizi1=Kirmizii.intValue();

Integer Mavii=new Integer(kutumavi.getText()); int mavi1=Mavii.intValue();

Integer Yesili=new Integer(kutuyesil.getText()); int yesil1=Yesili.intValue();

renk=new renk(kirmizi1,mavi1,yesil1); repaint();

}

public void paint(Graphics g)

{

g.setColor(renk); dikdortgen.filldikdortgen(g,xi1,yi1,en,boy,(aci1/180.0\*Math.PI)); g.setColor(renk.siyah); dikdortgen.drawdikdortgen(g,xi1,yi1,en,boy,(aci1/180.0\*Math.PI));

}

}

### BÖLÜM 7: GRAFIK APPLET VE ÇERÇEVE OLUSTURMA OLUSTURMA METOTLARI, GRAFIK KULLANICISI ARABIRIM PROGRAMLARI (GUI) ,

* 1. **GRAPHIC KULLANICISI ARABIRIM PROGRAMLARI, GUI, (GRAPHICS USER INTERFACE)**

Grafik kullanici arabirim programlari, GUI, programiniza kullanicinin daha iyi anliyabilecegi bir görünüs verme ve ayni zamanda grafik ekrandan girdi cve çikti yapabilme amaciyla olusturulur. Kullaniciya sunulan programin onun rahatlikla anlayabilecegi bir formata sahip olmasi günümüz programlamaciliginda oldukça önemlidir. Reklamlarla büyüyen bir kusak için paket bazen paketin içindekinden bile önemli hale gelebilmektedir. Grafik Kullanicisi Arabirim programlari kullaniciya sundugumuz paketi olusturmaktadir.

Java’da temel grafik arabirim kütüphaneleri **java.awt** sinifinda yer almaktadir**. Java 1.2** den itibare awt kütüphanesine ek olarak java swing grafik arabirim kütüphanesi **javax.swing** tanimlanmistir. Bu bölümde awt ve swing yapilarini paralel olarak vermeye çalisacagiz.. Yalniz swing kütüphanesi kapsam olarak çok daha genis oldugundan swing ile yapilan tüm islemlerin ve grafik arabirim alt programlarini awt’de karsiliklari mevcut degildir. Bir önceki grafik çiziminde oldugu gibi grafik arabirim elemanlarinin tanitimi da oldukça genis bir yelpaze olusturduguundan mümkün oldugunca detay verilmeye çalisilsa bile tüm kapsami bu kitapta islemek mümkün degildir. Örnegin java swing konusundaki kapsamli bir kitap olan “Java Swing , O’Reily basim evi” kitabi yaklasik 1200 sayfalik bir kitabi sadece bu konuya ayirmistir.

GUI metotlari **java.awt** (Abstract windowing toolkit) paketinde yer almaktadir. Bu pakette **Container** ve **Component** isimli iki alt pakette yer almaktadir. Her zaman kullandigimiz **Applet** sinifi Component sinifindan türiyen **Panel** sinifinin alt sinifidir. Yani Component Appletlerin süper sinifidir.

Burada Component sinifinin

 TextComponent

 TextField

 TextArea

 Button

 Label

 Checkbox

 List

 Choice

 Canvas

 Scrollbar

 Container

 Panel

Java.applet.Applet sinifinin

 ScrollPane

 Window

 Frame

 Dialog

Siniflarini incelemeye calisacagiz.

Paralel olarak **javax.swing** paketinde yer alan grafik arayüz programlarinda yer alan üst seviye paketler **Japplet, JDialog, JFrame,JWindow** ve **JComponent**’dir

**JComponent** sinifinin altinda yer alan siniflarin bazilari :

 JcomboBox

 JLabel

 JList

 JMenuBar

 JPanel

 JPopupMenu

 JScrollBar

 JScrollPane

 JTable

 JTree

 JInternalFrame

 JOptionPane

 JProgressBar

 JRootPane

 JSeperator

 JSlider

 JSplitPane

 JTabbedPane

 JToolBar

 JToolTip

 JViewePort

 JColorChooser

 JTextComponent

 JTextArea

 JTextField

 JPasswordField

 JEditorPAne

 JTextPane

 JFileChooser

 JLayeredPane

 AbstractButton

 JToggleButton

 JCheckBox

 JRadioButton

 JButton

 JMenuItem

 JMenu

 JRadioButtonMenuItem

 JCheckButtonMenuItem

Yine yineleyelim**, swing** sinifindaki tüm alt metodlari kapsamak bu kitabin kapsamini oldukça büyütecektir, bu yüzden simdilik buna imkan göremiyoruz, fakat pratikte ne olduklari hakkinda iyi bir fikir verebilecek genislikte bir spektrum vermeye çalisacagiz. Burada özellkle awt ve swing sinifi grafik islemcilerini bir arada vermeye çalistik. Kullanici her iki tip grafik ortamiyla da karsilasacagi için (Kendi yazmasa bile hazir programlar karsisina çikacaktir), her ikisini de bilmesi gereklidir. Zaten bu kitabin tüm konularinda da hem awt hem de swing ile yazilmis program örnekleri vermeye çalistik. Bu konudan sonra geriye dönüp tüm program örneklerini grafik ortami gözüyle inceliyebilirsiniz.

### AWT LABEL (ETIKET) SINIFI , JAVAX JLABEL VE ICON SINIFLARI

Label sinifi Appletin içine sadece okunabilen fakat degistirilemeyen yazilar yazma amaciyla kullanilirlar. **Label**

sinifinda tanimlanan metotlar sunlardir:

 Kurucu Metotlar :

**public Label()** // Label sinifi nesneyi olusturur herhangi bir yazi göstermez

**public Label(String s)** // Label sinifi nesneyi olusturur ve s stringini gösterir

**public void Label(String s,int pozisyon) //** Label sinifi nesneyi olusturur ve s stringi gösterir ayni zamanda Labeli pozisyonda verilen degere göre istenilen yere yerlestirir.

Yerlestirme konumlari :**Label.LEFT,Label.CENTER,Label.RIGHT** (sol,orta,sag) konumlaridir.

 Giris çikis metotlari

**public String getText()** // Label sinifinin String degiskenini okur.

**public void setText(String s) //** Label sinifinin String degiskenine yeni deger girer.

### public void setAlignment(int pozisyon)

Label pozisyon degiskeninde verilen degere göre istenilen yere yerlestirir.

Yerlestirme konumlari :**Label.LEFT,Label.CENTER,Label.RIGHT** (sol,orta,sag) konumlaridir.

Örnek olarak bir bir program verelim :

**Program 7.1 : LabelTesti.java** import java.applet.Applet; import java.awt.\*;

public class LabelTesti extends Applet

{

private Label L1,L2; public void init()

{

L1=new Label();

L1.setText("Bu etiket once bos tanimlandi"); add(L1);

L2=new Label("Bu etiket yazisi ilk tanimda tanimlandi"); add(L2);

}

public void paint(Graphics g)

{

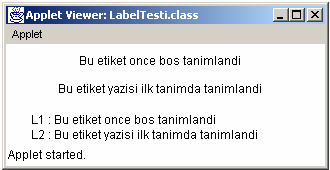
g.drawString("L1 : "+L1.getText(),25,80);

g.drawString("L2 : "+L2.getText(),25,95);

}

}

### 07001.JPG



**Sekil 7.1 LabelTesti.java appleti**

LabelTesti programinda kullanilan add metotu Label’i Container sinifina(Appletin alt sinifi oldugu) baglar, diger deyimle applete veya verilen üst komponente ilave eder.

**Swing** sinifi **Jlabel** Lable sinifindandan farkli olarak bize sadece yazi degil ayni zamanda görüntü ekleme olasiliklari da sunar. Temel sinif olarak kurucu metodlari :

### JLabel() JLabel(Icon Resim)

**Jlabel(Icon Resim,int yazi\_pozisyonu) Jlabel(String etiket)**

**Jlabel(String etiket, Icon Resim)**

**Jlabel(String etiket, Icon Resim, int yazi\_pozisyonu) Jlabel(String etiket, int yatay\_yazi\_pozisyonu)** Buradaki yatay yazi pozisyonu

### JLabel.LEFT JLabel .CENTER JLabel.RIGHT Veya

**SwingConstants.LEFT SwingConstants.CENTER SwingConstants.RIGHT**

Degerlerinden birini alabilir.

Jlabel sinifinin çok kullanacagimiz bazi metotlarini sayacak olursak : Jcomponent sinifindan gelen metodlar :

**void setBackground(Color c**) : arkaplan rengini degistiri **Color getBackground()** : arkaplan rengini okur **void setForeground(Color c)** : önpaln rengini degistiri **Color getForeground()** : ön plan rengini okur. **void setFont(Font f)** : yazi fontunu seçer

**Font getFont()** : yazi fontunu okur

**Boolean isVisible** : görünür olup olmadigini kontrol eder

**Void setVisible(Boolean b)** : görünür veya görünmez yapar. Jlabel sinifinda tanimlanmis bazi alt siniflar :

**void setText(String s) : JLabel**’in text degiskeninin degerine yeni verilen degeri atar

**String getText()** : JLabel’in String text degiskenini okur.

**void setVerticalAlignment(int düsey\_pozisyon) :** Label’in Pencere içindeki düsey pozisyonunu ayarlar Buradaki düsey pozisyon

### JLabel.TOP JLabel.CENTER JLabel.BOTTOM Veya SwingConstants.TOP

**SwingConstants.CENTER SwingConstants.BOTTOM** Degerlerinden birini alabilir. **int getVerticalAlignment()**

**void setVerticalTextPosition(int düsey\_pozisyon) :** Label’in içindeki yazinin Label içindeki düsey pozisyonunu ayarlar. Sabitler üsttekilerin aynidir.

**int getVerticalTextPosition() :** Label’in içindeki yazinin Label içindeki düsey pozisyonunu okur. **void setHorizontalTextPosition(int düsey\_pozisyon) :** Label’in içindeki yazinin Label içindeki yatay pozisyonunu ayarlar. Sabitler üsttekilerin aynidir(LEFT,CENTER,RIGHT).

**int getHorizontallTextPosition() :** Label’in içindeki yazinin Label içindeki yatay pozisyonunu okur.

**void setIcon(Icon resim) :** Label içindeki resmi degistirir veya eger yoksa ilk defa atar

**Icon getIcon() :** Label içindeki resmi okur.

**void setDisplayedMnemonic(int c)** : Mnemoic alt harf kontrolunu kullanarak belli bir girise ulasmayi saglar. (örnegin ‘m’ harfine set edilmisse alt-m bu komuta gider. Örnek problemlerle bu kavram açilacaktir.

### int getDisplayedMnemonic(int c) : Mnemonic degerini okur.

**Simdi örnek problemlerle swing JLabel kullanimina daha detayli olarak bakalim.**

Birinci örnegimizde basit bir sekilde String label yazdiracagiz. Çerçeve olarak JFrame kullandik.

### Program 7.2 : Label1SW.java swing java Jlabel örnegi

import javax.swing.\*; public class label1SW

{

public static void main(String[] args)

{

JLabel etiket=new JLabel("JLabel sinifina hos geldiniz"); JFrame cerceve=new JFrame(); cerceve.addWindowListener(new BasicWindowMonitor()); cerceve.getContentPane().add(etiket);

cerceve.pack(); cerceve.setVisible(true);

}

}

### 07002.JPG



**Sekil 7.2 Label1SW.java JFrame penceresi**

Ikinci örnegimizde resimli, hem resim hem yazili ve yalniz yazili bir JApplet programi olsturduk. Bu programin ayni zamanda main metodu da oldugundan hem applet hem de frame olarak çagirilmasi mümkündür.

Çerçeve olarak JFrame kullandik.

### Program 7.3 LabelTestiSW Japplet türü JLabel örnek programi

Import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class LabelTestiSW extends JApplet {

protected String iconname1="images/kus.gif"; ImageIcon icon1;

protected String iconname2="images/köpek.gif"; ImageIcon icon2;

protected JLabel j1,j2,j3;

public LabelTestiSW()

{

getRootPane().putClientProperty("defeatSystemEventQueueCheck", Boolean.TRUE);

}

public void init()

{

icon1=new ImageIcon(iconname1); icon2=new ImageIcon(iconname2); j1 = new JLabel(icon1);

j2 = new JLabel("Label (Etiket) yazisi",icon2,JLabel.CENTER); j2.setVerticalTextPosition(JLabel.BOTTOM); j2.setHorizontalTextPosition(JLabel.CENTER);

j3 = new JLabel("Font 16 mavi Label"); Font font=j3.getFont();

j3.setFont(new Font(font.getName(),Font.BOLD,16)); j3.setForeground(Color.blue);

JPanel pane=new JPanel(); JPanel pane1=new JPanel(); JPanel pane2=new JPanel(); pane1.add(j1); pane1.add(j2); pane1.add(j3);

pane1.setBackground(new Color(255,255,204)); pane1.setForeground(Color.black); pane1.setBorder(BorderFactory.createMatteBorder(1,1,2,2,Color.black)); pane.setLayout(new BorderLayout()); pane.add(pane1,BorderLayout.NORTH);

setContentPane(pane);

}

public static void main(String s[]) { JFrame f = new JFrame("Label testi ");

f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

JApplet applet = new LabelTestiSW(); f.getContentPane().add("Center", applet); applet.init();

f.pack();

f.setSize(new Dimension(550,200)); f.show();

}

}

### 07003.JPG



**Sekil 7.3 LabelTestiSW.java JFrame penceresi**

**Program 7.4 LabelDemo JPanel türü JLabel örnek programi**

import java.awt.GridLayout; import java.awt.BorderLayout;

import java.awt.event.WindowListener; import java.awt.event.WindowAdapter; import java.awt.event.WindowEvent; import javax.swing.JLabel;

import javax.swing.JPanel; import javax.swing.JFrame; import javax.swing.ImageIcon;

public class LabelDemo extends JPanel { JLabel label1, label2, label3;

public LabelDemo() {

ImageIcon icon = new ImageIcon("images/middle.gif",

"bir damla resmi"); setLayout(new GridLayout(3,1)); //3 satir, 1 sütun label1 = new JLabel("Resim ve Yazi",

icon, JLabel.CENTER);

//resme göreceli olarak yazi pozisyonunu ayarla: label1.setVerticalTextPosition(JLabel.BOTTOM); label1.setHorizontalTextPosition(JLabel.CENTER); label2 = new JLabel("Sadece yazili Label (Etiket)"); label3 = new JLabel(icon);

//Add labels to the JBufferedPane. add(label1);

add(label2); add(label3);

}

public static void main(String[] args) {

/\*

* Create a window. Use JFrame since this window will include
* lightweight components.

\*/

JFrame frame = new JFrame("LabelDemo");

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

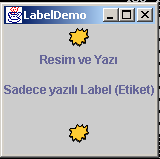
frame.getContentPane().add(new LabelDemo(), BorderLayout.CENTER); frame.pack();

frame.setVisible(true);

}

}

### 07004.JPG



**Sekil 7.4 LabelDemo.java JPanel penceresi**

Bu programda resim

### ImageIcon icon = new ImageIcon("images/middle.gif","bir damla resmi");

Tanimiyla ImageIcon sinifi araciligiyla resmi Jlabel’a aktardi. **images/middle.gif** middle.gif dosyasinin images alt direktorysinde yer aldigini belirtir.

Burada Icon interphase’ine biraz daha yakindan göz atalim. Icon interphase sinifi oldugundan tanimlanan sinifi içerisinde

### public int getIconWidth() public int getIconHeight()

**public void paintIcon(Component c, Graphics g, int x,int y)**

metodlari tanimlanmak zorundadir. Simdi Icon sinifinin alt sinifi olan OvalIcon sinifini tanimlayalim

### Program 7.5 Jlabel’da kullanilacak Icon olusturan OvalIcon sinifi

import javax.swing.\*; import java.awt.geom.\*; import java.awt.\*;

public class OvalIcon implements Icon

{

int genislik, yukseklik; public OvalIcon(int w,int h)

{

genislik=w; yukseklik=h;

}

public void paintIcon(Component c,Graphics g,int x,int y)

{

Graphics2D g2=(Graphics2D)g; g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

Ellipse2D elips1=new Ellipse2D.Double(x,y,genislik-1,yukseklik-1); g2.draw(elips1);

}

public int getIconWidth() {return genislik;} public int getIconHeight() {return yukseklik;}

}

Simdi de OvalIcon sinifini kullanan JLabel siniflarini olusturup gösterelim :

### Program 7.6 OvalIcon sinifini kullanarak JLabel’da daire Icon’u (resmi) gösteren label2SW programi

import javax.swing.\*; import java.awt.\*;

public class label2SW

{

public static void main(String[] args)

{

JFrame f=new JFrame();

f.addWindowListener(new BasicWindowMonitor()); JLabel etiket1=new JLabel(new OvalIcon(20,50)); JLabel etiket2=new JLabel(new OvalIcon(50,20));

JLabel etiket3=new JLabel("daire",new OvalIcon(60,60),SwingConstants.CENTER); etiket3.setHorizontalTextPosition(SwingConstants.CENTER);

Container c=f.getContentPane(); c.setLayout(new FlowLayout()); c.add(etiket1);

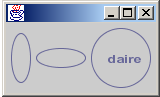
c.add(etiket2); c.add(etiket3); f.pack(); f.setVisible(true);

}

}

Bu sinifin çiktisi :

### 07005.JPG



**Sekil 7.5 label2SW.java JFrame penceresi**

Buradan da görüldügü gibi JLabel sinifini çok daha genis bir platformda, olusturdugumuz bir grafigin veya çizimin sonuçlarini grafik ortama aktarmak amaciyla da kullanabiliriz.

### BUTTON VE JBUTTON (DÜGME) SINIFLARI

Button (dügme) basilinca belirli islemlerin yapilmasini baslatan bir siniftir. Graphic arabirim kullanim (GUI) programlarini veya aplet programlarini konsol programlarindan ayiran en önemli özellik, GUI lerin islem kökenli olmasidir. Diger bir deyimle kullanicinin baslatacagi islemleri yapar ve kullanicinin yeni bir islem tanimlamasini beklerler. Bu islemleri yapan siniflar **java.awt.event** kütüphanesinde tanimlanmistir. Bu kütüphanede en çok kullanacagimiz siniflar hiyerarsi düzeninde söyle verilebilir :

### Java.util.EventObject

 Java.awt.AWTEvent

 ActionEvent // Bir dügme(Button) basildiginda,Listeden (List) iki klikle bir seçim yapildiginda veya menuden seçim yapildiginda kullanilir.

 AdjustmentEvent // asagi yukari kaydirma cubugunda (scroll bar) bir islem yapildiginda kullanilir.

 ItemEvent // List veya CheckBox kliklendiginde (bir kere) kullanilir

 ComponentEvent //appletteki elemanlarin gizlenmesi, boylarinin degistirilmesi, veya degisik yerlere alinmasi amaciyla kullanilir.

 ContainerEvent //Containera(applete) yeni bir eleman eklenilir veya çikarilirken kullanilir.

 FocusEvent//herhangi bir applet elemani kullanilmaya baslandiginda on plana cikarilmakk, ve kullanilmadiginda arka plana atilmak istendiginde kullanilir.

 PaintEvent

 WindowEvent //pencere acilip kapandiginda,kücültülüp büyütüldügünde vb kullanilir.

 InputEvent

 KeyEvent //keyboarddan bir girdi alindiginda kullanilir

 MouseEvent //mouse dan bir girdi alindiginda kullanilir

Kullanici bir applet veya pencere tipi baska Container programi yazarken dinleme ile ilgili iki islem yapar ilk islem herhangi bir GUI dan bir girdi olup olmadigini dinlemek olduysa kaydetmek ikincisi ise bu islemin sebep olacagi islemleri olusturmaktir. Dinleme islemini yapan metotlar sunlardir :

### Java.util.EventListener

 ActionListener

 AdjustmentListener

 ComponentListener

 ContainerListener

 FocusListener

 ItemListener

 KeyListener

 MouseListener

 MouseMotionListener

 TextListener

 WindowListener

Bu metotlarin bir çogunu ilgili örneklerde yeri geldiginde kullanacagiz. Simdi ayni sonucu veren iki button ornegi ile bu kavrami biraz daha açmaya çalisalim :

### Program 7.7 : ButtonTesti.java, dügme kullanim testi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class ButtonTesti extends Applet

{

private Button B1,B2; public void init()

{

B1=new Button("B1"); B1.addActionListener(new B1Basilinca(this)); add(B1);

B2=new Button("B2"); B2.addActionListener(new B2Basilinca(this)); add(B2);

}

}

class B1Basilinca implements ActionListener

{

Applet applet;

public B1Basilinca(Applet a) {applet=a;} public void actionPerformed(ActionEvent e)

{

applet.showStatus(" "+e.getActionCommand()+"dugmesine basildi");

}

}

class B2Basilinca implements ActionListener

{

Applet applet;

public B2Basilinca(Applet a) {applet=a;} public void actionPerformed(ActionEvent e)

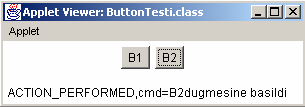
{

applet.showStatus(" "+e.paramString()+"dugmesine basildi");

}

}

### 07006JPG



**Sekil 7.6 ButtonTesti.java sinifi sonuçlarinin appletde görünmesi**

**Problem 7.8 ButtonTesti1.java sinifi (sonuclar ButtonTesti.java ile aynidir)**

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class ButtonTesti1 extends Applet implements ActionListener

{

private Button B1,B2; public void init()

{

B1=new Button("B1"); B1.addActionListener(this); add(B1);

B2=new Button("B2"); B2.addActionListener(this); add(B2);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==B1)

{showStatus(" "+e.getActionCommand()+"dugmesine basildi");} else if(e.getSource()==B2)

{showStatus(" "+e.paramString()+"dugmesine basildi");}

}

}

Birinci programda dügmelere basildiginda olusan eylemler (showStatus kösesine basilan dügmeyi yazdirma) ayri siniflarda yaptirilmis ve Applet sinifi üzerinden bizim ButtonTesti sinifina aktarilmistir. Ikinci programda ise ayni islemler tek bir sinifin içerisinde yer almistir.

Programdan da görülecegi gibi dügme basma eylemi ActionEvent sinifi üzerinden aktarilmistir.

**JButton** sinifi ile Button sinifi arasinda kullanim açisindan Label ve Jlabel sinifinin arasindakilere benzer farklar vardir.

**JButton** sinifinin sinif tanimi :

### public class JButton extends AbstractButton implements Accessible

**{**

**public JButton()**

**public JButton(Icon resim) public JButton(String yazi);**

**public JButton(String yazi, Icon resim)**

**public AccessibleContext getAccessibleContext(); public String getUIClassID();**

**public boolean isDefaultButton(); public boolean isDefaultCapable();**

**public void setDefaultCapable(boolean b); pubkic void updateUI();**

**}**

seklindedir. Görüldügü gibi JButton fazla metod barindirmaz bir çok önemli metodu Abstract button ve onun tepe sinifi JComponent siniflarindan alir. JButton sinifinda Jlabel sinifinda da kullandigimiz JComponent sinifindan gelen metodlar :

**public void setBackground(Color c**) : arkaplan rengini degistiri **public Color getBackground()** : arkaplan rengini okur **public void setForeground(Color c)** : önpaln rengini degistiri **public Color getForeground()** : ön plan rengini okur. **public void setFont(Font f)** : yazi fontunu seçer

**public Font getFont()** : yazi fontunu okur

**public Boolean isVisible()** : görünür olup olmadigini kontrol eder

**public Void setVisible(Boolean b)** : görünür veya görünmez yapar. AbstractButton sinifinda tanimlanmis bazi alt siniflar :

**public void setText(String s) : JLabel**’in text degiskeninin degerine yeni verilen degeri atar

**public String getText()** :AbstractButton’in String text degiskenini okur.

**public void setVerticalAlignment(int düsey\_pozisyon)** : AbstractButton’in Pencere içindeki düsey pozisyonunu ayarlar

Buradaki düsey pozisyon **SwingConstants.TOP SwingConstants.CENTER SwingConstants.BOTTOM** Degerlerinden birini alabilir. **public int getVerticalAlignment()**

**public void setVerticalTextPosition(int düsey\_pozisyon) :** AbstractButton‘in içindeki yazinin Label içindeki düsey pozisyonunu ayarlar. Sabitler üsttekilerin aynidir.

**public int getVerticalTextPosition() :** AbstractButton‘in içindeki yazinin AbstractButton içindeki düsey pozisyonunu okur.

public void setHorizontalTextPosition(int düsey\_pozisyon) : AbstractButton‘in içindeki yazinin AbstractButton içindeki yatay pozisyonunu ayarlar. Sabitler

### SwingConstants.LEFT SwingConstants.CENTER SwingConstants.RIGHT

**public int getHorizontallTextPosition() :** AbstractButton‘in içindeki yazinin AbstractButton‘ içindeki yatay pozisyonunu okur.

**Public void setIcon(Icon resim) :** AbstractButton‘in içindeki resmi degistirir veya eger yoksa ilk defa atar

**public Icon getIcon() :** AbstractButton içindeki resmi okur.

**public void setDisplayedMnemonic(int c)** : Mnemoic alt harf kontrolunu kullanarak belli bir girise ulasmayi saglar. (örnegin ‘m’ harfine set edilmisse alt-m bu komuta gider. Örnek problemlerle bu kavram açilacaktir.

**Public int getDisplayedMnemonic(int c) :** Mnemonic degerini okur.

Listener (pencere dinleme) metod ve neslerinden ise : **protected ActionListener actionListener; protected Itemlistener itemListener**

### protected ChangeListener changeListener public void addActionListener(ActionListener l)

**public void addChangeListener(ChangeListener l) public void addItemListener(ItemListener l)**

**public void removeActionListener(ActionListener l) public void removeChangeListener(ChangeListener l)**

**public void removeItemListener(ItemListener l) public String getActionCommand()**

**public void setActionCommand(String com)**

gibi metodlar mevcuttur. Simdi örnekler üzerinden JButton sinifini kullanmayi deneyelim :

### Program 7.9 Jbutton sinifinin gösterilmesini örnekleyen ButtonTestiSWF programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class ButtonTestiSWF extends JFrame implements ActionListener

{

private JButton B1,B2; private JTextArea cikti;

public ButtonTestiSWF()

{

super("button örnegi"); Container c=getContentPane(); c.setLayout(new FlowLayout());

ImageIcon icon1 = new ImageIcon("images/middle.gif"); B1=new JButton("B1",icon1); B1.setHorizontalTextPosition(SwingConstants.LEFT); B1.addActionListener(this);

c.add(B1);

B2=new JButton("B2",icon1); B2.addActionListener(this); c.add(B2);

cikti=new JTextArea(" "); c.add(cikti);

cikti.setBackground(c.getBackground());

}

public static void main(String[] args)

{

ButtonTestiSWF pencere= new ButtonTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

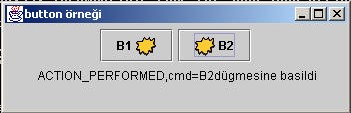
String gs=""; if(e.getSource()==B1)

gs = " "+e.getActionCommand()+"dügmesine basildi\n"; else if(e.getSource()==B2)

gs = " "+e.paramString() +"dügmesine basildi"; cikti.setText(gs);

}

} 07007.JPG



### Sekil 7. ButtonTestiSWF.java sonuçlarinin JFrame penceresinde görüntüsü

Button ve JButton arasindaki farklari görmek amaciyla birbirinin ayni olan ButtonDemoApplet.java awt applet programini, ButtonDemoApplet.java swing programiyla karsilastiralim. Not :isimler ayni oldugundan bu iki programi ayni dosyalarda sakliyamayiz.

### Program 7.10 Button sinifinin gösterilmesini örnekleyen ButtonDemoApplet programi

import java.awt.\*;

import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import java.applet.Applet;

public class ButtonDemoApplet extends Applet

implements ActionListener {

protected Button b1, b2, b3;

protected static final String DISABLE = "disable"; protected static final String ENABLE = "enable";

public void init() { b1 = new Button();

b1.setLabel("Disable middle button"); b1.setActionCommand(DISABLE);

b2 = new Button("Middle button");

b3 = new Button("Enable middle button"); b3.setEnabled(false); b3.setActionCommand(ENABLE);

//Listen for actions on buttons 1 and 3. b1.addActionListener(this); b3.addActionListener(this);

//Add Components to the Applet, using the default FlowLayout. add(b1);

add(b2);

add(b3);

}

public void actionPerformed(ActionEvent e) {

if (e.getActionCommand().equals(DISABLE)) { b2.setEnabled(false);

b1.setEnabled(false); b3.setEnabled(true);

} else {

b2.setEnabled(true); b1.setEnabled(true); b3.setEnabled(false);

}

}

}

### 07008.JPG



**Sekil 8. ButtonDemoApplet.java sonuçlarinin AppletViewer browserinda görüntüsü**

**Program 7.11 Button sinifinin gösterilmesini örnekleyen ButtonDemoApplet programi (bu program 7.10 ile ayni islevi yapmaktadir)**

/\*

\* Swing version.

\*/

import javax.swing.\*;

import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import java.awt.BorderLayout; import java.net.URL;

public class ButtonDemoApplet extends JApplet

implements ActionListener { protected JButton b1, b2, b3;

protected static final String DISABLE = "disable"; protected static final String ENABLE = "enable";

protected String leftButtonFilename = "images/right.gif"; protected String middleButtonFilename = "images/middle.gif"; protected String rightButtonFilename = "images/left.gif";

public void init() {

ImageIcon leftButtonIcon = new ImageIcon(

getURL(leftButtonFilename)); ImageIcon middleButtonIcon = new ImageIcon(

getURL(middleButtonFilename)); ImageIcon rightButtonIcon = new ImageIcon(

getURL(rightButtonFilename));

b1 = new JButton("Disable middle button", leftButtonIcon); b1.setVerticalTextPosition(AbstractButton.CENTER); b1.setHorizontalTextPosition(AbstractButton.LEFT); b1.setMnemonic('d');

b1.setActionCommand(DISABLE);

b2 = new JButton("Middle button", middleButtonIcon); b2.setVerticalTextPosition(AbstractButton.BOTTOM); b2.setHorizontalTextPosition(AbstractButton.CENTER); b2.setMnemonic('m');

b3 = new JButton("Enable middle button", rightButtonIcon);

//Use the default text position of CENTER, RIGHT. b3.setMnemonic('e'); b3.setActionCommand(ENABLE); b3.setEnabled(false);

//Listen for actions on buttons 1 and 3.

b1.addActionListener(this); b3.addActionListener(this);

//Add Components to a JPanel, using the default FlowLayout. JPanel pane = new JPanel();

pane.add(b1);

pane.add(b2);

pane.add(b3);

//Add JPanel to this applet, using the default BorderLayout. getContentPane().add(pane, BorderLayout.CENTER);

}

public void actionPerformed(java.awt.event.ActionEvent e) { if (e.getActionCommand().equals(DISABLE)) {

b2.setEnabled(false); b1.setEnabled(false); b3.setEnabled(true);

} else {

b2.setEnabled(true); b1.setEnabled(true); b3.setEnabled(false);

}

}

protected URL getURL(String filename) { URL codeBase = getCodeBase();

URL url = null;

try {

url = new URL(codeBase, filename);

} catch (java.net.MalformedURLException e) { System.out.println("Couldn't create image: badly specified URL"); return null;

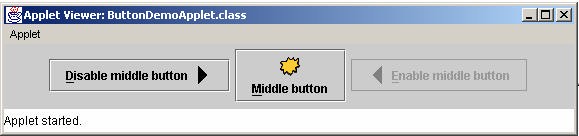
}

return url;

}

}

07009.JPG



### Sekil 9. ButtonDemoApplet.java sonuçlarinin swing JApplet olarak Appletviewer browserinda görüntüsü

Programdan da görüldügü gibi bu program bir önce verilmis olan programin swing frame benzeridir. Iki program da Applet olarak verilmistir. Programlarda setEnabled(Boolean b) metodu kullanilarak dügmenin çalismasi etkisisz hale getirilmistir. Ayrica swing versiyonunda resim dosyasini kullanirken URL sinifi üzerinden okuduk. URL network ortaminda bilgi aktarmaya yarayan bir siniftir. Resim dosyasi baska bir bilgisayarda olabilirdi.

### TEXTFIELD (YAZIM ALANI) SINIFI

TextField bir satirlik yazim alanlaridir. TextField’in metotlari asagida verilmistir :

###  Kurucu Metotlar :

**public TextField()**

**public TextField(int satirsayisi)** // satir sayisi uzunlugunda bir yazi kutusu açar.

**public TextField(String s)** // s uzunlugunda bir yazi kutusu açr ve s yi içine yazar

**public TextField(String s,int satirsayisi) )** // satir sayisi uzunlugunda bir yazi kutusu açr ve s yi içine yazar

###  Diger Metotlar:

**Public void setEchoChar(char c)** // kutudaki gerçek yaziyi gizleyerek onun yerine c karacter degiskenini yazar.

**Public void setEditable(boolean b**) //kutunun içine yazilan yazinin degistirileblmesini saglar veya engeller

b=true yazi yazilabilir b=false yazi yazilamaz.

Bu metotlarin kullanimini yine bir program örnegiyle izleyelim :

### Program 7.12 TextFieldTesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class TextFieldTesti extends Applet implements ActionListener

{

private TextField T1,T2,T3,T4; public void init()

{

setBackground(Color.lightGray);

//bos textField T1=new TextField();

T1.addActionListener(this); add(T1);

T2=new TextField("Buraya yazi yaziniz"); T2.addActionListener(this);

add(T2);

T2=new TextField("gizli yazi"); T2.setEchoChar('\*'); T2.addActionListener(this); add(T2);

T3=new TextField("gizli yazi"); T3.setEditable(false);

add(T3);

}

public void actionPerformed(ActionEvent e)

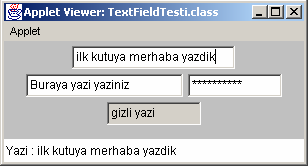
{

showStatus("Yazi : "+e.getActionCommand());

}

}

### 07010.JPG



**Sekil 7.10** TextFieldTesti.java **programi sonuçlarinin appletde görünümü**

Bu programda her yazi alanina (TextField) addActionListener metotu eklenmis ve bir yazi girildiginde actionPerformed metotundaki showStatus metotu yoluyla en son girilen yazi showStatus penceresine aktarilmistir.

TextField sinifinin java swing esdegeri JTextField sinifidir. JTextField sinifi kurucu metotlari :

### JTextField() JTextField(String) JTextField(String, int) JTextField(int)

**JTextField(Document, String, int)**

Seklindedir.

TextFielde Paralel olarak Özel bir TextField türü olan PasswordField de swingde tanimlanmistir. PassWord fieldin kurucu metod tanimlari :

### JPasswordField() JPasswordField(String) JPasswordField(String, int) JPasswordField(int) JPasswordField(Document, String, int) Seklinde tanimlanmistir.

Bu iki sinifin Text degerini girmak veya okumak için

**void setText(String) String getText()** metodlari tanimlanmistir. **void setEditable(boolean) boolean isEditable()**

metodlari Yazi alanindaki yaziyi kullanicinin degistirp degistiremeyacegini saptar veya bu müsadeyi degistirir.

### void setColumns(int); int getColumns()

**int getColumnWidth()**

metodlari yazi alaninin byutunu degistirmek veya saptamak amaciyla kullanilabilir.

### void setHorizontalAlignment(int); int getHorizontalAlignment()

metodlari yazii alanindaki yazinin ne sekilde yerlestirildigini saptar veya istege göre yerlestirir.

JTextField.LEFT, JTextField.CENTER, veya JTextField.RIGHT

### void setEchoChar(char) char getEchoChar()

***JpasswordField*** sinifinda tanimlanmis yazilan yazinin yerine geçecek karekteri tanimlayan veya soran metodlardir.

### void addActionListener(ActionListener)

metodlari ise TextFieldin dinlenmesini ve giris tusu basildiginda yapilacak eylemlerin verildigi ActionListener implement’inden gelen

### public void actionPerformed(ActionEvent e)

metodunu çalistirmaya yarar.

### void removeActionListener(ActionListener)

ise ActionListener dinleme metodunu iptal eder.

JTextField ve JPasswordField metodlarinin kullanilmasini bir örnek problemle inceleyelim. Bu bir önceki problemin hemen hemen ayni islemleri yapan JFrame ortaminda yazilmis bir örnek problemdir.

### Program 7.13 TextFieldTestiSWF.java programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class TextFieldTestiSWF extends JFrame implements ActionListener

{

private JTextField T1,T2,T3; private JPasswordField T4;

Container c;

public TextFieldTestiSWF()

{

super("JTestField ve JPassword Field Testleri"); c=getContentPane();

c.setLayout(new FlowLayout()); c.setBackground(Color.lightGray);

//bos textField

T1=new JTextField(10); T1.addActionListener(this); c.add(T1);

T2=new JTextField("Buraya yazi yaziniz"); T2.addActionListener(this);

c.add(T2);

T3=new JTextField("degistirilemez yazi"); T3.setEditable(false); T3.addActionListener(this);

c.add(T3);

T4=new JPasswordField("gizli yazi"); c.add(T4);

}

public void actionPerformed(ActionEvent e)

{

String s="Kaynak = "; if(e.getSource()==T1)

s+="T1 Yazi : "+e.getActionCommand(); else if(e.getSource()==T2)

s+="T2 Yazi : "+e.getActionCommand(); else if(e.getSource()==T3)

s+="T3 Yazi : "+e.getActionCommand(); else if(e.getSource()==T4)

s+="T4 Yazi : "+e.getActionCommand(); JOptionPane.showMessageDialog(this,s,"JTestField ve JPassword Field Testleri", JOptionPane.PLAIN\_MESSAGE);

}

public static void main(String[] args)

{

TextFieldTestiSWF pencere= new TextFieldTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

}

### 07011.JPG



**Sekil 7.11 TextFieldTestiSWF.java programi sonuçlarinin JFrame ve JoptionPane’de görünümü**

### YAZIM ALANI AWT TEXTAREA VE SWING JTEXTAREA SINIFLARI

awt kütüphanesinde tanimlanan Text Area sinifi yapi olarak TextField sinifina benzer. En onemli farki bir satir yerine birden fazla satir girdi alabilme olasiligidir. TextField ve TextArea siniflari TextComponent sinifinin alt siniflaridir. Bu yüzden üstte belirttigimiz

**Public void setEchoChar(char c)** // kutudaki gerçek yaziyi gizleyerek onun yerine c karacter degiskenini yazar.

**Public void setEditable(boolean b**) //kutunun içine yazilan yazinin degistirileblmesini saglar veya engeller b=true yazi yazilabilir b=false yazi yazilamaz.

Metotlari burada da aynen geçerlidir.TextArea’ninkurucu metotlarida söyledir:

### public TextArea()

**public TextArea(int sütun,int satir)** sütun x satir boyutunda yazi alani tanimlar **public TextArea(String s)**

String degiskeni s nin boyuna göre yazi alani tanimlar ve s yi içine yazar.

### public TextArea(String s,int sütun,int satir)

sütun x satir boyutunda yazi alani tanimlar ve s yi içine yazar

### public TextArea(String s,int sütun,int satir,int scrollbar)

scroll bar (kontrolçubugu) degiskeni su degerleri alabilir :

**SCROLLBARS\_BOTH** Hem yatay hemdüsey kontrol çubugu tanimlar **SCROLLBARS\_HORIZONTAL\_ONLY** Yatay kontrol çubugu tanimlar **SCROLBARS\_NONE** Kontrol çubugu tanimlamaz **SCROLLBARS\_VERTICAL\_ONLY** Düsey kontrol çubugu tanimlar

Simdi TextArea metotunu Küçük bir örnekte kullanalim :

### Program 7.14 TextAreaTesti.java programi

import java.applet.Applet; import java.awt.\*;

public class TextAreaTesti extends Applet

{

private TextArea T1; public void init()

{

setBackground(Color.lightGray);

//bos textField

String s="buraya denemek icin\nyazi yazildi\n\t sonrada bu yazi aktarildi";

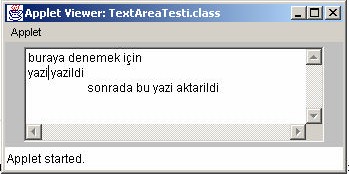
//not \n satirbasi kontrolu

// \t bir sonraki tab setine git T1=new TextArea(s,5,40); add(T1);

}

}

### 07012.JPG



**Sekil 7.12 TextAreaTesti.java Programi applet çiktisi**

Bu programda gördügünüz \n ve \t string komutlaridir. Alt satira geç ve gelecek tab noktasina git anlamina gelir Bu komutlar aslinda c dilinden alinmadir ve c dilindeki tüm \ komutlari java stringlerinde de geçerlidir.

Simdi TextArea metotunu biraz daha detayli bir programda kullanalim

### Program 7.15 : TextArea sinifini kullanan TextAreaTesti1.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class TextAreaTesti1 extends Applet implements ActionListener,TextListener

{

private TextArea T1,T2; private Button kopye; public void init()

{

setBackground(Color.lightGray);

//bos textField

String s="buraya denemek icin\nyazi yazildi\n\t sonrada bu yazi aktarildi";

//not \n sat• rbaŸ• kontrol•

// \t bir sonraki tab setine git

T1=new TextArea(s,5,20,TextArea.SCROLLBARS\_NONE); T1.addTextListener(this);

add(T1);

kopye=new Button("Kopye "); kopye.addActionListener(this); add(kopye);

T2=new TextArea(5,20); T2.setEditable(false); add(T2);

}

public void actionPerformed(ActionEvent e)

{

// T1 penceresinden seçilen yaziyi

// kopye dügmesine basildiginda T2 penceresine aktar T2.setText(T1.getSelectedText());

}

public void textValueChanged(TextEvent e)

{

//Birinci pencereye yeni yazi yazildiginda ikinci pencereye aktar TextComponent kaynak=(TextComponent)(e.getSource()); T2.setText(kaynak.getText());

}

}

### 07013.JPG



**Sekil 7.13 TextAreaTesti1.java Programi sonuçlarini gösteren appletler**

Bu programda **TextListener** kullanildigindan **textValueChange** metotu da kullanilmistir. Buradaki metot birinci kutuya yazilan yaziyi ikinci kutuya kopyalamaktadir.

Kopye dügmesine basilinca ise sadece T1.getSelectedText() metotuyla secilmis olan text actionPerforned metotu üzerinden T2 Yazim Alanina (TextArea) aktarilmaktadir.

Java swing **JTextArea** sinifi awt deki TextArea sinifinin benzeridir. Sinifin kurucu metodlarinin tanimi

### public JTextArea();

**public JTextArea(int rows,int cols); public JTextArea(Document doc);**

**public JTextArea(Document doc, String Text, int satirsayisi int sütun sayisi); public JTextArea(String Text)**

**public JTextArea(String Text,int satirsayisi int sütun sayisi)**

seklindedir.

JTextArea sinifini kullanan Bir örnek problem verelim:

### Program 7.16 TextAreaTesti1SWF.java programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class TextAreaTesti1SWF\_2000 extends JFrame implements ActionListener,TextListener

{

private JTextArea T1,T2; private JButton kopye;

public TextAreaTesti1SWF\_2000()

{

super("JTextAreaTesti"); Container c=getContentPane(); c.setLayout(new FlowLayout()); c.setBackground(Color.lightGray);

//bos textField

String s="buraya denemek icin\nyazi yazildi\n\t sonrada bu yazi aktarildi";

//not \n satirbasi kontrol

// \t bir sonraki tab setine git

Box b=Box.createHorizontalBox(); T1=new JTextArea(s,10,15); b.add(new JScrollPane(T1)); kopye=new JButton("Kopye >>>"); kopye.addActionListener(this); b.add(kopye);

c.add(kopye);

T2=new JTextArea(10,15); T2.setEditable(false); b.add(new JScrollPane(T2)); c.add(b);

}

public void actionPerformed(ActionEvent e)

{

// T1 penceresinden seçilen yaziyi

// kopye dügmesine basildiginda T2 penceresine aktar T2.setText(T1.getSelectedText());

}

public void textValueChanged(TextEvent e)

{

//Birinci pencereye yeni yazi yazildiginda ikinci pencereye aktar TextComponent kaynak=(TextComponent)(e.getSource()); T2.setText(kaynak.getText());

}

public static void main(String[] args)

{

TextAreaTesti1SWF\_2000 pencere= new TextAreaTesti1SWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(600,200);

pencere.setVisible(true);

}

}

### 07014.JPG



**Sekil 7.14 TextAreaTesti1SWF.java programi sonuçlarinin JFrame penceresindede görünümü**

### AWT CHOICE , SWING JCOMBOBOX SEÇIM SINIFLARI

awt kütüphanesi Choice (Seçim) sinifi birden fazla maddeden seçim yapilmasi gerektiginde kullanilan bir siniftir.

Choice sinifinin kurucu metotu ve diger metotlarinin listesi asagida verilmistir :

### public Choice() // Kurucu metot public String getItem(int indeks)

Indeksteki seçimi verir

### public synchronised void add(String s) verilen Stringi Choice listesine ekler.

**public synchronised String getSelectedItem()**

Fareyle seçilmis olan liste elemanini verir

### public synchronised String insert(String s, int indeks)

verilen stringi listeye indeks sirasindaki madde olarak ekle

### public synchronised void remove(String s)

verilen stringi listeden siler

Choice (Seçim) sinifini bir programda kullanalim :

### Problem 7.17 Choice sinifinin kullanimi, ChoiceTesti.java

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class ChoiceTesti extends Applet

{

private Choice F; private TextField T; public void init()

{

F=new Choice(); F.add("MonoSpaced"); // Courier F.add("SansSerif"); // Helvetica F.add("Serif"); // Times

T=new TextField(F.getItem(0),30); T.setEditable(false);

T.setFont(new Font(F.getItem(0),Font.PLAIN,12)); F.addItemListener(new FontIsmiKontrolu(T)); F.addItemListener(new TextFieldKontrolu(T)); add(F);

add(T);

}

}

class FontIsmiKontrolu implements ItemListener

{

private Component C;

public FontIsmiKontrolu(Component C)

{

this.C=C;

}

public void itemStateChanged(ItemEvent e)

{

C.setFont(new Font(e.getItem().toString(), C.getFont().getStyle(),

C.getFont().getSize()));

}

}

class TextFieldKontrolu implements ItemListener

{

private TextField T;

public TextFieldKontrolu(TextField T)

{

this.T=T;

}

public void itemStateChanged(ItemEvent e)

{

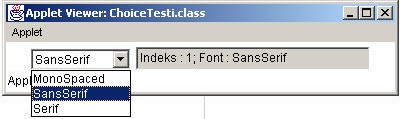
Choice C=(Choice)e.getItemSelectable();

T.setText("Indeks : "+ C.getSelectedIndex()+"; Font : "+e.getItem());

} }

Choice seçimlerini yukaridaki ChoiceTesti.html de de görülecegi gibi aktarilirken ItemListener ve ItemEvent kullanilmistir. Bu metotla kullanilmasi gereken method **ItemStateChanged metotudur.**

### 07015.JPG



**Sekil 7.15 ChoiceTesti.java programinin sonuçlarinin appletde görülmesi**

Java swing sinifinda Choice sinifinin benzeri Jchoice sinifi mevcut degildir. Onun yerini JComboBox sinifi almistir

JComboBox sinifinin Kurucu metodlari : public JComboBox();

public JComboBox(ComboBoxModel m); public JComboBox(Object obj[]);

public JComboBox(Vector v);

seklindedir. Vector gelismis bir boyutlu degisken türüdür detaylari 11 inci bölüm, gelismis java bilgi isleme yapilari kisminda verilmistir.

ComboBox Jcomponent sinifindan türetilmistir. ActionListener, ItemSelectable, ListDataListener ve Accessible siniflarini implement eder. JComboBox sinifi, seçilen sinifi size veren veya seçilme opsiyonunu tanimlayan su siniflari barindirir :

getSelectedItem() : bu noktadaki Object degerini çagirir. getSelectedIndex() : bu noktanin indeks degerini çagirir

getSelectedObjects() : birden fazla seçim yapildiginda tüm seçilen object listesini iletir. setSelectedIndex(int pozisyon) : pozisyon indeksini seçer

setSelectedItem(Object nesne) nesne ‘yi seçer

### Problem 7.18 JComboBox sinifinin kullanimi, JComboBoxTestiSWF.java

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

public class JComboBoxTestiSWF extends JFrame

{

private JComboBox F; private JTextField T;

public JComboBoxTestiSWF()

{

super("JComboBox örnegi"); Container co=getContentPane();

String Liste[]={"MonoSpaced","SansSerif","Serif"}; F=new JComboBox(Liste);

T=new JTextField((String)F.getItemAt(0)); T.setEditable(false);

T.setFont(new Font((String)F.getItemAt(0),Font.BOLD,12)); F.addItemListener(new FontIsmiKontrolu(T)); F.addItemListener(new TextFieldKontrolu(T)); co.add(F,BorderLayout.NORTH); co.add(T,BorderLayout.CENTER);

}

public static void main(String[] args)

{

JComboBoxTestiSWF pencere= new JComboBoxTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(300,150);

//pencere.pack(); pencere.setVisible(true);

}

}

class FontIsmiKontrolu implements ItemListener

{

private Component C;

public FontIsmiKontrolu(Component C)

{

this.C=C;

}

public void itemStateChanged(ItemEvent e)

{

C.setFont(new Font(e.getItem().toString(), C.getFont().getStyle(),

C.getFont().getSize()));

}

}

class TextFieldKontrolu implements ItemListener

{

private JTextField T;

public TextFieldKontrolu(JTextField T)

{

this.T=T;

}

public void itemStateChanged(ItemEvent e)

{

JComboBox C=(JComboBox)e.getItemSelectable(); T.setText("Indeks : "+ C.getSelectedIndex()+"; Font : "+e.getItem());

} }

### 07016.JPG



**Sekil 7.16 JComboBoxTestiSWF.java programinin sonuçlarinin appletde görülmesi**

### AWT, CHECKBOX VE CHECKBOXGROUP VE SWING JCHECKBOX VE JRADIOBUTTON SINIFLARI

awt sinifinda yer alan Check box ve CheckBoxGroup siniflari çesitli opsiyonlar arasindan seçim yapabilme olasiligini saglayan siniflardir.

### Kurucu metotlari :

**Public CheckBox(String s)**

Tek bir kare kontrol kutusu olusturur. Kontrol kutusu basta bostur

### Public CheckBox(String s, CheckBoxGroup c,boolean durum)

Tekbir daire seklinde kontrol kutusu olusturur ve bu kontrol kutusunu CheckBoxGroup nesnesine ilave eder.

### Public CheckboxGroup()

CheckboxGroup nesnesini olusturur bu gruba checkboxlar ilave edilebilir

CheckBox kullanimini anlamak amaciyla CheckBoxTesti.java programina göz atalim :

### Problem 7.19 CheckBoxTest.java, CheckBox kullanim test programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class CheckBoxTesti extends Applet implements ItemListener

{

private TextField T;

private Checkbox Kalin,Italic; public void init()

{

setBackground(Color.lightGray);

//bos textField

T=new TextField("Buradaki yazinin degismesini izleyiniz"); add(T);

Kalin=new Checkbox("Kalin"); Kalin.addItemListener(this); add(Kalin);

Italic=new Checkbox("Italik"); Italic.addItemListener(this); add(Italic);

}

public void itemStateChanged(ItemEvent e)

{

int KalinAyari; if(Kalin.getState())

{KalinAyari=Font.BOLD;} else

{KalinAyari=Font.PLAIN;} int ItalicAyari; if(Italic.getState())

{ItalicAyari=Font.ITALIC;} else

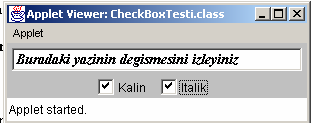
{ItalicAyari=Font.PLAIN;}

T.setFont(new Font("Serif",KalinAyari+ItalicAyari,14));

}

}

### 07017.JPG



**Sekil 7.17 CheckBoxTesti.java programinin sonuçlarinin appletde görülmesi**

Simdi de checkBoxGroup programinin çalismasini ornekleyen RadioButtonTesti programina göz atalim. Burada guruplanmis olan checkbox’lardan sadece bir tanesini seçme izni verilmistir.

### Problem 7.20 : RadioButtonTesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class RadiobuttonTesti extends Applet implements ItemListener

{

private TextField T;

private Font NormalFont,KalinFont,ItalicFont,KalinItalicFont; private Checkbox Normal,Kalin,Italic,KalinItalic;

private CheckboxGroup fontGurubu; public void init()

{

setBackground(Color.lightGray);

//bos textField

T=new TextField("Buradaki yazinin degismesini izleyiniz"); add(T);

fontGurubu=new CheckboxGroup();

Normal=new Checkbox("Normal",fontGurubu,true); Normal.addItemListener(this);

add(Normal);

Kalin=new Checkbox("Kalin",fontGurubu,true); Kalin.addItemListener(this);

add(Kalin);

Italic=new Checkbox("Italik",fontGurubu,true); Italic.addItemListener(this);

add(Italic);

KalinItalic=new Checkbox("Kalin Italik",fontGurubu,true); KalinItalic.addItemListener(this);

add(KalinItalic);

NormalFont=new Font("Serif",Font.PLAIN,14); KalinFont=new Font("Serif",Font.BOLD,14); ItalicFont=new Font("Serif",Font.ITALIC,14);

KalinItalicFont=new Font("Serif",Font.BOLD+Font.ITALIC,14); T.setFont(NormalFont);

}

public void itemStateChanged(ItemEvent e)

{

int KalinAyari; if(e.getSource()== Normal)

{T.setFont(NormalFont);} else if(e.getSource()== Kalin)

{T.setFont(KalinFont);}

else if(e.getSource()== Italic)

{T.setFont(ItalicFont);}

else if(e.getSource()== KalinItalic)

{T.setFont(KalinItalicFont);}

}

}

### 07018.JPG



**Sekil 7.18 RadioButtonTesti.java programi sonuçlarinin applette gösterilmesi**

Swing gurubundaki ayni isler için kullandigimiz siniflar JCheckBox, JRadioButton ve JButtonGroup siniflaridir. JCheckBox sinifinin kurucu metodlari :

### public JCheckBox()

**public JCheckBox(Icon resim)**

**public JCheckBox(Icon resim,boolean kutuisareti) public JCheckBox(String yazi)**

**public JCheckBox(String yazi, boolean kutuisareti) public JCheckBox(String yazi, Icon resim)**

**public JCheckBox(String yazi, , Icon resim ,boolean kutuisareti)**

Burada da daha önceki JLabel, JButton gibi siniflarda gördügümüz gibi yazinin yaninda resim yerlestirme seçimimiz de mevcuttur. Buradaki örnek problemde bir önceki problemin aynisi JcheckBox ile çözülmüstür.

### Problem 7.21 CheckBoxTestiSWF.java, JCheckBox kullanim test programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class CheckBoxTestiSWF extends JFrame implements ItemListener

{

private JTextField T;

private JCheckBox Kalin,Italik; int KalinAyari=Font.PLAIN;; int ItalikAyari=Font.PLAIN;

public CheckBoxTestiSWF()

{

super("check box örnegi"); Container c=getContentPane(); c.setBackground(Color.lightGray);

//c.setLayout(new FlowLayout()); JPanel Check=new JPanel(); Check.setLayout(new GridLayout(0,1));

//bos textField

Kalin=new JCheckBox("Kalin"); Kalin.setMnemonic('K'); Kalin.setSelected(false); Kalin.addItemListener(this); Check.add(Kalin);

Italik=new JCheckBox("Italik"); Italik.setMnemonic('I'); Italik.setSelected(false); Italik.addItemListener(this); Check.add(Italik); c.add(Check,BorderLayout.WEST);

T=new JTextField("Buradaki yazinin degismesini izleyiniz"); T.setFont(new Font("Serif",KalinAyari+ItalikAyari,20)); c.add(T,BorderLayout.CENTER);

}

public void itemStateChanged(ItemEvent e)

{

Object kutu=e.getItemSelectable();

if(Kalin==kutu && e.getStateChange()==ItemEvent.SELECTED)

{KalinAyari=Font.BOLD;}

else if(Kalin==kutu && e.getStateChange()==ItemEvent.DESELECTED)

{KalinAyari=Font.PLAIN;}

if(Italik==kutu && e.getStateChange()==ItemEvent.SELECTED)

{ItalikAyari=Font.ITALIC;}

else if(Italik==kutu && e.getStateChange()==ItemEvent.DESELECTED)

{ItalikAyari=Font.PLAIN;}

T.setFont(new Font("Serif",KalinAyari+ItalikAyari,20));

}

public static void main(String[] args)

{

CheckBoxTestiSWF pencere= new CheckBoxTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,100);

pencere.setVisible(true);

}

}

### 07019.JPG



**Sekil 7.19 CheckBoxTestiSWF.java programinin sonuçlarinin JFrame’de görülmesi**

Java swing JRadioButton sinifi JcheckBox tanimina benzerdir. Kurucu metodlari :

### public JRadioButton()

**public JRadioButton(Icon resim)**

**public JRadioButton(Icon resim,boolean kutuisareti) public JRadioButton(String yazi)**

**public JRadioButton(String yazi, boolean kutuisareti) public JRadioButton(String yazi, Icon resim)**

**public JRadioButton(String yazi, , Icon resim ,boolean kutuisareti)**

Burada JRadioButton sinifina ilave olarak bir tane daha ilave olarak bir tane daha yardimci sinif kullanacagiz. Bu sinif ButtonGroup sinifidir. Bu sinif dügmeleri bir gurup altinda toplayarak sadece bir tanesinin basili durumda olmasi islevini kontrol eder. Bu sinifin tanimi :

### public class ButtonGroup extends Object implements Serializable

**{**

**protected Vector dügmeler; public ButtonGroup();**

**public void add(AbstractButton ab); public Enumeration getElements(); public ButtonModel getSelection();**

**public boolean isSelected(ButtonModel bm); public void remove(AbstractButton ab);**

**public void setSelected(ButtonModel bm,boolean b);**

**}**

seklindedir. Simdi yine bir öncekine benzer bir örnekte JradioButton ve ButtonGroup siniflarinin birarada kullanimini görelim.

### Problem 7.22 RadioButtonTestiSWF.java, JRadioButton kullanim test programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class RadioButtonTestiSWF extends JFrame implements ItemListener

{

private JTextField T;

private JRadioButton normal,kalin,italik,kalinitalik; private ButtonGroup gurup;

int KalinAyari=Font.PLAIN;; int ItalikAyari=Font.PLAIN;

public RadioButtonTestiSWF()

{

super("Radio Button örnegi"); Container c=getContentPane(); c.setBackground(Color.lightGray); JPanel Check=new JPanel();

Check.setLayout(new GridLayout(4,0));

//bos textField

gurup=new ButtonGroup(); normal=new JRadioButton("Normal"); normal.setMnemonic('N'); normal.setSelected(true); normal.addItemListener(this); gurup.add(normal); Check.add(normal);

kalin=new JRadioButton("Kalin"); kalin.setMnemonic('K'); kalin.setSelected(false); kalin.addItemListener(this); gurup.add(kalin); Check.add(kalin);

italik=new JRadioButton("Italik"); italik.setMnemonic('t'); italik.setSelected(false); italik.addItemListener(this); gurup.add(italik); Check.add(italik);

kalinitalik=new JRadioButton("Kalin-Italik"); kalinitalik.setMnemonic('a'); kalinitalik.setSelected(false); kalinitalik.addItemListener(this); gurup.add(kalinitalik); Check.add(kalinitalik);

c.add(Check,BorderLayout.WEST);

T=new JTextField("Buradaki yazinin degismesini izleyiniz"); T.setFont(new Font("Serif",KalinAyari+ItalikAyari,20)); c.add(T,BorderLayout.CENTER);

}

public void itemStateChanged(ItemEvent e)

{

Object kutu=e.getItemSelectable(); if(kutu==normal)

{

KalinAyari=Font.PLAIN;

ItalikAyari=Font.PLAIN;

}

else if(kutu==kalin)

{

KalinAyari=Font.BOLD;

ItalikAyari=Font.PLAIN;

}

else if(kutu==italik)

{

KalinAyari=Font.PLAIN;

ItalikAyari=Font.ITALIC;

}

else if(kutu==kalinitalik)

{

KalinAyari=Font.BOLD;

ItalikAyari=Font.ITALIC;

}

T.setFont(new Font("Serif",KalinAyari+ItalikAyari,20)); repaint();

}

public static void main(String[] args)

{

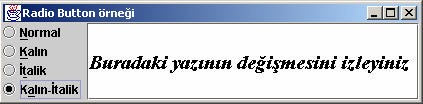
RadioButtonTestiSWF pencere= new RadioButtonTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,100);

pencere.setVisible(true);

}

}

### 07020.JPG



**Sekil 7.20 RadioButtonTestiSWF.java programinin sonuçlarinin JFrame’de görülmesi**

Son olarak bu guruptan JToggleButton sinifina deginelim. JToggleButton sinifi islevsel olarak JcheckBox sinifinin aynidir. Sadece basilim alani dügme(button) seklindedir. JcheckBox için verdigimiz programin JToggleButton’a dönüstürülmüs sekli asagida verilmistir.

### Problem 7.23 ToggleButtonTestiSWF.java, JToggleButton kullanim test programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class ToggleButtonSWF extends JFrame implements ItemListener

{

private JTextField T;

private JToggleButton Kalin,Italik; int KalinAyari=Font.PLAIN;;

int ItalikAyari=Font.PLAIN;

public ToggleButtonSWF()

{

super("ToggleButton örnegi"); Container c=getContentPane(); c.setBackground(Color.lightGray);

//c.setLayout(new FlowLayout()); JPanel Check=new JPanel(); Check.setLayout(new GridLayout(0,1));

//bos textField

Kalin=new JToggleButton("Kalin"); Kalin.setMnemonic('K'); Kalin.setSelected(false); Kalin.addItemListener(this); Check.add(Kalin);

Italik=new JToggleButton("Italik"); Italik.setMnemonic('t'); Italik.setSelected(false);

Italik.addItemListener(this); Check.add(Italik); c.add(Check,BorderLayout.WEST);

T=new JTextField("Buradaki yazinin degismesini izleyiniz"); T.setFont(new Font("Serif",KalinAyari+ItalikAyari,20)); c.add(T,BorderLayout.CENTER);

}

public void itemStateChanged(ItemEvent e)

{

Object kutu=e.getItemSelectable();

if(Kalin==kutu && e.getStateChange()==ItemEvent.SELECTED)

{KalinAyari=Font.BOLD;}

else if(Kalin==kutu && e.getStateChange()==ItemEvent.DESELECTED)

{KalinAyari=Font.PLAIN;}

if(Italik==kutu && e.getStateChange()==ItemEvent.SELECTED)

{ItalikAyari=Font.ITALIC;}

else if(Italik==kutu && e.getStateChange()==ItemEvent.DESELECTED)

{ItalikAyari=Font.PLAIN;}

T.setFont(new Font("Serif",KalinAyari+ItalikAyari,20));

}

public static void main(String[] args)

{

ToggleButtonSWF pencere= new ToggleButtonSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,100);

pencere.setVisible(true);

}

}

### 07021.JPG



**Sekil 7.21 ToggleButtonTestiSWF.java programinin sonuçlarinin JFrame’de görülmesi**

### MOUSE(FARE) KONTROLU

Modern bilgisayarlarda bilgisayar islemlerinin çok büyük bir kismi mouse (fare) ile kontrol edilir. Javada gurubunda temel fare kontrollarini saglayan siniflar MouseListener ve MouseMotionListener siniflaridir. Bu siniflarin en çok kullanilan metotlari sunlardir :

1. ***MouseListener Metotlari :***

**public void mousePressed(MouseEvent e)** bir mouse dügmesi basilinca çagirilir **public void mouseClicked(MouseEvent e)**

bir mouse dügmesi basilip birakilirsa çagirilir.

### public void mouseReleased(MouseEvent e)

Bir mouse dügmesi basili olarak mouse çekildikten sonra dügme birakilinca çagirilir.

### public void mouseEntered(MouseEvent e)

mouse pencere içine girince çagirilir.

### public void mouseExited(MouseEvent e)

mouse pencere çerçevesinin disindaysa çagirilir.

1. ***MouseMotionListener Metotlari***

**public void mouseDragged(MouseEvent e)**

Mouse dügmesi basili iken mouse hareket ettirilirse çagirilir

### public void mouseMoved(MouseEvent e)

Mouse hareket ederse çagirilir.

Bu metotlarin kullanimlarini göstermek amaciyla **FareTesti.java** programi asagida verilmistir. Bu program awt Applet programi olarak hazirlanmistir.

### Program 7.24 : FareTesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class FareTesti extends Applet

implements MouseListener, MouseMotionListener

{

private int nx,ny=-10; private String s=""; public void init()

{

addMouseListener(this); addMouseMotionListener(this);

}

public void paint(Graphics g)

{

g.drawString(s+"["+nx+","+ny+"]",nx,ny);

}

public void degerleriGir(String e,int x,int y)

{

s=e; nx=x; ny=y; repaint();

}

//fare dinleyicisinin dinledikleriyle ilgili cikti metotlari public void mouseClicked(MouseEvent e)

{ degerleriGir("Kliklendi",e.getX(),e.getY()); } public void mousePressed(MouseEvent e)

{ degerleriGir("Basildi",e.getX(),e.getY()); } public void mouseReleased(MouseEvent e)

{ degerleriGir("Birakildi",e.getX(),e.getY()); } public void mouseEntered(MouseEvent e)

{ showStatus("Fare applet alani icinde"); } public void mouseExited(MouseEvent e)

{ showStatus("Fare applet alani disinda"); }

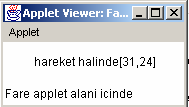
//MouseMotionListener (fare hareket dinleyicisi) metotlari public void mouseDragged(MouseEvent e)

{ degerleriGir("basilip cekiliyor",e.getX(),e.getY()); } public void mouseMoved(MouseEvent e)

{ degerleriGir("hareket halinde",e.getX(),e.getY()); }

}

### 07022.JPG



**Sekil 7.22 FareTesti.html applet çiktisi**

Ayni programin JFrame’de yazilmis bir benzeri asagida sunulmustur.

### Problem 7.25 FareTestiSWF.java, Fare(Mouse) kontrol siniflari kullanim test programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class FareTestiSWF extends JFrame implements MouseListener, MouseMotionListener

{

private int nx,ny=-10; private String s=""; private JLabel altbar;

public FareTestiSWF()

{

super("Fare Testi"); altbar=new JLabel();

getContentPane().add(altbar,BorderLayout.SOUTH); addMouseListener(this); addMouseMotionListener(this);

}

public void degerleriGir(String e,int x,int y)

{

s=e; nx=x; ny=y;

altbar.setText(e+"["+x+" , "+y+"] "); repaint();

}

//fare dinleyicisinin dinledikleriyle ilgili cikti metodlari public void mouseClicked(MouseEvent e)

{ degerleriGir("Kliklendi",e.getX(),e.getY()); } public void mousePressed(MouseEvent e)

{ degerleriGir("Basildi",e.getX(),e.getY()); } public void mouseReleased(MouseEvent e)

{ degerleriGir("Birakildi",e.getX(),e.getY()); } public void mouseEntered(MouseEvent e)

{ setTitle("Fare applet alani içinde"); } public void mouseExited(MouseEvent e)

{ setTitle("Fare applet alani disinda");

degerleriGir("Fare applet alani disinda",e.getX(),e.getY());

}

//MouseMotionListener (fare hareket dinleyicisi) metotlari public void mouseDragged(MouseEvent e)

{ degerleriGir("basilip cekiliyor",e.getX(),e.getY()); } public void mouseMoved(MouseEvent e)

{ degerleriGir("hareket halinde",e.getX(),e.getY()); } public static void main(String[] args)

{

FareTestiSWF pencere= new FareTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,150);

pencere.setVisible(true);

}

}

### 07023.JPG



**Sekil 7.23 FareTestiSWF.java JFrame penceresi çiktisi**

* 1. **SWING JTABLE SINIFI**

JTable su ana kadar gördügümüz siniflara göre daha kompleks bir yapi arzeder, fakat oldukça kullanisli bir yapidir ve sadece swing kütüphanesinde tanimlanmistir.

### JTable sinifinin kurucu siniflari :

public JTable();

public JTable(TableModel tm);

public JTable(TableModel tm, TableColumnModel tcm);

public JTable(TableModel tm, TableColumnModel tcm, ListSelectionModel lm); public JTable(int satirsayisi,int sütunsayisi);

public JTable(Object[][] tablogirdisi ,Object[] tablobasligigirdisi); public JTable(Vector tablogirdisi ,Vector tablobasligigirdisi);

seklindedir. Kurucu metodlarda girdi olarak görünen TableModel sinifi public interface TableModel{

public abstract void addTableModelListener(TableModelListener l); public abstract Class getColumnClass(int sütun);

public abstract int getColumnCount();

public abstract String getColumnName(int Column); public abstract int getRowCount();

public abstract Object getValueAt(int satir, int sütun); public abstract boolean isCellEditable(int satir, int sütun);

public abstract void removeTableModelListener(TableModelListener l); public abstract void setValueAt(Object o.int satir,int sütun);

}

görüldügü gibi bu sinif interface oldugundan bu siniftan türetilen siniflarin bu siniftaki tüm metodlari tanimlamalari gerekir. Bu yüzden AbstractTableModel gibi Table Model’den türetilen siniflar üzerinden yeni sinifin türetilmesine gidilebilir.

public abstract class AbstractTableModel implements TableModel

{

public AbstractTableModel();

public abstract void addTableModelListener(TableModelListener l); public int findColumn(String ColumnName);

public void fireTableCellUpdated(int satir,int sütun); public void fireTableChanged(TableModelEvent e); public void fireTableDataChanged();

public void fireTableRowsDeleted(int satir,int sütun); public void fireTableRowsInserted(int satir,int sütun);

public void fireTableRowsUpdated(int satir,int sütun); public void fireTableStructureChanged();

public abstract Class getColumnClass(int sütun); public abstract String getColumnName(int Column);

public abstract boolean isCellEditable(int satir, int sütun);

public abstract void removeTableModelListener(TableModelListener l); public abstract void setValueAt(Object o.int satir,int sütun);

}

Örnek problemlerde önce AbstractTableModel’den türetilen ve Tablonun giris degerlerini veren TableModel sinifi yaratilmis, sonra bu sinif tabloda girdi olarak kullanilmistir.

### Problem 7.26 tableSWF\_2000.java, TableModel, JTable ve TableModelListener siniflari kullanim test programi

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.table.\*; import javax.swing.event.\*;

class TableModel extends AbstractTableModel implements TableModelListener

{

Object[][] veri={{"satir bir - sütun bir","satir bir - sütun iki"},

{"satir iki - sütun bir","satir iki - sütun iki"}}; String[] baslik={"sütun bir","sütun iki"};

public TableModel()

{

addTableModelListener(this);

}

public int getRowCount() {return veri.length;} public int getColumnCount() {return baslik.length;}

public Object getValueAt(int satir,int sutun) {return veri[satir][sutun];} public String getColumnName(int c) {return baslik[c];}

public void setValueAt(Object val, int row, int col)

{

veri[row][col] = val;

// Degisimin oldugunu göster: fireTableDataChanged();

}

public void tableChanged(TableModelEvent e)

{

String s="Tablodaki deger degistirildi "; JOptionPane.showMessageDialog(null,s,"Tablodaki deger degistirildi", JOptionPane.WARNING\_MESSAGE);

}

public boolean isCellEditable(int row, int col) {return true;}

}

public class tableSWF\_2000 extends JFrame

{

private Container c; public tableSWF\_2000()

{

super("Tablo örnegi"); c=getContentPane(); c.setLayout(new BorderLayout()); setSize(300,300);

addWindowListener(new BasicWindowMonitor()); TableModel tm=new TableModel();

JTable jt=new JTable(tm); JScrollPane jsp=new JScrollPane(jt); c.add(jsp,BorderLayout.NORTH); f=new JTextField(); c.add(f,BorderLayout.SOUTH);

}

//=====================================================

public static void main(String[] args)

{

tableSWF\_2000 pencere= new tableSWF\_2000(); pencere.setVisible(true);

}

}

### 07024.JPG



**Sekil 7.24 26 tableSWF\_2000.java JFrame ve JoptionPane penceresi çiktisi**

**Problem 7.27 table1SWF\_2000.java, TableModel, JTable ve TableModelListener siniflari kullanim test programi**

// JTable örnegi import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*; import javax.swing.table.\*; import javax.swing.event.\*;

// The TableModel controls all the data:

class DataModel extends AbstractTableModel { Object[][] data = {

{"bir", "iki", "üç", "dört"},

{"bes", "alti", "yedi", "sekiz"},

{"dokuz", "on", "onbir", "oniki"},

};

String[] baslik={"sütun 1","sütun 2","sütun 3","sütun 4"};

// Tablo dinleme metodu ve sinifi : Tabloda bir degisiklik yapildiginda

// JOption pane ile yeni degisikligi göster class TML implements TableModelListener {

public void tableChanged(TableModelEvent e) {

String s="";

for(int i = 0; i < data.length; i++)

{

for(int j = 0; j < data[0].length; j++) s+=data[i][j] + " ";

s+="\n";

}

JOptionPane.showMessageDialog(null,s,"Tablodaki data degistirildi", JOptionPane.PLAIN\_MESSAGE);

}

}

//kurucu metod DataModel()

{

addTableModelListener(new TML());

}

public int getColumnCount() {return data[0].length;} public int getRowCount() { return data.length;}

public Object getValueAt(int row, int col) {return data[row][col];} public String getColumnName(int c) {return baslik[c];}

public void setValueAt(Object val, int row, int col)

{

data[row][col] = val;

// Degisimin oldugunu göster: fireTableDataChanged();

}

public boolean isCellEditable(int row, int col) { return true;

}

};

public class table1SWF\_2000 extends JFrame { public table1SWF\_2000()

{ super("Tablo örnegi"); Container c=getContentPane();

c.setLayout(new BorderLayout());

JTable table = new JTable(new DataModel()); JScrollPane jsp=new JScrollPane(table); c.add(jsp, BorderLayout.CENTER);

}

public static void main(String args[]) { table1SWF\_2000 pencere= new table1SWF\_2000(); pencere.setSize(300,300);

pencere.addWindowListener(new BasicWindowMonitor()); pencere.setVisible(true);

}

}

### 07025.JPG



**Sekil 7.25 table1SWF\_2000.java JFrame ve JOPtionPane çiktisin görünümü**

### AWT, LIST SINIFI VE JAVA SWING JLIST SINIFI , LISTEDEN SEÇIM

Bu gurup aslinda bölüm 7.6 da tanimlanan awt Choice , swing JcomboBox seçim siniflarinin bir devami niteligindedir.

List ve Jlist siniflari bir listeden seçim yapmaya ve bu seçim sonucu gerekli islemleri olusturmaya yarar. List sinifi ItemListener ve ActionListener sinif implementlerini kullanir. Örnek programimizda Listeden seçilen renge göre arkaplan rengi degismektedir.

### Program 7.28 ListTesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class ListTesti extends Applet implements ActionListener,ItemListener

{

private List renkListesi;

private String renkIsmi[]={"Siyah","Mavi","Cam Gobegi","Koyu Gri","Gri", "Yesil","Acik gri","Mor","Portakal","Pembe","Kirmizi","Beyaz","Sari"}; private Color C[]={Color.black,Color.blue,Color.cyan,Color.darkGray, Color.gray,Color.green,Color.lightGray,Color.magenta,Color.orange, Color.pink,Color.red,Color.white,Color.yellow};

public void init()

{

//Ekranda 5 isim g”r• lebilecel bir liste ac

//ayn• ismi birden fazla tekrarlama renkListesi=new List(5,false); renkListesi.addActionListener(this); renkListesi.addItemListener(this);

//listeye ekle

for(int i=0;i<renkIsmi.length;i++)

{

renkListesi.add(renkIsmi[i]);

}

add(renkListesi);

}

public void actionPerformed(ActionEvent e)

{

setBackground(C[renkListesi.getSelectedIndex()]); repaint();

}

public void itemStateChanged(ItemEvent e)

{

showStatus(renkListesi.getSelectedItem()+"; indeks : "+ renkListesi.getSelectedIndex());

}

}

### 07026.JPG



**Sekil 7.26 ListTesti.java programi sonuçlarin appletviewer da görünümü**

List sinifinin swing grubundaki paraleli daha gelismis bir liste kullanma metodu **JList** metodudur. Jlist metodu pencereye ilave edilirken direk olarak ilave edilmez, JscrollPane alt pencere programi üzerinden ilave edilir.

Ayni zamanda dinleme metodu olarak ListSelectionListener metodu kullanilir. Bu metod java swing kütüphanesi javax.swing.event de tanimlanmistir.

JList metoduyla bir seçim yapildigi gibi birden fazla seçim yapmakta mümkündür. JList metodunun kurucu metodlari :

### public JList();

**public JList(ListModel lm); public JList(Object o[]); public JList(Vector v);**

dir.

Seçimde kullanilan bazi önemli metodlari :

**public int GetSelectedIndex() :** seçilen (veya en son seçilen) elemanin indisini verir

**public int[] getSelectedIndices()** : seçilen elemanlarin (birden fazla olabilir) indislerini boyut üzerinden aktarir

**public Object getSelectedValue()** : son seçilen elemandaki degeri Object olarak aktarir.

**public object[] getSelectedValues()** : seçilen elemanlardaki degerleri boyutlu object olarak aktarir

**void setSelectionMode(int mode)** : tek eleman mi seçilecegini yoksa çok sayida eleman mi seçilecegini belirler. Girdi int degiskeni

**ListSelectionModel.SINGLE\_SELECTION** (tek eleman seçimi)

**ListSelectionModel .INTERVAL\_SELECTION** (bir gurup seçimi)

**ListSelectionModel .MULTIPLE\_INTERVAL\_SELECTION** (birden fazla gurup seçimi) Degerlerinden birini alabilir.

**Public void setVisibleRowCount(int sayi) :** listenin kaç elemaninin görüntülenecegini verir.

Listenin kenar kayma çubuklari, listenin içinde yer alacagi JScrollPane penceresi tarafindan olusturulur.

JScrollPane metodunun kurucu metodlari

### public JScrollPane();

**public JScrollPane(int düseyçubukçesidi,int yatayçubukçesidi); public JScrollPane(Component pencere);**

**public JScrollPane(Component pencere ,int düseyçubukçesidi,int yatayçubukçesidi);**

Buradaki pencere JScrollPane’in içinde yeralacagi container’i gösterir. Düseyçubukçesidi **JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS :** her zaman düsey kaydirma çubugu tanimla **JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED :** sadece gerektigi zaman zaman düsey kaydirma çubugu tanimla

**JScrollPane.VERTICAL\_SCROLLBAR\_NEVER :** hiçbir zaman düsey kaydirma çubugu tanimlama Degerlerini alir. Yatayçubukçesidi ise

**JScrollPane. HORIZONTAL \_SCROLLBAR\_ALWAYS :** her zaman yatay kaydirma çubugu tanimla **JScrollPane. HORIZONTAL \_SCROLLBAR\_AS\_NEEDED :** sadece gerektigi zaman yatay kaydirma çubugu tanimla

**JScrollPane. HORIZONTAL \_SCROLLBAR\_NEVER :** hiçbir zaman yatay kaydirma çubugu tanimlama Degerlerini alir.

Birkaç örnek problemle Jlist sinifini vermetye çalisalim. Ilk örnek yukaridaki JcomboBox problemlerin ayni islemini yapmaktadir. Listeden yapilan seçime göre JtextField alanindaki fontu degistirecektir. Listenin iki elemaninin görülmesi müsadesi verilmistir. Liste tek seçimli listedir.

### Problem 7.29 JListTestiSWF.java, JList sinifi kullanim test programi

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

import javax.swing.event.\*;

public class JListTestiSWF extends JFrame implements ListSelectionListener

{

private JList F; private JTextField T;

public JListTestiSWF()

{

super("JList ve JScrollPane örnegi"); Container co=getContentPane();

String Liste[]={"MonoSpaced","SansSerif","Serif"};

F=new JList(Liste);

F.setVisibleRowCount(2);

F.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION); F.setSelectedIndex(0);

T=new JTextField((String)F.getSelectedValue()); T.setEditable(false);

T.setFont(new Font((String)F.getSelectedValue(),Font.BOLD,12)); F.addListSelectionListener(this);

JScrollPane sp=new JScrollPane(F); co.add(sp,BorderLayout.NORTH); co.add(T,BorderLayout.CENTER);

}

public void valueChanged(ListSelectionEvent e)

{

T.setFont(new Font((String)F.getSelectedValue(), T.getFont().getStyle(),

T.getFont().getSize()));

T.setText("Indeks : "+ F.getSelectedIndex()+" Font : "+F.getSelectedValue());

}

public static void main(String[] args)

{

JListTestiSWF pencere= new JListTestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(300,200);

pencere.setVisible(true);

}

}

### 07027.JPG



**Sekil 7.27 JListTestiSWF.java programi sonuçlarin JFrame penceresinde görünümü**

Bu örnekte ise listeden yapacagimiz seçime göre arka plan rengini degistiriyoruz. Bu program yukarida verilen List programinin oldukça benzeridir. Listede dört elemanin görülmesi izni verilmistir. Liste tek seçimli bir listedir.

### Problem 7.30 JlistTesti1SWF.java, JList sinifi kullanim test programi

import javax.swing.\*; // java swing sinifini cagir import javax.swing.event.\*;

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import BasicWindowMonitor;

import renk;

public class JListTesti1SWF extends JFrame implements ListSelectionListener

{

// Renk secme ornegi

private String renkler[]={"siyah","Mavi","camgöbegi","Koyu Gri","Gri", "Yesil","Açik gri","mor","Portakal","Pembe","Kirmizi","beyaz","sari"}; private renk re[]={renk.siyah,renk.mavi,renk.camgobegi,renk.koyugri, renk.gri,renk.yesil,renk.acikgri,renk.mor,renk.portakal,renk.pembe, renk.kirmizi,renk.beyaz,renk.sari};

JList renklistesi;

Color r=Color.lightGray;

Container c;

// pencereyi baslatma metodu public JListTesti1SWF()

{

super("JListTesti1 renk seçici"); c=getContentPane(); c.setLayout(new FlowLayout()) ; renklistesi=new JList(renkler); renklistesi.setVisibleRowCount(4);

renklistesi.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION); c.add(new JScrollPane(renklistesi)); renklistesi.addListSelectionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void valueChanged(ListSelectionEvent e)

{

r=re[renklistesi.getSelectedIndex()]; if(r==null)

r=renk.acikgri; c.setBackground(r); repaint();

}

//=====================================================

public static void main(String[] args)

{

JListTesti1SWF pencere= new JListTesti1SWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

}

### 07028.JPG



**Sekil 7.28 JlistTesti1SWF.java programi sonuçlarin JFrame penceresinde görünümü**

Diger bir Liste kullanma örnegi : Bu örnekte Listeye JtextField üzerinden ilave yapabiliyoruz ve çikarabiliyoruz. Örnek Java döküman kütüphanesinden alinip adapte edilmistir.

### Problem 7.31 ListDemoSWF.java, JList sinifi kullanim test programi

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.event.\*;

public class ListDemoSWF extends JFrame implements ListSelectionListener { private JList list;

private DefaultListModel listModel;

private static final String hireString = "Ise al"; private static final String fireString = "Isten at"; private JButton fireButton;

private JTextField employeeName;

public ListDemoSWF() { super("ListDemo");

listModel = new DefaultListModel(); listModel.addElement("Turhan Çoban"); listModel.addElement("Irfan Çoban"); listModel.addElement("Birsen Çeliker"); listModel.addElement("Nurhan Erel");

//Listeyi olustur ve scrollPane'e yerlestir list = new JList(listModel);

list.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION); list.setSelectedIndex(0);

list.addListSelectionListener(this);

JScrollPane listScrollPane = new JScrollPane(list);

JButton hireButton = new JButton(hireString); hireButton.setActionCommand(hireString); hireButton.addActionListener(new HireListener());

fireButton = new JButton(fireString); fireButton.setActionCommand(fireString); fireButton.addActionListener(new FireListener());

employeeName = new JTextField(10); employeeName.addActionListener(new HireListener());

String name = listModel.getElementAt(list.getSelectedIndex()).toString(); employeeName.setText(name);

//Create a panel that uses FlowLayout (the default). JPanel buttonPane = new JPanel(); buttonPane.add(employeeName); buttonPane.add(hireButton); buttonPane.add(fireButton);

Container contentPane = getContentPane(); contentPane.setLayout(new BorderLayout()); contentPane.add(listScrollPane, BorderLayout.CENTER); contentPane.add(buttonPane, BorderLayout.SOUTH);

}

class FireListener implements ActionListener { public void actionPerformed(ActionEvent e) {

//Bu metod hala listede seçilebilecek eleman kalmissa çagirilir. int index = list.getSelectedIndex();

listModel.remove(index);

int size = listModel.getSize();

//Listede eleman kalmadi isten atmayi durdur if (size == 0) {

fireButton.setEnabled(false);

//seçimi düzenle

} else {

if (index == listModel.getSize()) index--;

list.setSelectedIndex(index);

}

}

}

class HireListener implements ActionListener { public void actionPerformed(ActionEvent e) {

//Eger yeni isim girilmemisse

if (employeeName.getText().equals("")) { Toolkit.getDefaultToolkit().beep(); return;

}

int index = list.getSelectedIndex(); int size = listModel.getSize();

if (index == -1 || (index+1 == size)) { listModel.addElement(employeeName.getText()); list.setSelectedIndex(size);

} else {

listModel.insertElementAt(employeeName.getText(), index+1); list.setSelectedIndex(index+1);

}

}

}

public void valueChanged(ListSelectionEvent e) { if (e.getValueIsAdjusting() == false) {

if (list.getSelectedIndex() == -1) { fireButton.setEnabled(false); employeeName.setText("");

} else {

fireButton.setEnabled(true);

String name = list.getSelectedValue().toString(); employeeName.setText(name);

}

}

}

public static void main(String s[])

{

JFrame frame = new ListDemoSWF(); frame.addWindowListener(new BasicWindowMonitor()); frame.pack();

frame.setVisible(true);

}

}

### 07029.JPG



**Sekil 7.29 ListDemoSWF.java programi sonuçlarin JFrame penceresinde görünümü**

Listelerin sadece tek seçim degil ayni zamanda birden fazla seçim yapmak içinde kullanildigindan bahsetmistik. Asagidaki örnekte listeden Liste türü degiskeninde yaptigimiz degisiklige göre bir veya birden fazla seçim yapabiliyoruz . Bu program hem JFrame hem de Japplet olarak çalisabilmektedir.

### Problem 7.32 ListSelectionDemo.java, JList sinifi kullanim test programi

import javax.swing.\*; import javax.swing.event.\*; import javax.swing.table.\*;

import java.util.\*; import java.awt.\*; import java.awt.event.\*;

public class ListSelectionDemo extends JApplet { JTextArea output;

JList list;

JTable table;

String newline = "\n"; ListSelectionModel listSelectionModel;

private boolean inAnApplet = true;

//bazi sistemlerde ana kurucu metod olmayinca hata verebilecegi

// için tanimlandi.

public ListSelectionDemo()

{

this(true);

}

public ListSelectionDemo(boolean inAnApplet) { this.inAnApplet = inAnApplet;

if (inAnApplet) { getRootPane().putClientProperty("defeatSystemEventQueueCheck",

Boolean.TRUE);

}

}

public void init() {

String[] listData = { "bir", "iki", "üç", "dört",

"bes", "alti", "yedi" };

String[] columnNames = { "Fransizca", "Ingilizce", "Italyanca" }; String[][] tableData = {{"un", "obe", "uno" },

{"deux", "two", "due" },

{"trois", "three", "tre" },

{ "quatre", "four", "quattro"},

{ "cinq", "five", "cinque" },

{ "six", "six", "sei" },

{ "sept", "seven", "sette" } }; list = new JList(listData);

listSelectionModel = list.getSelectionModel(); listSelectionModel.addListSelectionListener(

new SharedListSelectionHandler()); JScrollPane listPane = new JScrollPane(list);

table = new JTable(tableData, columnNames); table.setSelectionModel(listSelectionModel); JScrollPane tablePane = new JScrollPane(table);

//Kontrol alanini JPAnel olarak olustur (FlowLayout kullan) JPanel controlPane = new JPanel();

String[] modes = { "SINGLE\_SELECTION", "SINGLE\_INTERVAL\_SELECTION", "MULTIPLE\_INTERVAL\_SELECTION" };

final JComboBox comboBox = new JComboBox(modes); comboBox.setSelectedIndex(2); comboBox.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String newMode = (String)comboBox.getSelectedItem(); if (newMode.equals("SINGLE\_SELECTION")) {

listSelectionModel.setSelectionMode( ListSelectionModel.SINGLE\_SELECTION);

} else if (newMode.equals("SINGLE\_INTERVAL\_SELECTION")) { listSelectionModel.setSelectionMode(

ListSelectionModel.SINGLE\_INTERVAL\_SELECTION);

} else {

listSelectionModel.setSelectionMode( ListSelectionModel.MULTIPLE\_INTERVAL\_SELECTION);

}

});

}

output.append("----------"

+ "Liste seçim türü: " + newMode

+ "----------" + newline);

controlPane.add(new JLabel("Liste Türü :")); controlPane.add(comboBox);

//Çikti alani olustur

output = new JTextArea(10, 40); output.setEditable(false);

JScrollPane outputPane = new JScrollPane(output, ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS, ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_ALWAYS);

//JFrame'i Split pane kullanarak ikiye böl

JSplitPane splitPane = new JSplitPane(JSplitPane.VERTICAL\_SPLIT); getContentPane().add(splitPane, BorderLayout.CENTER);

//

JPanel topHalf = new JPanel();

topHalf.setLayout(new BoxLayout(topHalf, BoxLayout.X\_AXIS));

JPanel listContainer = new JPanel(new GridLayout(1,1)); listContainer.setBorder(BorderFactory.createTitledBorder(

"Türkçe Liste")); listContainer.add(listPane);

JPanel tableContainer = new JPanel(new GridLayout(1,1)); tableContainer.setBorder(BorderFactory.createTitledBorder( "Tablo"));

tableContainer.add(tablePane); tablePane.setPreferredSize(new Dimension(300, 100));

topHalf.setBorder(BorderFactory.createEmptyBorder(5,5,0,5)); topHalf.add(listContainer);

topHalf.add(tableContainer);

topHalf.setMinimumSize(new Dimension(400, 50));

topHalf.setPreferredSize(new Dimension(400, 110)); splitPane.add(topHalf);

JPanel bottomHalf = new JPanel(new BorderLayout()); bottomHalf.add(controlPane, BorderLayout.NORTH); bottomHalf.add(outputPane, BorderLayout.CENTER);

//XXX: the next line is necessary if bottomHalf is a scroll pane:

//bottomHalf.setMinimumSize(new Dimension(400, 50));

bottomHalf.setPreferredSize(new Dimension(450, 135)); splitPane.add(bottomHalf);

}

public static void main(String[] args) {

JFrame frame = new JFrame("Listeden seçme örnegi"); frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) { System.exit(0);

}

});

ListSelectionDemo listDemo = new ListSelectionDemo(false); listDemo.init();

frame.setContentPane(listDemo); frame.pack(); frame.setVisible(true);

}

class SharedListSelectionHandler implements ListSelectionListener { public void valueChanged(ListSelectionEvent e) {

ListSelectionModel lsm = (ListSelectionModel)e.getSource();

int firstIndex = e.getFirstIndex(); int lastIndex = e.getLastIndex();

boolean isAdjusting = e.getValueIsAdjusting(); output.append("Seçilen indeks bölgesi: "

+ firstIndex + " - " + lastIndex

+ "; isAdjusting " + isAdjusting

+ "; Listeden seçilen indeks sayisi:");

if (lsm.isSelectionEmpty()) { output.append(" <none>");

} else {

// Find out which indexes are selected.

int minIndex = lsm.getMinSelectionIndex(); int maxIndex = lsm.getMaxSelectionIndex(); for (int i = minIndex; i <= maxIndex; i++) {

if (lsm.isSelectedIndex(i)) { output.append(" " + i);

}

}

}

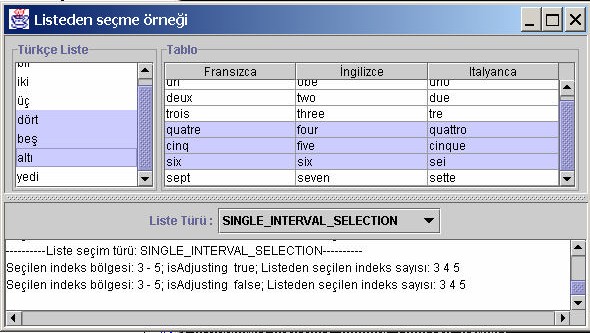
output.append(newline);

}

}

}

### 07030.JPG



**Sekil 7.30 ListSelectionDemo.java programi sonuçlarin JFrame penceresinde görünümü** (not bu program hem applet hem frame olark görüntülenebilir)

### SWING JMENU

Window türü programlamaya alisik olan herkes menu kullanmaya da alisiktir. Menuler pencerenin tepesinden açilan ve çesitli görevler yüklenen komponentlerdir. Java swing menu’leri temel olarak JMenuBar, JMenu ve JMenuItem siniflarindan olusur. Ayrica menulerde JcheckBoxMenuItem, JradioButtonMenuItem gibi alt elemanlar da bulunabilir.

**JmenuBar** kurucu metodu : Public JmenuBar(); Seklindedir.

Bazi önemli metodlari :

public JMenu add(JMenu menu); : menu ilave eder public Component getComponent();

public Component getComponentAtIndex(int indeks); public Insets getMargin();

public Jmenu getMenu(int) public int getMenuCount();

public SingleSelectionModel getSelectionModel(); public MenuElement[] getSubElements(); protected void paintBorder(Graphics g);

public void processMouseEvent(MouseEvent e,MouseElement path[],MouseSelectionManager mgr); public void setBorderPainted(boolean b);

public void setHelpMenu(Jmenu menu); public void setMargin(Insets in);

pubic void setSelected(Component c);

**Jmenu** kurucu metodlari : public JMenu();

public JMenu(String Label);

public JMenu(String label,boolean menudenayir);

### Bazi önemli JMenu metodlari :

public JMenuItem add(Action act); public JMenuItem add(JMenuItem item); public Component add(Component c); **public void add(String Jlabel);**

public void addMenuListener(MenuListener l); public void addSeperator();

public JMenuItem getItem(int pos); public int getItemCount();

public Component getMenuComponent(int pos); public int getMenuComponentCount();

public Component[] getMenuComponents(public MenuElement[] getSubElements(); public JMenuItem insert(JMenuItem item,int pos);

public void insert(String etiket,int pos); public void remove(int pos);

public void remove(MenuItem item); public void removeAll();

public void removeMenuListener(MenuListener l); public void setAccelerator(KeyStroke s);

public void setDealy(int delay);

public void setMenuLocation(int x,int y); public void setPopUpMenuVisible(boolean b) public void setSelected(boolean b);

JMenuItem kurucu metodlari : public JMenuItem();

public JMenuItem(Icon resim); public JMenuItem(String Label);

public JMenuItem(String Label,int mnemonic); public JMenuItem(String Label,Icon resim); bazi metodlari :

### publci Component getComponent();

public menuElement[] getSubElements();

JCheckBoxMenuItem, JMenuItem sinifinin bir alt sinifi olarak olusturulmustur. Davranisi JCheckBox sinifi gibidir. Kurucu metodlari :

public JCheckBoxMenuItem();

public JCheckBoxMenuItem(Icon resim); public JCheckBoxMenuItem(String Label);

public JCheckBoxMenuItem(String Label,Icon resim);

public JCheckBoxMenuItem(String Label,Boolean seçildimi);

public JCheckBoxMenuItem(String Label,Icon resim, Boolean seçildimi);

JRadioButtonMenuItem’ da JMenuItem’in alt sinifi olarak olusturulmustur. Kurucu metodlari : public JRadioButtonMenuItem ();

public JRadioButtonMenuItem (Icon resim); public JRadioButtonMenuItem (String Label);

public JRadioButtonMenuItem (String Label,Icon resim);

Simdi bu metodlari kullanan bir örnek menu olusturalim. Örnek menu sadece menu yapisini göstermek amaçli oldugundan fazla bir eyem yamamaktadir. Gerçek programlarinizda gerekli eylemleri yerestirebilirsiniz.

### Problem 7.33 MenuDemo.java, JMenu,JmenuBar,JMenuItem,JCheckBoxMenuItem ve JRadioButtonMenuItem siniflari kullanim test programi

import java.awt.\*; import java.awt.event.\*;

import javax.swing.JMenu; import javax.swing.JMenuItem;

import javax.swing.JCheckBoxMenuItem; import javax.swing.JRadioButtonMenuItem; import javax.swing.ButtonGroup;

import javax.swing.JMenuBar; import javax.swing.KeyStroke; import javax.swing.ImageIcon;

import javax.swing.JTextArea; import javax.swing.JScrollPane; import javax.swing.JFrame;

/\*

\* Menu Demo

\*/

public class MenuDemo extends JFrame

implements ActionListener, ItemListener { JTextArea output;

JScrollPane scrollPane;

String newline = "\n";

public MenuDemo() { JMenuBar menuBar; JMenu menu, submenu; JMenuItem menuItem;

JRadioButtonMenuItem rbMenuItem; JCheckBoxMenuItem cbMenuItem; addWindowListener(new BasicWindowMonitor());

//Add regular components to the window, using the default BorderLayout. Container contentPane = getContentPane();

output = new JTextArea(5, 30); output.setEditable(false);

scrollPane = new JScrollPane(output); contentPane.add(scrollPane, BorderLayout.CENTER);

//Create the menu bar. menuBar = new JMenuBar(); setJMenuBar(menuBar);

//Build the first menu.

menu = new JMenu("Menu Listesi"); menu.setMnemonic('M'); menu.getAccessibleContext().setAccessibleDescription(

"Programdaki sadece bu Menunun alt listeleri mevcuttur"); menuBar.add(menu);

//a group of JMenuItems

menuItem = new JMenuItem("Sadece yazi olan bir menu elemani", KeyEvent.VK\_T);

menuItem.setMnemonic('S'); //used constructor instead menuItem.setAccelerator(KeyStroke.getKeyStroke(

KeyEvent.VK\_1, ActionEvent.ALT\_MASK)); menuItem.getAccessibleContext().setAccessibleDescription(

"Bu sadece bir örnek oldugundan bu eleman bir sey yapmiyor"); menuItem.addActionListener(this);

menu.add(menuItem);

menuItem = new JMenuItem("Yazi ve resim",

new ImageIcon("images/middle.gif")); menuItem.setMnemonic(KeyEvent.VK\_B); menuItem.addActionListener(this); menu.add(menuItem);

menuItem = new JMenuItem(new ImageIcon("images/middle.gif")); menuItem.setMnemonic('d');

menuItem.addActionListener(this); menu.add(menuItem);

//a group of radio button menu items menu.addSeparator();

ButtonGroup group = new ButtonGroup();

rbMenuItem = new JRadioButtonMenuItem("radio dügmesi menu elemani"); rbMenuItem.setSelected(true);

rbMenuItem.setMnemonic('r'); group.add(rbMenuItem); rbMenuItem.addActionListener(this); menu.add(rbMenuItem);

rbMenuItem = new JRadioButtonMenuItem("ikinci bir radyo dügmesi"); rbMenuItem.setMnemonic('i');

group.add(rbMenuItem); rbMenuItem.addActionListener(this); menu.add(rbMenuItem);

//a group of check box menu items menu.addSeparator();

cbMenuItem = new JCheckBoxMenuItem("check box menu elemani"); cbMenuItem.setMnemonic('c');

cbMenuItem.addItemListener(this); menu.add(cbMenuItem);

cbMenuItem = new JCheckBoxMenuItem("ikinci bir check box menu elemani"); cbMenuItem.setMnemonic('k');

cbMenuItem.addItemListener(this); menu.add(cbMenuItem);

//a submenu menu.addSeparator();

submenu = new JMenu("Alt menu"); submenu.setMnemonic('A');

menuItem = new JMenuItem("Alt menude bir eleman"); menuItem.setAccelerator(KeyStroke.getKeyStroke(

KeyEvent.VK\_2, ActionEvent.ALT\_MASK)); menuItem.addActionListener(this); submenu.add(menuItem);

menuItem = new JMenuItem("Alt menude ikinci bir eleman"); menuItem.addActionListener(this);

submenu.add(menuItem); menu.add(submenu);

//Build second menu in the menu bar. menu = new JMenu("Iknci bir Menu"); menu.setMnemonic('n');

menu.getAccessibleContext().setAccessibleDescription( "Bu menu hiç bir sey yapmaz ve alt elemani yoktur");

menuBar.add(menu);

}

public void actionPerformed(ActionEvent e) { JMenuItem source = (JMenuItem)(e.getSource()); String s = "Action event kaydedildi"

+ newline

+ " Event kaynagi: " + source.getText()

+ " (sinifin adi " + getClassName(source) + ")"; output.append(s + newline);

}

public void itemStateChanged(ItemEvent e) { JMenuItem source = (JMenuItem)(e.getSource()); String s = "Item event kaydedildi."

+ newline

+ " Event kaynagi: " + source.getText()

+ " (sinifin adi " + getClassName(source) + ")"

+ newline

+ " New state: "

+ ((e.getStateChange() == ItemEvent.SELECTED) ? "seçildi":"seçim iptal edildi");

output.append(s + newline);

}

// Sinifin adini verir

protected String getClassName(Object o) { String classString = o.getClass().getName(); int dotIndex = classString.lastIndexOf("."); return classString.substring(dotIndex+1);

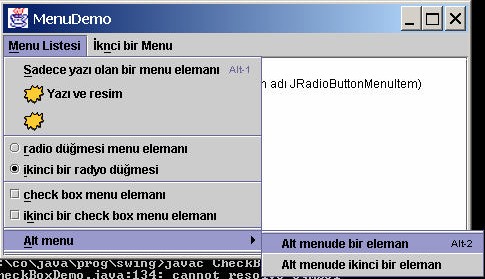
}

public static void main(String[] args) { MenuDemo window = new MenuDemo(); window.setTitle("MenuDemo"); window.setSize(450, 260); window.setVisible(true);

}

}

### 07031.JPG



**Sekil 7.31 MenuDemo.java programi sonuçlarin JFrame penceresinde görünümü**

### SWING JSLIDER SINIFI VE JPANEL TEMEL ÇIZIM ELEMANI (PANELI)

JSlider sinifi bir degiskenin degerini verilen sinirlar içerisinde degistirmeye yarayan bir grafik kullanici arayüz sinifidir. JSlider sinifinin kurucu metodlari :

### public JSlider();

**public JSlider(BoundedRAngeModel brm);**

**public JSlider(int konum);**

**public JSlider(int minimumdeger, int maksimumdeger);**

**Problem 7.34 ovalPanelSWP.java, JPanel sinifi**

import javax.swing.\*; import java.awt.\*; import yildizSW; import java.awt.geom.\*; import renk;

public class ovalPanelSWP extends JPanel

{

int xi,yi,yaricap;

public ovalPanelSWP(int xi1, int yi1,int yaricap1)

{

xi=xi1; yi=yi1;

yaricap=(yaricap1 >= 0 ? yaricap1:10);

}

public void setOvalPanelSWP(int xi1, int yi1,int yaricap1)

{

xi=xi1; yi=yi1;

yaricap=(yaricap1 >= 0 ? yaricap1:10); repaint();

}

public void paintComponent(Graphics g)

{

super.paintComponent(g); g.setColor(renk.mavi); Graphics2D g2=(Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

Ellipse2D elips1=new Ellipse2D.Double(xi,yi,yaricap,yaricap); g2.draw(elips1);

}

}

### Problem 7.35 ovalPanelTestSWP\_2000.java, JSlider sinifi test program

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import BasicWindowMonitor;

import ovalPanelSWP; import javax.swing.event.\*;

public class ovalTestSWF\_2000 extends JFrame implements ChangeListener

{

// Renk secme ornegi private ovalPanelSWP p; private JSlider cap; Color r=Color.lightGray; Container c;

// pencereyi baslatma metodu

public ovalTestSWF\_2000()

{

super("JSlider Testi"); c=getContentPane();

p=new ovalPanelSWP(30,30,30);

cap=new JSlider(SwingConstants.HORIZONTAL,0,200,10); cap.setMajorTickSpacing(20);

cap.setPaintTicks(true); cap.setPaintLabels(true); cap.addChangeListener(this); c.add(p,BorderLayout.CENTER); c.add(cap,BorderLayout.SOUTH);

}

// girdi alanindaki olan olaylari dinleme metodu public void stateChanged(ChangeEvent e)

{

p.setOvalPanelSWP(30, 30,cap.getValue()); repaint();

}

//=====================================================

public static void main(String[] args)

{

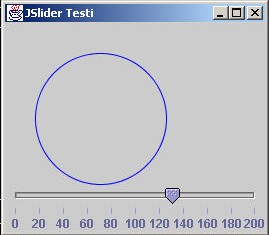
ovalTestSWF\_2000 pencere= new ovalTestSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

}

### 07032.JPG



**Sekil 7.32 ovalPanelTestSWP\_2000.java programi sonuçlarin JFrame penceresinde görünümü**

Bu programda uyguladigimiz bir çizim sekline daha detayli göz atalim. Daire JPanel sinifinda çizilmistir. Daireyi çizmek için ovalPanelSWP sinifini yarattik. JPanel sinifinda kurucu metodlarimizi herhangi bir sinifta oldugu gibi tanimladik ve **public void paintComponent(Graphics g)** metodu ve Graphics sinifi üzerinden çizimimizi tanimladik. Tekrar çizdirmek için **repaint()** metodunu kullandik. Ikinci JFrame proramimizda

private ovalPanelSWP p;

p=new ovalPanelSWP(30,30,30); Container c;

c=getContentPane(); c.add(p,BorderLayout.CENTER);

komutlarini kullanarak panelimizi ana panele monte ettik. StateChanged event metodundaki

p.setOvalPanelSWP(30, 30,cap.getValue());

komutuyla da çapini degistirip repaint();

komutuyla tekrar çizdirdik.

**Panel programinda** g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,RenderingHints.VALUE\_ANTIALIAS\_ON); komutu kullanildigindan dairenin oldukça düzgün çizilmis olduguna dikkatinizi çekeriz.

### FLOWLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI

Grafik arayüz programlarinin daha düzgün bir sekilde gösterilmesi amaciyla **Layout** siniflari kullanilir. **awt** grafik interface’inde default deger olarak kullanilan Layout sinifi FlowLayout sinifidir. Swing arayüz programlarinda ise BorderLayout sinifi default olarak kullanilar.

Flow layout sinifi grafik ortamina (Panel, Applet, Frame, Jpanel, Japplet, JFrame vs.) alt elemanlari yerlestirirken soldan saga dogru akleme (add) sirasiyla yerlestirir. Penceredeki yer bittiyse bir alt satira geçer.

Kurucu metotu su sekillerde tanimlanir :

### public Flowlayout()

**public Flowlayout(int formatbolgesi)**

format bölgesi degiskeni

### FlowLayout.RIGHT, FlowLayout.CENTER veya FlowLayout.LEFT

degerlerini alabilir. Bu degerlere göre içerdigi elemanlari saga ortaya veya sola yerlestirir.

### public Flowlayout(int formatbolgesi,int yatay\_bosluk,int dikey\_bosluk);

formatbolgesi FlowLayout.RIGHT, FlowLayout.CENTER veya FlowLayout.LEFT degerlerini alabilir.

Yatay\_bosluk pixel olarak yatay posisyonda elemanlar arasinda kalan boslugu gosterir. Dikey bosluk ise dikey dogrultudaki pixel olarak bosluktur.

Küçuk bir örnek problemle FlowLayout kullanimini daha açik hale getirebiliriz.

### Program 7.36 : FlowLayout.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class FlowLayoutTesti extends Applet implements ActionListener

{

private Button B[]; public void init()

{

B=new Button[3];

B[0]=new Button("Sol");

B[1]=new Button("Orta");

B[2]=new Button("Sag"); for(int i=0;i<B.length;i++)

{

B[i].addActionListener(this); add(B[i]);

}

}

public void actionPerformed(ActionEvent e)

{

int pozisyon=FlowLayout.LEFT;; if(e.getSource()==B[0])

{

pozisyon=FlowLayout.LEFT;

}

else if(e.getSource()==B[1])

{

pozisyon=FlowLayout.CENTER;

}

else if(e.getSource()==B[2])

{

pozisyon=FlowLayout.RIGHT;

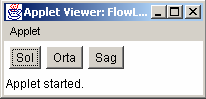
}

setLayout(new FlowLayout(pozisyon)); validate();

}

}

### 07033.JPG



**Sekil 7.33 FlowLayout sinifi ve sonuçlarinin Appletde görülmesi**

Ayni programin bir de swing JFrame versiyonunu inceleyelim.

### Program 7.37 : FlowLayoutTestiSWF\_2000.java programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class FlowLayoutTestiSWF\_2000 extends JFrame implements ActionListener

{

private JButton B[]; private Container c;

public FlowLayoutTestiSWF\_2000()

{

super("Flow Layout formatlama Grafik ara yüz (GUI) testi"); c=getContentPane();

c.setLayout(new FlowLayout()); B=new JButton[3];

B[0]=new JButton("Sol"); B[1]=new JButton("Orta"); B[2]=new JButton("Sag"); for(int i=0;i<B.length;i++)

{

B[i].addActionListener(this); c.add(B[i]);

}

}

public void actionPerformed(ActionEvent e)

{

int pozisyon=FlowLayout.LEFT;; if(e.getSource()==B[0])

{

pozisyon=FlowLayout.LEFT;

}

else if(e.getSource()==B[1])

{

pozisyon=FlowLayout.CENTER;

}

else if(e.getSource()==B[2])

{

pozisyon=FlowLayout.RIGHT;

}

c.setLayout(new FlowLayout(pozisyon)); c.validate();

}

public static void main(String[] args)

{

FlowLayoutTestiSWF\_2000 pencere= new FlowLayoutTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

}

### 07034.JPG



**Sekil 7.34 FlowLayout sinifi ve sonuçlarinin JFrame penceresinde görülmesi**

### BORDERLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI

BorderLayout sinifi GUI elemanlarini bes bolgeye ayirarak guruplar, kuzey(NORTH), güney(SOUTH), dogu(EAST),Bati(WEST),orta(CENTER), Kuzey Appletin üst ismi anlamina gelir. Bu sinif swing grafik kütüphanesinde default formatlama sinifi olarak seçilmistir. BorderLayout sinifinin kurucu metotlari sunlardir :

### public BorderLayout()

**public BorderLayout(int yatay\_bosluk,int dikey\_bosluk);**

yatay\_bosluk ve dikey\_basluk pixel olarak alinir. GUI elemanlarinin arasindaki bosluk miktarini tanimlar.

Simdi bir örnekle sinifi açiklamaya çalisalim.

### Program 7.38 : BorderLayoutTesti.java Programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class BorderLayoutTesti extends Applet implements ActionListener

{

private Button B[];

private String S[]={"Kuzeyi Gizle","Guneyi Gizle","Doguyu Gizle", "Batiyi Gizle","Ortayi Gizle"};

public void init()

{

setLayout(new BorderLayout(5,5)); B=new Button[5];

for(int i=0;i<B.length;i++)

{

B[i]=new Button(S[i]); B[i].addActionListener(this);

}

add(B[0],BorderLayout.NORTH); add(B[1],BorderLayout.SOUTH); add(B[2],BorderLayout.EAST); add(B[3],BorderLayout.WEST); add(B[4],BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<B.length;i++)

{

if(e.getSource()==B[i]) B[i].setVisible(false); else B[i].setVisible(true);

}

validate();

}

}

### Sekil 7.11.1 BorderLayoutTesti.java programi 07035.JPG

**Sekil 7.35 BorderLayoutTesti.java programinin sonuçlarinin applette görulmesi Program 7.39 : BorderLayoutTestiSWF\_2000.java Programi**

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class BorderLayoutTestiSWF\_2000 extends JFrame implements ActionListener

{

private JButton B[];

private String S[]={"Kuzeyi Gizle","Güneyi Gizle","Doguyu Gizle", "Batiyi Gizle","Ortayi Gizle"};

private Container c;

public BorderLayoutTestiSWF\_2000()

{

super("Border Layout formatlama Grafik ara yüz (GUI) testi"); c=getContentPane();

c.setLayout(new BorderLayout(5,5)); B=new JButton[5];

for(int i=0;i<B.length;i++)

{

B[i]=new JButton(S[i]); B[i].addActionListener(this);

}

c.add(B[0],BorderLayout.NORTH); c.add(B[1],BorderLayout.SOUTH); c.add(B[2],BorderLayout.EAST); c.add(B[3],BorderLayout.WEST); c.add(B[4],BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<B.length;i++)

{

if(e.getSource()==B[i]) B[i].setVisible(false); else B[i].setVisible(true);

}

c.validate();

}

public static void main(String[] args)

{

BorderLayoutTestiSWF\_2000 pencere= new BorderLayoutTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 07036.JPG



**Sekil 7.36 BorderLayoutTestiSWF\_2000.java programinin sonuçlarinin JFrame penceresinde görulmesi**

### GRIDLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI

GridLayout sinifi pencereyi (Applet,Japplet,Panel,JPAnel,Frame,JFrame vs.) esit boyutta gridlere böler ve her grid elemaninin içine bir GUI yerlestirilebilmesine olanak saglar. GridLayout sinifinin kurucu metotlari söyledir

:

### public GridLayout(int satirlar,int sütunlar)

**publicGridLayout(int satirlar,int sütunlar,int yatay\_bosluk,int dikey\_bosluk);** yatay\_bosluk ve dikey\_basluk pixel olarak alinir. GUI elemanlarinin arasindaki bosluk miktarini tanimlar.

Simdi bir örneklerle bu sinifi açiklamaya çalisalim.

### Program 7.40 GridLayoutTesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class GridLayoutTesti extends Applet

{

private Button B[];

private String S[]={"1","2","3",

"4","5","6"};

public void init()

{

setLayout(new GridLayout(3,2,5,5)); B=new Button[6];

for(int i=0;i<B.length;i++)

{

B[i]=new Button(S[i]);

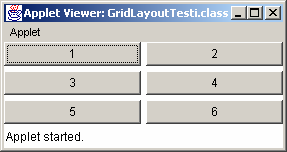
add(B[i]);

}

}

}

### 07037.JPG



**Sekil 7.17 GridLayoutTesti.java programi sonuçlarinin applette görulmesi**

Programin birde JFrame esdegerini inceleyelim.

### Program 7.41 : GridLayoutTestiSWF\_2000.java Programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class GridLayoutTestiSWF\_2000 extends JFrame implements ActionListener

{

private JButton B[];

private String S[]={"Dügme 1","Dügme 2","Dügme 3", "Dügme 4","Dügme 5","Dügme 6"};

private Container c;

public GridLayoutTestiSWF\_2000()

{

super("Grid Layout formatlama Grafik ara yüz (GUI) testi"); c=getContentPane();

c.setLayout(new GridLayout(3,2,5,5)); B=new JButton[6];

for(int i=0;i<B.length;i++)

{

B[i]=new JButton(S[i]); B[i].addActionListener(this); c.add(B[i]);

}

}

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<B.length;i++)

{

if(e.getSource()==B[i]) B[i].setVisible(false); else B[i].setVisible(true);

}

c.validate();

}

public static void main(String[] args)

{

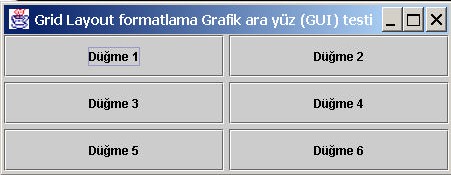
GridLayoutTestiSWF\_2000 pencere= new GridLayoutTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 07038.JPG



**Sekil 7.38 GridLayoutTestiSWF\_2000.java programi sonuçlarinin JFRame’de görülmesi**

* 1. **JTABBEDPANE SINIFI KULLANILARAK FORMATLAMA**

**JTabbedPane sadece swing sinifinda kullanilabilir. Degisik sayfalar halinde bir grafik arayüzü çagirir. Bu sayfalardan istenilen seçilerek kullanilabilir. Her sayfa bagimsiz bir grafik ortami olusturur.**

**Program 7.42 : JTabbedPaneTestiSWF\_2000.java Programi**

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class JTabbedPaneTestiSWF\_2000 extends JFrame implements ActionListener

{

private JButton B[];

private String S[]={"Dügme 1","Dügme 2","Dügme 3", "Dügme 4","Dügme 5","Dügme 6"};

private Container c;

public JTabbedPaneTestiSWF\_2000()

{

super("JTabbedPane formatlama Grafik ara yüz (GUI) testi"); c=getContentPane();

c.setLayout(new BorderLayout(5,5)); B=new JButton[6];

JTabbedPane jtp=new JTabbedPane(); for(int i=0;i<B.length;i++)

{

B[i]=new JButton(S[i]); B[i].addActionListener(this); jtp.addTab(S[i],B[i]);

}

c.add(jtp);

}

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<B.length;i++)

{

if(e.getSource()==B[i]) B[i].setText(S[i]+" dügmeye basildi");

else B[i].setText(S[i]);

}

c.validate();

}

public static void main(String[] args)

{

JTabbedPaneTestiSWF\_2000 pencere= new JTabbedPaneTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 07039.JPG



**Sekil 7.39 JTabbedPaneTestiSWF\_2000.java programi sonuçlarinin JFrame’de görülmesi**

### JSPLITPANE SINIFI KULLANILARAK FORMATLAMA

JSplitPane ekrani ikiye böler. Bu ikiye bölme islemlerini yineleyerek istedigimiz alt parçalari olusturabiliriz. JSplitPane ile bölünen alt parçalar fare ile çekilerek büyütülüp küçültülebilir. Örnek problemdeki

c=getContentPane();

c.setLayout(new BorderLayout(5,5)); B=new JButton[4];

jsp=new JSplitPane[3]; for(int i=0;i<B.length;i++)

{

B[i]=new JButton(S[i]); B[i].addActionListener(this);

}

jsp[0]=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,true,B[0],B[1]); jsp[1]=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,true,B[2],B[3]); jsp[2]=new JSplitPane(JSplitPane.VERTICAL\_SPLIT,true,jsp[0],jsp[1]); c.add(jsp[2]);

kod parçaciginda ekrani önce B[0] ve B[1] dügmelerini yerlestirdigimimiz yatay iki parçaya, sonra B[2] ve B[3] dügmelerini yerlestirdigimz iki yatay parçaya böldükten sonra bu parçalari (jsp[0] ve jsp[1]) iki düsey parçaya ayiriyoruz. Böylece birbirinden bagimsiz dört ayri alt parça (split) olusturuyoruz.

### Program 7.43 : JSplitPaneTestiSWF\_2000.java Programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class JSplitPaneTestiSWF\_2000 extends JFrame implements ActionListener

{

private JButton B[];

private String S[]={"Dügme 1","Dügme 2","Dügme 3","Dügme 4"}; private Container c;

JSplitPane jsp[];

public JSplitPaneTestiSWF\_2000()

{

super("JSplitPane formatlama Grafik ara yüz (GUI) testi"); c=getContentPane();

c.setLayout(new BorderLayout(5,5)); B=new JButton[4];

jsp=new JSplitPane[3];

for(int i=0;i<B.length;i++)

{

B[i]=new JButton(S[i]); B[i].addActionListener(this);

}

jsp[0]=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,true,B[0],B[1]); jsp[1]=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,true,B[2],B[3]); jsp[2]=new JSplitPane(JSplitPane.VERTICAL\_SPLIT,true,jsp[0],jsp[1]); c.add(jsp[2]);

}

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<B.length;i++)

{

if(e.getSource()==B[i]) B[i].setText(S[i]+" dügmeye basildi"); else B[i].setText(S[i]);

}

c.validate();

}

public static void main(String[] args)

{

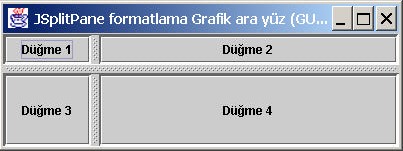
JSplitPaneTestiSWF\_2000 pencere= new JSplitPaneTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 07040.JPG



**Sekil 7.40 JSplitPaneTestiSWF\_2000.java programi sonuçlarinin JFrame’de görülmesi**

* 1. **JSCROLLPANE SINIFI KULLANILARAK FORMATLAMA**

JscrollPane sinifini Jlist sinifini incelerken görmüstük. Bu formatlama sinifi Scrollbar adini verdigimiz yatay ve düsey kaydirma çubuklari yardimiyla küçük bir pencereye sigmayacak kadar büyükolan grafiklerin kullanilmasini saglar.

### Program 7.44 : JScrollPaneTestiSWF\_2000.java Programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class JScrollPaneTestiSWF\_2000 extends JFrame implements ActionListener

{

private JButton B[];

private String S[]={"Dügme 1","Dügme 2","Dügme 3","Dügme 4",

"Dügme 5","Dügme 6","Dügme 7","Dügme 8","Dügme 9","Dügme 10","Dügme 11",

"Dügme 12","Dügme 13","Dügme 14","Dügme 15","Dügme 16"}; private Container c;

JScrollPane jscroll;

JPanel jsp;

public JScrollPaneTestiSWF\_2000()

{

super("JScrollPane formatlama Grafik ara yüz (GUI) testi"); c=getContentPane();

c.setLayout(new BorderLayout(5,5)); B=new JButton[16];

jsp=new JPanel();

for(int i=0;i<B.length;i++)

{

B[i]=new JButton(S[i]); B[i].addActionListener(this); jsp.add(B[i]);

}

jscroll=new JScrollPane(jsp); c.add(jscroll,BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

for(int i=0;i<B.length;i++)

{

if(e.getSource()==B[i]) B[i].setText(S[i]+" dügmeye basildi"); else B[i].setText(S[i]);

}

c.validate();

}

public static void main(String[] args)

{

JScrollPaneTestiSWF\_2000 pencere= new JScrollPaneTestiSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,200);

pencere.setVisible(true);

}

}

### 07041.JPG



**Sekil 7.41 JScrollPaneTestiSWF\_2000.java programi sonuçlarinin JFrame’de görülmesi**

### PANEL VE JPANEL SINIFI

Daha kompleks formatlamalara gereksinme duyuldugunda panel sinifini kullanabiliriz. Panel sinifi applete yerlestirebilecegimiz alt bolgeler olusturur. Hemen bir ornekle paneli nasil kulandigimizi açiklamaya çalisalim.

### Program 7.45 PanelTesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class Paneltesti extends Applet

{

private Panel buttonPanel; private Button buttons[]; public void init()

{

buttonPanel=new Panel(); buttons=new Button[5];

buttonPanel.setLayout(new GridLayout(1,buttons.length)); for(int i=0;i<buttons.length;i++)

{

buttons[i]=new Button("Button "+(i+1)); buttonPanel.add(buttons[i]);

}

setLayout(new BorderLayout()); add(buttonPanel,BorderLayout.SOUTH);

}

}

### 07042JPG



**Sekil 7.42PanelTesti.java programi ve sonuçlarinin applette görulmesi**

JPanel sinifini daha önce çesitli alistirmalarda zaten kullanmistik. JPanel’i formatlama isleminde ayni Panel sinifinda oldugu gibi bir alt formatlama elemani olarak kullanabiliriz. Zaten bütün swing programlarinin temeli de JPanel sinifidir. Swing gurubunda bir üstteki problemin aynisini JApplet olarak yineledik. Awt penceresi ve swing penceresi arasindaki en önemli farkin awt de add terimini appletin bir alt terimi gibi kullanabilirken, swingde getContentPane() deyimiyle ana JPanel’i çagirmak ve çizimleri bunun üzerine yapmak oldugunu bir kere daha yineleyelim.

### Program 7.46 : PanelTestiSWA.java Programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

public class PaneltestiSWA extends JApplet

{

private JPanel buttonPanel; private JButton buttons[];

public void init()

{

buttonPanel=new JPanel(); buttons=new JButton[5];

buttonPanel.setLayout(new GridLayout(1,buttons.length)); for(int i=0;i<buttons.length;i++)

{

buttons[i]=new JButton("Dügme "+(i+1)); buttonPanel.add(buttons[i]);

}

Container c=getContentPane(); c.setLayout(new BorderLayout()); c.add(buttonPanel,BorderLayout.SOUTH);

}

}

### 07043JPG



**Sekil 7.43 PanelTestiSWA.java programi ve sonuçlarinin JAppletde appletviewer ile görulmesi**

Su ana kadarki format örneklerinde anlatimi basitlestirmek amaciyla hep dügmeler (button) kullandik. Simdi de baska bir problemde Panel sinifinin kullanilmasina bakalim :

### Program 7.47 : PolinomunKokleri.java programi

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class PolinomunKokleri extends Applet implements ActionListener

{

// polinom a\*x^2+b\*x+c=0 private Panel YaziPaneli; Label prompt1;

Label prompt2; Label prompta; TextField inputa; // Label promptb; // TextField inputb; // Label promptc; // TextField inputc; //

double a,b,c; //polinomun katsayilari double delta=0.0; //determinant double x1,x2; //kokler

public void init()

{

setBackground(Color.lightGray); YaziPaneli=new Panel();

YaziPaneli.setFont(new Font("Serif",Font.BOLD,12)); YaziPaneli.setLayout( new GridLayout(4,2) ); prompt1=new Label("a\*x^2+b\*x+c = 0"); prompt2=new Label("Polinomunun kokleri"); prompta=new Label(" a : ");

inputa =new TextField(5); inputa.addActionListener(this); promptb=new Label(" b : "); inputb =new TextField(5); inputb.addActionListener(this); promptc=new Label(" c : "); inputc =new TextField(5); inputc.addActionListener(this); YaziPaneli.add(prompt1); YaziPaneli.add(prompt2); YaziPaneli.add(prompta); YaziPaneli.add(inputa); YaziPaneli.add(promptb); YaziPaneli.add(inputb); YaziPaneli.add(promptc); YaziPaneli.add(inputc);

add(YaziPaneli);

}

public void paint(Graphics g)

{

g.setColor(Color.blue);

g.setFont(new Font("Serif",Font.BOLD,14)); g.drawString("Polinomun Kokleri: ",70,175); g.setColor(Color.black);

g.setFont(new Font("Serif",Font.BOLD,12)); if(delta<0)

{

g.drawString("Kok 1 :"+(-b/2.0/a)+"+ i\* "+Math.sqrt(- delta)/2.0/a,100,190 ); g.drawString("Kok 2 :"+(-b/2.0/a)+"- i\* "+Math.sqrt(-delta)/2.0/a,100,205);

}

if(delta==0)

{

g.drawString("Kok 1 :"+(-b/2.0/a),100,190 );

g.drawString("Kok 2 :"+(-b/2.0/a),100,205 );

}

if(delta0)

{

x1=-b/2.0/a+Math.sqrt(delta)/20/a; x2=-b/2.0/a-Math.sqrt(delta)/2.0/a; g.drawString("Kok 1 :"+x1,100,190 ); g.drawString("Kok 2 :"+x2,100,205);

}

}

public void actionPerformed( ActionEvent e)

{

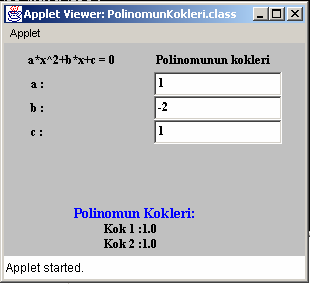
Double sayi1=new Double(inputa.getText() ); Double sayi2=new Double(inputb.getText() ); Double sayi3=new Double(inputc.getText() ); a=sayi1.doubleValue(); b=sayi2.doubleValue(); c=sayi3.doubleValue();

delta=b\*b-4.0\*a\*c; repaint();

}

}

### 07044.JPG



**Sekil 7.44 Ikinci dereceden denklemin koklerini hesaplayan PolinomunKokleri.java programi ve sonuçlarinin applette görulmesi**

Simdi ayni problemi JPanel kullanarak Japplet içinde gerçeklestirelim.

