

Java

Programlama Dili



Dr. Turhan Coban

JAVA PROGRAMLAMA DİLİ

00001.JPG



Dr. M. Turhan Çoban
TÜBİTAK Ulusal Metroloji Enstitüsü

Email : mcoban2@netscape.net

ÖNSÖZ.....	6
BÖLÜM 1 : JAVAYA GİRİŞ.....	8
1.1 NİÇİN JAVA PROGRAMLAMA DİLİ	8
1.2 JAVA PROGRAMINI YAZMA VE ÇALIŞTIRMA	9
1.3 JAVA PROGRAMLAMA DİLİ TEMEL DEĞİŞKEN TÜRLERİ.....	20
1.3.1 Boolean değişken türü	20
1.3.2 char (harf) değişken türü.....	20
1.3.3 Tam sayı değişken türleri (byte, short,int,long).....	21
1.3.4 Gerçek sayı değişken türleri (float, double,long double).....	22
1.4 JAVA NESNESİ OLARAK TANIMLANMIŞ TEMEL DEĞİŞKENLER	22
1.4.1 String nesne tipi değişkeni.....	22
1.4.2 Integer nesne tipi değişkeni	23
1.4.3 Double nesne tipi değişkeni.....	23
1.4.4 diğer nesne temelli değişken türleri	24
1.5 FİNAL TERİMİ VE SABİTLER	24
1.6 ARİTMETİK İŞLEMLER	24
1.7 JAVADA MANTIKSAL İŞLEMLER	27
1.8 JAVA DİLİNİN TEMEL KOMUTLARI.....	29
1.8.1 if - elseif - else mantıksal karşılaştırma yapısı	29
1.8.2 while tekrarlama yapısı	32
1.8.3 for tekrarlama yapısı	33
1.8.4 switch - case yapısı	35
1.9 ARİTMETİK İŞLEMLERDE DEĞİŞKEN TÜRÜ DEĞİŞTİRME (CASTİNG) OPERASYONU	39
1.10 SAYI DEĞİL VE SONSUZ SONUÇLARI	39
1.11 ALIŞTIRMALAR	39
BÖLÜM 2 : METODLAR VE OBJECT KULLANIMI.....	79
2.1 JAVA API KÜTÜPHANESİ	79
2.2 METOTLAR	82
2.3 NESNE (OBJECT) TANIMI VE METOTLarda KULLANIMI	84
2.4 METOTLARIN KENDİ KENDİNİ ÇAĞIRMASI (RECURSION).....	86
2.5 AYNI ADLI METOTLARIN BİR ARADA KULLANILMASI (OVERLOADİNG)	89
2.6 METOT (METHOD) VE SINIF(CLASS) DEĞİŞKENLERİ.....	91
2.7 ALISTIRMALAR	94
BÖLÜM 3 . SINIF (CLASS) YAPILARINA GİRİŞ.....	124
3.0 STANDART KAVRAMLARIN TÜRKÇE KARŞILIKLARI	124
3.1 SINIF YAPISININ GENEL TANIMI.....	124
3.2 KURUCU (CONSTRUCTOR) METOT.....	126
3.3 DİĞER METOTLAR	127
3.4 THIS DEYİMİNİN KULLANIMI	129
3.5 SINIF DEĞİŞKENLERİNİN DİŞ DÜNYADAN GİZLENMESİ	129
3.6 SINIFLarda KALITIM (INHERITANCE)	133
3.7 SINIFLARI BAŞKA BİR SINIFTA NESNE OLARAK ÇAĞIRARAK BİR ARAYA GETİRME (COMPOSITION).....	135
3.8 KALITIM YOLUYLA ALT SINIFA BAĞLANMIŞ ÜST SINIF REFERANSI ÜZERİNDEN ALT SINIFI ÇAĞIRMA	136
3.9 NESNEYİ DiNAMİK OLARAK SİLMEK FINALIZE() METOTU.....	137
3.10 ALIŞTIRMALAR	137
BÖLÜM 4 : BOYUTLU DEĞİŞKENLER VE NESNELER (ARRAYS)	202
4.1 TEK BOYUTLU DEĞİŞKENLER.....	202
4.2 TEK BOYUTLU NESNETİ DEĞİŞKENLER	206
4.3 ÇOK BOYUTLU DEĞİŞKENLER	206
4.4 BOYUTLU DEĞİŞKENLERİN METOTLARA AKTARIMI	210
4.5 BOYUTLU DEĞİŞKENLERDE BOYUT DEĞİŞTİRME	210
4.6 ALIŞTIRMALAR	213
BÖLÜM 5 : SINIFLarda HİYERARSİ, ABSTRACT SINIF VE INTERFACE.....	227
5.1 ABSTRACT SINIF	227
5.2 INTERFACE	231
5.3 ALIŞTIRMALAR	235
BÖLÜM 6 : GRAFİKLER, FONTLAR VE RENKLER.....	244

6.1 GİRİŞ	244
6.2 GRAPHICS VE GRAPHICS2D SINIFLARI	244
6.3 RENK KONTROLÜ	247
6.4 YAZI KONTROLU	253
6.5 CiZGi CiZiMi.....	257
6.6 DiKDÖRTGEN ÇİZİMİ.....	260
6.7 ÇİZİLEN ŞEKİLLERİN DEĞİŞTİRİLEREK ÇİZİMİ (TRANSFORM)	265
6.8 OVAL VE AÇILI OVAL ÇİZİMİ	267
6.9 ÇİZİMİN SEÇİLEN BİR RESİMLE DOLDURULMASI.....	270
6.10 POLYGON ÇİZİMİ	273
6.11 GENELLEŞTİRİLMİŞ EĞRİ ÇİZİMİ.....	276
6.12 RESİM GÖSTERİMİ	280
6.13 ALIŞTIRMALAR	281
BÖLÜM 7: GRAFİK APPLET VE ÇERÇEVE OLUŞTURMA OLUŞTURMA METOTLARI, GRAFİK KULLANICISI ARABİRİM PROGRAMLARI (GUI) ,.....	308
7.1 GRAPHIC KULLANICISI ARABİRİM PROGRAMLARI, GUI, (GRAPHICS USER INTERFACE)	308
7.2 AWT LABEL (ETİKET) SINIFI , JAVAX JLABEL VE ICON SINIFLARI.....	309
7.3 BUTTON VE JButton (DÜĞME) SINIFLARI	315
7.4 TEXTFIELD (YAZIM ALANI) SINIFI	322
7.5 YAZIM ALANI AWT TEXTAREA VE SWING JTEXTAREA SINIFLARI.....	326
7.6 AWT CHOICE , SWING JComboBox SEÇİM SINIFLARI	329
7.7 AWT, CHECKBOX VE CHECKBOXGROUP VE SWING JCHECKBOX VE JRADIOBUTTON SINIFLARI	332
7.8 MOUSE(FARE) KONTROLU	339
7.10 AWT, LIST SINIFI VE JAVA SWING JLIST SINIFI , LİSTEDEN SEÇİM	345
7.11 SWING JMENU.....	355
7.12 SWING JSLIDER SINIFI VE JPanel TEMEL ÇİZİM ELEMANI (PANELİ)	359
7.13 FLOWLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI.....	362
7.14 BORDERLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI	364
7.15 GRIDLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI	366
7.17 JSPLITPANE SINIFI KULLANILARAK FORMATLAMA.....	369
7.19 PANEL VE JPanel SINIFI	371
7.20 JOPTIONPANE SINIFI	376
7.22 JFILECHOOSEN SINIFI	379
7.23 JAVA JAR (JAVA ARCHIVES - JAVA ARŞİVİ) YAPILARININ KULLANIMI	381
7.24 ALIŞTIRMALAR	382
8. JAVADA HATA ANALİZİ VE YAKALANMASI	472
8.1 HATA ANALİZİ.....	472
8.2 ALIŞTIRMALAR	476
9. PARALEL KULLANIM (MULTITHREADING) , GERÇEK ZAMAN UYGULAMALARI, ANIMASYON.....	486
9.1 PARALEL KULLANIM(MULTITHREADING) VE GERÇEK ZAMAN PROGRAMLANMASI	486
9.2 PARALEL PROGRAM KULLANIMIDA HAFIZA SINKRONİZASYONU.....	491
9.3 ALIŞTIRMALAR	499
BÖLÜM 10 JAVA GİRDİ - ÇIKTI PROGRAMLANMASI	528
10.1 JAVANIN I/O (GİRDİ ÇIKTI) KÜTÜPHANESİNDE YER ALAN SINIFLAR VE INTERFACE'LER	528
10.2 FİLE (DOSYA) SINIFI	530
10.3 ARDIŞIK (SEQUENTIAL) DOSYA YARATILMASI	532
10.4 RASLATİSAL ULAŞIM DOSYASI OKU (RANDOM ACCESS FILE)	555
10.5 DOSYA SIKIŞTIRILMASI (GZIP,GUNZIP,ZIP,UNZIP)	563
10.6 ALIŞTIRMALAR	571
BÖLÜM 11 GELİŞMİŞ JAVA BİLGİ İŞLEME YAPILARI.....	644
11.1 STRINGTOKENIZER SINIFI.....	644
11.2 STRİNG BUFFER SINIFI	648
11.3 VECTOR SINIFI.....	650
11.4 LIST(LİSTE) SINIFI.....	660
11.5 DİZİ (STACK) SINIFI	667
11.6 SIRA (QUEUE) SINIFI	670
11.7 TREE(AGAÇ) SINIFI.....	672

11.8 DICTIONARY ve HASHTABLE SINIFLARI	675
11.9 ARRAYS SINIFI VE SIRALAMA	677
11.10 ALIŞTIRMALAR	678
BÖLÜM 12 ÖRNEKLERLE SAYISAL ANALİZ	695
12.1 SAYISAL ANALİZE GİRİŞ	695
12.2 MATRIX SINIFI	695
12.3 MATRIX SINIFI ÖRNEK PROGRAMLARI	725
12.4 NUMERIC SINIFI (SAYISAL ANALİZ PAKETİ) ÖRNEK PROBLEMLERİ	753
12.5 ALIŞTIRMALAR	777
BÖLÜM 13 DATABASE (VERİ TABANI) PROGRAMLAMASINA GİRİŞ	778
13.1 TEMEL KAVRAMLAR	778
13.2 JAVA VE SQL BAĞLANTISI	781
13.3 ALIŞTIRMALAR	789
BÖLÜM 14 JAVA ANADİL (NATIVE LANGUAGE) PROGRAMLAMASINA GİRİŞ	815
C++ ve C PROGRAMLAMA DİLLERİNİN JAVA İLE BİRLİKTE KULLANIMI.....	815
14.1 TEMEL KAVRAMLAR	815
14.2 DEĞİŞKENLERİN VE JAVA NESLERİNİN AKTARILMASI	816
BÖLÜM 15 JAVA BEANS (JAVA FASULYELERİ) PROGRAMLAMA	827
15.1 TEMEL KAVRAMLAR	827
15.2 ALIŞTIRMALAR	843
16. NETWORK PROGRAMLAMAYA GİRİŞ	847
16.1 TCP/IP PROTOKOLÜ	847
16.2 ALT SEVİYE İLETİŞİM : UDP KULLANIMI	848
16.3 TCP BİLGİ İLETİMİ, SOCKET SINIFI	850
16.4 İNTERNET SİTELERİYLE BİLGİ ALIŞVERİŞİ, URL SINIFI	853
17. GÜVENLİK	860
17.1 APPLETLERDE GÜVENLİK	860
17.2 KONSOL PROGRAMLARINDA GÜVENLİK	866
17.3 GÜVENLİK SERTİFİKALARI	868
17.4 JAR DOSYALARININ GÜVENLİK KODUYLA İMZALANMASI	869
17.5 BROWSERLARDA GÜVENLİK	869
EK A TEXT SINIFI	871
EK B JAVA DERSİ SINAV SORULARI	876
JAVA PROGRAMLAMA DİLİ GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ GÜZ DÖNEMİ 2000 BİTİRME SINAVI SORULARI	876
JAVA PROGRAMLAMA DİLİ GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ 20 OCAK 2000 BİTİRME SINAVI SORULARI	885
18 REFERANS LİSTESİ.....	898

ÖNSÖZ

Bilgisayar programlaması günümüz teknolojisinin önemli bir ögesidir. Benim için bu yolculuk 1975 de aldığım ilk Fortran IV programlama dili kurduya başladı diyebilirim. O dönemde genelde teorik bazda gördüğüm programlama dersinde yaptığımız tek gerçek program örneği kartlara delerek verdigimiz ve sonuçları ertesi gün aldığımız ikinci dereceden denklemenin köklerini hesaplar bir programdı. Kullandığımız ege üniversitesi bilgisayar merkezindeki IBM tabanlı bir "mainframe" bilgisayardı. 1978 yılının sonlarında Unix tabanlı siyah beyaz monitörler veya kağıt yazıcılı monitorler aracılığıyla çalışan IBM makinalarla çalışmaya başladım. Tabii yıl gereği Fortran 77 versiyonuna ulaşmıştım. 1982 yılında Digital VAX bilgisayarlarında programlamaya ve aynı zamanda sistem görevlisi olarak çalışmaya başladım. Bu bilgisayarlar virtual (sanal) hafiza özellikleriyle çok büyük programların kullanılmasına izin veriyorlardı. Dil olarak Fortranın yanında Pascal da vardı. Üç boyutlu çizim ortamı için özel olarak geliştirilmiş evans-sutherland bilgisayı ile de bu dönemde tanıma fırsatı oldu. Bu arada Commodore 64 bilgisayarları piyasaya çıkmıştı. Evinde kullandığım bu küçük aletle asemblerler ve basic dillerini kullanarak birçok program geliştirdim. hatta assemblerler da yazılmış türkçe bir kelime işlem programı bile vardı. 80 li yılların sonu ve 90 li yılların başında Pc ler, dos ortamı ve diliyle tanıştım. sonra PC ortamı Machintosh ve Amigadan çok sonra grafik programlama ortamını büyük bir buluş olarak ortaya attı ve Windows sistemlerini çıkardı. Ve ikinci bilgisayarı param oldukça aldığım parçaları birleştirerek kendim oluştururdum. Bu bir 80386 Pc bilgisayarı idi. Artık genelde c dilini kullanıyorum. C++ dilini gerçek anlamda kullanmaya başlamam, bilgisayar konusundaki lisansüstü kursunu alırken oldu. Bundan sonraki tüm programlama uygulamalarında nesne kökenli olan bu dili kullanır oldum. Aynı kursta Lisp, Parallaxis gibi değişik dillerle de çalışmıştım. Bu arada evde de yine PC tabanlı Pentium 100 bilgisayara terfi ettik. Bana Java dilini eşimin aldığı lisansüstü kursundaki Java kursu gösterdi. Java dili yeni çıkmış, fakat yeni olmasına rağmen okullardaki bilgisayar bölümleri eğitimde hızla bu dile yönelmişlerdi. Bu dile başlar başlamaz da çok sevdim. Her şey tüm diğer dillere göre çok daha iyi düşünülmüş ve planlanmıştı. Orijinal olarak C++ da yazılmış bir çok kodu fazla bir gayrete gerek duymadan java koduna çevirerek iş ortamında kullanmaya başladım. 1998 yılında Dokuz eylül üniversitesinde mühendislik bilimleri fakultesinde öğretim üyeleri ve master ve doktora öğrencileri için "Java programlama dili" dersi açtım. Bu kitabın temelini bu ders için hazırladığım ders notları oluşturmuştur. 1999 yılında Gebze ileri teknoloji Enstitüsünde aynı dersi yinelerken ders notları ilave problemlerle de zenginleşerek biraz daha kitabı alındı. Son bir gözden geçirmeden sonra elinizdeki haline geldi.

Java dili yapısı, kullanım olasılıkları ve kolay kullanımı, zengin kütüphaneleriyle geleceğin dili olacağına programcıların çoğunun inandığı bir dildir. C++ şu an itibariyle çok daha fazla kullanılan bir dil olam özelliğini sürdürmektedir, fakat C++ hatalara izin veren yapısıyla başlangıç seviyesi programcılara hitap eden bir dil değildir. Rahatlıkla hata yapabilirsiniz, ve yaptığınız bu hatalar rahatlıkla gözden kaçabilir. Hemen şunu ilave edeyim, java daha yeni emeklemeye başlamış bir bebektir, ama gelecek on yıl içinde çok iyi bir konuma yerleşmenin işaretlerini şimdiden vermektedir.

Türkiyede de programcılık henüz yeni yeni oluşmaya başlayan bir dal. Hem programcı kapasitesi, hem de isteklere cevap vermek için yeterli boyutta değil henüz. Programlama teknolojisinde iyi bir boyuta gelebilme paket kullanımından değil programlamadan geçmektedir. Bu yüzden bu kitabı eğer yeni programmacılarımıza bir şeyler verebilir ve onların gelecekte daha iyi bir noktada olabileceklerini sağlayabilirse ben emeğimin karşılığını almış olurum.

Yeni başlayanlara bir tavsiye bilgisayar dillerini öğrenme insan dillerini öğrenmeye benzer. Temel mantığını kapana kadar biraz zorlanabilirsiniz. Sakın ümitsizliğe kapılıp bırakmayın. Bir kere temel mantığını anladıkten sonra ne kadar kolay olduğunu göreceksiniz. İyi çalışmalar.

Dr. Turhan Çoban
TÜBİTAK, MAM
21 Mart 2000, 21.36
turhan@mam.gov.tr

İKİNCİ BASKININ ÖNSÖZÜ

Bu Java kitabının birinci baskısı umduğumun çok üstünde ilgi gördü. Sanıyorum bunun temel sebebi türkçe kaynak eksikliğinin oldukça fazla olması. Bu ilgi üzerine kendimi biraz daha ciddi bir çalışma oluşturmak için sizlere borçlu hissettim. Umuyorum bu ikinci baskı çok daha fazla işinize yarayacak ve sadece bir ders notu olmanın dışında biraz daha bir referans kitabına yaklaşacak bir eser olacak. Kitabın ilk baskısında okuyuculardan oldukça yoğun mektuplar aldım, bu mektuplarda beğenilerinin yanı sıra benim de haklı bulduğum eleştiriler de

yer alıyordu. Gelen eleştirilerden en yoğunu kitabındeki kodların bir CD olarak sunulmamış olmasıydı. Bu ikinci baskıda bunu sağlayabilmek için elimden geleni yapacağım, fakat başarılı olamazsam kodları www.geocities.com/turhan_coban/ adresi üzerinden yayılamaya çalışacağım. İkinci eleştiri Text sınıfını bulamamak veya çalıştırılamamak idi. Text sınıfını ben özellikle eklerde de vermiştim. Bu sınıfın amacı javanın çok yoğun kullandığı hata analizini aşarak en azından başlangıçta okuyuculara (öğrencilere?) kolaylık sağlamaktı. Derslerde öğrenciler için bu kolaylığı sağladı da, fakat sanırım kitapta kodların verilmeyişile birlikte kolaylık yerine zorluk getirdi. Bu ikinci baskıda birinci bölümde itibaren girdi çıktıda birden fazla alternatif bulacaksınız. Bunların birisi de swing sınıfından olan JoptionPane sınıfı, kullanılması gayet basit olan pencere tipi bir girdi çıktı sınıfı. Swing sınıfını bu baskıda bir öncekine göre oldukça yoğun kullandım. Bazı programlarında aradaki benzerlik ve farkları görebilmeniz amacıyla swing ve awt versiyonlarını arka arkaya koydum. Swing awt'ye göre çok daha kompleks bir yapı, yalnız olasılıkları arttırdığı için java dünyasına şimdiden tamamen hakim olmuş durumda. Bu baskıda ilk baskıda olmayan java güvenlik ve java anadil programlama bölümlerini bulacaksınız. Ayrıca daha öce ayrı bir bölüm olarak verdığımız swing ve 2D grafik programlama bölümünü de ilgili konulara yayarak kaldırıldım. Bence ilk baskiya göre elinizde daha kullanışlı bir kitap var. Elbette mükemmel değil, ama ilk baskının en azından dört katı bir emeğin sonucu. Umarım sizlere java öğrenme yolunda bir ilk adım olarak faydalı olur.

Bu yeni baskıyı geliştirmemde büyük payları olan Gebze Yüksek Teknoloji Üniversitesi, java dersi öğrencilerine, ayrıca bana iyi bir eğitim vermek için ellerinden gelen çabayı harcayan babam Osman ve annem Hatice'ye, Kardeşlerim Birsen, Nurhan ve İrfan'a ve bu kitabın yazılması sırasında verdiği destek için eşim Meral'e teşekkürlerimi bildirmek isterim.

Dr. Turhan Çoban
TÜBİTAK, MAM
11 Mart 2001, 23.34
Turhan.Coban@posta.mam.gov.tr

BÖLÜM 1 : JAVAYA GİRİŞ

1.1 NİCİN JAVA PROGRAMLAMA DİLİ

Java Programlama dili şu anda dünyadaki en popüler programlama dillerinden biri haline gelmiştir. Java SUN bilgisayar şirketince orijinal olarak elektrikli ev araçlarının (mikrodalga fırınları, buzdolapları, televizyonlar, uzaktan kumanda cihazları vs.) birbirileşmesini sağlaymayı amaçlayan bir proje içerisinde 1991 yılında geliştirilmeye başlandı. Orijinal adı bu dilin yaratıcıları James Gosling, Patrick Naughton, Chis Wartdh, Ed Frank ve Mike Sheridan tarafından **Oak** olarak konulan programlama dili daha sonra bu isimde başka bir programlama dili olduğu keşfedilince o anda bir kahvehanede kahve içen programlama gurubu tarafından kahve markasından esinlenerek **Java** olarak değiştirildi. Akıllı elektronik ev araçları pazarı SUN gurubunun tahminlerinden çok daha yavaş bir gelişme gösteriyordu. Bu yüzden Java dili projesi ticari bir geliştirme projesi olarak büyük olasılıkla iptal edilecekti. 1993 Yılında "World Wide Web" büyük bir atılım göstererek bütün dünyaya yayılmaya başladı. Javann Dinamik Web sayfaları hazırlamadaki büyük potansiyelini gören SUN şirketi projeyi bu tarafa yönlendirdi ve bu javaya yeni bir canlılık ve yaşama umudu sağladı.

Mayıs 1995 de SUN javayı büyük bir konferansta tanıttı. Program iş dünyası tarafından derhal büyük bir ilgiyle karşılandı. Java Modern bilgisayar dünyasının ses, grafik işlem, haberleşme gibi ihtiyaçlarına cevap verebilen ve Ticari gayeler için hazırlanan bir Program dili olarak daha önceki bilgisayar dillerinin hiç birinin kapsayamadığı özellikleri içermekteydi. Bunun yanı sıra dil komut yapısı olarak C++ diline çok yakın olması da öğrenilmesini kolaylaştırıyordu. SUN Javayı "World Wide Web" de kullanmak isteyen herkese ücretsiz olarak sundu. Java internette yayınlanmasıının ardından çok büyük bir patlama yaşadı. 1997 ye gelindiğinde dünyadaki bütün bilgisayar okullarında temel bilgisayar dili olarak gösterilmeye başlandı. Dünyada şu anda hala en çok kullanılan bilgisayar dili olan C++ dilinin yapılan hataları tam olarak denetlememesi programın çalışma hızını artırma yönünden iyi bir özellik olsa da profesyonel programcılar dışında kullanılmasını sınırlandırıcı bir etki yapıyordu. Java ise bütün hataları bildiren yapısı ve modern bilgisayarın bütün fonksiyonlarına ulaşabilen kütüphaneleriyle programcılar çok daha kolaylıkla öğrenebileceği bir dildir. Burada hemen şunu da belirtelim. C dili hızlı çalışma amacı birinci planda tutularak yaratılmış bir dildir. Javada ise emniyet ilk planda yer almıştır. Hız açısından düşünüldüğünde java C (ve C++) diliyle rekabet edemez. zaten program derleyicisi de C++ dilinde yazılmıştır.

Javanın diğer önemli bir temel özelliği Nesne kökenli (object oriented) bir dil olmasıdır. Nesne kökenli diller, nesneleri gerçek dünyadakine daha benzer bir yapıda tanımlayarak anlaşılmalarını kolaylaştırırlar. Nesneleri gerçek dünyadaki gibi masa, sandalye, bilgisayar, gerçek gaz, ısı eşanjörü gibi tanımlayarak programlamak insan beyninin anlaması açısından çok daha kolaydır. Bundan önceki tam nesnel kökenli programlama dillerinden hiçbirini çok yaygın kazanmamıştı. Bu tür dillere Smalltalk,u Örnek verebiliriz. C++ nesnel kökenli programlama yapabilen bir dildir. Fakat yapısal (structural) birprogramlama dili olan C dilinin bir uzantısı olarak geliştirildiğinden tam anlamıyla nesnel kökenli bir dil olduğu söylenemez.

Java dilini geleceğin dili yapan diğer bir özelliği de çok kullanımlı (multi-tasking) ve paralel kullanımlı(multi-treading) bir dil olmasıdır. Çok kullanımılık birden fazla işlemin aynı anda yapılabilmesinin tanımıdır. Paralel kullanımılık ise birden fazla programın aynı anda hafızayı beraber kullanarak kullanılabilirliğidir. Örnek olarak World ve Excel programlarının Windows NT ortamında aynı anda kullanılmasıdır. ADA gibi eski Program dillerinde çok kullanımılık programlanabiliyordu. Paralel kullanım olanaklarını sunan ilk bilgisayar dili ise javadır. Paralel kullanım paralel programlama kavramından ayrıdır ve karıştırılmamalıdır. Paralel programlanmada birden fazla Bilgi işlem Ünitesine (CPU) ayrı programlar veya bir programın ayrı parçaları gönderilir. Paralel kullanımda ise bir CPU nun kullanım zamanı küçük parçalara ayrılarak değişik Program veya Program parçacıkları bu zaman paketçiklerini paylaşarak kullanırlar.

Javayı önemli bir Program dili haline getiren en önemli özelliği ise kullanılan bilgisayardan bağımsız olmasıdır. Javada yazılan bir Program Unix, Machintosh, Windows 95 veya Windows NT veya herhangi bir 32 bit makinada hiç değiştirilmeden kullanılabilir. Java programlarının grafikleri "Wold Wide Web" sayfalarının programlama dili olan html (hypertext markup language) ile aktarılır. Bu yüzden html ve java programlarını birlikte kullanmak ve java programlarını gerçek zamanda www sayfalarında göstermek mümkündür. Kendi web sayfanızı veya webde yazılmış kitabınızı bütün dünyaya aktarırken yaptığınız analizleri de bu kitabın dinamik bir parçası olarak sunabilirsiniz. Bu Türkiye gibi ülkelerin dünyaya kapılarını daha kolay açmaları açısından oldukça önemli bir özellikleştir.

Javanın HTML diliyle kullanılabilmesi, bazı kişilerde htmlin bir parçası olduğu gibi bir kavram gelişmesine yol açmıştır. Programlamaya yeni başlayanlar sık sık html ile javayı birbirine karıştırırlar. veya javanın HTML'nin bir uzantısı olduğunu düşünürler. Bu temelde oldukça yanlış bir varsayımdır. HTML internet belgelerini birbirine bağlamak amacıyla geliştirilmiş bir belge işlem sistemidir ve bir programlama dili değildir. HTML ile javanın tek gerçek ilgisi HTML deki applet komutudur. bu komut yardımıyla java dilinde yazılmış programların sonuçları HTML ortamina dinamik olarak aktarılabilir.

1.2 JAVA PROGRAMINI YAZMA VE ÇALIŞTIRMA

Her programda olduğu gibi java programlarında da ilk evre programı yazmaktır. Java programları her editör programıyla yazılabilirler. Örneğin Unix sistemlerinde **vi** veya **emacs**, **windows 95** ve **windows NT** de **DOS Edit**, veya **Windows Notedefteri (notepad)** veya **WordPad** programları kullanılabilir. Bildiğiniz ve kullanmakta rahat olduğunuz herhangi bir editör programı varsa rahatlıkla kullanabilirsiniz. **Windows Notedefteri (notepad)** editöryüle ilgili karşılaşabileceğiniz bir programı burada hemen belirtmekte yarar var. Bu program tüm dosyaların arkasını txt eki getirmeye çalışır. Programı direk içinde yazıp çalıştırabilen Microsoft Visual J++, Symantec Visual Café, Jpad, Borland JBuilder gibi Java geliştirme paketleri de mevcuttur. Bu kitapta bu paketler yardımıyla program yapılmasına direkt olarak hiç dechinmemiştir. Kişisel olarak bu tür paketler java dilini öğrenme açısından bilhassa yeni programlama öğrenenlere çok fazla yardımcı olacağını sanıyorum, üstelik sadece o paketi kullanmayı öğreneceğiniz için bir tür bağımlılığa sebep olabilirler. Fakat bu paketler uzman hale geldikten sonra işinizi kolaylaştmak açısından çok yararlı olabilir.

Şüphesiz programı yazmadan önce Java compilerinin bir kopyasını edinmeniz de gereklidir. Java geliştirme seti (Java Developers kit - JDK) SUN bilgisayar şirketinin

<http://www.java.sun.com>

veya

<http://www.javasoftware.com>

internet adreslerinden bedava çekilebilir. Çekme zahmeti istemiyorsanız SUN bu programları CD formatında da satmaktadır. Sun programın kitaplar ve üçüncü el kaynaklarda verilmesine müsade etmemektedir. Ayrıca derleyici sürekli olarak yenilendiğinden orijinal kaynağından her zaman en yeni versiyonunu elde edebilirsiniz. Java ile yaptığınız lisans anlaşmasına göre program dilini bedava olarak kullanabilirsiniz, ancak satacağınız profesyonel bir program geliştirdiğinizde sun şirketine kazancınız üzerinden belirli bir yüzde ödemeniz gerekecektir.

Java programı **isim.java** şeklinde editörle yazılır. Java programınız yazıldıktan sonra ilk step java derleyici (compiler) programıyla **isim.class** ara bilgisayar kodunu oluşturmaktır. Java programını yazarken dikkat etmeniz gereken diğer bir özellik de javanın nesne sınıfının ismiyle dosyadaki isminin aynı olmasıdır. Bilgisayar konsol ekranında (DOS veya UNIX Console)

javac isim.java

Komutunu bilgisayara gönderdiğinizde programınızda hata yoksa java derleyicisi **isim.class** dosyasını yaratır. Eğer programınızda hata varsa hataların listesini verir. Programlamanın dili ingilizce olduğundan bu hata açıklamaları da ingilizcedir. Editör programına geri dönüp hataları düzeltmeniz, programı tekrar kaydetmeniz ve sonra bu komutu tekrar denemeniz gerekecektir. Java programları iki ortam için yaratılabilir. Birincisi Konsol ortamıdır (DOS, Unix console gibi). Konsol ortamını kullanırken java dilinin konsol komutu java kullanılarak isim.class dosyası yerel bilgisayarin makine diline program çalışırken dönüştürülür. Bu temelde bir tercüme (interpretation) işlemidir. Sonuç olarak java dilinde iki kademeli bir proses kullanılır, birinci işlemede java dosyası class dosyasına çevrilir (derlenir), ikinci işlemede ise class dosyası yerel bilgisayarin makine diline dönüştürülür. konsol ortamında çalıştırılan program sonuçları yine konsol ortamında alınacağı gibi pop-up pencere grafik ortamında da alınabilir. İkincisi ise internet browser penceresi (Applet) ortamıdır. Eğer Program konsol ortamında yazılmışsa

java isim

komutu kullanılarak sonuçlar ekranda veya açılan bir pencerede görülebilir. Eğer Program applet ortamında yazılmışsa java programına ilave olarak html dilinde **isim.html** dosyası açılır. En basit şekilde **html** dosyası şu

şekilde tanımlanabilir :

```
<html>
<applet code="isim.class" width=200 height=50>
</applet>
</html>
```

Buradaki width pencerenin enini pixel olarak, height'da yüksekliğini yine pixel(görüntü noktası) olarak vermektedir. daha sonra bu dosya konsoldan

appletviewer isim.html

komutuyla çalıştırılabilir. Buradaki appletviewer komutu java program paketindeki basitleştirilmiş bir browser programıdır. Bu komut yerine isim.html dosyası herhangi bir browser programına yüklenerek görülebilir. Burada hemen şunu not etmeliyiz: browser program yazılımcıları genelde java versiyonlarını kendi programlarına aktarmakta biraz yavaş kalmaktadırlar bu yüzden en yeni java derleyicisiyle derlediğiniz bir programı browser'ınız göstermeye bilir. Buna karşılık java "plug-in" denilen küçük programlar geliştirmiştir. Diğer bir olasılıkta elbette eski java derleyicınızı hemen atmayıp bir süre daha (yeni versiyonu browser programları tarafından tam olarak tanınıncaya kadar) profesyonel applet programları geliştirirken kullanmanızdır.

Programcılıkta adeta adet haline gelen ilk uygulama daima o dilde ekrana Hosgeldiniz çıktısi almaktır. Bizde bu kuralın dışına çıkmayalım. Aşağıdaki listede ilk konsol programımız **Hosgeldiniz.java** görülmektedir.

Program 1.1 *Hosgeldiniz.java* konsol programı

```
import java.io.*; //java girdi ciktigini sinifini cagir
class Hosgeldiniz
{
public static void main(String args[])
{
System.out.println("Java Sınıfına Hos Geldiniz!");
}
}
```

bu programı çalıştırmak için

```
javac Hosgeldiniz.java
java Hosgeldiniz
```

komutlarını konsolda vermemiz yeterlidir
konsol çıktısi olarak:

Java Sınıfına Hos Geldiniz!

Yazısını alırız.

Şimdi bu programdaki satırlara tek göz atalım.

import java.io.*; Bu deyim java konsol girdi çıktı kütüphanesindeki metotları programımıza aktarır.
//java girdi ciktigini sinifini cagir
// işaretiley baþlayan yazılar bilgisayar tarafından kullanılamaz sadece programı okuyanı bilgilendirmek amacı taþır /* */ işaretleriyle de verilebilir. Örneðin :

```
/* Bu kısımda yazılanlarla bilgisayar ilgilenmez
Sadece insanların programı anlaması içindir
*/
```

class Hosgeldiniz Class (sınıf) nesne kökenli programlama yapısının temel taşıdır. Her Class'in kendine özgü bir ismi olması gereklidir. Daha önce yazılmış Class'lar **import** deyiðimiyle programa çağırılabilirler. Yeniden

yazılmaları gerekmektedir. Class yapısı ve tanımlamaları üzerinde daha sonraki bölümlerde çok daha detaylı bilgi verilecektir.

{ } başlatma ve bitirme : Sınıf tanımlaması da dahil olmak üzere Tüm Program parçacıkları { işaretileyi başlar ve } işaretileyi son bulur. {} işaretleri arasında yazılan program parçacıklarını normal metin yazılmasındaki paragraf kavramına benzetebiliriz. Ayrıca program koduna baktığımızda ";" işaretine sık sık rastlıyoruz. Bu işaret java komutunun bittiğini belirtir. Cümle yağısında kullanılan nokta "." işaretini gibi işlev görür. Java programlarını istersek ";" ve "{}" ayraçlarını kullanarak tek bir satır olarak yazabilirmiz. Fakat bu programın bizim tarafımızdan okunurlüğünü zorlaştırmaktır. Program kodunu yazarken boşluk ve satır aralıklarını bırakarak insanlar tarafından programın okumasını kolaylaştırabiliyoruz. Modern programcılıktaki en büyük sorunlardan biri bigisayar kodlarının bakım ve onarımı sorundur. **Ticari olarak yazılan her programın belli bir ömrü olduğunu ve daha sonra başka bir programcı tarafından üzerinde tekrar çalışılacağını sürekli olarak hatırlamak zorundayız.**

public static void main(String args[]) diğer dillerde fonksiyon (function), subroutine, procedure gibi adlarla anılan Program alt parçacığının başlığı. Bu Program parçacıkları javada Metot (Method) adıyla anılır. Metotların girdileri ve çıktıları olabilir. Bu metotun adı **main** (ana Program) dır. String değişken türü args[] değişkeni girdi olarak girebilir. Bu metotta **void** çıktı türü kullanılmıştır. Void kelimesi bu metotdan hiçbir çıktı alınmayacağına belirtir. **public** metota dışarıdan ulaşılabilme izni verir.

System.out.println("Java Sinifina Hos Geldiniz!"); Bu satır System.out sınıfından println metodunu çalıştırır. Bu metot method girişinde gönderilen String tipi değişkeni konsol çıktısına aktarır. Fortran dilindeki write(*,*) veya print*, c dilindeki printf, c++ dilindeki cout<< terimlerinin javadaki tam karşılığıdır.

"Java Sinifina Hos Geldiniz!" stringi aynen aktarılırken içinde c dilinde kullanılan temel kontrol terimleri yer alabilir ve aynen c dilindeki anlamlarıyla kullanılabilir. Bu kontrol terimlerinden en yaygın olarak kullanılan bazları:

\n alt satıra geç
\t bir sonraki tab'a atla
\r gir tuşu bas
\\" işaretini yaz
\\" " işaretini yaz

şeklinde gösterilebilir.

İkinci konsol programımız HosgeldinizSW.java Program 1.2 de verilmiştir. Bu program SWING kütüphanesinin alt kütüphanesi olan JOptionPane kütüphanesi kullanılarak yaratılmıştır. Çıktı yazısı ekranda grafik pencere olarak belirir ve Java sınıfına Hos geldiniz! mesajını verir. Tüm bu kütüphanelerin detayları ilerideki bölümlerde verilecektir. Şimdi nasıl kullanıldıklarını öğrenmekle ve programımızı örnek programlara benzeterek yazmaya çalışacağız. Uzman programcılar isterlerse ilgili bölümlerden komut detaylarına derhal girebilirler.

Program 1.2 HosgeldinizSW.java programı

```
import javax.swing.JOptionPane;

class HosgeldinizSW {
    public static void main (String args[])
    {
        JOptionPane.showMessageDialog(null,"Java sınıfına hoş geldiniz!",
        "HosgeldinizSW programı",JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}
```

Bu programı

```
javac HosgeldinizSW.java
ve
java HosgeldinizSW
```

komutlarını kullanarak çalıştırduğımızda

01001.JPG



Şekil 1.1 HosgeldinizSW dosyasının konsol SWING pop-up grafik çıktıları kullanarak görülmesi

Bu çıktıyı alabilmek için java SWING kütüphanesi içerisinde yer alan **JOptionPane.showMessageDialog(...)** metodunu kullandık. Bu metodun görevi System.out.println metodıyla aynıdır. Farkı ise grafik ortamda çıktı vermesidir. Metot içindeki ikinci sırada verilen String tipi sabit, "Java sınıfına hoş geldiniz!" grafik çıktıya aktarılmıştır. Üçüncü sırada verilen string tipi "HosgeldinizSW programı" sabiti de grafik kutusunun adını belirtmiştir. JOptionPane.INFORMATION_MESSAGE komutu ise bilgi (information) mesajı penceresi açılacağını belirtmiştir. (grafikteki i harfi – information – bilgi kelimesinin baş harfidir). Bu alanda kullanabileceğimiz diğer alternatifler şunlardır :

JOptionPane.ERROR_MESSAGE : hata mesajı
JOptionPane.WARNING_MESSAGE : uyarı mesajı
JOptionPane.QUESTION_MESSAGE : soru mesajı
JOptionPane.PLAIN_MESSAGE : sade (ikonsuz) mesaj

System.exit(0); metodu çıktı penceresinde görülen OK düğmesine basıldığında pencerenin kapanmasını sağlar.
import javax.swing.JOptionPane; deyimi swing kütüphanesinde yer alan JOptionPane sınıfını çağırır.
Programdan da görüleceği gibi sınıfımızın (class) adı HosgeldinizSW dir. Bir önceki programdaki gibi bu programda kullandığımız komutlar **public static void main(String args[])** komutuyla tanımlanan ana metodun içinde yazılmıştır.

Üçüncü konsol program örneğimizde iki java programını birlestireceğiz İlk programımız BasicWindowAdaptor.java Program 1.3 de verilmiştir. Program 1.4 ise HosgeldinizSWF.java programını kapsamaktadır. BasicWindowMonitor programını bundan sonraki programlarımızda sık sık kullanacağız.

Program 1.3 [BasicWindowMonitor.java programı](#)

```
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 programı](#)

```
import javax.swing.*;
import BasicWindowMonitor;

public class HosgeldinizSWF
{
    public static void main(String[] args)
    {
        JLabel etiket=new JLabel("Java Programına Hoş geldiniz!");
        JFrame cerceve=new JFrame("HosgeldinizSWF");
        cerceve.addWindowListener(new BasicWindowMonitor());
```

```

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

```

Burada iki ayrı program bulunmaktadır. Bu iki programlık seti çalıştırmak için önce iki programı da derlemek gerekir.

**javac BasicWindowMonitor.java
javac HosgeldinizSWF.java**

sonra ikinci prgramı

java HosgeldinizSWF

komutunu kullanarak çalıştırabiliriz. Burada birinci programın görevi hosgeldinizSWF penceresini köşedeki X işaretine basduğımızda pencereyi kapatmaktadır. Bu programı bundan sonraki birçok uygulamamızda kullanacağınız. Programımız JFrame denilen grafik pencere ortamında yazılmıştır. Çıktı penceresi :

01002.JPG



Şekil 1.2 HosgeldinizSWF dosyasının konsol SWING JFrame pop-up grafik çıktıtı kullanarak görülmesi

Gördüğü gibi burada yazma işlemini yaptırmak için tek bir komut yerine bir komutlar ve metodlar zinciri kullanıldı. Bu program swing grafik kütüphanesiyle grafik formatında çıktı almanın çok daha standart bir yoludur. Temel olarak SWING grafik penceresine JLabel sınıfı tipi etiket eklenerek yazı grafik ekrana aktarılmıştır.

Program 1.5 de yine swing metodu kullanılmıştır. Buradaki küçük farklılık bir önceki HosgeldinizSWF programında kullanılan JLabel nesnesi üzerinden yazdırma yerine paint metodunun kullanılmış olmasıdır. Burada da BasicWindowMonitor programı kullanılmıştır. fakat bu programı daha önce derlemiş olduğumuzdan tekrar derlememiz gerekmek. Sadece

javac Hosgeldiniz1SWF.java

komutıyla bu programı derlememiz ve

java Hosgeldiniz1SWF

komutıyla çalıştırırmamız kafidir.

Program 1.5 Hosgeldiniz1SWF.java programı

```

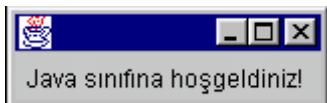
import javax.swing.*;
import java.awt.*;
import BasicWindowMonitor;
public class Hosgeldiniz1SWF extends Jframe
{
    public void paint(Graphics g)
    {
        g.drawString("Java sınıfına hoşgeldiniz!",10,40);
    }

    public static void main(String[] args)
    {
        Hosgeldiniz1SWF cerceve=new Hosgeldiniz1SWF();
        cerceve.addWindowListener(new BasicWindowMonitor());
        cerceve.setSize(160,50);
        cerceve.setVisible(true);
    }
}

```

Program çıktısı :

01003.JPG



Şekil 1.3 Hosgeldiniz1SWF.java dosyasının konsol SWING pop-up grafik çıktısı kullanarak görülmesi

Burada yazıyı temel olarak grafik metodu olan

```
g.drawString("Java sınıfına hoşgeldiniz!",10,40);
```

metodunu kullanarak pencere ortamına aktardık. bu metodda kullanılan 10,40 deyimi yazının grafik ortamındaki x ve y koordinatlarını göstermektedir. y koordinatı yukarı doğru tanımlanmıştır.
cerceve.setSize(160,50);
deyimi açılan pencerenin boyutunun 160 genişliğinde ve 50 yüksekliğinde olacağını belirtmektedir.
cerceve.setVisible(true);
terimi ise pencerenin görünmesini (çizilmesini) sağlamaktadır.

Buraya kadar kullandığımız programlar konsol programlarıydı. Bu programlarda da gördüğünüz gibi grafik ortamını yoğun olarak kullanabiliyoruz. Yalnız programların çalıştırılması konsoldan verilen **java isim** komutuyla gerçekleştirilmektedir.

Şimdi de applet grafik ortamında çıktı almaya örnek verelim. İlk applet programımız **HosgeldinizApplet.java** listesi Program 1.6 da verilmiştir

Program 1.6 **HosgeldinizApplet.java** programı

```
import java.applet.Applet;
import java.awt.*;
public class HosgeldinizApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Java sınıfına hoş geldiniz!",25,35);
    }
}
```

bu programı çalıştırmak için önce HosgeldinizApplet.html HTML dili programını yazmamız gereklidir

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 edilmiş olan **HosgeldinizApplet.class** dosyasının gösterilmesini istemektedir. Şüphesiz aynı komut çok daha kompleks bir html dosyasının içinde de aynı görevi görmek amacıyla kullanılabilirdi.
Programı çalıştırmak için

```
javac HosgeldinizApplet.java
appletviewer HosgeldinizApplet.html
```

komutlarını konsolda vermemiz yeterlidir. Elbette HosgeldinizApplet.html dosyasının yukarıda belirtildiği gibi hazırlanmış olması da gereklidir. Fakat Konsol programlarından farklı olarak sonucu konsol yerine Applette (browser programında) görürüz. Aynı sonucu **appletviewer** deyimini kullanmadan herhangi bir browser

programı çağrıarak da elde edebiliriz. Browser programında sonucu görebilmek için önce programı açmamız sonra da HosgeldinizApplet.html dosyasını bularak açmamız gereklidir. Çıktı görüntüsü aşağıdaki gibi olacaktır.

01004.JPG



Şekil 1.4 HosgeldinizApplet.html dosyasının appletviewer programı kullanarak görülmesi

Bu programında temel olarak bazı farklılıklar dışında bir önceki programlara benzediğini gözlemlayabiliriz. Bu programın temel farkı internet ortamından herhangi bir bilgisayardan da (gerekli ayarlama ve kopyelemelerden sonra) görülebilir olmasıdır.

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

deyimleri java kütüphanesindeki Applet ve pencere(awt) sınıfı grafik programlarını programımıza aktarır.

public class HosgeldinizApplet extends Applet deyimi HosgeldinizApplet isimli sınıfı (class) yaratır. Bu sınıf tüm Applet sınıfının bir devamı olarak tanımlanmaktadır (**extends Applet**) diğer bir deyimle Applet sınıfının bütün metodları extends deyi̇miyle HosgeldinizApplet sınıfımızın içine kopyalanmıştır.

public void paint(Graphics g) deyimi appletin grafik olarak çizilmesini sağlar. Bu Metot **Graphics** sınıfı **g** nesnesini girdi olarak kullanır. Metotun çıktısı yoktur.

g.drawString("Java sınıfına hos geldiniz!",25,35); Bu terim Appletin x = 25, y=35 noktasına **Java sınıfına hos geldiniz!** Yazısını çizer. DrawString Grafik sınıfı **g** nesnesine ait bir metottur.

Bu programda temel grafik kütüphanesi olarak abstract window toolkit (awt) kullanılmıştır. ayrıca konsol programlarından farklı olarak bu programda main metodunun kullanılmadığını da dikkat ediniz. bir program hem applet hem de konsol olarak da oluşturulabilir, yani hem main metodu olabilir hem de applet (html) ortamında çalışabilir.

Aynı appleti oldukça benzer bir şekilde SWING kütüphanesini kullanarak da oluşturabilirdik.

Program 1.8 HosgeldinizSWA.java programı

```
import javax.swing.JApplet;
import java.awt.Graphics;

public class HosgeldinizSWA extends JApplet
{
    public void paint(Graphics g)
    {
        g.drawString("Java sınıfına Hoşgeldiniz",25,25);
    }
}
```

01005.JPG



Şekil 1.5 HosgeldinizSWA.java dosyasının applet SWING pop-up grafik çıktısı kullanarak görülmesi

Bu programdaki

```
import javax.swing.JApplet;
import java.awt.Graphics;
```

deyimleri java swing kütüphanesindeki JApplet ve pencere(awt) sınıfındaki Graphics alt programlarını (sınıflarını) programımıza aktarır.

public class HosgeldinizSWA extends JApplet

deyimi HosgeldinizAppletSWA isimli sınıfı (class) yaratır. Bu sınıf swing sınıfındaki JApplet sınıfının bir devamı olarak tanımlanmaktadır (**extends JApplet**). Diğer bir deyimle JApplet sınıfının bütün metotları extends deyimiyle HosgeldinizAppletSWA sınıfımızın içine kopyalanmıştır.

Aynı işi yapmak için bir çok değişik program yazdık. Bazılarımız buna ne gerek vardı, tek bir şekilde bu işi halledemezmi idik diyebilirler. Haklıdırarda. Fakat modern bilgisayar dillerinde kullanıcıları da göz önüne almak durumundayız. Aynı işi daha renkli ve çeşitli şekillerde yapmamız temel olarak program kullanıcılarına daha çeşitli alternatifler sunabilir. Aynı zamanda diğer programlama alternatiflerine göre daha üstün bir çalışma ortamı yaratabilir.

Şimdi de programımıza bir basamak daha ekleyelim ve program kullanıcılarla isimlerini sorsun, yani ekrandan bir veri girelim. Bunu yapmak için ilk tanımlayacağımız program konsol ortamında çalışan **Hosgeldiniz2.java** programı olacaktır. Bu programın çıktısı dos konsol ortamında şekil 1.6 da görülmektedir. Program ekranda isminizi giriniz yazacak, ve ekranda isim geldikten sonra

java sınıfına hosgeldiniz! isim

çiktısını verecektir.

Program 1.9 *Hosgeldiniz2.java programı*

```
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 Sınıfına Hos Geldiniz! "+isim);
        cin.close();
    }
}
```

```
=====
C:\co\java\prog>java Hosgeldiniz2
isminizi giriniz :
Turhan
Java Sınıfına Hos Geldiniz! Turhan
C:\co\java\prog>
=====
```

Şekil 1.6 Hosgeldiniz2.java dosyasının sonuçlarının dos çıktısı kullanarak görülmesi

Bu programda okuma işlemini yapmak için önce

```
BufferedReader cin=new BufferedReader(new InputStreamReader(System.in));
Deyimini kullanarak bir okuma kanalı açtık, sonra
```

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

Deyimindeki **cin.readLine()** komutuyla ismi cin kanalından okuyarak isim değişkenine aktardık.

Aynı işlemi yapan fakat girdi çıktı sınıf tanımlarında Ek 1 de verilen Text sınıfını kullanan Hosgeldiniz3.java programı şekil 1.10 da verilmiştir.

Program 1.10 Hosgeldiniz3.java programı

```
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 programının çıktısı da Hosgeldiniz2.java programıyla aynı olacaktır.

Şimdi de swing kütüphanesindeki JoptionPane.showInputDialog() kullanarak girdi penceresi açan ve çıktı penceresine JoptionPane.showMessageDialog() kullanarak mesajı yazan Hosgeldiniz1SW.java programını göreceğiz.

Program 1.11 Hosgeldiniz1SW.java programı

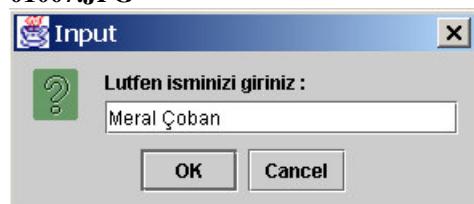
```
import javax.swing.JOptionPane;

class Hosgeldiniz1SW {
    public static void main (String args[])
    {
        String isim=JOptionPane.showInputDialog("Lutfen isminizi giriniz :");
        JOptionPane.showMessageDialog(null,"Java Sinifina Hos Geldiniz! "+isim,
        "Hosgeldiniz1SW programı",JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}
```

Bu programı çalıştırmak içinde

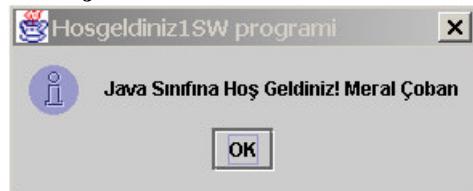
```
javac Hosgeldiniz1SW.java  
komutuyla bu programı derlememiz ve  
java Hosgeldiniz1SW  
komutuyla çalıştırmanız gerekecektir. Program girdi ve çıktı pencereleri şekil 1.7 ve Şekil 1.8 de verilmiştir.
```

01007.JPG



Şekil 1.7 Hosgeldiniz1SW.java dosyasının pencere popup girdi penceresi

01008.JPG



Şekil 1.8 Hosgeldiniz1SW.java dosyasının pencere popup çıktı penceresi

Simdi aynı girdi ve çıktı programlamasını JApplet sınıfını kullanarak yapalım. Input (girdi) için yine JOptionPane sınıfından yararlanacağız.

Program 1.12 Hosgeldiniz1SWA.java programı

```
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 sınıfına Hosgeldiniz "+isim,25,25);
    }
}
```

Bu programı çalıştırmak için compile prosesinde yine

javac Hosgeldiniz1SWA.java komutunu kullanacağız. sonra Hosgeldiniz1SWA.html dosyasını oluşturacağız.

Program 1.13 Hosgeldiniz1SWA.html programı

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

Hosgeldiniz1SWA.html dosyasını herhangi bir browser programında görebileceğimiz gibi javanın kendi basit browser programı appletviewer da da

appletviewer Hosgeldiniz1SWA.html

deyi̇mle gösterebiliriz. Program girdi ve çıktı :

Resim 01009.JPG



Şekil 1.9 Hosgeldiniz1SWA.java dosyasının applet popup girdi penceresi

Program çıktısi appletviewer browserında görüleceği şekliyle :

Resim 01010.JPG



Şekil 1.10 Hosgeldiniz1SWA.java dosyasının applet çıktı penceresi

olacaktır. Son olarak girdi ve çıktı işlemlerini aynı pencerede yapıldığı biraz daha kompleks olan Hosgeldiniz5SWA.java programına göz atalım.

Program 1.13 Hosgeldiniz5SWA.java programı

```
import javax.swing.*; // java JApplet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıflarını (awt kütüphanesi) çağır
import java.awt.event.*; // java pencereyi dinleme sınıflarını (awt kütüphanesi) çağır

public class Hosgeldiniz5SWA extends JApplet implements ActionListener
{
    //sınıf değişkenleri
    JTextArea ciktiAlani; //JTextArea sınıfı çıktı alanı değişkeni
    JLabel kutubasligi; //JLabel sınıfı değişkeni (nesnesi) kutubasligi
    JTextField kutugirdisi;//JTextField sınıfı degiskeni (nesnesi) kutugirdisi
    String isim;

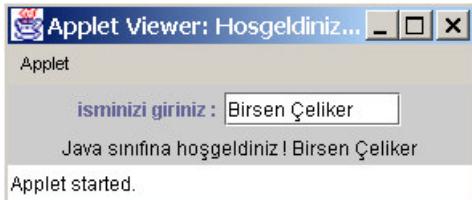
    public void init()
    {
        //girdi çıktı programlarının içine yazılacağı ana pencereyi tanımla
        Container c=getContentPane();
        c.setLayout(new FlowLayout());
        isim="Mürşide Gökağaç";
        kutubasligi=new JLabel("isminizi giriniz :");
        c.add(kutubasligi);//kutubasligini 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 sınıfına Hoşgeldiniz ! "+ isim);
        // kutuya yeni ilave edilecek komutları bekle
        // her yeni komutta actionPerformed metodunu calistir.
        kutugirdisi.addActionListener(this);
    }

    // girdi alanındaki olan olayları dinleme metodu
    public void actionPerformed(ActionEvent e)
    {
        //pencereden ismi oku
        isim=e.getActionCommand();
        ciktiAlani.setText("Java sınıfına hoşgeldiniz ! "+ isim);
        repaint();
    }
}
```

bu programı da yine
javac Hosgeldiniz5SWA.java deyimiyle compile edip,

appletviewer Hosgeldiniz5SWA.html komutuyla çalıştırırız. Hosgeldiniz5SWA.html dosyasının yukarıda anlatıldığı gibi hazırlanması gereklidir. Program çıktısı şekil 1.11 de gösterilmiştir.

Resim 01011.JPG



Şekil 1.11 Hosgeldiniz5SWA.java dosyasının applet çıktı penceresi

Bu programda verilen komut detaylarını şu anda bilmek zorunda değilsiniz. Benzer işlemleri yapmak için aynı formattan yararlanabilirsiniz. İlerki bölümlerde grafik programlarının alt metodları detaylı olarak incelenecaktır.

1.3 JAVA PROGRAMLAMA DİLİ TEMEL DEĞİŞKEN TÜRLERİ

Programlama dillerinde rakamlar bilgisayar belleğinin temel depolama birimlerine yazılırlar. Temel bilgisayar bellek birimi bit olarak adlandırılır. Bir bilgisayar belleğindeki tek bir transistörden oluşmuştur. Bu transistörden akım geçiyorsa tansistorun veya bitin bellek değeri 1 (bir) veya true (doğru) olarak alınır. Eğer akım geçmiyor veya düşük düzeyde bir akım geçiyorsa transistörün bellek değeri 0 (sıfır) veya false(yanlış) olarak alınır. Bilgisayar bit birimleri bir araya gelerek bilgisayar temel değişken türlerini yazabileceğimiz bir sistem oluştururlar. Değişken türü bilgisayar tarafından bilinmelidir, çünkü aynı bit topluluğu bir harfi simgeleyebileceği gibi bir rakamı da simgeleyebilir. Java dilinde Tablo 1.1 de görülen temel değişken türleri mevcuttur.

Tablo 1.1 Java Temel Değişken türleri

Değişken Türü	Türkçe karşılığı	Bit büyüklüğü	Sınır Değerleri
boolean	Mantık değişkeni	1	true(doğru) , false(yanlış)
char	harf değişkeni	16	"\u0000' den "\uFFFF"
byte	tam sayı değişkeni	8	-128 den 127 e
short	tam sayı değişkeni	16	-32768 den 32767 e
int	tam sayı değişkeni	32	-2157483648 den 2147483647 e
long	tam sayı değişkeni	64	-9223372036854775808 den 9223372036854775808 e
float	Gerçek sayı değişkeni	32	-3.40292347e+38 den 3.40292347e+38 e
double	Gerçek sayı değişkeni	64	-1.7976931348623157e+308 den 1.7976931348623157e+308 e

Şimdi bu temel değişken türlerinin bazılarını daha detaylı inceleyelim:

1.3.1 boolean değişken türü

boolean değişken türü mantık işlemlerinde kullanılır. Sadece true veya false değerleri alır. true doğru false yanlış anlamı taşır. Mantık değişkenlerine doğrudan true veya false değerleri yüklenebileceği gibi diğer değişkenleri mantık işlemlerini kullanarak karşılaştırarakta değerleri programların içinde hesaplanabilir.

Aşağıdaki küçük program parçası mantık değişkeni ilk mantık değişkenini tanımlamakta ve doğru değerini bu değişkene yüklemektedir.

```
boolean ilkmantikdegiskeni;  
ilkmantikdegiskeni=true;
```

1.3.2 char (harf) değişken türü

char değişken türü harflerin tanımlanmasında kullanılır. Harfler java dilinde ISO Unicode kodu ile bilgisayara aktarılır. Unicode 4 hekzagonal (16 tabanlı) sayının bir araya gelmesiyle oluşur. Hekzagonal sayı sisteminin onlu ve ikili sayı sistemiyle eşitliği söyledir :

Tablo 1.2 Heksagonal(onaltılı), onlu ve ikili sayı sistemleri eşitlikleri

Heksagonal	Onlu	İkili
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 tanımlanan '\u0041' kodu 'A' harfi anlamına gelir. Veya '\u03E1' kodu '\u03E1' harfini tanımlar. ISO Unicode karakter setinin ilk iki rakamı sıfır olduğunda ASCII karakter kodunu tanımlar. Unicode hakkında daha ayrıntılı bilgi için

<http://unicode.org>

internet adresini ziyaret edebilirsiniz. Aşağıdaki küçük program parçası **char** tipi A1, A2 ve alpha1,alpha2 değişkenlerine A ve \u03E1 harflerini yüklemektedir.

```
char A1,A2;
char alpha1,alpha2;
A1='\u0041';
A2='A';
alpha1='\u03E1';
alpha2='\u03E1';
```

Burada şunu kaydedelim alpha1 değişkenini her ortamda doğru olarak görebiliriz, fakat alpha2 değişkeni bazı ortamlarda doğru olarak görülmeyebilir.

1.3.3 Tam sayı değişken türleri (byte, short,int,long)

Tam sayı değişkenler hafızada işgal ettikleri yere göre byte(8 bit), short(16 bit), int(32 bit) ve long(64 bit) adını alırlar. Bir bitlik hafızaya sadece iki rakamın (0 veya 1) yazılabileceği göz önüne alınırsa örneğin sekiz bitlik byte türü tamsayı değişkenine 256 sayı (ikili sayı eşi 11111111) yazılabileceği ortaya çıkar. Bitlerden biri + veya - işaretini için kullanıldığından byte değişkeninin sınır değerleri -128den 127 e kadardır (toplam 256 sayı). Eğer bir tamsayı değişkenin sadece artı değerlerini kullanmak istersek unsigned terimini kullanırız. Örneğin **unsigned byte** tipi tamsayı değişkenin sınır değerleri 0 dan 256 ya kadardır. Tamsayı değişken türleri içinde en fazla kullanılan int türüdür. Aşağıdaki küçük program parçasında int türü **ilkTamsayı** değişkenine 32 rakamı yüklenecektir.

```
int ilkTamsayı;
ilkTamsayı=32;
```

bu iki satırlık program parçası tek bir satır olarak

```
int ilkTamsayı=32;
```

şeklinde de yazılabildi.

Javadaki tamsayı değişken türleri + ve – değeri alabilen türlerdir. Örneğin byte değişken türü -128 den +127 e kadar toplam 256 sayı kullanabilir. Eğer – bölgeyi kullanmayacaksa tamsayı değişkenlerin kullanım bölgesini **unsigned** deyimini kullanarak tamamen artı bölgeye çekebiliriz. Örneğin **unsigned byte artibolgetamsayisi;** işlemiyle tanımlanan **artibolgetamsayisi** değişkeni 0 ile 256 arasında değerler alabilir.

1.3.4 Gerçek sayı değişken türleri (float, double,long double)

Gerçek sayı sistemleri de 0 ve 1 bitleri kullanılarak oluşturulabilir. Gerçek sayıların tamsayıdan küçük kısımları 2 li tabanda eksi üstler kabul edilerek oluşturulur. Örnek olarak 16 bitlik gerçek bir sayıyı ikili sistemdeki bilgisayar belleğinde şu şekilde gösterebiliriz :

Tablo 1.3 gerçek sayı 22.625e-17 yazılmasında kullanılan onlu ve ikili sayıların bellekte gerçek sayı organizasyonu örneği

+/-	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	+/-	2^4	2^3	2^2	2^1	2^0
+	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	1	0	0	1

Yukarıda 16 bitle temsil edilen sayı : $+16+4+2+0.5+0.125)*10^{-(16+1)} = 22.625e-17$ rakamıdır. Gerçek sayı değişkende yeterli hassasiyeti sağlayabilmek için genelde 64 bit uzunluğundaki **double** değişken türü kullanılır. Javadaki Matematik kütüphaneleri de double değişken türü için tanımlanmıştır. Aşağıdaki küçük program parçacığında **double** türü **ilkgercekdegisken** değişkenine 22.625e-7 sayısını yükliyoruz.

```
double ilkgercekdegisken;
ilkgerceldegisken=22.625e-17;
```

eğer 64 bit double sayı hassaslığı yetmezse **long double** kullanılabilir. Bu değişken türü 128 bit boyutundadır.

1.4 JAVA NESNESİ OLARAK TANIMLANMIŞ TEMEL DEĞİŞKENLER

Java nesne temelli bir dildir. Javadaki tüm değişken türlerinin nesne kökenli eşdeğerleri mevcuttur. Bunun yanında temel değişken türleri arasında yer almış String gibi sadece nesne kökenli olarak tanımlanmış değişken türleri de mevcuttur. Şimdi bunların bir kısmını biraz daha detaylı inceleyelim.

1.4.1 String nesne tipi değişkeni

String değişkeni yazı yazdırma işleri için kullanılır. Nesne türü değişkendir. Java dilinde char değişken türü kullanılarak tanımlanmış bir nesne tipi değişkendir. Aşağıdaki küçük program parçacığında bu değişken türünün kullanılmasını görüyoruz.

```
String a="ali";
String c="veli";
String d;
d=a+b; // d nin değeri "ali veli"
System.out.println(d); // bu satır ali veli çıktısı verir
```

String türü sabitler her zaman " işaretleri arasında yazılırlar ve + işaretiley bir araya getirilebilirler.

String türü değişkenleri tanımlamak için aynı zamanda

```
String a=new String("ali");
String c=new String("veli");
String d=new String();
d=a+b; // d nin değeri "ali veli"
System.out.println(d); // bu satır ali veli çıktısı verir
```

Formunu da kullanabiliriz. İkinci verdigimiz form işlem olarak ilk verilenin tamamen aynıdır.

String türü bir değişkenin toplam boyutu length() alt metoduyla belirlenebilir. Örneğin
String a=new String("ali");
int x=a.length();

x değişkenine 3 değerini yükleyecektir.
Stringin alt metodu equals() karşılaştırma yapar. Örneğin
String a=new String("ali");
boolean y=a.equals("ali");
boolean z=a.equals("veli");

y boolean değişkenine true değerini, z boolean değişkenine false değerini yükleyecektir.

1.4.2 Integer nesne tipi değişkeni

Integer değişkeni, string değişkeni gibi

```
Integer i;  
i=new Integer(3);  
veya  
Integer i=new Integer(3);
```

Şeklinde tanımlanabilir. Integer tanımı String değişkeni üzerinden de tanımlanabilir. Örneğin

```
String s="15";  
Integer i=new Integer(s);
```

Tanımı 15 değerini yükler. Integer değerini int değerine (temel değişken) çevirebiliriz.

```
int x;  
Integer y=new Integer(3);  
x=Integer.IntegerValue(y);
```

String değerini int değerine direkt olarak çevirmek içinse

```
String s="15";  
int x=Integer.parseInt(s);
```

kullanılır.

int tipi (temel) değişkeni String değişkenine değiştirmek için

```
int x=3;  
String s=Integer.toString(x);
```

kullanılır.

1.4.3 Double nesne tipi değişkeni

Double değişkeni kullanım olarak Integer değişkeninden bir farkı yoktur. Aynı tanımlamalar Integer yerine Double kullanılarak yapılabilir. Değişkenler

```
Double x;  
X=new Double(3.66e5);  
veya  
Double x=new Double(3.66e5);
```

Şeklinde tanımlanabilir. Double tanımı String değişkeni üzerinden de tanımlanabilir. Örneğin

```
String s="15.66e-3";  
Double x=new Double(s);
```

Tanımı 15.66×10^{-3} değerini yükler. Double değerini double değerine (temel değişken) 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 değerini double (temel değişken) değerine direkt olarak çevirmek içinse

```
String s="15.25";  
double x=Double.parseDouble(s);
```

İşlemi kullanılabilir.

double tipi (temel) değişkeni String değişkenine değiştirmek için

```
double x=3.75;  
String s=Double.toString(x);
```

kullanılır.

1.4.4 diğer nesne temelli değişken türleri

Object, Long, Float, Boolean, Character, Vector gibi diğer nesne tipi değişken türleri de mevcuttur. Object nesne tipi tüm bu değişken nesne tiplerini içinde barındıran genel bir tiptir. Tüm nesne tipi değişkenlerin bizim için temel avantajı alt metodlarını kullanarak işlemler gerçekleştireme olasılıklarıdır. Fakat bu tür değişkenler temel değişken türlerine göre daha çok hafiza yeri işgal ettiklerinden mecbur kalınmadıkça da kullanılmamalıdır. Ayrıca Yine nesne tipi BigDecimal ve BigInteger türleri de hassasiyeti kullanıcı tarafından belirlenen nesne tipi değişkenler olarak kullanılabilirler.

1.5 FINAL TERİMİ VE SABİTLER

Java dilinde değişken yerine sabit kullanmak istersek tanımın başına **final** sözcüğünü getiririz. Final olarak tanımlanan sabitlerin bir kere değerleri verildikten sonra değiştirilemez.

Örneğin

```
final double pi=3.14159;  
pi sabitini tanımlar.
```

1.6 ARİTMETİK İŞLEMLER

Javadaki temel aritmetik işlemler şunlardır :

- + toplama
- çıkarma
- * çarpma
- / bölme
- % tamsayılar için bölmede kalan (modül)

Program 1.15 deki program örneği int tamsayı kullanarak bu aritmetik işlemlerin yapılmasını göstermektedir.

Program 1.15 : Aritmetik.java konsol programı

```
import java.io.*; //java girdi çıktı sınıfını çağır  
class Aritmetik  
{  
public static void main(String args[])  
{  
int sayi1,sayi2;  
int toplama,cıkarma,carpma,bolme,kalan;  
sayi1=2;  
sayi2=3;  
toplama = sayi1+sayi12;  
cıkarma = sayi1-sayi2;  
carpma = sayi1*sayı2;
```

```

bolme=sayı1/sayı2;
kalan=sayı1 % sayı2;
System.out.println("sayı1 + sayı2 =" +toplama);
System.out.println("sayı1 - sayı2 =" +cikarma);
System.out.println("sayı1 * sayı2 =" +carpma);
System.out.println("sayı1 / sayı2 =" +bolme);
System.out.println("sayı1 % sayı2 =" +kalan);
}
}

```

Aritmetik.java konsol programı çıktısı :

```

sayı1 + sayı2 =5
sayı1 - sayı2 =-1
sayı1 * sayı2 =6
sayı1 / sayı2 =0
sayı1 % sayı2 =2

```

şeklinde olacaktır. Swing grafik kütüphanesi çıktısı pratiği yapmak amacıyla aynı programı swing JOptionPane çıktı ile yazarsak :

Program 1.16 : AritmetikSW.java konsol programı

```

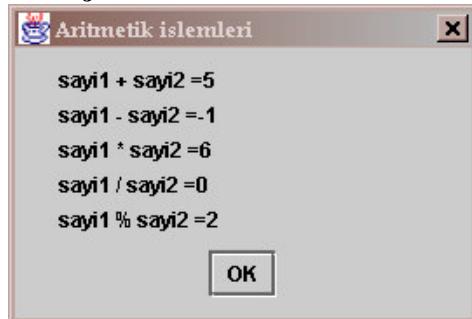
import javax.swing.JOptionPane; //java swing ciktı sinifini cagir

class AritmetikSW
{
    public static void main(String args[])
    {
        String s;
        int sayı1,sayı2;
        int toplama,cikarma,carpma,bolme,kalan;
        sayı1=Integer.parseInt(
            JOptionPane.showInputDialog("birinci sayıyı giriniz : ")
            );
        sayı2=Integer.parseInt(
            JOptionPane.showInputDialog("ikinci sayıyı giriniz : ")
            );
        toplama = sayı1+sayı2;
        cikarma = sayı1-sayı2;
        carpma = sayı1*sayı2;
        bolme = sayı1/sayı2;
        kalan = sayı1%sayı2;
        s="";
        s=s+sayı1+" + "+sayı2+" = "+toplama+"\n";
        s=s+sayı1+" - "+sayı2+" = "+cikarma+"\n";
        s=s+sayı1+" / "+sayı2+" = "+carpma+"\n";
        s=s+sayı1+" * "+sayı2+" = "+bolme+"\n";
        s=s+sayı1+" % "+sayı2+" = "+kalan+"\n";
        JOptionPane.showMessageDialog(null,s,
        "Aritmetik islemleri",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}

```

Bu programın çıktısı :

01012.JPG



Şekil 1.12 AritmetikSW.java dosyasının JOptionPane çıktı penceresinde görünümü

Java ve C dillerinde temel aritmetik işlemlerinin yanı sıra bu işlemlerin kısaltılmış sembolleri de oldukça sık kullanılır. Bu işlemler :

++ bir ilave et
-- bir çıkar
+= sağ taraftaki rakamı ilave et
-= sağ taraftaki rakamı çıkar
*= sağ taraftaki rakamla çarp
/= sağ taraftaki rakama böl
%= sağ taraftaki rakama bölüp kalanını al işlemeleridir.
++ ve -- işlemleri değişkenden önce veya sonra gelebilir. Bu iki şeitin değişken tek başlarına kullanıldığında bir farklılıklar yoktur, fakat ikinci bir değişkenle birlikte tek bir deyimde kullanıldıklarında ikinci değişkenin değeri değişir. Aşağıdaki iki program parçasına göz atalım.

++ işlemcisinin değişkenden sonra kullanılması

```
int a=2;  
int c;  
c=a++; // a nn değeri 3 c nin değeri 2
```

++ işlemcisinin değişkenden önce kullanılması

```
int a=2;  
int c;  
c=++a; // a nn değeri 3 c nin değeri 3
```

a değişkeninin değeri başlangıçta 2 ise görüldüğü gibi hem ++a işleminde hem de a++ işleminde değeri 3 e çökmaktadır. Ancak c değişkeninin değeri c=++a işleminde 2 olarak kalırken (önce değeri yükle sonra arttır), c=++a işleminde 3 olmaktadır(önce arttır, sonra değeri yükle). ++ ve -- işlemleri genelde tam sayı değişkenler için kullanılır.

Program 1.17 da aritmetik işlemleri gösteren bir program verilmiştir. Bu programın sonuçları da altta verilmektedir. Programı basamak basamak takip ederek sonuçların doğruluğunu elde hesaplayarak kontrol edebilirsiniz.

Program 1.17 : [Aritmetik1.java](#) aritmetik işlem örneği

```
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 sınıfının sonuçları

```
sayi1+= : 3  
sayi1-= : 1  
sayi1*= : 2  
sayi1/= : 1
```

1.7 JAVADA MANTIKSAL İŞLEMLER

Javadaki temel değişken türlerine bakarken boolean türü değişkenleri tanımlamıştık. Tam sayı ve gerçek değişkenlerin aritmetik işlemlerde kullanılabildiği gibi boolean değişkenler de mantıksal işlemlerde kullanılabilirler. Mantıksal işlem işlemcileri şunlardır :

Tablo 1.3 Java dilindeki mantıksal işlemler ve anlamları

Mantıksal İşlemci	Anlamı
&&	ve
	veya
>	büyüktür
<	küçüktür
==	eşittir
>=	büyük ve eşittir
<=	küçük ve eşittir
!	değildir
!=	eşit değildir

Mantıksal işlemleri pekiştirmek amacıyla Program 1.18 de bir örnek verilmiştir.

Program 1.18 : Java mantıksal işlemleri 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 );  
    }  
}
```

Aşağıda bu örnek problemin (Program 1.5) sonucunu verilmektedir. Satır satır programı takip ederek sonuçları 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
```

1.8 JAVA DİLİNDE BİT KOMUTLARI VE İŞLEMLERİ

Bit işlemi doğrudan doğruya bitler üzerinde yapılan işlemlerdir. Bit işlemleri genellikle veri trasferi (bilgisayardan bilgisayara veya çeşitli kontrol sistemlerinde) işlemlerinde kullanılır. Javadaki bit işlemleri Tablo 1.4 de verilmiştir.

Tablo 1.4 Java dilindeki bit işlemleri ve anlamları

Bit İşlemci	Anlamı
&	AND ve
	OR veya
^	XOR
~	Ters bit (NOT)
>>	Bir bit sağa kay
<<	Bir bit sola kay

Bit işleminde ne olduğuna bir örnekle bakalım.

```
int x=7;  
x=x<<1;  
bit işleminde x değişkeninin bit yapısı sola doğru bir bit kaydırılmıştır. 7 değerinin bit eşdeğeri  
00000000000000000000000000000000111  
şeklindedir. Sola doğru bir bit kaydığınıza  
000000000000000000000000000000001110  
bit yapısı olusacaktır. Bu yapının int karşılığı 14 dür.
```

```
int x=7;  
x=x<<1;  
x=-x;  
islemi  
1111111111111111111111110001
```

değerini verecektir.

Küçük bir programda bit işlemlerinin bazılarını kullanalım.

Program 1.19 : Java bit işlemleri 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 programın sonucu

```

x = 7
x = 14
x = 28
x = 7
x = -8

```

şeklinde olacaktır. Çıkan sonuçların bit eşdeğerlerini kontrol ediniz. Bit işlemleri aslında bilgisayarın ve digital elektronik devrelerin temel işlemleridir. Bu işlemler assembler dillerinin de temel işlemlerini teşkil ederler.

1.9 JAVA DİLİNİN TEMEL KOMUTLARI

Bilgisayar dillerinde değişkenlerden ve işlemlerden sonraki en temel yapı temel dil komutlarıdır. Şimdi bu komut yapılarına tek tek göz atalım.

1.9.1 if - elseif - else mantıksal karşılaştırma yapısı

Bu yapı programda karar mekanizmalarını oluşturmak için kullanılır. Temel yapısı :

```

if( boolean değişken yada sabit)
    { boolean değişken veya sabitin değeri true olduğunda yapılan işlemler}
else if( boolean değişken yada sabit)
    { boolean değişken veya sabitin değeri true olduğunda yapılan işlemler}
else
    { yukarıdaki boolean değişken vede sabitlerin hiçbir true değerine eşit değilse yapılacak olan
        işlemler}

```

else if yapısı probleme göre istenilen sayıda olabilir, if ve else yapısı sadece bir tane olarak kullanılır.

if yapısı java ve C dillerinde kısaltılmış bir şekilde de yazılabilir bu form

değişken adı = (boolean değişken yada sabit ? boolean true ise değişken değeri : boolean false ise değişken değeri);

şeklindedir. Küçük bir örnek programla if-elseif-else yapısının çalışmasını inceleyebiliriz.

Program 1.20 : if yapısını gösteren ifyapisi.java programı

```

import java.io.*; // giriş çıkış
import Text; // okuma sınıfı
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 yapısı programının çıktısı :

Ogrencinin notunu giriniz :

53

D

Bu programda öğrenci notu ekrandan girdi olarak girilmektedir. Bunu sağlayabilmek için önce

Text cin=new Text();

komutunu kullanarak ekrandan girdi girme kanalı olan cin değişkeni tanımlanmakta, sonra bu kanal üzerinden **not=cin.readDouble();**

deyi̇imle ekrandan girilen değer not değişkenine aktarılmaktadır. Okuma işlemi normalde daha detaylı tanımlar gerektirir. Bu tanımlar şimdilik Text sınıfının arkasında verilmiştir. Text sınıfının detaylı Yeri geldi̇inde okuma işlemi daha detaylı olarak incelenecektir.

not değişkeni bir ekran girdisi olduğundan girilen nota göre bilgisayar A,B,C,D,E veya F notlarından birini verecektir. Bu ilk programımızda if ve else if parantezlerinin içinde direkt olarak boolean sabit kullanılmıştır. Parantez içi boolean değişken de olabilir.

Aynı program kodunu java swing kütüphanesi JOptionPane alt sınıfını kullanarak yazacak olursak :

Program 1.21 : if yapısını gösteren ifyapisiSW.java programı

```

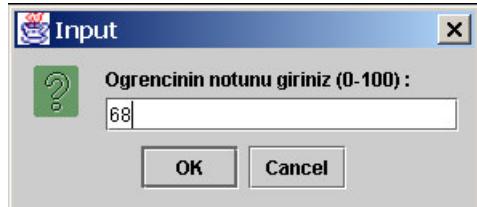
import javax.swing.JOptionPane;

class ifyapisiSW
{
    public static void main (String args[])
    {
        double not;
        //JOptionPane her zaman String turu çıktı verir. Bunu Double.parseDouble()
        //kullanarak double değerine 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 çıktısı :

01013.JPG



01014.JPG



Aşağıdaki program parçasında a degiskenenin değeri : true yazısını yazdıracaktır.

```
boolean a=true;
if(a)
{System.out.println("a degiskenenin değeri : true");}
else
{System.out.println("a degiskenenin değeri : false");}
```

if - elseif - else yapıları iç içe girmiş yapılar veya basit mantıkların **&&** (ve), **||** (veya) işlemleriyle bağlanmasıyla daha kompleks mantıksal şartlar oluşturulabilir.

İç içe geçmiş if yapısı kullanılarak mantıksal kontrol prosesi alttaki program parçasında açıklanmıştır :

İç içe geçmiş if yapısı

```
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 bağlanmış bir program parçası görüyorsunuz.

&& (ve) deyimiyle bağlanmış if yapısı

```
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.");}
```

```
}
```

Yukarıdaki program parçacıklarından da görebildiğimiz gibi her türlü kompleks mantığı bilgisayar programlarına aktarmamız mümkündür. If - else if - else yapıları programlamada en çok kullanılan mantık aktarımı yapı taşlarıdır.

1.9.2 while tekrarlama yapısı

While en çok kullanılan tekrarlama yapılarından biridir. () içindeki boolean terim true (doğru) olduğu sürece yeniden işlemi döndürür. Eğer programda boolean işlemini sonuçlandıracak bir ifade yoksa sonsuza dek veya program başka bir yöntemle durdurulana dek devam eder.

```
while( boolean değişken yada sabit)
{ boolean değişken veya sabitin değeri true olduğunda yapılan işlemler}
```

örneğin :

```
while(true)
{ System.out.println("Bu yazı sürekli olarak tekrar edecektir");}
```

yapısı sonsuz bir döngü oluşturur. Aşağıdaki program yapısı ise sayı 1000 den büyük veya 1000 e eşit olduğunda durur.

```
int sayı=2;
while(sayı <= 1000)
{ sayı *= 2; }
```

While tekrarlama yapısı bazen biraz daha farklı bir yapı içinde verilebilir. Aşağıda gösterilen bu yeni yapı, bir önceki while yapısına göre daha ender kullanılır.**do..while** yapısının while yapısından temel farkı, döngünün içine en az bir kere girilme zorunluğunu olmasıdır.

```
do
{
boolean değişken veya sabitin değeri true olduğunda yapılan işlemler.
Bu paranteze boolean değerinin ne olduğuna bakılmaksızın en az bir kere girilir
}
while( boolean değişken yada sabit)
```

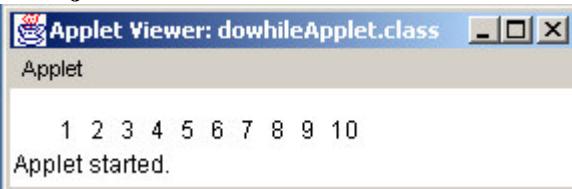
Program 1.22 de dowhile döngüsünün kullanıldığı [dowhileApplet.java](#) programı görülmektedir

Program 1.22 : do..while döngüsü örneği [dowhileApplet.java](#) programı

```
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 çalıştığında appletviewer Şekil 1.15 deki sonucu verecektir.

01015.JPG



Şekil 1.15 dowhileApplet.class dosyasının appletviewer programı kullanarak görülmesi

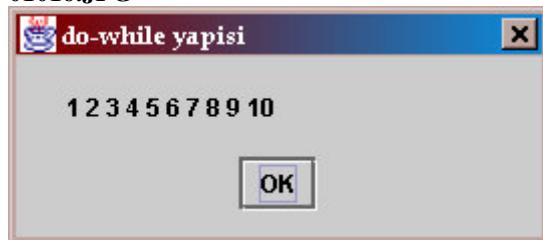
Aynı programı Swing JOptionPane çıktıları kullanarak yazalım :

Program 1.23 : do..while döngüsü örneği dowhileSW.java programı
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 yapısı",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Program çıktısı :

01016.JPG



Şekil 1.16 dowhileSW.class dosyasının JOptionPane.showMessageDialog metodu kullanarak görülmesi

1.9.3 for tekrarlama yapısı

For tekrarlama yapısı genellikle sayıları belli bir düzen içinde artırmak için kullanılır. Genel yapısı aşağıda gösterilmiştir. programlamadaki sayı saydırımıya dönük uygulamalarda en fazla kullanılan yapıdır.

for tekrarlama yapısı
for(baslangıç değerleri ; boolean değişken veya sabiti ; tekrarlama değişkeni artışı)
{ boolean değişken veya sabiti true (doğru) olduğunda yapılan işlemler }

Program 1.23 de 1 den yüze kadar sayıların toplamını hesaplayan bir konsol programı verilmiştir.

Program 1.24 : birden 100 e kadar sayıların toplamını hesaplayan foryapisi.java konsol programı

```
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 edildiğinde

1 den 100 e sayilarin toplami : 5050

sonucunu verecektir.

Program 1.25 : [faizOraniSW.java](#) *applet programı*

```

import java.applet.Applet; // java applet sınıfı
import java.awt.Graphics; // Java grafik sınıfı
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("Yıl",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 baslamalı veya string e dönüştürülmelidir.
            y+=15;
        } // for döngüsünün sonu
    } // paint metodunun sonu
} // faizOraniApplet sınıfının son

```

Program 1.24 de yine for döngüsünün uygulaması konusunda bir örnek problem verilmiştir. Programı nasıl çalıştığını inceleyiniz. Bu programda yıllık bileşik faizler on yıl için hesaplanmaktadır. İlk iki faiz oranını elde hesaplayınız. Programı çalıştırarak sonuçları kontrol ediniz. Program sonuçları Şekil 1.14 de listelenmiştir.

01017.JPG



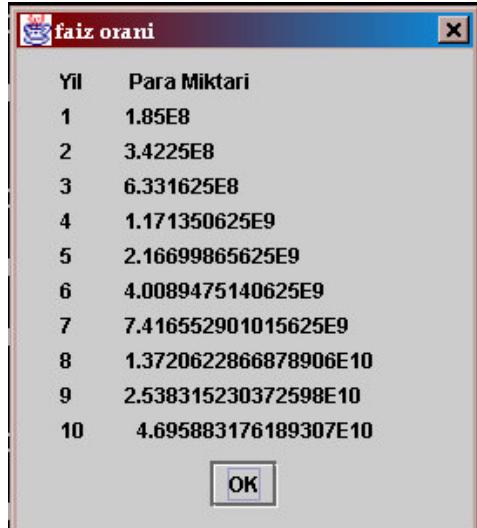
Şekil 1.17 faizOraniApplet.html appletinin appletviewer browser programında görünümü

Program 1.26 : faizOraniSW.java applet programı

```
import javax.swing.JOptionPane; //JOptionPane sınıfını çağır

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="Yıl      Para Miktarı\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 sınıfının sonu
```

01018.JPG



Şekil 1.18 faizOraniSW.java program çıktısının swing JOptionPane.showMessageDialog metodu kullanılarak görüntülenmesi

1.9.4 switch - case yapısı

Switch case yapısı çeşitli alternatifler içeren bir tablodan seçme yapma türü işlerde kullanılır. Temel yapı:

switch - case yapısı:

```
switch( değişken ismi)
{
    case değişkenin alacağı değer :
        bu değer oluştuğunda yapılacak işlemler
        break; // bu deyim döngünün bitmesini sağlar.
```

```

case değişkenin alacağı başka bir değer :
bu değer oluştugunda yapılacak işlemler
break; // bu deyim döngünün bitmesini sağlar.
.....
default :
eğer yukarıdaki case kutularından hiçbirine girilemediyse
yapılacak işlemler
break;
}

```

şeklindedir. Switch yapısından sonra değişken ismi gelir. Her case deyiminden sonra ise değişkenin alacağı yeni bir değer alternatif verilir. Eğer switch yapısında verilen alternatif hiçbir case bloğunda tanımlanmamışsa default bloğuna gider ve buradaki işlemler gerçekleştirilir. Case ve default blokları break deyimi ile sona erer. break deyimi herhangi bir döngü yapısından döngü bitmeden çıkışma amacıyla da kullanılabilir. break yapısıyla döngünün dışına çıkışma amacıyla diğer döngü yapılarında da kullanılabilir(if, for while gibi), fakat bu tür break kullanılarak döngü dışına çıkışması yapısal programlama teknikleri tarafından tavsiye edilmez. Bilgisayar listelerini yapısallıktan uzaklaştırıp, spaghetti programlama dediğimiz karmaşık programlamaya yol açabilen. Döngü yapılarından standart kontrolleri yoluyla çıkışması tavsiye edilir.

```

int i=0;
while(true)
{
    i++;
    if(i==3) break;
    System.out.println("Bu yazı uc kere tekrar edecektir");
}

```

Program 1.27 de verilen switchApplet.java programı switch deyiminin nasıl kullanıldığını göstermektedir. Programcılara göre kullanılma sıklığı değişebildiği varsayılsa bile, switch yapısı şu ana kadar bahsedilen yapılar içinde en az kullanılmıştır.

Program 1.27 switch - case yapısı örneği [switchApplet.java](#) applet programı . Bu program öğrenci notlarını appletden girer ve her nottan kaç tane girildiğini listeler.

```

import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
public class switchApplet extends Applet implements ActionListener
{
    //sınıf değişkenleri
    Label kutubasligi; //Label sınıfı değişkeni (nesnesi) kutubasligi
    TextField kutugirdisi;//Textfield sınıfı değişkeni (nesnesi) kutugirdisi
    char not;
    int Asayisi=0,Bsayisi=0,Csayisi=0,Dsayisi=0,Esayisi=0,Fsayisi=0;
    // pencereyi başlatma metodu
    public void init()
    {
        kutubasligi=new Label("Öğrencinin notunu giriniz (A B C.. : ");
        add(kutubasligi);//kutubasligini pencereye yaz
        kutugirdisi=new TextField(5);
        add(kutugirdisi);//kutuyu pencereye yerlestir
        // kutuya yeni ilave edilecek komutları bekle
        // her yeni komutta actionPerformed metotunu çalıştır.
        kutugirdisi.addActionListener(this);
    }
    public void paint(Graphics g)
    {
        g.drawString("Her harf notun toplamı : ",25,40);
        g.drawString("A : "+Asayisi,25,55);
        g.drawString("B : "+Bsayisi,25,70);
    }
}

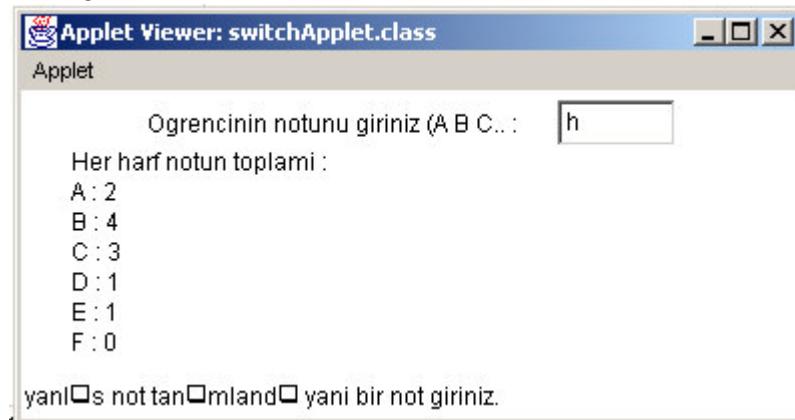
```

```

        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 olayları dinleme metodu
    public void actionPerformed(ActionEvent e)
    {
        //öğrencinin notunu pencereden oku
        String not1=e.getActionCommand();
        not=not1.charAt(0);
        showStatus(""); // sonuç bölgesindeki yazıyı 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("yanlış not tanımlandı yani bir not giriniz.");
                break;
        } // switch deyiminin sonu
        repaint(); // pencereyi yeniden paint metoduna göre çiz
    }
}

```

01019.JPG



Şekil 1.19 switch - case yapısı örneği switchApplet.html appletinin sonuçları

Program 1.28 switch - case yapısı örneği switchSWA.java swing applet programı . Bu program öğrenci notlarını appletden girer ve her nottan kaç tane girildiğini 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; // çıktıyı tanımlayacak alanı yaratır
    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 pencededen oku
        String ciktig="\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:

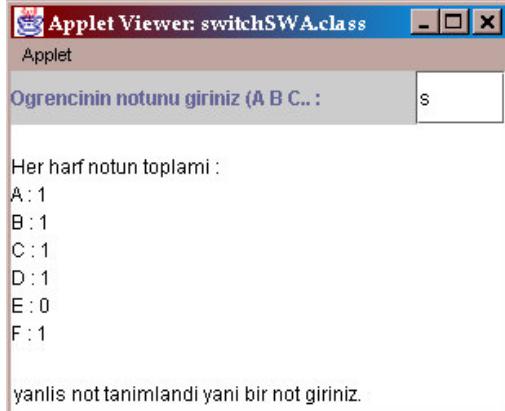
```

```

        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



Şekil 1.20 switch - case yapısı örneği switchSWA.html swing appletinin sonuçları

1.10 ARİTMETİK İŞLEMLERDE DEĞİŞKEN TÜRÜ DEĞİŞTİRME (CASTİNG) OPERASYONU

C veya C++ dilinde bir değişken türünden değişkeni diğer değişken türünden değişkene direk olarak atama mümkün değildir. örneğin

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

işlemi c++ da geçerli bir işlemidir. Javada ise bu işlem hata verir. fakat aynı işlem

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

şeklinde gerçekleştirilebilir (**int**) deyimi orijinal olarak char olarak tanımlanan a değişkeninin değerini int türüne dönüştürmiş sonra da in türünden b değişkenine yüklemiştir. Bu dönüşüm işlemlerini tüm değişken türleri için kullanabiliriz, yalnız burada programcının çok dikkatli olması ve değişken türlerini, ve sınırlarını iyi tanımması gerekmektedir. Değişken türlerini değiştirirken bazı bilgiler kaybolacağı gibi hiç istemediğimiz bir sonuca ulaşma olasılığımız da mevcuttur.

1.11 SAYI DEĞİL VE SONSUZ SONUÇLARI

Java gerçek sayı değişken türleri sayı değil (NaN) ve artı sonsuz (Pozitif infinity) ve eksiz sonsuz(negative infinity) sonuçları verebilirler. sayı değil (NaN) sonucu sıfır bölü sıfır, sıfır çarpı sonsuz gibi işlemlerden çıkar. sonsuz işlemi ise sayı bölü sıfır gibi işlemlerden çıkar. Bu işlemler sadece float ve double değişkenleri için tanımlıdır. Tamsayı değişkenler sıfıra bölünemezler.

1.12 ALIŞTIRMALAR

1. Notdefteri programını veya diğer herhangibir editörü kullanarak Hosgeldiniz.java programını incele. Konsole (MS DOS veya UNIX console) ortamında aşağıdaki komutları kullanarak çalıştır :

javac Hosgeldiniz.java
java Hosgeldiniz

Program 1.1 : Hosgeldiniz.java konsol programı

```
import java.io.*;
class Hosgeldiniz
{
public static void main (String args[])
{
    System.out.println("Java Sinifina Hos Geldiniz!")
}
}
```

2. Notdefteri programını kullanarak HosgeldinizApplet.java programını incele. Yine not defteriyle HosgeldinizApplet.html programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır:

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); }
}
```

3. Notdefteri programını kullanarak ikitamsayitopla.java programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır :

javac ikitamsayitopla.java
java ikitamsayitopla

Program 1.29 : ikitamsayitopla.java

```
import java.io.*; //giris çıkış
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 sayı giriniz : ");
        sayi1=m.readInt();
        System.out.println("Ikinci bir tam sayı girermisiniz : ");
        sayi2=m.readInt();
        toplam=sayi1+sayi2;
        System.out.println("İki tam sayının toplamı : "+toplam);
    }
}
```

5. ikitamsayitoplaSWA.java

Notdefteri programını kullanarak ikisayitoplaSWA.java ve ikisayitoplaSWA.html programını ve incele. Konsole (MS DOS) ortamında aşağıdaki komutları kullanarak çalıştır :

```
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 sayı giriniz : "));  
        sayi2=Integer.parseInt(JOptionPane.showInputDialog(" İkinci bir tam sayı giriniz : "));  
        toplam=sayi1+sayi2;  
    }  
  
    public void paint(Graphics g)  
    {  
        g.drawRect(15,10,270,20);  
        g.drawString("iki sayinin toplami : "+toplam,25,25);  
    }  
}
```

6. Notdefteri programını kullanarak **tamsayitoplaApplet.java** programını incele. Yine not defteriyle tamsayitoplaApplet.html programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır :

```
javac tamsayitoplaApplet .java  
appletviewer tamsayitoplaApplet.html
```

Program 1.31 tamsayitoplaApplet.java programı

```
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 sınıfı degiskeni (nesnesi) kutubasligi  
TextField kutugirdisi;//Textfield sınıfı degiskeni (nesnesi) kutugirdisi  
int sayi;  
int toplam;  
// pencereyi baslatma metodu  
public void init()  
{  
    kutubasligi=new Label("Bir tamsayı giriniz : ");  
    add(kutubasligi);//kutubasligini pencereye yaz  
    kutugirdisi=new TextField(10);  
    add(kutugirdisi);//kutuyu pencereye yerlestir  
    toplam=0;  
// kutuya yeni ilave edilecek komutları bekle  
// her yeni komutta actionPerformed metotunu calistir.
```

```

        kutugirdisi.addActionListener(this);
    }
    // girdi alanindaki olan olaylari dinleme metodu
    public void actionPerformed(ActionEvent e)
    {
        //sayiyi pencereden oku
        Integer sayi1=new Integer(e.getActionCommand());
        sayi=sayi1.intValue();
        // sayiyi okutma islemi asagidaki deyimle de yapabiliriz
        // sayi=Integer.parseInt( e.getActionCommand() );
        toplam+=sayi; // veya toplam=toplam+sayi;
        repaint();//pencereyi yeniden paint metotuna gore ciz
    }
    public void paint(Graphics g)
    { g.drawString("Su ana kadarki sayilarin toplami :" +toplam,25,50); }

```

01021.JPG



Sekil 1.21 tamsayitoplaApplet.html appletinin sonuçları

7. tamsayitoplaSWA.java

Notdefteri programını kullanarak tamsayitoplaSWA.java ve ikisayitoplaSWA.html programını ve incele. Konsole (MS DOS) ortamında aşağıdaki komutları kullanarak çalıştır :

```
javac tamsayitoplaSW.java
appletviewer tamsayitoplaSW.html
```

Program 1.32 tamsayitoplaSWA.java programı :

```

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

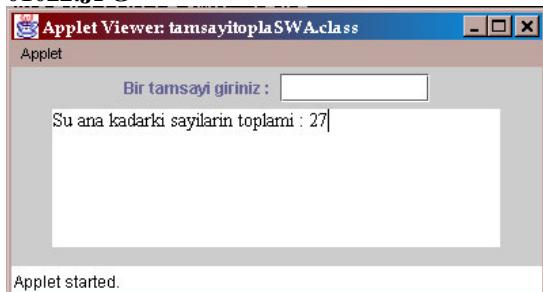
public class ikitamsayitoplaSWA extends JApplet
{
    int toplam;

    public void init()
    {
        int sayi1,sayi2;
        toplam=0;
        //
        sayi1=Integer.parseInt(JOptionPane.showInputDialog(" Bir tam sayı giriniz : "));
        sayi2=Integer.parseInt(JOptionPane.showInputDialog(" İkinci bir tam sayı 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



Şekil 1.22 tamsayitoplaSWA.html appletinin sonuçları

7. tamsayitoplaSWFrame.java

Not defteri programını kullanarak tamsayitoplaSW.java programını ve incele. Konsole (MS DOS) ortamında aşağıdaki komutları kullanarak çalıştır :

```
javac tamsayitoplaSWFrame.java  
java tamsayitoplaSWFrame
```

Program 1.33 tamsayitoplaSWFrame.java programı

```
import javax.swing.*; //java swing sınıfını çağırır  
import java.awt.*; // Java grafik sınıfını çağırır  
import java.awt.event.*; // java pencereyi dinleme sınıfını çağırır
```

```
public class tamsayitoplaSWFrame extends JFrame implements ActionListener  
{  
    //Layout container tanımı  
    //sınıf değişkenleri  
    JLabel kutubasligi; //Label sınıfı değişkeni (nesnesi) kutubasligi  
    JTextField kutugirdisi;//Textfield sınıfı değişkeni (nesnesi) kutugirdisi  
    JTextArea ciktig;  
    int sayi;  
    int toplam;  
  
    // pencereyi başlatma metodu  
  
    public tamsayitoplaSWFrame()  
    {  
        super("tam sayıların toplamı");  
        Container cerceve=getContentPane();  
        cerceve.setLayout( new FlowLayout() );  
        kutubasligi=new JLabel("Bir tamsayı giriniz : ");  
        cerceve.add(kutubasligi);//kutubasligini pencereye yaz  
        kutugirdisi=new JTextField(10);  
        cerceve.add(kutugirdisi);//kutuyu pencereye yerleştir  
        ciktig=new JTextArea(6,30);  
        ciktig.setFont(new Font("Times New Roman",Font.PLAIN,14));  
        cerceve.add(ciktig);  
        toplam=0;  
        // kutuya yeni ilave edilecek komutları bekle  
        // her yeni komutta actionPerformed metodunu çalıştır.  
        kutugirdisi.addActionListener(this);  
    }  
  
    // girdi alanındaki olan olayları 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();
}
}

```

5. Notdefteri programını kullanarak `dortArtiIslem.java` programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır :

`javac dortArtiIslem.java`

`java dortArtiIslem`

Program 1.34 `dortArtiIslem.java` programı

```

import java.io.*; //giris cikis
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 sayı giriniz : ");
        sayi1=Text.readInt(cin);
        System.out.println("Ikinci bir tam sayı 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 kalanı : " +(sayi1%sayi2));
        // Matematik tipi degisen islemleri :
        // + : topla
        // - : cikar
        // * : carp
        // / : bol
        // % : bolumun kalanı
        // += : 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 cevirma
// !=  : 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 sayı giriniz : 3
Ikinci bir tam sayı 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

6. Notdefteri programını kullanarak **gercekSayiOrtalamaApplet.java** programını incele. Yine not defteriyle **gercekSayiOrtalamaApplet.html** programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır :

javac [gercekSayiOrtalamaApplet .java](#)
appletviewer **gercekSayiOrtalamaApplet.html**

Program 1.34 : *gercekSayiOrtalamaApplet.java* programı

```

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

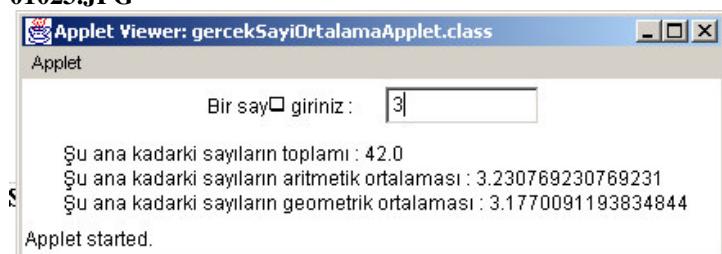
```

```

double carpim; // Gerçel degisken carpim
double aritmetikOrtalama; // Gerçel degisken aritmetik ortalama
double geometrikOrtalama; // Gerçel degisken geometrik ortalama
int sayici; // tam sayı degisken toplam rakam sayisini sayici
// pencereyi baslatma metodu
// pencere ilk basladiginda
// degiskenler buradaki degerleri alirlar
public void init()
{
    kutubasligi=new Label("Bir sayı 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 metodu
// Her yeni rakam girildiginde bu metot çağırılır
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*sayı;
    geometrikOrtalama=Math.pow(carpim,(1.0/sayici)); // carpimin koku
    kutugirdisi.setText(""); // kutudaki rakami sil
    repaint(); // pencereyi yeniden paint metoduna göre çiz
}
public void paint(Graphics g)
{
    // drawString Metotundaki eksen sistemi
    // ----- x eksen
    // |
    // |
    // |
    // V y eksen
    g.drawString("Su ana kadarki sayıların toplamı : "+toplam,25,50);
    g.drawString("Su ana kadarki sayıların aritmetik ortalaması : "+aritmetikOrtalama,25,65);
    g.drawString("Su ana kadarki sayıların geometrik ortalaması : "+geometrikOrtalama,25,80);
}
}

```

01023.JPG



Şekil 1.23 gercekSayiOrtalamaApplet.html çıktısı

7. Notdefteri programını kullanarak bessayitopla.java programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır :

```
javac bessayitopla.java
java bessayitopla
```

Program 1.35 : bessayitopla.java programı, beş gerçek sayıyı toplar.

```
import java.io.*; // giriş çıkış
import Text;      // okuma sınıfı
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 sayı giriniz : ");
            sayi=cin.readDouble();
            toplam+=sayi;
            saydirici++;
        }
        System.out.println("sayıların toplamı : "+toplam);
    }
}
```

```
Bir tam sayı giriniz : 1
Bir tam sayı giriniz : 2
Bir tam sayı giriniz : 3
Bir tam sayı giriniz : 4
Bir tam sayı giriniz : 5
sayıların toplamı : 15.0
```

8. Notdefteri programını veya herhangi bir editör programı kullanarak StringDegisen.java programını incele. Konsole (MS Dos) ortamında aşağıdaki komutları kullanarak çalıştır :

```
javac StringDegisen.java
java StringDegisen
```

Program 1.36 : StringDegisen.java programı, String değişkenlerini + operatöryle ekleme işlemi

```
import java.io.*; //giriş çıkış sınıfını çağır
class StringDegisen
{
    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

9. alistirma 7 deki [bessayi topla.java](#) programini iyice anladiktan sonra birden yüze (100) kadar sayıların ortalamasını hesaplayan [birdenyüzeOrtalama.java](#) program ve sinifini (Konsole) yaz ve calistir.

Program 1.37 : Birdenyüzeortalama.java programı, birden yüze kadar sayıların ortalamasını 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 ortalaması :" +toplam/100.0);
}
}
```

sayilarin toplami : 5050.0
sayilarin ortalaması : 50.5

10. If yapısını kullanma programı, karsilastirmaApplet.java , İki String türü değişkeni Applet TextField girdi alanından girer ve birbirıyla karşılaşır. Sonuçları Applet ortamında yazdırır. Bu program applet olarak dizayn edilmiştir. programı çalıştırınız ve sonuçlarını applet ortamında alınız.

Program 1.38 : karsilastirmaApplet.java programı

```
//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

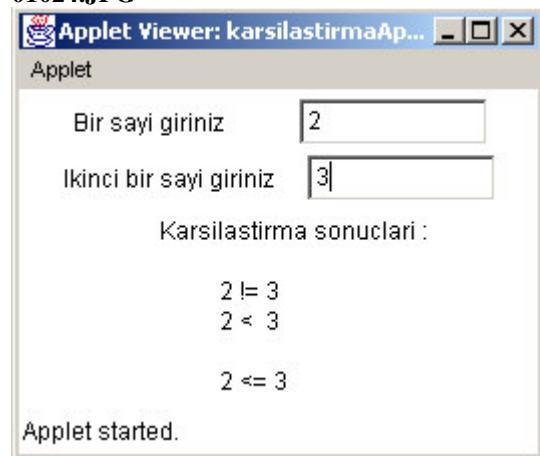
public class karsilastirmaApplet extends Applet implements ActionListener
{
Label soru1; //kullaniciya birinci degeri girme sorusu sorar
TextField girdi1; //birinci degeri girme kutusu
Label soru2; //kullaniciya ikinci degeri girme sorusunu sorar
TextField girdi2; //ikinci degeri girme kutusu
int sayi1,sayi2; //girilen iki sayi
//graphic kullanici baglantisindaki ilk degerleri tanimla
public void init()
{
soru1=new Label("Bir sayı giriniz ");
girdi1=new TextField(10);
soru2=new Label("Ikinci bir sayı giriniz");
girdi2=new TextField(10);
add(soru1); //soru1 yazisini applete cizer
```

```

add(girdi1); //girdi1 kutusunu applete cizer
add(soru2); //soru2 yazısını 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
}
// sonucları applete yazdır
public void paint(Graphics g)
{
g.drawString("Karsilaştırma sonucları : ",70,75);
if(sayı1 == sayı2)
g.drawString(sayı1+" == "+sayı2,100,90);
if(sayı1 != sayı2)
g.drawString(sayı1+" != "+sayı2,100,105);
if(sayı1 < sayı2)
g.drawString(sayı1+" < "+sayı2,100,120);
if(sayı1 == sayı2)
g.drawString(sayı1+" == "+sayı2,100,135);
if(sayı1 <= sayı2)
g.drawString(sayı1+" <= "+sayı2,100,150);
if(sayı1 = sayı2)
g.drawString(sayı1+" = "+sayı2,100,165);
}
// eger girdi(return) tusuna basıldıysa bu
// metodu çağır
public void actionPerformed( ActionEvent e)
{
// sayı 1 i girdi kutusundan oku
sayı1=Integer.parseInt(girdi1.getText());
// sayı 2 i girdi kutusundan oku
sayı2=Integer.parseInt(girdi2.getText());
repaint();
}
}

```

01024.JPG



Şekil 1.24 karsilastirmaApplet.html çıktısı

11. whileyapisi.java programını incele, çalıştır ve sonuçlarını konsol ortamında al.

Program 1.39 :[**whileyapisi.java**](#) programı. Bu program while yapısını kullanarak sayıların iki ile çarpımını hesaplar.

```

import java.io.*; //java girdi çıktı sınıfını çağır
class whileyapısı
{
    public static void main(String args[])
    {
        int sayı=2;
        while(sayı<=1000)
        {
            sayı*=2;
            System.out.println("sayı = "+sayı);
        }
    }

sayı = 4
sayı = 8
sayı = 16
sayı = 32
sayı = 64
sayı = 128
sayı = 256
sayı = 512
sayı = 1024

```

dowhileApplet.java programını incele ve çalıştır. Bu program do-while döngü yapısı kullanarak bir sayı dizisini applet ortamında yazdırmaktadır.

Program 1.40 :[dowhileApplet.java](#) programı

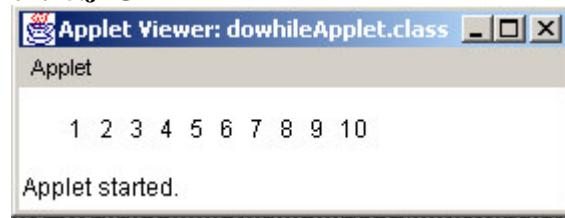
```

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

public class dowhileApplet extends Applet
{
    public void paint(Graphics g)
    {
        int saydırıcı=1;
        int x=25;
        do {
            g.drawString(Integer.toString(sayıdırıcı),x,25);
            x += 15;
        } while(++saydırıcı <= 10);
    }
}

```

01025.JPG



Şekil 1.25 dowhileApplet.html çıktısı

13. [foryapısı.java](#) programını incele ve çalıştır. Bu program for döngü yapısı kullanarak bir sayı dizisini applet ortamında yazdırmaktadır.

Program 1.41 :[foryapısı.java](#) programı

```

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 programı çıktısı :

1 den 100 e sayilarin toplami : 5050

14. [faizOraniApplet.java](#) programını incele ve çalıştır. Bu program döngü yapısı kullanarak faiz oranlarını hesaplamaktadır.

15. [switchApplet.java](#) programını incele ve çalıştır. Bu program **switch** yapısını kullanmaktadır..

16. Bir java konsol programı ([kilo.java](#)) yazarak On kişilik bir sınıfıncaki kişilerin ağırlıklarını girdi olarak giriniz.

- Ağırlıkları 40 kg dan az olanları zayıf olarak,
- Ağırlıkları 40 kilo ile 80 kilo arasında olanları normal,
- Ağırlıkları 80 kg in üstünde olanları sışman katagorisinde sınıflarsak

her sınıfda kaç kişi olduğunu yazdırınız. Dosyanızın adı **kilo.java**, sınıfınızın adı : **kilo** olacaktır. Bilmediğiniz yapıları örnek problemleri inceleyerek öğrenebilirsiniz.

Program 1.42 : [kilo.java](#) programı.

```

import java.io.*; // giriş çıkış
import Text; // okuma sınıfı
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("zayıf"); }
        else if( kilo>40 && kilo<80 )
            { System.out.println("normal"); }
        else
            { System.out.println("sıkıştırılmış"); }
    }
}

```

program çıktısı :

kilonuzu giriniz : 45
normal

kilonuzu giriniz : 95
sıkıştırılmış

17. yildiz3forApplet.java programı alta verilmiştir. bu programı inceleyiniz ve Applet çıktısını oluşturunuz.

Program 1.21 : [yildiz3forApplet.java](#) programı

```

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



Şekil 1.26 yildiz3forApplet.html çıktısı

18. yildiz1whileApplet.java programı alta verilmiştir. Bu programı inceleyiniz ve applet çıktısını oluşturunuz.

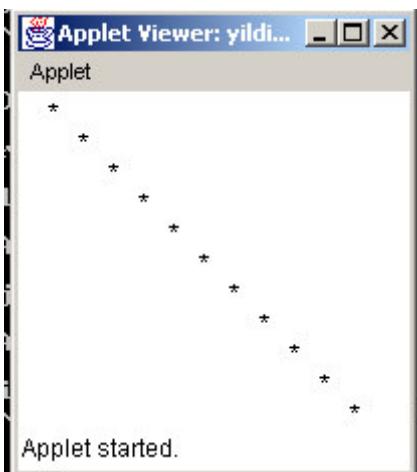
Program 1.22 : yildiz1whileApplet.java programı

```

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



Şekil 1.27 yildiz1whileApplet.html programı çıktısı

19. yildiz2forApplet.java programı altta verilmiştir. Bu programı inceleyiniz ve applet çıktısını oluşturunuz.