### Program 7.48 : PolinomunKokleriSWA.java programi

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

public class PolinomunKokleriSWA extends JApplet implements ActionListener

{

// polinom a\*x^2+b\*x+c=0 private JPanel YaziPaneli; JLabel prompt1;

JLabel prompt2; JLabel prompta; JTextField inputa; // JLabel promptb; // JTextField inputb; // JLabel promptc; // JTextField inputc; // JTextArea cikti;

double a,b,c; //polinomun katsayilari double delta=0.0; //determinant double x1,x2; //kokler

public void init()

{

Container c=getContentPane(); cikti=new JTextArea(); cikti.setBackground(c.getBackground()); setBackground(Color.lightGray); YaziPaneli=new JPanel();

YaziPaneli.setFont(new Font("Serif",Font.BOLD,12)); YaziPaneli.setLayout( new GridLayout(4,2) ); prompt1=new JLabel("a\*x^2+b\*x+c = 0"); prompt2=new JLabel("Polinomunun kokleri"); prompta=new JLabel(" a : ");

inputa =new JTextField(); inputa.addActionListener(this); promptb=new JLabel(" b : "); inputb =new JTextField(); inputb.addActionListener(this); promptc=new JLabel(" c : "); inputc =new JTextField(); inputc.addActionListener(this); YaziPaneli.add(prompt1); YaziPaneli.add(prompt2); YaziPaneli.add(prompta); YaziPaneli.add(inputa); YaziPaneli.add(promptb); YaziPaneli.add(inputb); YaziPaneli.add(promptc); YaziPaneli.add(inputc);

c.add(YaziPaneli,BorderLayout.NORTH); c.add(cikti,BorderLayout.CENTER);

}

public String toString()

{

String s="";

s=s+"Polinomun Kokleri: \n"; if(delta<0)

{

s+= "Kok 1 :"+(-b/2.0/a)+"+ i\* "+Math.sqrt(- delta)/2.0/a+"\n"; s+= "Kok 2 :"+(-b/2.0/a)+"- i\* "+Math.sqrt(-delta)/2.0/a+"\n";

}

if(delta==0)

{

s+= "Kok 1 :"+(-b/2.0/a)+"\n";

s+= "Kok 2 :"+(-b/2.0/a)+"\n";

}

if(delta>0)

{

x1=-b/2.0/a+Math.sqrt(delta)/20/a; x2=-b/2.0/a-Math.sqrt(delta)/2.0/a; s+= "Kok 1 :"+x1+"\n";

s+= "Kok 2 :"+x2+"\n";

}

return s;

}

public void actionPerformed( ActionEvent e)

{

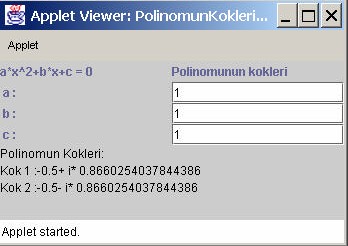
Double sayi1=new Double(inputa.getText() ); Double sayi2=new Double(inputb.getText() ); Double sayi3=new Double(inputc.getText() ); a=sayi1.doubleValue(); b=sayi2.doubleValue(); c=sayi3.doubleValue();

delta=b\*b-4.0\*a\*c; cikti.setText(toString()); repaint();

}

}

### 07045.JPG



**Sekil 7.45 Ikinci dereceden denklemin koklerini hesaplayan PolinomunKokleriSWA.java programi ve sonuçlarinin applette görulmesi**

### JOPTIONPANE SINIFI

Su ana kadar gördügümüz siniflar grafik arayüzü dizayni yapmaya yarayan siniflardi. Java swing gurubunda ayni zamanda kullanilmaya hazir bir çok grafik arayüzleri tanimlanmistir. Bunlarin en çok kullanilani, ve bu kitapta da su ana kadar yogun olarak kullanageldigimiz JOPtionPane sinifidir. Bu sinifin ana gayesi pop-up mesaj ve girdi ortami saglamaktir. JoptionPane sinifini programimizda kurucu metod kullanarak kurmamiz gerekmez. Temel olarak direk statik metodlarini çagiririz. JoptionPane panelinde genellikle bir resim,bir yazi alani, bir girdi alani (JtextField alani), birde Seçilebilen dügmeler bulunur.

JoptionPane’in girdi/çikti olarak kullanilabilen baslica metodlari sunlardir :

public static String showInputDialog(Object mesaj)

public static String showInputDialog(Component anapencere,Object mesaj)

public static String showInputDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü)

public static String showInputDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü, Icon resim, Object[] listeseçmedegerleri, Object girisdegeri)

public static String showInternalInputDialog(Component anapencere,Object mesaj)

public static String showInternalInputDialog(Component anapencere,Object mesaj, String baslik,int mesajtürü) public static String showInternalInputDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü, Icon resim, Object[] listeseçme degerleri,, Object girisdegeri)

public static void showMessageDialog(Component anapencere,Object mesaj);

public static void showMessageDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü);

public static void showMessageDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü, Icon resim);

public static void showInternalMessageDialog(Component anapencere,Object mesaj);

public static void showInternalMessageDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü); public static void showInternalMessageDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü, Icon resim);

public static int showConfirmDialog(Component anapencere,Object mesaj);

public static int showConfirmDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü);

public static int showConfirmDialog(Component anapencere,Object mesaj,String baslik,int opsiyontürü,int mesajtürü);

public static int showConfirmDialog(Component anapencere,Object mesaj,String baslik,int opsiyontürü,int mesajtürü,Icon resim);

public static int showInternalConfirmDialog(Component anapencere,Object mesaj);

public static int showInternalConfirmDialog(Component anapencere,Object mesaj,String baslik,int mesajtürü); public static int showInternalConfirmDialog(Component anapencere,Object mesaj,String baslik,int opsiyontürü,int mesajtürü);

public static int showInternalConfirmDialog(Component anapencere,Object mesaj,String baslik,int opsiyontürü,int mesajtürü,Icon resim);

public static int showOptionDialog(Component anapencere,Object mesaj,String baslik,int opsiyontürü,int mesajtürü,Icon resim, Object[] listeseçmedegerleri,Object listedekiilkdeger);

public static int showInternalOptionDialog(Component anapencere,Object mesaj,String baslik,int opsiyontürü,int mesajtürü,Icon resim, Object[] listeseçmedegerleri,Object listedekiilkdeger);

mesajtürü degiskeni (int) alabildigi degerler sunlardir : JOptionPane.ERROR\_MESSAGE JOptionPane.INFORMATION\_MESSAGE

JOptionPane.PLAIN\_MESSAGE JOptionPane.QUESTION\_MESSAGE JOptionPane.WARNING\_MESSAGE Opsiyon türü degiskeni

JOptionPane.DEFAULT\_OPTION : Sadece OK dügmesi verir JOptionPane.OK\_CANCEL\_OPTION : Ok ve Cancel(iptal et) dügmesi verir JOptionPane.YES\_NO\_CANCEL\_OPTION : “Yes (evet) ve No(hayir) ve Cancel(Iptal et) dügmeleri verir JOptionPane.YES\_NO\_OPTION : “Yes (evet) ve No(hayir) dügmeleri verir

OptionDialog kullanildiginda Metod bir integer deger gönderir JOptionPane.CANCEL\_OPTION JOptionPane.CLOSED\_OPTION

JOptionPane.NO\_OPTION JOptionPane.OK\_OPTION JOptionPane.YES\_OPTION JOptionPane.UNINITILIZED\_VALUE dur.

### Program 7.49 : JoptionPane1.java Programi

import javax.swing.\*; import java.awt.\*;

public class JOptionPane1

{

public static void main(String[] args)

{

Object[] o=new Object[]{"Turhan","Meral","Irfan","Hatice","Osman",

"Nurhan","Birsen","Ali","Veli","Mustafa","Ahmet","Mehmet","Ayhan"}; ImageIcon resim=new ImageIcon("images/Kus.gif"); JOptionPane.showInputDialog(null,"Lütfen bir isim seçin : ",

"JOptionPane örnek 1",JOptionPane.QUESTION\_MESSAGE,resim,o,"Turhan"); System.exit(0);

}

}

### 07046.JPG



**Sekil 7.46 JoptionPane1.java programi sonucunun JoptionPane çiktisinda görülmesi**

* 1. **SWING, JCOLORCHOOSER SINIFI**

JcolorChooser’da JoptionPane gibi kullanilmaya hazir bir dialog sinifidir. Renk seçmeye yarar ve renk seçilmesi gereken her yerde kullanilabilir. Bir önceki bölümdeki bir alistirmada bu özelligi kullanmistik. Ikinci bir örnekle pekistirelim :

### Program 7.50 : ColorChooserDemo.java Programi

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.event.\*;

import javax.swing.colorchooser.\*;

public class ColorChooserDemo extends JFrame { public ColorChooserDemo() {

super("ColorChooser Renk seçici");

//Pencerenin tepesinde bir JLabel açalim

final JLabel etiket = new JLabel("Java Programlama diline Hos geldiniz!", JLabel.CENTER);

etiket.setForeground(Color.yellow); etiket.setBackground(Color.blue); etiket.setOpaque(true);

etiket.setFont(new Font("SansSerif", Font.BOLD, 24)); etiket.setPreferredSize(new Dimension(100, 65));

JPanel etiketPaneli = new JPanel(new BorderLayout()); etiketPaneli.add(etiket, BorderLayout.CENTER); etiketPaneli.setBorder(BorderFactory.createTitledBorder("etiket"));

//JColorChooser sinifini çagirip rengi degistir

final JColorChooser tcc = new JColorChooser(etiket.getForeground()); tcc.getSelectionModel().addChangeListener(

new ChangeListener() {

public void stateChanged(ChangeEvent e) { Color newColor = tcc.getColor(); etiket.setForeground(newColor);

}

}

);

tcc.setBorder(BorderFactory.createTitledBorder(

"Yazi Rengini Seçiniz"));

//Add the components to the demo frame Container c = getContentPane(); c.add(etiketPaneli, BorderLayout.CENTER); c.add(tcc, BorderLayout.SOUTH);

}

public static void main(String[] args) { JFrame frame = new ColorChooserDemo();

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

frame.pack(); frame.setVisible(true);

}

}

### 07047.JPG



**Sekil 7.47 ColorChooserDemo.java programi sonucunun JFrame çiktisinda görülmesi**

### JFILECHOOSER SINIFI

JFileChooser’da JcolorChooser gibi hazir bir program demetidir. Bilgisayardan bir dosyayi seçme ileminde yararlanilir. Dosya prosesleri konusunda Java Girdi çikti programlanmasi bölümüne bakiniz.

### Program 7.51 : FileChooserDemo.java Programi

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import javax.swing.\*;

import javax.swing.filechooser.\*;

public class FileChooserDemo extends JFrame {

static private final String newline = System.getProperty("line.separator"); public FileChooserDemo() {

super("FileChooser Dosya seçme örnegi"); final JTextArea log = new JTextArea(5,20); log.setMargin(new Insets(5,5,5,5));

JScrollPane logScrollPane = new JScrollPane(log);

//dosya seçiciyi yarat

final JFileChooser fc = new JFileChooser();

//aç dügmesi yarat

ImageIcon openIcon = new ImageIcon("images/open.gif"); JButton acmaDugmesi = new JButton("Dosya aç", openIcon); acmaDugmesi.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int returnVal = fc.showOpenDialog(FileChooserDemo.this);

if (returnVal == JFileChooser.APPROVE\_OPTION) { File file = fc.getSelectedFile();

//gerçek dosyanin açilma yeri.

log.append("Dosya açiliyor: " + file.getName() + "." + newline);

} else {

log.append("Dosyanin açilmasi kullanici tarafindan iptal edildi." + newline);

}

}

});

//kaydet dügmesi yarat

ImageIcon saveIcon = new ImageIcon("images/save.gif"); JButton kayitDugmesi = new JButton("Dosya kaydet", saveIcon); kayitDugmesi.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int returnVal = fc.showSaveDialog(FileChooserDemo.this);

if (returnVal == JFileChooser.APPROVE\_OPTION) { File file = fc.getSelectedFile();

//gerçek dosyanin kaydedilme yeri..

log.append("Kaydediliyor : " + file.getName() + "." + newline);

} else {

log.append("Dosyanin kaydedilmesi kullanici tarafindan iptal edildi" + newline);

}

}

});

//Güzel bir görünüm için aç ve kapa dügmelerini ayri bir panele yerlestir. JPanel buttonPanel = new JPanel();

buttonPanel.add(acmaDugmesi); buttonPanel.add(kayitDugmesi);

//Dügmeleri pencereye ekle Container c = getContentPane();

c.add(buttonPanel, BorderLayout.NORTH); c.add(logScrollPane, BorderLayout.CENTER);

}

public static void main(String s[]) {

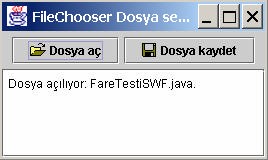
JFrame frame = new FileChooserDemo(); frame.addWindowListener(new BasicWindowMonitor()); frame.pack();

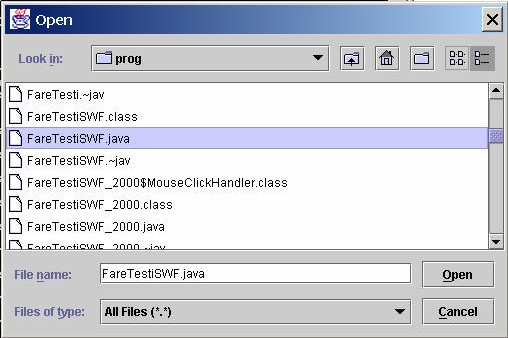
frame.setVisible(true);

}

}

### 07048.JPG



**Sekil 7.48 FileChooserDemo.java programi sonucunun JFrame çiktisinda görülmesi 07049.JPG**

**Sekil 7.49 FileChooserDemo.java programi dosya açicisinin JFileChooser çiktisinda görülmesi**

### JAVA JAR (JAVA ARCHIVES - JAVA ARSIVI) YAPILARININ KULLANIMI

**JAR** çesitli sinif ve ilgili dosyalari bir arada bir pakete dönüstürmek için gelistirilmis bir teknolojidir. JAR teknolojisi peogramlarin tasinmasini ve yeniden yüklenmesini daha kolay bir hale getirir. Ayni zamanda JAR dosyasinin içindeki elementler sikistirilmis halde bulunduklarindan daha az yer kaplarlar. Bu yüzden bir JAR dosyasinin internet üzerinden aktarilmasi, açik dosyalarin tek tek aktarilmasindan daha kisa süren bir prosestir.

Yeni JAR dosyalarini olusturmak için :

eger **isim.jar** dosyasini olusturmak ve bu dosyaya isim.class, program1.class ve resim.gif dosyalarini aktarmak istersek :

### jar cf isim.jar isim.class program1.class resim.gif

Alistirma 5deki H7AL1 örnegi için JAR dosyasi olusturmak istersek :

### jar cf H7AL1.jar H7AL1.class yildiz.class renk.class polar.class

komutunu kullanabiliriz. Problemi inceledignizde JAR dosyasinin içine ilave edilen tüm siniflarin bu problemle direk olarak ilgili oldugunu görebilirsiniz.

jar dosyasinin içindeki dosyalara bakmak için :

### jar tf isim.jar

jar dosyasi içindeki dosyalarin listesini almak için :

### jar xf isim.jar

jar dosyasini applet içinden çalistirmak için html içinde :

### <applet code=AppletClassIsmi.class archive="JarDosyasiIsmi.jar" width=width height=height

**</applet**

komutu kullanilir. Örnek olarak yine alistirma 5 de kullanilan jar dosyasinin html dosyasini verebiliriz.

<HTML

<HEAD

<TITLE yildiz cizim ornegi</TITLE

</HEAD

<BODY

<H3yildiz cizim ornegi </H3

<applet code=H7AL1.class archive="H7AL1.jar" width=600 height=500

</applet

</BODY

</HTML

Jar komutuyla kullanilan kontrol karakterleri Tablo 7.1 de verilmistir.

### Tablo 7.1 Jar kontrol komut karekterleri ve anlamlari

**c f m t x**

Yeni arsiv yarat

Listedeki ilk ismi arsiv dosyasinin ismi olarak tanimla Listedeki ikinci ismi arsiv dosyasinin ismi olarak tanimla arsiv içerigini listele

dosyalar arsivden geri çagirilacaktir. eger listede sadece bir isim varsa bunu jar dosyasi

olarak alir ve içindeki tüm dosyalari listeler, eger ilk simden sonra komutta baska isimler yer aliyorsa, sadece bu isimleri (eger jarda mevcutsa) listeler.

**0** sikistirma kullanma

### ALISTIRMALAR

1. FareTesti.java programini ve bir önceki bölümde inceledigimiz cizgiciz programini birlestirerek fareyi kullanarak ekrana cizgi çizdiriniz (cizginin baslangic ve bitis noktalari fare tarafindan belirlenecektir)
2. FareTesti.java programini ve gecen hafta inceledigimiz drawRect metotunu birlestirerek fareyi kallanarak ekrana dikdortrtgen çizdiriniz (dikdortgenin baslangic ve bitis noktalari fare tarafindan belirlenecektir)
3. FareTesti.java programini ve gecen hafta inceledigimiz drawOval metotunu birlestirerek fareyi kallanarak ekrana daire çizdiriniz (dairenin merkezi ve capi fare tarafindan belirlenecektir)

### Resimli bir JcheckBox örnegi. Java döküman kütüphanesinden alinip adapte edilen bu programda 16 resimden birisi JCheckBox kullanilarak yapilan seçime göre JLAbel kullanilarak gösteriliyor.

**Program 7.52 : CheckBoxDemo.java Programi**

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class CheckBoxDemo extends JPanel { JCheckBox chinButton;

JCheckBox glassesButton;

JCheckBox hairButton;

JCheckBox teethButton;

/\*

* Dört degisik seçim imkani tanimlanmistir.
* Bu seçimler 16 degisik dosyadaki resimlerden birini çagirmaktadir.
* resim isimleri "geek-XXXX.gif" seklinde olup
* buradaki XXXX 16 seçimden birini gösterir.
* seçim olasiliklari

---- // hiçbirsey seçilmemis

c--- // bir seçim

-g--

--h-

---t

cg-- // iki seçim c-h-

c--t

-gh-

-g-t

--ht

-ght // üç seçim c-ht

cg-t cgh-

cght // dört seçim seçimlerin anlamlari :

c : çene

g : gözlük h : saç

t : dis

\*/

StringBuffer choices;

JLabel pictureLabel;

public CheckBoxDemo() {

// Create the check boxes

chinButton = new JCheckBox("Çene"); chinButton.setMnemonic('e'); chinButton.setSelected(true);

glassesButton = new JCheckBox("Gözlük"); glassesButton.setMnemonic('G'); glassesButton.setSelected(true);

hairButton = new JCheckBox("Saç"); hairButton.setMnemonic('S'); hairButton.setSelected(true);

teethButton = new JCheckBox("Dis"); teethButton.setMnemonic('D');

teethButton.setSelected(true);

// listener programlarini çagir

CheckBoxListener myListener = new CheckBoxListener(); chinButton.addItemListener(myListener); glassesButton.addItemListener(myListener); hairButton.addItemListener(myListener); teethButton.addItemListener(myListener);

// orijinal olarak dörtlü resmi seç choices = new StringBuffer("cght");

// JLAbel'i resimli olarak olustur pictureLabel = new JLabel(new ImageIcon(

"images/geek/geek-"

+ choices.toString()

+ ".gif")); pictureLabel.setToolTipText(choices.toString());

// CheckBoxlari JPanel'e yerlestir. JPanel checkPanel = new JPanel();

checkPanel.setLayout(new GridLayout(0, 1)); checkPanel.add(chinButton); checkPanel.add(glassesButton); checkPanel.add(hairButton); checkPanel.add(teethButton);

setLayout(new BorderLayout()); add(checkPanel, BorderLayout.WEST); add(pictureLabel, BorderLayout.CENTER);

setBorder(BorderFactory.createEmptyBorder(20,20,20,20));

}

/\*\* check box'lari dinle. \*/

class CheckBoxListener implements ItemListener { public void itemStateChanged(ItemEvent e) {

int index = 0; char c = '-';

Object source = e.getItemSelectable();

if (source == chinButton) { index = 0;

c = 'c';

} else if (source == glassesButton) { index = 1;

c = 'g';

} else if (source == hairButton) { index = 2;

c = 'h';

} else if (source == teethButton) { index = 3;

c = 't';

}

if (e.getStateChange() == ItemEvent.DESELECTED) c = '-';

choices.setCharAt(index, c); pictureLabel.setIcon(new ImageIcon(

"images/geek/geek-"

+ choices.toString()

+ ".gif")); pictureLabel.setToolTipText(choices.toString());

}

}

public static void main(String s[])

{

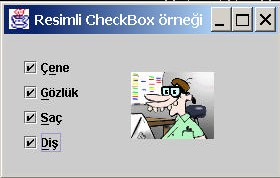
JFrame frame = new JFrame("Resimli CheckBox örnegi"); frame.addWindowListener(new BasicWindowMonitor()); frame.setContentPane(new CheckBoxDemo()); frame.pack();

frame.setVisible(true);

}

}

### 07050.JPG



**Sekil 7.50 CheckBoxDemo.java çiktisinin JFrame’de görülmesi**

1. Sirin bir JComboBox örnegi : Listeden yapilan seçime göre resim degisecektir. Bu programda java döküman kütüphanesinden alinip türkçeye adapte edilmistir.

### Program 7.53 : ComboBoxDemo.java Programi

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class ComboBoxDemo extends JPanel { JLabel picture;

public ComboBoxDemo() {

String[] petStrings = { "Kus", "Kedi", "Köpek", "Tavsan", "Domuz" };

// combo box yarat ve domuzu seç,

JComboBox petList = new JComboBox(petStrings); petList.setSelectedIndex(4); petList.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { JComboBox cb = (JComboBox)e.getSource(); String petName = (String)cb.getSelectedItem();

picture.setIcon(new ImageIcon("images/" + petName + ".gif"));

}

});

// Resmi JLabel'a yükle

picture = new JLabel(new ImageIcon("images/" +

petStrings[petList.getSelectedIndex()] + ".gif"));

picture.setBorder(BorderFactory.createEmptyBorder(10,0,0,0));

// tercih edilen boy olarak en büyük resmin boyu seçildi.

// bu deger daha sonra program tarafindan degistirilecektir. picture.setPreferredSize(new Dimension(177, 122+10));

setLayout(new BorderLayout()); add(petList, BorderLayout.NORTH); add(picture, BorderLayout.SOUTH);

setBorder(BorderFactory.createEmptyBorder(20,20,20,20));

}

public static void main(String s[])

{

JFrame frame = new JFrame("ComboBoxDemo"); frame.addWindowListener(new BasicWindowMonitor()); frame.getContentPane().add(new ComboBoxDemo(), BorderLayout.CENTER); frame.pack();

frame.setVisible(true);

}

}

### 07051.JPG



**Sekil 7.51 ComboBoxDemo.java çiktisinin JFrame’de görülmesi**

**5.** Appletin alt bölgesinde(SOUTH) iki sira, her sirada beser dügmeden on dügme bulunan bir applet olusturunuz..

Appletin tepesinde bir TextField olusturunuz. Her dügmeye basildiginda textfieldde degisik bir yazi görünsün. Yazilari 2x5 lik bir String degiskeninden okuyarak Text fielde yükleyiniz.

### Program 7.16 Panel10.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class Panel10 extends Applet implements ActionListener

{

private Panel buttonPanel; private Button buttons[]; String s[]=

{"Birinci Butona Basildi", "Ikinci Butona Basildi", "Ucuncu Butona Basildi", "Dorduncu Butona Basildi", "Besinci Butona Basildi", "Altinci Butona Basildi", "Yedinci Butona Basildi", "Sekizinci Butona Basildi",

"Dokuzunccu Butona Basildi", "Onuncu Butona Basildi"}; TextField T;

public void init()

{

T=new TextField(s[0]); buttonPanel=new Panel(); buttons=new Button[10];

buttonPanel.setLayout(new GridLayout(2,buttons.length/2)); for(int i=0;i<buttons.length;i++)

{

buttons[i]=new Button("Button "+(i+1)); buttons[i].addActionListener(this); buttonPanel.add(buttons[i]);

}

setLayout(new BorderLayout()); add(buttonPanel,BorderLayout.SOUTH); add(T,BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent e)

{

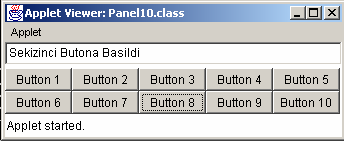
for(int i=0;i<buttons.length;i++)

{ if(e.getSource()==buttons[i]) T.setText(s[i]);} validate();

}

}

### 07052.JPG



**Sekil 7.52 Panel10.html çiktisi**

**4.** H7AL1.java programini incele ve çalistir.

### Program 7.54 H7AL1.java

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import yildiz;

import renk;

public class H7AL1 extends Applet implements ActionListener,ItemListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk Label kutu1;

Label kutu2; Label kutu3; Label kutu4; Label kutu5;

Label renk1;

Label renk2;

private String Renkler[]={"kirmizi","mavi","siyah","camgobegi","koyugri","gri","yesil","acikgri", "mor","portakal","pembe","beyaz","sari","acikmavi","lacivert"};

private List renkListesi,arkaPlanRenkListesi; private renk

r[]={renk.kirmizi,renk.mavi,renk.siyah,renk.camgobegi,renk.koyugri,renk.gri,renk.yesil,renk.acikgri,renk.mor, renk.portakal,renk.pembe,renk.beyaz,renk.sari,renk.acikmavi,renk.lacivert};

TextField kutugir1; TextField kutugir2; TextField kutugir3; TextField kutugir4; TextField kutugir5; Panel p,p1;

public void init(){

//not : init metodunda tum degiskenlerin ilk degerleri verilmelidir. setBackground(renk.beyaz);

p=new Panel();

p.setLayout(new GridLayout(3,4)); p.setBackground(renk.beyaz);

//3\*4 tablo p paneli haz• rland• p1=new Panel();

p1.setLayout(new GridLayout(1,4)); p1.setBackground(renk.beyaz); renkListesi=new List(3,false); arkaPlanRenkListesi=new List(3,false); kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

renk1=new Label("renk : "); renk2=new Label("arka plan rengi :"); xi1=300;

yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1);

kutugir5.setText(""+aci1); renk=new renk(255,200,0); p1.add(renk1); p1.add(renkListesi); p1.add(renk2); p1.add(arkaPlanRenkListesi);

//elemanlar p ve p1 paneline p ve p1 paneli de aplete eklendi kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); renkListesi.addActionListener(this); arkaPlanRenkListesi.addActionListener(this); renkListesi.addItemListener(this); arkaPlanRenkListesi.addItemListener(this);

for(int i=0;i<r.length;i++)

{

renkListesi.add(Renkler[i]); arkaPlanRenkListesi.add(Renkler[i]);

}

add(p,BorderLayout.NORTH); add(p1,BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex(); if(indeks!=-1)

{

setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk1.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

}

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void itemStateChanged(ItemEvent e)

{

showStatus(" renk: "+ renkListesi.getSelectedItem()+" arka plan: "+ arkaPlanRenkListesi.getSelectedItem());

}

public void paint(Graphics g)

{

int indeks=renkListesi.getSelectedIndex(); if(indeks!=-1)

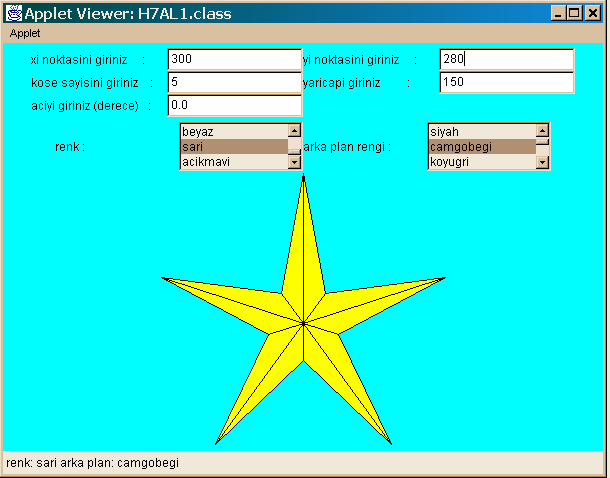
g.setColor(r[indeks]); else

g.setColor(renk.mavi); yildiz.fillYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); yildiz.drawYildiz1(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

}

### 07053.JPG



**Sekil 7.53 H7AL1.html çiktisi**

**5.** H7AL1\_2000.java programini incele ve çalistir.

### Program 7.54 H7AL1\_2000.java

import javax.swing.\*; import javax.swing.event.\*; import java.awt.\*;

import java.awt.event.\*; import yildizSW; import yildizPanelSWP; import renk;

public class H7AL1\_2000 extends JApplet implements ActionListener, ListSelectionListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk JLabel kutu1;

JLabel kutu2; JLabel kutu3; JLabel kutu4; JLabel kutu5; JLabel renk2;

private String Renkler[]={"kirmizi","mavi","siyah","camgobegi", "koyugri","gri","yesil","acikgri","mor","portakal","pembe","beyaz",

"sari","acikmavi","lacivert"}; private JList arkaPlanRenkListesi;

private renk r[]={renk.kirmizi,renk.mavi,renk.siyah,renk.camgobegi, renk.koyugri,renk.gri,renk.yesil,renk.acikgri,renk.mor,renk.portakal, renk.pembe,renk.beyaz,renk.sari,renk.acikmavi,renk.lacivert}; JTextField kutugir1;

JTextField kutugir2; JTextField kutugir3; JTextField kutugir4; JTextField kutugir5; JPanel p,p1,p2; yildizPanelSWP yp; Container c;

public void init(){

//not : init metodunda tüm degiskenlerin ilk degerleri verilmelidir. c=getContentPane();

setBackground(renk.beyaz); p=new JPanel();

p.setLayout(new GridLayout(3,4)); p.setBackground(renk.beyaz);

//3\*4 tablo p paneli hazirlandi p1=new JPanel();

p1.setLayout(new GridLayout(1,4)); p1.setBackground(renk.beyaz); arkaPlanRenkListesi=new JList(Renkler); arkaPlanRenkListesi.setVisibleRowCount(3);

arkaPlanRenkListesi.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION); kutu1=new JLabel("xi noktasini giriniz :");

p.add(kutu1);

kutugir1=new JTextField(3); p.add(kutugir1);

kutu2=new JLabel("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new JTextField(3); p.add(kutugir2);

kutu3=new JLabel("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new JTextField(3); p.add(kutugir3);

kutu4=new JLabel("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new JTextField(3); p.add(kutugir4);

kutu5=new JLabel("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new JTextField(3); p.add(kutugir5);

renk2=new JLabel("arka plan rengi :"); xi1=300;

yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1);

kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1); kutugir5.setText(""+aci1); renk=new renk(255,200,0);

//elemanlar p ve p1 paneline p ve p1 paneli de aplete eklendi kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); arkaPlanRenkListesi.addListSelectionListener(this); JScrollPane aprl=new JScrollPane(arkaPlanRenkListesi); p1.add(renk2);

p1.add(aprl);

yp=new yildizPanelSWP(xi1,yi1,yaricap1,n1,aci1); JPanel p2=new JPanel();

p2.setLayout(new GridLayout(2,1)); p2.add(p);

p2.add(p1); c.add(p2,BorderLayout.NORTH); c.add(yp,BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex(); if(indeks!=-1)

{

c.setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

}

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue(); yp.setYildizPanelSWP(xi1,yi1,yaricap1,n1,aci1); yp.setBackground(r[indeks]);

repaint();

}

public void valueChanged(ListSelectionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex(); if(indeks!=-1)

{

c.setBackground(r[indeks]);

yp.setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

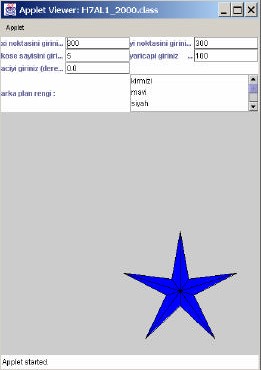
}

repaint();

}

}

### 07054.JPG



**Sekil 7.54 H7AL1\_2000.java çiktisi**

1. H7AL2.java programini incele ve çalistir. Bu program fare kullanarak yildizi çizdirmektedir. Fare’yi sol klik ettigimiz noktayi merkez olarak almakta, fareyi sürüdügümüz nokta ve açiyi çizdirme açisi olarak almakta fare dügmesini biraktigimiz noktada yildiz çapini almakta ve yilsizi çizmektedir. Renkler girdi listelerinden degistirilebilir.

### Program 7.55 H7AL2.java, Fare testi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import yildiz;

import renk;

public class H7AL2 extends Applet

implements ActionListener,ItemListener,MouseListener,MouseMotionListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk

Label kutu1; Label kutu2; Label kutu3; Label kutu4; Label kutu5; Label renk1; Label renk2;

private String Renkler[]={"kirmizi","mavi","siyah","camgobegi","koyugri","gri","yesil","acikgri", "mor","portakal","pembe","beyaz","sari","acikmavi","lacivert"};

private List renkListesi,arkaPlanRenkListesi; private renk

r[]={renk.kirmizi,renk.mavi,renk.siyah,renk.camgobegi,renk.koyugri,renk.gri,renk.yesil,renk.acikgri,renk.mor, renk.portakal,renk.pembe,renk.beyaz,renk.sari,renk.acikmavi,renk.lacivert};

TextField kutugir1; TextField kutugir2; TextField kutugir3; TextField kutugir4; TextField kutugir5; Panel p,p1;

public void init(){

//not : init metodunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. setBackground(renk.beyaz);

p=new Panel();

p.setLayout(new GridLayout(3,4)); p.setBackground(renk.beyaz);

//3\*4 tablo p paneli haz• rland• p1=new Panel();

p1.setLayout(new GridLayout(1,4)); p1.setBackground(renk.beyaz); renkListesi=new List(3,false); arkaPlanRenkListesi=new List(3,false); kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

renk1=new Label("renk : "); renk2=new Label("arka plan rengi :"); xi1=300;

yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1); kutugir5.setText(""+aci1); renk=new renk(255,200,0); p1.add(renk1); p1.add(renkListesi); p1.add(renk2); p1.add(arkaPlanRenkListesi);

//elemanlar p ve p1 paneline p ve p1 paneli de aplete eklendi kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); renkListesi.addActionListener(this); arkaPlanRenkListesi.addActionListener(this); renkListesi.addItemListener(this); arkaPlanRenkListesi.addItemListener(this); addMouseListener(this);

addMouseMotionListener(this); for(int i=0;i<r.length;i++)

{

renkListesi.add(Renkler[i]); arkaPlanRenkListesi.add(Renkler[i]);

}

add(p,BorderLayout.NORTH); add(p1,BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex(); if(indeks!=-1)

{

setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk1.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

}

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void itemStateChanged(ItemEvent e)

{

showStatus(" X : "+ xi1+" Y : "+yi1);

}

public void paint(Graphics g)

{

int indeks=renkListesi.getSelectedIndex(); if(indeks!=-1)

g.setColor(r[indeks]); else

g.setColor(renk.mavi); yildiz.fillYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); yildiz.drawYildiz1(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

public void mouseClicked(MouseEvent e)

{

}

public void mousePressed(MouseEvent e)

{

xi1=e.getX();

yi1=e.getY(); kutugir1.setText(""+xi1); kutugir2.setText(""+yi1);

showStatus("merkez : X : "+ xi1+" Y : "+yi1);

}

public void mouseReleased(MouseEvent e)

{

int xi2=e.getX(); int yi2=e.getY();

yaricap1=(int)Math.sqrt((xi1-xi2)\*(xi1-xi2)+(yi1-yi2)\*(yi1-yi2)); aci1=-Math.atan2((yi2-yi1),(xi2-xi1))\*180.0/Math.PI-90; kutugir4.setText(""+yaricap1);

kutugir5.setText(""+aci1);

showStatus(" X : "+ xi1+" Y : "+yi1+" yaricap : "+yaricap1); repaint();

}

public void mouseEntered(MouseEvent e)

{

}

public void mouseExited(MouseEvent e)

{

}

public void mouseMoved(MouseEvent e)

{

showStatus(" X : "+ e.getX()+" Y : "+e.getY());

}

public void mouseDragged(MouseEvent e)

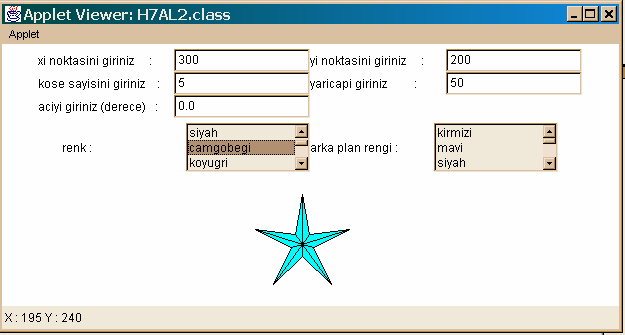
{

showStatus(" X : "+ e.getX()+" Y : "+e.getY());

}

}

### 07055.JPG



**Sekil 7.55 H7AL2.java çiktisi**

1. **H7AL3.**java programini incele bu program yildiz çizdirmektedir. Yildizin rengi scrollbar kullanilarak degistirilmektedir. Appleti çalistir.

### Program 7.56 H7AL3.java programi.

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import yildiz;

import renk;

public class H7AL3 extends Applet implements ActionListener,AdjustmentListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk

Label kutu1; Label kutu2; Label kutu3; Label kutu4; Label kutu5;

Label bos1;

Label bos2;

Label bos3;

Label renk1;

Label kirmizi;

Label mavi;

Label yesil;

TextField kutugir1; TextField kutugir2; TextField kutugir3; TextField kutugir4; TextField kutugir5;

Scrollbar kutukirmizi;

Scrollbar kutumavi;

Scrollbar kutuyesil;

public void init(){

//not : init metodunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. Panel p=new Panel();

p.setLayout(new GridLayout(5,4));

//5\*4 tablo p paneli haz• rland•

kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

bos1=new Label(" "); bos2=new Label(" "); bos3=new Label(" ");

renk1=new Label("renk"); kirmizi=new Label("kirmizi"); mavi=new Label("mavi"); yesil=new Label("yesil");

kutukirmizi=new Scrollbar(Scrollbar.HORIZONTAL,255,1,0,255); kutumavi=new Scrollbar(Scrollbar.HORIZONTAL,200,1,0,255); kutuyesil=new Scrollbar(Scrollbar.HORIZONTAL,0,1,0,255);

xi1=300; yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1); kutugir5.setText(""+aci1); renk=new renk(255,200,0);

p.add(bos1);

p.add(bos2);

p.add(bos3); p.add(kirmizi); p.add(yesil);

p.add(mavi);

p.add(renk1); p.add(kutukirmizi); p.add(kutumavi); p.add(kutuyesil);

//elemanlar p paneline p paneli de aplete eklendi add(p,BorderLayout.NORTH); kutukirmizi.addAdjustmentListener(this); kutumavi.addAdjustmentListener(this); kutuyesil.addAdjustmentListener(this); kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void adjustmentValueChanged(AdjustmentEvent e)

{

int kirmizi1=kutukirmizi.getValue(); int mavi1=kutumavi.getValue();

int yesil1=kutuyesil.getValue(); renk=new renk(kirmizi1,mavi1,yesil1); repaint();

}

public void paint(Graphics g)

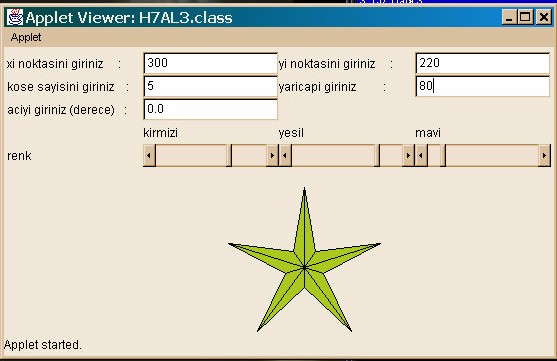
{

g.setColor(renk); yildiz.fillYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); yildiz.drawYildiz1(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

}

### 07056.JPG



**Sekil 7.56 Yildiz çizdiren ve scrollbar kullanarak renk kontrolu yapan H7AL3.html çiktisi**

1. **H7AL4.java** programi cardLayout layout düzenleyicisini kullanmaktadir. Bu düzenleyici appleti çesitli sayfalar seklinde düzenlemektedir. Programi ve cardLayout layout düzenleme sistemini inceleyiniz.

### Program 7.57 : H7AL4.java, cardLayout sistemini kullanan yildiz çizme appleti

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import yildiz;

import renk;

public class H7AL4 extends Applet implements ActionListener,AdjustmentListener

{

//card menager Layout kullan• m• private CardLayout c;

private Panel p,p1,p2,p3,p4; int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk Button B1,B2,B3,B4;

Label kutu1; Label kutu2; Label kutu3; Label kutu4; Label kutu5;

Label bos1;

Label bos2;

Label bos3;

Label renk1;

Label kirmizi;

Label mavi;

Label yesil;

TextField kutugir1; TextField kutugir2; TextField kutugir3; TextField kutugir4; TextField kutugir5;

Scrollbar kutukirmizi;

Scrollbar kutumavi;

Scrollbar kutuyesil;

public void init(){

//not : init metodunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. c=new CardLayout();

p=new Panel(); p.setLayout(c); p1=new Panel(); p2=new Panel(); p3=new Panel(); p4=new Panel();

p1.setLayout(new GridLayout(1,4)); p2.setLayout(new GridLayout(1,4)); p3.setLayout(new GridLayout(1,4)); p4.setLayout(new GridLayout(2,4));

//5\*4 tablo p paneli haz• rland•

kutu1=new Label("xi noktasini giriniz :"); p1.add(kutu1);

kutugir1=new TextField(3); p1.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p1.add(kutu2);

kutugir2=new TextField(3); p1.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p2.add(kutu3);

kutugir3=new TextField(3); p2.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p2.add(kutu4);

kutugir4=new TextField(3); p2.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p3.add(kutu5);

kutugir5=new TextField(3); p3.add(kutugir5);

bos1=new Label(" "); bos2=new Label(" "); bos3=new Label(" ");

renk1=new Label("renk"); kirmizi=new Label("kirmizi"); mavi=new Label("mavi"); yesil=new Label("yesil");

kutukirmizi=new Scrollbar(Scrollbar.HORIZONTAL,255,1,0,255); kutumavi=new Scrollbar(Scrollbar.HORIZONTAL,200,1,0,255); kutuyesil=new Scrollbar(Scrollbar.HORIZONTAL,0,1,0,255);

xi1=300; yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1); kutugir5.setText(""+aci1); renk=new renk(255,200,0); p3.add(bos1);

p3.add(bos2);

p4.add(bos3); p4.add(kirmizi); p4.add(yesil); p4.add(mavi); p4.add(renk1); p4.add(kutukirmizi); p4.add(kutumavi); p4.add(kutuyesil);

//elemanlar p paneline p paneli de aplete eklendi p.add(p1,BorderLayout.SOUTH); p.add(p2,BorderLayout.SOUTH); p.add(p3,BorderLayout.SOUTH); p.add(p4,BorderLayout.SOUTH);

B1=new Button("<<");

B2=new Button("<");

B3=new Button(">");

B4=new Button(">>");

add(B1);

add(B2);

add(B3);

add(B4);

add(p,BorderLayout.NORTH); kutukirmizi.addAdjustmentListener(this); kutumavi.addAdjustmentListener(this); kutuyesil.addAdjustmentListener(this); kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); B1.addActionListener(this); B2.addActionListener(this); B3.addActionListener(this); B4.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

//sayfa se‡imi

if(e.getSource()==B1)

{

c.first(p);

}

else if(e.getSource()==B2)

{

c.previous(p);

}

else if(e.getSource()==B3)

{

c.next(p);

}

else if(e.getSource()==B4)

{

c.last(p);

}

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void adjustmentValueChanged(AdjustmentEvent e)

{

int kirmizi1=kutukirmizi.getValue(); int mavi1=kutumavi.getValue();

int yesil1=kutuyesil.getValue(); renk=new renk(kirmizi1,mavi1,yesil1); repaint();

}

public void paint(Graphics g)

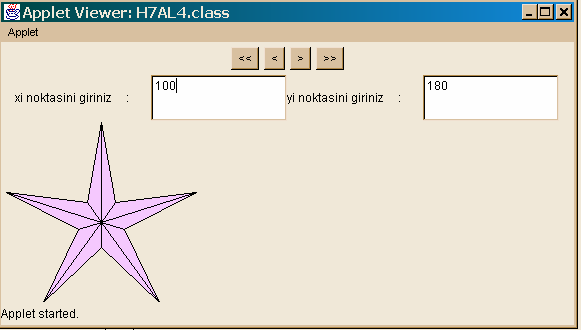
{

g.setColor(renk); yildiz.fillYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); yildiz.drawYildiz1(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

}

### 07057.JPG



**Sekil 7.57 H7AL4.html, cardLayout sistemini kullanan yildiz çizme appleti**

1. Appletde fare kullanarak çizgi çizdiren H7AL5.java programini inceleyiniz.

### Program 7.58 H7AL5.java

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import renk;

public class H7AL5 extends Applet

implements ActionListener,ItemListener,MouseListener,MouseMotionListener

{

renk renk; //sinifin adi renk, nesne adi renk Label renk1;

Label renk2;

private String Renkler[]={"kirmizi","mavi","siyah","camgobegi","koyugri","gri","yesil","acikgri", "mor","portakal","pembe","beyaz","sari","acikmavi","lacivert"};

private List renkListesi,arkaPlanRenkListesi; private renk

r[]={renk.kirmizi,renk.mavi,renk.siyah,renk.camgobegi,renk.koyugri,renk.gri,renk.yesil,renk.acikgri,renk.mor, renk.portakal,renk.pembe,renk.beyaz,renk.sari,renk.acikmavi,renk.lacivert};

Panel p,p1;

int xi1,yi1,xi2,yi2; int ii;

public void init(){

//not : init metodunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. setBackground(renk.beyaz);

p=new Panel();

p.setLayout(new GridLayout(3,4)); p.setBackground(renk.beyaz);

//3\*4 tablo p paneli haz• rland• p1=new Panel();

p1.setLayout(new GridLayout(1,4)); p1.setBackground(renk.beyaz); renkListesi=new List(3,false); arkaPlanRenkListesi=new List(3,false); xi1=0;

yi1=0; xi2=0; yi2=0;

renk1=new Label("renk : "); renk2=new Label("arka plan rengi :"); renk=new renk(255,200,0); p1.add(renk1);

p1.add(renkListesi); p1.add(renk2); p1.add(arkaPlanRenkListesi);

renkListesi.addActionListener(this); arkaPlanRenkListesi.addActionListener(this); renkListesi.addItemListener(this); arkaPlanRenkListesi.addItemListener(this); addMouseListener(this); addMouseMotionListener(this);

for(int i=0;i<r.length;i++)

{

renkListesi.add(Renkler[i]); arkaPlanRenkListesi.add(Renkler[i]);

}

add(p1,BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex();

if(indeks!=-1)

{

setBackground(r[indeks]); p1.setBackground(r[indeks]); renk1.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

}

repaint();

}

public void itemStateChanged(ItemEvent e)

{

showStatus(" X : "+ xi1+" Y : "+yi1);

}

public void paint(Graphics g)

{

int indeks=renkListesi.getSelectedIndex(); if(indeks!=-1)

g.setColor(r[indeks]); else

g.setColor(renk.mavi); g.drawLine(xi1,yi1,xi2,yi2);

}

public void update(Graphics g) { paint(g); }

public void mouseClicked(MouseEvent e)

{

xi1=e.getX();

yi1=e.getY();

showStatus("merkez : X : "+ xi1+" Y : "+yi1);

}

public void mousePressed(MouseEvent e)

{

xi1=e.getX();

yi1=e.getY();

showStatus("merkez : X : "+ xi1+" Y : "+yi1);

}

public void mouseReleased(MouseEvent e)

{

xi2=e.getX();

yi2=e.getY();

showStatus("merkez : X : "+ xi2+" Y : "+yi2); repaint();

}

public void mouseEntered(MouseEvent e)

{

}

public void mouseExited(MouseEvent e)

{

showStatus("mouse applet disina cikti");

}

public void mouseMoved(MouseEvent e)

{

}

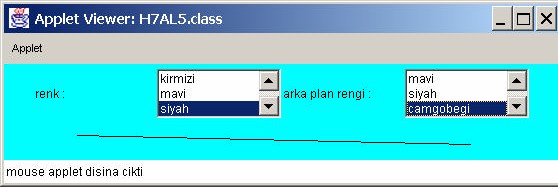
public void mouseDragged(MouseEvent e)

{

}

}

### 07058.JPG



**Sekil 7.58 Appletde fare kullanarak sekil çizdiren H7AL5.html applet görüntüsü**

1. **H7O1.java,** Alistirma H7AL2'de yildiz çizilmis ve mouse'u basip çekip birakarak yeni yildiz olusturulmustur. Ayni islemi yapan, fakat eskenar çokken çizen programi olusturunuz.

### Program 7.59 H7O1.java

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import eskenarcokgen; import renk;

public class H7O1 extends Applet

implements ActionListener,ItemListener,MouseListener,MouseMotionListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk Label kutu1;

Label kutu2; Label kutu3; Label kutu4; Label kutu5; Label renk1; Label renk2;

private String Renkler[]={"kirmizi","mavi","siyah","camgobegi","koyugri","gri","yesil","acikgri", "mor","portakal","pembe","beyaz","sari","acikmavi","lacivert"};

private List renkListesi,arkaPlanRenkListesi; private renk

r[]={renk.kirmizi,renk.mavi,renk.siyah,renk.camgobegi,renk.koyugri,renk.gri,renk.yesil,renk.acikgri,renk.mor, renk.portakal,renk.pembe,renk.beyaz,renk.sari,renk.acikmavi,renk.lacivert};

TextField kutugir1;

TextField kutugir2;

TextField kutugir3; TextField kutugir4; TextField kutugir5; Panel p,p1;

public void init(){

//not : init metodunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. setBackground(renk.beyaz);

p=new Panel();

p.setLayout(new GridLayout(3,4)); p.setBackground(renk.beyaz);

//3\*4 tablo p paneli haz• rland• p1=new Panel();

p1.setLayout(new GridLayout(1,4)); p1.setBackground(renk.beyaz); renkListesi=new List(3,false); arkaPlanRenkListesi=new List(3,false); kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

renk1=new Label("renk : "); renk2=new Label("arka plan rengi :"); xi1=300;

yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1); kutugir5.setText(""+aci1); renk=new renk(255,200,0); p1.add(renk1); p1.add(renkListesi); p1.add(renk2); p1.add(arkaPlanRenkListesi);

//elemanlar p ve p1 paneline p ve p1 paneli de aplete eklendi kutugir1.addActionListener(this); kutugir2.addActionListener(this);

kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); renkListesi.addActionListener(this); arkaPlanRenkListesi.addActionListener(this); renkListesi.addItemListener(this); arkaPlanRenkListesi.addItemListener(this); addMouseListener(this); addMouseMotionListener(this);

for(int i=0;i<r.length;i++)

{

renkListesi.add(Renkler[i]); arkaPlanRenkListesi.add(Renkler[i]);

}

add(p,BorderLayout.NORTH); add(p1,BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex(); if(indeks!=-1)

{

setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk1.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

}

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void itemStateChanged(ItemEvent e)

{

showStatus(" X : "+ xi1+" Y : "+yi1);

}

public void paint(Graphics g)

{

int indeks=renkListesi.getSelectedIndex(); if(indeks!=-1)

g.setColor(r[indeks]); else

g.setColor(renk.mavi);

eskenarcokgen.fillEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); eskenarcokgen.drawEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

public void mouseClicked(MouseEvent e)

{

}

public void mousePressed(MouseEvent e)

{

xi1=e.getX();

yi1=e.getY(); kutugir1.setText(""+xi1); kutugir2.setText(""+yi1);

showStatus("merkez : X : "+ xi1+" Y : "+yi1);

}

public void mouseReleased(MouseEvent e)

{

int xi2=e.getX(); int yi2=e.getY();

yaricap1=(int)Math.sqrt((xi1-xi2)\*(xi1-xi2)+(yi1-yi2)\*(yi1-yi2)); aci1=-Math.atan2((yi2-yi1),(xi2-xi1))\*180.0/Math.PI-90; kutugir4.setText(""+yaricap1);

kutugir5.setText(""+aci1);

showStatus(" X : "+ xi1+" Y : "+yi1+" yaricap : "+yaricap1); repaint();

}

public void mouseEntered(MouseEvent e)

{

}

public void mouseExited(MouseEvent e)

{

}

public void mouseMoved(MouseEvent e)

{

showStatus(" X : "+ e.getX()+" Y : "+e.getY());

}

public void mouseDragged(MouseEvent e)

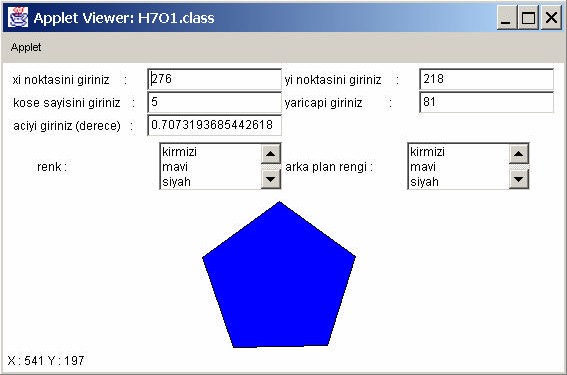
{

showStatus(" X : "+ e.getX()+" Y : "+e.getY());

}

}

### 07059.JPG



**Sekil 7.59 Appletde sekil çokgen çizdiren H7O1.html applet çiktisi**

1. **H7O2.java,** Choice kullanarak eskenarcokken veya yildiz çizebilen applet programini olusturunuz.

### Program 7.60 H7O2.java, choice kullanimi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import yildiz;

import renk;

public class H7O3 extends Applet

implements ActionListener,AdjustmentListener,ItemListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk Label kutu1;

Label kutu2; Label kutu3; Label kutu4; Label kutu5;

//Label bos1;

//Label bos2;

Label bos3;

Label renk1;

Label kirmizi;

Label mavi;

Label yesil;

TextField kutugir1;

TextField kutugir2;

TextField kutugir3;

TextField kutugir4;

TextField kutugir5;

Scrollbar kutukirmizi;

Scrollbar kutumavi;

Scrollbar kutuyesil;

Checkbox c1,c2;

public void init(){

//not : init metodunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. c1=new Checkbox("YILDIZ CIZ");

c2=new Checkbox("ESKENAR COKGEN CIZ"); Panel p=new Panel();

p.setLayout(new GridLayout(5,4));

//5\*4 tablo p paneli haz• rland•

kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

//bos1=new Label(" ");

//bos2=new Label(" "); bos3=new Label(" ");

renk1=new Label("renk"); kirmizi=new Label("kirmizi"); mavi=new Label("mavi"); yesil=new Label("yesil");

kutukirmizi=new Scrollbar(Scrollbar.HORIZONTAL,255,1,0,255); kutumavi=new Scrollbar(Scrollbar.HORIZONTAL,200,1,0,255); kutuyesil=new Scrollbar(Scrollbar.HORIZONTAL,0,1,0,255);

xi1=300; yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1); kutugir5.setText(""+aci1); renk=new renk(255,200,0); p.add(c1);

p.add(c2);

p.add(bos3); p.add(kirmizi); p.add(yesil);

p.add(mavi);

p.add(renk1); p.add(kutukirmizi); p.add(kutumavi); p.add(kutuyesil);

//elemanlar p paneline p paneli de aplete eklendi add(p,BorderLayout.NORTH); kutukirmizi.addAdjustmentListener(this); kutumavi.addAdjustmentListener(this); kutuyesil.addAdjustmentListener(this); kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); c1.addItemListener(this); c2.addItemListener(this);

}

public void actionPerformed(ActionEvent e)

{

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void adjustmentValueChanged(AdjustmentEvent e)

{

//renk de§iŸtirildiginde

int kirmizi1=kutukirmizi.getValue(); int mavi1=kutumavi.getValue();

int yesil1=kutuyesil.getValue(); renk=new renk(kirmizi1,mavi1,yesil1); repaint();

}

public void itemStateChanged(ItemEvent e)

{

//Choice komutu de§iŸtirildi§inde repaint();

}

public void paint(Graphics g)

{

if (F.getSelectedIndex()==0)

{

g.setColor(renk); yildiz.fillYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

g.setColor(renk.siyah); yildiz.drawYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

else if (F.getSelectedIndex()==1)

{

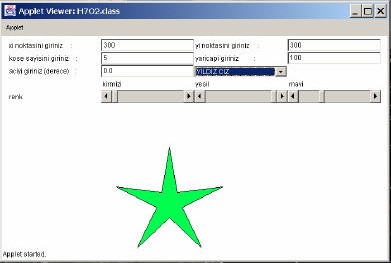
g.setColor(renk); eskenarcokgen.fillEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); eskenarcokgen.drawEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

}

}

### 07060.JPG



**Sekil 7.60 Appletde choice kullanarak çokgen veya yildiz çizdiren H7O2.html applet çiktisi**

1. **H7O3.java,** Checkbox kullanarak eskenarcokken veya yildiz veya herikisinide bir arada çizebilen (yildizin köselerinden geçen bir eskenar cokken) applet programini olusturunuz.

**Program 7.61 H7O3.java** import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import yildiz;

import eskenarcokgen; import renk;

public class H7O3 extends Applet implements ActionListener,ItemListener

{

int n1,xi1,yi1,yaricap1; double aci1;

renk renk; //sinifin adi renk, nesne adi renk Label kutu1;

Label kutu2; Label kutu3; Label kutu4; Label kutu5; Label renk1; Label renk2;

private String Renkler[]={"kirmizi","mavi","siyah","camgobegi","koyugri","gri","yesil","acikgri", "mor","portakal","pembe","beyaz","sari","acikmavi","lacivert"};

private List renkListesi,arkaPlanRenkListesi;

private renk r[]={renk.kirmizi,renk.mavi,renk.siyah,renk.camgobegi,renk.koyugri,renk.gri,renk.yesil,renk.acikgri,renk.mor,

renk.portakal,renk.pembe,renk.beyaz,renk.sari,renk.acikmavi,renk.lacivert}; TextField kutugir1;

TextField kutugir2;

TextField kutugir3;

TextField kutugir4;

TextField kutugir5;

Panel p,p1;

Checkbox c1,c2;

public void init(){

//not : init metotunda t• m de§iŸkenlerin ilk de§erleri verilmelidir. setBackground(renk.beyaz);

c1=new Checkbox("YILDIZ CIZ");

c2=new Checkbox("ESKENAR COKGEN CIZ"); c1.setState(true);

c2.setState(false); p=new Panel();

p.setLayout(new GridLayout(3,4)); p.setBackground(renk.beyaz);

//3\*4 tablo p paneli haz• rland• p1=new Panel();

p1.setLayout(new GridLayout(1,4)); p1.setBackground(renk.beyaz); renkListesi=new List(3,true); arkaPlanRenkListesi=new List(3,true); kutu1=new Label("xi noktasini giriniz :"); p.add(kutu1);

kutugir1=new TextField(3); p.add(kutugir1);

kutu2=new Label("yi noktasini giriniz :"); p.add(kutu2);

kutugir2=new TextField(3); p.add(kutugir2);

kutu3=new Label("kose sayisini giriniz :"); p.add(kutu3);

kutugir3=new TextField(3); p.add(kutugir3);

kutu4=new Label("yaricapi giriniz :"); p.add(kutu4);

kutugir4=new TextField(3); p.add(kutugir4);

kutu5=new Label("aciyi giriniz (derece) :"); p.add(kutu5);

kutugir5=new TextField(3); p.add(kutugir5);

p.add(c1);

p.add(c2);

renk1=new Label("renk : "); renk2=new Label("arka plan rengi :"); xi1=300;

yi1=300; n1=5;

yaricap1=100; aci1=0;

kutugir1.setText(""+xi1); kutugir2.setText(""+yi1); kutugir3.setText(""+n1); kutugir4.setText(""+yaricap1);

kutugir5.setText(""+aci1); renk=new renk(255,200,0); p1.add(renk1); p1.add(renkListesi); p1.add(renk2); p1.add(arkaPlanRenkListesi);

//elemanlar p ve p1 paneline p ve p1 paneli de aplete eklendi kutugir1.addActionListener(this); kutugir2.addActionListener(this); kutugir3.addActionListener(this); kutugir4.addActionListener(this); kutugir5.addActionListener(this); renkListesi.addActionListener(this);

arkaPlanRenkListesi.addActionListener(this); renkListesi.addItemListener(this); arkaPlanRenkListesi.addItemListener(this);

c1.addItemListener(this); c2.addItemListener(this); for(int i=0;i<r.length;i++)

{

renkListesi.add(Renkler[i]); arkaPlanRenkListesi.add(Renkler[i]);

}

renkListesi.select(0); arkaPlanRenkListesi.select(3);

int indeks=arkaPlanRenkListesi.getSelectedIndex(); setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk1.setBackground(r[indeks]); renk2.setBackground(r[indeks]); add(p,BorderLayout.NORTH); add(p1,BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent e)

{

int indeks=arkaPlanRenkListesi.getSelectedIndex(); if(indeks!=-1)

{

setBackground(r[indeks]); p.setBackground(r[indeks]); p1.setBackground(r[indeks]); kutu1.setBackground(r[indeks]); kutu2.setBackground(r[indeks]); kutu3.setBackground(r[indeks]); kutu4.setBackground(r[indeks]); kutu5.setBackground(r[indeks]); renk1.setBackground(r[indeks]); renk2.setBackground(r[indeks]);

}

Integer xi1i=new Integer(kutugir1.getText()); xi1=xi1i.intValue();

Integer yi1i=new Integer(kutugir2.getText()); yi1=yi1i.intValue();

Integer n1i=new Integer(kutugir3.getText()); n1=n1i.intValue();

Integer yaricapi=new Integer(kutugir4.getText()); yaricap1=yaricapi.intValue();

Double aci1i=new Double(kutugir5.getText()); aci1=aci1i.doubleValue();

repaint();

}

public void itemStateChanged(ItemEvent e)

{

showStatus(" renk: "+ renkListesi.getSelectedItem()+" arka plan: "+ arkaPlanRenkListesi.getSelectedItem());

repaint();

}

public void paint(Graphics g)

{

int indeks=renkListesi.getSelectedIndex(); if(indeks!=-1)

g.setColor(r[indeks]); else

g.setColor(renk.mavi);

if ( c1.getState() && c2.getState() )

{

g.setColor(r[indeks]); eskenarcokgen.fillEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); eskenarcokgen.drawEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); yildiz.drawYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

else if ( c1.getState())

{

g.setColor(r[indeks]); yildiz.fillYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); yildiz.drawYildiz(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

else if (c2.getState())

{

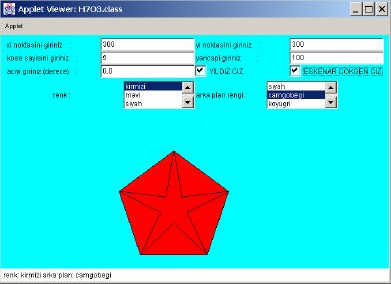
g.setColor(r[indeks]); eskenarcokgen.fillEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI); g.setColor(renk.siyah); eskenarcokgen.drawEskenarcokgen(g,xi1,yi1,n1,yaricap1,aci1/180.0\*Math.PI);

}

}

}

### 07061.JPG



**Sekil 7.61 Appletde checkBox kullanarak çokgen ve/veya yildiz çizdiren H7O3.html applet çiktisi**

1. **H7O4.java,** H7O3.java(alistirma 12) programinin aynisini olusturunuz, ancak renk seçimini Scrollbar ile yapiniz.

### 14. H7OD3\_2000

JSlider kullanarak çizdiginiz bir yildizin çapini büyütüp küçülten **H7OD3\_2000** programini yaziniz. ovalPanelSWP.java, ovalTestSWF\_2000.java, yildizSW.java,yildizPanelSWP.java

**Program 7.62** yildizPanelSWP.hava import javax.swing.\*;

import java.awt.\*; import yildizSW; import renk;

public class yildizPanelSWP extends JPanel

{

int xi,yi,yaricap; int n;

double aci;

public yildizPanelSWP(int xi1, int yi1,int yaricap1,int n1,double aci1)

{

xi=xi1; yi=yi1;

yaricap=yaricap1; aci=aci1;

n=n1;

}

public void setYildizPanelSWP(int xi1, int yi1, int yaricap1,int n1,double aci1)

{

xi=xi1; yi=yi1;

yaricap=yaricap1; aci=aci1;

n=n1; repaint();

}

public void paintComponent(Graphics g)

{

super.paintComponent(g);

g.setColor(renk.mavi); yildizSW.fillYildiz(g,xi,yi,n,yaricap,aci/180.0\*Math.PI); g.setColor(renk.siyah); yildizSW.drawYildiz1(g,xi,yi,n,yaricap,aci/180.0\*Math.PI);

}

}

### Program 7.63 H7OD3\_2000.java

import javax.swing.\*; // java swing sinifini cagir

import java.awt.\*; // java pencere kullanma sinifini cagir import java.awt.event.\*; // java pencereyi dinleme sinifini cagir import BasicWindowMonitor;

import ovalPanelSWP; import javax.swing.event.\*;

public class H7OD3\_2000 extends JFrame implements ChangeListener

{

// Renk secme ornegi private yildizpanelSWP p; private JSlider cap;

Color r=Color.lightGray;

Container c;

// pencereyi baslatma metodu public H7OD3\_2000()

{

super("JSlider Testi"); c=getContentPane(); p=new yildizpanelSWP(0);

cap=new JSlider(SwingConstants.HORIZONTAL,0,200,10); cap.setMajorTickSpacing(10);

cap.setPaintTicks(true); cap.addChangeListener(this); c.add(p,BorderLayout.CENTER); c.add(cap,BorderLayout.SOUTH);

}

// girdi alanindaki olan olaylari dinleme metodu public void stateChanged(ChangeEvent e)

{

p.setYildizPanelSWP(cap.getValue()); repaint();

}

//=====================================================

public static void main(String[] args)

{

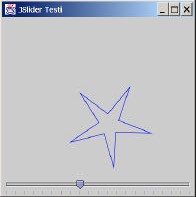
H7OD3\_2000 pencere= new H7OD3\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,400);

pencere.setVisible(true);

}

}

### 07062.JPG



**Sekil 7.62 H7OD3\_2000.java program çiktisinin JFrame penceresinde görünümü**

**15. H7OD4\_2000**

Dügme (Button) kontrolü ile yildiz vetya oval çizdirecek bir program yaziniz. Yildiz dügmesine basilinca ekrana yildiz gelsin, oval dügmesine basilinca ekrana oval gelsin

### Program 7.63 yildizovalPanelSWP.java

import javax.swing.\*; import java.awt.\*; import java.awt.geom.\*;

public class yildizovalPanelSWP extends JPanel

{

int anahtar;

int xi,yi,yaricap; int n;

double aci;

public yildizovalPanelSWP(int xi1, int yi1,int yaricap1,int n1,double aci1)

{

anahtar=1; xi=xi1; yi=yi1;

yaricap=yaricap1; aci=aci1;

n=n1;

}

public void setYildizOvalPanelSWP(int xi1, int yi1, int yaricap1,int n1,double aci1,int an)

{

anahtar=an; xi=xi1; yi=yi1;

yaricap=yaricap1; aci=aci1;

n=n1; repaint();

}

public void setAnahtar(int an)

{

anahtar=an;

}

public void paintComponent(Graphics g)

{

super.paintComponent(g); g.setColor(renk.mavi);

if(anahtar==1)

{

yildizSW.fillYildiz(g,xi,yi,n,yaricap,aci/180.0\*Math.PI); g.setColor(renk.siyah); yildizSW.drawYildiz1(g,xi,yi,n,yaricap,aci/180.0\*Math.PI);

}

else if(anahtar==2)

{

Graphics2D g2=(Graphics2D)g; g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

Ellipse2D elips1=new Ellipse2D.Double(xi,yi,yaricap,yaricap); g2.draw(elips1);

}

}

}

### Program 7.64 H7OD4\_2000.java

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*; import java.net.URL;

import yildizovalPanelSWP;

public class H7OD4\_2000 extends JFrame implements ActionListener

{

private JButton B1,B2;

private yildizovalPanelSWP p; private Container c;

public H7OD4\_2000()

{

super("dügme - yildiz - oval seçim testi"); c=getContentPane();

JPanel p1=new JPanel(); p1.setLayout(new GridLayout(1,2)); B1=new JButton("yildiz"); B1.addActionListener(this); p1.add(B1);

B2=new JButton("Oval"); B2.addActionListener(this); p1.add(B2); c.add(p1,BorderLayout.SOUTH); int x=(int)(c.getWidth()/2.0);

int y=(int)(c.getHeight()/2.0);

int r=(int)(Math.sqrt(x\*x+y\*y)/3.0); p=new yildizovalPanelSWP(x,y,r,5,0); c.add(p,BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

int x=(int)(c.getWidth()/2.0); int y=(int)(c.getHeight()/2.0);

int r=(int)(Math.sqrt(x\*x+y\*y)/3.0); if(e.getSource()==B1)

{p.setYildizOvalPanelSWP(x,y,r,5,0,1);} else if(e.getSource()==B2)

{p.setYildizOvalPanelSWP(x-r,y-r,2\*r,5,0,2);}

}

public static void main(String[] args)

{

H7OD4\_2000 pencere= new H7OD4\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(250,200);

pencere.setVisible(true);

}

}

### 07063.JPG - 07064.JPG



**Sekil 7.63 Sekil -7.64 H7OD4\_2000.java program çiktisinin JFrame penceresinde görünümü**

1. **swing ve awt karsilastirmali örnekler** 1

Su ana kadar çesitli örneklerde hem swing hem de awt olarak verdigimiz örneklerle iki grafik sistemindeli farklilik ve benzerlikleri belirtmeye çalistik. Simdi özellikle bu karsilastirmayi çesitli uygulama problemleri olarak görecegiz. Bu örnekteki problemler temel olarak java döküman kütüphanesinden alinmistir. Ilk örnegimiz TextEventDemo awt de yazilmis bir applet programi. TextArea penceresine giren her harfte TextEvent yardimiyla kontrollar yapiyoruz.

### Program 7.65 : TextEventDemo (awt)

/\*

* awt applet programi

\*/

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

public class TextEventDemo extends Applet

implements ActionListener { TextField textField;

TextArea textArea;

TextArea displayArea;

public void init() {

Button button = new Button("Sil"); button.addActionListener(this);

textField = new TextField(20); textField.addActionListener(new MyTextActionListener()); textField.addTextListener(new MyTextListener("Text Field"));

textArea = new TextArea(5, 20); textArea.addTextListener(new MyTextListener("Text Area"));

displayArea = new TextArea(5, 20); displayArea.setEditable(false);

//GridbagLayout oldukça kompleks formatlamaya izin verir GridBagLayout gridbag = new GridBagLayout(); GridBagConstraints c = new GridBagConstraints(); setLayout(gridbag);

c.fill = GridBagConstraints.BOTH; c.weightx = 1.0;

Panel leftPanel = new Panel(); leftPanel.setLayout(new BorderLayout()); leftPanel.add("North", textField); leftPanel.add("Center", textArea);

c.gridheight = 2; gridbag.setConstraints(leftPanel, c); add(leftPanel);

c.weighty = 1.0;

c.gridwidth = GridBagConstraints.REMAINDER; c.gridheight = 1; gridbag.setConstraints(displayArea, c); add(displayArea);

c.weighty = 0.0; gridbag.setConstraints(button, c); add(button);

textField.requestFocus();

}

class MyTextListener implements TextListener { String preface;

String newline;

public MyTextListener(String source) {

newline = System.getProperty("line.separator"); preface = source

+ " yazi degisti."

+ newline

+ " ilk on harf: \"";

}

public void textValueChanged(TextEvent e) { TextComponent tc = (TextComponent)e.getSource(); String s = tc.getText();

try {

s = s.substring(0, 10);

} catch (StringIndexOutOfBoundsException ex) {

}

displayArea.append(preface + s + "\"" + newline); if (displayArea.isValid()) {

displayArea.setCaretPosition(java.lang.Integer.MAX\_VALUE);

}

}

}

class MyTextActionListener implements ActionListener {

/\*\* Textfielddeki yaziyi döndürür \*/

public void actionPerformed(ActionEvent e) { int selStart = textArea.getSelectionStart(); int selEnd = textArea.getSelectionEnd();

textArea.replaceRange(textField.getText(), selStart, selEnd);

textField.selectAll();

}

}

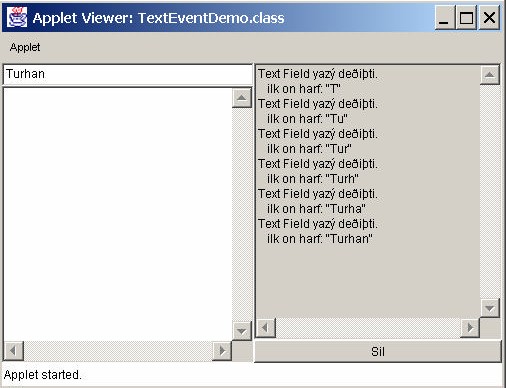
/\*\* dügmeye basilinca... \*/

public void actionPerformed(ActionEvent e) { displayArea.setText(""); textField.requestFocus();

}

}

### 07065.JPG



**Sekil 7.65 TextEventDemo.java sinifinin çikti programinin awt applet’te görülmesi**

1. **swing ve awt karsilastirmali örnekler** 2 : **TextEventDemoSW.java Program 7.66 TextEventDemo.java (swing) test programi**

/\*

* Swing versiyonu

\*/

import javax.swing.\*; import javax.swing.text.\*; import javax.swing.event.\*;

import java.awt.Dimension; import java.awt.BorderLayout; import java.awt.GridBagLayout;

import java.awt.GridBagConstraints; import java.awt.event.\*;

public class TextEventDemoSW extends JApplet

implements ActionListener {

JTextField textField;

JTextArea textArea;

JTextArea displayArea;

public void init() {

JButton button = new JButton("Sil"); button.addActionListener(this);

textField = new JTextField(20); textField.addActionListener(new MyTextActionListener());

textField.getDocument().addDocumentListener(new MyDocumentListener("Text Field"));

textArea = new JTextArea();

textArea.getDocument().addDocumentListener(new MyDocumentListener("Text Area")); JScrollPane scrollPane = new JScrollPane(textArea);

scrollPane.setPreferredSize(new Dimension(200, 75));

displayArea = new JTextArea(); displayArea.setEditable(false);

JScrollPane displayScrollPane = new JScrollPane(displayArea); displayScrollPane.setPreferredSize(new Dimension(200, 75));

JPanel contentPane = new JPanel(); GridBagLayout gridbag = new GridBagLayout(); GridBagConstraints c = new GridBagConstraints(); contentPane.setLayout(gridbag);

c.fill = GridBagConstraints.BOTH; c.weightx = 1.0;

JPanel leftPanel = new JPanel(); leftPanel.setLayout(new BorderLayout()); leftPanel.add(textField, BorderLayout.NORTH); leftPanel.add(scrollPane, BorderLayout.CENTER);

c.gridheight = 2; gridbag.setConstraints(leftPanel, c); contentPane.add(leftPanel);

c.weighty = 1.0;

c.gridwidth = GridBagConstraints.REMAINDER; c.gridheight = 1; gridbag.setConstraints(displayScrollPane, c); contentPane.add(displayScrollPane);

c.weighty = 0.0; gridbag.setConstraints(button, c); contentPane.add(button);

textField.requestFocus();

setContentPane(contentPane);

}

class MyDocumentListener implements DocumentListener { String preface;

String newline;

public MyDocumentListener(String source) { newline = System.getProperty("line.separator"); preface = source

+ " yazi degistirildi."

+ newline

+ " Ilk on karekter: \"";

}

public void insertUpdate(DocumentEvent e) { update(e);

}

public void removeUpdate(DocumentEvent e) { update(e);

}

public void changedUpdate(DocumentEvent e) {

}

public void update(DocumentEvent e) { Document doc = (Document)e.getDocument(); int length = doc.getLength();

String s = null; try {

s = doc.getText(0, (length > 10) ? 10 : length);

} catch (BadLocationException ex) {

}

displayArea.append(preface + s + "\"" + newline);

}

}

class MyTextActionListener implements ActionListener { public void actionPerformed(ActionEvent e) {

int selStart = textArea.getSelectionStart(); int selEnd = textArea.getSelectionEnd();

textArea.replaceRange(textField.getText(), selStart, selEnd);

textField.selectAll();

}

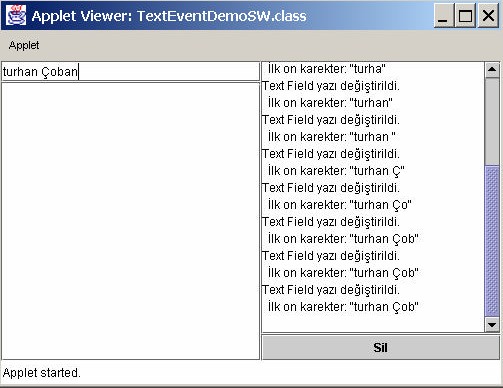
}

public void actionPerformed(ActionEvent e) { displayArea.setText(""); textField.requestFocus();

}

}

### 07066.JPG



**Sekil 7.66 TextEventDemo.java sinifinin çikti programinin swing JApplet’te görülmesi**

1. **swing ve awt karsilastirmali örnekler 3** : Birim dönüstürme örnegi : **TextEventDemoSW.java**

Birim dönüstürme örnegi :

### Tablo AWT ve Swing birim dönüstürme programlari

|  |  |
| --- | --- |
| **AWT Converter (birim dönüstürücü)** | **Swing Converter (birim dönüstürücü)** |
| **Converter.java** | **Converter.java** |
| **ConversionPanel.java** | **ConversionPanel.java** |
| **Unit.java** | **Unit.java** |
|  | **ConverterRangeModel.java** |
|  | **FollowerRangeModel.java** |
|  | **DecimalField.java** |
|  | **FormattedDocument.java** |

**Program 7.67 Converter.java AWT programi**

/\*

\* 1.1 version.

\*/

import java.awt.\*; import java.awt.event.\*; import java.util.\*;

import java.applet.Applet;

public class Converter extends Applet { ConversionPanel metricPanel, usaPanel; Unit[] metricDistances = new Unit[3]; Unit[] usaDistances = new Unit[4];

/\*\*

\* Create the ConversionPanels (one for metric, another for U.S.).

\* I used "U.S." because although Imperial and U.S. distance

\* measurements are the same, this program could be extended to

\* include volume measurements, which aren't the same.

\*/

public void init() {

//Use a GridLayout with 2 rows, as many columns as necessary,

//and 5 pixels of padding around all edges of each cell. setLayout(new GridLayout(2,0,5,5));

//Create Unit objects for metric distances, and then

//instantiate a ConversionPanel with these Units. metricDistances[0] = new Unit("Centimeters", 0.01); metricDistances[1] = new Unit("Meters", 1.0); metricDistances[2] = new Unit("Kilometers", 1000.0); metricPanel = new ConversionPanel(this, "Metric System",

metricDistances);

//Create Unit objects for U.S. distances, and then

//instantiate a ConversionPanel with these Units. usaDistances[0] = new Unit("Inches", 0.0254); usaDistances[1] = new Unit("Feet", 0.305); usaDistances[2] = new Unit("Yards", 0.914); usaDistances[3] = new Unit("Miles", 1613.0);

usaPanel = new ConversionPanel(this, "U.S. System", usaDistances);

//Add both ConversionPanels to the Converter. add(metricPanel);

add(usaPanel);

}

/\*\*

* Does the conversion from metric to U.S., or vice versa, and
* updates the appropriate ConversionPanel.

\*/

void convert(ConversionPanel from) { ConversionPanel to;

if (from == metricPanel) to = usaPanel;

else

to = metricPanel;

double multiplier = from.getMultiplier() / to.getMultiplier(); to.setValue(multiplier \* from.getValue());

}

/\*\* Draws a box around this panel. \*/ public void paint(Graphics g) {

Dimension d = getSize(); g.drawRect(0,0, d.width - 1, d.height - 1);

}

/\*\*

* Puts a little breathing space between
* the panel and its contents, which lets us draw a box
* in the paint() method.

\*/

public Insets getInsets() { return new Insets(5,5,5,5);

}

/\*\* Executed only when this program runs as an application. \*/ public static void main(String[] args) {

//Create a new window.

Frame f = new Frame("Converter Applet/Application");

f.addWindowListener(new WindowAdapter() { public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

//Create a Converter instance. Converter converter = new Converter();

//Initialize the Converter instance. converter.init();

//Add the Converter to the window and display the window. f.add("Center", converter);

f.pack(); //Resizes the window to its natural size. f.setVisible(true);

}

}

### Program 7.68 ConversionPanel.java AWT programi

/\*

* 1.1 version.

\*/

import java.awt.\*; import java.awt.event.\*; import java.util.\*;

import java.applet.Applet;

public class ConversionPanel extends Panel

implements ActionListener, AdjustmentListener, ItemListener {

TextField textField; Choice unitChooser; Scrollbar slider;

int max = 10000; int block = 100;

Converter controller; Unit[] units;

ConversionPanel(Converter myController, String myTitle, Unit[] myUnits) {

//Initialize this ConversionPanel to use a GridBagLayout. GridBagConstraints c = new GridBagConstraints(); GridBagLayout gridbag = new GridBagLayout(); setLayout(gridbag);

//Save arguments in instance variables. controller = myController;

units = myUnits;

//Set up default layout constraints.

c.fill = GridBagConstraints.HORIZONTAL;

//Add the label. It displays this panel's title, centered. Label label = new Label(myTitle, Label.CENTER);

c.gridwidth = GridBagConstraints.REMAINDER; //It ends a row. gridbag.setConstraints(label, c);

add(label);

//Add the text field. It initially displays "0" and needs

//to be at least 10 columns wide. textField = new TextField("0", 10);

c.weightx = 1.0; //Use maximum horizontal space... c.gridwidth = 1; //The default value. gridbag.setConstraints(textField, c);

add(textField); textField.addActionListener(this);

//Add the pop-up list (Choice). unitChooser = new Choice();

for (int i = 0; i < units.length; i++) { //Populate it. unitChooser.add(units[i].description);

}

c.weightx = 0.0; //The default value.

c.gridwidth = GridBagConstraints.REMAINDER; //End a row. gridbag.setConstraints(unitChooser, c);

add(unitChooser); unitChooser.addItemListener(this);

//Add the slider. It's horizontal, and it has the maximum

//value specified by the instance variable max. Its initial

//and minimum values are the default (0). A click increments

//the value by block units.

slider = new Scrollbar(Scrollbar.HORIZONTAL); slider.setMaximum(max + 10); slider.setBlockIncrement(block);

c.gridwidth = 1; //The default value. gridbag.setConstraints(slider, c); add(slider); slider.addAdjustmentListener(this);

}

/\*\*

* Returns the multiplier (units/meter) for the currently
* selected unit of measurement.

\*/

double getMultiplier() {

int i = unitChooser.getSelectedIndex(); return units[i].multiplier;

}

/\*\* Draws a box around this panel. \*/ public void paint(Graphics g) {

Dimension d = getSize(); g.drawRect(0,0, d.width - 1, d.height - 1);

}

/\*\*

* Puts a little breathing space between
* the panel and its contents, which lets us draw a box
* in the paint() method.
* We add more pixels to the right, to work around a
* Choice bug.

\*/

public Insets getInsets() { return new Insets(5,5,5,8);

}

/\*\*

* + Gets the current value in the text field.
  + It's guaranteed to be the same as the value
  + in the scroller (subject to rounding, of course).

\*/

double getValue() { double f;

try {

f = (double)Double.valueOf(textField.getText()).doubleValue();

} catch (java.lang.NumberFormatException e) { f = 0.0;

}

return f;

}

public void actionPerformed(ActionEvent e) { setSliderValue(getValue()); controller.convert(this);

}

public void itemStateChanged(ItemEvent e) { controller.convert(this);

}

/\*\* Respond to the slider. \*/

public void adjustmentValueChanged(AdjustmentEvent e) { textField.setText(String.valueOf(e.getValue())); controller.convert(this);

}

/\*\* Set the values in the slider and text field. \*/ void setValue(double f) {

setSliderValue(f); textField.setText(String.valueOf((float)f));

}

/\*\* Set the slider value. \*/

void setSliderValue(double f) { int sliderValue = (int)f;

if (sliderValue > max) sliderValue = max;

if (sliderValue < 0) sliderValue = 0;

slider.setValue(sliderValue);

}

}

### Program 7.69 Unit.java AWT programi

/\*

* 1.1 version.

\*/

public class Unit { String description; double multiplier;

Unit(String description, double multiplier) {

super();

this.description = description; this.multiplier = multiplier;

}

public String toString() {

String s = "Meters/" + description + " = " + multiplier; return s;

}

}

### Program 7.70 Conversion.java SWING programi

/\*

* 1.1+Swing version.

\*/

import javax.swing.\*; import javax.swing.event.\*; import java.awt.\*;

import java.awt.event.\*; import java.util.\*;

public class Converter {

ConversionPanel metricPanel, usaPanel; Unit[] metricDistances = new Unit[3]; Unit[] usaDistances = new Unit[4]; final static boolean COLORS = false; final static boolean DEBUG = false;

final static String LOOKANDFEEL = null;

ConverterRangeModel dataModel = new ConverterRangeModel(); JPanel mainPane;

/\*\*

* + Create the ConversionPanels (one for metric, another for U.S.).
  + I used "U.S." because although Imperial and U.S. distance
  + measurements are the same, this program could be extended to
  + include volume measurements, which aren't the same.

\*

* + Put the ConversionPanels into a frame, and bring up the frame.

\*/

public Converter() {

//Create Unit objects for metric distances, and then

//instantiate a ConversionPanel with these Units. metricDistances[0] = new Unit("Centimeters", 0.01); metricDistances[1] = new Unit("Meters", 1.0); metricDistances[2] = new Unit("Kilometers", 1000.0); metricPanel = new ConversionPanel(this, "Metric System",

metricDistances, dataModel);

//Create Unit objects for U.S. distances, and then

//instantiate a ConversionPanel with these Units. usaDistances[0] = new Unit("Inches", 0.0254); usaDistances[1] = new Unit("Feet", 0.305); usaDistances[2] = new Unit("Yards", 0.914); usaDistances[3] = new Unit("Miles", 1613.0); usaPanel = new ConversionPanel(this, "U.S. System",

usaDistances,

new FollowerRangeModel(dataModel));

//Create a JPanel, and add the ConversionPanels to it. mainPane = new JPanel();

if (COLORS) {

mainPane.setBackground(Color.red);

}

mainPane.setLayout(new GridLayout(2,1,5,5)); mainPane.setBorder(BorderFactory.createEmptyBorder(5,5,5,5)); mainPane.add(metricPanel);

mainPane.add(usaPanel); resetMaxValues(true);

}

public void resetMaxValues(boolean resetCurrentValues) { double metricMultiplier = metricPanel.getMultiplier(); double usaMultiplier = usaPanel.getMultiplier();

int maximum = ConversionPanel.MAX;

if (metricMultiplier > usaMultiplier) { maximum = (int)(ConversionPanel.MAX \*

(usaMultiplier/metricMultiplier));

}

if (DEBUG) {

System.out.println("in Converter resetMaxValues"); System.out.println(" metricMultiplier = "

+ metricMultiplier

+ "; usaMultiplier = "

+ usaMultiplier

+ "; maximum = "

+ maximum);

}

dataModel.setMaximum(maximum); if (resetCurrentValues) {

dataModel.setDoubleValue(maximum);

}

}

private static void initLookAndFeel() { String lookAndFeel = null;

if (LOOKANDFEEL != null) {

if (LOOKANDFEEL.equals("Metal")) {

lookAndFeel = UIManager.getCrossPlatformLookAndFeelClassName();

} else if (LOOKANDFEEL.equals("System")) {

lookAndFeel = UIManager.getSystemLookAndFeelClassName();

} else if (LOOKANDFEEL.equals("Mac")) {

lookAndFeel = "com.sun.java.swing.plaf.mac.MacLookAndFeel";

//PENDING: check!

} else if (LOOKANDFEEL.equals("Windows")) {

lookAndFeel = "com.sun.java.swing.plaf.windows.WindowsLookAndFeel";

} else if (LOOKANDFEEL.equals("Motif")) {

lookAndFeel = "com.sun.java.swing.plaf.motif.MotifLookAndFeel";

}

if (DEBUG) {

System.out.println("About to request look and feel: "

+ lookAndFeel);

}

try {

UIManager.setLookAndFeel(lookAndFeel);

} catch (ClassNotFoundException e) {

System.err.println("Couldn't find class for specified look and feel:"

+ lookAndFeel);

System.err.println("Did you include the L&F library in the class path?"); System.err.println("Using the default look and feel.");

} catch (UnsupportedLookAndFeelException e) { System.err.println("Can't use the specified look and feel ("

+ lookAndFeel

+ ") on this platform."); System.err.println("Using the default look and feel.");

} catch (Exception e) {

System.err.println("Couldn't get specified look and feel ("

+ lookAndFeel

+ "), for some reason."); System.err.println("Using the default look and feel."); e.printStackTrace();

}

}

}

public static void main(String[] args) { initLookAndFeel();

Converter converter = new Converter();

//Create a new window.

JFrame f = new JFrame("Converter"); f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) { System.exit(0);

}

});

//Add the JPanel to the window and display the window.

//We can use a JPanel for the content pane because

//JPanel is opaque. f.setContentPane(converter.mainPane); if (COLORS) {

//This has no effect, since the JPanel completely

//covers the content pane. f.getContentPane().setBackground(Color.green);

}

f.pack(); //Resizes the window to its natural size. f.setVisible(true);

}

}

### Program 7.71 ConversionPanel.java SWING programi

/\*

* 1.1+Swing version.

\*/

import javax.swing.\*; import javax.swing.event.\*; import java.awt.\*;

import java.awt.event.\*; import java.util.\*;

import java.text.NumberFormat;

public class ConversionPanel extends JPanel { DecimalField textField;

JComboBox unitChooser;

JSlider slider;

ConverterRangeModel sliderModel; Converter controller;

Unit[] units;

String title;

final static boolean DEBUG = false; final static boolean COLORS = false; final static int MAX = 10000;

ConversionPanel(Converter myController, String myTitle, Unit[] myUnits,

ConverterRangeModel myModel) { if (COLORS) {

setBackground(Color.cyan);

}

setBorder(BorderFactory.createCompoundBorder( BorderFactory.createTitledBorder(myTitle), BorderFactory.createEmptyBorder(5,5,5,5)));

//Save arguments in instance variables. controller = myController;

units = myUnits; title = myTitle;

sliderModel = myModel;

//Add the text field. It initially displays "0" and needs

//to be at least 10 columns wide.

NumberFormat numberFormat = NumberFormat.getNumberInstance(); numberFormat.setMaximumFractionDigits(2);

textField = new DecimalField(0, 10, numberFormat); textField.setValue(sliderModel.getDoubleValue()); textField.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { sliderModel.setDoubleValue(textField.getValue());

}

});

//Add the combo box. unitChooser = new JComboBox();

for (int i = 0; i < units.length; i++) { //Populate it. unitChooser.addItem(units[i].description);

}

unitChooser.setSelectedIndex(0); sliderModel.setMultiplier(units[0].multiplier); unitChooser.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//Set new maximums for the sliders. int i = unitChooser.getSelectedIndex();

sliderModel.setMultiplier(units[i].multiplier); controller.resetMaxValues(false);

}

});

//Add the slider.

slider = new JSlider(sliderModel); sliderModel.addChangeListener(new ChangeListener() {

public void stateChanged(ChangeEvent e) { textField.setValue(sliderModel.getDoubleValue());

}

});

//Make the textfield/slider group a fixed size. JPanel unitGroup = new JPanel() {

public Dimension getMinimumSize() { return getPreferredSize();

}

public Dimension getPreferredSize() { return new Dimension(150,

super.getPreferredSize().height);

}

public Dimension getMaximumSize() { return getPreferredSize();

}

};

if (COLORS) {

unitGroup.setBackground(Color.blue);

}

unitGroup.setBorder(BorderFactory.createEmptyBorder( 0,0,0,5));

unitGroup.setLayout(new BoxLayout(unitGroup,

BoxLayout.Y\_AXIS)); unitGroup.add(textField); unitGroup.add(slider);

setLayout(new BoxLayout(this, BoxLayout.X\_AXIS)); add(unitGroup);

add(unitChooser); unitGroup.setAlignmentY(TOP\_ALIGNMENT); unitChooser.setAlignmentY(TOP\_ALIGNMENT);

}

/\*\*

* + Returns the multiplier (units/meter) for the currently
  + selected unit of measurement.

\*/

public double getMultiplier() { return sliderModel.getMultiplier();

}

public double getValue() {

return sliderModel.getDoubleValue();

}

}

### Program 7.72 Unit.java SWING programi

/\*

* 1.1+Swing version. (NOTYET)

\*/

public class Unit { String description;

double multiplier;

Unit(String description, double multiplier) { super();

this.description = description; this.multiplier = multiplier;

}

public String toString() {

String s = "Meters/" + description + " = " + multiplier; return s;

}

}

### Program 7.73 ConverterRangeModel.java SWING programi

/\*

* 1.1+Swing version.

\*/

import javax.swing.\*; import javax.swing.event.\*;

/\*\*

* Based on the source code for DefaultBoundedRangeModel,
* this class stores its value as a double, rather than
* an int. The minimum value and extent are always 0.

\*\*/

public class ConverterRangeModel implements BoundedRangeModel { protected ChangeEvent changeEvent = null;

protected EventListenerList listenerList = new EventListenerList();

protected int maximum = 10000; protected int minimum = 0; protected int extent = 0; protected double value = 0.0; protected double multiplier = 1.0;

protected boolean isAdjusting = false; final static boolean DEBUG = false;

public ConverterRangeModel() {

}

public double getMultiplier() { if (DEBUG) {

System.out.println("In ConverterRangeModel getMultiplier");

}

return multiplier;

}

public void setMultiplier(double multiplier) { if (DEBUG) {

System.out.println("In ConverterRangeModel setMultiplier");

}

this.multiplier = multiplier; fireStateChanged();

}

public int getMaximum() { if (DEBUG) {

System.out.println("In ConverterRangeModel getMaximum");

}

return maximum;

}

public void setMaximum(int newMaximum) { if (DEBUG) {

System.out.println("In ConverterRangeModel setMaximum");

}

setRangeProperties(value, extent, minimum, newMaximum, isAdjusting);

}

public int getMinimum() { return (int)minimum;

}

public void setMinimum(int newMinimum) { System.out.println("In ConverterRangeModel setMinimum");

//Do nothing.

}

public int getValue() { if (DEBUG) {

System.out.println("In ConverterRangeModel getValue");

}

return (int)getDoubleValue();

}

public void setValue(int newValue) { if (DEBUG) {

System.out.println("In ConverterRangeModel setValue");

}

setDoubleValue((double)newValue);

}

public double getDoubleValue() { if (DEBUG) {

System.out.println("In ConverterRangeModel getDoubleValue");

}

return value;

}

public void setDoubleValue(double newValue) { if (DEBUG) {

System.out.println("In ConverterRangeModel setDoubleValue");

}

setRangeProperties(newValue, extent, minimum, maximum, isAdjusting);

}

public int getExtent() { return (int)extent;

}

public void setExtent(int newExtent) {

//Do nothing.

}

public boolean getValueIsAdjusting() { return isAdjusting;

}

public void setValueIsAdjusting(boolean b) { setRangeProperties(value, extent, minimum, maximum, b);

}

public void setRangeProperties(int newValue,

int newExtent, int newMin, int newMax,

boolean newAdjusting) {

System.out.println("In ConverterRangeModel setRangeProperties"); setRangeProperties((double)newValue,

newExtent, newMin, newMax, newAdjusting);

}

public void setRangeProperties(double newValue,

int unusedExtent, int unusedMin, int newMax,

boolean newAdjusting) {

if (DEBUG) {

System.out.println("setRangeProperties(): "

+ "newValue = " + newValue

+ "; newMax = " + newMax);

}

if (newMax <= minimum) { newMax = minimum + 1; if (DEBUG) {

System.out.println("maximum raised by 1 to " + newMax);

}

}

if (Math.round(newValue) > newMax) { //allow some rounding error newValue = newMax;

if (DEBUG) {

System.out.println("value lowered to " + newMax);

}

}

boolean changeOccurred = false; if (newValue != value) {

if (DEBUG) {

System.out.println("value set to " + newValue);

}

value = newValue; changeOccurred = true;

}

if (newMax != maximum) { if (DEBUG) {

System.out.println("maximum set to " + newMax);

}

maximum = newMax; changeOccurred = true;

}

if (newAdjusting != isAdjusting) { maximum = newMax; isAdjusting = newAdjusting; changeOccurred = true;

}

if (changeOccurred) { fireStateChanged();

}

}

/\*

* + The rest of this is event handling code copied from
  + DefaultBoundedRangeModel.

\*/

public void addChangeListener(ChangeListener l) { listenerList.add(ChangeListener.class, l);

}

public void removeChangeListener(ChangeListener l) { listenerList.remove(ChangeListener.class, l);

}

protected void fireStateChanged() {

Object[] listeners = listenerList.getListenerList(); for (int i = listeners.length - 2; i >= 0; i -=2 ) {

if (listeners[i] == ChangeListener.class) { if (changeEvent == null) {

changeEvent = new ChangeEvent(this);

}

((ChangeListener)listeners[i+1]).stateChanged(changeEvent);

}

}

}

}

### Program 7.74 FollowerRangeModel.java SWING programi

/\*

* 1.1+Swing version.

\*/

import javax.swing.\*; import javax.swing.event.\*;

public class FollowerRangeModel extends ConverterRangeModel

implements ChangeListener { ConverterRangeModel dataModel;

public FollowerRangeModel(ConverterRangeModel dataModel) { this.dataModel = dataModel; dataModel.addChangeListener(this);

}

public void stateChanged(ChangeEvent e) { fireStateChanged();

}

public int getMaximum() {

int modelMax = dataModel.getMaximum();

double multiplyBy = dataModel.getMultiplier()/multiplier; if (DEBUG) {

System.out.println("In FollowerRangeModel getMaximum"); System.out.println(" dataModel.getMaximum = " + modelMax

+ "; multiply by " + multiplyBy

+ "; result: " + modelMax\*multiplyBy);

}

return (int)(modelMax \* multiplyBy);

}

public void setMaximum(int newMaximum) { dataModel.setMaximum((int)(newMaximum \*

(multiplier/dataModel.getMultiplier())));

}

public int getValue() {

return (int)getDoubleValue();

}

public void setValue(int newValue) { setDoubleValue((double)newValue);

}

public double getDoubleValue() { return dataModel.getDoubleValue()

* + dataModel.getMultiplier()

/ multiplier;

}

public void setDoubleValue(double newValue) { dataModel.setDoubleValue(

newValue \* multiplier

/ dataModel.getMultiplier());

}

public int getExtent() { return super.getExtent();

}

public void setExtent(int newExtent) { super.setExtent(newExtent);

}

public void setRangeProperties(int value,

int extent, int min, int max,

boolean adjusting) {

double multiplyBy = multiplier/dataModel.getMultiplier(); dataModel.setRangeProperties(value\*multiplyBy,

extent, min, (int)(max\*multiplyBy), adjusting);

}

}

### Program 7.75 DecimalField.java SWING programi

import javax.swing.\*; import javax.swing.text.\*;

import java.awt.Toolkit; import java.text.\*;

public class DecimalField extends JTextField { private NumberFormat format;

public DecimalField(double value, int columns, NumberFormat f) { super(columns);

setDocument(new FormattedDocument(f)); format = f;

setValue(value);

}

public double getValue() { double retVal = 0.0;

try {

retVal = format.parse(getText()).doubleValue();

} catch (ParseException e) {

// This should never happen because insertString allows

// only properly formatted data to get in the field. Toolkit.getDefaultToolkit().beep(); System.err.println("getValue: could not parse: " + getText());

}

return retVal;

}

public void setValue(double value) { setText(format.format(value));

}

}

### Program 7.76 FormattedDocument.java SWING programi

import javax.swing.\*; import javax.swing.text.\*;

import java.awt.Toolkit; import java.text.\*; import java.util.Locale;

public class FormattedDocument extends PlainDocument { private Format format;

public FormattedDocument(Format f) { format = f;

}

public Format getFormat() { return format;

}

public void insertString(int offs, String str, AttributeSet a) throws BadLocationException {

String currentText = getText(0, getLength());

String beforeOffset = currentText.substring(0, offs);

String afterOffset = currentText.substring(offs, currentText.length()); String proposedResult = beforeOffset + str + afterOffset;

try {

format.parseObject(proposedResult); super.insertString(offs, str, a);

} catch (ParseException e) { Toolkit.getDefaultToolkit().beep();

System.err.println("insertString: could not parse: " + proposedResult);

}

}

public void remove(int offs, int len) throws BadLocationException { String currentText = getText(0, getLength());

String beforeOffset = currentText.substring(0, offs);

String afterOffset = currentText.substring(len + offs, currentText.length()); String proposedResult = beforeOffset + afterOffset;

try {

if (proposedResult.length() != 0) format.parseObject(proposedResult);

super.remove(offs, len);

} catch (ParseException e) { Toolkit.getDefaultToolkit().beep();

System.err.println("remove: could not parse: " + proposedResult);

}

}

}

### Program 7.77 FollowerRangeModel.java SWING programi

/\*

* Swing versiyonu.

\*/

import javax.swing.\*; import javax.swing.event.\*;

public class FollowerRangeModel extends ConverterRangeModel

implements ChangeListener { ConverterRangeModel dataModel;

public FollowerRangeModel(ConverterRangeModel dataModel) { this.dataModel = dataModel; dataModel.addChangeListener(this);

}

public void stateChanged(ChangeEvent e) { fireStateChanged();

}

public int getMaximum() {

int modelMax = dataModel.getMaximum();

double multiplyBy = dataModel.getMultiplier()/multiplier; if (DEBUG) {

System.out.println("In FollowerRangeModel getMaximum"); System.out.println(" dataModel.getMaximum = " + modelMax

+ "; multiply by " + multiplyBy

+ "; result: " + modelMax\*multiplyBy);

}

return (int)(modelMax \* multiplyBy);

}

public void setMaximum(int newMaximum) { dataModel.setMaximum((int)(newMaximum \*

(multiplier/dataModel.getMultiplier())));

}

public int getValue() {

return (int)getDoubleValue();

}

public void setValue(int newValue) { setDoubleValue((double)newValue);

}

public double getDoubleValue() { return dataModel.getDoubleValue()

* + dataModel.getMultiplier()

/ multiplier;

}

public void setDoubleValue(double newValue) { dataModel.setDoubleValue(

newValue \* multiplier

/ dataModel.getMultiplier());

}

public int getExtent() { return super.getExtent();

}

public void setExtent(int newExtent) { super.setExtent(newExtent);

}

public void setRangeProperties(int value,

int extent, int min, int max,

boolean adjusting) {

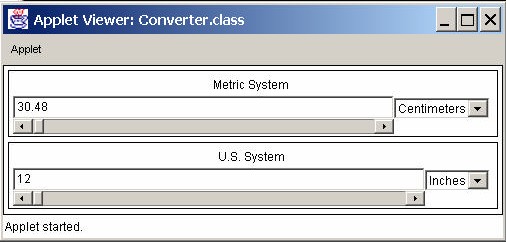
double multiplyBy = multiplier/dataModel.getMultiplier(); dataModel.setRangeProperties(value\*multiplyBy,

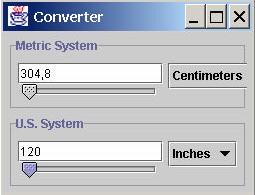
extent, min, (int)(max\*multiplyBy), adjusting);

}

}

### 07067.JPG



**Sekil 7.67 AWT Converter (birim dönüstürücü) 07068.JPG**

**Sekil 7.68 Swing Converter (birim dönüstürücü)**

1. **awt –swing karsilastirmali örnekler 4** : awt List örnegi : **ListDemo.java**

bu programin swing esdegeri ana metinde verilmisti. Simdi de swing versiyonuna göz atalim

### Program 7.78 ListDemo.java AWT programi

/\*

* awt versiyonu

\*/

import java.awt.\*; import java.awt.event.\*;

import java.applet.Applet;

public class ListDemo extends Applet

implements ActionListener, ItemListener {

TextArea output;

List turkce, italyanca;

String newline;

public void init() {

newline = System.getProperty("line.separator");

//Build

turkce = new List(4, true); //prefer 4 items visible turkce.add("bir");

turkce.add("iki");

turkce.add("üç");

turkce.add("dört");

turkce.add("bes");

turkce.add("alti");

turkce.add("yedi"); turkce.addActionListener(this); turkce.addItemListener(this);

//Build second list, which allows one selection at a time.

italyanca = new List(); //Defaults to none visible, only one selectable italyanca.add("uno");

italyanca.add("due"); italyanca.add("tre"); italyanca.add("quattro"); italyanca.add("cinque"); italyanca.add("sei"); italyanca.add("sette"); italyanca.addActionListener(this); italyanca.addItemListener(this);

//Add lists to the Applet.

GridBagLayout gridBag = new GridBagLayout(); setLayout(gridBag);

//Can't put text area on right due to GBL bug

//(can't span rows in any column but the first). output = new TextArea(10, 40); output.setEditable(false);

GridBagConstraints tc = new GridBagConstraints(); tc.fill = GridBagConstraints.BOTH;

tc.weightx = 1.0;

tc.weighty = 1.0;

tc.gridheight = 2; gridBag.setConstraints(output, tc); add(output);

GridBagConstraints lc = new GridBagConstraints(); lc.fill = GridBagConstraints.VERTICAL;

lc.gridwidth = GridBagConstraints.REMAINDER; //end row gridBag.setConstraints(turkce, lc);

add(turkce); gridBag.setConstraints(italyanca, lc); add(italyanca);

}

public void actionPerformed(ActionEvent e) { List list = (List)(e.getSource());

String language = (list == turkce) ?

"turkce" : "italyanca"; output.append("Action event olustu "

+ list.getSelectedItem() + "\" "

+ language + " dilinde." + newline);

}

public void itemStateChanged(ItemEvent e) { List list = (List)(e.getItemSelectable()); String language = (list == turkce) ?

"turkce" : "italyanca";

int index = ((Integer)(e.getItem())).intValue();

if (e.getStateChange() == ItemEvent.SELECTED) { output.append("item #"

+ index + " seçildi "

+ list.getItem(index) + " "

+ language + " dilinde. " + newline);

} else { //the item was deselected output.append("item #"

+ index + " iptal edildi "

+ list.getItem(index) + "\") "

+ language + " dilinde. " + newline);

}

}

}

### 07069.JPG



**Sekil 7.69 ListDemo.java (awt)**

**20.** bu program ListDialogu tanimlamakta ve kullanmaktadir.

### Program 7.79 ListDialog.java AWT programi

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

import BasicWindowMonitor;

public class ListDialog extends JDialog { private static ListDialog dialog; private static String value = "";

private JList list;

public static void initialize(Component comp,

String[] possibleValues, String title,

String labelText) {

Frame frame = JOptionPane.getFrameForComponent(comp); dialog = new ListDialog(frame, possibleValues,

title, labelText);

}

public static String showDialog(Component comp, String initialValue) { if (dialog != null) {

dialog.setValue(initialValue); dialog.setLocationRelativeTo(comp); dialog.setVisible(true);

} else {

System.err.println("ListDialog hatasi");

}

return value;

}

private void setValue(String newValue) {

value = newValue; list.setSelectedValue(value, true);

}

private ListDialog(Frame frame, Object[] data, String title, String labelText) {

super(frame, title, true);

//buttons

JButton cancelButton = new JButton("Iptal et"); final JButton setButton = new JButton("seç");

cancelButton.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent e) {

ListDialog.dialog.setVisible(false);

}

});

setButton.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent e) {

ListDialog.value = (String)(list.getSelectedValue()); ListDialog.dialog.setVisible(false);

}

});

getRootPane().setDefaultButton(setButton);

// ana dialog kismi list = new JList(data);

list.setSelectionMode(ListSelectionModel.SINGLE\_INTERVAL\_SELECTION); list.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) { if (e.getClickCount() == 2) {

setButton.doClick();

}

}

});

JScrollPane listScroller = new JScrollPane(list); listScroller.setPreferredSize(new Dimension(250, 80));

listScroller.setMinimumSize(new Dimension(250, 80)); listScroller.setAlignmentX(LEFT\_ALIGNMENT); JPanel listPane = new JPanel();

listPane.setLayout(new BoxLayout(listPane, BoxLayout.Y\_AXIS)); JLabel label = new JLabel(labelText);

label.setLabelFor(list); listPane.add(label);

listPane.add(Box.createRigidArea(new Dimension(0,5))); listPane.add(listScroller); listPane.setBorder(BorderFactory.createEmptyBorder(10,10,10,10)); JPanel buttonPane = new JPanel();

buttonPane.setLayout(new BoxLayout(buttonPane, BoxLayout.X\_AXIS)); buttonPane.setBorder(BorderFactory.createEmptyBorder(0, 10, 10, 10)); buttonPane.add(Box.createHorizontalGlue()); buttonPane.add(cancelButton);

buttonPane.add(Box.createRigidArea(new Dimension(10, 0))); buttonPane.add(setButton);

Container contentPane = getContentPane(); contentPane.add(listPane, BorderLayout.CENTER); contentPane.add(buttonPane, BorderLayout.SOUTH);

pack();

}

public static void main(String[] args) {

String[] names = {"Turhan", "Meral", "Nurhan", "Irfan", "Birsen", "Selin", "Osman", "Hatice", "Hayati", "Arzu", "Yagmur", "Bulut"};

JFrame f = new JFrame("Bebek ismini seçiniz"); f.addWindowListener(new BasicWindowMonitor()); JLabel intro = new JLabel("Seçilen isim :");

final JLabel name = new JLabel("Turhan"); intro.setLabelFor(name); name.setForeground(Color.black);

JButton button = new JButton("Yeni bir isim seçiniz..."); ListDialog.initialize(f, names, "Isim seçme dialogu",

"bebekler isim listesi :"); button.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

String selectedName = ListDialog.showDialog(null,name.getText()); name.setText(selectedName);

}

});

JPanel contentPane = new JPanel(); f.setContentPane(contentPane);

contentPane.setLayout(new BoxLayout(contentPane, BoxLayout.Y\_AXIS)); contentPane.setBorder(BorderFactory.createEmptyBorder(20,20,20,20)); contentPane.add(intro);

contentPane.add(name); contentPane.add(Box.createRigidArea(new Dimension(0,10))); contentPane.add(button); intro.setAlignmentX(JComponent.CENTER\_ALIGNMENT); name.setAlignmentX(JComponent.CENTER\_ALIGNMENT); button.setAlignmentX(JComponent.CENTER\_ALIGNMENT); f.pack();

f.setVisible(true);

}

}

### 07070.JPG



**Sekil 7.70 ListDialog.java**

1. bu program islem takip çubuk grafigi olan progressMonitor tanimlamakta ve kullanmaktadir.

**Program 7.80 :** ProgressMonitorDemo.java

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class ProgressMonitorDemo extends JFrame { public final static int ONE\_SECOND = 1000;

private ProgressMonitor progressMonitor; private Timer timer;

private JButton startButton; private LongTask task; private JTextArea taskOutput; private String newline;

public ProgressMonitorDemo() { super("ProgressMonitorDemo");

newline = System.getProperty("line.separator"); task = new LongTask();

//create the demo's UI

startButton = new JButton("Start"); startButton.setActionCommand("start"); startButton.addActionListener(new ButtonListener()); taskOutput = new JTextArea(5, 20); taskOutput.setMargin(new Insets(5,5,5,5)); taskOutput.setEditable(false);

JPanel contentPane = new JPanel(); contentPane.setLayout(new BorderLayout()); contentPane.add(startButton, BorderLayout.NORTH);

contentPane.add(new JScrollPane(taskOutput), BorderLayout.CENTER); contentPane.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20)); setContentPane(contentPane);

//create a timer

timer = new Timer(ONE\_SECOND, new TimerListener());

}

//the actionPerformed method in this class

//is called each time the Timer "goes off"

class TimerListener implements ActionListener { public void actionPerformed(ActionEvent evt) {

if (progressMonitor.isCanceled() || task.done()) { progressMonitor.close();

task.stop(); Toolkit.getDefaultToolkit().beep(); timer.stop(); startButton.setEnabled(true);

} else {

progressMonitor.setNote(task.getMessage()); progressMonitor.setProgress(task.getCurrent()); taskOutput.append(task.getMessage() + newline); taskOutput.setCaretPosition(taskOutput.getDocument().getLength());

}

}

}

//the actionPerformed method in this class

//is called when the user presses the start button class ButtonListener implements ActionListener {

public void actionPerformed(ActionEvent evt) {

progressMonitor = new ProgressMonitor(ProgressMonitorDemo.this, "Running a Long Task",

"", 0, task.getLengthOfTask()); progressMonitor.setProgress(0); progressMonitor.setMillisToDecideToPopup(2 \* ONE\_SECOND); startButton.setEnabled(false);

task.go(); timer.start();

}

}

public static void main(String[] args) {

JFrame frame = new ProgressMonitorDemo();

frame.addWindowListener(new WindowAdapter() {

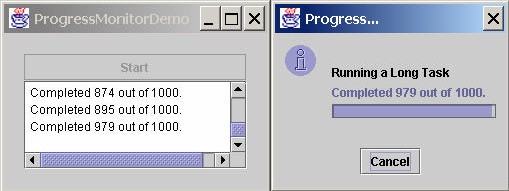
public void windowClosing(WindowEvent e) {System.exit(0);}

});

frame.pack(); frame.setVisible(true);

}

}



1. bu program detayli bir dosya seçme sistemi tanimlamaktadir.

**Program 7.81 :** FileChooserDemo.java

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import javax.swing.\*;

import javax.swing.filechooser.\*;

public class FileChooserDemo extends JFrame {

static private final String newline = System.getProperty("line.separator"); public FileChooserDemo() {

super("FileChooserDemo");

//Create the log first, because the action listeners

//need to refer to it.

final JTextArea log = new JTextArea(5,20); log.setMargin(new Insets(5,5,5,5));

JScrollPane logScrollPane = new JScrollPane(log);

//Create a file chooser

final JFileChooser fc = new JFileChooser();

//Create the open button

ImageIcon openIcon = new ImageIcon("images/open.gif"); JButton openButton = new JButton("Open a File...", openIcon); openButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int returnVal = fc.showOpenDialog(FileChooserDemo.this);

if (returnVal == JFileChooser.APPROVE\_OPTION) { File file = fc.getSelectedFile();

//this is where a real application would open the file. log.append("Opening: " + file.getName() + "." + newline);

} else {

log.append("Open command cancelled by user." + newline);

}

}

});

//Create the save button

ImageIcon saveIcon = new ImageIcon("images/save.gif"); JButton saveButton = new JButton("Save a File...", saveIcon); saveButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int returnVal = fc.showSaveDialog(FileChooserDemo.this);

if (returnVal == JFileChooser.APPROVE\_OPTION) { File file = fc.getSelectedFile();

//this is where a real application would save the file. log.append("Saving: " + file.getName() + "." + newline);

} else {

log.append("Save command cancelled by user." + newline);

}

}

});

//For layout purposes, put the buttons in a separate panel JPanel buttonPanel = new JPanel(); buttonPanel.add(openButton); buttonPanel.add(saveButton);

//Add the buttons and the log to the frame Container contentPane = getContentPane();

contentPane.add(buttonPanel, BorderLayout.NORTH); contentPane.add(logScrollPane, BorderLayout.CENTER);

}

public static void main(String s[]) {

JFrame frame = new FileChooserDemo();

frame.addWindowListener(new WindowAdapter() {

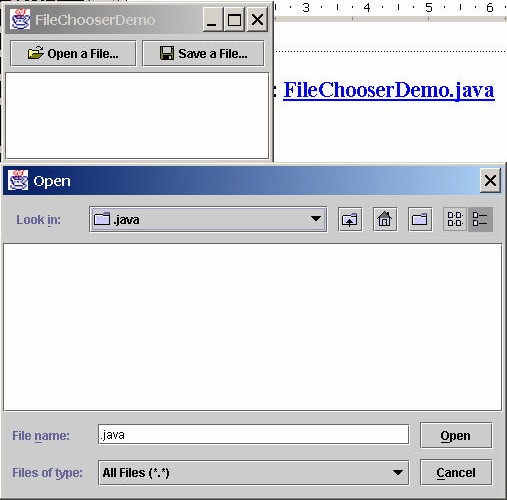
public void windowClosing(WindowEvent e) {System.exit(0);}

});

frame.pack(); frame.setVisible(true);

}

}



1. bu program detayli bir dosya seçme sistemi tanimlamaktadir.

**Program 7.82 :** FileChooserDemo1.java

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import javax.swing.\*;

import javax.swing.filechooser.\*;

public class FileChooserDemo2 extends JFrame {

static private String newline = System.getProperty("line.separator"); public FileChooserDemo2() {

super("FileChooserDemo2");

//Create the log first, because the action listener

//needs to refer to it.

final JTextArea log = new JTextArea(5,20); log.setMargin(new Insets(5,5,5,5));

JScrollPane logScrollPane = new JScrollPane(log);

JButton sendButton = new JButton("Attach..."); sendButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { JFileChooser fc = new JFileChooser(); fc.addChoosableFileFilter(new ImageFilter()); fc.setFileView(new ImageFileView()); fc.setAccessory(new ImagePreview(fc));

int returnVal = fc.showDialog(FileChooserDemo2.this, "Attach");

if (returnVal == JFileChooser.APPROVE\_OPTION) { File file = fc.getSelectedFile(); log.append("Attaching file: " + file.getName()

+ "." + newline);

} else {

log.append("Attachment cancelled by user." + newline);

}

}

});

Container contentPane = getContentPane(); contentPane.add(sendButton, BorderLayout.NORTH); contentPane.add(logScrollPane, BorderLayout.CENTER);

}

public static void main(String s[]) {

JFrame frame = new FileChooserDemo2();

frame.addWindowListener(new WindowAdapter() {

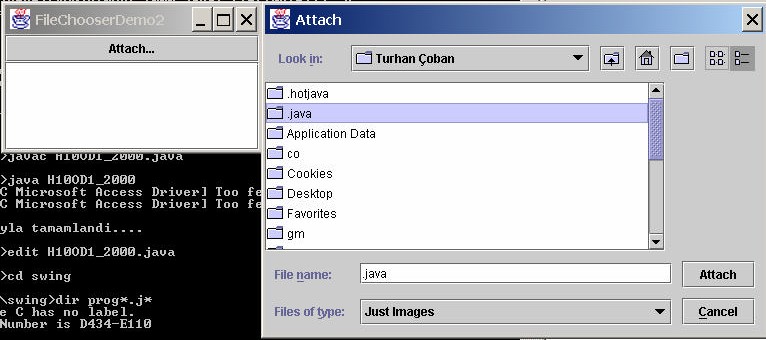
public void windowClosing(WindowEvent e) {System.exit(0);}

});

frame.pack(); frame.setVisible(true);

}

}



1. bu program detayli bir dosya seçme sistemi tanimlamaktadir.

**Program 7.83 :** TreeDemo.java

import javax.swing.JTree;

import javax.swing.tree.DefaultMutableTreeNode; import javax.swing.event.TreeSelectionListener; import javax.swing.event.TreeSelectionEvent; import javax.swing.tree.TreeSelectionModel; import java.net.URL;

import java.io.IOException; import javax.swing.JEditorPane; import javax.swing.JScrollPane; import javax.swing.JSplitPane;

import javax.swing.JFrame; import java.awt.\*;

import java.awt.event.\*;

public class TreeDemo extends JFrame { private JEditorPane htmlPane;

private static boolean DEBUG = false; private URL helpURL;

//Optionally play with line styles. Possible values are

//"Angled", "Horizontal", and "None" (the default). private boolean playWithLineStyle = false;

private String lineStyle = "Angled";

public TreeDemo() { super("TreeDemo");

//Create the nodes.

DefaultMutableTreeNode top = new DefaultMutableTreeNode("The Java Series"); createNodes(top);

//Create a tree that allows one selection at a time. final JTree tree = new JTree(top); tree.getSelectionModel().setSelectionMode

(TreeSelectionModel.SINGLE\_TREE\_SELECTION);

//Listen for when the selection changes. tree.addTreeSelectionListener(new TreeSelectionListener() {

public void valueChanged(TreeSelectionEvent e) { DefaultMutableTreeNode node = (DefaultMutableTreeNode)

tree.getLastSelectedPathComponent(); if (node == null) return;

Object nodeInfo = node.getUserObject(); if (node.isLeaf()) {

BookInfo book = (BookInfo)nodeInfo; displayURL(book.bookURL);

if (DEBUG) {

System.out.print(book.bookURL + ": \n ");

}

} else {

displayURL(helpURL);

}

if (DEBUG) {

System.out.println(nodeInfo.toString());

}

}

});

if (playWithLineStyle) { tree.putClientProperty("JTree.lineStyle", lineStyle);

}

//Create the scroll pane and add the tree to it. JScrollPane treeView = new JScrollPane(tree);

//Create the HTML viewing pane. htmlPane = new JEditorPane(); htmlPane.setEditable(false);

initHelp();

JScrollPane htmlView = new JScrollPane(htmlPane);

//Add the scroll panes to a split pane.

JSplitPane splitPane = new JSplitPane(JSplitPane.VERTICAL\_SPLIT); splitPane.setTopComponent(treeView); splitPane.setBottomComponent(htmlView);

Dimension minimumSize = new Dimension(100, 50); htmlView.setMinimumSize(minimumSize); treeView.setMinimumSize(minimumSize); splitPane.setDividerLocation(100); //XXX: ignored in some releases

//of Swing. bug 4101306

//workaround for bug 4101306:

//treeView.setPreferredSize(new Dimension(100, 100));

splitPane.setPreferredSize(new Dimension(500, 300));

//Add the split pane to this frame getContentPane().add(splitPane);

}

private class BookInfo { public String bookName; public URL bookURL; public String prefix = "file:"

+ System.getProperty("user.dir")

+ System.getProperty("file.separator"); public BookInfo(String book, String filename) {

bookName = book; try {

bookURL = new URL(prefix + filename);

} catch (java.net.MalformedURLException exc) { System.err.println("Attempted to create a BookInfo "

+ "with a bad URL: " + bookURL); bookURL = null;

}

}

public String toString() { return bookName;

}

}

private void initHelp() { String s = null;

try {

s = "file:"

+ System.getProperty("user.dir")

+ System.getProperty("file.separator")

+ "TreeDemoHelp.html"; if (DEBUG) {

System.out.println("Help URL is " + s);

}

helpURL = new URL(s); displayURL(helpURL);

} catch (Exception e) {

System.err.println("Couldn't create help URL: " + s);

}

}

private void displayURL(URL url) { try {

htmlPane.setPage(url);

} catch (IOException e) {

System.err.println("Attempted to read a bad URL: " + url);

}

}

private void createNodes(DefaultMutableTreeNode top) { DefaultMutableTreeNode category = null; DefaultMutableTreeNode book = null;

category = new DefaultMutableTreeNode("Books for Java Programmers"); top.add(category);

//original Tutorial

book = new DefaultMutableTreeNode(new BookInfo

("The Java Tutorial: Object-Oriented Programming for the Internet", "tutorial.html"));

category.add(book);

//Tutorial Continued

book = new DefaultMutableTreeNode(new BookInfo ("The Java Tutorial Continued: The Rest of the JDK", "tutorialcont.html"));

category.add(book);

//JFC Swing Tutorial

book = new DefaultMutableTreeNode(new BookInfo

("The JFC Swing Tutorial: A Guide to Constructing GUIs", "swingtutorial.html"));

category.add(book);

//Arnold/Gosling

book = new DefaultMutableTreeNode(new BookInfo ("The Java Programming Language", "arnold.html"));

category.add(book);

//FAQ

book = new DefaultMutableTreeNode(new BookInfo( "The Java FAQ", "faq.html"));

category.add(book);

//Chan/Lee

book = new DefaultMutableTreeNode(new BookInfo ("The Java Class Libraries: An Annotated Reference", "chanlee.html"));

category.add(book);

//Threads

book = new DefaultMutableTreeNode(new BookInfo

("Concurrent Programming in Java: Design Principles and Patterns", "thread.html"));

category.add(book);

category = new DefaultMutableTreeNode("Books for Java Implementers"); top.add(category);

//VM

book = new DefaultMutableTreeNode(new BookInfo ("The Java Virtual Machine Specification", "vm.html"));

category.add(book);

//Language Spec

book = new DefaultMutableTreeNode(new BookInfo ("The Java Language Specification",

"jls.html")); category.add(book);

}

public static void main(String[] args) { JFrame frame = new TreeDemo();

frame.addWindowListener(new WindowAdapter() {

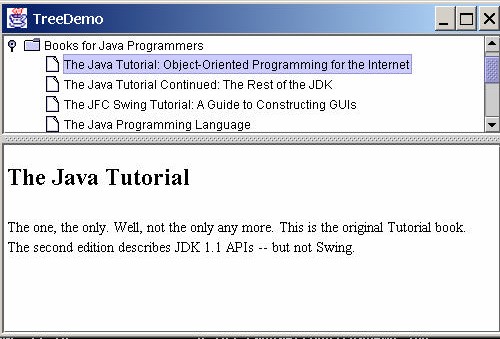
public void windowClosing(WindowEvent e) {System.exit(0);}

});

frame.pack(); frame.setVisible(true);

}

}



1. Simdi bir grafik çizim programina bakalim. Swing ve Graph2D olarak yazilmis olan bu program bir dosyadan (Plot.txt) okudugu veriye göre grafik çizmektedir.

Örnegin Plot.txt dosyasinda

Baslik x ekseni y ekseni 2

in.txt 20 0 0 0

out.txt 3 0 0 255

degerleri varsa Plot basligi olarak “Baslik”, x exseni yazisi olarak “x ekseni”, y ekseni yazisi basligi olarak “y ekseni”, toplam veri dosyasi olarak 2, veri dosyasi isimleri “in.txt” ve “out.txt”, çizim sekilleri 20 ve 3 ve renkler 00 255 ve 0 0 0 kullanmaktadir.