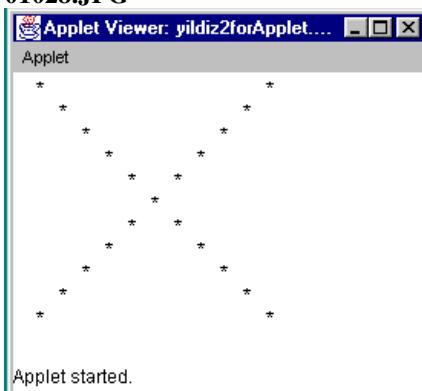
Program 1.23 : yildiz2forApplet.java

```
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



Şekil 1.28 yildiz2forApplet.html çıktısı

20. yildiz2whileApplet.java programı altta verilmiştir. Bu programı inceleyiniz ve applet çıktısını oluşturunuz.

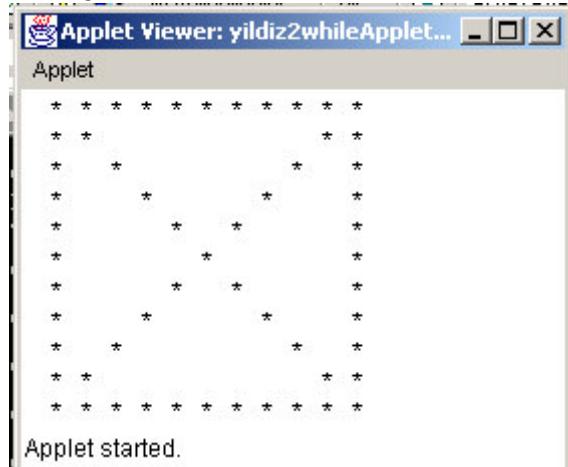
Program 1.24 : yildiz2whileApplet.java programı

```

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 degisen y 15
            g.drawString("*",15,x);      // x 15 y lineer degisen
            g.drawString("*",x,165);     // x lineer degisen y 165
            g.drawString("*",165,x);     // x 165 y lineer degisen
            g.drawString("*",x,x);       // çiz
            g.drawString("*",x,(180-x)); // çiz
        }
    }
}

```

01029.JPG



Şekil 1.29 yildiz2whileApplet.html çıktısı

21. yildiz1forApplet.java programı alta verilmiştir. Bu programı inceleyiniz ve applet çıktısını oluşturunuz.

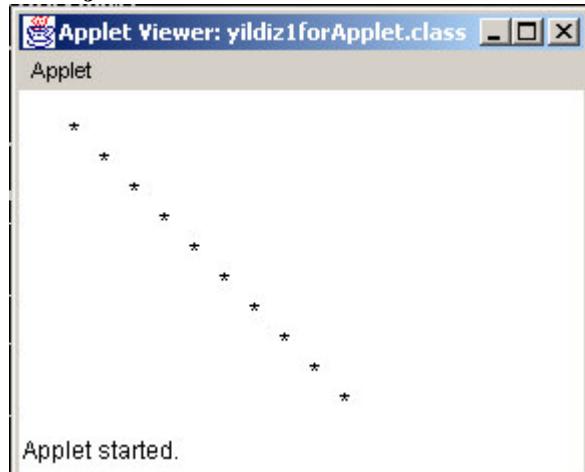
Program 1.24 yildiz1forApplet.java programı

```

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



Şekil 1.30 yildiz1forApplet.html çıktısı

22. piApplet.java programı alta verilmiştir. Bu programı inceleyiniz ve applet çıktısını oluşturunuz. Burada dizi formülü ve for döngüsünden yararlanılarak π sayısı hesaplanmaktadır. Kullanılan seri formülünün dönüşme hızının yüksek olmadığından seri 100000 terim için hesaplanmıştır. Bu serinin açık yazılımı :

$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$

şeklindedir.

Program 1.25 **piApplet.java** programı.

```
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);
    }
}
```

23. Pisagor üçlüsü : bir dik üçgende üç kenarın da tam sayı olması mümkündür. Bu tür sayı setine pisagor üçlüsü adı verilir. Pisagor üçlüsünde ilk iki sayının karelerinin toplamı üçüncü sayının karesine eşit olmalıdır. Birincidikkenar, ikincidikkenar ve hipotenüs tam sayı pisagor üçlülerini üçlü for döngüsü kulanarak bulan bir program yazınız.

24. Java applet programı kullanarak aşağıdaki şekli applet de çizdiriniz :

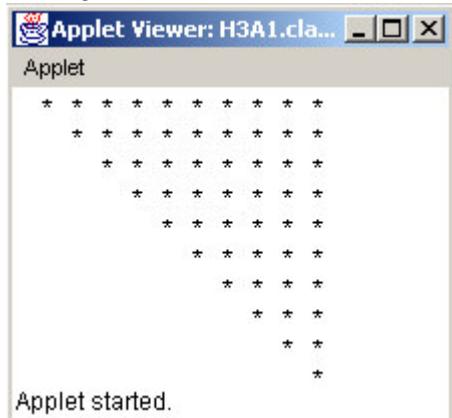
```
* * * * * * * * *
* * * * * * * *
* * * * * * *
* * * * * *
* * * * *
* * * *
* * *
```

```
* *
*
```

Program 1.26 H3A1.java programı. Bu program üçgen bir yıldızı alanı 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



Şekil 1.31 yıldız üçgeni çizen H3A1.html çıktısı

25. Java applet programı kullanarak aşağıdaki şekli appletde çizdiriniz

```
* * * * * * * * *
*           *
*           *
* * * * * * * * *
```

Program 1.27 : H3A2.java, Yıldızlardan oluşan 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);
    }
}
}

```

26. Java konsol programı kullanarak aşağıdaki şékli konsolda çizdiriniz. x eksenini ve y eksenindeki toplam yıldız sayısını ekrandan giriniz.

```

* * * * * * * * *
*           *
*           *
* * * * * * * * *

```

27. Java konsol programı **faktoriyel.java** programını oluşturunuz ve bir sayının faktoriyelini hesaplayınız.

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

Program 1.28 faktoriyel.java programı. Sayının 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 tamsayı giriniz:");
        faktoriyelsayi=cin.readInt();
        for(int sayi=1;sayi<=faktoriyelsayi;sayi++)
        { faktoriyel*=sayi;}
        System.out.println(faktoriyelsayi+" faktoriyel: "+faktoriyel);
    }
}

```

Bir tamsayı giriniz: 4

4 faktoriyel: 24

28. Java konsol programı kullanarak aşağıdaki formulü hesaplayınız:

$$e^x = 1 + x / 1! + x^2 / 2! + x^3 / 3! + x^4 / 4! + ..$$

not : e=2.7182818

Program 1.29 H3A4.java programı. 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 sayı 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("gerçek exp("+x") = "+Math.exp(x));
}
}

```

Bir sayı giriniz: 3
 $\exp(3.0) = 20.08553692318766$
 gerçek $\exp(3.0) = 20.08553692318767$

29. Aşağıda 28 deki exponent hesabını for döngüsü yerine while döngüsü kullanarak gerçekleştiriniz.

30. Hata fonksiyonu $\text{erf}(x)$ aşağıdaki formülle tanımlanabilir.

$$\text{erf}(x) = \frac{1}{\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{n!(2n+1)}$$

Bir konsol programı yazınız. Text sınıfı kullanarak dışarıdan x değerini okuyunuz. For döngüsü kullanarak formülü hesaplayınız ve sonucunu yazdırınız.

31. gerçek sayı faktoriyel fonksiyonu $x!$ 'nın 1.05 noktasındaki değeri $\approx(1.05)$ in değeri 0.9735 olarak bilinmektedir. erf fonksiyonu da faktoriyel gibi $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ ifadesiyle tanımlanıldığıne göre $\text{erf}(1.05)$ değerini hesaplayınız.

32. 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"
        // çok zevkli bir ders
        // yazdırır
        System.out.println("//Java Programlama dili\\ncok zevkli bir ders");
    }
}

```

- programını herhangi bir editör kullanarak yazınız.
- H1OD1.java adıyla kaydediniz.
- javac H1OD1.java terimiyle compile ediniz
- yanlışlarınızı düzeltiniz
- Java H1OD1 terimiyle işletiniz

33. 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"
        // çok zevkli bir ders
        // yazdırır
        g.drawString(("\"Java Programlama dili \"",25,25);
        g.drawString("cok zevkli bir ders");,25,40);
    }
}

```

- programını herhangi bir editör kullanarak yazınız.
- H1OD2.java adıyla kaydediniz.
- javac H1OD2.java terimiyle compile ediniz.
- Yanlışlarınızı düzeltiniz
- Herhangibir editor yardımıyla H1OD2.html dosyasını oluşturunuz.
- [appletviewer H1OD2.html](#) komutuyla programı çalıştırınız.
- Bir browser programında H1OD2.html dosyasını açınız

34. H1OD3.java

İki gerçek sayinin(double) (bu sayılar 2 ve 5 olsun) farklarını Konsol ortamında (MS DOS) hesaplayacak java programını yazıp işletiniz

35. H1OD4.java

İki gerçek sayinin(double) bölgümlerini Konsol ortamında (MS DOS) hesaplayacak java programını yazıp işletiniz. Sayılar ekrandan Text sınıfını kullanarak girilecektir (okunacaktır)

36. H1OD5.java

İki gerçek sayinin(double) çarpımlarını Applet(window) ortamında hesaplayacak java programını yazıp işletiniz.

37. H1OD6.java

Bes gerçek sayinin(double) çarpımlarını Konsol ortamında (MS DOS) hesaplayacak java programını yazıp işletiniz. Sayılar ekrandan Text sınıfını kullanarak girilecektir (okunacaktır)

38. H2AL1.java : if -elseif -else yapısı ve boolean değişken örneği, konsol programı. Programı incele ve çalıştır.

Program 1.32 . H2AL1.java programı boolean double ve String değişken karşılaştırma örneği

```

import java.io.*;

class H2AL1
{
    public static void main(String args[]) throws IOException
    {
        // Bu program konsolda(MS DOS)
        // if metodu 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 gerçek sayiyi karsilastir

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

```

39. H2AL2 : while döngüsü örneği applet programı, 10 dan 30 a kadar sayıları applete yazdırır. Programı incele ve çalıştır.

Program 1.33 : H2AL2.java, while döngüsü örneği applet programı, 10 dan 30 a kadar sayıları applete yazdırır.

```

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 sayıları yazdırır
        int i=0;
        double x=10.0;
        while(x<=20.0)
        {
            g.drawString(" "+(x+=1.0),i+=30,25);
        }
    }
}

```

40. H2AL3 : do - while döngüsü örneği applet programı. Programı incele ve çalıştır.

Problem 1.34 H2AL3.java : do-while döngüsü. Bu problem 10 dan 30 a kadar sayıları 2 şer 2 şer yazdırır.

```

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 sayıları ikişer ikişer yazdırır
        int i=0;
        double x=10.0;
        do
        {
            g.drawString(" "+(x+=2.0),i+=30,25);
        } while(x<=20.0)
    }
}

```

41. H2AL4 : for döngüsü örneği, applet programı . Programı incele ve çalıştır.

Problem 1.35 H2AL4.java : for döngüsü. Bu problem 10 dan 30 a kadar sayıları 3 er 3 er yazdırır.

```

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 sayıları 3 er 3 er yazdırır
        int i=0;
        for(double x=10.0;x<=20.0;x+=3.0)
        {
            g.drawString(" "+x,i+=30,25);
        }
    }
}

```

```
}
```

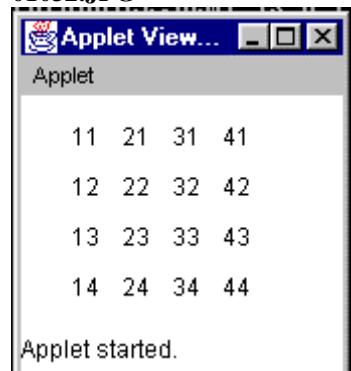
42. H2AL6: iç içe iki döngü, applet programı . Programı incele ve çalıştır.

Problem 1.36 H2AL6.java : for döngüsü. Bu problem iki applet ortamında boyutlu bir matrix yazdırır.

```
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 yazdırır
        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 ...döngüsünün kapanışı
        } //for(int i ..döngüsünün kapanışı
    } //paint metodunun kapanışı
} //H2AL6 sınıfının kapanışı
```

01032.JPG



Şekil 1.32 rakam matrisi çizen H2AL6.html çıktısı

43. H2AL7: sayı1 den sayı2 e n er ner sayilarin toplami, applet programı

Problem 1.37 H2AL7.java : for döngüsü. Bu problem sayı1 den sayı2 e n er ner sayilarin toplamını 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; //yazı alanları sınıfı
    TextField t1,t2,t3; //girdi alanları (kutuları)
```

```

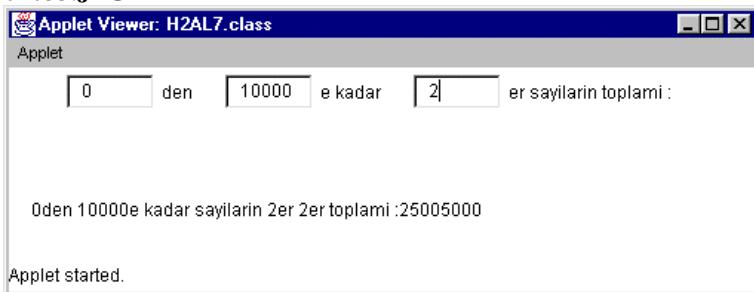
//pencereyi baslatma metodu
public void init()
{
    s1=new Label("den ");
    s2=new Label("e kadar ");
    s3=new Label("er sayilarin toplami : ");
    t1=new TextField(5);
    t2=new TextField(5);
    t3=new TextField(5);
    //pencereye ekle
    add(t1);
    add(s1);
    add(t2);
    add(s2);
    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 kutularidan 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



Şekil 1.33 değişken sınırlarla toplama yapan H2AL7.java programının applet çıktısı

44. [H2OD1](#)

while döngüsü ve konsol programı kullanarak 1 den n e kadar olan sayıların aritmetik ortalamasını hesaplayınız.

n sayisi konsoldan girilecektir.
Aritmetik ortalama $(1+2+3+\dots+n)/n$ formülüyle hesaplanır.

Problem 1.38 H2OD1.java : while döngüsü. Bu problem 1 den n e kadar sayilarin aritmetik ortalamasını 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);
    }
}
```

45. H2OD2

For döngüsü ve "*" karakterini kullanarak 5*4 yıldız dikdortgenini applet olarak çizdiriniz

```
* * * * *
* * * * *
* * * * *
* * * * *
```

Problem 1.39 H2OD2.java : for döngüsü. Bu problem 5*4 yıldız 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);
            }
        }
    }
}
```

46. H2OD3

do- while döngüsü kullanarak 1 den n e kadar olan sayıların küplerinin toplamını ve aritmetik ortalamasını applet programı olarak hesaplatınız. n sayısı applet kutu girdisinden girilecektir.

Problem 1.40 H2OD3.java : 1 den n e kadar sayıların küplerinin toplamını 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; //yazı alanları sınıfı
    TextField t1; //girdi alanları (kutuları)

    //pencereyi başlatma metodu
    public void init()
    {
        s1=new Label("küp ortalaması 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 küp ortalaması :" +ort,15,100);
    } //paint metodunun kapanışı

    //girdi alanından sinyal bekle ve sinyal gelince su işlemleri yap
    public void actionPerformed(ActionEvent e)
    {
        toplam=0;
        //sayi1 i kutulardan oku
        Integer Sayi1=new Integer(t1.getText());
        sayi1=Sayı1.intValue();

        int i=1;
        do
        {
            toplam+=i*i*i;
        }while(++i<=sayi1);
        ort=(double)toplam/(double)sayı1;
        repaint();
    } //ActionPerformed metodunun kapanışı
} //H2OD3 sınıfının kapanışı
```

46. [H2OD3_2000](#)

do- while döngüsü kullanarak 1 den n e kadar olan sayıların küplerinin toplamını ve aritmetik ortalamasını swing applet (JApplet) programı olarak hesaplatınız. n sayısı JApplet kutu girdisinden girilecektir.

Problem 1.40 H2OD3a_2000.java : 1 den n e kadar sayıların küplerinin toplamını swing applet olarak çizdir

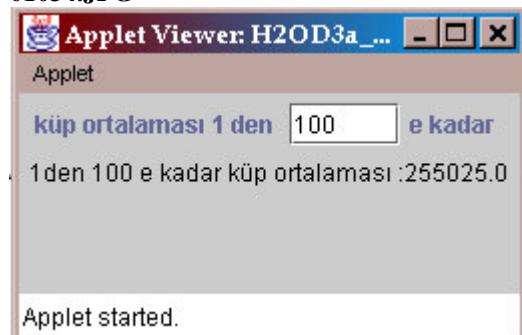
```

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; //yazı alanları sınıfı
    JTextField t1; //girdi alanları (kutuları)
    JTextArea ciktiAlani;
    //pencereyi başlatma metodu
    public void init()
    {
        Container c=getContentPane();
        c.setLayout(new FlowLayout());
        Color renk=c.getBackground();
        s1=new JLabel("küp ortalaması 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 alanından sinyal bekle ve sinyal gelince su işlemleri 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 ortalaması :" +ort);
        repaint();
    }
    //ActionPerfomed metodunun kapanışı
} //H2OD3_2000 sınıfının kapanışı

```

01034.JPG



Şekil 1.34 1 den n e kadar sayıların küplerinin ortalamasını hesaplayan H2OD3a.java programının JApplet çıktısı

H2OD3_2000 Aynı problemin ikinci bir çözümü
do- while döngüsü kullanarak 1 den n e kadar olan sayıların küplerinin toplamını ve aritmetik ortalamasını swing applet (JApplet) programı olarak hesaplatınız. n sayısı 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 sayı 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 sayıların kuplerinin toplamı : " + toplam3,25,25);
        g.drawString(n + "ye kadar olan sayıların aritmetik ortalaması : " + ortalama,25,65);
    }
}

```

47. H2OD4

herhangi bir döngü yapısı ve konsol programı kullanarak, n=200 değeri için

$1-x/1!+x^2/2!-x^3/3!+x^4/4!-x^5/5!+..+x^{n-1}/(n-1)!-x^n/n!$
formülünü hesaplayınız.

not: ! (faktoriyel) sayının birer eksilerek bire kadar çarpıldığı diziye denir. Örneğin $4!=4*3*2*1$

Problem 1.41 H2OD4.java : $1-x/1!+x^2/2!-x^3/3!+x^4/4!-x^5/5!+..+x^{n-1}/(n-1)!-x^n/n!$ formülünün hesabı.

```

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

48. H2OD5

Bir bilgisayar konsol bilgisayar programı yazınız. bu program kullanıcının ismini ve boyunu sorsun. eğer boyu 1.50 cm nin altındaysa "o **kullanıcısimi** siz pek kısaymışsınız" yazın,eğer kullanıcı boyu 1.90 cm nn üzerindeyse "o **kullanıcısimi** boyunuzda ne kadar uzunmuş" yazın. kullanıcı boyu bu değerlerin arasındaysa "merhaba kullanıcısimi sistemimize hoşgeldiniz" yazın.

Problem 1.42 H2OD5.java : if-elseif bloğu

```

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 kısaymışsınız.");}
else if(boy>=190)
{System.out.println(isim+" boyunuz da ne kadar uzunmus.");}
else
{System.out.println("merhaba "+isim+" sistemimize hoş geldiniz.");}
}
}

```

49. H2OD5_2000

Bir bilgisayar konsol bilgisayar programı yazınız. bu program kullanıcının ismini ve boyunu sorsun. eğer boyu 1.50 cm nin altındaysa "o **kullanıcısimi** siz pek kısaymışsınız" yazın,eğer kullanıcı boyu 1.90 cm nn üzerindeyse "o **kullanıcısimi** boyunuzda ne kadar uzunmuş" yazın. kullanıcı boyu bu değerlerin arasındaysa "merhaba kullanıcısimi sistemimize hoşgeldiniz" yazın. Çıktı için java swing JOPtionPane kullanınız.

Problem 1.42 H2OD5_2000.java : if-elseif bloğu

```

import javax.swing.JOptionPane;

class H2OD5_2000
{
public static void main(String args[])
{

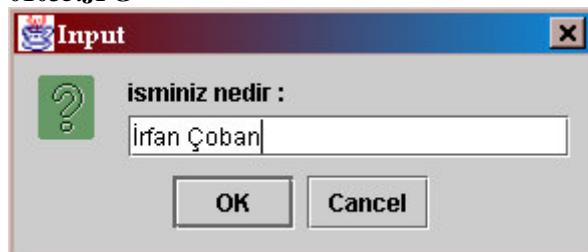
```

```

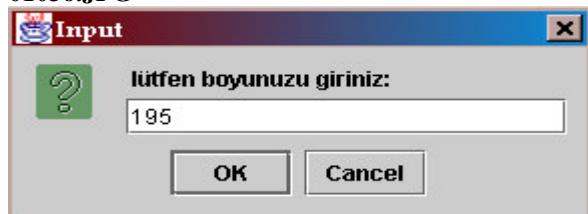
int boy;
String isim;
isim=JOptionPane.showInputDialog("isminiz nedir : ");
System.out.println(""+isim+" lutfen boyunuzu giriniz:");
boy=Integer.parseInt(
JOptionPane.showInputDialog(" lutfen boyunuzu giriniz:"));
if(boy<=150)
{
    JOptionPane.showMessageDialog(null,"sizde pek kısaymışsınız",
    "boy : "+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);
}
}

```

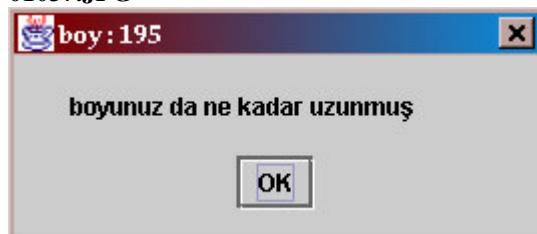
01035.JPG



01036.JPG



01037.JPG



Şekil 1.35,1.36,1.37 if yapısını test eden H2OD5_2000.java programının swing JOptionPane çıktısı

50 H2AL1_2000 : if -elseif -else yapısı ve boolean değişken örneği, JOptionPane konsol programı

Import javax.swing.JOptionPane;

```

class H2AL1_2000
{

```

```

public static void main(String args[])
{
    // Bu program konsolda(MS DOS)
    // if metodu ve boolean kullanarak
    // karşılaştırmalar yapar
    // ve sonucu yazdırır
    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 yapısının disı",JOptionPane.PLAIN_MESSAGE);
    if(b3)
    {
        JOptionPane.showMessageDialog(null,"if yapısının içi b3 = "+b3,
        "if yapısının içi",JOptionPane.PLAIN_MESSAGE);
    }
    else if(!b3)
    {
        JOptionPane.showMessageDialog(null,"else if yapısının içi b3 = "+b3,
        "if yapısının içi",JOptionPane.PLAIN_MESSAGE);
    }

    // iki gerçek sayıyı karşılaştır
    sayi1=Double.parseDouble(JOptionPane.showInputDialog("birinci gerçek sayıyı oku : "));
    sayi2=Double.parseDouble(JOptionPane.showInputDialog("ikinci gerçek sayıyı oku : "));
    b3=(sayi1>sayi2);
    if(sayi1>sayi2)
    {
        JOptionPane.showMessageDialog(null,""+sayi1+">"+sayi2+" = "+b3,
        "iki gerçek sayı karşılaştır",JOptionPane.PLAIN_MESSAGE);
    }
    else if(sayi1<sayi2)
    {
        JOptionPane.showMessageDialog(null,
        ""+sayi1+"<"+sayi2+" = +(sayi1<sayi2),
        "iki gerçek sayı karşılaştır",JOptionPane.PLAIN_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(null,""+sayi1+"=="+sayi2+" = +(sayi1==sayi2),
        "iki gerçek sayı karşılaştır",JOptionPane.PLAIN_MESSAGE);
        System.out.println();
    }

    // iki String karşılaştır

    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 gerçek sayı karşılaştır",JOptionPane.PLAIN_MESSAGE);
    }
}

```

```

else
{
JOptionPane.showMessageDialog(null,s1+" esit degildir "+s2+" b3= "+b3,
"iki gercek sayı karsilastir",JOptionPane.PLAIN_MESSAGE);
}
System.exit(0);
}//main metodunun sonu
} //H2AL1_2000 sınıfının sonu

```

51 H2AL2_2000 : while döngüsü örneği JOptionPane konsol programı

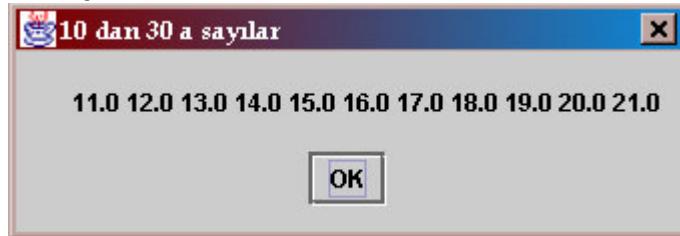
```

import javax.swing.JOptionPane;

public class H2AL2_2000
{
    public static void main(String args[])
    {
        // Bu program
        // 10 dan 20'ye kadar sayıları yazdırır
        double x=10.0;
        String s="";
        do
        {
            s=s+" "+(x+=1.0);
        }
        while(x<=20.0);
        JOptionPane.showMessageDialog(null,s,
        "10 dan 30 a sayılar",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}

```

01038.JPG



52 H2AL4_2000 : for döngüsü örneği, JOptionPane konsol programı

```

import javax.swing.JOptionPane;

public class H2AL4_2000
{
    public static void main(String args[])
    {
        // Bu program
        // 10 dan 20 ye kadar sayıları yazdırır
        String s="";
        for(double x=10.0;x<=30.0;x+=1.0)
        {
            s=s+" "+x;
        }
        JOptionPane.showMessageDialog(null,s,
        "10 dan 30 a sayılar",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}

```

53 H2AL5_2000 : switch- case yapısı, JOptionPane konsol programı

```
import javax.swing.JOptionPane;

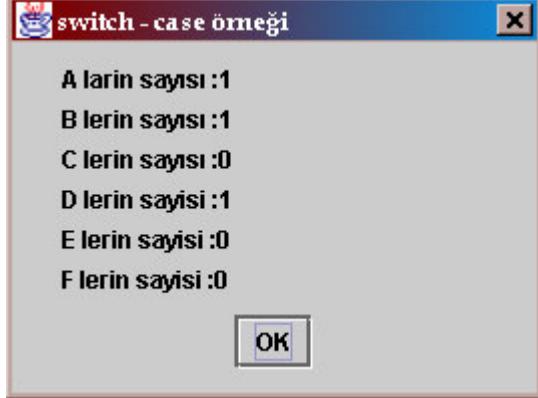
class H2AL5_2000
{
    public static void main(String args[])
    {
        // Bu program konsolda(MS DOS)
        // switch case metodunu kullanır
        char not;
        Text cin=new Text();
        int A=0;
        int B=0;
        int C=0;
        int D=0;
        int E=0;
        int F=0;
        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 örneği",JOptionPane.ERROR_MESSAGE);
                }
            }
        }
    }
}
```

```

        break;
    }
} // switch deyiminin sonu
s=s+"A larin sayısı :" +A +"\n";
s=s+"B lerin sayısı :" +B +"\n";
s=s+"C lerin sayısı :" +C +"\n";
s=s+"D lerin sayısı :" +D +"\n";
s=s+"E lerin sayısı :" +E +"\n";
s=s+"F lerin sayısı :" +F +"\n";
JOptionPane.showMessageDialog(null,s,
    "switch - case örneği",JOptionPane.PLAIN_MESSAGE);
}
}//main metodunun sonu
} //H2AL5_2000 sınıfının sonu

```

01039.JPG



01040.JPG



54 H2AL6_2000: iç içe iki döngü, JOptionPane konsol programı

```
import javax.swing.JOptionPane;
```

```

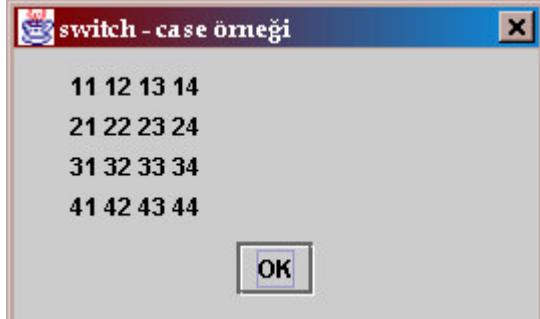
public class H2AL6_2000
{
    public static void main(String args[])
    {
        // Bu program
        // 11 12 13 14
        // 21 22 23 24
        // 31 32 33 34
        // 41 42 43 44 yazdirir
        String s="";
        for(int i=1;i<5;i++)
        {
            for(int j=1;j<5;j++)
            {
                s=s+" "+i+j;
            } // for(int j ...dongusunun kapanisi
            s=s+"\n";
        } //for(int i ..dongusunun kapanisi
    }
}
```

```

JOptionPane.showMessageDialog(null,s,
"switch - case örneği",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
} //main metodunun kapanisi
} //H2AL6_2000 sinifinin kapanisi

```

01041.JPG



55 H2AL7_2000: sayı1 den sayı2 e n er ner sayilarin toplami, swing Japplet programi, JTextArea çıktısı ile

```

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

public class H2AL7_2000 extends JApplet implements ActionListener
{ int sayı1,sayı2,sayı3;
int toplam;
double ort;
JLabel s1,s2,s3; //yazı alanları sınıfı
JTextField t1,t2,t3; //girdi alanları (kutuları)
JTextArea ciktiAlani;
//pencereyi baslatma metodu
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);
}

```

```

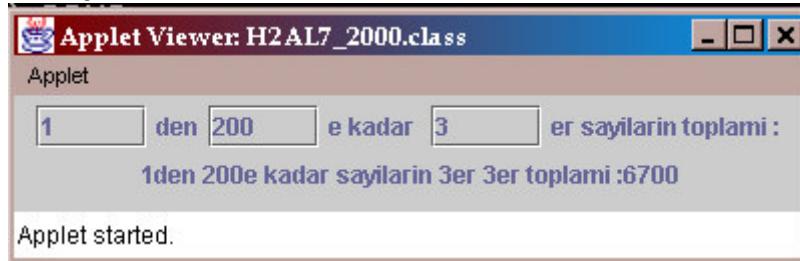
        c.add(s1);
        c.add(t2);
        c.add(s2);
        c.add(t3);
        c.add(s3);
        c.add(ciktigAlani);
        t3.addActionListener(this);

    }

//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;
    }
    ciktigAlani.setText(""+sayi1+"den "+sayi2+"e kadar sayilarin "+
    sayi3+"er "+sayi3+"er toplami :" +toplam);
    repaint();
}
//ActionPerfomed metotunun kapanisi
} //H2OD3 sinifinin kapanisi

```

01042.JPG



56 H2AL8_2000: sayının e inci kuvvetinin hesaplanması (for döngüsü örneği) JOptionPane konsol programı

```

import javax.swing.JOptionPane;

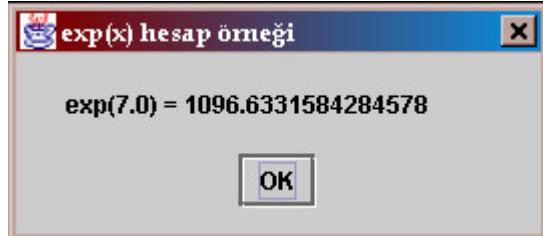
class H2AL8_2000
{
public static void main(String args[]) throws IOException
{
double faktoriyel=1;
double x;
double us=1;
double exponent=1;
x=Double.parseDouble(JOptionPane.showInputDialog("Bir sayı giriniz : "));
for(double sayı=1;sayı<=200;sayı++)
{
faktoriyel*=sayı;
us*=x;
exponent+=us/faktoriyel;
}

JOptionPane.showMessageDialog(null," $\exp(+x+)$  = "+exponent,
"exp(x) hesap örneği",JOptionPane.PLAIN_MESSAGE);

```

```
        System.exit(0);
    }
}
```

01043.JPG

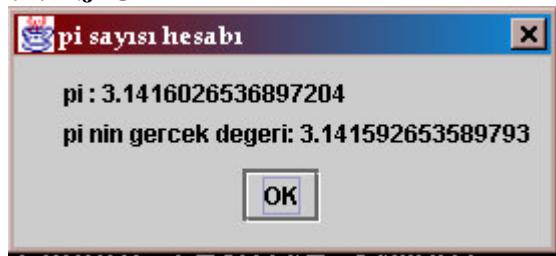


57 H2AL9_2000: pi sayısını hesaplayan bir seri formülü. JOptionPane konsol programı

```
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 gerçek değeri: "+Math.PI;
        JOptionPane.showMessageDialog(null,s,"pi sayısı hesabı",
        JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

01044.JPG



58 H2OD1_2000

for döngüsü ve konsol programı kullanarak 1 den n e kadar olan sayıların aritmetik ortalamasını hesaplayınız. n sayisi swing JOptionPane olarak girilecektir.
Aritmetik ortalama $(1+2+3+..+n)/n$ formülüyle hesaplanır.

```
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 "*" karakterini kullanarak 7*4 yıldız dikdörtgenini swing JOptionPane veya swing JApplet kullanarak çizdiriniz

```

* * * * *
* * * * *
* * * * *
* * * * *

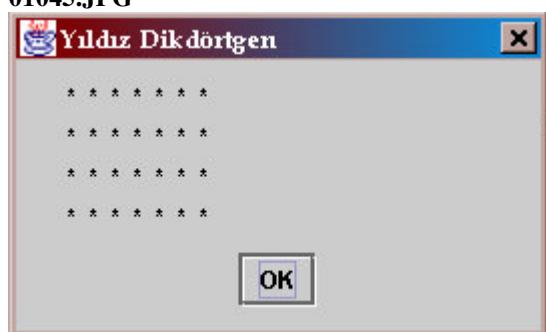
import java.io.*;
import javax.swing.JOptionPane;

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, "Yıldız Dikdörtgen"
        ,JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}

```

01045.JPG



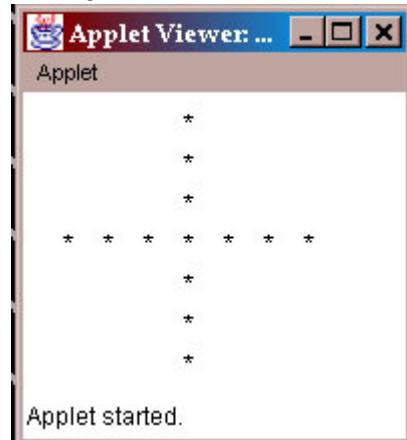
60 H2OD6_2000

for döngüsü ve "*" karakterini kullanarak grafik ortamında swing JApplet ve paint metodunu kullanarak aşağıdaki şekli 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

2.1 JAVA API KÜTÜPHANESİ

Gerçek problemlerin boyutları genelde birinci bölümde gördüğümüz problemlere göre çok daha büyüktür. Bu yüzden genelde programlar küçük parçalara bölünerek daha kolaylıkla anlaşılır şekilde getirilmeye çalışılır. Buna *böl parçala ve yönet teknigi* de diyebiliriz. Bu küçük parçalara javada metot adı verilir. Java programları yeni metotları içeren java sınıflarıyla(class) java sınıf kütüphanelerinde yer alan metotların birleşmesinden oluşur. Java kütüphanesinde matematik işlemlerinden grafik, ses, giriş çıkış işlemleri, yanlış kontrol gibi birçok konularda yazılmış metotlar mevcuttur. Java metotlar koleksiyonu **Java API** Adını alır. Java sınıf kütüphanesi adı da verilir. Şüphesiz bunun dışında java metodlar koleksiyonlarının olduğu çeşitli diğer kütüphaneler de mevcutur.

Java API kütüphanelerinin bazıları şunlardır :

java.applet : (**java applet paketi**) Java applet paketi applet grafik ortamını sağlar (grafik programlarının internet browser programları aracılığıyla görüntülenebilmesinin temellerini oluşturur.). Bu sınıfı paralel olarak Java 1.2 de grafik applet arayüz birimi **javax.swing.JApplet** geliştirilmiştir.

java.awt : (**java window araçları paketi**) Bu sınıf grafik arayüzleri tanımlamaya yara. Java 1.2 de paralel sınıf **javax.swing** geliştirilmiştir.

java.awt.color: renk tanımlar

java.awt.datatransfer: bilgisayarın geçic hafızasıyla java programı arasında bilgi akışını sağlar.

java.awt.dnd : ekle ve çalıştır türü programlar arasında geçiş yapabilen paketlerden oluşur.

java.awt.event : grafik programlama ortamıyla kullanıcı arasındaki ilişkiye sağlar. Örneğin grafik penceresinde return tuşuna basıldığında veya farenin düğmesine basıldığında ne yapılması gerektiğini belirler. **Java.awt** ve **javax.swing** sınıflarının ikisiyle de kullanılır.

java.awt.font : yazı fontları ve bu fontların değiştirilmesiyle ilgili programalrı içerir.

java.awt.geom : iki boyutlu geometrik şekilleri tanımlar.

java.awt.im : java programlarına çince, japonca ve kore dilinde girdi girebilmek için tanımlanmış bir arabirimdir.

java.awt.image : java görüntü işleme paketi

java.awt.image.renderable : bir program içindeki resimleri depolama ve değiştirme için gerekli olan programları içerir.

java.awt.print : java programlarından yazıcılara yazım için bilgi aktaran paketleri içerir.

java.beans : java fasulyeleri paketi, bu paketin detayları için ilgili bölümün inceleyiniz.

java.beans.beancontext : tekrar kullanılabilen ve otomatik olarak birleştirilebilen program parçacıkları tanımlar.

java.io : dosya ve ekranдан girdi ve çıktı yapmak için gerekli program paketleri tanımlanır.

java.lang : bu paket birçok temel program içerir Bunlardan biri olan Math paketi aşağıda açıklanacaktır. Lang paketi java programlarında otomatik olarak kullanılır. Import deyişiyle iave edilmesi gerekmek.

java.lang.ref : bu paket java programıyla garbage collector (çöp toplayıcısı) denilen otomatik kullanılmayan hafıza temizleme programıyla ilişkiye sağlar.

java.lang.reflect : bu paketteki programlar java programının çağrıldığında içinde bulunan değişken ve metodları dinamik olarak tespit etmesinive kullanma izni olmasını sağlar.

java.math : bu sınıf değişebilir hassasiyette arimetik işlemler yapmak için gereken tanımları ve sınıfları içerir.

java.net : ağ işlemlerinde kullanılan çeşitli paketleri tanımlar

java.rmi , java.rmi.activation, java.rmi.dgc, java.rmi.registry, java.rmi.server: programların ve metodların uzaktan çalışabilmeleri için gerekli tanımlamaları içerir.

java.security, java.security.acl, java.security.cert, java.security.interfaces, java.security.spec : güvenlik ile ilgili programlar demetini içerir.

java.sql : database programıyla java programlarını bağlantısını sağlar.

java.text : sayı karakter ve string tarih gibi işlemlerle ilgili programlar demeti içerir. Örneğin çok dilli programlar yazmayı sağlayan **internalisation** paketi bu paket içinde yer alır.

java.util : bu sınıf tarih ve zamanla ilgili fonksiyonlar, tesadüfi sayı üreticileri, **StringTokenizer** gibi programları barındırır.

java.util.jar bu paket java programlarını paketlemeye yarıyan jar yapılarını tanımlamada gerekli olan programları tanımlar.

java.util.zip : bu paket programları sıkıştırma yarayan programları tanımlar.

Java.accessibility : bu paket özürlülerin kullanabileceği ara üniteler tanımlar. Ekran okuyucular ve ekran büyütmeçileri gibi.

javax.swing : grafik kullanıcı arabirimini tanımlar. Bu sınıf takımları aracılığıyla ekrandan grafik ortamında veri transferi kolaylıkla yapılabilir.

javax.swing.borders : grafik arabirimini sınır yaratımında çeşitli sınır şekilleri yaratmakta kullanılır.

Javax.swing.colorchooser : renk seçimi için tanımlanmış grafik kullanıcı arabirimini tanımlar.

Java API dosyalarındaki metodların sadece isimlerinin listesi yaklaşık 200 sayfa boyutunda bir liste tuttuğundan burada sadece yeri geldiğine programları kullanırken örneklerde gerekli API isimlerini göreceğiz. Şu ana kadar kullandığımız java programlarında java.applet, java.awt, java.io javax.swing, javax.swing.JApplet ve javax.awt.event API paketlerini kullandık. Java.applet paketi appletleri oluşturma, appletlere giriş çıkışları ve dosyaları applet içerisinde çalıştırmayı sağlar. java.awt pencere(window) ortamındaki tüm programlama sınıflarını ve metodlarını içerir. Java swing (javax.swing) paketi yine aynı tür pencere(window) ortamında programlama için gerekli girdi çıktı grafik ortamlarını içerir. awt grafik ortamına göre daha zengin ve gelişmiş bir koleksiyondur. Bu metodları ilerideki bölümlerde daha detaylı inceleyeceğiz. java.io paketi javaya dışarıdan(dosya, ekran v.b) bilgi giriş çıkışını ayarlar.

Java metodlarının tamamının listesini (İngilizce olarak) Java doküman kütüphanesindeki api alt gurubunda bulabilirsiniz. Java döküman kütüphanesi <http://java.sun.com> adresinden çekilebilir.

Java dilinde en çok kullanılan API sınıflarından birisi Math sınıfıdır. Bu sınıf(class) java.lang paketinde yer alır. Java.lang paketi java programı açılırken otomatik olarak çağırılır. Bütün diğer paketler **import** deyimi kullanılarak programa ilave edilirler. Math sınıfında tanımlanan metodların bazıları Şekil 2.1 de listelenmiştir. Tablodaki x veya y değişkenleri **double** değişken türündendir. Math sınıfında(class) iki tane de sabit tanımlanmıştır. Bu sabitlerden birisi **Math.PI** 3.14159265358979323846 (pi) sayısına eşittir. Diğer **Math.E** 2.7182818284590452354 (e) sayısına eşittir.

Tablo 2.1 Math sınıfında(class) çalışan metodlardan bazıları

Metot	Tanım	Örnek
abs(x)	x değişkeninin mutlak değeri TÜRKÇE : Mutlak	Math.abs(9.2) = 9.2 Math.abs(-9.2) = 9.2
ceil(x)	x değişkenini bir üst tamsayıya dönüştür (TÜRKÇE : tavan)	Math.ceil(9.2) = 10 Math.ceil(-9.8) = -9
floor(x)	x değişkenini bir alt tamsayıya dönüştür. (TÜRKÇE : taban)	Math.floor(9.2) = 9 Math.floor(-9.8) = -10
cos(x)	x in trigonometrik cosinusü (x radyan cinsinden tanımlanmıştır.)	Math.cos(1.0)=0.54030230568
sin(x)	x in trigonometrik sinüsü (x radyan cinsinden tanımlanmıştır.)	Math.sin(1.0)=0.8414709840709
tan(x)	x in trigonometrik tanjantı (x radyan cinsinden tanımlanmıştır.)	Math.tan(1.0)=1.557040724655
acos(x)	x in trigonometrik cosinusünün tersi (sonuç radyan cinsindendir)	Math.acos(0.54030230568)=1.0
asin(x)	x in trigonometrik sinüsü (x radyan cinsinden tanımlanmıştır.)	Math.asin(0.8414709840709)=1.0
atan(x)	x in trigonometrik tanjantı (x radyan cinsinden tanımlanmıştır.)	4.0*Math.atan(1.0)=3.14159265359
atan2(x,y)	x,y noktanın x,y koordinatlarındır	4.0*Math.atan2(1.0,1.0)=3.14159265359
exp(x)	e(2.7182818284590452354) nin x	Math.exp(1.0) = 2.718281828459

	inci kuvveti	Math.exp(2.0) =7.389056098931
log(x)	e tabanına 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 olanı	Math.max(2.3,12.7)=12.7
min(x,y)	x ve y nin küçük olanı	Math.min(2.3,12.7)=2.3
random()	Raslantısal sayı (0 ile 1 arasında)	Math.random() = 0 ila bir arasında her sayı çıkabilir

Program 2.1 de Math sınıfındaki metodları kullanan [Matematik1](#) sınıfını görüyoruz.

Program 2.1 Math sınıfında(class) çalışan metodlardan bazılarını gösteren [Matematik1.java](#) programı

```
import javax.swing.JOptionPane; // giriş çıktı

class matematik1SW
{
    public static void main (String args[])
    {
        double x,y;
        String s="";
        x=Double.parseDouble(JOptionPane.showInputDialog
("Bir gerçek sayı giriniz :"));
        y=Double.parseDouble(JOptionPane.showInputDialog
("İkinci bir gerçek sayı giriniz :"));

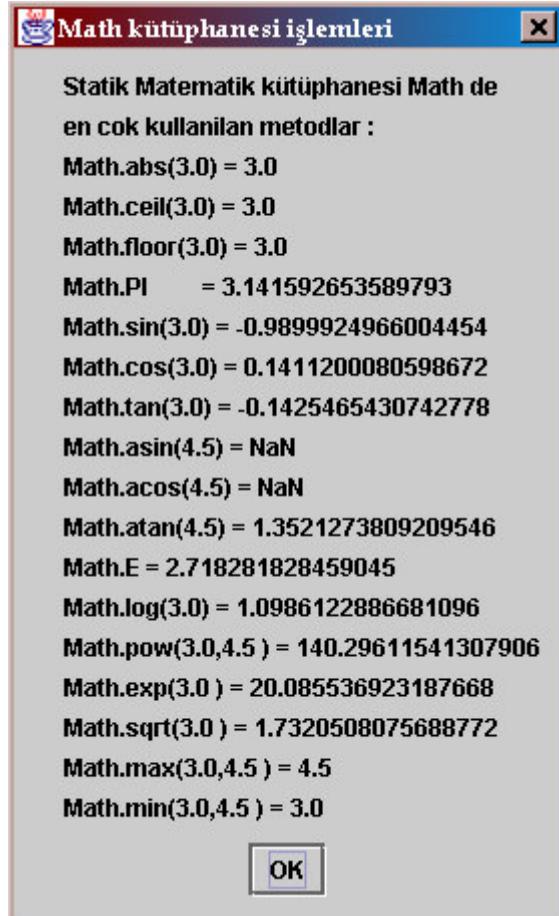
        s += "Statik Matematik kütüphanesi Math de \n";
        s+="en çok kullanılan metodlar :\n" ;
        // Math.abs(3.2)=3.2 Math.abs(-3.2)=3.2
        s+="Math.abs("+x+") = "+Math.abs(x)+"\n";
        // Math.ceil(9.2)=10 Math.ceil(-9.8)=-9
        // Math.ceil(x) x den büyük en küçük tam sayıya yuvarlar
        s+="Math.ceil("+x+") = "+Math.ceil(x)+"\n";
        // Math.floor(9.2)=9 Math.floor(-9.8)=-10
        // Math.floor(x) x den küçük en büyük tam sayıya yuvarlar
        s+="Math.floor("+x+") = "+Math.floor(x)+"\n";
        // Trigonometrik fonksiyonlar sin(x) cos(x) tan(x) x:radyan
        // Math.PI pi sayısı
        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) doğal logaritma (e) tabanına göre
        // Math.E e sayısı = 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 kökü
        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 işlemleri", JOptionPane.PLAIN_MESSAGE);
    System.exit(0);
}
}

```

02001.JPG



Şekil 2.1 matematik1SW.java programının çıktısı

2.2 METOTLAR

Metotlar java programlarının ana parçalarıdır. Metotlar sınıfların(class) içinde yer alan küçük program parçacıklarıdır. Metotların çoğu değişken parametreler metotlar ve sınıflar arasında iletişimini sağlarlar. Ayrıca her metotun kendine özgü değişkenleri de vardır. Metot yapısının ana sebebi programları modüler hale getirmektir. Aynı zamanda aynı program parçacığının tekrarlanmasılığını önlemeyi de sağlar. Her metot çağrıldığı program parçacığına belli bir değişkeni aktarabilir. Metotların tanımlarında aktardıkları değişken türü tanımlanır. Eğer metot hiçbir değişken aktarmıyorsa **void** sözcüğüyle tanımlanır. Metotların genel tanımı aşağıdaki gibidir. Parantez içindeki terimler kullanılabilir.

Genel Metot tanımı

```

(public) (static) sınıf değişken türü sınıf ismi( sınıf değişken girdi listesi)
{
    sınıf içinde geçerli değişken tanımları
    Metotun ana gövdesi
    return metot çıktı değişkeni
}

```

Örnek olarak aşağıdaki metodu 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) değişkenleri metotun girdi değişkenleridir. z ve f değişkenleri metotun yerel değişkenleridir ve bu metot dışında tanımları yoktur. Metot f değişkeninin değerini çıktı olarak metotun dışına aktarmaktadır. Eğer yukarıdaki metot aynı sınıfın içindeki başka bir metotda veya main metotunda aşağıdaki gibi bir örnekte kullanılırsa,

```
double z = 2.5;
double r ;
int i = 3;
r = metotornegi(i,z);
System.out.println("r = "+r);
```

r = 34 sonucuna ulaşılır. Şimdi aynı metotun statik metot olarak bir appletin içinde kullanılmasını görelim.

Program 2.2 : [metotornegi1.java](#) programında kare statik metotunun kullanılması

```
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 sayı giriniz : ");
sayi=cin.readDouble();
System.out.println("girilen sayinin karesi : "+kare(sayi));
}
}
```

Program 2.3 : [metotornegi2.java](#) programında kare metotunun kullanılması

```
import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır

public class metotornegi2 extends Applet
{
double sayi=5; // Gerçek değişken sayı
// 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



Şekil 2.2 [metotnegi2.java](#) programında kare metotunun kullanılması

Şekil 2.2.3 de tanımlanan **public double kare(double x)** metodu, **public void paint(Graphics g)** metotundaki
g.drawString(sayı+"nin karesi = "+kare(sayı),25,50);

işleminde çağrılmıştır. Bilgisayar **kare(sayı)** ifadesini gördüğünde metotun içine gider, orada tanımlanan tüm işlemleri yaptıktan sonra **return** kelimesinin karşısındaki işlemleri çağrıldığı yere geri ileter. Metotneği1.java programında çağrılan **kare(sayı)** metotunda sadece sayının kendisiyle çarpımı **return** kelimesinin karşısına yazıldığından, **kare(sayı)** ifadesi **sayı*sayı** ifadesiyle eşdeğerdir. Burada kare metotundandaki değişken adının **sayı** değil **x** olarak verildiğini de burada not edelim. Sayı değişkeni metotun girişinde **x** değişkenine yüklenir. Gerekli işlemler metotda yapıldıktan sonra sadece metotun sonuçları **return** deyimiyle metodun değişken türü olarak (bu metod için **double**) geri döner.

2.3 NESNE (OBJECT) TANIMI VE METOTLARDA KULLANIMI

Birinci bölümde temel değişken türlerini (**double,int,boolean** v.b.) nasıl tanımladığımızı görmüştük. Nesne tipi programmanın en önemli özelliği kendi değişken türlerimizi yaratabilmemizdir. İlerideki bölgelerde kendi değişken türlerimizi sınıf (class) yapısını kullanarak nasıl yaratacağımızı daha detaylı olarak inceleyeceğiz. Bu bölümde nesnelerin program içindeki tanımlanmasına göz atalım. Daha önce tanımladığımız bazı applet programlarında nesne tanımları zaten geçmişi. Örneğin daha önceki programlarımızda bilgi okutmak için kullandığımız

Text cin=new Text();

Terimi **Text** sınıfından(class) **cin** nesnesini tanımlar. Nesne(object) tanımı yaparken aynı metodlarda olduğu gibi değişken veya nesneleri girdi olarak tanımlamamız mümkündür.

Diğer bir örnek olarak yine daha önceki örnek problemlerde kullandığımız **Label** sınıfından(class) kutubaslığı nesnesinin tanımını ve **TextField** sınıfından(class) kutugirdisi nesnesini gösterebiliriz.

**Label kutubasligi; //Label sınıfı değişkeni (nesnesi) kutubasligi
 kutubasligi=new Label("Ogrencinin notunu giriniz (A B C.. : ");**

**TextField kutugirdisi;//Textfield sınıfı değişkeni (nesnesi) kutugirdisi
 kutugirdisi=new TextField(5);**

Bu örneklerde de görüldüğü gibi Nesne tanımı şu şekilde yapılmaktadır:

**Sınıf (class) ismunesne(object) ismi;
nesne(object) ismi = new Sınıf (class) ismi(değişken veya nesne girdi listesi)**

Nesne tanımını daha iyi anlamak için metotnegi2 sınıfını yazdığımız programı biraz değiştirerek metotnegi3 ve metotnegi3a sınıflarını oluşturalım ve kare metotumuzu metotnegi3a sınıfına yerlestirelim.

Program 2.4 : [metotnegi.java](#) programında kare metotunun kullanılması

```
import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
class metotnegi3a
{
// sayının 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 programında iki tane sınıf(class) yaratılmıştır. İlk sınıf metotornegi3a da kare metodu tanımlanmıştır. İkinci sınıf(class) metotornegi3 de ise metotornegi3a sınıfı nesne1 nesnesi tanımlanmış, ve kare metodu nesne1 nesnesi üzerinden **nesne1.kare(sayı)** olarak çağrılmıştır. Bu metot nesne1 nesnesi için bilgisayar belleğinde bir yer ayırır. Karel metotunu çağrıırken bu yerin adresini kullanır. Metotornegi2.java programında kare(sayı) doğrudan kullanılmıştı. Çünkü kare metodu aynı sınıfın(class) içinde tanımlanmıştı. Metotornegi3.java programında ise kare metodu ayrı bir sınıf(class) ta tanımlandığından yerinin tanımlanması gereklidir. Bu bizim kendi odamızın yatak odasını tanımlarken sadece yatak odası dememizle baska bir evin yatak odasını tanımlarken Ahmet beyin evinin yatak odası dememiz gibidir. Metotlar statik olarak da tanımlanabilirler. Statik olarak tanımlanan metotlar nesne kullanılmadan direk olarak sınıf(class) adları kullanılarak çağrılabılır.

Program 2.5 : [metotornegi4.java](#) programında static kare metotunun kullanılması

```

import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
class metotornegi4a
{
// sayının karesi dinamik metodu
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);
}
}

```

Gördüğü gibi metotornegi4 sınıfında(class) kare metodu metotornegi4a.kare(sayı) olarak tanımlanmış ve ek bir nesne adresi kullanılmamıştır. Static olarak tanımlanan metodların kendikendine yeterli metotlar olaması gereklidir. Yani içinde bulunduğu sınıfta veya baska sınıflarla direkt olarak veri alışverişi olamaz. Girdileri sadece girdi parantezleri () arasında verilen değişkenlerdir. Dinamik metodlar ise kendi sınıfları ve diğer sınıflarla değişik yollarla bilgi ve değişken aktarımı yapabilirler

Metotornegi3.java ve Metotornegi4.java programlarında iki sınıf(class) aynı dosyada yer almıştır. Her sınıfı ayrı bir dosyada da tanımlamak mümkün değildir. Bu yapıldığında diğer dosya import(ithal) terimiyle diğer programa aktarılır. Şekil 2.6 ve 2.7 de Şekil 2.5 de verilen programın iki ayrı dosyada yazılımını görüyoruz.

Program 2.6 : [metotornegi6.java](#) programında kare metotunun tanımlanması

```

class metotornegi6
{
// sayinin karesi dinamik metotu
public double kare(double x)
{
return x*x;
}
}

```

Program 2.7 : [metotornegi7.java](#) programında kare metotunun metotornegi6 sınıfı x nesnesi üzerinden kullanılmış.

```

import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import metotornegi6; // metotornegi6 sınıfını çağır

public class metotornegi7 extends Applet
{
double sayı=5; // Gerçek değişken sayı
metotornegi6 x=new metotornegi6();
public void paint(Graphics g)
{
g.drawString(sayı+"nin karesi = "+ x.kare(sayı),25,50);
}
}

```

2.4 METOTLARIN KENDİ KENDİNİ ÇAĞIRMASI (RECURSION)

Bazı problemlerin çözümünde bir metotun kendi kendini çağırması yararlı olabilir. Java bir metotun kendi kendini çağrımasına izin verir. Burada dikkatli olunması gereken nokta bu kendi kendine çağrıılma döngüsünün sonsuza kadar sürmesinin bir kontrol yapısı kullanılarak engellenmesidir. Kendi kendini çağrıran programlarda mutlaka bir döngü çıkış şartı tanımlanmalıdır.

Daha önceki alıştırmalarımızda faktoriyel sınıfı tanımlanmıştı. Önce bu programdaki faktoriyel hesabını ayrı metot haline dönüştürelim. Program 2.8 de daha önce de gördüğümüz faktoriyeli ana metotumuz olan main içinde hesaplayan faktoriyel programını görmekteyiz. Program 2.9 da faktoriyel hesabı for döngüsü kullanılarak faktoriyel metotonda gerçekleştirilmektedir.

Program 2.8 : Metot kullanılmadan (main metotunun içinde) faktoriyel hesaplayan faktoriyel.java programı

```

import java.io.*;
import Text;
class faktoriyel
{
public static void main(String args[]) throws IOException
{
int faktoriyel=1;
Text cin=new Text();
int faktoriyelsayı;
System.out.print("Bir tamsayı giriniz : ");
faktoriyelsayı=cin.readInt();
for(int sayı=1;sayı<=faktoriyelsayı;sayı++)
{ faktoriyel*=sayı;}
System.out.println(faktoriyelsayı + " faktoriyel: " + faktoriyel);
}
}

```

değişik giriş çıkış opsiyonlarını biraz daha anlamak için aynı programın JOptionPane kullanılarak yazılmış eşdeğerini de verelim :

Program 2.9 : Metot kullanılmadan (main metodunun içinde) faktoriyel hesaplayan faktoriyeSW.java programı

```
import javax.swing.JOptionPane;

class faktoriyelSW
{
    public static void main(String args[])
    {
        int faktoriyel=1;
        int faktoriyelsayi;
        faktoriyelsayi=Integer.parseInt(JOptionPane.showInputDialog("Bir tam sayı giriniz : "));
        for(int sayi=1;sayi<=faktoriyelsayi;sayi++)
        { faktoriyel*=sayi;}
        String s=faktoriyelsayi+" faktoriyel: "+faktoriyel;
        JOptionPane.showMessageDialog(null,s, "Metod örneği faktoriyelSW.java ",
        JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Simdi aynı programın faktoriyel metodu ayrı yazılmış fakat yine for döngüsü kullanan şekillerine göz atalım. Program 2.10 ve 2.11 de birbirinin aynı olu, program 2.10 da çıktı System.out.println deyişiyle alınırken 2.11 de çıktı JOptionPane.showMessageDialog metodu kullanılarak aktarılmıştır.

Program 2.10 : for döngülü faktoriyel metotunu kullanarak faktoriyel hesaplayan faktoriyel1.java programı

```
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 tamsayı giriniz:");
        faktoriyelsayi=cin.readInt();
        System.out.println(faktoriyelsayi+" faktoriyel:" + faktoriyel(faktoriyelsayi));
    }
}
```

Program 2.11 : for döngülü faktoriyel metotunu kullanarak faktoriyel hesaplayan faktoriyel1SW.java programı

(JOptionPane girdi çıktı sistemi kullanarak hesaplıyor)

```
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 sayı giriniz : "));
    String s=faktoriyelsayi+" faktoriyel: "+faktoriyel(faktoriyelsayi);
    JOptionPane.showMessageDialog(null,s,"Metod örneği faktoriye1SW.java
",JOptionPane.PLAIN_MESSAGE);
    System.exit(0);
}

```

Program 2.12 ve 2.13 de faktoriyeli kendi kendini çağrıran(recursive) faktoriyel metotunu kullanarak hesaplayalım.

Program 2.12 : kendi kendini çağrıran (recursive) faktoriyel metotunu kullanarak faktoriyel hesaplayan [faktoriyel2.java](#) programı

```

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 tamsayı giriniz:");
        faktoriyelsayi=Text.readInt(cin);
        System.out.println(faktoriyelsayi+" faktoriyel: " + faktoriyel(faktoriyelsayi));
    }
}

```

Bir tamsayı giriniz: 4
4 faktoriyel:24

Program 2.13 : kendi kendini çağrıran (recursive) faktoriyel metotunu kullanarak faktoriyel hesaplayan [Faktoriyel2SW.java](#) programı (JOPtionPage girdi/çıktı)

```

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 sayı giriniz : "));
    String s=faktoriyelsayi+" faktoriyel: "+faktoriyel(faktoriyelsayi);
    JOptionPane.showMessageDialog(null,s, "kendi kendini çağırın (recursive) Metod örneği faktoriyel2SW.java",
    JOptionPane.PLAIN_MESSAGE);
    System.exit(0);
}
}

```

02003.JPG



Şekil 2.3 : kendi kendini çağırın (recursive) faktoriyel metotunu kullanarak faktoriyel hesaplayan Faktoriyel2SW.java programının sonuçlarının JOptionPane çıktıları olarak görülmesi

Program 2.132 veya 2.13'ü çalıştırıldığımızda faktoriyel hesabı şu şekilde yapılacaktır :

$$5! = 5 * 4! = 4 * 3! = 3 * 2! = 2 * 1! = 1$$

faktoriyel metodu 1 e ulastığında tekrar kendisini çağrılmayıcağından geriye doğru yaptığı hesapları göndermeye başlıyacaktır.

$$1 = (2 * 1! (=1)) = 2 = (3 * 2! (=2)) = 6 = (4 * 3! (=6)) = 24 = (5 * 4! (=24)) = 120 = 5! (=120)$$

sonuç 120 olarak ana programa gönderilecektir. Kendi kendini çağırın metodlar mutlak bir gereksinim olmadıkça tercih edilmez. Bu tür programlamada hem sonsuz döngülerin görülmemesi daha güç olabilir, hemde for, while gibi döngüler kullanılarak programlamaya göre programlar daha yavaş çalışır. Gerçek uygulamalarda kendi kendini çağırın programlar genellikle bir döngü yapısı kurmadan çok tek kerelik çağrımları gerçekleştirmek için kullanılırlar.

2.5 AYNI ADLI METOTLARIN BİR ARADA KULLANILMASI (OVERLOADİNG)

Java aynı adlı metodların aynı sınıf içerisinde kullanılmasına izin verir. Aynı sınıfta(class) kullanılan metodların girdi değişkenlerinin değişken türlerinin veya değişken sayılarının farklı olması gereklidir. Java aynı isimli ve aynı değişken türlü iki metodu birbirinden ayıramaz. Örneğin

Public double Bmetodu(double Adeğerkeni)

Ve

Public double Bmetodu(double Bdeğerkeni)

Veya

Public int Bmetodu(double Cdeğerkeni)

Java tarafından ayırt edilemez , Fakat

Public double Cmetodu(double Adeğerkeni)

Ve

Public double Cmetodu(int Bdeğerkeni)

veya

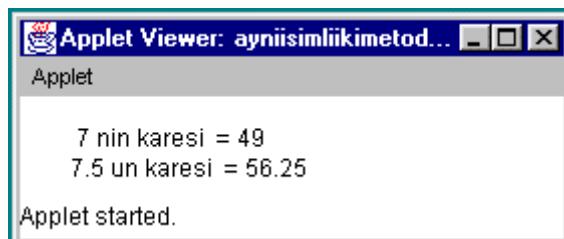
Public double Cmetodu(double Adeğerkeni, double Bdeğerkeni)

Java tarafından ayrıt edilir ve birlikte aynı sınıfın (class) içinde yer alabilir. Eğer birbirinin tam olarak eşiti metotlar değişik sınıflarda yer alıysa bunun herhangi bir sakıncası yoktur. Program 2.11'de `ayniisimlikitmetot.java` programı görülmektedir. Bu programda daha önceki programlarda da kullandığımız kare metotunu int ve double girdi değişkenleriyle iki kere aynı sınıfın içerisinde tanımlayacağız.

Program 2.14 : [ayniisimlikitmetot.java](#) programı

```
import java.awt.Graphics;
import java.applet.Applet;
public class ayniisimlikitmetot extends Applet
{
// void tipi paint metodu
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 metodu
int kare(int x)
{
return x*x;
}

// double tipi kare metodu
double kare(double x)
{
return x*x;
}
```



Şekil 2.4 ayniisimlikitmetot.html programıyla ayniisimlikitmetot.class in gösterilmesi

Bu programda parantez içerisinde 7 (tamsayı) verildiğinde java tamsayı metodu `kare(int x)`, 7.5 (gerçek sayı) verildiğinde gerçek sayı metodu `kare(double x)` çağırılır.

Benzer bir örneği swing applet olarak verelim :

Program 2.15 : H3AL9_2000 programı

```
import java.awt.Container;
import javax.swing.*;

public class H3AL9_2000 extends JApplet
{
JTextArea ciktigiciAlani;

public void init()
{
ciktigiciAlani=new JTextArea();
Container c=getContentPane();
c.add(ciktigiciAlani);
```

```

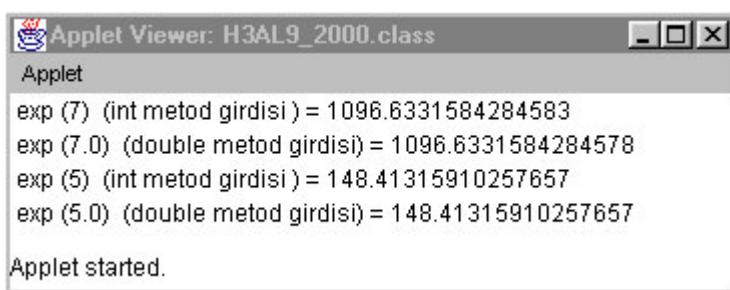
int x1=7;
double x2=7.0;
ciktiAlani.setText(" exp ("+x1+) (int metod girdisi ) = "+exp(x1)+"\n");
ciktiAlani.append(" exp ("+x2+) (double metod girdisi) = "+exp(x2)+"\n");
x1=5;
x2=5.0;
ciktiAlani.append(" exp ("+x1+) (int metod girdisi ) = "+exp(x1)+"\n");
ciktiAlani.append(" exp ("+x2+) (double metod girdisi) = "+exp(x2)+"\n");
}

// double girişli 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 girişli 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



Şekil 2.5 Gerçek ve tamsayı değişkenleri çağıran aynı adlı metodların çağırılması örneği

2.6 METOT (METHOD) VE SINIF(CLASS) DEĞİŞKENLERİ

Metotların içinde kullanılan değişkenler sadece metotlara aittir. O metotun dışında tanımları yoktur. Eğer değişkenler metotların dışında sınıf(class) değişkenleri olarak tanımlanırsa metota da aynen aktarılırlar. Sınıf değişkeniyle aynı isimde bir değişken mettotta da tanımlanmışsa bu değişken metotun içinde kullanılan metotun değişkenidir

Program 2.16 : [metotvesinifdeg.java](#) programında metot değişkenlerinin sınıfa aktarılması

```

import java.awt.Graphics;
import java.applet.Applet;
public class metotvesinifdeg extends Applet
{
//buradaki değişkenler tüm sınıfa aittir
double x ;
int y ;
void metot1( Graphics g) //metota hiç bir değişken girmiyor
{
x=5.5;
y=6;
g.drawString("metot1 in içi : 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 programının sonuçları

Metot1 in disi : x=2.5 y=3
 Metot1 in içi : x=5.5 y=6
 Metot1 in disi : x=5.5 y=6

Program 2.17 : [metotvesinifdeg1.java](#) programında metot değişkenlerinin sınıfı aktarılması

```

import java.awt.Graphics;
import java.applet.Applet;
public class metotvesinifdeg1 extends Applet
{
//buradaki değişkenler tüm sınıfa aittir
double x ;
int y ;
void metot1( Graphics g) //metota hiç bir değişken girmiyor
{
double x;
int y;
x=5.5;
y=6;
g.drawString("metot1 in içi : 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 programının sonuçları

Metot1 in disi : x=2.5 y=3
 Metot1 in içi : x=5.5 y=6
 Metot1 in disi : x=2.5 y=3

Program 2.16'daki metotvesinifdeg sınıfındaki metot1 metodunun içinde değiştirilen x ve y değişkenleri tüm sınıf(class) aittir. Bu yüzden paint metodundaki drawString metodу tekrar çağrııldığında x ve y nin değerlerinin değiştigini görürüz.

Program 2.17 deki metotvesinifdeg1 sınıfındaki metot1 metodunun içinde ise yerel değişkenler x ve y tanımlanmıştır. Bu yüzden metot1'in içindeki x ve y değişkenlerinin değişmesi paint metodundaki drawString metodу tekrar çağrııldığında x ve y nin değerlerinin değişmediği görülür.

Nesne tanımı yapıldığında Bilgisayar bu nesne için bir adres yaratır. Yeni adres yarat komutu **new** komutudur.

Pogram 2.18 : [Point1.java](#) programda nokta1 ve nokta2 nesnelerinin oluşturulması ve adres paylaşımı.

```
// java kutuphanelerinde tanımlı bir sınıfın çağırılıp
// nesne olarak tanımlanması ve adres paylaşımı
import java.io.*; //java girdi çıktı sınıfını çağır
import java.awt.Point; //Point sınıfını çağır

class Point1
{
    public static void main(String args[])
    {
        //Point sınıfı x,y koordinatlı bir nokta tanımlar
Point nokta1,nokta2;
nokta1=new Point(100,100);
nokta2=nokta1;
        // nokta1 ve nokta2 bilgisayarda aynı
        // adres kutusunu paylaşıyor
        System.out.println("nokta1 ve nokta2 aynı değerleri taşıyor");
        System.out.println("ve aynı bilgisayar adresini paylaşıyor");
        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 değeri değiştirildi");
        System.out.println("nokta 1 :" +nokta1.x+ " , "+nokta1.y);
        System.out.println("nokta 2 :" +nokta2.x+ " , "+nokta2.y);
        // nokta1 ve nokta2'nin değerleri aynı fakat bilgisayarda
        // değişik adreslerde tanımlanmış
nokta1=new Point(100,100);
nokta2=new Point(100,100);
        System.out.println("nokta1 ve nokta2 aynı değerleri taşıyor");
        System.out.println("fakat aynı bilgisayar adresini paylaşmıyor");
        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 değeri değiştirildi");
        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;
```

tanımında Point sınıfından(class) nokta1 ve nokta2 değişkenleri tanımlanmış, nokta1 değişkeni için yeni(new) adres tanımlanmış, nokta2 değişkenine de aynı adresi (nokta1'in adresini) kullanması söylemiştir. Bu yüzden programda nokta1'e yeni değer verildiğinde nokta2'nin değeri de aynı değeri alır.

daha sonra kullanılan

```
nokta1=new Point(100,100);
nokta2=new Point(100,100);
```

deyimi ile ise nokta1 ve nokta2 için iki ayrı adres yaratılmıştır. Bu yüzden nokta1 değiştiğinde nokta2 eski değerinde kalır.

Aynı adresin iki değişken tarafından kullanılması ancak gerektiği zaman ve çok dikkat sarf ederek yapılmalıdır. Gereksiz yere kullanılmasından kesin olarak kaçınılmalıdır. Program güvenlik ve kontrol problemleri yaratabildiği gibi, istemediğimiz değişken değerlerinin silinmesi sonucunu da verebilir.

2.7 ALISTIRMALAR

1. Math kütüphanesindeki Math.random() metotunu kullanarak **static zar** metotunu yazınız. Bu metotu [H4A1a](#) sınıfında(class) tanımlayınız.

Not : 1-den 6 ya kadar tesadüfi (random) sayıları $1+(int)(Math.random())*6$ 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](#) sınıfında(class) java konsol ana metotu (public void main) yazarak zar değerini yazdırınız.

Program 2.20 H4A1b.java

```
import java.io.*; // giriş çıkış
import H4A1a;
class H4A1b
{
    public static void main (String args[]) throws IOException
    {
        System.out.println("zar değeri : "+H4A1a.zar());
    }
}
```

[H4A1c](#) sınıfında java konsol ana metotunda (public void main) metotunda yüz(100) kere zar attıracak ortalamasını hesaplayınız.

Not : for veya while döngüsü kullanabilirsiniz.

Program 2.21 H4A1c.java

```
import java.io.*; // giriş çıkış
import H4A1a;
class H4A1c
{
    //yuz zarın ortalaması
    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 ortalaması = "+ortalama);
    }
}

```

H4A1d sınıfında java konsol ana metotunda (public void main) bin(1000) kere zar attırarak her rakamın kaç kere geldiğini bulunuz ve yazdırınız.

Not : switch – case deyimi bu tür programlar için idealdir

Program 2.22 H4A1d.java

```

import java.io.*; // giriş çıkış
import H4A1a;
public class H4A1d
{
//bin zar atımında her zar kaç 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 sayısı : "+zar1);
System.out.println("toplam zar 2 sayısı : "+zar2);
System.out.println("toplam zar 3 sayısı : "+zar3);
System.out.println("toplam zar 4 sayısı : "+zar4);
System.out.println("toplam zar 5 sayısı : "+zar5);

```

```
        System.out.println("toplam zar 6 sayisi : "+zar6);
    }
}
```

```
toplam zar 1 sayisi : 169
toplasm zar 2 sayisi : 165
toplasm zar 3 sayisi : 165
toplasm zar 4 sayisi : 146
toplasm zar 5 sayisi : 178
toplasm zar 6 sayisi : 177
```

2. Math kütüphanesindeki Math.random() metotunu kullanarak **static zar** metotunu yazılmıştır. Bu metot [H4A1a](#) sınıfında(class) Problem 1 de tanımlanmıştır.

[H4A1bSW](#) sınıfında(class) java konsol ana metodu (public void main) yazarak zar değerini yazdırınız. Yazdırma işlemini java swing sınıfından JOptionPane sınıfını kullanarak yapınız.

Program 2.23 H4A1bSW.java

```
import javax.swing.JOptionPane; // giriş çıkış
import H4A1a;

class H4A1bSW
{
    public static void main (String args[])
    {
        JOptionPane.showMessageDialog(null,"atılan zar = "+H4A1a.zar()
        ,"sınıf örneği, zar atımı ",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

[H4A1cSW](#) sınıfında java konsol ana metotunda (public void main) metotunda yüz(100) kere zar attırarak ortalamasını hesaplayınız. Yazdırma işlemini java swing sınıfından JOptionPane sınıfını kullanarak yapınız. Not : for veya while döngüsü kullanabilirsiniz.

Program 2.24 H4A1cSW.java

```
import javax.swing.JOptionPane; // giriş çıkış
import H4A1a;

class H4A1cSW
{
    //yüz zarın ortalaması
    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 zarın ortalaması = "+ortalama
        ,"sınıf örneği, 100 kez zar atımı ortalaması H4A1c",
        JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

}

[H4A1dSW](#) sınıfında java konsol ana metodunda (public void main) bin(1000) kere zar attırarak her rakamın kaç kere geldigini bulunuz ve yazdırınız. Yazdırma işlemini java swing sınıfından JOptionPane sınıfını kullanarak yapınız.

Not : switch – case deyimi bu tür programlar için idealdir

Program 2.25 H4A1dSW.java

```
import javax.swing.JOptionPane; // giriş çıkış
import H4A1a;

public class H4A1dSW
{
    //bin zar atımında her zar kaç 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 sayısı : "+zar1+"\n";
        s+="toplam zar 2 sayısı : "+zar2+"\n";
        s+="toplam zar 3 sayısı : "+zar3+"\n";
        s+="toplam zar 4 sayısı : "+zar4+"\n";
        s+="toplam zar 5 sayısı : "+zar5+"\n";
        s+="toplam zar 6 sayısı : "+zar6+"\n";
        JOptionPane.showMessageDialog(null,s)
```

```

        , "sınıf örneği, 1000 kez zar atımı 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](#) sınıfına yerlestir. ZarApplet.java programindaki zar metotunu sil ve import H4A2b; deyimini ekle.
- ZarApplet.java programını H4A2c olarak değiştir. sınıfından yeni nesne z yi H42c.java programına ekle ve Paint methodunda çağırılan zar metotunda gerekli değişiklikleri yap. Programı yeni sekliyle çalıştır.

Program 2.26 : ZarApplet.java, zar atma metot örneği

```

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 düğmesi
int zar1,zar2;
public static int zar()
{
return 1+(int)(Math.random()*6);
}
public void init()
{
//programı baslat
ilkzar=new TextField(10); // Textfield sınıfı ilkzar nesnesini yarat
add(ilkzar); // ilk zar nesnesini pencereye ekle
ikincizar=new TextField(10); // Textfield sınıfı ikincizar nesnesini yarat
add(ikincizar); // ikinci zar nesnesini pencereye ekle
salla=new Button("Zarı salla ve at");
add(salla);
salla.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
zar1=zar();
zar2=zar();
}

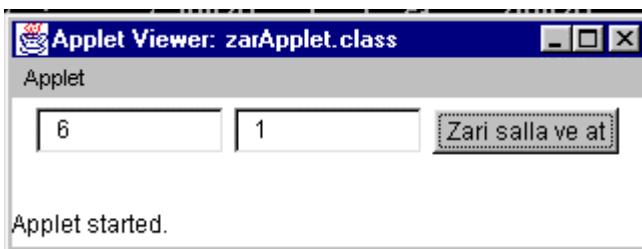
```

```

ilkzar.setText(Integer.toString(zar1));
ikincizar.setText(Integer.toString(zar2));
toplamsar+=zar1+zar2;
toplamsar++; 
repaint();
}
public void paint( Graphics g)
{
g.drawString("toplamsar = "+toplamsar+" Atılan zar sayısı = "+toplamsar,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 toplamsar=0;
int toplam=0;
TextField ilkzar,ikincizar;
Button salla; //zar atma düğmesi
int zar1,zar2;
H4A2b z=new H4A2b();
public void init()
{
//programı başlat
ilkzar=new TextField(10); // Textfield sınıfı ilkzar nesnesini yarat
add(ilkzar); // ilk zar nesnesini pencereye ekle
ikincizar=new TextField(10); // Textfield sınıfı ikincizar nesnesini yarat
add(ikincizar); // ikinci zar nesnesini pencereye ekle
salla=new Button("Zarı 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));
toplamsar+=zar1+zar2;
toplamsar++; 
repaint();
}
public void paint( Graphics g)
{
g.drawString("toplamsar = "+toplamsar+" Atılan zar sayısı = "+toplamsar,25,50);
}
}

```

02007.JPG



Şekil 2.7 ZarApplet.html çıktısı

4. faktoriyeltesti1.java programını inceleyiniz, ve applet ortamında çalıştırınız.

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);
    }
}
```

5. fibonachitesti1.java programını inceleyiniz ve applet ortamında çalıştırınız.

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("Fibonacci("+i+") = "+fibonacci(i),25,y);
            y+=15;
        }
    }

    public long fibonacci(long sayi)
    {
        if(sayi <=0) return 0;
        else if(sayi ==1) return 1;
        else return fibonacci(sayi - 1)+fibonacci(sayi - 2);
    }
}
```

```
}
```

6. fibonnachitesti1SWA.java swing applet programını inceleyiniz ve appletviewer ile çalıştırınız

Program 2.29 Fibobachi testi, swing applet programı fibonnachitesti1SWA.java

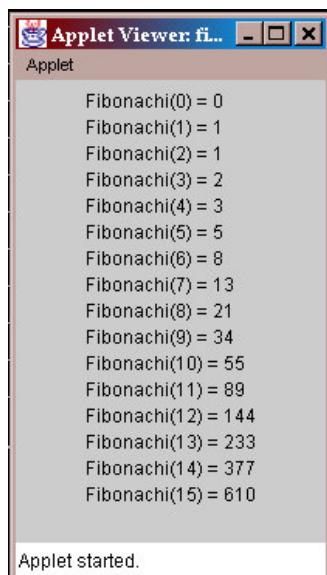
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

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+="Fibonacci("+i+") = "+fibonacci(i)+"\n";
        }
        cikti.setText(s);
    }

    public long fibonacci(long sayi)
    {
        if(sayi <=0) return 0;
        else if(sayi ==1) return 1;
        else return fibonacci(sayi - 1)+fibonacci(sayi - 2);
    }
}
```

02008.JPG



Şekil 2.8 fibonnachitesti1SWA swing applet çıktısı

7. fibonnachitesti1SWF.java swing Frame programını inceleyiniz ve java komutu ile çalıştırınız

Program 2.30 Fibobachi testi, swing applet programı fibonnachitesti1SWA.java

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

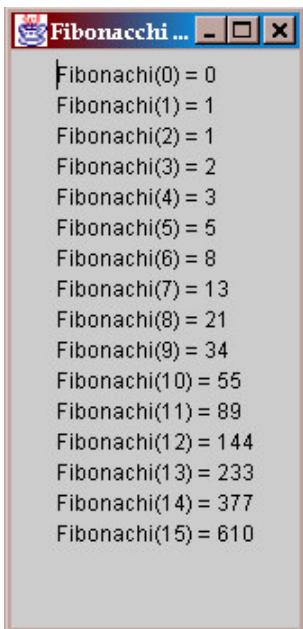
public class fibonnachitesti1SWF extends JFrame
{
    private JTextArea cikti;

    public fibonnachitesti1SWF()
    {
        super("Fibonacci 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+="Fibonacci("+i+") = "+fibonacci(i)+"\n";
        }
        cikti.setText(s);
    }

    public long fibonacci(long sayi)
    {
        if(sayi <=0) return 0;
        else if(sayi ==1) return 1;
        else return fibonacci(sayi - 1)+fibonacci(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



Şekil 2.9 fibonnachitesti1SWF çıktısı

8. **fibonachitesti2.java** programını inceleyiniz ve applet ortamında çalıştırınız.

Problem 2.31 fibonachitesti2.java programı

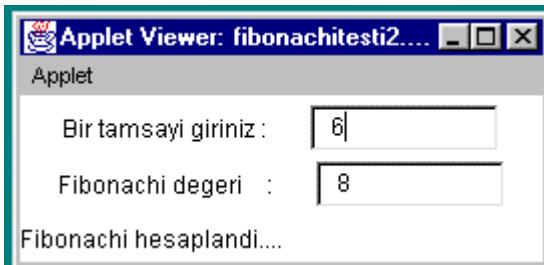
```
import java.awt.*;  
import java.applet.Applet;  
import java.awt.event.*;  
public class fibonachitesti2 extends Applet implements ActionListener  
{  
    Label rakametiketi,sonucetiketi;  
    TextField rakam,sonuc;  
    public void init()  
    {  
        rakametiketi=new Label("Bir tamsayı giriniz : ");  
        rakam=new TextField(10);  
        rakam.addActionListener(this);  
        sonucetiketi=new Label("Fibonacci değeri : ");  
        sonuc=new TextField(10);  
        sonuc.setEditable(false);  
        add(rakametiketi);  
        add(rakam);  
        add(sonucetiketi);  
        add(sonuc);  
    }  
    public void actionPerformed( ActionEvent e)  
    {  
        long sayı;  
        long fibonachisayisi;  
        sayı=Long.parseLong(rakam.getText());  
        showStatus("Fibonacci hesaplanıyor....");  
        fibonachisayisi=fibonacci(sayı);  
        showStatus("Fibonacci hesaplandı....");  
        sonuc.setText(Long.toString(fibonachisayisi));  
    }  
    public long fibonacci(long sayı)  
    {  
        if(sayı <=0) return 0;
```

```

else if(sayı ==1) return 1;
else return fibonachi(sayı - 1)+fibonachi(sayı - 2);
}
}

```

02010.JPG



Şekil 2.10 Fibonacci testi 2 çıktısı (AWT Aplet)

9. fibonachitesti2SWA.java programında bir önceki programda yazılan fibonacci testi swing applet ortamında yazılmıştır. Programı inceleyiniz ve applet ortamında çalıştırınız.

Problem 2.32 fibonachitesti2SWA.java programı

```

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 tamsayı giriniz : ");
        rakam=new JTextField(10);
        rakam.addActionListener(this);
        sonucetiketi=new JLabel("Fibonacci değeri : ");
        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 sayı;
        long fibonachisayisi;
        sayı=Long.parseLong(rakam.getText());
        showStatus("Fibonacci hesaplanıyor....");
        fibonachisayisi=fibonacci(sayı);
        showStatus("Fibonacci hesaplandı....");
        sonuc.setText(Long.toString(fibonachisayisi));
    }

    public long fibonacci(long sayı)

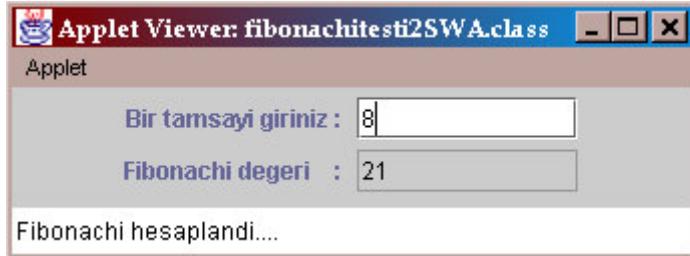
```

```

{
if(sayı <=0) return 0;
else if(sayı ==1) return 1;
else return fibonachi(sayı - 1)+fibonachi(sayı - 2);
}
}

```

02011.JPG



Şekil 2.11 Fibonacci testi 2 çıktısı (swing applet)

10. Bir kürenin hacmini hesaplayacak bir applet programı yazınız. Kürenin hacmi küreHacmi isimli metot tarafından hesaplanınsın. Applet bir girdi alanına kürenin yarıçapını sorsun ve yarıçap girildiğinde yeni küre hacmi ekranda gösterilsin.

Not : Kürenin hacmi : $(4.0/3.0)*\text{Math.PI}*\text{Math.pow}(r,3);$

Problem 2.33 : Bir kürenin hacmini hesaplayan H5A1.java programı

```

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 s̄n̄fa aittir
Label kutubasligi; //Label sinifi degiskeni (nesnesi) kutubasligi
TextField kutugirdisi;// TextField sinifi degiskeni (nesnesi) kutugirdisi
double yaricap; // Gercek degisen yar̄cap
double hacim; // Gercek degisen hacim
//=====
// pencereyi baslatma metodu
// 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 alanındaki olan olayları dinleme metodu
// Her yeni rakam girildiğinde bu metot cagrilir
public void actionPerformed(ActionEvent e)
{
Double sayı1=new Double(e.getActionCommand());
yaricap=sayı1.doubleValue();
hacim=kureninhacmi(yaricap);
}

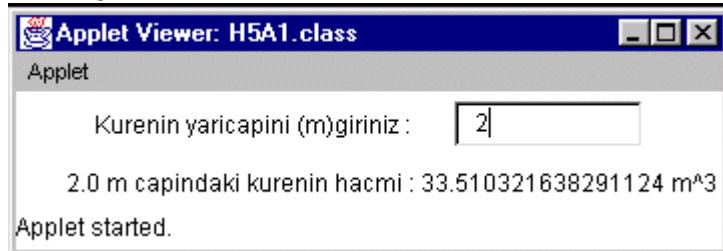
```

```

kutugirdisi.setText(" ");//kutudaki rakami sil
repaint();//pencereyi yeniden paint metotuna gore ciz
}
//=====
// paint (boya=pencereye ciz metodu)
public void paint(Graphics g)
{
g.drawString(yaricap+" m capindaki kurenin hacmi : "+hacim+" m^3",25,50);
}
//=====
//kurenin hacmini hesaplama metodu
public double kureninhacmi(double r)
{
return (4.0/3.0)*Math.PI*r*r*r;
}
}

```

02012.JPG



Şekil 2.12 Kürenin hacmini hesaplayan H5A1.html applet çıktısı

11. Bir dik üçgenin hipotenüsünü hesaplayan bir applet programı yazınız. Program dik üçgenin iki dik kenarını applet de girsın ve hipotenus metotuyla hesaplasın ve sonucu applete yazdırın.

Not : Dik üçgenin hipotenüsü :

Hipotenüs = $\sqrt{dikkennar1 * dikkennar1 + dikkennar2 * dikkennar2}$;

Problem 2.34 : Bir dik üçgenin hipotenüsünü hesaplayan H5A2.java programı

```

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 sınıf dik ucgenin hipotenusunu hesaplar
//=====
// sınıf değişkenleri
// bu değişkenler tüm sınıfa aittir
Label kutubasligi1; //Label sınıfı değişkeni (nesnesi) kutubasligi1
TextField kutugirdisi1; // TextField sınıfı değişkeni (nesnesi) kutugirdisi1
Label kutubasligi2; //Label sınıfı değişkeni (nesnesi) kutubasligi2
TextField kutugirdisi2; // TextField sınıfı değişkeni (nesnesi) kutugirdisi2
double dikkennar1; // Gerçek değişken dikkennar1
double dikkennar2; // Gerçek değişken dikkennar1
//=====
// pencereyi başlatma metodu
// pencere ilk basladığında
// değişkenler buradaki değerleri alırlar
public void init()
{
kutubasligi1=new Label("Birinci dik kenarı giriniz : ");

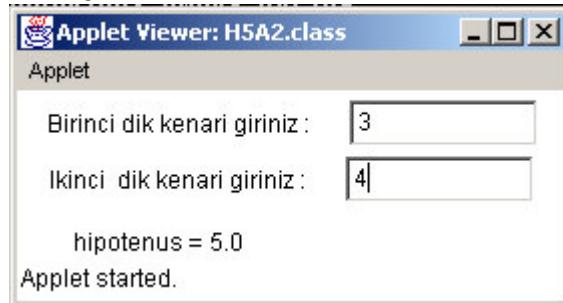
```

```

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 metodu
// Her yeni rakam girildiginde bu metot cagrilir
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 metodu)
public void paint(Graphics g)
{
g.drawString(" hipotenus = "+hipotenus(dikkenar1,dikkenar2),25,80);
}
//=====
//ucgenin hipotenusunu hesaplama metodu
public double hipotenus(double dk1,double dk2)
{
return Math.sqrt(dk1*dk1+dk2*dk2);
}
}

```

02013.JPG



Şekil 2.13 Bir dik üçgenin hipotenüsünü hesaplayan H5A2.html applet çıktısı

12. H3OD5_2000

iki kenar ve aralarındaki açı (derece cinsinden) verildiğinde üçgenin üçüncü kenarını hesaplayacak

**public static double ucgenkarsikenar(double birinciyen, double ikinciyen,
double aciderece)**

metodunu yazınız.

not:karsikenar=Math.sqrt(birinciyen*birinciyen+ikinciyen*ikinciyen-
2*birinciyen*ikinciyen*Mathd.cosd(aradakıacı derece));

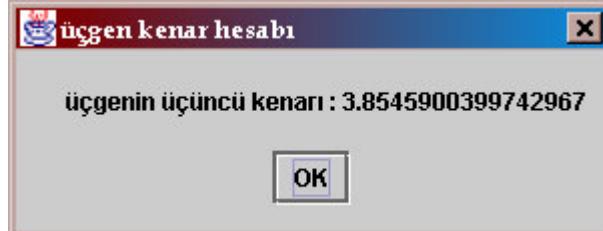
bu metodu H3OD5_2000 sınıfı **JoptionPane** konsol programı main metodunda kullanınız (iki yan kenar ve aradaki açı derece cinsinden verilecek, üçüncü kenar hesaplanarak yazılacak)

Problem 2.35 H3OD5_2000.java programı diküçgenin hipotenüsünü hesaplar

```
import javax.swing.JOptionPane; // giriş çıkış  
import Mathd;
```

```
public class H3OD5_2000  
{  
// ucgenin kenarının bulunması 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 kenarı giriniz : "));  
sayi2=Double.parseDouble(JOptionPane.showInputDialog  
("ikinci kenarı giriniz : "));  
sayi3=Double.parseDouble(JOptionPane.showInputDialog  
(" iki kenar arasındaki açıyı derece cinsinden giriniz : "));  
JOptionPane.showMessageDialog(null,  
"üçgenin üçüncü kenarı : "+kkenar(sayi1,sayi2,sayi3),  
"üçgen kenar hesabı",  
JOptionPane.PLAIN_MESSAGE);  
System.exit(0);  
}  
}
```

02014.JPG



Şekil 2.14 Bir dik üçgenin hipotenüsünü hesaplayan H3OD5_2000.java programının JoptionPane çıktısında görülmesi

13. H3AL2 sınıfında **static double exp(double x)** metodunu yaz ve aynı sınıf içinde çağrıarak ekranдан girilen bir x değeri için sonucu konsol ortamında hesaplettir.

Problem 2.36 : H3AL2.java exp metotunun çağrılması

```
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 sayı giriniz : ");
sayi=cin.readDouble();
System.out.println("girilen sayının exponenti : "+exp(sayi));
}
}

```

Bir sayı giriniz : girilen sayının exponenti : 2.7182818284590455

14. H3AL2_2000 sınıfında **static double exp(double x)** metodunu yaz ve aynı sınıf içinde çağrıarak ekrandan **JoptionPane** kullanılarak girilen bir x değeri için sonucu konsol ortamında hesaplaştır.

Problem 2.37 : H3AL2_2000.java exp metodunun çağrılması
import javax.swing.JOptionPane; // giriş çıkış

```

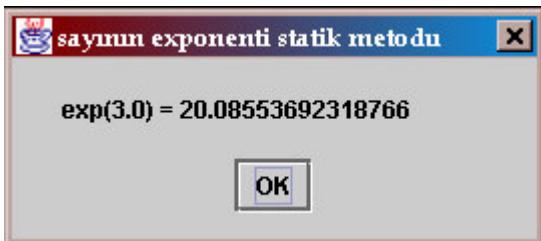
public class H3AL2_2000
{
// sayının exponenti static metodu

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 sayı giriniz : "));
JOptionPane.showMessageDialog(null,"exp("+sayi+") = "+exp(sayi),
"sayının exponenti statik metodu",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```

02015.JPG



Şekil 2.15 Sayının exponentini statik metodla hesaplayan H3AL2_2000.java programını JOptionPane kullanılarak çıktı

15. H3AL2 sınıfındaki **static double exp(double x)** metodunu H3AL3 sınıfında çağırarak ekrandan girilen bir x değeri için sonucu konsol ortamında hesaplettir.

Problem 2.38 : [H3AL3.java](#) exp metodunun çağrılması

```
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 sayı giriniz : ");
        sayi=cin.readDouble();
        System.out.println("girilen sayının exponenti : "+H3AL2.exp(sayi));
    }
}
```

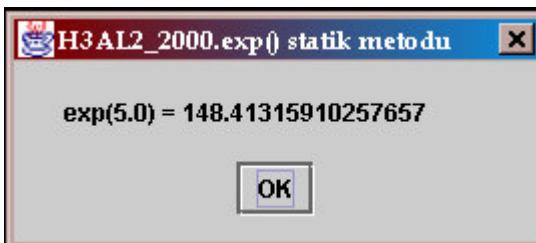
16. H3AL2_2000 sınıfındaki **static double exp(double x)** metodunu H3AL3_2000 sınıfında çağırarak ekrandan girilen bir x değeri için sonucu konsol ortamında hesaplettir.

Problem 2.39 : [H3AL3.java](#) exp metodunun çağrılması

```
import javax.swing.JOptionPane; // giriş çıkış
import H3AL2_2000;

public class H3AL3_2000
{
    public static void main(String[] args)
    {
        double sayi;
        sayi=Double.parseDouble(JOptionPane.showInputDialog
        ("Bir gerçek sayı giriniz : "));
        JOptionPane.showMessageDialog(null,"exp("+sayi+") = "+H3AL2_2000.exp(sayi),
        "H3AL2_2000.exp() statik metodu",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

02016.JPG



Şekil 2.16 Sayının exponentini statik metodla hesaplayan H3AL3_2000.java programını JOptionPane kullananan çıktısı

17. H3AL4 sınıfında yazılan **double exp(double x)** metodunu aynı sınıfda çağırarak ekranдан girilen bir x değeri için sonucu konsol ortamında hesaplattır.

Problem 2.40 : [H3AL4.java](#) exp metodunun çağrılması

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 sayı giriniz : ");
sayi=cin.readDouble();
System.out.println("girilen sayinin exponenti : "+y.exp(sayi));
}
```

18. H3AL4_2000 sınıfında yazılan **double exp(double x)** metodunu aynı sınıfda çağırarak ekranдан girilen bir x değeri için sonucu JOptionPane kullanarak konsol ortamında hesaplattır.

Problem 2.41 : [H3AL4_2000.java](#) exp metodunun çağrılması

import javax.swing.JOptionPane; // giriş çıkış

```
public class H3AL4_2000
{
// sayinin exponenti dinamik metodu

public double exp(double x)
{
double faktoriyel=1;
double us=1;
double exponent=1;
```

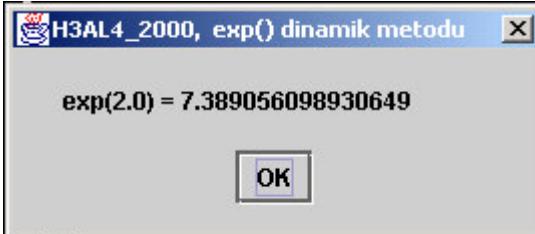
```

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 sayı 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



Şekil 2.17 Sayının exponentini dinamik metodla hesaplayan H3AL3_2000.java programını JoptionPane kullanan çıktısı

19. H3AL5a sınıfında yazılan double exp(double x) metodunu H3AL5 sınıfında çağırarak ekranдан girilen bir x değeri için sonucu konsol ortamında hesaplattır.

Problem 2.42 : [H3AL5.java](#), exp metodunun çağrılması

```

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 sayı;
Text cin=new Text();
H3AL5a y=new H3AL5a();
System.out.println("Bir sayı giriniz : ");
sayı=cin.readDouble();
System.out.println("girilen sayının exponenti : "+y.exp(sayı));
}
}

```

20. H3AL5a sınıfında yazılan **double exp(double x)** metodunu **H3AL5_2000** sınıfında çağrıarak ekrandan girilen bir x değeri için sonucu swing JOPtionPane kullanarak konsol ortamında hesaplettir.

Problem 2.43 : [H3AL5.java](#), exp metodunun çağrılması

```
import javax.swing.JOptionPane; // giriş çıkış
```

```

class H3AL5a
{
// sayının 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 sayı;
sayı=Double.parseDouble(JOptionPane.showInputDialog
("Bir gerçek sayı giriniz : "));
H3AL5a y=new H3AL5a();
JOptionPane.showMessageDialog(null,"exp("+sayı+") = "+y.exp(sayı),
"H3AL5a, exp() dinamik metodu",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}
```

21. H3AL5a sınıfında yazılan **double exp(double x)** metodunu **H3AL6 sınıfında ve H3AL6.java dosyasından** çağrıarak ekrandan girilen bir x değeri için sonucu konsol ortamında hesaplettir.

Problem 2.44 : [H3AL6.java](#), exp metodunun çağrılması

```

import java.io.*;
import H3AL5a;
public class H3AL6
{
public static void main(String[] args) throws IOException

```

```

{
double sayı;
Text cin=new Text();
H3AL5a y=new H3AL5a();
System.out.println("Bir sayı giriniz : ");
sayı=cin.readDouble();
System.out.println("girilen sayının ekseni : "+y.exp(sayı));
}
}

```

22. H3AL5a sınıfında yazılan **double exp(double x)** metodunu **H3AL6_2000 sınıfında ve H3AL6_2000.java dosyasından** çağrıarak ekrandan girilen bir x değeri için sonucu konsol ortamında hesapla.

Problem 2.45 : H3AL6_2000, exp metodunun çağrılması

```

import javax.swing.JOptionPane; // giriş çıkış
import H3AL5a;

public class H3AL6_2000
{

public static void main(String[] args)
{
double sayı;
sayı=Double.parseDouble(JOptionPane.showInputDialog
("Bir gerçek sayı giriniz : "));
H3AL5a y=new H3AL5a();
JOptionPane.showMessageDialog(null,"exp("+sayı+") = "+y.exp(sayı),
"H3AL5a, exp() dinamik metodu",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```

23 . H3OD1

Math kütüphanesinde aşağıdaki metodlar tanımlanmıştır.

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 değeri : -pi/2 den pi/2 ye kadar.

public static native double acos(double a)

sonuç değeri 0 dan pi' ye kadar

public static native double atan(double a)

sonuç değeri : -pi/2 den pi/2 ye kadar.

public static native double atan2(double a,

double b)

sonuç değeri : -pi den pi ye kadar

bu metodların hepsi radyan cinsinden tanımlıdır. Bu metodların derece eşitlerini tanımlamak istiyoruz. [Mathd](#)

sınıfı yaratarak, bu sınıfta

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 değeri : -90 dan 90 dereceye kadar.
 public static double acosd(double a)
 sonuç değeri 0 dan 180 dereceye kadar
 public static double atand(double a)
 sonuç değeri : -90 dan 90 a kadar.
 public static double atan2d(double a,
 double b)
 sonuç değeri : -180 den 180 dereceye kadar
 statik metodlarını oluşturunuz. [H3OD1.java](#) applet programını yazınız. Bu programa derece cinsinden bir girdi giriniz ve hesaplanmış sin,cos,tan değerlerini apletde yazdırınız.
 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;

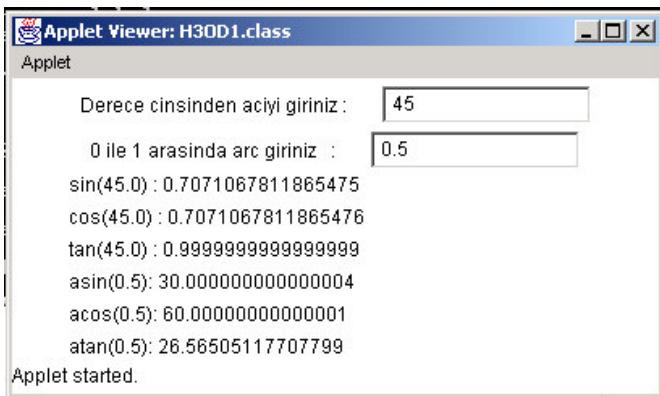
public class H3OD1 extends Applet implements ActionListener
{
    double a;
    double b;
    Label s1;
    TextField t1;
    Label s2;
    TextField t2;
    public void init()
    {
        s1=new Label("Derece cinsinden aciyi giriniz : ");
        t1=new TextField(15);

        s2=new Label("0 ile 1 arasında 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 fonksiyonların derece cinsinden yazılmış 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



Şekil 2.18 Bir derece cinsinden trigonometrik fonksiyonları hesaplayan H3OD1.html çıktısı

24. H3OD2

Bir konsol programı oluşturunuz. Bu programa 0-1 arasında bir girdi giriniz. Mathd sınıfından atand,acosd ve asind statik metodlarını kullanarak bu metodların sonuçlarını konsol ortamında çıktıya yazdırınız.

Problem 2.48 : H3OD2.java,

```
import java.io.*;
import Text;
import Mathd;
public class H3OD2
{
// trigonometrik fonksiyonların derece cinsinden yazılmış hali
public static void main(String args[]) throws IOException
{
    double a;
    Text cin=new Text();
    System.out.println("0-1 arasında bir sayı 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");
}
}
```

25. H3OD3

kendi kendini çağırınan (recursive), tabanın kuvvetinci kuvvetini hesaplayacak (taban^{kuvvet}, $2^3=8$ gibi)

public static pow(double taban, double kuvvet)

metotunu H3OD3 sınıfında, konsol programı olarak yazınız. main metodunda dışarıdan bir sayı okuyarak bu metot yardımıyla sonucu hesaplayınız.

not: H3OD3.pow(taban,1) çağrıını taban değerini geri verecektir.

Problem 2.49 : H3OD3.java,

```
import java.io.*;

public class H3OD3
{
// sayının eksponenti dinamik metodu

public double kuvvet(double x,double y)
{
// y nin pozitif olduğu 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));
}
}

```

26. H3OD4

döngü yapısıyla hesap yapan, tabanın kuvvetinci kuvvetini hesaplayacak (taban^bkuvvet, $2^3=8$ gibi)
 public pow(double taban, double kuvvet)
 H3OD4a sınıfında yazınız. H3OD4 sınıfında bir applet programı oluşturarak H3OD4a sınıfındaki pow metodu
 yardımıyla verilen bir sayının verilen bir kuvvetini hesaplayınız. ve appletde çıktı olarak yazdırınız.

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 fonksiyonların derece cinsinden yazılmış hali
    public void paint(Graphics g)
    {
        g.drawString("taban "+a+" nin "+b+"inci kuvveti : "+ H3OD4a.kuvvet(a,b),35,70);
    }
}

```

27. H3OD4_2000

döngü yapısıyla hesap yapan, tabanın kuvvetinci kuvvetini hesaplayacak (taban^kkuvvet, $2^3=8$ gibi)
 public pow(double taban, double kuvvet)
 H3OD4a sınıfında yazınız. H3OD4_2000 sınıfında bir siwng JApplet programı oluşturarak H3OD4a sınıfındaki
 pow metodu yardımıyla verilen bir sayının verilen bir kuvvetini hesaplayınız. ve appletde çıktı olarak yazdırınız.

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



Şekil 2.19 H3OD1.html çıktısı

28. H3OD5

iki kenar ve aralarındaki açı (derece cinsinden) verildiğinde üçgenin üçüncü kenarını hesaplayacak
public static double ucgenkarsikenar(double birinciyan, double ikinciyan,
double aciderece)

metotunu yazınız.

not : karsikenar=Math.sqrt(birinciyan*birinciyan+ikinciyan*ikinciyan-
2*birinciyan*ikinciyan*Mathd.cosd(aradaki açı derece));

bu metodu H3OD5 sınıfı konsol programı main metodunda kullanınız (iki yan kenar ve aradaki açı derece cinsinden verilecek, üçüncü kenar hesaplanarak yazılacak)

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));
}
}
```

29. Hata fonksiyonu erf(x) aşağıdaki formülle tanımlanabilir.

$$erf(x) = \frac{1}{\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{n!(2n+1)}$$

Bu formülü hesaplayacak olan statik erf metodunu yazınız ve java swing çıktısı verecek bir örnekte deneyiniz.

30. Bir konsol programı yazınız. Text sınıfı kullanarak dışarıdan x değerini, celcius, fahrenheit cinsinden girilen değerin celcius eşitini hesaplasın. Bunun için

C=5/9*(F-32)

formülünü kullanabilirsiniz.

İkinci metot, fahrenheit, derece celcius cinsinden girilen değerin fahrenheit eşdeğerini hesaplasın.

31. Bir applet programı yazınız. Bu programın içindeki kubunhacmi metodu kübün hacmini hesaplasın. Applet kullanıcıya kübün kenar uzunluğunu sorsun ve hacmini versin. okuyunuz. erf statik metodu yaratarak formülü hesaplayınız. Ana programda erf metodunu çağırınız ve sonucunu yazdırınız. Hata analizi yaparak döngüyü kaç terim için hesaplamamanız gerekiğine karar veriniz.

32. Sıcaklık.java programını denemek için bir swing JApplet programı yazınız.

33. mesafe metodunu yazınız. Bu metod iki noktanın (x1,y1) ve (x2,y2) arasındaki mesafeyi hesaplasın. Bu metodu bir konsol test programında deneyin.

34. Bir tam sayı eğer kendisinden başka bir tam sayıyla bölünemiyorsa, asal sayı adını alır. Örneğin 2,3,5,7,11..

Asal sayıları bulan bir metot yazınız. Bu metodu bir applette kullanarak 1 den 1000 e kadar olan sayılar arasındaki asal sayıları belirleyiniz.

35. Üç (gerçek veya tamsayı) sayının arasından minimum olanı bulan **minimum3** metodunu yazınız ve bir konsol programında test ediniz

36. Üç sayının (gerçek veya tamsayı) arasından maksimum olanı bulan **maksimum3** metodunu yazınız ve bir konsol programında test ediniz

37. tekmi isimli boolean bir metot yazınız. bu metot eğer girilen sayı tekse true değeri, sayı çiftse false değeri versin. Bu metodu bir applet programında test ediniz.

38. kürenin hacmini hesaplayan hacim(double x) metodunu içeren kureApplet applet programını olusturunuz. Bu program kürenin yarıçapını sorsun, ve sonuç olarak hacim metodunda hesaplanan kürenin hacmini versin.

Problem 2.53 kürenin hacmini hesaplayan küreapplet.java programı

```
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 tamsayı giriniz : ");
        rakam=new TextField(10);
        rakam.addActionListener(this);
        sonucetiketi=new Label("Kürenin 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 GİRİŞ

3.0 STANDART KAVRAMLARIN TÜRKÇE KARŞILIKLARI

Class : sınıf ana nesne kökenli programlama elemanı.

Object : nesne sınıf (Class) yapıları bilgisayar belleğinde bir işlem için kullanıldığında aldıkları (özel) isim.

new : sınıfların kurucu metotunu çağrıp nesne tanımını ve bilgisayar adreslerini oluşturan deyim

Method: metot sınıfların içindeki işlevsel program parçacıkları

Constructor : Kurucu metot nesne ilk defa oluşturulurken hangi sınıf değişkenlerinin hangi değerleri alacağını belirleyen metot

3.1 SINIF YAPISININ GENEL TANIMI

Şu na kadarki bütün programlarımızda sınıf(class) yapısını kullandık. Fakat tam olarak sınıfların ne olduğunu ve neler yapabileceklerini anlatmadık. Sınıf yapısının en onemli ozelliği yeni bir değişken türü tanımlıyor olmasıdır. Şu ana kadar programlarımıza tanımlamak amacıyla çeşitli sınıflar tanımladık. Bu bölümde sınıfların değişken türü olarak tanımlamasını ve kullanmasını göreceğiz.

Sınıf (Class) deyince herhangi bir nesnenin genel tanımlanmasını anlıyoruz. Bu sınıflardan değişkenler tanımlandıklarında nesneleri oluştururlar. Örneğin kalem dediğimizde genel bir sınıfı anlatıyoruz. Alının cebindeki kırmızı kalem ise belirli bir nesnedir. Sınıfın genel tanımı şu şekilde yapılabilir :

```
class sınıf_ismi
{
//sınıfa ait değişken listesi
değişken_türü değişken_ismi;
değişken_türü değişken_ismi;
.....
//sınıfa ait metot listesi
değişken türü metotu(metot girdi değişken listesi)
{
//metotun içindeki hesaplanan işlemler
return çıktı_değişkeni;
}
.....}
```

Sınıf (class) tanımında metotların (veya değişkenlerin) bulunması şart değildir. Bulundukları takdirde bunlar o sınıfa ait veya sınıfın üyesi metotlar olarak düşünülür. İlk sınıf örneği olarak içinde hiçbir metotun yer almadığı bir kutu sınıfı tanımlyalım. Bir kutuyu üç geometrik boyutu (eni,boyu,yüksekliği) cinsinden tanımlayabiliriz.

```
class Kutu
{
double en;
double boy;
double yükseklik;
}
```

Üstte bahsettiğimiz gibi sınıf yeni bir değişken türü yaratır. Yukardaki sınıfın yarattığı değişken türünün adı Kutu'dur. Kutu değişkenini başka bir programda tanımlarken budeğişken türünden bir nesne yaratılır. Örneğin

Kutu kutu1= new Kutu();

kutu1, kutu türünde tanılmış bir nesnedir. Bu nesnenin kendisinin de en boy ve yükseklik olmak üzere üç alt değişkeni mevcuttur. Eğer bir programda kutu1 değişkeni tanımlanmışsa, bu değişkenin alt değişkenlerine şu şekilde ulaşılabilir :

kutu1.en=1.0;

Şimdi Kutu sınıfından nesneyi kullanacak ufak bir kutatesti sınıfı oluşturalım ve Kutu sınıfını bu sınıfta kullanalım.

```

import javax.swing.JOptionPane; // giriş çıkış

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,
        "sınıf yapısı: kututesti1",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}

```

Bu programın adının [kututesti.java](#) olması zorunludur. Çünkü ana metot(main) kututesti sınıfında tanımlanmıştır. Fakat programı javac terimiyle bilgisayar diline çevirdiğimizde iki tane sınıf(class) dosyasının olduğunu görürüz. Birisi **Kutu.class** diğeride **kututesti.class**. Ana program kututesti olduğu için bu programın işletimi **java kututesti**

deyimini kullanarak yapılır. Program işletildiğinde

03001.JPG



Şekil 3.1 kututesti.java programının JOptionPane grafik çıktısındaki görünümü

Sonucunu verir. kutu1 nesnesi yaratıldığında bilgisayar belleğinde kutu1 için bir adres tanımlanır. New Kutu() deyimi verildiğinde ise bilgisayarın belleginde en,boy ve yükseklik değişkenleri için yer ayrıılır ve bu yerlerin adresleriyle kutu1 nesnesinin adresi arasında bağlantı sağlanır. Bunu grafik olarak anlatalım :

Tablo 3.1 Nesne adreslerinin bilgisayar belleğindeki yerleri ve yerleşmesi

Deyim	Bilgisayar ana nesne belleği	Nesne alt hafıza belleği
Kutu kutu1;	null	Mevcut değil
kutu1=new Kutu();	Kutu1 kutu1 adresi (örneğin: 23436A3F)	en Boy Yükseklik

Not: null terimi bilgisayar beleginin boş olduğunu gösterir. Yukardaki grafikte ilk olarak

Kutu kutu1;

tanımı yapıldığında bilgisayar belleğinde 1 byte boyutunda bir yer tanımlar. Ve bu yere boş anlamına gelen null değerini yükler. İkinci durumda ise (**kutu1=new Kutu()** tanımında) kutu1 nesne adresindeki kutuda en,boy,yükseklik değişkenlerinin adresleri kaydedilmiştir. Bu adres aracılığıyla en boy ve yükseklik değişkenlerine benimkutum degişkeni üzerinden ulaşılabilir. Diğer deyimle en,boy,yükseklik kutu1 nesnesinin alt değişkenleridir.

3.2 KURUCU (CONSTRUCTOR) METOT

İkinci bir programda Kutu sınıfı iki nesne tanımlama gereği duyduğumuzu varsayıyalım. Buna örnek olarak Şekil 3.2.1 de kututesti1.java programı tanımlanmıştır.

Program 3.1 : Kurucu metot örneği, Kutu sınıfı iki nesne tanımlayan [kututesti1.java](#) programı

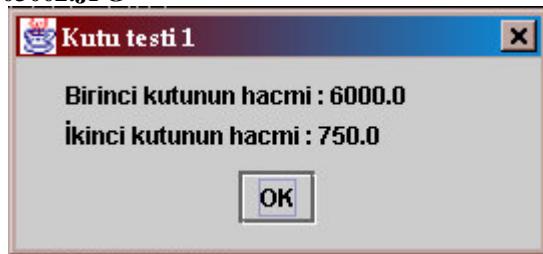
```
import javax.swing.JOptionPane; // giriş çıkış
```

```
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+="İkinci kutunun hacmi : "+hacim+"\n";
        JOptionPane.showMessageDialog(null,s,
        "Kutu testi 1",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Bu programın çıktısı :

03002.JPG



Şekil 3.2 kututesti1.java

Bu programda Kutu sınıfından **kutu1** ve **kutu2** değişkenleri tanımlanmıştır. Bu basit programda bile en,boy ve yükseklik değişkenlerini iki defa çağrırmak program listesinin uzamasına sebep olmuştur. Gerçek uygulamalarda ise çok daha yüksek sayıda değişken içeren sınıflar oluşturulabilir ve bu sınıflarda oluşturulan toplam nesne

sayısı da çok fazla olabilir. Bu yüzden nesneyi ilk tanımlarken aynı zamanda değişkenlerinin değerlerini de tanımlayabilirsek, çok daha okunaklı ve kısa bir bilgisayar listesiyle aynı işi yapabiliriz. Bu iş için kurucu (constructor) adı verilen metodlar kullanılır. Kurucu metodları diğer metodlardan biraz farklıdır. Once metod önüne metot değişken türü gelmez ve metod dışına hiçbir değişken göndermezler. İsimleri de her zaman sınıf(class) ismiyle aynı olur. Program 3.1 deki programı kurucu (constructor) metodunu kullanarak tekrar yazalım.

Program 3.2 : Kurucu metod örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan [kututesti2.java](#) programı

```
import javax.swing.JOptionPane; // giriş çıkış
```

```
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 aynı sonucu verir. Fakat burada bir önceki programda yapılan

Kutu kutu1=new Kutu();

Tanımı yerini

Kutu kutu1=new Kutu(10.0,20.0,30.0);

Tanımına bırakmış ve kututesti2 sınıfı yazılarken en ,boy yükseklik değişkenlerinin yeniden tanımlanması ihtiyacını ortadan kaldırılmıştır.