Önce Plot sekillerini tanimalyan PlotShapesSW.java programina göz atalim

**Program 7.84 :** plotShapesSW.java

//======================================================

// Numerical Analysis package in java

// PlotShapes class

// This class convert graphic draw methods to

// plot coordinates and gives additional plotting methods

// Dr. Turhan Coban

// =====================================================

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

public class PlotShapesSW

{

Graphics2D g;

int xabsmin,yabsmin; int absheight,abswidth;

double xmin,xmax,ymin,ymax; Font f;

final static float dash1[] = {10.0f};

final static BasicStroke dashed = new BasicStroke(10.0f,

BasicStroke.CAP\_BUTT, BasicStroke.JOIN\_MITER, 10.0f, dash1, 0.0f);

public PlotShapesSW(Graphics2D gi,int xabsmini ,int yabsmini, int absheighti,int abswidthi,

double xmini,double xmaxi, double ymini,double ymaxi)

{

// xabsmin : absulute stating point x axis

// yabsmin : absolute starting point y axis

// absheight : absoulute height of plotting window

// abswidth : absolute width of plotting window

// xmin : minimum x value (real number)

// xmax : maximum x value (real number)

// ymin : minimum y value (real number)

// ymax : maximum y value (real number)

// g : graphic object that actual drawing is done through g=gi;

//Font fonts[]=GraphicsEnvironment.getLocalGraphicsEnvironment().getAllFonts();

//f=fonts[2].deriveFont(Font.BOLD,12);

//g.setFont(f);

xabsmin=xabsmini; yabsmin=yabsmini; absheight=absheighti; abswidth=abswidthi; xmin=xmini; xmax=xmaxi; ymin=ymini; ymax=ymaxi;

}

public void drawLine(int plottype,double x1,double y1,double x2,double y2)

{

// draw a line from (x1,y1) to (x2,y2) if(plottype==0) //draw a continuous line

{

g.draw(new Line2D.Double( (int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight), (int)(xabsmin+(x2-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y2-ymin)/(ymax-ymin)\*absheight)));

}

} //end of drawLine

public void drawChar(char ch,double x1,double y1)

{

// draws a single character at (x1,y1) char ch1[]=new char[1]; ch1[0]=ch;

int h=g.getFontMetrics().getHeight(); int w=h-2;

g.drawChars(ch1,0,1,

(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth)-(int)(abswidth/200.0), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+h/2);

} //end of PlotShapes.drawChar

public void drawChars(char ch[],int firstspace,int numberofchars,

double x1,double y1)

{

// draws a character array from space firstspace to (firstspace+numberofchars)

// starting at (x1,y1)

int h=g.getFontMetrics().getHeight(); int w=h-2;

//g.drawChars(ch,firstspace,numberofchars,

// (int)((xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth)+abswidth/80.0),

// (int)((yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+h/2)); g.drawChars(ch,firstspace,numberofchars,

(int)((xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth)+abswidth/80.0), (int)((yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)));

} //end of PlotShapes.drawChars

public void drawString(String s,double x1,double y1)

{

//draws a String at (x1,y1)

int h=g.getFontMetrics().getHeight(); int w=h-2;

g.drawString(s,

(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight));

} //end of PlotShapes.drawStrings

public void drawRect(double x1,double y1,int rectwidth,int rectheight)

{

// draw a rectangle starting at (x1,y1)

// with dimensions of (rectwidth,rectheight)

g.draw(new Rectangle2D.Double((int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-rectwidth/2), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight-rectheight/2),

rectwidth,rectheight));

}//end of drawRect

public void drawEllipse(double x1,double y1,int width,int height)

{

// draw an oval with the centre of (x1,y1)

// with dimension of (width,height)

g.draw(new Ellipse2D.Double((int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-width/2), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight-height/2),

width,height));

}//end of drawRect

public void drawPolygon(double x1,double y1,int radius,int side)

{

//draw a polygon of n sides n=3(trinagle),n=4(dimond).... double Pi=Math.PI;

int xvalues[]=new int[side+1]; int yvalues[]=new int[side+1];

GeneralPath polygon=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD, xvalues.length);

double angle\_increase; double angle; angle\_increase=2.0\*Pi/side; angle=Pi/2.0;

for(int i=0;i<side;i++)

{

xvalues[i]= (int)(Math.floor(xabsmin+(x1-xmin)/ (xmax-xmin)\*abswidth+radius\*Math.cos(angle)));

yvalues[i]= (int)(Math.floor(yabsmin+absheight-(y1-ymin)/ (ymax-ymin)\*absheight-radius\*Math.sin(angle)));

if(i==0)

{

xvalues[side]=xvalues[i];yvalues[side]=yvalues[i]; polygon.moveTo(xvalues[0],yvalues[0]);

}

else

{

polygon.lineTo(xvalues[i],yvalues[i]);

}

}

g.draw(polygon);

}//end of drawPolygon

public void drawXTic(int ticNumber,double ticHeight,int ticSide)

{

//draw a series of x axis tics

double dtic=(xmax-xmin)/ticNumber; double x1=xmin;

double y1=ymin; if(ticSide==0)

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,x1,(y1-ticHeight)); x1=x1+dtic;

}

}

else

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,x1,(y1+ticHeight)); x1=x1+dtic;

}

}

}

public void drawYTic(int ticNumber,double ticWidth,int ticSide)

{

//draw a series of y axis tics

double dtic=(ymax-ymin)/ticNumber; double x1=xmin;

double y1=ymin; if(ticSide==0)

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,(x1-ticWidth),y1); y1=y1+dtic;

}

}

else

{

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,y1,(x1+ticWidth),y1); y1=y1+dtic;

}

}

}

public void drawXGrid(int ticNumber)

{

//draw x gridlies |||||||

double dtic=(xmax-xmin)/ticNumber; double x1=xmin;

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,x1,ymin,x1,ymax); x1=x1+dtic;

}

}

public void drawYGrid(int ticNumber)

{

//draw y gridlines =====

double dtic=(ymax-ymin)/ticNumber; double y1=ymin;

for(int i=0;i<=ticNumber;i++)

{

drawLine(0,xmin,y1,xmax,y1); y1=y1+dtic;

}

}

public void drawXNumbers(int ticNumber)

{

//draw x numbers

double dtic=(xmax-xmin)/ticNumber; double x1=xmin;

double y1=ymin;

String s;

s=" ";

for(int i=0;i<=ticNumber;i++)

{

s=Double.toString(Math.floor(x1\*100.0)/100.00); g.drawString(s,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth),

(int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)+(int)(abswidth/20.0)); x1=x1+dtic;

}

}

public void drawYNumbers(int ticNumber)

{

// draw y numbers

double dtic=(ymax-ymin)/ticNumber; double x1=xmin;

double y1=ymin;

String s;

s=" ";

for(int i=0;i<=ticNumber;i++)

{

s=Double.toString(Math.floor(y1\*100)/100.0)+" "; g.drawString(s,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth-abswidth/10.0), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight));

y1=y1+dtic;

}

}

public void drawXLabel(String xLabel)

{

// draw x labels

double x1=xmin+(xmax-xmin)/2.0; double y1=ymin;

g.drawString(xLabel,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)

+(int)(abswidth/12.0));

}

public void drawYLabel(String yLabel)

{

// draw y labels

double x1=xmin-(xmax-xmin)/6.0; double y1=ymax-(ymax-ymin)/3.0; int n=yLabel.length();

char ch[]=new char[n]; yLabel.getChars(0,n,ch,0);

//AffineTransform at=AffineTransform.getRotateInstance(3.0\*Math.PI/2.0,

//(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth),

//(int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight));

//g.setTransform(at);

//g.drawString(yLabel,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth),

//(int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight));

//at=AffineTransform.getRotateInstance(Math.PI/2.0,

//(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth),

//(int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight));

//g.setTransform(at); for(int i=0;i<n;i++)

{

drawChar(ch[i],x1,y1); y1=y1-(ymax-ymin)/30;

}

}

public void drawLabel(String Label)

{

//draw graphic label

double x1=xmin+(xmax-xmin)/2.0; double y1=ymax;

g.drawString(Label,(int)(xabsmin+(x1-xmin)/(xmax-xmin)\*abswidth), (int)(yabsmin+absheight-(y1-ymin)/(ymax-ymin)\*absheight)-(int)(abswidth/40.0));

}

public void

drawPlotLines(int i,int plottype[],double x[][],double y[][],int n[],char ch[])

{

int j;

//draw lines

if((plottype[i] >= 0) && (plottype[i] < 10) )

{

switch ( i )

{

case 1 : g.setStroke(new BasicStroke(1.0f) ); break; case 2 : g.setStroke(new BasicStroke(2.0f) ); break; case 3 : g.setStroke(dashed); break;

}

for(j=0;j<n[i]-1;j++)

{

if((x[i][j]>=xmin && x[i][j]<=xmax )

&& (y[i][j]>=ymin && y[i][j]<=ymax))

{

if((x[i][j+1]>=xmin && x[i][j+1]<=xmax) && (y[i][j+1]>=ymin && y[i][j+1]<=ymax))

{

drawLine(0,x[i][j],y[i][j],x[i][j+1],y[i][j+1]);

}

else if(x[i][j+1]>xmax)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j]-b\*x[i][j]; drawLine(0,x[i][j],y[i][j],xmax,(a+b\*xmax));

}

else if(y[i][j+1]>ymax)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j]-b\*x[i][j]; drawLine(0,x[i][j],y[i][j],(ymax-a)/b,ymax);

}

else if(x[i][j+1]>xmax && y[i][j+1]>ymax)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j]-b\*x[i][j]; drawLine(0,x[i][j],y[i][j],(ymax-a)/b,(a+b\*xmax));

}

}

else if((x[i][j+1]>=xmin && x[i][j+1]<=xmax ) && (y[i][j+1]>=ymin && y[i][j+1]<=ymax))

{

if(x[i][j]<xmin)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j+1]-b\*x[i][j+1];

drawLine(0,xmin,(a+b\*xmin),x[i][j+1],y[i][j+1]);

}

if(y[i][j]<ymin)

{

double b=(y[i][j+1]-y[i][j])/(x[i][j+1]-x[i][j]); double a=y[i][j+1]-b\*x[i][j+1]; drawLine(0,(ymin-a)/b,ymin,x[i][j+1],y[i][j+1]);

}

}

}//end of for(j=0

g.setStroke(new BasicStroke(1.0f));

}//end of if(plottype

//draw characters

else if(plottype[i]==10)

{

for(j=0;j<n[i];j++)

{

if((x[i][j]>=xmin && x[i][j]<=xmax )

&& (y[i][j]>=ymin && y[i][j]<=ymax))

{ drawChar(ch[i],x[i][j],y[i][j]); }

}//end of for(j=0;

}//end else if(plottype[i]==10)

//draw rectangles

else if(plottype[i]==20)

{

for(j=0;j<n[i];j++)

{

if((x[i][j]>=xmin && x[i][j]<=xmax )

&& (y[i][j]>=ymin && y[i][j]<=ymax))

{ drawRect(x[i][j],y[i][j],abswidth/100,abswidth/80); }

}//end of for(j=0;

}//end else if(plottype[i]==20)

//draw circle

else if(plottype[i]==21)

{

for(j=0;j<n[i];j++)

{

if((x[i][j]>=xmin && x[i][j]<=xmax )

&& (y[i][j]>=ymin && y[i][j]<=ymax))

{ drawEllipse(x[i][j],y[i][j],abswidth/100,abswidth/80); }

}//end of for(j=0;

}//end else if(plottype[i]==21)

else if(plottype[i]>=22 && plottype[i]<=27)

{

for(j=0;j<n[i];j++)

{ if((x[i][j]>=xmin && x[i][j]<=xmax )

&& (y[i][j]>=ymin && y[i][j]<=ymax))

{drawPolygon(x[i][j],y[i][j],5,(plottype[i]-(int)(abswidth/20.0)));}

}//end of for(j=0;

}//end else if(plottype[i]==21..27)

}

}

Burada tanimlanan sekil, çizgi vs gerçek plot koordinat sistemini kullanmakta ve pencere koordinat sistemiyle gerçek koordinat sistemleri arasinda otomatik dönüsüm yapmaktadir.

**Program 7.85 :** Plot2D.java

import java.io.\*;

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*;

import java.awt.print.PrinterJob; import java.awt.print.\*;

public class Plot2D extends JPanel implements Printable

{

public Plot p1;

final static Color bg = Color.white; final static Color fg = Color.black; final static Color red = Color.red; final static Color white = Color.white;

final static BasicStroke stroke = new BasicStroke(1.0f); final static BasicStroke boldStroke = new BasicStroke(2.0f);

public Plot2D()

{

//Initialize drawing colors setBackground(Color.white); setForeground(Color.black);

try{

p1=new Plot();

} catch(IOException ioe) {System.err.println("IOExceptionin opening plot");}

}

public void yenidanPlotDatasiOku()

{

try{

p1=new Plot();

} catch(IOException ioe) {System.err.println("IOExceptionin opening plot");}

}

public void yenidenciz()

{

repaint();

}

public void yazdir()

{

PrinterJob printJob = PrinterJob.getPrinterJob(); printJob.setPrintable(this);

if (printJob.printDialog()) { try {

printJob.print();

} catch (Exception ex) {ex.printStackTrace();}

}

}

public int print(Graphics g, PageFormat pf, int pi) throws PrinterException { if (pi >= 1) {

return Printable.NO\_SUCH\_PAGE;

}

drawShapes((Graphics2D)g); return Printable.PAGE\_EXISTS;

}

public void drawShapes(Graphics2D g2)

{

Dimension d=getSize(); g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

int gridWidth = d.width; int gridHeight = d.height;

int rowspacing=(int)(d.width/100.0); int columnspacing=(int)(d.height/50.0);

int rectWidth=gridWidth-columnspacing; int rectHeight=gridHeight-rowspacing; p1.xabsmin=(int)(0.25\*rectWidth); p1.yabsmin=(int)(0.2\*rectHeight); p1.abswidth=(int)(0.65\*rectWidth); p1.absheight=(int)(0.7\*rectHeight);

//g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,

RenderingHints.VALUE\_ANTIALIAS\_OFF); g2.setStroke(stroke);

g2.setPaint(fg); g2.drawRect(p1.xabsmin,p1.yabsmin,p1.abswidth,p1.absheight); PlotShapesSW ps=new PlotShapesSW(g2,p1.xabsmin,p1.yabsmin, p1.absheight,p1.abswidth,p1.xmin,p1.xmax,p1.ymin,p1.ymax); if(p1.xgridon!=0)

ps.drawXGrid(p1.xntic); if(p1.ygridon!=0)

ps.drawYGrid(p1.yntic); int i,j; for(i=0;i<p1.nline;i++)

{

// Select plot colors

g2.setPaint(new Color(p1.red[i],p1.green[i],p1.blue[i])); ps.drawPlotLines(i,p1.plottype,p1.x,p1.y,p1.n,p1.ch);

}//end of for(i=0 g2.setPaint(fg);

ps.drawXTic(p1.xntic,(p1.ymax-p1.ymin)/80.0,0); ps.drawYTic(p1.yntic,(p1.xmax-p1.xmin)/80.0,0); ps.drawXNumbers(p1.xntic); ps.drawYNumbers(p1.xntic); ps.drawLabel(p1.label); ps.drawXLabel(p1.xlabel); ps.drawYLabel(p1.ylabel);

}

public void paintComponent(Graphics g)

{

super.paintComponent(g); Graphics2D g2 = (Graphics2D) g; drawShapes(g2);

}

}

bu program plot sekillerini Plot olusturacak sekilde bir araya koymaktadir.

**Program 7.85 :** PlotWindowSWF\_2000.java

//======================================================

// Numerical Analysis package in java

// Plot (Çizim ) programi

// Dr. Turhan Coban

// =====================================================

/\*

* Swing version.

\*/

import java.lang.Integer; import java.awt.\*; import java.awt.event.\*; import java.awt.font.\*; import java.awt.geom.\*; import java.awt.image.\*; import javax.swing.\*; import Plot2D;

import PlotShapesSW;

import BasicWindowMonitor;

public class PlotWindowSWF\_2000 extends JFrame implements ItemListener,ActionListener

{

boolean inAnApplet = true;

final static String KONTROLPANEL = "Kontrol sayfasi"; final static String PLOTPANEL = "Plot sayfasi "; Plot2D jta;

JLabel promptXmin; // Label prompt in Xmin field JLabel promptXmax; // Label prompt in Xmax field JLabel promptYmin; // Label prompt in Ymin field JLabel promptYmax; // Label prompt in Ymax field JLabel promptLabel; // Label prompt Plot Label JLabel promptXLabel; // Label prompt Plot XLabel JLabel promptYLabel; // Label prompt Plot YLabel JLabel promptXntic; // Label prompt in Xmin field JLabel promptYntic; // Label prompt in Xmax field JLabel promptXgridon; // Label prompt in Ymin field JLabel promptYgridon; // Label prompt in Ymax field JTextField inputXmin; // input field Xmin

JTextField inputXmax; // input field Xmax JTextField inputYmin; // input field Ymin JTextField inputYmax; // input field Ymax JTextField inputLabel; // input field Label JTextField inputXLabel; // input field XLabel JTextField inputYLabel; // input field YLabel JTextField inputXntic; // input field xntic JTextField inputYntic; // input field yntic JCheckBox inputXgridon; // input field xgridon JCheckBox inputYgridon; // input field ygridon JButton printButton;

public PlotWindowSWF\_2000()

{

super("Plot çizimi");

Container contentPane = getContentPane(); JTabbedPane tabbedPane = new JTabbedPane();

promptXmin=new JLabel("Xmin "); inputXmin=new JTextField(5); promptXmax=new JLabel("Xmax "); inputXmax=new JTextField(5); promptYmin=new JLabel("Ymin "); inputYmin=new JTextField(5); promptYmax=new JLabel("Ymax "); inputYmax=new JTextField(5);

//\*\*\*\*\*\*\*

promptLabel=new JLabel(" Plot basligi : "); promptXLabel=new JLabel(" x ekseni basligi : ");

promptYLabel=new JLabel(" y ekseni basligi : "); inputLabel=new JTextField(30);

inputXLabel=new JTextField(30); inputYLabel=new JTextField(30);

//\*\*\*\*\*\*\*

promptXntic=new JLabel("X tik no"); inputXntic=new JTextField(5); promptYntic=new JLabel("Y tik no"); inputYntic=new JTextField(5);

promptXgridon=new JLabel("X grid (küçük kare)"); inputXgridon=new JCheckBox(" "); promptYgridon=new JLabel("Y grid (küçük kare)"); inputYgridon=new JCheckBox(" ");

//\*\*\*\*\*\*\*

JPanel pane1 = new JPanel()

{

public Dimension getPrefferedSize()

{

Dimension size=super.getPreferredSize(); size.width=800;

return size;

}

};

JPanel mpane=new JPanel(); mpane.setLayout(new GridLayout(1,8)); mpane.add(promptXmin); mpane.add(inputXmin); mpane.add(promptXmax); mpane.add(inputXmax); mpane.add(promptYmin); mpane.add(inputYmin); mpane.add(promptYmax); mpane.add(inputYmax); pane1.add(mpane,BorderLayout.NORTH); JPanel xpane=new JPanel(); xpane.setLayout(new GridLayout(1,8)); xpane.add(promptXntic); xpane.add(inputXntic); xpane.add(promptYntic); xpane.add(inputYntic); xpane.add(promptXgridon); xpane.add(inputXgridon); xpane.add(promptYgridon); xpane.add(inputYgridon); pane1.add(xpane,BorderLayout.NORTH);

//\*\*\*\*\*\*\*\*

JPanel lpane=new JPanel(); lpane.setLayout(new GridLayout(3,2)); lpane.add(promptLabel); lpane.add(inputLabel); lpane.add(promptXLabel); lpane.add(inputXLabel); lpane.add(promptYLabel); lpane.add(inputYLabel); pane1.add(lpane,BorderLayout.SOUTH);

//\*\*\*\*\*\*\*\*\* inputXmin.addActionListener(this); inputXmax.addActionListener(this); inputYmin.addActionListener(this); inputYmax.addActionListener(this);

inputLabel.addActionListener(this); inputXLabel.addActionListener(this); inputYLabel.addActionListener(this); inputXntic.addActionListener(this); inputYntic.addActionListener(this); inputXgridon.addItemListener(this); inputYgridon.addItemListener(this); jta=new Plot2D();

inputXmin.setText(Double.toString(jta.p1.xmin)); inputXmax.setText(Double.toString(jta.p1.xmax)); inputYmin.setText(Double.toString(jta.p1.ymin)); inputYmax.setText(Double.toString(jta.p1.ymax)); inputXntic.setText(Integer.toString(jta.p1.xntic)); inputYntic.setText(Integer.toString(jta.p1.yntic)); inputLabel.setText(jta.p1.label); inputXLabel.setText(jta.p1.xlabel); inputYLabel.setText(jta.p1.ylabel); printButton=new JButton("Yazdir");

JPanel pane2 = new JPanel(); pane2.setLayout(new BorderLayout()); pane2.add(jta); pane1.add(printButton,BorderLayout.SOUTH); printButton.addActionListener(this); tabbedPane.addTab(PLOTPANEL, pane2); tabbedPane.addTab(KONTROLPANEL, pane1);

contentPane.add( tabbedPane, BorderLayout.CENTER);

}

public void itemStateChanged(ItemEvent e)

{

inputXmin.setText(Double.toString(jta.p1.xmin)); inputXmax.setText(Double.toString(jta.p1.xmax)); inputYmin.setText(Double.toString(jta.p1.ymin)); inputYmax.setText(Double.toString(jta.p1.ymax)); inputXntic.setText(Integer.toString(jta.p1.xntic)); inputYntic.setText(Integer.toString(jta.p1.yntic)); Object source=e.getItemSelectable(); if(source==inputXgridon)

{

if (e.getStateChange() == ItemEvent.DESELECTED)

{

jta.p1.xgridon=0;

}

else

{

jta.p1.xgridon=1;

}

}

else if(source==inputYgridon)

{

if (e.getStateChange() == ItemEvent.DESELECTED)

{

jta.p1.ygridon=0;

}

else

{

jta.p1.ygridon=1;

}

}

inputLabel.setText(jta.p1.label); inputXLabel.setText(jta.p1.xlabel); inputYLabel.setText(jta.p1.ylabel); jta.yenidenciz();

}

public void actionPerformed(ActionEvent e)

{

if (e.getSource()==printButton )

{

jta.yazdir();

}

Double valXmin=new Double(inputXmin.getText()); jta.p1.xmin=valXmin.doubleValue();

Double valXmax=new Double(inputXmax.getText()); jta.p1.xmax=valXmax.doubleValue();

Double valYmin=new Double(inputYmin.getText()); jta.p1.ymin=valYmin.doubleValue();

Double valYmax=new Double(inputYmax.getText());

//\*\*\*\*\*

Integer valXntic=new Integer(inputXntic.getText()); jta.p1.xntic=valXntic.intValue();

Integer valYntic=new Integer(inputYntic.getText()); jta.p1.yntic=valYntic.intValue();

//\*\*\*\*\* jta.p1.ymax=valYmax.doubleValue(); jta.p1.label=inputLabel.getText(); jta.p1.xlabel=inputXLabel.getText(); jta.p1.ylabel=inputYLabel.getText(); jta.yenidenciz();

}

public static void main(String[] args)

{

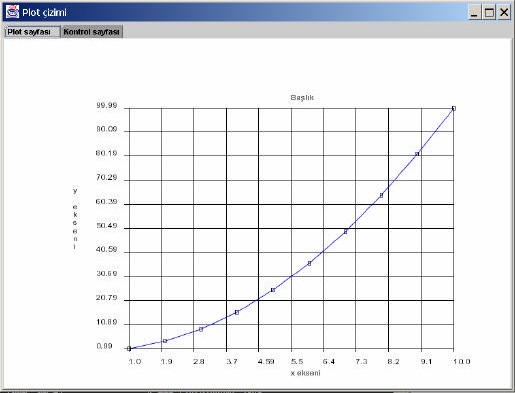
PlotWindowSWF\_2000 pencere= new PlotWindowSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(600,600);

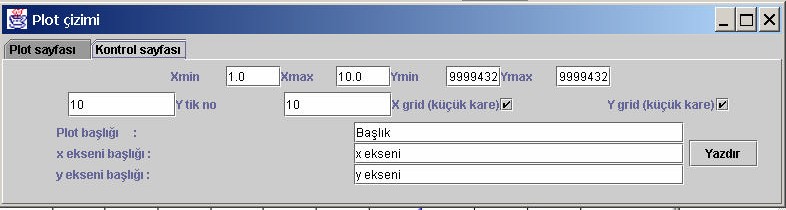
pencere.setVisible(true);

}

}

Asil Plot’i çizen bu programdadir.





### JAVADA HATA ANALIZI VE YAKALANMASI

### HATA ANALIZI

Javada diger dillerden farkli olarak dil yapisinin içinde hatalari yakalamak için oldukça gelismis bir sistem mevcuttur. Eger javada kontrol edilmemis bir hata olusursa program durur ve hatayi yazar. Biz hata olustugunda hata kontrol sisteminde hatayi kontrol edip programi çalistirmaya devam edebiliriz. Javadaki hata kontrol blogu su sekilde yazilir :

### try { java deyimleri;

**java kaynaklarindan herhangi birini kullanan deyimler**

**}**

**catch( herhangibirtürException e1)**

**{ exception kontrol deyimleri }**

**catch( digerherhangibirtürException e1)**

**{ exception kontrol deyimleri } finally**

**{**

**deyimler**

**java kaynagini serbest birakma deyimleri**

**}**

Javadaki Exception türleri asagidaki tabloda verilmistir :

### Tablo 8.1 : Java hata ve exception listesi Java.lang hata listesi

AbstractMethodError ClassCircularityError ClassFormatError Error

ExceptionInInitializerError IllegalAccessError IncompatibleClassChangeError InstantiationError

InternalError LinkageError NoClassDefFoundError NoSuchFieldError NoSuchMethodError OutOfMemoryError StackOverflowError ThreadDeath UnknownError UnsatisfiedLinkError VerifyError VirtualMachineError

### Java.lang Exception listesi

ArithmeticException ArrayIndexOutOfBoundsException ArrayStoreException ClassCastException ClassNotFoundException CloneNotSupportedException Exception

IllegalAccessException IllegalArgumentException IllegalMonitorStateException IllegalStateException IllegalThreadStateException IndexOutOfBoundsException InstantiationException InterruptedException NegativeArraySizeException NoSuchFieldException NoSuchMethodException NullPointerException NumberFormatException RuntimeException SecurityException StringIndexOutOfBoundsException

### Java.util exception listesi

EmptyStackException MissingResourceException NoSuchElementException TooManyListenersException

### Java.io exception listesi

CharConversionException EOFException FileNotFoundException IOException InterruptedIOException InvalidClassException InvalidObjectException NotActiveException NotSerializableException ObjectStreamException OptionalDataException StreamCorruptedException SyncFailedException UTFDataFormatException UnsupportedEncodingException WriteAbortedException

Java hata olustugununda hatanin olustugu metotun girisinde kullanilan **throws** kelimesini takip eden Exception sinifiyla ayni exception gurubunu yasiyan **try-catch** gurubuna aktarir.

Bu metotun içinde gerekli noktada (hatanin olusacagi sartda) **throw** deyimi kullanilarak hata olustugu aktarilir ve bu aktarim **try-catch** gurubu tarafindan yakalanir ve isleme alinir.

Hata kontrolunu daha iyi açiklayabilmek amaciyla bir örnek vermek istiyoruz. SifiraBölünmeException sinifinda yeni bir hata kontrol sinifi yaratiyoruz. Bu sinif ve java kütüphanesinde mevcut olan tamsayi format hatasi (NumberFormatException) sifirebolmetesti programinda test edilmistir. Sonuçlar applet çiktilarinda görülmektedir. Eger bu hatalar kontrol edilmeseydi, java programi hata vererek çalismasini durdururdu.

Programdaki hata kontrol deyimleri kalin harfle verilmistir. Hata kontrol blogu ise italik olarak verilmistir.

### Problem 8.1 sifiraBolunmeException.java programi, sisira bolme hatasi tanimlar

public class sifiraBolunmeException extends ArithmeticException

{

public sifiraBolunmeException()

{

super("sifira bolmeye calistiniz ");

}

}

### Problem 8.2 Bizim tarafimizdan olusturulan sifiraBolunmeException.java hata kontrolü ve sayi formati hata kontrolü (NumberFormatException) yapan sifirabolmetesti.java programi

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*;

import java.text.DecimalFormat; import sifiraBolunmeException;

public class sifirebolmetesti extends Applet implements ActionListener

{

private Label L1,L2; private TextField T1,T2; private int sayi1,sayi2; private double sonuc; public void init()

{

sayi1=0; sayi2=1;

L1=new Label("Payi gir : "); L2=new Label("Paydayi gir : "); T1=new TextField(10); T2=new TextField(10); T2.addActionListener(this); add(L1);

add(T1);

add(L2);

add(T2);

}

public double bol(int s1,int s2) **throws sifiraBolunmeException**

{

*if(s2==0)* **throw new sifiraBolunmeException();**

return (double) s1/s2;

}

public void actionPerformed(ActionEvent e)

{

DecimalFormat hassaslik3=new DecimalFormat("#.000");

**try**{ *sayi1=Integer.parseInt(T1.getText()); sayi2=Integer.parseInt(T2.getText()); T1.setText("");*

*T2.setText(""); sonuc=bol(sayi1,sayi2);*

*showStatus(sayi1+" / "+sayi2+" = "+hassaslik3.format(sonuc));*

}

### catch(NumberFormatException nfe)

*{showStatus("iki tam sayi girmelisiniz.");}*

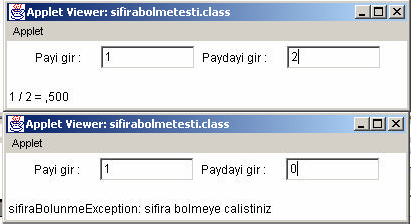
### catch(sifiraBolunmeException sbh)

**{showStatus(sbh.toString());}**

}

}

### 08001.JPG, 08002.JPG,08003.JPG



**Sekil 8.1-3 sifirabolmetesti.html appletinde sifira bolme ve tamsayi hata mesajlarinin görünümü**

Programlarda da görüldügü gibi hata programin calismasina engel olmaksizin kullaniciya bildirilmektedir. Bu program gurubunda once ArithmeticException sinifindan sifiraBolunmeException sinifi türetilmistir. Sonra programin içindeki

public double bol(int s1,int s2) **throws sifiraBolunmeException**

{

*if(s2==0)* **throw new sifiraBolunmeException();**

return (double) s1/s2;

}

metotunda metotun once throws sifiraBolunmeException deyimiyle exception gonderecegi bildirilmis, ve metotun içinde

*if(s2==0)* ***throw new sifiraBolunmeException();***

deyimiyle exception ( hata degiskeni) gönderilmistir. Gönderilen hata degiskeni (exception)

**try{ *sayi1=Integer.parseInt(T1.getText()); sayi2=Integer.parseInt(T2.getText()); T1.setText("");***

***T2.setText(""); sonuc=bol(sayi1,sayi2);***

***showStatus(sayi1+" / "+sayi2+" = "+hassaslik3.format(sonuc));***

**}**

**catch(NumberFormatException nfe)**

***{showStatus("iki tam sayi girmelisiniz.");}***

**catch(sifiraBolunmeException sbh)**

**{showStatus(sbh.toString());}**

**}**

**}**

gurubu tarafindan yakalanmis, ve eger sifir hatasi varsa

**catch(sifiraBolunmeException sbh)** deyimi tarafindan yakalanmis ve

showStatus("iki tam sayi girmelisiniz."); islemini çagirmisti bu islem sonunda applet sifiraBolunmeException sinifinin toString metotunu kullanarak

### sifiraBolunmeException:sifira bolmeye calistiniz

mesajini vermistir. Ayrica yukardaki program parçaciginda da görüldügü gibi programimiz ayni zamanda **NumberFormatException nfe** terimiyle integer olan rakamimizin integer olarak gönderilip gönderilmedigini kontrol etmektedir. Bu exception java kütüphanesindeki **parsInt** metotu tarafindan gönderilmektedir.

### ALISTIRMALAR

1. H6O1java Hata (Exception) kontrolu programi. gercekSayiException ve bayagikesirException tanimlanmistir.

gercekSayiException.java programininda gercek sayi girildiginde hata(Exception) vermek için gerceksayiException tanimlanmistir. bayagikesir1.java programindaki

public bayagikesir(double npay,double npayda) throws gercekSayiException

{

//\*\*\* tanimlayiniz

}

kurucu metotunu bu exception'i kullanacak sekilde olusturunuz. H5O1.java programinin benzeri H6O1.java programinda bayagikesir yerine bayagikesir1 kullanarak exception yapinizi kontrol ediniz.

### Problem 8.3 : gercekSayiException, gercek sayi hata tanim programi

public class gercekSayiException extends NumberFormatException

{

public gercekSayiException()

{

super("gercek sayi girdiniz tam sayi girmelisiniz");

}

}

### Problem 8.4 : bayagikesirException, bayagi kesir sifira bölme hata tanim programi

class bayagikesirException extends Exception

{

public bayagikesirException()

{

super("Bayagi kesir paydasinin degeri 0.");

}

}

### Problem 8.5 : bayagikesir1.java, hata kontrollü bayagikesir sinifi

//===========================================

// bayagikesir sinifi tanimi

// Dr. Turhan Coban

//===========================================

import java.io.\*;

import bayagikesirException; import gercekSayiException;

// sinif bayagikesir

// bu sinif bayagikesir sayilarin matematik

// islemlerini tanimlar

//

class bayagikesir {

// sinif degiskenleri int pay;

int payda;

// kurucu metodlar

public bayagikesir()

{

pay=0; payda=1;

}

public bayagikesir(int npay,int npayda)

{

pay=npay; payda=npayda;

}

public bayagikesir(double npay,double npayda) throws gercekSayiException

{

//\*\*\* tanimlayiniz

double tmp1,tmp2; tmp1=(int)npay; tmp2=(int)npayda;

if (tmp1!=npay) throw new gercekSayiException();

else if (tmp2!=npayda) throw new gercekSayiException(); else

{

pay=(int)tmp1; payda=(int)tmp2;

}

}

public bayagikesir(int numer)

{

pay=numer; payda=1;

}

public bayagikesir(bayagikesir c )

{

pay=c.Pay(); payda=c.Payda();

}

// giriŸ - ‡• k• Ÿ metotlar•

public int enbuyukortakbolen()

{

int n=pay; int m=payda;

// iki tam sayinin en buyuk ortak bolenini hesaplar if(n==0)

return m; if(m==0)

return n; while(m != n)

{

if(n>m) n=n-m;

else

m=m-n;

}

return n;

}

public void sadelestir() throws bayagikesirException

{

//sadelestir int isaret=1; if(pay<0)

{

isaret=-isaret; pay=-pay;

}

if(payda<0)

{

isaret=-isaret; payda=-payda;

}

if(payda==0) {throw new bayagikesirException();} int ebob=enbuyukortakbolen(); ebob=Math.abs(ebob);

pay=isaret\*pay/ebob; payda=payda/ebob;

}

public int Pay()

{

return pay;

}

public int Payda()

{

return payda;

}

public void payGir(int r)

{

pay=r;

}

public void paydaGir(int i)

{

payda=i;

}

public void bayagikesirGir(bayagikesir sag)

{

pay=sag.Pay(); payda=sag.Payda();

}

public void bayagikesirGir(int nr,int ni)

{

pay=nr; payda=ni;

}

public void bayagikesirGir(int nr)

{

pay=nr;

payda=1;

}

public void bayagikesirGir(double d) throws bayagikesirException

{

// tam sayinin bayagikesir esitini y• kle

// eh birazc• k yaklas• m var tabi bayagikesirGir((int)d\*10000,10000); sadelestir();

}

public double toDouble()

{

//bayagikesir sayinin gercek sayi esidi return ((double)pay/(double)payda);

}

public static bayagikesir BayagikesireCevir(double d) throws bayagikesirException

{

// tam sayinin bayagikesir esiti

// eh birazc• k yaklas• m var tabi bayagikesir b=new bayagikesir(); b.bayagikesirGir((int)d\*10000,10000); b.sadelestir();

return b;

}

public void topla(bayagikesir sag) throws bayagikesirException

{

//ikinci bir bayagikesirle topla

pay = pay\*sag.Payda() + sag.Pay()\*payda; payda = payda\*sag.Payda();

sadelestir();

}

public void topla(int sag) throws bayagikesirException

{

//bir gercek say• yla topla pay = pay + sag\*payda; sadelestir();

}

public void cikar(bayagikesir sag) throws bayagikesirException

{

//ikinci bir bayagikesirle topla

pay = pay\*sag.Payda() - sag.Pay()\*payda; payda = payda\*sag.Payda();

sadelestir();

}

public void cikar(int sag) throws bayagikesirException

{

//bir gercek say• yla topla pay = pay - sag\*payda; sadelestir();

}

public void carp(bayagikesir sag ) throws bayagikesirException

{

//bir bayagikesir sayiyla carp pay = pay\*sag.Pay();

payda = payda\*sag.Payda(); sadelestir();

}

public void carp(int sag ) throws bayagikesirException

{

//bir gercek sayiyla carp pay = pay\*sag; sadelestir();

}

public void bol(bayagikesir sag ) throws bayagikesirException

{

//bir bayagikesir say• yla bol pay = pay\*sag.Payda(); payda = payda\*sag.Pay(); sadelestir();

}

public void bol(int sag ) throws bayagikesirException

{

//bir Pay say• yla bol payda = payda\*sag; sadelestir();

}

public static bayagikesir topla(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{

// iki bayagikesir say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1= sol.Pay() \* sag.Payda() + sag.Pay() \* sol.Payda(); int i1= sol.Payda()\* sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir topla(bayagikesir sol, int sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1=sol.Pay() + sag\*sol.Payda(); int i1=sol.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir topla(int sol, bayagikesir sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1=sag.Pay() + sol\*sag.Payda(); int i1=sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1);

sonuc.sadelestir(); return sonuc;

}

//===================================

public static bayagikesir cikar(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{

// iki bayagikesir say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1= sol.Pay() \* sag.Payda() - sag.Pay() \* sol.Payda(); int i1= sol.Payda()\* sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir cikar(bayagikesir sol, int sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1=sol.Pay() - sag\*sol.Payda(); int i1=sol.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir cikar(int sol, bayagikesir sag) throws bayagikesirException

{

// bir bayagikesir ve bir Pay say• n• n toplam• n• return deyimiyle bayagikesir olarak

// aktar• r

int r1=sag.Pay() - sol\*sag.Payda(); int i1=sag.Payda();

bayagikesir sonuc; sonuc=new bayagikesir(r1,i1); sonuc.sadelestir();

return sonuc;

}

//===================================

public static bayagikesir carp(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{ // iki bayagikesir sayinin carpimini aktar bayagikesir sonuc;

sonuc=new bayagikesir(sol.Pay()\*sag.Pay(),sol.Payda()\*sag.Payda()); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir carp(bayagikesir sol, int sag) throws bayagikesirException

{ // bir bayagikesir ve bir double sayinin carpimini aktar bayagikesir sonuc;

sonuc=new bayagikesir(sol.Pay()\*sag,sol.Payda()); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir carp(int sol, bayagikesir sag) throws bayagikesirException

{ // bir bayagikesir ve bir double sayinin carpimini aktar bayagikesir sonuc;

sonuc=new bayagikesir(sag.Pay()\*sol,sag.Payda()); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir bol(bayagikesir sol, bayagikesir sag) throws bayagikesirException

{ // iki bayagikesir sayinin b”l• mlerini aktar

double a=sag.Pay()\*sag.Pay()+sag.Payda()\*sag.Payda(); bayagikesir sonuc;

sonuc=new bayagikesir((int)(sol.Pay()\*sag.Payda()),(int)(sol.Payda()\*sag.Pay())); sonuc.sadelestir();

return sonuc;

}

public static bayagikesir bol(bayagikesir sol, int sag) throws bayagikesirException

{ // bayagikesir say• y• double say• ya bol bayagikesir sonuc;

sonuc=new bayagikesir(sol.Pay(),(sol.Payda()\*sag)); sonuc.sadelestir();

return sonuc;

}

public boolean kucuktur(bayagikesir sol,bayagikesir sag)

{

// less then comparison of two bayagikesir numbers return (sol.toDouble() < sag.toDouble());

}

public boolean kucuktur\_esittir(bayagikesir sol,bayagikesir sag)

{

// less then and esittir comparison of two bayagikesir numbers return (sol.toDouble() <= sag.toDouble());

}

public boolean buyuktur(bayagikesir sol,bayagikesir sag)

{

// buyuktur then comparison of two bayagikesir numbers return sol.toDouble() > sag.toDouble();

}

public boolean buyuktur\_esittir(bayagikesir sol,bayagikesir sag)

{

// buyuktur then and esittir comparison of two bayagikesir numbers return sol.toDouble() >= sag.toDouble();

}

public boolean esittir(bayagikesir sol,bayagikesir sag)

{

// esittir comparison of two bayagikesir numbers return sol.toDouble() == sag.toDouble();

}

public boolean esit\_degildir(bayagikesir sol,bayagikesir sag)

{

// not esittir comparison of two bayagikesir numbers return sol.toDouble() != sag.toDouble();

}

public static String toString(bayagikesir value)

{

String b=""; if(Math.abs(value.Payda())!=1)

{

b=b+"( "+value.Pay()+" / "+value.Payda()+" )";

}

else

{

b=b+value.Pay()+" ";

}

return b;

}

public String toString()

{

// yazima haz• r bayagikesir formda String de§iskeni iletir. String b="";

if(Math.abs(Payda())!=1)

{

b=b+"( "+pay+" / "+Payda()+" )";

}

else

{

b=b+Pay()+" ";

}

return b;

}

};

//bayagikesir sinifinin taniminin sonu

### Problem 8.6 : bayagikesirException, bayagi kesir sifira bölme hata test programi H6O1.java

/bu sinif sinif degiskeni kompleksi boyutlu olarak çcagirir.

//not buradaki bayagikesir sinifi bayagikesir1.java programi

//tarafindan yaratilmistir.

import java.io.\*; //java girdi cikti sinifini cagir import bayagikesir;

class H6O1

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); double n1,n2;

System.out.print("toplam bayagi kesir sayisi, n = "); int n=cin.readInt();

bayagikesir sayi[]=new bayagikesir[n]; bayagikesir toplam=new bayagikesir(); int i;

for(i=0;i<sayi.length;i++)

{

### try{

System.out.print("bir bayagikesir giriniz : "); n1=cin.readDouble();

n2=cin.readDouble(); sayi[i]=new bayagikesir(n1,n2);

}

### catch (gercekSayiException e)

**{**

**System.err.println(e.getMessage()); i--;**

**}**

}

for(i=0;i<sayi.length;i++)

{

try{

toplam.topla(sayi[i]);

}

catch (bayagikesirException e)

{

System.err.println(e.getMessage());

}

}

System.out.println("Toplam = "+toplam.toString());

try{

toplam.bol(sayi.length);

}

catch (bayagikesirException e)

{

System.err.println(e.getMessage());

}

System.out.println("Ortalama = "+toplam.toString());

}

}

1. H8AL1\_2000, Bayagikesir hata kontrolu programi çiktisi, JoptionPane kullaniyor. import javax.swing.JOptionPane;

import java.io.\*; //java girdi cikti sinifini cagir import bayagikesir;

class H8AL1\_2000

{

public static void main(String args[]) throws IOException

{

Text cin=new Text(); double n1,n2;

System.out.print("toplam bayagi kesir sayisi, n = "); int n=cin.readInt();

bayagikesir sayi[]=new bayagikesir[n]; bayagikesir toplam=new bayagikesir(); int i;

String s=""; for(i=0;i<sayi.length;i++)

{

try{

System.out.print("bir bayagikesir giriniz : "); n1=cin.readDouble();

n2=cin.readDouble(); sayi[i]=new bayagikesir(n1,n2);

}

catch (gercekSayiException e)

{

JOptionPane.showMessageDialog(null,e.getMessage(),"Gerçek Sayi Hatasi", JOptionPane.ERROR\_MESSAGE);

i--;

}

catch (bayagikesirException e)

{

JOptionPane.showMessageDialog(null,e.getMessage(),"Bayagi kesir Hatasi", JOptionPane.ERROR\_MESSAGE);

}

}

for(i=0;i<sayi.length;i++)

{

try{

toplam.topla(sayi[i]);

}

catch (bayagikesirException e)

{

JOptionPane.showMessageDialog(null,e.getMessage(),"Bayagi kesir Hatasi", JOptionPane.ERROR\_MESSAGE);

}

}

s+="Ortalama = "+toplam.toString();

try{

toplam.bol(sayi.length);

}

catch (bayagikesirException e)

{

JOptionPane.showMessageDialog(null,e.getMessage(),"Bayagi kesir Hatasi",

JOptionPane.ERROR\_MESSAGE);

}

s+="Ortalama = "+toplam.toString(); JOptionPane.showMessageDialog(null,s,

"Bayagi kesir Ortalamasi",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 9. PARALEL KULLANIM (MULTITHREADING) , GERÇEK ZAMAN UYGULAMALARI, ANIMASYON

### PARALEL KULLANIM(MULTITHREADING) VE GERÇEK ZAMAN PROGRAMLANMASI

Islemleri tek tek sirayla ve iyi bir sekilde yapabilseydik belki de çok mükemmel sonuçlar alabilirdik. Fakat Dünyadaki hemen hemen hiçbirsey tek tek sirayla çalismaz. Insan vücudu da dahil bir çok dogal sistem paralel olarak çalisir. Yani bir islem yapilirken parelelinde diger bir islem yapilmaya devam eder. Bizim su anda kullandigimiz bilgisayarlarin büyük bir bölümü tek bir islemciye(processor) sahiptir. Birden fazla islemciye sahip paralel bilgisayarlarda, her bir islemcide bir is tanimlayarak paralel kullanim mümkündür. Fakat tek islemcili sistemler için(seri bilgisayar) yazilmis olan program dilleri genel olarak birden fazla islemin ayni anda yapilmasina izin vermezler. Modern bilgisayar dilleri arasinda sadece **Ada** ve **Java** dilleri bir islemcide birden fazla islemin (programin) ayni anda kullanilmasina izin verir. C++ dili birden fazla programin ayni anda kullanilmasina izin vermez. Bu yüzden tek kullanimli bir programlama dilidir.

Parelel Kullanim(Multithreading) Tek bir islemcinin birden fazla program tarafindan kullanilmasidir. Bu kullanim CPU (merkez program islemcisi) adini verdigimiz temel bilgisayar islem ünitesinin çesitli programlar tarafindan ortaklasa kullanilmasi prensibine göre yapilir. Bilgisayar CPU zamani küçük parçalara bölünerek her parçada ayri bir bilgisayar programinin çalismasi ve diger bilgisayar programlarinin beklemesi saglanir. Programlar küçük paketçikler halinde program islemcisine gönderilir ve sonuçlar tekrar hafizaya aktarilir. Sonra ikinci bir program parçacigi islemci bellegine gönderilir. Modern seri bilgisayarlarda cache adi verilen bu program parçaciklari çalisirken diger program parçaciklarini tutabilen hizli hafiza birimleri de bulunur.Tüm bu kontrollari ve birden fazla programin tek islemcide çalisma islemini java dilinde dogrudan programlayabiliriz.

CPU nun Parelel kullanimi Runnable interface’i veya abstract Thread sinifini kullanarak gerçeklestirilir. Asagida Bu iki yolu kullanarak bir digital saat programi olusturulmustur. Bu programlarda saat sadece tek bir program olmasina ragmen bilgisayarin belleginde eger devamli çalistirilacak olursa baska hiçbir programin çalistirilamiyacagi düsünülerek paralel kullanimli olarak verilmistir.

Zaman içinde programlari bilgisayarda kontrollu yapida kullanmak ayni zamanda bize gerçek zaman programlama olasiliklarini da sunar. Paralel ve gerçek zama kullanimi javayi kuvvetli bir modern programlama dili yapan en önemli özelliklerden biridir.

### Program 9.1 Saat.java sinifinin thread sinifindan türemis olarak yazilimi.

import java.awt.Graphics; import java.awt.Font; import java.util.\*;

import java.applet.Applet; import java.text.\*;

class saatThread **extends Thread**

{

Applet a;

public saatThread(Applet ai)

{a=ai;}

### public void run()

**{**

**while(true)**

**{**

**a.repaint();**

**try {Thread.sleep(1000);} catch(InterruptedException e)**

**{System.err.println(e.toString());}**

**}**

**}**

}

public class saat extends Applet

{

Font F;

Date D; DateFormat df; Locale tr; **saatThread r;** public void init()

{

F=new Font("TimesRoman",Font.BOLD,24); tr=new Locale("tr","TR");

D=new Date();

df= DateFormat.getDateInstance(DateFormat.LONG,tr);

### r=new saatThread(this); r.start();

}

public void paint(Graphics g)

{

df.format(D); D=new Date(); g.setFont(F); try

{

g.drawString(D.toString(),10,50);

}

catch(NullPointerException e){}

}

}

### Program 9.2 SaatSWF.java sinifinin thread sinifindan türemis olarak yazilimi.

import java.awt.Graphics; import java.awt.\*; import java.util.\*;

import javax.swing.\*; import java.text.\*;

class saatThreadSWF extends Thread

{

JFrame a;

public saatThreadSWF(JFrame ai)

{a=ai;}

public void run()

{

while(true)

{

a.validate();

a.repaint();

try {Thread.sleep(1000);} catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

}

public class saatSWF extends JFrame

{

JLabel jb;

Font F;

Date D;

saatThreadSWF r;

GregorianCalendar takvim;

public saatSWF()

{

super("Digital Saat extends Thread"); Container cc=getContentPane(); jb=new JLabel("");

F=new Font("TimesRoman",Font.BOLD,24); jb.setFont(F);

// Türkiye standart (Greenwitch zamanina göre iki saat ileride) String[] ids = TimeZone.getAvailableIDs(+2 \* 60 \* 60 \* 1000);

SimpleTimeZone pdt = new SimpleTimeZone(+2 \* 60 \* 60 \* 1000, ids[0]); takvim=new GregorianCalendar(pdt);

D=new Date(); takvim.setTime(D);

r=new saatThreadSWF(this); cc.add(jb);

r.start();

}

public void paint(Graphics g)

{

D=new Date(); takvim.setTime(D); String s=" ["+

takvim.get(Calendar.DATE) +" / "+ takvim.get(Calendar.MONTH)+" / "+ takvim.get(Calendar.YEAR) +" ] "+ takvim.get(Calendar.HOUR\_OF\_DAY)+" : "+ takvim.get(Calendar.MINUTE)+" : "+ takvim.get(Calendar.SECOND);

jb.setText(s);

}

public static void main(String[] args)

{

saatSWF pencere= new saatSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,100);

pencere.setVisible(true);

}

}

### 09001.JPG



**Sekil 9.1 saaatSWF.java digital saat JFRame çiktisi**

**Program 9.3 Saat.java sinifinin Runnable interface’i ile yazilimi ve applet çiktisi**

import java.awt.Graphics; import java.awt.Font; import java.util.Date; import java.applet.Applet;

public class saat extends Applet **implements Runnable**{ Font theFont=new Font("TimesRoman",Font.BOLD,24);

Date theDate; **Thread runner; public void start()**

### {

**if(runner==null)**

**{**

**runner=new Thread(this); runner.start();**

**}**

**}**

**public void stop()**

**{**

**if(runner!=null)**

**{**

**runner.stop(); runner=null;**

**}**

**}**

**public void run()**

**{**

**while(true)**

**{**

**theDate=new Date(); repaint();**

**try {Thread.sleep(1000);} catch(InterruptedException e) {}**

**}**

**}**

public void paint(Graphics g)

{

g.setFont(theFont); try

{

g.drawString(theDate.toString(),10,50);

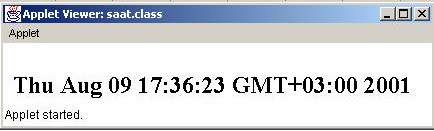
}

catch(NullPointerException e){}

}

}

### 09002.JPG



**Sekil 9.2 saat.html appletinde saat theradinin isletilmesi**

**Program 9.4 saatRunnableSWF sinifinin Runnable interface’i ile yazilimi**

import java.awt.Graphics; import java.awt.\*; import java.util.\*;

import javax.swing.\*; import java.text.\*;

public class saatRunnableSWF extends JFrame implements Runnable

{

JLabel jb;

Font F;

Date D;

saatThreadSWF r; GregorianCalendar takvim; Thread runner;

public saatRunnableSWF()

{

super("Digital Saat Runnable implement Thread"); Container cc=getContentPane();

jb=new JLabel("");

F=new Font("TimesRoman",Font.BOLD,24); jb.setFont(F);

// Türkiye standart (Greenwitch zamanina göre iki saat ileride) String[] ids = TimeZone.getAvailableIDs(+2 \* 60 \* 60 \* 1000);

SimpleTimeZone pdt = new SimpleTimeZone(+2 \* 60 \* 60 \* 1000, ids[0]); takvim=new GregorianCalendar(pdt);

D=new Date(); takvim.setTime(D); setBackground(renk.beyaz); r=new saatThreadSWF(this); cc.add(jb);

r.start();

}

public void start()

{

if(runner==null)

{

runner=new Thread(this); runner.start();

}

}

public void run()

{

while(true)

{

repaint();

try {Thread.sleep(1000);} catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

public void paint(Graphics g)

{

D=new Date(); takvim.setTime(D); String s=" ["+

takvim.get(Calendar.DATE) +" / "+ takvim.get(Calendar.MONTH)+" / "+ takvim.get(Calendar.YEAR) +" ] "+ takvim.get(Calendar.HOUR\_OF\_DAY)+" : "+ takvim.get(Calendar.MINUTE)+" : "+

takvim.get(Calendar.SECOND); jb.setText(s);

}

public static void main(String[] args)

{

saatRunnableSWF pencere= new saatRunnableSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,100);

pencere.setVisible(true);

}

}

Programlardan da görüldügü gibi thread adi verilen ve bilgisayar belleginde diger programlarla beraber çalismasini istedigimiz program parçaciklarini java programinin içinde baslatabiliriz(start), kontrol edebiliriz veya durdurabiliriz(stop). Program parçaciginin çalismadan bekleyecegi sureyi de sleep metotunu kullanarak ayarliyabiliriz.

### PARALEL PROGRAM KULLANIMIDA HAFIZA SINKRONIZASYONU

Ikinci olarak biraz daha karisik bir örnege göz atalim. Bu örnekte, iki tane program bilgisayar belleginde ayni anda çalisacaktir. Birinci program ikinci programla ortak olarak kullanilan bir bilgisayar int hafiza noktasina sirasiyla birden ona kadar degisen bir rakam yazacak, ikinci program da ayni noktadan (OrtakHucreden) yazilan degeri okuyacaktir.

### Program 9.5 OrtakHucre.java programi

class OrtakHucre{ public static void main(String args[])

{

### TutInteger h=new TutInteger(); UretInteger p=new UretInteger(h); KullanInteger c= new KullanInteger(h); p.start();

**c.start();** }

}

class UretInteger **extends Thread**

{

private TutInteger pTut;

public UretInteger(TutInteger h)

{

pTut=h;

}

public void **run()**

{

for(int count=0; count<10; count++)

{

pTut.setOrtakInt(count);

System.out.println("OrtakInteger "+count+" degerini uretti");

### try{

**sleep( (int) (Math.random()\*5000 ) );**

**}**

**catch(InterruptedException e)**

**{**

**System.err.println("Exeption "+e.toString());**

**}**

}

}

}

class KullanInteger **extends Thread**

{

private TutInteger cTut;

public KullanInteger(TutInteger h)

{

cTut=h;

}

public void **run()**

{

int Intdegeri=0; while (Intdegeri !=9)

{

### try{

**sleep( (int) (Math.random()\*5000));**

**}**

**catch(InterruptedException e)**

**{**

**System.err.println("Exeption "+e.toString());**

**}**

Intdegeri=cTut.getOrtakInt();

System.out.println("Kullan Integer"+Intdegeri+" degerini okudu");

}

}}

class TutInteger{ private int OrtakInt; public void setOrtakInt(int Intdegeri)

{

OrtakInt=Intdegeri;

}

public int getOrtakInt()

{

return OrtakInt;

}

}

program sonucundaki çikti :

OrtakInteger 0 degerini uretti OrtakInteger 1 degerini uretti OrtakInteger 2 degerini uretti Kullan Integer2 degerini okudu OrtakInteger 3 degerini uretti OrtakInteger 4 degerini uretti OrtakInteger 5 degerini uretti Kullan Integer5 degerini okudu Kullan Integer5 degerini okudu OrtakInteger 6 degerini uretti OrtakInteger 7 degerini uretti Kullan Integer7 degerini okudu OrtakInteger 8 degerini uretti OrtakInteger 9 degerini uretti Kullan Integer9 degerini okudu

### Program 9.6 Runnable interface’i ile çalisan OrtakHucreR.java sinifi

import java.util.\*; import java.awt.\*;

import java.io.\*;//runnable interfacei kullanarak

class UretInteger **implements Runnable**

{

private TutInteger pTut; private Thread thread;

public UretInteger(TutInteger h)

{

pTut=h;

### thread = new Thread(this); thread.start();

}

public void **run()** {

for(int count=0; count < 10; count++) { pTut.setOrtakInt(count);

System.out.println("Ortak Integer " + count+" Olarak olusturuldu");

### try { thread.sleep((int)(Math.random()\*3000));

**}**

**catch(InterruptedException exception) { System.err.println("Exception " + exception.toString());**

**}**

}

}

}

class KullanInteger **implements Runnable**

{

private TutInteger cTut; private Thread thread;

public KullanInteger(TutInteger h) { cTut = h;

### thread= new Thread(this); thread.start();

}

public void **run()** { int Intdegeri=0;

while(Intdegeri != 9) {

### try { thread.sleep((int)(Math.random()\*3000));

**}**

**catch(InterruptedException e) { System.err.println("Exception " + e.toString());**

**}**

Intdegeri = cTut.getOrtakInt();

System.out.println("Kullan Integer " + Intdegeri+"degerini yakaladi");

}

}

}

class TutInteger { private int OrtakInt;

public void setOrtakInt(int Intdegeri) {OrtakInt = Intdegeri;} public int getOrtakInt() { return OrtakInt;}

}

public class OrtakHucreR {

public static void main(String[] args) { TutInteger h= new TutInteger();

new UretInteger(h); new KullanInteger(h);

}

}

### Program 9.7 Buffer ile uyum saglanmis OrtakHucreBuf.java sinifi

import java.awt.\*;

import java.applet.Applet; import java.text.DecimalFormat;

public class OrtakHucreBuf extends Applet

{

private TextArea cikti; public void init()

{

setLayout(new BorderLayout()); cikti = new TextArea(); add(cikti,BorderLayout.CENTER);

}

public void start()

{

Tutinteger h=new Tutinteger(cikti); Uretinteger p=new Uretinteger(h); Consumeinteger c= new Consumeinteger(h); p.start();

c.start();

}

}

class Uretinteger extends Thread

{

private Tutinteger pTut; private TextArea cikti;

public Uretinteger(Tutinteger h)

{

pTut=h;

}

public void run()

{

for(int count=0; count<10; count++)

{

pTut.setOrtakint(count); try{

sleep( (int) (Math.random()\*500 ) );

}

catch(InterruptedException e)

{

System.err.println("Hata : "+e.toString());

}

}

}

}

class Consumeinteger extends Thread

{

private Tutinteger cTut;

public Consumeinteger(Tutinteger h)

{

cTut=h;

}

public void run()

{

int intdegeri; intdegeri=cTut.getOrtakint(); while (intdegeri !=9)

{

try{

sleep( (int) (Math.random()\*500));

}

catch(InterruptedException e)

{

System.err.println("Hata "+e.toString());

}

intdegeri=cTut.getOrtakint();

}

}

}

class Tutinteger

{

private int Ortakint[] = {9,9,9,9,9}; private boolean writeable = true; private boolean okunabilir = false; private int YerelOku =0, YerelYaz=0; private TextArea cikti;

public Tutinteger(TextArea out)

{

cikti = out;

}

public synchronized void setOrtakint(int intdegeri)

{

while (!writeable)

{

try{

cikti.append(" Beklemede "+intdegeri); wait();

}

catch (InterruptedException e)

{

System.err.println("Hata :"+e.toString() );

}

}

Ortakint[YerelYaz]=intdegeri; okunabilir = true;

cikti.append("\n Uretilen "+intdegeri+" Yazilan "+YerelYaz); YerelYaz = ++YerelYaz % 5;

cikti.append("\t yazilan hucre "+YerelYaz+"\t okunan hucre "+YerelOku); printBuffer(cikti, Ortakint);

if(YerelYaz==YerelOku)

{

writeable=false;

cikti.append("\nDEPOLAMA YERLER˜ DOLU ");

}

notify();

}

public synchronized int getOrtakint()

{

int intdegeri;

while (!okunabilir)

{

try{

cikti.append(" KULLANMAK iCiN BEKLiYOR "); wait();

}

catch (InterruptedException e)

{

System.err.println("hata:"+e.toString() );

}

}

writeable=true; intdegeri=Ortakint[YerelOku];

cikti.append("\n Kullanilan "+intdegeri+" Okundugu Hucre "+YerelOku); YerelOku = ++YerelOku % 5;

cikti.append("\t yazilan h• cre "+YerelYaz+"\t okunan hucre "+YerelOku); printBuffer(cikti, Ortakint);

if (YerelOku==YerelYaz)

{okunabilir=false; cikti.append("\niNTEGER DEPOSU BOS");

}

notify();

return intdegeri;

}

public void printBuffer(TextArea out,int buf[])

{

out.append("\t depo: "); for(int i=0; i<buf.length; i++)

out.append(" "+buf[i]);

}

}

### Buffer ile uyum saglanmis OrtakHucreBuf.java sinifi çiktisi Uretilen 0 Yazilan 0 yazilan hucre 1 okunan hucre 0 depo: 0 9 9 9 9

**Kullanilan 0 Okundugu Hucre 0 yazilan hucre 1 okunan hucre 1 depo: 0 9 9 9 9**

**INTEGER DEPOSU BOS**

**Uretilen 1 Yazilan 1 yazilan hucre 2 okunan hucre 1 depo: 0 1 9 9 9**

**Kullanilan 1 Okundugu Hucre 1 yazilan hucre 2 okunan hucre 2 depo: 0 1 9 9 9 INTEGER DEPOSU BOS**

**Uretilen 2 Yazilan 2 yazilan hucre 3 okunan hucre 2 depo: 0 1 2 9 9**

**Uretilen 3 Yazilan 3 yazilan hucre 4 okunan hucre 2 depo: 0 1 2 3 9**

**Kullanilan 2 Okundugu Hucre 2 yazilan hucre 4 okunan hucre 3 depo: 0 1 2 3 9**

**Uretilen 4 Yazilan 4 yazilan hucre 0 okunan hucre 3 depo: 0 1 2 3 4**

**Kullanilan 3 Okundugu Hucre 3 yazilan hucre 0 okunan hucre 4 depo: 0 1 2 3 4**

**Uretilen 5 Yazilan 0 yazilan hucre 1 okunan hucre 4 depo: 5 1 2 3 4**

**Uretilen 6 Yazilan 1 yazilan hucre 2 okunan hucre 4 depo: 5 6 2 3 4**

**Uretilen 7 Yazilan 2 yazilan hucre 3 okunan hucre 4 depo: 5 6 7 3 4**

**Kullanilan 4 Okundugu Hucre 4 yazilan hucre 3 okunan hucre 0 depo: 5 6 7 3 4**

**Kullanilan 5 Okundugu Hucre 0 yazilan hucre 3 okunan hucre 1 depo: 5 6 7 3 4**

**Uretilen 8 Yazilan 3 yazilan hucre 4 okunan hucre 1 depo: 5 6 7 8 4**

**Kullanilan 6 Okundugu Hucre 1 yazilan hucre 4 okunan hucre 2 depo: 5 6 7 8 4**

**Uretilen 9 Yazilan 4 yazilan hucre 0 okunan hucre 2 depo: 5 6 7 8 9**

**Kullanilan 7 Okundugu Hucre 2 yazilan hucre 0 okunan hucre 3 depo: 5 6 7 8 9**

**Kullanilan 8 Okundugu Hucre 3 yazilan hucre 0 okunan hucre 4 depo: 5 6 7 8 9**

**Kullanilan 9 Okundugu Hucre 4 yazilan hucre 0 okunan hucre 0 depo: 5 6 7 8 9 INTEGER DEPOSU BOS**

**Problem 9.8 Uyum saglanmis OrtakHucreSync.java sinifi**

class OrtakHucreSyn

{

public static void main(String args[])

{

TutInteger h=new TutInteger(); UretInteger p=new UretInteger(h); KullanInteger c= new KullanInteger(h); p.start();

c.start();

}

}

class UretInteger extends Thread

{

private TutInteger pTut;

public UretInteger(TutInteger h)

{

pTut=h;

}

public void run()

{

for(int count=0; count<10; count++)

{

pTut.setOrtakInt(count);

System.out.println("UretInteger OrtakInteger degerini"+count+"olarak olusturdu"); try{

sleep( (int) (Math.random()\*3000 ) );

}

catch(InterruptedException e)

{

System.err.println("Exeption "+e.toString());

}

}

}

}

class KullanInteger extends Thread

{

private TutInteger cTut;

public KullanInteger(TutInteger h)

{

cTut=h;

}

public void run()

{

int Intedegeri=0; while (Intedegeri !=9)

{

try{

sleep( (int) (Math.random()\*3000));

}

catch(InterruptedException e)

{

System.err.println("Exeption "+e.toString());

}

Intedegeri=cTut.getOrtakInt(); System.out.println("KullanInteger "+Intedegeri+" yakalad• ");

}

}

}

class TutInteger

{

private int OrtakInt;

private boolean Yazilabilir = true;

public synchronized void setOrtakInt(int Intedegeri)

{

while (!Yazilabilir)

{

try{

wait();

}

catch (InterruptedException e)

{

System.err.println("Exception:"+e.toString() );

}

}

OrtakInt=Intedegeri; Yazilabilir=false; notify();

}

public synchronized int getOrtakInt()

{

while (Yazilabilir)

{

try{

wait();

}

catch (InterruptedException e)

{

System.err.println("Exception:"+e.toString() );

}

}

Yazilabilir=true; notify();

return OrtakInt;

}

}

### Uyum saglanmis OrtakHucreSync.java sinifi sonuçlari

**UretInteger OrtakInteger degerini 0 olarak olusturdu KullanInteger 0 yakaladi**

**UretInteger OrtakInteger degerini 1 olarak olusturdu KullanInteger 1 yakaladi**

**UretInteger OrtakInteger degerini 2 olarak olusturdu KullanInteger 2 yakaladi**

**UretInteger OrtakInteger degerini 3 olarak olusturdu KullanInteger 3 yakaladi**

**UretInteger OrtakInteger degerini 4 olarak olusturdu KullanInteger 4 yakaladi**

**UretInteger OrtakInteger degerini 5 olarak olusturdu KullanInteger 5 yakaladi**

**UretInteger OrtakInteger degerini 6 olarak olusturdu KullanInteger 6 yakaladi**

**UretInteger OrtakInteger degerini 7 olarak olusturdu KullanInteger 7 yakaladi**

**UretInteger OrtakInteger degerini 8 olarak olusturdu KullanInteger 8 yakaladi**

**UretInteger OrtakInteger degerini 9 olarak olusturdu KullanInteger 9 yakaladi**

Programlardan da görülecegi gibi UretInteger sinifi (thread’i) TutInteger sinifini kullanarak yeni bir integer degeri yaratmakta ve KullanInteger sinifi da bu degeri ortak hafizadan okumaktadir. Burada orijinal programlarda iki thread’in çalisma siralarinin sirayla gitmamesi problemi çikmistir. Bazen UretInteger programi birden fazla çalismakta bazende KullanInteger programi birden fazla çalismaktadir. Bunu onlemek için OrtakHucreSync programinda TutInteger sinifinin içinde Uyum kontrolü yapilmistir. boolean yazilabilir degiskenini bir trafik isareti gibi kullanip UretInteger ve TutInteger threat programlarinin sirasiyla isleme girmesi, bu arada digerinin beklemesi saglanmistir. Buradaki zorluk. Bazen bir programin çok fazla

bekleyebilmesi olasiligidir. Bunu engellemek içinde OrtakHucreBuf programinda yazmak için bir yerine bes integer yeri tanimlanmis, ve okutma ve yazma islemlerinde iki ayri kontrol boolean deyimi kullanilmistir.

### ALISTIRMALAR

1. daha önce tanimladigimiz yildiz.java (problem 6.19) programini da kullanarak burada analog bir saat tanimi yapilmistir. Analog saat her saniyede bir çalismakta ve kendini tekrar çizmektedir. Programi inceleyiniz.

### Program 9.9 : analogSaat.java programi, yildizCiz2Thread be analogSaat siniflari programi

import java.applet.Applet; import java.util.\*;

import java.awt.\*; import yildiz; import polar;

import java.awt.event.\*;

class yildizCiz2Thread extends Thread

{

Applet a;

public yildizCiz2Thread(Applet ai)

{

a=ai;

}

public void run()

{

while(true)

{

a.repaint();

try {Thread.sleep(1000);}

catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

} //yildizciz2Thread sinifinin sonu

public class analogSaat extends Applet

{

yildizCiz2Thread y;

double aci1,aci2,aci3;

Date c;

polar p;

public void init()

{

c=new Date(); p=new polar();

setBackground(renk.beyaz);

aci1=-c.getSeconds()/60.0\*Math.PI\*2; aci2=-c.getMinutes()/60.0\*Math.PI\*2;

aci3=-(c.getHours()%12)/12.0\*Math.PI\*2+aci2/12.0;

y=new yildizCiz2Thread(this);

y.start();

}

public void paint(Graphics g)

{

try{

g.setColor(renk.camgobegi);

yildiz.fillDaire(g,200,200,170);

g.setColor(renk.mavi);

yildiz.fillAkrep(g,200,200,16,160,aci2,0);

g.setColor(renk.yesil);

yildiz.fillAkrep(g,200,200,8,110,aci3,0);

g.setColor(renk.lacivert);

yildiz.fillAkrep(g,200,200,10,60,aci1,0);

g.setColor(renk.siyah); yildiz.drawDaire(g,200,200,165); yildiz.drawDaire(g,200,200,170);

yildiz.drawAkrep(g,200,200,16,160,aci2,0);

yildiz.drawAkrep(g,200,200,8,110,aci3,0);

yildiz.drawYildiz1(g,200,200,10,60,aci1);

aci1-=Math.PI/30.0;

aci2-=Math.PI/1800.0;

aci3-=Math.PI/21600.0;

for(int i=0;i<12;i++)

{

p.polarGir(165,Math.PI\*i/6);

g.setColor(renk.camgobegi);

if(i==0||i==3||i==6||i==9||i==12)

{

yildiz.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),5,20,0);

g.setColor(renk.siyah);

yildiz.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),5,20,0);

}

else

{

g.setColor(renk.lacivert);

yildiz.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

//yildiz.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

}

}

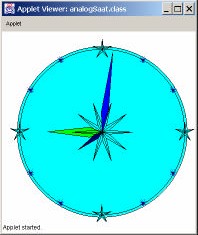
}

catch(NullPointerException e){}

}

}

### 09003.JPG



**Sekil 9.3 analogSaat.html appleti**

Ayni programin Swing JFrame versiyonu :

### Program 9.10 : analogSaatSWF.java JFrame programi, analog saati swing Frame olarak tanimlar

import javax.swing.\*; import java.util.\*;

import java.awt.\*; import yildizSW; import polar;

import java.awt.event.\*;

class yildizCiz2ThreadSWF extends Thread

{

JFrame a;

public yildizCiz2ThreadSWF(JFrame ai)

{

a=ai;

}

public void run()

{

while(true)

{

a.repaint();

try {Thread.sleep(1000);} catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

} //y• ld• zciz2Thread s• n• f• n• n sonu

public class analogSaatSWF extends JFrame

{

yildizCiz2ThreadSWF y; double aci1,aci2,aci3; Date c;

polar p;

public analogSaatSWF()

{

super("Analog Saat"); c=new Date();

// Türkiye standart (Greenwitch zamanina göre iki saat ileride) String[] ids = TimeZone.getAvailableIDs(+2 \* 60 \* 60 \* 1000); if (ids.length == 0)

System.exit(0);

SimpleTimeZone pdt = new SimpleTimeZone(+2 \* 60 \* 60 \* 1000, ids[0]); GregorianCalendar takvim=new GregorianCalendar(pdt); takvim.setTime(c);

p=new polar(); setBackground(renk.beyaz);

aci1=-takvim.get(Calendar.SECOND)/60.0\*Math.PI\*2; aci2=-takvim.get(Calendar.MINUTE)/60.0\*Math.PI\*2;

aci3=-((takvim.get(Calendar.HOUR))%12)/12.0\*Math.PI\*2+aci2/12.0; y=new yildizCiz2ThreadSWF(this);

y.start();

}

public void paint(Graphics g)

{

try{

g.setColor(renk.camgobegi); yildizSW.fillDaire(g,200,200,170); g.setColor(renk.mavi); yildizSW.fillAkrep(g,200,200,16,160,aci2,0); g.setColor(renk.yesil);

yildizSW.fillAkrep(g,200,200,8,110,aci3,0); g.setColor(renk.lacivert); yildizSW.fillAkrep(g,200,200,10,60,aci1,0); g.setColor(renk.siyah); yildizSW.drawDaire(g,200,200,165); yildizSW.drawDaire(g,200,200,170); yildizSW.drawAkrep(g,200,200,16,160,aci2,0); yildizSW.drawAkrep(g,200,200,8,110,aci3,0); yildizSW.drawYildiz1(g,200,200,10,60,aci1); aci1-=Math.PI/30.0;

aci2-=Math.PI/1800.0; aci3-=Math.PI/21600.0; for(int i=0;i<12;i++)

{

p.polarGir(165,Math.PI\*i/6); g.setColor(renk.camgobegi); if(i==0||i==3||i==6||i==9||i==12)

{

yildizSW.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),5,20,0); g.setColor(renk.siyah); yildizSW.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),5,20,0);

}

else

{

g.setColor(renk.lacivert); yildizSW.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

//yildiz.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

}

}

}

catch(NullPointerException e){}

}

public static void main(String[] args)

{

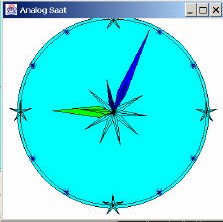
analogSaatSWF pencere= new analogSaatSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,400);

pencere.setVisible(true);

}

}

### 09004.JPG



**Sekil 9.4 analogSaatSWF.java JFrame çiktisi**

Ayni programin swing Japplet versiyonu :

### Program 9.11 : analogSaatSWA.java JApplet programi, analog saati swing Applet olarak tanimlar

import javax.swing.\*; import java.util.\*; import java.awt.\*; import yildizSW; import polar;

import java.awt.event.\*;

class yildizCiz2ThreadSWA extends Thread

{

JApplet a;

public yildizCiz2ThreadSWA(JApplet ai)

{

a=ai;

}

public void run()

{

while(true)

{

a.repaint();

try {Thread.sleep(1000);} catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

} //y• ld• zciz2Thread s• n• f• n• n sonu

public class analogSaatSWA extends JApplet

{

yildizCiz2ThreadSWA y; double aci1,aci2,aci3; Date c;

polar p;

public void init()

{

// Türkiye standart (Greenwitch zamanina göre iki saat ileride) String[] ids = TimeZone.getAvailableIDs(+2 \* 60 \* 60 \* 1000); if (ids.length == 0)

System.exit(0);

SimpleTimeZone pdt = new SimpleTimeZone(+2 \* 60 \* 60 \* 1000, ids[0]); GregorianCalendar takvim=new GregorianCalendar(pdt);

c=new Date(); takvim.setTime(c); p=new polar();

setBackground(renk.beyaz);

aci1=-takvim.get(Calendar.SECOND)/60.0\*Math.PI\*2; aci2=-takvim.get(Calendar.MINUTE)/60.0\*Math.PI\*2;

aci3=-((takvim.get(Calendar.HOUR))%12)/12.0\*Math.PI\*2+aci2/12.0; y=new yildizCiz2ThreadSWA(this);

y.start();

}

public void paint(Graphics g)

{

try{

g.setColor(renk.camgobegi); yildizSW.fillDaire(g,200,200,170); g.setColor(renk.mavi); yildizSW.fillAkrep(g,200,200,16,160,aci2,0); g.setColor(renk.yesil); yildizSW.fillAkrep(g,200,200,8,110,aci3,0); g.setColor(renk.lacivert); yildizSW.fillAkrep(g,200,200,10,60,aci1,0); g.setColor(renk.siyah); yildizSW.drawDaire(g,200,200,165); yildizSW.drawDaire(g,200,200,170); yildizSW.drawAkrep(g,200,200,16,160,aci2,0); yildizSW.drawAkrep(g,200,200,8,110,aci3,0); yildizSW.drawYildiz1(g,200,200,10,60,aci1); aci1-=Math.PI/30.0;

aci2-=Math.PI/1800.0; aci3-=Math.PI/21600.0; for(int i=0;i<12;i++)

{

p.polarGir(165,Math.PI\*i/6); g.setColor(renk.camgobegi); if(i==0||i==3||i==6||i==9||i==12)

{

yildizSW.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),5,20,0); g.setColor(renk.siyah); yildizSW.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),5,20,0);

}

else

{

g.setColor(renk.lacivert); yildizSW.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

//yildiz.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

}

}

}

catch(NullPointerException e){}

}

}

1. trafikIsigi.java programini inceleyiniz.

### Program 9.12 : trafikIsigi.java programi

import java.awt.\*; import java.applet.\*; import java.awt.event.\*;

class isikKontroluThread extends Thread

{

Applet a;

public double zaman=12.0;

public isikKontroluThread(Applet ai)

{

a=ai;

}

public void run()

{

while(true)

{

a.repaint();

//12 saniye bekle

try {Thread.sleep((int)(zaman\*1000));}

catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

public void setZaman(double zi)

{

zaman=zi;

}

public double getZaman()

{

return zaman;

}

}

public class trafikIsigi extends Applet implements ActionListener

{

isikKontroluThread yik;

private char which\_light='y';

private Button light\_switch;

private static int x=10;

private static int y= 40;

private static int h=360;

private static int w=100;

private static int radius=40;

private static int hight\_diff=30;

TextField t1;

Label l1,l2;

public void init() {

light\_switch = new Button("Switch");

light\_switch.setForeground(Color.black);

light\_switch.setBackground(Color.lightGray);

t1=new TextField("12");

l1=new Label("isik yanma zamani : ");

l2=new Label("saniye");

add(light\_switch); add(l1);

add(t1);

add(l2);

light\_switch.addActionListener(this);

t1.addActionListener(this);

yik=new isikKontroluThread(this);

yik.start();

}

public void paint(Graphics g) {

g.setColor(Color.black);

g.fillRect(x,y,w,h);

g.setColor(Color.white); g.drawRect(x-1,y-1,w+2,h+2);

g.drawRect(x-2,y-2,w+4,h+4);

switch(which\_light) {

case 'r':

which\_light='g';

g.setColor(Color.lightGray);

g.fillOval(x+10,y+hight\_diff,radius\*2,radius\*2);

g.fillOval(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2);

g.setColor(Color.green);

g.fillOval(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2);

break;

case 'y':

which\_light='r';

g.setColor(Color.lightGray); g.fillOval(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2); g.fillOval(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2);

g.setColor(Color.red);

g.fillOval(x+10,y+hight\_diff,radius\*2,radius\*2);

break;

case 'g':

which\_light='y';

g.setColor(Color.lightGray);

g.fillOval(x+10,y+hight\_diff,radius\*2,radius\*2);

g.fillOval(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2);

g.setColor(Color.yellow);

g.fillOval(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2);

repaint(1000);

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==light\_switch)

{

repaint();

try {Thread.sleep(1000);} catch (Exception exc) {}

}

else if(e.getSource()==t1)

{

Double i1=new Double(t1.getText());

yik.setZaman(i1.doubleValue());

}

}

}

### Program 9.13 : trafikIsigiSWP.java JPanel programi, trafik isigi panelini tanimlar

import javax.swing.\*; import java.awt.\*; import yildizSW; import java.awt.geom.\*; import renk;

public class trafikIsigiSWP extends JPanel

{

private char which\_light='y'; private static int x=10; private static int y= 40; private static int h=360; private static int w=100; private static int radius=40;

private static int hight\_diff=30;

public trafikIsigiSWP()

{

which\_light='y';

}

public trafikIsigiSWP(char l)

{

which\_light=l;

}

public void setIsik( char l)

{

which\_light=l;

}

public char getIsik()

{

return which\_light;

}

public void paintComponent(Graphics g) { Graphics2D g2 = (Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

g2.setColor(Color.black);

Rectangle2D dikdortgen1=new Rectangle2D.Double(x,y,w,h); g2.fill(dikdortgen1);

g2.setColor(Color.white);

Rectangle2D dikdortgen2=new Rectangle2D.Double(x-1,y-1,w+2,h+2); g2.draw(dikdortgen2);

Rectangle2D dikdortgen3=new Rectangle2D.Double(x-2,y-2,w+4,h+4); g2.draw(dikdortgen3);

Rectangle2D dikdortgen4=new Rectangle2D.Double(x-4,y-4,w+8,h+8); g2.setColor(Color.black);

g2.draw(dikdortgen3);

Ellipse2D elips1=new Ellipse2D.Double(x+10,y+hight\_diff,radius\*2,radius\*2);

Ellipse2D elips2=new Ellipse2D.Double(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2); Ellipse2D elips3=new Ellipse2D.Double(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2); switch(which\_light) {

case 'r':

which\_light='g'; g2.setColor(Color.lightGray); g2.fill(elips1);

g2.fill(elips2); g2.setColor(Color.green); g2.fill(elips3);

break; case 'y':

which\_light='r'; g.setColor(Color.lightGray); g2.fill(elips2); g2.fill(elips3); g2.setColor(Color.red); g2.fill(elips1);

break; case 'g':

which\_light='y'; g.setColor(Color.lightGray); g2.fill(elips3); g2.fill(elips1); g2.setColor(Color.yellow); g2.fill(elips2);

break;

//repaint(2000);

}

}

}

### Program 9.14 : trafikIsigiSWF.java JFrame programi, trafik isigi tanimlar ve çalistirir

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

public class trafikIsigi1SWF extends JFrame implements ActionListener

{

isikKontroluThread1 yik;

JButton light\_switch;

JTextField t1,t2;

JLabel l0,l1,l2,l3,l4; trafikIsigi1SWP ti1; Container c;

public trafikIsigi1SWF()

{

c=getContentPane();

JPanel p1=new JPanel();

p1.setLayout(new GridLayout(2,4)); yik=new isikKontroluThread1(this); yik.start();

ti1=new trafikIsigi1SWP(yik);

light\_switch = new JButton("Isigi degistir"); light\_switch.setForeground(Color.black); light\_switch.setBackground(Color.lightGray); t1=new JTextField(""+yik.getYesilZaman()); t2=new JTextField(""+yik.getKirmiziZaman()); l0=new JLabel(" ");

l1=new JLabel("yesil isik yanma zamani : "); l2=new JLabel("saniye ");

l3=new JLabel("kirmizi isik yanma zamani : "); l4=new JLabel("saniye "); p1.add(light\_switch);

p1.add(l1);

p1.add(t1);

p1.add(l2);

p1.add(l0);

p1.add(l3);

p1.add(t2);

p1.add(l4); light\_switch.addActionListener(this); t1.addActionListener(this); t2.addActionListener(this); c.add(p1,BorderLayout.NORTH); c.add(ti1,BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==light\_switch)

{

repaint();

}

else if(e.getSource()==t1)

{

Double i1=new Double(t1.getText()); yik.setYesilZaman(i1.doubleValue());

}

else if(e.getSource()==t2)

{

Double i1=new Double(t1.getText()); yik.setKirmiziZaman(i1.doubleValue());

}

}

public static void main(String[] args)

{

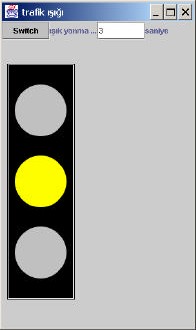
trafikIsigi1SWF pencere= new trafikIsigi1SWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(300,600);

pencere.setVisible(true);

}

}

### 09005.JPG



**Sekil 9.5 trafikIsigiSWF.java çiktisi**

**4.** H8O1

trafikIsigi programini inceleyiniz. Bu programda yesil ve kirmizinin ayni sürede yandigini göreceksiniz. (sari 1 saniye yanmaktadir.) Yesil ve

kirmizinin ayri ayri sürelerde yanmasini saglayiniz. Yanma sürelerini TextField alanindan kontrol ediniz.

### Program 9.15 H8O1.java programi trafik isigi kontrolu

import java.awt.\*; import java.applet.\*; import java.awt.event.\*;

class isikKontroluThread extends Thread

{

Applet a;

public double yesilZaman; public double kirmiziZaman; public double sariZaman; public char hangi\_isik;

public isikKontroluThread(Applet ai)

{

a=ai; yesilZaman=8.0; kirmiziZaman=4.0; sariZaman=1.0; hangi\_isik='y';

}

public void run()

{

while(true)

{

try {

if(hangi\_isik=='r')

{

Thread.sleep((int)(yesilZaman\*1000));

}

else if(hangi\_isik=='g')

{

Thread.sleep((int)(sariZaman\*1000));

}

else if(hangi\_isik=='y')

{

Thread.sleep((int)(kirmiziZaman\*1000));

}

a.repaint();

}

catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

public void setYesilZaman(double zi)

{

yesilZaman=zi;

}

public void setKirmiziZaman(double zi)

{

kirmiziZaman=zi;

}

public double getYesilZaman()

{

return yesilZaman;

}

public double getKirmiziZaman()

{

return kirmiziZaman;

}

public char getIsik()

{

return hangi\_isik;

}

public void setIsik(char hi)

{

hangi\_isik=hi;

}

}

public class H8O1 extends Applet implements ActionListener

{

isikKontroluThread yik; private Button light\_switch; private static int x=10; private static int y= 40;

private static int h=360; private static int w=100; private static int radius=40; private static int hight\_diff=30; TextField t1;

TextField t2; Label l1,l2,l3,l4; public void init()

{

yik=new isikKontroluThread(this); yik.start();

light\_switch = new Button("Switch"); light\_switch.setForeground(Color.black); light\_switch.setBackground(Color.lightGray); t1=new TextField(""+yik.getYesilZaman()); t2=new TextField(""+yik.getKirmiziZaman()); l1=new Label("yesil isik yanma zamani : "); l2=new Label("saniye");

l3=new Label("kirmizi isik yanma zamani : "); l4=new Label("saniye");

add(light\_switch); add(l1);

add(t1);

add(l2);

add(l3);

add(t2);

add(l4); light\_switch.addActionListener(this); t1.addActionListener(this); t2.addActionListener(this);

}

public void paint(Graphics g) { g.setColor(Color.black); g.fillRect(x,y,w,h); g.setColor(Color.white); g.drawRect(x-1,y-1,w+2,h+2);

g.drawRect(x-2,y-2,w+4,h+4); switch(yik.getIsik()) {

case 'r':

yik.setIsik('g'); g.setColor(Color.lightGray);

g.fillOval(x+10,y+hight\_diff,radius\*2,radius\*2); g.fillOval(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2); g.setColor(Color.green); g.fillOval(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2); break;

case 'y':

yik.setIsik('r'); g.setColor(Color.lightGray);

g.fillOval(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2); g.fillOval(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2); g.setColor(Color.red); g.fillOval(x+10,y+hight\_diff,radius\*2,radius\*2);

break; case 'g':

yik.setIsik('y'); g.setColor(Color.lightGray);

g.fillOval(x+10,y+hight\_diff,radius\*2,radius\*2); g.fillOval(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2); g.setColor(Color.yellow);

g.fillOval(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2); break;

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==t1)

{

Double i1=new Double(t1.getText()); yik.setYesilZaman(i1.doubleValue());

}

else if(e.getSource()==t2)

{

Double i1=new Double(t2.getText()); yik.setKirmiziZaman(i1.doubleValue());

}

else if(e.getSource()==light\_switch)

{

repaint();

try {Thread.sleep(1000);} catch (Exception exc) {}

}

}

}

**Program 9.16** trafikIsigi1SWP.java, Jpanel trafik isigi panelleri

import javax.swing.\*; import java.awt.\*; import yildizSW; import java.awt.geom.\*; import renk;

public class trafikIsigi1SWP extends JPanel

{

private static int x=10; private static int y= 40; private static int h=360; private static int w=100; private static int radius=40;

private static int hight\_diff=30; private isikKontroluThread1 yik;

public trafikIsigi1SWP(isikKontroluThread1 ikt)

{

yik=ikt;

}

public void paintComponent(Graphics g) { Graphics2D g2 = (Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

g2.setColor(Color.black);

Rectangle2D dikdortgen1=new Rectangle2D.Double(x,y,w,h); g2.fill(dikdortgen1);

g2.setColor(Color.white);

Rectangle2D dikdortgen2=new Rectangle2D.Double(x-1,y-1,w+2,h+2); g2.draw(dikdortgen2);

Rectangle2D dikdortgen3=new Rectangle2D.Double(x-2,y-2,w+4,h+4);

g2.draw(dikdortgen3);

Rectangle2D dikdortgen4=new Rectangle2D.Double(x-4,y-4,w+8,h+8); g2.setColor(Color.black);

g2.draw(dikdortgen3);

Ellipse2D elips1=new Ellipse2D.Double(x+10,y+hight\_diff,radius\*2,radius\*2);

Ellipse2D elips2=new Ellipse2D.Double(x+10,y+hight\_diff\*2+radius\*2,radius\*2,radius\*2); Ellipse2D elips3=new Ellipse2D.Double(x+10,y+hight\_diff\*3+radius\*4,radius\*2,radius\*2); switch(yik.getIsik()) {

case 'r':

yik.setIsik('g'); g2.setColor(Color.lightGray); g2.fill(elips1);

g2.fill(elips2); g2.setColor(Color.green); g2.fill(elips3);

break; case 'y':

yik.setIsik('r'); g.setColor(Color.lightGray); g2.fill(elips2); g2.fill(elips3); g2.setColor(Color.red); g2.fill(elips1);

break; case 'g':

yik.setIsik('y'); g.setColor(Color.lightGray); g2.fill(elips3); g2.fill(elips1); g2.setColor(Color.yellow); g2.fill(elips2);

break;

}

}

}

**Program 9.17** trafikIsigi1SWF.java, JFrame e trafik isigi programi import java.awt.\*;

import javax.swing.\*; import java.awt.event.\*;

public class trafikIsigi1SWF extends JFrame implements ActionListener

{

isikKontroluThread1 yik;

JButton light\_switch;

JTextField t1,t2;

JLabel l0,l1,l2,l3,l4; trafikIsigi1SWP ti1; Container c;

public trafikIsigi1SWF()

{

c=getContentPane();

JPanel p1=new JPanel();

p1.setLayout(new GridLayout(2,4)); yik=new isikKontroluThread1(this); yik.start();

ti1=new trafikIsigi1SWP(yik);

light\_switch = new JButton("Isigi degistir"); light\_switch.setForeground(Color.black); light\_switch.setBackground(Color.lightGray); t1=new JTextField(""+yik.getYesilZaman()); t2=new JTextField(""+yik.getKirmiziZaman()); l0=new JLabel(" ");

l1=new JLabel("yesil isik yanma zamani : "); l2=new JLabel("saniye ");

l3=new JLabel("kirmizi isik yanma zamani : "); l4=new JLabel("saniye "); p1.add(light\_switch);

p1.add(l1);

p1.add(t1);

p1.add(l2);

p1.add(l0);

p1.add(l3);

p1.add(t2);

p1.add(l4); light\_switch.addActionListener(this); t1.addActionListener(this); t2.addActionListener(this); c.add(p1,BorderLayout.NORTH); c.add(ti1,BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==light\_switch)

{

repaint();

}

else if(e.getSource()==t1)

{

Double i1=new Double(t1.getText()); yik.setYesilZaman(i1.doubleValue());

}

else if(e.getSource()==t2)

{

Double i1=new Double(t1.getText()); yik.setKirmiziZaman(i1.doubleValue());

}

}

public static void main(String[] args)

{

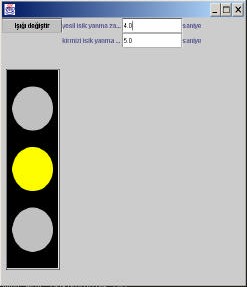
trafikIsigi1SWF pencere= new trafikIsigi1SWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(300,600);

pencere.setVisible(true);

}

}

### 09006.JPG



**Sekil 9.6 trafikIsigi1.java trafik isigi kontrol programi JFrame çiktisi**

**3. H8O2**

Birbirinden bagimsiz çalisan iki tarafik isigini iki ayri isik olarak simule edin. Bu isiklardan birincisinde yesil veya sari yanarken ikincisinde kirmizi yansin, ikincisinde yesil veya sari yanarken birincisinde kirmizi yansin. Kirmizi ve yesil süreleri ayni olmayabilir, fakat birbirinin tersi olacaktir. Örnegin birinci isikta 6 saniye kirmizi, 10 saniye yesil, 1 saniye sari yaniyorsa, ikinci isikta 11 saniye kirmizi, 5 saniye yesil, 1 saniye sari yanacaktir. Birinci isik saridan kirmiziya geçtigi anda ikinci isik yesil yanacaktir.

### Program 9.18 : IkiTrafikIsigi.java programi

/Title: 8. Hafta Odevi

//Version: 1.1

//Copyright: Copyright (c) 1999

//Author: Hasan Er

//Company: Gebze Yuksek Teknoloji Enstitusu

//Description:

// version 1.1 Turhan Coban tarafindan paint metotu

// degistirilmistir. import java.awt.\*; import java.applet.\*; import java.awt.event.\*; import java.applet.\*;

public class IkiTrafikIsigi extends Applet implements ActionListener

{

int bekleyen = 0;

boolean kontrolEdenVar = false;

IkiIsikKontroluThread yik[] = new IkiIsikKontroluThread[2]; char which\_light[]= new char[] {'r', 'r'};

TextField t1, t2; Label l1,l2,l3,l4; public void init()

{

t1=new TextField("12"); t2=new TextField("12");

l1=new Label("kirmizi isik yanma zamani : "); l2=new Label("saniye");

l3=new Label("yesil isik yanma zamani : "); l4=new Label("saniye");

add(l1);

add(t1);

add(l2);

add(l3);

add(t2);

add(l4); t1.addActionListener(this); t2.addActionListener(this); for(int i=0;i<2;i++)

{

yik[i]=new IkiIsikKontroluThread( this, i); yik[i].start();

}

}

public synchronized void kontrolAl()

{

try

{

if ( kontrolEdenVar )

{

bekleyen++; wait();

}

else

{

kontrolEdenVar = true;

}

}

catch ( InterruptedException interruptedEx)

{ interruptedEx.printStackTrace();}

}

public synchronized void kontrolBirak()

{

if ( bekleyen0 )

{

notify(); bekleyen--;

}

else

{

kontrolEdenVar = false;

}

}

public void paintIsik(Graphics g,int xi,int yi,int c)

{

int h1=360; int w1=100; int radius=40;

int hight\_diff=30; g.setColor(Color.black); g.fillRect(xi,yi,w1,h1); g.setColor(Color.white); g.drawRect(xi-1,yi-1,w1+2,h1+2); g.drawRect(xi-2,yi-2,w1+4,h1+4); switch(which\_light[c])

{

case 'g':

g.setColor(Color.lightGray); g.fillOval(xi+10,yi+hight\_diff,radius\*2,radius\*2); g.fillOval(xi+10,yi+hight\_diff\*2+radius\*2,radius\*2,radius\*2); g.setColor(Color.green); g.fillOval(xi+10,yi+hight\_diff\*3+radius\*4,radius\*2,radius\*2);

break; case 'r':

g.setColor(Color.lightGray); g.fillOval(xi+10,yi+hight\_diff\*2+radius\*2,radius\*2,radius\*2); g.fillOval(xi+10,yi+hight\_diff\*3+radius\*4,radius\*2,radius\*2); g.setColor(Color.red); g.fillOval(xi+10,yi+hight\_diff,radius\*2,radius\*2);

break; case 'y':

g.setColor(Color.lightGray); g.fillOval(xi+10,yi+hight\_diff,radius\*2,radius\*2); g.fillOval(xi+10,yi+hight\_diff\*3+radius\*4,radius\*2,radius\*2); g.setColor(Color.yellow); g.fillOval(xi+10,yi+hight\_diff\*2+radius\*2,radius\*2,radius\*2); break;

}

}

public void paint(Graphics g)

{

paintIsik(g,10,40,0); paintIsik(g,210,40,1);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==t1)

{

Double i1=new Double(t1.getText()); for ( int i=0; i<yik.length; i++)

yik[i].setKirmiziZaman(i1.doubleValue());

}

else if(e.getSource()==t2)

{

Double i1=new Double(t2.getText()); for ( int i=0; i<yik.length; i++)

yik[i].setYesilZaman(i1.doubleValue());

}

}

}

**Program 9.19 :** IkiIsikThread.java **programi**

//Title: 8. Hafta Ödevi

//Version:

//Copyright: Copyright (c) 1999

//Author: Hasan Er

//Company: Gebze Yüksek Teknoloji Enstitüsü

//Description:

public class IkiIsikKontroluThread extends Thread { double kirmiziZaman = 12.0;

double yesilZaman = 12.0; IkiTrafikIsigi ti;

int isikNo;

public IkiIsikKontroluThread(IkiTrafikIsigi tiparam, int isikNo)

{

ti = tiparam; this.isikNo = isikNo;

}

public void run()

{

char which = '\0';

try {

Thread.sleep( (int)( Math.random()\*1000 ) );

}

catch( InterruptedException interruptedEx ) { interruptedEx.printStackTrace();

}

while(true)

{

which = ti.which\_light[ isikNo ];

//12 saniye bekle try {

switch ( which ) {

case 'y' : ti.which\_light[ isikNo ] = 'r'; ti.repaint();

break;

case 'g' : ti.which\_light[ isikNo ] = 'y'; ti.repaint();

Thread.sleep( 1000 ); ti.kontrolBirak(); break;

case 'r' : ti.kontrolAl();

ti.which\_light[ isikNo ] = 'g'; ti.repaint(); Thread.sleep((int)(yesilZaman\*1000)); break;

}

}

catch(InterruptedException e)

{System.err.println(e.toString());}

}

}

public void setKirmiziZaman(double zi)

{

kirmiziZaman=zi;

}

public double getKirmiziZaman()

{

return kirmiziZaman;

}

public void setYesilZaman(double zi)

{

yesilZaman=zi;

}

public double getYesilZaman()

{

return yesilZaman;

}

}

### 09007.JPG



**Sekil 9.7 sinkronize olarak çalisan iki trafik isigi**

4. **H8OD2\_2000**

1. m3 boyutunda bir depomuz var. Bu depoya birinci pompa su doldurmakta, ikinci pompa is suyu bosaltmaktadir. Pompalarin debileri degistirilebilmektedir, ve kullanici tarafindan gerektiginde elle açilip kapatilabilmektedir. Depo tamamen bosalirsa bosaltan pompa otomatik olarak durdurulacaktir. Depo tamamen dolarsa dolduran pompa otamatik olarak durdurulacaktir. Bu prosesi gerçek zamanda simule eden iki pompa ve

bir depodan olusan bir sistem olusturunuz, sonuçlari belli zaman araliklariyla ekrana yazdiriniz (pompa1, pompa2 debileri ve depodaki su miktari ve baslangiçtan itibaren geçen zaman) veya bir grafik çikti üzerinden simulasyon çiktisini olusturunuz. (ayni degerleri grafik çiktida gösteriniz) Depodaki su miktarini bir ortak double degeri olarak düsünebiliriz.

### Program 9.20 : H8OD2\_2000.java programi

class H8OD2\_2000

{

public static void main(String args[])

{

havuz h=new havuz();

Pompa1 p1 = new Pompa1(h); Pompa2 p2 = new Pompa2(h);

p1.start();

p2.start();

}

}

class Pompa1 extends Thread

{

private havuz pTut; private double debi;

public Pompa1(havuz h)

{

pTut=h;

}

public void run()

{

while(true)

{

while((debi = Math.abs((Math.random()\*5))) == 0);

pTut.ekleSu(debi);

try{

sleep( (int) (Math.random()\*3000 ) );

}

catch(InterruptedException e)

{

System.err.println("Exeption "+e.toString());

}

}

}

}

class Pompa2 extends Thread

{

private havuz cTut; private double debi;

public Pompa2(havuz h)

{

cTut=h;

}

public void run()

{

while (true)

{

try{

sleep( (int) (Math.random()\*3000));

}

catch(InterruptedException e)

{

System.err.println("Exeption "+e.toString());

}

while((debi = Math.abs((Math.random()\*5))) == 0); cTut.cikarSu(debi);

}

}

}

class havuz

{

private double OrtakSu = 0.0; private boolean pompa1 = true; private boolean pompa2 = false; private final double kapasite = 10.0;

public synchronized void ekleSu(double suMiktari)

{

while (!pompa1)

{

try{

wait();

}

catch (InterruptedException e)

{

System.err.println("Exception:"+e.toString() );

}

}

OrtakSu += suMiktari;

if (OrtakSu >= kapasite){

pompa1 = false;

System.out.println("Pompa1'in debisi " + suMiktari + " dir. Depo Dolu!!! (10)."); OrtakSu = kapasite;

}

OrtakSu);

else

System.out.println("Pompa1'in debisi " + suMiktari + " dir. Depo daki su miktari :" +

pompa2 = true; notify();

}

public synchronized void cikarSu(double suMiktari)

{

while (!pompa2)

{

try{

wait();

}

catch (InterruptedException e)

{

System.err.println("Exception:"+e.toString() );

}

}

OrtakSu -= suMiktari;

if (OrtakSu <= 0)

{

pompa2=false;

}

else

System.out.println("Pompa2'in debisi " + suMiktari + " dir. Depo Bos!!! (0)."); OrtakSu = 0;

System.out.println("Pompa2'in debisi " + suMiktari + " dir. Depo daki su miktari :" +

OrtakSu);

pompa1 = true;

notify();

}

}

### AWT Animasyon örnegi

Bu problemde animasyon’un thread kullanarak nasil yapilacagini görecegiz. Aslinda kavram yeni degil aslinda su ana kadar bu bölümde yaptigimiz tüm uygulamalar bir animasyon. Bu program java döküman kütüphanesinden alinmistir.

### Program 9.21 : AnimatorApplication.java programi

/\*

\* awt animasyon

\*/

import java.awt.\*; import java.awt.event.\*;

/\*

* Arthur van Hoff'un animasyon örnegi.
* Bu örnek tüm animasyonlar için örnek olarak kullanilabilir.

\*/

public class AnimatorApplication extends Frame

implements Runnable { int frameNumber = -1;

int delay;

Thread animatorThread; boolean frozen = false;

AnimatorApplication(int fps, String windowTitle) { super(windowTitle);

delay = (fps > 0) ? (1000 / fps) : 100;

addMouseListener(new MouseAdapter() { public void mousePressed(MouseEvent e) {

if (frozen) { frozen = false;

startAnimation();

} else {

frozen = true; stopAnimation();

}

}

});

addWindowListener(new WindowAdapter() { public void windowIconified(WindowEvent e) {

stopAnimation();

}

public void windowDeiconified(WindowEvent e) { startAnimation();

}

public void windowClosing(WindowEvent e) { System.exit(0);

}

});

}

public void startAnimation() { if (frozen) {

//Hiç bir sey yapmaniz gerekmez

//

} else {

//Animasyon basliyor...

if (animatorThread == null) { animatorThread = new Thread(this);

}

animatorThread.start();

}

}

public void stopAnimation() {

//Animasyon tread'ini durdur animatorThread = null;

}

public void run() {

//Önce bu thread'in önceligini düsürüyoruz,

// böylece öbür programlara engel olmaz Thread.currentThread().setPriority(Thread.MIN\_PRIORITY);

//Baslama zamanini hatirla

long startTime = System.currentTimeMillis();

//hangi thread oldugumuzu hatirla

Thread currentThread = Thread.currentThread();

//gerçek zaman animasyon döngüsü

while (currentThread == animatorThread) {

//Animasyon ekranindaki rakami artir frameNumber++;

//rakami göster repaint();

//beklemeye al try {

startTime += delay; Thread.sleep(Math.max(0,

startTime-System.currentTimeMillis()));

} catch (InterruptedException e) { break;

}

}

}

//animasyon ekranini çiz public void paint(Graphics g) {

g.drawString("Frame " + frameNumber, 5, 50);

}

public static void main(String args[]) { AnimatorApplication animator = null; int fps = 10;

// saniyedeki çerçeve sayisini hesapla if (args.length > 0) {

try {

fps = Integer.parseInt(args[0]);

} catch (Exception e) {}

}

animator = new AnimatorApplication(fps, "Animasyon awt"); animator.setSize(200, 60);

animator.setVisible(true); animator.startAnimation();

}

}

### 09008.JPG



**Sekil 9.8 awt animasyon örnegi**

1. **swing Animasyon örnegi**

Bu örnek problemde bir öncekinin temel olarak aynisi, sadece swing kullanilmis hali

### Program 9.22 : AnimatorApplicationSW.java programi

/\*

* Swing versiyonu

\*/

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

/\*

* Arthur van Hoff'un animasyon örnegi.
* Bu örnek tüm animasyonlar için örnek olarak kullanilabilir.

\*/

public class AnimatorApplicationSW extends JFrame

implements Runnable { AnimappPanel animappPanel;

int frameNumber = -1; int delay;

Thread animatorThread; boolean frozen = false;

AnimatorApplicationSW(int fps, String windowTitle) { super(windowTitle);

delay = (fps > 0) ? (1000 / fps) : 100;

animappPanel = new AnimappPanel(); getContentPane().add(animappPanel);

addMouseListener(new MouseAdapter() { public void mousePressed(MouseEvent e) {

if (frozen) { frozen = false;

startAnimation();

} else {

frozen = true; stopAnimation();

}

}

});

addWindowListener(new WindowAdapter() { public void windowIconified(WindowEvent e) {

stopAnimation();

}

public void windowDeiconified(WindowEvent e) { startAnimation();

}

public void windowClosing(WindowEvent e) { System.exit(0);

}

});

}

public void startAnimation() { if (frozen) { }

else {

//animasyonu baslat

if (animatorThread == null) { animatorThread = new Thread(this);

}

animatorThread.start();

}

}

public void stopAnimation() {

//animasyon treadini durdur. animatorThread = null;

}

public void run() {

//önceligi düsür. Thread.currentThread().setPriority(Thread.MIN\_PRIORITY);

//baslama zamani

long startTime = System.currentTimeMillis();

//thread tanimi

Thread currentThread = Thread.currentThread();

//animasyon döngüsü

while (currentThread == animatorThread) {

//animasyon çerçevesini bir artir frameNumber++;

//göster repaint();

//geciktir try {

startTime += delay; Thread.sleep(Math.max(0,

startTime-System.currentTimeMillis()));

} catch (InterruptedException e) { break;

}

}

}

class AnimappPanel extends JPanel { public AnimappPanel() {}

//animasyon çerçevesini çiz

public void paintComponent(Graphics g) { super.paintComponent(g); g.drawString("Frame " + frameNumber, 5, 50);

}

}

public static void main(String args[]) { AnimatorApplicationSW animator = null; int fps = 10;

// Get frames per second from the command line argument if (args.length > 0) {

try {

fps = Integer.parseInt(args[0]);

} catch (Exception e) {}

}

animator = new AnimatorApplicationSW(fps,"Animasyon swing versiyonu"); animator.setSize(200, 60);

animator.setVisible(true); animator.startAnimation();

}

}

### 09009.JPG



**Sekil 9.9 swing animasyon örnegi**

1. **zaman ayarli swing Animasyon örnegi**

**Program 9.23 : AnimatorApplicationTimer.java programi**

/\*

* Tam anlamiyla swing'e dönüstürülmüs animasyon

\*/

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

/\*

* Arthur van Hoff'un animasyon örneginden yazilmistir.

\*/

public class AnimatorApplicationTimer extends JFrame

implements ActionListener { int frameNumber = -1;

int delay;

boolean frozen = false;

JLabel label;

Timer timer;

AnimatorApplicationTimer(int fps, String windowTitle) { super(windowTitle);

delay = (fps > 0) ? (1000 / fps) : 100; timer = new Timer(delay, this); timer.setInitialDelay(0); timer.setCoalesce(true);

addWindowListener(new WindowAdapter() { public void windowIconified(WindowEvent e) {

stopAnimation();

}

public void windowDeiconified(WindowEvent e) { startAnimation();

}

public void windowClosing(WindowEvent e) { System.exit(0);

}

});

Container contentPane = getContentPane(); contentPane.addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent e) { if (frozen) {

frozen = false; startAnimation();

} else {

}

}

});

frozen = true; stopAnimation();

label = new JLabel("Frame ", JLabel.CENTER); contentPane.add(label, BorderLayout.CENTER);

}

public void startAnimation() { if (frozen) {

} else {

timer.start();

}

}

public void stopAnimation() { timer.stop();

}

public void actionPerformed(ActionEvent e) { frameNumber++;

label.setText("Frame " + frameNumber);

}

public static void main(String args[]) { AnimatorApplicationTimer animator = null; int fps = 10;

if (args.length > 0) { try {

fps = Integer.parseInt(args[0]);

} catch (Exception e) {}

}

animator = new AnimatorApplicationTimer(fps, "zamani ayarlanabilen swing animasyonu"); animator.pack();

animator.setVisible(true); animator.startAnimation();

}

}

### 09010.JPG



**Sekil 9.10 zaman ayarli swing animasyon örnegi**

### BÖLÜM 10 JAVA GIRDI - ÇIKTI PROGRAMLANMASI

### JAVANIN I/O (GIRDI ÇIKTI ) KÜTÜPHANESINDE YER ALAN SINIFLAR VE INTERFACE’LER

Javadaki tüm girdi çikti programlamasi veri akis (stream) kavramina dayanir. Veri akisini bir su akisi veya bir kablolu sistem üzerinden bilgi akisi gibi düsünebiliriz. Javada sadece dosya degil yazici, scanner gibi herhangi bir kaynaga veri iletmek istedigimizde bu veri akis sistemini kullaniriz. Javada oldukça genis bir veri akis kütüphanesi bulunmaktadir.

java.io kütüphanesinde yer alan girdi çikti programlama siniflari sunlardir :

InputStream

FilterInputStream

LineNumberInputStream BufferedInputStream DataInputStream PushbackInputStream

ByteArrayInputStream FileInputStream ObjectInputStream PipedInputStream SequenceInputStream StringBufferInputStream

OutputStram

ByteArrayOutputStream FileOutputStream FilterOutputStream

BufferedOutputStream DataOutputStream PrintStream

ObjectOutputStream PipedOutputStream

ObjectStreamClass StreamTokenizer Reader

FileReader BufferedReader

LineNumberReader CharArrayReader FilterReader

PushbackReader InputStreamReader

FileReader PipedReader StringReader

Writer

BufferedWriter CharArrayWriter FilterWriter OutputStreamWriter PipedWriter PrintWriter StringWriter

File FileDescriptor

RandomAccessFile

Bu siniflarin hepsinin genel görevi javaya girdi saglamak ve çikti almaktir. Burada bunlardan bir kisminin islevlerini ve nasil çalistiklarini inceleyecegiz. Önce temel ana siniflarin bazilarinin tanimlarina daha detayli

bakalim. InputStream sinifi byte türü yazi okuma sinifidir. Tüm byte türü bilgi akis kanali siniflarinin ana sinifidir. Tanimi :

public abstract class InputStream extends Object

{

public InputStream();

public int available() throws IOException; public void close() throws IOException; public synchronized void mark(int readlimit); public boolean markSupported();

public abstract int read() throws IOException; public int read(byte b[]) throws IOException;

public int read(byte b[],int off,int len) throws IOException; public synchronized void reset() throws IOException; public long skip(long n) throws IOException;

}

InputStream sinifinin byte türü yazma pareleli olan OutputStream sinifinin tanimi ise : public abstract class OutputStream extends Object

{

public outputStream();

public void close() throws IOException; public void flush() throws IOException;

public abstract void write(int b) throws IOException; public void write(byte b[]) throws IOException;

public void write(byte b[],int baslamanoktasi,int uzunluk) throws IOException;

}

Abstract sinif Reader, InputStream sinifina olduça yakindir, yalniz data transferinde byte yerine char kullanir, böylece girdi çiktida unicode stream kullanabilme olasiligi yakalar.

public abstract class Reader extends Object

{

protected Reader(); protected Reader(Object o);

public abstract void close() throws IOException; public void mark(int readlimit) throws IOException; public boolean markSupported() ;

public abstract int read() throws IOException; public int read(char c[]) throws IOException;

public int read(char c[],int baslamanoktasi,int uzunluk) throws IOException; public boolean ready() throws IOException;

public void reset() throws IOException; public long skip(long n) throws IOException;

}

Abstract sinif Writer da Output straem sinifina esdegerdir. Fakat ayni readerda oldugu gibi bu sinif da veri akisinda char ve string kullanir.

public abstract class Writer extends Object

{

public outputStream();

public void close() throws IOException; public void flush() throws IOException;

public abstract void write(int b) throws IOException; public void write(char c[]) throws IOException;

public void write(char c[],int baslamanoktasi,int uzunluk) throws IOException; public void write(String s) throws IOException;

public void write(String s,int baslamanoktasi,int uzunluk) throws IOException;

}

Bu siniflar temel siniflardir. Bunlarin altinda yer alan çesitli siniflara asagidaki bölümlerde daha detayli bakacagiz.

### FILE (DOSYA) SINIFI

File sinifi girdi çikti islemlerini saglarken, dosya isimleri ve bulunduklari dizin ile ilgili bilgi verir. File sinifinin tanimi :

Public class File extends Object implements Serializable, Comparable

{

public static final String pathSeperator; public static final char pathSeperatorChar; public static final String seperator;

public static final char seperatorChar; public File(String dosya\_ismi);

public File(String dizin\_ismi, String dosya\_ismi); public File(File file\_tipi\_nesne,String dosya\_ismi); public boolean canRead();

public boolean canWrite();

public boolean compareTo(Object o);

public boolean compareTo(File baskabirdosya); public boolean createNewFile();

public static File createTempFile(String isim) throws IOException;

public static File createTempFile(String isim,File dosya) throws IOException; public boolean delete();

public void deleteOnExit(); public boolean equals(Object o); public boolean exists(Object o); public File getAbsooluteFile(); public File getAbsoolutePath();

public File getCanonicalFile() throws IOException; public File getCanonicalPath() throws IOException; public String getName();

public String getParent(); public File getParentFile(); public String getPath(); public int hashCode(); public boolean isAbsolute(); public boolean isDirectory(); public boolean isFile(); public boolean isHidden(); public long lastModified(); public long length();

public String[] list();

public String[] list(FilenameFilter filitrefonksiyonu); public static File[] listRoots();

public boolean mkdir(); public boolean mkdirs();

public boolean renameTo(File yeniisim);

public boolean setLastModified(long nezaman); public boolean setReadOnly();

public String toString();

public URL toURL() throws MAlformedURLException;

}

Kurucu metotlarin çagirilmasina örnek olarak : File f1=new File(“/”);

File f2=new File(“/”,”autoexec.bat”);

File f3=new File(f1,”autoexec.bat”);

tanimlarini verebiliriz. File sinifinda tanimlanmis olan Metotlardan bazilarinin tanimlari sunlardir.

### canRead()

Tanimlanan dosyadan bilgi okunulabilirligini test eder.

### canWrite()

Tanimlanan dosyaya bilgi yazilabilirligini test eder.

**delete()** Dosyayi siler **equals(Object)**

Dosya ismini Object te verilen dosya ismiyle karsilastirir.

### exists()

Dosyanin mevcut olup olmadigini kontrol eder.

### getAbsolutePath()

Dosyanin tam dizin ismini ve dosya ismini verir( co/java/prog/Hosgeldiniz.java).

### getName()

Dosyanin ismini verir.(Hosgeldiniz.java)

### getParent()

Dosyanin içinde bulundugu dizinin ismini verir.(prog)

### getPath()

dosyanin ismini ve içinde bulundugu dizinin ismini verir.(prog/Hosgeldiniz.java)

### isAbsolute()

Dosya isminin mutlak isim olup olmadigini kontrol eder.(Eger dosya ismi co/java/prog/Hosgeldiniz.java olarak verilmisse true degeri verir.) **isDirectory()**

verilen isimin bir dizin ismi olup olmadigini kontrol eder.

### isFile()

verilen isimin birdosya isimi olup olamdigini kontrol eder.

### lastModified()

Dosyanin en son degistirildigi tarihi verir

### length()

Dosyanin boyutunu Byte olarak verir.

### list()

Verilen dizinin içindeki dosyalarin listesini verir.

### list(FilenameFiltre)

Verilen dizinin içindeki Filtre nesnesindeki tanima uyan dosyalarin listesini verir.

### mkdir()

Yeni bir dizin olusturur.

### mkdirs()

O anda tanimli olan dizinin içine bir alt dizin olusturur.

**renameTo(File)** Dosyanin ismini degistirir. **toString()**

Dosya ve dizin isimleri toplulugunun String degiskeni esdegerini verir.

File(dosya) sinfindaki metodlardan da görülebilecegi gibi bu sinif dosyadan okuyup yazma gibi bir islem yürütmez. File sinifinin temel islevi girdi çikti dosyasinin adlandirilmasi ve isimle ilgili fonksiyonlara ulasilmasidir. File sinifinin bazi metotlarini ufak bir program içinde kullanalim.

### Program 10.1 . FileTestiSW.java programi

import java.awt.\*; import java.io.File;

import java.util.HashMap; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.filechooser.\*; import javax.swing.event.\*;

public class FileTestiSW

{

public static void main(String args[])

{

//önce JFileChooser kullanarak dosyayi seç, sonra dosya özelliklerini göster. File f=new File("Hosgeldiniz.java");

JFileChooser dosyasec=new JFileChooser(); int onay = dosyasec.showOpenDialog(null);

//veya kayit için int onay = dosyasec.showSaveDialog(null);

if(onay==JFileChooser.APPROVE\_OPTION)

{

f=dosyasec.getSelectedFile();

}

String s="";

s+="Dosya ismi : "+f.getName()+"\n"; s+="Dizin ismi : "+f.getPath()+"\n";

s+="Mutlak Dizin ismi : "+f.getAbsolutePath()+"\n"; s+="Üst Dizin ismi : "+f.getParent()+"\n"; s+=f.exists() ? "Mevcut " : " Mevcut degil "+"\n"; s+=f.canWrite() ? "Yazilabilir " : " Yazilamaz"+"\n"; s+=f.canRead() ? "Okunabilir " : " Okunamaz"+"\n"; s+=f.isDirectory() ? "Dizin " : " Dizin degil"+"\n"; s+=f.isFile() ? "Dosya " : " Dosya degil"+"\n";

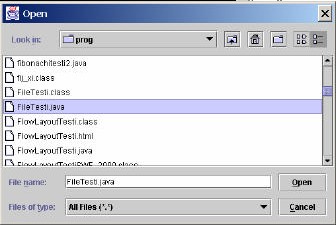
s+=f.isAbsolute() ? "Mutlak dizin ismi " : " Mutlak dizin ismi degil"+"\n"; s+="Son Degisiklik : "+f.lastModified()+"\n";

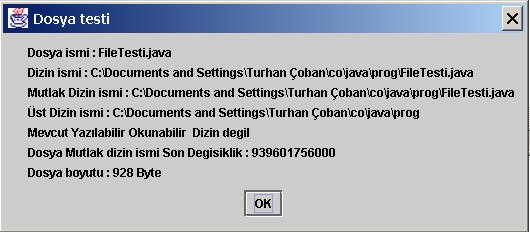
s+="Dosya boyutu : "+f.length()+" Byte"+"\n"; JOptionPane.showMessageDialog(null,s,"Dosya testi",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

### 10001.JPG



**Sekil 10.2 FileTesti.java programinin içindeki dosya seçim penceresi 10002.JPG**

**Sekil 10.2 FileTesti.java programinin sonuclari**

Burada swing hazir dosya seçme programi JfileChooser kullanilarak dosya ismi seçilmis, sonar bu dosya ile ilgili çesitli bilgiler JoptionPane grafik ortaminda aktarilmistir.

### ARDISIK (SEQUENTIAL) DOSYA YARATILMASI

Java bilgi dosyalarini birbiri ardisira gelen byte veya char (unicode) akisi olarak görür. Her dosya dosya-bitis isaretiyle sonlanir. Yeni bir dosya açildiginda bu dosyayla ilgili bir nesne olusturulur. Yeni bir java programi açildiginda üç akis nesnesi otomatik olarak açilir. Bunlar **System.in , System.out, System.err** nesneleridir. Eger kendimiz bir girdi veya çikti akis nesnesi olusturmak istersek **FileInputStream** veya **FileOutputStream** sinifi bir nesne tanimliyabiliriz. Akisin içindeki degiskenleri sadece byte olarak okumak yerine direk olarak double veya integer gibi degiskenler türünden okumak istersek **DataInputStream ve DataOutputStream** siniflarini kullanabiliriz. Genel olarak FileOutputStream ve DataOutputStream siniflarini asagidaki gibi bir arada kullanabiliriz:

### DataOutputStream ciktiakimi;

**try{ ciktiakimi=new DataOutputStream(new FileOutputStream(“isim.dat”));} catch(IOException e) {System.err.println(“Dosya acilamadi\n”+e.toString());}**

Bu deyim bize isim.dat isimli dosyadan bir fileOutputStream araciligiyla açilan dosyayi DataOutputStream’e çevirip gerçek degiskenler cinsinden akis kanali açar. DataOutputStream ve FileOutputStream siniflarinin tanimlari :

public class DataOutputStream extends FilterOutputStream implements DataOutput

{

public DataOutputStream(OutputStram o); protected int written

public void flush() public final void size()

public synchronized void write(byte[],int baslangiçnoktasi,int uzunluk); public synchronized void write(int);

public final void writeBoolean(boolean b); public final void writeBytes(String s) public final void writeChar(int)

public final void writeChars(String) public final void writeDouble(double) public final void writeFloat(float) public final void writeInt(int)

public final void writeLong(long) public final void writeShort(int) public final void writeUTF(String)

}

FileOutputStream ise dosya sinifi File ile DataOutputStream sinifi arasindaki gerekli tanimlari saglar

public class FileOutputStream extends OutputStream

{

public FileOutputStream(File file) throws IOException;

public FileOutputStream(FileDescriptor fd) throws IOException; public FileOutputStream(String s) throws IOException;

public FileOutputStream(String s, boolean b) throws IOException; public void close() throws IOException;

protected void finalize() throws IOException; public void getFD() throws IOException; public void write(byte[]) throws IOException;

public void write(byte[], int, int) throws IOException; public void write(int) throws IOException;

public final FileDescriptor getFD() throws IOException;

}

Kanal açildiktan sonra **DataOutputStream** sinifinin metodlarini kullanarak dosyaya yazi yazabiliriz. Yazma metodlarinin islevlerine bir göz atarsak :

### write(byte B[] )

byte tipi boyutludegiskeni açilmis olan DataOutputStream kanalina yazar.

### write(byte B[], int baslangicindeksi, int indeksboyutu)

byte tipi boyutludegiskeni açilmis olan DataOutputStream kanalina yazar.

Baslangiç indeksinden baslar ve indeksboyutu uzunlugundaki kismi yazar.

**write(int B)** int datayi data output stream’e yazar. **writeBoolean(boolean B)** Boolean degiskeni DataOutputStream’e yazar. **writeByte(int)** int degiskeni byte olarak DataOutputStream’e yazar.

**writeBytes(String)** String degiskeni byte boyutlu degisken olarak DataOutputStream’e yazar. **writeChar(int)** int degiskeni char degisken olarak DataOutputStream’e yazar. **writeChars(String)** String degiskeni Char degisken olarak DataOutputStream’e yazar. **writeDouble(double)** double degiskeni double degisken olarak DataOutputStream’e yazar. **writeFloat(float)** float degiskeni float degisken olarak DataOutputStream’e yazar. **writeInt(int)** float degiskeni float degisken olarak DataOutputStream’e yazar. **writeLong(long)** long degiskeni long degisken olarak DataOutputStream’e yazar. **writeShort(int)** int degiskeni short (short integer) degisken olarak DataOutputStream’e yazar. **writeUTF(String)** String degiskeni unicode UTF-8 formati karekter boyutu olarak yazar.

Bu sekilde yazdigimiz bir veri akisin okumak için ise :

### DataInputStream girdiakimi; try{

**girdiakimi=new DataInputStream(new FileInputStream(“isim.dat"));**

**}**

**catch(IOException e)**

**{**

**System.err.println("Dosya acilamadi\n"+e.toString()); System.exit(1);**

**}**

tanimini kulanabiliriz. Buradaki DataInputStream sinifinin tanimi :

public class DataInputStream extends FilterInputStream implements DataInput

{

public DataInputStream(InputStream in); public DataInputStream(InputStream in);

public final int read(byte b[]) throws IOException;

public final int read(byte b[], int baslangiçdegeri, int uzunluk) throws IOException; public final void readFully(byte b[]) throws IOException;

public final void readFully(byte b[],int baslangiçdegeri, int uzunluk) throws IOException; public final int skipBytes(int n) throws IOException;

public final boolean readBoolean() throws IOException; public final byte readByte() throws IOException;

public final int readUnsignedByte() throws IOException; public final short readShort() throws IOException;

public final int readUnsignedShort() throws IOException; public final char readChar() throws IOException;

public final int readInt() throws IOException; public final long readLong() throws IOException; public final float readFloat() throws IOException;

public final double readDouble() throws IOException; public final String readLine() throws IOException; public final String readUTF() throws IOException;

public final static String readUTF(DataInput in) throws IOException;

}

seklindedir. FileInputStream ise :

public FileInputStream(File file) throws FileNotFoundException

{

public FileInputStream(FileDescriptor fdObj); public int read() throws IOException;

public int read(byte b[]) throws IOException;

public int read(byte b[],int baslangiçnoktasi,int uzunluk) throws IOException; public long skip(long n) throws IOException;

public int available() throws IOException; public void close() throws IOException;

public final FileDescriptor getFD() throws IOException; protected void finalize() throws IOException;

}

seklinde tanimlidir. **DataInputStream** metdunun alt okuma metodlarina biraz daha detayli göz atarsak:

### read(byte[])

byte veri akisini direk okur.

### read(byte[], int baslangiçnoktasi, int boyut)

byte veri akisini baslangiçnoktasi indeksinden boyut uzunluguna kadar okurdirek okur.

### readBoolean()

boolean degisken (true veya false) okur.

### readByte()

8-bit byte integer okur

### readChar()

unicode karecter(char) okur

**readDouble()** double okur **readFloat()** float okur

### readFully(byte[])

byte boyutlu degiskenini, byte[] boyutlu girdi degiskeninin boyutuna göre okur.

### readFully(byte[], int, int) readInt()

Integer okur

### readLine()

Bir satiri satirbasi yap komutuna kadar okur. Java 1.0 komutudur. 1.1 ve üzerinde bazi okuma hatalari yaptigi görüldügünden kullanilmamasi tavsiye edilir.

### readLong()

Long tamsayi degiskeni okur.

### readShort()

Short tamsayi degiskeni okur

### readUnsignedByte()

8-bitlik isaretsiz tamsayi degiskeni okur.

### readUnsignedShort()

16-bitlik isaretsiz tamsayi degiskeni okur.

### readUTF()

Unicode karekteri okur.

### readUTF(DataInput)

Unicode Karakterini (Character) DataInput girdi akisindan okur.

### skipBytes(int n)

n byte degeri okumadan atlar.

Simdi bir örnek problem ile bu yazma ve okuma islemine daha yakindan bir göz atalim. Program 10.2 de yeni bir dosya açilmakta, ve dosyaya veri girilmektedir. Program 10.3 de ayni programin swing kullanilarak yazilmis versiyonu mevcuttur. Program 10.4 de ise yaratilan dosyaya girilen veriler okunmaktadir.

### Program 10.2 ardisikDosyaYarat.java programi

import java.io.\*; import java.awt.\*;

import java.awt.event.\*;

public class ardisikDosyaYarat extends Frame implements ActionListener

{

private TextField hesapIsmiKutusu,isimKutusu,soyIsimKutusu,hesapKutusu; private Button enter,done;

private DataOutputStream cikti; private Label H,I,S,P;

public ardisikDosyaYarat()

{

super("Musteri dosyasi Ac");

### try{

**cikti=new DataOutputStream(new FileOutputStream("musteri.txt"));**

**} catch(IOException e)**

**{ System.err.println("Dosya dogru acilamadi\n"+e.toString()); System.exit(1);**

**}**

setSize(300,150);

setLayout(new GridLayout(5,2)); H=new Label("Hesap numarasi :"); add(H);

hesapIsmiKutusu=new TextField(); add(hesapIsmiKutusu);

I=new Label("isim : "); add(I);

isimKutusu=new TextField(20); add(isimKutusu);

S=new Label("Soyisim : "); add(S);

soyIsimKutusu=new TextField(20); add(soyIsimKutusu);

P=new Label("Hesaptaki para : "); add(P);

hesapKutusu=new TextField(20); add(hesapKutusu);

enter=new Button("Gir"); enter.addActionListener(this); add(enter);

done=new Button("Cikis"); done.addActionListener(this); add(done);

setVisible(true);

}

public void hesapEkle()

{

int accountNumber=0;

Double d;

if(!hesapIsmiKutusu.getText().equals(""))

{

try{ accountNumber=Integer.parseInt(hesapIsmiKutusu.getText()); if(accountNumber0)

{

cikti.writeInt(accountNumber); cikti.writeUTF(isimKutusu.getText()); cikti.writeUTF(soyIsimKutusu.getText()); d=new Double(hesapKutusu.getText()); cikti.writeDouble(d.doubleValue());

}

hesapIsmiKutusu.setText(""); isimKutusu.setText(""); soyIsimKutusu.setText(""); hesapKutusu.setText("");

}

catch(NumberFormatException nfe)

{

System.err.println("Hesap numarasi tamsayi degisken olamalidir");

}

catch(IOException io)

{

System.err.println("Dosyaya yazarken hata olustu\n"+io.toString()); System.exit(1);

}

}

}

public void actionPerformed(ActionEvent e)

{

hesapEkle(); if(e.getSource()==done)

{

try{ cikti.close();} catch(IOException io)

{

System.err.println("Dosya kapatilamadi\n"+io.toString());

}

System.exit(0);

}

}

public static void main(String args[])

{

new ardisikDosyaYarat();

}

}

### 10003.JPG



**Sekil 10.3 ArdisikDosyaYarat.java programi ve sonuclarinin Frame çiktisinda görülmesi**

**Program 10.3 ardisikDosyaYaratSWF\_200.java programi**

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import javax.swing.\*;

public class ardisikDosyaYaratSWF\_2000 extends JFrame implements ActionListener

{

private JTextField hesapIsmiKutusu,isimKutusu,soyIsimKutusu,hesapKutusu; private JButton enter,done;

private DataOutputStream cikti; private JLabel H,I,S,P; Container c;

public ardisikDosyaYaratSWF\_2000()

{

super("Müsteri dosyasi Aç"); c=getContentPane();

try{

cikti=new DataOutputStream(new FileOutputStream("musteri.txt"));

} catch(IOException e)

{ System.err.println("Dosya dogru açilamadi\n"+e.toString()); System.exit(1);

}

c.setLayout(new GridLayout(5,2)); H=new JLabel("Hesap numarasi :"); c.add(H);

hesapIsmiKutusu=new JTextField(); c.add(hesapIsmiKutusu);

I=new JLabel("isim : "); c.add(I);

isimKutusu=new JTextField(20); c.add(isimKutusu);

S=new JLabel("Soyisim : "); c.add(S);

soyIsimKutusu=new JTextField(20); c.add(soyIsimKutusu);

P=new JLabel("Hesaptaki para : "); c.add(P);

hesapKutusu=new JTextField(20); c.add(hesapKutusu);

enter=new JButton("Gir"); enter.addActionListener(this); c.add(enter);

done=new JButton("Cikis"); done.addActionListener(this); c.add(done);

setVisible(true);

}

public void hesapEkle()

{

int accountNumber=0;

Double d;

if(!hesapIsmiKutusu.getText().equals(""))

{

try{ accountNumber=Integer.parseInt(hesapIsmiKutusu.getText()); if(accountNumber>0)

{

cikti.writeInt(accountNumber); cikti.writeUTF(isimKutusu.getText()); cikti.writeUTF(soyIsimKutusu.getText()); d=new Double(hesapKutusu.getText()); cikti.writeDouble(d.doubleValue());

}

hesapIsmiKutusu.setText(""); isimKutusu.setText(""); soyIsimKutusu.setText(""); hesapKutusu.setText("");

}

catch(NumberFormatException nfe)

{

System.err.println("Hesap numarasi tamsayi degisken olamalidir");

}

catch(IOException io)

{

System.err.println("Dosyaya yazarken hata olustu\n"+io.toString()); System.exit(1);

}

}

}

public void actionPerformed(ActionEvent e)

{

hesapEkle(); if(e.getSource()==done)

{

try{ cikti.close();} catch(IOException io)

{

System.err.println("Dosya kapatilamadi\n"+io.toString());

}

System.exit(0);

}

}

public static void main(String[] args)

{

ardisikDosyaYaratSWF\_2000 pencere= new ardisikDosyaYaratSWF\_2000(); pencere.setSize(300,150);

pencere.addWindowListener(new BasicWindowMonitor()); pencere.setVisible(true);

}

}

### 10004.JPG



**Sekil 10.4 ArdisikDosyaYaratSWF.java programi ve sonuclarinin Frame çiktisinda görülmesi**

**Program 10.4 ardisikDosyaOku.java programi**

import java.io.\*; import java.awt.\*;

import java.awt.event.\*;

public class ardisikDosyaOku extends Frame implements ActionListener

{

private TextField accountField,firstNameField,lastNameField,balanceField; private Button next,done;

private DataInputStream input; public ardisikDosyaOku()

{

super("Musteri dosyasini oku");

### try{

**input=new DataInputStream(new FileInputStream("musteri.dat"));**

**} catch(IOException e) { System.err.println("Dosya acilamadi\n"+e.toString()); System.exit(1);**

}

setSize(300,150);

setLayout(new GridLayout(5,2)); add(new Label("Hesap Numarasi :")); accountField=new TextField(); add(accountField);

add(new Label("isim : "));

firstNameField=new TextField(20); add(firstNameField);

add(new Label("soyisim : ")); lastNameField=new TextField(20); add(lastNameField);

add(new Label("Hesaptaki para : ")); balanceField=new TextField(20); add(balanceField);

next=new Button("bir sonraki hesap"); next.addActionListener(this); add(next);

done=new Button("cikis"); done.addActionListener(this); add(done);

setVisible(true);

}

public void readRecord()

{

int account; String first,last; double balance;

{

try{ account=input.readInt(); first=input.readUTF(); last=input.readUTF(); balance=input.readDouble();

accountField.setText(String.valueOf(account)); firstNameField.setText(first); lastNameField.setText(last); balanceField.setText(String.valueOf(balance));

}

catch(EOFException eof)

{

closeFile();

}

catch(IOException io)

{

System.err.println("Dosyay• okurken hata olustu\n"+io.toString()); System.exit(1);

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==next) readRecord();

else closeFile();

}

private void closeFile()

{

try{ input.close(); System.exit(0);

}

catch(IOException e)

{

System.err.println("Dosya Kapama Hatasi olustu\n"+e.toString()); System.exit(1);

}

}

public static void main(String args[])

{

new ardisikDosyaOku();

}

}

### 10004.JPG



**Sekil 10.4 ArdisikDosyaOku.java programi ve sonuclarinin Frame çiktisinda görülmesi**

ArdisikDosyaYarat.java programinda **musteri.txt** dosyasi

### try{

**cikti=new DataOutputStream(new FileOutputStream("musteri.txt"));**

**}**

**catch(IOException e)**

**{**

**System.err.println("Dosya dogru acilamadi\n"+e.toString()); System.exit(1);**

**}**

gurubunu kullanarak açilmistir. Bu gurubu File sinifini da kullanarak

### File f=new File(“musteri.txt”); f=f.getAbsolutePath(f);

**try{**

**cikti=new DataOutputStream(new FileOutputStream(f));**

**}**

**catch(IOException e)**

**{ System.err.println("Dosya dogru acilamadi\n"+e.toString()); System.exit(1);**

**}**

seklinde de yaratabilirdik. Ikinci sekilde dosyayi tanimlarken tam dizin ismini de otomatik olarak tanimlamis olurduk.

Dosyanin okunmasi içinse

### try{

**input=new DataInputStream(new FileInputStream("musteri.dat"));**

**}**

**catch(IOException e)**

**{**

**System.err.println("Dosya acilamadi\n"+e.toString()); System.exit(1);**

**}**

gurubunu kullandik. Eger dosya ismi de degiskende disaridan okuduktan sonra ismi File sinifina yükleyip sonra akis kanali açabiliriz.

Dosya ile islemler bittikten sonra dosya kapanir.

### private void closeFile()

**{**

**try{ input.close(); System.exit(0);**

**}**

**catch(IOException e)**

**{**

**System.err.println("Dosya Kapama Hatasi olustu\n"+e.toString()); System.exit(1);**

**}**

**}**

**closeFile()** metotu dosya akisini kapatmak amaciyla olusturulmustur.

Bu örnekte kullandigimiz, **DataOutputStream, FileOutputStream, DataInputStream, FileInputStream** girdi çikti akis kontrollari temel olarak daha önce tanimlarini vermis oldugumuz **OutputStream** ve **InputStream** siniflarindan türetilmis siniflardi. Ve temel olarak byte degisken türü üzerinden veri akisini saglamaktadirlar.

Yine yukarida tanimlanmis olan Reader ve Writer siniflari ise ayni islemi char sinifi veri akisi üzerinden yapmaktadir ve char sinfi direk olarak unicode olarak tanimlandigindan herhangibir hataya sebep vermeden daha iyi bir veri akisi saglar. Bu sinifin alt siniflari olarak tanimlanan ve yazma isleminde kullanilan siniflarin kullanilmasina bir göz atalim.

Genel olarak Writer sinifindan türetilen ve en çok kullanilan Yazma kanali **PrintWriter**, **BufferedWriter, FileWriter** veya **OutputStreamWriter** siniflarinin beraber kullanilmasiyla gerçeklestirilebilir. Bu siniflarin birlikte kulanilmalarini su deyimlerle örnekleyebiliriz :

dosya "a.txt" e yazdirmak için :

PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt"))); tanimi kullanilabilir.

**PrintWriter** sinifinin tanimi :

public class PrintWriter extends Writer

### {

public PrintWriter(OutputStream o) ;

public PrintWriter(OutputStream o, boolean b) ; public PrintWriter(Writer w) ;

public PrintWriter(Writer w, boolean b); public void flush();

public void close();

public boolean checkError(); protected void setError(); public void write(int c); public void write(char buf[]);

public void write(String s,int baslamaindeksi,int uzunluk); public void write(String s);

public void print(boolean b); public void print(char c); public void print(int i); public void print(long l); public void print(float f); public void print(double d); public void print(Object obj); public void println();

public void println(boolean x);

public void println(char x); public void println(int x); public void println(long x); public void println(float x); public void println(double x); public void println(char x[]); public void println(Object x);

### }

**PrintWriter** sinifinin mevcut metotlarinin islevleri sunlardir :

checkError() Tüm veriyi gönderir ve hata kontrolu yapar. close() veri akis kanalini kapatir.

flush() veri akis kanalindaki tüm veriyi gönderir (ve bufferi yeni veri için bosaltir). print (boolean) boolean degisken yazar.

print (char) character degisken yazar.

print (char[]) boyutlu chracter degiskenleriyazar. print (double) double yazar.

print (float) float yazar.

print (int) integer yazar.

print (long) long yazar.

print (Object) object sinifinin tanimladigi çiktiyi (object türüne göre degisebilir) yazar. print(String) String yazar

println () satir sonu yapar alt satira geçer.

println (boolean) boolean yazar ve satir sonu yapar alt satira geçer. println (char) character yazar ve satir sonu yapar alt satira geçer.

println (char[]) boyutlu character degiskenleri yazar ve satir sonu yapar alt satira geçer. println (double) double yazar ve satir sonu yapar alt satira geçer.

println (float) float yazar ve satir sonu yapar alt satira geçer. println (int) Print an integer, and then finish the line. println (long) long yazar ve satir sonu yapar alt satira geçer.

println (Object) object sinifinin tanimladigi çiktiyi (object türüne göre degisebilir) yazar ve satir sonu yapar alt satira geçer.

println(String) String yazar ve satir sonu yapar alt satira geçer. setError() bir hata olustugunu gösterir

write (char[]) boyutlu character degiskenlerini yazar

write (char[], int, int) boyutlu character degiskenlerini birinci integer(tamsayi) boyutundan ikinci integer boyutuna kadar yazar.

write (int) tek bir character yazar.

write (String) string yazar

write(String, int, int) string degiskenini birinci integer(tamsayi) boyutundan ikinci integer boyutuna kadar yazar.

BufferedWriter sinifinin tanimi :

public class BufferedWriter extends Writer

{

public BufferedWriter(Writer cikti);

public BufferedWriter(Writer cikti,int boyut); public void write(int c) throws IOException;

public void write(char c[],int baslamaindeksi,int uzunluk) throws IOException; public void write(String s, int baslamaindeksi,int uzunluk) throws IOException; public void newLine() throws IOException;

public void flush() throws IOException; public void close() throws IOException;

}

seklindedir. Yazma verimini arttirmak için akisa ilave edilmektedir.

FileWriter sinifi ise File sinifi ile baglantimizi kuran (dosyayi tanimlayan sinifimizdir.). Writer sinifinin altinda yer alan OutputStreamWriter Sinifinin alt sinifidir. tanimi :

public class FileWriter extends OutputStreamWriter

{

public FileWriter(String fileName) throws IOException;

public FileWriter(String fileName,boolean append) throws IOException; public FileWriter(File file) throws IOException;

public FileWriter(FileDescriptor fd);

}

OutputStreamWriter sinifinin tanimi :

public class OutputStreamWriter extends Writer

{

public OutputStreamWriter(OutputStream o,String enc) throws UnsupportedEncodingException; public OutputStreamWriter(OutputStream o);

public String getEncoding();

public void write(int c) throws IOException;

public void write(char c[],int baslamaindeksi,int uzunluk) throws IOException; public void write(String str,int baslamaindeksi,int uzunluk) throws IOException; public void flush() throws IOException;

public void close() throws IOException;

}

Bu siniflari kullanarak bir yazma kanai açmak istersek :

### PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt"))); cfout.println("hello");

Komutunu kullanabiliriz. Bu komut

hello string degiskenini a.txt dosyasina yazdirir.

Dosya veya ekrandan okumak için ise paralel olarak **BufferedReader, FileReader ve InputStreamReader**

siniflarini kullanabiliriz. BufferedReader sinifinin tanimi :

public class BufferedReader extends Reader

{

public BufferedReader(Reader giris,int boyut); public BufferedReader(Reader giris);

public int read(char c[],int off,int len) throws IOException; public int read() throws IOException;

public String readLine() throws IOException; public long skip(long n) throws IOException; public boolean ready() throws IOException; public boolean markSupported();

public void mark(int readAheadLimit) throws IOException; public void reset() throws IOException;

public void close() throws IOException;

}

BufferedReader ve FileReader siniflarini birarada kullanarak bir okuma kanali açabiliriz. örnegin a.txt dosyasindan okumak için :

### BufferedReader cfin=new BufferedReader(new FileReader("a.txt"));

deyimini kullanabiliriz.

Bu terimleri kullanarak dosyaya yazma ve okumayla ilgili ufak bir örnek verelim

### Program 10.5 : YazOku.java programi

import java.io.\*; class YazOku

{

public static void main (String args[]) throws IOException

{

String s1="ilk deger"; String s2="ilk deger"; String s3="ilk deger";

### PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt")));

cfout.println("Merhaba"); cfout.println("isminiz nedir");

cfout.println("sizinle tanistigima memnun oldum"); cfout.close();

### BufferedReader cfin=new BufferedReader(new FileReader("a.txt"));

s1=cfin.readLine(); s2=cfin.readLine(); s3=cfin.readLine(); cfin.close(); System.out.println("s1 = "+s1); System.out.println("s2 = "+s2); System.out.println("s3 = "+s3);

}

}

Program çiktisi :

s1 = Merhaba

s2 = isminiz nedir

s3 = sizinle tanistigima memnun oldum dosya a.txt :

Merhaba isminiz nedir

sizinle tanistigima memnun oldum

Seklinde olacaktir. Programimiz a.txt dosyasini olusturup, üç string degerini yazdiktan sonra okuma kanali açarak bu veriyi okudu.

Benzer bir okuma sinifi olarak su ana kadar kullandigimiz Text sinifini da verebiliriz. Aslinda Text sinifinin içerigine baktigimizda yukarida verilen siniflarin organize bir sekilde kullanilmasindan ibaret oldugunu rahatlikla görebiliriz. Text sinifinin listesi :

### Program 10.6 Text.java programi

//============================================

// Java nümerik analiz paketi

// Class Text to read data from screen or file

// and write (by using print) to screen or file

// formatted c printf like comand structure

// Dr. Turhan Coban

//============================================

import java.io.\*; import java.util.\*;

class Text

{

BufferedReader in;

/\*

---------------------------------------------------

Static routines provided are:

---------------------------------------------------

public static void prompt(String s);

public static int readInt(DataInputStream in); public static int readInt(BufferedReader in);

public static double readDouble(DataInputStream in); public static double readDouble(BufferedReader in); public static String readString(DataInputStream in); public static String readString(BufferedReader in); public static char readChar(DataInputStream in); public static char readChar(BufferedReader in);

----------------------------------------------------

Dynamic routines provided are :

----------------------------------------------------

public void Text()

public void Text(String s1) public void Text(File f1) public int readInt();

public double readDouble(); public String readString(); public char readChar(); Sample use :

--------- reading a double--------------

DataInputStream cin=new DataInputStream(System.in); double number;

number=Text.readDouble(cin); veya

BufferedReader cin=new BufferedReader(new InputStreamReader(System.in)); double number;

number=Text.readDouble(cin); veya

double number;

Text cin=new Text(); number=cin.readDouble();

------ reading data from a file "datafile.dat"------ DataInputStream fin=new DataInputStream (new FileInputStream("datafile.dat"));

double number; number=Text.readDouble(fin);

veya

double number;

BufferedReader fin=new BufferedReader(new FileReader("datafile.dat")); double number;

number=Text.readDouble(fin); veya

double number;

Text cin=new Text("dataFile.dat"); number=cin.readDouble();

------ printing data into a file "printfile.dat"----

PrintStream fout=new PrintStream(new FileOutputStream("printfile.dat")); String a="turhan";

fout.println(a); veya

BufferedWriter fout=new BufferedWriter(new FileWriter("datafile.dat")); String a="Turhan");

fout.println(a);

----------------------------------------------------

\*/

private static StringTokenizer T; private static String S;

public Text()

{

in=new BufferedReader(new InputStreamReader(System.in));

}

public Text(String s1) throws IOException

{

in=new BufferedReader(new FileReader(s1));

}

public static void prompt (String s) { System.out.print(s + " "); System.out.flush();

}

public static int readInt (DataInputStream in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

String item = T.nextToken();

return Integer.valueOf(item.trim()).intValue();

}

catch (NoSuchElementException e1) { refresh (in);} catch(NumberFormatException e2)

{ //System.err.println("Error in number, try again.");

}

}

}

public int readInt() throws IOException

{

return Text.readInt(in);

}

public String readStringLine() throws IOException

{

return Text.readStringLine(in);

}

public double readDouble() throws IOException

{

return Text.readDouble(in);

}

public String readString() throws IOException

{

return Text.readString(in);

}

public char readChar() throws IOException

{

return Text.readChar(in);

}

public static int readInt (BufferedReader in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

String item = T.nextToken();

return Integer.valueOf(item.trim()).intValue();

}

catch (NoSuchElementException e1) { refresh (in);

} catch(NumberFormatException e2)

{ //System.err.println("Error in number, try again.");

}

}

}

public static char readChar (DataInputStream in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

return T.nextToken().charAt(0);

}

catch(NoSuchElementException e1) {refresh (in);}

}

}

public static char readChar (BufferedReader in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

return T.nextToken().charAt(0);

}

catch(NoSuchElementException e1) {refresh (in);}

}

}

public static double readDouble(DataInputStream in) throws IOException

{

if(T==null) refresh(in); while (true) {

try {

String item = T.nextToken();

return Double.valueOf (item.trim()).doubleValue();

} catch(NoSuchElementException e1) { refresh (in);

} catch(NumberFormatException e2)

{

//System.err.println("Error in number, try again.");

}

}

}

public static double readDouble(BufferedReader in) throws IOException

{

if(T==null) refresh(in); while (true) {

try {

String item = T.nextToken();

return Double.valueOf (item.trim()).doubleValue();

} catch(NoSuchElementException e1) { refresh (in);

} catch(NumberFormatException e2)

{

//System.err.println("Error in number, try again.");

}

}

}

// this method is deprecated, but still kept for historical reasons

// prefer BufferedReader version

public static String readString(DataInputStream in) throws IOException

{

if(T==null) refresh (in);

while (true) {

try {return T.nextToken();}

catch (NoSuchElementException e1) { refresh (in);

}

}

}

public static String readString(BufferedReader in) throws IOException

{

if(T==null) refresh (in); while (true) {

try {return T.nextToken();}

catch (NoSuchElementException e1) { refresh (in);

}

}

}

public static String readStringLine(DataInputStream in) throws IOException

{

//reads a line of strings from DataInputStream in int ch;

String r = ""; boolean done = false; while (!done)

{ try

{ ch = in.read();

if (ch < 0 || (char)ch == '\n' || (char)ch == '\0') done = true;

else

r = r + (char) ch;

}

catch(java.io.IOException e)

{ done = true;

}

}

return r.substring(0,(r.length()-1));

}

public static String readStringLine(BufferedReader in) throws IOException

{

//reads a line of strings from BufferedReader in int ch;

String r = ""; boolean done = false; while (!done)

{ try

{ ch = in.read();

if (ch < 0 || (char)ch == '\n' || (char)ch == '\0') done = true;

else

r = r + (char) ch;

}

catch(java.io.IOException e)

{ done = true;

}

}

return r.substring(0,(r.length()-1));

}

private static void refresh (DataInputStream in) throws IOException

{

String s = in.readLine();

if (s==null) throw new EOFException(); T = new StringTokenizer(s);

}

private static void refresh (BufferedReader in) throws IOException

{

String s = in.readLine();

if (s==null) throw new EOFException(); T = new StringTokenizer(s);

}

}

Text sinifini kullanarak sequential bilgiyi dosyadan okuyan YazOku1.java programi altta verilmistir.

### Program 10.7 YazOku1.java programi

import java.io.\*; import Text; class YazOku1 {

public static void main (String args[]) throws IOException

{

String s1="bos String"; String s2="bos String"; String s3="Bos String"; String s4="Bos String";

PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt"))); cfout.println("Merhaba");

cfout.println("isminiz nedir"); cfout.println("Hosgeldiniz"); cfout.close();

**Text cfin=new Text("a.txt");** s1=cfin.readString(); s2=cfin.readString(); s3=cfin.readString(); s4=cfin.readString(); System.out.println("s1 = "+s1); System.out.println("s2 = "+s2); System.out.println("s3 = "+s3); System.out.println("s4 = "+s4);

}

}

bu programin çiktisi :

s1 = Merhaba s2 = isminiz s3 = nedir

s4 = Hosgeldiniz

seklindedir. Burada görüldügü gibi her kelime ayri bir string olarak algilanmistir. Bu yapiyi gerçeklestiren Text sinifi içerisinde kullanilan StringTokenizer sinifidir.

### String item = T.nextToken();

ifadesiyle okunmakta olan Stringi cümlelere böler.

Ardisik dosya okuma örnegi olarak Atom.java, AtomTest.java, Atom.txt, gurubunu verebiliriz. Atom.java Atomlarin bazi fiziksel özelliklerini tanimlayan bir programdir. Program atomlarla ilgili veriyi Atom.txt dosyasindan okumaktadir.

### Program 10.8 Atom.java programi

//======================================================

// Thermodynamic Package in java

// Class Atom

// Properties of single atom

// -----------------------------------------------------

// Programmer : Dr. Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Reasearch Institute

// [turhan@mam.gov.tr](mailto:turhan@mam.gov.tr)

// -----------------------------------------------------

// File Name : Atom.java

// This file contains the atom class

// this class sets basic properties of requested atom

// required data is read from atom.txt

// atom.txt should be copied to the same directory as your

// atom.java file

// =====================================================

import java.io.\*; import java.util.\*; import Text;

class Atom

{

public int number; // atomic number public String name; // name of the the atom public String symbol; // symbol of the atom public double mass; // mass of the atom public double N; // number of the atoms **BufferedReader fina;**

public Atom()

{

//empty constructor number=0; symbol=""; name="";

mass=0;

N=1;

}

//=================================================================

public Atom(int anumber,double NN) throws IOException

{

//this constructor will load Atom data from a given atomic number

//and number of atoms

//type

String atom\_name; String atom\_symbol; double atom\_mass; int atom\_number;

### try{

**fina=new BufferedReader(new FileReader("Atom.txt"));**

**} catch(IOException e)**

**{**

**System.err.println("Error Opening File Atom.dat\n"+e.toString()); System.exit(1);**

**}**

**try {**

**while(fina != null)**

**{**

**atom\_number=Text.readInt(fina); atom\_name= Text.readString(fina); atom\_symbol= Text.readString(fina);**

**atom\_mass=Text.readDouble(fina); if(atom\_number==anumber)**

**{ N=NN;**

**number=atom\_number; name=atom\_name; symbol=atom\_symbol; mass=atom\_mass\*N; break;**

**}**

**}**

**} catch(EOFException e\_eof) {fina.close(); }**

**}**

public Atom(String st1,double NN) throws IOException

{

// this constructor will load Atom data from a given atom name

// or atom symbol and number of atoms

// type

String atom\_name; String atom\_symbol; double atom\_mass; int atom\_number; **try{**

### fina=new BufferedReader(new FileReader("Atom.txt"));

**} catch(IOException e)**

**{**

**System.err.println("Error Opening File Atom.dat\n"+e.toString()); System.exit(1);**

**}**

**try { while(fina != null)**

**{**

**atom\_number=Text.readInt(fina); atom\_name= Text.readString(fina); atom\_symbol= Text.readString(fina); atom\_mass=Text.readDouble(fina);**

if(st1.equals(atom\_name) || st1.equals(atom\_symbol))

{ N=NN;

number=atom\_number; name=atom\_name; symbol=atom\_symbol; mass=atom\_mass\*N; break;

}

}

### } catch(EOFException e\_eof) {fina.close(); }

}

public Atom(String st1) throws IOException

{

// this constructor will load Atom data from a given atom name

// or atom symbol and number of atoms

// type

double NN=1; String atom\_name; String atom\_symbol; double atom\_mass; int atom\_number; try{

fina=new BufferedReader(new FileReader("Atom.txt"));

} catch(IOException e)

{

System.err.println("Error Opening File Atom.dat\n"+e.toString()); System.exit(1);

}

try { while(fina != null)

{

atom\_number=Text.readInt(fina); atom\_name= Text.readString(fina); atom\_symbol= Text.readString(fina); atom\_mass=Text.readDouble(fina);

if(st1.equals(atom\_name) || st1.equals(atom\_symbol))

{ N=NN;

number=atom\_number; name=atom\_name; symbol=atom\_symbol; mass=atom\_mass\*N; break;

}

}

} catch(EOFException e\_eof) {fina.close(); }

}

public Atom(Atom a)

{

number=a.number; name=a.name; symbol=a.symbol; mass=a.mass; N=a.N;

}

public Atom(Atom a, double NN)

{

number=a.number; name=a.name; symbol=a.symbol; mass=a.mass; N=NN;

}

//=================================================================

public void assign(Atom a)

{

number=a.number; name=a.name; symbol=a.symbol; mass=a.mass; N=a.N;

}

//=================================================================

//boolean equals logical comparisons public boolean equals(String s)

{

boolean b;

if(name.equals(s) || symbol.equals(s)) return true;

else

return false;

}

public boolean equals(int n)

{

boolean b; if(number==n) return true; else

return false;

}

//=================================================================

public String toString()

{

String st=symbol; if(N != 1.0)

if(N==Math.floor(N)) st = st + (int)N; else st = st + N;

return st;

}

}

//=================================================================

### Program 10.9 AtomTest.java programi

//======================================================

// Thermodynamic Package in java

// Class Atom Test//Properties of single atom

// Dr. Turhan Coban

// File Name : AtomTest.java

// This file contains the atom class

// this class sets basic properties of requested atom

// required data is read from atom.dat.

// =====================================================

import java.io.\*; class AtomTest

{

public static void main(String args[]) throws IOException

{

Atom n2=new Atom("N",2); System.out.println(n2.toString()); int n=8;

Atom o2=new Atom("O",n); System.out.println(o2.toString());

}

}

//=================================================================

### Atom.txt girdi dosyasi

1. Hydrogen H 1.00794
2. Helium He 4.002602
3. Lithium Li 6.941
4. Berylium Be 9.012182
5. Boron B 10.811
6. Carbon C 12.011

7Nitrogen N 14.00674

8 Oxygen O 15.9994

9 Florine F 18.9984032

10 Neon Ne 20.1797

11 Sodium Na 22.989768

1. Magnesium Mg 24.305
2. Aliminium Al 26.981539
3. Silicon Ag 28.0855
4. Phosphourus P 30.973762
5. Sulphur S 32.066
6. Chlorine Cl 35.4527 18 Argon Ar 39.938
7. Potassium K 39.0983
8. Calcium Ca 40.078
9. Scandium Se 44.95591 22 Titanium Ti 47.88
10. Vanadium V 50.9415
11. Chromium Cr 51.9961
12. Manganese Mn 54.93805 26 Iron Fe 55.587
13. Cobalt Co 58.9332
14. Nickel Ni 58.69
15. Copper Cu 63.546
16. Zink Zn 65.39
17. Gallium Ga 69.723
18. Germanium Ge 72.61
19. Arsenic As 74.92159
20. Selenium Si 78.96
21. Bromine Br 79.904
22. Kyrpton Kr 83.8
23. Rubidium Ru 85.4678
24. Strontium Sr 87.62
25. Yttirum Yt 88.90585
26. Zirconium Zr 91.224
27. Niobium Nb 92.90638
28. Molybdenium Mo 95.94
29. Technetium Tc 98
30. Ruthenium Sm 101.007
31. Rhodium Rh 102.9055
32. Palladium Pd 106.42
33. Silver Ag 107.8682
34. Cadmium Cd 112.411
35. Indium In 114.82

50 Tin Sn 118.71

1. Anthimony Sb 121.75
2. Tellerium Te 127.6

53 Iodine I 126.90447

54 Xenon Xe 131.29

55 Cesium Cs 132.90543

1. Barium Ba 137.327
2. Lanthanum La 138.9055
3. Cerium Ce 140.115

59Praseodymium Pr 140.90765

1. Neodymium Nd 144.24
2. Promethium Pm 145
3. Samarium Sc 150.36
4. Europium Eu 151.965
5. Gadolinium Gd 157.25
6. Terbium Tb 158.92534
7. Dysprosium Dy 162.5
8. Holmium Ho 164.93032
9. Erbium Er 167.26
10. Thulium Tm 168.93421
11. Ytterbium Yb 173.04
12. Lutetium Lu 174.967
13. Hafnium Hf 178.49
14. Tantalum Ta 180.9479
15. Tungsten W 183.85
16. Rhenium Re 186.207
17. Osmium Os 190.2
18. Iridium Ir 192.22
19. Platinium Pt 195.08

79 Gold Au 196.96654

1. Mercury Hg 200.59
2. Thallium Tl 204.3833
3. Lead Pb 207.2
4. Bismuth Bi 208.98037
5. Polonium Po 209
6. Astatine At 210
7. Radon Rn 222
8. Francium Fr 223
9. Radium Ra 226
10. Actinium Ac 227
11. Thorium Th 232.0381
12. Protactinium Pa 231
13. Uranium U 238.0289
14. Neptunium Np 237
15. Plutonium Pu 244
16. Americium Am 243
17. Curium Cm 247
18. Berkelium Bk 247
19. Californium Cf 251
20. Einsteinium Es 252 100 Fermium Fm 257 101 Mendelevium Md 258 102 Nobelium No 259 103 Lawrencium Lr 260

Program AtomText.java çiktisi :

N2 O8

Burada hemen sunu kaydedelim. Programdan da anlasilacagi gibi Atom.java java’yi ögretme amaciyla hazirlanmamistir. Gazlarin termodinamik özelliklerini hesaplayan bir program gurubunun parçasi olarak hazirlanmistir.

### RASLANTISAL ULASIM DOSYASI OKU (RANDOM ACCESS FILE)

Raslantisal Ulasim Dosyasinin (Random Access File) Ardisik Dosya sisteminden temel farki, Teyp kaydiyla CD kaydi arasindaki fark gibidir. Ardisik dosyalar Teyp kaydi seklinde birbiri ardi sira gelen bilgilerden olusur.

Raslantisal Ulasim sisteminde ise CD gibi herhangi bir kayita kayit numarasi kullanilarak dogrudan ulasmak mümkündür. Raslantisal Ulasim dosyasi islemleri RandomAccessFile sinifi üzerinden yapilir. Bu sinifin tanimi :

public class RandomAccessFile extends Object implements DataOutput, DataInput

{

public RandomAccessFile(String name,String mode) throws IOException public RandomAccessFile(File file,String mode) throws IOException public final FileDescriptor getFD() throws IOException

public int read() throws IOException

public int read(byte b[],int off,int len) throws IOException public int read(byte b[]) throws IOException

public final void readFully(byte b[]) throws IOException

public final void readFully(byte b[],int off,int len) throws IOException public int skipBytes(int n) throws IOException

public void write(int b) throws IOException public void write(byte b[]) throws IOException

public void write(byte b[],int off,int len) throws IOException; public long getFilePointer() throws IOException;

public void seek(long pos) throws IOException; public void seek(long pos) throws IOException; public long length() throws IOException; public void close() throws IOException;

public final byte readByte() throws IOException;

public final int readUnsignedByte() throws IOException; public final short readShort() throws IOException;

public final int readUnsignedShort() throws IOException;

public final char readChar() throws IOException; public final int readInt() throws IOException; public final long readLong() throws IOException; public final float readFloat() throws IOException;

public final double readDouble() throws IOException; public final String readLine() throws IOException; public final String readUTF() throws IOException;

public final void writeBoolean(boolean v) throws IOException; public final void writeByte(int v) throws IOException;

public final void writeShort(int v) throws IOException; public final void writeChar(int v) throws IOException; public final void writeInt(int v) throws IOException; public final void writeLong(long v) throws IOException; public final void writeFloat(float v) throws IOException;

public final void writeDouble(double v) throws IOException; public final void writeBytes(String s) throws IOException; public final void writeChars(String s) throws IOException; public final void writeUTF(String str) throws IOException;

}

Burada tanimlanan metodlarin görevlerine kisaca bir bakacak olursak :

**getFD()** : dosya (File) tanimini verir

**getFilePointer()** : Dosyanin (File) o anda hangi dosya referansini gösterdigini belirtir.

**length()** : Dosyadaki toplam referans sayisini verir.

**read()** : Byte (char degiskeni karsiligi) olarak dosyadan bilgiyi okur (bir byte).

**read(byte[])** : Byte (char degiskeni karsiligi) olarak dosyadan bilgiyi okur (byte[] degiskeninin boyu kadar) ve byte degiskenine aktarir.

**read(byte[], int baslangiç\_indisi, int indis\_boyutu)** : Byte (char degiskeni karsiligi) olarak dosyadan bilgiyi okur (byte[] degiskeninin boyu kadar, baslangiç indisinden baslayarak, indis\_boyutu uzunlugunda) ve byte degiskenine aktarir.

**readBoolean()** : boolean degisken okur

**readByte()** : dosyadan integer tipi byte degisken (isaretli 8-bit ) okur. **readChar()** : Dosyadan Unicode karekter (character) okur. **readDouble()** : Dosyadan double degiskeni okur.

**readFloat()** : Dosyadan float degiskeni okur.

**readFully(byte[] b)** : dosyadan byte olarak okur

**readFully(byte[] b, int baslangiç\_indisi, int toplam\_boy)** : dosyadan bte olarak sadece byte boyutlu degiskeninin baslangic\_indisi’nden baslamak üzere taplam\_boy kadar kismini okur.

**readInt()** : Dosyadan int degiskeni okur.

**readLine()** : Dosyadan “\n” = yeni satir isaretini veya “\r” satirbasi isaretini veya herikisini birden arka arkaya görene kadar yazilan her seyi bütün bir satir olarak okur.

**readLong()**: Dosyadan Long (Long integer) degiskeni okur. **readShort():** Dosyadan Short (Short integer) degiskeni okur. **readUnsignedByte()**: Isaretsiz Byte degeri okur. **readUnsignedShort():** Isaretsiz 16 bitlik Short integer degeri okur. **readUTF()**: UTF stringi okur

**seek(long)** : indisi herhangi bir bilgi gurubu indisine ayarlar.

**skipBytes(int n)** : n bit input degerini okumadan atlar

Asagidaki tanimlar read tanimlarina parelel olan write terimleridir.

**write(byte[])** : byte boyutlu degiskenini yazar.

**write(byte[],int baslangiç\_indisi, int toplam\_boy)** : dosyaya byte olarak sadece byte boyutlu degiskeninin baslangic\_indisi’nden baslamak üzere taplam\_boy kadar kismini yazar.

**write(int)** : dosyaya byte boyutlu degiskenini yazar **writeBoolean(boolean)** : dosyaya boolean boyutlu degiskenini yazar **writeByte(int)** : dosyaya int girdisini byte olarak yazar

**writeBytes(String)** : dosyaya String girdisini byte boyutlu degiskeni olarak yazar

**writeChar(int)** : dosyaya int girdisini Char olarak yazar

**writeChars(String)** : dosyaya string girdisini char boyutlu degiskeni olarak yazar

**writeDouble(double)** : dosyaya double girdisini double olarak yazar.

**writeFloat(float)** : dosyaya float girdisini float olarak yazar

**writeInt(int)** : dosyaya int girdisini int olarak yazar.

**writeLong(long)** : dosyaya long(integer) girdisini long(integer) olarak yazar **writeShort(int)** : dosyaya integer gidisini short(integer) olarak yazar. **writeUTF(String) : dosyaya String girdisini UTF string olarak yazar.**

RandomAccessFile sinifinin tanimindan da görüldügü gibi b sinif DataOutput ve DataInput siniflarini implement eder. DataOutput ve DataInput siniflarinini tanimi :

public interface DataOutput

{

public abstract void write(int b) throws IOException public abstract void write(byte b[]) throws IOException

public abstract void write(byte b[],int off,int len) throws IOException public abstract void writeBoolean(boolean v) throws IOException public abstract void writeByte(int v) throws IOException

public abstract void writeShort(int v) throws IOException public abstract void writeChar(int v) throws IOException public abstract void writeInt(int v) throws IOException public abstract void writeLong(long v) throws IOException public abstract void writeFloat(float v) throws IOException

public abstract void writeDouble(double v) throws IOException public abstract void writeBytes(String s) throws IOException public abstract void writeChars(String s) throws IOException public abstract void writeUTF(String str) throws IOException

}

public interface DataInput

{

public abstract void readFully(byte b[]) throws IOException

public abstract void readFully(byte b[],int off,int len) throws IOException public abstract int skipBytes(int n) throws IOException

public abstract boolean readBoolean() throws IOException public abstract byte readByte() throws IOException

public abstract int readUnsignedByte() throws IOException public abstract short readShort() throws IOException

public abstract int readUnsignedShort() throws IOException public abstract int readInt() throws IOException

public abstract long readLong() throws IOException public abstract float readFloat() throws IOException public abstract double readDouble() throws IOException public abstract String readLine() throws IOException public abstract String readUTF() throws IOException

}

RandomAccessFile Çok daha çabuk ulasim olanaklari yaratir. Raslantisal Ulasim dosyasi örnegi olarak asagidaki paketi veriyoruz. Kayit sinifi tek bir kayitin yazilip okunabilmesi için gerekli olan bilgileri içeriyor. Kayitlarin hesap numarasi, isim, soyisim ve hesaptaki paradan olustugunu kabul ediyoruz. Bütün bu kayitlari bir arada yapmak için oku ve yaz metotlari bu sinifin içindr tanimlanmistir. Raslantisal ulasim kanalinin dosyaya açilmasi için :

### try{

**girdi=new RandomAccessFile("musteri1.dat","rw");**

**} catch(IOException e)**

**{**

**System.err.println("Dosya acma hatasi\n"+e.toString()); System.exit(1);**

**}**

yapisi kullanbilir. Buradaki “rw” yapisi dosyaya hem yazi yazilip hem okunabilecegini belirtir. “r” sadece okumak için “w” sadece yazmak için kullanilir.

**RandomAccessFile** içinde yer alan

**close()** metodu RandomAccessFile (Raslantisal Ulasim Dosyasi) kanalini ve ilgili tüm siniflari kapatir. örnek olarak :

### private void closeFile()

**{**

**try{ girdi.close(); System.exit(0);**

**}**

**catch(IOException e)**

**{**

**System.err.println("Error closing filr\n"+e.toString()); System.exit(1);**

**}**

**}**

metotu verilebilir.

Simdi de örnek programin listelerini verelim:

### Program 10.10 Kayit.java raslantisal ulasim dosyasi kayit programi

import java.io.\*; public class Kayit

{

private int hesap; private String soyIsim; private String Isim;

private double hesaptakiPara;

public void oku(RandomAccessFile dosya) throws IOException

{

//RandomAccessFile = Raslantisal Ulasim dosyasi hesap=dosya.readInt();

char first[]=new char[15]; for(int i=0;i<first.length;i++)

{ first[i]=dosya.readChar(); } Isim=new String(first);

char last[]=new char[15]; for(int i=0;i<first.length;i++)

{ last[i]=dosya.readChar(); } soyIsim=new String(last); hesaptakiPara=dosya.readDouble();

}

public void yaz(RandomAccessFile dosya) throws IOException

{

StringBuffer buf; dosya.writeInt(hesap); if(Isim!=null)

buf=new StringBuffer(Isim); else

buf=new StringBuffer(15); buf.setLength(15); dosya.writeChars(buf.toString()); if(soyIsim!=null)

buf=new StringBuffer(soyIsim); else

buf=new StringBuffer(15);

buf.setLength(15); dosya.writeChars(buf.toString()); dosya.writeDouble(hesaptakiPara);

}

public void yazhesap(int a) {hesap = a;} public int okuhesap() {return hesap;} public void yazIsim(String f){Isim=f;} public String okuIsim() {return Isim;}

public void yazsoyIsim(String f){soyIsim=f;} public String okusoyIsim() {return soyIsim;}

public void yazhesaptakiPara(double b) {hesaptakiPara=b;} public double okuhesaptakiPara() {return hesaptakiPara;} public static int boyut() {return 72;}

}

### Program 10.11 RaslantisalUlasimDosyasiYarat.java programi

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import Kayit;

public class RaslantisalUlasimDosyasiYarat

{

private Kayit hesapDosyasi; private RandomAccessFile girdi;

public RaslantisalUlasimDosyasiYarat()

{

hesapDosyasi = new Kayit(); try{

girdi = new RandomAccessFile("musteri1.dat","rw"); for(int i=0;i<100;i++)

{ hesapDosyasi.yaz(girdi); }

} catch(IOException e)

{

System.err.println("Dosya acma hatasi\n"+e.toString()); System.exit(1);

}

}

public static void main(String args[])

{

RaslantisalUlasimDosyasiYarat H= new RaslantisalUlasimDosyasiYarat();

}

}

### Program 10.12 RaslantisalUlasimDosyasiYaz.java programi

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import Kayit;

public class RaslantisalUlasimDosyasiYaz extends Frame implements ActionListener

{

private TextField hesapAlani,isimAlani,soyisimAlani,hesaptakiParaAlani; private Button birsonraki,kapat;

private RandomAccessFile girdi; private Kayit hesapDosyasi;

public RaslantisalUlasimDosyasiYaz()

{

super("Tesadufi ulasim dosyasina yaz");

hesapDosyasi=new Kayit(); try{

girdi=new RandomAccessFile("musteri1.dat","rw");

} catch(IOException e)

{

System.err.println("Dosya acma hatasi\n"+e.toString()); System.exit(1);

}

setSize(300,150);

setLayout(new GridLayout(5,2)); add(new Label("Hesap numarasi:")); hesapAlani=new TextField(); add(hesapAlani);

add(new Label("isim : ")); isimAlani=new TextField(20); add(isimAlani);

add(new Label("Soyisim : ")); soyisimAlani=new TextField(20); add(soyisimAlani);

add(new Label("Hesaptaki para : ")); hesaptakiParaAlani=new TextField(20); add(hesaptakiParaAlani); birsonraki=new Button("Gir"); birsonraki.addActionListener(this); add(birsonraki);

kapat=new Button("cikis"); kapat.addActionListener(this); add(kapat);

setVisible(true);

}

public void addKayit()

{

int accountNumber=0;

Double d;

if(!hesapAlani.getText().equals(""))

{

try{ accountNumber=Integer.parseInt(hesapAlani.getText()); if(accountNumber0 && accountNumber <=100)

{

hesapDosyasi.yazhesap(accountNumber); hesapDosyasi.yazIsim(isimAlani.getText()); hesapDosyasi.yazsoyIsim(soyisimAlani.getText()); d=new Double(hesaptakiParaAlani.getText()); hesapDosyasi.yazhesaptakiPara(d.doubleValue()); girdi.seek((long)(accountNumber-1)\*Kayit.boyut()); hesapDosyasi.yaz(girdi);

}

hesapAlani.setText(""); isimAlani.setText(""); soyisimAlani.setText(""); hesaptakiParaAlani.setText("");

}

catch(NumberFormatException nfe)

{

System.err.println("Hesap numarasi tamsayi degisken olamalidir");

}

catch(IOException io)

{

System.err.println("Dosyaya yazarken hata olustu\n"+io.toString());

System.exit(1);

}

}

}

public void actionPerformed(ActionEvent e)

{

addKayit(); if(e.getSource()==kapat)

{

try{ girdi.close();} catch(IOException io)

{

System.err.println("Dosya kapatilamadi\n"+io.toString());

}

System.exit(0);

}

}

public static void main(String args[])

{

new RaslantisalUlasimDosyasiYaz();

}

}

### Program 10.13 RaslantisalUlasimDosyasiOku.java programi

import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import Kayit;

public class RaslantisalUlasimDosyasiOku extends Frame implements ActionListener

{

private TextField hesapAlani,isimAlani,soyisimAlani,hesaptakiParaAlani; private Button birsonraki,kapat;

private RandomAccessFile girdi; private Kayit hesapDosyasi;

public RaslantisalUlasimDosyasiOku()

{

super("Musteri dosyasini oku"); try{

girdi=new RandomAccessFile("musteri1.dat","r");

} catch(IOException e)

{ System.err.println("Dosya acilamadi\n"+e.toString()); System.exit(1);

}

hesapDosyasi=new Kayit(); setSize(300,150);

setLayout(new GridLayout(5,2)); add(new Label("Hesap numarasi :")); hesapAlani=new TextField(); add(hesapAlani);

add(new Label("isim : ")); isimAlani=new TextField(20); add(isimAlani);

add(new Label("soyisim : ")); soyisimAlani=new TextField(20); add(soyisimAlani);

add(new Label("Hesaptaki para : ")); hesaptakiParaAlani=new TextField(20); add(hesaptakiParaAlani);

birsonraki=new Button("bir sonraki hesap");

birsonraki.addActionListener(this); add(birsonraki);

kapat=new Button("cikis"); kapat.addActionListener(this); add(kapat);

setVisible(true);

}

public void okuKayit()

{

int hesap; String ilk,son; double balance;

{

try{ do{

hesapDosyasi.oku(girdi);

} while(hesapDosyasi.okuhesap()==0); hesapAlani.setText(String.valueOf(hesapDosyasi.okuhesap())); isimAlani.setText(String.valueOf(hesapDosyasi.okuIsim())); soyisimAlani.setText(String.valueOf(hesapDosyasi.okusoyIsim())); hesaptakiParaAlani.setText(String.valueOf(hesapDosyasi.okuhesaptakiPara()));

}

catch(EOFException eof)

{

closeFile();

}

catch(IOException io)

{

System.err.println("Dosyayi okurken hata olustu\n"+io.toString()); System.exit(1);

}

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==birsonraki) okuKayit();

else closeFile();

}

private void closeFile()

{

try{ girdi.close(); System.exit(0);

}

catch(IOException e)

{

System.err.println("Error closing filr\n"+e.toString()); System.exit(1);

}

}

public static void main(String args[])

{

new RaslantisalUlasimDosyasiOku();

}

}

### 10003.JPG



**Sekil 10.3 RaslantisalUlasimDosyasiOku.java Frame çiktisi**

Yukardaki programin ilginç bir yönü de **Frame** sinifini kullanmis olmasidir. Frame ve swing esiti JFrame konsol ortaminda kullanilabilen grafik ortami programlamasidir. Temel olarak applet yapilarinda kullanilan heryerde Frame de kullanilabilir. Frame **main** programdan baslatilir. Html dosyasindan baslatilmaz.

### DOSYA SIKISTIRILMASI (GZIP,GUNZIP,ZIP,UNZIP)

java.util.zip paketi veri sikistirmaya yarayan dosyalariiçerir. Günümüz bilgisayar kullaniminda GZIP ve ZIP adi verilen bu yapilar ZLIB sikistirma algoritmasinda tanimlanmistir. Bu algoritm RFC 1950,RFC 1951 ve RFC 1952 dökümanlarinda tanimlanmistir. Bu dökümanlara <http://www.faqs.org/rfcs>adresinden ulasilabilir.

Kullanma açisindan bu dökümanlara ihtiyaciniz yoktur. Gerekli programlar java zip paketinde bulunmaktadir.

GZIP tek bir dosyayi sikistirarak isim.gz ismiyle sikistirilmis dosya olusturur. Bunun için GZIPOutputStream dosyasina bir FileOutptStream açmamiz kafidir. Program 10.4 den de görülecegi gibi

### GZIPOutputStream zipout;

try {

### FileOutputStream out = new FileOutputStream(zipname); zipout = new GZIPOutputStream(out);

}

catch (IOException e) {

JOptionPane.showInputDialog(null,"Dosya olusturma hatasi : " + zipname + "."); return;

}

deyimi dosyanin gzip olarak yazilmasi için gerekli kanali olusturur.

### Program 10.14 Gzip.java dosya sikistirma programi

//dosya: GZip.java import java.io.\*; import java.util.zip.\*;

import javax.swing.JOptionPane;

public class GZip {

public static int sChunk = 8192;

public static void main(String[] args) { if (args.length != 1) {

JOptionPane.showInputDialog(null,"KULLANIM: java GZip girisdosyasi"); return;

}

// çikti doyasi olustur.

String zipname = args[0] + ".gz"; GZIPOutputStream zipout;

try {

FileOutputStream out = new FileOutputStream(zipname); zipout = new GZIPOutputStream(out);

}

catch (IOException e) {

JOptionPane.showInputDialog(null,"Dosya olusturma hatasi : " + zipname + "."); return;

}

byte[] buffer = new byte[sChunk];

// dosyayi sikistir try {

FileInputStream in = new FileInputStream(args[0]); int length;

while ((length = in.read(buffer, 0, sChunk)) != -1) zipout.write(buffer, 0, length);

in.close( );

}

catch (IOException e) {

JOptionPane.showInputDialog(null,"Dosya sikistirma hatasi : " + args[0] + ".");

}

try { zipout.close( ); } catch (IOException e) {}

}

}

GZIP isleminin tersini yapmak içinse parelel olarak :

### GZIPInputStream zipin;

try {

### FileInputStream in = new FileInputStream(zipname); zipin = new GZIPInputStream(in);

}

catch (IOException e) {

JOptionPane.showInputDialog(null,"Dosya açma hatasi : " + zipname + "."); return;

}

yapisi kullanilir.

Programlarin çalismasi için java isim deyiminden sonra dostya ismini vermek kafidir.

C:\co\java\prog>java Gzip a.dat veya

C:\co\java\prog >java Gunzip a.dat.gz gibi.

### Program 10.15 GUnzip.java dosya sikistirilmis dosyayi açma programi

//dosya : GUnzip.java import java.io.\*; import java.util.zip.\*;

import javax.swing.JOptionPane; public class GUnzip {

public static int sChunk = 8192; public static void main(String[] args) { if (args.length != 1) {

JOptionPane.showInputDialog(null,"KULLANIM: java GUnzip girisdosyasi"); return;

}

// girdi dosyasini incele String zipname, source;

if (args[0].endsWith(".gz")) { zipname = args[0];

source = args[0].substring(0, args[0].length( ) - 3);

}

else {

zipname = args[0] + ".gz";

source = args[0];

}

GZIPInputStream zipin; try {

FileInputStream in = new FileInputStream(zipname); zipin = new GZIPInputStream(in);

}

catch (IOException e) {

JOptionPane.showInputDialog(null,"Dosya açma hatasi : " + zipname + "."); return;

}

byte[] buffer = new byte[sChunk];

// dosyayi aç (decompress) try {

FileOutputStream out = new FileOutputStream(source); int length;

while ((length = zipin.read(buffer, 0, sChunk)) != -1) out.write(buffer, 0, length);

out.close( );

}

catch (IOException e) {

JOptionPane.showInputDialog(null,"Dosya sikistirilmasi açilma hatasi : " + args[0] + ".");

}

try { zipin.close( ); } catch (IOException e) {}

}

}

ZIP ve UNZIP biraz daha kompleks bir yapiya sahiptir. Çünki bu proseste direktory içindeki dosyalarin açilma islemi söz konusudur.

Prosesi izah etmek için önce ZIP islemi yapan bir programa göz atalim :

### Program 10.16 zip.java dosya sikistirma programi

// dosya : zip.java

// Turhan Çoban 24.2.2001 import java.io.\*;

import java.util.\*; import java.util.zip.\*;

import javax.swing.JOptionPane;

public class zip {

public static int sChunk = 8192;

public static String[] dosyalistesi(String p)

{

File path=new File(p); if(path.isDirectory())

{

Vector x=new Vector(); File files[]; files=path.listFiles(); int n=files.length;

int n1=0;

for(int i=0;i<n;i++)

{

if(!files[i].isDirectory())

{

x.addElement(files[i].toString()); n1++;

}

}

String s1[]; s1=new String[n1];

Enumeration nn=x.elements(); int k=0; while(nn.hasMoreElements())

{

s1[k++]=(String)nn.nextElement();

}

return s1;

}

else

{

String[] s1 = new String[1]; s1[0]=p;

return s1;

}

}

public static void main(String[] args) { String str="";

FileOutputStream out; ZipOutputStream zout; String zipname, source; String s="KULLANIM: \n"+

"java zip zipdosyasi dosya(veyadirectory)ismi1 dosya(veyadirectory)ismi1...\n"+ " zip dosyasi yarat ve dosya(veya directorylerdeki dosyalari) sikistir "+

" alt direktoryler atlanmaktadir tekrar tanimla";

// anahtari kontrol et if(args.length<1 )

{

String s1="zipdosyasi isim.zip tanimlanmadi lütfen argümanlari giriniz\n"+s; JOptionPane.showMessageDialog(null,s1);

System.exit(0); return;

} // girdi dosyasini incele else if(args.length<2)

{

String s1="sikistirilacak dosya(veya direktory) isimleri tanimlanmadi\n"+ " lütfen argümanlari giriniz\n"+s; JOptionPane.showMessageDialog(null,s1);

System.exit(0); return;

}

if (args[0].endsWith(".zip")) { zipname = args[0];

source = args[0].substring(0, args[0].length( ) - 3);

}

else {

zipname = args[0] + ".zip"; source = args[0];

}

//zip kanalini tanimla try {

out = new FileOutputStream(zipname); zout=new ZipOutputStream(out);

}

catch (IOException e) {

JOptionPane.showMessageDialog(null,"Dosya olusturma hatasi : " + zipname + "."); return;

}

int k=1; while(k<args.length)

{

try

{

String ss[]; ss=dosyalistesi(args[k]); int j=0;

byte[] buffer=new byte[sChunk]; while(j<ss.length)

{

ZipEntry giris=new ZipEntry(ss[j]); zout.putNextEntry(giris);

try {

FileInputStream in=new FileInputStream(ss[j]); int length; while((length=in.read(buffer,0,sChunk))!=-1)

zout.write(buffer,0,length); in.close();

} //try sonu

catch (IOException e)

{

JOptionPane.showMessageDialog(null,"Dosya sikistirma hatasi : " +ss[j] );

} //catch sonu }

str+="ss["+j+"] = "+ss[j]+" "+giris+" iceri girdi\n"; j++;

}//while sonu

} //try sonu catch(IOException io) {} k++;

} //while(k< sonu try{

zout.close();

out.close();

}

catch(IOException e){} JOptionPane.showMessageDialog(null,str); System.exit(0);

}

}

örnek program girdisi :

### java zip zip1.zip c:\co\java\prog\a d.dat c:\co\java\prog\a\b 10004.JPG

**Sekil 10.3 zip.java programini çiktisi**

simdi de ZIP yapilmis bir dosyayi açan Unzip.java programina bir göz atalim :

### Program 10.16 Unzip.java dosya sikistirilmis dosyayi açma programi

//dosya : Unzip.java import java.io.\*; import java.util.zip.\*;

import javax.swing.JOptionPane;

public class Unzip {

public static int sChunk = 8192;

public static void main(String[] args) { String zipname, source;

String s="";

if (args.length != 1) {

JOptionPane.showMessageDialog(null,"KULLANIM: java Unzip girisdosyasi"); System.exit(0);

return;

}

// girdi dosyasini incele

if (args[0].endsWith(".zip")) { zipname = args[0];

source = args[0].substring(0, args[0].length( ) - 3);

}

else {

zipname = args[0] + ".zip"; source = args[0];

}

s+="zip dosyasi : "+zipname+" içideki sikistirilmis \n"; try {

FileInputStream in = new FileInputStream(zipname); ZipInputStream zin=new ZipInputStream(in);

byte[] buffer = new byte[sChunk];

// dosyayi aç (decompress try{

ZipEntry z; do

{

z=zin.getNextEntry(); try {

FileOutputStream out = new FileOutputStream(z.getName()); int length;

while ((length = zin.read(buffer, 0, sChunk)) != -1) out.write(buffer, 0, length);

out.close( );

}

catch (IOException e) { JOptionPane.showMessageDialog(null, "Dosya sikistirma hatasi : " + args[0] + "."); System.exit(0);

}

s+=" "+z.getName()+" \n";

}while(z!=null);

} catch(NullPointerException npe)

{ s+="dosyalari açildi";JOptionPane.showMessageDialog(null,s); System.exit(0);

}

try { zin.close(); } catch (IOException e) {}

}

catch(IOException e)

{

JOptionPane.showMessageDialog(null,"Dosya açilma hatasi : " + zipname); System.exit(0);

}

}

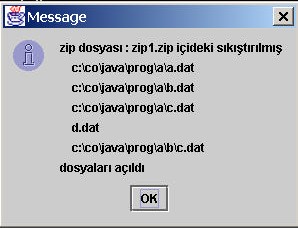
}

programi

### java unzip zip1.zip

komutu kullanarak çalistirirsak :

### 10005.JPG



**Sekil 10.4 unzip.java programini çiktisi**

Çiktisini elde ederiz.

Burada zip dosyasina ulasmak için :

### FileInputStream in = new FileInputStream(zipname); ZipInputStream zin=new ZipInputStream(in);

Deyimini kullandik. Zipli dosyayi okumak için ise :

### ZipEntry z;

do

{

### z=zin.getNextEntry();

try {

FileOutputStream out = new FileOutputStream(z.getName()); int length;

### while ((length = zin.read(buffer, 0, sChunk)) != -1)

out.write(buffer, 0, length); out.close( );

}

gurubunu kullandik. Burada temel olarak iki yeni sinif kullanildi. Bunlardan birincisi ZipInputStream, digeri ise ZipEntry siniflari idi. Simdi bu iki sinifin tanimlarina göz atalim :

public class **ZipInputStream** extends InflaterInputStream implements ZipConstants { private ZipEntry entry;

private CRC32 crc; private long remaining; private byte[] tmpbuf ;

private static final int STORED; private static final int DEFLATED; private boolean closed ;

private boolean entryEOF;

private void ensureOpen() throws IOException; public ZipInputStream(InputStream in) ;

public ZipEntry getNextEntry() throws IOException; public void closeEntry() throws IOException;

public int available() throws IOException ;

public int read(byte[] b, int off, int len) throws IOException; public long skip(long n) throws IOException ;

public void close() throws IOException;

private ZipEntry readLOC() throws IOException;

private static String getUTF8String(byte[] b, int off, int len) throws IOException; protected ZipEntry createZipEntry(String name) throws IOException;

private void readEnd(ZipEntry e) throws IOException;

private void readFully(byte[] b, int off, int len) throws IOException; private static final int get16(byte b[], int off);

private static final long get32(byte b[], int off);

}

ZipOutputStream’in tanimi ise :

public class **ZipOutputStream** extends DeflaterOutputStream implements ZipConstants { private ZipEntry entry;

private Vector entries; private Hashtable names ; private CRC32 crc; private long written; private long locoff ; private String comment; private int method ; private boolean finished;

private boolean closed = false;

private void ensureOpen() throws IOException public static final int STORED ;

public static final int DEFLATED;

public ZipOutputStream(OutputStream out) public void setComment(String comment) public void setMethod(int method

public void setLevel(int level)

public void putNextEntry(ZipEntry e) throws IOException public void closeEntry() throws IOException

public void close() throws

private void writeLOC(ZipEntry e) throws IOException private void writeEXT(ZipEntry e) throws IOException private void writeCEN(ZipEntry e) throws IOException private void writeEND(long off, long len) throws IOException private void writeShort(int v) throws IOException

private void writeInt(long v) throws IOException

private void writeBytes(byte[] b, int off, int len) throws IOException private static byte[] getUTF8Bytes(String s)

}

seklindedir. Ayrica zip dosyasinin içindeki elemanlar ZipEntry sinifinda tanimlanmaktadir. ZipEntry sinifi : public class ZipEntry implements ZipConstants, Cloneable

{

String name; long time; long crc ; long size; long csize; int method; byte[] extra;

String comment; int flag;

int version;

long offset;

public static final int STORED ; public static final int DEFLATED; private static native void initIDs(); public ZipEntry(String name) public ZipEntry(ZipEntry e ZipEntry(String name, long jzentry ZipEntry(long jzentry)

public String getName()

public void setTime(long time) public long getTime()

public void setSize(long size) public long getSize()

public long getCompressedSize()

public void setCompressedSize(long csize) public void setCrc(long crc)

public long getCrc()

public void setMethod(int method) public int getMethod()

public void setExtra(byte[] extra) public byte[] getExtra()

public void setComment(String comment) public String getComment()

public boolean isDirectory() public String toString()

private static long dosToJavaTime(long dtime) private static long javaToDosTime(long time) public int hashCode()

public Object clone()

}

seklinde tanimlanmistir. Burada su ana kadar verdigimiz sinif tanimlari ile ilgili sunu söyliyelim. Tanimlamalar sadece sinif ve degisken adlarini vermektedir, gerçek programi açtiginizda göreceginiz koda benzemezler sadece metod isimlerini tasirlar.

### ALISTIRMALAR

1. **H9O1.java** programini inceleyiniz. Bu programda sayilar dosyadan okunup ortalamalari hesaplanmaktadir.

### Program 10.14 H9O1.java programi

import java.io.\*; import Text;

// (c) Ali SAYGIVAR public class H9O1

{

public static void main(String[] args) throws IOException

{

DataInputStream input; int[] i= new int[100]; String s1;

Text cin= new Text();

System.out.print("Lutfen dosya ismini giriniz: "); s1 = cin.readString();

File myfile = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(myfile)); int toplam=0;

boolean EOF=false; int j=0;

while (!EOF)

{

try

{

i[j]= Text.readInt(b); toplam+=i[j];

++j;

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

System.out.println("Dosyanin icindeki sayilarin adedi: "+j); System.out.println("Ortalama: "+(double)toplam/20);

} //main

} //class

1. **Karsilastir.java** verilen iki sosyayi karsilastirmaktadir. Dosya isimleri program kullanimi sirasinda verilecektir. Örnek :

java Karsilastir a.dat a1.dat

### Program 10.15 Karsilastir.java programi

import java.io.\*;

public class Karsilastir {

//——————————————————————————————————————

// RandomAccessFile örnegi

// program main()

//

// disaridan iki tane dosya ismi giriniz

// dosyalari açar ve içindekileri yükler

// dosyalari degerlendirir (karsilastirir)

// dosyalari kapatir

//———————————————————————————————————

public static void main (String args[]) { RandomAccessFile fh1 = null; RandomAccessFile fh2 = null;

int bufsize; // en küçük dosyanin byutu long filesize1 = -1;

long filesize2 = -1; byte buffer1[]; byte buffer2[];

// disaridan okunan veriyi kontrol et

if (args.length == 0 || args[0].equals("?") ) {

System.err.println ("Random Acsess dosya Karsilastirma : "); System.err.println ("-----");

System.err.println ("KULLANIM : java Karsilastir <dosya1> <dosya2> | ?"); System.err.println ();

System.exit(0);

}

// birinci dosyayi okumak için aç

try {

fh1 = new RandomAccessFile(args[0], "r"); filesize1 = fh1.length();

} catch (IOException ioErr) {

System.err.println ("Dosya bulunamadi : " + args[0]); System.err.println (ioErr);

System.exit(100);

}

// ikinci dosyayi okumak için aç

try {

fh2 = new RandomAccessFile (args[1], "r"); filesize2 = fh2.length();

} catch (IOException ioErr) {

System.err.println ("Dosya bulunamadi :" + args[1]); System.err.println (ioErr);

System.exit(100);

}

if (filesize1 != filesize2) {

System.out.println ("Dosya boyutlari ayni degil! ");

System.out.println ("Dosya 1 : " + args[0] + "boyutu " + filesize1 + " bytes"); System.out.println ("Dosya 2 : " + args[1] + "boyutu " + filesize2 + " bytes");

}

// iki dosyaninda içerigini alabilecek bir yer ayir

bufsize = (int) Math.min(filesize1, filesize2); buffer1 = new byte [bufsize];

buffer2 = new byte [bufsize];

// dosyalari toptan oku try {

fh1.readFully (buffer1, 0, bufsize); fh2.readFully (buffer2, 0, bufsize);

// simdi gelelim asil isimize...

for (int i = 0; i < bufsize; i++) { if (buffer1[i] != buffer2[i]) {

System.out.println ("dosyalar index " + i+"de degisiyor"); break;

}

}

} catch (IOException ioErr) {

System.err.println ("HATA: dosyalari icelerken bir hata olustu"); System.err.println (ioErr.toString());

} finally { try {

fh1.close();

fh2.close();

} catch (IOException ignored) {}

}

}

}

1. **File sinifi uygulamasi :** dir.java programi istenilen directory ve alt dirctory dosyalarini sirayla listeler.

### Program 10.16 dir.java programi

**//------------------------------------**

import java.io.\*; import java.util.\*; public class dir {

static int indentLevel = -1; dir (String path) {

listPath (new File (path));

}

void listPath (File path) {

File files[]; // directory'dekidosyalarin listesi indentLevel++; // sayiliyor...

// bu directory'deki dosyalarin listesini hazirla files = path.listFiles();

// Dosya isimlerini siraya sok Arrays.sort (files);

for (int i=0, n=files.length; i < n; i++) {

for (int indent=0; indent < indentLevel; indent++) { System.out.print(" ");

}

System.out.println(files[i].toString());

if (files[i].isDirectory()) { listPath(files[i]);

}

}

indentLevel--;

}

public static void main (String args[]) { new dir(args[0]);

}

}

java dir c:\Driver\hpscanner\disk1 komutu sonucu :

C:\Drive\hpscanner\disk1\!hwp2002.cfg C:\Drive\hpscanner\disk1\!hwp2080.cfg [C:\Drive\hpscanner\disk1\@621f.adf](mailto:\Drive\hpscanner\disk1\@621f.adf) [C:\Drive\hpscanner\disk1\@631f.adf](mailto:\Drive\hpscanner\disk1\@631f.adf) C:\Drive\hpscanner\disk1\\_inst32i.ex\_ C:\Drive\hpscanner\disk1\\_isdel.exe

1. zip.java sinifi içerisindeki public static String[] dosyalistesi(String p) metodu alt direktoryleri okumadan atlamaktadir. Alt direktoryleri de okuyup zipleyecek yeni bir zip1.java programi yaziniz.
2. Bu problemde gerçek bir bilgisayar programlama örnegine göz atacagiz. Program 10.8 de Atom.java programina bakmistik, bu problemde Atom.java programi kullanilarak yazilmis olan Gas.java programina yer verecegiz. Bu program ideal gazlarin termodinamik özelliklerin hesaplamak amaciyla yazilmistir. Program ideal

gazlarin verilerini Gas.txt dosyasindan akumaktadir. Gaz isimleri kurucu metod tarafindan aranan gaz isimlerine uyum sagladiginda program geriye kalan veriyi okumaktadir. Gazlar atomlardan yapildigindan veride okunan Atom isimlerine göre Atom özellikleri de bu sinifin kurucu metodunda Atom.txt dosyasindan okunmakta ve degerlendirilmektedir. Program çiktisi almak için GasPropertySWF.java programi da ayrica verilmistir. Burada verilen kod özellikle bu kitap için yazilmamistir. Uluslararasi projelerde kullanildigindan açiklamalari ingilizcedir.

### Program 10.16 Gas.java programi

//======================================================

// Thermodynamic Package in java

// Class Gas Properties of perfect gases

// Dr. Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Research Institute

// email : [turhan@mam.gov.tr](mailto:turhan@mam.gov.tr)

// File Name : Gas.java

// This file contains the Gas class

// this class sets basic properties of perfect gases

// required data is read from Gas.txt

// =====================================================

// Description : This file contains the gas class

// class gas calculates thermophysical properties of

// perfect gasses

// following properties can be calculated

// T() : Temperature degree K

// h(T) : enthalpy KJ/kmol

// hf : formation enthalpy KJ/kg

// ht(T) : total enthalpy KJ/kg (h+hf)

// M : molar mass kg/kmol

// HT(T) : total enthalpy KJ : M\*ht(T)

// P() : presuure bar

// s(T,P) : entropy KJ/kmol K

// Cp(T) : specific heat at constant pressure KJ/kmol K

// Cv(T) : specific heat at constant volume KJ/kg K

// gamma(T): adiabatic constant Cp/Cv

// u(T) : Internal energy KJ/kmol

// c(T) : speed of sound m/s

// vis(T) : viscosity Ns/m^2

// k(T) : thermal conductivity KJ/kg K

// DATA FILE DEFINATION

// gas datas are written in the data file "Gas.txt"

// if gas data is not given in the data file, it can be curve fitted

// and added to the data file. Additional curve fitting programs supplied

// in Numerical Analysis package. Each data has the following form :

//------------------

// gasName

// n\_equation M h0 hf sf

// xa[0] xb[0] xc[0] xd[0] tl[0] th[0]

// .........

// xa[n\_equation-1] xb[n\_equation-1]......th[n\_equation-1]

// n\_vis

// xvis[0]

// .........

// xvis[n\_vis-1]

// n\_k

// xk[0]

// .........

// xk[n\_k-1]

//-------------------

// unit of the xa : kcal/kmol

// note : if any curvefitting applied for a new gas temperature values

// in K nad enthalpy values in the unit of Kcal.kmol should be supply

// for the Cp curve fitting

//============================================================

// VARIABLE IDENTIFICATION

// all the variables that type is not defined is a double variable

// PROTECTED VARIABLES :

// xa,xb,xc,xd ,tl,th : double pointers. This values used to calculate

// specific heat at constant pressure from the following equation :

// Cp(T) = xa[i]+xb[i]\*1e-3\*T+xc[i]\*1.0e5/T^2+xd[i]\*1e-6\*T^3

// where tl[i] <= T <= th[i]

// n\_equation : number of equations (xa,xb,xc,xd,tl,th) for a gas

// xvis : real pointers to define viscosity according to formula :

// vis(T)=sum(xvis(i)\*T^i) , for(i=0;i<n\_vis;i++)

// n\_vis : number of coefficients in polynomial viscosity curve fitting

// xk :real pointers to define thermal conductivity according to formula

// k(t)=sum(xk(i)\*T^i) , for(i=0;i<n\_k;i++)

// PUBLIC VARIABLES :

// gasName : name of the gas example : H2O :variable class str

// (class str is defined at file cstr.h and str.cpp, written by Timotyhy A. Budd)

// M : mol number of the gas example mol number of H2O is 18.016 kg/kmol

// h0 : value of enthalpy at 298 K in the unit of Kcal/kmol

// hf : formation enthalpy at 298 K in the unit of Kcal/kmol

// sf : value of enthalpy at 298 K and 1 bar pressure

// N : molar weight of the gas, kmol

// ierror : integer variable, error flag

//============================================================

import java.io.\*; import Text; import Atom;

class Gas

{

double xa[],xb[],xc[],xd[],tl[],th[]; int n\_equation;

int n\_vis; int n\_k;

double xvis[]; double xk[]; int natom;

String gasName;

Atom atomList[];

double M; // molar mass of atom kg/kmol double h0; // enthalpy at T=298 K

double hf; // enthalpy of formation

double sf; // entropy of formation kJ/kmol K double N; // moles of gas kmol

int ierror;

BufferedReader fin;

File gasDir;

// definations of class functions

// constructors

//===================================================================

public Gas()

{

//empty constructor int i;

gasName="\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; natom=1;

atomList=new Atom[natom]; M=0;

N=1.0; h0=0;

hf=0; sf=0;

n\_equation=6; n\_vis=10; n\_k=10;

xa=new double[6]; xb=new double[6]; xc=new double[6]; xd=new double[6]; tl=new double[6]; th=new double[6]; xvis=new double[10]; xk=new double[10]; for(i=0;i<n\_equation;i++)

{xa[i]=0.0;xb[i]=0.0;xc[i]=0.0;xd[i]=0.0;tl[i]=293.0;th[i]=293.0;}

for(i=0;i<n\_vis;i++)

{xvis[i]=0.0;} for(i=0;i<n\_k;i++)

{xk[i]=0.0;}

}

//===================================================================

public Gas(String gName,double Nnew) throws IOException

{

int i;

String aName; double aN; ierror=1; N=Nnew;

String tempGasName=""; try{

fin=new BufferedReader(new FileReader("Gas.txt"));

} catch(IOException e)

{

System.err.println("Error Opening File Gas.dat\n"+e.toString()); System.exit(1);

}

try{ while(fin!=null)

{

tempGasName=Text.readString(fin);

if (tempGasName.equals(gName)) {ierror=0;break;}

}

} catch(EOFException e\_eof)

{

System.out.println("error required gas "+tempGasName+" is not found"); fin.close();return;

}

gasName=gName; natom=Text.readInt(fin); atomList=new Atom[natom]; M=0;

for(i=0;i<natom;i++)

{

aName=Text.readString(fin); aN=Text.readDouble(fin);

atomList[i]=new Atom(aName,aN); M+=atomList[i].mass;

}

n\_equation=Text.readInt(fin); h0=Text.readDouble(fin); hf=Text.readDouble(fin); sf=Text.readDouble(fin); hf\*=4.1868;

xa=new double[n\_equation]; xb=new double[n\_equation]; xc=new double[n\_equation]; xd=new double[n\_equation]; tl=new double[n\_equation]; th=new double[n\_equation];

for(i=0;i<n\_equation;i++)

{

xa[i]=Text.readDouble(fin); xb[i]=Text.readDouble(fin); xc[i]=Text.readDouble(fin); xd[i]=Text.readDouble(fin); tl[i]=Text.readDouble(fin); th[i]=Text.readDouble(fin);

}

n\_vis=Text.readInt(fin); xvis=new double[n\_vis]; for(i=0;i<n\_vis;i++)

{

xvis[i]=Text.readDouble(fin);

}

n\_k=Text.readInt(fin); xk=new double[n\_k]; for(i=0;i<n\_k;i++)

{

xk[i]=Text.readDouble(fin);

}

}

public Gas(String gName) throws IOException

{

double Nnew=1; int i;

String aName; double aN; ierror=1; N=Nnew;

String tempGasName=""; try{

fin=new BufferedReader(new FileReader("Gas.txt"));

} catch(IOException e)

{

System.err.println("Error Opening File Gas.dat\n"+e.toString()); System.exit(1);

}

try{ while(fin!=null)

{

tempGasName=Text.readString(fin);

if (tempGasName.equals(gName)) {ierror=0;break;}

}

} catch(EOFException e\_eof)

{

System.out.println("error required gas "+tempGasName+" is not found"); fin.close();return;

}

gasName=gName;

//fin>>natom; natom=Text.readInt(fin); atomList=new Atom[natom]; M=0;

for(i=0;i<natom;i++)

{

//fin>>aName>>aN; aName=Text.readString(fin); aN=Text.readDouble(fin); atomList[i]=new Atom(aName,aN); M+=atomList[i].mass;

}

//fin>>n\_equation>>h0>>hf>>sf; n\_equation=Text.readInt(fin); h0=Text.readDouble(fin); hf=Text.readDouble(fin); sf=Text.readDouble(fin); hf\*=4.1868;

xa=new double[n\_equation]; xb=new double[n\_equation]; xc=new double[n\_equation]; xd=new double[n\_equation]; tl=new double[n\_equation]; th=new double[n\_equation];

for(i=0;i<n\_equation;i++)

{

xa[i]=Text.readDouble(fin); xb[i]=Text.readDouble(fin); xc[i]=Text.readDouble(fin); xd[i]=Text.readDouble(fin); tl[i]=Text.readDouble(fin); th[i]=Text.readDouble(fin);

}

n\_vis=Text.readInt(fin); xvis=new double[n\_vis]; for(i=0;i<n\_vis;i++)

{

xvis[i]=Text.readDouble(fin);

}

n\_k=Text.readInt(fin); xk=new double[n\_k]; for(i=0;i<n\_k;i++)

{

xk[i]=Text.readDouble(fin);

}

}

public String readGasNames() throws IOException

{

String temp1=""; int natom;

String temp=new String("");

String pgasName; String aName; double aN; double ppercent; int n\_equation; double h0,hf,sf;

double xai,xbi,xci,xdi,tli,thi; int n\_vis,n\_k;

double xvisi,xki; int i;

try{

fin=new BufferedReader(new FileReader("Gas.txt")); try {

while(fin!=null)

{

temp1=temp1+Text.readString(fin)+ " "; natom=Text.readInt(fin); for(i=0;i<natom;i++)

{

aName=Text.readString(fin); aN=Text.readDouble(fin);

}

n\_equation=Text.readInt(fin); h0=Text.readDouble(fin); hf=Text.readDouble(fin); sf=Text.readDouble(fin); for(i=0;i<n\_equation;i++)

{

xai=Text.readDouble(fin); xbi=Text.readDouble(fin); xci=Text.readDouble(fin); xdi=Text.readDouble(fin); tli=Text.readDouble(fin); thi=Text.readDouble(fin);

}

n\_vis=Text.readInt(fin); for(i=0;i<n\_vis;i++)

{

xvisi=Text.readDouble(fin);

}

n\_k=Text.readInt(fin); for(i=0;i<n\_k;i++)

{

xki=Text.readDouble(fin);

}

} //end of while

} catch(EOFException e\_eof) {fin.close();}

}

catch(FileNotFoundException fnfe) {System.out.println("File Gmix.dat not found");} return temp1;

}

// ==============================================================

public Gas(Gas g) throws IOException

{

int i; gasName=g.gasName; natom=g.natom;

atomList=new Atom[natom]; for(i=0;i<natom;i++)

{

atomList[i]=g.atomList[i];

} M=g.M;

N=g.N;

h0=g.h0; hf=g.hf; sf=g.sf;

n\_equation=g.n\_equation; n\_vis=g.n\_vis; n\_k=g.n\_k;

xa=new double[n\_equation]; xb=new double[n\_equation]; xc=new double[n\_equation]; xd=new double[n\_equation]; tl=new double[n\_equation]; th=new double[n\_equation]; xvis=new double[n\_vis]; xk=new double[n\_k]; for(i=0;i<n\_equation;i++)

{xa[i]=g.xa[i];xb[i]=g.xb[i];xc[i]=g.xc[i];xd[i]=g.xd[i];tl[i]=g.tl[i];th[i]=g.th[i];} for(i=0;i<n\_vis;i++)

{xvis[i]=g.xvis[i];} for(i=0;i<n\_k;i++)

{xk[i]=g.xk[i];}

}

//===================================================================

//change/assign new molar mass

// see also \* operator for the similar function public void changeN(double Nnew)

{

N=Nnew;

}

//===================================================================

public String toString()

{

//return the chemical symbol of the gas String s="";

for(int i=0;i<natom;i++) s=s+atomList[i].toString(); return s;

}

//===================================================================

public double vis(double T)

{

// dynamic viscosity of the gas double visg=0;

if(n\_vis!=0.0)

{

visg=xvis[n\_vis-1]; for(int i=n\_vis-2;i>=0;i--)

{ visg=visg\*T+xvis[i]; } visg\*=1.0e-7;

}

else

visg=0; return visg;

}

//===================================================================

public double k(double T)

{

// thermal conductivity of the gas double kg;

if(n\_k!=0.0)

{

int nk=n\_k-1; kg=xk[nk];

for(int i=n\_k-2;i>=0;i--)

{ kg+=kg\*T+xk[i]; }

kg\*=1.0e-3;

}

else

kg=0; return kg;

}

//===================================================================

public double h(double T)

{

// enthalpy KJ/kmol

//integration of function dh=Cp(T)\*dT double hh = - h0;

for(int i=0;i<n\_equation;i++)

{

if(((T>th[i]) && (i== (n\_equation-1) ) )

|| ((T<tl[i]) && (i== 0 ) ) )

{

hh+= xa[i]\*(T- tl[i])

+ xb[i]\*1.0e-3/2.0\*(T\*T-tl[i]\*tl[i])

- xc[i]\*1e5\*(1/T-1/tl[i])

+ xd[i]\*1e-6\*(T\*T\*T-tl[i]\*tl[i]\*tl[i])/3.0;

}

else if((T<= th[i]) && (T> tl[i]))

{

hh+= xa[i]\*(T- tl[i])

+ xb[i]\*1.0e-3/2.0\*(T\*T-tl[i]\*tl[i])

- xc[i]\*1e5\*(1/T-1/tl[i])

+ xd[i]\*1e-6\*(T\*T\*T-tl[i]\*tl[i]\*tl[i])/3.0;

}

else if(T>th[i])

{

hh+= xa[i]\*(th[i]- tl[i])

+ xb[i]\*1.0e-3/2.0\*(th[i]\*th[i] - tl[i]\*tl[i])

- xc[i]\*1e5\*(1/th[i]-1/tl[i])

+ xd[i]\*1e-6\*(th[i]\*th[i]\*th[i]-tl[i]\*tl[i]\*tl[i])/3.0;

}

}

return (hh\*4.1868);

}

//===================================================================

public double ht( double t)

{

return h(t)+hf;

}

//===================================================================

public double H(double t)

{

return h(t)\*N;

}

//===================================================================

public double HT(double t)

{

return ht(t)\*N;

}

//===================================================================

public double u(double T)

{

// internal energy KJ/kmol

// Integration of function du = Cv(T)\*dT return (h(T) - 8.3145\*T);

}

//===================================================================

public double v(double T,double P)

{

// specific volume of the gas m^3/kmol return 8314.5\*T/(P\*1e5);

}

public double v(double T)

{

double P=1.0;

// specific volume of the gas m^3/kmol return 8314.5\*T/(P\*1e5);

}

//===================================================================

public double c(double t)

{

// speed of sound m/s

return Math.sqrt(8314.5/M\*t\*gamma(t));

}

//===================================================================

public double s(double T, double P)

{

//entropy KJ/kmol K

//integration of function

// ds = Cp(T) \* dt/T - R dP/P double ss=sf;

for(int i=0;i<n\_equation;i++)

{

if( ( T > th[i] && i==n\_equation - 1 )

||( T < tl[i] && i==0 ))

{

ss+=xa[i]\*Math.log(T/tl[i])

+ xb[i]\*1.0E-3\*(T-tl[i])

- xc[i]\*1e5/2.0\*(1.0/(T\*T) - 1.0/(tl[i]\*tl[i]))

+ xd[i]\*1e-6/2.0\*(T\*T-tl[i]\*tl[i]);

}

else if((T <= th[i]) && (T > tl[i]))

{

ss+=xa[i]\*Math.log(T/tl[i])

+ xb[i]\*1.0E-3\*(T-tl[i])

- xc[i]\*1e5/2.0\*(1.0/(T\*T) - 1.0/(tl[i]\*tl[i]))

+ xd[i]\*1e-6/2.0\*(T\*T-tl[i]\*tl[i]);

}

else if( T > th[i] )

{

ss+=xa[i]\*Math.log(th[i]/tl[i])

+ xb[i]\*1.0E-3\*(th[i]-tl[i])

- xc[i]\*1e5/2.0\*(1.0/(th[i]\*th[i]) - 1.0/(tl[i]\*tl[i]))

+ xd[i]\*1e-6/2.0\*(th[i]\*th[i] - tl[i]\*tl[i]);

}

}

ss\*=4.1868;

return (ss-8.3145\*Math.log(P));

}

public double S(double T, double P)

{

return s(T,P)\*N;

}

public double s(double T)

{

//entropy KJ/kmol K

//integration of function

// ds = Cp(T) \* dt/T - R dP/P double ss=sf;

for(int i=0;i<n\_equation;i++)

{

if( ( T > th[i] && i==n\_equation - 1 )

||( T < tl[i] && i==0 ))

{

ss+=xa[i]\*Math.log(T/tl[i])

+ xb[i]\*1.0E-3\*(T-tl[i])

- xc[i]\*1e5/2.0\*(1.0/(T\*T) - 1.0/(tl[i]\*tl[i]))

+ xd[i]\*1e-6/2.0\*(T\*T-tl[i]\*tl[i]);

}

else if((T <= th[i]) && (T > tl[i]))

{

ss+=xa[i]\*Math.log(T/tl[i])

+ xb[i]\*1.0E-3\*(T-tl[i])

- xc[i]\*1e5/2.0\*(1.0/(T\*T) - 1.0/(tl[i]\*tl[i]))

+ xd[i]\*1e-6/2.0\*(T\*T-tl[i]\*tl[i]);

}

else if( T > th[i] )

{

ss+=xa[i]\*Math.log(th[i]/tl[i])

+ xb[i]\*1.0E-3\*(th[i]-tl[i])

- xc[i]\*1e5/2.0\*(1.0/(th[i]\*th[i]) - 1.0/(tl[i]\*tl[i]))

+ xd[i]\*1e-6/2.0\*(th[i]\*th[i] - tl[i]\*tl[i]);

}

}

ss\*=4.1868;

return ss;

}

//===================================================================

public double s0(double T)

{

return s(T);

}

//===================================================================

public double g(double T,double P)

{

return h(T)-T\*s(T,P);

}

public double gt(double T,double P)

{

return h(T)+hf-T\*s(T,P);

}

public double gt(double T)

{

return h(T)+hf-T\*s(T,1);

}

public double g(double T)

{

double P=1.0;

return h(T)-T\*s(T,P);

}

public double G(double T,double P)

{

return g(T,P)\*N;

}

public double G(double T)

{

return g(T)\*N;

}

public double GT(double T)

{

return gt(T)\*N;

}

public double GT(double T,double P)

{

return gt(T,P)\*N;

}

//===================================================================

public double g0(double T)

{

return h(T)-T\*s0(T);

}

//===================================================================

public double Cp(double T)

{

//specific heat at constant pressure KJ/kmol K double cp=0.0;

for (int i=0;i<n\_equation;i++)

{

if( ( T > th[i] && i==n\_equation - 1 )

||( T < tl[i] && i==0 ))

{

cp=xa[i]+xb[i]\*1.0e-3\*T+xc[i]\*1.0e5/T/T+xd[i]\*1.0e-6\*T\*T; break;

}

else if((T <= th[i]) && (T > tl[i]) )

{

cp=xa[i]+xb[i]\*1.0e-3\*T+xc[i]\*1.0e5/T/T+xd[i]\*1.0e-6\*T\*T; break;

}

}

return (cp\*4.1868);

}

//===================================================================

public double Cv(double T)

{

//specific heat at constant volume KJ/kmol K double cv;

cv=Cp(T) - 8.3145;

return cv;

}

//===================================================================

public double gamma(double T)

{

//adiabatic constant return Cp(T)/Cv(T);

}

//===================================================================

public double T( char name,double y0,double p)

{

// name can have values h : for enthalpy

// u : for internal energy

// s : for entropy

// v : specific volume

// yo : the value of the variable given by variable name double t=300;

if(name=='v') {t= p\*1e5\*y0/8.3145e3;} else

{

double dt=0; int nmax=400;

double tolerance=1.0e-8; for(int i=0;i<nmax;i++)

{

// apply newtons method for finding roots of equation if (name=='h') dt=-( h(t) - y0 ) /Cp(t);

else if(name=='u') dt=-( u(t) - y0 ) /Cv(t);

else if(name=='s') dt=-( s(t,p) - y0 ) /(Cp(t)/t);

else { System.out.println("wrong name defined please try h,u,s ot v");} t+=dt;

// if error range is less than tolerance, exit if(Math.abs(dt)<tolerance) break;

}

}

return t;

}

public double T( char name,double y0)

{

// name can have values h : for enthalpy

// u : for internal energy

// s : for entropy

// v : specific volume

// yo : the value of the variable given by variable name double t=300;

double p=1.0;

if(name=='v') {t= p\*1e5\*y0/8.3145e3;} else

{

double dt=0; int nmax=400;

double tolerance=1.0e-8; for(int i=0;i<nmax;i++)

{

// apply newtons method for finding roots of equation if (name=='h') dt=-( h(t) - y0 ) /Cp(t);

else if(name=='u') dt=-( u(t) - y0 ) /Cv(t);

else if(name=='s') dt=-( s(t,p) - y0 ) /(Cp(t)/t);

else { System.out.println("wrong name defined please try h,u,s ot v");} t+=dt;

// if error range is less than tolerance, exit if(Math.abs(dt)<tolerance) break;

}

}

return t;

}

//===================================================================

public double P( char name,double y0,double t1)

{

// name can have values v : for specific volume

// s : for entropy

// note : for a perfect gas enthalpy and internal energy

// is not function of pressure

// yo : the value of the variable given by variable name if(name=='v') return 8.3145e3\*t1/y0\*1e-5;

else if (name=='s') return Math.exp((s(t1,1.0)-y0)/8.3145);

else { System.out.println("wrong name defined please try s or v"); return 1.0;}

}

//===================================================================

public double Pr(double t)

{

// Prandtl number

return Cp(t)\*vis(t)/k(t)/M\*1e3;

}

//===================================================================

public void assign(Gas g1) throws IOException

{

// assign operator (assigning a new gas to the gas variable) int i;

ierror=1; gasName=g1.gasName; N=g1.N;

n\_equation=g1.n\_equation; n\_k=g1.n\_k; n\_vis=g1.n\_vis;

M=g1.M;

h0=g1.h0; hf=g1.hf; sf=g1.sf; natom=g1.natom;

atomList=new Atom[natom]; M=0;

for(i=0;i<natom;i++)

{

atomList[i]=new Atom(g1.atomList[i].name,g1.atomList[i].N);

}

xa=new double[n\_equation]; xb=new double[n\_equation]; xc=new double[n\_equation]; xd=new double[n\_equation]; tl=new double[n\_equation]; th=new double[n\_equation]; xvis=new double[n\_vis]; xk=new double[n\_k]; for(i=0;i<n\_equation;i++)

{ xa[i]=g1.xa[i];

xb[i]=g1.xb[i];

xc[i]=g1.xc[i];

xd[i]=g1.xd[i];

tl[i]=g1.tl[i];

th[i]=g1.th[i];

}

for(i=0;i<n\_vis;i++)

{ xvis[i]=g1.xvis[i];} for(i=0;i<n\_k;i++)

{xk[i]=g1.xk[i];}

}

//===================================================================

public Gas multiply(double Nnew, Gas g1) throws IOException

{

Gas g2=new Gas(g1); g2.N\*=Nnew;

return g2;

}

public boolean equals(Gas g)

{

if(gasName.equals(g.gasName)) return true;

else

return false;

}

//=================================================================

}

//=================================================================

Gaz özelliklerinin yer aldigi bilgi dosyasi Gas.txt (sadece iki örnek gaz ch4 ve c2h6 için veri listelenmistir.) : ch4

2

C 1

H 4

6 0 -17883.1088181 44.4764979459

.569592791684135E+01 .230885866354099E+02 .257021911461535E-03 -.514500509673528E+02 298. 300.0

.549418872803147 23.348756180407 1.4381268667348 -6.89019972775113 300.0 1000.00

11.8834319588735 9.20574503493484 -21.4292411101953 -1.8024921550441 1000.0 2000.0

23.037791919106 1.1029857730146 -87.1157168947875 -.127020658512245 2000.0 3000.0

22.0766220965729 1.36002847842152 -59.4550709113699 -0.1402094192352699 3000.0 4000.0

25.8471557797976 -0.0210627042222442 -144.63391429225 0.00254109611861818 4000.0 6000.0

4

4

c2h6 2

C 2

H 6

-.317434898401931E+01

.455271007474814E+00

-.273399388607163E-03

.895312831114117E-07

.399296677905716E+01

.551053536678452E-01

.172999763063988E-03

-.657213363465470E-07

3 0 -20226.42591 54.8394955574

5.255293329509001 29.744079588767550 -1.128964437911497 -4.086756508274219 298.15 600.0

3.855945449740792 35.946345289228690 -1.504708264372915 -10.330104368611350 600.0 1000.0

21.017162073226170 15.065627352784930 -34.594573805677190 -3.233353234903328 1000.0 1473.15

3

-0.492854013060778E+01 0.379062761757185E+00

-0.148297458234132E-03 3

-0.173496859087276E+02 0.117899036192513E+00

0.399524288778580E-04

### Program 10.17 GasModel.java : bu program veri çiktisinin bir parçasidir, çiktiyi tablo formunda elde etmemiz amaciyla kullanilmistir.

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.table.\*; import javax.swing.event.\*;

class GasModel extends AbstractTableModel

{

Object[][] veri={

{"Gas Formula : ","0"," "},

{"M : ","0","kg/kmol"},

{"h, enthalpy : ","0","KJ/kmol"},

{"u, Internal energy : ","0","KJ/kmol"},

{"s, Entropy : ","0","KJ/kmol K"},

{"v, Specific volume : ","0","m^3/kmol"},

{"k, Thermal conductivity : ","0","W/m K"},

{"h+hf, enthalpy+formation enth. : ","0","KJ/kmol"},

{"Density : ","0","kg/m^3"},

{"Cp : ","0"," KJ/kmol K"},

{"Cv : ","0","KJ/kmol K"},

{"Cp/Cv adiabatic constant : ","0"," "},

{"c speed of sound : ","0","m/s"},

{"Viscosity : ","0","N s/m^2"},

{"Pr,Prandtl Number : ","0"," "}}; String[] baslik={"Property ","Value ","Units"};

public GasModel(Gas g1,double TK,double P)

{

setValues(g1,TK,P);

}

public int getRowCount() {return veri.length;} public int getColumnCount() {return baslik.length;}

public Object getValueAt(int satir,int sutun) {return veri[satir][sutun];} public String getColumnName(int c) {return baslik[c];}

public void setValueAt(Object val, int row, int col)

{

veri[row][col] = val;

}

public void setValues(Gas g1,double T,double P)

{

double TK=T+273.0; setValueAt(g1.toString(),0,1); setValueAt((""+g1.M),1,1); setValueAt(Double.toString(g1.h(TK)),2,1); setValueAt(Double.toString(g1.u(TK)),3,1); setValueAt(Double.toString(g1.v(TK,P)),4,1); setValueAt(Double.toString(g1.s(TK,P)),5,1); setValueAt(Double.toString(g1.k(TK)),6,1); setValueAt(Double.toString(g1.ht(TK)),7,1); setValueAt(Double.toString(g1.M/g1.v(TK)),8,1); setValueAt(Double.toString(g1.Cp(TK)),9,1); setValueAt(Double.toString(g1.Cv(TK)),10,1); setValueAt(Double.toString(g1.gamma(TK)),11,1); setValueAt(Double.toString(g1.c(TK)),12,1); setValueAt(Double.toString(g1.vis(TK)),13,1); setValueAt(Double.toString(g1.Pr(TK)),14,1);

}

public boolean isCellEditable(int row, int col) {return true;}

}

### Program 10.18 GasPropertySWF.java

// ======================================================

// Thermodynamics package in java

// GasPropertySWF class to calculate properties of gases

// user interface (JFrame)

// Dr. Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Research Institute

// email : [Turhan.Coban@posta.mam.gov.tr](mailto:Turhan.Coban@posta.mam.gov.tr)

// =====================================================

import java.io.\*;

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import java.util.\*; import javax.swing.\*;

import javax.swing.table.\*; import GasModel;

public class GasPropertySWF extends JFrame implements ActionListener,ItemListener

{

JPanel inputPanel;

JPanel outputPanel;

JPanel totalPanel;

JTextField promptGasName; // Label prompt GasName

JTextField promptTemperature; // Label prompt Temperature JTextField promptPressure; // Label prompt Pressure JComboBox inputGasName; // input GasName

JTextField inputTemperature; // input (from list )Temperature JTextField inputPressure; // input Pressure

JTextArea outputTextArea; GasModel gm;

JTable jt; double T; double P;

String gasName;

Gas g1;

protected File gmixFile; protected StringTokenizer token; String st[];

final static Color bg=Color.lightGray; final static Color fg=Color.black; final static Color kirmizi=Color.red; final static Color beyaz=Color.white;

public GasPropertySWF()

{

super("Properties of perfect gases "); Container c=getContentPane(); c.setLayout(new FlowLayout());

//adding max-min prompts and input fields gasName=new String("air");

T=27.0;

P=1.0;

inputPanel=new JPanel(); inputPanel.setLayout(new GridLayout(3,2)); outputPanel=new JPanel(); outputPanel.setLayout(new BorderLayout()); totalPanel=new JPanel(); totalPanel.setLayout(new BorderLayout()); try{

g1=new Gas(gasName);

token=new StringTokenizer(g1.readGasNames()); st=new String[token.countTokens()];

} catch(IOException ioe) {System.out.println("IOException");} int i=0;

while(token.hasMoreTokens())

{

st[i++]=new String((String)token.nextToken());

}

inputGasName=new JComboBox(st); gasName=(String)inputGasName.getSelectedItem(); try{

g1=new Gas(gasName);

} catch(IOException ioe) {System.out.println("IOException");} promptGasName= new JTextField(20); promptGasName.setText("Name of gas mixture : "); promptGasName.setBackground(Color.lightGray); promptTemperature=new JTextField(20); promptTemperature.setText("Temperature (°C) : "); promptTemperature.setBackground(Color.lightGray); inputTemperature=new JTextField(20); inputTemperature.setBackground(Color.lightGray); promptPressure= new JTextField(20); promptPressure.setText("Pressure (Bars) : ");

promptPressure.setBackground(Color.lightGray); outputTextArea = new JTextArea(); outputTextArea.setBackground(Color.lightGray); inputPressure=new JTextField(20); inputPressure.setBackground(Color.lightGray); inputPanel.add(promptGasName); inputPanel.add(inputGasName); inputPanel.add(promptTemperature); inputPanel.add(inputTemperature); inputPanel.add(promptPressure); inputPanel.add(inputPressure); inputGasName.addItemListener(this); inputTemperature.addActionListener(this); inputPressure.addActionListener(this);

gm=new GasModel(g1,T,P); jt=new JTable(gm);

jt.setBackground(c.getBackground()); setArea();

outputPanel.add(outputTextArea,BorderLayout.NORTH); outputPanel.add(jt,BorderLayout.SOUTH); totalPanel.add(inputPanel,BorderLayout.NORTH); totalPanel.add(outputPanel,BorderLayout.SOUTH); c.add(totalPanel,BorderLayout.NORTH);

}

public void setArea()

{

double TK=T+273.0; inputTemperature.setText(Double.toString(T)); inputPressure.setText(Double.toString(P));

String s="additional data can be added to Gas.txt file \n"; s+="Dr. Turhan Çoban, TUBITAK Marmara Research Center\n"; s+="Energy Systems & Environmental Research Institute\n"; s+="PO Box 21, Gebze - Kocaeli, Turkey\n";

s+="email: Turhan.Coban@posta.mam.gov.tr\n";

//note if you would like to list additional information on the screen

//add to string s

// s+="Gas Formula : "+g1.toString("formula")+"\n";

// s+= "Gas Composition : "+g1.toString("composition")+"\n"; gm.setValues(g1,T,P);

outputTextArea.setText(s);

}

public void itemStateChanged(ItemEvent ev)

{

gasName=(String)inputGasName.getSelectedItem(); try{

g1=new Gas(gasName);

} catch(IOException ioe1) {System.out.println("IOException");} setArea();

repaint();

}

public void actionPerformed( ActionEvent e)

{

Double valTemp=new Double(inputTemperature.getText()); T=valTemp.doubleValue();

Double valPressure=new Double(inputPressure.getText()); P=valPressure.doubleValue();

getContentPane().setLayout(new FlowLayout(FlowLayout.RIGHT)); setArea();

repaint();

}

public static void main(String[] args)

{

GasPropertySWF pencere= new GasPropertySWF(); pencere.addWindowListener(

new WindowAdapter()

{

public void windowClosing(WindowEvent e) {System.exit(0);}

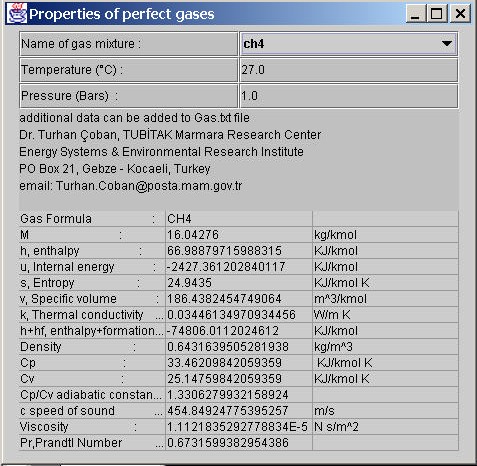
} );

pencere.pack(); pencere.setVisible(true);

}

}

### Program çiktisi :



1. Bu problemde bir önceki problemde verilen uygulamanin bir kademe daha ilerine gidecegiz. Burada Gazlari karistirip bir gaz karisimi elde edecek ve bu gaz karisiminin termodinamik özelliklerine bakacagiz. Programimizin ismi Gmix.java, gazlari karistirdigimiz veri dosyasi Gmix.txt ve çikti dosyalarimiz GmixModel.java ve GmixPropertySWF.java

### Program 10.19 Gmix.java

// ==============================================================

// File Name : Gmix.java

// Author : Dr. Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Research Institute

// email : [Turhan.Coban@posta.mam.gov.tr](mailto:Turhan.Coban@posta.mam.gov.tr)

// Description : This file contains class gmix which calculates

// thermophysical properties of mixture of perfect gases.

// following properties can be calculated

// T() : Temperature degree K

// h(T) : enthalpy KJ/kmol

// hf : formation enthalpy KJ/kg

// ht(T) : total enthalpy KJ/kg (h+hf)

// M : molar mass kg/kmol

// HT(T) : total enthalpy KJ : M\*ht(T)

// P() : presure bar

// s(T,P) : entropy KJ/kmol K

// Cp(T) : specific heat at constant pressure KJ/kmol K

// Cv(T) : specific heat at constant volume KJ/kg K

// gamma(T): adiabatic constant Cp/Cv

// c(T) : speed of sound m/s

// u(T) : Internal energy KJ/kmol

// vis(T) : viscosity

// k(T) : thermal conductivity KW/kg K

// DATA FILE DEFINATION

// gas mixture definations are given the data file "Gmix.txt"

// if gas mixture data is not given in the Gmix.txt user can be add

// his own data to the file which has the following format

//------------------

// gmixName

// ngas

// gname\_0 N\_0

// .........

// gname\_ngas-1 N\_ngas-1

//-------------------

// and defination : gmix a(gmixName); will defined this gas mixture

// the same mixture can be defined directly in the main program as :

// -------------------

// Gas a\_0=new Gas("a\_0");

//.......

// Gas a\_ngas=new Gas("a\_ngas);

// Gmix a=new Gmix;

// a=N\_0\*a\_0+...+N\_ngas\*a\_ngas;

// -------------------

//============================================================

// VARIABLE IDENTIFICATION

// all the variables that type is not defined is a double variable

// PUBLIC VARIABLES :

// gasName : String class variable of gas mixture name

// ngas : int variable, number of simple gasses

// N : total molar mass of the gas mixture

// gasList :gas class vector variables

// All the other variables defined for class gas is also valid for gmix import java.io.\*;

import java.util.\*; import Text; import Atom; import Gas;

// ==============================================================

class Gmix{

// this class calculates perfect gas thermodynamic

// properties when the perfect gas constitutes of

// several single gases

public int ngas; // number of simple gasses inside of the gas mixture String gasName;

public double M; // molar mass of atom kg/kmol public double h0; // enthalpy at T=298 K

public double hf; // enthalpy of formation

public double sf; // entropy of formation kJ/kmol K public double N; // moles of gas kmol

int ierror;

int natom; //number of unique atoms in the atom list

Gas gasList[]; //list of the component gasses Atom atomList[]; //list of component atoms File gmixFile; // File name and directory BufferedReader cfin;

// construction methods :

//========definations of class gmix =============================

//constructor functions

public Gmix(String name) throws IOException

{

// class complex gas construction function

// this function reads the initial gases in

// the mixture and their molar weight from

// the file Gmix.txt and construct mixed gas try{

cfin=new BufferedReader(new FileReader("Gmix.txt")); int ierror=1;

int i,j;

N=0;

M=0;

hf=0; natom=0;

try{ while(cfin!=null)

{

gasName=Text.readString(cfin); if(gasName.equals(name)) { ierror=0; break;}

} //end of while

} catch(EOFException e\_eof)

{

System.out.println("error required gas mixture "+name+" is not found"); cfin.close();return;

}

//cfin>>ngas; ngas=Text.readInt(cfin);

gasList=new Gas[ngas];

//ierror=0; String pgasName; double ppercent; Gas tempgas;

for(i=0;i<ngas;i++)

{

pgasName=Text.readString(cfin); ppercent=Text.readDouble(cfin);

//cfin>>pgasName>>ppercent; tempgas=new Gas(pgasName,ppercent); ierror=tempgas.ierror;

{

if(ierror!=1)

{

try{

gasList[i]=new Gas(tempgas);

} catch(IOException ioe) {System.out.println("IOException");} N+=tempgas.N;

M+=tempgas.N\*tempgas.M; hf+=tempgas.N\*tempgas.hf;

}

else

{

System.out.println("gas is not found in the list");

System.out.println("this gas is not added to the list");

i--;

ngas--;

}

}

}

M=M/N;

hf=hf/N; arrange\_atoms();

} catch(FileNotFoundException fnfe) {System.out.println("File not found");}

}

// ==============================================================

public String readGmixNames() throws IOException

{

String temp=new String(""); String pgasName;

double ppercent; try{

cfin=new BufferedReader(new FileReader("Gmix.txt")); try {

while(cfin!=null)

{

temp=temp+Text.readString(cfin)+ " "; ngas=Text.readInt(cfin);

for(int i=0;i<ngas;i++)

{

pgasName=Text.readString(cfin); ppercent=Text.readDouble(cfin);

} //end of while

}

} catch(EOFException e\_eof) {cfin.close();}

}

catch(FileNotFoundException fnfe) {System.out.println("File Gmix.txt not found");} return temp;

}

// ==============================================================

public void arrange\_atoms() throws IOException

{

int i,j;

for(i=0;i<ngas;i++)

{

for(j=0;j<gasList[i].natom;j++)

{

add\_atom(i,j);

}

}

}

public int add\_atom(int i,int j) throws IOException

{

int k;

//Atom tempAtom=new Atom(gasList[i].atomList[j].symbol,gasList[i].atomList[j].N\*gasList[i].N/N); for(k=0;k<natom;k++)

{

if(gasList[i].atomList[j].symbol.equals(atomList[k].symbol))

{

atomList[k]=new Atom(atomList[k].symbol,atomList[k].N+gasList[i].atomList[j].N\*gasList[i].N/N); return 1;

}

}

Atom atomL[];

atomL=new Atom[natom+1]; for(k=0;k<natom;k++) atomL[k]=new Atom(atomList[k]);

atomL[natom]=new Atom(gasList[i].atomList[j].symbol,gasList[i].atomList[j].N\*gasList[i].N/N); atomList=atomL;

natom+=1; return 2;

}

//===============================================================

public Gmix() throws IOException

{

//empty construction function N=0;

M=0;

String pgasname="\0"; ngas=0;

}

// ==============================================================

public Gmix(Gmix g1) throws IOException

{

gasName=g1.gasName; gasName=gasName; N=g1.N;

M=g1.M;

hf=g1.hf;

ngas=g1.ngas; natom=g1.natom;

gasList=new Gas[ngas];

for(int i=0;i<ngas;i++) gasList[i]=new Gas(g1.gasList[i]); for(int i=0;i<natom;i++) atomList[i]=new Atom(g1.atomList[i]);

}

// ==============================================================

public Gmix(Gas g1) throws IOException

{

gasName=g1.gasName; gasName=gasName; N=g1.N;

M=g1.M;

hf=g1.hf; ngas=1; natom=g1.natom;

gasList=new Gas[ngas];

for(int i=0;i<ngas;i++) gasList[i]=new Gas(g1);

for(int i=0;i<natom;i++) atomList[i]=new Atom(g1.atomList[i]);

}

// ==============================================================

public void changeN(double newN) throws IOException

{

normalise();

for(int i=0;i<ngas;i++)

{ gasList[i].N\*=newN; } N=newN;

}

// ==============================================================

public void add(Gas g1) throws IOException

{

// this function adds a single gas to the mixture int gasflag=1;

// if the gas exist in the list simply change N and M values for(int i=0;i<ngas;i++)

{

if(gasList[i].gasName.equals(g1.gasName))

{

gasflag=0;

M=M\*N+g1.M\*g1.N;

gasList[i].N+=g1.N;

N+=g1.N;

M=M/N;

}

}

if(gasflag!=0)

{

Gas newGas[]; newGas=new Gas[ngas+1]; for(int i=0;i<ngas;i++)

{

newGas[i]=new Gas(gasList[i]);

}

ngas++;

double MT=M\*N+g1.M\*g1.N; N+=g1.N;

M=MT/N;

newGas[ngas-1]=new Gas(g1); gasList=newGas;

}

arrange\_atoms();

}

// ==============================================================

public void remove(String name) throws IOException

{

// this function removes a single gas

// from the list int i,k;

for(i=0;i<ngas;i++)

{

if(gasList[i].gasName.equals(name))

{

M=0; N=0;

Gas newGas[]; newGas=new Gas[ngas];

for(k=0;k<i;k++)

{ newGas[k]=new Gas(gasList[k]); M+=newGas[k].M\*newGas[k].N; N+=newGas[k].N;

}

for( k=i;k<(ngas-1);k++)

{

newGas[k]=new Gas(gasList[k+1]); M+=newGas[k].M\*newGas[k].N;

N+=newGas[k].N;

} M=M/N;

ngas--; gasList=newGas; break;

}

}

arrange\_atoms();

// correct dynamic memory size

}

// ==============================================================

public void add(String name,double Nnew) throws IOException

{

// this function adds a single gas to the mixture Gas g1=new Gas(name,Nnew);

int gasflag=1;

// if the gas exist in the list simply change N and M values for(int i=0;i<ngas;i++)

{

if(gasList[i].gasName.equals(g1.gasName))

{

gasflag=0;

M=M\*N+g1.M\*g1.N;

gasList[i].N+=g1.N;

N+=g1.N;

M=M/N;

}

}

if(gasflag!=0)

{

Gas newGas[]; newGas=new Gas[ngas+1]; for(int i=0;i<ngas;i++)

{

newGas[i]=new Gas(gasList[i]);

}

ngas++;

double MT=M\*N+g1.M\*g1.N; N+=g1.N;

M=MT/N;

newGas[ngas-1]=new Gas(g1); gasList=newGas;

}

arrange\_atoms();

}

// ==============================================================

public void simplify() throws IOException

{

// this function combines any single gas

// that repeated in the list double ngasold=ngas;

int i,j,k;

for(i=0;i<ngas;i++)

{

for(j=i+1;j<ngas;j++)

{

if(gasList[i].gasName.equals(gasList[j].gasName))

{

gasList[i].N+=gasList[j].N; for(k=j;k<(ngas-1);k++)

{ gasList[k]=new Gas(gasList[k+1]); } ngas--;

}

}

}

// correct dynamic memory size if(ngasold!=ngas)

{

Gas newGas[]; newGas=new Gas[ngas]; for(i=0;i<ngas;i++)

{

newGas[i]=new Gas(gasList[i]);

}

gasList=newGas;

}

}

// ==============================================================

public void normalise() throws IOException

{

for(int i=0;i<ngas;i++)

{

gasList[i].N=gasList[i].N/N;

} N=1.0;

arrange\_atoms();

}

// ==============================================================

public void changeMix(double Nmix[]) throws IOException

{

// this function changes

// all the molar weights in the mixture

N=0; M=0;

{

for(int i=0;i<ngas;i++)

gasList[i].N=Nmix[i]; N+=Nmix[i]; M+=Nmix[i]\*gasList[i].M;

} M=M/N;

arrange\_atoms();

}

// ==============================================================

public double vis(double T)

{

// dynamic viscosity of the mixture

// note that viscosity of the mixture IS NOT the simple addition

// of viscosity of component gasses double vmix=0;

for(int i=0;i<ngas;i++)

{ T=T;

double fij; double xj=0; double p1; double c1,c2,c3;

for(int j=0;j<ngas;j++)

{

double vratio;

double xi=gasList[i].M/gasList[j].M; if(gasList[j].vis(T)!=0)

{

c1=gasList[i].vis(T)/gasList[j].vis(T);

else

vratio=Math.sqrt(c1);

}

vratio=0; c2=Math.pow(xi,0.25); p1=(1+vratio/c2); c3=8.0+8.0\*xi;

fij=p1\*p1 / Math.sqrt(c3);

xj+=fij\*gasList[j].N/N;

}

vmix+=gasList[i].N/N\*gasList[i].vis(T)/xj;

}

return vmix;

}

// ==============================================================

public double k(double T)

{

// thermal conductivity of the mixture

// note that thermal conductivity of the mixture IS NOT the

// simple addition of the thermal conductivity of component gasses double vmix=0;

for(int i=0;i<ngas;i++)

{ T=T;

double fij; double xj=0; double p1; double c1,c2,c3;

for(int j=0;j<ngas;j++)

{

double vratio;

double xi=gasList[i].M/gasList[j].M; if(gasList[j].k(T)!=0)

{

c1=gasList[i].k(T)/gasList[j].k(T);

// if(c1<0) System.out.println("negative c1 value "); vratio=Math.sqrt(c1);

}

else

vratio=0;

c2=Math.pow(xi,0.25); p1=(1+vratio/c2);

c3= 8.0+8.0\*xi;

// if(c2<0) System.out.println(" negative c2 value "); fij=p1\*p1/Math.sqrt(c3);

xj+=fij\*gasList[j].N/N;

}

vmix+=gasList[i].N/N\*gasList[i].k(T)/xj;

}

return vmix;

}

// ==============================================================

public double Pr(double t)

{

// Prandtl number

return Cp(t)\*vis(t)/k(t)/M\*1e3;

}

// ==============================================================

public double h(double T)

{

//specific enthalpy of the mixture KJ/kmol double HH=0;

for(int i=0;i<ngas;i++)

{ HH+=gasList[i].h(T)\*gasList[i].N; } return HH/N;

}

// ==============================================================

public double ht(double T)

{

//specific enthalpy of the mixture ht=h+hf

// hf : formation enthalpy double HH=0;

for(int i=0;i<ngas;i++)

{ HH+=gasList[i].ht(T)\*gasList[i].N; } return HH/N;

}

// ==============================================================

public double H(double t)

{

//total enthalpy of the mixture KJ return h(t)\*N;

}

// ==============================================================

public double HT(double t)

//total enthalpy of the mixture HT=N\*(h+hf) KJ

{

return ht(t)\*N;

}

// ==============================================================

public double u(double T)

{

// specific internal energy of the mixture KJ/kmol double UU=0;

for(int i=0;i<ngas;i++)

{ UU+=gasList[i].u(T)\*gasList[i].N; } return UU/N;

}

// ==============================================================

public double Cp(double T)

{

// Specific energy at constant pressure KJ/kmol K double C=0;

for(int i=0;i<ngas;i++)

{ C+=gasList[i].Cp(T)\*gasList[i].N; } return C/N;

}

// ==============================================================

public double Cv(double T)

//Specific energy at constant volume KJ/kmol K

{

double C=0;

for(int i=0;i<ngas;i++)

{ C+=gasList[i].Cv(T)\*gasList[i].N; } return C/N;

}

// ==============================================================

public double gamma(double T)

{

//adiabatic constant return Cp(T)/Cv(T);

}

// ==============================================================

public double c(double T)

//speed of sound m/s

{

return Math.sqrt(8314.5/M\*T\*gamma(T));

}

// ==============================================================

public double s(double T, double P)

{

//specific entropy KJ/kmol K double SS=0;

for(int i=0;i<ngas;i++)

{

SS+=gasList[i].s(T,P)\*gasList[i].N;

}

return SS/N;

}

public double s(double T)

{

//specific entropy KJ/kmol K double P=1.0;

double SS=0;

for(int i=0;i<ngas;i++)

{

SS+=gasList[i].s(T,P)\*gasList[i].N;

}

return SS/N;

}

// ==============================================================

public double v(double T, double P)

{

double VV=0;

for(int i=0;i<ngas;i++)

{ VV+=gasList[i].v(T,P)\*gasList[i].N; } return VV/N;

}

// ==============================================================

public double T( char name,double y0,double p)

{

double t=300;

if(name=='v') {t= p\*1e5\*y0/8.314e3;} else

{

double dt;

int nmax=400;

double tolerance=1.0e-8; for(int i=0;i<nmax;i++)

{

if (name=='h') dt=-( h(t) - y0 ) /Cp(t);

else if(name=='u') dt=-( u(t) - y0 ) /Cv(t);

else if(name=='s') dt=-( s(t,p) - y0 ) /(Cp(t)/t);

else { System.out.println("wrong name defined please try h,u,s ot v"); break;}

t+=dt;

if(Math.abs(dt)<tolerance) break;

}

}

return t;

}

// ==============================================================

public double P( char name,double y0,double t1)

{

if(name=='v') return 8.314e3\*t1/y0\*1e-5;

else if (name=='s') return Math.exp((s(t1,1.0)-y0)/8.314);

else { System.out.println("wrong name defined please try s or v"); return 1.0;}

}

// ==============================================================

public void multiplyassign(double Nx)

{

for(int i=0;i<ngas;i++)

{ gasList[i].N\*=Nx; } N\*=Nx;

}

// ==============================================================

public void addassign(Gas g1) throws IOException

{

// this function adds a single gas to the mixture int gasflag=1;

// if the gas exist in the list simply change N and M values for(int i=0;i<ngas;i++)

{

if(gasList[i].gasName==g1.gasName)

{

gasflag=0;

M=M\*N+g1.M\*g1.N;

gasList[i].N+=g1.N;

N+=g1.N;

M=M/N;

}

}

if(gasflag!=0)

{

Gas newGas[]; newGas=new Gas[ngas+1]; for(int i=0;i<ngas;i++)

{

newGas[i]=new Gas(gasList[i]);

}

ngas++;

double MT=M\*N+g1.M\*g1.N; N+=g1.N;

M=MT/N;

newGas[ngas-1]=new Gas(g1); gasList=newGas;

}

}

// ==============================================================

public void addassign(Gmix right) throws IOException

{

// this function adds a gas mixture to the mixture for(int i=0;i<right.ngas;i++)

{

add(right.gasList[i]);

}

}

// ==============================================================

public Gmix multiply( double Nx,Gmix right) throws IOException

{

Gmix g1=new Gmix(right);

g1.N\*=Nx;

for(int i=0;i<g1.ngas;i++)

{

g1.gasList[i].N\*=Nx;

}

return g1;

}

// ==============================================================

public Gmix add(Gas l,Gas r) throws IOException

{

Gmix g1=new Gmix(r); Gas g2=new Gas(l); g1.add(g2);

return g1;

}

// ==============================================================

public Gmix add(Gmix l,Gas r) throws IOException

{

Gmix g1=new Gmix(l); Gas g2=new Gas(r); g1.add(g2);

return g1;

}

// ==============================================================

public Gmix add(Gas l,Gmix r) throws IOException

{

Gmix g1=new Gmix(r); Gas g2=new Gas(l); g1.add(g2);

return g1;

}

// ==============================================================

public Gmix add(Gmix l,Gmix r) throws IOException

{

Gmix g1=new Gmix(l); for(int i=0;i<r.ngas;i++)

{

g1.add(r.gasList[i]);

}

return g1;

}

// ==============================================================

public void assign(Gmix g1) throws IOException

{

gasName=g1.gasName;

N=g1.N;

M=g1.M;

ngas=g1.ngas;

Gas newGas[];

newGas=new Gas[ngas];

for(int i=0;i<ngas;i++) newGas[i]=g1.gasList[i]; gasList = newGas;

}

// ==============================================================

public void assign(Gas g1) throws IOException

{

// a single gas is assigned to the mixture gasName=g1.gasName;

N=g1.N;

M=g1.M;

ngas=1;

gasList=new Gas[ngas]; gasList[0]=new Gas(g1);

}

public String toString(String ch)

{

//return the c String s=""; int i,j;

if(ch.equals("name")) s=s+gasName+"\n";

else if(ch.equals("formula"))

{

for(i=0;i<ngas;i++)

{ s=s+" "+gasList[i].toString()+" "+gasList[i].N+"\n"; }

}

else if(ch.equals("composition"))

{

for(i=0;i<natom;i++) s=s+atomList[i].toString()+"\n";

}

return s;

}

}//end of class

### Gmix veri dosyasi Gmix.txt (kismi liste )

diesel1 25

c9h20 0.0122

c10h22 0.0243

c11h24 0.0517

c12h26 0.0912

c13h28 0.2007

c14h30 0.1959

c15h32 0.098

c16h34 0.049

c17h36 0.0245

c18h38 0.0122

c19h40 0.0061

c20h42 0.0031

c11h16 0.0027

c12h18 0.0041

c13h20 0.0055

c14h22 0.0058

c15h24 0.0059

c16h26 0.0065

c17h28 0.0030

c18h30 0.0020

c10h8 0.0302

c11h10 0.0654

c12h12 0.0453

c13h14 0.0322

c14h16 0.0215

metanolreformeryanmaodasi 4

co2 1

h2o 2.1

n2 7.5238

o2 0.5

biogaz 5

h2 0.116

co 0.174

ch4 0.082

co2 0.131

n2 0.382

yanmahavasi 2

o2 2.463

n2 9.26557

### Program 10.20 GmixMethod.java

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.table.\*; import javax.swing.event.\*;

class GmixModel extends AbstractTableModel

{

Object[][] veri={

{"M : ","0","kg/kmol"},

{"h, enthalpy : ","0","KJ/kmol"},

{"u, Internal energy : ","0","KJ/kmol"},

{"s, Entropy : ","0","KJ/kmol K"},

{"v, Specific volume : ","0","m^3/kmol"},

{"k, Thermal conductivity : ","0","W/m K"},

{"h+hf, enthalpy+form. enth. : ","0","KJ/kmol"},

{"Density : ","0","kg/m^3"},

{"Cp : ","0"," KJ/kmol K"},

{"Cv : ","0","KJ/kmol K"},

{"Cp/Cv adiabatic constant : ","0"," "},

{"c speed of sound : ","0","m/s"},

{"Viscosity : ","0","N s/m^2"},

{"Pr,Prandtl Number : ","0"," "}}; String[] baslik={"Property ","Value ","Units"};

public GmixModel(Gmix g1,double TK,double P)

{

setValues(g1,TK,P);

}

public int getRowCount() {return veri.length;} public int getColumnCount() {return baslik.length;}

public Object getValueAt(int satir,int sutun) {return veri[satir][sutun];} public String getColumnName(int c) {return baslik[c];}

public void setValueAt(Object val, int row, int col)

{

veri[row][col] = val;

}

public void setValues(Gmix g1,double T,double P)

{

double TK=T+273.0; setValueAt((""+g1.M),0,1);

setValueAt(Double.toString(g1.h(TK)),1,1); setValueAt(Double.toString(g1.u(TK)),2,1); setValueAt(Double.toString(g1.v(TK,P)),3,1); setValueAt(Double.toString(g1.s(TK,P)),4,1); setValueAt(Double.toString(g1.k(TK)),5,1); setValueAt(Double.toString(g1.ht(TK)),6,1); setValueAt(Double.toString(g1.M/g1.v(TK,P)),7,1); setValueAt(Double.toString(g1.Cp(TK)),8,1); setValueAt(Double.toString(g1.Cv(TK)),9,1); setValueAt(Double.toString(g1.gamma(TK)),10,1); setValueAt(Double.toString(g1.c(TK)),11,1); setValueAt(Double.toString(g1.vis(TK)),12,1); setValueAt(Double.toString(g1.Pr(TK)),13,1);

}

public boolean isCellEditable(int row, int col) {return false;}

}

### Program 10.21 GmixPropertySWF.java

// ======================================================

// Thermodynamics package in java

// GmixPropertySWF class to calculate properties of perfect

// gas mixtures user interface (JFrame)

// Dr. Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Research Institute

// email : [Turhan.Coban@posta.mam.gov.tr](mailto:Turhan.Coban@posta.mam.gov.tr)

// =====================================================

import java.io.\*;

import java.applet.Applet; import java.awt.\*;

import java.awt.event.\*; import java.util.\*; import javax.swing.\*;

import javax.swing.table.\*; import GmixModel;

public class GmixPropertySWF extends JFrame implements ActionListener,ItemListener

{

final static String MAIN = "Main Page";

final static String GASFORMULA = "Gas Formula"; JTabbedPane tabbedPane;

JPanel inputPanel;

JPanel outputPanel;

JPanel totalPanel;

JTextField promptGmixName; // Label prompt GmixName JTextField promptTemperature; // Label prompt Temperature JTextField promptPressure; // Label prompt Pressure JComboBox inputGmixName; // input GmixName JTextField inputTemperature; // input (from list )Temperature JTextField inputPressure; // input Pressure

JTextArea outputTextArea; JTextArea outputTextArea1; GmixModel gm;

JTable jt; double T; double P;

String GmixName;

Gmix g1;

protected File gmixFile;

protected StringTokenizer token; String st[];

final static Color bg=Color.lightGray; final static Color fg=Color.black; final static Color kirmizi=Color.red; final static Color beyaz=Color.white;

public GmixPropertySWF()

{

super("Properties of perfect gas mixtures "); Container c=getContentPane(); c.setLayout(new FlowLayout()); tabbedPane = new JTabbedPane();

//adding max-min prompts and input fields GmixName=new String("air");

T=27.0;

P=1.0;

inputPanel=new JPanel(); inputPanel.setLayout(new GridLayout(3,2)); outputPanel=new JPanel(); outputPanel.setLayout(new BorderLayout()); totalPanel=new JPanel(); totalPanel.setLayout(new BorderLayout()); try{

g1=new Gmix(GmixName);

token=new StringTokenizer(g1.readGmixNames()); st=new String[token.countTokens()];

} catch(IOException ioe) {System.out.println("IOException");} int i=0;

while(token.hasMoreTokens())

{

st[i++]=new String((String)token.nextToken());

}

inputGmixName=new JComboBox(st); GmixName=(String)inputGmixName.getSelectedItem(); try{

g1=new Gmix(GmixName);

} catch(IOException ioe) {System.out.println("IOException");} promptGmixName= new JTextField(20); promptGmixName.setText("Simple Gmix Name : "); promptGmixName.setBackground(Color.lightGray); promptTemperature=new JTextField(20); promptTemperature.setText("Temperature (°C) : "); promptTemperature.setBackground(Color.lightGray); inputTemperature=new JTextField(20); inputTemperature.setBackground(Color.lightGray); promptPressure= new JTextField(20); promptPressure.setText("Pressure (Bars) : "); promptPressure.setBackground(Color.lightGray); outputTextArea = new JTextArea(); outputTextArea.setBackground(Color.lightGray); outputTextArea1 = new JTextArea(); outputTextArea1.setBackground(Color.lightGray); inputPressure=new JTextField(20); inputPressure.setBackground(Color.lightGray); inputPanel.add(promptGmixName); inputPanel.add(inputGmixName); inputPanel.add(promptTemperature); inputPanel.add(inputTemperature); inputPanel.add(promptPressure);

inputPanel.add(inputPressure); inputGmixName.addItemListener(this); inputTemperature.addActionListener(this); inputPressure.addActionListener(this); gm=new GmixModel(g1,T,P);

jt=new JTable(gm); jt.setBackground(c.getBackground()); setArea();

outputPanel.add(outputTextArea,BorderLayout.CENTER); outputPanel.add(jt,BorderLayout.NORTH); totalPanel.add(inputPanel,BorderLayout.NORTH); totalPanel.add(outputPanel,BorderLayout.CENTER); tabbedPane.addTab(MAIN, totalPanel); tabbedPane.addTab(GASFORMULA, outputTextArea1); c.add(tabbedPane);

}

public void setArea()

{

double TK=T+273.0; inputTemperature.setText(Double.toString(T)); inputPressure.setText(Double.toString(P));

String s="additional data can be added to Gmix.txt file \n"; s+="Dr. Turhan Çoban, TUBITAK Marmara Research Center\n"; s+="Energy Systems & Environmental Research Institute\n"; s+="PO Box 21, Gebze - Kocaeli, Turkey\n";

s+="email: [Turhan.](mailto:Turhan.Coban@posta.mam.gov.tr)[Coban@posta.mam.gov.tr";](mailto:Coban@posta.mam.gov.tr)

//note if you would like to list additional information on the screen

//add to string s String s1="";

s1+="Gas mixture formula :\n"; s1+="=========================\n"+g1.toString("formula")+"\n"; s1+="Gas mixture composition :\n"; s1+="=========================\n"+g1.toString("composition")+"\n"; gm.setValues(g1,T,P);

outputTextArea.setText(s); outputTextArea1.setText(s1);

}

public void itemStateChanged(ItemEvent ev)

{

GmixName=(String)inputGmixName.getSelectedItem(); try{

g1=new Gmix(GmixName);

} catch(IOException ioe1) {System.out.println("IOException");} setArea();

repaint();

}

public void actionPerformed( ActionEvent e)

{

Double valTemp=new Double(inputTemperature.getText()); T=valTemp.doubleValue();

Double valPressure=new Double(inputPressure.getText()); P=valPressure.doubleValue();

getContentPane().setLayout(new FlowLayout(FlowLayout.RIGHT)); setArea();

repaint();

}

public static void main(String[] args)

{

GmixPropertySWF pencere= new GmixPropertySWF(); pencere.addWindowListener(

new WindowAdapter()

{

public void windowClosing(WindowEvent e) {System.exit(0);}

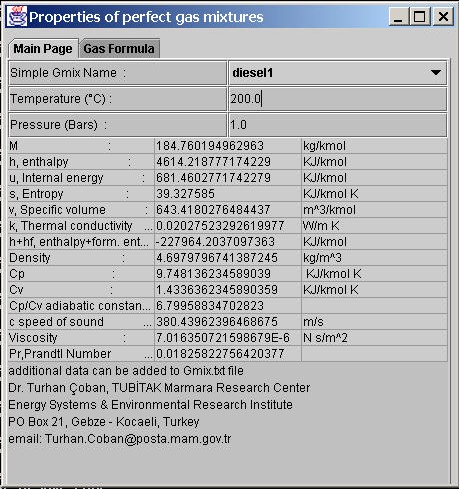
} );

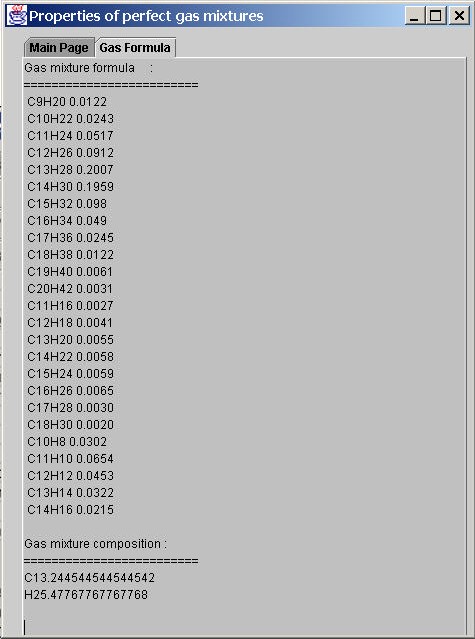
pencere.pack(); pencere.setVisible(true);

}

}

Veri çiktisi :





Burada hemen bir noktaya dikkat çekelim. Gerçek programlamalarda asil veriyi isleyen siniflar ve insan grafik arayüzü programlari genelde ayri tutulur. Bu problem seti ayni zamanda siniflarin birbiri üzerine insasinin da güzel bir örnegini teskil etmektedir.