3.3 DİĞER METOTLAR

Tekrar yukarıda incelediğimiz kutu sınıfını göz önüne alalım. Hacim bir kutunun özellikleri arasında yer alır, ve her zaman kutunun hacminin hesaplanması gerekebilir. Bu yüzden hacim işlemini Kutu sınıfının içinde

tanımlamak daha ideal olacaktır. Şekil 3.3.1 deki kutuornegi3.java programında hacim hesaplarını yapan hacim metodu Kutu sınıfının içinde tanımlanmıştır.

Program 3.3 : Kurucu metot örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan ve hacmi hacim() metotuyla hesaplayan kututesti3.java programı

```
import javax.swing.JOptionPane; // giriş çıkış
```

```
class Kutu
{
double en;
double boy;
double yukseklik;

//kurucu metod kutu
Kutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklik=y;
}

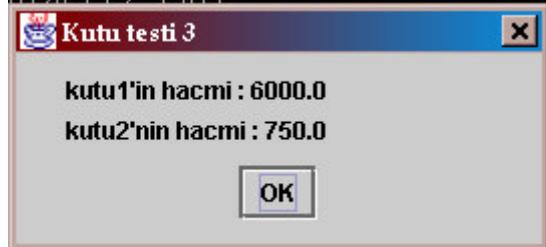
//bu metot kutunun hacmini hesaplar
double hacim()
{
return en*boy*yukseklik;
}

}

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 çıktısı yine aynı olacaktır.

03003.JPG



Şekil 3.3 kututesti3.java

Bir önceki bölümde metot yazımı konusunda anlattıklarımızın hepsini buraya ilave edebiliriz.

3.4 THIS DEYİMİNİN KULLANIMI

Bazen bir metotun içinde o metotun ait olduğu sınıfın yaratılacak nesneyi veya o nesnenin bir alt değişkenini tanımlamak gereklidir. Nesne daha tanımlanmadığından direkt olarak nesne ismini kullanamayız. Bunun yerine Java **this** deyimini kullanır. This deyimi bilhassa sınıfın ait değişkenlerinin aynı ismi metotda kullanılmışsa da işe yarar. Bu durumda tüm sınıfın ait değişkenler **this.değişken_ismi** komutuyla çağrılabılır. geçen bölümde sınıf ve metoda ait değişkenleri izlerken metotvesinifdeg.java ve metotvesinifdeg1.java isimlerinde iki örnek problemi incelemiştik.

Şimdi de aynıgruptan metotvesinifdeg3.java programını inceleyelim.

Program 3.4 : [metotvesinifdeg3.java](#) programıyla ve **this** deyimi yardımıyla metotun içinde yerel ve sınıfın ait değişkenlere ulaşım.

```
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 hic bir degişken 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);
}
}
```

Sınıfın ait her değişkeni başına **this.** getirerek kullanabiliriz. Fakat **this** kullanımı mutlaka gerekli değilse kullanılmaması tavsiye edilir.

Örneğin yukarıdaki kurucu metot kutu

```
//kurucu metot kutu
kutu(double en,double boy,double yukseklik)
{
this.en=en;
this.boy=boy;
this.yukseklik=yukseklik;
}
```

şeklinde **this** değişkenini kullanarak da yazılabildi. Burada girdi değişkeni ile sınıf değişkeninin adı aynı verilmiştir. **This** sol taraftaki değişkenin sınıf değişkeni olduğunu bilgisayara bildirmektedir.

3.5 SINIF DEĞİŞKENLERİNİN DIŞ DÜNYADAN GİZLENMESİ

Yukarda incelediğimiz kutu metodu. Program 3.5 de daha detaylı yazılmıştır. Burada en boy ve yükseklik değişkenleri private olarak tanımlanmıştır. Private olarak tanımlanan değişkenlere dışarıdan direk olarak

```
Kutu x=new Kutu(3.0,5.0,7.0);
x.en=8.0;
```

gibi direk ulaşım mevcut değildir. Bu değişkenlere ulaşım (değerlerini değiştirme veya çağrıma) ancak ulaşım metodları adını verdigimiz metodlar aracılığıyla yapılabilir.

Program 3.5 : Kurucu metot örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan ve hacmi hacim() metotuyla hesaplayan ve diğer metod tanımlarının da bulunduğu kututesti4.java programı.

```
import javax.swing.JOptionPane; // giriş çıkış

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;
    }

    //sınıf değişkenlerini değiştirme set metodu
    public void setKutu(double e,double b,double y)
    {
        en=e;
        boy=b;
        yukseklik=y;
    }

    //sınıf değişkenlerine ulaşma (get) metodları
    public double getEn()
    {return en;}

    public double getBoy()
    {return boy;}

    public double getTukseklik()
    {return yukseklik;}

    //bu metot kutunun hacmini hesaplar (işlem- hesap) metodu
    public double hacim()
    {
        return en*boy*yukseklik;
    }

    //string çıktı metodu
    public String toString()
    {
        return("en = "+en+" boy = "+boy+" yükseklik = "+yukseklik+"\n"+
               "Hacim = "+hacim()+"\n");
    }
}
```

```

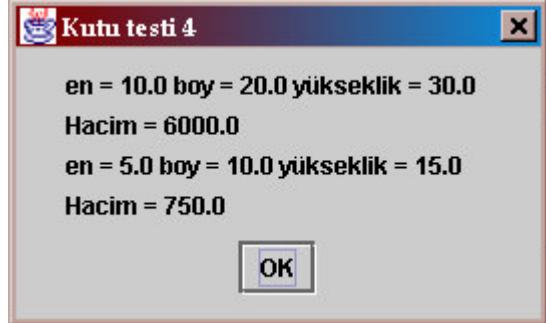
}

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 programın çıktısı :

03004.JPG



Şekil 3.4 kututesti4.java

İkinci bir örnek olarak dörtislemhesapmakini sınıfına bakalım. Burada da sınıf değişkenleri protected olarak tanımlanmıştır. Aşağıda dört işlem (toplama çıkarma çarpma bolme yapan dörtislemhesapmakini sınıfını görüyorsunuz.

Program 3.5 : dörtislemhesapmakini.java programı ve sınıf değişkenlerinin değerlerinin metot kullanılarak değiştirilmesi

```

public class dörtislemhesapmakini
{
    //sınıf değişkenleri x sonuc hafiza
    protected double x,sonuc,hafiza;
    // kurucu metot dörtislemhesapmakinesi
    public dörtislemhesapmakini(double y)
    {
        //constructor
        sonuc=y;
        hafiza=0;
    }
    // ikinci bir kurucu metot dörtislemhesapmakinesi
    public dörtislemhesapmakini()
    {
        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 değeri değiştirilir
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()
{
//hafızayı çağır
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 sınıfı (class) sınıf değişkeni **sonuc** değerlerinin değiştirilmesi için **gir** metodu yazılmıştır. Bunun sebebi programlarda sınıfı ait değişkenlere dışarıdan direk ulaşımın önlenmesini sağlamaktır. İyi programlama şartlarından birisi sınıfın değişkenlerine sadece o sınıfın metotları üzerinden ulaşmaktadır. Herkese serbest ulaşım verilmez. Hatta bunu sağlamak için değişkenlerin başına sadece bu sınıfı aittir dışarıdan kullanılamaz anlamındaki **private** sözcüğü getirilebilir. **private** sözcüğü yerine ancak dışarda bu sınıfın yaratılana yeni sınıflar (akraba sınıflar) kullanabilir, yabancı sınıflar kullanamaz anlamındaki **protected** sözcüğü de sık olarak kullanılır. Protected sözcüğü aynı directorydeki diğer dosyaları da (public) gibi kabula eder, fakat başka direktoryelerdeki dosyalar için private olarak davranış gösterir.

dortislemhesapmakinası.java programının x, sonuc ve hafiza değişkenleri için **protected** sözcüğünü kullandığımız için

```
dortislemhesapmakinası h = new dortislemhesapmakinası();
h.sonuc = 2.0;
```

şeklinde ulaşılabilir. Aynı işlemi yapmak için aşağıdaki üç yoldan birini kullanabiliriz.

```
dortislemhesapmakinası h = new dortislemhesapmakinası();
h.gir(2.0);
```

veya

```
dortislemhesapmakinası h = new dortislemhesapmakinası(2.0);
şeklinde sonuç değişkenine 2.0 değeri atanır.
```

Hafiza değişkeninin değerinin değiştirilmesi için ise :

```
h.M(3.0);
metodu çağırılır.
```

Sınıf değişkenlerinin dış dünyadan gizlenmesi niçin gereklidir? Bunun en önemli sebebi program güvenliğinin sağlanmasıdır. Değişkenlere dışarıdan direk müdahale edilebilirse, değerleri de değiştirilebilir. Örneğin değişkeni bir banka hesabı veya şirketin özel hesabı olarak düşünübiliriz. Dış dünyadan gizlenmemiş bir değişkenin değeri değiştirilebilir. Private veya protected değişkenler ise ancak metotlar üzerinden değiştirilebilir, burada yeterli güvenlik önlemlerini almak mümkündür.

3.6 SINIFLARDA KALITIM (INHERITANCE)

Bir sınıfın kurulmasında o sınıfla direk olarak ilişkisi olan bir sınıfın onun tüm metotları ve değişkenleri **kalıtım (inheritance)** yoluyla yeni kurulan bir sınıfın aktarılabilir. Kalıtım için sınıf(class) tanımında **extends** kelimesi kullanılır. Eğer bir sınıfın daha alt sınıflara kalıtım yoluyla aktarılacaksa **private** değişken türü yerini **protected** değişken türüne bırakır. Protected private deyiminin aynıdır. Fakat kalıtımıyla değişkenlerin aktarıldığı alt sınıflara değişkenleri görme izni verir. Private terimi ise kalıtımıla aktarılan alt sınıflardan bile o değişkenleri saklar.

Program 3.6 da nokta sınıfı tanımlanmıştır. Program 3.7 de ise nokta sınıfından kalıtım yoluyla yaratılan daire sınıfını görüyoruz. Programda da görüleceği gibi daire sınıfının tanımı

```
public class daire extends nokta
```

terimiyle yapılmıştır. Ayrıca daire sınıfının kurucusu yazılırken nokta sınıfının kurucusu direk olarak **super(a,b);** terimiyle çağırılmıştır.

Burada önemli bir noktayı hatırlatalım :

"Java dilindeki bir sınıf sadece bir tane başka sınıfın metot ve değişkenlerini kalıtım yoluyla aktarabilir."

Program 3.6 : [nokta.java](#) programı ve nokta sınıfının tanımı.

```
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](#) programı ve daire sınıfının tanımı. Daire sınıfı nokta sınıfından Kalıtım (inheritance) yoluyla türetilmiştir.

```
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;
}
}

```

Kalıtım yoluyla aktarılan metotlar ve değişkenler yeni programda o programın içinde yazılmış gibi aynen kullanılabilir.

3.7 SINIFLARI BAŞKA BIR SINIFTA NESNE OLARAK ÇAĞIRARAK BiR ARAYA GETİRME (COMPOSITION)

Sınıftaki değişken ve metodları kalıtım yoluyla aktarabildiğimiz gibi, sadece üst sınıfa ait bir nesne (object) tanımını alt sınıfta tanımlıyalarak ta yapabiliriz. Buna nesne yoluyla aktarım (ingilizce composition) diyoruz.

Program 3.8 : 1 daire1.java programı ve daire sınıfının tanımı. Nokta sınıfı daire1 sınıfında merkez adlı nesne olarak tanımlanmıştır.

```

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 programında

3.8 KALITIM YOLUYLA ALT SINIFA BAĞLANMIŞ ÜST SINIF REFERANSI ÜZERINDEN ALT SINIFI ÇAĞIRMA

Yukarıda nokta ve daire sınıflarını tanımlamıştık. Baska bir sınıfın kalıtım yoluya türeyen sınıfın(dairenin noktadan kalıtım yoluya türemesi gibi), üst sınıfın adresi(tanımı) üzerinden programın içinde aktarımı mümkündür.

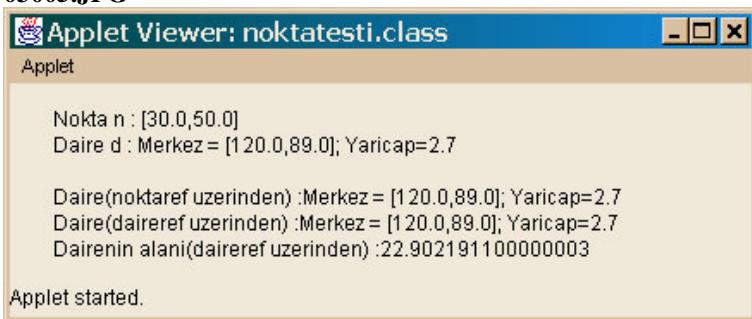
Program 3.9 : [noktatesti.java](#) sınıfında üst sınıf üzerinden adres referansı kullanımı

```

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 yazdıralım
        g.drawString("Nokta n : "+n.toString(),25,25);
        g.drawString("Daire d : "+d.toString(),25,40);
        //c daresinin adresini nokta olarak tanımlanan noktaref e aktaralım
        noktaref=d;
        g.drawString("Daire(noktaref üzerinden) :" +noktaref.toString(),25,70);
        //noktaref nesnesini daireref değişkenine(nesnesine) aktaralım
        daireref=(daire)noktaref;
        g.drawString("Daire(daireref üzerinden) :" +daireref.toString(),25,85);
        g.drawString("Dairenin alanı(daireref üzerinden) :" +daireref.alan(),25,100);
    }
}

```

03005.JPG



Şekil 3.5 Noktatesti.html

Program 3.9 da görülen noktatesti.java sınıfında, n nokta sınıfı nesnesi, ve noktaref noktasını nesnesi, ve d daresını nesnesi, ve daireref daire sınıfı nesnesi tanımlanmıştır. Init metotunda p ve c nesnelerinin adresleri

kurucu metodu new deyimiyle taşıyarak tanımlanmıştır. Dahasonra noktaref=d; deyimiyle aslında daire sınıfından olan d nesnesi nokta sınıfından tanımlanan noktaref nesnesinin adresine yüklenmiştir. Programda da görüldüğü gibi noktaref üzerinden daireyi kullanmak mümkündür çünkü **nokta dairenin kalıtım yoluyla üst sınıfıdır**. Aynı zamanda **cast** işlemcisi (**daire**) terimini kullanarak noktaref'i daireref'e aktarmak da mümkün olmuştur. Bu aktarma nokta dairenin super sınıfı olmama başarılamazdı. Bazen sadece bu süper sınıf aktarımını saglamak amacıyla sınıf hiyerarşisinin en tepesine aslında belli bir görevi olmayan **abstract** sınıflar yerleştirilir. Bu sınıfın alt sınıfları olarak tanımlanan tüm sınıflar bu sınıf adresinden yararlanılarak aktarılabilir. Abstract sınıfların tanımında **abstract** sözcüğü kullanılır. Abstract tür metodların içinde metodların sadece başlıklarını verip tanımları yapılmayıbilir. (metotun ana gövdesi tanımlanmayıbilir). Abstract super sınıfının kullanılmasına söyle bir örnek verebiliriz : sekil isimli bir abstract sınıfın alt sınıfları olarak daire, üçgen, dikdörtgen ve kare sınıflarını oluşturduğumuzu ve bu sınıfların herbirinin kendi şeklini çizecek çiz metotunu içerdigini düşünelim. Programı yazarken abstract sınıf şekilde de alan metodu tanımı yaparız. Ve sekil sınıfından bir nesne tanımlarız. Program çalışırken sekil üzerinden istediğimiz alt şeklin çizimi dinamik olarak her biri için ayrı bir adres tanımlamadan mümkün olur. Aynı şekilde bir nümerik analiz programı yazarken, örneğin her değişik fonksiyon için ayrı ayrı integral metodu yazma yerine abstract tür bir fonksiyon için integral programını yazar ve gerçek foksiyonumuzu tanımladığımız sınıfı bu abstract sınıfı olarak tanımlarız.

Bu kavramı daha sonraki bölümlerimizde daha geniş örneklerle açıklamaya çalışacağız.

3.9 NESNEYİ DİNAMİK OLARAK SİLMEK FINALIZE() METOTU

Java dili kullanılmayan nesneleri silmek için arka planda çalışan bir sisteme sahiptir. Bu yüzden C++ gibi dillerde dinamik nesne tanımlarında mutlaka yer olması gereken nesneyi silme metodları jada o kadar gerekli değildir. Fakat eğer kendiniz metlaka bir nesneyi acil olarak silmek isteseniz **finalize()** isimli bir metodu sınıfınızda tanımlayıp kullanabilirsiniz. Örnek olarak aşağıdaki metodu verebiliriz.

```
protected void finalize() throws Throwable
{
super.finalize();
}
```

3.10 ALIŞTIRMALAR

1. kututesti1.java, kututesti2.java, kututesti3.java ve kututesti4.java programlarını karşılaştırınız. Aralarındaki fark ve benzerlikleri açıklayınız.
2. dairetest.java programını inceleyiniz

Program 3.10 dairetest.java, bu program nokta sınıfından (Program 3.5) kalıtım yoluyla türüyen daire sınıfının (Program 3.6) kullanılmasını 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 kalıtımla nokta sınıfından aktarılmıştır
        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 alanı : 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 alanı : 3.14159

```

3. dairetest_2000.java programını inceleyiniz

Program 3.11 dairetest_2000.java, bu program nokta sınıfından (Program 3.5) kahtım yoluyla türüyen daire sınıfının(Program 3.6) kullanılmasını göstermektedir.

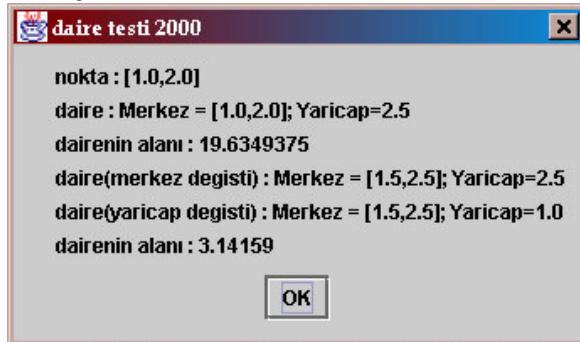
```

import javax.swing.JOptionPane; // giriş - çıkış
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 alanı : "+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 alanı : "+d1.alan()+"\n";
        JOptionPane.showMessageDialog(null,s,
        "daire testi 2000",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}

```

03006.JPG



Şekil 3.6 daire testi 2000 JoptionPage çıktısı

5. Nokta sınıfını kullanarak doğru sınıfını yarat. Doğru iki noktayla tanımlanır. Doğru sınıfının kurucu metotunu, boyunu hesaplayan boy metotunu, üç noktalarını String değişkeni olarak tanımlayan toString metotunu ve gerekli olan diğer metodları tanımla. Konsol programı olarak doğrultestı sınıfı oluşturarak bu sınıfın

İçinde **cokdogrubirdogr** isimli bir nesne oluştur be bu nesnenin başlangıç ve bitiş noktalarıyla boyunu yazdır. Girdi değerlerini program içinde sabit olarak tanımla.

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 cokdogrubirdogr=new dogru(1.0,1.0,2.0,2.0);
        System.out.println(cokdogrubirdogr.toString());
        System.out.println("dogru boyu : "+cokdogrubirdogr.boy());
    }
}
```

([1.0,1.0],[2.0,2.0])
dogru boyu : 1.4142135623730951

6. Nokta sınıfını kullanarak dikdörtgen sınıfını yarat. Dikdörtgen iki noktaya tanımlanır. Dikdörtgen sınıfının kurucu metotunu, çevresini hesaplayan çevre metotunu, Alanını hesaplayan alan metotonu, dikdörtgeni tanımlayan iki noktayı String değişkeni olarak tanımlayan toString metotunu ve gerekli olan diğer metotları tanımla. Applet programı olarak dikdortgentesti sınıfı oluşturarak bu sınıfın içinde **cokguzelbirdikdortgen**

isimli bir nesne oluştur be bu nesnenin başlangıç ve bitiş noktalarıyla boyunu yazdır. Girdi değerlerini program içinde sabit olarak tanımla.

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



Şekil 3.7 dikdortgentesti.html çıktısı

7. dogrutesti1 sınıfını yaz. Bu sınıfda dogruyu tanımlayan noktaları (toplam iki nokta) swing JoptionPane konsol programına ekrandan değişken olarak gir. Dogrunun iki noktasını ve doğru boyunu ekranı çıktı olarak yazdır.

8. Dikdortgentesti1 sınıfını yaz. Bu sınıfda dogruyu tanımlayan üç noktaları JApplet programına dışardan değişken olarak gir.

9. Hesapmakinesi sınıfını yarat. Bu sınıf iki gerçek değişkeni sınıf değişkeni olarak tanımlasın. Topla,cikar,carp ve bol metodlarını yaz ve bu işlemleri metodların içinde tanımla.
bir test programını yaz ve bu sınıf hesapmakinesi sınıfını kullanarak dışarıdan girilen iki gerçek sayının dört işlemini hesapla. **dortislemhesapmakinası** programı Program 3.5 de verilmiştir.

Program 3.15 Dörtişlem hesap makinası appleti tanımlayan H4AL1.java programı

```
import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import dortislemhesapmakinası;

public class H4AL1 extends Applet implements ActionListener
{
    // Dört işlem hesap makinası

    TextField sonucgirdisi;
    TextField kutugirdisi;
    Button Gir,Arti,Eksi,Carpı,Bolu,M,MR,MArtı,MEksi,Isaret,C;
    //Button sınıfı değişkenleri
    double sayı;
    dortislemhesapmakinası D;
    // pencereyi başlatma metodu
    public void init()
    {
        D=new dortislemhesapmakinası();
        sonucgirdisi=new TextField(30);
        sonucgirdisi.setEditable(false);
        add(sonucgirdisi);
        kutugirdisi=new TextField(30);
        add(kutugirdisi);//kutuyu pencereye yerleştir
        Gir=new Button("gir");
        Artı=new Button("+");
        Eksi=new Button("-");
        Carpı=new Button("*");
        Bolu=new Button("/");
        M=new Button("M");
        MR=new Button("MR");
        MArtı=new Button("M+");
        MEksi=new Button("M-");
        Isaret=new Button("+/-");
        C=new Button("C");
        add(Gir);
        add(Artı);
        add(Eksi);
        add(Carpı);
        add(Bolu);
        add(M);
        add(MR);
        add(MArtı);
        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 alanındaki olan olayları 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("");
}
}

```

03008.JPG



Sekil 3.8 Dörtişlem hesap makinası çıktı appleti H4AL1.html

10. Hesapmakinesi sınıfını için. bir JApplet test programı yaz ve bu sınıf hesapmakinesi sınıfını kullanarak dışarıdan girilen iki gerçek sayının dört işlemini hesapla. **dortislemhesapmakinası** programı yukarıda verilmiştir.

Program 3.16 Dörtişlem hesap makinası swing JAppleti tanımlayan H4AL1_2000.java programı

```
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 dortislemhesapmakinası;

public class H4AL1_2000 extends JApplet implements ActionListener
{
    // Dört işlem hesap makinası
    JTextField sonucgirdisi;
    JTextField kutugirdisi;
    JButton Gir,Arti,Eksi,Carpı,Bolu,M,MR,MArtı,MEksi,Isaret,C;
    //JButton sınıfı degiskenleri
    double sayı;
    dortislemhesapmakinası D;
    // pencereyi baslatma metodu
    public void init()
    {
        D=new dortislemhesapmakinası();

        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");
        Artı=new JButton("+");
        Eksi=new JButton("-");
        Carpı=new JButton("*");
        Bolu=new JButton("/");
        M=new JButton("M");
        MR=new JButton("MR");
        MArtı=new JButton("M+");
        MEksi=new JButton("M-");
        Isaret=new JButton("+/-");
        C=new JButton("C");
        c.add(Gir);
        c.add(Artı);
        c.add(Eksi);
        c.add(Carpı);
        c.add(Bolu);
        c.add(M);
        c.add(MR);
        c.add(MArtı);
        c.add(MEksi);
        c.add(Isaret);
        c.add(C);
        Gir.addActionListener(this);
        Artı.addActionListener(this);
        Eksi.addActionListener(this);
        Carpı.addActionListener(this);
        Bolu.addActionListener(this);
        M.addActionListener(this);
        MR.addActionListener(this);
```

```

MArti.addActionListener(this);
MEksi.addActionListener(this);
Isaret.addActionListener(this);
C.addActionListener(this);
}

// girdi alanındaki olan olayları 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("");
}
}

```

10. Hesapmakinesi sınıfını için bir **JFrame** test programı yaz ve bu sınıf hesapmakinesi sınıfını kullanarak dışarıdan girilen iki gerçek sayının dört işlemini hesapla. **dortislemhesapmakinası** programı yukarıda verilmiştir.

Program 3.17 Dörtişlem hesap makinası appleti tanımlayan H4AL1SWF_2000.java programı (java komutuyla çalıştıracağız)

```

import javax.swing.*; // java swing sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import dortislemhesapmakinası;
import BasicWindowMonitor;

//dörtişlem hesap makinası sınıfını çağırır
//swing frame versiyonu
public class H4AL1SWF_2000 extends JFrame implements ActionListener
{
// Dört işlem hesap makinası
JTextField sonucgirdisi;
JTextField kutugirdisi;
JButton Gir,Arti,Eksi,Carpi,Bolu,M,MR,MArti,MEksi,Isaret,C;
//JButton sınıfı değişkenleri
double sayı;
dortislemhesapmakinası D;

```

```

// pencereyi baslatma metodu
public H4AL1SWF_2000()
{
D=new dortislemhesapmakinas();

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(sayı);
else if(e.getSource() == Bolu) D.bol(sayı);
else if(e.getSource() == M) D.M(sayı);
else if(e.getSource() == MR) D.MR();
else if(e.getSource() == MArti) D.MTopla(sayı);
else if(e.getSource() == MEksi) D.Mcikar(sayı);
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);
}
}

```

11. Bayağı kesirler sanırım ilk okulda sizleri epey zorlamıştır. Daha fazla aklımızı karıştırmalarını önlemek amacıyla **bayagikesir** sınıfını tanımlayalım. Bu sınıf bayağı kesirlerin dört işlemini ve sadeleştirilmelerini yapabilme kapasitesinde olsun. Aynı zamanda bayağı kesirleri yazdırın `toString` metotunu da tanımlayalım. Bu metot [2/3] veya 3[5/4] şeklinde bayağı kesirleri yazdırıbsın.
 Bayağı kesirleri sadeleştirirken en büyük ortak bölen hesabı gerekebilir. Bu amaçla aşağıdaki metot verilmiştir.

```

public int enbuyukortakbolen(int n, int m)
{
    // iki tam sayının en büyük 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;
}
not : sadeleştirme işlemi pay ve paydayı en büyük ortak bölene bölgerek yapılır.
Bayagi kesir sınıfının çalışmasını kontrol amacıyla bayagikesirstest.java applet programını yaz. Bu programa iki bayağı kesir gir ve dört işlemini hesapla.

```

Program 3.18 : bayagikesir.java,bayagikesir sınıfı tanımı

```

//=====
// bayagikesir sınıfı tanımı
// 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();
}
    // giriş- çıkış metodları

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()
{
//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 sayı 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 sayıyla carp
pay = pay*sag.Pay();
payda = payda*sag.Payda();
sadelestir();
}

public void carp(int sag ) throws bayagikesirException
{
//bir gercek sayıyla 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 sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
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
// aktarır
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 sayının farkını return deyimiyle bayagikesir olarak
// aktarır
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"melerini 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 sayıyi double sayıya 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()
{
// yazma hazır bayagikesir formda String deşiskeni 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 sınıfı test programı bkTesti.java

```

// Dr. Turhan Çoban
// Bayağı kesir testi
//-----
import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import bayagikesir;
public class bkTesti extends Applet implements ActionListener
{
    //
//=====
// sınıf değişkenleri
// bu değişkenler tüm sınıfa aittir
Label kutubasligi1; //Label sınıfı değişkeni (nesnesi) kutubasligi1
TextField kutugirdisi1; // TextField sınıfı değişkeni (nesnesi) kutugirdisi1
Label kutubasligi2; //Label sınıfı değişkeni (nesnesi) kutubasligi2
TextField kutugirdisi2; // TextField sınıfı değişkeni (nesnesi) kutugirdisi2
TextField kutugirdisi3; // TextField sınıfı değişkeni (nesnesi) kutugirdisi3
TextField kutugirdisi4; // TextField sınıfı değişkeni (nesnesi) kutugirdisi4
bayagikesir k1; // bayagikesir değişken k1
bayagikesir k2; // bayagikesir değişken k2
//=====
// pencereyi başlatma metodu
// pencere ilk basladığında
// değişkenler buradaki değerleri alırlar
public void init()
{
k1=new bayagikesir(1,2);
k2=new bayagikesir(2,4);
kutubasligi1=new Label("Birinci bayagikesir sayıyi 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 sayıyi giriniz : ");
add(kutubasligi2);
kutugirdisi3=new TextField(5);

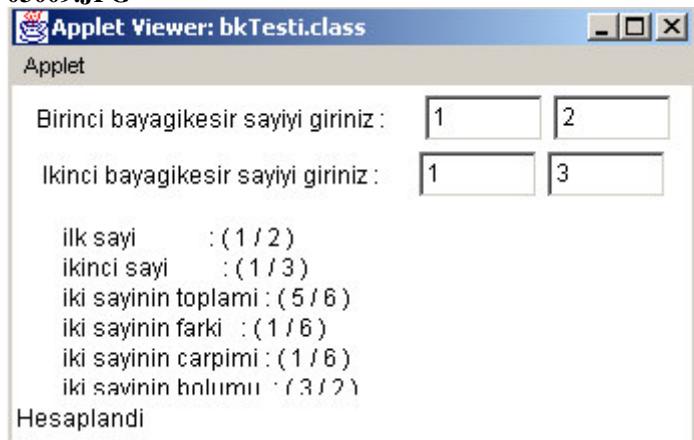
```

```

kutugirdisi3.addActionListener(this);
add(kutugirdisi3);
kutugirdisi4=new TextField(5);
kutugirdisi4.addActionListener(this);
add(kutugirdisi4);
}
//=====
// girdi alanindaki olan olaylari dinleme metodu
// Her yeni rakam girildiginde bu metot cagrilir
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 metoduna gore ciz
}
//=====
// paint (boya=pencereye ciz metodu)
public void paint(Graphics g)
{
    g.drawString("ilk sayı : "+k1.toString(),25,80);
    g.drawString("ikinci sayı : "+k2.toString(),25,95);
    try{
        g.drawString("iki sayının toplamı : "+bayagikesir.toString(bayagikesir.topla(k1,k2)),25,110);
        g.drawString("iki sayının farkı : "+bayagikesir.toString(bayagikesir.cikar(k1,k2)),25,125);
        g.drawString("iki sayının çarpımı : "+bayagikesir.toString(bayagikesir.carp(k1,k2)),25,140 );
        g.drawString("iki sayının bölümü : "+bayagikesir.toString(bayagikesir.bol(k1,k2)),25,155 );
    }
    catch (bayagikesirException e)
    {
        showStatus(e.getMessage());
    }
}
}

```

03009.JPG



Şekil 3.9 Bayağı kesir test programı bkTest.html applet çıktısı

Program 3.20 :Bayagikesir sınıfı Java swing applet test programı bkTesti_2000.java

```
import javax.swing.*; // java swing sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import bayagikesir;

public class bkTesti_2000 extends JApplet implements ActionListener
{
    //
    //=====
    // sınıf değişkenleri
    // bu değişkenler tüm sınıfa aittir
    JLabel kutubasligi1; //JLabel sınıfı değişkeni (nesnesi) kutubasligi1
    JTextField kutugirdisi1; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi1
    JLabel kutubasligi2; //JLabel sınıfı değişkeni (nesnesi) kutubasligi2
    JTextField kutugirdisi2; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextField kutugirdisi3; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextField kutugirdisi4; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextArea ciktig; // JTextArea sınıfı değişkeni (nesnesi) çıktı
    bayagikesir k1; // bayagikesir değişken k1
    bayagikesir k2; // bayagikesir değişken k2
    //
    //=====
    // pencereyi başlatma metodu
    // pencere ilk basladığında
    // değişkenler buradaki değerleri alırlar
    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 sayınızı 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 sayınızı giriniz : ");
        c.add(kutubasligi2);
        kutugirdisi3=new JTextField(5);
        kutugirdisi3.addActionListener(this);
        c.add(kutugirdisi3);
        kutugirdisi4=new JTextField(5);
        kutugirdisi4.addActionListener(this);
        c.add(kutugirdisi4);
        ciktig=new JTextField(" ");
    }
    //
    // girdi alanındaki olan olayları dinleme metodu
    // Her yeni rakam girildiğinde bu metod çağırılır
    public void actionPerformed(ActionEvent e)
    {
        Integer sayı1,sayı2,sayı3,sayı4;
        sayı1=new Integer(kutugirdisi1.getText() );
        sayı2=new Integer(kutugirdisi2.getText() );
        sayı3=new Integer(kutugirdisi3.getText() );
        sayı4=new Integer(kutugirdisi4.getText() );
```

```

        showStatus("Hesaplanıyor....");
        k1.bayagikesirGir(sayı1.intValue(),sayı2.intValue());
        k2.bayagikesirGir(sayı3.intValue(),sayı4.intValue());
        showStatus("Hesaplandı");
        String s="";
        s+="ilk sayı      : "+k1.toString()+"\n";
        s+="ikinci sayı   : "+k2.toString();
        s+="iki sayının toplamı : "+bayagikesir.toString(bayagikesir.topla(k1,k2))+"\n";
        s+="iki sayının farkı  : "+bayagikesir.toString(bayagikesir.cikar(k1,k2))+"\n";
        s+="iki sayının çarpımı : "+bayagikesir.toString(bayagikesir.carp(k1,k2))+"\n";
        s+="iki sayının bolumu : "+bayagikesir.toString(bayagikesir.bol(k1,k2))+"\n";
        cikti.setText(s);
        repaint();//pencereyi yeniden paint metoduna gore ciz
    }
//=====
}

```

03010.JPG



Şekil 3.10 bkTesti_2000 swing JApplet çıktısı

Program 3.21 :Bayagikesir sınıfı test programı bkTestiSWF_2000.java swing Frame test programı

```

import javax.swing.*; // java swing sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import bayagikesir;
import BasicWindowMonitor;

public class bkTestiSWF_2000 extends JFrame implements ActionListener
{
    //
//=====
// sınıf değişkenleri
// bu değişkenler tüm sınıfa aittir
JLabel kutubasligi1; //JLabel sınıfı değişkeni (nesnesi) kutubasligi1
JTextField kutugirdisi1; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi1
JLabel kutubasligi2; // JLabel sınıfı değişkeni (nesnesi) kutubasligi2
JTextField kutugirdisi2; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
JTextField kutugirdisi3; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi3
JTextField kutugirdisi4; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi4
JTextArea cikti; // JTextArea sınıfı değişkeni (nesnesi) cikti
bayagikesir k1; // bayagikesir değişken k1
bayagikesir k2; // bayagikesir değişken k2
//=====
// pencereyi başlatma metodu
// pencere ilk basladığında

```

```

// degiskenler buradaki degerleri alirlar
public bkTestiSWF_2000()
{
super("bayağı 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 cagrilir

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 çıktısı

Kompleks sınıfı örnek olarak burada verilmiştir. Problem 11 ve 12 ile ilgili önemli bir nokta bunların gerçek hayatı kullanabilecek sınıflar olmalarıdır. Sadece bir eksersiz olarak kullanılmak amacıyla değil, bu sınıflara ihtiyacımız olduğunda bize hizmet vermek amacıyla ve tüm detay metodları düşünülerek yazılmışlardır. Bu sınıflar her zaman kütüphanenizde kalabilecek ve gerektiğinde direkt olarak gerçek programlarınızda kullanılabilecek özelliklerde sınıflardır:

Program 3.22 : kompleks sınıfının tanımı : kompleks.java

```

=====
// kompleks sınıfı tanımı
// Dr. Turhan Coban
=====
import java.io.*;
// sınıf kompleks
// bu sınıf kompleks sayıların matematik
// işlemlerini tanımlar
//
class kompleks {
    // sınıf değişkenleri
    double ger; // kompleks sayının gerçek kısmı
    double imaj; // kompleks sayının imajineri kısmı

    // 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();
}

// giriş - çıkış metodları

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 sayıyla carp
ger = ger*sag.gercek() - imaj*sag.imajineri();
imaj = ger*sag.imajineri() + imaj*sag.gercek();
}
public void carp(double sag )
{
//bir gercek sayıyla 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 gerçek sayı  $\square$   $\square$  n toplam  $\square$   $\square$  return deyimiyle kompleks olarak
// aktar  $\square$  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 sayının sonuçlarını aktar
kompleks sonuc;
sonuc=new kompleks((sol.gercek() - sag.gercek()),
(sag.imajineri() - sag.imajineri()));
return sonuc;
}
public static kompleks cikar(kompleks sol, double sag)
{ // bir kompleks sayıdan bir gerçek sayıyı çıkartır sonuçlarını aktar
kompleks sonuc;
sonuc=new kompleks((sol.gercek() - sag),sag.imajineri());
return sonuc;
}
public static kompleks cikar(double sol, kompleks sag)
{ // bir double sayıdan bir kompleks sayıyı çıkartır sonuçlarını aktar
kompleks sonuc;
sonuc=new kompleks((sol - sag.gercek()),-sag.imajineri());
return sonuc;
}
public static kompleks carp(kompleks sol, kompleks sag)
{ // iki kompleks sayının çarpımını aktar
kompleks sonuc;
sonuc=new kompleks
((sol.gercek()*sag.gercek() - sol.imajineri()*sag.imajineri()),
(sag.gercek()*sag.imajineri() + sol.imajineri()*sag.gercek()));
return sonuc;
}
public static kompleks carp(kompleks sol, double sag)
{ // bir kompleks ve bir double sayının çarpımını aktar
kompleks sonuc;
sonuc=new kompleks(sol.gercek()*sag,sag.imajineri()*sag);
return sonuc;
}
public static kompleks carp(double sol, kompleks sag)
{ // bir kompleks ve bir double sayının çarpımını aktar
kompleks sonuc;
sonuc=new kompleks(sag.gercek()*sol,sag.imajineri()*sol);
return sonuc;
}
public static kompleks bol(kompleks sol, kompleks sag)
{ // iki kompleks sayının b'lımlerini 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 hazır kompleks formda String deşiskeni 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 sınıfının tanımının sonu

```

Program 3.21 : kompleks sınıfını test eden H8A1.java programı

```

import java.applet.Applet; // java applet sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import kompleks;
public class H8A1 extends Applet implements ActionListener
{
    //
//=====
// sınıf değişkenleri
// bu değişkenler tüm sınıflarda aittir
Label kutubasligi1; //Label sınıfı değişkeni (nesnesi) kutubasligi1
TextField kutugirdisi1; // TextField sınıfı değişkeni (nesnesi) kutugirdisi1
Label kutubasligi2; //Label sınıfı değişkeni (nesnesi) kutubasligi2
TextField kutugirdisi2; // TextField sınıfı değişkeni (nesnesi) kutugirdisi2
TextField kutugirdisi3; // TextField sınıfı değişkeni (nesnesi) kutugirdisi3
TextField kutugirdisi4; // TextField sınıfı değişkeni (nesnesi) kutugirdisi4
kompleks k1; // kompleks değişken k1
kompleks k2; // kompleks değişken k2
//=====

```

```

// pencereyi baslatma metodu
// 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 metodu
// Her yeni rakam girildiginde bu metot cagrilir
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 metodu)
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 );
}
}

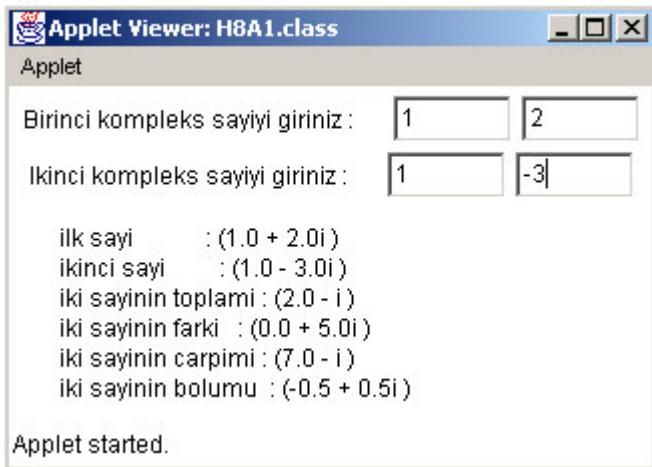
import java.applet.Applet; // java applet sinifini cagir
import java.awt.*; // java pencere kullanma sinifini cagir
import java.awt.event.*; // java pencereyi dinleme sinifini cagir
import kompleks;
public class H8A1 extends Applet implements ActionListener
{
    //
}
//=====================================================================
// sinif degiskenleri
// bu degiskenler tum snnfa 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 metodu
// 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 metodu
// Her yeni rakam girildiginde bu metot cagrilir
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 metodu)
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 );
}
}

```



Şekil 3.12 Kompleks sınıfı testi H8A1.html applet çıktısı

Program 3.23 : kompleks sınıfını test eden java swing applet programı kompleksTesti_2000.java

```

import javax.swing.*; // java swing sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import kompleks;

public class kompleksTesti_2000 extends JApplet implements ActionListener
{
    //
    //=====
    // sınıf değişkenleri
    // bu değişkenler tüm sınıfa aittir
    JLabel kutubasligi1; //JLabel sınıfı değişkeni (nesnesi) kutubasligi1
    JTextField kutugirdisi1; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi1
    JLabel kutubasligi2; //JLabel sınıfı değişkeni (nesnesi) kutubasligi2
    JTextField kutugirdisi2; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextField kutugirdisi3; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextField kutugirdisi4; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextArea cikti; // JTextArea sınıfı değişkeni (nesnesi) çıktı
    kompleks k1; // kompleks değişken k1
    kompleks k2; // kompleks değişken k2
    //=====
    // pencereyi başlatma metodu
    // pencere ilk basladığında
    // değişkenler buradaki değerleri alırlar
    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 sayıyı 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("İkinci kompleks sayıyı giriniz : ");
        c.add(kutubasligi2);
    }
}

```

```

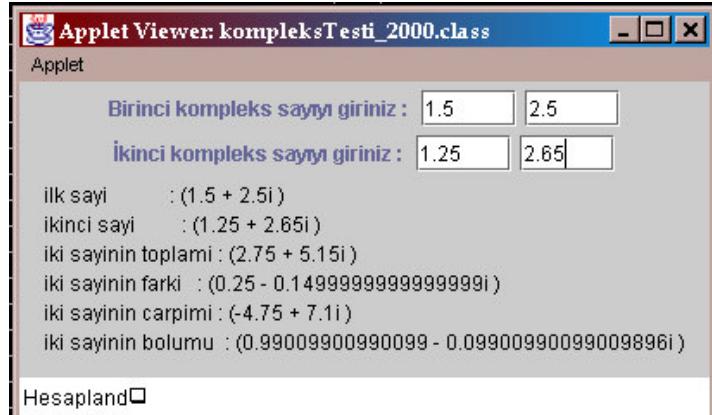
        kutugirdisi3=new JTextField(5);
        kutugirdisi3.addActionListener(this);
        c.add(kutugirdisi3);
        kutugirdisi4=new JTextField(5);
        kutugirdisi4.addActionListener(this);
        c.add(kutugirdisi4);
        cikti=new JTextArea( toString());
        cikti.setBackground(c.getBackground());
        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



Şekil 3.13 Kompleks sınıfı testi kompleksTesti_2000.html JApplet çıktısı

Program 3.24 : kompleks sınıfını test eden kompleksTestiSWF_2000.java JFrame programı

```

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; // JTextField 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 degiskeni k1
    kompleks k2; // kompleks degiskeni k2
    //=====
    // pencereyi baslatma metodu
    // pencere ilk basladiginda
    // degiskenler buradaki degerleri alirlar
    public kompleksTestiSWF_2000()
    {
        super("kompleks sayı 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 sayı      :" +k1.toString()+"\n";
        s+="ikinci sayı   :" +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



Şekil 3.14 Kompleks sınıfı testi kompleksTestiSWF_2000.html JFrame konsol çıktısı

13. Kalıtım yoluyla dortslemhesapmakinasi sınıfından türetilen bilimselhesapmakinasi1 sınıfı verilmiştir. Bu sınıfı inceleyiniz.

Program 3.25 : bilimselhesapmakinasi1.java programı

```

import dortslemhesapmakinasi;

public class bilimselhesapmakinasi1 extends dortslemhesapmakinasi
{
//bu metot miras (inheritance) yoluyla aktarma yapmaktadır.
// kurucu metot dortslemhesapmakinasi

double arasonuc;

```

```
double arasonuc2;

public bilimselhesapmakinası1(double y)
{
//constructor
super(y);
arasonuc=0;
arasonuc2=0;
}

public bilimselhesapmakinası1()
{
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 `dortislemhesapmakinası` sınıfından türetilen `bilimselhesapmakinası2` sınıfı verilmiştir. Bu sınıfı inceleyiniz. Alıştırma 10'da verilen `bilimselhesapmakinası1` programından farklarını açıklayınız.

Program 3.26 : bilimselhesapmakinası2.java programı

```

import dortislemhesapmakinası;

public class bilimselhesapmakinası2
{

```

```
//bu metot nesne (object) yoluyla aktarma yapmaktadır.  
// kurucu metot dortislemhesapmakinesi  
  
dortislemhesapmakinesi DH;  
double arasonuc;  
double arasonuc2;  
  
public bilimselhesapmakini2(double y)  
{  
//constructor  
DH=new dortislemhesapmakini(y);  
arasonuc=0;  
arasonuc2=0;  
}  
  
public bilimselhesapmakini2()  
{  
DH=new dortislemhesapmakini();  
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 hesapmakinası1 sınıfını kullanan hesap makinası simulasyon programı H4O1.java tanımlanmıştır. Bu programı inceleyiniz. Aynı işlemi bilimselhesapmakinası2 sınıfını kullanarak yapan H4O1b.java programını yazınız. Bilimselhesapmakinası1.java ve bilimselhesapmakinası2.java programlarının temel farklarını (miras-inheritance ve nesne-object yoluyla aktarma-composition) izah ediniz

Program 3.27 : H4O1.java programı

```

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 bilimselhesapmakinası1;

public class H4O1 extends Applet implements ActionListener
{
    // bilimsel hesap makinasi

    TextField sonucgirdisi;
    TextField kutugirdisi;
    Button Gir,Arti,Eksi,Carpı,Bolu,M,MR,MArtı,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 sayı;
    bilimselhesapmakinası1 D;
    // pencereyi baslatma metodu
    public void init()
    {
        D=new bilimselhesapmakinası1();
        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 ");
        Artı=new Button(" + ");
        Eksi=new Button(" - ");
        Carpı=new Button(" * ");
        Bolu=new Button(" / ");
        M=new Button("M");
        MR=new Button("MR");
        MArtı=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(Artı);
        add(Eksi);
        add(Carpı);
    }
}

```

```

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);
Carpı.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()==Carp) 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()==MArtile.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



Şekil 3.15 H4O1.html bilimsel hesap makinası simulasyonu

Program 3.28 : H4O1b.java programı, Bilimsel hesap makinası, bu programda hesap makinası düğmeleri formatlanmıştır.

```

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 bilimselhesapmakinası1;

public class H4O1b extends Applet implements ActionListener
{
    // bilimsel hesap makinası

    TextField sonucgirdisi;
    TextField kutugirdisi;
    Button Gir,Arti,Eksi,Carpı,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 simifi degiskenleri
    double sayi;
    bilimselhesapmakinası1 D;
    // pencereyi baslatma metodu
    public void init()
    {
        D=new bilimselhesapmakinası1();
        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(" - ");
        Carpı=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("xy");
        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()==MArtille.getSource()==MEksi)
{
    kutugirdisi.setText(D.MtoString());
}
else if(e.getSource()==Cos ||
e.getSource()==Sin||e.getSource()==Tan||e.getSource()==RtoD ||

```

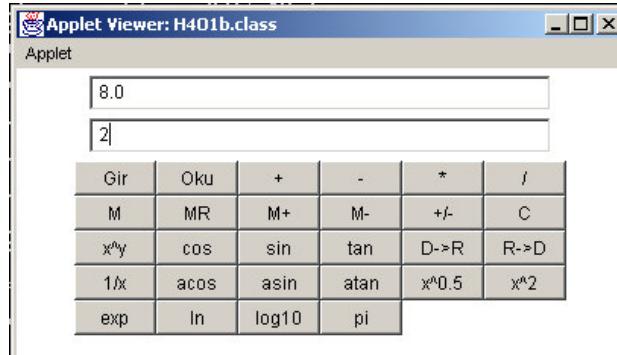
```

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



Şekil 3.16 H4O1b.html bilimsel hesap makinasi simulasyonu. Bu program üsttekinin ayndır,yalnız düğmeleri formattı bir şekilde yerleştirilmiştir. Bu format (layout) işlemini daha sonraki bölümlerde detaylı olarak inceleyeceğiz.

Program 3.29 : H4O1bSW.java programı, Bilimsel hesap makinasi, bu programda hesap makinasi düğmeleri formatlanmıştır ve temel olarak swing applet sınıfı kullanılmıştır.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import bilimselhesapmakinası1;

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 sınıfı degiskenleri
    double sayı;
    bilimselhesapmakinası1 D;
    // pencereyi baslatma metodu

    public void init()
    {
        D=new bilimselhesapmakinası1();
        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(" - ");
Carpı=new JButton(" * ");
Bolu=new JButton(" / ");
M=new JButton("M");
MR=new JButton("MR");
MArtı=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(Artı);
m.add(Eksi);
m.add(Carpı);
m.add(Bolu);
m.add(M);
m.add(MR);
m.add(MArtı);
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);
Carpı.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()==Carpı) 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.In(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() == MArtille.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 Makinası 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



Şekil 3.17 H4O1bSW.html bilimsel hesap makinası simulasyonu. Bu program üsttekinin aynıdır,yalnız Applet yerine Swing Applet Japplet kullanılmıştır.

14. H4O2

[nokta3D.java](#) ve [dogru3D.java](#) programlarını inceleyiniz. Bu programlardan yararlanarak [vektor3D.java](#) programını (vektor sınıfını) oluşturunuz. Testprogramı H4O2.java'da aynı başlangıç noktalı iki vektörü toplatınız.

Not : vektörler doğrular gibi iki nokta ve bir fiziksel boyut (uzay veya başka bir fiziksel boyut olabilir) ile tanımlanırlar. bir yönleri mevcuttur (daima P0 dan P1 e doğru).

Program 3.30 : H4O2.java programı

```
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 noktasını gir : ");
        xi=cin.readDouble();
        xj=cin.readDouble();
        xk=cin.readDouble();
        n1=new nokta3D(xi,xj,xk);
        System.out.print("vektor deşerini gir : ");
        Fi=cin.readDouble();
        Fj=cin.readDouble();
        Fk=cin.readDouble();
        v1=new vektor3D(n1,Fi,Fj,Fk);
        System.out.print("vektor deşerini 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 değeri . "+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 programı

```
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 arasindali mesafe
return Math.sqrt(xi*xi+xj*xj+xk*xk);
}

public double cosai()
```

```

{
//noktanın i ekseniyle yaptıgı açının cosinusunu
return xi/R();
}

public double cosaj()
{
//noktanın j ekseniyle yaptıgı açının cosinusunu
return xj/R();
}

public double cosak()
{
//noktanın k ekseniyle yaptıgı açının cosinusunu
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 programı

```

import nokta3D;

public class dogru3D
{
public nokta3D P0,P1;
double xi,xj,xk;

//***** kurucu metodlar 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();
}

```

*/****** sənəf deşiyken giriyi ..Gir (veya set..) metodları*

```

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);
}

```

*/****** sənəf deşiyken şəkər Yə ..Oku veya get.. metodları*

```

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;
}

/**sınıf islem metodları

public double R()
{
//iki nokta arasındaki mesafe
return Math.sqrt(xi*xi+xj*xj+xk*xk);;

}

public double cosai()
{
//noktanın i eksenile yaptığı açının kosinüsü
return xi/R();;

}

public double cosaj()
{
//noktanın j eksenile yaptığı açının kosinüsü
return xj/R();;

}

public double cosak()
{
//noktanın k eksenile yaptığı açının kosinüsü
return xk/R();;

}

/**sınıf mantıksal karşılaştırma metodları

public boolean esittir(dogr3D 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());
}

//***** sınıf string çıktı metodu 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 programı

```

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) noktasından 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 doğrudan gecen vektorun tanımı
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 noktasından baslayan vektor tanımla
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 noktasından baslayan vektor
super(y0);
Fi=Fi0;
Fj=Fj0;
Fk=Fk0;
}

public vektor3D(nokta3D y0,nokta3D y1,double F)
{
//iki noktadan gecen vektorun tanımı
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 büyüklüğü
return Math.sqrt(Fi*Fi+Fj*Fj+Fk*Fk);
}

//not R() metodu, vektorun merkezden uzaklığını nokta3D de tanımlanmıştır.
// tekrar tanımı gerekmeyez.
/*
public double R()
{
//koordinat merkezi ile nokta arasındaki mesafe
return Math.sqrt(xi*xi+xj*xj+xk*xk);
}
*/

public double cosFi()
{
//vektorun i ekseniyle yaptığı açının kosinüsü
return Fi/F();
}

public double cosFj()
{
//vektorun j ekseniyle yaptığı açının kosinüsü
return Fj/F();
}

public double cosFk()
{
//vektorun k ekseniyle yaptığı açının 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 noktası : "+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** programlarını inceleyiniz. Bu programlardan yararlanarak **vektor3D.java** programını (vektor sınıfını) oluşturunuz. Java swing JFrame Test programı H4OD3_2000.java'da aynı başlangıç noktalı iki vektörü toplatınız.

Not : vektorler doğrular gibi iki nokta ve bir fiziksel boyut (uzay veya başka bir fiziksel boyut olabilir) ile tanımlanırlar. bir yönleri mevcuttur (daima P0 dan P1 e doğru).

Program 3.34 : vektor3D sınıfını test eden swing JFrame, H4OD3_2000.java programı

```

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 ciktig; // JTextArea sınıfı değişkeni (nesnesi) çıktı
    vektor3D k[]={new vektor3D[2]; // vector3D boyutlu değişken
    vektorpanel p[]={new vektorpanel[2]; //vektorPanel değişken 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

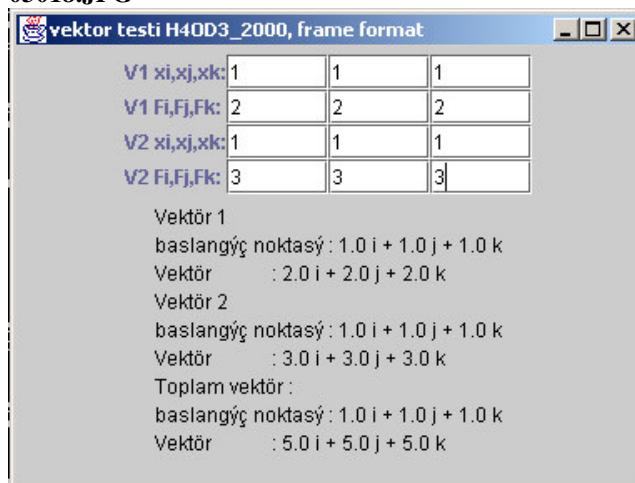
public void actionPerformed(ActionEvent e)
{
    for(int i=0;i<2;i++)
        k[i].vektorGir(Double.parseDouble(p[i].x[0].getText()),
                      Double.parseDouble(p[i].x[1].getText()),
                      Double.parseDouble(p[i].x[2].getText()),
                      Double.parseDouble(p[i].F[0].getText()),
                      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



Şekil 3.18 vektor3D sınıfını test eden H4OD3_2000 java frame programı

yukarıdaki programda kullanılan alt sınıf vektorpanel.java

Program 3.35 : vektor3D sınıfını test eden swing JFrame, H4OD3_2000.java programında kullanılan vektorpanel alt sınıfı (vektorpanel.java programı)

```
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 geçtiğini biliyoruz. düzlemin formülü $z=a*x+b*y+c$ dir (a,b,c sabit). 3 noktadan geçen düzlemi tanımlayan `duzlem.java` programını ve `duzlem` sınıfını yazınız.
not: birim vektorler x_i , x_j y doğrultusunda, x_k z doğrultusundadır. `H4O3.java` konsol türü örnek test programında yarattığınız bir düzlem nesnesinin(`object`) başlangıç kordinatlarına uzaklığını (başlangıç noktasından geçen ve düzleme dik olan bir doğrunun boyunu) hesaplayınız.

Program 3.36 : `vektor3D.java`, `nokta3D.java` test programı `H4O3.java`

```

import duzlem;
import vektor3D;
import nokta3D;

class H4O3
{
    public static void main(String args[])
    {
//bu test düzlem ile nokta arasındaki 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.scalararp(v1)/v.F();
        System.out.println("duzlem d = "+d.toString());
        System.out.println("nokta P0 = "+P0.toString());
        System.out.println("duzlem(d) ile nokta arasındaki mesafe:"+sonuc);
    }
}

```

duzlem d = z = -2.33333333333335 + 1.66666666666667x + 0.666666666666666y
nokta P0 = 2.0 i - 1.0 j + 3.0 k
duzlem(d) ile nokta arasindaki mesafe:1.6222142113076254

- **H4OD4_2000**

kompleks sınıfına kompleks sayının karekökünü hesaplayacak

public static kompleks karekok(kompleks sol)

metodunu ekleyiniz. Bir test programı yazarak (**H4OD4_2000.java**) girdığınız kompleks sayının karesini hesaplatınız.

not : **public static kompleks pow(kompleks sol)** metodu kompleks sayının üssünü hesaplar.