6. Bu problemde bir önceki problemde verilen uygulamanin bir kademe daha ilerine gidecegiz. Burada Gaz karisimlarini bir borudan akitacagiz. Borunun basinç düsümü isi transferi gibi çesitli özelliklerini bu program yardimiyla degerlendirecegiz. Borularin özellikleri Pipe.txt dosyasinda tanimlanmistir. Gaz karisimi için Boru basinç düsümü ve isi transferini hesaplayan programimizin ismi Pipe.java’dir.

### Program 10.22 Pipe.java

//===================================================================

// Thermodynamic package in java

// File Name : pipe.java

// Author : Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Research Institute

// email : [turhan@mam.gov.tr](mailto:turhan@mam.gov.tr)

// Description : This file contains the pipe class

// class pipe calculates pressure drops

// and internal heat transfer of pipes

// DATA FILE DEFINATION

// pipe datas are written in the data file "Pipe.txt"

// if gas data is not given in the data file, it can be curve fitted

// and added to the data file. Each data has the following form :

//------------------

// pipeName

// shape

// gasName

// d1 d2 (or only d1 depends on shape)

// mass\_flow\_rate

//-------------------

// a constructor is also available to define this data as passing values

//============================================================

// VARIABLE IDENTIFICATION

// all the variables that type is not defined is a double variable

// pipeName : String class variable for name

// shape : String class variable for shape of the cross-section

// options :

// circular

// rectangular

// concentric\_annulus

// circular\_segment

// circular\_sector

// right\_triangle

// gasName : String class variable for name of the gas mixtures

// g : gmix class variable

// mass\_flow\_rate : mass flow rate of gas mixture flowing

// through pipe, kg/s

// eps : pipe equivalent roughness m

// -------------------------------------------

// value of equivalent roughness for some pipes:

// Pipe equivalent rougness m

// ------------- ---------------------

// Riveted steel 0.9e-3 - 9e-3

// Concrete 0.3e-3 - 3e-3

// Wood stave 0.18e-3 - 0.9 e-3

// Cast iron 0.26e-3

// Galvanised iron 0.15e-3

// Commercial steel 0.045e-3

// wrought iron 0.045e-3

// drawn tubing 0.0015e-3

// Plastic,glass 0.0 (smooth)

// -------------------------------------------

// P : Pressure bar

// l : length m

// dh : hydraulic diameter m

// A : pipe area m^2

// KL : local pressure drop coefficient

//

//============================================================

import java.io.\*; import Text; import Gmix;

class pipe{

protected double d1,d2; // m pipe dimensions

// actual meaning of d1 and d2

// can be change according to shape public String pipeName;

public String shape; //pipe shape "circular","rectangular"... public String gasName; //name of a gas mixture

public Gmix g;

public double mass\_flow\_rate; // kg/s public double eps ; // m

public double P ; // Pressure bar

public double l ; // length

public double dh; // hydraulic diameter public double A; // pipe area m^2

public double KL; //additional local pressure drop

// pipe functions

//=============================================

public static double tanh(double x)

{

//I couldnt find hyperbolic tangent in java

return (Math.exp(x)-Math.exp(-x))/(Math.exp(x)+Math.exp(-x));

}

public static double f(double Re,double eod)

{

// friction factor for turbulent flow 5000<Re<10^8

double f1=Math.log(eod/3.7+5.74/ Math.pow(Re,0.9))/Math.log(19);; f1=1.325/(f1\*f1);

return f1;

}

public static double fx(double X,double Re,double eod)

{

// colebrook equation to solve

// friction factor for turbulent flow 2000 < Re

double xx=2.0\*Math.log(eod/3.7+2.51/Re\*X)/Math.log(10.0)+X; return xx;

}

public static double dfx(double X,double Re,double eod)

{

//derivative of colebrook equation double xx;

xx = 1+2.0/(eod/3.7+2.51/Re\*X)/Math.log(10.0)\*2.51/Re; return xx;

}

public static double fcol(double Re,double eod)

{

//solution of the colebrook equation

// by using newton method double fi=f(Re,eod);

double x=1.0/ Math.pow(fi,0.5); int nmax=50;

double tolerance=1.0e-10; for(int i=0;i<nmax;i++)

{

double fx1=fx(x,Re,eod); x-=fx1/dfx(x,Re,eod);

if(Math.abs(fx1)<tolerance) return 1.0/(x\*x);

}

return 1.0/(x\*x);

}

//==============================================

// implementation of constructors functions

// for class pipe

public pipe(String name,double le)

{

//this function reads pipe configuration

//from file Pipe.txt l=le;

//beginning of try block try{

BufferedReader cfin=new BufferedReader(new FileReader("Pipe.txt")); int ierror=1;

//\*\*\*\*\*\*\*\*\*\*\*\* try{

while(cfin!=null)

{

pipeName=Text.readString(cfin); if(pipeName.equals(name)) { ierror=0; break;}

} //end of while

} catch(EOFException e\_eof)

{

System.out.println("error required pipe "+name+" is not found"); cfin.close();return;

}

//\*\*\*\*\*\*\*\*\*\*\*\* if(ierror==1)

{System.out.println("error required pipe name cannot be found in the data file\n");return;} shape=Text.readString(cfin);

gasName=Text.readString(cfin); g=new Gmix(gasName); if(g==null)

{

System.out.println("gmix is not found\n");

}

double pi=4.0\*Math.atan(1.0); if(shape.equals("circular"))

{

d1=Text.readDouble(cfin); d2=0.0;

A=pi\*d1\*d1/4;dh=d1;

}

else if(shape.equals("rectangular"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); A=d1\*d2;dh=4.0\*d1\*d2/(2.0\*d1+2.0\*d2);

}

else if(shape.equals("concentric\_annulus"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); dh=Math.abs(d2-d1); A=pi\*dh\*dh/4.0;

}

else if(shape.equals("circular\_segment"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2)); A=pi\*dh\*dh/4.0;

}

else if(shape.equals("circular\_sector"))

{

d1=Text.readDouble(cfin);

d2=Text.readDouble(cfin); dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0;

}

else if(shape.equals("right\_triangle"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin);;

dh=4.0\*d1\*d2/(d1+d2+ Math.pow((d1\*d1+d2\*d2),0.5));

}

eps=0.03e-3; mass\_flow\_rate=Text.readDouble(cfin); P=1.0;;

KL=0.0;

//end of try block

}

catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

}

public pipe(String name)

{

//this function reads pipe configuration

//from file Pipe.txt l=1.0;

//beginning of try block try{

BufferedReader cfin=new BufferedReader(new FileReader("Pipe.txt")); int ierror=1;

while(cfin!=null)

{

pipeName=Text.readString(cfin); if(pipeName.equals(name)) {ierror=0;break;}

}

if(ierror==1)

{System.out.println("error required pipe name cannot be found in the data file\n");return;} shape=Text.readString(cfin);

gasName=Text.readString(cfin); g=new Gmix(gasName); if(g==null)

{

System.out.println("gmix is not found\n");

}

double pi=4.0\*Math.atan(1.0); if(shape.equals("circular"))

{

d1=Text.readDouble(cfin); d2=0.0;

A=pi\*d1\*d1/4;dh=d1;

}

else if(shape.equals("rectangular"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); A=d1\*d2;dh=4.0\*d1\*d2/(2.0\*d1+2.0\*d2);

}

else if(shape.equals("concentric\_annulus"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); dh=Math.abs(d2-d1); A=pi\*dh\*dh/4.0;

}

else if(shape.equals("circular\_segment"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2)); A=pi\*dh\*dh/4.0;

}

else if(shape.equals("circular\_sector"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin); dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0;

}

else if(shape.equals("right\_triangle"))

{

d1=Text.readDouble(cfin); d2=Text.readDouble(cfin);;

dh=4.0\*d1\*d2/(d1+d2+ Math.pow((d1\*d1+d2\*d2),0.5));

}

eps=0.03e-3; mass\_flow\_rate=Text.readDouble(cfin); P=1.0;;

KL=0.0;

//end of try block

}

catch(IOException e)

{

System.err.println("Error Opening File \n"+e.toString()); System.exit(1);

}

}

void change\_d1(double dim1)

{

d1=dim1;

double pi=4.0\*Math.atan(1.0); if(shape=="circular")

{ A=pi\*d1\*d1/4;dh=d1;} else if(shape=="rectangular")

{

A=d1\*d2;dh=4.0\*d1\*d2/(2.0\*d1+2.0\*d2);

}

else if(shape=="concentric\_annulus")

{dh=Math.abs(d2-d1);A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_segment"))

{dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2));A=pi\*dh\*dh/4.0;} else if(shape=="circular\_sector")

{dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0; } else if(shape=="right\_triangle")

{

dh=4.0\*d1\*d2/(d1+d2+ Math.pow((d1\*d1+d2\*d2),0.5)); A=d1\*d2/2.0;

}

}

void change\_d2(double dim2)

{

d2=dim2;

double pi=4.0\*Math.atan(1.0); if(shape=="circular")

{ d2=0;A=pi\*d1\*d1/4;dh=d1;} else if(shape.equals("rectangular"))

{

A=d1\*d2;dh=4.0\*d1\*d2/(2.0\*d1+2.0\*d2);

}

else if(shape.equals("concentric\_annulus"))

{dh=Math.abs(d2-d1);A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_segment"))

{dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2));A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_sector"))

{dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0; } else if(shape.equals("right\_triangle"))

{

dh=4.0\*d1\*d2/(d1+d2+ Math.pow((d1\*d1+d2\*d2),0.5)); A=d1\*d2/2.0;

}

}

pipe(String pn,

String sh, String gn, double dim1,

double dim2,

double m,double e, double pp,

double k,

double length)

{

// direct assignment constructor

// this constructor can assign all

// the required values directly without

// going to the data file pipeName=pn; l=length;

shape=sh; try{

g=new Gmix(gn);

} catch(IOException ioe) {System.out.println("IOException");}

double pi= 4.0\*Math.atan(1.0); d1=dim1;

d2=dim2; if(shape.equals("circular"))

//d1 is diameter of the pipe

{ A=pi\*d1\*d1/4;dh=d1;d2=0;} else if(shape.equals("rectangular"))

// d1 and d2 are two sides of the rectangle

{ A=d1\*d2;dh=4.0\*d1\*d2/(2.0\*d1+2.0\*d2);

}

else if(shape.equals("concentric\_annulus"))

// d1 and d2 are inrernal and external diameters

//of the concentric annulus

{ dh=Math.abs(d2-d1);A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_segment"))

// d1 is diameter and d2 is the angle of the circular segment

{ dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2));A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_sector"))

//d1 is the diameter, d2 is the height of the circular section

{ dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0; } else if(shape.equals("right\_triangle"))

// d1 and d2 two sides of right triangle

{

dh=4.0\*d1\*d2/(d1+d2+ Math.pow((d1\*d1+d2\*d2),0.5)); A=d1\*d2/2.0;

}

eps=e; mass\_flow\_rate=m; P=pp;

KL=k;

}

pipe(String pn,

String sh, Gmix gmix1, double dim1,

double dim2,

double m, double e,

double pp, double k,

double length)

{

// direct assignment constructor

// this constructor can assign all

// the required valued directly without

// going to the data file pipeName=pn; shape=sh;

//setup gas name try{

g=new Gmix(gmix1);

} catch(IOException ioe) {System.out.println("IOException");}

double pi= 4.0\*Math.atan(1.0); d1=dim1;

d2=dim2; l=length;

if(shape.equals("circular"))

//d1 is diameter of the pipe

{ A=pi\*d1\*d1/4;dh=d1;d2=0;} else if(shape.equals("rectangular"))

// d1 and d2 are two sides of the rectangle

{ A=d1\*d2;dh=4.0\*d1\*d2/(2.0\*d1+2.0\*d2);

}

else if(shape.equals("concentric\_annulus"))

// d1 and d2 are inrernal and external diameters

//of the concentric annulus

{ dh=Math.abs(d2-d1);A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_segment"))

// d1 is diameter and d2 is the angle of the circular segment

{ dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2));A=pi\*dh\*dh/4.0;}

else if(shape.equals("circular\_sector"))

//d1 is the diameter, d2 is the height of the circular section

{ dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0; } else if(shape.equals("right\_triangle"))

// d1 and d2 two sides of right triangle

{ dh=2.0\*d1\*Math.sin(d2)/(1.0+Math.sin(d2)+Math.cos(d2)); A=pi\*dh\*dh/4.0;} eps=e;

mass\_flow\_rate=m;

P=pp;

KL=k;

}

void change\_gas(String str1)

{

try{

g=new Gmix(str1);

}

catch(IOException ioe) {System.out.println("IOException");}

}

void change\_pipe( String pn,

String sh, String gn, double dim1,

double dim2,

double e, double m,

double pp, double k)

{

// change assignment

// this constructor can assign all

// the required valued directly without

// going to the data file pipeName=pn; shape=sh;

try{

g=new Gmix(gn);

} catch(IOException ioe) {System.out.println("IOException");} double pi= 4.0\*Math.atan(1.0);

d1=dim1; d2=dim2;

if(shape.equals("circular"))

//d1 is diameter of the pipe

{ A=pi\*d1\*d1/4;dh=d1;}

else if(shape.equals("rectangular"))

// d1 and d2 are two sides of the rectangle

{ A=d1\*d2;dh=2.0\*d1\*d2/(d1+d2);

}

else if(shape.equals("concentric\_annulus"))

// d1 and d2 are inrernal and external diameters

//of the concentric annulus

{ dh=Math.abs(d2-d1);A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_segment"))

// d1 is diameter and d2 is the angle of the circular segment

{ dh=d1\*(1+Math.sin(2.0\*d2)/2.0/(pi-d2));A=pi\*dh\*dh/4.0;} else if(shape.equals("circular\_sector"))

//d1 is the diameter, d2 is the height of the circular section

{ dh=d2/(1+d2)\*d1; A=pi\*dh\*dh/4.0; } else if(shape.equals("right\_triangle"))

// d1 and d2 two sides of right triangle

{ dh=4.0\*d1\*d2/(d1+d2+ Math.pow((d1\*d1+d2\*d2),0.5)); A=d1\*d2/2.0;

}

eps=e; mass\_flow\_rate=m; P=pp;

KL=k;

}

public double get\_d1()

//getting the first dimension

{

return d1;

}

public double get\_d2()

//getting the first dimension

{

return d2;

}

public double velocity(double t)

{

// velocity of the pipe m/x double d =g.M/g.v(t,P);

return (double)(mass\_flow\_rate/d/A);

}

public double Re(double t)

{

// reynold's number double d=g.M/g.v(t,1.0); double vis=g.vis(t); double vv=velocity(t); double res=(dh\*vv\*d/vis); return res;

}

public double f(double t)

{

// this function calculated friction coefficient

// for both laminar and turnulent flow for various

// cross sections

double Rey=Re(t); //Reynold's number double C=64;

if(Rey<2000)

{

String c = "circular"; String r = "rectangular";

String co\_an = "concentric\_annulus"; String cir\_seg = "circular\_segment"; String cir\_sec = "circular\_sector"; String rig\_tri;

rig\_tri=new String("right\_triangle"); if(shape.equals(c))

//circular cross section, laminar flow

{ C=64.0; }

else if(shape.equals(r))

{

// Laminar rectangular channel friction factor double x=d1/d2;

if(x>1) x=1/x; double a1[]; a1=new double[6];

a1[0]= .959923253176985E+02; a1[1]= -.130364643234979E+03; a1[2]= .188053177394400E+03; a1[3]= -.166159826967448E+03 ; a1[4]= .952739160606451E+02; a1[5]= -.258949934514987E+02 ; C=a1[5];

for(int i=4;i>=0;i--) { C=C\*x+a1[i];}

}

else if(shape.equals(co\_an))

{

//concentric annulus laminar friction factor double x=d2/d2;

if(x>1) x=1/x; double a1[]; a1=new double[5];

a1[0]= .717068909524062E+02; a1[1]= .932037998655195E+03; a1[2]= -.947586946520906E+04; a1[3]= .204346570871727E+05; a1[4]= -.118665325115712E+05; C=a1[4];

for(int i=4;i>=0;i--) {C=C\*x+a1[i];}

}

else if(shape.equals(cir\_seg))

{

//circular segment laminar friction factor double a1[];

a1=new double[5];

a1[0]= .639999999999997E+02; a1[1]= -.162338014624008E+01; a1[2]= .162620455430327E+01; a1[3]= -.827251633703104E+00 ; a1[4]= .138590724447624E+00; C=a1[4];

for(int i=3;i>=0;i--) {C=C\*d2+a1[i];}

}

else if(shape.equals(cir\_sec))

{

//circular section laminar friction factor double a1[];

a1=new double[4];

a1[0]= .480000000000000E+02; a1[1]= .227909935649730E+02; a1[2]= -.134959915493607E+02; a1[3]= .325095868081599E+01; C=a1[3];

for(int i=2;i>=0;i--) {C=C\*d2+a1[i];}

}

else if(shape.equals(rig\_tri))

{

//right triangle laminar friction factor double a1[];

a1=new double[5];

a1[0]= .480001792918036E+02; a1[1]= .127207559700323E+02; a1[2]= -.108590518340935E+02; a1[4]= .158423382029386E+01; a1[5]= .115425033696128E+01; C=a1[5];

for(int i=4;i>=0;i--) {C=C\*d2+a1[i];}

}

return C/Rey;

}

else

{

double eod=eps/dh; double fc=fcol(Rey,eod); return fc;

}

}

public double lentry(double t)

{

//hydraulic entry length double Rey=Re(t); double le;

if(Rey<2000) le=0.06\*dh\*Rey;

else le=4.4\*dh\* Math.pow(Rey,(1.0/6.0)); return le;

}

public double dP(double t,double P)

{

// pipe pressure drop for the given length plus

// local KL factors double V=velocity(t);

return (KL+f(t)\*l/dh)\*(g.M/g.v(t,P))\*V\*V/2.0\*1e-5;

}

public double dP(double t)

{

double P=1;

// pipe pressure drop for the given length plus

// local KL factors double V=velocity(t);

return (KL+f(t)\*l/dh)\*(g.M/g.v(t,P))\*V\*V/2.0\*1e-5;

}

public double dPl\_e(double t,double P)

{

double Q=mass\_flow\_rate\*g.v(t,P)/g.M; double a=d1/2.0;

double b=d2/2.0; double x=0;

double pi=4.0\*Math.atan(1.0); for(int i=1;i<8000;i+=2)

{

x+=tanh(pi\*b\*i/d1)/(double)(i\*i\*i\*i\*i);

}

x=1.0-192.0\*a/(pi\*pi\*pi\*pi\*pi\*b)\*x;

double xx=b\*a\*a\*a/(6.0\*g.vis(t))\*x; xx=Q/xx\*l\*1e-5;

return xx;

}

void change\_length(double l1)

{ l=l1; }

void change\_mass\_flow\_rate(double mfr)

{ mass\_flow\_rate=mfr; }

void change\_P(double Pin)

//change inlet pressure

{ P=Pin; }

void change\_KL(double klin)

//change local pressure drop coefficient

{KL=klin; }

void change\_e(double e)

{ eps=e; }

public double hc(double t)

{

//convective heat transfer coefficient double Rey=Re(t);

double Nu=0; if(Rey<2000)

{

if(shape.equals("circular"))

{ Nu=3.66; }

else if(shape.equals("rectangular"))

{

// Laminar rectangular channel friction factor double x=d1/d2;

if(x>1) x=1/x; double a1[]; a1=new double[7];

a1[0]= .754000000000580E+01; a1[1]= -.193789080028494E+02; a1[2]= .333861212302836E+02; a1[3]= -.170496495070800E+02 ; a1[4]= -.304226226912837E+02; a1[5]= .495297442580787E+02; a1[6]= -.206246852871550E+02; Nu=a1[6];

for(int i=5;i>=0;i--) {Nu=Nu\*x+a1[i];}

}

else if(shape.equals("concentric\_annulus"))

{

double x=d2/d2; if(x>1) x=1/x; double a1[]; a1=new double[4];

a1[0]= .401698044199492E+01; a1[1]= .921803331830132E+00; a1[2]= -.307845192109755E+00; a1[3]= .229092480046921E+00; Nu=a1[3];

for(int i=2;i>=0;i--) {Nu=Nu\*x+a1[i];}

}

else if(shape.equals("circular\_segment"))

{ Nu=3.66;

}

else if(shape.equals("circular\_sector"))

{ Nu=3.66;

}

else if(shape.equals("right\_triangle"))

{ Nu=2.47;

}

}

else

{

double fr = f(t);

Rey = Re(t);

double Prt = g.Pr(t);

// Gnielinski equation

// Gnielinski, V, Int. Chem Eng., 16, 359,1976

// Valid for 2000 < Re <5e5, 0.5 < Pr < 2000 Nu=fr/8.0\*(Rey-1000.0)\*Prt/

(1+12.7\* Math.sqrt(fr/8.0)\*( Math.pow(Prt,(2.0/3.0))-1.0));

}

double kt = g.k(t);

// double Prt = g.Pr(t); double result; result=Nu\*kt/dh; return result;

}

public double Nu(double t)

//Nusselt's Number

{

return hc(t)\*dh/g.k(t);

}

} //end of class pipe

Program veri dosyasi Pipe.txt (sadece birkaç örnek durum listelenmistir.) example

circular

air 0.1

0.11

p18\_natural circular natural

18e-3

1.0589996e-4

p18\_air circular air

18e-3

4.244105e-3

pipe\_6 rectangular

prereformer\_output 30e-3 1.6e-3

8.727136e-4

veri dosyasi boru ismi, boru sekli(circular, rectangular..), gazkarisiminin adi(Gmix.txt dosyasinda tanimlanmis olmalidir), boru çapi veya boru en ve boyu ve akis debisi(kg/s) bilgilerini içerir. Bu programin kullanildigi çikti veren bir insan arayüzü programina göz atalim :

### Program 10.23 pipeProperty.java

//======================================================

// Thermodynamics package in java

// pipeProperty class to calculate pipe

// user interface (applet)

// Dr. Turhan Coban

// =====================================================

import pipe; import java.io.\*; import java.awt.\*;

import java.awt.event.\*; import java.applet.Applet; import java.util.\*;

public class pipeProperty extends Applet implements ActionListener,ItemListener

{

protected Panel inputPanel;

protected Label promptGmixName; // Label prompt GmixName protected Label promptTemperature; // Label prompt Temperature protected Label promptPressure; // Label prompt Pressure protected Label promptPipeLength; // Label prompt Pipe Length

protected Label promptShape; // Label prompt cross-section shape protected Label promptD1; // Label prompt first dimension protected Label promptD2; // Label prompt second dimension protected Label promptKL; // LP local PD coefficient

protected Label promptMass; // LP mass flow rate protected Label prompte; // LP pipe equivalent roughness protected Choice inputGmixName; // input GmixName protected TextField inputTemperature; // input Temperature protected TextField inputPressure; // input Pressure

protected TextField inputPipeLength; // input pipe length protected Choice inputShape;

protected String pipeShapeArray[]={"circular","rectangular", "concentric\_annulus","circular\_segment","circular\_sector","right\_triangle"}; protected TextField inputD1;

protected TextField inputD2; protected TextField inputKL; protected TextField inputMass; protected TextField inpute; double T;

double P; double TK;

double pipeLength; String pipeShape; double D1,D2; String GmixName; String pipeName;

double mass\_flow\_rate;

Gmix g1;

pipe p1; double KL; double e;

StringTokenizer token;

public void init()

{

//adding max-min prompts and input fields inputPanel=new Panel(); inputPanel.setLayout(new GridLayout(10,2)); GmixName=new String("natural");

try{

g1=new Gmix(GmixName);

token=new StringTokenizer(g1.readGmixNames());

} catch(IOException ioe) {System.out.println("IOException");} pipeName=new String("P18\_natural");

T=27.0;

TK=T+273;

P=1.0;

pipeLength=1.0; KL=0.0; D1=0.005; D2=0.005; e=0.00001;

mass\_flow\_rate=0.01;

pipeShape =new String("circular");

promptGmixName= new Label(" Gas mixture Name : "); inputGmixName=new Choice(); while(token.hasMoreTokens())

{

inputGmixName.add(token.nextToken());

}

//inputGmixName=new TextField(15);

promptTemperature= new Label(" Temperature (øC) : "); inputTemperature=new TextField(15);

promptPressure= new Label(" Pressure (Bars) : "); inputPressure=new TextField(15);

promptPipeLength= new Label(" Pipe Length, m : "); inputPipeLength=new TextField(15);

promptShape= new Label(" Pipe Shape : "); inputShape=new Choice();

for(int i=0;i<5;i++)

{

inputShape.add(pipeShapeArray[i]);

}

promptD1= new Label(" Pipe Dimension D1, m : "); inputD1= new TextField(15);

promptD2= new Label(" Pipe Dimension D2, m : "); inputD2= new TextField(15);

promptKL= new Label(" Pipe Local PD coefficient, : "); inputKL= new TextField(15);

promptMass= new Label(" mass flow rate , kg/s : "); inputMass= new TextField(15);

prompte= new Label(" Pipe equivalent roughness, m: "); inpute= new TextField(15);

//\*\*\*\*\*\*\*\* inputGmixName.addItemListener(this); inputTemperature.addActionListener(this); inputPressure.addActionListener(this); inputPipeLength.addActionListener(this);

inputShape.addItemListener(this); inputD1.addActionListener(this); inputD2.addActionListener(this); inputKL.addActionListener(this); inputMass.addActionListener(this); inpute.addActionListener(this);

//\*\*\*\*\*\*\*\* inputPanel.add(promptGmixName); inputPanel.add(inputGmixName); inputPanel.add(promptTemperature); inputPanel.add(inputTemperature); inputPanel.add(promptPressure); inputPanel.add(inputPressure); inputPanel.add(promptPipeLength); inputPanel.add(inputPipeLength); inputPanel.add(promptShape); inputPanel.add(inputShape); inputPanel.add(promptD1); inputPanel.add(inputD1); inputPanel.add(promptD2); inputPanel.add(inputD2); inputPanel.add(promptKL); inputPanel.add(inputKL); inputPanel.add(promptMass); inputPanel.add(inputMass); inputPanel.add(prompte); inputPanel.add(inpute);

inputTemperature.setText(Double.toString(T)); inputPressure.setText(Double.toString(P)); inputPipeLength.setText(Double.toString(pipeLength)); inputD1.setText(Double.toString(D1)); inputD2.setText(Double.toString(D2)); inputKL.setText(Double.toString(KL)); inputMass.setText(Double.toString(mass\_flow\_rate)); inpute.setText(Double.toString(e));

//inputGmixName.setText(GmixName); inputD1.setText(Double.toString(D1)); inputD2.setText(Double.toString(D2)); inputD2.setEditable(false); inputKL.setText(Double.toString(KL)); inputMass.setText(Double.toString(mass\_flow\_rate)); inpute.setText(Double.toString(e));

setLayout(new BorderLayout());

add( inputPanel,BorderLayout.NORTH); p1= new pipe(pipeName,

pipeShape, GmixName, D1,

D2,

mass\_flow\_rate, e,

P,

KL,

pipeLength);

}

public void itemStateChanged(ItemEvent ev)

{

//pipeShape=pipeShapeArray[inputShape.getSelectedIndex()];

pipeShape=inputShape.getSelectedItem(); if(pipeShape.equals("circular"))

{

inputD2.setEditable(false);

}

else

{

inputD2.setEditable(true);

}

GmixName=inputGmixName.getSelectedItem(); p1= new pipe(pipeName,

pipeShape, GmixName, D1,

D2,

mass\_flow\_rate, e,

P,

KL,

pipeLength); repaint(); showStatus(pipeShape);

}//end of itemStateChange

public void paint(Graphics g)

{

TK=T+273;

g.drawString("Pressure Drop , bars : ",20,250); g.drawString(""+p1.dP(TK,P),250,250); g.drawString("f friction : ",20,265); g.drawString(""+p1.f(TK),250,265); g.drawString("Velocity , m/s : ",20,280); g.drawString(""+p1.velocity(T),250,280);

g.drawString("Reynold's number Re : ",20,295); g.drawString(""+p1.Re(TK),250,295); g.drawString("Prandl's number Pr : ",20,310); g.drawString(""+p1.g.Pr(TK),250,2310); g.drawString("Nusselt's number Nu : ",20,325); g.drawString(""+p1.Nu(TK),250,325); g.drawString("Hydraulic entry length, m : ",20,340); g.drawString(""+p1.lentry(TK),250,340); g.drawString("Viscosity , N s/m^2 : ",20,355); g.drawString(""+p1.g.vis(TK),250,355); g.drawString(p1.g.toString("formula"),20,370); g.drawString(p1.g.toString("composition"),20,385); g.drawString("shape : "+pipeShape,20,400);

}//end of method

public void actionPerformed( ActionEvent ee)

{

GmixName=inputGmixName.getSelectedItem();

Double valTemp=new Double(inputTemperature.getText()); T=valTemp.doubleValue();

Double valPressure=new Double(inputPressure.getText()); P=valPressure.doubleValue();

Double valpipeLength=new Double(inputPipeLength.getText()); pipeLength=valpipeLength.doubleValue();

Double valD1=new Double(inputD1.getText());

D1=valD1.doubleValue();

Double valD2=new Double(inputD2.getText()); D2=valD2.doubleValue();

Double valKL=new Double(inputKL.getText()); KL=valKL.doubleValue();

Double valMass=new Double(inputMass.getText()); mass\_flow\_rate=valMass.doubleValue();

Double valuee=new Double(inpute.getText()); e=valuee.doubleValue();

p1= new pipe(pipeName, pipeShape, GmixName,

D1,

D2,

mass\_flow\_rate, e,

P,

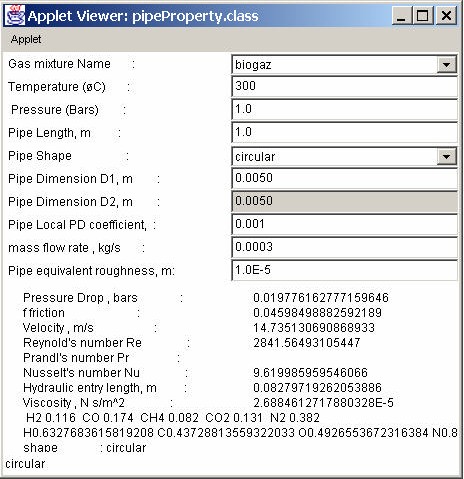
KL,

pipeLength); repaint();

}

}

Appletviewer olarak program çiktisi:



1. **Thermodinamik** özellikler serisinden bir adim daha ileri gidip gazlari kimyasal reaksiyonlara sokabiliriz. Bu sinifimiz Reaction.java ismini tasiyor. Gazlarin termodinamik reaksiyona girdiklerinde enerji dengelerini,adyabatik alev sicakliklarini ve reaksiyon sabitlerini (k) hesaplama kapasitesinde bir program. Program veridosyasi Reaction.txt adinda. Bu dosyada reaksiyonun ismi, toplam gaz sayisi, bu gazin reaksiyona giren ve reaksiyondan çikan moleküllerinin sayisi liste olarak veriliyor. Reaksiyon kütle denklik hesabi program tarafindan yapilmamaktadir. Dogru molekül sayilarinin verilmesi kullaniciya birakilmistir.

Bu dosyadan birkaç örnek veri setine göz atalim:

hidrojen 3

h2o 1 0

h2 0 1

o2 0 0.5

c2h4comb 5

c2h4 1 0

o2 3 0

n2 11.28 11.28

co2 0 2

h2o 0 2

LPG 6

c3h8 0.3 0

c4h10 0.7 0

o2 6.05 0

n2 22.7595 22.7595

co2 0 3.7

h2o 0 4.7

dogal\_1.05 8

ch4 0.906 0

c2h6 0.056 0

c3h8 8e-4 0

c4h10 2e-4 0

co2 0.018 0

n2 8.116375 8.116375

o2 2.155125 0.102625

h2o 0 2.017

dogal\_1.1 8

ch4 0.906 0

c2h6 0.056 0

c3h8 8e-4 0

c4h10 2e-4 0

co2 0.018 0

n2 8.5024405 8.5024405

o2 2.27575 0.20525

h2o 0 2.017

Programimizin listesi:

### Program 10.24 Reaction.java

import java.io.\*; import java.util.\*; import Text;

class Reaction

{

//======================================================//

// Dr. Turhan Coban //

// TUBITAK Marmara Research Center //

// Energy Systems and Environmental Research Institute //

// Phone : 90 (262) 641 2300-3917 //

// email : [turhan@mam.gov.tr](mailto:turhan@mam.gov.tr) //

//======================================================//

// Class Description :

//------------------------------------------------------

// This program calculates energy, adiabatic flame temperature

// entropy and gibbs free energy for a given reaction

// Basic reaction data is given in the file called Reaction.dat

// Format of the data :

// reactionname

// number\_of\_the\_chemicals

// gas\_name reactant\_mole prodoct\_mole

// .......

// gas\_name reactant\_mole prodoct\_mole

//-----------------

// sample data :

//-----------------

// c2h6combustion

// 5

// c2h6 1 0

// o2 3 0

// n2 11.28 11.28

// co2 0 2

// h2o 0 2

//------------------

// the available gas\_name's can be checked from data file

// Gas.dat

// Variable Identification

//-----------------------------------------------------

// ngas : number of gases in reaction (integer variable)

// reacname : reaction name (String variable)

// gasR,gasP: gas objects for reactants and products

// (array objects of class Gas)

// see Gas.java for details

// natom : number of atoms in the reaction

// nr,np : number of Moles of reactants and products

// (double vector array)

// nrt,npt : total number of moles for reactants and products

// reacFile : reaction file channel variable (Class File)

// atomList : list of the atoms in the reaction public int ngas;

String reacName; Gas gasR[],gasP[]; public int natom; Atom atomList[]; File reacFile; double nr[],np[]; double nrt,npt;

public Reaction(String name) throws IOException

{

reacFile=new File("Reaction.txt"); reacFile=new File(reacFile.getAbsolutePath()); try

{

DataInputStream cfin=new DataInputStream(new FileInputStream(reacFile)); int ierror=1;

natom=0; try

{

while(cfin!=null)

{

reacName=Text.readString(cfin); if(reacName.equals(name)) {ierror=0;break;}

}

}

catch(EOFException e\_eof)

{

System.out.println("error required gas mixture "+name+" is not found"); cfin.close();return;

}

ngas=Text.readInt(cfin); gasR=new Gas[ngas]; gasP=new Gas[ngas]; nr=new double[ngas]; np=new double[ngas]; nrt=0;

npt=0;

String tempGasName;

Double n1,n2;

for(int i=0;i<ngas;i++)

{

tempGasName=Text.readString(cfin); nr[i]=Text.readDouble(cfin); np[i]=Text.readDouble(cfin); nrt+=nr[i];

npt+=np[i];

gasR[i]=new Gas(tempGasName,nr[i]); gasP[i]=new Gas(tempGasName,np[i]);

}

}

catch(FileNotFoundException fnfe) {System.out.println("File Not Found");} arrange\_atoms();

} //end of constructor

public void arrange\_atoms() throws IOException

{

// atomic structure of the reactants int i,j;

for(i=0;i<ngas;i++)

{

for(j=0;j<gasR[i].natom;j++)

{

add\_atom(i,j);

}

}

}

public int add\_atom(int i,int j) throws IOException

{

//determine and order the atomic structure int k;

for(k=0;k<natom;k++)

{

if(gasR[i].atomList[j].symbol.equals(atomList[k].symbol))

{

atomList[k]=new Atom(atomList[k].symbol,atomList[k].N+gasR[i].atomList[j].N\*gasR[i].N); return 1;

}

}

Atom atomL[];

atomL=new Atom[natom+1];

for(k=0;k<natom;k++) atomL[k]=new Atom(atomList[k]);

atomL[natom]=new Atom(gasR[i].atomList[j].symbol,gasR[i].atomList[j].N\*gasR[i].N); atomList=atomL;

natom+=1; return 2;

}

public String toString()

{

// writes chemical reaction in chemistry norm String s="";

for(int i=0;i<ngas;i++)

{

if(nr[i]==1)

{ s+=gasR[i].toString(); } else if(nr[i]!=0)

{ s+=nr[i]+" "+gasR[i].toString(); }

if(i!=(ngas-1) && nr[i]!=0 ) s+=" + ";

}

if(s.endsWith(" + ")) s=s.substring(0,s.length()-3); s+=" --> ";

for(int i=0;i<ngas;i++)

{

if(np[i]==1)

{ s+=gasP[i].toString(); } else if(np[i]!=0)

{ s+=np[i]+" "+gasP[i].toString(); }

if(i!=(ngas-1) && np[i]!=0 ) s+=" + ";

}

if(s.endsWith(" + ")) s=s.substring(0,s.length()-3);

return s;

}

public String toString(String ch)

{

//return the c String s=""; int i,j;

if(ch.equals("name")) s=s+reacName+"\n";

else if(ch.equals("formula"))

{

s=toString();

}

else if(ch.equals("composition"))

{

for(i=0;i<natom;i++) s=s+atomList[i].toString()+" ";

}

return s;

}

public double H(double TR,double TP)

{

//reaction enthalpy double HH=0;

for(int i=0;i<ngas;i++)

{

HH+=gasP[i].HT(TP);

HH-=gasR[i].HT(TR);

}

return HH;

}

public double S(double TR,double TP,double pt)

{

//reaction entropy double SS=0;

for(int i=0;i<ngas;i++)

{

if(np[i]!=0) SS+=gasP[i].S(TP,np[i]/npt\*pt);

if(nr[i]!=0)

SS-=gasR[i].S(TR,nr[i]/nrt\*pt);

}

return SS;

}

public double S(double TR,double TP)

{

//reaction entropy double SS=0;

for(int i=0;i<ngas;i++)

{

if(np[i]!=0) SS+=gasP[i].S(TP,np[i]/npt);

if(nr[i]!=0)

SS-=gasR[i].S(TR,nr[i]/nrt);

}

return SS;

}

public double G(double TR,double TP,double pt)

{

// reaction gibbs energy double GG=0;

for(int i=0;i<ngas;i++)

{

if(np[i]!=0) GG+=gasP[i].GT(TP,np[i]/npt\*pt);

if(nr[i]!=0)

GG-=gasR[i].GT(TR,nr[i]/nrt\*pt);

}

return GG;

}

public double G(double TR,double TP)

{

// reaction gibbs energy double GG=0;

for(int i=0;i<ngas;i++)

{

if(np[i]!=0) GG+=gasP[i].GT(TP,np[i]/npt);

if(nr[i]!=0)

GG-=gasR[i].GT(TR,nr[i]/nrt);

}

return GG;

}

public double G0(double TR,double TP)

{

// reaction gibbs free energy double GG=0;

for(int i=0;i<ngas;i++)

{

if(np[i]!=0)

GG+=gasP[i].GT(TP);

if(nr[i]!=0)

GG-=gasR[i].GT(TR);

}

return GG;

}

public double K(double TR,double TP)

{

//Equilibrium costant. Usually TR and Tp will be in

// thesame temperature

return Math.exp(G0(TR,TP)/(-8.3145\*TP));

}

public double Taf(double TR,double xl,double xu)

{

// Adyabatic flame temperature using Bisection method

// bisection method to find roots of zero energy change

// Q=0

// defination of variables :

// xl : lower guess

// xu : upper guess

// xr : root estimate

// es : stopping criterion

// ea :approximate error

// maxit : maximum iterations

// iter : number of iteration double test;

double xr=0; double es,ea; double fxl,fxr;

int maxit=500,iter=0; es=0.000001;

ea=1.1\*es; while((ea>es)&&(iter<maxit))

{

xr=(xl+xu)/2.0; iter++; if((xl+xu)!=0)

{ ea=Math.abs((xu-xl)/(xl+xu))\*100;} fxl= H(TR,xl);

fxr= H(TR,xr); test= fxl\*fxr;

if(test==0.0) ea=0; else if(test<0.0) xu=xr; else

{

xl=xr;

}

} //end of while

return xr;

}

}

programimizi bir örnekte test edelim.

### Program 10.25 ReactionTest.java

import java.io.\*; import Reaction;

class ReactionTest

{

public static void main(String args[]) throws IOException

{

Reaction R2;

R2=new Reaction("hid"); System.out.println(R2.toString("name")); System.out.println(R2.toString("formula")); System.out.println(R2.toString("composition")); double Tadyabatikalev= R2.Taf(299,299, 3000)-273;

System.out.println("T adyabatik alev = "+Tadyabatikalev+" derece C");

}

}

program çiktisi :

hid

0.17 H2 + 1.72 O2 + 6.47 N2 --> 1.635 O2 + 6.47 N2 + 0.17 H2O H0.34 O3.44 N12.94

T adyabatik alev = 194.51268483046442 derece C

1. **Termodinamik** serisinden bir diger program. Bu program gerçek gazlarin termodinamik özelliklerini hesapliyor. Programin ismi LeeKesler.java, veri dosyasi RealGas.txt

Program Listesi :

//======================================================

// Thermodynamic Package in java

// Class Lee Kesler Properties of real gases

// Dr. Turhan Coban

// TUBITAK Marmara Research Center

// Energy Systems and Environmental Research Institute

// email : [turhan@mam.gov.tr](mailto:turhan@mam.gov.tr)

// File Name : LeeKesler.java

// This file contains the LeeKesler class

// this class sets basic properties of real gases

// required data is read from LeeKesler.txt

// =====================================================

// Description : This file contains the LeeKesler class

// class LeeKesler calculates thermophysical properties of

// perfect gasses

// DATA FILE DEFINATION

// datas are written in the data file "RealGas.txt"

//============================================================

// VARIABLE IDENTIFICATION

//============================================================

// GasEquation : String type variable

// values : "LeeKesler"

//

//

// Formula : Fluid chemical formula

// Name : Fluid name

// atomList : Array of atom class, structral information

// natom : number of atoms

// M : Molecular weight as calculated from atomic structure

// through class atom and atomList

// MolWt : Molecular Weigth from data file RealGas

// Tfp : Normal freezing point (1 atm), K

// Tb : Normal Boiling point (1 atm), K

// Tc : Critical temperature, K

// Pc : Critical pressure, bar

// Vc : Critical volume, cm^3/mole

// Zc : Critical compresibility factor PcVc/RTc

// Omega : Pitzer's accentric factor

// Dipm : Dipole moment, debyes

// CPVAPA,CPVAPB,CPVAPC,CPVAPD : constants to calculate the

// isobaric heat capacity of the ideal gas Cp kJ/(kmolK)

// DELHF : formation enthalpy, 298.2 K, 1.01325 bar, KJ/kmol

// DELGF : formation gibbs free energy, 298.2 K, 1.01325 bar,KJ/kmol

// Vpeq : vapor pressure equation number

// VPA,VPB,VPC,VPD : vapor pressure equation coefficients

// Tmin : minimum twmp. K

// Tmax : maximum temp. K

// LDEN : liquid density, g/cm^3

// TDEN : temperature that liquid density is given, K

//============================================================

// Reference : The Properties of Gases & Liquids,

// Robert C. Reid, John M. Prausnitz, Bruce E. Pouling

// McGraw-Hill Book Company, ISBN 0-07-051799-1

//==============================================

import java.io.\*; import Text; import Atom; import complex;

//==============================================

// implementation of constructors functions

// for class cubicEquationofState

public class LeeKesler

{

String GasEquation; int natom;

Atom atomList[]; int No;

String Formula,Name;

double M,MolWt,Tfp,Tb,Tc,Pc,Vc,Zc,Omega,Dipm;

double CPVAPA,CPVAPB,CPVAPC,CPVAPD,DELHF,DELGF;

int VPEq;

double VPA,VPB,VPC,VPD,Tmin,Tmax,LDEN,TDEN;

int ierror;

BufferedReader fin;

double R=8314.5; //(J/kmolK) double a,b,u,w;

//000000

//================================//

// Equation of State coefficients //

//================================//

//Simple Fluid (omega=0) :

//------------------------

double b0[]=new double[4]; double c0[]=new double[4]; double d0[]=new double[2]; double beta0;

double gama0; double omega0;

//Reference Fluid (n-octane) :

//------------------------

double bR[]=new double[5]; double cR[]=new double[5]; double dR[]=new double[3]; double betaR;

double gamaR;

public LeeKesler(String s1,double le) throws IOException

{

GasEquation="Peng Robinson";

//setting the simple fluid coefficients : b0[0]=0.1181193;

b0[1]=0.265728; b0[2]=0.15479; b0[3]=0.030323; c0[0]=0.0236744; c0[1]=0.0186984; c0[2]=0.0; c0[3]=0.042724; d0[0]=0.155488e-4; d0[1]=0.623689e-4; beta0=0.65392; gama0=0.060167;

omega0=0.3978;

//setting the reference fluid coefficients : bR[0]=0.2026579;

bR[1]=0.331511; bR[2]=0.027655; bR[3]=0.203488; cR[0]=0.0313385; cR[1]=0.0503618; cR[2]=0.016901; cR[3]=0.041577; dR[0]=0.48736e-4; dR[1]=0.0740336e-4;

betaR=1.226; gamaR=0.03754;

int i;

String atomName[]; double atomN[];

int anatom; ierror=1;

String tempName="";

String tempFormula="";

String aFormula,aName;

double aMolWt,aTfp,aTb,aTc,aPc,aVc,aZc,aOmega,aDipm ;

double aCPVAPA,aCPVAPB,aCPVAPC,aCPVAPD,aDELHF,aDELGF;

int aVPEq;

double aVPA,aVPB,aVPC,aVPD,aTmin,aTmax,aLDEN,aTDEN ;

try{

fin=new BufferedReader(new FileReader("RealGas.txt"));

} catch(IOException e)

{

System.err.println("Error Opening File RealGas.dat\n"+e.toString()); System.exit(1);

}

try{

while(fin!=null)

{

aFormula=Text.readString(fin); aName=Text.readString(fin); anatom=Text.readInt(fin); atomName=new String[anatom]; atomN=new double[anatom]; for(i=0;i<anatom;i++)

{

atomName[i]=Text.readString(fin); atomN[i]=Text.readDouble(fin);

}

aMolWt=Text.readDouble(fin); aTfp=Text.readDouble(fin); aTb=Text.readDouble(fin); aTc=Text.readDouble(fin); aPc=Text.readDouble(fin); aVc=Text.readDouble(fin); aZc=Text.readDouble(fin); aOmega=Text.readDouble(fin); aDipm=Text.readDouble(fin); aCPVAPA=Text.readDouble(fin); aCPVAPB=Text.readDouble(fin); aCPVAPC=Text.readDouble(fin); aCPVAPD=Text.readDouble(fin); aDELHF=Text.readDouble(fin); aDELGF=Text.readDouble(fin); aVPEq=Text.readInt(fin); aVPA=Text.readDouble(fin); aVPB=Text.readDouble(fin); aVPC=Text.readDouble(fin); aVPD=Text.readDouble(fin); aTmin=Text.readDouble(fin); aTmax=Text.readDouble(fin); aLDEN=Text.readDouble(fin); aTDEN=Text.readDouble(fin);

if (aName.equals(s1) || aFormula.equals(s1) )

{

Name=aName;

Formula=aFormula;

MolWt=aMolWt;

Tfp=aTfp;

Tb=aTb; Tc=aTc; Pc=aPc; Vc=aVc; Zc=aZc;

Omega=aOmega;

Dipm=aDipm;

CPVAPA=aCPVAPA;

CPVAPB=aCPVAPB;

CPVAPC=aCPVAPC;

CPVAPD=aCPVAPD;

DELHF=aDELHF;

DELGF=aDELGF;

VPEq=aVPEq;

VPA=aVPA;

VPB=aVPB;

VPC=aVPC;

VPD=aVPD;

Tmin=aTmin;

Tmax=aTmax;

LDEN=aLDEN;

TDEN=aTDEN;

natom=anatom; atomList=new Atom[natom]; M=0;

for(i=0;i<natom;i++)

{

atomList[i]=new Atom(atomName[i],atomN[i]); M+=atomList[i].mass;

}

ierror=0; break;

}

}

} catch(EOFException e\_eof)

{

System.out.println("error required gas "+s1+" is not found"); fin.close();return;

}

fin.close();

}

public String toString()

{

//return the chemical symbol of the gas String s="";

for(int i=0;i<natom;i++) s=s+atomList[i].toString();

return s;

}

public double Z(double T, double V)

{

double Tr=T/Tc; double Vr=V/Vc; double B0,C0,D0; double BR,CR,DR; double Z0,ZR,ZX;

B0=b0[0]-b0[1]/Tr-b0[2]/(Tr\*Tr)-b0[3]/(Tr\*Tr\*Tr); C0=c0[0]-c0[1]/Tr+c0[2]/(Tr\*Tr\*Tr); D0=d0[0]+d0[1]/Tr;

Z0=1+B0/Vr+C0/(Vr\*Vr)+D0/(Vr\*Vr\*Vr\*Vr\*Vr)

+ c0[3]/(Tr\*Tr\*Tr\*Vr\*Vr)\*(beta0+gama0/(Vr\*Vr))\*Math.exp(-gama0/(Vr\*Vr)); BR=bR[0]-bR[1]/Tr-bR[2]/(Tr\*Tr)-bR[3]/(Tr\*Tr\*Tr);

CR=cR[0]-cR[1]/Tr+cR[2]/(Tr\*Tr\*Tr); DR=dR[0]+dR[1]/Tr;

ZR=1+BR/Vr+CR/(Vr\*Vr)+DR/(Vr\*Vr\*Vr\*Vr\*Vr)

+ cR[3]/(Tr\*Tr\*Tr\*Vr\*Vr)\*(betaR+gamaR/(Vr\*Vr))\*Math.exp(-gamaR/(Vr\*Vr));

ZX=Z0+Omega/0.3978\*(ZR-Z0);

return ZX;

}

double P(double T, double V)

{

double Tr=T/Tc; double Vr=V/Vc; return Z(T,V)\*Tr/Vr;

}

/\*

public double V(double T, double P)

{

return Z(T,P)\*R\*T/(P\*1e5);

}

public double Pvp(double T)

{

switch ( VPEq) double cPvp; case 1:

{

cPvp=Pc\*Math.exp((VPA\*x+VPB\*Math.pow(x,1.5)+VPC\*x\*x\*x+VPD\*x\*x\*x\*x\*x\*x)/(1-x)); break;

}

case 2:

{

cPvp=VPA-VPB/T+VPC\*Math.log(T)+VPD break;

}

case 3:

{

break;

}

return cPvp;

}

\*/

}

Programin veri dosyasindan bazi gazlarin özelliklerini örnek olarak verirsek:

H2 Hydrogen 1

H 2

2.016 14 20.3 33 12.9 64.3 0.303 -0.216 0

27.14 9.274e-3 -1.381e-5 7.645e-9 0 0

1 -5.57929 2.60012 -0.85506 1.70503 14 33 0.071 20

H2O Water 2

H 2

O 1

18.015 273.2 373.2 647.3 221.2 57.1 0.235 0.334 1.8

3.224 1.924e-3 1.055e-5 -3.596e-9 -2.42e5 -2.288e5

1 -7.76451 1.45838 -2.7758 -1.23303 275 647.3 0.998 293

CH4 Methane 2

C 1

H 4

16.043 90.7 111.6 190.4 46 99.2 0.288 0.011 0

19.25 5.213e-2 1.197e-5 -1.132e-8 -7.49e4 -5.087e4

1 -6.00435 1.1885 -0.83408 -1.22833 91 190.4 0.425 112

Sanirim son birkaç örnek gerçek programlama ortaminda veri dosyalarinin kullanilmasi hakkinda daha iyi bir fikir vermistir. Buradaki tüm örnekler sirali (sequential) tip veri dosyasi islemektedir. Ayni islemler veri tabani kullanilarak da yapilabilirdi.

### BÖLÜM 11 GELISMIS JAVA BILGI ISLEME YAPILARI

* 1. **GIRIS**

Bu bölümde çesitli ileri bilgi isleme ve boyut olusturma kavramlarina göz atacagiz.

### STRINGTOKENIZER SINIFI

Biz bir cümle okudugumuzda beynimiz cümleyi otomatik olarak kelimelerine ayirir. Java Dilinde bu görevi StringTokenizer sinifi yerine getirir. StringTokenizer sinifinin yapisi :

Public class StringTokenizer implements Enumeration { private int currentPosition;

private int maxPosition; private String str;

private String delimiters; private boolean retTokens;

public StringTokenizer(String str, String delim, boolean returnTokens) public StringTokenizer(String str, String delim)

public StringTokenizer(String str) private void skipDelimiters() public boolean hasMoreTokens() public String nextToken()

public String nextToken(String delim) public boolean hasMoreElements() public Object nextElement()

public int countTokens()

}

StringTokenizer sifini açiklamak için önce küçük bir program verelim :

### Program 11.1 : StringTokenizerTest.java programi

import java.io.\*; import java.util.\*;

public class StringTokenizerTest

{

public static void main(String arg[])

{

String s="Ali veli 49 elli"; System.out.println("cumle : "+s); **StringTokenizer t=new StringTokenizer(s);**

System.out.println("kelime sayisi : "+**t.countTokens()**); int i=0;

while(**t.hasMoreTokens()**)

{

System.out.println("kelime indeksi : "+(i++)+" kelime : "+**t.nextToken()**);

}

}

}

Programdan da görülecegi gibi StringTokenizer sinifi

### StringTokenizer t=new StringTokenizer(s);

deyimiyle tanimlanmistir. Ayrica stringin içindeki cümle sayisi **t.countTokens();** metotuyla çagirilmis, **t.hasMoreTokens()** sinifi ise daha fazla kelime kalip kalmadiginin boolean kontrolu metotu olarak çagirilmistir. **t.nextToken()** metotu o andaki kelimeyi disariya (String cinsinden) aktarirken indeksi de (currentPosition) bir sonraki kelimeye kaydirir.

Program çiktisi :

cumle : Ali veli 49 elli kelime sayisi : 4

kelime indeksi : 0 kelime : Ali kelime indeksi : 1 kelime : veli kelime indeksi : 2 kelime : 49 kelime indeksi : 3 kelime : elli olacaktir.

StringTokenizer sinifinin kullanilmasiyla ilgili bir örnek daha verelim. Bu örnekte ninci dereceden bir polinomun kokleri Matrix.java kütüphanesi (detaylar için bakiniz bölüm 12) kullanilarak çözülür. girdi alaninda tüm polinom katsayilari bir alan içinde tanimlanmakta ve sonra bu bilesik alandan StringTokenizer sinifi yardimi ile katsayilar tek tek çekilmektedir.

### Program 11.2 kokN.java progami, n'inci dereceden polinomun köklerini hesaplar. Tek bir string olarak girilen katsayilar matrisinden katsayilari StringTokenizer sinifini kullanarak çeker.

import java.util.\*; import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*; import Matrix;

public class kokN extends Applet implements ActionListener

{

private Label prompt1,prompt2; private TextField input; TextArea t;

Panel YaziPaneli; int n;

String s;

public void init()

{

setBackground(Color.lightGray); YaziPaneli=new Panel();

YaziPaneli.setFont(new Font("Serif",Font.BOLD,12)); YaziPaneli.setLayout( new GridLayout(3,1) );

t=new TextArea(8,47);

prompt1= new Label("a[0]+a[1]\*x+...+a[n]\*x^n=0");

prompt2= new Label("n inci dereceden polinomun katsayilarini giriniz : "); input = new TextField(30);

YaziPaneli.add(prompt1); YaziPaneli.add(prompt2); YaziPaneli.add(input); add(YaziPaneli);

add(t); input.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

s=input.getText();

### StringTokenizer token=new StringTokenizer(s);

t.setText(""); **n=token.countTokens()-1;** int m=n+1;

double a[]=new double[m]; complex z[]=new complex[n]; for(int i=0;i<n;i++)

{

z[i]=new complex();

}

int j=0;

while(**token.hasMoreTokens()**)

{

Double ax=new Double(token.nextToken()); a[j++]=ax.doubleValue();

}

z=Matrix.poly\_rootsC(a); int i=0;

t.setText(Matrix.toStringT(z)); input.setText("");

}

}

### 11001.JPG



**Sekil 11.1 StringTokenizer sinifini ve matrix sinifini kullanarak ninci dereceden bir polinomun koklerini hesaplayan kokN.java programinin applet çiktisi**

Ayni programin swinf JFrame versiyonu :

### Program 11.3 kokNSWF\_2000.java progami, n'inci dereceden polinomun köklerini hesaplar. Tek bir string olarak girilen katsayilar matrisinden katsayilari StringTokenizer sinifini kullanarak çeker.

import java.util.\*; import java.awt.\*; import javax.swing.\*; import java.awt.event.\*; import Matrix;

public class kokNSWF\_2000 extends JFrame implements ActionListener

{

private JLabel prompt1,prompt2; private JTextField input; JTextArea t;

JPanel YaziPaneli; int n;

String s;

Container c;

public kokNSWF\_2000()

{

super("n inci dereceden polinomun kökleri"); c=getContentPane();

c.setLayout(new FlowLayout()); YaziPaneli=new JPanel();

YaziPaneli.setFont(new Font("Serif",Font.BOLD,12)); YaziPaneli.setLayout( new GridLayout(3,1) );

t=new JTextArea(); t.setBackground(c.getBackground());

prompt1= new JLabel("a[0]+a[1]\*x+...+a[n]\*x^n=0");

prompt2= new JLabel("n'inci dereceden polinomun katsayilarini giriniz : "); input = new JTextField(30);

input.setBackground(c.getBackground()); YaziPaneli.add(prompt1); YaziPaneli.add(prompt2); YaziPaneli.add(input); c.add(YaziPaneli);

c.add(t); input.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

s=input.getText();

StringTokenizer token=new StringTokenizer(s); t.setText("");

n=token.countTokens()-1; int m=n+1;

double a[]=new double[m]; complex z[]=new complex[n]; for(int i=0;i<n;i++)

{

z[i]=new complex();

}

int j=0;

while(token.hasMoreTokens())

{

Double ax=new Double(token.nextToken()); a[j++]=ax.doubleValue();

}

z=Matrix.poly\_rootsC(a); int i=0;

t.setText(Matrix.toStringT(z)); input.setText("");

}

public static void main(String[] args)

{

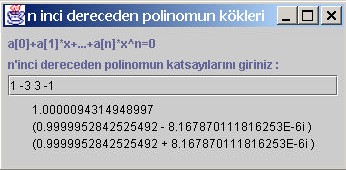
kokNSWF\_2000 pencere= new kokNSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,300);

pencere.setVisible(true);

}

}

### 11002.JPG



**Sekil 11.2 StringTokenizer sinifini ve matrix sinifini kullanarak ninci dereceden bir polinomun koklerini hesaplayan kokNSWF\_2000.java programinin JFrame çiktisi**

StringTokenizer sinifinin alt degiskenleri :

**nval** : eger seçilmis olan token bir sayi ise nval bu sayinin degerini içerir.

### sval

eger seçilmis olan token bir kelime(String) ise sval bu String degiskenini içerir

### TT\_EOF

girdi stringinin sonuna ulasildigini gösterir

### TT\_EOL

satir sonu okundugunu bildirir

### TT\_NUMBER

bir sayi tokeni okundugunu bildirir

### TT\_WORD

bir kelime(String) token'i okundugunu bildirir..

### ttype

nextToken metotu çagrildikrtan sonra bu degisken en son okunan token (kelimenin) türünü bildiri

StringTokenizer sinifinin metotlari : kurucu metot : **StreamTokenizer(Reader) eolIsSignificant(boolean)**

satir sonu kumutunun yeni kelime tanimlayip tanimlamadigini belirler.

### lineno()

satir sayisini verir.

### lowerCaseMode(boolean)

kelimelerin otomatik olarak küçük harfe çevrilip çevrilmeyecegini tanimlar

### nextToken()

giris Stringinin içindeki bir sonraki kelimeyi bulur.

### ordinaryChar(int)

Tokanizer'in normalde kontrol için kullandigi cosluk,satir sonu gibi özel kontrol karakterlerinin normal karakter olarak okunmasini saglar

### parseNumbers()

Direk olarak bir sayi dizininin geldigini haber verir

### pushBack()

bir sonraki nextToken çagirilmasinda kelimenin su andaki kelime olarak kalmasini saglar.

### quoteChar(int)

eger " karecteri varsa bunun içindeki karakterleri bir String degiskeni olarak algilar.

### slashSlashComments(boolean)

Tokenizer'in C++-tipi (//.....)comment **deyimlerini** comment olarak taniyim tanimiyacagini belirler. slashStarComments(boolean)

Tokenizer'in C-tipi comment (/\*...\*/) deyimlerini comment olarak taniyip tanimiyacagini belirler.

### toString()

o anda seçilmis olan klimeyi (token) gönderir.

### whitespaceChars(int, int)

birinciinteger <= c <= ikinciinteger arasindaki karekterleri bosluk olarak tanimlar

### wordChars(int, int)

birinciinteger <= c <= ikinciinteger arasindaki karekterleri tek kelime olarak tanimlar

### STRING BUFFER SINIFI

Yeri gelmisken **String** sinifinin kardes sinifi, **StringBuffer** sinifindan da bahsedelim. StringBuffer genel olarak String sinifinin aynidir, fakat genellikle dinamik olarak program içinde degeri sikça degisen Stringler için kullanilir. String ise genellikle statik olarak verilen String degiskenleri içindir. StringBuffer sinifina Append metotuyla dinamik olarak ilave yapabiliriz.

StringBuffer sinifinin metod yapisi söyledir :

package java.lang;

public final class StringBuffer implements java.io.Serializable { private int count;

private boolean shared;

static final long serialVersionUID = 3388685877147921107L; public StringBuffer

public StringBuffer(int length) public StringBuffer(String str) public int length()

public int capacity() private final void copy()

public synchronized void ensureCapacity(int minimumCapacity) private void expandCapacity(int minimumCapacity)

public synchronized void setLength(int newLength) public synchronized char charAt(int index)

public synchronized void getChars(int srcBegin, int srcEnd, char dst[], int dstBegin) public synchronized void setCharAt(int index, char ch)

public synchronized StringBuffer append(Object obj) public synchronized StringBuffer append(String str) public synchronized StringBuffer append(char str[])

public synchronized StringBuffer append(char str[], int offset, int len) public StringBuffer append(boolean b)

public synchronized StringBuffer append(char c) String) public StringBuffer append(int i)

public StringBuffer append(long l) public StringBuffer append(float f) public StringBuffer append(double d)

public synchronized StringBuffer delete(int start, int end) public synchronized StringBuffer deleteCharAt(int index)

public synchronized StringBuffer replace(int start, int end, String str) public String substring(int start)

public synchronized String substring(int start, int end)

public synchronized StringBuffer insert(int index, char str[], int offset, int len) public synchronized StringBuffer insert(int offset, Object obj)

public synchronized StringBuffer insert(int offset, char str[]) public StringBuffer insert(int offset, boolean b)

public synchronized StringBuffer insert(int offset, char c) public StringBuffer insert(int offset, int i)

public StringBuffer insert(int offset, long l) public StringBuffer insert(int offset, float f) public StringBuffer insert(int offset, double d) public synchronized StringBuffer reverse() public String toString

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException

}

Simdi küçük bir test programinda StringBuffer sinifini kullanalim :

### Program 11.4 TestString.java, giren String degiskenini tersine çevirir.

class TersString {

public static String reverseIt(String source) { int i, len = source.length();

StringBuffer dest = new StringBuffer(len); for (i = (len - 1); i = 0; i--) {

dest.append(source.charAt(i));

}

return dest.toString();

}

}

diger bir StringBuffer metotu **insert** metotudur. Bu metot i inci elementten sonra verilen yeni parçayi ilave edecektir.

### Program 11.5 javaKahvesi.java, StringBuffer, insert metodu örnegi

import java.io.\*; class **javaKahvesi** {

public static void main (String args[])

{

StringBuffer sb = new StringBuffer("**Java kahvesi ic!**"); sb.insert(12, "**ni sicak**"); System.out.println(sb.toString());

}

}

bu programi çalistirdigimizda

### Java kahvesini sicak iç!

mesajini verecektir.

### VECTOR SINIFI

Daha önce boyutlu degiskenleri görmüstük. Boyutlu degiskenlerin boyutlari bilgisayar belleginde olusturulurken birbiri ardina sirayla olusturulur.

Örnegin **double b[]=new double[3];**

Deyimi bilgisayar belleginde

### Tablo 11.1 boyutlu degiskenlerin bilgisayar belleginde yerlesimi

|  |  |
| --- | --- |
| 1023400 | b[0] |
| 1023464 | b[1] |
| 1023528 | b[2] |

yerlerini birbirini izleyen adreslerde ayirir.

Buradaki sorun eger bilgisayar bellegindeki toplam yer degistirilmek istenirse bunun zor olmasidir. Ayni zamanda bilgisayar boyutlu degiskenin adreslerini bilgisayar hafizasinda bir referans tablosu olarak tuttugundan boyut degistirme bilgisayar hafizasini da zorlayan bir islemdir. Pratik olarak yapilan uygulama yeni bir boyutlu degisken olusturup adresleri degistirmektir.

### double c[]=new double[5]; for(int i=0,i<b.length;i++)

**{ c[i]=b[i];} b=c;**

Daha gelismis bir bilgi yapisi olarak Vector sinifini kullanabiliriz. Vector sinifi **java.util** kütüphanesinde tanimlanmistir. Vector sinifinda her bilgi elemani ayni zamanda bir sonraki bilgi elemaninin adresini tasir. Boylece istenildiginde dizinin son elemanina yeni bir adres ilave edilerek kolayca yeni elemanlar ilave edilebilir.

### 11003.JPG



**Sekil 11.3 Vectör sinifinin hafizada yerlesimi**

Dizinin arasindan herhangibir elemani da kalaylikla sadece referans adresini degistirerek silebiliriz.

### Vector sinifinin tanimi :

public class Vector extends AbstractList implements List, Cloneable,

java.io.Serializable {

protected int elementCount;

protected int capacityIncrement;

private static final long serialVersionUID = -2767605614048989439L; public Vector(int initialCapacity, int capacityIncrement);

public Vector(int initialCapacity); public Vector();

public Vector(Collection c);

public synchronized void copyInto(Object anArray[]); public synchronized void trimToSize();

public synchronized void ensureCapacity(int minCapacity); private void ensureCapacityHelper(int minCapacity); public synchronized void setSize(int newSize);

public int capacity(); public boolean isEmpty();

public Enumeration elements(); public boolean hasMoreElements(); public Object nextElement();

public boolean contains(Object elem); public int indexOf(Object elem);

public synchronized int indexOf(Object elem, int index); public int lastIndexOf(Object elem);

public synchronized int lastIndexOf(Object elem, int index) public synchronized Object elementAt(int index);

public synchronized Object firstElement(); public synchronized Object lastElement();

public synchronized void setElementAt(Object obj, int index); public synchronized void removeElementAt(int index);

public synchronized void insertElementAt(Object obj, int index); public synchronized void addElement(Object obj) ;

public synchronized boolean removeElement(Object obj); public synchronized void removeAllElements();

public synchronized Object clone(); public synchronized Object[] toArray();

public synchronized Object[] toArray(Object a[]); public boolean remove(Object o);

public void add(int index, Object element); public synchronized Object remove(int index); public void clear();

public synchronized boolean containsAll(Collection c); public synchronized boolean addAll(Collection c); public synchronized boolean removeAll(Collection c); public synchronized boolean retainAll(Collection c);

public synchronized boolean addAll(int index, Collection c); public synchronized boolean equals(Object o);

public synchronized int hashCode(); public synchronized String toString();

public List subList(int fromIndex, int toIndex);

protected void removeRange(int fromIndex, int toIndex);

}

Vector sinifinda tanimlanan metotlarin bazilarinin anlamlari sunlardir :

### addElement(Object)

Vektöre object türü (herhangi bir sinif olabilir, vectörün hepsi bu sinfta tanimlanmis olmalidir.) bir eleman ekler.

### capacity()

vektörün toplam kapasitesini verir.

### clone()

vectorun bir kopye vectörünü verir.

### contains(Object)

Vectörün içinde sorgulanan elemanin olup olmadigini kontrol eder.

### copyInto(Object[])

vectorü boyutlu degiskene aktarir.

### elementAt(int indeks)

Verilen indeksteki elemani verir..

### elements()

Tüm vektörü enumeration sinifina aktarir.

### ensureCapacity(int yeni\_kapasite)

eger yeni\_kapasite vector kapasitesinin üstündeyse vector yeni\_kapasite’ye aktarilir.

### firstElement()

Vectörün ilk elemanini verir.

### indexOf(Object)

Object’in indeks degerini verir

### indexOf(Object, int indeks)

Object’in indeks degerini verir, arama indeks degiskeninden baslayarak yapilir.

**insertElementAt(Object, int indeks)** Vectörün ideks noktasina Object ilave edilir. **isEmpty()**

Vectörün bos olup olmadigi kontrol edilir.

### lastElement()

vectörün son elemaninin indeksini verir.

### lastIndexOf(Object)

Object’in vectördeki son tekrarinin indeksini verir.

### lastIndexOf(Object, int indeks)

indeks ten baslayarak geriye dogru Objectin ilk tekrarlandigi yeri bulur.

**removeAllElements()** Vectördeki bütün elemanlari siler. **removeElement(Object)**

Vectördeki Object elemanini siler. **removeElementAt(int indeks)** indeksle verilen elemani siler. **setElementAt(Object, int indeks)** Objectin ideksini indeks olarak belirler. **setSize(int)**

Sets the size of this vector.

### size()

Vectördeki toplam eleman sayisini verir.

### toString()

Vectörün String esitini verir.

### trimToSize()

Vectörün boyutunu dolu olan elemanlarinin boyutuna kisaltir.

Asagida bu metotlarin önemlilerinden bazilarini kullanan bir örnek program verilmistir.

### Program 11.6 : VectorTesti.java programi, vector sinifinin kullanimi

import java.util.\*; import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

public class VectorTesti extends Applet implements ActionListener

{

private Vector v; private Label prompt; private TextField input;

private Button addBtn, removeBtn, firstBtn, lastBtn, emptyBtn, containsBtn, locationBtn, trimBtn, statsBtn, displayBtn;

Panel YaziPaneli; public void init()

{

setBackground(Color.lightGray); YaziPaneli=new Panel();

YaziPaneli.setFont(new Font("Serif",Font.BOLD,12)); YaziPaneli.setLayout( new GridLayout(5,2) );

v = new Vector(1);

prompt= new Label("Bir String degiskeni giriniz "); input = new TextField(10);

addBtn = new Button("Ekle"); removeBtn = new Button("Cikar"); firstBtn = new Button("Ilk"); lastBtn = new Button("Son");

emptyBtn = new Button("Bos mu ?"); containsBtn = new Button("Listede varmi?"); locationBtn = new Button("Adresi");

trimBtn = new Button("Kes"); statsBtn = new Button("Istatistikler");

displayBtn = new Button("Listeyi Goster"); add(prompt);

add(input); addBtn.addActionListener(this); YaziPaneli.add(addBtn); removeBtn.addActionListener(this); YaziPaneli.add(removeBtn); firstBtn.addActionListener(this); YaziPaneli.add(firstBtn); lastBtn.addActionListener(this); YaziPaneli.add(lastBtn); emptyBtn.addActionListener(this); YaziPaneli.add(emptyBtn); containsBtn.addActionListener(this); YaziPaneli.add(containsBtn); locationBtn.addActionListener(this); YaziPaneli.add(locationBtn); trimBtn.addActionListener(this); YaziPaneli.add(trimBtn); statsBtn.addActionListener(this); YaziPaneli.add(statsBtn); displayBtn.addActionListener(this); YaziPaneli.add(displayBtn); add(YaziPaneli);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==addBtn)

{

v.addElement(input.getText());

showStatus("Listenin sonuna eklendi: "+input.getText());

}

else if(e.getSource()==removeBtn)

{

if(v.removeElement(input.getText())) showStatus("Silindi :"+input.getText()); else

showStatus(input.getText()+"not in vector");

}

else if(e.getSource() == firstBtn)

{

try{

showStatus("Ilk eleman : "+v.firstElement());

}

catch(NoSuchElementException exception)

{

showStatus(exception.toString() );

}

}

else if(e.getSource()==lastBtn)

{

try{

showStatus("Son eleman : "+v.lastElement());

}

catch(NoSuchElementException exception)

{

showStatus(exception.toString() );

}

}

else if(e.getSource() == emptyBtn)

{

showStatus(v.isEmpty()? "Vector bos ":"vector dolu ");

}

else if(e.getSource()==containsBtn)

{

String searchKey = input.getText(); if(v.contains(searchKey) )

showStatus("Vektor elemanlar• " +searchKey); else

showStatus("Aranan eleman Vektorde bulunamadi "+searchKey);

}

else if(e.getSource()==locationBtn)

{

showStatus("Eleman "+v.indexOf(input.getText() )+" pozisyonunda bulundu");

}

else if(e.getSource()==trimBtn)

{

v.trimToSize();

showStatus("Vector boyu eleman boyuna k• salt• ld• ");

}

else if(e.getSource()== statsBtn)

{

showStatus("Boyut = "+v.size()+"; Toplam kapasite = "+v.capacity());

}

else if(e.getSource()== displayBtn)

{

Enumeration enum=v.elements(); StringBuffer buf = new StringBuffer(); while(enum.hasMoreElements())

{

buf.append(enum.nextElement() ); buf.append(" ");

}

showStatus(buf.toString() );

}

input.setText("");

}

}

### 11004.JPG



**1105.JPG**



**Sekil 11.4-11.5 VectorTesti.java programi applet çiktisi**

Bundan sonraki program iki kümenin bilesim ve kesisim kümelerini hesaplamaktadir. Kümelerin kaç elemandan olustugu tanimlanmamistir. Islemler vector sinifini kullanarak yapilmaktadir. Ayrica bu yapida vector sinifini bir döngü içinde kullanmak için vector sinifi **Enumeration** sinifina yüklenmektedir. Enumeration sinifinin görevi, StringTokenizer sinifinin görevini andirir. Vector yapisinin içinden her vector elemanini ayri ayri çagirma ve indeksleme görevi görür. Enumeration interface’inin tanimi :

public interface Enumeration { boolean hasMoreElements(); Object nextElement();

}

seklindedir.

**Enumeration n1=list1.elements();** Enumeration n2=list2.elements(); while(**n1.hasMoreElements()**)

{

s1=(String)**n1.nextElement()**; bilesimVectoru.addElement(s1);

}

kod parçaciginda vector sinifinin elements metotu,Enumeration sinifi ve Enumeration sinifinin **hasMoreElements** ve **nextElement** metotlari kullanilarak nasil döngü olusturuldugu görülmektedir. Asagidaki örnekte setA sinifi verilmistir. Bu örnekte bir kümelerin bilesim ve kesisim kümelerinin vector metodunu kullanarak nasil olusturulabilecegi gösterilmektedir.

### Program 11.7 setA.java setTest.java siniflari, bir kümenin bilesim ve kesisim kümelerini hesaplar.

// Bu program vector ve Enumeration siniflarini

// Kullanmaktadir. Bilesim, kesisim kümelerini hesaplar

// Ayni zamanda StringTokaniser sinifini kullanir. import java.util.\*;

import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

class SetA

{

public Vector bilesim(Vector list1, Vector list2)

{

Vector bilesimVectoru = new Vector(); String s1,s2;

Enumeration n1=list1.elements(); Enumeration n2=list2.elements(); while(n1.hasMoreElements())

{

s1=(String)n1.nextElement(); bilesimVectoru.addElement(s1);

}

while(n2.hasMoreElements())

{

s2=(String)n2.nextElement(); if(!bilesimVectoru.contains(s2))

bilesimVectoru.addElement(s2);

}

return bilesimVectoru;

}

public Vector kesisim(Vector list1,Vector list2)

{

Vector kesisimVectoru = new Vector(); String s;

Enumeration n = list1.elements(); while(n.hasMoreElements())

{

s=(String)n.nextElement(); if(list2.contains(s))

kesisimVectoru.addElement(s);

}

return kesisimVectoru;

}

}

public class SetTest extends Applet implements ActionListener

{

SetA set = new SetA(); // SetA Label L1, P1, P2;

TextField T1,T2;

TextArea cikti;

// dinamik boyutlu degisken sinifi Vector'u kullanarak

// iki vector listesi yarat Vector list1 = new Vector(); Vector list2 = new Vector();

// Sonuclari yine vektor cinsinden degiskenler

// kullanarak aktar Vector ansUn; Vector ansInt; public void init() {

P1 = new Label("Birinci listeyi gir : "); T1 = new TextField(50);

P2 = new Label("Ikinci listeyi gir : "); T2 = new TextField(50);

// Sonuclarin yazildigi yazi alanini ac cikti = new TextArea(10,40); cikti.setEditable(false); cikti.setText("");

add(P1);

add(T1); T1.addActionListener(this); add(P2);

add(T2); T2.addActionListener(this); add(cikti);

}

public void girdiA(String string, Vector list)

{

// StringTokeniser sinifi String degiskenleri icin

// Enumeration sinifinin yaptigina paralel gorev

// gorur String'i siraya sokarak degerlerini sirayla verir StringTokenizer tokens = new StringTokenizer(string); while(tokens.hasMoreTokens()) {

String test = tokens.nextToken(); list.addElement(test);

}

}

public void ciktiA(Vector v)

{

Enumeration enum = v.elements(); while(enum.hasMoreElements())

{

String ans = (String)enum.nextElement(); cikti.append(ans + "\n");

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==T1)

{

String stringToTokenize = T1.getText(); girdiA(stringToTokenize,list1);

}

else if(e.getSource()==T2)

{

String stringToTokenize = T2.getText(); girdiA(stringToTokenize,list2);

// Iki listenin bilesim ve kesisim k• melerini hesapla. ansUn = set.bilesim(list1,list2);

ansInt = set.kesisim(list1,list2);

// sonuclari cikti alanina gonder cikti.append("\nIki listenin bilesim kumesi : \n"); ciktiA(ansUn);

cikti.append("\nIki listenin kesisim kumesi : \n"); ciktiA(ansInt);

}

} //actionPerformed metotunun sonu

}

Bu programda ayni zamanda StringTokaniser sinifi kullanilmistir. StringTokaniser String degiskenini bosluk kullanarak alt degiskenlere ayirmaya yarayan Enumeration türü bir siniftir.

### 11006.JPG



**Sekil 11.6 Vector sinifini kullanarak iki kumenin bilesim ve kesisim kümelerini hesaplayan SetTest.java**

**programi applet çiktisi**

**Ayni programin Swing versiyonuSetTestSWF\_2000.java asagida verilmistir :**

// Bu program vector ve Enumeration siniflarini

// Kullanmaktadir. Bilesim ve kesisim kümelerini hesaplar

// Ayni zamanda StringTokeniser sinifini kullanir.

### Program 11.8 setTestSWF\_2000.java, bir kümenin bilesim ve kesisim kümelerini hesaplar.

import java.util.\*; import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

public class SetTestSWF\_2000 extends JFrame implements ActionListener

{

SetA set = new SetA(); // SetA JLabel L1, P1, P2;

JTextField T1,T2;

JTextArea cikti;

// dinamik boyutlu degisken sinifi Vector'u kullanarak

// iki vector listesi yarat Vector list1 = new Vector(); Vector list2 = new Vector();

// Sonuclari yine vektor cinsinden de§iskenler

// kullanarak aktar Vector ansUn; Vector ansInt; Container c;

public SetTestSWF\_2000()

{

super("küme testi bilesim ve kesisim kümeleri"); c=getContentPane();

c.setLayout(new FlowLayout());

P1 = new JLabel("Birinci listeyi giriniz : "); T1 = new JTextField(30); T1.setBackground(c.getBackground());

P2 = new JLabel("Ikinci listeyi giriniz : "); T2 = new JTextField(30); T2.setBackground(c.getBackground());

// Sonuclarin yazildigi yazi alanini ac cikti = new JTextArea(); cikti.setEditable(false); cikti.setText("");

cikti.setBackground(c.getBackground()); c.add(P1);

c.add(T1); T1.addActionListener(this); c.add(P2);

c.add(T2); T2.addActionListener(this);

JScrollPane jp=new JScrollPane(cikti); jp.setPreferredSize(new Dimension(320,200)); c.add(jp);

}

public void girdiA(String string, Vector list)

{

// StringTokeniser sinifi String degiskenleri icin

// Enumeration sinifinin yaptigina paralel gorev

// gorur String'i s• raya sokarak degerlerini s• rayla verir StringTokenizer tokens = new StringTokenizer(string); while(tokens.hasMoreTokens()) {

String test = tokens.nextToken(); list.addElement(test);

}

}

public void ciktiA(Vector v)

{

Enumeration enum = v.elements(); while(enum.hasMoreElements())

{

String ans = (String)enum.nextElement(); cikti.append(ans + " ");

}

cikti.append("\n");

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==T1)

{

String stringToTokenize = T1.getText(); girdiA(stringToTokenize,list1);

}

else if(e.getSource()==T2)

{

String stringToTokenize = T2.getText(); girdiA(stringToTokenize,list2);

// Iki listenin bilesim ve kesisim kümelerini hesapla. ansUn = set.bilesim(list1,list2);

ansInt = set.kesisim(list1,list2);

// sonuclari cikti alanina gonder cikti.setText("");

cikti.append("Iki listenin bilesim kümesi : \n"); ciktiA(ansUn);

cikti.append("Iki listenin kesisim kümesi : \n"); ciktiA(ansInt);

}

} //actionPerformed metodunun sonu public static void main(String[] args)

{

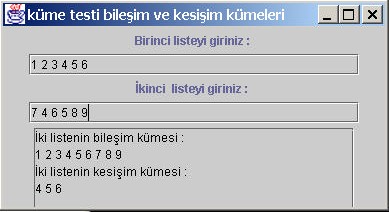
SetTestSWF\_2000 pencere = new SetTestSWF\_2000(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(400,350);

pencere.setVisible(true);

}

}

### 11007.JPG



**Sekil 11.7 Vector sinifini kullanarak iki kumenin bilesim ve kesisim kümelerini hesaplayan SetTestSWF\_2000 programi JFrame çiktisi**

### LIST(LISTE) SINIFI

Yukaridaki vektor sinifinda ilk defa dinamik hafiza kullanabilen (vektor boyutunu program çalisirken degistirebilen) Vector sinifini inceledik. Eger vektor boyutlari program çalisirken azaliyor veya çogaliyorsa, boyutlu degiskenler yerine dinamik hafiza kullanimi toplam bilgisayar hafizasi kullanimi açisinda çok daha verimlidir. Dinamik data yapilari sadece vektorden ibaret degildir. List(liste), Stack(dizin) ve Queue (sira) ve Tree(agaç) yapilari da oldukça sik kullanilar dinamik yapilari teskil eder. Bunlardan List yapisinin özellikleri sunlardir :

 Listenin ilk elemanindan önce veya son elemanindan sonra listeye dinamik eleman ilavesi yapilabilir.

 Listenin herhangibir ara noktasina yeni eleman ilave edilemez.

 Listenin ilk elemani veya son elemani listeden çekilebilir.

 Listenin herhangi bir ara noktasindan eleman çekilemez.

Burada List kavraminin daha iyi anlasilabilmesi için önce kendi List sinifimiz olusturulmus ve bir örnek problemde kullanlmistir. Bu program örneklerine bir göz atalim.

### Program 11.9 List.java programi. Bu program List yapisini tanimlayan ListNode ve List siniflarini içerir

class **ListNode**

{

Object data;

ListNode next;

ListNode(Object o) {this(o,null);} ListNode(Object o,ListNode nextNode)

{

data=o; next=nextNode;

}

Object getObject() {return data;} ListNode getnext() {return next;}

}

public class **List**

{

//Liste

### private ListNode firstNode; private ListNode lastNode; private String name;

public **List**(String s)

{

name=s; firstNode=lastNode=null;

}

public **List**(){this("Liste");}

public synchronized void **insertAtFront**(Object insertItem)

{

if( isEmpty() )

firstNode = lastNode = new ListNode(insertItem); else

firstNode = new ListNode(insertItem,firstNode);

}

public synchronized void **insertAtBack**(Object insertItem)

{

if(isEmpty())

firstNode=lastNode=new ListNode(insertItem); else

lastNode=lastNode.next=new ListNode(insertItem);

}

public synchronized Object **removeFromFront**() throws EmptyListException

{

Object removeItem=null; if(isEmpty())

throw new EmptyListException(name); removeItem=firstNode.data; if(firstNode.equals(lastNode))

firstNode=lastNode=null; else

firstNode=firstNode.next; return removeItem;

}

public synchronized Object **removeFromBack**() throws EmptyListException

{

Object removeItem=null; if(isEmpty())

throw new EmptyListException(name); removeItem=lastNode.data; if(firstNode.equals(lastNode))

firstNode=lastNode=null; else

{

ListNode current=firstNode; while(current.next != lastNode)

current=current.next; lastNode=current; current.next=null;

}

return removeItem;

}

public boolean **isEmpty**() {return firstNode==null; }

### //türkçe esdeger metodlar

public boolean **bosmu**() {return isEmpty(); }

public synchronized void **oneEkle**(Object o) {insertAtFront(o);}

public synchronized void **arkayaEkle**(Object o) {insertAtBack(o);} public synchronized Object **ondenCikar**() {return removeFromFront();} public synchronized Object **arkadanCikar**() {return removeFromBack();} public void **print**()

{

if(**bosmu**())

{

System.out.print(" Bos "+name); return;

}

System.out.print(" "+name+" : "); ListNode current=firstNode; while(current!=null)

{

System.out.print(current.data.toString()+" "); current=current.next;

}

System.out.println("\n");

}

}

### Program 11.10 : EmptyListException sinifinin EmptyListException dosyasindaki tanimi

public class **EmptyListException** extends RuntimeException

{

public EmptyListException(String name)

{

super(" "+name+" bos ");

}

}

### Program 11.11 : List yapisini kullanan örnek program ListTest.java

import List;

import EmptyListException; public class ListTest

{

public static void main(String args[])

{

List objList=new List(); Boolean b=new Boolean(true); Character c=new Character('$'); Integer i=new Integer(34567); String s=new String("hello"); objList.insertAtFront(b); objList.print(); objList.insertAtFront(c); objList.print(); objList.insertAtFront(i); objList.print(); objList.insertAtFront(s); objList.print();

Object removedObj; try{

removedObj=objList.removeFromFront(); System.out.print(removedObj.toString()+" cikarildi"); objList.print(); removedObj=objList.removeFromFront(); System.out.print(removedObj.toString()+" cikarildi"); objList.print();

removedObj=objList.removeFromBack(); System.out.print(removedObj.toString()+" cikarildi"); objList.print(); removedObj=objList.removeFromBack(); System.out.print(removedObj.toString()+" cikarildi"); objList.print();

}

catch(EmptyListException e) {System.out.println("\n"+e.toString());}

}

}

ListTest programinin sonucu :

### List yapisini kullanan örnek program ListTest.java nin çiktisi

Liste : true Liste : $ true

Liste : 34567 $ true

Liste : hello 34567 $ true

hello cikarildi Liste : 34567 $ true 34567 cikarildi Liste : $ true

true cikarildi Liste : $

$ cikarildi Bos Liste

Burada List sinifinin girdisinin Object sinifi oldugun dikkatinizi çekelim. Object sinifi yine dinamik object degisken yapilari olarak tanimlanan **Double, Byte, String, Integer, Boolean, Float, Short, Long** gibi degisken türleri için kullanilan siniflarin abstract sinifini teskil eder. Bu yüzden bu siniflarin hepsini temsil edebilir ve yerlerine kullanilabilir (genel bir siniftir.). Bu siniflar normal degisken türlerine dönüstürülebilirler. (daha önce de kullandigimiz intValue(), doubleValue() gibi metotlari kullanarak)

Object sinifinin temel tanimi (metodtanimlari olmadan) su sekildedir. package java.lang;

public class Object {

private static native void registerNatives(); static {

registerNatives();

}

public final native Class getClass(); public native int hashCode(); public boolean equals(Object obj

protected native Object clone() throws CloneNotSupportedException; public String toString()

public final native void notify(); public final native void notifyAll();

public final native void wait(long timeout) throws InterruptedException; public final void wait(long timeout, int nanos) throws InterruptedException ; public final void wait() throws InterruptedException ;

protected void finalize() throws Throwable;

}

burada geçen native sözcügünün anlamak istiyorsaniz, 14 üncü bölümü inceleyebilirsiniz. Bu terim metodun anadilde (native) yazildigini belirtir.

Java.util paketinde List sinifi tanimlanmistir. Yukarida kendi tanimladigimiz List ile yapabildiklerimizi ve daha fazlasini tanimli List paketiyle a gerçeklestirebiliriz.

package java.util;

public interface List extends Collection {

int size();

boolean isEmpty();

boolean contains(Object o); Iterator iterator();

Object[] toArray();

Object[] toArray(Object a[]); boolean add(Object o); boolean remove(Object o);

boolean containsAll(Collection c); boolean addAll(Collection c);

boolean addAll(int index, Collection c); boolean removeAll(Collection c); boolean retainAll(Collection c);

void clear();

boolean equals(Object o); int hashCode();

Object get(int index);

Object set(int index, Object element); void add(int index, Object element); Object remove(int index);

int indexOf(Object o);

int lastIndexOf(Object o); ListIterator listIterator(); ListIterator listIterator(int index);

List subList(int fromIndex, int toIndex);

}

List sinifiyla birlikte kullanilmak üzere ListIterator interface’i tanimlanmistir.

/\*

package java.util;

public interface ListIterator extends Iterator { boolean hasNext();

Object next();

boolean hasPrevious();

Object previous();

int nextIndex();

int previousIndex(); void remove(); void set(Object o); void add(Object o);

}

Simdi bu iki sinifin kullanimini bir örnekle gösterelim.

### Program 11.12 : List yapisini kullanan örnek program MovingPlanets.java

import java.util.List;

import java.util.ListIterator; import java.util.Iterator; import java.util.ArrayList;

import javax.swing.JOptionPane;

public class MovingPlanets {

public static void main (String args[]) {

String names[] = {"Mercür", "Venüs", "Dünya", "Mars", "Jupiter", "Satürn", "Uranüs", "Neptün", "Pluto"};

int namesLen = names.length; List planets = new ArrayList(); for (int i=0; i < namesLen; i++) {

planets.add (names[i]);

}

ListIterator lit = planets.listIterator(); String s;

lit.next();

lit.next();

s = (String)lit.next(); lit.remove();

lit.next();

lit.next();

lit.next();

lit.add(s);

lit.next(); lit.previous(); lit.previous();

s = (String)lit.previous(); lit.remove();

lit.next();

lit.next();

lit.add(s);

Iterator it = planets.iterator(); String ss="";

while (it.hasNext()) {ss+=it.next()+"\n";} JOptionPane.showMessageDialog(null,ss); System.exit(0);

}

}

### 11008.JPG



**Sekil 11.08 List sinifi, MovingPlanets çiktisi**

List sinifinin bir alt sinifi da LinkedList sinifidir. LinkedList sinifi List sinifinda olmayan ilave metodlar tanimlamistir.

public class LinkedList extends AbstractSequentialList

implements List, Cloneable, java.io.Serializable

{

private transient Entry header; private transient int size; public LinkedList()

public LinkedList(Collection c) public Object getFirst()

public Object getLast() public Object removeFirst() public Object removeLast()

public void addFirst(Object o) public void addLast(Object o) public boolean contains(Object o) public int size()

public boolean add(Object o) public boolean remove(Object o) public boolean addAll(Collection c)

public boolean addAll(int index, Collection c) public void clear()

public Object get(int index)

public Object set(int index, Object element public void add(int index, Object element) public Object remove(int index)

public int indexOf(Object o) public int lastIndexOf(Object o) \*/

public ListIterator listIterator(int index)

}

Bu sinifi kullanan küçük bir örnek problemi inceleyelim :

### Program 11.13 : LinkedList yapisini kullanan örnek program MovingPlanets.java

import java.util.List;

import java.util.ListIterator; import java.util.Iterator; import java.util.ArrayList;

import javax.swing.JOptionPane;

public class GezegenlerVeaylarListesi { static class Gezegen {

private String isim; private int aySayisi;

Gezegen (String s, int aylar) { isim = s;

aySayisi = aylar;

}

public String toString() {

return getClass().getName() + "[" + isim + "-" + aySayisi + "]";

}

public final String getName() { return isim;

}

public final int getaySayisi () { return aySayisi;

}

}

public static void main (String args[]) {

String names[] = {"Mercür", "Venüs", "Dünya", "Mars", "Jupiter", "Satürn", "Uranüs", "Neptün", "Pluto"};

int aylar[] = {0, 0, 1, 2, 16, 18, 17, 8, 1};

int namesLen = names.length;

List Gezegens = new ArrayList (namesLen); for (int i=0; i < namesLen; i++) {

Gezegens.add (new Gezegen (names[i], aylar[i]));

}

String ss="";

for (int i=Gezegens.size()-1; i >= 0; --i)

{

Gezegen p = (Gezegen)(Gezegens.get (i));

ss+=p.getName() + " : " + p.getaySayisi()+"\n";

}

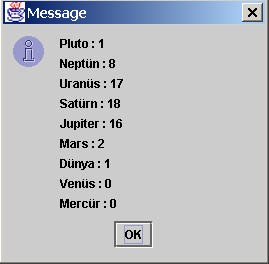
JOptionPane.showMessageDialog(null,ss); System.exit(0);

}

}

Program çiktisi

### 11009.JPG



**Sekil 11.09 Linkedlist sinifi, GezegenlerVeaylarListesi çiktisi**

LinkList Bölüm 11.4’de inceledigimiz Stack(dizi) yapisi ve bölüm 11.5 de inceledigimiz queue(sira) yapisi yerine kullanilabilir.

### DIZI (STACK) SINIFI

Diziler veya ingilizce adiyla Stack aslinda üstte tanimladigimiz List sinifinin bir alt sinifidir. dizin listeye göre daha sinirli olan su kosullara sahiptir :

 Dizide son elemandan sonra listeye dinamik eleman ilavesi yapilabilir.

 Dizinin herhangibir ara noktasina yeni eleman ilave edilemez.

 Dizinin sadece son elemani listeden çekilebilir.

 Listenin herhangi bir ara noktasindan eleman çekilemez.

Yukardaki kendi tanimladigimiz List (Program 11.5,11.6) kullanarak olusturdugumuz dizi programi ve diziyi kullnan bir test programi asagida verilmistir.

### Program 11.14 : dizi sinifinin dizi.java dosyasindaki tanimlari

public class dizi extends List

{

public dizi() {super("dizi");}

public synchronized void **push**(Object o)

{insertAtFront(o);}

public synchronized void **ekle**(Object o)

{insertAtFront(o);}

public synchronized Object **pop**() throws EmptyListException

{return removeFromFront();}

public synchronized Object **cek**() throws EmptyListException

{return removeFromFront();}

public boolean **isEmpty**() {return super.isEmpty();} public boolean **bosmu(**) {return super.isEmpty();}

public void **print**() {super.print();}

}

### Program 11.15 : dizi sinifinin diziTesti.java dosyasindaki örnek kullanimi

import bolum11.dizi;

import bolum11.EmptyListException; public class diziTesti

{

public static void main(String args[])

{

dizi d=new dizi();

Boolean b=new Boolean(true); Character c=new Character('$'); Integer i=new Integer(34567); String s=new String("hello"); d.ekle(b);

d.print();

d.ekle(c);

d.print();

d.ekle(i);

d.print();

d.ekle(s);

d.print(); Object cekilen; try

{

while(true)

{

cekilen=d.cek();

System.out.println(cekilen.toString()+" diziden cekildi"); d.print();

}

}

catch(EmptyListException e)

{System.err.println("\n"+e.toString());}

}

}

### dizi yapisini kullanan örnek program diziTesti.java’nin çiktisi

dizi : true dizi : $ true

dizi : 34567 $ true

dizi : hello 34567 $ true hello diziden cekildi dizi : 34567 $ true 34567 diziden cekildi dizi : $ true

$ diziden cekildi dizi : true

true diziden cekildi Bos dizi

yukarida tanimlanan **LinkedList** sinifinin addFirst()/removeFirst() metodlarini kullanarak da stack olusturabiliriz.

**java.util** paketinde **vector** sifinin yani sira **Stack** sinifi da tanimlanmistir. Java Stack sinifinda

push, pop ve peek metotlari vardir. push ve pop dizin programini aynisidir. peek ise listenin en sonundaki

elemanin degerini bu elemani listeden uzaklastirmadan alir. (pop ile push beraber kullanilarak da ayni isi yapabilirler)

Bu sinifin tanimi :

public class Stack extends Vector

{

public Stack();

public Boolean empty();

public synchronized Object peek(); public synchronized Object pop();

public synchronized Object push(Object obj); public synchronized int search(Object obj);

}

Stack sinifinin metotlarinin ne oldugunu daha detayli inceleyecek olursak:

### push

**public Object push(Object item)**

stack'in üstüne yeni eleman ilave eder

### pop

**public synchronized Object pop()**

Stakin en tepesindeki elemani çeker (Stack dizisinden kaldirir ve degerini return eder) Stack'in bos olmasi halinde **EmptyStackException** gönderir.

### peek

**public synchronized Object peek()**

Stack'in en tepesindeki elemanin degerini okur(Stack dizisinden kaldirmaz ve sadece degerini return eder) Stack'in bos olmasi halinde **EmptyStackException** gönderir.

### empty

**public boolean empty()**

Stack'in bos olup olmadigini kontrol eder, bos ise true, dolu ise false degeri gönderir. Tests if this stack is empty.

### search

**public synchronized int search(Object o)**

Stack'in içindeki istenen Object'in yerini verir. Eger object Stack'de yok ise -1 degeri verir

Burada hemen Stack sinifinin Vector sinifinin alt sinifi oldugunu ve Vector sinifinda mevcut olan tüm metotlar Stack metotunda da geçerli oldugunu da hatirlatalim.

simdi bir örnek problemde bu sinifin nasil kullanildigina bakalim :

### Program 11.16 : java.util kütüphanesinde yer alan Stack yapisini kullanan örnek program StackTesti.java

import java.util.\*;

public class StackTesti

{

public static void main(String args[])

{

StackTesti st=new StackTesti(); Stack d=new Stack();

Boolean b=new Boolean(true); Character c=new Character('$'); Integer i=new Integer(34567); String s=new String("hello"); d.push(b);

st.print(d);

d.push(c);

st.print(d);

d.push(i);

st.print(d);

d.push(s);

st.print(d); Object bakilan; Object cekilen; try

{

while(true)

{

bakilan=d.peek();

System.out.println(bakilan.toString()+" de§erine bakildi"); st.print(d);

cekilen=d.pop();

System.out.println(cekilen.toString()+" de§eri listeden cekildi "); st.print(d);

System.out.println("---------------------");

}

}

catch(EmptyStackException e)

{System.err.println("\n"+e.toString());}

}

public void print(Stack di)

{

Enumeration enum=di.elements(); StringBuffer buf=new StringBuffer(); while(enum.hasMoreElements()) buf.append(enum.nextElement()).append(" "); System.out.println(buf.toString());

}

}

### java.util kütüphanesinde yer alan Stack yapisini kullanan örnek program StackTesti.java nin çiktisi

true true $

true $ 34567

true $ 34567 hello hello degerine bakildi true $ 34567 hello

hello degeri listeden cekildi true $ 34567

---------------------

34567 degerine bakildi

true $ 34567

34567 degeri listeden cekildi true $

---------------------

$ degerine bakildi true $

$ degeri listeden cekildi true

---------------------

true degerine bakildi true

true degeri listeden cekildi

---------------------

### SIRA (QUEUE) SINIFI

Sira veya ingilizce adiyla queue aslinda üstte tanimladigimiz List sinifinin bir alt sinifidir. sira listeye göre daha sinirli olan su kosullara sahiptir :

 Sirada son elemandan sonra listeye dinamik eleman ilavesi yapilabilir.

 Siranin herhangibir ara noktasina yeni eleman ilave edilemez.

 Siranin sadece ilk elemani listeden çekilebilir.

 Siranin herhangi bir ara noktasindan eleman çekilemez.

Sira kavraminin oldukça yaygin olarak kullanildigi ülkemizde oldukça yararli olabilecek bir sinifi degilmi? Yukardaki List tanimini kullanarak olusturdugumuz sira programi ve sirayii kullanan bir test programi asagida verilmistir.

### Program 11.17 : sira sinifinin sira.java dosyasindaki tanimlari

public class sira extends List

{

public sira() {super("sira");}

public synchronized void sirayagir(Object o)

{arkayaEkle(o);}

public synchronized Object siranGeldi() throws EmptyListException

{return ondenCikar();}

public boolean isEmpty() {return super.isEmpty();} public boolean bosmu() {return super.isEmpty();} public void print() {super.print();}

}

### Program 11.18 : sira sinifinin siraTesti.java dosyasindaki test programi

import sira;

import EmptyListException; public class siraTesti

{

public static void main(String args[])

{

sira s=new sira();

Boolean b=new Boolean(true); Character c=new Character('$'); Integer i=new Integer(34567); String st=new String(" merhaba "); s.sirayagir(b);

s.print(); s.sirayagir(c); s.print(); s.sirayagir(i); s.print(); s.sirayagir(st); s.print(); Object cekilen; try

{

while(true)

{

cekilen=s.siranGeldi();

System.out.println(cekilen.toString()+" sirasi geldi ”nden cikti"); s.print();

}

}

catch(EmptyListException e)

{System.err.println("\n"+e.toString());}

}

}

### sira sinifini kullanan siraTesti.java dosyasinin çiktisi

sira : true sira : true $

sira : true $ 34567

sira : true $ 34567 merhaba true sirasi geldi onden cikti sira : $ 34567 merhaba

$ sirasi geldi onden cikti sira : 34567 merhaba

34567 sirasi geldi onden cikti sira : merhaba

merhaba sirasi geldi onden cikti Bos sira

Queue sinifi Java'da direk olarak tanimlanmamistir, ancak üstteki sira örneginin bir benzeri Vector sinifindan bizim tarafimizdan rahatlikla yazilabilir, veya yine LinkedList sinifinin addFirst()/removeLast() metodlarini kullanarak olusturulabilir.

### TREE(AGAÇ) SINIFI

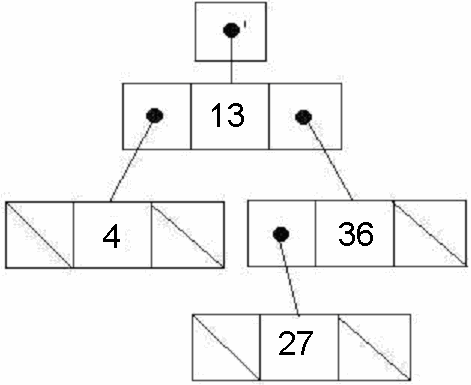
Elimizdeki bir gurup boyutlu degiskenin belli bir yapida guruplandirilmasi veya aranmasi gerektiginde bu standart boyut kavraminda oldukça kompleks bir yapi arzeder. Bilgi guruplandirma islemini çok daha basit bir sekilde yapabilmek için Tree yapilari öngörülmektedir. Tree temel olarak Vector gibi bir yapidir. fakat burada her Tree noktasi (Nod'u) iki degisik adrese isaret eder bu adreslerden birisi o nodedaki objenin kokdekinden daha küçük oldugu bir degeri, digeri ise daha büyük oldugu bir degeri tasir. hiç bir deger tasimiyorsa da null degeriyle o Tree-agaç dali sona erer. Bu yapiyi kullanarak sayilari büyükten küçüge veyaküçükten büyüge veya daha degisik mantiklarla direk olarak siralamak mümkündür.

### 11010.JPG



**Sekil 11.10 : Binary search Tree (ikili arama agaci) yapisinin grafiksel gösterimi**

**11011.JPG**



**Sekil 11.11 Binary search Tree (ikili arama agaci) yapisinin bir rakam setine uygulanmasinin grafiksel gösterimi**

Sekil 11.10 de ikili arama agaci yapisinin seti görülmektedir. Sekil 11.11 de ise sayisal bir örnekle yapi gösterilmistir. Her bir dal kökü kendinden sonra gelen rakami küçükse sol dala büyükse sag dala göndermektedir. ve her yeni veri yeni bir dallanma olusturmaktadir. Simdi Tree yapisini olusturacak bir örnek

sinifi ve onun kullanildigi bir örnek problem sinifini olusturarak kavrami biraz daha açmaya çalisalim :

### Program 11.19 : Binary search Tree (ikili arama agaci) yapisinin programlandigi TreeNode ve Tree siniflarinin ter aldigi Tree.java programi

class TreeNode

{

TreeNode sol; int data; TreeNode sag;

public TreeNode(int d)

{

data=d; sol=sag=null;

}

public synchronized void gir(int d)

{

// not ayni deger iki kere girilmeye calisilirsa

// ikinci giris dikkate alinmiyacaktir if(d<data)

{ if(sol==null) {sol=new TreeNode(d);} else {sol.gir(d);}

}

else if(ddata)

{ if(sag==null) {sag=new TreeNode(d);} else {sag.gir(d);}

}

}

}

public class Tree

{

private TreeNode kok; public Tree() {kok=null;}

public synchronized void gir(int d)

{

if(kok==null) kok=new TreeNode(d); else kok.gir(d);

}

public void node\_soldal\_sagdal\_sirala(){node\_soldal\_sagdal\_siralayici(kok);} public void soldal\_node\_sagdal\_sirala(){soldal\_node\_sagdal\_siralayici(kok);} public void soldal\_sagdal\_node\_sirala(){soldal\_sagdal\_node\_siralayici(kok);} public void node\_sagdal\_soldal\_sirala(){node\_sagdal\_soldal\_siralayici(kok);} public void sagdal\_node\_soldal\_sirala(){sagdal\_node\_soldal\_siralayici(kok);} public void sagdal\_soldal\_node\_sirala(){sagdal\_soldal\_node\_siralayici(kok);} public void node\_soldal\_sagdal\_siralayici(TreeNode node)

{

if(node==null) return; System.out.print(node.data+" "); node\_soldal\_sagdal\_siralayici(node.sol); node\_soldal\_sagdal\_siralayici(node.sag);

}

public void node\_sagdal\_soldal\_siralayici(TreeNode node)

{

if(node==null) return; System.out.print(node.data+" "); node\_sagdal\_soldal\_siralayici(node.sag); node\_sagdal\_soldal\_siralayici(node.sol);

}

public void soldal\_node\_sagdal\_siralayici(TreeNode node)

{

if(node==null) return; soldal\_node\_sagdal\_siralayici(node.sol); System.out.print(node.data+" "); soldal\_node\_sagdal\_siralayici(node.sag);

}

public void sagdal\_node\_soldal\_siralayici(TreeNode node)

{

if(node==null) return; sagdal\_node\_soldal\_siralayici(node.sag); System.out.print(node.data+" "); sagdal\_node\_soldal\_siralayici(node.sol);

}

public void soldal\_sagdal\_node\_siralayici(TreeNode node)

{

if(node==null) return; soldal\_node\_sagdal\_siralayici(node.sol); soldal\_node\_sagdal\_siralayici(node.sag); System.out.print(node.data+" ");

}

public void sagdal\_soldal\_node\_siralayici(TreeNode node)

{

if(node==null) return; sagdal\_node\_soldal\_siralayici(node.sag); sagdal\_node\_soldal\_siralayici(node.sol); System.out.print(node.data+" ");

}

}

### Program 11.20 : Binary search Tree (ikili arama agaci) yapisinin programlandigi TreeNode ve Tree siniflarinin kullanilmasini örnekleyen TreeTest.java programi

import Tree;

public class TreeTest

{

public static void main(String args[])

{

Tree agac=new Tree(); int sayi;

System.out.println("orijinal olarak Tree'ye girilen sayi dizini"); for(int i=0;i<=6;i++)

{

sayi=(int)(Math.random()\*45)+1; System.out.print(sayi+" "); agac.gir(sayi);

}

System.out.println("\nSol dal - kok - sag dal siralama : "); **agac.soldal\_node\_sagdal\_sirala();** System.out.println("\nkok -Sol dal - sag dal siralama : "); agac.node\_soldal\_sagdal\_sirala(); System.out.println("\nSol dal - sag dal - kok siralama : "); agac.soldal\_sagdal\_node\_sirala(); System.out.println("\nSag dal - kok - sol dal siralama : "); **agac.sagdal\_node\_soldal\_sirala();** System.out.println("\nkok -Sag dal - sol dal siralama : "); agac.node\_sagdal\_soldal\_sirala(); System.out.println("\nSag dal - sol dal - kok siralama : "); agac.sagdal\_soldal\_node\_sirala();

}

}

### Program 11.17 : Binary search Tree (ikili arama agaci) yapisinin programlandigi TreeNode ve Tree siniflarinin kullanilmasini örnekleyen TreeTest.java programinin sonuçlari. renkli olarak isaretlenen satirlarin sayi dizisini küçükten büyüge ve büyükten küçüge siraladigina dikkat ediniz.

orijinal olarak Tree'ye girilen sayi dizini 13 11 25 34 21 26 14

Sol dal - kok - sag dal siralama :

### 11 13 14 21 25 26 34

kok -Sol dal - sag dal siralama : 13 11 25 21 14 34 26

Sol dal - sag dal - kok siralama : 11 14 21 25 26 34 13

Sag dal - kok - sol dal siralama :

### 34 26 25 21 14 13 11

kok -Sag dal - sol dal siralama : 13 25 34 26 21 14 11

Sag dal - sol dal - kok siralama : 34 26 25 21 14 11 13

### DICTIONARY ve HASHTABLE SINIFLARI

Su ana kadar bu konuda gördügümüz elemanlarda belli bir baglanti kurali kullanarak elemanlari birbirine baglamistik. Bu hafiza açisindan yogun bir islemdir. Her seferinde hafiza pozisyonunun arastirilmasini gerektirir. Java anahtar referans saglayan daha degisik bir yapiyi da içerir. Burada her elemanin kendimiz tarafindan verilen bir referans degeri mevcuttur, ve elemani bulmak için bu referansi kullanabiliriz. Bu gurubun abstract sinifi Dictionary sinifidir. Dictionary sinifinin altinda Hashtable sinifi yer alir. Dictionary sinifinin tanimi :

package java.util;

public abstract class Dictionary

{

public Dictionary() abstract public int size();

abstract public boolean isEmpty(); abstract public Enumeration keys(); abstract public Enumeration elements(); abstract public Object get(Object key);

abstract public Object put(Object key, Object value); abstract public Object remove(Object key);

}

HashTable sinifinin tanimi :

public class Hashtable extends Dictionary implements Map, Cloneable,

java.io.Serializable {

private transient Entry table[]; private transient int count; private int threshold;

private float loadFactor;

private transient int modCount = 0;

private static final long serialVersionUID = 1421746759512286392L; public Hashtable(int initialCapacity, float loadFactor)

public Hashtable(int initialCapacity) public Hashtable(Map t)

public int size()

public boolean isEmpty()

public synchronized Enumeration keys() public synchronized Enumeration elements()

public synchronized boolean contains(Object value) public boolean containsValue(Object value)

public synchronized boolean containsKey(Object key) public synchronized boolean equals(Object key) public synchronized Object get(Object key)

protected void rehash()

public synchronized Object put(Object key, Object value) public synchronized Object remove(Object key)

public synchronized void putAll(Map t) public synchronized void clear()

public synchronized Object clone() public synchronized String toString() public Set entySet()

public Set keySet()

public Collection Values()

public synchronized int hashCode()

}

Hashtables sinifi temel olarak Dictionary ve Map siniflarindan türetilmistir. Map sinifi tanimi

package java.util; public interface Map {

int size();

boolean isEmpty();

boolean containsKey(Object key); boolean containsValue(Object value); Object get(Object key);

Object put(Object key, Object value); Object remove(Object key);

void putAll(Map t); void clear();

public Set keySet(); public Collection values(); public Set entrySet(); boolean equals(Object o); int hashCode();

public interface Entry

{

Object getKey();

Object getValue();

Object setValue(Object value); boolean equals(Object o);

int hashCode();

}

}

seklindedir.

En çok kullanacagimiz Hashtable metodlarinin anlamlarina bir göz atalim : Bir referans çifti girmek için : put(Object anahtar, Object deger) Anahtari vererek referans degerini çagirmak için : get(Object anahtar)

Bir elemani silmek için : remove((Object anahtar) Boyutu kontrol için : size()

Bos olup olmadiginin kontralu için empty()

Tüm anahtarlarin listesini almak için : keys() veya keySet() Tüm degerlerin listesini almak için : elements() veya entrySet()

### Program 11.21 : Hashtable yapisinin programlandigi GezegenCaplari.java programi

import java.util.Enumeration; import java.util.Hashtable; import javax.swing.JOptionPane;

public class GezegenCaplari {

public static void main (String args[]) {

String names[] = {"Merkür", "Venüs", "Dünya", "Mars", "Jüpiter", "Satürn", "Uranüs", "Neptün", "Pluton"};

float diameters[] = {4800f, 12103.6f, 12756.3f, 6794f, 142984f, 120536f, 51118f, 49532f, 2274f};

Hashtable hash = new Hashtable();

for (int i=0, n=names.length; i < n; i++) { hash.put (names[i], new Float (diameters[i]));

}

Enumeration enum = hash.keys(); Object obj;

String ss="";

while (enum.hasMoreElements()) { obj = enum.nextElement();

ss+=obj + ": " + hash.get(obj)+"\n";

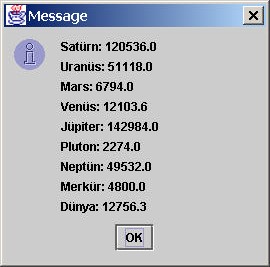
}

JOptionPane.showMessageDialog(null,ss); System.exit(0);

}

}

### 11012.JPG



**Sekil 11.12 HashTable sinifini kullanan GezegenCaplari programi**

Programdan da görüldügü gibi çikti hiçbir mantiksal siralama düzeni içermemektedir, fakat burada önemli olan gezegenlerin ve çaplarinin birbirleriyle olan iliskilerinin listelenmesidir. Hashtable bu tür islevleri en iyi bir sekilde yerine getirir.

### ARRAYS SINIFI VE SIRALAMA

Arrays sinifi boyutlu degiskenleri siralamaya yariyan bir java sinfidir. Metodlari: public static List asList(object array[]);

b degerinin a degerinin hangi elemaninda oldugunu bulan :

public static int binarySearch(<degiskentürü> a[],<degiskentürü>b);

<degiskentürü> char,double,float,int,Object,long,short degerleri alabilir, elbette Object üzerinden Objectin alt siniflari olan String, Double, Integer gibi siniflari da kabul eder.

Iki boyutlu degiskenin birbirine esit olup olmadigini denetleyen : public static boolean equals(<degiskentürü> a[],<degiskentürü>b[]);

boyutlu degiskenin tüm degerlerini veya tanimlanan bir bölümünü bir bölümünü ayni tür bir degisken atayan public static void fill(<degiskentürü> a[],<degiskentürü>b);

public static void fill(<degiskentürü> a[],int baslangiçindisi,int bitisindisi,<degiskentürü>b); boyutlu degiskeni siraya sokan :

public static void sort(<degiskentürü> a[]);

public static void sort(Object a[],Comparator comp);

listeden de görüldügü gibi tüm metodlar static metodlardir.

### Program 11.22 : Arrays yapisinin kullanildigi Gezegensirasi.java programi

import java.util.\*;

import javax.swing.JOptionPane;

public class Gezegensirasi {

static class InsensitiveComp implements Comparator { public int compare (Object a1, Object a2) {

String s1 = a1.toString().toLowerCase(); String s2 = a2.toString().toLowerCase(); return s1.compareTo (s2);

}

}

public static void main (String args[]) {

String names[] = {"Mercür", "Venüs", "Dünya", "Mars", "Jüpiter", "Satürn", "Uranüs", "Neptün", "Pluto",

"mercür", "venüs", "dünya",

"mars", "jüpiter", "satürn", "uranüs", "neptün", "pluto"

};

Arrays.sort(names);

int namesLen = names.length; String ss="";

for (int i=0; i<namesLen; i++) { ss+=names[i] + " ";

}

ss+="\n";

Arrays.sort(names, new InsensitiveComp()); for (int i=0; i<namesLen; i++) { ss+=names[i] + " ";

}

ss+="\n"; JOptionPane.showMessageDialog(null,ss); System.exit(0);

}

}

### 11013.JPG



**Sekil 11.13 Arrays sinifini kullanan Gezegensirasi programi**

programda liste ikinci kere Comparator sinifi InsensitiveComp() kullanarak büyük ve küçük harfi göz önüne almadan siraya sokulmustur.

### ALISTIRMALAR

1. **H10AL1.java :** Vector sinifi örnegi, H9O1 de boyutlu degiskenle çözülen dosyadan okunan rakamlarin ortalamasinin bulunmasi.

### Program 11.23 : H10Al1.java programi, vector ve dosya girdi örnegi

import java.io.\*; import Text; import java.util.\*;

public class H10AL1

{

public static void main(String[] args) throws IOException

{

DataInputStream input;

Vector i= new Vector(1);

String s1;

Text cin= new Text();

System.out.print("Lutfen dosya ismini giriniz: "); s1 = cin.readString();

File myfile = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(myfile)); int toplam=0;

boolean EOF=false; int j=0;

while (!EOF)

{

try

{

j++;

i.addElement(Text.readString(b));

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

Enumeration enum=i.elements(); StringBuffer buf = new StringBuffer();

while(enum.hasMoreElements())

{

buf.append(enum.nextElement()); Double d=new Double(buf.toString()); toplam+=d.doubleValue();

}

System.out.println("Dosyanin icindeki sayilarin adedi: "+j); System.out.println("Ortalama: "+(double)toplam/i.size());

} //main

} //class

Lutfen dosya ismini giriniz: c.txt

1.0 toplam: 1

2.0 toplam: 3

3.0 toplam: 6

4.0 toplam: 10

5.0 toplam: 15

6.0 toplam: 21

7.0 toplam: 28

8.0 toplam: 36

9.0 toplam: 45

10.0 toplam: 55

11.0 toplam: 66

12.0 toplam: 78

13.0 toplam: 91

14.0 toplam: 105

15.0 toplam: 120

16.0 toplam: 136

17.0 toplam: 153

18.0 toplam: 171

19.0 toplam: 190

20.0 toplam: 210

dosyanin ve yuklendigi vektorun icindeki sayilarin adedi: 20 Ortalama: 10.5

1. **H10AL2.java : List** sinifi örnegi, H9O1 de boyutlu degiskenle çözülen dosyadan okunan rakamlarin ortalamasinin bulunmasi.

### Program 11.24 : List sinifi ve dosyadan okuma örnegi H10AL1.java

import java.io.\*; import Text; import java.util.\*;

import bolum11.List;

import bolum11.EmptyListException; public class H10AL2

{

public static void main(String[] args) throws IOException

{

DataInputStream input;

List l= new List("H10AL2"); String s1;

Text cin= new Text();

System.out.print("Lutfen dosya ismini giriniz: "); s1 = cin.readString();

File myfile = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(myfile)); int toplam=0;

boolean EOF=false; int j=0;

while (!EOF)

{

try

{

l.arkayaEkle(Text.readString(b)); j++;

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

while(!l.bosmu())

{

Double d=new Double((String)l.ondenCikar()); toplam+=d.doubleValue(); System.out.println(""+d+" toplam: "+toplam);

}

System.out.println("dosyan• n ve y• klendigi listenin icindeki sayilarin adedi: "+j); System.out.println("Ortalama: "+(double)toplam/j);

} //main

} //class

Lutfen dosya ismini giriniz: c.txt

1.0 toplam: 1

2.0 toplam: 3

3.0 toplam: 6

4.0 toplam: 10

5.0 toplam: 15

6.0 toplam: 21

7.0 toplam: 28

8.0 toplam: 36

9.0 toplam: 45

10.0 toplam: 55

11.0 toplam: 66

12.0 toplam: 78

13.0 toplam: 91

14.0 toplam: 105

15.0 toplam: 120

16.0 toplam: 136

17.0 toplam: 153

18.0 toplam: 171

19.0 toplam: 190

20.0 toplam: 210

dosyanin ve yuklendigi listenin icindeki sayilarin adedi: 20 Ortalama: 10.5

1. **H10AL3.java : dizi** sinifi örnegi, H9O1 de boyutlu degiskenle çözülen dosyadan okunan rakamlarin ortalamasinin bulunmasi.