```
public static kompleks karekok(kompleks sol)
{
    return pow(sol,0.5);
}
```

Program 3.37 : vektor3D.java, nokta3D.java test programı H4O3.java

```
import javax.swing.*; // java swing sınıfını çağır
import java.awt.*; // java pencere kullanma sınıfını çağır
import java.awt.event.*; // java pencereyi dinleme sınıfını çağır
import kompleks;
import BasicWindowMonitor;

public class H4OD4_2000 extends JFrame implements ActionListener
{
    //
    //=====
    // sınıf değişkenleri
    // bu değişkenler tüm sınıfa aittir
    JLabel kutubasligi1; //JLabel sınıfı değişkeni (nesnesi) kutubasligi1
    JTextField kutugirdisi1; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi1
    JTextField kutugirdisi2; // JTextField sınıfı değişkeni (nesnesi) kutugirdisi2
    JTextArea cikti; // JTextArea sınıfı değişkeni (nesnesi) çıktı
    kompleks k1; // kompleks değişken k1
    //
    //=====
    // pencereyi başlatma metodu
    // pencere ilk basladığında
    // değişkenler buradaki değerleri alırlar
    public H4OD4_2000()
    {
        super("kompleks sayı karekök testi H4OD4_2000, frame format");
        k1=new kompleks(1.0,1.0);
        Container c=getContentPane();
        c.setLayout(new FlowLayout());
        kutubasligi1=new JLabel("kompleks sayıyı 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 sayı : "+k1.toString()+"\n";
s+="kompleks sayının karekökü : "+kompleks.toString(kompleks.karekok(k1))+"\n";
s+="kompleks sayının karesi : "+kompleks.toString(kompleks.kare(k1))+"\n";
return s;
}
// girdi alanindaki olan olaylari dinleme metodu
// Her yeni rakam girildiginde bu metod cagrilir

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 DEĞİŞKENLER VE NESNELER (ARRAYS)

4.1 TEK BOYUTLU DEĞİŞKENLER

Boyutlu değişkenler aynı değişken türü veya sınıfından birden fazla nesne veya değişkenin tek bir isimle tanımlanmasıdır. Boyutlu değişkenlere referans indeks numarası kullanılarak ulaşılır. Genel olarak bir boyutlu değişkenler

Değişken_türü değişken_ismi[];

veya

Değişken_türü[] değişken_ismi;

Şeklinde tanımlanır. Birinci tanımın kullanılması daha yaygındır. İki tanım arasında bilgisayar açısından bir fark mevcut değildir. Bir örnek verecek olursak

```
int ayin_gunleri[]; veya  
int[] ayin_gunleri;
```

Bu tanımlar her ne kadar haftanın_gunleri nin boyutlu bir değişken olduğunu belirtiyorsa da kesin boyutunu bildirmemektedir. Boyutlu değişkenin tam boyutunu tanımlamak ve bilgisayar hafızasındaki yerini tam olarak saptamak için

Değişken_ismi=new Değişken_türü[değişken_boyutu];

Terimi kullanılır. Bunu bir önceki örneğe uygularsak :

```
Ayin_gunleri=new int[12];
```

Boyutlu değişkeni bir kere tanımladıktan sonra onun içindeki her alt değere indeksi üzerinden ulaşmak mümkündür.

Öregin:

```
ayin_gunleri[0] = 31;  
ayin_gunleri[1] = 28;
```

gibi. . İndeks değişkeni her zaman sıfırdan başlar.

Küçük bir örnek sınıfta bu kavramı daha açık olarak vermeye çalışalım.

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 ayı "+ayin_gunleri[3]+" gün ceker");
}
}

```

Bu programı daha kısa olarak yazmak için Şekil 4.1.2 de görülen şekilde bütün değişkenlerin değerlerini aynı anda tanımlayabiliriz.

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 ayı "+ayin_gunleri[3]+" gün ceker");
}
}

```

Program 4.3'de boyut kavramının for döngüsüyle birlikte kullanışını göriyorsunuz. Boyutlu değişkenin toplam boyutuna da **sayı.length** değişkeniyle ulaşıyoruz. Daha önceki birçok programlama dilinde bu mümkün değildir ve boyutlu değişkenin boyutunun bildirilmesi gereklidir. Program 4.3 ün çıktısını step step takip ederek ve bir hesap makinası kullanarak hesaplayınız.

Program 4.3 : Aritmetik2.java

```

import java.io.*; //java girdi çıktı sınıfını çağır
class Aritmetik2
{
public static void main(String args[])
{
int sayı[]={10,5,7,9,11,13,14,18};
int toplam=0;
int i;
for(i=0;i<sayı.length;i++)
{
toplam+=sayı[i];
}
System.out.println("Ortalama = "+toplam/sayı.length);
}
}

```

Şimdi de bir applet örneğinde tek boyutlu değişkenleri kullanalım. Daha önce zarApplet.java programını incelemiştik. Şimdi bu programa her bir zar yüzeyinin atılış frekanslarını ekleyelim. Toplam altı yüz olduğundan birden altıya kadar her sayının kaç kere geldiğini saydırımız gereklidir. Bunun için bir boyutlu **zarfrekansi** değişkenini tanımlayacağız.

Program listesi ve sonuç appleti Program 4.4 ve şekil 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 düğmesi

```

```

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));
toplum+=(zar1+zar2);
toplumzaratisi++;
// not boyutlu degisken indeksi 0 dan basliyor.
zarfrekansi[zar1-1]++;
zarfrekansi[zar2-1]++;
repaint();
}
public void paint( Graphics g)
{
g.drawString("toplum = "+toplum+" Atilan zar sayisi = "+toplumzaratisi,25,50);
for(int i=0;i<6;i++)
{
g.drawString(zarfrekansi[i]+" kere "+(i+1)+" atildi ",25,(65+15*i));
}
}
}

```

04001.JPG



Şekil 4.1 ZarfrekansiApplet.java appletinin görünümü

zarfrekansiApplet.java programının bir de swing eşdeğeri zarfrekansiSWF.java swing Frame programını verelim :

Program 4.5 : zarfrekansiSWF, java swing JFrame programı

```

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 düğmesi
    JTextArea ciktig;
    int zar1,zar2;
    int zarfrekansi[]; 

    public static int zar()
    {
        return 1+(int)(Math.random()*6);
    }

    public zarfrekansiSWF()
    {
        //programı baslat
        super("Zar frekansı swing JFrame");
        zarfrekansi=new int[6];
        Container c=getContentPane();
        c.setLayout(new FlowLayout());
        ilkzar=new JTextField(10); // JTextField sınıfı ilkzar nesnesini yarat
        c.add(ilkzar);           // ilk zar nesnesini pencereye ekle
        ikincizar=new JTextField(10); // JTextField sınıfı ikincizar nesnesini yarat
        c.add(ikincizar);         // ikinci zar nesnesini pencereye ekle
        salla=new JButton("Zarı salla ve at");
        c.add(salla);
        salla.addActionListener(this);
        ciktig=new JTextArea();
        ciktig.setBackground(c.getBackground());
        c.add(ciktig);
    }

    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 basiyor.
        zarfrekansi[zar1-1]++;
        zarfrekansi[zar2-1]++;
        s+="toplam = "+toplam+" Atilan zar sayısı = "+toplamzaratisi+"\n";
        for(int i=0;i<6;i++)
        {
            s+=zarfrekansi[i]+" kere "+(i+1)+" atıldı \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



Şekil 4.2 ZarfrekansiAppletSWF.java appletinin görünümü

4.2 TEK BOYUTLU NESNE TİPİ DEĞİŞKENLER

Nesne tipi değişkenler için de aynı normal basit değişkenlerde olduğu gibi boyut tanımı yapılır. Normal boyut tanımı yapıldıktan sonra, boyuttaki her nesne teker teker tanımlanır. Örneğin :

```

kompleks sayı[]=new kompleks[5];
sayı[0]=new kompleks(1,2);
sayı[1]=new kompleks(1,-1);
sayı[2]=new kompleks(2,0);
sayı[3]=new kompleks(1.1,-0.5);
sayı[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.

4.3 ÇOK BOYUTLU DEĞİŞKENLER

İki ve daha fazla boyutlu değişkenler de tanımlamak mümkündür. İki boyutlu değişkenler en fazla tablo veya matris gibi satır ve sütun olmak üzere iki boyutta gösterilmesi gereken büyülükleri oluşturmak için kullanılır. Java iki boyutlu değişkenleri direk olarak açamaz. Tek boyutlu değişkenlerin yine tek boyutlu değişkenini açar. Sonuç olarak iki boyut sağlanmış olur. Örnek verecek olursak

Int b[][]=new int[3][4]. Genel olarak ilk parantes satır sayısı, ikinci parantez sütun sayısı olarak kabul edilir. Buna göre b değişkenini şöyle düşünebiliriz :

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]

İki boyutlu değişken de bir boyutlu değişkenlerde olduğu gibi değerleri direk olarak yükleyebiliriz, ve yükleme sırasında boyutları da tayin edebiliriz.

Örneğin :

```
Int b[][] = {{1,2},{3,4}};
```

Terimi bize aşağıdaki tabloyu tanımlar:

1	2
3	4

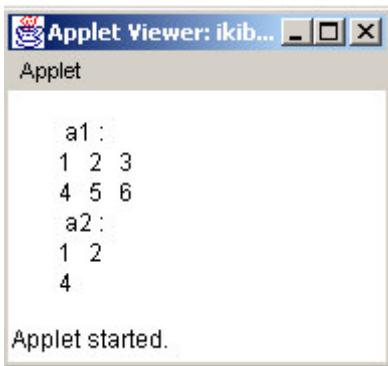
b[0][0]	b[0][1]
b[1][0]	b[1][1]

İki boyutlu değişkenlerin kullanılmasını göstermek amacıyla bir örnek problem verelim:

Program 4.6 : İki boyutlu değişkenler örneği : ikiboyut.java

```
import java.awt.Graphics;
import java.applet.Applet;
public class ikiboyut extends Applet
{
// void tipi paint metodu
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



Şekil 4.3 İki boyutlu değişkenler örneği [ikiboyut.html](#) dosyasının appletviewer ile görünümü

Aynı programı swing JOptionPane çıktıları olarak da tanımlayalım :

Program 4.7 : İki boyutlu değişkenler örneği : ikiboyut_2000.java

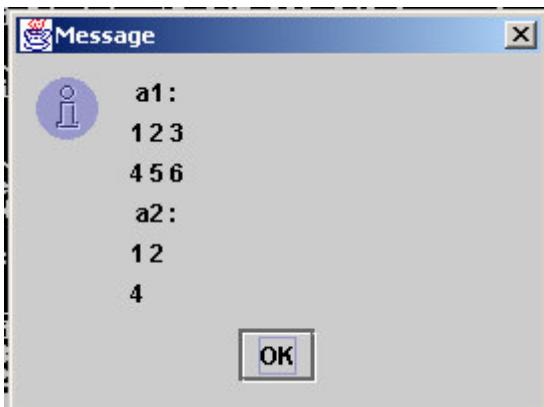
import javax.swing.JOptionPane;

```
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+=ikiboyutyazdir(a1);
        s+=" a2 :\n";
        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 çıktıları :

04004.JPG



Şekil 4.4 İki boyutlu değişkenler örneği [ikiboyut_2000.html](#) dosyasının JOptionPane çıktısı olarak görünümü

Aynı programın swing JFrame versiyonunu da inceleyelim :

Program 4.8 : İki boyutlu değişkenler örneği : 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 değişken örneği");
        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+=ikiboyutyazdir(a1);
        s+=" a2 :\n";
        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());
```

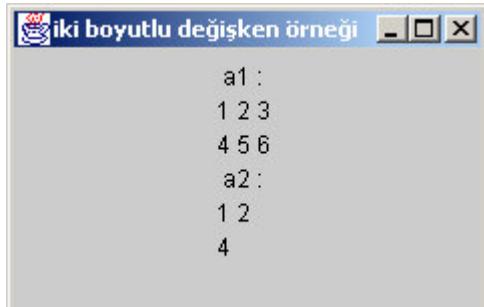
```

        pencere.setSize(200,200);
        pencere.setVisible(true);
    }
}

```

Swing frame çıktısı :

04005.JPG



Şekil 4.5 İki boyutlu değişkenler örneği [ikiboyutSWF_2000.html](#) doyasının JFrame çıktısı olarak görünümü

Boylara ulaşmak için yukarıdaki programlardan da görüleceği gibi for döngüsü kullandık. Boyutlu değişkenlere ulaşım. for döngülerinin en çok kullanıldığı yerlerden biridir. Satırların boyutuna **a.length**, sütunların boyutuna ise **a[satır].length** terimiyle ulaştık. Bu tanım bize toplam satır ve sütun boyutunu direk tanımlar, ve kullanımda yeterince esneklik sağlar.

4.4 BOYUTLU DEĞİŞKENLERİN METOTLARA AKTARIMI

Boylu değişkenler aynen boyutsuz değişkenler gibi metotlara aktarılırlar. Örneğin eğer
int sıcaklık[] = new int[24];
deyimiyle tanımlanan sıcaklık;
sıcaklığıdegistir(sıcaklık);
terimiyle sıcaklığıdegistir metotuna aktarılabilir. Metot tanımlanırken, sıcaklığın boyutlu değişken olduğu tanımlanmalıdır:

```

double void sıcaklığıdegistir(int sıcaklık[])
{
.....
}

```

gibi.

Metotların çıktı değişkenleri de çok boyutlu olarak tanımlanabilir ve kullanılabilir. Örneğin :
public static double[][] inversematris(double[][] a) gibi.

4.5 BOYUTLU DEĞİŞKENLERDE BOYUT DEĞİŞTİRME

Boylu değişkenlerin boyutları ilk tanımlamalarında belirtildiğinden normal olarak değiştirilemez. Ancak indirek yollarla boyutun değiştirilmesi mümkündür. Bu yol önce yeni boyutta bir boyutlu değişken tanımlamak, sonra boyutlu değişkenin içerisindeki değerleri bu yeni değişkene aktarmak ve sonra orijinal boyutlu değişkenin adresini yeni oluşturulan boyutlu değişken adresiyle değiştirmek şeklinde yapılır.

```

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 parçasığında a değişkeninin boyutu üçten 4 e değiştirilmiştir. Daha detaylı bir örnekle boyut değiştirmeyi inceleyelim.

Program 4.9 : boyutlu değiştirme örneği : doubleBoyut.java

```
public class doubleBoyut
{
    //sınıf değişkenleri
    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 sınıfında boyutu değiştirilebilen değişken (nesne) tanımladık. Bu değişkeni boyutDegistir sınıfında test edelim :

Program 4.10 : boyutlu değiştirme testi : doubleBoyut sınıfını 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();
}
}

```

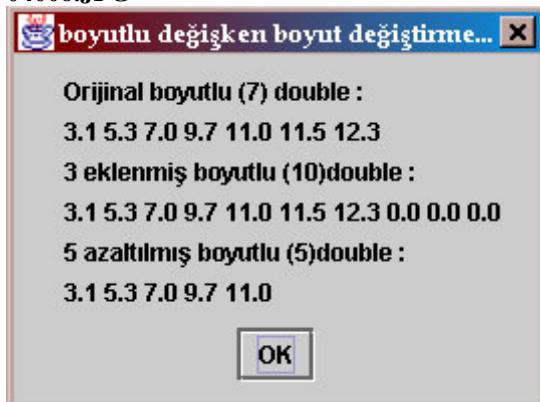
```

y.boyutEkle(3);
s+="3 eklenmiş boyutlu ("+y.length+)double : \n";
s+=y.toString();
y.boyutAzalt(5);
s+="5 azaltılmış boyutlu ("+y.length+)double : \n";
s+=y.toString();
JOptionPane.showMessageDialog(null,s,
"boyutlu değişken boyut değiştirme eksersizi",
JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```

Program çıktısı :

04006.JPG



Şekil 4.6 boyut değiştirme örneği boyutu değiştirilebilen doubleBoyut sınıfının boyutDegistir sınıfında JOptionPane çıktısı olarak görünümü

4.6 ALIŞTIRMALAR

1. nokta3 ve dogru3 sınıflarının tanımları verilmiştir. Bu tanımları kullanarak konsol ortamında 5 adet doğru3 tanımlayınız. Doğruların başlangıç ve bitiş noktalarını konsol ortamından giriniz. ve doğruların boyalarının ortalamasını konsol ortamında ekrana yazdırınız.

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());
    }
    //giriŞ çıkış metotları
    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());
    }
    // giriş çıkış metotları
    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 düzlemindeki açı
    return Math.atan2(Y(),X());
}
public double ac❑_n1_xz()
{
    // dogrunun n1 noktasından x z düzlemindeki açısı
}

```

```

// radyan
return Math.atan2(Z(),X());
}
public double acn_n1_yz()
{
//dogrunun n1 noktasndan y z düzlemindeki açı 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();
}
}

```

2. Sizin için kompleks sınıfı oluşturulmuş ve bu sınıfın nasıl kullanıldığını açıklayan H8A1 applet programı verilmiştir. Bu iki programı incele. Ayrıca Aritmetik2 programını incele ve boyutlu değişkenlerin kullanımını öğren. Sonra 5 kompleks sayının toplamını hesaplayan bir konsol programı yaz.

Program 4.13 H8A1.java programı

```

import java.io.*; //java girdi çıktı sınıfını çağır
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 programı

```

import java.io.*; //java girdi çıktı sınıfını çağır
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 sayı 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 = "+toplams.toString());
}
}

```

bir kompleks sayı giriniz : 1 1
 bir kompleks sayı giriniz : 2 2
 bir kompleks sayı giriniz : 3 3
 bir kompleks sayı giriniz : 4 4
 bir kompleks sayı giriniz : 5 5
 Ortalama = (3.0 + 3.0i)

3. Üç boyutlu nokta ve doğru tanımları nokta3 ve doğru3 sınıflarında verilmiştir. 3 tane doğruya boyutlu değişken olarak **H8A2** sınıfında tanımlayın ve doğruların toplam boyalarını ve baslangıç noktasının xy xz ve yz düzlemlerinde yaptığı açıları derece cinsinden yazdırın.

Program 4.15 H8A1.java programı

```

import java.io.*; //java girdi çıktı sınıfını çağır
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
```

12. H5AL3.java programı. . Programı inceleyiniz.

Program 4.16 H5AL3.java programı

```
//bu sınıf sınıfı degiskeni kompleksi boyutlu olarak çağırır  
import java.io.*; //java girdi çıktı sınıfını çağır  
import kompleks;  
  
class H5AL3  
{  
    public static void main(String args[]) throws IOException  
    {  
        Text cin=new Text();  
        double n1,n2;  
        System.out.print("kompleks sayıların miktarı, 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 sayı 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());  
    }  
}
```

13. **H5O1** : java konsol programında toplam bayağı kesir sayısını sorduktan sonra ekrandan verilen sayıda bayağı kesiri giriniz ve ortalamalarını hesaplayınız.

Program 4.17 H5O1.java programı

```
//bu sınıf sınıfı degiskeni kompleksi boyutlu olarak çağırır  
import java.io.*; //java girdi çıktı sınıfını çağır  
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 sayısı, 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());
}
}

```

14. **H5O3**

Üç boyutlu n kompleks sayıyı java konsol programında toplam kompleks sayıların sayısını sorduktan sonra ekranдан verilen sayıda kompleks sayıyı giriniz. Kompleks sayılar girildikten sonra, kompleks sayıların ortalamasını hesaplayınız.

Program 4.18 H5O3.java programı

```

//bu sınıf sınıfı degiskeni kompleksi boyutlu olarak çağırır
import java.io.*; //java girdi çıktı sınıfını çağır
import kompleks;

```

```

class H5O3
{
public static void main(String args[]) throws IOException
{
Text cin=new Text();
double n1,n2;
System.out.print("toplam kompleks sayı, 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 sayıyi 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 programını inceleyiniz. Java konsol programında (Text sınıfını kullanabilirsiniz) toplam kompleks sayısını sorduktan sonra ekrandan verilen sayıda kompleks sayıyı giriniz ve kareköklerinin ortalamalarını hesaplayınız.

Program 4.19 H5OD1_2000.java programı

```

//bu sınıf sınıfı degiskeni kompleksi boyutlu olarak çağırır
import java.io.*; //java girdi çıktı sınıfını çağır
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 sayı, 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 sayı 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 sayıların kareköklerinin toplamı = "+toplam.toString());
toplam.bol(sayi.length);
System.out.println("kompleks sayıların kareköklerinin ortalaması = "+toplam.toString());
}
}

```

15. H5OD2_2000

Double boyutlu sayıların boyutlarını değiştirek kullanabilen doubleBoyut sınıfı yukarıda tanımlanmıştır. Integer (tamsayı) değişkenlerin boyutlarını değiştirek kullanabileceğimiz intBoyut sınıfını oluşturunuz. Bir test programı yazınız. Bu programda bir değişkene 5 boyutlu bir integer değişken gurubu yükleyiniz, sonra boyutu 6 ya çıkarınız ve sonradan boyutu 4 e düşürünüz. Sonuçları yazdırınız.

Program 4.20 intBoyut.java programı

```
public class intBoyut
{
    //sınıf değişkenleri
    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 programı

```

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();
}

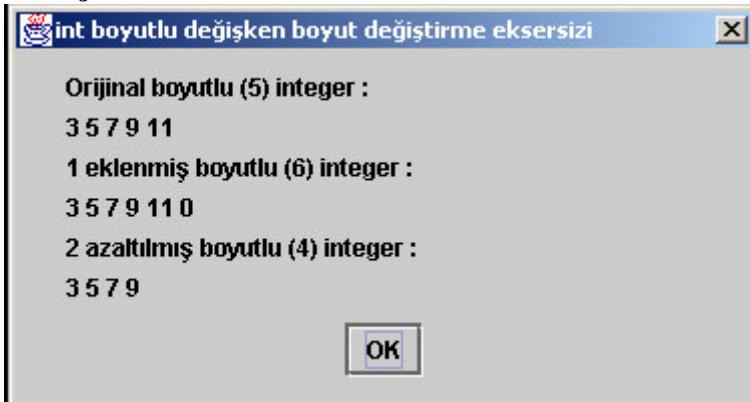
```

```

s+="1 eklenmiş boyutlu ("+y.length+) integer : \n";
s+=y.toString();
y.boyutAzalt(2);
s+="2 azaltılmış boyutlu ("+y.length+) integer : \n";
s+=y.toString();
JOptionPane.showMessageDialog(null,s,
"int boyutlu değişken boyut değiştirme eksersizi",
JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```

04007.JPG



Şekil 4.7 intBoyut, boyutlu tam sayı boyut değiştirme eksersizi, H5OD2_2000.java, JOptionPane çıktısı

16. H5OD3_2000

```

String isim;
String Soysisim;
int not;

```

sınıf değişkenleri ve gerekli metodları kapsayan “ogrenci” sınıfını yaratınız. Sınıftaki öğrenci sayısını sorup, daha sonra tüm öğrenciler için öğrenci ismi, soyismi, ve notunu giriniz (sınıfnızda minimum üç öğrenci olsun). Çıktı alanında öğrencilerin isim soyisim ve not listesini sıraladıktan sonra sınıf ortalamasını da verin.

Program 4.22 ogrenci.java programı

```

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 programı

```

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("sınıf Ortalaması = " + toplam);
}
```

H5OD5_2000.java programının çıktısı :

Turhan Çoban 25
Meral Çoban 75
Ali Velizade 99
sınıf Ortalaması = 66.33333333333333

BÖLÜM 5 : SINIFLARDA HİYERARSİ, ABSTRACT SINIF VE INTERFACE

5.1 ABSTRACT SINIF

Üçüncü bölümde sınıfların diğer sınıflardan kalıtım (Inheritance) yoluyla türetilmesini görmüştük. Sınıfları kalıtım yoluyla birbirinden türetirken eğer ortak özellikli sınıflar yaratıyorsak, bütün bu sınıf gurubunun en tepesine abstract bir sınıf koyabiliriz. Abstract sınıf diğer sınıfların kullanılmasında extends yoluyla tepe sınıflık etmek ve bütün alt sınıflara ortak bir adres çıkış noktası sağlamak dışında bir görevi yoktur. Diğer bir deyimle bu sınıflar hiçbir zaman doğrudan kullanılmazlar. Ancak alt sınıflarından birini referans olarak göstermek ve onlara dolaylı yoldan ulaşmak amacıyla kullanılırlar.

Kavramı daha iyi verebilmek için bir örnek problem oluşturalım. Daha önce sınıf kavramını açıklarken kullandığımız nokta, daire sınıflarını hatırliyacaksınız. Şimdi bu sınıfların üzerine sekil isimli bir abstract sınıf ekleyerek tekrar oluşturalım.

Program 5.1 Abstract sınıf sekil (sekilX.java dosyasında yer alıyor)

```
public abstract class sekilX
{
    //not burada başka değişkenlerde olabilir
    public double alan(){return 0.0;}
    public double hacim() {return 0.0;}
    public abstract String isim();
}
```

Program 5.2: sekil sınıfından kalıtım yoluyla türetilen noktaX sınıfı

```
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 sınıfından kalıtım yoluyla türetilen daireX sınıfı

```
import noktaX;  
  
public class daireX extends noktaX  
{  
    protected double yaricap;  
  
    public daireX()  
    {  
        //daire kalıtım yaptığı nokta sınıfının kurucu metodunu  
        // super deyimi ile çağırabilir.  
        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 sınıfından kalıtım yoluyla türetilen silindirX sınıfı

```
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 sınıf ve indirek referanslamayı test eden abstracttesti sınıfı

```

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);
}

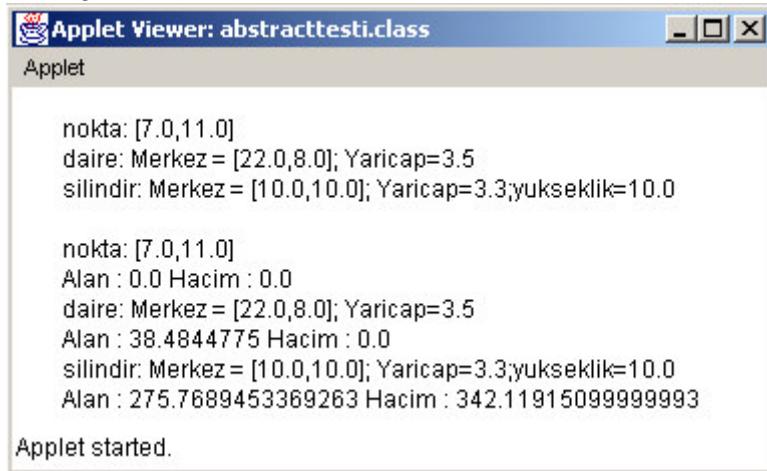
```

```

g.drawString(d.isim()+" : "+d.toString(),25,40);
g.drawString(s.isim()+" : "+s.toString(),25,55);
//indirek cagırma
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



Şekil 5.1 Abstract sınıf ve indirek referanslamayı test eden abstractTesti sınıfının sonuçlarının applet de görülmesi

Program 5.5 : Abstract sınıf ve indirek referanslamayı 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";

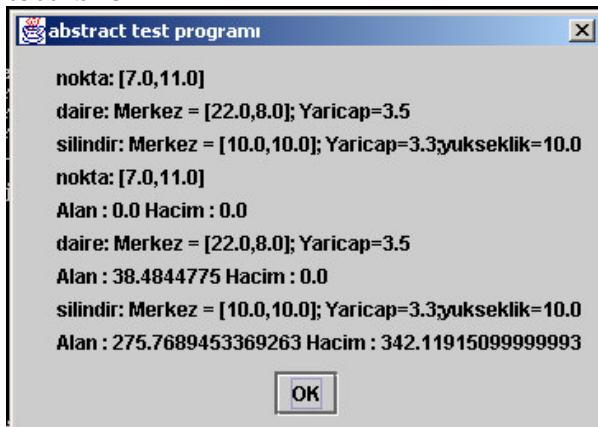
```

```

st+=d.isim()+": "+d.toString()+"\n";
st+=s.isim()+": "+s.toString()+"\n";
//indirek çağrıma
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 programı",
JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```

05002.JPG



Şekil 5.2 Abstract sınıf ve indirek referanslamayı test eden abstractTesti_2000 sınıfının sonuclarının JOptionPane çıktısı olarak görülmesi

Örnekten de görüldüğü gibi nokta, daire ve silindir abstract sekil sınıfının alt sınıfları olarak oluşturulmuş, ve aynı zamanda bu sınıflara indirek referans olarak kullanılmıştır. Diğer bir deyimle nokta daire ve silindirin adresleri sekil sınıfı nesneye aktarılmış ve bu nesne üzerinden üç sınıfta indirek olarak kullanılmıştır. Bu indirek kullanımın bize temel faydası, Programımıza yeni sınıflar eklediğimizde daha kolay ve az program değiştirerek uyumu sağlayabilme yeteneğidir. Orneğin yukarıdaki program sınıfları gurubuna koni eklemek istersek bütün yapacağımız koni sınıfını tanımladıktan sonra, sekil nesnesi a nın boyutunu bir arttırarak aynı hesaplamaya koniyide ilave etmekten ibaret olacaktır. Bu tür uygulamalar bilgisayar kodunun değişim gereksiminini minimuma indirgediinden nesne kokenli bilgisayar programcılığının önemli uygulamalarından biridir.

5.2 INTERFACE

Abstract sınıfların dezavantajı sadece bir sınıfın kalıtım yoluyla ve extends sözcüğü kullanılarak bağlanabilir olmasıdır. Ve eğer başka bir sınıf halihazırda o sınıfa kalıtım yoluyla bağlanmışsa, bizim yeni bir sınıfı üst sınıf olarak kullanma olasılığımız yok olmuş demektir. Fakat bazı uygulamalar için bu gerekebilir. Bu zaman **interface** kullanınız. Interface de abstract sınıf gibi kendi içinde bir işlem yapmaz. Sadece öbür sınıfların bağlanmasıını teşkil eden bir adres noktası olarak iş görür. Interface’ı bir başka sınıfa bağlamak gerektiğinde extends yerine **implements** sözcüğü kullanılır. Abstract sınıflar kendi başına bir bütün teşkil edebilmelerine rağmen interface’ın sadece kendi başına bir anlamı yoktur. Interface sınıfında yer alan tüm metod ve değişkenlerin alt sınıflarda tanımlanması gereklidir.

Örnek olarak bir önceki problemi interface ile tanımlıyalım.

Program 5.5 interface sekil ([sekilY.java](#) dosyasında yer alıyor)

```

public interface sekilY
{
//burada deşıiÝken veya nesnelerde olabilir

```

```
public abstract double alan();
public abstract double hacim();
public abstract String isim();
}
```

Program 5.6 : interface sekilden implements kelimesiyle türeyen noktaY sınıfı ([noktaY.java](#) dosyasında yer alıyor)

```
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";
}
}
```

Gördüğü gibi burada nokta sınıfının bir öncekinden farklı implements kelimesinin kullanılmış olması ve boş alan ve hacim metodlarının bu dosyada yazılmasının gereklmesinden ibarettir. Interface programla yapılan ve alt sınımlarda bu metodların (veya değişkenlerin) tanımlanacağına dair yapılan bir kontrattır. DaireY ve silindirY sınımlarında bir fark mevcut degildir. İki seklin işlemsel sonuçları arasında da bir fark mevcut değildir. Yukardaki örneğe parel olarak burada da daireY,SilindirY tanımlarını yapalım ve interfaceTesti_2000.java programında test edelim.

Program 5.6 : interface sekilY'den implements kelimesiyle türeyen noktaY sınıfından extends yoluya türeyen daireY sınıfı([daireY.java](#) dosyasında yer alıyor)

```
import noktaY;

public class daireY extends noktaY
{
protected double yaricap;

public daireY()
{
//daire kalıtım yaptığı nokta sınıfının 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 sınıfından extends yoluya türeyen daireY sınıfından extends yoluya türeyen silindirY sınıfı([silindirY.java](#) dosyasında yer alıyor)

```

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 sınıfları test eden ve indirek olarak referanslıyan interfaceTesti_2000.java programı

```

import javax.swing.JOptionPane;
import silindirY;
import noktaY;
import daireY;

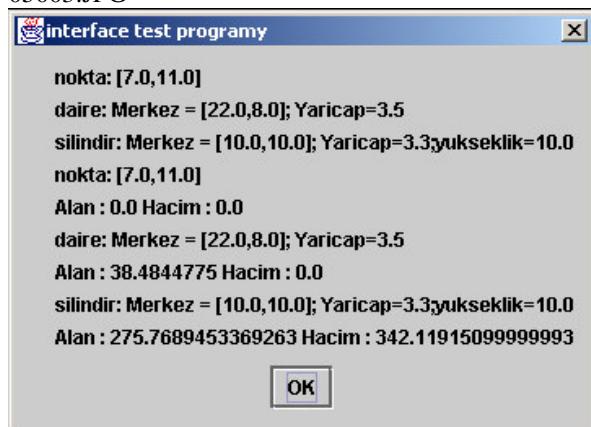
public class interfaceTesti_2000
{

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 çağrıma
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 programı",
JOptionPane.PLAIN_MESSAGE);
}

```

```
System.exit(0);
}
}
```

05003.JPG



Şekil 5.2 interface sınıf ve indirek referanslamayı test eden interfaceTesti_2000 sınıfının sonuçlarının JOptionPane çıktıları olarak görülmesi

5.3 ALIŞTIRMALAR

- Derste gördüğümüz **abstract class sekilX**, **noktaX**, **daireX**, **silindirX** sınıflarına ve ilave olarak **koniX** sınıfını yaratınız. Bu sınıfların **sekilstestiX.java** test sınıfında alanlarını ve hacimlerini **sekilX** referansını kullanarak yazdırınız.
- Derste gördüğümüz **interface sekilY**, **noktaY**, **daireY**, **silindirY** sınıflarına ve ilave olarak **küreY** sınıfını yaratınız. Bu sınıfların **sekilstesti.java** testY sınıfında alanlarını ve hacimlerini **sekilY** referansını kullanarak yazdırınız.
- H5AL1.java** programını inceleyiniz. Bu program bir boyutlu abstract sınıf **fx** ve bu sınıfta tanımlanan abstract metot **f(double x)** i kullanan turev ve integral metodlarını içerir. Integral ve türev metodlarını niçin abstract kavramı kullanarak yazdığınıizi açıklayınız.

Program 5.7 H5AL1.java, fx ve H5AL1 sınıflarının tanımı

```
//abstract sınıf ve boyutlu değişkenler listirmasi

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 t̄revini hesaplar
        // metodun hata miktarı h0 and n degiskenlerinin degisimi ile
        // degistirilebilir. fonksiyon girişi fx abstract s̄n̄f̄ ̄zerinden
        // yapılmalıdır.
        // Bu problem boyutlu dēiskenerin kullan̄lması ile ilgili de bir
        // örnektir.
        double h0=0.0256;
        int i,m;
        int n=7;
        //basit bir fonksiyonun t̄revi
        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=f1.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=f1.f(x);
z+=(b-a)/2.0*w[i]*y;
}
return z;
}

```

4. H5AL2.java programı alıştırma 3 de verilen turev ve integral sınıflarını konsol ortamında kullanmaktadır. Programın 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[]);
}
```

tanımını ve H5AL1 sınıfındaki turev metoduna H5O2 sınıfında

```
public static double turev(fx fi,double x)
```

gerekli değişiklikleri yaparak

```
public static double turev(fxi fi,double x[],int i)
```

metotunu oluşturunuz ve i^{inci} x değeri için denklemin türevini hesaplayınız.

($\frac{df_i(x[])}{dx_i}$ türevini hesaplayınız.)

Program 5.9 H5O2.java

```
//abstract sınıf ve boyutlu değişkenler 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 t̄revini hesaplar
// metotun hata miktarı h0 and n degiskenlerinin degisimi ile
// degistirilebilir. fonksiyon girisi fxi abstract s̄n̄f̄ ̄zerinden
// yapilmalidir.
// Bu problem boyutlu de§iskenerin kullan̄lmasi ile ilgili de bir
// ornekтир.
double h0=0.0256;
int i,m;
int n=7;
//basit bir fonksiyonun t̄revi
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=f1.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=f1.f(x);
z+=(b-a)/2.0*w[i]*y;
}
x[k]=tempxk;
return z;
}
}

```

6. Program 5.10 da verilen polar sınıfını inceleyiniz. Polar sınıfının kompleks sınıfıyla oldukça benzer olduğunu gözlemleyebilirsiniz. Bu iki sınıfın temel özelliklerinden yararlanan abstract bir sınıf oluşturunuz, bu sınıfın alt sınıfları olarak polar ve kompleks sınıflarını yeniden tanımlayınız. Bir örnek programda abstract sınıf üzerinden polar ve kompleks sınıfları çağırarak kullanınız.

Program 5.10 Polar sınıfı tanımı 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 teta)
{
R=Ri;
teta=teta;
}

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 : GRAFİKLER, FONTLAR VE RENKLER

6.1 GİRİŞ

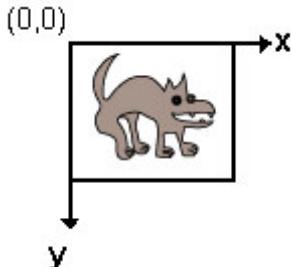
Daha önce applet çizerken **java.awt** kütüphanesindeki grafik (Graphics) sınıfını kullanmıştık. Ve grafik (Graphics) sınıfının ait `drawString` metodunu applete yazı yazdırmıştık. Şimdi bu sınıfın diğer özelliklerini ve metodlarını öğreneceğiz.

Graphics sınıfı `java.lang.Object` sınıfının altında yer alır. Bu gurupta aynı zamanda grafik çiziminin çeşitli fazlarında kullanılan `java.awt.Color`, `java.awt.Font`, `java.awt.FontMetrics`, `java.awt`, `java.awt.Graphics`, `java.awt.Polygon` yer alır. Daha detaylı grafik çizme işlemleri tanımlayabilen `java.awt`. `Graphics2D` paketi `java.awt.Graphics` paketinden türetilmiştir. Ayrıca `java.awt.BasicStroke`, `java.awt.GradientPaint`, `java.awt.TexturePaint` gibi temel grafik alt programları da `java.lang.Object` altında tanımlanmıştır. Temel şekilleri çizdirmeye yarayan `java.awt.geom.GeneralPath`, `java.awt.geom.Line2D`, `java.awt.geom.RectangularShape` de `java.lang.Object` altında tanımlanır. `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 tanımlanmıştır.

Graphics sınıfı, yazı yazma, çizgi çizme, dikdörtgen, oval çizme gibi bir dizi metodu barındırır. Bunun dışında awt kütüphanesinin önemli bir sınıfı da `Color` sınıfıdır. `Color` sınıfı rekle ri tanımlar ve değişik renklerin kullanımına imkan verir. `Font` sınıfı grafikte kullanılan yazıların fontlarının (yazı tipinin ve boyutunun) belirlenmesi amacıyla kullanılır. `FontMetrics` sınıfı fontların boyutlarının belirlenmesiyle ilgili metodları içerir. `Polygon` sınıfı `Polygon` çizimiyle ilgili metodları barındırır. Bu metodların bir kısmını burada göreceğiz. Temel olarak `Graphics` ve `Graphics2D` sınıflarını grafik çiziminde kullanacağız, Bu işlemi yaparken de diğer tüm yardımcı grafik sınıflarından yararlanacağız. `Graphics2D` sınıfı daha yeni olarak tanımlanmış ve `Graphics` sınıfından türetilmiş bir sınıfır. Fakat çok daha kompleks grafik çizim kapasitelerini barındırır.

Java appletlerinde çizim yaparken ilk hatırlamamız gereken nokta, koordinat merkezinin ekranın sol üst köşesi olduğudur. X eksenleri sola doğru, Y eksenleri aşağıya doğru gider.

06001.JPG



Şekil 6.1 Java grafik koordinat sistemi, x ve y eksenleri

Y ekseninin alta doğru gitmesi konsol programlarında (ve ilk konsol programlama ortamı olan satır yazıcılarında) satırların alta doğru ilerlemesi gibi tarihi bir nedene dayanmaktadır. Bütün bilgisayar diileri ve ekranı grafik programlamalarında standartdır.

6.2 GRAPHICS VE GRAPHICS2D SINIFLARI

Graphics sınıfı grafik çizimi için gerekli olan bir çok metodu içinde barındırır. Grafik sınıfı `java.lang.Object` sınıfının alt sınıflarındandır. Grafik çizme amacıyla genellikle Graphics sınıfından bir nesne `paint` veya `paintComponent` metodunun içinde çağırılır. Paint metodunun tanımı şöyledir

```
public void paint (Graphics g)
```

Bu tanım Applet, Japplet, JFrame extend etmiş programlardan çağrılabılır. Paint metodу tüm bu üç sınıfın türetildiği Component sınıfında tanımlanmıştır. javax.swing sınıfı altında tanımlanan JComponent sınıfında ise `paintComponent` metodу tanımlanmıştır. JComponent'in alt metodу JPanel bu metodу kullanır. Tanımı `paint` metoduna benzer

public void paintComponent (Graphics g)

paint veya paintComponent metodları Applet, Japplet, Frame, JFrame, Panel, JPanel sınıfları tarafından direkt olarak program açıldığında çağırılır. Kullanıcı tarafından çağrılmaz. Tekrar kullanıcı tarafından çağrılmaması gerektiğinde ise **repaint** metodu kullanılır ve repaint metodu üzerinden indirek olarak paint metodu çağrılr. Bu metodun tanımı :

public void repaint ()

şeklindedir.

İkinci ilginç metod update metodudur. Bu metod

public void update (Graphics g)

şeklinde tanımlanmıştır. Update metodu çağrıldığında Graphics metodu sistem tarafından otomatik olarak gönderilir. Graphics2D sınıfı Graphics sınıfının bir alt sınıfı olarak tanımlanmıştır. Bu sınıf Graphicss sınıfına göre çok daha kompleks çizimler yapabilen metodlara sahiptir. Burada graphics ve Graphics2D de yer alan bazı sınıf ve çizim tekniklerini yakından inceleyeceğiz.

6.1.1 drawString, drawChars ve drawBytes metodları

Graphics sınıfının altmetodları olan bu metodların tanımları şu şekilde verilmiştir.

```
public abstract void drawString( String stringYazi, int x,int y)
public void drawChars( char charYazi[], int baslangicindeksi,int yazılacakharfsayısı, int x,int y)
public void drawBytes( byte byteYazi[],
int başlangicindeksi,int yazılacakharfsayısı, int x,int y)
```

drawString metodu bir string değişkenini verilen x ve y koordinatlarından başlayarak çizer.

DrawChars bir boyutlu Char tipi değişken dizisini başlangicindeksi indeksinden baslayarak yazılacakharfsayı kadar kısmını x ve y koordinatlarından başlıyarak çizer. DrawByte metodu da drawChars metodu gibidir tek değişikliği Byte türü boyutlu değişken kullanmasıdır.

Program 6.1. de bu metodların kullanılmasını açıklayan bir program verilmiştir.

Program 6.1 : ciz.java programı : drawString,drawChars,drawBytes metodlarının kullanılması

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

public class ciz extends Applet
{
    private String s ="Bunu drawString Metoduyla yazdır";
    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 metodlarını kullanan ciz programının Applet çıktısı

Şimdi aynı işlemi ve aynı metodları Graphics2D üzerinden yaptıralım :

Program 6.2 : ciz2DSWF_2000.java programı : drawString,drawChars,drawBytes metodlarının swing JFrame ve Graphics2D ile birlikte kullanılması

```
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 yazdır";
    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);
    }

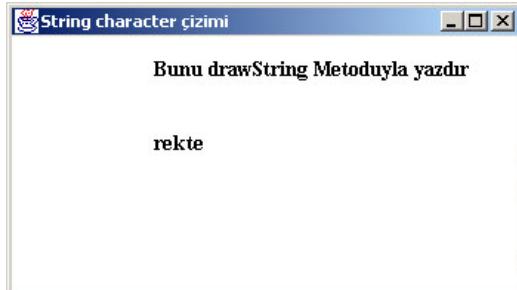
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D)g;
        g2.setFont(new Font("TimesRoman",Font.BOLD,14));
        g2.drawString(s,100,50);
        //ikinci charecterden başlayarak 5 character yaz.
        g2.drawChars(c,2,5,100,100);

    }

//=====
    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 sınıflarını kullanırken Graphics2D sınıfları kullanmak zorunda değiliz, ama burada genel olarak swing ve Graphics2D kullanımlarını birlikte sunacağız. Bu programın çıktısı da bir önceki programın çıktısında olduğu gibidir :

0603.JPG



Şekil 6.3 drawString drawChars metotlarını kullanan ciz2DSWF_2000 programının swing JFrame çıktısı

Programdan da görüleceği gibi paint metodu yine Graphics cinsinden çağrılmakta, daha sonra **Graphics2D g2 = (Graphics2D)g;** Kullanılarak adres Graphics2D sınıfından nesneye yüklenmektedir.

6.3 RENK KONTROLÜ

Appletlerde renk kontrolu **Color** sınıfını kullanarak yapılır. Renk olusumunda monitör ve Tv de kullanılan **RGB** (**Red-Green-Blue=Kırmızı-Yeşil-Mavi**) sistemi temel alınır. Bu üç renk monitorde 0-255 arası değerler alabilir. Üç rengin karışımı toplam 256*256*256 renk tanımlar. Color sınıfı kurucu metodları şu şekilde tanımlanır.

```
public Color( int kirmizi,int yesil,int mavi ) //her renk 0-255 arası
public Color(float kirmizi,float yesil,float mavi) //her renk 0.0-1.0 arası
Color sınıfında ayrıca renk değerlerini okuyabildiğimiz
public int getRed() // Kırmızı tonunu oku
public int getGreen()//Yeşil tonunu oku
public int getBlue()//Mavi tonunu oku
public abstract Color getColor() //renyi oku
metotları ve rengi değiştirebildiğimiz
public abstract Color setColor(Color c)
metotu mevcuttur. Color sınıfında sabit olarak tanımlanmış :
```

Tablo 6.1 Color sınıfında tanımlanmış sabit renkler:

Renk sabiti	Renk	RGB değeri
Public final static Color orange	portakal	255,200,0
Public final static Color pink	Pembe	255,175,175
public final static Color cyan	camgöbeği	0,255,255
public final static Color magenta	Mor	255,0,255
public final static Color yellow	Sarı	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	Açık gri	192,192,192
public final static Color darkGray	Koyu Gri	64,64,64
public final static Color red	kırmızı	255,0,0
public final static Color green	Yeşil	0,255,0
public final static Color blue	mavi	0,0,255

Color sınıfının daha iyi anlaşılabilmesi amacıyla Color sınıfının bir alt sınıfı olan renk sınıfını tanımladık.

Program 6.3 : Color sınıfının alt sınıfı renk sınıfını tanımlayan renk.java programı :

```
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 sınıfını bir çok yerde Color sınıfı yerine kullanabiliriz. Kullanamadığımız durumlarda cast işlemcisi (Color) kullanılabilir.

Applette renk değiştirdiğimiz küçük bir örnek program yazalım :

Program 6.4 : renkTest.java

```

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

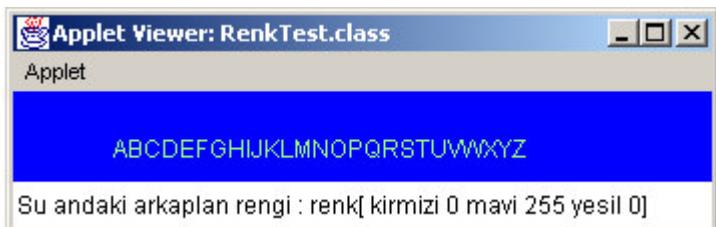
public class RenkTest extends Applet
{
private int kirmizi,yesil,mavi;

public void init()
{
kirmizi=150;
yesil=255;
mavi=125;
// arka planın rengi mavi olarak veriliyor
setBackground(renk.mavi);
}

public void paint(Graphics g)
{
// yazıının rengi 100,255,125 olarak alındı
g.setColor(new renk(kirmizi,yesil,mavi));
g.drawString("ABCDEFGHIJKLMNPQRSTUVWXYZ",50,33);
showStatus("Su andaki arkaplan rengi : "+getBackground());
}
}

```

06004.JPG



Şekil 6.4 Renk.html çıktısı

Bu programda kırmızı değeri 100, yeşil değeri 255 ve mavi değeri de 125 alınmış ve setColor deyi̇imiyle yeni renk tanımlanmıştır. Arka plan rengi ise **public void setBackground(Color c)** Metotunu kullanarak maviye dönüştürülmüştür.
Aynı programın JApplet eşidi olan, aynı zamanda java komutuyla kendi çerçevesini de oluşturabilen bir eşdegerini verelim. Renk2D sınıfında Graphics2D sınıfı kullanılmaktadır. Grafik kalitesini karşılaştıralırsınız.

Program 6.5 : Renk2D.java programı

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;

public class Renk2D extends JApplet {

    private int kirmizi,yesil,mavi;

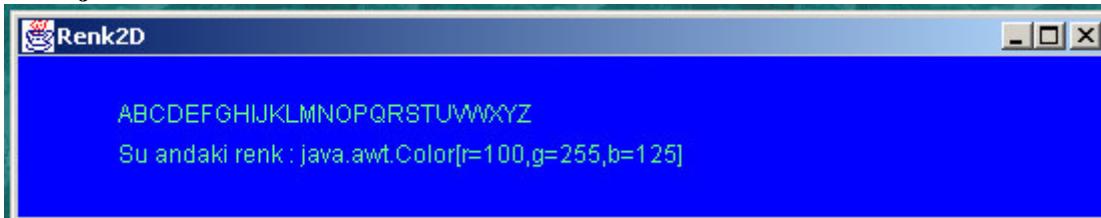
    public void init()
    {
        kirmizi=100;
        yesil=255;
        mavi=125;
        setBackground(new Color(0,0,255));
        setForeground(new Color(kirmizi,yesil,mavi));
    }

    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 sınıfının çıktısı :

06005.JPG



Şekil 6.4 Renk2D.java çıktısı

Swing sınıfında renk seçimini yapabilmek amacıyla JColorChooser adlı bir program tanımlanmıştır. Bu programı kullanarak renk grafik kutusunun rengini değiştirebildiğimiz renkSeciciSWF swing JFrame programı aşağıda verilmiştir.

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;

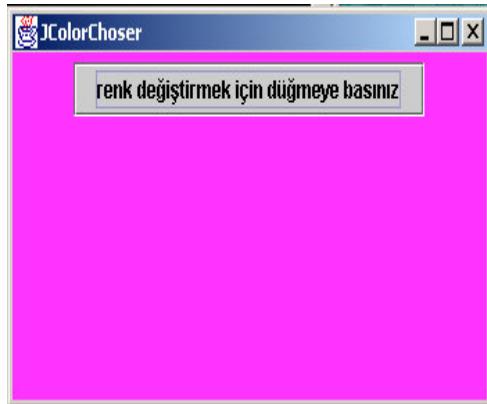
    // pencereyi baslatma metodu
    public renkSeciciSWF_2000()
    {
        super("JColorChoser ");
        c=getContentPane();
        c.setLayout(new FlowLayout());
        renkdegistir=new JButton("renk degistirmek için düğmeye basınız");
        renkdegistir.addActionListener(this);
        c.add(renkdegistir);
    }

    // girdi alanındaki olan olayları dinleme metodu
    public void actionPerformed(ActionEvent e)
    {
        r=JColorChooser.showDialog(this,"Renk seçiniz :",r);
        if(r==null)
            r=renk.acikgri;
        c.setBackground(r);
        repaint();
    }

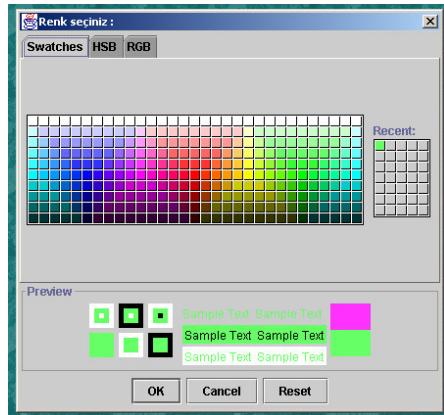
    //=====
    public static void main(String[] args)
    {
        //ana program
        renkSeciciSWF_2000 pencere= new renkSeciciSWF_2000();
        pencere.addWindowListener(new BasicWindowMonitor());
        pencere.setSize(350,200);
        pencere.setVisible(true);
    }
}
```

06006.JPG

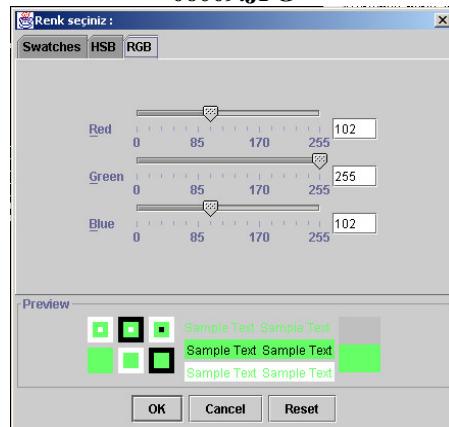
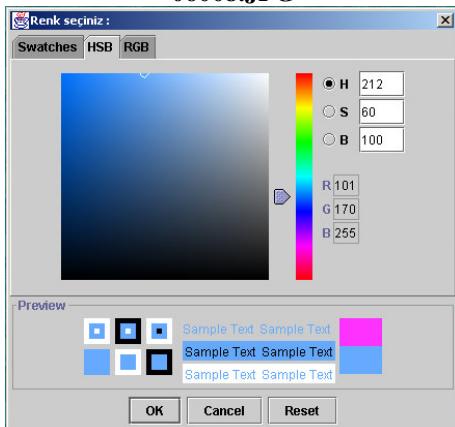
06007.JPG



06008.JPG



06009.JPG



Şekil 6.6-6.9 renkSeciciSWF_2000.java java swing frame programı ve JcolorChooser renk seçme programı

Renk seciciSWF_2000.java programında temel olarak JFrame grafik çıktıları kullanılmıştır. Bu programın içerisinde çağırılan

```
r=JColorChooser.showDialog(this,"Renk seciniz :",r);
```

metodu java kütüphanesinde yer almaktadır. ShowDialog metodu bize Color sınıfındaki renk değişkenini aktarmakta ve
c.setBackground(r);
komutuyla arka plan renklerini değiştirmektedir.

JFrame ve JApplet arasındaki farkı anlamaya biraz daha katkıda bulunması amacıyla aynı programı 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();
```

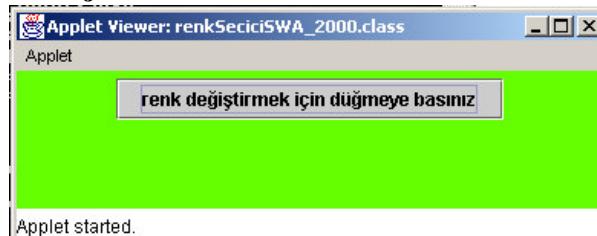
```

c.setLayout(new FlowLayout());
renkdegistir=new JButton("renk değiştirmek için düğmeye basınız");
renkdegistir.addActionListener(this);
c.add(renkdegistir);
}
// girdi alanındaki olan olayları 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 programlarının en büyük farkı JFrame ana programdan (main) çalıştığı halde JApplet'in bir browser programından çalıştırılıyor olmasıdır. Burada da en dış çerçeveye dışında JFrame ile aynı görüntüyü elde ederiz.

06010.JPG



Şekil 6.10 renkSeciciSWA_2000.java programının Appletviewer'daki çıktısı

6.4 YAZI KONTROLU

Appletlerde yazı kontrolu **Font** sınıfını kullanarak yapılır. Şekil 5.3.1 de java 1.1.5 de kullanılabilen tüm yazı stillerinin (font) listesini veren stil.java programını görüyoruz.

Program 6.8 : [stil.java](#) programı, java kütüphanesinde mevcut olan yazı 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



Şekil 6.11 stil.java programı, java kütüphanesinde mevcut olan yazı sitillerini gösterir applet programının sonuçları.

Yine aynı programın swing JApplet eşitini verelim :

Program 6.9 : [stil2D.java](#) programı, java kütüphanesinde mevcut olan yazı 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



Şekil 6.12 : [stil2D.java](#) programı çıktısı

Bu yazı sitillerinden birisi aşağıda verilen yazı tiplerinden birisiyle beraber ve verilen belli bir yazı boyutunda gösterilebilir.

public final static int PLAIN : düz yazı sitili
public final static int BOLD : Kalın yazı sitili
public static int ITALIC : İtalik (yatık) yazı sitili

bu tanımlardaki **final** sözcüğü bu ifadelerin değişken değil **sabit** olduğunu gösterir. Font sınıfının kurucu metodu şu şekildedir.

Public Font(String s, int yazı_sitili, int harf_boyutu);

Bir appletin yazı sitili Graphics sınıfındaki setFont metotuyla değiştirilebilir.

Public abstract void setFont(Font f)

Örnek olarak aşağıdaki deyimi verebiliriz :

g.setFont(new Font("Serif",Font.BOLD,12);

Şimdi setFont metotunu küçük bir örnekte kullanarak gösterelim :

Program 6.10 : [RenkFont.java](#) programı, java kütüphanesindeki Font uygulamasını 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("ABCDEFGHIJKLMNPQRSTUVWXYZ",50,33);  
showStatus("Su andaki font : "+g.getFont());  
}  
}
```

06013.JPG



Şekil 6.13 RenkFont.html programı, java kütüphanesindeki Font uygulamasını gösterir

bu programın da Japplet eşdeğerini verelim:

Program 6.11 : [RenkFont2D.java](#) swing JApplet programı, java kütüphanesindeki Font uygulamasını 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



Şekil 6.14 RenkFont2D.java JFrame programı, java kütüphanesindeki Font uygulamasını gösterir

6.5 ÇİZGİ ÇİZİMİ

Javada çizgi çizmek için kullanılan temel metot Graphics sınıfında **drawLine** metotudur. Bu metot Graphics sınıfında tanımlanmıştır. Kurucu metodu :

```
Public abstract void drawLine(  
int x1, // ilk noktanın x koordinatı  
int y1, //ilk noktanın y koordinatı  
int x2, // ikinci noktanın x koordinatı  
int y2) //ikinci noktanın y koordinatı
```

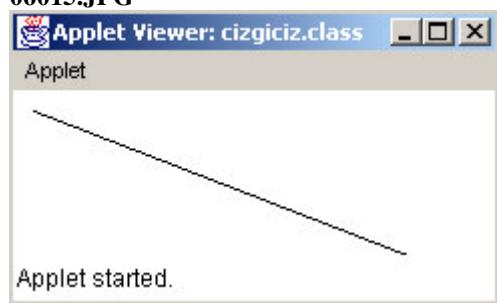
Graphics2D sınıfında ise çizgi çiziminde java.awt.geom kütüphanesinde tanımlanan Line2D sınıfı, Graphics2D sınıfındaki draw metoduyla birlikte kullanılır.

Bir küçük örnek problem [cizgiciz.java](#)'da drawLine metotunun kullanımını görelim:

Program 6.12 : Çizgi çizim örneği 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



Şekil 6.15 ve çizginin appletde görünümü

Şimdi aynı problemin yaklaşık eşidini cizgiSWF_2000.java programında görelim :

Program 6.13 : Çizgi çizim örneği 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



Şekil 6.16 ve çizginin cizgiSWF_2000 JFrame çıktısında görünümü

Bu programdaki temel çizim metodlarına göz atalım :

Graphics2D g2 = (Graphics2D)g;

Graphics sınıfı g nesnesini Graphics2D sınıfı g2 nesnesine aktarır.

g2.setPaint(Color.black);

Grafik çizim rengini siyah olarak tanımlar.

```
Line2D l=new Line2D.Double(50,50,100,75);
```

İlk devamlı çizginin koordinatlarını tanımlar

```
g2.draw(l);
```

İlk devamlı çizgiyi çizer

```
g2.setStroke(new BasicStroke(1f,BasicStroke.CAP_ROUND,  
BasicStroke.JOIN_ROUND,1f,new float[] {5f},0f));
```

Çizgiyi noktalı çizme komutunu tanımlar

```
Line2D l1=new Line2D.Double(50,150,100,175);
```

Noktalı çizginin koordinatlarını tanımlar

```
g2.draw(l1);
```

Noktalı çizgiyi çizer.

Burada bizim için yeni bir kavram olan noktalı çizgi çizmenin detaylarına biraz göz atalım. Noktalı çizgi oluşturmak için temel olarak setStroke metodunu kullanıyoruz. Bu metodun içinde Basic Stroke metodu bir nesne yaratıyoruz. BasicStroke türü nesne çizgi kalınlığı, çizgi bağlama sitili, çizgi ucu bitiş sitili ve çizgi-boşluk bilgisi içerir. Çizgi kalınlığı çizgi istikametine dik olarak ölçülür. Çizgi kalınlığı 1f (buradaki f sayının float temel değişken türü olduğunu belirtir) 0.35277 milimetreye eşittir. Yalnız transformasyon kullanıldığında bu değişimler olabilir. Transformasyonlara biraz sonra değineceğiz.

Çizgi bağlama sitili iki çizginin birbirine bağlandığı noktadaki şeklini belirler. BasicStroke sınıfında 3 bağlama sitili mevcuttur. Bunlar :



JOIN_BEVEL



JOIN_MITER

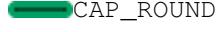


JOIN_ROUND

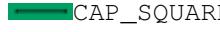
sitilleridir. Çizgi ucu bitiş sitilleri boş çizgi ucunun şeklini belirtir. Yine temel olarak 3 BasicStroke çizgi ucu bitiş sitili mevcuttur. Bunlar :



CAP_BUTT



CAP_ROUND



CAP_SQUARE

sitilleridir. Çizgi boşluk bilgisi birbirini takip eden şeffaf ve görünür çizgi parçacıkları ve bunların boyalarını tanımlar. Bir ara boşluk değişkeni ve çizgi boşluk çizgi boşluk olmak üzere bir seriden oluşur. Son olarak son rakam serinin hangi elemanından çizime başlanacağını gösterir. Örneğin:

```
g2.setStroke(new BasicStroke(1,BasicStroke.CAP_ROUND,
```

```
BasicStroke.JOIN_ROUND,0,new float[] {10,5,2,5},0));
```

10 boyutu bir çizgi 5 boşluk 2 boyutu çizgi 5 boşluk belirtir.

```
g2.setStroke(new BasicStroke(1,BasicStroke.CAP_ROUND,
```

```
BasicStroke.JOIN_ROUND,5,new float[] {5},0));
```

5 boşluk ve 5 çizgi belirtir. cizgiSWF_2000.java programında çeşitli stroke tiplerini deneyebilirsiniz.

6.6 DİKDÖRTGEN ÇİZİMİ

Java Graphics sınıfında dikdörtgen çizmek için kullanılan temel metod **drawRect** metotudur. Bu metod Graphics sınıfında tanımlanmıştır. Bu metodların tanımı :

```
Public abstract void drawRect(  
int x1, // üst tepe noktanın x koordinatı  
int y1, // üst tepe noktanın y koordinatı  
int en, // dikdörgenin genişliği  
int yükseklik) // dikdörgenin yüksekliği
```

Bu metoda ilave olarak dikdörtgenin içini verilen renkte boyayan

```
Public abstract void fillRect(  
int x1, // üst tepe noktanın x koordinatı  
int y1, // üst tepe noktanın y koordinatı  
int en, // dikdörgenin genişliği  
int yükseklik) // dikdörgenin yüksekliği
```

ve dikdörtgeni arkaplan renginde boyayan (silen)

```
Public abstract void clearRect(  
int x1, // üst tepe noktanın x koordinatı  
int y1, // üst tepe noktanın y koordinatı  
int en, // dikdörgenin genişliği  
int yükseklik) // dikdörgenin yüksekliği
```

metotları mevcuttur. DrawRect ve fillRec metodunun kullanımı Program 6.6 da verilmiştir.

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



Şekil 6.17 dikdortgenCiz.java programı ve dikdortgenCiz.html appleti

Graphics2D metodunda dikdörtgen çizmek için
g2.draw(new Rectangle2D.Double(x, y,dikdortgenEn,dikdortgenYükseklik));
kullanılır.

Dikdörtgenin içini maviye boyamak için

```
g2.setPaint(Color.blue);
Rectangle2D dikdortgen2=new Rectangle2D.Double(50,50,100,40);
g2.fill(dikdortgen2);
```

kullanılabilir. Graphics2D de tek renk yerine değişen bir renk profiliyle boyamak da mümkündür. Örneğin :

```
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 kırmızıdan beyaza dönüßen 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);
    }
}
```

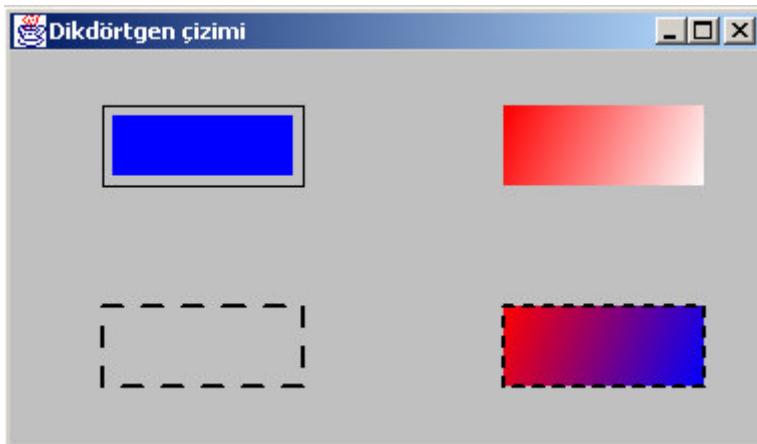
```

=====
public static void main(String[] args)
{
dikdortgenSWF_2000 pencere= new dikdortgenSWF_2000();
pencere.addWindowListener(new BasicWindowMonitor());
pencere.setSize(400,300);
pencere.setVisible(true);
}
}

```

Bu programın çıktısı :

06018.JPG



Şekil 6.18 çeşitli dikdörtgen çizimlerinin **dikdortgenSWF_2000** JFrame çıktısında görünümü

Graphics sınıfında Köşeleri yuvarlatılmış dikdörtgen çizmek içinse **drawRoundRect** metodu kullanılır

```

Public abstract void drawRoundRect(
int x1, // üst tepe noktasının x koordinatı
int y1, // üst tepe noktasının y koordinatı
int en, // dikdörgenin genişliği
int yükseklik, // dikdörtgenin yüksekliği
int koseeni, // yuvarlak kosenin eni
int koseyuksekligi )// yuvarlak kosenin yüksekliği

```

Bu metoda ilave olarak dikdörtgenin içini verilen renkte boyayan

```

Public abstract void fillRoundRect(
int x1, // üst tepe noktasının x koordinatı
int y1, // üst tepe noktasının y koordinatı
int en, // dikdörgenin genişliği
int yükseklik, // dikdörtgenin yüksekliği
int koseeni, // yuvarlak kosenin eni
int koseyuksekligi )// yuvarlak kosenin yüksekliği

```

metotu da paralel olarak tanımlanmıştır. Yine bir örnek problemde bu tanımı kullanacak olursak :

Program 6.16 : yuvarlakKoseliDikdortgenCiz.java programı

```

import java.applet.Applet;
import java.awt.*;
public class yuvarlakKoseliDikdortgenCiz extends Applet
{
public void paint(Graphics g)
{

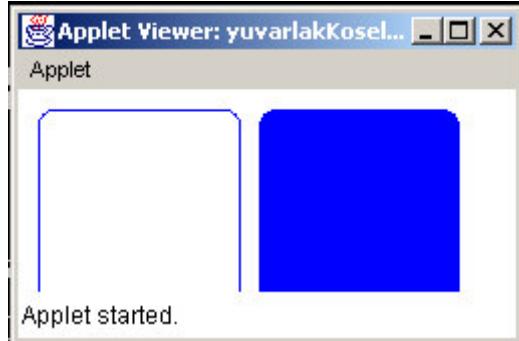
```

```

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



Şekil 6.19 yuvarlakKoseliDikdortgenCiz.html appleti

Graphics2D sınıfında köşeleri yuvarlatılmış dikdörtgen çizmek için :

```

g2.draw(new Rectangle2D.Double(x,
y,dikdortgenEn,dikdortgenYukseklik,xKöşeYarıçapı,yKöşeYarıçapı));

```

metodları kullanılır. Rakamsal örnek verirsek :

```

g2.setPaint(Color.blue);
RoundRectangle2D dikdortgen2=new RoundRectangle2D.Double(50,50,100,40,15,15);
g2.draw(dikdortgen2);

```

kullanılabilir. Dikdörtgenin içini boyamak içinse:

```

g2.setPaint(Color.blue);
RoundRectangle2D dikdortgen2=new RoundRectangle2D.Double(50,50,100,40,15,15);
g2.fill(dikdortgen2);

```

kodu kullanılabilir. Elbette bir önceki örnekte olduğu 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 Köşeli 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);

```

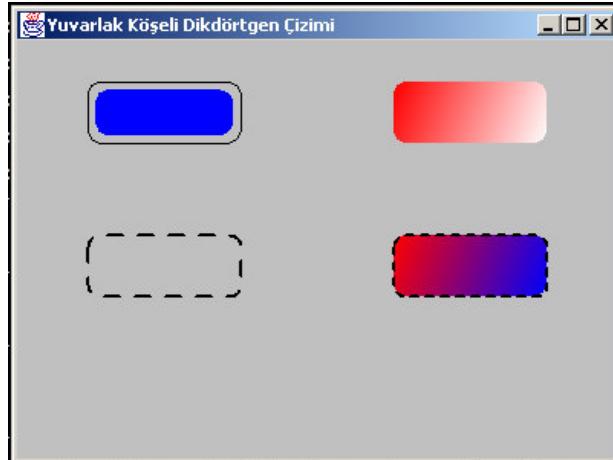
```

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 çeşitli köşesi yuvarlatılmış dikdörtgen çizimlerinin yuvarlakKoselidikdortgenSWF_2000 JFrame çıktısında görünümü

Graphics'de 3 boyutlu dikdortgen çizen **draw3DRect** ve **fill3DRect** metotları mevcuttur. Bu metodların tanımları şu şekilde verilmiştir.

```

Public abstract void draw3DRect(
int x1, // üst tepe noktanın x koordinatı
int y1, // üst tepe noktanın y koordinatı
int en, // dikdörgenin genişliği

```

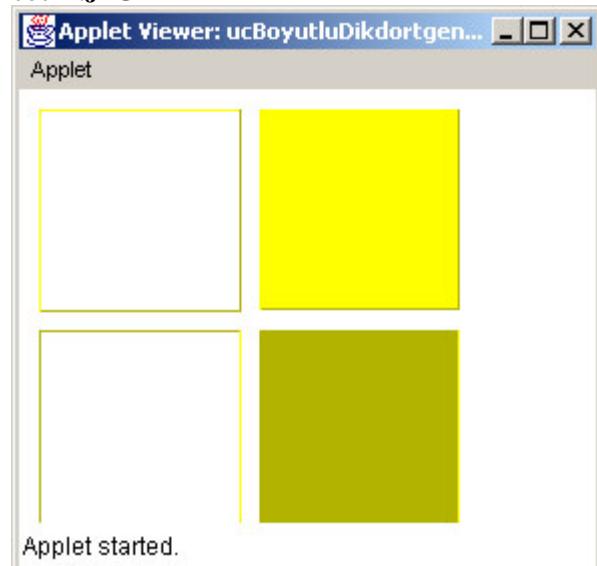
```
int yükseklik, // dikdörtgenin yüksekliği  
boolean yukarıkaldırma) // yukarı kalkık 3üncü boyut
```

```
Public abstract void fill3DRect(  
int x1, // üst tepe noktanın x koordinatı  
int y1, // üst tepe noktanın y koordinatı  
int en, // dikdörgenin genişliği  
int yükseklik, // dikdörtgenin yüksekliği  
boolean yukarıkaldırma) // yukarı kalkık 3üncü boyut
```

Program 61.8 : [ucBoyutluDikdortgenCiz.java](#) programı

```
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



Şekil 6.21 [ucBoyutluDikdortgenCiz.html](#) appleti

6.7 ÇİZİLEN ŞEKİLLERİN DEĞİŞİTİRİLEREK ÇİZİMİ (TRANSFORM)

Graphics2D sınıfında çizilen şekiller dinamik olarak değiştirilebilirler. Bu değişimler döndürme, köşedenekip uzatma gibi değişik işlemler olabilir. Bu değiştirme prosesini önce bir örnekle gösterelim, sonra izah etmeye çalışalım.

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("Dödürümüş - çekilmiş 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);
    }
}

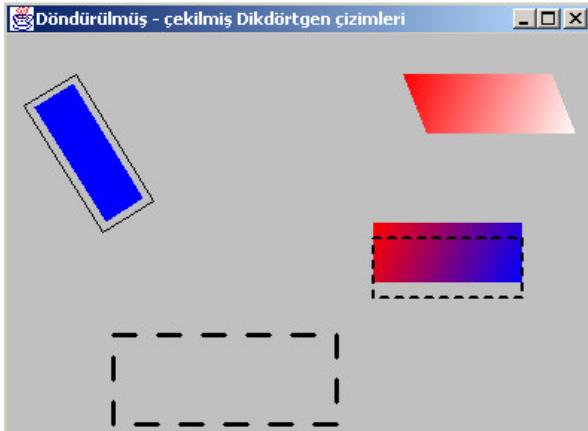
```

```

        pencere.setVisible(true);
    }
}

```

06022.JPG



Şekil 6.22 çeşitli şekli değiştirilmiş dikdörtgen çizimlerinin donmusDikdortgenSWF-2000 JFrame çıktısında görünümü

AffineTransform at= `AffineTransform.getRotateInstance(x,y,açıDerece);`

Şekli x,y noktası etrafında açıDerece kadar saat akranının tersi yönde döndürür.

AffineTransform at2= `AffineTransform.getShearInstance(xÇekişi,yÇekişi);`

Şekli x ekseninde xÇelişi kadar, y ekseninde yÇekişi kadar uzatır (orijinal şeilden elastik ve üst köşesi tutturılmış şekilde)

AffineTransform at1= `AffineTransform.getScaleInstance(xBoyutFaktörü,yBoyutFaktörü);`

Şeklin boyutunu xBoyutFaktörü,yBoyutFaktörü faktörleriyle çarparak büyütür.

AffineTransform at3= `AffineTransform.getTranslateInstance(xTaşımaFaktörü,yTaşımaFaktörü);`

Şekli yerinden hareket ettirir. Orijinal transforma geri dönmek için :

g2.setTransform(new AffineTransform());

metodu çağrırlabilir.

6.8 OVAL VE AÇILI OVAL ÇİZİMİ

Graphics sınıfı Oval çizimi için drawOval ve fillOval metotları kullanılır bu metodların tanımı şöyledir :

```

Public abstract void drawOval(
int x1, // üst tepe noktasının x koordinatı
int y1, // üst tepe noktasının y koordinatı
int en, // ovalin genişliği
int yükseklik) // ovalin yüksekliği

```

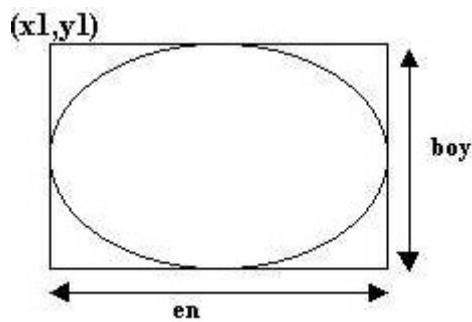
```

Public abstract void fillOval(
int x1, // üst tepe noktasının x koordinatı
int y1, // üst tepe noktasının y koordinatı
int en, // ovalin genişliği
int yükseklik) // ovalin yüksekliği

```

(x1,y1) koordinatının ovalle ilişkisini daha iyi anlayabilmek için aşağıdaki grafiği verebiliriz :

06023.JPG



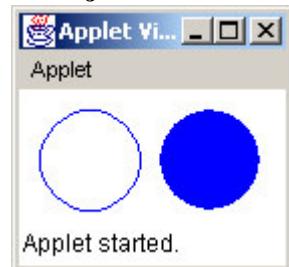
Şekil 6.23 Oval tanımı koordinat sistemi

Oval programını küçük bir örnek programda kullanalım :

Program 6.20 : daireCiz.java programı

```
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



Şekil 6.24 daireCiz.html apleti

Graphics ile bir ovalin tamamını değilde sadece bir bölümünü çizmek dilenilirse, **drawArc** ve **fillArc** metotları kullanılabilir. Bu metodların tanımı :

```
Public abstract void drawArc(
int x1, // üst tepe noktanın x koordinatı
int y1, // üst tepe noktanın y koordinatı
int en, // ovalin genişliği
int yükseklik, // ovalin yüksekliği
int başlangıçacısı, //arkın başlangıç açısı, derece
int çizimaçısı) // arkın çizim açısı, derece
```

```
Public abstract void fillArc(
int x1, // üst tepe noktanın x koordinatı
int y1, // üst tepe noktanın y koordinatı
int en, // ovalin genişliği
int yükseklik, // ovalin yüksekliği
int başlangıçacısı, //arkın başlangıç açısı, derece
int çizimaçısı) // arkın çizim açısı, derece
```

Şimdi de bu metodu kullanan bir bilgisayar programı yazalım :

Program 6.21 : arcCiz.java, arc çizimi programı

```
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 sınıfında 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. İçini boyamak için ise :

```
g2.fill(elips1);
```

metodunu kullanırız.

Yine örneğimizde 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);
    }
}
```

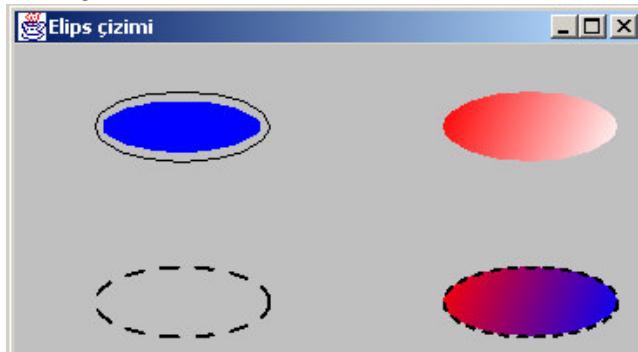
```

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



Şekil 6.26 çeşitli oval (elips) çizimlerinin elipsSWF_2000JFrame çıktısında görünümü

6.9 ÇİZİMİN SEÇİLEN BİR RESİMLE DOLDURULMASI

Graphics2D sınıfında bir grafiği boyarken bir resim dosyasının içinde tanımlanan 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);
    }
}
```

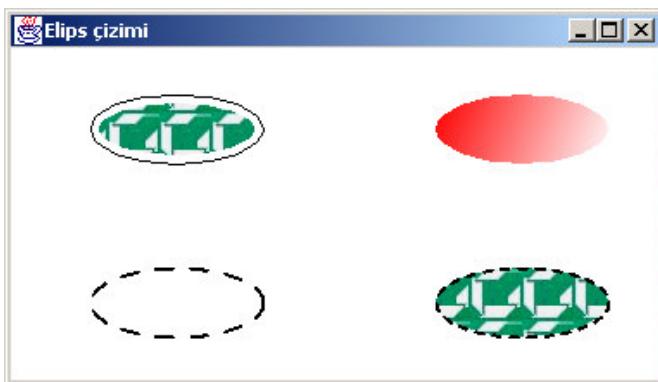
```

    }

//=====
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

```



Şekil 6.27 çeşitli oval (elips) çizimlerinin ve arkaplan resimli boyama işleminin **textureElipsSWF_2000JFrame** çıktısında görünümü

Burada texture sybex.gif dosyasından

public TexturePaint getImageTexture(String imageFile)

metodu kullanılarak aktarılmış ve aynen yeni bir renk gibi çizim ortamında kullanılmıştır.

Şimdi de Graphics2D metoduyla arc çizdirmeye bakalım arc yukarıda bahsettiğimiz gibi tamamlanmamış 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);
    }
}

```

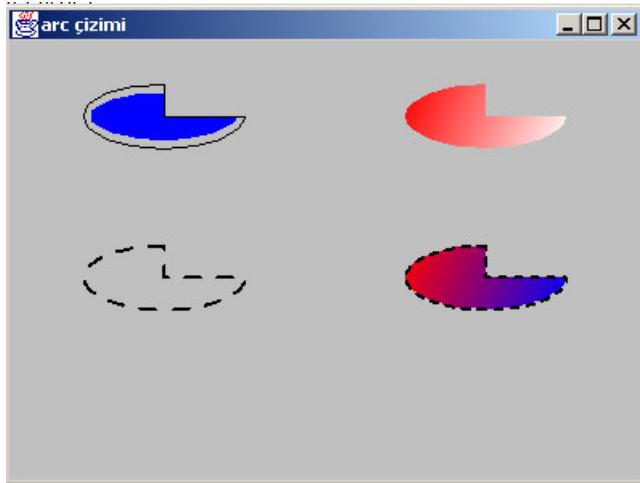
```

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



Şekil 6.28 çeşitli tamamlanmamış oval (ark) çizimlerinin arcSWF_20000 JFrame çıktısında görünümü

6.10 POLYGON ÇİZİMİ

Graphics sınıfında Açık ve kapalı polygon şekilleri çizmek için **drawPolygon** ve **fillPolygon** metotları kullanılır.
Metotların tanımı :

Program 6.24 : drawPolygon.java

```

public abstract void drawPolygon(
int x[], // x koordinat vektörü
int y[], // y koordinatı vektörü

```

```

int nokta sayısı) // nokta sayısı

public abstract void drawPolyline(
int x[], // x koordinat vektörü
int y[], // y koordinati vektörü
int nokta sayısı) // nokta sayısı

public abstract void drawPolygon( Polygon p) //Graphics sınıfı metodu

public Polygon() //Polygon sınıfı

public Polygon(
int x[], // x koordinat vektörü
int y[], // y koordinati vektörü
int nokta sayısı) // nokta sayısı

```

Metot tanımlarından da görüldüğü gibi polygon int türü vektörle çizilebileceği gibi bu vektörü içinde barındıran Polygon sınıfı bir nesne kullanılarak ta çizilebilir.

Örnek olarak üçgen çizen bir program verelim :

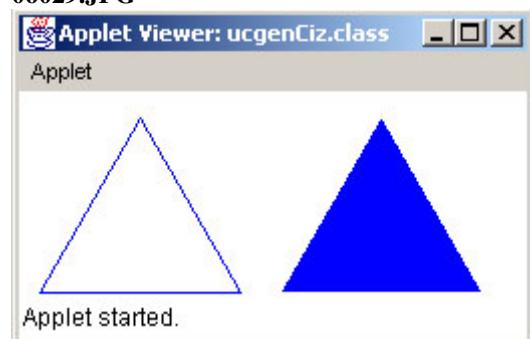
Program 6.25 : ucgenCiz.java, poligon çizim programı

```

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



Şekil 6.29 ucgenCiz.java programı ve apleti

Graphics2D programında grafik çizdirmek için kullanılan metodlar Graphics sınıfına göre oldukça değişiktir. Burada önce örnek programımızı verelim:

Program 6.26 : polygonSWF.java

```
import java.awt.*;
```

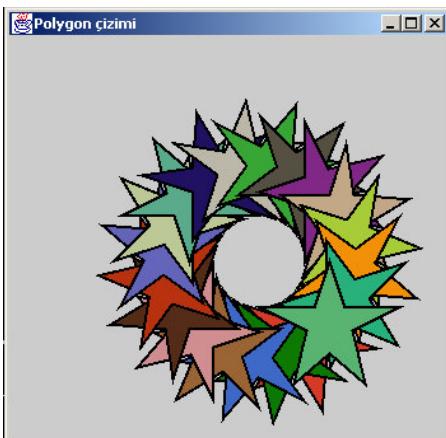
```
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;

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



Şekil 6.30 çeşitli yıldızların (polygonun) çizim ve döndürme işlemi polygonSWF JFrame çıktısında 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 yıldız tanımlanmış, ve

g2.draw(yildiz);

ile çizilmiş, ve

g2.fill(yildiz);

ile içi boyanmıştır. Ayrıca

g2.translate(200,200); ile çizim başlama noktası 200,200 noktasına kaydırılmış ve

g2.rotate(Math.PI/10.0);

komutu ile döndürülmüştür. Programda setPaint kullanarak tesadüfi renkler seçilmiş ve yıldız çizimi doldurulmuştur.

Programda kullanılan

GeneralPath yildiz=new GeneralPath();

Terime özellikle dikkatinizi çekmek isteriz. Bu terim GeneralPath nesnesi yıldızı tanımlar.

6.11 GENELLEŞTİRİLMİŞ EĞRİ ÇİZİMİ

GeneralPath nesnesi kullanarak polinomdan çok daha kompleks çekiller de tanımlayabiliriz.

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()  
    {
```

```

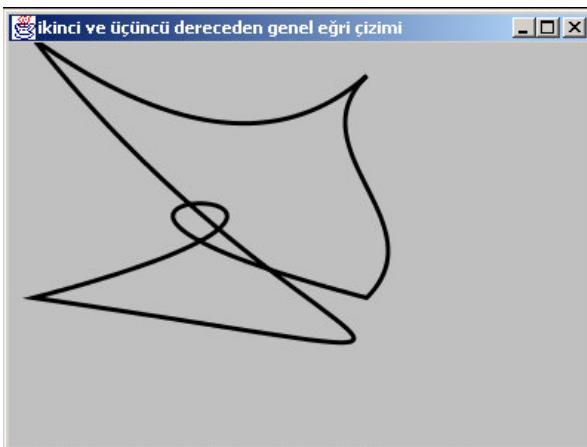
        super("ikinci ve üçüncü dereceden genel eğri çizimi");
    }

    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);
        //quadratic ekleme
        sekil.quadTo(160,120,245,45);
        //cubic 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



Şekil 6.31 ikinci ve üçüncü dereceden genel eğri çizimi, generalPathSWF_2000, JFrame çıktısında görünümü

Bu çizimdeki çizginin oldukça düzgün olduğuna dikkatinizi çekeriz. Bunu sağlayan temel komut :

**g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);**

Komutudur. Bu komut çizgileri daha düzgün hale getirir, fakat belli bir hesap ve hafıza kapasitesi harcar. Programı bu komutu çıkararak da çalıştırınız ve aradaki farkları inceleyiniz.

Şekli oluşturan 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);
komutlarıdır. Daha sonra bu şekiller
g2.draw(sekil);
komutıyla çizdirilmiştir.
g2.fill(sekil);
```

komutu burada kullanılmamakla birlikte, geçerli bir komuttur. Programda deneyiniz. Bu programda çizdirilen ikinci dereceden (quadratik eğriler) quadTo ve curveTo metodları kullanılarak oluşturulmuştur. Şimdi de ikinci derece polynom uydurmasını QuadCurve2D sınıfını kullanarak yapan bir programı 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 eğri 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



Şekil 6.32 ikinci dereceden genel eğri çizimi, curve2SWF_2000, JFrame çıktısında görünümü

Şimdide üçüncü dereceden eğri uyduran bir program koduna göz atalım :

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 eğri 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);
    }
}

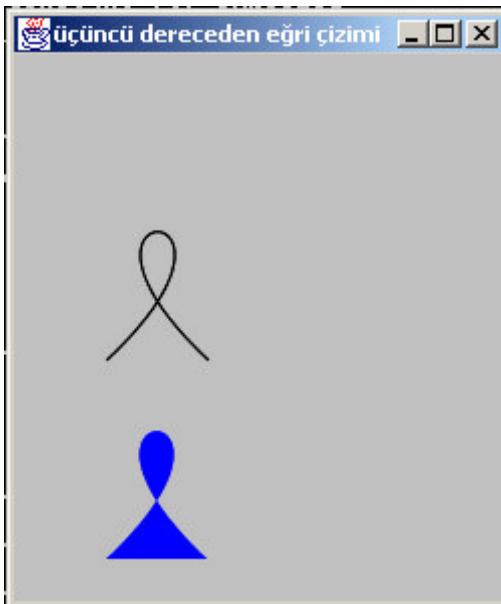
```

```

//=====
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



Şekil 6.33 üçüncü dereceden genel eğri çizimi, curve3SWF_2000, JFrame çıktısında görünümü

6.12 RESİM GÖSTERİMİ

Java dili sadece .gif ve .jpg standartlarındaki resimleri gösterebilir. Resim önce **getImage** metoduyla bir Image sınıfı nesneye yüklenir, sonra **drawImage** metoduyla gösterilir.

Orneğin Image **resim=getImage(getDocumentBase(),"resim.gif");**

Ve resim **g.drawImage(resim,1,1,this);** deyimiyle çizilir. Burada ikinci ve üçüncü noktalar başlangıç x ve y koordinatlarıdır. Son olarak **this** resmin çizileceği appletinde bulunan sınıfın appleti olduğunu belirtmektedir.

Bir örnek probleme resim çizimini biraz daha belirgin hale getitelim :

Program 6.31: resim.java resim aktarım programı

```

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 katı boyunda ciziniz
int en=res.getWidth(this);
int yukseklik=res.getHeight(this);
g.drawImage(res,190,en*2,yukseklik*2,this);
*/
}
}

```

06034.JPG



Şekil 6.34 : Resmin java üzerinden aktarılması, resim.html

Java grafik çizim olasılıkları burada verilenden çok aha geniş boyutludur. Bu bölümde veremediğimiz bir çok detayı javanın parasız olarak çekebileceğiniz döküman kütüphanesinde bulabilirsiniz.

6.13 ALIŞTIRMALAR

- ciz1.java programını 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



Şekil 6.15 ciz1.html appleti

- İki boyutlu plot çizimi amacıyla bir program paketi hazırlanmıştır. Bu paket temel olarak üç java programından oluşmaktadır. Birinci program dosyadan plot bilgisini okuyan Plot.java programıdır. Çizilecek Plotla ilgili en temel bilgi Plot.txt dosyasında bulunmaktadır. Bu dosyadaki ilk satır Pot başlığını, ikinci satır x ekseni başlığını, üçüncü satır y ekseni başlığını, dördüncü satır çizilecek iki boyutlu plot sayısını (okunacak dosya sayısını), beşinci ve sonraki satırlar plot bilgisinin olduğu dosya ismi, çizgi çizim türü, ve integer RGB renk değerlerini vermektedir. Çizgi çizim türlerini bazıları

```

// plottype = 0 sürekli çizgi
// plottype = 10 kullanıcı tarafından tanımlanan karakter
// plottype = 20 dikdörtgen
// plottype = 21 daire
// plottype = 22 üçgen
// plottype = 23 eşkenar dörtgen
// plottype = 24 beşgen
// plottype = 25 altigen

```

eger çizgi türü 10 olarak tanımlanmışsa kullanılacak olan karakterinde verilmesi gereklidir.
a.dat 10 * 0 0 255 gibi. Aşağıda örnek bir Plot.txt dosyası değeri verilmektedir.

```

TÜRKÇE YAZI
X EKSENI
Y EKSENI
2
A.DAT 26 0 0 255
A.DAT 5 0 0 255

```

Plot bilgi dosyasında iki kolon olarak x,y plot bilgisi verilir. Küçük bir örnek dosyası verecek olursak :

```

1 1
2 4
3 9
4 16
5 25
6 36
7 49

```

Burada bilgi dosyadan okunmaktadır. Dosya okuma yazma komutlarını daha detaylı olarak 10. bölümde inceleyeceğiz.

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(xtemp>xmax) xmax=xtemp;
if(ytemp<ymin) ymin=ytemp;
if(ytemp>ymax) 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(xtemp>xmax) xmax=xtemp;
            if(ytemp<ymin) ymin=ytemp;
            if(ytemp>ymax) 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 maxi,
                  double ymini,double maxi)
{
// xabsmin : absolute 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.drawString
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 gridlines
    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)/100.0);
            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*100.0)/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);
}

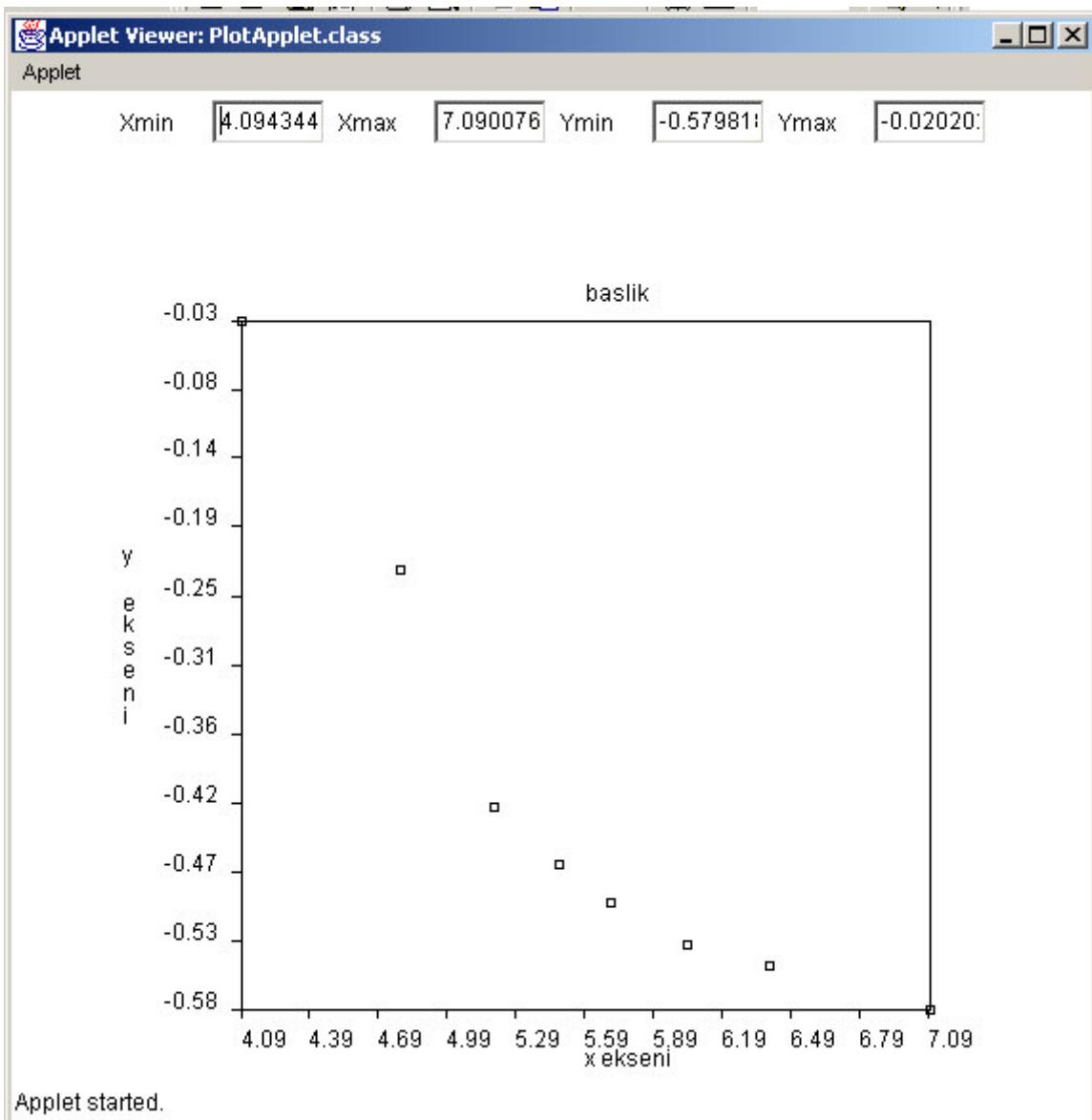
```

```

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,p1ymax);
    g.drawRect(p1.xabsmin,p1.yabsmin,p1.abswidth,p1.absheight);
    ps.drawXTic(p1.xtic,(p1.ymax-p1.ymin)/80.0,0);
    ps.drawY Tic(p1.ytic,(p1.xmax-p1.xmin)/80.0,0);
    ps.drawXNumbers(p1.xtic);
    ps.drawYNumbers(p1.xtic);
    ps.drawLineLabel(p1.label);
    ps.drawXLabel(p1.xlabel);
    ps.drawYLabel(p1.ylabel);
    if(p1.xgridon!=0)
        ps.drawXGrid(p1.xtic);
    if(p1.ygridon!=0)
        ps.drawYGrid(p1.ytic);
    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 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());
    p1ymax=valYmax.doubleValue();
    repaint();
}
}

```

06016.JPG



Şekil 6.16 : PlotApplet.html çıktısı

3. drawRect ve fillRect metodlarını kullanarak biri dolu diğeri sadece bir çizgi olan iki dikdörtgen çiz.
4. Oval ve fillOval metodlarını kullanarak biri dolu diğeri sadece bir çizgi olan iki daire çiz.
5. Paint2D metodlarından yararlanarak biri dolu diğeri sadece bir çizgi olan iki eşkenar dörtgen çizdir.
3. resim.java sınıfında çizilen resmi iki kat büyülüğnde 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);
}
}
```

4. drawPolygon ve fillPolygon metodlarını kullanarak biri dolu diğeri sadece bir çizgi olan iki eşkenar üçgen çiz.

5. Türkçe renkleri tanımlıyan renk sınıfını yaz ([renk.java](#)).

Program 6.18 . renk.java, türkçe renk tanımı sınıfı

```
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 tanımlayabilirsiniz.
    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()
}

```

6. Problem 5.10 da verilen Polar.java, yildiz.java ve yildizCiz.java programlarını 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 yıldız çizime  $\theta = \pi/2 + \text{aci}$  radyandan başlar
    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 yıldız çizime 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 yıldızın içini 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 yıldız çizime teta=pi/2+aci radyandan baslar
    // yıldızın i\d{f}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 : YıldızCiz.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 programını 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 sınıfını test eden renkTest.java programını inceleyiniz. Ve applet çıktısını çalıştırınız.

Program 6.21 renkTest.java

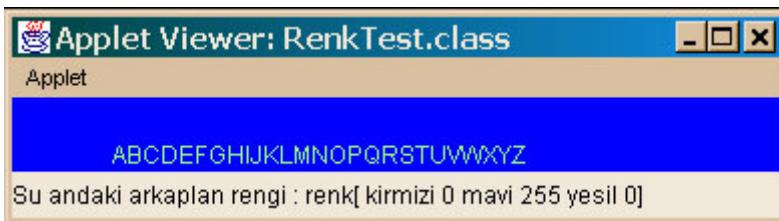
```
import java.awt.*;
import java.applet.Applet;
import renk;

public class RenkTest extends Applet
{
    private int kirmizi,yesil,mavi;

    public void init()
    {
        kirmizi=150;
        yesil=255;
        mavi=125;
        // arka planın rengi mavi olarak veriliyor
        setBackground(renk.mavi);
    }

    public void paint(Graphics g)
    {
        // yazının rengi 100,255,125 olarak alındı

        g.setColor(new renk(kirmizi,yesil,mavi));
        g.drawString("ABCDEFGHIJKLMNPQRSTUVWXYZ",50,33);
        showStatus("Su andaki arkaplan rengi : "+getBackground());
    }
}
```



Şekil 6.17 RenkTest.html appletinin görünümü

11. H6O2

polar.java, yildiz.java, yildizCiz.java programlarının inceleyiniz. aynı programlama prensiplerini kullanarak n eşkenar kenarlı çokken çizecek [eskenarcokgen.java](#) sınıfını yaratınız. Bu sınıf

public static void drawEskenarCokgen(Graphics g,int xi,int yi, int n,int eskenarcokkencapi,double aci)
public static void fillEskenarCokgen(Graphics g,int xi,int yi, int n,int yildizboyu,double aci)

metotlarını kullanarak eşkenar çokgen çizsin ve içini boyasin.

not xi,yi : çokken merkezinin koordinatları

n : çokken köşe sayısı

eskenarcokkencapi: çokken merkezinden çokken köşesine olan mesafe

aci : 90 derece açı çokkenin başlama açısı olarak tanımlanmıştır. bu açıdan sapış 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 çokgen merkez koordinatları
        // n : eskenar dörtgen köse sayısı
        // eskenarcokkenboyu : eskenar çokgen merkezinden
        // kenarına 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 metodu 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 metodu sonu
}

```

8. eskenarcokgen sınıfını test etmek amacıyla bir applet programı yazınız ve ekrana beşken çizdiriniz.
9. eskenarcokgen sınıfını test etmek amacıyla bir JFrame programı yazınız ve ekrana üçken çizdiriniz.

13.

H6O3

grafik sınıfındaki:

public abstract void drawRect(int x1,int y1, int en,int yükseklik)
metotundan yararlanarak H6O3 sınıfında
public static void drawAciliDortgen(Graphics g, int x1,int y1, int en,int yükseklik,double aci)
metotunu yazınız. Bu metot dikdortgeni verilen açı kadar bir açıyla çizdirsin.
x1,y1 : dörtgenin sağ üst köşe koordinatları
en,yükseklik : dörtgenin eni ve yüksekliği
aci: dörgenin radyan cinsinden x1,y1 noktası etrafında dönme miktarı.
not :polar sınıfı ve drawPolygon metodlarından yararlanabilirsiniz.

Program 6.23 açılı dikdortgen çizdireن dikdortgen sınıfı

```

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 sınıfını kullanan H6O2.java programı

```

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 metodunda tı̄m deşı̄kenlerin ilk deşerleri verilmelidir.
        Panel p=new Panel();
        p.setLayout(new GridLayout(5,4));
        //5*4 tablo p paneli hazırlandı̄
    }
}

```

```
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: GRAFİK APPLET VE ÇERÇEVE OLUŞTURMA OLUŞTURMA METOTLARI, GRAFİK KULLANICI ARABİRİM PROGRAMLARI (GUI) ,

7.1 GRAPHIC KULLANICI ARABİRİM PROGRAMLARI, GUI, (GRAPHICS USER INTERFACE)

Grafik kullanıcı arabirim programları, GUI, programınıza kullanıcının daha iyi anlayabileceği bir görünüş verme ve aynı zamanda grafik ekrandan girdi cve çıktı yapabilme amacıyla oluşturulur. Kullanıcıya sunulan programın onun rahatlıkla anlayabileceği bir formata sahip olması günümüz programlamacılığında oldukça önemlidir. Reklamlarla büyüyen bir kuşak için paket bazen paketin içindekinden bile önemli hale gelebilmiştir. Grafik Kullanıcı Arabirim programları kullanıcıya sunduğumuz paketi oluşturmaktadır.

Java'da temel grafik arabirim kütüphaneleri **java.awt** sınıfında yer almaktadır. **Java 1.2** den itibare awt kütüphanesine ek olarak java swing grafik arabirim kütüphanesi **javax.swing** tanımlanmıştır. Bu bölümde awt ve swing yapılarını paralel olarak vermeye çalışacağiz.. Yalnız swing kütüphanesi kapsam olarak çok daha geniş olduğundan swing ile yapılan tüm işlemlerin ve grafik arabirim alt programlarını awt'de karşılıkları mevcut değildir. Bir önceki grafik çiziminde olduğu gibi grafik arabirim elemanlarının tanıtımı da oldukça geniş bir yelpaze oluşturduğundan mümkün olduğunda detay verilmeye çalışılsa bile tüm kapsamı bu kitapta işlemek mümkün değildir. Örneğin java swing konusundaki kapsamlı bir kitabı olan "Java Swing , O'Reily basım evi" kitabı yaklaşık 1200 sayfalık bir kitabı sadece bu konuya ayırmıştır.

GUI metodları **java.awt** (Abstract windowing toolkit) paketinde yer almaktadır. Bu pakette **Container** ve **Component** isimli iki alt pakette yer almaktadır. Her zaman kullandığımız **Applet** sınıfı Component sınıfından türiyen **Panel** sınıfının alt sınıfıdır. Yani Component Appletlerin süper sınıfıdır.

Burada Component sınıfının

- TextComponent
- TextField
- TextArea
- Button
- Label
- Checkbox
- List
- Choice
- Canvas
- Scrollbar
- Container
- Panel

Java.applet.Applet sınıfının

- ScrollPane
- Window
- Frame
- Dialog

Sınıflarını incelemeye çalışacağız.

Paralel olarak **javax.swing** paketinde yer alan grafik arayüz programlarında yer alan üst seviye paketler **Japplet**, **JDialog**, **JFrame**, **JWindow** ve **JComponent**'dir

JComponent sınıfının altında yer alan sınıfların bazıları :

- JComboBox
- JLabel
- JList
- JMenuBar
- JPanel
- JPopupMenu
- JScrollPane

- JScrollPane
- JTable
- JTree
- JInternalFrame
- JOptionPane
- JProgressBar
- JRootPane
- JSeparator
- JSlider
- JSplitPane
- JTabbedPane
- JToolBar
- JToolTip
- JVewPort
- JColorChooser
- JTextComponent
- JTextArea
- JTextField
- JPasswordField
- JEditorPAn
- JTextPane
- JFileChooser
- JLayeredPane
- AbstractButton
- JToggleButton
- JCheckBox
- JRadioButton
- JButton
- JMenuItem
- JMenu
- JRadioButtonMenuItem
- JCheckButtonMenuItem

Yine yineleyelim, **swing** sınıfındaki tüm alt metodları kapsamak bu kitabın kapsamını oldukça büyütücektir, bu yüzden şimdilik buna imkan göremiyoruz, fakat pratikte ne oldukları hakkında iyi bir fikir verebilecek genişlikte bir spektrum vermeye çalışacağız. Burada özellikle awt ve swing sınıfı grafik işlemcilerini bir arada vermeye çalıştık. Kullanıcı her iki tür grafik ortamıyla da karşılaşacağı için (Kendi yazmama bile hazır programlar karşısına çıkacaktır), her ikisini de bilmesi gereklidir. Zaten bu kitabın tüm konularında da hem awt hem de swing ile yazılmış program örnekleri vermeye çalıştık. Bu konudan sonra geriye dönüp tüm program örneklerini grafik ortamı gözüyle inceliyebilirsiniz.

7.2 AWT LABEL (ETİKET) SINIFI , JAVAX JLABEL VE ICON SINIFLARI

Label sınıfı Appletin içine sadece okunabilen fakat değiştirilemeyen yazılar yazma amacıyla kullanılır. **Label** sınıfında tanımlanan metodlar şunlardır:

- Kurucu Metotlar :
- ```
public Label() // Label sınıfı nesneyi oluşturur herhangi bir yazı göstermez
public Label(String s) // Label sınıfı nesneyi oluşturur ve s stringini gösterir
public void Label(String s,int pozisyon) // Label sınıfı nesneyi oluşturur ve s stringi gösterir aynı zamanda
Labeli pozisyonda verilen değere göre istenilen yere yerleştirir.
```

Yerleştirme konumları :**Label.LEFT,Label.CENTER,Label.RIGHT** (sol,orta,sağ) konumlarıdır.

- Giriş çıkış metodları
- ```
public String getText() // Label sınıfının String değişkenini okur.
public void setText(String s) // Label sınıfının String değişkenine yeni değer girer.
public void setAlignment(int pozisyon)
```

Label pozisyon değişkeninde verilen değere göre istenilen yere yerleştirir.

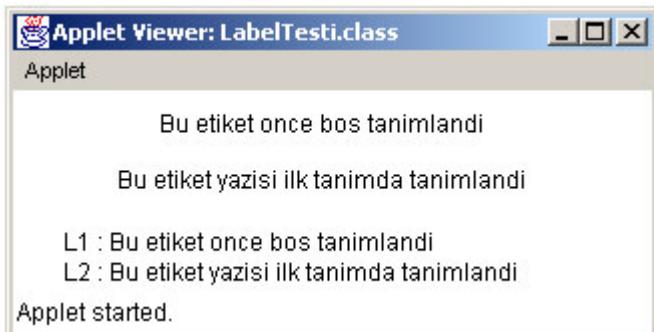
Yerleştirme konumları :**Label.LEFT,Label.CENTER,Label.RIGHT** (sol,orta,sağ) konumlarıdır.

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 programında kullanılan add metodu Label'i Container sınıfına(Appletin alt sınıfı olduğu) bağlar, diğer deyimle applete veya verilen üst komponente ilave eder.

Swing sınıfı JLabel sınıfından farklı olarak bize sadece yazı değil aynı zamanda görüntü eklemeye olasılıkları da sunar. Temel sınıf olarak kurucu metodları :

```
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 yazı pozisyonu

JLabel.LEFT
JLabel.CENTER
JLabel.RIGHT

Veya

SwingConstants.LEFT
SwingConstants.CENTER
SwingConstants.RIGHT

Değerlerinden birini alabilir.

Jlabel sınıfının çok kullanacağımız bazı metodlarını sayacak olursak :

Jcomponent sınıfından gelen metodlar :

void setBackground(Color c) : arkaplan rengini değiştirir

Color getBackground() : arkaplan rengini okur

void setForeground(Color c) : önpalan rengini değiştirir

Color getForeground() : ön plan rengini okur.

void setFont(Font f) : yazı fontunu seçer

Font getFont() : yazı fontunu okur

Boolean isVisible

: görünür olup olmadığını kontrol eder

Void setVisible(Boolean b) : görünür veya görünmez yapar.

Jlabel sınıfında tanımlanmış bazı alt sınıflar :

void setText(String s) : JLabel'ın text değişkeninin değerine yeni verilen değeri atar

String getText() : JLabel'ın String text değişkenini okur.

void setVerticalAlignment(int düşey_pozisyon) : Label'in Pencere içindeki düşey pozisyonunu ayarlar

Buradaki düşey pozisyon

JLabel.TOP

JLabel.CENTER

JLabel.BOTTOM

Veya

SwingConstants.TOP

SwingConstants.CENTER

SwingConstants.BOTTOM

Değerlerinden birini alabilir.

int getVerticalAlignment()

void setVerticalTextPosition(int düşey_pozisyon) : Label'in içindeki yazının Label içindeki düşey pozisyonunu ayarlar. Sabitler üsttekilerin aynıdır.

int getVerticalTextPosition() : Label'in içindeki yazının Label içindeki düşey pozisyonunu okur.

void setHorizontalTextPosition(int düşey_pozisyon) : Label'in içindeki yazının Label içindeki yatay pozisyonunu ayarlar. Sabitler üsttekilerin aynıdır(LEFT,CENTER,RIGHT).

int getHorizontalTextPosition() : Label'in içindeki yazının Label içindeki yatay pozisyonunu okur.

void setIcon(Icon resim) : Label içindeki resmi değiştirir veya eğer yoksa ilk defa atar

Icon getIcon() : Label içindeki resmi okur.

void setDisplayedMnemonic(int c) : Mnemonic alt harf kontrolunu kullanarak belli bir girişe ulaşmayı sağlar. (örneğin 'm' harfine set edilmişse alt-m bu komuta gider. Örnek problemlerle bu kavram açılacaktır.

int getDisplayedMnemonic(int c) : Mnemonic değerini okur.

Şimdi örnek problemlerle swing JLabel kullanımına daha detaylı olarak bakalım.

Birinci örneğimizde basit bir şekilde String label yazdıracağız. Çerçeve olarak JFrame kullandık.

Program 7.2 : Label1SW.java swing java Jlabel örneği

```
import javax.swing.*;
public class label1SW
{
    public static void main(String[] args)
    {
        JLabel etiket=new JLabel("JLabel sınıfına hoş geldiniz");
        JFrame cerceve=new JFrame();
        cerceve.addWindowListener(new BasicWindowMonitor());
        cerceve.getContentPane().add(etiket);
        cerceve.pack();
        cerceve.setVisible(true);
    }
}
```

07002.JPG



Şekil 7.2 Label1SW.java JFrame penceresi

İkinci örneğimizde resimli, hem resim hem yazılı ve yalnız yazılı bir JApplet programı oluşturduk. Bu programın aynı zamanda main metodu da olduğundan hem applet hem de frame olarak çağrılması mümkündür. Çerçeve olarak JFrame kullandık.

Program 7.3 LabelTestiSW Japplet türü JLabel örnek programı

```
Import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;

public class LabelTestiSW extends JApplet {

    protected String iconname1="images/kuş.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) yazısı",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



Şekil 7.3 LabelTestiSW.java JFrame penceresi

Program 7.4 LabelDemo JPanel türü JLabel örnek programı

```

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 satır, 1 sütun
        label1 = new JLabel("Resim ve Yazı",
            icon,
            JLabel.CENTER);
        //resme göreceli olarak yazı pozisyonunu ayarla:
        label1.setVerticalTextPosition(JLabel.BOTTOM);
        label1.setHorizontalTextPosition(JLabel.CENTER);
        label2 = new JLabel("Sadece yazılı 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



Şekil 7.4 LabelDemo.java JPanel penceresi

Bu programda resim

ImageIcon icon = new ImageIcon("images/middle.gif","bir damla resmi");
Tanımiyla ImageIcon sınıfı aracılığıyla resmi JLabel'a aktardı. **images/middle.gif** middle.gif dosyasının images alt direktorysinde yer aldığı belirtir.