### Program 11.25 : H10AL3.java,

import java.io.\*; import Text; import java.util.\*;

import bolum11.dizi;

import bolum11.EmptyListException;

//dizi ornegi

public class H10AL3

{

public static void main(String[] args) throws IOException

{

DataInputStream input; dizi l= new dizi(); String s1;

Text cin= new Text();

System.out.print("Lutfen dosya ismini giriniz: "); s1 = cin.readString();

File myfile = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(myfile)); int toplam=0;

boolean EOF=false; int j=0;

while (!EOF)

{

try

{

l.ekle(Text.readString(b)); j++;

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

while(!l.bosmu())

{

Double d=new Double((String)l.cek()); toplam+=d.doubleValue(); System.out.println(""+d+" toplam: "+toplam);

}

System.out.println("dosyanin ve yuklendigi listenin icindeki sayilarin adedi: "+j); System.out.println("Ortalama: "+(double)toplam/j);

} //main

} //class

Lutfen dosya ismini giriniz: c.txt

20.0 toplam: 20

19.0 toplam: 39

18.0 toplam: 57

17.0 toplam: 74

16.0 toplam: 90

15.0 toplam: 105

14.0 toplam: 119

13.0 toplam: 132

12.0 toplam: 144

11.0 toplam: 155

10.0 toplam: 165

9.0 toplam: 174

8.0 toplam: 182

7.0 toplam: 189

6.0 toplam: 195

5.0 toplam: 200

4.0 toplam: 204

3.0 toplam: 207

2.0 toplam: 209

1.0 toplam: 210

dosyanin ve yuklendigi listenin icindeki sayilarin adedi: 20 Ortalama: 10.5

1. **H10AL4.java : dizi** sinifi örnegi, H9O1 de boyutlu degiskenle çözülen dosyadan okunan rakamlarin ortalamasinin bulunmasi.

### Problem 11.26 : H10AL4.java dizi sinifi örnegi

import java.io.\*; import Text; import java.util.\*;

import bolum11.sira;

import bolum11.EmptyListException;

//sira ornegi

public class H10AL4

{

public static void main(String[] args) throws IOException

{

DataInputStream input; sira l= new sira(); String s1;

Text cin= new Text();

System.out.print("Lutfen dosya ismini giriniz: "); s1 = cin.readString();

File myfile = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(myfile));

int toplam=0; boolean EOF=false; int j=0;

while (!EOF)

{

try

{

l.sirayagir(Text.readString(b)); j++;

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

while(!l.bosmu())

{

Double d=new Double((String)l.siranGeldi()); toplam+=d.doubleValue(); System.out.println(""+d+" toplam: "+toplam);

}

System.out.println("dosyanin ve yuklendigi listenin icindeki sayilarin adedi: "+j); System.out.println("Ortalama: "+(double)toplam/j);

} //main

} //class

Lutfen dosya ismini giriniz: c.txt

1.0 toplam: 1

2.0 toplam: 3

3.0 toplam: 6

4.0 toplam: 10

5.0 toplam: 15

6.0 toplam: 21

7.0 toplam: 28

8.0 toplam: 36

9.0 toplam: 45

10.0 toplam: 55

11.0 toplam: 66

12.0 toplam: 78

13.0 toplam: 91

14.0 toplam: 105

15.0 toplam: 120

16.0 toplam: 136

17.0 toplam: 153

18.0 toplam: 171

19.0 toplam: 190

20.0 toplam: 210

dosyanin ve yuklendigi listenin icindeki sayilarin adedi: 20 Ortalama: 10.5

1. **H10AL5.java : Stack** sinifi örnegi, H9O1 de boyutlu degiskenle çözülen dosyadan okunan rakamlarin ortalamasinin bulunmasi.

### Problem 11.27 H10Al5.java Stack sinifi örnegi

import java.io.\*; import Text; import java.util.\*;

//Stack - dizi ornegi

public class H10AL5

{

public static void main(String[] args) throws IOException

{

DataInputStream input;

Stack l= new Stack();

String s1;

Text cin= new Text();

System.out.print("Lutfen dosya ismini giriniz: "); s1 = cin.readString();

File myfile = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(myfile)); int toplam=0;

boolean EOF=false; int j=0;

while (!EOF)

{

try

{

l.push(Text.readString(b)); j++;

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

while(!l.empty())

{

Double d=new Double((String)l.pop()); toplam+=d.doubleValue(); System.out.println(""+d+" toplam: "+toplam);

}

System.out.println("dosyanin ve yuklendigi listenin icindeki sayilarin adedi: "+j); System.out.println("Ortalama: "+(double)toplam/j);

} //main

} //class

Lutfen dosya ismini giriniz: c.txt

20.0 toplam: 20

19.0 toplam: 39

18.0 toplam: 57

17.0 toplam: 74

16.0 toplam: 90

15.0 toplam: 105

14.0 toplam: 119

13.0 toplam: 132

12.0 toplam: 144

11.0 toplam: 155

10.0 toplam: 165

9.0 toplam: 174

8.0 toplam: 182

7.0 toplam: 189

6.0 toplam: 195

5.0 toplam: 200

4.0 toplam: 204

3.0 toplam: 207

2.0 toplam: 209

1.0 toplam: 210

dosyanin ve yuklendigi listenin icindeki sayilarin adedi: 20 Ortalama: 10.5

Alistirmalarda kullanilan c.txt dosyasi :

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

**6.** SetTest.java programini inceleyiniz. Kesisim kumesiyle bilesim kumesinin farkini olusturan kümeyi hesaplayan bir metot yaziniz ve test programina da ekleyerek çalistiriniz.

17. **H1OD1.java**

veri dosyasi e.txt'e e dizisini giriniz. d.txt'e d dizisini giriniz. program setText'ti inceleyiniz. iki dizinin bilesim kümesini ed.txt dosyasina yazdiriniz.

### Program 11.28 H10OD1.java iki dizinin bilesim kümesini vektor sinifini kullanarak hesaplayan program

import java.io.\*; import Text; import java.util.\*;

public class H10OD1

{

public void girdiA(String string, Vector list)

{

// StringTokeniser sinifi String degiskenleri icin

// Enumeration sinifinin yaptigina paralel gorev

// gorur String'i s• raya sokarak degerlerini s• rayla verir StringTokenizer tokens = new StringTokenizer(string); while(tokens.hasMoreTokens()) {

String test = tokens.nextToken(); list.addElement(test);

}

}

public static String ciktiA(Vector v)

{

StringBuffer cikti=new StringBuffer(); Enumeration enum = v.elements(); while(enum.hasMoreElements())

{

String ans = (String)enum.nextElement(); cikti.append(ans + " ");

}

return cikti.toString();

}

public static void main(String[] args) throws IOException

{

SetA set=new SetA(); DataInputStream input; Vector v1= new Vector(1); Vector v2= new Vector(1); Vector v=new Vector(1); String s1,s2;

Text cin= new Text();

System.out.print("Lutfen ilk dosya ismini giriniz: "); s1 = cin.readString();

File myfile1 = new File(s1);

BufferedReader b1= new BufferedReader(new FileReader(myfile1)); System.out.print("Lutfen ikinci dosya ismini giriniz: ");

s2 = cin.readString();

File myfile2 = new File(s2);

BufferedReader b2= new BufferedReader(new FileReader(myfile2)); int toplam=0;

boolean EOF=false; while (!EOF)

{

try

{

v1.addElement(Text.readString(b1));

}

catch (EOFException e)

{

b1.close();

EOF=true;

}

} //while EOF=false; while (!EOF)

{

try

{

v2.addElement(Text.readString(b2));

}

catch (EOFException e)

{

b2.close();

EOF=true;

}

} //while v=set.bilesim(v1,v2); System.out.println(ciktiA(v));

PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("de.txt"))); cfout.println(ciktiA(v));

cfout.close(); toplam=0;

Enumeration enum=v.elements(); while(enum.hasMoreElements())

{

Double d=new Double((String)enum.nextElement()); toplam+=d.doubleValue();

}

System.out.println("dosyanin ve yuklendigi vektorun icindeki sayilarin adedi: "+v.size());

System.out.println("Ortalama: "+(double)toplam/v.size());

} //main

} //class

sonuçlar :

Lutfen ilk dosya ismini giriniz: e.txt Lutfen ikinci dosya ismini giriniz: f.txt

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 1 2 3 4 5 6 7 8 9

dosyanin ve yuklendigi vektorun icindeki sayilarin adedi: 29 Ortalama: 15.0

e.txt:

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

d.txt

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

de.txt

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 1 2 3 4 5 6 7 8 9

### H10OD2

alistirma 7 deki sonuçlari Tree sinifini kullanarak küçükten büyüge dogru siralanmis olarak ed.txt dosyasina yazdiriniz.

### Program 11.29 : H10OD2.java

import java.io.\*; import Text; import java.util.\*;

import bolum11.Tree1; public class H10OD2

{

public void girdiA(String string, Vector list)

{

// StringTokeniser sinifi String degiskenleri icin

// Enumeration sinifinin yaptigina paralel gorev

// gorur String'i s• raya sokarak degerlerini s• rayla verir StringTokenizer tokens = new StringTokenizer(string); while(tokens.hasMoreTokens()) {

String test = tokens.nextToken(); list.addElement(test);

}

}

public static String ciktiA(Vector v)

{

StringBuffer cikti=new StringBuffer(); Enumeration enum = v.elements(); while(enum.hasMoreElements())

{

String ans = (String)enum.nextElement(); cikti.append(ans + " ");

}

return cikti.toString();

}

public static void main(String[] args) throws IOException

{

SetA set=new SetA(); Tree1 t=new Tree1(); DataInputStream input; Vector v1= new Vector(1);

Vector v2= new Vector(1); Vector v=new Vector(1); String s1,s2;

Text cin= new Text();

System.out.print("Lutfen ilk dosya ismini giriniz: "); s1 = cin.readString();

File myfile1 = new File(s1);

BufferedReader b1= new BufferedReader(new FileReader(myfile1)); System.out.print("Lutfen ikinci dosya ismini giriniz: ");

s2 = cin.readString();

File myfile2 = new File(s2);

BufferedReader b2= new BufferedReader(new FileReader(myfile2)); int toplam=0;

boolean EOF=false; while (!EOF)

{

try

{

v1.addElement(Text.readString(b1));

}

catch (EOFException e)

{

b1.close();

EOF=true;

}

} //while EOF=false; while (!EOF)

{

try

{

v2.addElement(Text.readString(b2));

}

catch (EOFException e)

{

b2.close();

EOF=true;

}

} //while v=set.bilesim(v1,v2);

Enumeration enum=v.elements(); while(enum.hasMoreElements())

{

Integer d=new Integer((String)enum.nextElement()); t.gir(d.intValue());

}

PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("de.txt"))); System.out.println(t.soldal\_node\_sagdal\_sirala()); cfout.println(t.soldal\_node\_sagdal\_sirala());

cfout.close();

} //main

} //class de.txt :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

e.txt :

11 13 24 15 17 22 18 19 12 20 21 23 25 14 26 27 16 28 29 10

d.txt

1 8 9 11 12 3 4 13 10 18 19 14 15 16 2 5 6 7 17 20

sonuç (ekran)

Lutfen ilk dosya ismini giriniz: d.txt Lutfen ikinci dosya ismini giriniz: e.txt

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

### Program 11.30 Tree1.java ,

package bolum11; class TreeNode

{

TreeNode sol; int data; TreeNode sag;

public TreeNode(int d)

{

data=d; sol=sag=null;

}

public synchronized void gir(int d)

{

// not ayni deger iki kere girilmeye calisilirsa

// ikinci giris dikkate alinmiyacaktir if(d<data)

{ if(sol==null) {sol=new TreeNode(d);} else {sol.gir(d);}

}

else if(ddata)

{ if(sag==null) {sag=new TreeNode(d);} else {sag.gir(d);}

}

}

}

public class Tree1

{

private TreeNode kok; public Tree1() {kok=null;}

public synchronized void gir(int d)

{

if(kok==null) kok=new TreeNode(d); else kok.gir(d);

}

public String node\_soldal\_sagdal\_sirala(){return node\_soldal\_sagdal\_siralayici(kok);} public String soldal\_node\_sagdal\_sirala(){return soldal\_node\_sagdal\_siralayici(kok);} public String sagdal\_node\_soldal\_sirala(){return sagdal\_node\_soldal\_siralayici(kok);} public String node\_soldal\_sagdal\_siralayici(TreeNode node)

{

String s=""; if(node==null) return s; s=s+node.data+" ";

s=s+node\_soldal\_sagdal\_siralayici(node.sol); s=s+node\_soldal\_sagdal\_siralayici(node.sag); return s;

}

public String soldal\_node\_sagdal\_siralayici(TreeNode node)

{

String s=""; if(node==null) return s;

s=s+soldal\_node\_sagdal\_siralayici(node.sol); s=s+node.data+" "; s=s+soldal\_node\_sagdal\_siralayici(node.sag);

return s;

}

public String sagdal\_node\_soldal\_siralayici(TreeNode node)

{

String s=""; if(node==null) return s;

s=s+sagdal\_node\_soldal\_siralayici(node.sol); s=s+node.data+" "; s=s+sagdal\_node\_soldal\_siralayici(node.sag); return s;

}

}

### H10OD3

Ekrandan bir rakam dizisini tek bir String olarak giriniz. Girdiginiz diziyi StringTokenezer sinifi bir gerçek sayi rakam dizisine dönüstürünüz ve vector sinifina yükleyiniz. daha sonra vector elemanlarinin karesinin ortalamasini çikti olarak veriniz.

### Program 11.31 : H10OD3.java

import java.io.\*; import Text; import java.util.\*;

import bolum11.Tree1; public class H10OD3

{

public static void main(String[] args) throws IOException

{

String s1;

Text cin= new Text();

Vector v=new Vector();

System.out.print("tum rakamlari giriniz : "); s1 = cin.readStringLine();

StringTokenizer st=new StringTokenizer(s1); while(st.hasMoreTokens())

{

v.addElement((String)st.nextToken());

}

Enumeration enum=v.elements(); double toplam=0; while(enum.hasMoreElements())

{

Double d=new Double((String)enum.nextElement()); double x=d.doubleValue();

toplam+=x\*x;

System.out.println("x = "+x+"toplam = "+toplam);

}

System.out.println("Ortalama: "+(double)toplam/v.size());

} //main

} //class sonuclar:

tum rakamlari giriniz : 1 2 3 4 5 6 7 8 9 10

x = 1.0 toplam = 1.0

x = 2.0 toplam = 5.0

x = 3.0 toplam = 14.0

x = 4.0 toplam = 30.0

x = 5.0 toplam = 55.0

x = 6.0 toplam = 91.0

x = 7.0 toplam = 140.0

x = 8.0 toplam = 204.0

x = 9.0 toplam = 285.0

x = 10.0 toplam = 385.0

Ortalama: 38.5

* 1. H10OD4

Ekrandan bir rakam dizisini tek bir String olarak giriniz. Girdiginiz diziyi StringTokenezer sinifi bir gerçek sayi rakam dizisine dönüstürünüz ve Stack sinifina yükleyiniz. daha sonra Stack elemanlarinin karesinin ortalamasini çikti olarak veriniz.

### Program 11.32 : H10OD4.java

import java.io.\*; import Text; import java.util.\*;

import bolum11.Tree1;

// not stack vektor s• n• f• n• n bir alt s• n• f• d• r.

// bu uygulamada bu ”zellikten yararlan• lm• Ÿt• r. public class H10OD4

{

public static void main(String[] args) throws IOException

{

String s1;

Text cin= new Text();

Stack v= new Stack();

System.out.print("tum rakamlari giriniz : "); s1 = cin.readStringLine();

StringTokenizer st=new StringTokenizer(s1); while(st.hasMoreTokens())

{

v.push((String)st.nextToken());

}

Enumeration enum=v.elements(); double toplam=0; while(enum.hasMoreElements())

{

Double d=new Double((String)enum.nextElement()); double x=d.doubleValue();

toplam+=x\*x;

System.out.println("x = "+x+"toplam = "+toplam);

}

System.out.println("Ortalama: "+(double)toplam/v.size());

} //main

} //class sonuçlar :

tum rakamlari giriniz : x = 1.0toplam = 1.0 x = 2.0toplam = 5.0

x = 3.0toplam = 14.0 x = 4.0toplam = 30.0 x = 5.0toplam = 55.0 x = 6.0toplam = 91.0 x = 7.0toplam = 140.0 x = 8.0toplam = 204.0 x = 9.0toplam = 285.0

x = 10.0toplam = 385.0 Ortalama: 38.5

**Program 11.33 : H10OD4a.java:** ikinci program (ayni program 11.20'nin degisik yazilisi ) : **H10OD4a.java**

import java.io.\*; import Text;

import java.util.\*; import bolum11.Tree1; public class H10OD4a

{

public static void main(String[] args) throws IOException

{

String s1;

Text cin= new Text();

Stack v= new Stack();

System.out.print("tum rakamlari giriniz : "); s1 = cin.readStringLine();

StringTokenizer st=new StringTokenizer(s1); while(st.hasMoreTokens())

{

v.push((String)st.nextToken());

}

double toplam=0; double sayi=v.size(); while(!v.isEmpty())

{

Double d=new Double((String)v.pop()); double x=d.doubleValue(); toplam+=x\*x;

System.out.println("x = "+x+"toplam = "+toplam);

}

System.out.println("Ortalama: "+(double)toplam/sayi);

} //main

} //class sonuçlar :

tum rakamlari giriniz : 1 2 3 4 5 6 7 8 9 10

x = 10.0 toplam = 100.0

x = 9.0 toplam = 181.0

x = 8.0 toplam = 245.0

x = 7.0 toplam = 294.0

x = 6.0 toplam = 330.0

x = 5.0 toplam = 355.0

x = 4.0 toplam = 371.0

x = 3.0 toplam = 380.0

x = 2.0 toplam = 384.0

x = 1.0 toplam = 385.0

Ortalama: 38.5

### Java Vector uygulamasi Gezegenler.java Program 11.34 . Gezegenler.java vectör test programi

import java.util.\*;

public class Gezegenler { static class Gezegen { private String isim; Gezegen (String s) {

isim = s;

}

public String toString() {

return getClass().getName() + "[" + isim + "]";

}

}

public static void main (String args[]) {

String isimler[] = {"Mercür", "Venüs", "Dünya", "Mars", "Jupiter", "Satürn", "Uranüs",

"Neptün", "Pluto"};

int namesLen = isimler.length;

Vector gezegenler = new Vector (namesLen); for (int i=0; i < namesLen; i++) {

gezegenler.addElement (new Gezegen (isimler[i]));

}

for (int i=0, n=gezegenler.size(); i < n; i++) { System.out.println (gezegenler.elementAt (i));

}

}

}

### GezegenlerveAylar.java, Vector uygulamasi Program 11.35 . GezegenlerveAylar.java vector test programi

import java.util.\*;

import javax.swing.JOptionPane;

public class GezegenlerveAylar { static class Gezegen {

private String isim; private int aySayisi;

Gezegen (String s, int aylar) { isim = s;

aySayisi = aylar;

}

public String toString() {

return getClass().getName() + "[" + isim + "-" + aySayisi + "]";

}

public final String getName() { return isim;

}

public final int getMoonCount () { return aySayisi;

}

}

public static void main (String args[]) {

String isimler[] = {"Mercür", "Venüs", "Dünya", "Mars", "Jupiter", "Satürn", "Uranüs", "Neptün", "Pluto"};

int aylar[] = {0, 0, 1, 2, 16, 18, 17, 8, 1};

int namesLen = isimler.length;

Vector planets = new Vector (namesLen); for (int i=0; i < namesLen; i++)

{

planets.addElement (new Gezegen (isimler[i], aylar[i]));

}

Enumeration enum = planets.elements(); Gezegen p;

String ss="";

while (enum.hasMoreElements()) { p = (Gezegen)(enum.nextElement());

ss+=p.getName() + " : " + p.getMoonCount()+"\n";

}

JOptionPane.showMessageDialog(null,ss); System.exit(0);

}

}

### BÖLÜM 12 ÖRNEKLERLE SAYISAL ANALIZ

### SAYISAL ANALIZE GIRIS

Bu bölümümüzde bazi pratik matematik problemlerinin java ile nasil çözülebilecegine göz atacagiz. Sayisal analiz islemlerini yapmak amaciyla iki sinif olusturduk bunlar Matrix ve Numeric siniflaridir. Bu serinin devami olan java ile numerik analiz kitabinda bu siniflardaki metotlari tek tek incelemeye çalisacagim. Burada ise sadece program listeleri ve örnek problemler vermekle yetinecegiz. Buradaki metodlar uluslararasi projelerde kullanildigindan metot isim ve tanimlamalari ingilizcedir. Sadece bu kitabin kapsami için degil gerçek nümerik problemlerde kullanilmak amaciyla gelistirilmislerdir. Burada ayrica hemen sunu da belirtelim: sayisal metodlarin kullanimi bu metodlar, sayisal analiz ve bilgisayar degiskenlerinin degisimi ile ilgili iyi bilgi gerektirmektedir. Bu sinif ve metodlari ciddi bir uygulamada kullanmadan önce verilen metodlar hakkinda nümerik analiz kitaplarndan yeterince bilgi sahibi olmanizi ve test fonksiyonlari kullanarak sizin istediginiz görevi yapip yapamiyacaklarini iyice kontrol etmenizi tavsiye ederim. Türkiyedeki nümerik analizle ilgilenen arkadaslar eger ilgilenirlerse benim kendi projelerimde kullanmak için gelistirdigim bu gurubu daha da gelistirerek genel bir nümerik analiz kütüphanesine dönüstürmemiz mümkün olabilir.

### MATRIX SINIFI

Matrix sinifi double ve complex vektör (bir boyutlu degisken) ve matris (iki boyutlu degisken) degiskenlerinin matematik islemlerini yapmak amaciyla tanimlanmistir. complex (kompleks degisken sinifi) daha önce türkçe olarak tanimlanan kompleks sinifinin aynidir, fakat metot isimleri burada ingilizce oldugundan bu sinifin tanimini tekrar verelim :

### Program 12.1 complex.java, ingilizce metot isimleriyle kompleks degiskenler sinifi.

//===========================================

// Numerical Analysis Package in Java

// Complex class defination

// Dr. Turhan Coban

//===========================================

import java.io.\*;

// class complex

// complex number abstraction

//

class complex {

// constructors public complex()

{

nreal=0; nimag=0;

}

public complex(double nre,double nim)

{

nreal=nre; nimag=nim;

}

public complex(double numer)

{

nreal=numer; nimag=0;

}

public complex(complex value )

{

nreal=value.real(); nimag=value.imaginary();

}

// accessor functions

public double real()

{

return nreal;

}

public double imaginary()

{

return nimag;

}

public void setReal(double r)

{

nreal=r;

}

public void setImaginary(double i)

{

nimag=i;

}

public double R()

{

return Math.sqrt(nreal\*nreal+nimag\*nimag);

}

public double theta()

{

return Math.atan2(nimag,nreal);

}

public double dtheta()

{

return Math.atan2(nimag,nreal)\*45.0/Math.atan(1.0);

}

// assignments

public void assign(complex right)

{

nreal=right.real(); nimag=right.imaginary();

}

public void assign(double nr,double ni)

{

nreal=nr; nimag=ni;

}

public void add(complex right)

{

nimag = nimag + right.imaginary(); nreal = nreal + right.real();

}

public void substract(complex right)

{

nimag = nimag - right.imaginary();

nreal = nreal - right.real();

}

public void multiply(complex right )

{

nreal = nreal\*right.real() - nimag\*right.imaginary(); nimag = nreal\*right.imaginary() + nimag\*right.real();

}

public void divide(complex right )

{

double a=nreal\*nreal+nimag\*nimag;

nreal = ( nreal\*right.real() + nimag\*right.imaginary())/a; nimag = (-nreal\*right.imaginary() + nimag\*right.real())/a;

}

public static complex add(complex left, complex right)

{ // return sum of two complex numbers double r1=(left.real() + right.real());

double i1=(left.imaginary() + right.imaginary()); complex result;

result=new complex(r1,i1); return result;

}

public static complex substract(complex left, complex right)

{ // return substraction of two complex numbers complex result;

result=new complex((left.real() - right.real()), (left.imaginary() - right.imaginary()));

return result;

}

public static complex multiply(complex left, complex right)

{ // return multiplication of two complex numbers complex result;

result=new complex

((left.real()\*right.real() - left.imaginary()\*right.imaginary()), (left.real()\*right.imaginary() + left.imaginary()\*right.real())); return result;

}

public static complex divide(complex left, complex right)

{ // return division of two complex numbers

double a=right.real()\*right.real()+right.imaginary()\*right.imaginary(); complex result;

result=new complex

((left.real()\*right.real() + left.imaginary()\*right.imaginary())/a, (-left.real()\*right.imaginary() + left.imaginary()\*right.real())/a); return result;

}

public static complex pow(complex left, double right)

{ // return sum of two complex numbers double Rad,th; Rad=Math.pow(left.R(),right); th=right\*left.theta();

complex result;

result =new complex((Rad\*Math.cos(th) ),

(Rad\*Math.sin(th) ) ); return result;

}

public boolean smaller(complex left,complex right)

{

// less then comparison of two complex numbers return (left.R() < right.R());

}

public boolean smaller\_equal(complex left,complex right)

{

// less then and equal comparison of two complex numbers return (left.R() <= right.R());

}

public boolean greater(complex left,complex right)

{

// greater then comparison of two complex numbers return left.R() > right.R();

}

public boolean greater\_equal(complex left,complex right)

{

// greater then and equal comparison of two complex numbers return left.R() >= right.R();

}

public boolean equal(complex left,complex right)

{

// equal comparison of two complex numbers return left.R() == right.R();

}

public boolean not\_equal(complex left,complex right)

{

// not equal comparison of two complex numbers return left.R() != right.R();

}

public static String toString(complex value)

{

String b=""; if(Math.abs(value.imaginary())!=1)

{

if(value.imaginary() >= 0)

b=b+"("+value.real()+" + "+value.imaginary()+"i )"; else

b=b+"("+value.real()+" - "+(-value.imaginary())+"i )";

}

else

{

if(value.imaginary() >= 0) b=b+"("+value.real()+" + i )"; else

b=b+"("+value.real()+" - i )";

}

return b;

}

public String toString()

{

String b=""; if(Math.abs(imaginary())!=1)

{

if(imaginary() > 0)

b=b+"("+real()+" + "+imaginary()+"i )"; else if(imaginary() <0 )

b=b+"("+real()+" - "+(-imaginary())+"i )"; else

b=b+real()+ " ";

}

else

{

if(imaginary() > 0) b=b+"("+real()+" + i )"; else if(imaginary() < 0) b=b+"("+real()+" - i )"; else

b=b+real()+" ";

}

return b;

}

// data areas double nreal; double nimag;

};

//end of class complex definations

Simdi Matrix metoduna daha detayli bakabiliriz. Ilkönce Matrix sinifinin metot basliklarini verelim :

### toString Metotlari

*public static String toString(double left)*

left double degiskeninin String esdegerini yazima hazir hale getirir.

*public static String toString(double[][] left)*

left double matrisini String degiskeni olarak yazima hazir hale getirir.

*public static String toString(complex[][] left)*

left complex matrisini String degiskeni olarak yazima hazir hale getirir.

*public static String toStringT(complex[] left)*

left complex vektörünü String degiskeni sütun olarak yazima hazir hale getirir.

*public static String toString(complex[] left)*

left complex vektörünü String degiskeni satir olarak yazima hazir hale getirir.

*public static String toStringT(double[] left)*

left double vektörünü String degiskeni sütun olarak yazima hazir hale getirir.

*public static String toString(double[] left)*

left complex vektörünü String degiskeni satir olarak yazima hazir hale getirir.

### inverse matris ve denklem sistemi çözüm hesaplari

Denklem çözümlerinde temel olarak iki sayisal analiz tekniginden yararlanilmistir. Birinci metod tam pivotlu gauss eliminasyon yöntem,, digeri ise LU ayristirma yöntemidir. Gauss eliminasyon yöntemi matrisi eliminasyon yöntemi kullanarak bir üst üçgen matris haline dönüstürür, yani diagonal elementlerin altinda kalan elemanlari sifirlar. Böylece denklem sistemi tek bilinmiyenli bir lineer denklemden basayarak iki bilinmiyenli, üç bilinmiyenli vs bir denklem sistemine dönüsür. Bu denklem sistemi bir bilinmiyenli denklemin çözülüp degerinin iki blinmiyenli denklemde yerine konulmasiyla ikincinin çözülmesi ve ayni islemin seri halde devamiyla tüm bilinmeyenler çözülür. LU ayristirma metodunda ise matris önce bir üst üçgen matris(U) ve bir alt üçgen matris (L) ye ayristirilir. Sistem L ve U matrislerini kullanarak çözülür. Inverse (ters) matris hesabi da yine bu iki yöntemin kullanilmasiyla gerçeklestirilir. Eger sadece bir matris çözülecekse gauss eliminasyon ve LU yöntemleri arasinda hesaplama verimi farki yoktur, ancak sadece denklemin sag tarafi degisiyorsa LU daha

verimli olarak hesap yapar. Hesaplama hassasiyetini arttirmak için pivoting adiverilen bir metod kullanilmistir. Pivoting matrisin satir ve sütunlarinin yerini degistirerek yuvarlama hatalarinin minimize edilmesi teknigidir.

*public static double[][] inv(double[][] a)*

inverse matrix hesaplar (tam pivotlu Gauss eliminasyon yöntemi)

*public static double[][] inverse(double a[][])*

inverse matrix hesaplar (tam pivotlu Gauss eliminasyon yöntemi)

*public static double[] LUaxb(double a[][],double x[],int indx[])*

lineer denklem sistemini LU metodu yardimiyla çözer.

*public static double[] AXB(double a[][],double b[])*

lineer denklem sistemini tam pivotlu gauss eliminasyon metodu yardimiyla çözer.

### Determinant hesaplari

Determinant özel bir matris çarpim prosesidir. Tek bir sayiyla ifade edilir. Determinant hesabi için LU metodu daha uygun oldugundan bu metod kullanilmistir. Asagidaki metodlar aslinda tek bir metod olup, degisik isimleri verebilmek amaciyla üç ayri metod olarak sunulmustur.

*public static double det(double a[][])*

LU metodu yardimiyla determinant hesaplar

*public static double determinant(double a[][])*

LU metodu yardimiyla determinant hesaplar

*public static double D(double a[][])*

LU metodu yardimiyla determinant hesaplar

### Matrislerin dört islemi

Burada matrislerin çarpimi (multiply) metodlarinda vektörel matris çarpimlarindan bahsediyoruz.

*public static double[][] multiply(double[][] left,double[][] right)*

iki double matris çarpimi

*public static double[] multiply(double[][] left,double[] right)*

double matris - vektör çarpimi

*public static double[] multiply(double[] left,double[][] right) double vektör - matris çarpimi*

*public static double[][] multiply(double left,double[][] right)*

double sayi - matris çarpimi

*public static double[][] multiply(double[][] left,double right) double matris - sayi çarpimi*

*public static double[] multiply(double left,double[] right) double sayi - vektör çarpimi*

*public static double[] multiply(double[] left,double right) double sayi - vektör çarpimi*

Matrislerin toplami(add) ayni konumda olan elemanlarin toplamidir. Fark islemi de (subtract) ayni konumda olan iki matrisden ilk verilenden ikinci verilenin isaratinin degistirilerek toplanmasidir.

*public static double[][] add(double[][] left,double[][] right) iki double matris toplami*

*public static double[] add(double[] left,double[] right) iki double vektör toplami*

*public static double[][] substract(double[][] left,double[][] right) iki double mtrisin farki*

*public static double[] substract(double[] left,double[] right) iki double vektörün farki*

vectorlerin bölümüyle birinci vectorün tersinin (inverse) alinarak ikinci matrisle çarpilmasi anlasilir. A/B = B\*inv(A)

*public static double[][] divide(double[][] left,double[][] right)*

*iki matrisin bölümü (birinci matrisin ikinci matrisin inversüyle çarpimi) public static double[][] LUdivide(double[][] left,double[][] right)*

*iki matrisin LU metodu kullanilarak bölümü (birinci matrisin ikinci matrisin inversüyle çarpimi) public static double[] LUdivide(double[] left,double[][] right)*

*vektörün matrise bölümü (LU metodu)*

*public static double[] divide(double[] left,double[][] right) vektörün matrise bölümü*

*public static double[] divide(double[] left,double[][] right)*

*vektörün matrise bölümü*

**Üst islemi** : üst isleminden anlasilan matristeki her elemanin ayri ayri üssünü almaktir

*public static double[][] pow(double[][] right,double left)*

matrisin double kuvveti (her elemaninin ayri ayri kuvveti)

*public static double[] pow(double[] right,double left)*

vektörün double kuvveti ( (her elemaninin ayri ayri kuvveti)

**Absulute matris** : absolute matris, matris elemanlarinin tümünün degerini positife çevirir.

*public static double abs(double[][] left)*

matrisin mutlak degeri

*public static double abs(double[] left)*

vektörün mutlak degeri

**Transpose matris** : transpose matris matrisin satir ve sütünlarinin yerini degistirir, yani satirlar sütun, sütunlar satir haline dönüsür. (aij)T=aji

*public static double[][] Transpose(double [][] left)*

*Transpose matrix (satir ve sütunlarin yer degistirmis hali) public static double[][] T(double [][] left)*

*Transpose matrix (satir ve sütunlarin yer degistirmis hali) public static complex[][] T(complex [][] left)*

*Transpose matrix (satir ve sütunlarin yer degistirmis hali)*

**Birim matris** : birim matris diagonal elemaninda (aii) 1 degeri olan diger elemanlari ise 0 tasiyan (aij,j!=I) matristir.

*public static double[][] I(int n)*

*Birim matris*

*public static double[] one(int n)*

Birim vektör (tüm degerler 1'e esit)

**EigenValue Problemi** : EigenValue, veya sinir deger problemi matris islemlerindeki ve matematikteki oldukça önemli bir islemdir. Sinir deger islemine kisaca bir denklem sisteminn köklerini bulma da diyebiliriz. Genel tanim olarak [A] matrisinin sinir degeri

[A]{X}-[I]=0 olarak tanimlanir. Burada {X} denklemi saglayan vektör setidir, eigen vektörü (sinir deger vektörü) olarak adlandirilir.  degeri ise EigenValue(sinir degeri) adini alir. Buradaki [I] birim matrisdir. Süphesiz, verilen bir matris için birden fazla eigenvektör ve eigenvalue bulunabilir.

*public static double[][] balance(double b[][])*

balance islemi eigenvalue hesaplamalarini kolaylastiran bir ilk iterasyon prosesidir. Bazi durumlarda sonuca ulasmayabilir. Bu yüzden dikkatli kullanilmalidir.

*public static double[][] Hessenberg(double b[][])*

Hessenberg matris dönüsümü, eigenvalue hesaplamalarinda kullanilir.

*public static double[][] QR(double b[][])*

Eigenvalue (sinir-deger hesaplama metodu, simetrik olmayan matrislere de uygulanabilir.) sonuçlar n\*2 (gerçek ve imajineri) matris olarak aktarilir.

*public static double[][] eigenValue(double b[][])*

QR metodunun aynidir.

*public static complex[] eigenValueC(double b[][])*

*QR metodunun aynidir tek farki sonuçlari kompleks vektör olarak vermesidir. public static double[][] balancedEigenValue(double b[][])*

QR metodu uygulanmadan önce balance metodunu uygular.

*public static double[][] eigenQL(double a[][])*

Eigenvalue (sinir-deger hesaplama metodu, sadece simetrik matrislere uygulanabilir, simetrik eigenvalue degerleri gerçektir,, komplex kökler vermez)

### Polinomun kökleri

Polinomun köklerinin bulunmasi aslinda bir eigenvalue (sinirdeger) problemidir. Herhangi bir dereceden polinomun katsayilari verildiginde karekteristikmatris adi verilen bir matris esdegeri olusturulabilir. Bu matrisin eigenvalue degerleri verilen polinomun kökleridir.

*public static double[][] poly\_roots(double c[])*

QR Eigenvalue (sinir-deger hesaplama metodu, simetrik olmayan matrislere de uygulanabilir.)

metodu kullanarak polinomun köklerini hesaplar. polinom katsayilarini önce karektteristik matrise çevirir. sonuçlar n\*2 (gerçek ve imajineri) matris olarak aktarilir.

*public static complex[] poly\_rootsC(double c[])*

Yukaridakinin aynidir. ancak sonuçlar kompleks vektör olarak aktarilir.

### 12.1.1 Matrix Sinifinin Listesi :

**Problem 12.2** Matrix.java, **Matris proseslerini tanimlayan Matrix sinifi**

//=============================================================

// Numerical Analysis Packages in Java

// Matrix calculations class Matrix

// Author : Dr. Turhan Coban

//=============================================================

//Turhan Coban import java.io.\*; import complex;

public class Matrix

{

// This class defines matrix and vector

// calculation methods

public static String toString(double left)

{

//arrange double to string conversion so that all the

//matrix double variables nicely printed in the same column String s="";

if(left=0) s=s+" "; s=s+left;

double n=s.length(); while(n<13)

{

s=s+" ";

n=s.length();

}

return s;

}

public static String toString(double[][] left)

{

//return a string representation of a matrix int n,m;

String b; b="";

n=left.length; m=left[0].length; for(int i=0;i<n;i++)

{

for(int j=0;j<m;j++)

{

b=b+toString(left[i][j]);

}

b=b+"\n";

}

return b;

}

public static String toString(complex[][] left)

{

//return a string representation of a complex matrix

int n,m; String b; b="";

n=left.length; m=left[0].length; for(int i=0;i<n;i++)

{

for(int j=0;j<m;j++)

{

b=b+left[i][j].toString()+"\t";

}

b=b+"\n";

}

return b;

}

public static String toStringT(complex[] left)

{

// returns a horizontal string representation of

// a complex vector int n,m;

String b; b="";

n=left.length; for(int i=0;i<n;i++)

{

b=b+left[i].toString()+"\n";

}

return b;

}

public static String toString(complex[] left)

{

// returns a vertical string representation of

// a complex vector int n,m;

String b; b="";

n=left.length; for(int i=0;i<n;i++)

{

b=b+left[i].toString()+" ";

}

b=b+"\n"; return b;

}

public static String toStringT(double[] left)

{

// returns a vertical string representation

// of a double vector int n,m;

String b; b="";

n=left.length; for(int i=0;i<n;i++)

{

b=b+toString(left[i])+"\n";

}

return b;

}

public static String toString(double[] left)

{

// returns a horizontal string representation

// of a double vector int n,m;

String b; b="";

n=left.length; for(int i=0;i<n;i++)

{

b=b+toString(left[i]);

}

b=b+"\n"; return b;

}

public static String toString(int[] left)

{

// returns a horizontal string representation

// of a integer vector int n,m;

String b; b="";

n=left.length; for(int i=0;i<n;i++)

{

b=b+left[i]+"\t";

}

b=b+"\n"; return b;

}

public static double SIGN(double a,double b)

{

//returns the value of double a with sign of double b;

//if a=-2, b= 3 SIGN(a,b) returns 2

//if a=-2, b=-3 SIGN(a,b) returns -2

//if a= 2, b=-3 SIGN(a,b) returns -2

//if a= 2, b= 3 SIGN(a,b) returns 2 if(b!=0)

return Math.abs(a)\*b/Math.abs(b); else

return Math.abs(a);

}

public static double[][] inv(double[][] a)

{

// INVERSION OF A MATRIX

// inversion by using gaussian elimination

// with full pivoting int n=a.length;

int m=a[0].length; double b[][];

b=new double[n][n]; int indxc[];

int indxr[]; double ipiv[]; indxc=new int[n]; indxr=new int[n];

ipiv=new double[n]; int i,j,k,l,ll,ii,jj;

int icol=0; int irow=0;

double big,dum,pivinv,temp;

if(n!=m)

{

System.out.println("Matrix must be square "); for(ii=0;ii<n;ii++)

for(jj=0;jj<n;jj++) b[ii][jj]=0.0;

return b;

}

for(i=0;i<n;i++) for(j=0;j<n;j++)

b[i][j]=a[i][j]; for(i=0;i<n;i++)

{

big=0.0; for(j=0;j<n;j++)

{

if(ipiv[j] != 1) for(k=0;k<n;k++)

{

if(ipiv[k] == 0)

{

if(Math.abs(b[j][k]) = big)

{

big=Math.abs(b[j][k]); irow=j;

icol=k;

}

}

else if(ipiv[k] 1 )

{

System.out.println("error : inverse of the matrix : singular matrix-1"); for(ii=0;ii<n;ii++)

for(jj=0;jj<n;jj++) b[ii][jj]=0.0;

return b;

}

}

}

++ ipiv[icol]; if(irow != icol) for(l=0;l<n;l++)

{

temp=b[irow][l]; b[irow][l]=b[icol][l]; b[icol][l]=temp;

}

indxr[i]=irow; indxc[i]=icol; if(b[icol][icol] == 0.0)

{

System.out.println("error : inverse of the matrix : singular matrix-2"); for(ii=0;ii<n;ii++)

for(jj=0;jj<n;jj++) b[ii][jj]=0.0;

return b;

}

pivinv=1.0/b[icol][icol]; b[icol][icol]=1.0;

for(l=0;l<n;l++) b[icol][l] \*=pivinv; for(ll=0;ll<n;ll++)

if(ll != icol)

{

dum=b[ll][icol];

b[ll][icol]=0.0;

for(l=0;l<n;l++) b[ll][l]-= b[icol][l]\*dum;

}

}

for(l=n-1;l=0;l--)

{

if(indxr[l] != indxc[l]) for(k=0;k<n;k++)

{

temp=b[k][indxc[l]]; b[k][indxc[l]]=b[k][indxr[l]]; b[k][indxr[l]]=temp;

}

}

return b;

}

public static double[][] inverse(double a[][])

{

//inverse of a matrix

//this method enable usage of inv or inverse as

//name of mtrix inversion method return Matrix.inv(a);

}

//LU decomposition method

public static double[][] LU(double c[][],int indx[],int d[])

{

//returns LU decomposition of matrix c and index indx double a[][];

int n=c.length; a=new double[n][n]; double vv[]; vv=new double[n];

double sum,dum,big,temp; int i,j,k;

int imax;

int nmax=100; double tiny=1.0e-40; imax=0; for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

a[i-1][j-1]=c[i-1][j-1];

}

d[0]=1;

for(i=1;i<=n;i++)

{

big=0.0; for(j=1;j<=n;j++)

{

if(Math.abs(a[i-1][j-1])big) big=Math.abs(a[i-1][j-1]);

}

if(big==0) {System.out.println("singular matrix");return a;} vv[i-1]=1.0/big;

}

for(j=1;j<=n;j++)

{

for(i=1;i<j;i++)

{

sum=a[i-1][j-1]; for(k=1;k<i;k++)

{

sum-=a[i-1][k-1]\*a[k-1][j-1];

}

a[i-1][j-1]=sum;

}

big=0; for(i=j;i<=n;i++)

{

sum=a[i-1][j-1]; for(k=1;k<j;k++)

{

sum-=a[i-1][k-1]\*a[k-1][j-1];

}

a[i-1][j-1]=sum;

dum=vv[i-1]\*Math.abs(sum); if(dum=big)

{

imax=i; big=dum;

}

} //end of i=0 if(j != imax)

{

for(k=1;k<=n;k++)

{

dum=a[imax-1][k-1]; a[imax-1][k-1]=a[j-1][k-1]; a[j-1][k-1]=dum;

}

d[0]=-d[0];

vv[imax-1]=vv[j-1];

} //end of if indx[j-1]=imax;

if(a[j-1][j-1]==0) a[j-1][j-1]=tiny; if(j!=n)

{

dum=1.0/a[j-1][j-1]; for(i=j+1;i<=n;i++) a[i-1][j-1]\*=dum;

}//endif

} //end for j= return a;

}

public static double[] LUaxb(double a[][],double x[],int indx[])

{

//solves AX=B system of linear equation of LU decomposed matrix a

//(calculated by method LU) int ii=0;

int i,j,ll=0; double sum=0; int n=a.length; double b[]; b=new double[n]; for(i=1;i<=n;i++)

{

b[i-1]=x[i-1];

}

for(i=1;i<=n;i++)

{

ll=indx[i-1];

sum=b[ll-1]; b[ll-1]=b[i-1]; if(ii!=0)

{

for(j=ii;j<=(i-1);j++)

{

sum-=a[i-1][j-1]\*b[j-1];

}

}

else if(sum!=0) ii=i; b[i-1]=sum;

}

for(i=n;i=1;i--)

{

sum=b[i-1]; if(i<n)

{

for(j=(i+1);j<=n;j++)

{

sum-=a[i-1][j-1]\*b[j-1];

}

}

b[i-1]=sum/a[i-1][i-1];

}

return b;

}

public static double[] AXB(double a[][],double b[])

{

//Solution of system of linear equations by LU method

// note that the same calculation can be done by divide method. int n=a.length;

double c[]=new double[n]; int d[]={1};

int indx[]=new int[n];

double e[][]=new double[n][n]; e=Matrix.LU(a,indx,d); c=Matrix.LUaxb(e,b,indx); return c;

} //end of AXB

public static double[][] LUinv(double a[][])

{

//inverse of a matrix by using LU decomposition method

//this method is more efficient than inv (or inverse) int n=a.length;

double c[][]=new double[n][n];

double b[][]=Matrix.I(n); int d[]={0};

int indx[]=new int[n];

double e[][]=new double[n][n]; e=Matrix.LU(a,indx,d);

for(int i=0;i<n;i++)

{

c[i]=Matrix.LUaxb(e,b[i],indx);

}

return Matrix.T(c);

} //end of LUinv

public static double det(double a[][])

{

//determinant of a matrix int n=a.length;

int indx[]=new int[n]; int d[]={1};

double e;

double b[][]=new double[n][n]; b=Matrix.LU(a,indx,d); e=d[0];

for(int i=0;i<n;i++) e=e\*b[i][i];

return e;

} //end of det

public static double determinant(double a[][])

{

//determinant of a matrix return Matrix.det(a);

}

public static double D(double a[][])

{

//determinant of a matrix return Matrix.det(a);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\* multiply methods definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* public static double[][] multiply(double[][] left,double[][] right)

{

//multiplication of two matrices int ii,jj,i,j,k;

int m1=left[0].length; int n1=left.length;

int m2=right[0].length; int n2=right.length; double[][] b;

b=new double[m1][n2]; if(n1 != m2)

{

System.out.println("inner matrix dimensions must agree"); for(ii=0;ii<n1;ii++)

{

for(jj=0;jj<m2;jj++) b[ii][jj]=0;

}

return b;

}

for(i=0;i<m1;i++)

{

for(j=0;j<n2;j++)

{

for(k=0;k<n1;k++) b[i][j]+=left[i][k]\*right[k][j];

}

}

return b;

//end of multiply of two matrices

}

public static double[] multiply(double[][] left,double[] right)

{

//multiplication of one matrix with one vector

int ii,jj,i,j,k;

int m1=left[0].length; int n1=left.length;

int m2=right.length; double[] b;

b=new double[m2]; if(n1 != m2)

{

System.out.println("inner matrix dimensions must agree"); for(ii=0;ii<n1;ii++)

{

b[ii]=0;

}

return b;

}

for(i=0;i<m1;i++)

{

b[i]=0;

for(k=0;k<n1;k++) b[i]+=left[i][k]\*right[k];

}

return b;

//end of multiply of a matrix and a vector

}

public static double[] multiply(double[] left,double[][] right)

{

//multiplication of one vector with one matrix int ii,jj,i,j,k;

int m2=right[0].length; int n2=right.length;

int m1=left.length; double[] b;

b=new double[m1]; if(n2 != m1)

{

System.out.println("inner matrix dimensions must agree"); for(ii=0;ii<n2;ii++)

{

b[ii]=0;

}

return b;

}

for(i=0;i<m2;i++)

{

b[i]=0;

for(k=0;k<m1;k++) b[i]+=right[i][k]\*left[k];

}

return b;

//end of multiply of a vector and a matrix

}

public static double[][] multiply(double left,double[][] right)

{

//multiplying a matrix with a constant int i,j;

int n=right.length;

int m=right[0].length; double b[][];

b=new double[n][m]; for(i=0;i<n;i++)

{

for(j=0;j<m;j++) b[i][j]=right[i][j]\*left;

}

return b;

//end of multiplying a matrix with a constant double

}

public static double[][] multiply(double[][] left,double right)

{

//multiplying a matrix with a constant int i,j;

int n=left.length;

int m=left[0].length; double b[][];

b=new double[n][m]; for(i=0;i<n;i++)

{

for(j=0;j<m;j++) b[i][j]=left[i][j]\*right;

}

return b;

//end of multiplying a matrix with a constant double

}

public static double[] multiply(double left,double[] right)

{

//multiplying a vector with a constant int i;

int n=right.length; double b[]; b=new double[n]; for(i=0;i<n;i++)

{

b[i]=left\*right[i];

}

return b;

}

public static double[] multiply(double[] left,double right)

{

//multiplying a vector with a constant int i;

int n=left.length; double b[]; b=new double[n]; for(i=0;i<n;i++)

{

b[i]=right\*left[i];

}

return b;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end of multiply methods definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*

//=============== defination of power methods pow ============== public static double[][] pow(double[][] right,double left)

{

// power of a matrix int i,j;

double b[][];

int n=right.length;

int m=right[0].length; b=new double[n][m]; for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

if(left==0.0)

{

b[i][j]=1.0;

}

else

{

b[i][j]=Math.pow(right[i][j],left);

}

}

}

return b;

//end of power of a matrix

}

public static double[] pow(double[] right,double left)

{

// power of a vector int i;

int n=right.length; double b[]; b=new double[n]; for(i=0;i<n;i++)

{

if(left==0.0)

{

b[i]=1.0;

}

else

{

b[i]=Math.pow(right[i],left);

}

}

return b;

//end of power of a vector

}

//=================end of power method pow definitions =============

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* addition add methods \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* public static double[][] add(double[][] left,double[][] right)

{

//addition of two matrices int n1=left.length;

int m1=left[0].length; int n2=right.length;

int m2=right[0].length; int nMax,mMax;

int i,j;

if(m1=m2) mMax=m1; else mMax=m2; if(n1=n2) nMax=n1; else nMax=n2; double b[][];

b=new double[nMax][mMax]; for(i=0;i<n1;i++)

{

for(j=0;j<m1;j++)

{

b[i][j]=b[i][j]+left[i][j];

}

}

for(i=0;i<n2;i++)

{

for(j=0;j<m2;j++)

{

b[i][j]=b[i][j]+right[i][j];

}

}

return b;

//end of matrix addition method

}

public static double[] add(double[] left,double[] right)

{

//addition of two vectors int n1=left.length;

int n2=right.length; int nMax;

int i;

if(n1=n2) nMax=n1; else nMax=n2; double b[];

b=new double[nMax]; for(i=0;i<n1;i++)

{

b[i]=b[i]+left[i];

}

for(i=0;i<n2;i++)

{

b[i]=b[i]+right[i];

}

return b;

//end of vector addition method

}

public static double[][] substract(double[][] left,double[][] right)

{

//addition of two matrices int n1=left.length;

int m1=left[0].length; int n2=right.length;

int m2=right[0].length; int nMax,mMax;

int i,j;

if(m1=m2) mMax=m1; else mMax=m2; if(n1=n2) nMax=n1; else nMax=n2; double b[][];

b=new double[nMax][mMax]; for(i=0;i<n1;i++)

{

for(j=0;j<m1;j++)

{

b[i][j]=b[i][j]+left[i][j];

}

}

for(i=0;i<n2;i++)

{

for(j=0;j<m2;j++)

{

b[i][j]=b[i][j]-right[i][j];

}

}

return b;

//end of matrix substraction method

}

public static double[] substract(double[] left,double[] right)

{

//addition of two vectors int n1=left.length;

int n2=right.length; int nMax;

int i;

if(n1=n2) nMax=n1; else nMax=n2; double b[];

b=new double[nMax]; for(i=0;i<n1;i++)

{

b[i]=b[i]+left[i];

}

for(i=0;i<n2;i++)

{

b[i]=b[i]-right[i];

}

return b;

//end of vector substraction method

}

//============== division of the matrices

public static double[][] divide(double[][] left,double[][] right)

{

//division of two matrices int n=right.length;

int m=right[0].length; double b[][];

b=new double[n][m]; b=Matrix.multiply(Matrix.inv(right),left); return b;

}

public static double[][] LUdivide(double[][] left,double[][] right)

{

//division of two matrices utilises LUinv method instead of inv int n=right.length;

int m=right[0].length; double b[][];

b=new double[n][m]; b=Matrix.multiply(Matrix.LUinv(right),left); return b;

}

public static double[] divide(double[] left,double[][] right)

{

//division of two matrices int n=right.length;

int m=right[0].length; double b[];

b=new double[n]; b=Matrix.multiply(Matrix.inv(right),left); return b;

}

public static double[] LUdivide(double[] left,double[][] right)

{

//division of two matrices utilises AXB (LU decomposition method)

//in fact this method is exactly same as AXB except spacing of the

//arguments

return AXB(right,left);

}

//============== absolute value of a matrix=================== public static double abs(double[][] left)

{

// absoulute value of a matrix int i,j;

int n=left.length;

int m=left[0].length; double b=0; for(i=0;i<n;i++) for(j=0;j<m;j++)

{

b=b+Math.abs(left[i][j]);

}

return b;

}

public static double abs(double[] left)

{

// absolute value of a vector int i;

int n=left.length; double b=0; for(i=0;i<n;i++)

{

b=b+Math.abs(left[i]);

}

return b;

}

//===============special matrices==============================

public static double[][] Transpose(double [][] left)

{

//transpose matrix (if A=a(i,j) Transpose(A)=a(j,i) int i,j;

int n=left.length;

int m=left[0].length; double b[][];

b=new double[m][n]; for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

b[j][i]=left[i][j];

}

}

return b;

}

public static double[][] T(double [][] left)

{

//transpose matrix (if A=a(i,j) T(A)=a(j,i) int i,j;

int n=left.length;

int m=left[0].length; double b[][];

b=new double[m][n];

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

b[j][i]=left[i][j];

}

}

return b;

}

public static complex[][] T(complex [][] left)

{

//transpose matrix (if A=a(i,j) T(A)=a(j,i) int i,j;

int n=left.length;

int m=left[0].length; complex b[][];

b=new complex[m][n]; for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

b[j][i]=new complex(left[i][j]);

}

}

return b;

}

public static double[][] I(int n)

{

//unit matrix double b[][];

b=new double[n][n]; for(int i=0;i<n;i++)

b[i][i]=1.0;

return b;

}

public static double[] one(int n)

{

//one matrix double b[]; b=new double[n];

for(int i=0;i<n;i++) b[i]=1.0;

return b;

}

public static double[][] characteristic\_matrix(double c[])

{

//this routine converts polynomial coefficients to a matrix

//with the same eigenvalues (roots) int n=c.length-1;

int i;

double a[][]=new double[n][n]; for(i=0;i<n;i++)

{

a[0][i]=-c[i+1]/c[0];

}

for(i=0;i<n-1;i++)

{

a[i+1][i]=1;

}

return a;

}

//===========Eigen value calculations ============== public static double[][] balance(double b[][])

{

// balance of a matrix for more accurate eigenvalue

// calculations double radix=2.0;

double sqrdx=radix\*radix; double c,r,f,s,g;

int m,j,i,last; int n=b.length; last=0;

double a[][];

a=new double[n][n]; f=1;

s=1;

for(i=1;i<=n;i++) for(j=1;j<=n;j++)

a[i-1][j-1]=b[i-1][j-1];

while(last==0)

{

last=1; for(i=1;i<=n;i++)

{ c=0;r=0;

for(j=1;j<=n;j++)

{

if(j != i)

{

c+=Math.abs(a[j-1][i-1]);

r+=Math.abs(a[i-1][j-1]);

} //end of if(j!=..

} //end of for(j=1... if(c != 0 && r != 0 )

{

g=r/radix; f=1.0;

s=c+r; while(c<g)

{

f\*=radix; c\*=sqrdx;

}

g=r\*radix; while(cg)

{

f/=radix; c/=sqrdx;

}

} //end of if(c != 0 && .... if( (c+r)/f < 0.95\*s )

{

last=0; g=1.0/f;

for(j=1;j<=n;j++) { a[i-1][j-1]\*=g; }

for(j=1;j<=n;j++) { a[j-1][i-1]\*=f; }

} //end of if( ((c+r..

}//end of for(i=1;i<=n....

} //end of while last==0

return a;

}

public static double[][] Hessenberg(double b[][])

{

// Calculates the hessenberg matrix

// it is used in QR method to calculate eigenvalues

// of a matrix(symmetric or non-symmetric) int m,j,i;

int n=b.length; double a[][];

a=new double[n][n]; for(i=0;i<n;i++) for(j=0;j<n;j++)

a[i][j]=b[i][j]; double x,y; if(n2)

{

for(m=2;m<=(n-1);m++)

{ x=0.0;

i=m; for(j=m;j<=n;j++)

{

if(Math.abs(a[j-1][m-2]) Math.abs(x))

{

x=a[j-1][m-2];

i=j;

} //end of if(Math.abs(..

}//end of for(j=m,j<=n... if(i!=m)

{

for(j=(m-1);j<=n;j++)

{

y=a[i-1][j-1];

a[i-1][j-1]=a[m-1][j-1]; a[m-1][j-1]=y;

}//end of for(j=(m-1).. for(j=1;j<=n;j++)

{

y=a[j-1][i-1];

a[j-1][i-1]=a[j-1][m-1]; a[j-1][m-1]=y;

}//end of for(j=1;j<=n....

} //end of if(i!=m) if(x != 0.0)

{

for(i=(m+1);i<=n;i++)

{

y=a[i-1][m-2]; if(y!=0.0)

{

y=y/x;

a[i-1][m-2]=y;

for(j=m;j<=n;j++)

{

a[i-1][j-1]-=y\*a[m-1][j-1];

}

for(j=1;j<=n;j++)

{

a[j-1][m-1]+=y\*a[j-1][i-1];

}

}//end of if(y!=0..

}//end of for(i=(m+1)...

} //end of if(x != 0.0...

}//end of for(m=2;m<=(n-1)..

}//end of Hessenberg for(i=1;i<=n;i++) for(j=1;j<=n;j++)

{

if(i(j+1)) a[i-1][j-1]=0;

}

return a;

}

public static double[][] QR(double b[][])

{

//calculates eigenvalues of a Hessenberg matrix int n=b.length;

double rm[][]=new double[2][n]; double a[][]=new double[n+1][n+1]; double wr[]=new double[n+1]; double wi[]=new double[n+1];

int nn,m,l,k,j,its,i,mmin;

double z,y,x,w,v,u,t,s,r=0,q=0,p=0,anorm; for(i=0;i<n;i++)

for(j=0;j<n;j++) a[i+1][j+1]=b[i][j];

anorm=Math.abs(a[1][1]); for (i=2;i<=n;i++)

for (j=(i-1);j<=n;j++)

anorm += Math.abs(a[i][j]); nn=n;

t=0.0;

while (nn = 1) { its=0;

do {

for (l=nn;l=2;l--) {

s=Math.abs(a[l-1][l-1])+Math.abs(a[l][l]); if (s == 0.0) s=anorm;

if ((double)(Math.abs(a[l][l-1]) + s) == s) break;

}

x=a[nn][nn]; if (l == nn) {

wr[nn]=x+t; wi[nn--]=0.0;

}

else {

y=a[nn-1][nn-1]; w=a[nn][nn-1]\*a[nn-1][nn]; if (l == (nn-1)) {

p=0.5\*(y-x); q=p\*p+w;

z=Math.sqrt(Math.abs(q)); x += t;

if (q = 0.0) { z=p+Matrix.SIGN(z,p);

wr[nn-1]=wr[nn]=x+z; if (z!=0) wr[nn]=x-w/z;

wi[nn-1]=wi[nn]=0.0;

}

else {

wr[nn-1]=wr[nn]=x+p; wi[nn-1]= -(wi[nn]=z);

}

nn -= 2;

}

else {

if (its == 30) System.out.println("Too many iterations in hqr"); if (its == 10 || its == 20) {

t += x;

for (i=1;i<=nn;i++) a[i][i] -= x; s=Math.abs(a[nn][nn-1])+Math.abs(a[nn-1][nn-2]); y=x=0.75\*s;

w = -0.4375\*s\*s;

}

++its;

for (m=(nn-2);m=l;m--) { z=a[m][m];

r=x-z;

s=y-z;

p=(r\*s-w)/a[m+1][m]+a[m][m+1]; q=a[m+1][m+1]-z-r-s; r=a[m+2][m+1];

s=Math.abs(p)+Math.abs(q)+Math.abs(r); p /= s;

q /= s; r /= s;

if (m == l) break;

u=Math.abs(a[m][m-1])\*(Math.abs(q)+Math.abs(r));

v=Math.abs(p)\*(Math.abs(a[m-1][m-1])+

Math.abs(z)+Math.abs(a[m+1][m+1])); if ((double)(u+v) == v) break;

}

for (i=m+2;i<=nn;i++) { a[i][i-2]=0.0;

if (i != (m+2)) a[i][i-3]=0.0;

}

for (k=m;k<=nn-1;k++) { if (k != m) {

p=a[k][k-1];

q=a[k+1][k-1]; r=0.0;

if (k != (nn-1)) r=a[k+2][k-1];

if ((x=Math.abs(p)+Math.abs(q)+Math.abs(r)) != 0.0) { p /= x;

q /= x; r /= x;

}

}

if ((s=Matrix.SIGN(Math.sqrt(p\*p+q\*q+r\*r),p)) != 0.0) { if (k == m) {

if (l != m)

a[k][k-1] = -a[k][k-1];

}

else

a[k][k-1] = -s\*x; p += s;

x=p/s; y=q/s; z=r/s; q /= p;

r /= p;

for (j=k;j<=nn;j++) { p=a[k][j]+q\*a[k+1][j];

if (k != (nn-1)) {

p += r\*a[k+2][j];

a[k+2][j] -= p\*z;

}

a[k+1][j] -= p\*y;

a[k][j] -= p\*x;

}

mmin = nn<k+3 ? nn : k+3; for (i=l;i<=mmin;i++) {

p=x\*a[i][k]+y\*a[i][k+1];

if (k != (nn-1)) {

p += z\*a[i][k+2];

a[i][k+2] -= p\*r;

}

a[i][k+1] -= p\*q;

a[i][k] -= p;

}

}

}

}

}

} while (l < nn-1);

}

for(i=0;i<n;i++)

{

rm[0][i]=wr[i+1];

rm[1][i]=wi[i+1];

}

return rm;

} //end of QR

public static double[][] eigenValue(double b[][])

{

// this routine input a matrix (non symetric or symmetric)

// and calculate eigen values

// method balance can be used prior to this method to balance

// the input matrix int n=b.length;

double d[][]=new double[2][n]; d=Matrix.QR(Matrix.Hessenberg(b)); return d;

}

public static complex[] eigenValueC(double b[][])

{

// this routine input a matrix (non symetric or symmetric)

// and calculate eigen values

// method balance can be used prior to this method to balance

// the input matrix

//output eigenvalues will be in a vector of complex form int n=b.length;

double d[][]=new double[2][n]; d=Matrix.QR(Matrix.Hessenberg(b)); complex c[]=new complex[n];

for(int i=0;i<n;i++)

{

c[i]=new complex(d[0][i],d[1][i]);

}

return c;

}

//roots of a polynomial

public static double[][] poly\_roots(double c[])

{

//roots of a degree n polynomial

// P(x)=c[n]\*x^n+c[n-1]\*x^(n-1)+....+c[1]\*x+c[0]=0; int n=c.length-1;

double a[][]=new double[n][n]; a=characteristic\_matrix(c); double d[][]=new double[2][n]; d=balancedEigenValue(a); return d;

}

public static complex[] poly\_rootsC(double c[])

{

// roots of a degree n polynomial

// P(x)=c[n]\*x^n+c[n-1]\*x^(n-1)+....+c[1]\*x+c[0]=0;

// roots are returned as complex variables int n=c.length-1;

double a[][]=new double[n][n]; a=characteristic\_matrix(c); double d[][]=new double[2][n]; d=balancedEigenValue(a); complex e[]=new complex[n]; for(int i=0;i<n;i++)

e[i]=new complex(d[0][i],d[1][i]); return e;

}

public static double[][] balancedEigenValue(double b[][])

{

// this routine input a matrix (non symetric or symmetric)

// and calculates eigen values

// method balance is used to balance the matrix previous to

// actual calculations int n=b.length;

double d[][]=new double[2][n]; d=Matrix.QR(Matrix.Hessenberg(Matrix.balance(b))); return d;

}

public static double[][] tridiagonal(double b[][], double d[], double e[])

{

//reduces matrix to tridiaonal form by using householder transformation

//this method is used by QL method to calculate eigen values

//and eigen vectors of a symmetric matrix int l,k,j,i;

int n=b.length; double scale,hh,h,g,f;

double a[][]=new double[n+1][n+1]; double c[][]=new double[n][n]; for(i=0;i<n;i++)

for(j=0;j<n;j++) a[i][j]=b[i][j];

for (i=n;i=2;i--) { l=i-1; h=scale=0.0;

if (l 1) {

for (k=1;k<=l;k++)

scale += Math.abs(a[i-1][k-1]); if (scale == 0.0)

e[i-1]=a[i-1][l-1];

else {

for (k=1;k<=l;k++) { a[i-1][k-1] /= scale;

h += a[i-1][k-1]\*a[i-1][k-1];

}

f=a[i-1][l-1];

g=(f = 0.0 ? -Math.sqrt(h) : Math.sqrt(h)); e[i-1]=scale\*g;

h -= f\*g;

a[i-1][l-1]=f-g; f=0.0;

for (j=1;j<=l;j++) {

a[j-1][i-1]=a[i-1][j-1]/h; g=0.0;

for (k=1;k<=j;k++)

g += a[j-1][k-1]\*a[i-1][k-1];

for (k=j+1;k<=l;k++)

g += a[k-1][j-1]\*a[i-1][k-1];

e[j-1]=g/h;

f += e[j-1]\*a[i-1][j-1];

}

hh=f/(h+h);

for (j=1;j<=l;j++) { f=a[i-1][j-1];

e[j-1]=g=e[j-1]-hh\*f; for (k=1;k<=j;k++)

a[j-1][k-1] -= (f\*e[k-1]+g\*a[i-1][k-1]);

}

}

} else

e[i-1]=a[i-1][l-1];

d[i-1]=h;

}

d[1-1]=0.0;

e[1-1]=0.0;

/\* Contents of this loop can be omitted if eigenvectors not wanted except for statement d[i-1]=a[i-1][i-1]; \*/

for (i=1;i<=n;i++) { l=i-1;

if (d[i-1] != 0) {

for (j=1;j<=l;j++) { g=0.0;

for (k=1;k<=l;k++)

g += a[i-1][k-1]\*a[k-1][j-1];

for (k=1;k<=l;k++)

a[k-1][j-1] -= g\*a[k-1][i-1];

}

}

d[i-1]=a[i-1][i-1];

a[i-1][i-1]=1.0;

for (j=1;j<=l;j++) a[j-1][i-1]=a[i-1][j-1]=0.0;

}

return a;

}

public static double pythag(double a, double b)

{

//this method is used by QL method double absa,absb; absa=Math.abs(a);

absb=Math.abs(b);

if (absa absb) return absa\*Math.sqrt(1.0+(absb/absa)\*(absb/absa));

else return (absb==0.0 ? 0.0 : absb\*Math.sqrt(1.0+(absa/absb)\*(absa/absb)));

}

public static double[][] QL(double d[], double e[], double a[][])

{

// QL algorithm : eigenvalues of a symmetric matrix reduced to tridiagonal

// form by using method tridiagonal int n=d.length;

int m,l,iter,i,j,k;

double s,r,p,g,f,dd,c,b;

for (i=2;i<=n;i++) e[i-2]=e[i-1]; e[n-1]=0.0;

double z[][]=new double[n][n]; for(i=0;i<n;i++) for(j=0;j<n;j++)

z[i][j]=a[i][j]; for (l=1;l<=n;l++) { iter=0;

do {

for (m=l;m<=n-1;m++) { dd=Math.abs(d[m-1])+Math.abs(d[m]);

if ((double)(Math.abs(e[m-1])+dd) == dd) break;

}

if (m != l) {

if (iter++ == 30) System.out.println("Too many iterations in QL"); g=(d[l]-d[l-1])/(2.0\*e[l-1]);

r=Matrix.pythag(g,1.0);

g=d[m-1]-d[l-1]+e[l-1]/(g+Matrix.SIGN(r,g)); s=c=1.0;

p=0.0;

for (i=m-1;i=l;i--) { f=s\*e[i-1];

b=c\*e[i-1]; e[i]=(r=Matrix.pythag(f,g)); if (r == 0.0) {

d[i] -= p;

e[m-1]=0.0;

break;

}

s=f/r; c=g/r; g=d[i ]-p;

r=(d[i-1]-g)\*s+2.0\*c\*b; d[i ]=g+(p=s\*r);

g=c\*r-b;

for (k=1;k<=n;k++) { f=z[k-1][i ];

z[k-1][i ]=s\*z[k-1][i-1]+c\*f;

z[k-1][i-1]=c\*z[k-1][i-1]-s\*f;

}

}

if (r == 0.0 && i = l) continue; d[l-1] -= p;

e[l-1]=g;

e[m-1]=0.0;

}

} while (m != l);

}

return z;

}

public static double[][] eigenQL(double a[][])

{

// QL algoritm to solve eigen value problems

// symmetric matrices only (real eigen values)

// first column of the matrix returns eigen values

// second..n+1 column returns eigen vectors.

// Note : If matrix is not symmetric DO NOT use

// this method use eigenValue method (a QR algorithm) int i,j;

int n=a.length;

double sum[]=new double[n];; double d[]=new double[n]; double b[][]=new double[n][n]; double e[]=new double[n]; double z[][]=new double[n+1][n]; b=tridiagonal(a,d,e); b=QL(d,e,b);

for(j=0;j<n;j++)

{

z[0][j]=d[j]; for(i=0;i<n;i++)

{

z[i+1][j]=b[i][j]/b[0][j];

if(z[i+1][j]<1e-13) z[i+1][i]=0;

}

}

return z;

}

//end of eigen value programs

//end of class Matrix

}

### MATRIX SINIFI ÖRNEK PROGRAMLARI

**Inverse matris :** Inverse matrisi hesaplar. Buradaki temel hesaplama yolu tam pivotlu gauss eliminasyon metodudur. Bu metod göreceli olarak küçük matrisler içindir. dev boyutlardaki matrisler için tavsiye edilmez. Çok büyük boyutlarda iterativ metodlar daha az sayida islemle çözüme gidebilir. Bu metod hassas sonuç verebilme özelliginden dolayi seçilmistir. ayni zamanda LU (alt ve üst üçgen matris) parçalama metoduyla da çözebiliriz. Bu metodun avantaji lineer bir denklem sistemi çözerken eger ikinci taraf sürekli degisiyorsa ortaya çikar. tek bir çözüm için ilave bir avantaj getirmez.

### Problem 12.3 Matrix3.java, inverse matris problemi örnegi

import java.io.\*; import Numeric; class Matrix3

{

public static void main(String args[]) throws IOException

{

/\*

double a[][];

a=new double[5][5]; a[0][0]=1;

a[0][1]=2;

a[0][2]=0;

a[0][3]=0;

a[0][4]=0;

a[1][0]=-2;

a[1][1]=3;

a[1][2]=0;

a[1][3]=0;

a[1][4]=0;

a[2][0]=3;

a[2][1]=4;

a[2][2]=50;

a[2][3]=0;

a[2][4]=0;

a[3][0]=-4;

a[3][1]=5;

a[3][2]=-60;

a[3][3]=7;

a[3][4]=0;

a[4][0]=-5;

a[4][1]=6;

a[4][2]=-70;

a[4][3]=8;

a[4][4]=-9;

\*/

double c[][]=new double[5][5];

double a[][]={{1,2,0,0,0},{-2,3,0,0,0},{3,4,50,0,0},{-4,5,-60,7,0},

{-5,6,-70,8,-9}};

double b[]=new double[5]; int d[]=new int[1];

b[0]=1;

b[1]=0;

b[2]=0;

b[3]=0;

b[4]=0;

System.out.println("Original Matrix : "); System.out.println(Matrix.toString(a)); System.out.println("Inverse Matrix : (Method inv) "); System.out.println(Matrix.toString(Matrix.inv(a)));

System.out.println("Matrix \* Inverse Matrix : (Method multiply) "); System.out.println(Matrix.toString(Matrix.multiply(a,Matrix.inv(a)))); System.out.println("Solution of system of equation : "); System.out.println("with second side : "); System.out.println(Matrix.toStringT(b)); System.out.println(Matrix.toString(Matrix.divide(b,a)));

int indx[]=new int[5]; d[0]=0;

c=Matrix.LU(a,indx,d);

System.out.println("LU decomposed matrix : (Method LU) "); System.out.println(Matrix.toString(c));

System.out.println("Matrix inversion by LU decomposition : (Method LUinv)"); System.out.println(Matrix.toString(Matrix.LUinv(a))); System.out.println("Matrix \* Inverse Matrix (LU Decomposition) :"); System.out.println(Matrix.toString(Matrix.multiply(a,Matrix.LUinv(a))));

}

}

### inverse matrix metotlari sonucu :

**Original Matrix :**

1.0 2.0 0.0 0.0 0.0

-2.0 3.0 0.0 0.0 0.0

3.0 4.0 50.0 0.0 0.0

-4.0 5.0 -60.0 7.0 0.0

-5.0 6.0 -70.0 8.0 -9.0

### Inverse Matrix : (Method inv)

0.4285714285714286 -0.2857142857142857 0.0 -1.1895246692412392E-17 3.96508223080413E-18

0.2857142857142857 0.14285714285714285 0.0 5.947623346206196E-18-1.9825411154020644E-18

-0.048571428571428564 0.005714285714285717 0.019999999999999997-3.8659551750340293E-19 1.898283117997477E-18

-0.37551020408163255-0.21632653061224483 0.1714285714285714 0.14285714285714282 2.2431036619977644E-17

-0.003628117913832162 0.01723356009070296-0.0031746031746031824 0.12698412698412698-

0.11111111111111112

### Matrix \* Inverse Matrix : (Method multiply)

1.0 0.0 0.0 0.0 1.5407439555097887E-33

-1.1102230246251565E-16 1.0 0.0 4.163336342344337E-17-1.3877787807814454E-17 4.440892098500626E-16 1.6653345369377348E-16 0.9999999999999999-3.122502256758254E-17

9.887923813067798E-17

0.0 2.220446049250313E-16 0.0 0.9999999999999999 1.7347234759768022E-17

-2.8449465006019636E-16 1.942890293094024E-16 1.734723475976807E-16 0.0 1.0

### Solution of system of equation : with second side :

1.0

0.0

0.0

0.0

0.0

0.4285714285714286

0.2857142857142857

-0.048571428571428564

-0.37551020408163255

-0.003628117913832162

### LU decomposed matrix : (Method LU)

-2.0 3.0 0.0 0.0 0.0

-0.5 3.5 0.0 0.0 0.0

2.5 -0.42857142857142855-70.0 8.0 -9.0

-1.5 2.4285714285714284-0.7142857142857143 5.714285714285714-6.428571428571429

2.0 -0.2857142857142857 0.8571428571428571 0.025000000000000064 7.875

### Matrix inversion by LU decomposition : (Method LUinv)

0.42857142857142855-0.2857142857142857 -0.0 -0.0 -0.0

0.2857142857142857 0.14285714285714285 0.0 0.0 0.0

-0.04857142857142856 0.005714285714285712 0.02 -0.0 -0.0

-0.37551020408163255-0.21632653061224494 0.17142857142857143 0.14285714285714285 0.0

-0.0036281179138321806 0.017233560090702916-0.003174603174603183 0.12698412698412698-

0.1111111111111111

### Matrix \* Inverse Matrix (LU Decomposition) :

1.0 0.0 0.0 0.0 0.0

0.0 1.0 0.0 0.0 0.0

4.440892098500626E-16-1.1102230246251565E-16 1.0 0.0 0.0

-4.440892098500626E-16-4.440892098500626E-16 0.0 1.0 0.0

-1.1796119636642288E-16-8.326672684688674E-17-4.5102810375396984E-17 0.0 1.0

### Problem 12.4 Matrix5.java, inverse matris problemi örnegi

import java.io.\*; import Numeric;

class Matrix5

{

public static void main(String args[]) throws IOException

{

//calculation of an inverse matrix

double a[][];

a=new double[2][2]; a[0][0]=1;

a[0][1]=2;

a[1][0]=3;

a[1][1]=4;

System.out.println("Matrix : "); System.out.println(Matrix.toString(a)); System.out.println("Inverse Matrix : "); System.out.println(Matrix.toString(Matrix.inv(a))); System.out.println("Matrix \* Inverse Matrix :");

System.out.println(Matrix.toString(Matrix.multiply(a,Matrix.inv(a)))); System.out.println("Determinant = "+Matrix.det(a));

}

}

Matrix :

1.0 2.0

3.0 4.0

Inverse Matrix :

-2.0 1.0

1.5 -0.5

Matrix \* Inverse Matrix : 1.0 0.0

0.0 1.0

Determinant = -2.0

**EigenValue - Sinir deger problemi :** matislerle ilgili belki de en ilginç proses eigen value - sinir deger problemidir. Sinir deger hesaplamalari iteratif bir problemdir. Simetrik matrislerde sonuçlar gerçek sayi olarak çikar. bu tür uygulamalarda QL metodu kullanilabilir. Simetrik olmayan matrislerin sinir degerleri kompleks degerlerdir (kompleks ve gerçek degerlerin karisimi olabilir.) burada temel olarak QR formülü kullanilmistir. Eigen value problemleriyle direk olarak ilgili ilginç bir problem de ninci dereceden bir polinomun köklerinin hesaplanmasi problemidir. polinom karakteristik matris ismi verilen özel bir matrise dönüstürülebilir. Bu matrisin eigen-value degerleri polinomun da kökleridir.

Karekterisitik matrisin tanimi

public static double[][] characteristic\_matrix(double c[])

{

int n=c.length-1; int i;

double a[][]=new double[n][n]; for(i=0;i<n;i++)

{ a[0][i]=-c[i+1]/c[0]; }

for(i=0;i<n-1;i++)

{ a[i+1][i]=1; }

return a; }

metodu ile verilmistir.

**Problem 12.5** Matrix1.java , **eigenvalue problemi**

import java.io.\*; import Numeric;

class Matrix1

{

public static void main(String args[]) throws IOException

{

//Eigen Value calculations double c[][];

c=new double[5][5];

double a[][]={{1,2,0,0,0}, {-2,3,0,0,0}, {3,4,50,0,0},

{-4,5,-60,7,0},{-5,6,-70,8,-9}};

System.out.println("Matris : "); System.out.println(Matrix.toString(a)); complex c1[];

int n=a.length; c1=new complex[n];

c1=Matrix.eigenValueC(a); System.out.println("Kompleks Eigenvalue vektörü : "); System.out.println(Matrix.toStringT(c1));

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| } |  | | | |
| Matris : |
| 1.0 | 2.0 | 0.0 | 0.0 | 0.0 |
| -2.0 | 3.0 | 0.0 | 0.0 | 0.0 |
| 3.0 | 4.0 | 50.0 | 0.0 | 0.0 |
| -4.0 | 5.0 | -60.0 | 7.0 | 0.0 |
| -5.0 | 6.0 | -70.0 | 8.0 | -9.0 |

Kompleks Eigenvalue vectörü : 50.0

(2.000000000000002 - 1.732050807568873i ) (2.000000000000002 + 1.732050807568873i ) 7.000000000000001

-8.999999999999998

### Problem 12.6 Matrix4.java , ninci dereceden denklemin kökleri ve eigenvalue problemleri

import java.io.\*; import Numeric;

class Matrix4

{

public static void main(String args[]) throws IOException

{

//Eigen Value calculations using QL method double p[]={-1, 8, -19, 12};

double a[][]=Matrix.characteristic\_matrix(p); System.out.println("Orijinal Polinom : "); System.out.println(Matrix.toString(p)); System.out.println("Karekteristik matris : "); System.out.println(Matrix.toString(a)); System.out.println("Eigen Value (sinir deger) : "); System.out.println(Matrix.toString(Matrix.eigenValue(a)));

System.out.println("Eigen Value (sinir deger) balancedeigenValue metodu : "); System.out.println(Matrix.toString(Matrix.balancedEigenValue(a))); System.out.println("Polinomun kokleri : (poly\_roots) "); System.out.println(Matrix.toString(Matrix.poly\_roots(p))); System.out.println("Polinomun kokleri : (poly\_rootsC) "); System.out.println(Matrix.toString(Matrix.poly\_rootsC(p)));

}

}

### cözüm :

Orijinal Polinom :

-1.0 8.0 -19.0 12.0

Karekteristik matris :

|  |  |  |
| --- | --- | --- |
| 8.0 | -19.0 | 12.0 |
| 1.0 | 0.0 | 0.0 |
| 0.0 | 1.0 | 0.0 |

Eigen Value (sinir deger) :

3.9999999999999964 3.0000000000000067 0.9999999999999991

0.0 0.0 0.0

Eigen Value (sinir deger) balancedeigenValue metodu : 4.000000000000007 2.999999999999993 1.0000000000000004

0.0 0.0 0.0

Polinomun kokleri : (poly\_roots)

4.000000000000007 2.999999999999993 1.0000000000000004

0.0 0.0 0.0

Polinomun kokleri : (poly\_rootsC)

4.000000000000007 2.999999999999993 1.0000000000000004

### Determinant ,

**Problem 12.5** Matrix5.java , **matris çarpim, inverse ve determinant islemleri**

import java.io.\*; import Numeric;

class Matrix5

{

public static void main(String args[]) throws IOException

{

//calculation of an inverse matrix double a[][];

a=new double[2][2]; a[0][0]=1;

a[0][1]=2;

a[1][0]=3;

a[1][1]=4;

System.out.println("Matrix : "); System.out.println(Matrix.toString(a)); System.out.println("Inverse Matrix : "); System.out.println(Matrix.toString(Matrix.inv(a))); System.out.println("Matrix \* Inverse Matrix :");

System.out.println(Matrix.toString(Matrix.multiply(a,Matrix.inv(a)))); System.out.println("Determinant = "+Matrix.det(a));

}

}

### cözümü :

Matrix : 1.0 2.0

3.0 4.0

Inverse Matrix :

-2.0 1.0

1.5 -0.5

Matrix \* Inverse Matrix :

1.0 0.0

0.0 1.0

Determinant = -2.0

**Eigenvalue, EigenVektör :** Burada verilen matris simetrik bir matristir. sinir deger vektörü QL metodu kullanilarak hesaplanmistir. Simetrik matris oldugundan sinir degerler gerçek degerler olarak elde edilmistir.

### Problem 12.5 Matrix6.java , eigenvalue (sinir-deger) problemi

import java.io.\*; import Numeric; class Matrix6

{

public static void main(String args[]) throws IOException

{

//Eigen Value calculations using QL method double a[][]={{3,-1,0},{-1,2,-1},{-0,-1,3}};

System.out.println("Original Matrix : "); System.out.println(Matrix.toString(a)); System.out.println("eigen values and Eigen Vectors : "); System.out.println(Matrix.toString(Matrix.eigenQL(a)));

}

}

### çözümü :

Original Matrix : 3.0 -1.0 0.0

-1.0 2.0 -1.0

0.0 -1.0 3.0

eigen values and Eigen Vectors :

4.0 2.9999999999999996 1.0000000000000004

1.0 1.0 1.0

-1.0000000000000004 0.0 2.0000000000000004

1.0000000000000007-0.9999999999999997 1.0000000000000004

### 12.3 NUMERIC SINIFI (SAYISAL ANALIZ PAKETI)

Numeric sinifi genel bir sayisal analiz paketi olarak düsünülmüstür. Bu pakette benim kisisel ihtiyaçlarim için yazdigim yada baska dillerde daha önceden yazilmis olan kodlardan adepte ettigim bazi programlar verilmistir. Bir paketin genel olarak herhangi bir fonksiyonla kullanilabilmesi için herhangi bir fonksiyonla birlikte kullanilmasi gerekir. Bu yüzden programlarda fonksiyonlar tanimlanmis abstract siniflar içinde yer almistir.

Gerçek fonksiyonlar yazilirken abstract siniflardan türetilmeleri gerekmektedir. Numeric paketindeki ve abstract siniflar ve metodlar su sekilde tanimlanmistir :

### Numerik sinifi ve fonksiyon girdi absract siniflari ve metotlari : abstract class f\_x

tek degiskenli tek fonksiyonlari aktarmak için kullanilan abstract siniftir.

Bu sinifla kullanilabilecek fonksiyona örnek olarak **f=x\*x** fonksiyonu verilebilir. Buradaki abstract fonksiyonun tanimi :

abstract double func(double x); seklindedir. Örnek verecek olursak :

### class f1 extends f\_x

{ double func(double x) { return x\*x; } }

Bu sinif dinamik olarak tanimlandigindan kullanilacaginda tanimlanmasi gerekir.

double y=Numeric.derivative(new f1(),x); veya

f1 xkare=new f1();

double y=Numeric.derivative(xkare,x); seklinde çagirilir.

### abstract class f\_xi

birden fazla fonksiyon birden fazla degisken ile kullanilacaksa f\_xi abstract sinifi kullanilir. Bu sinifta tanimlanan func metodu denklem\_referansi indisinin degerine göre tek bir fonksiyon girdisi verir.

örnek

f[0] = x[0]+sin(x[1]);

f[1] = x[0]\*x[0]-x[1] ;

func(x,1) f[1]degerini verir func(x,0) f[0]degerini verir abstract metodun tanimi :

abstract double func(double x[],int equation\_ref); seklindedir.

Örnek problemin java kodu esdegeri : class f2 extends f\_xi

{ double func(double x[],int x\_ref) { double a=0;

switch(x\_ref) { {case 0: a = x[0]+sin(x[1]); break;}

{case 1: a = x[0]\*x[0]-x[1]; break;}}

return a; } }

seklinde verilebilir. Bu funksiyonun asil programda çagirilmasi : double x[]={1,2};

double y=Numeric.derivative(new f2(),x,0); veya

double x[]={1,2}; f2 xkare=new f2();

double y=Numeric.derivative(xkare,x,0); seklinde gerçeklestirilir.

### abstract class fi\_xi

birden fazla fonksiyon birden fazla degisken ile kullanilacaksa fi\_xi abstract sinifi kullanilir. Bu sinifta tanimlanan func metodu tüm fonksiyon degerlerini vektör olarak iletir.

örnek f[0]=x[0]+sin(x[1]);

f[1]=x[0]\*x[0]-x[1];

func(x) f[0] ve f[1] degerlerini verir. abstract double[] func(double x[]);

Örnek problemin java kodu esdegeri : class f3 extends fi\_xi

{ double[] func(double x[])

{ double a[]=new double[2]; a[0] = x[0]+sin(x[1]);

a[1] = x[0]\*x[0]-x[1];

return a; }

}

seklinde verilebilir. Bu funksiyonun asil programda çagirilmasi : double x[]={1,2};

double y[]=Numeric.derivative(new f3(),x); veya

double x[]={1,2}; f3 fx=new f3();

double y[]=Numeric.derivative(fx,x); seklinde gerçeklestirilir.

### abstract class fij\_xi

{

iki boyutlu fonksiyon matrisi ve bir boyutlu degisken vectörü verilmisse fij\_xi sinifini kullanabiliriz. Bu tür

fonksiyonlar özellikle iki boyutlu fonksiyon setinin türev fonksiyonlarini hesaplarken olusabilir. Çikti matris olarak tüm fonksiyonlari verir.

### örnek :

f[0][0]=x[0]+Math.sin(x[1]) f[0][1]=x[0]-x[1]

f[1][0]=x[0]\*x[0]-x[1] f[1][1]=Math.exp(x[0]+x[1]\*x[1]

f[0][0], f[0][1]

f[1][0], f[1][1]

degerlerin iki boyutlu matrix olarak verir.

abstract double[][] func(double x[]); örnegin java kodu esdegeri :

class f4 extends fij\_xi

{

double[][] func(double x[])

{

double b[][];

f=new double[2][2];

f[0][0]= x[0]+Math.sin(x[1]);

f[0][1]= x[0]-x[1];

f[1][0]= x[0]\*x[0]-x[1];

f[1][1]= Math.exp(x[0]+x[1]\*x[1]; return f;

}

}

metod ve sinifin java programinda kullanilisi :

double x[]={1,2};

double y[][]=Numeric.derivative(new f4(),x); veya

double x[]={1,2}; f4 fx=new f4();

double y[][]=Numeric.derivative(fx,x);

### public class Numeric

Numeric sinifindaki metodlar genelde statik metodlardir. Bu yüzden bagimsiz olarak kullanilabilirler. Simdi bu metodlari daha detayli inceleyelim :

**Türev :** bu metotlar gurubu yukarida tanimlanan double sayi, vektor ve matris olarak verilen bir fonksiyonun türevlerini alir.

### public static double derivative(f\_x f\_deriv,double x) df/dx

tek fonksiyon, tek degisken türevi, örnek fonksiyon f=x\*x fonksiyon abstract sinifi : f\_x

türevin alinacagi nokta : double x

### public static double derivative(f\_xi f\_deriv,int equation\_ref, double x[],int x\_ref)

**fi/xj**

birden fazla fonksiyon birden fazla degisken türevi

denklem equation\_ref indisinin ve x\_ref x indisinindegerine göre sadece tanimlanan fonksiyonun tanimlanan x\_ref indisine göre olan türevini verir

örnek fonksiyon

f[0] = x[0]+sin(x[1]);

f[1] = x[0]\*x[0]-x[1] ;

fonksiyon abstract sinifi : f\_xi

yukaridaki örnek için türevin alinacagi nokta : double x[] ile belirlenmistir

public static double derivative(fi\_xi f\_deriv, int equation\_ref,double x[],int x\_ref)

### public static double derivative(fi\_xi f\_deriv, int equation\_ref,double x[],int x\_ref)

**fi/xj**

birden fazla fonksiyon birden fazla degisken türevi

denklem equation\_ref indisinin ve x\_ref x indisinindegerine göre sadece tanimlanan fonksiyonun tanimlanan x\_ref indisine göre olan türevini verir

örnek fonksiyon

f[0] = x[0]+sin(x[1]);

f[1] = x[0]\*x[0]-x[1] ;

fonksiyon abstract sinifi : f\_xi

yukaridaki örnek için türevin alinacagi nokta : double x[] ile belirlenmistir

### not1 : bir üsteki türev formülüyle bu türev formülünün temel farki, burada tanimlanan fonksiyonun tüm vektörü çikti olarak vermesidir. sonuçlar her ikisinde de aynidir.

**not2: denklemlerin hassasiyeti metotlarin içerisinde verilen h0 ve n degiskenlerinin degistirilmesiyle arttirilabilir. Su andaki degerleri optimize edilmistir ve denenen tüm fonksiyonlar için hassas degerler vermistir. Nümerik türevlerle çözülmesi oldukça zor olan boyutlu newton-raphson tekniginde bile basarili sonuçlar vermistir. Daha küçük degerler kullanirken önce türevini bildiginiz fonksiyonlarla sonuçlari kontrol etmeniz tavsiye edilir.**

**Integral:** Sayisal olarak integral almak için çok çesitli integrasyon metotlari bulunmaktadir. Burada verilen metodlarda ana kriteria sonuçlarin hassasligidir. Denklemler nümerik olarak dogru sonuç çikarmasi oldukça zor olan siyah cisim radyasyon fonksiyonu ve istatistik fonksiyonlarda bile tam hassaslik vermistir. integral 30 noktali Gauss - Legendre integral formülü kullanmaktadir. trap, trapezoidal integral formülü romberg integrasyonunun temelini teskil etmek için hazirlanmistir, tekbasina da kullanilabilir. Çok hassasiyet gereken uygulamalariniz için kullanmanizi tavsiye etmiyoruz. Romberg integrasyonu iterativ bir integrasyon teknigidir. Trapezoidal integrasyonla hesaplanan kök degerlerini alarak iterasyonla hata miktarini azaltir.

### public static double integral (f\_x f\_xnt,double a,double b)

f(x) fonksiyonunun double a ve double b sinirlari arasinda integrali, Gauss-Legendre metodu örnek fonksiyon f=x\*x

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.integral(new fx(),a,b);

**public static void trap(f\_x ff,double a,double h, int j,int m,double R[][])** f(x) fonksiyonunun a ve b sinirlari arasinda integrali, trapezoidal metodu örnek fonksiyon f=x\*x

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.trap(new fx(),a,h,j,m,R);

### public static double integral\_romberg(f\_x ff,double a,double b)

f(x) fonksiyonunun a ve b sinirlari arasinda integrali, Romberg integrasyon metodu örnek fonksiyon f=x\*x

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.integral\_romberg(new fx(),a,b);

**Non-linear denklem kökleri :** non linear denklem veya denklem sistemlerinin çözümü nümerik analizde en çok karsimiza çikan problemlerden biridir. Genelde tüm metodlar iteratif yöntemler kullanirlar. Bu metodlar içinde en verimli olarak hesap yapan metodun newton metodu oldugu söylenebilir, fakat newton metodu fonksiyonun türevinin de hesaplanmasini gerekli kilar. buradaki metodlarin bir kisminda türev girdi olarak alinirken, bir kisminda da yine nümerik metodlarla hesaplanmistir. denklemin ikinci türevini de göz önüne alan newton\_2nd\_derivative gibi metodlar da verilmistir. ayrica secant, bisection gibi türev hesabi gerektirmeyen metodlar da mevcuttur. Bisection metodu temelde bir arama metodudur ve verimi en düsük metoddur, fakat verilen bölgeyi tam olarak taradigi ve bir kök varsa muhakkak bulabildigi için verimsiz bir metod için oldukça sik tercih gören bir yöntemdir.

### public static double newton (double x,f\_x y,f\_x dy)

y(x) fonksiyonunun kökleri, dy(x)/dx fonksiyonu da verilmektedir. örnek fonksiyon y=x\*x-2

dy=2\*x fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.newton(x,new fx(),new dfx());

public static double newton(double x,f\_x y)

y(x) fonksiyonunun kökleri, dy(x)/dx fonksiyonu nümerik olarak hesaplanmaktadir örnek fonksiyon y=x\*x-2

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.newton(x,new fx());

public static double newton\_2nd\_derivative(double x,f\_x y,f\_x dy)

y(x) fonksiyonunun kökleri, 2inci türev formülü, dy(x)/dx fonksiyonu da verilmektedir. örnek fonksiyon y=x\*x-2

dy=2\*x fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.newton\_2nd \_derivative (x,new fx(),new dfx(),new dfx());

public static double newton\_2nd\_derivative(double x,f\_x y)

y(x) fonksiyonunun kökleri, 2nci türev formülü, dy(x)/dx fonksiyonu nümerik olarak hesaplanmaktadir örnek fonksiyon y=x\*x-2

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.newton\_2nd \_derivative (x,new fx(),new dfx());

public static double secant\_2nd\_derivative(double x,f\_x y)

y(x) fonksiyonunun kökleri, 2nci türev formülü, secand metodu kullanilmistir (türev formülü gerekmez) örnek fonksiyon y=x\*x-2

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.secant\_2nd \_derivative (x,new fx(),new dfx());

public static double bisection(double xl,double xu,f\_x y)

y(x) fonksiyonunun kökleri, 2nci türev formülü, bisection metodu kullanilmistir (türev formülü gerekmez, temel olarak bir arama metodudur.)

örnek fonksiyon y=x\*x-2

fonksiyon abstract sinifi : f\_x

örnek çagirma : double y=Numeric.bisection(xl,xu,new fx());

public static double[] newton( double x[],fi\_xi y,fij\_xi dy)

yi(xi) fonksiyon sisteminin kökleri kökleri, dyi(x)/dxj fonksiyonu da matrix fonksiyonu olarak verilmektedir. örnek fonksiyon

y1=2\*x1\*x1\*x2-3\*x1 y2=2\*x1+2\*x2

örnek türev fonksiyonu

dy1/dx1=4\*x1\*x2-3 dy1/dx2=2\*x1 dy2/dx1=2 dy2/dx2=2

fonksiyon abstract sinifi : fi\_xi

türev fonksiyonu abstract sinifi . fi\_xij

örnek çagirma : double y[]=Numeric.newton(x,new fixi(),new dfixij());

public static double[] newton( double x[],fij\_xi y)

yi(xi) fonksiyon sisteminin kökleri kökleri, dyi(x)/dxj fonksiyonu nümerik olarak hesaplanmaktadir, girdi degildir.

örnek fonksiyon

y1=2\*x1\*x1\*x2-3\*x1 y2=2\*x1+2\*x2

fonksiyon abstract sinifi : fi\_xi

örnek çagirma : double y[]=Numeric.newton(x,new fixi());

### Least Square (en küçük kareler metodu) polinom uydurma

**public static double[] poly\_least\_square (double xi[],double yi[],int n)**

polynom least square (en küçük kareler metodu) polinom uydurma, çikti olarak polinomun katsayilarini

verir

örnek çagirma : double y[]=Numeric. poly\_least\_square(xi,yi,n);

### public static double f\_least\_square(double e[],double x)

polynom least square (en küçük kareler metodu) polinom uydurma, polinomun katsayilarini kullanarak fonksiyon degerini hesaplar

örnek çagirma : double y=Numeric. f\_least\_square (e,x);

### public static double error(double x[],double y[],double e[])

polynom least square (en küçük kareler metodu) polinom uydurma, polinomun katsayilarini kullanarak fonksiyon degerini hesaplar ve toplam hata miktarini bulur.

örnek çagirma : double y=Numeric. error(x,y,e);

### public static double[][] orthogonal\_polynomial\_least\_square (double xi[],double fi[],int m)

ortagonal polynom least square (en küçük kareler metodu) polinom uydurma, çikti olarak polinomun katsayilarini verir. hesaplama yöntemi matrislere dayanmadigindan matris çözümlerine dayanan hatalari yapmaz. yalniz polinom formülü daha kompleksdir

örnek çagirma : double y[]=Numeric. Orthogonal\_polynomial\_least\_square(xi,yi,n);

### public static double f\_orthogonal\_least\_square(double e[],double x)

ortogonal polynom least square (en küçük kareler metodu) polinom uydurma, polinomun katsayilarini kullanarak fonksiyon degerini hesaplar

örnek çagirma : double y=Numeric. f\_orthogonal\_least\_square (e,x);

### Diferansiyel denklemler

**public static double[] RK4(f\_xi fp,double x0,double xn,double f0,int N)**

4üncü dereceden Runge-Kutta diferansiyal denklem çözüm metodu, tek bir denklemi çözmek için örnek çagirma : double y[]=RK4 (new fxi(),x0,xn,N);

### public static double[][] RK4(f\_xi fp,double x0,double xn,double f0[],int N)

4üncü dereceden Runge-Kutta diferansiyal denklem sistemi çözüm metodu, bir denklemin sistemini veya daha yüksek dereceden diferansiyel denklemleri çözmek için kulanilir.

örnek çagirma : double y[][]=RK4 (new fxi(),x0,xn,f0,N);

### public static double[][] RKF45(f\_xi fp,double x0,double xn,double f0,int N) throws IOException

4-5 incü dereceden Runge-Kutta-fehlenberg diferansiyal denklem sistemi çözüm metodu, bir denklemin sistemini veya daha yüksek dereceden diferansiyel denklemleri çözmek için kulanilir.

örnek çagirma : double y[][]=RKF45 (new fxi(),x0,xn,f0,N);

**Problem 12.6 Numeric sinifi listesi :** Numeric.java

//=============================================================

// Numerical Analysis Package in Java

// Numerical analysis calculation class Numeric

// Author : Dr. Turhan Coban

//=============================================================

import java.io.\*; import Matrix;

### abstract class f\_x

{

//single function single independent variable

// example f=x\*x

abstract double func(double x);

}

### abstract class f\_xi

{

// multifunction multi independent variable

// a single value is returned indiced to equation\_ref

// example f[0]=x[0]+sin(x[1])

// f[1]=x[0]\*x[0]-x[1]

// func(x,1) returns the value of f[1]

// func(x,0) returns the value of f[0]

abstract double func(double x[],int equation\_ref);

}

### abstract class fi\_xi

{

// multifunction multi independent variable

// vector of dependent variables are returned

// example f[0]=x[0]+sin(x[1])

// f[1]=x[0]\*x[0]-x[1]

// func(x) returns the value of f[0] and f[1]

// as a two dimensional vector abstract double[] func(double x[]);

}

### abstract class fij\_xi

{

// multifunction multi independent variable

// for n independent variable n\*n matrix of

// dependent variables are returned

// example

// f[0][0]=x[0]+Math.sin(x[1]) f[0][1]=x[0]-x[1]

// f[1][0]=x[0]\*x[0]-x[1] f[1][1]=Math.exp(x[0]+x[1]\*x[1]

// func(x) returns the value of f[0][0], f[0][1]

// f[1][0], f[1][1]

// as a two dimensional matrix

abstract double[][] func(double x[]);

}

### public class Numeric

{

//This is a library of numerical methods

//

### public static double derivative(f\_x f\_deriv,double x)

{

// This method calculates derivatives of a simple function

// accuracy of method can be adjusted by changing

// variables h0 and n

// function input should be in the form given in abstract class

// f\_x

double h0=0.0256;

// accuracy can be change by adjusting h0 and n int i,m;

int n=7;

//derivative of a simple function double T[][];

T=new double[n][n]; double h[];

h=new double[n];

//vector<double h(n,0); for(i=0;i<n;i++)

{

h[i]=0;

for(int j=0;j<n;j++) T[i][j]=0;

}

h[0]=h0;

double r=0.5; for( i=1;i<n;i++)

{

h[i]=h0\*Math.pow(r,i);

}

for(i=0;i<n;i++)

{

T[i][0]=( f\_deriv.func(x + h[i]) - f\_deriv.func( x - h[i]))/(2.0\*h[i]);

}

for(m=1;m<n;m++)

{

for(i=0;i<n-m;i++)

{

T[i][m]=(h[i]\*h[i]\*T[i+1][m-1] - h[i+m]\*h[i+m]\*T[i][m-1])/ (h[i]\*h[i]- h[i+m]\*h[i+m]);

}

}

double xx=T[0][n-1]; return xx;

}

### public static double derivative(f\_xi f\_deriv,int equation\_ref, double x[],int x\_ref)

{

// This method calculates derivative of a function selected from a set of

// functions. Accuracy of method can be adjusted by changing

// variables h0 and n

// function input should be in the form given in abstract class

// f\_xi

// df(equation\_ref)/dx(x\_ref) double h0=0.0256;

int i,m; int n=7;

double x1[];

x1=new double[x.length]; double x2[];

x2=new double[x.length]; for(i=0;i<x.length;i++)

{

x1[i]=x[i];

x2[i]=x[i];

}

//derivative of a simple function double T[][];

T=new double[n][n]; double h[];

h=new double[n];

//vector<double h(n,0); for(i=0;i<n;i++)

{

h[i]=0;

for(int j=0;j<n;j++) T[i][j]=0;

}

h[0]=h0;

double r=0.5; for( i=1;i<n;i++)

{

h[i]=h0\*Math.pow(r,i);

}

for(i=0;i<n;i++)

{

x1[x\_ref]+=h[i];

x2[x\_ref]-=h[i];

T[i][0]=( f\_deriv.func(x1,equation\_ref) - f\_deriv.func(x2,equation\_ref))/(2.0\*h[i]);

x1[x\_ref]=x[x\_ref]; x2[x\_ref]=x[x\_ref];

}

for(m=1;m<n;m++)

{

for(i=0;i<n-m;i++)

{

T[i][m]=(h[i]\*h[i]\*T[i+1][m-1] - h[i+m]\*h[i+m]\*T[i][m-1])/(h[i]\*h[i] - h[i+m]\*h[i+m]);

}

}

double xx=T[0][n-1]; return xx;

}

### public static double derivative(fi\_xi f\_deriv, int equation\_ref,double x[],int x\_ref)

{

// This method calculates derivative of a function selected from a set of

// functions. Accuracy of method can be adjusted by changing

// variables h0 and n

// function input should be in the form given in abstract class

// f\_xi

double h0=0.0256; int i,m;

int n=7; double f1[];

f1=new double[x.length]; double f2[];

f2=new double[x.length]; double x1[];

x1=new double[x.length]; double x2[];

x2=new double[x.length]; for(i=0;i<x.length;i++)

{

x1[i]=x[i];

x2[i]=x[i];

}

//derivative of a simple function double T[][];

T=new double[n][n]; double h[];

h=new double[n];

//vector<double h(n,0); for(i=0;i<n;i++)

{

h[i]=0;

for(int j=0;j<n;j++) T[i][j]=0;

}

h[0]=h0;

double r=0.5; for( i=1;i<n;i++)

{

h[i]=h0\*Math.pow(r,i);

}

for(i=0;i<n;i++)

{

x1[x\_ref]+=h[i];

x2[x\_ref]-=h[i]; f1=f\_deriv.func(x1); f2=f\_deriv.func(x2);

T[i][0]=( f1[equation\_ref] - f2[equation\_ref])/(2.0\*h[i]); x1[x\_ref]=x[x\_ref];

x2[x\_ref]=x[x\_ref];

}

for(m=1;m<n;m++)

{

for(i=0;i<n-m;i++)

{

T[i][m]=(h[i]\*h[i]\*T[i+1][m-1] - h[i+m]\*h[i+m]\*T[i][m-1])/(h[i]\*h[i]

- h[i+m]\*h[i+m]);

}

}

double xx=T[0][n-1]; return xx;

}

### public static double integral(f\_x f\_xnt,double a,double b)

{

//integral of a function by using gauss-legendre quadrature

//

double s[],w[]; int i;

s=new double[30]; w=new double[30];

s[ 0] = .15532579626752470000E-02; s[ 1] = .81659383601264120000E-02; s[ 2] = .19989067515846230000E-01; s[ 3] = .36899976285362850000E-01; s[ 4] = .58719732103973630000E-01; s[ 5] = .85217118808615820000E-01; s[ 6] = .11611128394758690000E+00; s[ 7] = .15107475260334210000E+00; s[ 8] = .18973690850537860000E+00; s[ 9] = .23168792592899010000E+00; s[10] = .27648311523095540000E+00; s[11] = .32364763723456090000E+00; s[12] = .37268153691605510000E+00; s[13] = .42306504319570830000E+00; s[14] = .47426407872234120000E+00; s[15] = .52573592127765890000E+00; s[16] = .57693495680429170000E+00; s[17] = .62731846308394490000E+00; s[18] = .67635236276543910000E+00; s[19] = .72351688476904450000E+00; s[20] = .76831207407100990000E+00; s[21] = .81026309149462140000E+00; s[22] = .84892524739665800000E+00; s[23] = .88388871605241310000E+00; s[24] = .91478288119138420000E+00; s[25] = .94128026789602640000E+00; s[26] = .96310002371463720000E+00; s[27] = .98001093248415370000E+00; s[28] = .99183406163987350000E+00; s[29] = .99844674203732480000E+00; w[ 0] = .39840962480827790000E-02; w[ 1] = .92332341555455000000E-02; w[ 2] = .14392353941661670000E-01; w[ 3] = .19399596284813530000E-01; w[ 4] = .24201336415292590000E-01; w[ 5] = .28746578108808720000E-01; w[ 6] = .32987114941090080000E-01; w[ 7] = .36877987368852570000E-01; w[ 8] = .40377947614710090000E-01; w[ 9] = .43449893600541500000E-01; w[10] = .46061261118893050000E-01; w[11] = .48184368587322120000E-01; w[12] = .49796710293397640000E-01; w[13] = .50881194874202750000E-01; w[14] = .51426326446779420000E-01; w[15] = .51426326446779420000E-01; w[16] = .50881194874202750000E-01; w[17] = .49796710293397640000E-01; w[18] = .48184368587322120000E-01; w[19] = .46061261118893050000E-01; w[20] = .43449893600541500000E-01; w[21] = .40377947614710090000E-01; w[22] = .36877987368852570000E-01; w[23] = .32987114941090080000E-01; w[24] = .28746578108808720000E-01;

w[25] = .24201336415292590000E-01; w[26] = .19399596284813530000E-01; w[27] = .14392353941661670000E-01; w[28] = .92332341555455000000E-02; w[29] = .39840962480827790000E-02;

int n=30; double z=0; double x,y; for(i=0;i<n;i++)

{

x=(b+a)/2.0+(b-a)/2.0\*s[i]; y=f\_xnt.func(x);

z+=(b-a)/2\*w[i]\*y;

}

for(i=0;i<n;i++)

{

x=(b+a)/2.0+(b-a)/2.0\*(-s[i]); y=f\_xnt.func(x);

z+=(b-a)/2.0\*w[i]\*y;

}

return z;

}

### public static void trap(f\_x ff,double a,double h, int j,int m,double R[][])

{

// successive trapezoidal integration rule

// this program will be utilised in romberg integration double sum=0;

int p; for(p=1;p<=m;p++)

{ sum+=ff.func(a+h\*(2\*p-1)); }

R[j][0]=R[j-1][0]/2.0+h\*sum;

}

### public static double integral\_romberg(f\_x ff,double a,double b)

{

//romberg integration

int n=8;//increase n to increase accuracy double R[][];

R=new double[n+1][n+1]; int m=1;

double h=b-a; double close=1; double tol=1e-40; int j=0,k=0; double ret=0;

R[0][0]=h/2.0\*(ff.func(a)+ff.func(b)); do

{

j++; h/=2.0;

Numeric.trap(ff,a,h,j,m,R); m\*=2;

for(k=1;k<=j;k++)

{

R[j][k]=R[j][k-1]+(R[j][k-1]-R[j-1][k-1])/(Math.pow(4,k)-1.0);

close=Math.abs(R[j-1][j-1]-R[j][j]); ret=R[j][k];

}

}while(closetol && j<n);

return ret;

}

//==================================================

//Finding Non-linear Roots of Functions

### public static double newton(double x,f\_x y,f\_x dy)

{

// Newton - Raphson method for single equation

// with single variable

// required function and its derivative int nmax=500;

double tolerance=1.0e-15; double fx,dfx;

for(int i=0;i<nmax;i++)

{

fx=y.func(x); dfx=dy.func(x); x-=fx/dfx;

if(Math.abs(fx)<tolerance)

{

return x;

}

}

return x;

}

### public static double newton(double x,f\_x y)

{

// Newton - Raphson method for single equation

// required function only derivative is calculated

// numerically by using method derivative int nmax=500;

double tolerance=1.0e-15; double fx,dfx;

for(int i=0;i<nmax;i++)

{

fx=y.func(x); dfx=Numeric.derivative(y,x); x-=fx/dfx; if(Math.abs(fx)<tolerance)

{

return x;

}

}

return x;

}

### public static double newton\_2nd\_derivative(double x,f\_x y,f\_x dy)

{

// Newton - Raphson type method for single equation

// includes 2nd order derivative calculations

//function and first order derivative is required int nmax=500;

double dx=1e-15; double x1m;

double tolerance=1.0e-15; double fx,fxm,dfx,dfxm,d2fx; for(int i=0;i<nmax;i++)

{

fx=y.func(x); fxm=y.func(x-dx);

dfx=dy.func(x); dfxm=dy.func(x-dx);

d2fx=-6.0/dx/dx\*(fx-fxm)+2.0/dx\*(2.0\*dfx+dfxm); x-=(fx/dfx+.5\*fx\*fx/(dfx\*dfx\*dfx)\*d2fx); if(Math.abs(fx)<tolerance)

{

return x;

}

}

return x;

}

### public static double newton\_2nd\_derivative(double x,f\_x y)

{

// Newton - Raphson type method for single equation

// includes 2nd order derivative calculations

// function is required, first order derivative calculated

// numerically by derivative method. int nmax=500;

double dx=1e-3; double x1m;

double tolerance=1.0e-15; double fx,fxm,dfx,dfxm,d2fx; for(int i=0;i<nmax;i++)

{

fx=y.func(x); fxm=y.func(x-dx);

dfx=Numeric.derivative(y,x); dfxm=Numeric.derivative(y,(x-dx));

d2fx=-6.0/dx/dx\*(fx-fxm)+2.0/dx\*(2.0\*dfx+dfxm); x-=(fx/dfx+.5\*fx\*fx/(dfx\*dfx\*dfx)\*d2fx); if(Math.abs(fx)<tolerance)

{

return x;

}

}

return x;

}

### public static double secant\_2nd\_derivative(double x,f\_x y)

{

// Newton - Raphson type method for single equation

// includes 2nd order derivative calculations

// function should be supplied int nmax=500;

double dx=1.0e-3; double dx1=2.0e-3; double x1m;

double tolerance=1.0e-15;

double fx,fxm,fxm1,dfx,dfxm,d2fx; for(int i=0;i<nmax;i++)

{

fx=y.func(x); fxm=y.func(x-dx); fxm1=y.func(x-dx1); dfx=(fx-fxm)/dx; dfxm=(fx-fxm1)/dx1;

d2fx=-6.0/dx/dx\*(fx-fxm)+2.0/dx\*(2.0\*dfx+dfxm); x-=(fx/dfx+.5\*fx\*fx/(dfx\*dfx\*dfx)\*d2fx); if(Math.abs(fx)<tolerance)

{

return x;

}

}

return x;

}

### public static double bisection(double xl,double xu,f\_x y)

{

//bisection method to find roots of a function y.func(x)

//function should be supplied

// defination of variables :

// xl : lower guess

// xu : upper guess

// xr : root estimate

// es : stopping criterion

// ea :approximate error

// maxit : maximum iterations

// iter : number of iteration double test;

double xr=0; double es,ea; double fxl,fxr;

int maxit=500,iter=0; es=0.000001;

ea=1.1\*es; while((eaes)&&(iter<maxit))

{

xr=(xl+xu)/2.0; iter++; if((xl+xu)!=0)

{ ea=Math.abs((xu-xl)/(xl+xu))\*100;} fxl= y.func(xl);

fxr= y.func(xr); test= fxl\*fxr;

if(test==0.0) ea=0; else if(test<0.0) xu=xr; else

{

xl=xr;

}

} //end of while return xr;

}

### public static double[] newton( double x[],fi\_xi y,fij\_xi dy)

{

//solution of nonlinear system of equations

//by using newton raphson method.

//ti :weigth factor

//x independent value vector:

//y dependent function vector

//dy derivative of dependent function matrix double ti=1.0;

int i;

int nmax=500;

double tolerance=1.0e-30; int n=x.length;

double b[]; b=new double[n];

for(i=0;i<n;i++)

{ b[i]=1.0;

} i=0;

while( i++ < nmax && Matrix.abs(b) tolerance )

{

b=Matrix.multiply(Matrix.divide(y.func(x),dy.func(x)),-ti); x=Matrix.add(x,b);

}

if(i = nmax) System.out.println("warning maximum number"+ " of iterations results may not be valid");

return x;

}

### public static double[] newton( double x[],fi\_xi y)

{

//solution of nonlinear system of equations

//by using newton raphson method.

//this function does not require derivatives

//it is utilised method derivative to calculate derivatives double ti=1.0;

int i,ii,jj;

int nmax=500;

double tolerance=1.0e-15; int n=x.length;

double b[]; b=new double[n]; double dy[][];

dy=new double[n][n]; i=0;

for(i=0;i<n;i++)

{ b[i]=1.0;

}

while( i++ < nmax && Matrix.abs(b) tolerance )

{

for(ii=0;ii<n;ii++)

{

for(jj=0;jj<n;jj++)

{

dy[ii][jj]=Numeric.derivative(y,ii,x,jj);

}

}

b=Matrix.multiply(Matrix.divide(y.func(x),dy),-ti); x=Matrix.add(x,b);

}

if(i = nmax) System.out.println("warning maximum number of iterations"+ " results may not be valid" );

return x;

}

//========= least square curve fitting methods ==========

//-------- polynomial least square curve fitting --------

### public static double[] poly\_least\_square (double xi[],double yi[],int n)

//polynomial least square curve fitting

//variables xi,yi vector of data points

//degree of curve fitting

{

//xi vector of independent variable

//yi vector of dependent variable

//n : degree of curve fitting int l=xi.length;

int i,j,k;

int np1=n+1; double A[][];

A=new double[np1][np1]; double B[];

B=new double[np1]; double X[];

X=new double[np1]; for(i=0;i<n+1;i++)

{

for(j=0;j<n+1;j++)

{

if(i==0 && j==0) A[i][j]=l;

else

for(k=0;k<l;k++)

A[i][j] += Math.pow(xi[k],(i+j));

}

for(k=0;k<l;k++)

{

if(i==0) B[i]+= yi[k]; else

B[i] += Math.pow(xi[k],i)\*yi[k];

}

}

X=Matrix.divide(B,A);

//X=B/A; double max=0;

for(i=0;i<n+1;i++)

if(Math.abs(X[i]) max) max = Math.abs(X[i]); for(i=0;i<n+1;i++)

if((Math.abs(X[i]/max) 0) && (Math.abs(X[i]/max) < 1.0e-11)) X[i]=0; return X;

}

### public static double f\_least\_square(double e[],double x)

{

// this function calculates the value of

// least square curve fitting function int n=e.length;

double ff; if(n!=0.0)

{ ff=e[n-1];

for(int i=n-2;i=0;i--)

{ ff=ff\*x+e[i]; }

}

else

ff=0; return ff;

}

### public static double error(double x[],double y[],double e[])

{

//calculates absolute square root error of a least square approach double n=x.length;

int k;

double total=0; for(k=0;k<n;k++)

{

total+=(y[k]-f\_least\_square(e,x[k]))\*(y[k]-f\_least\_square(e,x[k]));

}

total=Math.sqrt(total); return total;

}

//-------End least square methods --------------------

//-------Orthogonal polynomial least square curve fitting ---

### public static double[][] orthogonal\_polynomial\_least\_square (double xi[],double fi[],int m)

//orthogonal polynomial least square curve fitting

//this method do not require any matrix solution

//variables xi,fi vector of data points wi weight functions

//m degree of curve fitting

{

int i,j,k;

int n=xi.length; int mp2=n+2; int mp1=n+1;

//matrix<double p(m+2,n,0); double p[][];

p=new double[mp2][n];

//vector<double gamma(m+1,0); double gamma[];

gamma=new double[mp1];

//vector<double beta(m+1,0); double beta[];

beta=new double[mp1];

//vector<double omega(m+1,0); double omega[];

omega=new double[mp1];

//vector<double alpha(m+1,0); double alpha[];

alpha=new double[mp1];

//vector<double b(m+1,0); double b[];

b=new double[mp1];

//vector<double wi(n,1.); double wi[];

wi=new double[n];

//matrix<double a(3,m+1); double a[][];

a=new double[3][mp1]; for(i=0;i<n;i++)

{

p[1][i]=1.0;

p[0][i]=0.0;

}

gamma[0]=0; for(i=0;i<n;i++)

{

gamma[0]+=wi[i];

}

beta[0]=0.0; for(j=0;j<m+1;j++)

{

omega[j]=0;

for(i=0;i<n;i++)

{

omega[j]+=wi[i]\*fi[i]\*p[j+1][i];

}

b[j]=omega[j]/gamma[j]; if( j != m)

{

alpha[j+1]=0; for(i=0;i<n;i++)

{

alpha[j+1]+=wi[i]\*xi[i]\*p[j+1][i]\*p[j+1][i]/gamma[j];

}

for(i=0;i<n;i++)

{

p[j+2][i]=(xi[i]-alpha[j+1])\*p[j+1][i]-beta[j]\*p[j][i];

}

gamma[j+1]=0; for(i=0;i<n;i++)

{

gamma[j+1]+=wi[i]\*p[j+2][i]\*p[j+2][i];

}

beta[j+1]=gamma[j+1]/gamma[j];

}

}//end of j for(j=0;j<m+1;j++)

{

a[0][j]=b[j];

a[1][j]=alpha[j];

a[2][j]=beta[j];

}

return a;

}

### public static double f\_orhogonal\_polynomial(double a[][],double x)

{

double yy=0; int k;

int m=a[0].length-1; int mp2=m+2; double q[];

q=new double[mp2];

//vector<double q(m+2,0.0); for(k=m-1;k=0;k--)

{

q[k]=a[0][k]+(x-a[1][k+1])\*q[k+1]-a[2][k+1]\*q[k+2];

yy=q[k];

}

return yy;

}

//minimization of a function

public static double amotry(double p[][], double y[], double psum[], int ndim, f\_xi ff, int ihi, double fac)

{

int j;

double fac1,fac2,ytry;

double ptry[]=new double[ndim]; fac1=(1.0-fac)/ndim;

fac2=fac1-fac;

for (j=1;j<=ndim;j++) ptry[j-1]=psum[j-1]\*fac1-p[ihi-1][j-1]\*fac2; ytry=ff.func(ptry,0);

if (ytry < y[ihi-1]) { y[ihi-1]=ytry;

for (j=1;j<=ndim;j++) {

psum[j-1] += ptry[j-1]-p[ihi-1][j-1];

p[ihi-1][j-1]=ptry[j-1];

}

}

return ytry;

}

### public static void amoeba(double p[][], double y[], int ndim, double ftol, f\_xi ff, int nfunk)

{

int i,ihi,ilo,inhi,j,mpts=ndim+1; int NMAX=5000;

double rtol,sum,swap,ysave,ytry; double psum[]=new double[ndim]; nfunk=0;

for (j=1;j<=ndim;j++)

{

for (sum=0.0,i=1;i<=mpts;i++) sum += p[i-1][j-1]; psum[j-1]=sum;

}

for (;;) { ilo=1;

//ihi = y[1]y[2] ? (inhi=2,1) : (inhi=1,2); if(y[0]y[1])

{

inhi=2;

}

ihi=1;

else

{

inhi=1;

}

ihi=2;

for (i=1;i<=mpts;i++) {

if (y[i-1] <= y[ilo-1]) ilo=i;

if (y[i-1] y[ihi-1]) { inhi=ihi;

ihi=i;

} else if (y[i-1] y[inhi-1] && i != ihi) inhi=i;

}

rtol=2.0\*Math.abs(y[ihi-1]-y[ilo-1])

/(Math.abs(y[ihi-1])+Math.abs(y[ilo-1])); if (rtol < ftol) {

swap=y[0]; y[0]=y[ilo-1];

y[ilo-1]=swap; for (i=1;i<=ndim;i++)

{

swap=p[0][i-1];

p[0][i-1]=p[ilo-1][i-1];

p[ilo-1][i-1]=swap;

}

break;

}

if (nfunk = NMAX) System.out.println("NMAX exceeded"); nfunk += 2;

ytry=amotry(p,y,psum,ndim,ff,ihi,-1.0);

if (ytry <= y[ilo-1]) ytry=amotry(p,y,psum,ndim,ff,ihi,2.0); else if (ytry = y[inhi-1]) {

ysave=y[ihi-1]; ytry=amotry(p,y,psum,ndim,ff,ihi,0.5); if (ytry = ysave) {

for (i=1;i<=mpts;i++) { if (i != ilo) {

for (j=1;j<=ndim;j++)

psum[j-1]=0.5\*(p[i-1][j-1]+p[ilo-1][j-1]);

p[i-1][j-1]=psum[j-1];

y[i]=ff.func(psum,0);

}

}

nfunk += ndim;

for (j=1;j<=ndim;j++)

{

for (sum=0.0,i=1;i<=mpts;i++) sum += p[i][j]; psum[j]=sum;

}

}

} else --nfunk;

}

}

// differential equation solutions

### public static double[] RK4(f\_xi fp,double x0,double xn,double f0,int N)

{

//4th order Runge Kutta Method

//fp :given derivative function df/dx(f,x)

// xo : initial value of the independent variable

// xn : final value of the independent variable

// f0 : initial value of the dependent variable

// N : number of dependent variable to be calculated

// fi : dependent variable double h=(xn-x0)/N;

double fi[]=new double[N+1]; double xi[]=new double[2]; double k[];

k=new double[4]; int i;

double x; fi[0]=f0;

for(x=x0,i=0;x<xn;x+=h,i++)

{

xi[0]=x; xi[1]=fi[i];

k[0]=h\*fp.func(xi,0); xi[0]=x+h/2.0; xi[1]=fi[i]+k[0]/2;

k[1]=h\*fp.func(xi,0);

xi[1]=fi[i]+k[1]/2.0;

k[2]=h\*fp.func(xi,0); xi[0]=x+h; xi[1]=fi[i]+k[2];

k[3]=h\*fp.func(xi,0); fi[i+1]=fi[i]+k[0]/6.0+k[1]/3.0+k[2]/3.0+k[3]/6.0;

}

return fi;

}

### public static double[][] RK4(f\_xi fp,double x0,double xn,double f0[],int N)

{

//4th order Runge Kutta Method

//fp : given set of derivative functions dfj/dxi(fj,x)

// xo : initial value of the independent variable

// xn : final value of the independent variable

// f0 : initial value of the dependent variable

// N : number of dependent variable to be calculated

// fi : dependent variable double h=(xn-x0)/N;

int M=f0.length; double fi[][];

fi=new double[M][N+1]; double xi[]=new double[M+1]; double k[]=new double[4];

int i,j; double x;

for(j=0;j<=M;j++) fi[j][0]=f0[j]; for(x=x0,i=0;x<xn;x+=h,i++)

{

for(j=1;j<=M;j++)

{

xi[0]=x; xi[1]=fi[j-1][i];

k[0]=h\*fp.func(xi,j-1); xi[0]=x+h/2.0; xi[1]=fi[j-1][i]+k[0]/2;

k[1]=h\*fp.func(xi,j-1);

xi[1]=fi[j-1][i]+k[1]/2.0;

k[2]=h\*fp.func(xi,j-1); xi[0]=x+h;

xi[1]=fi[j-1][i]+k[2];

k[3]=h\*fp.func(xi,j-1);

fi[j-1][i+1]=fi[j-1][i]+k[0]/6.0+k[1]/3.0+k[2]/3.0+k[3]/6.0;

}

}

return fi;

}

### public static double[][]

**RKF45(f\_xi fp,double x0,double xn,double f0,int N) throws IOException**

{

// Runge Kutta - Fehlberg Method

// fp : derivative function df/dx(f,x) (defined as a class)

// Tol :Tolerance

//Hmax : maximum possible step size

//Hmin : minimum possible step size

//k : Runge kutta coefficients

//Err : error

//x[i] : input variable to fp = df/dx(f,x) = df/dx(x[i])

//j : actual step size

//fi[][]:solution matrix

//fj[][]:solution matrix in exact size(j) (return function) double h=(xn-x0)/N;

double fi[][]=new double[2][1000]; double Tol=2.0e-3;

double Hmin=h/64.0; double Hmax=h\*64.0; double k[]=new double[6]; double Err;

double s;

double xi[]=new double[2]; int j=0;

fi[0][j]=x0;

fi[1][j]=f0;

while(fi[0][j] < xn )

{

if(fi[0][j]+h xn) h=xn-fi[0][j]; xi[0]=fi[0][j];

xi[1]=fi[1][j];

k[0]=h\*fp.func(xi,0);

xi[0]=fi[0][j]+h/4.0;

xi[1]=fi[1][j]+k[0]/4.0;

k[1]=h\*fp.func(xi,0);

xi[0]=fi[0][j]+3.0/8.0\*h; xi[1]=fi[1][j]+3.0/32.0\*k[0]+9.0/32.0\*k[1];

k[2]=h\*fp.func(xi,0);

xi[0]=fi[0][j]+12.0/13.0\*h;

xi[1]=fi[1][j]+1932.0/2197.0\*k[0]-7200.0/2197.0\*k[1]+7296.0/2197.0\*k[2];

k[3]=h\*fp.func(xi,0);

xi[0]=fi[0][j]+h;

xi[1]=fi[1][j]+439.0/216.0\*k[0]-8.0\*k[1]+3680.0/513.0\*k[2]-845.0/4104.0\*k[3];

k[4]=h\*fp.func(xi,0);

xi[0]=fi[0][j]+0.5\*h;

xi[1]=fi[1][j]-8.0/27.0\*k[0]+2.0\*k[1]

-3544/2565\*k[2]+1859.0/4104.0\*k[3]-11.0/40.0\*k[4];

k[5]=h\*fp.func(xi,0); Err=Math.abs(1.0/360.0\*k[0]-128/4275\*k[2]

-2197.0/75240\*k[3]+1.0/5.0\*k[4]+2.0/55.0\*k[5]);

if(Err<Tol || h<2\*Hmin)

{

fi[0][j+1]=fi[0][j]+h; fi[1][j+1]=fi[1][j]+16.0/135.0\*k[0]+6656.0/12825.0\*k[2]+ 28561.0/56430.0\*k[3]-9.0/50.0\*k[4]+2.0/55.0\*k[5];

j++;

}

else

{

if(Err==0) s=0;

else s=0.84\*(Math.pow(Tol\*h/Err,0.25)); if(s<0.75 && h(2\*Hmin) ) h/=2.0; if(s1.5 && Hmax (2\*h) ) h\*=2.0;

}

}

double fj[][]=new double[2][j+1]; for(int i=0;i<=j;i++)

{

fj[0][i]=fi[0][i];

fj[1][i]=fi[1][i];

}

return fj;

}

//End of Class Numeric

}

### NUMERIC SINIFI (SAYISAL ANALIZ PAKETI) ÖRNEK PROBLEMLERI Non linear denklem çözümleri :

**Problem 12.7 non lineer denklem sistemi çözümü örnegi ,** Matrix2.java

import java.io.\*; import Numeric; import complex;

### class f1 extends fi\_xi

{

double[] func(double[] x)

{

double b[]; b=new double[2];

b[0]=x[0]\*x[0]-x[1]-1;

b[1]=(x[0]-2)\*(x[0]-2)+(x[1]-0.5)\*(x[1]-0.5)-1.0;

return b;

}

}

### class df1 extends fij\_xi

{

double[][] func(double x[])

{

double b[][];

b=new double[2][2]; b[0][0]=2.0\*x[0];

b[0][1]=-1.0; b[1][0]=2.0\*x[0]-4.0;

b[1][1]=2.0\*x[1]-4.0;

return b;

}

}

### class f2 extends f\_x

{

double func(double x)

{

return 4.2+0.45\*x+0.0025\*Math.pow(x,2.8)-(60+8\*x-0.16\*x\*x);

}

}

### class df2 extends f\_x

{

double func(double x)

{

return Math.cos(x)-0.5;

}

}

### class Matrix2

{

public static void main(String args[]) throws IOException

{

//Solution of zero of functions of one or multivariable equations f1 b1;

b1=new f1(); df1 db1; db1=new df1(); f2 b2;

b2=new f2(); df2 db2;

db2=new df2(); double x[]; x=new double[2]; x[0]=0;

x[1]=0.5;

double xx=13;

System.out.println("Roots of equations : ");

System.out.println("Newtons method y=f(x)=0 f(x) and df(x)/dx is given"); System.out.println("Newton y,dy x= "+Numeric.newton(xx,b2,db2)); System.out.println("");

System.out.println("Newtons method y=f(x)=0 f(x) is given"); System.out.println("Newton y, x= "+Numeric.newton(xx,b2)); System.out.println("");

System.out.println("Newtons method with 2nd derivative y=f(x)=0 "); System.out.println("f(x) and df(x)/dx is given "); System.out.println("Newton\_2nd\_derivative y,dy x= "+ Numeric.newton\_2nd\_derivative(xx,b2,db2));

System.out.println("");

System.out.println("Newtons method with 2nd derivative y=f(x)=0 "); System.out.println("f(x) is given only "); System.out.println("Newton\_2nd\_derivative y, x= "+ Numeric.newton\_2nd\_derivative(xx,b2));

System.out.println("");

System.out.println("Secant method with 2nd derivative y=f(x)=0 "); System.out.println("f(x) is given only "); System.out.println("Secant\_2nd\_derivative y, x= "+ Numeric.secant\_2nd\_derivative(xx,b2));

System.out.println("");

System.out.println("Newtons method for system of equations"); System.out.println("yi=fi(xi)=0,i=0,n "); System.out.println("fi(xi) and dfi(xi)/dxj,i=1,n,j=1,n is given");

System.out.println(" "+Matrix.toStringT(Numeric.newton(x,b1,db1))); System.out.println("");

System.out.println("Newtons method for system of equations"); System.out.println("yi=fi(xi)=0,i=0,n ");

System.out.println("fi(xi),i=1,n is given(derivative matrix is notrequired)"); System.out.println(" "+Matrix.toStringT(Numeric.newton(x,b1)));

}

}

### Çözüm seti :

Roots of equations :

Newtons method y=f(x)=0 f(x) and df(x)/dx is given Newton y,dy x= 1.895494267033981

Newtons method y=f(x)=0 f(x) is given Newton y, x= 1.895494267033981

Newtons method with 2nd derivative y=f(x)=0 f(x) and df(x)/dx is given

Newton\_2nd\_derivative y,dy x= 1.895494267033981 Newtons method with 2nd derivative y=f(x)=0

f(x) is given only

Newton\_2nd\_derivative y, x= 1.895494267033981 Secant method with 2nd derivative y=f(x)=0

f(x) is given only

Secant\_2nd\_derivative y, x= 1.895494267033981 Newtons method for system of equations yi=fi(xi)=0,i=0,n

fi(xi) and dfi(xi)/dxj,i=1,n,j=1,n is given 1.0673460858066892

0.13922766688686042

Newtons method for system of equations yi=fi(xi)=0,i=0,n

fi(xi),i=1,n is given(derivative matrix is notrequired) 1.0673460858066897

0.1392276668868614

### Integral -türev : Integral ve türev örnegi

**Problem 12.8 integral ve türev örnegi, integ1.java**

//======================================================

// Numerical Analysis package in java

// example to show utilisation of integration (integral)

// and differentiation (derivative) functions

// Dr. Turhan Coban

// =====================================================

import java.io.\*; import Numeric; import complex;

class f1 extends f\_x

{

double func(double x)

{

return x\*x;

}

}

class f2 extends f\_x

{

double func(double x)

{

return x;

}

}

class fm1 extends f\_xi

{

//multivariable function

double func(double x[],int x\_ref)

{

double a=0; switch(x\_ref)

{

case 0: a=(x[0]+x[1]\*x[1]+x[2]\*2);break;

case 1: a=(x[0]\*3+x[1]\*x[1]\*x[1]+x[2]);break;

case 2: a=(x[0]-5\*x[1]\*x[1]+x[2]\*x[2]\*x[2]);break;

}

return a;

}

}

class fm2 extends fi\_xi

{

//multivariable function double[] func(double x[])

{

double a[]; a=new double[3];

a[0]=(x[0]+x[1]\*x[1]+x[2]\*2);

a[1]=(x[0]\*3+x[1]\*x[1]\*x[1]+x[2]);

a[2]=(x[0]-5\*x[1]\*x[1]+x[2]\*x[2]\*x[2]);

return a;

}

}

class integ1

{

public static void main(String args[]) throws IOException

{

f1 b1; b1=new f1(); f2 b2; b2=new f2(); fm1 b3;

b3=new fm1(); double x[]; fm2 b4; b4=new fm2();

x=new double[3]; x[0]=1;

x[1]=2;

x[2]=3;

System.out.println("integral of class f1 : "+ Numeric.integral(b1,0.0,1.0)); System.out.println("romberg integral of class f1 : "+ Numeric.integral\_romberg(b1,0.0,1.0)); System.out.println("integral of class f2 : "+ Numeric.integral(b2,0.0,1.0)); System.out.println("Derivative of class f1 : "+ Numeric.derivative(b1,1.0)); System.out.println("Derivative of class f2 : "+ Numeric.derivative(b2,1.0));

System.out.println("Derivative of fm1,0 : "+ Numeric.derivative(b3,1,x,0)); System.out.println("Derivative of fm2,0 : "+ Numeric.derivative(b4,1,x,0)); System.out.println("Derivative of fm1,0 : "+ Numeric.derivative(b3,1,x,0)); System.out.println("Derivative of fm2,0 : "+ Numeric.derivative(b4,1,x,0));

System.in.read();

}

}

integ1.java

### çözüm :

integral of class f1 : 0.333333333333329

romberg integral of class f1 : 0.3333333333333333 integral of class f2 : 0.49999999999999384 Derivative of class f1 : 1.9999999999994864 Derivative of class f2 : 0.9999999999998606 Derivative of fm1,0 : 3.0000000000028932 Derivative of fm2,0 : 3.0000000000028932 Derivative of fm1,0 : 3.0000000000028932 Derivative of fm2,0 : 3.0000000000028932

### Least Square polinomu uydurma

**Problem 12.9 least square polinom uydurma,** least.java

//===========================================

// Polynomial least square curve fitting

// Dr. Turhan Coban

//===========================================

import java.io.\*; import Numeric; import Text;

class least

{

public static void main(String args[]) throws IOException

{

//polynomial least square curve fitting

double e[]; //coefficients of polynomial least square e=new double[20];

double xi[];

xi=new double[500]; double yi[];

yi=new double[500];

DataInputStream cin=new DataInputStream(System.in); System.out.println("name of input file : ");

String in\_name=Text.readString(cin); System.out.println(" ");

DataInputStream fin=new DataInputStream(new FileInputStream(in\_name)); PrintStream fout=new PrintStream(new FileOutputStream("out.dat"));

int i=-1; try {

while(fin != null)

{

i++;

xi[i]=Text.readDouble(fin); yi[i]=Text.readDouble(fin); System.out.println(" "+xi[i]+" "+yi[i]);

}

} catch(EOFException e\_eof) {System.out.println("end of file"); }

System.out.println("degree of curve fitting : "); int n=Text.readInt(cin); e=Numeric.poly\_least\_square(xi,yi,n); System.out.println(Matrix.toString(e)); fout.println(Matrix.toString(e));

System.out.println("least square error : "+Numeric.error(xi,yi,e)); System.out.println("would you like to send curve fitted data to a data file (y/n) "); char answer=Text.readChar(cin);

if(answer=='y')

{

System.out.println("name of output file : "); String out\_name=Text.readString(cin);

PrintStream fout1=new PrintStream(new FileOutputStream(out\_name)); double xmin,xmax;

System.out.println("minimum x : "); xmin=Text.readDouble(cin); System.out.println("maximum x : "); xmax=Text.readDouble(cin); System.out.println("number of data points : "); int n\_data=Text.readInt(cin);

double x; for(i=0;i<n\_data;i++)

{

x=xmin+(xmax-xmin)/(n\_data-1.0)\*i;

System.out.println(" "+x+" "+Numeric.f\_least\_square(e,x)); fout1.println(" "+x+" "+Numeric.f\_least\_square(e,x));

} //end of for

}//end of if

}

}

least square polinom uydurma islevini biraz daha kullaniciya cazip hale getirmek için grafik formatindaki leastSW.java programi gelistirilmistir.

### Problem 12.10 least square polinom uydurma, leastSW.java jFrame grafik programi

//======================================================

// Numerical Analysis package in java

// En küçük kareler metodu polinom uydurma

// Dr. Turhan Coban

// =====================================================

/\*

* Java 2 Swing JFrame version.

\*/

import java.lang.Integer; import java.awt.\*; import java.awt.event.\*; import java.awt.font.\*; import java.awt.geom.\*; import java.awt.image.\*;

import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import java.awt.event.ItemListener; import java.awt.event.ItemEvent; import javax.swing.\*;

import javax.swing.border.\*; import Numeric;

import Text; import java.util.\*; import java.io.\*; import Plot2D;

import PlotShapesSW; import java.security.\*;

public class leastSW extends JFrame implements ItemListener,ActionListener

{

boolean inAnApplet = true;

final static String girdi = "DATA girdi sayfasi";

final static String leastcikti = "En kucuk kareler çikti sayfasi"; final static String KONTROLPanel = "PLOT Kontrol sayfasi"; final static String PLOTPanel = "Plot sayfasi";

Plot2D jta;

StringTokenizer t;

// least degisken seti \*\*\*\*\*\*\*\*

double ee[];

double xi[];

double yi[];

BufferedReader fin;

File girdiDosyasi;

File ciktiDosyasi; File plotDosyasi; int nGirdi;

int nCikti;

double minX,maxX; int PolinomDerecesi;

// \*\*\*\*\*\* girdi sayfasi \*\*\*\*\*\*\* JLabel promptGirdi; JTextField inputGirdi; JLabel promptPD; JTextField inputPD; JTextArea outputGirdi;

// \*\*\*\*\*\* çikti sayfasi \*\*\*\*\*\*\* JLabel promptCikti; JTextField inputCikti; JTextArea outputCikti; JTextArea outputFormul;

JLabel promptP1\_1,promptP1\_2,promptP1\_3,promptP1\_4; JTextField inputP1\_1,inputP1\_2,inputP1\_3,inputP1\_4;

// \*\*\*\*\*\* plot kontrol sayfasi \*\*\*\*

JLabel promptXmin; // Label prompt in Xmin field JLabel promptXmax; // Label prompt in Xmax field JLabel promptYmin; // Label prompt in Ymin field JLabel promptYmax; // Label prompt in Ymax field JLabel promptLabel; // Label prompt Plot Label; JLabel promptXLabel; // Label prompt Plot XLabel; JLabel promptYLabel; // Label prompt Plot YLabel; JLabel promptXntic; // Label prompt in Xmin field JLabel promptYntic; // Label prompt in Xmax field JLabel promptXgridon; // Label prompt in Ymin field JLabel promptYgridon; // Label prompt in Ymax field JTextField inputXmin; // input field Xmin

JTextField inputXmax; // input field Xmax JTextField inputYmin; // input field Ymin JTextField inputYmax; // input field Ymax JTextField inputLabel; // input field Label JTextField inputXLabel; // input field XLabel JTextField inputYLabel; // input field YLabel JTextField inputXntic; // input field xntic JTextField inputYntic; // input field yntic JCheckBox inputXgridon; // input field xgridon JCheckBox inputYgridon; // input field ygridon JButton printButton;

//===========================================================

public leastSW(String s) throws IOException

{

super(s);

Border b=BorderFactory.createRaisedBevelBorder(); girdiDosyasi=new File("in.txt");

girdiDosyasi=new File(girdiDosyasi.getAbsolutePath()); ciktiDosyasi=new File("out.txt");

ciktiDosyasi=new File(ciktiDosyasi.getAbsolutePath()); plotDosyasi=new File("Plot.txt");

plotDosyasi=new File(plotDosyasi.getAbsolutePath()); oku();

nGirdi=xi.length; nCikti=nGirdi; minX=xi[0]; maxX=xi[nGirdi-1]; ee=new double[nGirdi]; PolinomDerecesi=2;

ee=Numeric.poly\_least\_square(xi,yi,PolinomDerecesi); Container contentPane = getContentPane(); JTabbedPane tabbedPane = new JTabbedPane();

JPanel pane1 = new JPanel(); JPanel pane2 = new JPanel();

pane2.setLayout(new BorderLayout()); JPanel pane3=new JPanel();

JPanel pane4=new JPanel(); JPanel mpane=new JPanel();

mpane.setLayout(new GridLayout(1,8)); JPanel xpane=new JPanel(); xpane.setLayout(new GridLayout(1,8)); JPanel lpane=new JPanel(); lpane.setLayout(new GridLayout(3,2)); JPanel PanelCikti1=new JPanel(); PanelCikti1.setLayout(new GridLayout(4,2)); JPanel PanelCikti2=new JPanel(); PanelCikti2.setLayout(new BorderLayout());

//JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS; JPanel PanelGirdi1=new JPanel(); PanelGirdi1.setLayout(new GridLayout(2,2));

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

promptGirdi=new JLabel("Girdi dosya ismi"); inputGirdi=new JTextField("in.txt"); promptPD=new JLabel("polinom derecesi"); inputPD=new JTextField(""+PolinomDerecesi); outputGirdi=new JTextArea(yazxiyi(),20,40); outputGirdi.setLineWrap(true); outputGirdi.setBorder(b);

promptCikti=new JLabel("Cikti dosya ismi"); inputCikti=new JTextField("out.txt"); promptP1\_2=new JLabel("veri sayisi"); promptP1\_3=new JLabel("Minimum x : "); promptP1\_4=new JLabel("Maximum x : "); inputP1\_2=new JTextField(""+nCikti); inputP1\_3=new JTextField(""+minX); inputP1\_4=new JTextField(""+maxX); outputCikti=new JTextArea(yazCikti(),20,40); outputCikti.setBorder(b);

outputFormul=new JTextArea(yaze(),10,40); outputFormul.setLineWrap(true); outputFormul.setAutoscrolls(true); outputFormul.setBorder(b); PanelCikti1.add(promptCikti); PanelCikti1.add(inputCikti); PanelCikti1.add(promptP1\_2); PanelCikti1.add(inputP1\_2); PanelCikti1.add(promptP1\_3); PanelCikti1.add(inputP1\_3); PanelCikti1.add(promptP1\_4); PanelCikti1.add(inputP1\_4); PanelCikti2.add(PanelCikti1,BorderLayout.NORTH);

PanelCikti2.add(new JScrollPane(outputFormul),BorderLayout.CENTER); PanelCikti2.add(new JScrollPane(outputCikti),BorderLayout.SOUTH);

PanelGirdi1.add(promptGirdi); PanelGirdi1.add(inputGirdi); PanelGirdi1.add(promptPD); PanelGirdi1.add(inputPD); pane3.add(PanelGirdi1);

pane3.add(new JScrollPane(outputGirdi)); pane4.add(PanelCikti2);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

promptXmin = new JLabel("Xmin "); inputXmin = new JTextField(5); promptXmax = new JLabel("Xmax "); inputXmax = new JTextField(5); promptYmin = new JLabel("Ymin "); inputYmin = new JTextField(5); promptYmax = new JLabel("Ymax "); inputYmax = new JTextField(5);

//\*\*\*\*\*\*\*

promptLabel = new JLabel("Plot basligi : "); promptXLabel = new JLabel("x ekseni basligi : "); promptYLabel = new JLabel("y ekseni basligi : "); inputLabel = new JTextField(30);

inputXLabel = new JTextField(30); inputYLabel = new JTextField(30);

//\*\*\*\*\*\*\*

promptXntic=new JLabel("X tic no"); inputXntic=new JTextField(5); promptYntic=new JLabel("Y tic no"); inputYntic=new JTextField(5); promptXgridon=new JLabel("X grid"); inputXgridon=new JCheckBox(" "); promptYgridon=new JLabel("Y grid"); inputYgridon=new JCheckBox(" ");

//\*\*\*\*\*\*\* mpane.add(promptXmin); mpane.add(inputXmin); mpane.add(promptXmax); mpane.add(inputXmax); mpane.add(promptYmin); mpane.add(inputYmin); mpane.add(promptYmax); mpane.add(inputYmax);

pane1.add(mpane,BorderLayout.NORTH); xpane.add(promptXntic); xpane.add(inputXntic); xpane.add(promptYntic); xpane.add(inputYntic); xpane.add(promptXgridon); xpane.add(inputXgridon); xpane.add(promptYgridon); xpane.add(inputYgridon); pane1.add(xpane,BorderLayout.NORTH);

//\*\*\*\*\*\*\*\* lpane.add(promptLabel); lpane.add(inputLabel); lpane.add(promptXLabel); lpane.add(inputXLabel); lpane.add(promptYLabel); lpane.add(inputYLabel);

pane1.add(lpane,BorderLayout.SOUTH);

//\*\*\*\*\*\*\*\*\* jta=new Plot2D();

inputXmin.setText(Double.toString(jta.p1.xmin)); inputXmax.setText(Double.toString(jta.p1.xmax)); inputYmin.setText(Double.toString(jta.p1.ymin)); inputYmax.setText(Double.toString(jta.p1.ymax)); inputXntic.setText(Integer.toString(jta.p1.xntic)); inputYntic.setText(Integer.toString(jta.p1.yntic)); inputLabel.setText(jta.p1.label); inputXLabel.setText(jta.p1.xlabel); inputYLabel.setText(jta.p1.ylabel); printButton=new JButton("Yazdir"); pane2.add(jta); pane1.add(printButton,BorderLayout.SOUTH); tabbedPane.addTab(girdi, pane3); tabbedPane.addTab(leastcikti, pane4); tabbedPane.addTab(PLOTPanel, pane2); tabbedPane.addTab(KONTROLPanel, pane1);

contentPane.add( tabbedPane, BorderLayout.CENTER); printButton.addActionListener(this); inputXmin.addActionListener(this); inputXmax.addActionListener(this); inputYmin.addActionListener(this); inputYmax.addActionListener(this); inputLabel.addActionListener(this); inputXLabel.addActionListener(this); inputYLabel.addActionListener(this); inputXntic.addActionListener(this); inputYntic.addActionListener(this); inputXgridon.addItemListener(this); inputYgridon.addItemListener(this);

//en küçük kareler metodu action listener baglantisi inputP1\_2.addActionListener(this); // nCikti inputP1\_3.addActionListener(this); // minX inputP1\_4.addActionListener(this); // maxX inputGirdi.addActionListener(this); // Girdi dosyasi inputCikti.addActionListener(this); // Cikti dosyasi inputPD.addActionListener(this); // PolinomDerecesi leastCiktisiniPlotGirdisiOlarakYazdir();

}

//===========================================================

public void itemStateChanged(ItemEvent e)

{

inputXmin.setText(Double.toString(jta.p1.xmin)); inputXmax.setText(Double.toString(jta.p1.xmax)); inputYmin.setText(Double.toString(jta.p1.ymin)); inputYmax.setText(Double.toString(jta.p1.ymax)); inputXntic.setText(Integer.toString(jta.p1.xntic)); inputYntic.setText(Integer.toString(jta.p1.yntic)); Object source=e.getItemSelectable(); if(source==inputXgridon)

{

if (e.getStateChange() == ItemEvent.DESELECTED)

{

jta.p1.xgridon=0;

}

else

{

jta.p1.xgridon=1;

}

}

else if(source==inputYgridon)

{

if (e.getStateChange() == ItemEvent.DESELECTED)

{

jta.p1.ygridon=0;

}

else

{

jta.p1.ygridon=1;

}

}

inputLabel.setText(jta.p1.label); inputXLabel.setText(jta.p1.xlabel); inputYLabel.setText(jta.p1.ylabel); jta.yenidenciz();

}

//===========================================================

public void leastCiktisiniPlotGirdisiOlarakYazdir()

{

try

{

BufferedWriter fplot=new BufferedWriter(new FileWriter(plotDosyasi)); fplot.println(inputLabel.getText());

fplot.println(inputXLabel.getText()); fplot.println(inputYLabel.getText()); fplot.println(inputGirdi.getText()+" 20 0 0 0 ");

fplot.println(inputCikti.getText()+" 3 0 0 255 "); fplot.close();

}

catch(IOException e1)

{

System.err.println("girdi cikti hatasi : Plot.txt");

}

catch(AccessControlException ace)

{

System.err.println("least ciktisini plot girdisi olarak yazdir access kontrol hatasi : "+ plotDosyasi);

}

}

public void sayfalariyenile()

{

//en küçük kareler metoduna göre degisik

// sayfalardaki de§erleri yeniler girdiDosyasi=new File(inputGirdi.getText()); Integer d1=new Integer(inputPD.getText()); PolinomDerecesi=d1.intValue(); ciktiDosyasi=new File(inputCikti.getText()); Integer d2=new Integer(inputP1\_2.getText()); nCikti=d2.intValue();

Double d3=new Double(inputP1\_3.getText()); minX=d3.doubleValue();

Double d4=new Double(inputP1\_4.getText()); maxX=d4.doubleValue();

try{ oku();

}

catch(IOException e1)

{

System.err.println("least girdi dosyasi hatasi : "+ girdiDosyasi);

}

catch(AccessControlException ace)

{

System.err.println("least girdi access kontrol hatasi : "+ girdiDosyasi);

}

ee=new double[PolinomDerecesi+1]; ee=Numeric.poly\_least\_square(xi,yi,PolinomDerecesi); outputGirdi.setText(yazxiyi()); outputCikti.setText(yazCikti()); outputFormul.setText(yaze()); jta.yenidanPlotDatasiOku(); leastCiktisiniPlotGirdisiOlarakYazdir();

}

//===========================================================

public void actionPerformed(ActionEvent e)

{

// en kucuk kareler metodu girdileri verildiginde bu yeni action

// eventi olusturur

if( e.getSource()==inputGirdi || e.getSource()==inputPD || e.getSource()==inputCikti || e.getSource()==inputP1\_2 || e.getSource()==inputP1\_3 || e.getSource()==inputP1\_4 )

{

sayfalariyenile();

}

//grafigi yaziciya gönderir

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (e.getSource()==printButton )

{

jta.yazdir();

}

//grafik min max degerlerini yeniden okur

Double valXmin=new Double(inputXmin.getText()); jta.p1.xmin=valXmin.doubleValue();

Double valXmax=new Double(inputXmax.getText()); jta.p1.xmax=valXmax.doubleValue();

Double valYmin=new Double(inputYmin.getText()); jta.p1.ymin=valYmin.doubleValue();

Double valYmax=new Double(inputYmax.getText());

//\*\*\*\*\*

//grafik tic sayisini yeniden okur

Integer valXntic=new Integer(inputXntic.getText()); jta.p1.xntic=valXntic.intValue();

Integer valYntic=new Integer(inputYntic.getText()); jta.p1.yntic=valYntic.intValue();

//\*\*\*\*\*

//grafik basliklarini yeniden okur jta.p1.ymax=valYmax.doubleValue();

jta.p1.label=inputLabel.getText(); jta.p1.xlabel=inputXLabel.getText(); jta.p1.ylabel=inputYLabel.getText();

//Plot'i (grafik) yeniden çizer jta.yenidenciz();

}

//===========================================================

//girdi dosyasindan girdi degerlerini okur.

public void oku() throws IOException

{

//dosyadan bilgileri okur xi,yi boyutlu matrisine y• kler. try{

fin=new BufferedReader(new FileReader(girdiDosyasi));

}

catch(FileNotFoundException e2)

{

System.err.println(e2.toString());

}

nGirdi=0;

Vector v1,v2;

v1=new Vector(1); v2=new Vector(1); String X1,X2; boolean EOF=false; while(!EOF)

{

try{ X1=Text.readString(fin); X2=Text.readString(fin); v1.addElement(X1); v2.addElement(X2); nGirdi++;

}catch(EOFException e)

{

fin.close();

EOF=true;

}

}

xi=new double[nGirdi]; yi=new double[nGirdi];

Enumeration enum1=v1.elements(); Enumeration enum2=v2.elements(); StringBuffer b=new StringBuffer();

//for(int i=0;i<n;i++) int i=0;

while(enum1.hasMoreElements())

{

X1=(String)enum1.nextElement(); Double d1=new Double(X1); X2=(String)enum2.nextElement(); Double d2=new Double(X2); xi[i]=d1.doubleValue(); yi[i]=d2.doubleValue();

i++;

}

} //oku metodu sonu

//===========================================================

//girdi alanina girdi de§erlerini yazar

public String yazxiyi()

{

String b=new String(); for(int i=0;i<xi.length;i++)

{

b=b+" "+xi[i]+"\t "+yi[i]+"\n";

}

return b;

}

//========================================================

// Polinom katsayilarini yazar public String yaze()

{

String b="Polinom katsayilari : ";

b=b+"f(x) = a[0]+a[1]\*x+a[2]\*x^2+....a[n]\*x^n \n"; for(int i=0;i<=PolinomDerecesi;i++)

{

b=b+"a["+i+"] = "+ee[i]+"\n";

}

return b;

}

//=========================================================

// cikti dosyasina kayit yapar ve cikti alanina sonuclar• yazar public String yazCikti()

{

String b=new String(); try{

ObjectOutputStream fout=new ObjectOutputStream(new FileOutputStream(ciktiDosyasi)); b="cikti dosyasi : \n";

double x,y;

double nn=((maxX-minX)/(nCikti-1)); for(int i=0;i<nCikti;i++)

{

x=minX+nn\*i; y=Numeric.f\_least\_square(ee,x); b=b+" "+x+"\t "+y+"\n";

}

fout.writeObject(b); try{

fout.close();

}

catch(IOException io)

{

System.exit(1);

}

}

catch(IOException e1)

{

System.err.println("input output error");

}

catch(AccessControlException ece)

{

System.err.println("yaz Cikti dosyasi"+ciktiDosyasi+ " excess kontrol");

}

return b;

}

//===========================================================

public static void main( String argv[] ) throws IOException {

leastSW frame = new leastSW( "En küçük kareler metodu egri uydurma" ); frame.addWindowListener(new BasicWindowMonitor());

frame.setSize( 600, 600 ); frame.setVisible(true);

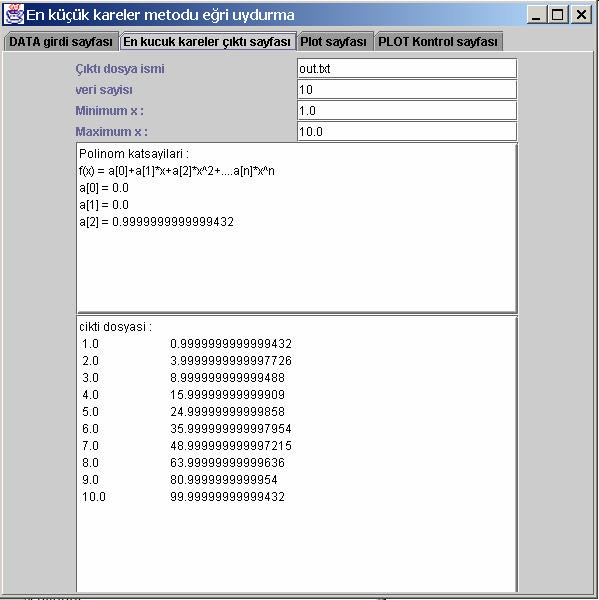
}

}

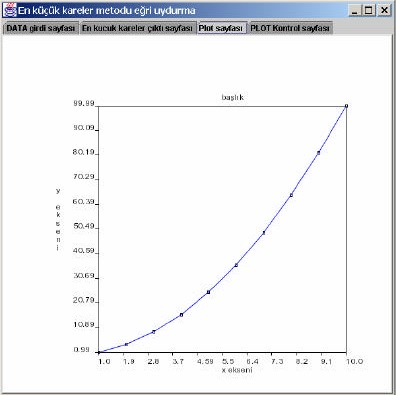
### 12001.JPG



**12002.JPG**



**12003.JPG**



**12004.JPG**



Sekil 12.1-4 en küçük kareler metoduyla polinaom egri uyduran leastSW.java sinifi Jframe çiktilarinin çesitli sayfalarinin görünümü.

Örnek çoktidaki girdi dosyalari :

Plot.txt baslik

x ekseni y ekseni 2

in.txt 20 0 0 0

out.txt 3 0 0 255

### in.txt

1.0 0.9999999999999432

2.0 3.9999999999997726

3.0 8.999999999999488

4.0 15.99999999999909

5.0 24.99999999999858

6.0 35.999999999997954

7.0 48.999999999997215

8.0 63.99999999999636

9.0 80.9999999999954

10.0 99.99999999999432

### out.txt

cikti dosyasi :

1.0 0.9999999999999432

|  |  |
| --- | --- |
| 2.0 | 3.9999999999997726 |
| 3.0 | 8.999999999999488 |
| 4.0 | 15.99999999999909 |
| 5.0 | 24.99999999999858 |
| 6.0 | 35.999999999997954 |
| 7.0 | 48.999999999997215 |
| 8.0 | 63.99999999999636 |
| 9.0 | 80.9999999999954 |
| 10.0 | 99.99999999999432 |

### Diferansiyel Denklemler :

**Problem 12.11 diferansiyel denklem testi, RK4 metodu dif1.java**

//======================================================

// Numerical Analysis package in java

// example to show differential equation solution

// and differentiation (derivative) functions

// Dr. Turhan Coban

// =====================================================

import java.io.\*; import Numeric; import complex; import Matrix;

class fm1 extends f\_xi

{

//multivariable function

double func(double x[],int x\_ref)

{

//x[0] is x, x[1] is y

//this is a representation of equation : dy/dx=y

//solution of this equation is e(x) if(x\_ref==0) return x[1];

else return 0;

}

}

class dif1

{

public static void main(String args[]) throws IOException

{

fm1 b3=new fm1(); double x[];

x=new double[1]; x[0]=1;

//RK4 differential equation to solve differential equation dy/dx=y

//with limits 0 to 1 exact solution is e=2.7182818

//boundary condition x=0 y=1 is given

//final value x=1 to be calculated by the method System.out.println(Matrix.toStringT(Numeric.RK4(b3,0.0,1.0,1.0,50)));

}

}

Çözüm seti : 1.0

1.02020134

1.0408107741377959

1.0618365464618167

1.0832870675613175

1.1051709179307265

1.1274968514019572

1.1502737986460576

1.1735108707455981

1.197217362839226

1.2214027578398445

1.2460767302279048

1.271249149921327

1.2969300862235986

1.3231298118516308

1.3498588070449815

1.3771277637580916

1.4049475899372086

1.4333294138837107

1.4622845887055762

1.4918246968587778

1.5219615547804188

1.5527072176154666

1.5840739840389706

1.6160744011756962

1.648721269619143

1.682027648551951

1.7160068609697496

1.7506724990105325

1.7860384293916938

1.8221187989569014

1.8589280403350215

1.896480877713363

1.9347923327275491

1.9738777304703714

2.0137527056220317

2.0544332087042223

2.095935512460547

2.1382762183658373

2.1814722632669596

2.2255409261577848

2.270499835091013

2.3163669742296302

2.363160691040814

2.4108997036351645

2.4596031082541976

2.5092903869090972

2.5599814151737794

2.611696470135386

2.6644562385053905

2.7182818248945586

### Problem 12.12 : çok boyutlu RKF45 diferansiyel denklem çözüm metodu örnegi, dif2.java

//======================================================

// Numerical Analysis package in java

// example to show differential equation solution

// and differentiation (derivative) functions

// Dr. Turhan Coban

// =====================================================

import java.io.\*; import Numeric; import complex;

import Matrix;

class fm1 extends f\_xi

{

//multivariable function

double func(double x[],int x\_ref)

{

//x[0] is x, x[1] is y

if(x\_ref==0) return -2.0\*x[0]-x[1]; else return 0;

}

}

class dif2

{

public static void main(String args[]) throws IOException

{

fm1 b3=new fm1(); double x[];

x=new double[1]; x[0]=1;

//RK5 differential equation to solve equation dy/dx=-2x-y

//initial value is given as x=0 y=-1 at x=0.4 y will be determined System.out.println (Matrix.toString(Matrix.T(Numeric.RKF45(b3,0.0,0.4,-1.0,10))));

}

}

### Çözüm :

0.0 -1.0

0.0050 -0.9950377826066799

0.01 -0.9901501844020889

0.015 -0.9853368332473136

0.02 -0.9805973588593618

0.025 -0.9759313928019077

0.030000000000000002-0.9713385684760812

0.035 -0.9668185211113052

0.04 -0.9623708877561767

0.045 -0.9579953072693951

0.049999999999999996-0.9536914203107341

0.05499999999999999-0.94945886933206

0.05999999999999999-0.9452972985683943

0.06499999999999999-0.9412063540290199

0.06999999999999999-0.9371856834886335

0.075 -0.9332349364785408

0.08 -0.9293537642778962

0.085 -0.9255418199049855

0.09000000000000001-0.9217987581085532

0.09500000000000001-0.9181242353591716

0.10000000000000002-0.9145179098406548

0.10500000000000002-0.9109794414415133

0.11000000000000003-0.9075084917464531

0.11500000000000003-0.9041047240279163

0.12000000000000004-0.9007678032376644

0.12500000000000003-0.8974973959984025

0.13000000000000003-0.8942931705954467

0.13500000000000004-0.891154796968432

0.14000000000000004-0.8880819467030621

0.14500000000000005-0.8850742930228999

0.15000000000000005-0.8821315107811996

0.15500000000000005-0.8792532764527788

0.16000000000000006-0.8764392681259321

0.16500000000000006-0.8736891654943832

0.17000000000000007-0.8710026498492798

0.17500000000000007-0.8683794040712255

0.18000000000000008-0.8658191126223544

0.18500000000000008-0.8633214615384427

0.19000000000000009-0.8608861384210608

0.1950000000000001-0.8585128324297647

0.2000000000000001-0.856201234274326

0.2050000000000001-0.8539510362070005

0.2100000000000001-0.8517619320148359

0.2150000000000001-0.849633617012017

0.2200000000000001-0.8475657880322502

0.22500000000000012-0.8455581434211851

0.23000000000000012-0.843610383028874

0.23500000000000013-0.8417222082022695

0.24000000000000013-0.8398933217777589

0.24500000000000013-0.8381234280737365

0.2500000000000001-0.8364122328832122

0.2550000000000001-0.8347594434664576

0.2600000000000001-0.8331647685436885

0.2650000000000001-0.8316279182877833

0.27000000000000013-0.8301486043170393

0.27500000000000013-0.8287265396879636

0.28000000000000014-0.8273614388881004

0.28500000000000014-0.8260530178288946

0.29000000000000015-0.8248009938385901

0.29500000000000015-0.8236050856551641

0.30000000000000016-0.8224650134192967

0.30500000000000016-0.8213804986673754

0.31000000000000016-0.8203512643245341

0.31500000000000017-0.8193770346977273

0.3200000000000002-0.8184575354688391

0.3250000000000002-0.8175924936878259

0.3300000000000002-0.8167816377658933

0.3350000000000002-0.8160246974687078

0.3400000000000002-0.8153214039096416

0.3450000000000002-0.8146714895430517

0.3500000000000002-0.8140746881575915

0.3550000000000002-0.8135307348695573

0.3600000000000002-0.8130393661162659

0.3650000000000002-0.812600319649468

0.3700000000000002-0.8122133345287916

0.3750000000000002-0.8118781511152201

0.3800000000000002-0.8115945110646018

0.38500000000000023-0.8113621573211925

0.39000000000000024-0.8111808341112298

0.39500000000000024-0.8110502869365394

0.4 -0.8109702625681743

### Problem 12.13 diferansiyel denklem çözüm örnegi, RK4 metodu, dif3.java

//======================================================

// Numerical Analysis package in java

// example to show differential equation solution

// and differentiation (derivative) functions

// Dr. Turhan Coban

// =====================================================

import java.io.\*; import Numeric; import complex; import Matrix;

class fm1 extends f\_xi

{

//multivariable function

double func(double x[],int x\_ref)

{

//x[0] is x, x[1] is y

if(x\_ref==0) return -2.0\*x[0]-x[1]; else return 0;

}

}

class dif3

{

public static void main(String args[]) throws IOException

{

fm1 b3=new fm1(); double x[];

x=new double[1]; x[0]=1;

//RK4 differential equation to solve equation dy/dx=-2x-y System.out.println(Matrix.toStringT(Numeric.RK4(b3,0.0,0.4,-1.0,100))); System.in.read();

}

}

### Çözüm :

-1.0

-0.9960239680320001

-0.992095744511233

-0.9882151385858677

-0.9843819601659565

-0.9805960199203918

-0.9768571292738785

-0.9731651004039156

-0.9695197462377918

-0.9659208804495921

-0.9623683174572164

-0.9588618724194105

-0.955401361232808

-0.9519866005289858

-0.9486174076715289

-0.945293600753109

-0.942014998592574

-0.9387814207320486

-0.9355926874340476

-0.9324486196785992

-0.9293490391603817

-0.9262937682858702

-0.9232826301704945

-0.9203154486358103

-0.9173920482066786

-0.9145122541084599

-0.9116758922642163

-0.9088827892919272

-0.9061327725017148

-0.903425669893081

-0.900761310152156

-0.898139522648956

-0.8955601374346547

-0.8930229852388627

-0.8905278974669202

-0.888074706197199

-0.8856632441784156

-0.8832933448269559

-0.8809648422242088

-0.8786775711139124

-0.8764313668995095

-0.8742260656415141

-0.872061504054888

-0.8699375195064296

-0.8678539500121697

-0.8658106342347813

-0.8638074114809979

-0.8618441216990421

-0.8599206054760645

-0.8580367040355941

-0.8561922592349969

-0.854387113562946

-0.8526211101369017

-0.8508940927006006

-0.8492059056215563

-0.8475563938885688

-0.8459454031092447

-0.8443727795075264

-0.8428383699212321

-0.8413420217996048

-0.8398835832008722

-0.8384629027898148

-0.8370798298353451

-0.8357342142080957

-0.8344259063780173

-0.8331547574119857

-0.8319206189714199

-0.8307233433099079

-0.8295627832708429

-0.8284387922850687

-0.8273512243685349

-0.826299934119961

-0.8252847767185101

-0.8243056079214711

-0.8233622840619519

-0.8224546620465801

-0.8215825993532133

-0.8207459540286591

-0.8199445846864039

-0.8191783505043502

-0.8184471112225643

Su ana kadarki örneklerimizin hepsi konsol programi olarak verildi. simdi grafik (applet) çiktisina da bir örnek olmak üzere ninci dereceden denklemin köklerini hesaplayan kokN.java programini verelim.

### Problem 12.14 ninici dereceden polinomun köklerini bulma, kokN.java programi

import java.util.\*; import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*; import Matrix;

public class kokN extends Applet implements ActionListener

{

private Label prompt1,prompt2; private TextField input; TextArea t;

Panel YaziPaneli; int n;

String s;

public void init()

{

setBackground(Color.lightGray); YaziPaneli=new Panel();

YaziPaneli.setFont(new Font("Serif",Font.BOLD,12)); YaziPaneli.setLayout( new GridLayout(3,1) );

t=new TextArea(8,47);

prompt1= new Label("a[0]+a[1]\*x+...+a[n]\*x^n=0");

prompt2= new Label("n inci dereceden polinomun katsayilarini giriniz : "); input = new TextField(30);

YaziPaneli.add(prompt1); YaziPaneli.add(prompt2); YaziPaneli.add(input); add(YaziPaneli);

add(t); input.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

s=input.getText();

StringTokenizer token=new StringTokenizer(s); t.setText("");

n=token.countTokens()-1; int m=n+1;

double a[]=new double[m]; complex z[]=new complex[n]; for(int i=0;i<n;i++)

{

z[i]=new complex();

}

int j=0;

while(token.hasMoreTokens())

{

Double ax=new Double(token.nextToken()); a[j++]=ax.doubleValue();

}

z=Matrix.poly\_rootsC(a); int i=0;

t.setText(Matrix.toStringT(z)); input.setText("");

}

}

Bu progamin sonucu :

### 12005.JPG



**Sekil 12.5 n inci dereceden polinomun köklerini bulan kokN programi, StringTokenizer ve Matrix sinif**

**örnegi**

### ALISTIRMALAR

* + 1. f(x)=x\*x\*x-3 denkleminin köklerini newton metodunu kullanarak çözünüz (3 ün küp kökünü hesaplayiniz), Bunun için bir java programi yazip çiktiyi hesaplatiniz.
    2. f(x)=x\*x\*x-3 denkleminin köklerini poly\_roots metodunu kullanarak çözünüz.
    3. f(x)=x\*x\*x-3 denkleminin köklerini bisection metodunu kullanarak çözünüz (3 ün küp kökünü hesaplayiniz), bunun için bir java programi yazip çiktiyi hesaplatiniz.
    4. f(x)=sin(x) denkleminin 0 ila pi arasindaki integralini hesaplayiniz. Bunun için bir java programi yazip çiktiyi hesaplatiniz.
    5. f(x)=sin(x) denkleminin pi noktasindaki türevini aliniz. Bunun için bir java programi yazip çiktiyi hesaplatiniz.
    6. f(x,y)=x\*x+y\*y-R\*R (R=sabit) denkleminin 0 ile 1 arasindaki türevini hesaplayiniz.
    7. df(x)/dx=-f(x) denkleminin x=1 den 2 ye kadar çözümünü yapiniz, f(0)=1, Bunun için bir java programi yazip çiktiyi hesaplatiniz.

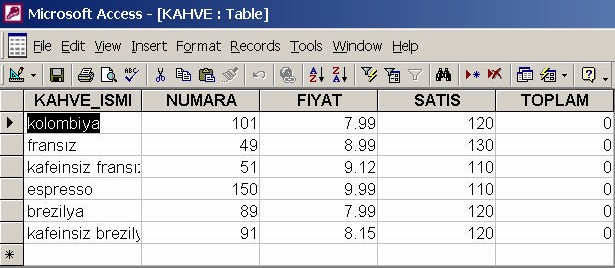
### BÖLÜM 13 DATABASE (VERI TABANI) PROGRAMLAMASINA GIRIS

### TEMEL KAVRAMLAR

Database programlarina Java dili üzerinden erismek için java JDBC API kullanilir. JDBC bize herhangi bir Data Base (veri Tabani) yapisina java üzerinden ulasmamizi ve Veri tabani dili olan SQL (Structured Query Language) dilinde direk olarak programlama yapabilmemizi saglar. JDBC ye geçmeden önce SQL dilinin bazi temel yapilarina göz atalim. Burada görecegimiz oldukça kompleks islemleri yapabilecegimiz üçüncü kusak bir bilgisayar dilidir. Bu kitabin konusu Veri tabanlari olmadigi için biz fazla detaya girmeyecegiz. Dileyen arkadaslar Veri Tabani ile ilgili kitaplarda SQL ile ilgili çok daha detayli bilgi bulabilirler.

Örnek olarak asagidaki Veri tabani dosyasinin oldugunu kabul edelim :

### 13001.JPG



**Sekil 13.1 Kahve veri tabani dosyasinin acsees veri tabani programinda görünüsü**

Bu dosyayi SQL dilinde yaratmak için :

**CREATE TABLE KAHVE** (KAHVE\_ISMI VARCHAR(32), NUMARA INTEGER,

FIYAT FLOAT, SATIS INTEGER, TOPLAM INTEGER)

komutunu kullanabiliriz. Buradaki temel komut CREATE TABLE komutudur. Veri tabani tablosu adi **KAHVE** ve Tablo sütun isimleri KAHVE\_ISMI, NUMARA, FIYAT, SATIS ve TOPLAM olarak tanimlanmistir. degisken isminden sonra SQL degisken türleri verilmistir. Deqsken türlerinin isimleri degisik SQL dillerinde degisebilir.

Bu tablodan belli degerleri girmek için INTERT INTO deyimi kullanilir. Örnegin tablodaki ilk satir

**INSERT INTO KAHVE VALUES**('Fransiz',49,8.99,130,0)

SQL deyimiyle girilebilir.

Tablodan belli degerleri geri çekmek için ise SELECT deyimi kullanilir. SELECT deyiminin en genel sekli SELECT [DISTINCT|ALL] {\*| liste}

FROM [tablo ismi][, ikinci tablo ismi]…. [WHERE sart]

[GROUP BY kolon listesi] [HAVING sart]

[ORDER BY kolon listesi] seklindedir. SLECT ve FROM herzaman için kullanilir, digerleri istenirse kullanilir. SELECT komutundan sontra çiktida görmek istedigimiz kolon adlari bulunur. Tablonun tüm kolonlarini görmek istersek \* kullanabiliriz. FROM komutundan sonra seçilen tablonun (veya tablolarin) isimleri yer alir. Tablo

isimleri ‘,’ ile ayrilir. ALL veya DISTINCT anahtarlari eger tekrar varsa bunlarin verilip verilmeyecegini bildirir. DISTINCT secildiginde tüm tekrarlar elimine edilir. ALL seçildiginde tüm girdiler tekrar da olsa verilir. WHERE deyimi javadaki if deyimi gibidir. Sart olusturur. WHERE ile birlikte kullanilan bazi mantik islemleri sunlardir :

AND : ve OR : veya

NOT : olumsuzluk

BETWEEN : arasinda (genellikle AND ile birlikte BETWEEN 20 AND 30 gibi kullanilir) IN : içinde

LIKE : gibi,benzer(karecter degiskenleri için esittir yerine geçen bir yapidir, kullanilirken bazi kisimlar bos birakilabilir LIKE ‘%ali%’ içinde ali geçen her seyi kapsar.)

NULL : bos

ANY : herhangibiri ALL : hepsi EXIST : varsa

Rakamsal karsilastirmalarda

= : esittir

<> : esit degildir

> : büyüktür

< : küçüktür

>= : büyüktür ve esittir.

<= : küçüktür ve esittir. Kullanilabilir.

ORDER BY siraya sokar ASC kelmesiyle birlikte kullanildiginda küçükten büyüge dogru, DESC kelimesiyle birlikte kullanildiginda büyükten küçüge dogru siralar.

GROUP BY belli bir kolona göre veriyi tasnif edebilir. HAVING deyimi GROUP by deyimiyle birlikte kullanilir be GROUP By terimi için ilave sart olusturur.

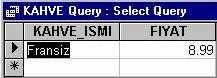
Select deyiminibir kaç örnekle açalim:

**SELECT** KAHVE\_ISMI, FIYAT

### FROM KAHVE

**WHERE** KAHVE\_ISMI like "Fransiz" deyimi

### 13002.JPG



**Sekil 13.2 Query sonuçlarinin access veri tabaninda görünümü**

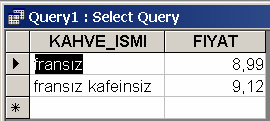
sonucunu verir.

**SELECT** KAHVE\_ISMI, FIYAT

### FROM KAHVE

**WHERE** KAHVE\_ISMI like "\*Fransiz\*"

### 13003.JPG



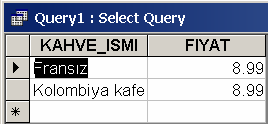
**Sekil 13.3 Query sonuçlarinin access veri tabaninda görünümü**

sonucunu verir.

SELECT KAHVE\_ISMI, FIYAT FROM KAHVE

WHERE FIYAT > 8.5 AND FIYAT < 9.0

### 13003.JPG



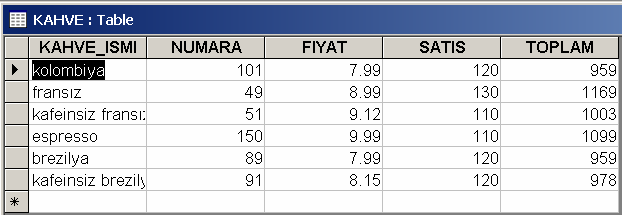
**Sekil 13.3 SELECT..WHERE querry sonuçlarinin acsess veri tabaninda görünümü**

Fiyati 8.5 ila 9.0 arasinda olan kahvelerin listesi sonucunu verecektir. UPDATE KAHVE

SET TOPLAM=FIYAT\*SATIS;

Komutu kullanarak TOPLAM kolonunun degerlerini hesaplayabiliriz. Tablonun yeni hali :

### 13004.JPG



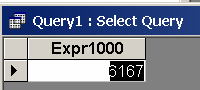
**Sekil 13.4 UPDATE querry sonuçlarinin acsess veri tabaninda görünümü**

Sonucunu verir.

SQL de direk olarak hesap yapabilen fonksiyonlar da mevcuttur. Örnegin SUM(kolon ismi) degerleri toplar, MAX, maksimum degeri verir, MIN, minumum degeri verir, COUNT toplam sütun sayisini sayar. Bu fonksiyonlara örnek olarak SUM fonksiyonunu kullanalim :

SELECT SUM(TOPLAM) FROM KAHVE;

### 13005.JPG



**Sekil 13.5 querry sonuçlarinin acsess veri tabaninda görünümü**

Veri tabanini siraya koymak için SELECT ..ORDER BY komutunu kullanabiliriz

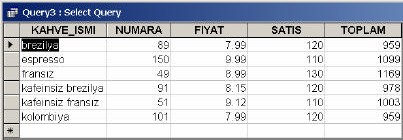
ASC küçükten büyüge DESC büyükten küçüge siraya koyar. Örnegin veri tabani tablomuz kahveyi KAHVE\_ISMI’ne göre siralamak istersek

SELECT \* FROM KAHVE

ORDER BY KAHVE\_ISMI ASC;

SQL komutunu kullanabiliriz. Sonuç :

### 13006.JPG



**Sekil 13.6 SELECT..ORDERBY querry sonuçlarinin acsess veri tabaninda görünümü**

### JAVA VE SQL BAGLANTISI

Simdi bu islemleri direk olarak java programlama dili üzerinden nasil yapacagimizi inceleyelim.

Java dilinde data base tanimlamak için önce database ile bir baglanti kurulmalidir. Burada örnek olarak MS Access ile Window ortaminda nasil baglanti tanimliyacagimizi step-step verelim

Window (95-98) ortaminda access database ile baglanti kurulmasi: prog (veya baska bir )isimli bir access db programini aç ve save et baslat,ayarlar,denetim masasi,32 bit ODBC

User DSN sayfasindan MS Access Database seç ve Add dügmesine bas Microsoft Access Driver(\*.mbd) seç, Son dügmesini tikla

data source name olarak prog (veya baska bir isim) seç

Description alanina JDBC - odbc database yaz, select dügmesine bas prog DB dosyasinin oldugu directory 'i seç

OK dügmesine bas, Advanced dügmesinden login ve password da tanimliyabilirsiniz.

Windows 200-NT ortaminda access database ile baglanti kurulmasi (benim winows 2000nim ingilizce, türkçe olursa win 98’e parelel isimler olmali)

prog (veya baska bir )isimli bir access db programini aç ve save et start,settings,control panel,Administrative Tools, Data Sources

ODBC Data Source Administrator, User DSN sayfasindan MS Access Database seç ve Add dügmesine bas Microsoft Access Driver(\*.mbd) seç, Son dügmesini tikla

data source name olarak prog (veya baska bir isim) seç

Description alanina JDBC - odbc database yaz, select dügmesine bas prog DB dosyasinin oldugu directory 'i seç

OK dügmesine bas, Advanced dügmesinden login ve password da tanimliyabilirsiniz.

### Program 13.1 kahve veri tabani dosyasini tanmlayan CreateKahve.java programi

import java.sql.\*; import java.io.\*; import Text;

public class CreateKahve{

public static void main(String args[]) throws IOException

{

String url = "jdbc:odbc:prog"; Connection con;

String createString;

createString = "create table KAHVE " + "(KAHVE\_ISMI varchar(32), " + "NUMARA int, " +

"FIYAT float, " + "SATIS int, " + "TOPLAM int)";

Statement stmt; try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

} catch(java.lang.ClassNotFoundException e) {

System.err.print("ClassNotFoundException: "); System.err.println(e.getMessage());

}

try {

stmt.close();

con.close();

con = DriverManager.getConnection(url,"Login", "Password"); stmt = con.createStatement(); stmt.executeUpdate(createString);

}

}

Buradaki

} catch(SQLException ex)

{System.err.println("SQLException: " + ex.getMessage());}

### Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

deyimi java JdbcOdbcDriver ile baglanti açmaktadir. Eger varsa kendi database driver'inizi kullanabilirsiniz Bunun size getirecegi en büyük avantaj daha önce yazilmis bir database programina erismeyi mümkün kilmasidir. JDBC driverlerinin en son listesi

http:/java.sun.com/products/jdbc/jdbc.drivers.html sitesinden ögrenilebilir.

### String url = "jdbc:odbc:prog";

**Connection con = DriverManager.getConnection(url,"Login", "Password");**

deyimi ise daha önce ismini prog olarak tanimladigimiz (ismi siz database kaynak kotunun ismine göre tanimliyabilirsiniz) prog database'ine ulasmamizi saglamaktadir.

Veri tabani sonuçlarini almak için ResultSet kullanilir. ResultSet metodunun baslica alt metodlari : **ResultSet executeQuerry(String sql) throws SQLException** : SQL deyimini çalistirir ve sonuçlariniResultSET tipi Object olarak aktarir.

**int executeUpdate(String sql) throws SQLException :** bir sonuç tablosu vermeyen SQL terimlerini çalistirir. Komuttan etkilenen satir sayisini iletir.

**Boolean execute(String sql) throws SQLException :** genel maksatli SQL deyimi çalistirici

### ResultSet getResultSet() Throws SQLException: int getUpdateCount() throws SQLException : Boolean getMoreResults() : throws SQLException :

Kahve verisini veri tabanina yazdiran ve select deyimi kullanarak okutan program :

### Program 13.2 kahve veri tabani dosyasina veri girisi saglayan InsertKahve.java programi

import java.sql.\*;

public class InsertKahve {

public static void main(String args[]) { String url = "jdbc:odbc:prog";

Connection con; Statement stmt;

String query = "select KAHVE\_ISMI, FIYAT from KAHVE"; try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

} catch(java.lang.ClassNotFoundException e)

{

System.err.print("ClassNotFoundException: "); System.err.println(e.getMessage());

}

try {

con = DriverManager.getConnection(url, "myLogin", "myPassword");

stmt = con.createStatement(); stmt.executeUpdate("insert into KAHVE " + "values('kolombiya',101, 7.99, 120, 0)"); stmt.executeUpdate("insert into KAHVE " + "values('fransiz', 49, 8.99, 130, 0)"); stmt.executeUpdate("insert into KAHVE " + "values('kafeinsiz fransiz', 51, 9.12, 110, 0)"); stmt.executeUpdate("insert into KAHVE " + "values('espresso', 150, 9.99, 110,0)"); stmt.executeUpdate("insert into KAHVE " + "values('brezilya', 89, 7.99, 120, 0)"); stmt.executeUpdate("insert into KAHVE " + "values('kafeinsiz brezilya', 91, 8.15, 120, 0)");

ResultSet rs = stmt.executeQuery(query); System.out.println("Coffee Break Coffees and Prices:"); while (rs.next()) {

String s = rs.getString("KAHVE\_ISMI"); float f = rs.getFloat("FIYAT"); System.out.println(s + " " + f);

}

stmt.close();

con.close();

} catch(SQLException ex) {

System.err.println("SQLException: " + ex.getMessage());

}

}

}

Program çiktisi :

Coffee Break Coffees and Prices: kolombiya 7.99

fransiz 8.99

kafeinsiz fransiz 9.12

Espresso 9.99

Brezilya 7.99

Kafeinsiz brezilya 8.15

Ayni programin simdi de applet olarak yazilmis bir esdegerine göz atalim.

### Program 13.3 KahveApplet.java programi InsertKahve programinin applet esdegeridir. programin thread olarak yazildigina dikkat ediniz.

import java.applet.Applet; import java.awt.Graphics; import java.util.Vector; import java.sql.\*;

public class KahveApplet extends Applet implements Runnable { private Thread worker;

private Vector queryResults;

private String message = "Baslatiliyor...";

public synchronized void start() {

// "start" her cagirildiginda

// kahve database query tekrar ‡agirilacaktir. if (worker == null) {

message = "Veritabani ile baglanti kuruluyor";

worker = new Thread(this); worker.start();

}

}

public void run() {

String url = "jdbc:odbc:prog";

String query = "select KAHVE\_ISMI, FIYAT from KAHVE";

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

} catch(Exception ex) {

setError("Can't find Database driver class: " + ex); return;

}

try {

Vector results = new Vector();

Connection con = DriverManager.getConnection(url, "myLogin", "myPassword");

Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(query); while (rs.next()) {

String s = rs.getString("KAHVE\_ISMI"); float f = rs.getFloat("FIYAT");

String text = s + " " + f; results.addElement(text);

}

stmt.close();

con.close(); setResults(results);

} catch(SQLException ex) { setError("SQLException: " + ex);

}

}

public synchronized void paint(Graphics g) { if (queryResults == null)

{

g.drawString(message, 5, 50); return;

}

// Display the results. g.drawString("Kahve fiyatlari : ", 5, 10); int y = 30;

java.util.Enumeration enum = queryResults.elements(); while (enum.hasMoreElements()) {

String text = (String)enum.nextElement(); g.drawString(text, 5, y);

y = y + 15;

}

}

private synchronized void setError(String mess) { queryResults = null;

message = mess;

worker = null;

// And ask AWT to repaint this applet. repaint();

}

private synchronized void setResults(Vector results) { queryResults = results;

worker = null; repaint();

}

}

Bu programin çiktisi, bir önceki programda oldugu gibidir.

### 13007.JPG



**Sekil 13.7 KahveApplet.html programinin Netscape browserdaki çiktisi**

Statement interface’inin alt metodlarina biraz daha detayli bakalim. Statement SQL komutlarini veritabani programina gönderen ve sonuçlarini geri çagiran ana siniftir. SQL komut istemi çesitli metodlar üzerinden yapilabilir

### ResultSet executeQuery(String sqldeyimi) throws SQLException

Bu metod çalistirildiginda tek bir SQL query çalistirilir ve sonuçlari ResultSet sinifindan geri alinir.

### int executeUpdate(String sqldeyimi) throws SQLException

Bu metod çalistirildiginda tek bir SQL Query çalistirilir, geriye sonuç iletmez, sadece kaç satir verinin bu deyimden etkilendigini belirtir.

### boolean execute(String sqldeyimi) throws SQLException

Bu metod çalistirildiginda genel bir SQL Query deyimi çalistirilirilabilir. Bilhassa çalistirilan SQL deyiminin ne yapabilecegi tam olarak bilinmiyorsa veya degisken sonuçlar elde ediliyorsa kullanilabilir. Bu deyim sonucu iletilen veriler GetResultSet(),getUpdateCount() ve getMoreResults() metodlari kullanilarak alinabilir.

### ResultSet getResultSet() throws SQLException

En son çikmis olan sonuçlari iletir. Eger hiç bir sonuç mevcut degildse null sonucu verir. Bir kere okunduktan sonra sonuçlar silinir.

### int getUpdateCount() throws SQLException

Update, insert ve delet deyimlerinin sonuçlarinda aç satirin etkilendigini verir. Eger SQL tarafiindan bir sonuç seti verilmisse veya herhangi bir veri degistirilmemisse –1 degerini verir.

### int getMoreResults() throws SQLException

Birden fazla sonuçveya update sayimi yapilan durumlarda bir sonraki deger setine geçer. Bu metod ayni zamanda daha önce açilmis olan tüm ResultSet deyimi degerlerini kapatir.

ResultSet sinifi temel olarakveritabani dosyasindandegerleri okumak için kullanilir. SQL veri türleri çesitli SQL veritabanlarinin varligindan dolayi oldukça çesitlilik gösterir. Temel olak kullanilan metodlar ve olarin SQL veri larsiliklari sunlardir :

**getByte : TINYINT**, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR

**getsHORT :** TINYINT, **SMALLINT**, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR

**getInt :** TINYINT, SMALLINT, **INTEGER**, BIGINT, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR

**getLong :** TINYINT, SMALLINT, INTEGER, **BIGINT**, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR

**getFloat :** TINYINT, SMALLINT, INTEGER, BIGINT, **REAL**, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR

**getDouble :** TINYINT, SMALLINT, INTEGER, BIGINT, REAL, **FLOAT, DOUBLE**, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR

**getBigDecimal :** TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, **DECIMAL, NUMERIC**, BIT, CHAR, VARCAHR, LONGVARCHAR

**getBoolean :** TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, **BIT**, CHAR, **VARCAHR**, LONGVARCHAR

**getString :** TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, **CHAR**, VARCAHR, LONGVARCHAR

**getBytes : BINARY,VARBINARY**,LONGVARBINARY

**getDate :** CHAR, VARCAHR, LONGVARCHAR**,** BINARY, **DATE**, TIMESTAMP

**getTime :** CHAR, VARCAHR, LONGVARCHAR**,** BINARY, **TIME**, TIMESTAMP

**getTimeStamp :** CHAR, VARCAHR, LONGVARCHAR**,** BINARY,DATE,TIME,**TIMESTAMP getAsciiStream :** CHAR, VARCAHR, **LONGVARCHAR,** BINARY, VARBINARY, LONGVARBINARY **getUnicodeStream :** CHAR, VARCAHR, **LONGVARCHAR,** BINARY, VARBINARY,

LONGVARBINARY **getBinaryStream :** BINARY, VARBINARY, **LONGVARBINARY**

**getObject :** TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, DECIMAL, NUMERIC, BIT, CHAR, VARCAHR, LONGVARCHAR**,** BINARY, VARBINARY, LONGVARBINARY, DATE, TIME, TIMESTAMP

kalin harfle yazili olan SQL tipleri bu metod için en uygun tiplerdir. Metodlar SQL tablo kolonuna kolon ismi veya indeks numarasi kullanarak ulasabilirler. Örnegin getDouble metodu

double getDouble(int sütunindeksi) throws SQLException; double getDouble(String sütunismi) throws SQLException; seklinde tanimlanmistir.

GetObject metodu her türlü degisken için kullanilabilir.

ResultSetMetaData sinifi ResultSet’te tanimlanan bilgi ile ilgili ilave sütun bilgisi aktarnmak için kullanilir.

ResultSet rs=stmt.executeQuery(“SELECT Toplam FROM KAHVE”); ResultSetMetaData rsmd=rs.getMetaData();

Seklinde tanimlanir. Önemli metodlari :

int getColumnCount() throws SQLException : kaç sütunluk bir set oldugunu tanimlar String getColumnLabel(int column) throws SQLException :

String getColumnName(int column) throws SQLException : veri tabani sütun ismini verir String getTableName(int column) throws SQLException : veri tabani tablosunun ismini verir

Booean isReadOnly(int sütun) throws SQLException : sütunun sadece okunabilir olarak izni olup olmadigini verir.

Su ana kadar gelistirdigimiz veri tabani programlari uygulamaya özel idi. Simdi biraz daha genel bir uygulama olusturalim. Bu uygulamada swing JTable sinifini kullanacagiz. Bu programda olusturdugumuz ilk sinif QuerryTableModel sinifi tabloya girmek için qury sonuçlarini tablo giris formatinda hazirlamaktadir. Bu metodun bilgi islerken Vector sinifini kullandigina dikkatinizi çekeriz. Vector sinifi hakkinda daha detayli bilgi için ilgili bölüme bakiniz.

### Program 13.4 QueryTableModel.java programi

import java.sql.\*;

import java.io.\*; import java.util.Vector; import javax.swing.\*;

import javax.swing.table.\*;

public class QueryTableModel extends AbstractTableModel

{

Vector cache; int colCount; String[] headers; Connection db;

Statement statement;

String currentURL;

public QueryTableModel()

{

cache=new Vector();

//new gsl.sql.driv.Driver(); try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{

System.err.println("Class (sinif) bulunamadi hatasi : "); System.err.println(e.getMessage());

}

}

public String getColumnName(int i) {return headers[i];} public int getColumnCount() {return colCount;}

public int getRowCount() {return cache.size();}

public Object getValueAt(int row,int col)

{

return ((String[])cache.elementAt(row))[col];

}

public void setHostURL(String url)

{

if(url.equals(currentURL))

{return;} closeDB(); initDB(url); currentURL=url;

}

public void setQuery(String q)

{

cache= new Vector(); try{

ResultSet rs=statement.executeQuery(q); ResultSetMetaData meta=rs.getMetaData(); colCount=meta.getColumnCount(); headers=new String[colCount];

for (int h=1;h<=colCount;h++)

{

headers[h-1]=meta.getColumnName(h);

}

while(rs.next())

{

String[] record=new String[colCount]; for(int i=0;i<colCount;i++)

{record[i]=rs.getString(i+1);} cache.addElement(record);

} //while'in sonu fireTableChanged(null);

} //try'in sonu catch(Exception e)

{

cache=new Vector(); e.printStackTrace();

}

} // setQuery sonu

public void initDB(String url)

{

try {

db=DriverManager.getConnection(url); statement=db.createStatement();

}

catch(Exception e)

{

System.out.println("DataBase baslatilamadi"); e.printStackTrace();

}

} //initDB sonu

public void closeDB()

{

try {

if(statement!= null) {statement.close();} if(db != null) {db.close();}

}

catch(Exception e)

{

System.out.println("database kapatilamadi"); e.printStackTrace();

}

} //closeDB sonu

}

Ikinci sinifimiz (Tabloyu gösterme sinifi) olan DataBaseTest programi ise girdi alanindan SQL komutunu ve veritabani baglantisini okumakta ve SQL sonuçlarini tabloda vermektedir.

### Program 13.5 DatabaseTest.java programi

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

import javax.swing.table.\*;

public class DatabaseTest extends JFrame

{

JTextField hostField; JTextArea queryField; QueryTableModel qtm; public DatabaseTest()

{

super("Database Örnek Penceresi"); addWindowListener(new BasicWindowMonitor()); setSize(350,200);

qtm=new QueryTableModel(); JTable table=new JTable(qtm);

JScrollPane scrollpane=new JScrollPane(table); JPanel p1=new JPanel();

p1.setLayout(new GridLayout(3,2)); p1.add(new JLabel(" url ismini giriniz :"));

p1.add(hostField=new JTextField("jdbc:odbc:prog"));

p1.add(new JLabel("SQL(Sequential Query Language) deyiminizi giriniz :")); p1.add(queryField=new JTextArea("select \*\n from KAHVE"));

p1.add(new JLabel("Query'nizi göndermek icin dügmeye basiniz:")); JButton jb=new JButton("SQL deyimini calistir"); jb.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

qtm.setHostURL(hostField.getText().trim()); qtm.setQuery(queryField.getText().trim());

}

}); // jb.addActionlistener p1.add(jb);

getContentPane().add(p1,BorderLayout.NORTH); getContentPane().add(scrollpane,BorderLayout.CENTER);

} //DatabaseTest()

public static void main(String args[])

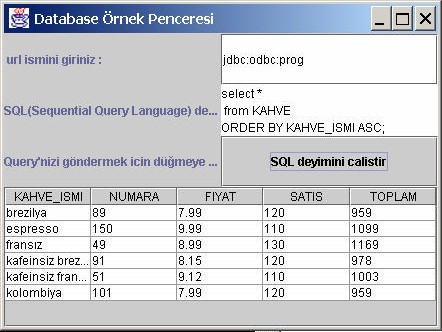
{

DatabaseTest tt=new DatabaseTest(); tt.setVisible(true);

}

}

### 13008.JPG



**Sekil 13.8 DatabaseTest.java programinin JFrame’de ve JTable kullanilarak olusturulan çiktisi**

### ALISTIRMALAR

**1.** Program örnegi database dosyasini girdi dosyasi olarak kullanan ve içindeki sayilarin ortalamasini alan H9O2.java programi : Bu programda sayilar dosyadan okunmakta, ortalamasi hem querry dilinde (SQL) hemde java dilinde hesaplanmaktadir. Programi inceleyiniz.

### Program 13.6 H9O2.java, database dosyasindan alinan bilgilerle database dosyasinin girdi dosyasi olarak kullanilmasi

import java.sql.\*; public class H9O2

{

public static void main(String[] args)

{

String url="jdbc:odbc:h9o2"; Connection con;

Statement stmt;

String query2="select AVG(SAYI) as T from H9O2"; String query1="select SAYI from H9O2";

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{ System.err.println(e.getMessage()); } try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

ResultSet rs= stmt.executeQuery(query1); int j=0;

double toplam=0; while (rs.next())

{

double sayi= rs.getDouble("SAYI"); toplam+=sayi; System.out.println("Sayi("+(j++)+") = "+sayi);

}

System.out.println("metot 1 ortalamalar query dilinde hesaplandi"); rs= stmt.executeQuery(query2);

while (rs.next())

{

double i= rs.getDouble("T"); System.out.println("Ortalama = "+i);

}

System.out.println("metot 2 ortalamalar java dilinde hesaplandi"); double ortalama=toplam/j;

System.out.println("metot 2 ortalama : "+ortalama); stmt.close();

con.close();

}

catch (SQLException e)

{

System.err.println(e.getMessage());

} //catch

} //main

} //class

program H9O2.java çiktisi : Sayi(0) = 1.232

Sayi(1) = 1.654

Sayi(2) = 3.562462

Sayi(3) = 2.6546

Sayi(4) = 2.64784

Sayi(5) = 1.4262

Sayi(6) = 2.98963

Sayi(7) = 3.76487648

Sayi(8) = 1.7368365

Sayi(9) = 4.2222487

Sayi(10) = 1.86327653

Sayi(11) = 1.973836

Sayi(12) = 3.687522

Sayi(13) = 1.7637836

Sayi(14) = 2.0832863

metot 1 ortalamalar query dilinde hesaplandi Ortalama = 2.4841598739999995

metot 2 ortalamalar java dilinde hesaplandi metot 2 ortalama : 2.4841598739999995

============================================================================

**2.** H9OD2\_2000

prog.mdb acsess veri tabaninda H9OD2\_2000 veri tabani tablosunu açiniz. Bu dosyada x ve y ve y2 basliklari altinda üç kolon yaratiniz. x ve y kolonlarina

|  |  |
| --- | --- |
| 1 | 1 |
| 1 | 4 |
| 2 | 9 |
| 3 | 16 |
| 4 | 25 |
| 5 | 36 |

veri setini giriniz. H9OD2\_2000.java programinda veri tabani doyasindan okutup ekranda gösteriniz. Ayni zamanda y2 kolonuna x\*x degerlerini yazdiriniz.

### Program 13.7 H9O2\_2000.java, database dosyasindan alinan bilgilerle database dosyasinin girdi dosyasi olarak kullanilmasi

import java.sql.\*;

public class H9OD2\_2000

{

public static void main(String[] args)

{

String url="jdbc:odbc:prog"; Connection con;

Statement stmt;

String query1="select x,y from H9OD2\_2000";

String query2="UPDATE H9OD2\_2000 SET y2=x\*x;"; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{ System.err.println(e.getMessage()); } try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

ResultSet rs= stmt.executeQuery(query1); while (rs.next())

{

double xi= rs.getDouble("x"); double yi= rs.getDouble("y"); System.out.println(""+xi+" "+yi);

}

// SQL komutu kullanarak y\*y degerini veri tabani dosyasina yazdir. int jj = stmt.executeUpdate(query2); System.out.println(""+jj+"kolon degistirildi");

stmt.close();

con.close();

}

catch (SQLException e)

{

System.err.println(e.getMessage());

} //catch

} //main

} //class

### Program çiktisi : 1.0 1.0

**2.0 4.0**

**3.0 9.0**

**4.0 16.0**

**5.0 25.0**

**6.0 36.0**

**7.0 49.0**

**7 kolon degistirildi**

**program sonunda H9OD2\_2000 veritabani dosyasi:**

**Program 13.8 H9O2\_2000a.java, database dosyasindan alinan bilgilerle database dosyasinin girdi dosyasi olarak kullanilmasi, sonuçlarin tablo olarak gösterimi**

import java.sql.\*; import javax.swing.\*; import java.awt.\*; import java.util.Vector;

public class H9OD2\_2000a extends JFrame

{

ResultSet rs; ResultSetMetaData rsm; public H9OD2\_2000a()

{

super("Data base sonuçlari"); Container c=getContentPane(); String url="jdbc:odbc:prog"; Connection con;

Statement stmt;

String query1="select \* from H9OD2\_2000";

String query2="UPDATE H9OD2\_2000 SET y2=x\*x;"; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{ System.err.println(e.getMessage()); } try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

int jj=stmt.executeUpdate(query2); rs= stmt.executeQuery(query1); rsm=rs.getMetaData();

int j=rsm.getColumnCount(); String[] baslik=new String[j]; for (int h=1;h<=j;h++)

{

baslik[h-1]=rsm.getColumnName(h);

}

JTable jt=new JTable(aa(j),baslik); JScrollPane jsp=new JScrollPane(jt); c.add(jsp,BorderLayout.CENTER); stmt.close();

con.close();

}

catch (SQLException e)

{

System.err.println(e.getMessage());

} //catch

} //kurucu metod

public String[][] aa(int j) throws SQLException

{

int n=0;

String a[][]=new String[8][j]; while(rs.next())

{

for(int k=0;k<j;k++)

{

a[n][k]=""+rs.getDouble(rsm.getColumnName(k+1));

} n++;

}

String b[][]=new String[n][j]; for(int i=0;i<n;i++)

{

for(int k=0;k<j;k++)

{

b[i][k]= a[i][k];

}

}

return b;

}

public static void main(String[] args)

{

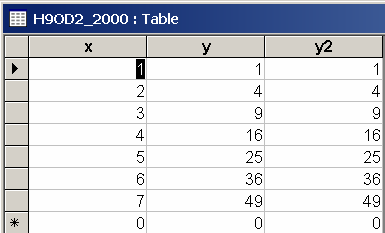
H9OD2\_2000a pencere= new H9OD2\_2000a(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

} //class

### 13009.JPG



**Sekil 13.9 H9O2\_2000a.java çiktisi, database dosyasindan alinan bilgilerle database dosyasinin girdi dosyasi olarak kullanilmasi, çikti ortami olarak tablo kullanan bir alternatifi**

**Program 13.10 H9O2\_2000b.java, database dosyasindan alinan bilgilerle database dosyasinin girdi dosyasi olarak kullanilmasi, çikti ortami olarak QueryTableModel sinifini kullanarak olusturdugu tabloyu kullanan bir alternatif**

import java.sql.\*; import javax.swing.\*; import java.awt.\*;

import QueryTableModel;

// Vector (Querytable model’da ) kullaniyor public class H9OD2\_2000b extends JFrame

{

public H9OD2\_2000b()

{

super("Data base sonuçlari"); Container c=getContentPane(); String url="jdbc:odbc:prog";

QueryTableModel qtm=new QueryTableModel(); qtm.setHostURL(url);

String query1="select x,y from H9OD2\_2000"; qtm.setQuery(query1);

JTable jt=new JTable(qtm); JScrollPane jsp=new JScrollPane(jt); c.add(jsp,BorderLayout.CENTER);

} //kurucu metod

public static void main(String[] args)

{

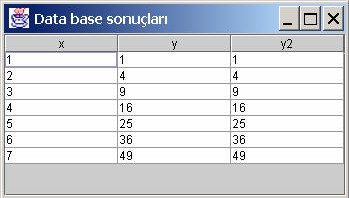
H9OD2\_2000b pencere= new H9OD2\_2000b(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

} //class

### 13010.JPG



**Sekil 13.10 H9O2\_2000b.java, database dosyasindan alinan bilgilerle database dosyasinin girdi dosyasi olarak kullanilmasi, çikti ortami olarak QueryTableModel sinifini kullanarak olusturdugu tabloyu kullanan bir alternatif**

============================================================================

**3.** Daha önce de kullandigimiz prog veritabaninda e isimli bir tablo olusturunuz, bu tablodaki e sütununa gerçek sayi veri giriniz. Diger bir tablo olarak e tablosunda e sütununu olusturunuz, bu tabloya da gerçek sayi degerler giriniz. Iki veri tabani tablosundan degerleri okuyup, bilesim kümesini hesaplayiniz ve yarattiginiz ed tablosuna ed degerleri olarak yazdiriniz.

### Program 13.10 H10O1\_2000b.java

import java.sql.\*; import set; import java.util.\*;

import java.io.\*;

import javax.swing.JOptionPane;

public class H10OD1\_2000

{

public static void main(String[] args)

{

String s=""; double e,d,ed; Vector ve,vd,ved; ve =new Vector(); vd =new Vector();

ved =new Vector();

String url="jdbc:odbc:prog"; Connection con;

Statement stmt;

String query1="select \* from e"; String query2="select \* from d"; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException er)

{ System.err.println(er.getMessage()); } try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

ResultSet rs= stmt.executeQuery(query1); s+="e veritabani:\n";

while (rs.next())

{

e = rs.getDouble("e"); ve.addElement(""+e); s+=""+e+"\n";

}

JOptionPane.showMessageDialog(null,s); rs= stmt.executeQuery(query2);

s+="d veritabani: \n"; while (rs.next())

{

d= rs.getDouble("d"); vd.addElement(""+d); s+=""+d+"\n";

}

JOptionPane.showMessageDialog(null,s); ved=set.bilesim(ve,vd);

s+="bilesim vektörü :\n"; Enumeration n=ved.elements(); int i=1; while(n.hasMoreElements())

{

String s1=(String)n.nextElement();

String s2="INSERT INTO H10OD1\_2000 VALUES('"+(i++)+"',"+s1+")";

s+=s1+"\n"; stmt.executeUpdate(s2);

}

stmt.close();

con.close();

JOptionPane.showMessageDialog(null,s); System.exit(0);

}

catch (SQLException er)

{

System.err.println(er.getMessage());

} //catch

} //main

} //class

### Program 13.11 set.java

// Bu program vector ve Enumeration siniflarini

// Kullanmaktadir. Bilesim ve kesisim kümelerini hesaplar

// Ayni zamanda StringTokeniser sinifini kullanir. import java.util.\*;

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

public class set

{

public static Vector bilesim(Vector list1, Vector list2)

{

Vector bilesimVectoru = new Vector(); Object s1,s2;

Enumeration n1=list1.elements(); Enumeration n2=list2.elements(); while(n1.hasMoreElements())

{

s1=n1.nextElement(); bilesimVectoru.addElement(s1);

}

while(n2.hasMoreElements())

{

s2=n2.nextElement(); if(!bilesimVectoru.contains(s2))

bilesimVectoru.addElement(s2);

}

return bilesimVectoru;

}

public static Vector kesisim(Vector list1,Vector list2)

{

Vector kesisimVectoru = new Vector(); Object s;

Enumeration n = list1.elements(); while(n.hasMoreElements())

{

s=n.nextElement(); if(list2.contains(s))

kesisimVectoru.addElement(s);

}

return kesisimVectoru;

}

public static Vector fark(Vector list1, Vector list2)

{

Vector subst=new Vector(); Object ans;

Enumeration enum= list1.elements(); while(enum.hasMoreElements())

{

ans=enum.nextElement(); if(list2.contains(ans))

continue; else

subst.addElement(ans);

}

return subst;

}

public static Vector bilesimeksikesisim(Vector list1, Vector list2)

{

Vector UMI=new Vector(); Object ans1, ans2;

Enumeration enum1= list1.elements(); Enumeration enum2= list2.elements();

while(enum1.hasMoreElements())

{

ans1=enum1.nextElement(); if(list2.contains(ans1))

continue; else

UMI.addElement(ans1);

}

while(enum2.hasMoreElements())

{

ans2 = enum2.nextElement(); if(list1.contains(ans2))

continue; else

UMI.addElement(ans2);

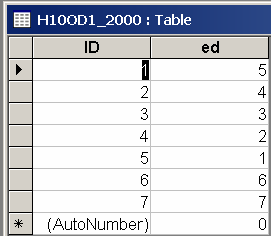
}

return UMI;

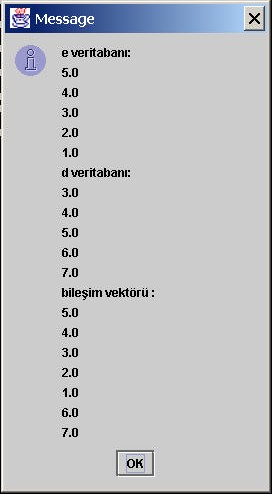
}

}

### 13011.JPG



**Sekil 13.11 H10OD1\_2000.java, programinin olusturdugu H10OD1\_2000 veritabani dosyasi 13012.JPG**



**Sekil 13.12 H10OD1\_2000.java, programinin sonuçlarinin JoptionPane çiktisi olarak görüntülenmesi**

1. Daha önce de kullandigimiz prog veritabaninda e isimli bir tablo olusturunuz, bu tablodaki x sütununa gerçek sayi veri giriniz. Diger bir tablo olarak e tablosunda x sütununu olusturunuz, bu tabloya da gerçek sayi degerler giriniz. Iki veri tabani tablosundan degerleri okuyup, bilesim kümesini hesaplayiniz ve yarattiginiz H10OD1\_2000 tablosuna x degerleri olarak yazdiriniz. Proseste boyutlu degiskenleri vektör üzerinden tasiyiniz

### Program 13.12 H10OD1\_2000.java

Bu program Gebze Yüksek teknoloji enstitüsü Java programla dili dersi, 2000 Güz dönemi ögrencilerinden Deniz KAYA tarafindan yazilmistir.

// Deniz KAYA

// Hafta 10 Ödevi

// Tablolardan vektore, vektor bileþiminden tabloya

// yazdýrma iþlemi.

import java.io.\*; import java.util.\*; import java.sql.\*;

class SetA

{

public Vector tablodanIntYukle(String d)

{

// ismi d deðiskeni ile gelen tablodan

// Rakamlari Vektor Icine Yukle

Vector v = new Vector(); String url="jdbc:odbc:prog"; Connection con;

Statement stmt;

String query1="select x from "+d; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{

System.err.println(e.getMessage());

}

try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

ResultSet rs= stmt.executeQuery(query1); while (rs.next())

{

int xi= rs.getInt("x"); v.addElement(new Integer(xi));

}

stmt.close();

con.close();

}

catch (SQLException e)

{

System.err.println(e.getMessage());

} //catch return v;

} //

public Vector bilesim(Vector list1, Vector list2)

{

Vector bilesimVectoru = new Vector(); Object s1,s2;

Enumeration n1=list1.elements(); Enumeration n2=list2.elements(); while(n1.hasMoreElements())

{

s1=n1.nextElement(); bilesimVectoru.addElement(s1);

}

while(n2.hasMoreElements())

{

s2=n2.nextElement(); if(!bilesimVectoru.contains(s2))

bilesimVectoru.addElement(s2);

}

return bilesimVectoru;

}

public Vector kesisim(Vector list1,Vector list2)

{

Vector kesisimVectoru = new Vector(); Object s;

Enumeration n = list1.elements(); while(n.hasMoreElements())

{

s=n.nextElement(); if(list2.contains(s))

kesisimVectoru.addElement(s);

}

return kesisimVectoru;

}

public Vector fark(Vector list1, Vector list2)

{

Vector subst=new Vector(); Object ans;

Enumeration enum= list1.elements(); while(enum.hasMoreElements())

{

ans=enum.nextElement(); if(list2.contains(ans))

continue; else

subst.addElement(ans);

}

return subst;

}

public Vector bilesimeksikesisim(Vector list1, Vector list2)

{

Vector UMI=new Vector(); Object ans1, ans2;

Enumeration enum1= list1.elements(); Enumeration enum2= list2.elements();

while(enum1.hasMoreElements())

{

ans1=enum1.nextElement(); if(list2.contains(ans1))

continue; else

UMI.addElement(ans1);

}

while(enum2.hasMoreElements())

{

ans2 = enum2.nextElement(); if(list1.contains(ans2))

continue; else

UMI.addElement(ans2);

}

return UMI;

}

}

public class H10OD1\_2000 {

public static void main( String arg[]) throws IOException { SetA A = new SetA();

Vector list1 = A.tablodanIntYukle("d"); // list1 'e (d) tablosunun deðerlerinin yüklüyor.

Vector list2 = A.tablodanIntYukle("e"); // list2 'ye (e) tablosunun deðerlerinin yüklüyor. Vector bilesim = A.bilesim(list1,list2); // listeleri birleþtir.

Enumeration e = bilesim.elements();

// Aþagýdaki kesim bilesim kumesini tabloya ve ekrana yazdýrýr. String url="jdbc:odbc:prog";

Connection con;

Statement stmt;

String query1="select x from H10OD1\_2000"; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException ex)

{

System.err.println(ex.getMessage());

}

try

{

con=DriverManager.getConnection(url); stmt=con.createStatement(); while(e.hasMoreElements())

{

Integer s = (Integer)e.nextElement();

stmt.executeUpdate("insert into H10OD1\_2000 values("+s+")"); System.out.println(s.toString());

}

System.out.println("\n Ýþleminiz basayla tamamlandi...."); stmt.close();

con.close();

}

catch (SQLException ex)

{

System.err.println(ex.getMessage());

} //catch

} // Main

}// H10OD1\_2000

1. Program 4 de H10OD1\_2000 veri tabani dosyasina yazilan veriyi okuyunuz. Binary Tree yapilarini kullanarak siraya koyunuz ve H10OD2\_2000 dosyasina yazdiriniz.

### Program 13.13 Tree.java

class TreeNode

{

TreeNode sol; int data; TreeNode sag;

public TreeNode(int d)

{

data=d; sol=sag=null;

}

public synchronized void gir(int d)

{

// not ayni deger iki kere girilmeye calisilirsa

// ikinci giris dikkate alinmiyacaktir if(d<data)

{ if(sol==null) {sol=new TreeNode(d);} else {sol.gir(d);}

}

else if(d>data)

{ if(sag==null) {sag=new TreeNode(d);} else {sag.gir(d);}

}

}

}

public class Tree

{

public TreeNode kok; public Tree() {kok=null;}

public synchronized void gir(int d)

{

if(kok==null) kok=new TreeNode(d); else kok.gir(d);

}

public void node\_soldal\_sagdal\_sirala(){node\_soldal\_sagdal\_siralayici(kok);} public void soldal\_node\_sagdal\_sirala(){soldal\_node\_sagdal\_siralayici(kok);} public void soldal\_sagdal\_node\_sirala(){soldal\_sagdal\_node\_siralayici(kok);} public void node\_sagdal\_soldal\_sirala(){node\_sagdal\_soldal\_siralayici(kok);} public void sagdal\_node\_soldal\_sirala(){sagdal\_node\_soldal\_siralayici(kok);} public void sagdal\_soldal\_node\_sirala(){sagdal\_soldal\_node\_siralayici(kok);}

public void node\_soldal\_sagdal\_siralayici(TreeNode node)

{

if(node==null) return; System.out.print(node.data+" "); node\_soldal\_sagdal\_siralayici(node.sol); node\_soldal\_sagdal\_siralayici(node.sag);

}

public void node\_sagdal\_soldal\_siralayici(TreeNode node)

{

if(node==null) return; System.out.print(node.data+" "); node\_sagdal\_soldal\_siralayici(node.sag); node\_sagdal\_soldal\_siralayici(node.sol);

}

public void soldal\_node\_sagdal\_siralayici(TreeNode node)

{

if(node==null) return; soldal\_node\_sagdal\_siralayici(node.sol); System.out.print(node.data+" "); soldal\_node\_sagdal\_siralayici(node.sag);

}

public void sagdal\_node\_soldal\_siralayici(TreeNode node)

{

if(node==null) return; sagdal\_node\_soldal\_siralayici(node.sag); System.out.print(node.data+" "); sagdal\_node\_soldal\_siralayici(node.sol);

}

public void soldal\_sagdal\_node\_siralayici(TreeNode node)

{

if(node==null) return; soldal\_node\_sagdal\_siralayici(node.sol); soldal\_node\_sagdal\_siralayici(node.sag); System.out.print(node.data+" ");

}

public void sagdal\_soldal\_node\_siralayici(TreeNode node)

{

if(node==null) return; sagdal\_node\_soldal\_siralayici(node.sag); sagdal\_node\_soldal\_siralayici(node.sol); System.out.print(node.data+" ");

}

}

### 3.

**Program 13.14 H10OD2\_2000.java**

Bu program Gebze Yüksek teknoloji enstitüsü Java programla dili dersi, 2000 Güz dönemi ögrencilerinden Deniz KAYA tarafindan yazilmistir.

// Deniz KAYA

// H10OD2\_2000

// Tablodan-Treeye, Tree-den tabloya(recursive) yazdirma islemi.

import Tree; import TreeNode; import java.io.\*; import java.util.\*; import java.sql.\*;

public class H10OD2\_2000

{

public void veriyazdir(int data)

{

String url="jdbc:odbc:prog"; Connection con;

Statement stmt;

String query1="select x from H10OD2\_2000"; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{

System.err.println(e.getMessage());

}

try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

stmt.executeUpdate("insert into H10OD2\_2000 values("+data+")"); stmt.close();

con.close();

}

catch (SQLException e)

{

System.err.println(e.getMessage());

} //catch

};

// Tree de bulunan verileri sirali bir sekilde tabloya atar.

// Tabloya yazdirmak için veriyazdir metodunu kullanir.

public void soldal\_node\_sagdal\_siralayici(TreeNode node)

{

if(node==null) return; soldal\_node\_sagdal\_siralayici(node.sol);

// veriyi tabloya yazdir. veriyazdir(node.data); soldal\_node\_sagdal\_siralayici(node.sag);

}

public static void main(String args[])

{

Tree agac=new Tree(); int sayi;

System.out.println("orijinal olarak Tree'ye girilen sayi dizini"); String url="jdbc:odbc:prog";

Connection con;

Statement stmt;

String query1="select x from H10OD1\_2000"; try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(java.lang.ClassNotFoundException e)

{

System.err.println(e.getMessage());

}

try

{

con=DriverManager.getConnection(url); stmt=con.createStatement();

ResultSet rs= stmt.executeQuery(query1); while (rs.next())

{

int xi= rs.getInt("x"); agac.gir(xi);

}

H10OD2\_2000 h = new H10OD2\_2000();

// Küçükten büyüge sirali bir sekilde ekrana yazdirir. agac.soldal\_node\_sagdal\_sirala();

stmt.close();

con.close();

// H10OD2\_2000 tablosuna verileri küçükten büyüge sirali yazdirir. h.soldal\_node\_sagdal\_siralayici(agac.kok);

System.out.println("\n H10OD2\_2000 tablosuna veriler yazildi.");

}

catch (SQLException e)

{

System.err.println(e.getMessage());

} //catch

}

}

1. Bu alistirmada bir adres defteri olusturacagiz. Önce adres defterinde tanimlanan alanlari verelim ID anahtar alan

isim

soyisim adres

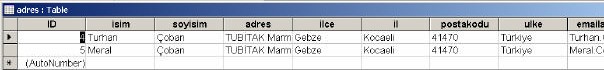
ilce il

postakodu ulke emailadresi telefonev telefonis telefoncep faks webadresi evadresi

alanlari tanimlanacaktir.

Temel access veri tabani tanimlamasini acsess’de yaptik. veri tabani dosyasinin ismi adres olarak tanimlandi. Veri tabaninin görünümü:

### 13013.JPG



**Sekil 13.13 adres.mpg veritabani tablosu adres**

seklindedir. Bu programi olusturabilmek çin bir dizi program tanim yapildi. Bunlar adresdefteri.java, adresgir.java, adresbul.java, adressil, adresdegistir.java, yardim.java, adrespanel.java ve ciktipaneli.java programlaridir. Bu programlarin kodlarini verelim :

### Program 13.15 adresdefteri.java

// adresdefteri ana programi import java.sql.\*;

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class adresdefteri extends JFrame { private ciktipaneli controls;

private adrespanel scrollArea; private JTextArea output; private String url;

private Connection connect; private JScrollPane textpane;

public adresdefteri()

{

super( "Adres defteri " ); Container c = getContentPane(); scrollArea = new adrespanel(); output = new JTextArea( 6, 30 );

c.setLayout( new BorderLayout() ); c.add( new JScrollPane( scrollArea ),

BorderLayout.CENTER ); textpane = new JScrollPane( output );

c.add( textpane, BorderLayout.SOUTH );

// database baglantisini olustur try {

url = "jdbc:odbc:adres";

Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" ); connect = DriverManager.getConnection( url ); output.append( "Baglanti saglandi\n" );

}

catch ( ClassNotFoundException cnfex ) { cnfex.printStackTrace();

output.append( "baglanti saglanamadi\n" + cnfex.toString() );

}

catch ( SQLException sqlex ) { sqlex.printStackTrace();

output.append( "baglanti saglanamadi\n" + sqlex.toString() );

}

catch ( Exception ex ) { ex.printStackTrace(); output.append( ex.toString() );

}

controls =

new ciktipaneli( connect, scrollArea, output); c.add( controls, BorderLayout.NORTH ); setSize( 500, 500 );

show();

}

public static void main( String args[] )

{

adresdefteri app = new adresdefteri(); app.addWindowListener(new BasicWindowMonitor());

}

}

### Program 13.16 adresgir.java

// adresgir.java import java.awt.\*;

import java.awt.event.\*; import java.sql.\*; import javax.swing.\*;

public class adresgir implements ActionListener { private adrespanel fields;

private JTextArea output; private Connection connection;

public adresgir( Connection c, adrespanel f, JTextArea o )

{

connection = c; fields = f; output = o;

}

public void actionPerformed( ActionEvent e )

{

try {

Statement statement = connection.createStatement();

if ( !fields.soyisim.getText().equals( "" ) &&

!fields.isim.getText().equals( "" ) ) { String query = "INSERT INTO adres (" +

"isim, soyisim, adres, ilce, " + "il, postakodu, ulke, " +

"emailadresi, telefonev, telefonis" +

",telefoncep, faks, webadresi,evadresi ) VALUES ('" +

fields.isim.getText() + "', '" + fields.soyisim.getText() + "', '" + fields.adres.getText() + "', '" +

fields.ilce.getText() + "', '" +

fields.il.getText() + "', '" + fields.postakodu.getText() + "', '" + fields.ulke.getText() + "', '" + fields.emailadresi.getText() + "', '" + fields.telefonev.getText() + "', '" + fields.telefonis.getText() + "', '" + fields.telefoncep.getText() + "', '" + fields.faks.getText() + "', '" + fields.webadresi.getText() + "', '" + fields.evadresi.getText() + "')";

//System.out.println(query);

//System.exit(0);

output.append( "\nSQL query gönderiliyor: " + connection.nativeSQL( query )

+ "\n" );

int result = statement.executeUpdate( query );

if ( result == 1 )

output.append( "\nQuery basariyla çalisti\n" ); else {

output.append( "\nQuery hata verdi\n" ); fields.isim.setText(" "); fields.soyisim.setText(" "); fields.adres.setText(" "); fields.ilce.setText(" "); fields.il.setText(" "); fields.postakodu.setText(" "); fields.ulke.setText(" "); fields.emailadresi.setText(" "); fields.telefonev.setText(" "); fields.telefonis.setText(" "); fields.telefoncep.setText(" "); fields.faks.setText(" "); fields.webadresi.setText(" "); fields.evadresi.setText(" ");

}

}

else

output.append( "\nen azindan isim ve soy isim girin ve sonra " + "ekle dugmesine basin\n" );

statement.close();

}

catch ( SQLException sqlex ) { sqlex.printStackTrace(); output.append( sqlex.toString() );

}

}

}

### Program 13.17 adresbul.java

// adresbul.java import java.awt.\*;

import java.awt.event.\*; import java.sql.\*; import javax.swing.\*;

public class adresbul implements ActionListener { private adrespanel fields;

private JTextArea output; private Connection connection;

public adresbul( Connection c, adrespanel f, JTextArea o )

{

connection = c; fields = f; output = o;

}

public void actionPerformed( ActionEvent e )

{

try {

if ( !fields.soyisim.getText().equals( "" ) && !fields.soyisim.getText().equals( "" )) {

Statement statement =connection.createStatement(); String query = "SELECT \* FROM adres " +

"WHERE soyisim = '" + fields.soyisim.getText() + "'"+ "AND isim = '" + fields.isim.getText() + "'";

output.append( "\nisim+soyisim Query'si gönderiliyor: " + connection.nativeSQL( query )

+ "\n" );

ResultSet rs = statement.executeQuery( query ); display( rs );

output.append( "\nisim + soyisim Query'si basariyla sonuçlandi\n" ); statement.close();

}

else if ( !fields.isim.getText().equals( "" ) ) { Statement statement =connection.createStatement(); String query = "SELECT \* FROM adres " +

"WHERE isim = '" +

fields.isim.getText() + "'"; output.append( "\nisim Query'si gönderiliyor: " +

connection.nativeSQL( query )

+ "\n" );

ResultSet rs = statement.executeQuery( query ); display( rs );

output.append( "\nsim Query'si basariyla sonuçlandi\n" ); statement.close();

}

else if ( !fields.soyisim.getText().equals( "" ) ) { Statement statement =connection.createStatement(); String query = "SELECT \* FROM adres " +

"WHERE soyisim = '" + fields.soyisim.getText() + "'";

output.append( "\nQuery gönderiliyor: " + connection.nativeSQL( query )

+ "\n" );

ResultSet rs = statement.executeQuery( query ); display( rs );

output.append( "\nQuery basariyla sonuçlandi\n" ); statement.close();

}

else

fields.soyisim.setText(

"Burada soyismi girin ve bul dügmesine basin" );

}

catch ( SQLException sqlex ) { sqlex.printStackTrace(); output.append( sqlex.toString() );

}

}

public void display( ResultSet rs )

{

try {

rs.next();

int recordNumber = rs.getInt( 1 ); if ( recordNumber != 0 ) {

fields.id.setText( String.valueOf( recordNumber)); fields.isim.setText(rs.getString( 2 ));

fields.soyisim.setText(rs.getString( 3 ));

fields.adres.setText(rs.getString( 4 ));

fields.ilce.setText(rs.getString( 5 ));

fields.il.setText(rs.getString( 6 ));

fields.postakodu.setText(rs.getString( 7 ));

fields.ulke.setText(rs.getString( 8 ));

fields.emailadresi.setText(rs.getString( 9 ));

fields.telefonev.setText(rs.getString( 10 ));

fields.telefonis.setText(rs.getString( 11 ));

fields.telefoncep.setText(rs.getString( 12 ));

fields.faks.setText(rs.getString( 13 ));

fields.webadresi.setText(rs.getString( 14 ));

fields.evadresi.setText(rs.getString( 15 ));

}

else

output.append( "\nKayit bulunamadi\n" );

}

catch ( SQLException sqlex ) { sqlex.printStackTrace(); output.append( sqlex.toString() );

}

}

}

### Program 13.18 adresdegistir.java

import java.awt.\*; import java.awt.event.\*; import java.sql.\*; import javax.swing.\*;

public class adresdegistir implements ActionListener { private adrespanel fields;

private JTextArea output; private Connection connection;

public adresdegistir( Connection c, adrespanel f, JTextArea o )

{

connection = c; fields = f; output = o;

}

public void actionPerformed( ActionEvent e )

{

try {

Statement statement = connection.createStatement();

if ( ! fields.id.getText().equals( "" ) ) { String query = "UPDATE adres SET " + "isim ='"+ fields.isim.getText() + "',soyisim ='"+ fields.soyisim.getText() + "',adres ='"+ fields.adres.getText() + "',ilce ='"+ fields.ilce.getText() +

"',il ='"+ fields.il.getText() +

"',postakodu ='"+ fields.postakodu.getText() + "',ulke ='"+ fields.ulke.getText() +

"',emailadresi ='"+ fields.emailadresi.getText() + "',telefonev ='"+ fields.telefonev.getText() + "',telefonis ='"+ fields.telefonis.getText() + "',telefoncep ='"+ fields.telefoncep.getText() + "',faks='"+ fields.faks.getText() +

"',webadresi ='"+ fields.webadresi.getText() + "',evadresi ='"+ fields.evadresi.getText()+

"' WHERE ID=" + fields.id.getText(); output.append( "\nquery gönderiliyor : " + connection.nativeSQL( query ) + "\n" );

int result = statement.executeUpdate( query ); if ( result == 1 )

output.append( "\nkayit basariyla degistirildi \n" ); else {

output.append( "\nkayit degistirme islemi basarilamadi\n" ); fields.isim.setText(" ");

fields.soyisim.setText(" "); fields.adres.setText(" "); fields.ilce.setText(" "); fields.il.setText(" "); fields.postakodu.setText(" "); fields.ulke.setText(" "); fields.emailadresi.setText(" "); fields.telefonev.setText(" " ); fields.telefonis.setText(" "); fields.telefoncep.setText(" "); fields.faks.setText(" "); fields.webadresi.setText(" "); fields.evadresi.setText(" ");

}

statement.close();

}

else

output.append( "\nsadece mevcut olan kayitlari degistirebilirsiniz\n" );

}

catch ( SQLException sqlex ) { sqlex.printStackTrace(); output.append( sqlex.toString() );

}

}

}

### Program 13.19 adressil.java

import java.awt.\*; import java.awt.event.\*;

public class adressil implements ActionListener { private adrespanel fields;

public adressil( adrespanel f )

{

fields = f;

}

public void actionPerformed( ActionEvent e )

{

fields.isim.setText(" "); fields.soyisim.setText(" "); fields.adres.setText(" "); fields.ilce.setText(" "); fields.il.setText(" "); fields.postakodu.setText(" "); fields.ulke.setText(" "); fields.emailadresi.setText(" "); fields.telefonev.setText(" "); fields.telefonis.setText(" "); fields.telefoncep.setText(" "); fields.faks.setText(" "); fields.webadresi.setText(" ");

}

}

### Program 13.20 adrespanel.java

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class adrespanel extends JPanel { private JPanel labelPanel, fieldsPanel; private String labels[] =

{ "ID :", "isim :", "soy isim :",

"adres :", "ilçe:", "il :",

"posta kodu :", "ülke :", "Email adresi :", "ev telefonu :", "is telefonu :",

"cep telefonu :", "faks :",

"web adresi :", "ev adresi :"};

JTextField id, isim, soyisim, adres, // alan isimleri ilce, il,postakodu,ulke,emailadresi,telefonev, telefonis,telefoncep,faks,webadresi,evadresi;

public adrespanel()

{ // Label

labelPanel = new JPanel(); labelPanel.setLayout(

new GridLayout( labels.length, 1 ) ); ImageIcon ii = new ImageIcon( "images/icon.jpg" ); for ( int i = 0; i < labels.length; i++ )

labelPanel.add( new JLabel( labels[ i ], ii, 0) );

// TextField panel fieldsPanel = new JPanel(); fieldsPanel.setLayout(

new GridLayout( labels.length, 1 ) ); id = new JTextField( 20 );

id.setEditable( false ); fieldsPanel.add( id );

isim = new JTextField( 20 ); fieldsPanel.add( isim );

soyisim = new JTextField( 20 ); fieldsPanel.add( soyisim ); adres = new JTextField( 20 ); fieldsPanel.add( adres );

ilce = new JTextField( 20 ); fieldsPanel.add( ilce );

il = new JTextField( 20 ); fieldsPanel.add( il );

postakodu = new JTextField( 20 ); fieldsPanel.add( postakodu );

ulke = new JTextField( 20 ); fieldsPanel.add( ulke );

emailadresi = new JTextField( 20 ); fieldsPanel.add( emailadresi); telefonev = new JTextField( 20 ); fieldsPanel.add( telefonev ); telefonis = new JTextField( 20 ); fieldsPanel.add( telefonis ); telefoncep = new JTextField( 20 ); fieldsPanel.add( telefoncep );

faks = new JTextField( 20 ); fieldsPanel.add( faks );

webadresi = new JTextField( 20 ); fieldsPanel.add( webadresi ); evadresi = new JTextField( 20 ); fieldsPanel.add( evadresi ); setLayout( new GridLayout( 1, 2 ) ); add( labelPanel );

add( fieldsPanel );

}

}

### Program 13.21 ciktipaneli.java

import java.awt.\*; import java.awt.event.\*; import java.sql.\*; import javax.swing.\*;

public class ciktipaneli extends JPanel { private JButton findName, addName,

updateName, clear, help;

public ciktipaneli( Connection c, adrespanel s, JTextArea t )

{

setLayout( new GridLayout( 1, 5 ) );

findName = new JButton( "BUL" ); findName.addActionListener( new adresbul( c, s, t ) ); add( findName );

addName = new JButton( "EKLE" );

addName.addActionListener( new adresgir( c, s, t ) ); add( addName );

updateName = new JButton( "DEGISTIR" ); updateName.addActionListener(

new adresdegistir( c, s, t ) ); add( updateName );

clear = new JButton( "SIL" ); clear.addActionListener( new adressil( s ) ); add( clear );

help = new JButton( "YARDIM" ); help.addActionListener( new yardim( t ) ); add( help );

}

}

### Program 13.22 yardim.java

import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class yardim implements ActionListener { private JTextArea output;

public yardim( JTextArea o )

{

output = o;

}

public void actionPerformed( ActionEvent e )

{

output.append( "\nbir kayit bulmak için isim ve/veya soyisim alanina yazip BUL dügmesine basiniz\n" + "Yeni bir kayit girmek için EKLE dügmesine basiniz\n" +

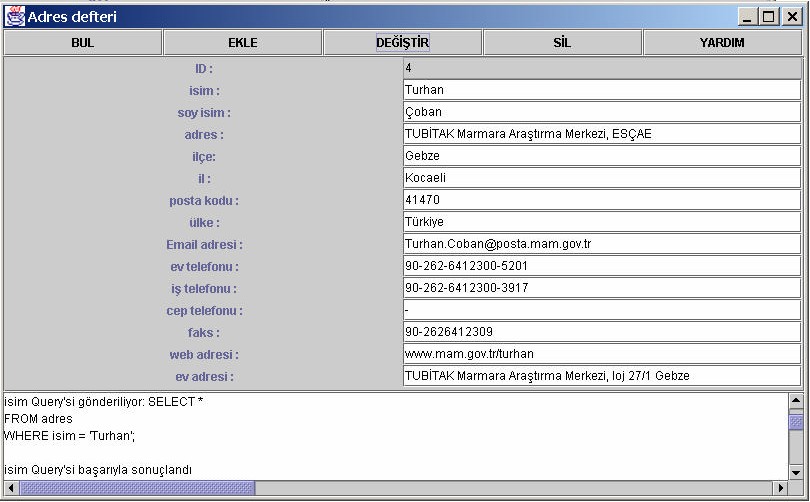
"Kayittaki degerleri degistirmek için DEGISTIR dügmesine basiniz\n" + "kaydi silmek için SIL dügmesine basiniz\n" );

}

}

bu program setinin ana programi olan adresdefteri programini çalistirdigimizda adres defterimiz açilir. Adres defterinde verilen bul, degistir, ekle, sil gibi alanlari kullanarak adres defterimizi kullanabiliriz. Belli bir komut verdigimizde SQL Querry dilinde olusan ve gönderilen komutlar alt pencereden izlenebilir. Ayrica herhangi bir hata olustu ise de bu prncerede hata mesaji verilecektir.

### 13014.JPG



**Sekil 13.14 adres.mpg veritabani tablosunu kullanan adresdefteri.java programi Jframe çiktisi.**

**BÖLÜM 14 JAVA ANADIL (NATIVE LANGUAGE) PROGRAMLAMASINA GIRIS C++ ve C PROGRAMLAMA DILLERININ JAVA ILE BIRLIKTE KULLANIMI**

### TEMEL KAVRAMLAR

Java programlama dili C dilinde yazilmistir. Bu yüzden C’den javaya geçis oldukça kolay bir sekilde yapilabilir. Bir örnekle birden fazla programlama dilinin nasil kullanilabildigine bakmadan önce, bir temel kavrami iyi anlamamiz gerekir. Java güvenlik açisindan maksimum güvenligi verebilen bir dildir. Öte taraftan C maksimum hizlara ulasmayi saglayabilen bir dildir. C programlarini java programlariyla birlikte kullanmamizin temel nedeni hiz gereken yerlerde bu hizi saglamaktir. Ancak hiz saglarken güvenlikten olan kaybimizi her zaman hesaba katmamiz ve ikisi arasinda bir denge bulmamiz biz programcilara birakilmistir. Bu yüzden iki dilli programlamada oldukça dikkatli bir sekilde çalisma gereksinimi olusturur. Bu konu C++ dilini iyi bilen programcilar içindir. Burada C programlamasina girmeyecegiz. Eger bu dili bilmiyorsaniz, bu konuya baslamadan önce ögrenmenizi veya java anadil programlamasini hiç kullanmamanizi tavsiye ederim. C++ dilini kullanabilmek için, hele iki dili bir arada kullanabilmek için iyi bir programci olmaniz gerekiyor, eger kendinizi güveniyorsaniz, bundan sonraki kisma geçebilirsiniz.

### Örnek problemle step step iki dilli program kodunun hazirlanmasi ve çalistirilmasi

Çok dilli programlama birden fazla basamagi gerektiren bir prosestir. Bir örnek problemler bu basamaklari inceleyelim :

C++ dili ile java dilinin beraber çalisacagi bir örnek olusturalim. Ilk step olarak java dilindeki programimizi yazalim :

### Program 14.1 JavaCpp.java programi

class JavaCpp

{

public native void cdenaktar(); static

{

System.loadLibrary("jcpp");

}

public static void main(String[] args)

{

new JavaCpp().cdenaktar();

}

}

JavaCpp programinda iki temel metod görüyoruz. Birinci metod cdenaktar() metodudur. Bu metod biraz sonra C++ dilinde olusturacagimiz fonksiyonu Java’nin bir parçasi olarak çagirir. Ikinci metod ise main metodudur. Cdenaktar metoduna bakarken bunun java metodu olmadigini belirten native sözcügü yer almaktadir. Metodun türü void’dir, yani hiçbir sey aktarilmiyacaktir. Programimizi, herhangi bir java programinda oldugu gibi önce **java JavaCpp.java**

Komutunu kullanarak derleriz. Ikinci stepte javah komutunu kullanarak

### javah JavaCpp

JavaCpp.h dosyasini olustururuz. Bu dosyanin içerigi :

### Program 14.2 JavaCpp.h arabaglanti programi

/\* DO NOT EDIT THIS FILE - it is machine generated \*/

#include <jni.h>

/\* Header for class JavaCpp \*/

#ifndef \_Included\_JavaCpp

#define \_Included\_JavaCpp

#ifdef cplusplus

extern "C" {

#endif

/\*

* Class: JavaCpp
* Method: cdenaktar
* Signature: ()V

\*/

JNIEXPORT void JNICALL Java\_JavaCpp\_cdenaktar (JNIEnv \*, jobject);

#ifdef cplusplus

}

#endif

#endif

seklinde olacaktir. Bundan sonraki stepte C++ dilinde asagidaki programi olusturalim :

### Program 14.3 JavaCpp\_c.cpp C++ programi

#include <iostream.h>

#include <jni.h>

#include "JavaCpp.h"

JNIEXPORT void JNICALL

Java\_JavaCpp\_cdenaktar(JNIEnv \*env, jobject obj)

{

cout<<"Java-C++ programlama dillerini birlikte kullaniyoruz "; return;

}

c++ programini derlemek için elbette bir derleyiciye ihtiyaç vardir. Biz burada MS Visual C++ versiyon 6 derleyicisini kullandik, herhangi bir dll dosyasi hazirlayan derleyici (MS Windows için) ayni isi görebilir.

Bu derleyici ile derleme için ya direk olarak dos’ta

cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LD %JavaCpp\_c.cpp –Fejcpp.dll

komutunu kullaniriz, yada bu uzun komutu her sefer yazmamak için bir isim.bat dosyasi yaratip bunu kullanabiliriz. Örnegin hazirladigimiz pcomp.bat dosyasi

### Program 14.4 pcomp.bat MSdos programi

cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LD %1.cpp -Fe%2.dll

kullanilarak yukaridaki komut pcomp JavaCpp\_c jcpp

seklinde verilebilir. Bu islemleri yaptigimizda programimiz çalismaya hazir olacaktir. Programi çalistirmak için :

### java JavaCpp

komutunu kullanmamiz yeterli olacaktir. Program çiktisi :

### Java-C++ programlama dillerini birlikte kullaniyoruz

Seklinde olacaktir. Burada derlemenin jcpp.dll dosyasina yapildigini hatirlatalim, ve bu dosya java programi üzerinden C++ metodunu çalistirirken kullandigimiz ana programdi.

### 14.2 DEGISKENLERIN VE JAVA NESLERININ AKTARILMASI

Ilk programimizda java ve C++ arasinda herhangi bir degisken aktarilmasi olmamis, sadece void türü fonksiyon kullanilarak belli bir yazi yazan C++ fonksiyonu çagirilmistir. Ikinci örnegimizde java programimizdan String türü degiskeni C++ programlama diline aktaracak ve kullanacagiz.

Ilk java programimiz Showmessage.java. bu programda ekrandan girilen bir string, yahut ekrandan girilen bir string degeri yoksa programin içinde tanimlanan bir stringle birlikte C++ programina aktarilip burada ekrana yazilmaktadir.

### Program 14.5 ShowMessage.java programi

public class ShowMessage {

private native void ShowMessage(String msg); static { System.loadLibrary("MsgI"); }

public static void main(String[] args) { ShowMessage app = new ShowMessage(); if(args.length==0)

app.ShowMessage("Bu mesaj javadan C++ ya gönderildi ve orada yazildi"); else

app.ShowMessage(args[0]);

}

}

bu program **javac ShowMessage.java** ve **javah ShowMessage** komutlariyla islendikten sonra

### Program 14.6 MsgImpl.cpp programi

//dosya : MsgImpl.cpp

#include <jni.h>

#include <iostream.h>

#include <stdio.h>

#include "ShowMessage.h"

extern "C" JNIEXPORT void JNICALL Java\_ShowMessage\_ShowMessage(JNIEnv\* env, jobject, jstring jMsg)

{

const char\* msg=env->GetStringUTFChars(jMsg,0); printf("JNI cikti stringi : %s\n",msg) ;

env->ReleaseStringUTFChars(jMsg, msg);

}

c++ dilindeki MsgImpl.cpp programi hazirlanmis ve üstte hazirladigimiz pcomp.bat yardimiyla

### pcomp MsgImpl MsgI

komutu kullanilarak derlenmistir. Program çiktisi

**java ShowMessage “Merhaba Dünya”** komutu kullanilarak :

### JNI cikti stringi : Merhaba Dünya

veya

**java ShowMessage** komutu kullanilarak

JNI cikti stringi : Bu mesaj javadan C++ ya gönderildi ve orada yazdirildi

Çiktisi alinabilir. Programda veri aktarimi yapmak amaciyla java tarafinda metod tanimi yapilirken

### private native void ShowMessage(String msg);

tanimi verilmistir. Buradaki String msg metodun dolayisiyla C++ fonksiyonunun girdisini teskil etmektedir. C++ tarafinda mesaji alirken,

Java\_ShowMessage\_ShowMessage(JNIEnv\* env, jobject, **jstring jMsg**)

{

const char\* msg=env->GetStringUTFChars(jMsg,0);

jstring jMsg tanimiyla aktarilmis ve const char\* msg=env->GetStringUTFChars(jMsg,0); tanimiyla c degiskenine dönüstürülmüstür.

Javada Stringler unicode (16 bit) olarak tanimlanmistir. C stringleri ise 8 bitlik ascii char boyutlu degiskeni olarak tanimlanir. Bu yüzden string degiskenlerini javadan cye veya cden javaya aktarirken dönüsümler yapmak gerekir. Ayni zamnda java string degiskenleri length (boyut) degiskenini de bünyelerinde tasirlar. Cde ise boyut tanimlanmamistir. Ayri bir degisken üzerinden tanimlanmasi gerekebilir. Bu dönüsümleri yapmak için bir dizi metod tanimlanmistir.

Bu tanimalarin tamami söyledir :

GetStringChars GetStringLength GetStringUTFChars GetStringUTFLength NewString NewStringUTF ReleaseStringChars ReleaseStringUTFChars

Buradaki ikinci örnegimizde java programinda c++ dilindeki bir fonksiyonun (metod) yaptigi hesap sonuçlarini aktaracagiz. Programimiz java’da çagirilan bir C++ kare metodunu kullanarak java dilinde girilen sayinin karesini hesaplayacaktir.

### Program 14.7 karesinifi.java programi

class karesinifi {

private native double kare(double x); public static void main(String args[]) { karesinifi p = new karesinifi();

double x=2.0; double y = p.kare(x);

System.out.println("" +x+" kare = "+ y);

}

static { System.loadLibrary("kareI");

}

}

bu programi javac karesinifi.java deyimiyle derleyip, javah karesinifi deyimiyle anadil kütüphanesine çevirirsek

### Program 14.8 karesinifi.h programi

/\* DO NOT EDIT THIS FILE - it is machine generated \*/

#include <jni.h>

/\* Header for class karesinifi \*/

#ifndef \_Included\_karesinifi

#define \_Included\_karesinifi

#ifdef cplusplus extern "C" {

#endif

/\*

* Class: karesinifi
* Method: kare
* Signature: (D)D

\*/

JNIEXPORT jdouble JNICALL Java\_karesinifi\_kare (JNIEnv \*, jobject, jdouble);

#ifdef cplusplus

}

#endif

#endif

karesinifi.h bize java programindaki private native double kare(double x); metodunun c++ anadil (Native) karsiliginin

JNIEXPORT jdouble JNICALL Java\_karesinifi\_kare (JNIEnv \*, jobject, jdouble);

oldugunu göstermektedir. Simdi kare metodunun aslini C++ dilinde yazarsak :

### Program 14.9 karesinifi.cpp C++ programi

#include <stdio.h>

#include <jni.h>

#include "karesinifi.h"

JNIEXPORT jdouble JNICALL Java\_karesinifi\_kare(JNIEnv \*, jobject, jdouble x )

{

return x\*x;

}

burada metod isminin kare yerine **Java\_karesinifi\_kare** seklini aldigini görüyoruz. Double degiskeni de yerini jdouble degiskenine birakmis durumda. Java anadilinde programlari yazarken metod isimleri Java\_sinifismi\_metodismi

Seklini alir. Basit java degiskenlerinin Anadil karsiliklari ise Tablo 14.1 Java degisken tipleri ve ana dil (native) esdegerleri

|  |  |  |
| --- | --- | --- |
| Java degisken tipi | Anadil (Native) degisken esdegeri | Degisken bit degeri |
| Boolean | jboolean | 8, isaretsiz |
| Byte | jbyte | 8 |
| Char | jchar | 16, isaretsiz |
| Short | jshort | 16 |
| Int | jint | 32 |
| Long | jlong | 64 |
| Float | jfloat | 32 |
| Double | jdouble | 64 |
| Object | jobject |  |
| Void | void | Yok |

Seklindedir. C++ programi pcomp karesinifi kareI deyimiyle veya açik yazilisiyla

cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LDkaresinifi.cpp –KareI.dll deyimiyle derlenirse (MS Visual C++ 6), sonuç :

2.0 kare = 4.0 seklinde çikacaktir.

Simdi de bu islemin tam tersini inceleyelim. Java’da yazilmis olan kare metodunu C++ ve C dillerinde çagiralim. Java programimiz karesinifi1.java asagidaki gibi tanimlanmistir.

### Program 14.10 karesinifi1.java java programi

class karesinifi1 {

private native void nativeMethod(); private double kare(double x)

{

return x\*x;

}

public static void main(String args[])

{

karesinifi1 c=new karesinifi1(); c.nativeMethod();

}

static { System.loadLibrary("kare1I");

}

}

sinifin alt sinifi olan kare’de girilen sayinin karesi hesaplanmaktadir. NativeMethod metodu da C (veya c++) dilleriyle baglanti saglamaktadir. Metodumuzu yine java karesinifi1.java ile derledikten sonra javah karesinifi1 komutuyla karesinifi1.h dosyasini olustururuz. Bu dosya :

### Program 14.11 karesinifi1.h programi

/\* DO NOT EDIT THIS FILE - it is machine generated \*/

#include <jni.h>

/\* Header for class karesinifi1 \*/

#ifndef \_Included\_karesinifi1

#define \_Included\_karesinifi1

#ifdef cplusplus extern "C" {

#endif

/\*

* Class: karesinifi1
* Method: nativeMethod
* Signature: ()V

\*/

JNIEXPORT void JNICALL Java\_karesinifi1\_nativeMethod (JNIEnv \*, jobject);

#ifdef cplusplus

}

#endif

#endif

anadil tarafinda program yazilirken bize yardimci olabilecek bir ara java programi javap de mevcuttur.

Bu islemi yapmak zorunda degiliz. Yapma sebebimizi c++ dili programini inceleyince daha iyi anlayacaksiniz. javap -s -p karesinifi1

### Program 14.12 javap komutuyla olusturulan ara kod

Compiled from karesinifi1.java

class karesinifi1 extends java.lang.Object { karesinifi1();

/\* ()V \*/

private native void nativeMethod();

/\* ()V \*/

private double kare(double);

/\* (D)D \*/

public static void main(java.lang.String[]);

/\* ([Ljava/lang/String;)V \*/ static {};

/\* ()V \*/

}

simdi c++ koduna bakalim :

### Program 14.13 karesinifi1.cpp programi

#include <stdio.h>

#include <iostream.h>

#include <jni.h>

#include "karesinifi1.h"

JNIEXPORT void JNICALL Java\_karesinifi1\_nativeMethod(JNIEnv \*env, jobject obj)

{

jclass cls=env->GetObjectClass(obj);

jmethodID mid=env->GetMethodID(cls,"kare","(D)D"); double x=2.0;

double y=env->CallDoubleMethod(obj,mid,x); cout<<x<<"nin karesi = "<<y;

}

Bu programda önce java sinifinin (karesinifi1) aktarilmasi **jobject obj** degiskeni ve jclass cls=env->GetObjectClass(obj);

üzerinden yapilir. Burada sinif adresi cls degiskenine yüklenmis olur. Alt metod karenin tanimi için jmethodID mid=env->GetMethodID(cls,"kare","(D)D");

tanimi kullanilmistir. Buradaki “(D)D” tanimi program 14.12 de bize verilmisti. Anlami java programindaki kare metodunun double degisken girisi oldugu ve çikisinda double degisken istedigidir. “kare” deyimiyle metodun javadaki ismi verilmistir. Cls ise üstte tanimladigimiz karesinifi1 java sinifinin adresine belirtmektedir.

double y=env->CallDoubleMethod(obj,mid,x);

deyimiyle de metodu çagirip sonuçlarini y degiskenine yüklemekteyiz.

Burada kullandigimiz D isareti java degisken türünün double oldugunu belirtiyordu. Java degisken tiplerinin isaret karsiliklari Tablo 14.2 de verilmistir.

Tablo 14.2 Java tiplerinin esdeger isaretleri

|  |  |
| --- | --- |
| Isaret | Java degisken tipi |
| Z | boolean |
| B | Byte |
| C | Char |
| S | Short |
| I | Int |
| J | Long |
| F | Float |
| D | Double |
| L sinif ismi | sinif ismi |
| [Degisken türü | Degisken türü[] |

Bu programda sinifi çagirmak için GetObjectClass metodu, metodu tanimlamak içinde GetMethodID metodunu kullandik. CallDoubleMethod metodu ile de metodu çalistirdik. Paralel islemleri yapmak için mevcut olan metod listesi ve anlamlari söyledir :

GetObjectClass : sinif adresini tanimlar GetMethodID : dinamik metod adresini tanimlar GetStaticMethodID : statik metod adresini tanimlar Call<degiskentürü>Method : çesitli metodlari çagirir

CallBooleanMethod CallByteMethod CallCharMethod CallDoubleMethod CallFloatMethod CallIntMethod CallLongMethod CallObjectMethod CallShortMethod CallVoidMethod

CallStatic< degiskentürü >Method : statik metodlari çagirir, degisken türleri olarak yukaridakinin ayni

türleri kabul eder.

CallNonvirtual< degiskentürü >Method Call< degiskentürü >MethodV

Call< degiskentürü >MethodA

Ayni programin c dilinde yazilmasi küçük farkliliklar tasir. C dilideki program :

### Program 14.14 karesinifi1.c programi

#include <stdio.h>

#include <jni.h>

#include "karesinifi1.h"

JNIEXPORT void JNICALL Java\_karesinifi1\_nativeMethod(JNIEnv \*env, jobject obj)

{

jclass cls=(\*env)->GetObjectClass(env,obj);

jmethodID mid=(\*env)->GetMethodID(env,cls,"kare","(D)D"); double x=2.0;

double y=(\*env)->CallDoubleMethod(env,obj,mid,x); printf("%f nin karesi = %f ",x,y);

}

programdan da görüldügü gibi C++ daki : jclass cls=env->GetObjectClass(obj);

komutu

jclass cls=(\*env)->GetObjectClass(env,obj); halini almistir.

jmethodID mid=env->GetMethodID(cls,"kare","(D)D"); komutu

jmethodID mid=(\*env)->GetMethodID(env,cls,"kare","(D)D"); halini almistir.

double y=env->CallDoubleMethod(obj,mid,x); komutu ise

double y=(\*env)->CallDoubleMethod(env,obj,mid,x);

halini almistir. Sonuç olarak env yerine referans degiskeni (\*env) gelmektedir ve env referans degiskeni ayni zamanda metod degiskeni olarakda çagirilmaktadir. Bunun disinda islemler arasinda bir fark yoktur.

Program **java karesinifi1** deyimiyle çalistirildiginda

### 2nin karesi = 4

sonucunu aliriz.

Ikinci bir sinif ulasim örnegi olarak UseObjects programini verelim. Burada sinif, MyJavaClass içinde bir int degisken ve bir void metod mevcuttur. Ikinci bir metod olan UseObjects metodu kullanilarak native metod changeObject’e ulasilmakta ve bu metod c++ programinda kullanilarak sonuçlar hesaplanmaktadir.

### Program 14.15 UseObjects.java programi

//UseObjects.java class MyJavaClass { public int aValue;

public void divByTwo() { aValue /= 2; }

}

public class UseObjects { private native void

changeObject(MyJavaClass obj); static { System.loadLibrary("UseObjImpl");

}

public static void main(String[] args) { UseObjects app = new UseObjects(); MyJavaClass anObj = new MyJavaClass(); anObj.aValue = 2;

app.changeObject(anObj); System.out.println("Java: " + anObj.aValue);

}

}

### Program 14.16 javah UseObject komutu kullanilarak olusturulan UseObjects.h programi

/\* DO NOT EDIT THIS FILE - it is machine generated \*/

#include <jni.h>

/\* Header for class UseObjects \*/

#ifndef \_Included\_UseObjects

#define \_Included\_UseObjects

#ifdef cplusplus extern "C" {

#endif

/\*

* Class: UseObjects
* Method: changeObject
* Signature: (LMyJavaClass;)V

\*/

JNIEXPORT void JNICALL Java\_UseObjects\_changeObject (JNIEnv \*, jobject, jobject);

#ifdef cplusplus

}

#endif

#endif

### Program 14.17 UseObjImp.cpp programi

#include <jni.h>

extern "C" JNIEXPORT void JNICALL Java\_UseObjects\_changeObject( JNIEnv\* env, jobject, jobject obj)

{

jclass cls = env->GetObjectClass(obj);

jfieldID fid = env->GetFieldID(cls, "aValue", "I");

jmethodID mid = env->GetMethodID(cls, "divByTwo", "()V"); int value = env->GetIntField(obj, fid);

printf("Native: %d\n", value); env->SetIntField(obj, fid, 6);

env->CallVoidMethod(obj, mid); value = env->GetIntField(obj, fid); printf("Native: %d\n", value);

}

Program 14.15 de tanimlanan MyJavaClass sinifi bir int degisken, aValue, ve bir metod, divByTwo , içermektedir. Metod int de verilen degeri 2 ye bölmekte ve sonucu aktarmaktadir. Bu programda int sinif alt degiskenini tanimlamak için

jfieldID fid = env->GetFieldID(cls, "aValue", "I"); deyimini kullandik. Void aValue metodu

jmethodID mid = env->GetMethodID(cls, "divByTwo", "()V"); deyimiyle tanimlandi. Ve

env->SetIntField(obj, fid, 6); tanimiyla 6 degeri atandi.

Siniflarin alt degiskenlerine deger atamak (ve okumak )için GetFieldID

GetStaticFieldID

Get< degiskentürü >Field GetBooleanField GetByteField

GetCharField GetDoubleField GetFloatField GetIntField GetLongField GetObjectField GetShortField

Set< degiskentürü >Field GetStatic< degiskentürü >Field SetStatic< **degiskentürü** >Field

Metodlari mevcuttur.

Simdi de Çok boyutlu degiskenleri nasil kullanabildigimizi inceleyelim :

### Program 14.18 IntArray.java programi

class IntArray {

private native int sumArray(int arr[]); public static void main(String args[]) { IntArray p = new IntArray();

int arr[] = new int [10]; for (int i = 0; i < 10; i++) arr[i] = i;

int sum = p.sumArray(arr); System.out.println("sum = " + sum);

}

static { System.loadLibrary("MyImpOfIntArray");

}

}

### Program 14.19 IntArray.c programi

#include <jni.h>

#include "IntArray.h"

JNIEXPORT jint JNICALL

Java\_IntArray\_sumArray(JNIEnv \*env, jobject obj, jintArray arr)

{

jsize len = (\*env)->GetArrayLength(env, arr); int i, sum = 0;

jint \*body = (\*env)->GetIntArrayElements(env, arr, 0); for (i=0; i<len; i++) {

sum += body[i];

}

(\*env)->ReleaseIntArrayElements(env, arr, body, 0); return sum;

}

burada tanimlanan java programinda boyutlu degisken arr tanimlanmis, ve boyutlu degiskenin toplami c programinda hesaplanmistir. C metodunda boyutlu degiskeni tanimlamak için jintArray türü arr degiskeni kullanilmistir. Boyutlu degiskenin boyutu jsize tipi len degiskenine

jsize len = (\*env)->GetArrayLength(env, arr); deyimiyle aktarilmistir.

arr degiskeninin degeri jint tipi body boyutlu degiskene, indeksi 0dan baslamak üzere jint \*body = (\*env)->GetIntArrayElements(env, arr, 0);

metoduyla aktarilir.

arr boyutlu degiskeninin degeri (degerler degismiste olabilir) tekrar kullanilabilmek üzere (\*env)->ReleaseIntArrayElements(env, arr, body, 0);

deyimi kullanilarak serbest birakilir (javaya gönderilir). Bu islemlerin sonucunda java programi : sum = 45

sonucunu verecektir. JNI boyutlu degisken çagirma fonksiyonlari(metodlari) söyledir : GetArrayLength

Get< degiskentürü >ArrayElements GetBooleanArrayElements GetByteArrayElements GetCharArrayElements GetDoubleArrayElements GetFloatArrayElements GetIntArrayElements GetLongArrayElements GetShortArrayElements

Release< degiskentürü >ArrayElements Get< degiskentürü >ArrayRegion

Set< degiskentürü >ArrayRegion GetObjectArrayElement SetObjectArrayElement

Java dilinde tanimlanmis, fakat c dilinde direk tanimlanmamis islemlerden birisi de hata analiz islemidir (Catch- throw exception). Ana dil programlamasi kullanirken, bu islemi de yapmak mümkündür. Asagidaki CatchThrow program setinde hata sisteminin kullanilmasini görüyoruz.

### Program 14.20 CatchThrow.java programi

class CatchThrow {

private native void catchThrow() throws IllegalArgumentException; private void callback() throws NullPointerException {

throw new NullPointerException("thrown in CatchThrow.callback");

}

public static void main(String args[]) { CatchThrow c = new CatchThrow(); try {

c.catchThrow();

} catch (Exception e) { System.out.println("In Java:\n " + e);

}

}

static { System.loadLibrary("MyImpOfCatchThrow");

}

}

### Program 14.21 CatchThrow.c programi

#include <jni.h>

#include "CatchThrow.h"

JNIEXPORT void JNICALL

Java\_CatchThrow\_catchThrow(JNIEnv \*env, jobject obj)

{

jclass cls = (\*env)->GetObjectClass(env, obj);

jmethodID mid = (\*env)->GetMethodID(env, cls, "callback", "()V"); jthrowable exc;

if (mid == 0) { return;

}

(\*env)->CallVoidMethod(env, obj, mid);

exc = (\*env)->ExceptionOccurred(env); if (exc) {

/\*

Biz aslinda c’de hata konusunda gerçekten bir sey yapmiyoruz, sadece bir mesaj iletiyoruz, hata degiskenini temizliyoruz ve yeni bir exception gönderiyoruz. \*/

jclass newExcCls;

(\*env)->ExceptionDescribe(env); (\*env)->ExceptionClear(env);

newExcCls = (\*env)->FindClass(env, "java/lang/IllegalArgumentException"); if (newExcCls == 0) { /\* Unable to find the new exception class, give up. \*/ return;

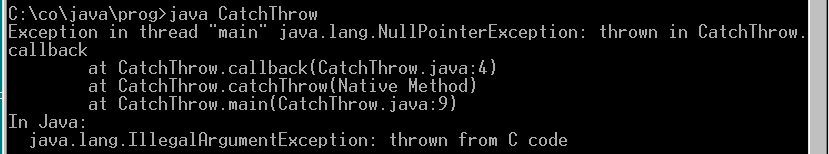
}

(\*env)->ThrowNew(env, newExcCls, "thrown from C code");

}

}

bu programi çalistirdigimizda :



sonucunu aliriz. Buradaki In Java:

java.lang.IllegalArgumentException: thrown from C code mesaji c tarafindan olusturulup gönderilmistir.

JNI Hata fonksiyonlari sunlardir :

ExceptionClear ExceptionDescribe ExceptionOccurred

Burada iki dilli programlama kavramina bir giris yaptik. Buradaki kavramlar size biraz zor geldiyse lütfen umutsuzluga kapilmayin, java ve C dillerini çok iyi ögrendikten sonra geri dönün. Iki dilli programlama her zaman programlama dünyasinin en zor islerindendir, ayrica jbuilder, Microsoft j++ gibi paketler java anadil (native) programlamasinda isleri kolaylastiran otomatik kod olusturma sisyemleri sunarlar, bu tür programlama araçlariyla isinizi daha kolay hale getirebilirsiniz.

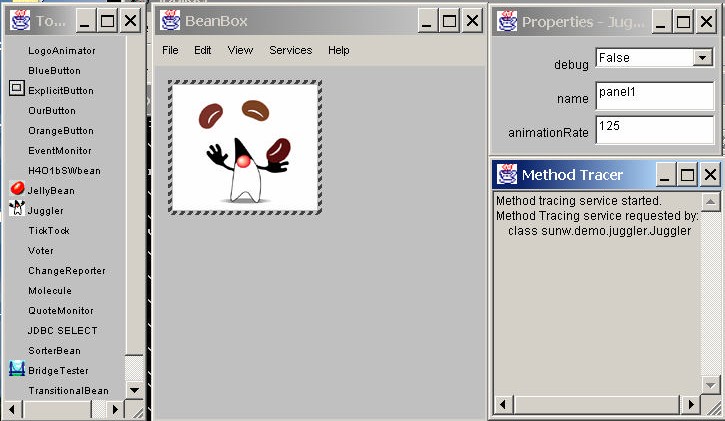
### BÖLÜM 15 JAVA BEANS (JAVA FASULYELERI) PROGRAMLAMA

### TEMEL KAVRAMLAR

Java beans tekrar kullanilabilir paketler halinde java programlamasi gelistiren, ayni zamanda hizli yeni gelistirmelere de ön ayak olan bir programlama sistemidir. Daha önceden programlanmis hazir program paketçiklerinin ve apletlerinin bilgisayarlar arasinda hizli bir sekilde alisverisine yardimci olan bir sistemdir. "Java beans" kavrami yeni bir programlama seklini getirebilir. Bu kod yazma yerine program parçaciklarini birlestirerek programlama yapan bir programci olabilir.

Java beans sistemini kullanmak için "JavaBeans development kit" Java bean kullanma seti'ne ihtiyaciniz vardir. Bu seti <http://java.sun.com/beans/software/index.html> adresinden kopyalayabilirsiniz. Java bean kullanma setinin Java 1.1 versiyonu da olmakla birlikte biz burada 1.2 versiyonunun kullanildigini varsayacagiz.. Java Bean gelistirme setini kullanmak için önce programi kurmamiz gerekir. Paketi java dosyalarinizin oldugu ana dosyanin bir alt dosyasi veya bagimsiz bir dosya sisteminde tanimlayabilirsiniz. Bu dosyanin alt dosyasi olarak **beanbox alt** dosyasi mevcuttur. Bu dosyanin altinda window sisteminde **run.bat**, unix sisteminde **run.sh** dosyasini bulacaksiniz. bu dosyayi açtiginizda beanbox açilacaktir. Beanbox programini açtigimizda, yaninda Toolbox(aletkutusu), Properties (özellikler) ve Method tracer( metot izleyici) adi verilen programlari da görecegiz. ToolVox kutusunda jars alt dosyasinda yer aln bean program paketlerinin listesini görecegiz bunlarin birini secip BeanBox kutusuna getirince bu program çalismaya basliyacaktir. Örnegin Sekil 15.1 de Bean Box ve içinde çalisan Jogler programini görmekteyiz.

### 15001.JPG



**Sekil 15.1 Juggler bean programinin BeanBox'da görünümü**

* 1. **Bean programini gelistirme**

Simdi kendi bean programimizi nasil yazabilecegimize bir göz atalim. Daha önceki bölümlerde yildiz.java programini gelistirmis ve bu programdan yola çikarak analog saat ve çesitli çizim programlari yazmistik. Önce yilsiz.java programinin swing ve Graph2D olarak degistirlmis sekline bir göz atalim

### Program 15.1 yildizSW.java programi, yildiz sinifi

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*; import polar;

public class yildizSW

{

public static void drawYildiz1(Graphics g2,int xi,int yi, int n,int yildizboyu,double aci )

{

// bu yildiz cizime teta=pi/2+aci radyandan baslar

Graphics2D g=(Graphics2D)g2; double teta=2.0\*Math.PI/n; double R=yildizboyu;

double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar(); for(int i=0;i<n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2);

g.drawLine((xi+(int)P1.xi()),(yi-(int)P1.xj()),

(xi+(int)P2.xi()),(yi-(int)P2.xj())); double teta3=teta\*(i+1)+Math.PI/2.0+aci; P3.polarGir(R,teta3); g.drawLine((xi+(int)P2.xi()),(yi-(int)P2.xj()),

(xi+(int)P3.xi()),(yi-(int)P3.xj())); g.drawLine(xi,yi,

(xi+(int)P1.xi()),(yi-(int)P1.xj())); g.drawLine(xi,yi,

(xi+(int)P2.xi()),(yi-(int)P2.xj()));

}

} //drawYildiz1 metodu sonu

public static void drawYildiz1(Graphics g2,int xi,int yi, int n,int yildizboyu )

{

drawYildiz1(g2,xi,yi,n,yildizboyu,0);

} //drawYildiz1 metodu sonu

public static void drawYildiz(Graphics g2,int xi,int yi, int n,int yildizboyu,double aci )

{

Graphics2D g=(Graphics2D)g2;

// bu yildiz cizime teta=pi/2+aci radyandan baslar double teta=2.0\*Math.PI/n;

double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2];

GeneralPath polygon=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD, x.length);

for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2.0+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2); x[2\*i]=xi+(int)P1.xi();

y[2\*i]=yi-(int)P1.xj();

x[2\*i+1]=xi+(int)P2.xi();

y[2\*i+1]=yi-(int)P2.xj(); if(i==n)

{

x[2\*i]=x[0];

y[2\*i]=y[0];

}

}

for(int i=0;i<=2\*n;i++)

{

if(i==0)

{

polygon.moveTo(x[0],y[0]);

}

else

{

polygon.lineTo(x[i],y[i]);

}

}

g.draw(polygon);

} //drawYildiz1 metodu sonu

public static void fillYildiz(Graphics g2,int xi,int yi, int n,int yildizboyu,double aci )

{

Graphics2D g=(Graphics2D)g2;

// bu y• ld• z cizime teta=pi/2+aci radyandan baslar double teta=2.0\*Math.PI/n;

double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[2\*n+2]; int y[]=new int[2\*n+2];

GeneralPath polygon=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD, x.length);

for(int i=0;i<=n;i++)

{

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta/2.0+teta1; P1.polarGir(R,teta1); P2.polarGir(r,teta2); x[2\*i]=xi+(int)P1.xi();

y[2\*i]=yi-(int)P1.xj();

x[2\*i+1]=xi+(int)P2.xi();

y[2\*i+1]=yi-(int)P2.xj(); if(i==n)

{

x[2\*i]=x[0];

y[2\*i]=y[0];

}

}

for(int i=0;i<=2\*n;i++)

{

if(i==0)

{

polygon.moveTo(x[0],y[0]);

}

else

{

polygon.lineTo(x[i],y[i]);

}

}

g.fill(polygon);

} //fillYildiz metodu sonu

public static void fillAkrep(Graphics g2,int xi,int yi, int n,int yildizboyu,double aci,int i )

{

// bu yildiz cizime teta=pi/2+aci radyandan baslar

// yildizin i‡ini boyar Graphics2D g=(Graphics2D)g2; double teta=2.0\*Math.PI/n; double R=yildizboyu;

double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[5]; int y[]=new int[5];

GeneralPath polygon=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD, x.length);

x[0]=xi;

y[0]=yi;

x[4]=xi;

y[4]=yi;

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta1+teta/2.0;

double teta3=teta1-teta/2+Math.PI\*2; P1.polarGir(R,teta1); P2.polarGir(r,teta2); P3.polarGir(r,teta3); x[1]=xi+(int)P3.xi();

y[1]=yi-(int)P3.xj();

x[2]=xi+(int)P1.xi();

y[2]=yi-(int)P1.xj();

x[3]=xi+(int)P2.xi();

y[3]=yi-(int)P2.xj(); for(i=0;i<5;i++)

{

if(i==0)

{

polygon.moveTo(x[0],y[0]);

}

else

{

polygon.lineTo(x[i],y[i]);

}

}

g.fill(polygon);

} //fillAkrep metodu sonu

public static void drawAkrep(Graphics g2,int xi,int yi, int n,int yildizboyu,double aci,int i )

{

Graphics2D g=(Graphics2D)g2; double teta=2.0\*Math.PI/n;

double R=yildizboyu; double r=yildizboyu\*0.25; polar P1=new polar(); polar P2=new polar(); polar P3=new polar();

int x[]=new int[5]; int y[]=new int[5];

GeneralPath polygon=new GeneralPath(GeneralPath.WIND\_EVEN\_ODD, x.length);

x[0]=xi;

y[0]=yi;

x[4]=xi;

y[4]=yi;

double teta1=teta\*i+Math.PI/2.0+aci; double teta2=teta1+teta/2.0;

double teta3=teta1-teta/2+Math.PI\*2; P1.polarGir(R,teta1); P2.polarGir(r,teta2); P3.polarGir(r,teta3); x[1]=xi+(int)P3.xi();

y[1]=yi-(int)P3.xj();

x[2]=xi+(int)P1.xi();

y[2]=yi-(int)P1.xj();

x[3]=xi+(int)P2.xi();

y[3]=yi-(int)P2.xj(); for(i=0;i<5;i++)

{

if(i==0)

{

polygon.moveTo(x[0],y[0]);

}

else

{

polygon.lineTo(x[i],y[i]);

}

}

g.draw(polygon);

} //drawAkrep metodu sonu

public static void drawDaire(Graphics g2, int xi,int yi, int R)

{

Graphics2D g=(Graphics2D)g2;

g.draw(new Ellipse2D.Double((int)(xi-R),(int)(yi-R),2\*R,2\*R));

}

public static void fillDaire(Graphics g2, int xi,int yi, int R)

{

Graphics2D g=(Graphics2D)g2;

g.fill(new Ellipse2D.Double((int)(xi-R),(int)(yi-R),2\*R,2\*R));

}

}

Simdi ilk bean programimizi hazirliyalim. Önce programimizi yazalim. program 15.2 de yildizciz2Dbean.java programi görülmektedir. Bu programda **Serializable** sinifi implement edilmistir. bunun disinda normal bir java programidir. Programin Jpanel olarak yazilmis olduguna dikkat edelim. Bu yüzden

### java yildizciz2Dbean

terimi kullanilarak çalistirilabilir.

### Program 15.2 yildizciz2Dbean.java programi

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*; import javax.swing.\*; import java.net.\*; import java.io.\*;

public class yildizciz2Dbean extends JPanel implements ActionListener,Serializable

{

final static Color bg=Color.white; final static Color fg=Color.black; final static Color kirmizi=Color.red; final static Color beyaz=Color.white;

public yildizciz2Dbean()

{

setSize(getPrefferedSize());

}

public void init()

{

setBackground(bg); setForeground(fg);

}

public Dimension getMinimumSize()

{

return getPrefferedSize();

}

public Dimension getPrefferedSize()

{

return new Dimension(180,100);

}

public void paintComponent(Graphics g2)

{

super.paintComponent(g2); Graphics2D g = (Graphics2D) g2; Dimension boyut=getSize();

int dx=boyut.width/2; int dy=boyut.height;

g.setColor(getBackground()); g.draw3DRect(0,0,boyut.width-1,boyut.height-1,true); g.draw3DRect(3,3,boyut.width-7,boyut.height-7,false);

//cizgiciz

yildizSW.drawYildiz(g,dx/2,dy/2, 3,dy/2-5,0 ); g.setPaint(renk.kirmizi); yildizSW.fillYildiz(g,dx/2,dy/2, 3,dy/2-5,0 ); g.setPaint(renk.mavi); yildizSW.drawYildiz1(g,3\*dx/2,dy/2, 3,dy/2-5,0 );

}

public void actionPerformed(ActionEvent e)

{

repaint();

}

public static void main(String s[]) {

yildizciz2Dbean yildiz=new yildizciz2Dbean(); JFrame f = new JFrame("yildiz ciz 2D ");

f.getContentPane().add(yildiz,BorderLayout.CENTER); f.addWindowListener(

new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

f.setSize(yildiz.getPrefferedSize().width+10, yildiz.getPrefferedSize().height+30);

f.show();

}

}

Bölüm 7.14 de bir tür program paketleme yolu olan jar deyimi ve kullanilmasindan kisaca bahsedilmisti. Burada programimizi bir jar dosyasi haline getirecegiz. Fakat bunu yaparken ayni zamanda hem jarin içinde hangi dosyanin ana programi içerdigini, hemde Java-bean olarak kullanilma izni veren Java\_Bean komutunu veren **MANIFEST:MF** dosyasini olusturacagiz. Bunu yapmak için **manifest.tmp** isimli birdosya hazirlayacagiz. Bu dosyanin içerigi :

### Main-Class: yildiz2Dbean

**Name: yildiz2Dbean.class Java-Bean: True**

seklinde olacaktir. yildiz.jar isimli jar dosyasini olusturmak için:

### jar cfm yildiz.jar manifest.tmp yildizciz2Dbean.class yildizciz2Dbean$1.class renk.class yildizSW.class polar.class renk.class

komutunu kullanabiliriz. Burada yildizSW da kullanilan polar, renk siniflarinin da jar gurubuna ilave edildigine dikkat edelim. Bu komut yildiz.jar dosyasini yaratacaktir. Simdi programimizi

### java -jar yildiz.jar

komutunu kullanarak çalistirabiliriz. Program

### java -cp yildiz.jar yildiz2Dbean

komutuyla da çalistirilabilir.

yildiz.jar dosyasinin içerigini görmek istersek

### jar tvf yildiz.jar

komutunu kullanabiliriz. Bu komut bize

0 Sat Mar 04 11:18:32 GMT+02:00 2000 META-INF/

143 Sat Mar 04 11:18:32 GMT+02:00 2000 META-INF/MANIFEST.MF

2454 Sat Mar 04 11:18:26 GMT+02:00 2000 yildizciz2Dbean.class

410 Sat Mar 04 11:18:26 GMT+02:00 2000 yildizciz2Dbean$1.class

1924 Mon Dec 13 15:21:50 GMT+02:00 1999 renk.class

3345 Sat Mar 04 08:46:04 GMT+02:00 2000 yildizSW.class

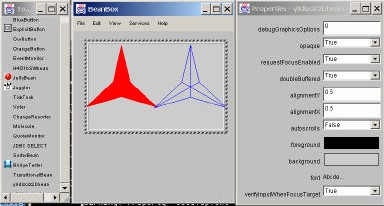
1775 Sat Mar 04 08:45:48 GMT+02:00 2000 polar.class

sonucunu verecektir. Tanimladigimiz yildiz.jar dosyasi ayni zamanda bir JavaBean dosyasidir. Bu dosyayi jars alt direktörüsüne ytasidiktan sonra beanbox direktörüüsndeki run.bat komutunu kullanarak beanbox programimizi açabilir ve bu ortamda gösterebiliriz. Sekil 15.2 de Java Bean haline gelmis olan programimiz görülmektedir.

Ikinci bir örnek olarak ayni yildiz çizme prosesini amimasyon olarak yapacak yildizAnimasyon.java programini veriyoruz. Bu programda kullanilan stopAnimation() ve startAnimation() metodlari animasyonu baslatma ve durdurma islevlerini görmektedirler. Animasyon basladiginda çizilen yildiz dönmektedir. Program 15.3 de bu programin listesi verilmistir.

yildizAnimator programi bir önceki yildizciz2Dbean programina göre biraz daha detaylidir. örnegin yildizin kanat sayisini degistirebilen metodlara sahiptir. Burada da bir öceki programla ayni yollari izleyerek yildizAnimator.jar dosyasini olusrurduk ve jars alt direktörüsüne yerlestirdik.

### 15002.JPG



**Sekil 15.2 yildiz2Dbean programinin BeanBox'da görünümü Program 15.3 yildizAnimator.java programi**

import java.awt.\*; import java.awt.event.\*; import java.io.\*;

import java.net.\*; import javax.swing.\*;

public class yildizAnimator extends JPanel implements ActionListener, Serializable

{

protected int animationDelay; protected Timer animationTimer; protected int yildizkosesayisi; protected double aci;

protected double piaci=Math.PI/360;

public yildizAnimator()

{

setSize( getPreferredSize() ); startAnimation(); yildizkosesayisi=5;

aci=0;

animationDelay = 1000; // 1000 milisaniye gecikme setForeground(renk.mavi);

}

public void paintComponent( Graphics g2 )

{

super.paintComponent( g2 ); Graphics2D g = (Graphics2D) g2; Dimension boyut=getSize();

int dx=boyut.width; int dy=boyut.height;

g.setColor(getBackground()); g.setPaint(getForeground());

yildizSW.drawYildiz1(g,dx/2,dy/2, yildizkosesayisi,dy/2-5,aci); aci+=piaci;

}

public void setYildizkosesayisi(int syks)

{

yildizkosesayisi=syks;

}

public int getYildizkosesayisi()

{

return yildizkosesayisi;

}

public void setAci(int iaci)

{

aci=iaci;

}

public double getAcii()

{

return aci;

}

public void setAnimationDelay(int iad)

{

animationDelay=iad;

}

public int getAnimationDelay()

{

return animationDelay;

}

public void actionPerformed( ActionEvent e )

{

repaint();

}

public void startAnimation()

{

if ( animationTimer == null )

{

animationTimer = new Timer( animationDelay, this ); animationTimer.start();

}

else // continue

if ( ! animationTimer.isRunning() )

animationTimer.restart();

}

public void stopAnimation()

{

animationTimer.stop();

}

public Dimension getMinimumSize()

{

return getPreferredSize();

}

public Dimension getPreferredSize()

{

return new Dimension( 160, 80 );

}

public static void main( String args[] )

{

yildizAnimator anim = new yildizAnimator();

JFrame app = new JFrame( "yildiz Animator testi" ); app.getContentPane().add( anim, BorderLayout.CENTER );

app.addWindowListener( new WindowAdapter() {

public void windowClosing( WindowEvent e )

{

System.exit( 0 );

}

}

);

app.setSize( anim.getPreferredSize().width + 10, anim.getPreferredSize().height + 30 );

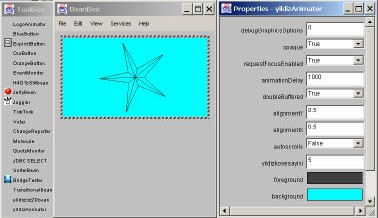
app.show();

}

}

yildizAnimasyon.jar dosyasinin BeanBox'daki görüntüsü :

### 15003.JPG

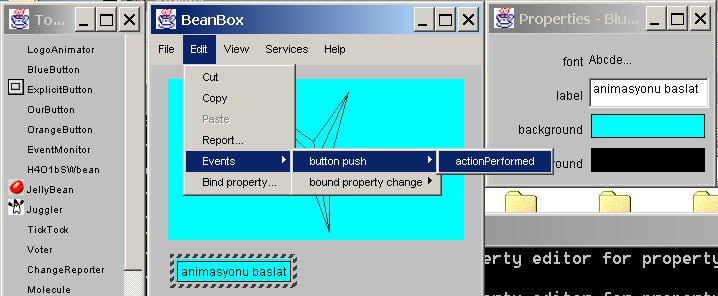


**Sekil 15.3 yildizAnimator.jar programinin BeanBox'ta görünümü**

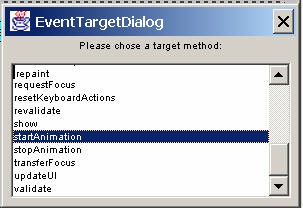
BeanBox Properties penceresini kullanarak program çalisirken özelliklerini degistirebilirsiniz. Örnegin yildizin ve arka planin çizim renklerini degistirebilirsiniz.

Su ana kadar yeni bir ortamda çalismakla birlikte bu bölümün basinda söyledigimiz gibi programlama olmadan çalistigimizi söylemek pek mümkün degil. Fakat bundan sonraki kisimlarda programi bilgisayar olusturacaktir. Örnegin yildiz döndüren yildizAnaimator programimiza animasyonu baslatacak ve durduracak komutlar ilave etmek isteyelim. Bunun için yeni bir program yazmak mümkündür. Fakat program yazmadan bu yeni düzenlemeyi BeanBox üzerinden gerçeklestirebiliriz. Bunu yapmak için önce beanBox'ta daha önce hazir olarak tanimlanmis bulunan **ExplicitButton** bean'ini seçelim ve bean box'a ilave edelim. Daha sonra properties penceresindeki , "press" yazisini "animasyonu baslat" olarak degistirelim. Ayni islemi tektrarlayarak "animasyonu durdur dügmesi olusturalim. Sonra Animasyomu baslat dügmesini seçelim. BeanBox da **Edit, Events, button push,actionPerformed** dizisini seçtikten sonra faremizin ucunda bir dogru olusacaktir. bu dogruyu yildizAnimasyon penceresinin içine çekerek sol mouse dügmesine basalim.

### 15004.JPG



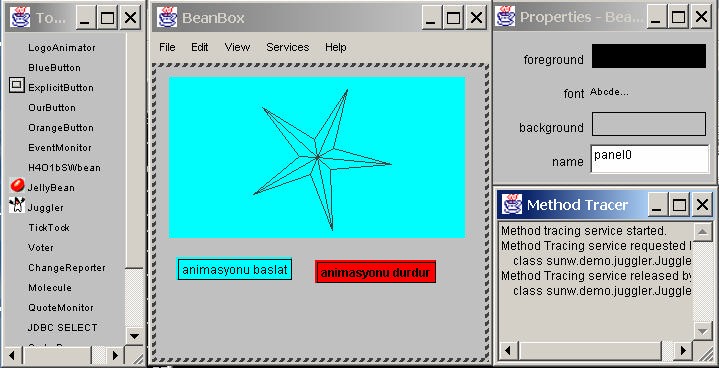
**15005.JPG**



**Sekil 15.4-15.5 BeanBox'ta startAnimation metodunun" Animasyonu baslat" dügmesine actionPerformed olarak eklenmesi.**

EventTargetDialog penceresi ekrana gelecektir. Bu pencerede startAnimation metodunu seçelim ve OK dügmesine basalim. Bu yeni program yeni bir kod olusturaracaktir. Ayni seyi Animasyonu durdur dügmesi için de tekrarliyalim. yalniz bu kez stopAnimation metodunu seçelim. Yeni programimiz animasyonu durduran ve tekrar çalistiran dügmeleriyle birlikte hazir hale gelecektir. Dügmelere basarak çalismasini deneyiniz.

### 15006.JPG



**Sekil 15.6 BeanBox'ta startAnimation metodunun" tamamlanmis sekli**

Olusan yeni programi kaydedebiliriz. Bilgisayarin olusturdugu java program kodu asagidaki gibidir.

### Program 15.4 yildizAnimator.java programina BeanBox'da animasyon baslatma ve durdurma dügmelerinin eklenmesinden sonra otomatik olarak olusturulan serizalizable MyApplet.java program

import java.util.Hashtable; import java.io.Serializable; import java.io.InputStream; import java.io.OutputStream;

import java.io.ObjectOutputStream; import java.io.ObjectInputStream; import java.awt.Component;

import java.awt.Rectangle; import java.awt.Dimension; import java.applet.Applet; import java.lang.reflect.Method; import java.beans.Beans;

import sunw.beanbox.AppletSupport; import sunw.beanbox.PropertyHookup;

public class MyApplet extends Applet implements Serializable {

// Public 0-argument constructor public MyApplet() {

InputStream is = this.getClass().getResourceAsStream("MyAppletData");

if (is == null) {

System.err.println("Could not locate the Applet data at MyAppletData"); throw new Error("Could not locate the Applet data at MyAppletData");

}

this.setLayout(null); try {

ObjectInputStream ois = new ObjectInputStream(is); initContentsFromStream(ois);

} catch (Exception ex) {

System.err.println("trouble reading from serialized data: "+ex); throw new Error("trouble reading from serialized data: "+ex);

}

}

// Preferred size

public Dimension getPreferredSize() { return getMinimumSize();

}

// Preferred size

public Dimension getMinimumSize() { return new Dimension(382, 161);

}

// Read hidden-state beans from stream

private void initContentsFromStream(java.io.ObjectInputStream ois) throws java.lang.ClassNotFoundException,

java.io.IOException

{

Object[] data = (Object[]) ois.readObject();

String id = (String) data[0];

if (! id.equals("MyApplet")) { throw new Error("Wrong data!");

}

// Get references to hidden-state beans

explicitButton1 = (sunw.demo.buttons.ExplicitButton) data[1]; explicitButton2 = (sunw.demo.buttons.ExplicitButton) data[2];

// Initialize the remainder of the applets contents

// including acquiring its nested beans and reconnecting hookups. initContents();

}

// Initialize nested beans private void initContents()

throws java.lang.ClassNotFoundException, java.io.IOException

{

myLoader = this.getClass().getClassLoader(); propInstances = new Hashtable();

// Create nested beans

yildizAnimator1 = (yildizAnimator) Beans.instantiate(myLoader,"yildizAnimator");

// position all nested beans - we don't have it initially acquire(yildizAnimator1, new Rectangle(10, 3, 160, 153));

acquire(explicitButton1, new Rectangle(205, 9, 150, 24));

acquire(explicitButton2, new Rectangle(206, 46, 148, 24));

// Add their connections addConnections();

}

private void addConnections() { try {

//

hookup0 = new tmp.sunw.beanbox. Hookup\_163021aed5(); hookup0.setTarget(yildizAnimator1); explicitButton1.addActionListener(hookup0);

//

hookup1 = new tmp.sunw.beanbox. Hookup\_163021cf3b();

hookup1.setTarget(yildizAnimator1); explicitButton2.addActionListener(hookup1);

//

// No property adaptors. A typical example is:

// hookup0 = addPropertyTarget(

// ourButton1,

// "foreground",

// "background",

// "setBackground", new String[] {"java.awt.Font"});

} catch (Exception ex) {

System.err.println("Problems adding a target: "+ex); ex.printStackTrace();

}

}

private void addReconnections() { try {

//

//

//

// No property adaptors. A typical example is:

// hookup0 = addPropertyTarget(

// ourButton1,

// "foreground",

// "background",

// "setBackground", new String[] {"java.awt.Font"});

// No method adaptors. A typical example is:

// hookup0 = new MyActionAdaptor();

// ourButton1.addActionListener(hookup0);

} catch (Exception ex) {

System.err.println("Problems adding a target: "+ex); ex.printStackTrace();

}

}

// Serialization code - readObject

private void readObject(java.io.ObjectInputStream ois) throws java.lang.ClassNotFoundException, java.io.IOException

{

// Initialize object from stream

Object[] data = (Object[]) ois.readObject();

myLoader = this.getClass().getClassLoader(); propInstances = new Hashtable();

String id = (String) data[0];

if (! id.equals("MyApplet")) { throw new Error("Wrong data!");

}

// Get references to nested beans yildizAnimator1 = (yildizAnimator) data[1];

explicitButton1 = (sunw.demo.buttons.ExplicitButton) data[2]; explicitButton2 = (sunw.demo.buttons.ExplicitButton) data[3];

// Don't position nested beans acquire(yildizAnimator1, null); acquire(explicitButton1, null); acquire(explicitButton2, null);

// Reconnect their connections addReconnections();

}

// Serialization code - writeObject

private void writeObject(java.io.ObjectOutputStream oos) throws java.io.IOException

{

Object data[] = new Object[4]; data[0] = "MyApplet";

data[1] = yildizAnimator1; data[2] = explicitButton1; data[3] = explicitButton2;

// Write the object out oos.writeObject(data);

}

// The fields used to hold the beans private yildizAnimator yildizAnimator1;

private sunw.demo.buttons.ExplicitButton explicitButton1; private sunw.demo.buttons.ExplicitButton explicitButton2;

// The hookups

private tmp.sunw.beanbox. Hookup\_163021aed5 hookup0; private tmp.sunw.beanbox. Hookup\_163021cf3b hookup1;

// No property adaptors. A typical example is:

// private PropertyHookup hookup1;

// the loader so we can locate the resource file private ClassLoader myLoader;

// ===========================================

// Support code

// ===========================================

// It really belongs in support.jar but it is here for your reading pleasure

// Acquire a bean

private void acquire(Object bean, Rectangle boundsData) { if (!(bean instanceof Component)) {

return;

}

if (bean instanceof Applet) { AppletSupport.assignStub((Applet) bean,

myLoader, bean.getClass());

}

add((Component) bean); if (boundsData != null) {

((Component) bean).setBounds(boundsData);

}

((Component)bean).invalidate(); // not needed?

if (bean instanceof Applet) {

// Start the Applet ((Applet)bean).start();

}

}

// Add a property bound via an adaptor Hashtable propInstances = new Hashtable();

private PropertyHookup addPropertyTarget(Object source, String propertyName,

Object targetObject,

String setterName, String[] setterTypeNames) throws Exception

{

Object args[] = new Object[1]; // arguments Class types[] = new Class[1]; // types

Class pClass = java.beans.PropertyChangeListener.class;

PropertyHookup hook = (PropertyHookup) propInstances.get(source); if (hook == null) {

// This is the first property hookup on this source object

// Push a PropertyHookup adaptor onto the source hook = new PropertyHookup(source); propInstances.put(source, hook);

// find the adder types[0] = pClass;

Method adder = source.getClass().getMethod("addPropertyChangeListener", types);

// invoke the adder args[0] = hook;

adder.invoke(source, args);

}

// get setter

Method setter = targetObject.getClass().getMethod(setterName, getClassFromTypes(setterTypeNames));

hook.attach(propertyName, targetObject, setter); return hook;

}

private Class[] getClassFromTypes(String types[]) throws Exception { Class[] back = new Class[types.length];

for (int i=0; i<back.length; i++) {

Class c = unwrapPrimitiveStringToClass(types[i]);

if (c == null)

back[i] = myLoader.loadClass(types[i]); else

back[i] = c;

}

return back;

}

private Class unwrapPrimitiveStringToClass(String s) {

if (s.equals(Byte.TYPE.getName())) return byte.class; if (s.equals(Short.TYPE.getName())) return short.class; if (s.equals(Integer.TYPE.getName())) return int.class; if (s.equals(Long.TYPE.getName())) return long.class;

if (s.equals(Double.TYPE.getName())) return double.class; if (s.equals(Float.TYPE.getName())) return float.class;

if (s.equals(Character.TYPE.getName())) return char.class;

if (s.equals(Boolean.TYPE.getName())) return boolean.class; if (s.equals(Void.TYPE.getName())) return void.class; return null;

}

// ===========================================

// End of Support code

// ===========================================

}

Süphesiz bu program kodunu biz çok daha kisa ve etkili ekilde yazabilirdik, ancak otomatik kod olusturmanin zamansal olarak sagladigi avantaj ile bizim harcayacagimiz zamanin bize ilave maliyeti hesaplanarak optimal çözümler olusturulabilir.

### 15.2 ALISTIRMALAR

1. Alistirma 3.13 ve alistirma 14.10 da üzerinde çalistigimiz bilimsel hesap makinasi programini Java bean olarak tekrar düzenleyelim .

import javax.swing.\*; // java swing sinifini cagir import java.awt.\*;

import java.awt.event.\*; import java.io.\*;

import java.net.\*;

public class H4O1bSWbean extends JApplet implements ActionListener,Serializable

{

// bilimsel hesap makinasi

JTextField sonucgirdisi;

JTextField kutugirdisi;

JButton Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;

JButton Pow,Cos,Sin,Tan,RtoD,DtoR,Acos,Asin,Atan,Exp,Kok,Kare,birBoluX; JButton Ln,Log10,Oku,PI;

//Button sinifi degiskenleri double sayi; bilimselhesapmakinasi1 D;

// pencereyi baslatma metodu

public H4O1bSWbean()

{

D=new bilimselhesapmakinasi1(); Container c=getContentPane(); c.setLayout(new BorderLayout()); sonucgirdisi=new JTextField(); sonucgirdisi.setEditable(false); sonucgirdisi.setBackground(Color.green);

sonucgirdisi.setFont(new Font("SansSerif",Font.BOLD,14)); JPanel e=new JPanel();

e.setLayout(new GridLayout(2,1)); kutugirdisi=new JTextField(40);

kutugirdisi.setFont(new Font("SansSerif",Font.BOLD,14)); e.add(sonucgirdisi);

e.add(kutugirdisi); c.add(e,BorderLayout.NORTH); Gir=new JButton(" Gir "); Oku=new JButton(" Oku "); Arti=new JButton(" + "); Eksi=new JButton(" - "); Carpi=new JButton(" \* "); Bolu=new JButton(" / "); M=new JButton("M"); MR=new JButton("MR");

MArti=new JButton("M+"); MEksi=new JButton("M-"); Isaret=new JButton("+/-"); C=new JButton("C"); Pow=new JButton("x^y"); Cos=new JButton("cos"); Sin=new JButton("sin"); Tan=new JButton("tan"); RtoD=new JButton("R->D"); DtoR=new JButton("D->R"); Acos=new JButton("acos"); Asin=new JButton("asin"); Atan=new JButton("atan"); Exp=new JButton("exp"); Ln=new JButton("ln"); Log10=new JButton("log10"); Kok=new JButton("x^0.5"); Kare=new JButton("x^2"); birBoluX = new JButton("1/x"); PI=new JButton("pi");

JPanel m=new JPanel(); m.setLayout(new GridLayout(5,8)); m.add(Gir);

m.add(Oku);

m.add(Arti);

m.add(Eksi);

m.add(Carpi);

m.add(Bolu);

m.add(M);

m.add(MR);

m.add(MArti);

m.add(MEksi); m.add(Isaret); m.add(C);

m.add(Pow);

m.add(Cos);

m.add(Sin);

m.add(Tan);

m.add(DtoR);

m.add(RtoD); m.add(birBoluX); m.add(Acos);

m.add(Asin);

m.add(Atan);

m.add(Kok);

m.add(Kare);

m.add(Exp);

m.add(Ln);

m.add(Log10);

m.add(PI); c.add(m,BorderLayout.SOUTH); Pow.addActionListener(this); Cos.addActionListener(this); Sin.addActionListener(this); Tan.addActionListener(this); DtoR.addActionListener(this); RtoD.addActionListener(this); Acos.addActionListener(this); Asin.addActionListener(this); Atan.addActionListener(this);

Exp.addActionListener(this); Ln.addActionListener(this); Log10.addActionListener(this); Gir.addActionListener(this); Oku.addActionListener(this); Arti.addActionListener(this); Eksi.addActionListener(this); Carpi.addActionListener(this); Bolu.addActionListener(this); M.addActionListener(this); MR.addActionListener(this); MArti.addActionListener(this); MEksi.addActionListener(this); Isaret.addActionListener(this); C.addActionListener(this); Kok.addActionListener(this); Kare.addActionListener(this); birBoluX.addActionListener(this); PI.addActionListener(this);

}

// girdi alanindaki olan olaylari dinleme metodu public void actionPerformed(ActionEvent e)

{

String ss=kutugirdisi.getText(); if(ss.equals("")) sayi=0.0;

else

{

Double sayi1=new Double(kutugirdisi.getText() ); sayi=sayi1.doubleValue();

}

if(e.getSource()==Gir) D.gir(sayi); if(e.getSource()==Oku) D.oku(); if(e.getSource()==PI) D.pi();

else if(e.getSource()==Arti) D.topla(sayi); else if(e.getSource()==Eksi) D.cikar(sayi); else if(e.getSource()==Carpi) D.carp(sayi); else if(e.getSource()==Bolu) D.bol(sayi); else if(e.getSource()==M) D.M(sayi); else if(e.getSource()==MR) D.MR();

else if(e.getSource()==MArti) D.MTopla(sayi); else if(e.getSource()==MEksi) D.Mcikar(sayi); else if(e.getSource()==Isaret) D.isaretdegistir(sayi); else if(e.getSource()==C) D.C();

else if(e.getSource()==Pow) D.pow(sayi); else if(e.getSource()==Cos) D.cos(sayi); else if(e.getSource()==Sin) D.sin(sayi); else if(e.getSource()==Tan) D.tan(sayi); else if(e.getSource()==RtoD) D.RtoD(sayi); else if(e.getSource()==DtoR) D.DtoR(sayi); else if(e.getSource()==Acos) D.acos(sayi); else if(e.getSource()==Asin) D.asin(sayi); else if(e.getSource()==Atan) D.atan(sayi); else if(e.getSource()==Exp) D.exp(sayi); else if(e.getSource()==Ln) D.ln(sayi);

else if(e.getSource()==Log10) D.log10(sayi); else if(e.getSource()==Kok) D.kok(sayi); else if(e.getSource()==Kare) D.kare(sayi);

else if(e.getSource()==birBoluX) D.bir\_bolu\_x(sayi); sonucgirdisi.setText(D.toString());

if(e.getSource()==MR || e.getSource()==M || e.getSource()==MArti||e.getSource()==MEksi)

{

kutugirdisi.setText(D.MtoString());

}

else if(e.getSource()==Cos || e.getSource()==Sin||e.getSource()==Tan||e.getSource()==RtoD || e.getSource()==DtoR || e.getSource()==Acos || e.getSource()==Ln || e.getSource()==Asin||e.getSource()==Atan||e.getSource()==Exp || e.getSource()==Log10||e.getSource()==Isaret ||e.getSource()==Kok || e.getSource()==Kare || e.getSource()==birBoluX || e.getSource()==Oku || e.getSource()==PI)

{

kutugirdisi.setText(D.aratoString());

}

else

{

kutugirdisi.setText("");

}

}

public static void main(String s[]) {

JFrame f = new JFrame("Hesap Makinasi Java Programlama dili"); f.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {System.exit(0);}

});

JApplet applet = new H4O1bSWbean(); f.getContentPane().add("Center", applet); applet.init();

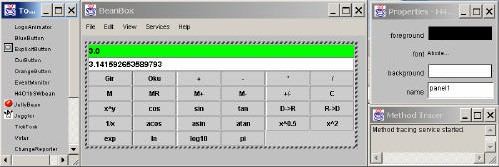
f.pack();

f.setSize(new Dimension(450,220)); f.show();

}

}

### 15005.JPG



Sekil 15.5 H4O1bSWbean.java programinin java BeanBox'ta görünümü

### NETWORK PROGRAMLAMAYA GIRIS

### TCP/IP PROTOKOLÜ

Java network protokolü olarak TCP/IP protokolünü kullanir. TCP ingilizce Transmission interface protokol (Veritransferi arayüz protokolü) IP ise Internet protokolü anlamina gelir.

IP kodu veriyi su alt paketcikler olarak tasir : 0-10 : baslik

10-12 : kontrol

12-16 gönderici adresi

16-20 gidecegi adres

20-24 : seçenekler 24....gönderilen bilgi

IP bilgi paketlerinin yerine ulasma garantisiyoktur. TCP protokolü gitme garantili paketler olusturur. IP paketini telgrafa benzetebiliriz. TCP paketleri ise telefon servisi gibidir. Karsilikli iki tarafi baglar. TCP paketinde alt birimler sunlardir :

0-2 gönderici adresi

2-4 gidecek yerin adresi 4-8 paket numarasi

8-12 paket varis numarasi 12-20 basliklar

20+ veri

TCP kanallarindan bazilari özel görevler görürler. Örnegin kanal 7 gönderilen bilgiyi geriye yansitir. Kanal 13 verinin gittigi makinanin yerel zamanini bildirir. Kanal 20-21 ftp protokolü, kanal 23 telnet protokolü, kanal 80 http protokolü tarafindan kullanilir. Javada internet adreslerini proses aden sinifimiz InetAdress sinifidir.

package java.net;

public final class InetAddress implements java.io.Serializable { InetAddress() ;

InetAddress(String hostName, byte addr[]);

public static InetAdress[] getAllByName(String Host) throws UnknownHostException; public static InetAdress getByName(String Host) throws UnknownHostException ; public static synchronised InetAdress getLocalHost() throws UnknownHostException; public boolean isMulticastAddress();

public byte[9 getAdress[] ; public String getHostName();

String getHostName(boolean check); public String getHostAddress() ; public int hashCode() ;

public boolean equals(Object obj); public String toString();

}

Inet adres sinifini kullanarak DSN server sitelerinin adresine ulasabiliriz. Bunu yapan bir program kodu asagida verilmistir.

Program 16.1 DNSadres.java programi import java.net.\*;

public class DNSadres {

public static void main (String args[]) throws UnknownHostException { InetAddress someHost;

byte bytes[];

int fourBytes[] = new int[4];

if (args.length == 0) {

someHost = InetAddress.getLocalHost();

} else {

someHost = InetAddress.getByName (args[0]);

}

System.out.print ("site '"+ someHost.getHostName() +"' sitenin adresi : "); bytes = someHost.getAddress();

for (int i=0; i<4; i++) { fourBytes[i] = bytes[i] & 255;

}

System.out.println (fourBytes[0] + "." +

fourBytes[1] + "." + fourBytes[2] + "." + fourBytes[3]);

System.out.println("Local bilgisayarin adresi : "+InetAddress.getLocalHost());

}

}

program çalistirilinca :

site 'www.mam.gov.tr' sitenin adresi : 193.140.76.10

Local bilgisayarin adresi : turhan-czvbccoqtf/193.140.79.15 sonucunu verecektir.

### ALT SEVIYE ILETISIM : UDP KULLANIMI

network üzerinden kurabilecegimiz en basit iletisim yollarindan biri olan UDP DatagramPacket ve DatagramSocket siniflarini kullanarak gerçeklestirilebilir.

DatagramPacket sinifi dört kurucu metod içermektedir. Bu metodlardan ikisi veri almak, ikisi de veri göndermek amaçlidir. Hem gönderirkenhemde veri alirken byte olarak veriyi ve veri uzunlugunu bildirme zorunlugu mevcuttur. Bu sinif asil gönderme islemlerini yapmaz, bu parelel sinifi DatagramSocket tarafindan gerçeklestirilir.

package java.net;

public final class DatagramPacket { byte[] buf;

int offset; int length;

InetAddress address; int port;

public DatagramPacket(byte buf[], int offset, int length) public DatagramPacket(byte buf[], int length)

public DatagramPacket(byte buf[], int offset, int length,

InetAddress address, int port) public DatagramPacket(byte buf[], int length,

InetAddress address, int port) public synchronized InetAddress getAddress()

public synchronized int getPort() public synchronized byte[] getData() public synchronized int getOffset public synchronized int getLength()

public synchronized void setData(byte[] buf, int offset, int length) public synchronized void setAddress(InetAddress iaddr)

public synchronized void setPort(int iport) public synchronized void setData(byte[] buf) public synchronized void setLength(int length) private native static void init()

}

package java.net;

public class DatagramSocket { DatagramSocketImpl impl; boolean connected = false;

InetAddress connectedAddress = null; int connectedPort = -1;

public DatagramSocket() throws SocketException

public DatagramSocket(int port) throws SocketException

public DatagramSocket(int port, InetAddress laddr) throws SocketException void create(int port, InetAddress laddr) throws SocketException

public void connect(InetAddress address, int port) public void disconnect()

public InetAddress getInetAddress() public int getPort

public void send(DatagramPacket p) throws IOException { public synchronized void receive(DatagramPacket p) throws public InetAddress getLocalAddress()

public int getLocalPort()

public synchronized void setSoTimeout(int timeout) throws SocketException public synchronized int getSoTimeout() throws SocketException

public synchronized void setSendBufferSize(int size) throws SocketException public synchronized int getSendBufferSize() throws SocketException

public synchronized void setReceiveBufferSize(int size) throws SocketException public synchronized int getReceiveBufferSize() throws SocketException

public void close()

}

Bu iki sinifi bir örnek problemde kullanalim. **Program 16.2 GetDate.java programi** import java.net.\*;

public class GetDate {

final static int PORT\_DAYTIME = 13; // well-known daytime port

public static void main (String args[]) throws Exception { DatagramSocket dgSocket;

DatagramPacket datagram; InetAddress destination; byte msg[] = new byte[256];

dgSocket = new DatagramSocket();

destination = InetAddress.getByName ("mam.gov.tr");

datagram = new DatagramPacket (msg, msg.length, destination,

PORT\_DAYTIME);

dgSocket.send(datagram);

datagram = new DatagramPacket (msg, msg.length); dgSocket.receive(datagram);

String received = new String (datagram.getData()); System.out.println ("TÜBITAK MAM'da su anda saat: " + received); dgSocket.close();

}

} // end of class GetDate program çiktisi :

TÜBITAK MAM'da su anda saat: Sun Mar 11 16:14:20 2001

Program çiktisindan da görüldügü gibi bu program zaman port’undan mam.gov.tr bilgisayarinin zamanini okumaktadir.

### TCP BILGI ILETIMI, SOCKET SINIFI

Direk olarak TCP üzerinden bilgi aktarma UDP üzerinden bilgi aktarimi gibidir. En önemli fark, TCP kullanirken veri miktarinin önemli olmamasidir. Istenilen miktarda veri TCP lanali üzerinden aktarilabilir. TCP veri aktarimi için socket sinifini kullaniyoruz. Socket sinifinin tanimi :

package java.net; public class Socket {

protected Socket()

protected Socket(SocketImpl impl) throws public Socket(String host, int port)

throws UnknownHostException, IOException

public Socket(InetAddress address, int port) throws IOException public Socket(String host, int port, InetAddress localAddr,

int localPort) throws IOException

public Socket(InetAddress address, int port, InetAddress localAddr, int localPort) throws IOException

public Socket(String host, int port, boolean stream) throws IOException public Socket(InetAddress host, int port, boolean stream) throws

public InetAddress getInetAddress() public InetAddress getLocalAddress() public int getPort()

public InputStream getInputStream() throws IOException public OutputStream getOutputStream() throws IOException

public void setTcpNoDelay(boolean on) throws SocketException public boolean getTcpNoDelay() throws SocketException

public void setSoLinger(boolean on, int linger) throws SocketException public int getSoLinger() throws SocketException

public synchronized void setSoTimeout(int timeout) throws SocketException public synchronized int getSoTimeout() throws SocketException

public synchronized void setSendBufferSize(int size) throws SocketException

public synchronized int getSendBufferSize() throws SocketException public synchronized void setReceiveBufferSize(int size)

throws SocketException

public synchronized int getReceiveBufferSize() throws SocketException

public String toString()

public static synchronized void setSocketImplFactory(SocketImplFactory fac) throws IOException

}

seklindedir.

### Program 16.3 email.java programi (TCP Socket sinifi örnegi)

import java.io.\*; import java.net.\*;

import javax.swing.JOptionPane; public class email {

public static void main(String args[])

throws IOException, UnknownHostException { String s;

String msgFile;

String from, to, mailHost;

if (args.length != 4) {

JOptionPane.showMessageDialog(null,"Dogru çagirma: java email gönderilendosya gönderenin\_adresi gittigi\_adres mektup\_server\_adresi");

System.exit (10);

}

msgFile = args[0]; from = args[1];

to = args[2]; mailHost = args[3];

checkEmailAddress (from); checkEmailAddress (to);

SMTP mail = new SMTP(mailHost); if (mail != null) {

if (mail.send (new FileReader(msgFile), from, to) ) { JOptionPane.showMessageDialog(null,"Mektup gönderildi");

} else {

JOptionPane.showMessageDialog(null,"STMP server'a baglanti saglanamadi");

}

}

System.exit(0);

}

static void checkEmailAddress (String address) { if (address.indexOf('@') == -1) {

JOptionPane.showMessageDialog(null,"Geçersiz e-mail adresi '" + address + "'"); System.exit (10);

}

}

}

//-------------------------------------------------------------------

class SMTP {

public final static int SMTP\_PORT = 25; InetAddress mailHost;

InetAddress ourselves;

BufferedReader in;

PrintWriter out;

public SMTP (String host) throws UnknownHostException {

mailHost = InetAddress.getByName(host); ourselves= InetAddress.getLocalHost();

System.out.println ("mailhost = " + mailHost); System.out.println ("localhost= " + ourselves); System.out.println ("SMTP constructor done\n");

}

public boolean send (FileReader msgg, String from, String to) throws IOException { Socket smtpPipe;

InputStream inn;

OutputStream outt;

BufferedReader msg;

msg = new BufferedReader (msgg);

smtpPipe = new Socket (mailHost, SMTP\_PORT); if (smtpPipe == null) {

return false;

}

// get raw streams

inn = smtpPipe.getInputStream(); outt = smtpPipe.getOutputStream();

// turn into usable ones

in = new BufferedReader (new InputStreamReader (inn)); out = new PrintWriter (new OutputStreamWriter (outt), true);

if (inn==null || outt==null) {

System.out.println ("Failed to open streams to socket."); return false;

}

String initialID = in.readLine(); System.out.println (initialID);

System.out.println ("HELO " + ourselves.getHostName()); out.println ("HELO " + ourselves.getHostName());

String welcome = in.readLine(); System.out.println (welcome);

System.out.println ("MAIL From:<" + from + ">"); out.println ("MAIL From:<" + from + ">");

String senderOK = in.readLine(); System.out.println (senderOK);

System.out.println ("RCPT TO:<" + to + ">"); out.println ("RCPT TO:<" + to + ">");

String recipientOK = in.readLine(); System.out.println (recipientOK);

System.out.println ("DATA"); out.println ("DATA");

String line;

while ((line = msg.readLine()) != null) { out.println(line);

}

System.out.println (".");

out.println (".");

String acceptedOK = in.readLine(); System.out.println (acceptedOK);

System.out.println ("QUIT"); out.println ("QUIT");

return true;

}

}

programi çalistirmak için :

>java email a.dat [**Turhan.Coban@posta.mam.gov.tr**](mailto:Turhan.Coban@posta.mam.gov.tr)[***turhan@mam.gov.tr***](mailto:turhan@mam.gov.tr)mam.gov.tr komutunu kullanabiliriz.

Program çiktisi :

mailhost = posta.mam.gov.tr/193.140.72.10 localhost= turhan-czcvoqtf/193.140.78.17 SMTP constructor done

220 posta.mam.gov.tr ESMTP Server (Microsoft Exchange Internet Mail Service 5.5.2653.13) ready HELO turhan-czcvoqtf

250 OK

MAIL [From:<turhan@mam.gov.tr>](mailto:turhan@mam.gov.tr)

250 OK - mail from [<turhan@mam.gov.tr>](mailto:turhan@mam.gov.tr) RCPT [TO:<Turhan.Coban@posta.mam.gov.tr>](mailto:Turhan.Coban@posta.mam.gov.tr)

250 OK - Recipient [<Turhan.Coban@posta.mam.gov.tr>](mailto:Turhan.Coban@posta.mam.gov.tr) DATA

.

354 Send data. End with CRLF.CRLF QUIT

Seklinde olusur.

### INTERNET SITELERIYLE BILGI ALISVERISI, URL SINIFI

Java'nin Internet ortaminda programlama için oldukça iyi bir araç oldugundan kitabin basinda bahsetmistik. Bu konuda javanin çok iyi bir araç olmasi nazilarinin Java dilini sadece bir internet araci oldugunu sanmalarina kadar gitmektedir. Java dilinde internet ve serverlerin haberlesmeleri için bir çok araç bulunmaktadir. Bu araçlarin en önemlilerinden biri de süphesiz URL sinifidir. bu sinif yardimiyla bir internet adresiyle (html adresiyle) rahatlikla baglanti kurulabilir. url sinifinin genel tanimi söyledir :

public final class URL implements java.io.Serializable {

static final long serialVersionUID = -7627629688361524110L;

private static final String protocolPathProp = "java.protocol.handler.pkgs"; private String protocol;

private String host;

private transient InetAddress hostAddress; private int port = -1;

private String file; private String ref;

transient URLStreamHandler handler; private int hashCode = -1;

public URL(String protocol, String host, int port, String file) throws MalformedURLException public URL(String protocol, String host, String file) throws MalformedURLException

public URL(String protocol, String host, int port, String file,URLStreamHandler handler) throws MalformedURLException

public URL(String spec) throws

public URL(URL context, String spec) throws MalformedURLException

public URL(URL context, String spec, URLStreamHandler handler) throws MalformedURLException private boolean isValidProtocol(String protocol)

private void checkSpecifyHandler(SecurityManager sm)

protected void set(String protocol, String host, int port, String file, String ref) public int getPort()

public String getProtocol() public String getHost() public String getFile() public String getRef()

public boolean equals(Object obj public synchronized int hashCode()

private synchronized InetAddress getHostAddress() private static boolean hostsEqual(URL u1, URL u2) public boolean sameFile(URL other)

public String toString()

public String toExternalForm()

public URLConnection openConnection() throws java.io.IOException public final InputStream openStream() throws java.io.

public final Object getContent() throws java.io.IOException

public static synchronized void setURLStreamHandlerFactory(URLStreamHandlerFactory fac static synchronized URLStreamHandler getURLStreamHandler(String protocol)

private synchronized void readObject(java.io.ObjectInputStream s) throws IOException, ClassNotFoundException

}

Internetten URL sinifini kullanarak veri aktarimi konusunda iki örnek program verelim. Birinci program seçilen sitelere ulasim yapacak bir internet sitesi seçme programidir.

### Program 16.4 internetsitesisecici.java, bu program applet kullanarak choice deyimiyle listelenmis (programin içinde) internet siteleri arasinda seçim yapar ve sonucda bu internet sitesini browsere çagirir.

import java.awt.\*; import java.awt.event.\*; import java.net.\*;

import java.util.Hashtable; import java.applet.Applet;

public class internetsitesecici extends Applet implements ItemListener

{

private Hashtable site; private Choice siteChoice; public void init()

{

site=new Hashtable(); siteChoice=new Choice(); String baslik,yer;

URL url;

int counter=0; try

{

baslik="TUBITAK Marmara Arastirma Merkezi"; [yer="http://www.mam.gov.tr";](http://www.mam.gov.tr/)

url=new URL(yer); site.put(baslik,url); siteChoice.add(baslik); baslik="Turhan Coban";

[yer="http://www.mam.gov.tr/~turhan";](http://www.mam.gov.tr/%7Eturhan) url=new URL(yer);

site.put(baslik,url); siteChoice.add(baslik);

baslik="TUBITAK Marmara Arastirma Merkezi network sitesi"; [yer="http://www.mam.net.tr";](http://www.mam.net.tr/)

url=new URL(yer); site.put(baslik,url); siteChoice.add(baslik); baslik="Java sitesi"; [yer="http://java.sun.com";](http://java.sun.com/) url=new URL(yer); site.put(baslik,url); siteChoice.add(baslik);

}

catch(MalformedURLException e){e.printStackTrace();} add(new Label("internet sitesi seciniz:")); siteChoice.addItemListener(this);

add(siteChoice);

}

public void itemStateChanged(ItemEvent e)

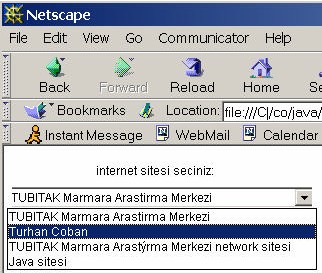
{

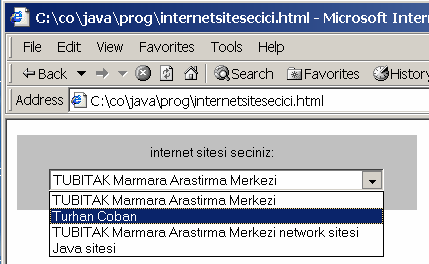
URL url=(URL)site.get(e.getItem()); getAppletContext().showDocument(url);

}

}

### 16001.JPG



**Sekil 16.1 internetsitesecici programini kullanarak bir internet sitesinin seçilmesi (netscape ) 16002.JPG**

**Sekil 16.2 internetsitesecici programini kullanarak bir internet sitesinin seçilmesi (Microsoft internet explorer)**

**16003.JPG**



**Sekil 16.3 internetsitesecici programini kullanarak seçilen internet sitesinin Netscape browsereyle gösterilmesi**

**Program 16.5 internetsitesiseciciSW.java, bu program Japplet (swing) kullanarak listelenmis (programin diinda html dosyasinda tanimlanmis) internet sitelari arasinda seçim yapar ve sonucda bu internet sitesini browsere çagirir.**

import java.net.\*; import java.util.\*; import javax.swing.\*;

import javax.swing.event.\*; import java.awt.\*;

import java.applet.AppletContext;

public class internetsitesiseciciSW extends JApplet { private Hashtable internetsitesi;

private Vector siteIsimleri;

public void init()

{

internetsitesi = new Hashtable(); siteIsimleri = new Vector();

getinternetsitesiFromHTMLParameters(); Container c = getContentPane();

c.add( new JLabel( "Internet internetsitesi seciniz : " ), BorderLayout.NORTH );

final JList siteChooser = new JList( siteIsimleri ); siteChooser.addListSelectionListener(

new ListSelectionListener() {

public void valueChanged( ListSelectionEvent e )

{

Object o = siteChooser.getSelectedValue();

URL newDocument = (URL) internetsitesi.get( o ); AppletContext browser = getAppletContext(); browser.showDocument( newDocument );

}

}

);

c.add( new JscrollPane( siteChooser ), BorderLayout.CENTER );

}

private void getinternetsitesiFromHTMLParameters()

{

//HTML dökümanindaki applet parametrelerini incele ve site isimlerini oku String baslik, yer;

URL url;

int counter = 0;

while ( true ) {

baslik = getParameter( "baslik : " + counter );

if ( baslik != null ) {

yer = getParameter( "adres :" + counter );

try {

url = new URL( yer ); internetsitesi.put( baslik, url ); siteIsimleri.addElement( baslik );

}

catch ( MalformedURLException e ) { e.printStackTrace();

}

}

else

break;

++counter;

}

}

}

bu programin çiktisi da bir önceki programda oldugu gibidir. Bir sonraki programda internet üzerinden verilen adrese giderek dosyayi okumaktayiz. Yani bir anlamda küçük bir browser programi yarattik.

### Program 16.6 InternetDosyaOku.java, bu program küçük bir browser tanimlar. JtextField alanina yazilan URL adresindeki dosyayi JeditorPane kullanarak ekrana getirir.

// Bu program JEditorPane kullanarak

// bir Web sayfasini ekranda göstermektedir. import java.awt.\*;

import java.awt.event.\*; import java.net.\*; import java.io.\*;

import javax.swing.\*; import javax.swing.event.\*;

public class InternetDosyaOku extends JFrame { private JTextField gir;

private JEditorPane contents;

public InternetDosyaOku()

{

super( "Java Programlama dili WEB browser" ); Container c = getContentPane();

gir = new JTextField( "Ulasmak istediginiz adresi buraya yaziniz : " ); gir.addActionListener(

new ActionListener() {

public void actionPerformed( ActionEvent e )

{

getThePage( e.getActionCommand() );

}

}

);

c.add( gir, BorderLayout.NORTH );

contents = new JEditorPane(); contents.setEditable( false ); contents.addHyperlinkListener(

new HyperlinkListener() {

public void hyperlinkUpdate( HyperlinkEvent e )

{

if ( e.getEventType() == HyperlinkEvent.EventType.ACTIVATED ) getThePage( e.getURL().toString() );

}

}

);

c.add( new JScrollPane( contents ), BorderLayout.CENTER );

setSize( 400, 300 ); show();

}

private void getThePage( String adres )

{

setCursor( Cursor.getPredefinedCursor( Cursor.WAIT\_CURSOR ) );

try {

contents.setPage( adres ); gir.setText( adres );

}

catch ( IOException io ) { JOptionPane.showMessageDialog( this,

"internet URL adresine erisilemedi", "URL adresinde hata var",

JOptionPane.ERROR\_MESSAGE );

}

setCursor( Cursor.getPredefinedCursor( Cursor.DEFAULT\_CURSOR ) );

}

public static void main( String args[] )

{

InternetDosyaOku dosya = new InternetDosyaOku();

dosya.addWindowListener( new WindowAdapter() {

public void windowClosing( WindowEvent e )

{

System.exit( 0 );

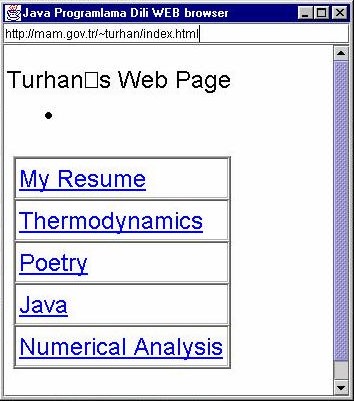
}

}

);

}

}



### Sekil 16.4 internetdosyaoku programini kullanarak seçilen internet sitesinin gösterilmesi

### GÜVENLIK

Güvenlik denince bir programin girdi çikti fonksiyonlari araciligiyla yapabilecegi degisiklikler, bu degisikliklerin etkileri anlasilir. Javada güvenligi saglaak için program girdi çiktilar

### APPLETLERDE GÜVENLIK

Appletler güvenlik açisindan sinirlandirilmis programlardir. Genel olarak yerel bilgisayara yazma ve okuma iznine sahip degillerdir. Bunu bir örnekle açiklayalim. Program 17.1’de bir applet programi olan yazitesti.java verilmistir. Bu program yazitesti.txt dosyasina “bu yazi deneme amaciyla dosya yazitesti.txt içine Appletden yazildi” yazisini yazmak için olusturulmustur.

### Program 17.1 yazitesti.java

import java.awt.\*; import java.io.\*; import java.lang.\*; import javax.swing.\*;

public class yazitesti extends JApplet { String myFile = "yazitesti.txt";

File f = new File(myFile);

PrintWriter dos;

public void paint(Graphics g) { try {

dos=new PrintWriter(new BufferedWriter(new FileWriter(myFile)));

dos.println("bu yazi deneme amaciyla dosya yazitesti.txt içine Appletden yazildi\n"); dos.close();

g.drawString("Yazim islemi basari ile gerçeklesti : " + myFile + " -- dosyasini açip bakabilirsiniz ", 10, 10);

}

catch (SecurityException e) {

g.drawString("yazitesti: güvenlik kontrolu asilamadi (güvenik hatasi) : " + e, 10, 10);

}

catch (IOException ioe) {

g.drawString("yazitesti: giris çikis hatasi", 10, 10);

}

}

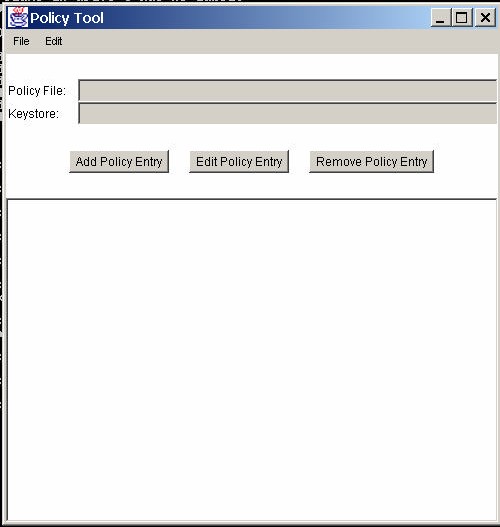
}

Program çalistirildiginda (appletviewer yazitesti.html) 17001.JPG

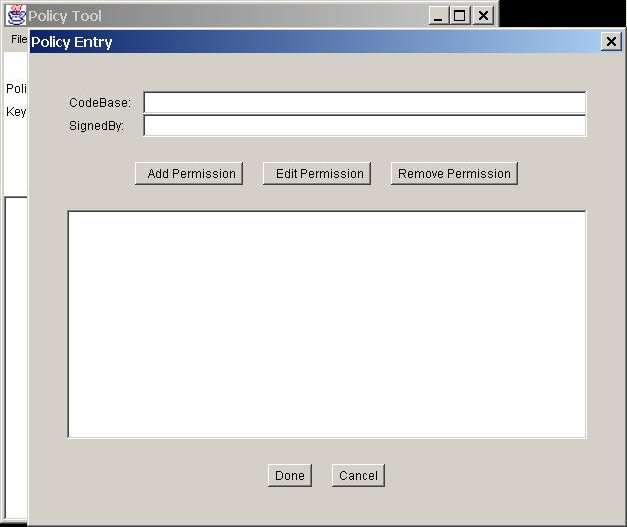
Figure 17.1 yazitesti.html applet çiktisi

Yazitesti:güvenlik kontrolu asilamadi(güvenlik hatasi): java security.AccessControlException:access denied(java.io.FilePermission yazitesti.txt write

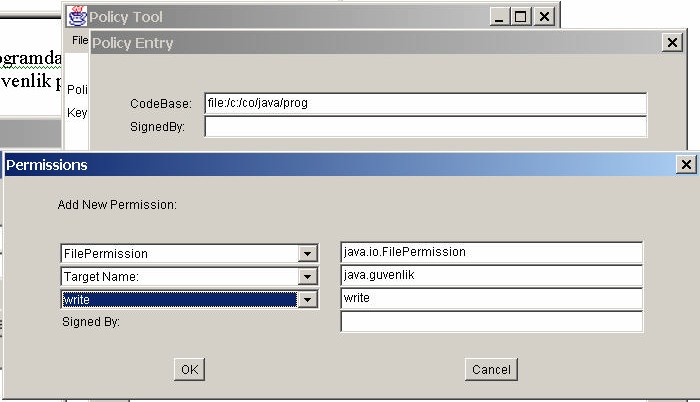
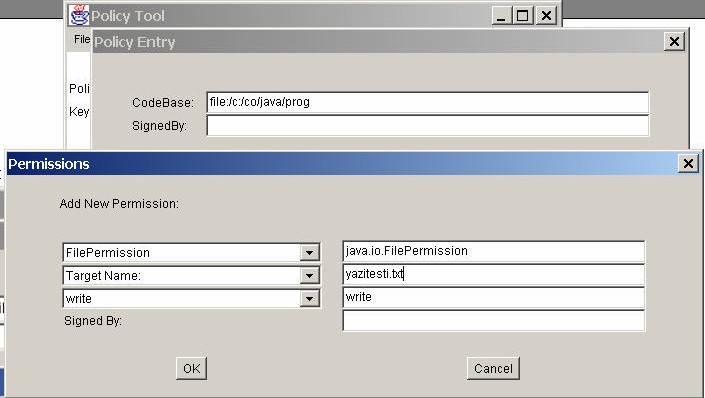
Mesaji verilmistir. yazitesti.txt dosyasi incelendiginde de hiç bir sey yazilmadigi görülür. Bu applet programini kullanarak nasil dosyaya yazi yazdirabiliriz. Bunun için önce **policytool** isimli programi çalistiralim. Ekrana



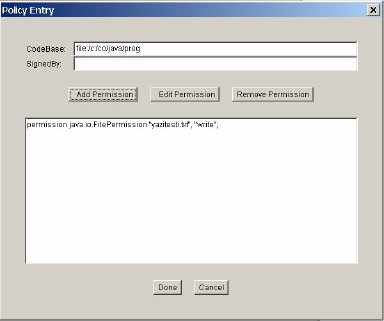
bir java programi gelecektir. Programdaki Add Policy Entry (güvenlik kurali ekle) dügmesine basalim. Yeni bir ekran açilacaktir. Bu ekranda güvenlik programimizi içerecek dosya url adresini yazip (örnegin file:/c:/co/java/prog veya [html://www.mam.gov.tr/~turhan)](http://www.mam.gov.tr/%7Eturhan)) Add Permission tusuna basalim.



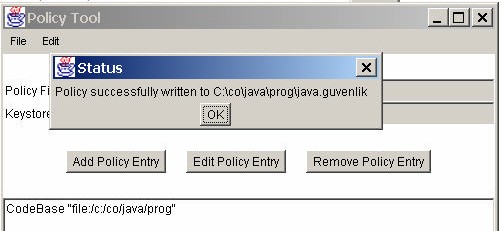
File permission (dosya güvenlik izni) penceresinden file permission (dosya izni) seçtikten sonra Target Name alanina yazma izni istedigimiz yazitesti.txt dosyasinin ismini yazalim. Ve son alanda da yazma izni (“write”) seçelim ve ok dügmesine basalim.



son pencere kapanacak ve bir önceki pencere gelecektir. Burada da Done dügmesine basalim.



ilk pencerede file, save as seçeneklerini seçtikten sonra c:\co\java\prog (veya sizin çalistiginiz dosyaya) java.guvenlik adiyla kaydedelim. Program policy succesfully written to c:\co\java\prog\java.guvenlik mesajini verecektir.



simdi policy tool programini exit deyimiyle kapatabiliriz. Policytool programi bize otomatik olarak

**java.guvenlik** dosyasini yaratti. Bu dosyanin kapsamina bakacak olursak :

/\* AUTOMATICALLY GENERATED ON Sun Mar 11 00:45:40 PST 2001\*/

/\* DO NOT EDIT \*/

grant codeBase "file:/c:/co/java/prog" {

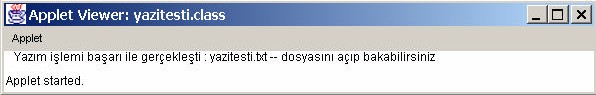
permission java.io.FilePermission "yazitesti.txt", "write";

};

biz ayni komutu dosyaya elde de girebilirdik. Bu komutun anlami c:\co\java\prog direktory’sindeki yazitesti.txt dosyasina yazma (“write”) müsadesi ver demektir.

Simdi appletviewer programini

appletviewer –J-Djava.security.policy=java.guvenlik yazitesti.html komutuyla çalistirdigimizda :



çiktisini alacagiz. Yazitesti.txt dosyasini açtigimizda

### bu yazi deneme amaciyla dosya yazitesti.txt içine Appletden yazildi

dosyaya yazisinin yazildigini görecegiz.

Burada yazimi gerçeklestirmek için **–J-Djava.security.policy=java.guvenlik** gibi özel bir anahtar kullandik. Ayni isi bu anahtari kullanmadan yapmak için **java.security** dosyasina yeni bir komut ilave etmemiz gerekir. Bu dosya sizin kullandiginiz ana java directory’sinin altindaki lib\security de bulunacaktir. Örnegin benim dosyam c:\co\java\jre\lib\security directory’sinde yer almaktadir. Bu dosyanin görünümü :

------------

#

# This is the "master security properties file".

#

# In this file, various security properties are set for use by

# java.security classes. This is where users can statically register

# Cryptography Package Providers ("providers" for short). The term

# "provider" refers to a package or set of packages that supply a

# concrete implementation of a subset of the cryptography aspects of

# the Java Security API. A provider may, for example, implement one or

# more digital signature algorithms or message digest algorithms.

#

# Each provider must implement a subclass of the Provider class.

# To register a provider in this master security properties file,

# specify the Provider subclass name and priority in the format

#

# security.provider.<n>=<className>

#

# This declares a provider, and specifies its preference

# order n. The preference order is the order in which providers are

# searched for requested algorithms (when no specific provider is

# requested). The order is 1-based; 1 is the most preferred, followed

# by 2, and so on.

#

# <className> must specify the subclass of the Provider class whose

# constructor sets the values of various properties that are required

# for the Java Security API to look up the algorithms or other

# facilities implemented by the provider.

#

# There must be at least one provider specification in java.security.

# There is a default provider that comes standard with the JDK. It

# is called the "SUN" provider, and its Provider subclass

# named Sun appears in the sun.security.provider package. Thus, the

# "SUN" provider is registered via the following:

#

# security.provider.1=sun.security.provider.Sun

#

# (The number 1 is used for the default provider.)

#

# Note: Statically registered Provider subclasses are instantiated

# when the system is initialized. Providers can be dynamically

# registered instead by calls to either the addProvider or

# insertProviderAt method in the Security class.

#

# List of providers and their preference orders (see above):

#

security.provider.1=sun.security.provider.Sun security.provider.2=com.sun.rsajca.Provider

#

# Class to instantiate as the system Policy. This is the name of the class

# that will be used as the Policy object.

#

policy.provider=sun.security.provider.PolicyFile

# The default is to have a single system-wide policy file,

# and a policy file in the user's home directory. policy.url.1=file:${java.home}/lib/security/java.policy policy.url.2=file:${user.home}/.java.policy

# whether or not we expand properties in the policy file

# if this is set to false, properties (${...}) will not be expanded in policy

# files. policy.expandProperties=true

# whether or not we allow an extra policy to be passed on the command line

# with -Djava.security.policy=somefile. Comment out this line to disable

# this feature. policy.allowSystemProperty=true

# whether or not we look into the IdentityScope for trusted Identities

# when encountering a 1.1 signed JAR file. If the identity is found

# and is trusted, we grant it AllPermission. policy.ignoreIdentityScope=false

#

# Default keystore type.

#

keystore.type=jks

#

# Class to instantiate as the system scope:

#

system.scope=sun.security.provider.IdentityDatabase

#

# List of comma-separated packages that start with or equal this string

# will cause a security exception to be thrown when

# passed to checkPackageAccess unless the

# corresponding RuntimePermission ("accessClassInPackage."+package) has

# been granted. package.access=sun.

#

# List of comma-separated packages that start with or equal this string

# will cause a security exception to be thrown when

# passed to checkPackageDefinition unless the

# corresponding RuntimePermission ("defineClassInPackage."+package) has

# been granted.

#

# by default, no packages are restricted for definition, and none of

# the class loaders supplied with the JDK call checkPackageDefinition.

#

#package.definition= policy.url.1=file${java.home}/lib/security/java.policy policy.url.2=file${user.home}/.java.policy **policy.url.3=file:/C:/co/java/prog/java.guvenlik**

------------

seklindedir. Bu dosyaya biz yazitesti.txt izni olan java.guvenlik dosyasini

### policy.url.3=file:/C:/co/java/prog/java.guvenlik

komutunu kullanarak ekledik. 3 rakaminin anlami daha önce yazilan iki güvenlik kodunun olmasidir. Simdi sadece **appletviewer yazitesti.html** komutu kulanarak programimizi çalistirip yazma islemini basarabiliriz.

### KONSOL PROGRAMLARINDA GÜVENLIK

Applet disi programlarda (konsol programlarinda) java direk olarak güvenlik kontrolu istemez. Yukardaki programin bir Jframe esdegerini olusturalim

import java.awt.\*; import java.io.\*; import java.lang.\*; import javax.swing.\*;

public class yazitestiSWF extends JFrame { String myFile = "yazitestiSWF.txt";

File f = new File(myFile);

PrintWriter dos;

public yazitestiSWF()

{

super("yazi testi swing frame konsol uygulamasi");

}

public void paint(Graphics g) { Graphics2D g2 = (Graphics2D)g;

try {

dos=new PrintWriter(new BufferedWriter(new FileWriter(myFile)));

dos.println("bu yazi deneme amaciyla dosya yazitesti.txt içine Appletden yazildi\n"); dos.close();

g2.drawString("Yazim islemi basari ile gerçeklesti : " + myFile + " -- dosyasini açip bakabilirsiniz ", 10, 50);

}

catch (SecurityException e) {

g2.drawString("yazitesti: güvenlik kontrolu asilamadi (güvenik hatasi) : " + e, 10, 50);

}

catch (IOException ioe) {

g2.drawString("yazitesti: giris çikis hatasi", 10, 50);

}

}

public static void main(String[] args)

{

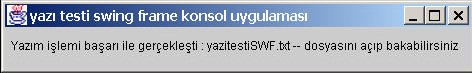
yazitestiSWF pencere= new yazitestiSWF(); pencere.addWindowListener(new BasicWindowMonitor()); pencere.setSize(350,200);

pencere.setVisible(true);

}

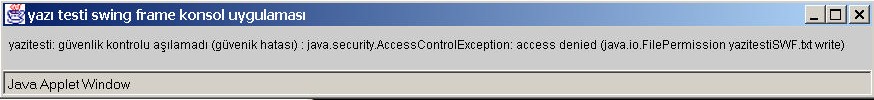
}

bu programi javac yazitestiSWF.java deyimiyle derleyip java yazitestiSWF ile çalistirirsak



programin direk olarak hiçbir güvenlik islemine gereksinim duymadan çalistigini görürüz. Eger bu güvenlik kontrolunu özel olarak arzularsak, programimizi

**java –Djava.security.manager yazitestiSWF** komutuyla çagirabiliriz. Bu durumda daha yazitestiSWF.txt dosyasini güvenlik sisteminde yanimlamadigimiz için tipki appletde oldugu gibi



sonucuyla karsilasiriz. Güvenlik sistemine bu yeni dosyayi da tanimlayarak konsol programini da güvenli modda çalistirabiliriz.

Örnegin java.guvenlik dosyasinin yazitestiSWF.txt dosyasi içinde yazma izni almis hali:

/\* AUTOMATICALLY GENERATED ON Sun Mar 11 10:27:31 PST 2001\*/

/\* DO NOT EDIT \*/

grant codeBase "file:/c:/co/java/prog" {

permission java.io.FilePermission "yazitesti.txt", "write"; permission java.io.FilePermission "yazitestiSWF.txt", "write";

};

direk olarak java güvenligini saglayan komut

### java –Djava.security.manager-Djava.security.policy=java.guvenlik yazitestiSWF

seklinde olacaktir.

Simdi size sistem özelliklerini veren bir program verelim. import java.lang.\*;

import java.security.\*;

import javax.swing.JOptionPane;

class ozellikler {

public static void main(String[] args) { String s,s1;

try {

s = System.getProperty("os.name", "not specified"); s1="isletim sistemi : " + s+"\n";

s = System.getProperty("java.version", "not specified"); s1+=" su anda çalisan java programi versiyonu : " + s+"\n"; s = System.getProperty("user.home", "not specified"); s1+=" (user.home) directory'si : " + s+"\n"; s=System.getProperty("java.home", "not specified"); s1+="(java.home) directory'si: " + s+"\n";

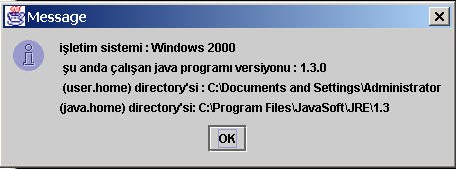
} catch (Exception e) { s1="Hata yakalandi " + e.toString();}

JOptionPane.showMessageDialog(null,s1); System.exit(0);

}

}

bu programi çalistirdigimizda :



çiktisini verecektir. Simdi programi güvenli modda çalistiralim, çikti :



olacaktir. Burada temel olarak sistem özelliklerini okuma izni vermemiz gerekiyordu. Bu yüzden java.util.PropertyPermission (java özellikleri izni) read (oku) komutuyla verildi

/\* AUTOMATICALLY GENERATED ON Sun Mar 11 11:44:51 PST 2001\*/

/\* DO NOT EDIT \*/

grant codeBase "file:/c:/co/java/prog" {

permission java.io.FilePermission "yazitesti.txt", "write"; permission java.io.FilePermission "yazitestiSWF.txt", "write"; permission java.util.PropertyPermission "ozellikler", "read";

};

### java –Djava.security.manager-Djava.security.policy=java.guvenlik ozellikler

komutu sonucu bize hatasiz olarak iletecektir.

### GÜVENLIK SERTIFIKALARI

Bilgisayarlardaki genel güvenlik sistemi genelde iki anahtarli bir sistemdir. Bu anahtarlardan birincisi, özel anahtar, sadece sizin bildiginiz bir anahtardir. Ikinci anahtar güvenli olarak iletisim kurmak istediginiz her yere gönderilebilir. Sertifika almanin ilk asamasini javada verilen keytool adli programi kullanarak yapabiliriz. Bunun için

### keytool –genkey

Komutunu kullanmamiz ve sorulan sorulara (ingilizce) cevap vermemiz kafidir. Keytool –genkey su sorulari soracaktir : kalin ile belirttiklerim cevaplar, parantez içerisindekiler türkçe karsiliklaridir.

Enter keystrore password : **ali** (pasword’u giriniz)

What is your first and last name? (isminiz ve soyisminiz nedir?) [unknown]: **Turhan Coban** (burada isminizi ve soyisminizi giriniz) What is name of your organisational unit? (is yerinizin is türü nedir?) [unknown]: **research (arastirma)**

What is name of your organisation? (is yerinizin adi nedir?) [unknown]: **TUBITAK MRC**

What is the name of your City or Locality? (sehrinizin veya bulundugunuz yerin adi nedir?) [unknown]: **Gebze**

What is the name of your State or Province? (eyaletinizin veya bölgenizin adi nedir?) [unknown]: **Kocaeli**

What is the two-letter country code for this unit? (iki harfli ülke kodunuz nedir?) [unknown]: **TR**

Is <CN=Turhan Coban, OU=Research, O=TUBITAK MAM, L=Gebze, ST=Kocaeli,C=TR> correct? (bu verilen bilgiler dogru mu?)

[no]: **yes** (evet)

bu bilgiler verildikten sonra program Enter key password for <mykey>

(RETURN if same as keystrore password):

(mykey anahtari için sifreyi giriniz daha önce girdiginiz sifreyle ayniysa sadece gir tusuna basiniz) sorusunu soracak ve sifre isteyecektir. Burada sizin sifreniz mykey ismini almistir.

Eger ikinci bir kisi size sifre gönderirse bunu keytool –import –file anahtarismi

komutuyla depolayabilirsiniz. Anahtari olusturduktan sonra anahtariniza bir anahtar sertifikasyon kurumundan sertifika alabilirsiniz sertifika alabilirsiniz, bunun için önce sertifika onay belgesi olusturabilirsiniz. Bunun için **keytool –certreq** komutu kullanabilirsiniz. Bu komut

-----BEGIN NEW CERTIFICATE REQUEST----- MIICcjCCAi8CAQAwbDELMAkGA1UEBhMCVFIxEDAOBgNVBAgTB0tvY2FlbGkxDjAMBgNVBAcTBUdl YnplMRQwEgYDVQQKEwtNUkNfVFVCSVRBSzEOMAwGA1UECxMFRVNFUkkxFTATBgNVBAMTDFR1cmhh

biBDb2JhbjCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3UjzvRADD Hj+AtlEmaUVdQCJR+1k9jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7qdf+ t8Yb+DtX58aophUPBPuD9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAJdgUI8VIwvM spK5gqLrhAvwWBz1AoGBAPfhoIXWmz3ey7yrXDa4V7l5lK+7+jrqgvlXTAs9B4JnUVlXjrrUWU/m cQcQgYC0SRZxI+hMKBYTt88JMozIpuE8FnqLVHyNKOCjrh4rs6Z1kW6jfwv6ITVi8ftiegEkO8yk 8b6oUZCJqIPf4VrlnwaSi2ZegHtVJWQBTDv+z0kqA4GFAAKBgQCGLh1t/Vo3fMkMFLXNu9dhueKG dac3vGCUmd1FzqBALncr+YEQw2DZqtxiwqIeivbXOoean2516S0cDBi333ArOGtK0pm/pHQ3Yyeh W/ePbV6ljh/KbmuHaa3ZSfAHKseuHBcMkS4SNC5DeTVifzT1QLRwGeAkP5xdL72EX/dt8KAAMAsG ByqGSM44BAMFAAMwADAtAhUAlm7LpK8FaOco1hgARAkCX0ik9CECFDqrUbnw9Fyp5CRUYs0+RCFy L+U9

-----END NEW CERTIFICATE REQUEST-----

gibi bir çikti olusturacaktir. Bu çikti anahtar sertifika otoritelerine direk olarak email ile gönderilebilir. Sertifikanin gayesi sizi tanimiyan kisilerce bile programlarinizin güvenliginin onaylanmasidir. Genellikle sertifika kuruluslari bunu bir ücret karsiligi yapar. Örnegin [www.baltimore.ie](http://www.baltimore.ie/) asdresinde böyle bir kurulusla ilgili detayli bilgi bulabilirsiniz. Sertifika kurulusundan onaylanmis olarak gelen sertifikanizi

### keytool –import –file \directory\yenisertifika

komutuyla yeni bir isimle depolayabilirsiniz. Sertifikanizi baska birine aktarirken

**keytool –export –rfc** komutu ile önderme anahtarini olusturup gönderebilirsiniz.

### JAR DOSYALARININ GÜVENLIK KODUYLA IMZALANMASI

Jar dosyalarinin hazirlanmasindan daha önce bahsetmistik. Java jar dosyalari jarsigner komutu kullanilarak sertifikalandirilabilir. Komut

**jarsigner arsiv.jar mykey** seklindedir. Buradaki mykey sizin anahtarinizin adi, arsiv.jar ise kullandigimiz jar dosyasidir. Dosyayi kullanmadan önce –verify ve –verbose terimleriyle açmamiz gerekir.

jarsigner –verify –verbose arsiv.jar

### BROWSERLARDA GÜVENLIK

Browserlerin güvenlik sistemi burada anlattiklarimizla benzer olmakla beraber aynisi degildir. Microsoft ve netscape birbirinden ayri güvenlik sistemlerine sahiptirler. Bunlarla ilgili bilgi

Microsoft için : [**http://microsoft.com/java/security/default.htm**](http://microsoft.com/java/security/default.htm)

Netscape için : [**http://home.netscape.com/info/security-doc.html**](http://home.netscape.com/info/security-doc.html)

adreslerinden edinilebilir.

### EK A TEXT SINIFI

**Bu kitapta okuma islemlerinde Text sinifi oldukça yogun olarak kullanilmistir. Bu yüzden kodu Ekte bir daha vermekte yarar görüyoruz.**

**Text.java**

//============================================

// Numerical Analysis Package in Java

// Class Text to read data from screen or file

// Dr. Turhan Coban

//============================================

import java.io.\*; import java.util.\*;

class Text

{

BufferedReader in;

/\*

---------------------------------------------------

Static routines provided are:

---------------------------------------------------

public static void prompt(String s);

public static int readInt(DataInputStream in); public static int readInt(BufferedReader in);

public static double readDouble(DataInputStream in); public static double readDouble(BufferedReader in); public static String readString(DataInputStream in); public static String readString(BufferedReader in); public static char readChar(DataInputStream in); public static char readChar(BufferedReader in);

----------------------------------------------------

Dynamic routines provided are :

----------------------------------------------------

public void Text()

public void Text(String s1) public void Text(File f1) public int readInt();

public double readDouble(); public String readString(); public char readChar(); Sample use :

--------- reading a double--------------

DataInputStream cin=new DataInputStream(System.in); double number;

number=Text.readDouble(cin); veya

BufferedReader cin=new BufferedReader(new InputStreamReader(System.in)); double number;

number=Text.readDouble(cin); veya

double number;

Text cin=new Text(); number=cin.readDouble();

------ reading data from a file "datafile.dat"------ DataInputStream fin=new DataInputStream (new FileInputStream("datafile.dat"));

double number; number=Text.readDouble(fin); veya

double number;

BufferedReader fin=new BufferedReader(new FileReader("datafile.dat"));

double number; number=Text.readDouble(fin); veya

double number;

Text cin=new Text("dataFile.dat"); number=cin.readDouble();

------ printing data into a file "printfile.dat"----

PrintStream fout=new PrintStream

(new FileOutputStream("printfile.dat")); String a="turhan";

fout.println(a); veya

BufferedWriter fout=new BufferedWriter(new FileWriter("datafile.dat")); String a="Turhan");

fout.println(a);

----------------------------------------------------

\*/

private static StringTokenizer T; private static String S;

public Text()

{

in=new BufferedReader(new InputStreamReader(System.in));

}

public Text(String s1) throws IOException

{

in=new BufferedReader(new FileReader(s1));

}

public static void prompt (String s) { System.out.print(s + " "); System.out.flush();

}

public static int readInt (DataInputStream in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

String item = T.nextToken();

return Integer.valueOf(item.trim()).intValue();

}

catch (NoSuchElementException e1) { refresh (in);} catch(NumberFormatException e2)

{ //System.err.println("Error in number, try again.");

}

}

}

public int readInt() throws IOException

{

return Text.readInt(in);

}

public String readStringLine() throws IOException

{

return Text.readStringLine(in);

}

public double readDouble() throws IOException

{

return Text.readDouble(in);

}

public String readString() throws IOException

{

return Text.readString(in);

}

public char readChar() throws IOException

{

return Text.readChar(in);

}

public static int readInt (BufferedReader in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

String item = T.nextToken();

return Integer.valueOf(item.trim()).intValue();

}

catch (NoSuchElementException e1) { refresh (in);

} catch(NumberFormatException e2)

{ //System.err.println("Error in number, try again.");

}

}

}

public static char readChar (DataInputStream in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

return T.nextToken().charAt(0);

}

catch(NoSuchElementException e1) {refresh (in);}

}

}

public static char readChar (BufferedReader in) throws IOException

{

if (T==null) refresh(in); while (true)

{

try {

return T.nextToken().charAt(0);

}

catch(NoSuchElementException e1) {refresh (in);}

}

}

public static double readDouble(DataInputStream in) throws IOException

{

if(T==null) refresh(in); while (true) {

try {

String item = T.nextToken();

return Double.valueOf (item.trim()).doubleValue();

} catch(NoSuchElementException e1) { refresh (in);

} catch(NumberFormatException e2)

{

//System.err.println("Error in number, try again.");

}

}

}

public static double readDouble(BufferedReader in) throws IOException

{

if(T==null) refresh(in); while (true) {

try {

String item = T.nextToken();

return Double.valueOf (item.trim()).doubleValue();

} catch(NoSuchElementException e1) { refresh (in);

} catch(NumberFormatException e2)

{

//System.err.println("Error in number, try again.");

}

}

}

// this method is deprecated, but still kept for historical reasons

// prefer BufferedReader version

public static String readString(DataInputStream in) throws IOException

{

if(T==null) refresh (in); while (true) {

try {return T.nextToken();}

catch (NoSuchElementException e1) { refresh (in);

}

}

}

public static String readString(BufferedReader in) throws IOException

{

if(T==null) refresh (in); while (true) {

try {return T.nextToken();}

catch (NoSuchElementException e1) { refresh (in);

}

}

}

public static String readStringLine(DataInputStream in) throws IOException

{

//reads a line of strings from DataInputStream in int ch;

String r = ""; boolean done = false; while (!done)

{ try

{ ch = in.read();

if (ch < 0 || (char)ch == '\n' || (char)ch == '\0') done = true;

else

r = r + (char) ch;

}

catch(java.io.IOException e)

{ done = true;

}

}

return r.substring(0,(r.length()-1));

}

public static String readStringLine(BufferedReader in) throws IOException

{

//reads a line of strings from BufferedReader in int ch;

String r = ""; boolean done = false; while (!done)

{ try

{ ch = in.read();

if (ch < 0 || (char)ch == '\n' || (char)ch == '\0') done = true;

else

r = r + (char) ch;

}

catch(java.io.IOException e)

{ done = true;

}

}

return r.substring(0,(r.length()-1));

}

private static void refresh (DataInputStream in) throws IOException

{

String s = in.readLine();

if (s==null) throw new EOFException(); T = new StringTokenizer(s);

}

private static void refresh (BufferedReader in) throws IOException

{

String s = in.readLine();

if (s==null) throw new EOFException(); T = new StringTokenizer(s);

}

}

### EK B JAVA DERSI SINAV SORULARI

**Java derslerindeki bazi sinav sorulari bu ekde verilmistir.**

**JAVA PROGRAMLAMA DILI** GEBZE YÜKSEK TEKNOLOJI ENSTITÜSÜ GÜZ DÖNEMI 2000 BITIRME SINAVI SORULARI

|  |  |
| --- | --- |
| Isim – soyisim |  |
| Ögrenci numarasi |  |
| Master/doktora |  |
| Email |  |
| Telefon |  |
| Not |  |

AÇIKLAMALAR

 Toplam süre 3 saattir.

 Sizin için verilen bos alanlara sorulari cevaplandiriniz.

 Verilen yerler yetmezse ilave kagit kullanabilirsiniz

 Her soru on puandir (toplam 100 puan)

 Iyi sanslar

### SORU 1

Asagida verilen programin girdisi :

ilk gerçek sayiyi giriniz : 2 ikinci ger‡ek sayiyi giriniz : 2

seklindedir. Program çiktisini yaziniz.

import java.io.\*; import Text;

import javax.swing.JOptionPane;

class SORU1\_2000{

public static void main (String args[]) throws IOException

{

double sayi1,sayi2; double toplam;

Text cin=new Text();

System.out.print("ilk ger‡ek sayiyi giriniz : "); sayi1=cin.readDouble();

System.out.print("ikinci ger‡ek sayiyi giriniz : "); sayi2=cin.readDouble();

toplam=sayi1+sayi2;

JOptionPane.showMessageDialog(null,sayi1+" + "+sayi2+" = "+toplam); System.exit(0);

}

}

### SORU 2

Asagida verilen kutuya girdi olarak 2 girildiginde olusacak program çiktisini yaziniz

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;



public class SORU2\_2000 extends JApplet implements ActionListener

{ int sayi1; int toplam; double ort;

JLabel s1,s2; //yazi alanlari sinifi JTextField t1; //girdi alanlari (kutulari) JTextArea ciktiAlani; //cikti alani

//pencereyi baslatma metodu public void init()

{

Container c=getContentPane(); c.setLayout(new FlowLayout()); Color renk=c.getBackground();

s1=new JLabel("küp ortalamasi 1 den "); s2=new JLabel("e kadar ");

t1=new JTextField(5); //pencereye ekle ciktiAlani=new JTextArea(); ciktiAlani.setBackground(renk); c.add(s1);

c.add(t1);

c.add(s2); c.add(ciktiAlani); t1.addActionListener(this);

//girdi alanindan sinyal bekle ve sinyal gelince su islemleri yap

public void actionPerformed(ActionEvent e)

{

toplam=0; //sayi1 i kutulardan

oku

sayi1=Integer.parseInt(t1.getText()); int i=1;

do { toplam+=i\*i\*i;

}

while(++i<=sayi1);

ort=(double)toplam/(double)sayi1; ciktiAlani.setText(""+"1den "+say

kadar küp ortalamasi :"+ort); repaint();

}

//ActionPerformed metotunun kapanisi

} //H2OD3\_2000 sinifinin kapanisi

}

### SS

**SORU 3**

Asagida verilen programdaki sayinin kuvvetini hesaplayan

### public static double pow(double taban,double kuvvet)

metodunu yaziniz. Bu metod kendi kendini çagiran bir program kodu veya bir döngü yapisi kullanan bir sekilde olusturulabilir.

Import javax.swing.JOptionPane; public class SORU3\_2000

{

public static double pow(double taban,double kuvvet)

{

\*\*\*buraya gerekli program kodu girilecektir.\*\*\*

}

public static void main (String args[])

{

double taban,kuvvet; taban=Double.parseDouble(JOptionPane.showInputDialog ("taban sayiyi giriniz : ")); kuvvet=Double.parseDouble(JOptionPane.showInputDialog ("kuvveti giriniz : "));

JOptionPane.showMessageDialog(null,taban+" nin "+kuvvet+" inci kuvveti: "

+pow(taban,kuvvet),"H3OD3\_2000.java programi",JOptionPane.PLAIN\_MESSAGE); System.exit(0);

}

}

Cevap :

public static double pow(double taban,double kuvvet)

{

}

### SORU 4

Asagida verilen kompleks2 programinin içerisinde yer almasi gereken ve bir kompleks sayinin karesini hesaplayan kare metodunu yaziniz. kompleks2 sinifini kullanarak (1+2i) kompleks sayisinin karesini alan ve sonuçlarini konsol ortaminda yazdiran bir küçük bir çikti programi yaziniz

public class kompleks2

{

public double ger; public double imaj;

public kompleks2(double nger,double nimaj)

{

ger=nger; imaj=nimaj;

}

public static kompleks2 pow(kompleks sol, double sag)

{ // kompleks bir sayinin gerçek kuvvetini hesaplar. double Rad,th;

Rad=Math.pow(sol.R(),sag); th=sag\*sol.theta(); kompleks2 sonuc;

sonuc =new kompleks2((Rad\*Math.cos(th) ), (Rad\*Math.sin(th) ) );

return sonuc;

}

}

//kompleks sinifinin taniminin sonu

### Kare metodu :

public static kompleks2 kare(kompleks sol)

{

}

### SORU4\_2000.java çikti programi

**SORU 5**

Boyutlu degisken kullaniminda en büyük sorunlardan biri boyutlarin sabit olmasi, ve degistirilememesidir. Double tipi degiskenlerin boyutlarinin degistirilmesi amaciyla doubleBoyut sinifi gelistirilmistir. Fakat program dosyasina kaza sonucu dökülen kahve sonucu programin bir kismi okunamamaktadir. Program kodu bilgisayara tekrar girilecektir. BoyutEkle alt metodundaki okunamiyan yerin tekrar yazilmasinda yardimci olabilirmisiniz.

public class doubleBoyut

{

//sinif degiskenleri public double a[]; public int length;

public doubleBoyut(double x[] )

{

length=x.length; a=new double[length];

for(int i=0;i<length;i++) a[i]=x[i];

}

public doubleBoyut(int n)

public void setValue(double x,int i)

{

a[i]=x;

}

public void setValue(double[] x )

{

length=x.length; a=new double[length];

for(int i=0;i<length;i++)

{

a[i]=x[i];

}

}

{

a=new double[n]; length=a.length;

}

public void boyutEkle(int n)

{

if(n>0)

{

int z=a.length+n;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*buradaki kodu düzeltiniz\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* a=b;

length=a.length;

}

}

public void boyutEkle()

{

boyutEkle(1);

}

public void boyutAzalt()

{

boyutAzalt(1);

}

public double getValue(int i)

{

return a[i];

}

public double[] getValue()

{

return a;

}

public String toString(int i)

{

return ""+a[i];

}

public String toString()

{

String s="";

for(int i=0;i<length;i++)

{

s+=a[i]+" ";

}

s+="\n"; return s;

}

}

public void boyutAzalt(int n)

{

int z=a.length-n; if(z>1)

{

double[] b=new double[a.length-n]; for(int i=0;i<z;i++)

b[i]=a[i]; a=b;

}

else

{

double[] b=new double[1]; b[0]=a[0];

a=b;

}

length=a.length;

}

public void boyutEkle(int n)

{

if(n>0)

{

int z=a.length+n;

a=b; length=a.length;}}

### SORU 6

OvalPanelSWP panel programi verilmistir. Biz bu programi kullanarak bir elips çizdirmek istiyoruz. Bunun için yazilan JFrame programi SORU6\_2000 nin kurucu metodu SORU6\_2000metodunu tamamlayiniz.

import javax.swing.\*; import java.awt.\*; import yildizSW; import java.awt.geom.\*; import renk;

public class ovalPanelSWP extends JPanel

{

int xi,yi,yaricap;

public ovalPanelSWP(int xi1, int yi1,int yaricap1)

{

xi=xi1; yi=yi1;

yaricap=(yaricap1 >= 0 ? yaricap1:10);

}

public void setOvalPanelSWP(int xi1, int yi1,int yaricap1)

{

xi=xi1; yi=yi1;

yaricap=(yaricap1 >= 0 ? yaricap1:10); repaint();

}

public void paintComponent(Graphics g)

{

super.paintComponent(g); g.setColor(renk.mavi); Graphics2D g2=(Graphics2D)g;

g2.setRenderingHint(RenderingHints.KEY\_ANTIA LIASING,

RenderingHints.VALUE\_ANTIALIAS\_ON); Ellipse2D elips1=new Ellipse2D.Double(xi,yi,yaricap,yaricap); g2.draw(elips1);

}

}

================================================================

import java.awt.event.\*; // java pencereyi

dinleme sinifini cagir

import BasicWindowMonitor; import ovalPanelSWP;

import javax.swing.event.\*;

public class **SORU6\_2000** extends JFrame implements ChangeListener

{

// Renk secme ornegi private ovalPanelSWP p; Color r=Color.lightGray; Container c;

// pencereyi baslatma metodu

public ovalTestSWF\_2000()

{

super("JSlider Testi"); c=getContentPane(); c.setLayout();new Flowlayout());

}

public **SORU6\_2000**()

{

super("JSlider Testi"); c=getContentPane(); c.setLayout(new Flowlayout());

\*\*\*\*\*\* tamamlayiniz \*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

//============================

public static void main(String[] args)

{

SORU6\_2000 pencere= new SORU6\_2000(); pencere.addWindowListener(new

BasicWindowMonitor()); pencere.setSize(350,200); pencere.setVisible(true);

}}

**SORU 7**

Program çikti tablosunu çiziniz ve olusan degerleri yaziniz.

import java.sql.\*; import java.io.\*;

import Text;

public class CreateKahve{

public static void main(String args[]) throws IOException

{

String driverName;

String url = "jdbc:odbc:prog"; Connection con;

String createString;

createString = "create table KAHVE " + "(KAHVE\_ISMI varchar(32), " + "NUMARA int, " +

"FIYAT float, " + "SATIS int, " + "TOPLAM int)";

Statement stmt; try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

} catch(java.lang.ClassNotFoundException e)

{

System.err.print("ClassNotFoundException: "); System.err.println(e.getMessage());

}

try {

con = DriverManager.getConnection(url,"Login", "Password"); stmt = con.createStatement(); stmt.executeUpdate(createString);

stmt.executeUpdate("insert into KAHVE " + "values('Brezilya', 00101, 7.99, 0, 0)"); stmt.close();

con.close();

} catch(SQLException ex)

{System.err.println("SQLException: " + ex.getMessage());}

}

}

Program çiktisi :

### SORU 8

**gir1.txt dosyasi incelendiginde 1 2**

**2 3**

**3 4**

**4 5**

**5 6**

**verisinin kaydedildigi gözlenmistir. Asagidaki program çalistirildiginda cik1.txt dosyasinda bulunmasi gereken veriyi yaziniz**

import java.io.\*; import Text;

public class SORU8\_2000

{

public static void main(String[] args) throws IOException

{

double x[],y[]; DataInputStream input; x= new double[200]; y=new double[200]; String s1="gir1.txt"; File fgir = new File(s1);

BufferedReader b= new BufferedReader(new FileReader(fgir)); boolean EOF=false;

int j=0; while (!EOF)

{

try

{

x[j]= Text.readDouble(b); y[j]= Text.readDouble(b);

++j;

}

catch (EOFException e)

{

b.close();

EOF=true;

}

} //while

String s2="cik1.txt";

File fcik = new File(s2);

PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter(fcik))); for(int i=0;i<j;i++)

{cfout.println(""+x[i]+" "+(y[i]\*y[i]));} cfout.close();

} //main

} //class

### SORU 9

* + - 1. programin islevini izah ediniz
      2. verilen girdi için çiktiyi yaziniz

import java.util.\*; import java.awt.\*;

import javax.swing.\*; import java.awt.event.\*;

### class SetA

c.add(jp);

}

### {

**public Vector bilesim(Vector list1, Vector list2)**

**{**

**Vector bilesimVectoru = new Vector(); String s1,s2;**

**Enumeration n1=list1.elements(); Enumeration n2=list2.elements(); while(n1.hasMoreElements())**

**{**

**s1=(String)n1.nextElement(); bilesimVectoru.addElement(s1);**

**}**

**while(n2.hasMoreElements())**

**{**

**s2=(String)n2.nextElement(); if(!bilesimVectoru.contains(s2))**

**bilesimVectoru.addElement(s2);**

**}**

**return bilesimVectoru;**

**}**

**}**

public class SORU9\_2000 extends JFrame implements ActionListener

{

SetA set = new SetA(); // SetA JLabel L1, P1, P2;

JTextField T1,T2; JTextArea cikti;

Vector list1 = new Vector(); Vector list2 = new Vector(); Vector ansUn;

Container c;

public SetTestSWF\_2000()

public void girdiA(String string, Vector list)

{

StringTokenizer tokens = new StringTokenizer(string);

while(tokens.hasMoreTokens()) { String test = tokens.nextToken(); list.addElement(test);

}

}

public void ciktiA(Vector v)

{

Enumeration enum = v.elements(); while(enum.hasMoreElements())

{

String ans = (String)enum.nextElement(); cikti.append(ans + " ");

}

cikti.append("\n");

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==T1)

{

String stringToTokenize = T1.getText(); girdiA(stringToTokenize,list1);

}

else if(e.getSource()==T2)

{

String stringToTokenize = T2.getText(); girdiA(stringToTokenize,list2);

ansUn = set.bilesim(list1,list2); cikti.setText("");

cikti.append("Iki listenin bilesim kümesi :

\n");

ciktiA(ansUn);

{ }

super("küme testi bilesim ve kesisim kümeleri");

c=getContentPane(); c.setLayout(new FlowLayout());

P1 = new JLabel("Birinci listeyi giriniz : "); T1 = new JTextField(30);

P2 = new JLabel("Ikinci listeyi giriniz : "); T2 = new JTextField(30);

T2.setBackground(c.getBackground()); cikti = new JTextArea(); cikti.setEditable(false); cikti.setText("");

cikti.setBackground(c.getBackground()); c.add(P1);

c.add(T1); T1.addActionListener(this); c.add(P2);

c.add(T2); T2.addActionListener(this);

JScrollPane jp=new JScrollPane(cikti); jp.setPreferredSize(new Dimension(320,200));

} //actionPerformed metodunun sonu public static void main(String[] args)

{

SORU9\_2000 pencere = new SORU9\_2000 (); pencere.addWindowListener(new

BasicWindowMonitor()); pencere.setSize(400,350); pencere.setVisible(true);

}

}



### SORU 10

Birden yüze kadar sayilarin ortalamasini bulan bir java bilgisayar programi yaziniz.

### JAVA PROGRAMLAMA DILI GEBZE YÜKSEK TEKNOLOJI ENSTITÜSÜ 20 OCAK 2000 BITIRME SINAVI SORULARI

Dr. Turhan Çoban

|  |  |
| --- | --- |
| isim soyisim |  |
| ogrenci numarasi |  |
| email /telefon |  |
| not |  |

AÇIKLAMALAR :

 Toplam süre 3 saattir. (9.00 -12.00)

 Sizin için verilen bos alanlara sorulari cevaplandiriniz

 verilen yerler yetmezse ilave kagit kullanabilirsiniz.

 Iyi sanslar

### SORU 1

import java.io.\*;

public class Aritmetik1

{ public static void main(String[] args)

{

int sayi1=1; int sayi2=2;

sayi1 += sayi2; System.out.println("sayi1+= : "+ sayi1); sayi1 -= sayi2; System.out.println("sayi1-= : "+ sayi1); sayi1 \*= sayi2; System.out.println("sayi1\*= : "+ sayi1); sayi1 /= sayi2; System.out.println("sayi1/= : "+ sayi1);

}

}

Verilen programin sonuçlarini yaziniz :

### SORU 2

import java.applet.Applet; import java.awt.\*;

public class soru2 extends Applet

{

public void paint(Graphics g)

{

double not=60;

if( not >= 90)

{ g.drawString("A",25,35); } else if(not >=75)

{ g.drawString("B",25,35); } else if(not >=60)

{ g.drawString("C",25,35); } else if(not >=50)

{ g.drawString("D",25,35); } else if(not >=40)

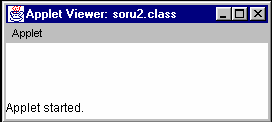
{ g.drawString("E",25,35); } else

{ g.drawString("F",25,35); }

}

}

Verilen programin sonuçlarini yaziniz :



### SORU 3

import java.applet.Applet; import java.awt.\*;

public class soru3 extends Applet

{

public void paint(Graphics g)

{

for(int x=10;x<100;x+=20)

{

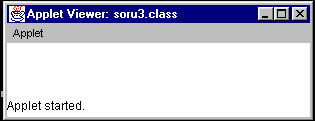
g.drawString(" "+x,x,25);

}

}

}

Verilen programin sonuçlarini yaziniz :



### SORU 4

import java.awt.Graphics; import java.applet.Applet;

public class soru4 extends Applet

{

public void paint(Graphics g)

{

int x,y; y=1;

while(y++<5)

{

x=1;

while(x++<6)

{

g.drawString ("\*",15\*x,15\*y);

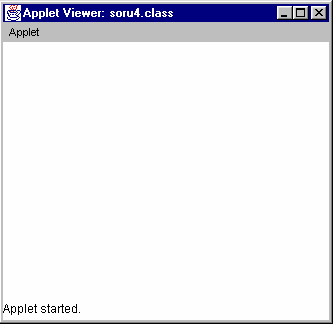
}

}

}

}

Verilen programin sonuçlarini çiziniz :



### SORU 5

asagidaki program kodunda \*\*\*\*\*\*\* ile verilen yerde kare metodunu kullanarak sonucu yazdiracak System.out.println(...) komutunu yeniden yaziniz

import java.io.\*;

public class soru5

{

// sayinin karesi static metodu

public static double kare(double x)

{

return x\*x;

}

public static void main(String[] args) throws IOException

{

double sayi=2;

System.out.println("sayinin karesi : "+ \*\*\*\*\*\*\*\* );

}

}

### SORU6

soru6 metodunda, apletdeki sonucu alabilmek için gerekli olan kare metodunu yaziniz.

import java.applet.Applet; // java applet sinifini cagir import java.awt.\*; // java pencere kullanma sinifini cagir

class soru6

{

// sayinin karesi dinamik metodu

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

public class soru6a extends Applet

{

double sayi=5; // Gercek degisken sayi

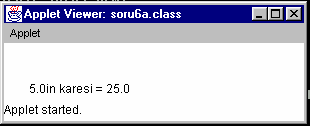
soru6 nesne=new soru6(); public void paint(Graphics g)

{

g.drawString(sayi+"in karesi = "+nesne.kare(sayi),25,50);

}

}



### SORU 7

Verilen programin çiktisini yaziniz. import java.io.\*;

class soru7

{

public static boolean ve(boolean b1,boolean b2)

{

return b1&&b2;

}

public static boolean veya(boolean b1,boolean b2)

{

return b1||b2;

}

}

public class soru7a

{

public static void main(String arg[])

{

boolean b1=true; boolean b2=false;

System.out.println(""+soru7.ve(b1,b2)); System.out.println(""+soru7.veya(b1,b2));

}

}

### SORU 8

Asagida verilen noktaX sinifinda bos birakilan public noktaX(double a, double b)

kurucu metodunu yaziniz.

public class noktaX

{

public double x, y;

public noktaX()

{ x=0; y=0;

}

public noktaX(double a, double b)

{

\*\*\*\*\*\*\*\*\*\*\*\*\*

}

public noktaX(noktaX a)

{

x=a.x;

y=a.y;

}

public String toString()

{

return "["+x+","+y+"]";

}

}

public noktaX(double a, double b)

{

}

### SORU 9

soru 8 de verilen noktaX sinifini kullanan dogruX sinifi verilmistir. public dogruX(nokta nokta1,nokta nokta2)

kurucu metodunu yaziniz. import noktaX;

public class dogruX

{

public nokta n1,n2;

//kurucu metod metot girdisi x1,y1,x2,y2

public dogruX(double x1, double y1,double x2,double y2)

{

n1=new nokta(x1,y1); n2=new nokta(x2,y2);

}

//kurucu metod, method girdisi iki nokta public dogruX(nokta nokta1,nokta nokta2)

{

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

}

public dogruX(nokta nokta1,nokta nokta2)

{

}

### SORU 10

Asagidaki programin çiktisini yaziniz import java.io.\*;

class boyut1

{

public static void main(String args[] )

{

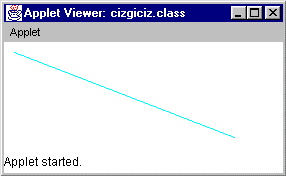
int ayin\_gunleri[]={31,28,31,30,31,30,31,31,30,31,30,31};

System.out.println("Nisan ayi "+ayin\_gunleri[3]+" gun ceker");

}

}

### SORU 10

**drawLine** metodu kullanarak asagidaki çiktiyi alacak bir program yaziniz. çizgi koordinatlarinin ne oldugu önemli degildir.

### SORU 11

**drawRect** metodu kullanarak asagidaki çiktiyi alacak bir program yaziniz. dikdortgen koordinatlarinin ne oldugu önemli degildir.



### SORU 12

Bu programin çiktisini yaziniz :

import java.applet.Applet; import java.awt.\*;

public class LabelTesti extends Applet

{

private Label L1,L2;

public void init()

{

L1=new Label(); L1.setText("etiket 1"); add(L1);

L2=new Label("etiket 2"); add(L2);

}

public void paint(Graphics g)

{

g.drawString("L1 : "+L1.getText(),25,80);

g.drawString("L2 : "+L2.getText(),25,95);

}

}

### SORU 13

Aplete "Merhaba" yazdiran bir applet programi yaziniz.

### SORU 14

Asagidaki programin sonuçlarini (aplete) yaziniz.

import java.applet.Applet; import java.awt.\*;

public class soru14 extends Applet

{

private TextArea T1; public void init()

{

setBackground(Color.lightGray);

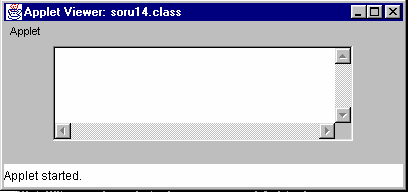
String s="buraya denemek icin\nyazi yazildi\n\t sonrada bu yazi aktarildi";

//not \n satirbasi kontrolu

// \t bir sonraki tab setine git T1=new TextArea(s,5,40); add(T1);

}

}



### SORU 15

Birden 10 a kadar sayilarin ortalamasini hesaplayan bir application (konsol) programi yaziniz.

### SORU 16

Asagidaki kodun çiktisi ne olacaktir :

public class sifiraBolunmeException extends ArithmeticException

{

public sifiraBolunmeException()

{ super("sifira bolmeye calistiniz "); }

}

import java.io.\*;

import sifiraBolunmeException;

public class soru15

{

public static void main(String arg[])

{

int sayi1=0; int sayi2=1; try

{

System.out.println(bol(sayi2,sayi1));

}

catch(sifiraBolunmeException e) {System.err.println(e.toString());}

}

public static double bol(int s1,int s2) throws sifiraBolunmeException

{

if(s2==0) throw new sifiraBolunmeException(); return (double) s1/s2;

}

}

### SORU 17

1. Bu programin ne yaptigini izah ediniz
2. program çiktisini yaziniz. import java.io.\*;

class YazOku

{

public static void main (String args[]) throws IOException

{

String s1="ilk deger"; String s2="ilk deger"; String s3="ilk deger";

PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt"))); cfout.println("Merhaba");

cfout.println("isminiz nedir");

cfout.println("sizinle tanistigima memnun oldum"); cfout.close();

BufferedReader cfin=new BufferedReader(new FileReader("a.txt")); s1=cfin.readLine();

s2=cfin.readLine(); s3=cfin.readLine(); cfin.close();

System.out.println("s1 = "+s1); System.out.println("s2 = "+s2); System.out.println("s3 = "+s3);

}

}

a)

b)

### SORU 18

Bu programin çiktisini yaziniz.

import java.io.\*; import java.util.\*;

public class StringTokenizerTest

{

public static void main(String arg[])

{

String s="Ali veli 49 elli"; System.out.println("cumle : "+s); StringTokenizer t=new StringTokenizer(s);

System.out.println("kelime sayisi : "+t.countTokens()); int i=0;

while(t.hasMoreTokens())

{

System.out.println("kelime indeksi : "+(i++)+" kelime : "+t.nextToken());

}

}

}

### SORU 19

Asagida verilen **bilesim** metodunun ne yaptigini izah ediniz

import java.util.\*; import java.awt.\*;

import java.applet.Applet; import java.awt.event.\*;

class kume

{

public Vector bilesim(Vector list1, Vector list2)

{

Vector bilesimVectoru = new Vector(); String s1,s2;

Enumeration n1=list1.elements(); Enumeration n2=list2.elements(); while(n1.hasMoreElements())

{

s1=(String)n1.nextElement(); bilesimVectoru.addElement(s1);

}

while(n2.hasMoreElements())

{

s2=(String)n2.nextElement(); if(!bilesimVectoru.contains(s2))

bilesimVectoru.addElement(s2);

}

return bilesimVectoru;

}

}

### SORU 20

Asagida verilen Tree sinifinin ve **soldal\_node\_sagdal\_sirala();** metodunun ne yaptigini izah ediniz.

package bolum11; class TreeNode

{

TreeNode sol; int data; TreeNode sag;

public TreeNode(int d)

{

data=d; sol=sag=null;

}

public synchronized void gir(int d)

{

// not ayni deger iki kere girilmeye calisilirsa

// ikinci giris dikkate alinmiyacaktir if(d<data)

{ if(sol==null) {sol=new TreeNode(d);} else {sol.gir(d);}

}

else if(d>data)

{ if(sag==null) {sag=new TreeNode(d);} else {sag.gir(d);}

}

}

}

public class Tree

{

private TreeNode kok; public Tree() {kok=null;}

public synchronized void gir(int d)

{

if(kok==null) kok=new TreeNode(d); else kok.gir(d);

}

public void soldal\_node\_sagdal\_sirala(){soldal\_node\_sagdal\_siralayici(kok);}

public void soldal\_node\_sagdal\_siralayici(TreeNode node)

{

if(node==null) return; soldal\_node\_sagdal\_siralayici(node.sol); System.out.print(node.data+" "); soldal\_node\_sagdal\_siralayici(node.sag);

}

}

### 18 REFERANS LISTESI

1. Java How to Program, Deitel & Deitel, Second Edition, Prentice Hall, [http://www.deitel.com,](http://www.deitel.com/) second edition,1998, ISBN 0-13-906249-1
2. Java How to Program, Java 2 version, Deitel & Deitel, Third Edition, Prentice Hall, http://www.deitel.com,1999, ISBN 0-13-012507-5
3. Java 1.1 The Complete Reference, Patrick Naughton, Herbert Schildt, Second Adition, Mc Graw Hill, 1998,ISBN 0-07-882436-2
4. Java Software Solutions, faundations of Programming design, John Lewis, William Loftus, Addison Wesley publishing, ISBN 0-201-57164-1
5. Java Language reference, Mark Grand, O'reilly publishing, 1997, ISBN 1-56592-204-2
6. JDBC Database Access with Java, A tutorial and Annotated Reference, Hamilton, Cattell, Fisher, Addison Wesley, 1997, ISBN 0-201-30995-5
7. Teach yourself java in 21 Days, Laura Lemay, Charles L. Perkins, Sams net, 1996, ISBN 1-57521-030- 4
8. Java 1.1 Uygulama Gelistirme klavuzu, uzmanlar için, Philip Heller, S. Robberts,P. Seymour, T. McGin, Türkçe basim editörü Dr. Cahit Akin, ISBN 975-322-011-1
9. Java, Hazirlayan: Gurup Java, Beta basin yayim dagitim AS,Istanbul 1997, ISBN 975-486-581-7
10. Java döküman kütüphanesi : http:\\java.sun.com'dan çekilebilir.
11. Mastering Java 1.2 (Renamed Java 2 by sun microsystems), John Zukowski, Sybex publishing, 1998, ISBN 0-7821-2180-2
12. Learning Jave, Covers Java 1.3, Patrick Niemeyer & Jonathan Knudsen, O’Reilly , 2000, ISBN 1- 56592-718-4
13. Java Swing, Robert Eckstein, Marc Loy & Dave Wood, O’Reilly , 1998, ISBN 1-56592-455-X
14. JDBC Database Access with Java, The Java Series, Graham Hamilton, Rick Cattell, Maydene Fisher, Addison Wesley, 1997, ISBN 0-201-30995-5
15. Java Algorithms, Scott Robert Ladd, Mc Graw-Hill, 1998, ISBN 0-07-913696-6
16. Fundamentals of Database Systems, Elmasri & Navathe, Benjamin/Cummings, 1994, ISBN 0-8053- 1753-8
17. Select…SQL the relational database language, Larry R. Newcomer, Macmillian /Maxwell, 1992, ISBN 0-02-386693-4
18. Thinking in Java, Bruce Eckel, Prentice Hall, 2000, ISBN 013-659723-8