Burada Icon interphase'ine biraz daha yakından göz atalım. Icon interphase sınıfı olduğundan tanımlanan sınıfı içerisinde

```

public int getIconWidth()
public int getIconHeight()
public void paintIcon(Component c, Graphics g, int x,int y)

```

metodları tanımlanmak zorundadır. Şimdi Icon sınıfının alt sınıfı olan OvalIcon sınıfını tanıyalayalım

Program 7.5 JLabel'da kullanılacak Icon oluşturan OvalIcon sınıfı

```

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 yükseklik;}
}

```

Şimdi de OvalIcon sınıfını kullanan JLabel sınıflarını oluşturup gösterelim :

Program 7.6 OvalIcon sınıfını kullanarak JLabel'da daire Icon'u (resmi) gösteren label2SW programı

```

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 sınıfın çıktısı :

07005.JPG



Şekil 7.5 label2SW.java JFrame penceresi

Buradan da görüldüğü gibi JLabel sınıfını çok daha geniş bir platformda, oluşturduğumuz bir grafiğin veya çizimin sonuçlarını grafik ortama aktarmak amacıyla da kullanabiliriz.

7.3 BUTTON VE JButton (DÜĞME) SINIFLARI

Button (düğme) basılınca belirli işlemlerin yapılmasını başlatan bir sınıfır. Graphic arabirim kullanım (GUI) programlarını veya applet programlarını konsol programlarından ayıran en önemli özellik, GUI lerin işlem kökenli olmasıdır. Diğer bir deyimle kullanıcının başlatacağı işlemleri yapar ve kullanıcının yeni bir işlem tanımlamasını beklerler. Bu işlemleri yapan sınıflar **java.awt.event** kütüphanesinde tanımlanmıştır. Bu kütüphanede en çok kullanacağımız sınıflar hiyerarşî dizendinde şöyle verilebilir :

Java.util.EventObject

- Java.awt.AWTEvent
- ActionEvent // Bir düğme(Button) basıldığında, Listeden (List) iki klike bir seçim yapıldığında veya menuden seçim yapıldığında kullanılır.

- AdjustmentEvent // aşağı yukarı kaydırma cubugunda (scroll bar) bir işlem yapıldığında kullanılır.
- ItemEvent // List veya CheckBox kliklendiginde (bir kere) kullanılır
- ComponentEvent //appletteki elemanların gizlenmesi, boyalarının değiştirilmesi, veya değişik yerlere alınması amacıyla kullanılır.
- ContainerEvent //Containera(applete) yeni bir eleman eklenir veya çıkarılırken kullanılır.
- FocusEvent//herhangi bir applet elemamı kullanılmaya baslandığında on plana çıkarılmakk, ve kullanılmadığında arka plana atılmak istendiğinde kullanılır.
- PaintEvent
- WindowEvent //pencere açılıp kapandığında,küçültülüp büyütüldüğünde vb kullanılır.
- InputEvent
- KeyEvent //keyboarddan bir girdi alındığında kullanılır
- MouseEvent //mouse dan bir girdi alındığında kullanılır

Kullanıcı bir applet veya pencere tipi başka Container programı yazarken dinleme ile ilgili iki işlem yapar ilk işlem herhangi bir GUI dan bir girdi olup olmadığı dinlemek oldusya kaydetmek ikincisi ise bu işlemin sebep olacağı işlemleri oluşturmaktır. Dinleme işlemini yapan metodlar şunlardır :

Java.util.EventListener

- ActionListener
- AdjustmentListener
- ComponentListener
- ContainerListener
- FocusListener
- ItemListener
- KeyListener
- MouseListener
- MouseMotionListener
- TextListener
- WindowListener

Bu metodların bir çوغunu ilgili örneklerde yeri geldiğinde kullanacağız. Şimdi aynı sonucu veren iki button ornegi ile bu kavramı biraz daha açmaya çalışalım :

Program 7.7 : ButtonTesti.java, düğme kullanım testi

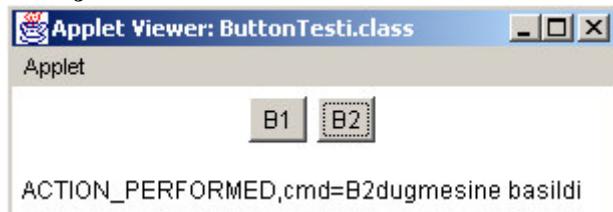
```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class ButtonTesti extends Applet
{
private Button B1,B2;
public void init()
{
B1=new Button("B1");
B1.addActionListener(new B1Basilinca(this));
add(B1);
B2=new Button("B2");
B2.addActionListener(new B2Basilinca(this));
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



Şekil 7.6 ButtonTesti.java sınıfı sonuçlarının appletde görünmesi

Problem 7.8 ButtonTesti1.java sınıfı (sonuçlar ButtonTesti.java ile aynıdır)

```

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 düğmelere basıldığında oluşan eylemler (showStatus köşesine basılan düğmeyi yazdırma) ayrı sınıflarda yaptırılmış ve Applet sınıfı üzerinden bizim ButtonTesti sınıfına aktarılmıştır. İkinci programda ise aynı işlemler tek bir sınıfın içerisinde yer almıştır.

Programdan da görüleceği gibi düğme basma eylemi ActionEvent sınıfı üzerinden aktarılmıştır.

JButton sınıfı ile Button sınıfı arasında kullanım açısından Label ve Jlabel sınıfının arasındaki benzer farklar vardır.

JButton sınıfının sınıf tanımı :

```

public class JButton extends AbstractButton implements Accessible
{

```

```

public JButton()
public JButton(Icon resim)
public JButton(String yazı);
public JButton(String yazı, Icon resim)
public AccessibleContext getAccessibleContext();
public String getUIClassID();
public boolean isDefaultButton();
public boolean isDefaultCapable();
public void setDefaultCapable(boolean b);
public void updateUI();
}

```

şeklindedir. Görüldüğü gibi JButton fazla metod barındırmaz bir çok önemli metodu Abstract button ve onun tepe sınıfı JComponent sınıflarından alır. JButton sınıfında JLabel sınıfında da kullandığımız JComponent sınıfından gelen metodlar :

public void setBackground(Color c)	: arkaplan rengini değiştirir
public Color getBackground()	: arkaplan rengini okur
public void setForeground(Color c)	: önpalan rengini değiştirir
public Color getForeground()	: ön plan rengini okur.
public void setFont(Font f)	: yazı fontunu seçer
public Font getFont()	: yazı fontunu okur
public Boolean isVisible()	: görünür olup olmadığını kontrol eder
public void setVisible(Boolean b)	: görünür veya görünmez yapar.

AbstractButton sınıfında tanımlanmış bazı alt sınıflar :

public void setText(String s)	: JLabel'ın text değişkeninin değerine yeni verilen değeri atar
public String getText()	: AbstractButton'ın String text değişkenini okur.

public void setVerticalAlignment(int düşey_pozisyon)	: AbstractButton'in Pencere içindeki düşey pozisyonunu ayarlar
---	--

Buradaki düşey pozisyon

SwingConstants.TOP

SwingConstants.CENTER

SwingConstants.BOTTOM

Değerlerinden birini alabilir.

public int getVerticalAlignment()

public void setVerticalTextPosition(int düşey_pozisyon) : AbstractButton'in içindeki yazının Label içindeki düşey pozisyonunu ayarlar. Sabitler üsttekilerin aynıdır.

public int getVerticalTextPosition() : AbstractButton'in içindeki yazının AbstractButton içindeki düşey pozisyonunu okur.

public void setHorizontalTextPosition(int düşey_pozisyon) : AbstractButton'in içindeki yazının AbstractButton içindeki yatay pozisyonunu ayarlar. Sabitler

SwingConstants.LEFT

SwingConstants.CENTER

SwingConstants.RIGHT

public int getHorizontalTextPosition() : AbstractButton'in içindeki yazının AbstractButton' içindeki yatay pozisyonunu okur.

Public void setIcon(Icon resim) : AbstractButton'in içindeki resmi değiştirir veya eğer yoksa ilk defa atar

public Icon getIcon() : AbstractButton içindeki resmi okur.

public void setDisplayedMnemonic(int c) : Mnemonic alt harf kontrolunu kullanarak belli bir girişe ulaşmayı sağlar. (örneğin 'm' harfine set edilmişse alt-m bu komuta gider. Örnek problemlerle bu kavram açılacaktır.

Public int getDisplayedMnemonic(int c) : Mnemonic değerini okur.

Listener (pencere dinleme) metod ve neslerinden ise :

```

protected ActionListener actionPerformed;
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. Şimdi örnekler üzerinden JButton sınıfını kullanmayı deneyelim :

Program 7.9 JButton sınıfının gösterilmesini örnekleyen ButtonTestiSWF programı

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

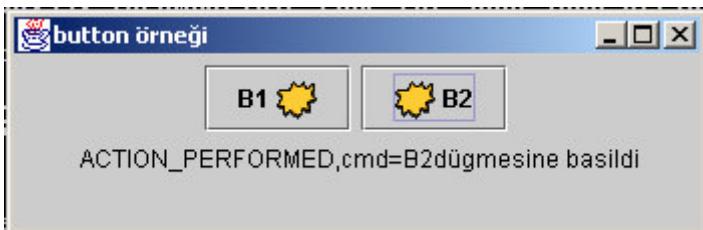
public class ButtonTestiSWF extends JFrame implements ActionListener
{
    private JButton B1,B2;
    private JTextArea cikti;

    public ButtonTestiSWF()
    {
        super("button örneği");
        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()+" düğmesine basıldı\n";
        else if(e.getSource()==B2)
            gs = " "+e paramString() +" düğmesine basıldı";
        cikti.setText(gs);
    }
}
```

07007.JPG



Şekil 7. ButtonTestiSWF.java sonuçlarının JFrame penceresinde görüntüsü

Button ve JButton arasındaki farkları görmek amacıyla birbirinin aynı olan ButtonDemoApplet.java awt applet programını, ButtonDemoApplet.java swing programıyla karşılaştırıralım. Not :isimler aynı olduğundan bu iki programı aynı dosyalarda saklıyamayız.

Program 7.10 Button sınıfının gösterilmesini örnekleyen ButtonDemoApplet programı

```
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");

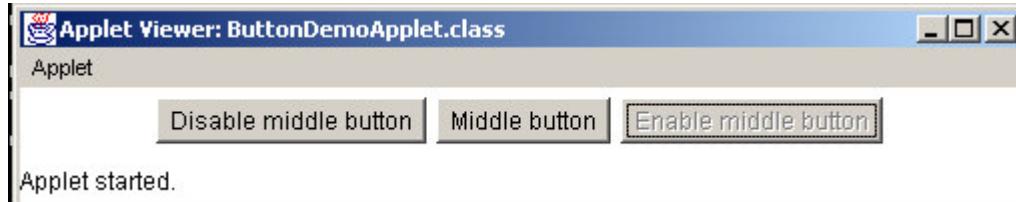
        b3 = new Button("Enable middle button");
        b3.setEnabled(false);
        b3.setActionCommand(ENABLE);

        //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



Şekil 8. ButtonDemoApplet.java sonuçlarının AppletViewer browserında görüntüsü

Program 7.11 Button sınıfının gösterilmesini örnekleyen ButtonDemoApplet programı (bu program 7.10 ile aynı işlevi yapmaktadır)

```
/*
 * 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);
        }
    }

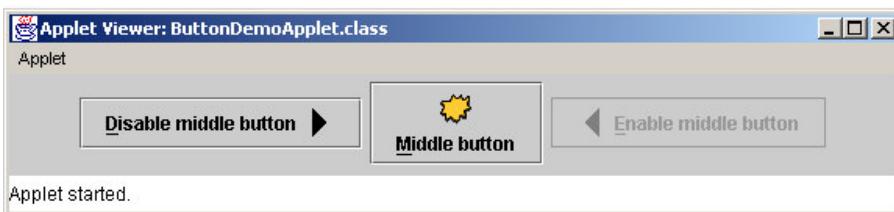
    protected URL getURL(String filename) {
        URL codeBase = getCodeBase();
        URL url = null;

        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



Şekil 9. ButtonDemoApplet.java sonuçlarının swing JApplet olarak Appletviewer browserında görüntüsü

Programdan da görüldüğü gibi bu program bir önce verilmiş olan programın swing frame benzeridir. İki program da Applet olarak verilmiştir. Programlarda setEnabled(Boolean b) metodu kullanılarak düğmenin çalışması etkisiz hale getirilmiştir. Ayrıca swing versiyonunda resim dosyasını kullanırken URL sınıfı üzerinden okuduk. URL network ortamında bilgi aktarmaya yarayan bir sınıfıtır. Resim dosyası başka bir bilgisayarda olabilir.

7.4 TEXTFIELD (YAZIM ALANI) SINIFI

TextField bir satırlık yazım alanlarıdır. TextField'in metotları aşağıda verilmiştir :

- **Kurucu Metotlar :**

```

public TextField()
public TextField(int satırsayısı) // satır sayısı uzunluğunda bir yazı kutusu açar.
public TextField(String s) // s uzunluğunda bir yazı kutusu açır ve s yi içine yazar
public TextField(String s,int satırsayısı ) // satır sayısı uzunluğunda bir yazı kutusu açır ve s yi içine yazar

```

- **Diger Metotlar:**

```

Public void setEchoChar(char c) // kutudaki gerçek yazıyı gizleyerek onun yerine c karakter
değişkenini yazar.
Public void setEditable(boolean b) //kutunun içine yazılan yazının değiştirilebilmesini sağlar veya
engeller
b=true yazı yazılabilir b=false yazı yazılamaz.

```

Bu metodların kullanımını yine bir program örneğiyle izleyelim :

Program 7.12 TextFieldTesti.java programı

```

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 yazı yazınız");
        T2.addActionListener(this);
        add(T2);
        T2=new TextField("gizli yazı");
        T2.setEchoChar('*');
        T2.addActionListener(this);
        add(T2);
        T3=new TextField("gizli yazı");
        T3.setEditable(false);
        add(T3);
    }
    public void actionPerformed(ActionEvent e)
    {
        showStatus("Yazı : "+e.getActionCommand());
    }
}

```

07010.JPG



Şekil 7.10 [TextFieldTesti.java](#) programı sonuçlarının appletde görünümü

Bu programda her yazı alanına (TextField) addActionListener metodu eklenmiş ve bir yazı girildiğinde actionPerformed metotundaki showStatus metodu yoluyla en son girilen yazı showStatus penceresine aktarılmıştır.

TextField sınıfının java swing eşdeğeri JTextField sınıfıdır. JTextField sınıfı kurucu metodları :

JTextField()
JTextField(String)
JTextField(String, int)
JTextField(int)
JTextField(Document, String, int)

Şeklindedir.

TextFielde Paralel olarak Özel bir TextField türü olan PasswordField de swingde tanımlanmıştır. PassWord fieldin kurucu metod tanımları :

JPasswordField()
JPasswordField(String)
JPasswordField(String, int)
JPasswordField(int)
JPasswordField(Document, String, int)

Şeklinde tanımlanmıştır.

Bu iki sınıfın Text değerini girmak veya okumak için

void setText(String)

String getText()

metodları tanımlanmıştır.

void setEditable(boolean)

boolean isEditable()

metodları Yazı alanındaki yazıyı kullanıcının değiştirdip değiştiremeyeceğini saptar veya bu müsadeyi değiştirir.

void setColumns(int);

int getColumns()

int getColumnWidth()

metodları yazı alanının boyutunu değiştirmek veya saptamak amacıyla kullanılabilir.

void setHorizontalAlignment(int);

int getHorizontalAlignment()

metodları yazı alanındaki yazının ne şekilde yerleştirildiğini saptar veya istege göre yerleştirir.

JTextField.LEFT, JTextField.CENTER, veya JTextField.RIGHT

void setEchoChar(char)

char getEchoChar()

JPasswordField sınıfında tanımlanmış yazılan yazının yerine geçecek karakteri tanımlayan veya soran

metodlardır.

void addActionListener(ActionListener)

metodları ise TextFieldin dinlenmesini ve giriş tuşu basıldığında yapılacak eylemlerin verildiği ActionListener implement'inden gelen

public void actionPerformed(ActionEvent e)

metodunu çalıştırılmaya yarar.

void removeActionListener(ActionListener)

ise ActionListener dinleme metodunu iptal eder.

JTextField ve JPasswordField metodlarının kullanılmasını bir örnek probleme inceleyelim. Bu bir önceki problemin hemen aynı işlemleri yapan JFrame ortamında yazılmış bir örnek problemdir.

Program 7.13 TextFieldTestiSWF.java programı

```
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("JTextField 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 yazı yazınız");
T2.addActionListener(this);
c.add(T2);
T3=new JTextField("değiştirilemez yazı");
T3.setEditable(false);
T3.addActionListener(this);
c.add(T3);
T4=new JPasswordField("gizli yazı");
c.add(T4);
}
}

public void actionPerformed(ActionEvent e)
{
String s="Kaynak = ";
if(e.getSource()==T1)
    s+="T1 Yazı : "+e.getActionCommand();
else if(e.getSource()==T2)
    s+="T2 Yazı : "+e.getActionCommand();
else if(e.getSource()==T3)
    s+="T3 Yazı : "+e.getActionCommand();
else if(e.getSource()==T4)
    s+="T4 Yazı : "+e.getActionCommand();
JOptionPane.showMessageDialog(this,s,"JTextField 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



Şekil 7.11 TextFieldTestiSWF.java programı sonuçlarının JFrame ve JOptionPane'de görünümü

7.5 YAZIM ALANI AWT TEXTAREA VE SWING JTEXTAREA SINIFLARI

awt kütüphanesinde tanımlanan Text Area sınıfı yapı olarak TextField sınıfına benzer. En önemli farkı bir satır yerine birden fazla satır girdi alabilme olasılığıdır. TextField ve TextArea sınıfları TextComponent sınıfının alt sınıflarıdır. Bu yüzden üstte belirttiğimiz

Public void setEchoChar(char c) // kutudaki gerçek yazıyı gizleyerek onun yerine c karakter değişkenini yazar.

Public void setEditable(boolean b) //kutunun içine yazılan yazının değiştirilebilmesini sağlar veya engeller b=true yazı yazılabilir b=false yazı yazılamaz.

Metotları burada da aynen geçerlidir.TextArea'nın kurucu metotlarında şöyledir:

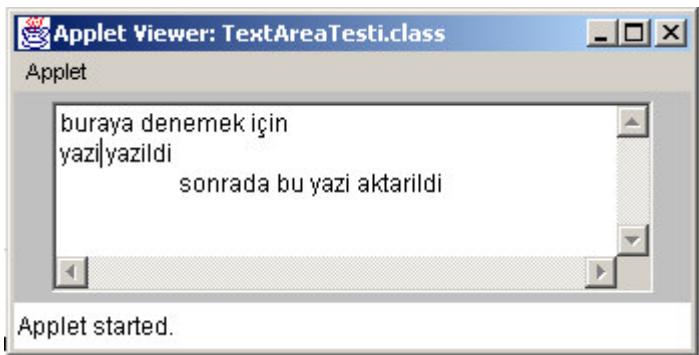
```
public TextArea()
public TextArea(int sütun,int satır)
    sütun x satır boyutunda yazı alanı tanımlar
public TextArea(String s)
    String değişkeni s nin boyuna göre yazı alanı tanımlar ve s yi içine yazar.
public TextArea(String s,int sütun,int satır)
    sütun x satır boyutunda yazı alanı tanımlar ve s yi içine yazar
public TextArea(String s,int sütun,int satır,int scrollbar)
    scroll bar (kontrolçubuğu) değişkeni şu değerleri alabilir :
    SCROLLBARS_BOTH    Hem yatay hem düşey kontrol çubuğu tanımlar
    SCROLLBARS_HORIZONTAL_ONLY  Yatay kontrol çubuğu tanımlar
    SCROLLBARS_NONE Kontrol çubuğu tanımlamaz
    SCROLLBARS_VERTICAL_ONLY Düşey kontrol çubuğu tanımlar
```

Şimdi TextArea metotunu Küçük bir örnekte kullanalım :

Program 7.14 TextAreaTesti.java programı

```
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\n\tsonradan bu yazi aktarildi";
//not \n satırbaşı kontrolü
//\t bir sonraki tab setine git
T1=new TextArea(s,5,40);
add(T1);
}
}
```

07012.JPG



Şekil 7.12 TextAreaTesti.java Programı applet çıktısı

Bu programda gördüğünüz \n ve \t string komutlarıdır. Alt satır geç ve gelecek tab noktasına git anlamına gelir. Bu komutlar aslında c dilinden alınmadır ve c dilindeki tüm \ komutları java stringlerinde de geçerlidir.

Şimdi TextArea metodunu biraz daha detaylı bir programda kullanalım

Program 7.15 : TextArea sınıfını kullanan TextAreaTesti1.java programı

```
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 yazı aktarıldı";
//not \n satırbaşına kontrol
//\t bir sonraki tab setine git
T1=new TextArea(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 yazıyı
// kopye düğmesine basıldığında T2 penceresine aktar
T2.setText(T1.getSelectedText());
}
public void textValueChanged(TextEvent e)
{
//Birinci pencereye yeni yazı yazıldığında ikinci pencereye aktar
TextComponent kaynak=(TextComponent)(e.getSource());
T2.setText(kaynak.getText());
}
}
```

07013.JPG



Şekil 7.13 **TextAreaTesti1.java** Programı sonuçlarını gösteren appletler

Bu programda **TextListener** kullanıldığından **textValueChange** metodu da kullanılmıştır. Buradaki metot birinci kutuya yazılan yazıyı ikinci kutuya kopyalamaktadır. Kopye düğmesine basılıncı ise sadece T1.getSelectedText() metotuyla seçilmiş olan text actionPerformed metodu üzerinden T2 Yazım Alanına (TextArea) aktarılmaktadır.

Java swing **JTextArea** sınıfı awt deki **TextArea** sınıfının benzeridir. Sınıfın kurucu metodlarının tanımı

```
public JTextArea();
public JTextArea(int rows,int cols);
public JTextArea(Document doc);
public JTextArea(Document doc, String Text, int satırsayı int sütun sayısı);
public JTextArea(String Text)
public JTextArea(String Text,int satırsayı int sütun sayısı)
şeklindedir.
```

JTextArea sınıfını kullanan Bir örnek problem verelim:

Program 7.16 **TextAreaTesti1SWF.java** programı

```
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\nyazı yazıldı\n\tsonrada bu yazı aktarıldı";
//not \n satırbaşı 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(b);
}
```

```

T2=new JTextArea(10,15);
T2.setEditable(false);
b.add(new JScrollPane(T2));
c.add(b);
}

public void actionPerformed(ActionEvent e)
{
// T1 penceresinden seçilen yazıyı
// kopye düğmesine basıldığında T2 penceresine aktar
T2.setText(T1.getSelectedText());
}
public void textValueChanged(TextEvent e)
{
//Birinci pencereye yeni yazı yazıldığında 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



Şekil 7.14 TextAreaTesti1SWF.java programı sonuçlarının JFrame penceresindeki görünümü

7.6 AWT CHOICE , SWING JCOMBOBOX SEÇİM SINIFLARI

awt kütüphanesi Choice (Seçim) sınıfı birden fazla maddeden seçim yapılması gerekiğinde kullanılan bir sınıfır.

Choice sınıfının kurucu metodu ve diğer metodlarının listesi aşağıda verilmiştir :

```

public Choice() // Kurucu metot
public String getItem(int indeks)
    İndeksteki seçimi verir
public synchronised void add(String s)
    verilen Stringi Choice listesine ekler.
public synchronised String getSelectedItem()
    Fareyle seçilmiş olan liste elemanını verir
public synchronised String insert(String s, int indeks)
    verilen stringi listeye indeks sırasındaki madde olarak ekle

```

```
public synchronized void remove(String s)
verilen stringi listeden siler
```

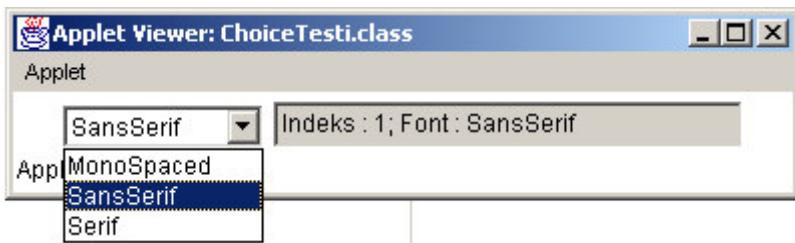
Choice (Seçim) sınıfını bir programda kullanalım :

Problem 7.17 Choice sınıfının kullanımı, 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 yukarıdaki ChoiceTesti.html de de görüleceği gibi aktarılırken ItemListener ve ItemEvent kullanılmıştır. Bu metotla kullanılması gereken method **ItemStateChanged metotudur.**

07015.JPG



Sekil 7.15 ChoiceTesti.java programının sonuçlarının appletde görülmesi

Java swing sınıfında Choice sınıfının benzeri Jchoice sınıfı mevcut değildir. Onun yerini JComboBox sınıfı almıştır

JComboBox sınıfının Kurucu metodları :

```
public JComboBox();
public JComboBox(ComboBoxModel m);
public JComboBox(Object obj[]);
public JComboBox(Vector v);
```

şeklindedir. Vector gelişmiş bir boyutlu değişken türüdür detayları 11 inci bölüm, gelişmiş java bilgi işleme yapıları kısmında verilmiştir.

ComboBox Jcomponent sınıfından türetilmiştir. ActionListener, ItemSelectable, ListDataListener ve Accessible sınıflarını implement eder. JComboBox sınıfı, seçilen sınıfı size veren veya seçilme opsyonunu tanımlayan şu sınıfları barındırır :

```
getSelectedItem() : bu noktadaki Object değerini çağırır.
getSelectedIndex() : bu noktanın indeks değerini çağırır
getSelectedObjects() : birden fazla seçim yapıldığında tüm seçilen object listesini iletir.
setSelectedIndex(int pozisyon) : pozisyon indeksini seçer
setSelectedItem(Object nesne) nesne 'yi seçer
```

Problem 7.18 JComboBox sınıfının kullanımı, 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 örneği");
        Container co=getContentPane();
        String Liste[]={ "MonoSpaced","SansSerif","Serif" };
        F=new JComboBox(Liste);
        T=new JTextField((String)F.getSelectedItem());
        T.setEditable(false);
        T.setFont(new Font((String)F.getSelectedItem(),Font.BOLD,12));
        F.addItemListener(new FontIsmiKontrolu(T));
        F.addItemListener(new TextFieldKontrolu(T));
        co.setLayout(NORTH);
        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 JCheckBoxTestiSWF.java programının sonuçlarının appletde görülmesi

7.7 AWT, CHECKBOX VE CHECKBOXGROUP VE SWING JCHECKBOX VE JRADIOBUTTON SINIFLARI

awt sınıfında yer alan Check box ve CheckBoxGroup sınıfları çeşitli opsiyonlar arasından seçim yapabilme olasılığını sağlayan sınıflardır.

Kurucu metotları :

Public CheckBox(String s)

Tek bir kare kontrol kutusu oluşturur. Kontrol kutusu başta boştur

Public CheckBox(String s, CheckBoxGroup c,boolean durum)

Tekbir daire şeklinde kontrol kutusu oluşturur ve bu kontrol kutusunu CheckBoxGroup nesnesine ilave eder.

Public CheckboxGroup()

CheckboxGroup nesnesini oluşturur bu gruba checkboxlar ilave edilebilir

CheckBox kullanımını anlamak amacıyla CheckBoxTesti.java programına göz atalım :

Problem 7.19 CheckBoxTest.java, CheckBox kullanım test programı

```

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("Italic");
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 programının sonuçlarının appletde görülmesi

Şimdi de checkBoxGroup programının çalışmasını ornekleyen RadioButtonTesti programına göz atalım. Burada guruplanmış olan checkbox'lardan sadece bir tanesini seçme izni verilmiştir.

Problem 7.20 : RadioButtonTesti.java programı

```

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("Italic",fontGurubu,true);
Italic.addItemListener(this);
add(Italic);
KalinItalic=new Checkbox("Kalin Italic",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



Şekil 7.18 RadioButtonTesti.java programı sonuçlarının applette gösterilmesi

Swing gurubundaki aynı işler için kullandığımız sınıflar JCheckBox, JRadioButton ve JButtonGroup sınıflarıdır. JCheckBox sınıfının kurucu metodları :

```

public JCheckBox()
public JCheckBox(Icon resim)
public JCheckBox(Icon resim,boolean kutuişareti)
public JCheckBox(String yazı)
public JCheckBox(String yazı, boolean kutuişareti)
public JCheckBox(String yazı, Icon resim)
public JCheckBox(String yazı, , Icon resim ,boolean kutuişareti)

```

Burada da daha önceki JLabel, JButton gibi sınıflarda gördüğümüz gibi yazının yanında resim yerleştirme seçimi de mevcuttur. Buradaki örnek problemde bir önceki problemin aynısı Jcheckbox ile çözülmüştür.

Problem 7.21 CheckBoxTestiSWF.java, JCheckBox kullanım test programı

```
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 örneği");
        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("İtalik");
        Italik.setMnemonic('İ');
        Italik.setSelected(false);
        Italik.addItemListener(this);
        Check.add(Italik);
        c.add(Check,BorderLayout.WEST);
        T=new JTextField("Buradaki yazının değişmesini 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



Şekil 7.19 CheckBoxTestiSWF.java programının sonuçlarının JFrame'de görülmesi

Java swing JRadioButton sınıfı Jcheckbox tanımına benzerdir. Kurucu metodları :

```

public JRadioButton()
public JRadioButton(Icon resim)
public JRadioButton(Icon resim,boolean kutuişareti)
public JRadioButton(String yazı)
public JRadioButton(String yazı, boolean kutuişareti)
public JRadioButton(String yazı, Icon resim)
public JRadioButton(String yazı, , Icon resim ,boolean kutuişareti)

```

Burada JRadioButton sınıfına ilave olarak bir tane daha ilave olarak bir tane daha yardımcı sınıf kullanacağız. Bu sınıf ButtonGroup sınıfıdır. Bu sınıf düğmeleri bir gurup altında toplayarak sadece bir tanesinin basılı durumda olması işlevini kontrol eder. Bu sınıfın tanımı :

```

public class ButtonGroup extends Object implements Serializable
{
protected Vector düğmeler;
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);
}

```

Şeklindedir. Şimdi yine bir öncekine benzer bir örnekte JradioButton ve ButtonGroup sınıflarının birarada kullanımını görelim.

Problem 7.22 RadioButtonTestiSWF.java, JRadioButton kullanım test programı

```

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,kalinalitalik;
private ButtonGroup gurup;
int KalinAyari=Font.PLAIN;;
int ItalikAyari=Font.PLAIN;

public RadioButtonTestiSWF()
{
super("Radio Button örneği");
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("Kalink");
kalin.setMnemonic('K');
kalin.setSelected(false);
kalin.addItemListener(this);
gurup.add(kalin);
Check.add(kalin);

italik=new JRadioButton("İtalik");
italik.setMnemonic('t');
italik.setSelected(false);
italik.addItemListener(this);
gurup.add(italik);
Check.add(italik);

kalinitalik=new JRadioButton("Kalink-İtalik");
kalinitalik.setMnemonic('a');
kalinitalik.setSelected(false);
kalinitalik.addItemListener(this);
gurup.add(kalinitalik);
Check.add(kalinitalik);

c.add(Check,BorderLayout.WEST);

T=new JTextField("Buradaki yazının değişmesini 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



Şekil 7.20 RadioButtonTestiSWF.java programının sonuçlarının JFrame'de görülmesi

Son olarak bu guruptan JToggleButton sınıfına degenelim. JToggleButton sınıfı işlevsel olarak Jcheckbox sınıfının aynıdır. Sadece basılım alanı düğme(button) şeklindedir. Jcheckbox için verdigimiz programın JTtoggleButton'a dönüştürülülmüş şekli aşağıda verilmiştir.

Problem 7.23 ToggleButtonTestiSWF.java, JToggleButton kullanım test programı

```

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

public class ToggleButtonSWF extends JFrame implements ItemListener
{
    private JTextField T;
    private JTtoggleButton Kalin, Italik;
    int KalinAyari=Font.PLAIN;;
    int ItalikAyari=Font.PLAIN;

    public ToggleButtonSWF()
    {
        super("ToggleButton örneği");
        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 JTtoggleButton("Kalink");
        Kalin.setMnemonic('K');
        Kalin.setSelected(false);
        Kalin.addItemListener(this);
        Check.add(Kalin);
        Italik=new JTtoggleButton("İtalik");
        Italik.setMnemonic('t');
        Italik.setSelected(false);
    }
}

```

```

Italik.addItemListener(this);
Check.add(Italik);
c.add(Check,BorderLayout.WEST);
T=new JTextField("Buradaki yazının değişmesini 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



Şekil 7.21 ToggleButtonTestiSWF.java programının sonuçlarının JFrame'de görülmesi

7.8 MOUSE(FARE) KONTROLU

Modern bilgisayarlarda bilgisayar işlemlerinin çok büyük bir kısmı mouse (fare) ile kontrol edilir. Javada gurubunda temel fare kontrollerini sağlayan sınıflar MouseListener ve MouseMotionListener sınıflarıdır. Bu sınıfların en çok kullanılan metodları şunlardır :

1. MouseListener Metotları :

```

public void mousePressed(MouseEvent e)
bir mouse düğmesi basılınca çağırılır
public void mouseClicked(MouseEvent e)
bir mouse düğmesi basılıp bırakılırsa çağırılır.
public void mouseReleased(MouseEvent e)
Bir mouse düğmesi basılı olarak mouse çekildikten sonra düğme bırakılınca çağırılır.
public void mouseEntered(MouseEvent e)
mouse pencere içine girince çağırılır.
public void mouseExited(MouseEvent e)
mouse pencere çerçevesinin dışındaysa çağırılır.

```

2. MouseMotionListener Metotları

```
public void mouseDragged(MouseEvent e)
Mouse düğmesi basılı iken mouse hareket ettirilirse çağırılır
public void mouseMoved(MouseEvent e)
Mouse hareket ederse çağırılır.
```

Bu metodların kullanımlarını göstermek amacıyla [FareTesti.java](#) programı aşağıda verilmiştir. Bu program awt Applet programı olarak hazırlanmıştır.

Program 7.24 : FareTesti.java programı

```
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 çıktı metotları
    public void mouseClicked(MouseEvent e)
    { degerleriGir("Kliklendi",e.getX(),e.getY()); }
    public void mousePressed(MouseEvent e)
    { degerleriGir("Basıldı",e.getX(),e.getY()); }
    public void mouseReleased(MouseEvent e)
    { degerleriGir("Bırakıldı",e.getX(),e.getY()); }
    public void mouseEntered(MouseEvent e)
    { showStatus("Fare applet alanı içinde"); }
    public void mouseExited(MouseEvent e)
    { showStatus("Fare applet alanı dışında"); }
    //MouseMotionListener (fare hareket dinleyicisi) metotları
    public void mouseDragged(MouseEvent e)
    { degerleriGir("basılıp çekiliyor",e.getX(),e.getY()); }
    public void mouseMoved(MouseEvent e)
    { degerleriGir("hareket halinde",e.getX(),e.getY()); }
}
```

07022.JPG



Şekil 7.22 FareTesti.html applet çıktısı

Aynı programın JFrame'de yazılmış bir benzeri aşağıda sunulmuştur.

Problem 7.25 FareTestiSWF.java, Fare(Mouse) kontrol sınıfları kullanım test programı

```
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 çıktı metodları
public void mouseClicked(MouseEvent e)
{ degerleriGir("Kliklendi",e.getX(),e.getY()); }
public void mousePressed(MouseEvent e)
{ degerleriGir("Basıldı",e.getX(),e.getY()); }
public void mouseReleased(MouseEvent e)
{ degerleriGir("Bırakıldı",e.getX(),e.getY()); }
public void mouseEntered(MouseEvent e)
{ setTitle("Fare applet alanı içinde"); }
public void mouseExited(MouseEvent e)
{ setTitle("Fare applet alanı dışında");
degerleriGir("Fare applet alanı dışında",e.getX(),e.getY());
}

//MouseMotionListener (fare hareket dinleyicisi) metodları
public void mouseDragged(MouseEvent e)
{ degerleriGir("basılıp çekiliyor",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



Şekil 7.23 FareTestiSWF.java JFrame penceresi çıktısı

7.9 SWING JTABLE SINIFI

JTable şu ana kadar gördüğümüz sınıflara göre daha kompleks bir yapı arzeder, fakat oldukça kullanışlı bir yapıdır ve sadece swing kütüphanesinde tanımlanmıştır.

JTable sınıfının kurucu sınıfları :

```

public JTable();
public JTable(TableModel tm);
public JTable(TableModel tm, TableColumnModel tcm);
public JTable(TableModel tm, TableColumnModel tcm, ListSelectionModel lm);
public JTable(int satirsayısı,int sütunsayısı);
public JTable(Object[][] tablogirdisi ,Object[] tablobaşlığıgirdisi);
public JTable(Vector tablogirdisi ,Vector tablobaşlığıgirdisi);

```

şeklindedir. Kurucu metodlarda girdi olarak görünen TableModel sınıfı

```

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 satır, int sütun);
    public abstract boolean isCellEditable(int satır, int sütun);
    public abstract void removeTableModelListener(TableModelListener l);
    public abstract void setValueAt(Object o,int satır,int sütun);
}

```

görüldüğü gibi bu sınıf interface olduğundan bu sınıfın türetilen sınıfların bu sınıfın tüm metodları tanımlamaları gereklidir. Bu yüzden AbstractTableModel gibi Table Model'den türetilen sınıflar üzerinden yeni sınıfın 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 satır,int sütun);
    public void fireTableChanged(TableModelEvent e);
    public void fireTableDataChanged();
    public void fireTableRowsDeleted(int satır,int sütun);
    public void fireTableRowsInserted(int satır,int sütun);

```

```

public void fireTableRowsUpdated(int satır,int sütun);
public void fireTableStructureChanged();
public abstract Class getColumnClass(int sütun);
public abstract String getColumnName(int Column);
public abstract boolean isCellEditable(int satır, int sütun);
public abstract void removeTableModelListener(TableModelListener l);
public abstract void setValueAt(Object o,int satır,int sütun);
}

```

Örnek problemlerde önce AbstractTableModel'den türetilen ve Tablonun giriş değerlerini veren TableModel sınıfı yaratılmış, sonra bu sınıf tabloda girdi olarak kullanılmıştır.

Problem 7.26 tableSWF_2000.java, TableModel, JTable ve TableModelListener sınıfları kullanım test programı

```

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={{ "satır bir - sütun bir","satır bir - sütun iki"},  

                    {"satır iki - sütun bir","satır 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 satır,int sütun) {return veri[satır][sütun];}
    public String getColumnName(int c) {return baslik[c];}
    public void setValueAt(Object val, int row, int col)
    {
        veri[row][col] = val;
        // Değişimin olduğunu göster:
        fireTableDataChanged();
    }
    public void tableChanged(TableModelEvent e)
    {
        String s="Tablodaki değer değiştirildi ";
        JOptionPane.showMessageDialog(null,s,"Tablodaki değer değiştirildi",
        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 örneği");
        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



Şekil 7.24 26 tableSWF_2000.java JFrame ve JOptionPane penceresi çıktısı

Problem 7.27 table1SWF_2000.java, TableModel, JTable ve TableModelListener sınıfları kullanım test programı

```

// JTable örneği

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"},
        {"beş", "altı", "yedi", "sekiz"},
        {"dokuz", "on", "onbir", "oniki"},
    };
    String[] baslik={ "sütun 1","sütun 2","sütun 3","sütun 4"};

    // Tablo dinleme metodu ve sınıfı : Tabloda bir değişiklik yapıldığında
    // JOptionPane ile yeni değişikliği 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 değiştirildi",
                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;
    // Değişimin olduğunu göster:
    fireTableDataChanged();
}
public boolean
isCellEditable(int row, int col) {
    return true;
}
};

public class table1SWF_2000 extends JFrame {
    public table1SWF_2000()
    {
        super("Tablo örneği");
        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



Şekil 7.25 table1SWF_2000.java JFrame ve JOptionPane çıktılarını görünümü

7.10 AWT, LIST SINIFI VE JAVA SWING JLIST SINIFI , LİSTEDEN SEÇİM

Bu gurup aslında bölüm 7.6 da tanımlanan awt Choice , swing JComboBox seçim sınıflarının bir devamı niteliğindedir.

List ve Jlist sınıfları bir listeden seçim yapmaya ve bu seçim sonucu gerekli işlemleri oluşturmaya yarar. List sınıfı ItemListener ve ActionListener sınıf implementlerini kullanır. Örnek programımızda Listeden seçilen renge göre arkaplan rengi değişmektedir.

Program 7.28 ListTesti.java programı

```

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 ḡrilebilecel bir liste ac
        //ayn̄ 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



Şekil 7.26 ListTesti.java programı sonuçların appletviewer da görünümü

List sınıfının swing grubundaki paraleli daha gelişmiş bir liste kullanma metodu **JList** metodudur. Jlist metodu pencereye ilave edilirken direkt olarak ilave edilmez, JScrollPane alt pencere programı üzerinden ilave edilir. Aynı zamanda dinleme metodu olarak ListSelectionListener metodu kullanılır. Bu metod java swing kütüphanesi javax.swing.event de tanımlanmıştır.

JList metoduyla bir seçim yapıldığı gibi birden fazla seçim yapmakta mümkündür.

JList metodunun kurucu metodları :

public JList();

```
public JList(ListModel lm);
public JList(Object o[]);
public JList(Vector v);
```

dir.

Seçimde kullanılan bazı önemli metodları :

```
public int GetSelectedIndex() : seçilen (veya en son seçilen) elemanın indisini verir
public int[] getSelectedIndices() : seçilen elemanların (birden fazla olabilir) indislerini boyut üzerinden aktarır
public Object getSelectedValue() : son seçilen elemandaki değeri Object olarak aktarır.
public object[] getSelectedValues() : seçilen elemanlardaki değerleri boyutlu object olarak aktarır
void setSelectionMode(int mode) : tek eleman mı seçileceğini yoksa çok sayıda eleman mı seçileceğini belirler.
Girdi int değişkeni
ListSelectionModel.SINGLE_SELECTION (tek eleman seçimi)
ListSelectionModel .INTERVAL_SELECTION (bir gurup seçimi)
ListSelectionModel .MULTIPLE_INTERVAL_SELECTION (birden fazla gurup seçimi)
Değerlerinden birini alabilir.
Public void setVisibleRowCount(int sayı) : listenin kaç elemanın görüntüleneceğini verir.
```

Listenin kenar kayma çubukları, listenin içinde yer alacağı JScrollPane penceresi tarafından oluşturulur.

JScrollPane metodunun kurucu metodları

```
public JScrollPane();
public JScrollPane(int düşeyçubukçeşidi,int yatayçubukçeşidi);
public JScrollPane(Component pencere);
public JScrollPane(Component pencere ,int düşeyçubukçeşidi,int yatayçubukçeşidi);
```

Buradaki pencere JScrollPane'in içinde yeralacağı container'ı gösterir. Düşeyçubukçeşidi
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS : her zaman düşey kaydırma çubuğu tanımla
JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED : sadece gerektiği zaman zaman düşey kaydırma çubuğu tanımla
JScrollPane.VERTICAL_SCROLLBAR_NEVER : hiçbir zaman düşey kaydırma çubuğu tanımlama
Değerlerini alır. Yatayçubukçeşidi ise
JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS : her zaman yatay kaydırma çubuğu tanımla
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED : sadece gerektiği zaman yatay kaydırma çubuğu tanımla
JScrollPane.HORIZONTAL_SCROLLBAR_NEVER : hiçbir zaman yatay kaydırma çubuğu tanımlama
Değerlerini alır.

Birkaç örnek probleme Jlist sınıfını vermekte çalışalım. İlk örnek yukarıdaki JComboBox problemlerin aynı işlemiini yapmaktadır. Listededen yapılan seçime göre JTextField alanındaki fontu değiştirecektir. Listenin iki elemanın görünmesi müsadesi verilmiştir. Liste tek seçimli listedir.

Problem 7.29 JListTestiSWF.java, JList sınıfı kullanım test programı

```
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 örneği");
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



Şekil 7.27 JListTestiSWF.java programı sonuçlarının JFrame penceresinde görünümü

Bu örnekte ise listeden yapacağımız seçime göre arka plan rengini değiştiriyoruz. Bu program yukarıda verilen List programının oldukça benzeridir. Listede dört elemanın görülmesi izni verilmiştir. Liste tek seçimli bir listedir.

Problem 7.30 JlistTesti1SWF.java, JList sınıfı kullanım test programı

```

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", "camgöbeği", "Koyu Gri", "Gri",
    "Yeşil", "Açık gri", "mor", "Portakal", "Pembe", "Kırmızı", "beyaz", "sarı" };
    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 alanındaki olan olayları 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



Şekil 7.28 JlistTesti1SWF.java programı sonuçlarının JFrame penceresinde görünümü

Diger bir Liste kullanma örneği : Bu örnekte Listeye JTextField üzerinden ilave yapabiliyoruz ve çıkarabiliyoruz. Örnek Java döküman kütüphanesinden alınıp adapte edilmiştir.

Problem 7.31 ListDemoSWF.java, JList sınıfı kullanım test programı

```
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 = "İşe al";
    private static final String fireString = "İşten at";
    private JButton fireButton;
    private JTextField employeeName;

    public ListDemoSWF() {
        super("ListDemo");
        listModel = new DefaultListModel();
        listModel.addElement("Turhan Çoban");
        listModel.addElement("İrfan Çoban");
        listModel.addElement("Birsen Çeliker");
        listModel.addElement("Nurhan Erel");

        //Listeyi oluştur ve scrollPane'e yerleştir
        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 kalmışsa çağırılır.
            int index = list.getSelectedIndex();
            listModel.remove(index);
        }
    }
}
```

```

        int size = listModel.getSize();
        //Listede eleman kalmadı işten atmayı 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) {

        //Eğer yeni isim girilmemişse
        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



Şekil 7.29 ListDemoSWF.java programı sonuçların JFrame penceresinde görünümü

Listelerin sadece tek seçim değil aynı zamanda birden fazla seçim yapmak içinde kullanıldığından bahsetmiştık. Aşağıdaki örnekte listeden Liste türü değişkeninde yaptığımız değişiklikle göre bir veya birden fazla seçim yapabiliyoruz. Bu program hem JFrame hem de Japplet olarak çalışabilmektedir.

Problem 7.32 ListSelectionDemo.java, JList sınıfı kullanım test programı

```
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;

    //bazı sistemlerde ana kurucu metod olmayınca hata verebileceği
    // için tanımlandı.
    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",
            "beş", "altı", "yedi" };
        String[] columnNames = { "Fransızca", "İngilizce", "İtalyanca" };
        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 alanını JPanel olarak oluştur (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);
//Çıktı alanı oluştur
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 örneği");
    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();

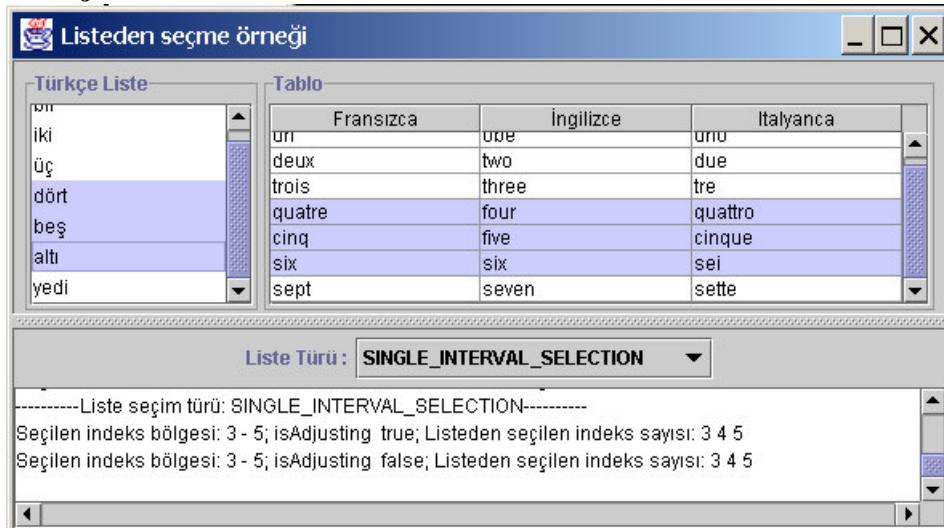
        int firstIndex = e.getFirstIndex();
        int lastIndex = e.getLastIndex();
        boolean isAdjusting = e.getValueIsAdjusting();
        output.append("Secilen indeks bölgesi: "
            + firstIndex + " - " + lastIndex
            + "; isAdjusting " + isAdjusting
            + "; Listeden seçilen indeks sayısı:");

        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



Şekil 7.30 ListSelectionDemo.java programı sonuçların JFrame penceresinde görünümü (not bu program hem applet hem frame olark görüntülenebilir)

7.11 SWING JMENU

Window türü programlamaya alışık olan herkes menu kullanmaya da alışktır. Menuler pencerenin tepesinden açılan ve çeşitli görevler yüklenen komponentlerdir. Java swing menu'leri temel olarak JMenuBar, JMenu ve JMenuItem sınıflarından oluşur. Ayrıca menulerde JcheckboxMenuItem, JradioMenuItem gibi alt elemanlar da bulunabilir.

JMenuBar kurucu metodu :

```
Public JMenuBar();
```

Seklindedir.

Bazı önemli metodları :

```
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,Mouse
public void setBorderPainted(boolean b);
public void setHelpMenu(Jmenu menu);
public void setMargin(Insets in);
public void setSelected(Component c):
```

Jmenu kurucu metodları :

```
public JMenu();  
public JMenu(String Label);  
public JMenu(String label boolean menuudenavır);
```

Bazı önemli JMenu metodları :

```
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 addSeparator();
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 setDelay(int delay);
public void setMenuLocation(int x,int y);
public void setPopUpMenuVisible(boolean b)
public void setSelected(boolean b);
```

JMenuItem kurucu metodları :

```
public JMenuItem();
public JMenuItem(Icon resim);
public JMenuItem(String Label);
public JMenuItem(String Label,int mnemonic);
public JMenuItem(String Label,Icon resim);
bazı metodları :
public Component getComponent();
public menuElement[] getSubElements();
```

JCheckBoxMenuItem, JMenuItem sınıfının bir alt sınıfı olarak oluşturulmuştur. Davranışı JCheckBox sınıfı gibidir. Kurucu metodları :

```
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'ın alt sınıfı olarak oluşturulmuştur. Kurucu metodları :

```
public JRadioButtonMenuItem ();
public JRadioButtonMenuItem (Icon resim);
public JRadioButtonMenuItem (String Label);
public JRadioButtonMenuItem (String Label,Icon resim);
```

Şimdi bu metodları kullanan bir örnek menu oluşturalım. Örnek menu sadece menu yapısını göstermek amaçlı olduğundan fazla bir eyem yamamaktadır. Gerçek programlarınızda gerekli eylemleri yereştirebilirsiniz.

Problem 7.33 MenuDemo.java, JMenu,JMenuBar,JMenuItem,JCheckBoxMenuItem ve JRadioButtonMenuItem sınıfları kullanım test programı

```
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 yazı 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 olduğundan bu eleman bir şey yapmıyor");
        menuItem.addActionListener(this);
        menu.add(menuItem);

        menuItem = new JMenuItem("Yazı 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 düğmesi menu elemani");
rbMenuItem.setSelected(true);
rbMenuItem.setMnemonic('r');
group.add(rbMenuItem);
rbMenuItem.addActionListener(this);
menu.add(rbMenuItem);
rbMenuItem = new JRadioButtonMenuItem("ikinci bir radyo düğmesi");
rbMenuItem.setMnemonic('t');
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("İkinci bir Menu");
menu.setMnemonic('n');
menu.getAccessibleContext().setAccessibleDescription(
    "Bu menu hiç bir şey yapmaz ve alt elemanı yoktur");
menuBar.add(menu);
}

```

```

public void actionPerformed(ActionEvent e) {
    JMenuItem source = (JMenuItem)(e.getSource());
    String s = "Action event kaydedildi"
        + newline
        + "  Event kaynağı: " + source.getText()
        + " (sınıfın adı " + getClassName(source) + ")";
    output.append(s + newline);
}

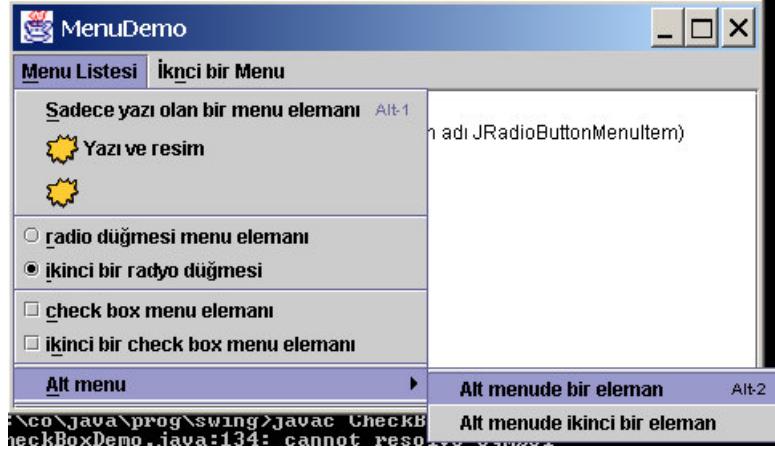
public void itemStateChanged(ItemEvent e) {
    JMenuItem source = (JMenuItem)(e.getSource());
    String s = "Item event kaydedildi."
        + newline
        + "  Event kaynağı: " + source.getText()
        + " (sınıfın adı " + getClassName(source) + ")"
        + newline
        + "  New state: "
        + ((e.getStateChange() == ItemEvent.SELECTED) ?
            "seçildi":"seçim iptal edildi");
    output.append(s + newline);
}

// Sınıfın adını 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



Şekil 7.31 MenuDemo.java programı sonucların JFrame penceresinde görünümü

7.12 SWING JSLIDER SINIFI VE JPANEL TEMEL ÇİZİM ELEMANI (PANELİ)

JSlider sınıfı bir değişkenin değerini verilen sınırlar içerisinde değiştirmeye yarayan bir grafik kullanıcı arayüzü sınıfıdır. JSlider sınıfının kurucu metodları :

```

public JSlider();
public JSlider(BoundedRangeModel brm);

```

```
public JSlider(int konum);
public JSlider(int minimumdeğer, int maksimumdeğer);
```

Problem 7.34 ovalPanelSWP.java, JPanel sınıfı

```
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 sınıfı test programı

```
import javax.swing.*;      // java swing sınıfını çağır
import java.awt.*;         // java pencere kullanma sınıfını çağır
import java.awt.event.*;   // java pencereyi dinleme sınıfını çağır
import BasicWindowMonitor;
import ovalPanelSWP;
import javax.swing.event.*;

public class ovalTestSWF_2000 extends JFrame implements ChangeListener
{
    // Renk seçme örneği
    private ovalPanelSWP p;
    private JSlider cap;
    Color r=Color.lightGray;
    Container c;
    // pencereyi başlatma metodu
```

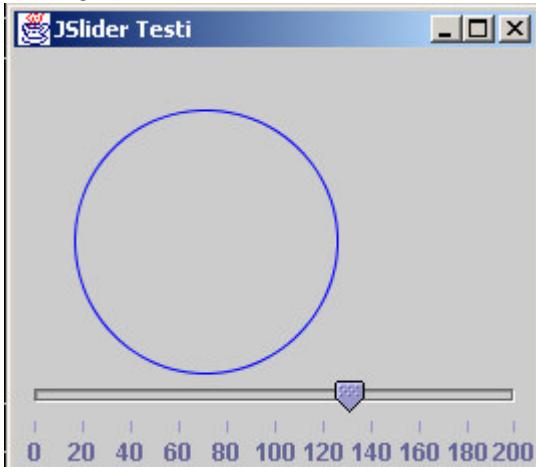
```

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 alanındaki olan olayları 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



Şekil 7.32 ovalPanelTestSWF_2000.java programı sonuçlarının JFrame penceresinde görünümü

Bu programda uyguladığımız bir çizim şecline daha detaylı göz atalım. Daire JPanel sınıfında çizilmiştir. Daireyi çizmek için ovalPanelSWP sınıfını yarattık. JPanel sınıfında kurucu metodlarını herhangi bir sınıfta olduğu gibi tanımladık ve **public void paintComponent(Graphics g)** metodu ve Graphics sınıfı üzerinden çizimimizi tanımladık. Tekrar çizdirmek için repaint() metodunu kullandık. İkinci JFrame proramımızda

```

private ovalPanelSWP p;
p=new ovalPanelSWP(30,30,30);
Container c;
c=getContentPane();
c.add(p,BorderLayout.CENTER);

```

komutlarını kullanarak panelimizi ana panele monte ettik. StateChanged event metodundaki

```
p.setOvalPanelSWP(30, 30,cap.getValue());
```

komutuyla da çapını değiştirip
repaint();

komutuyla tekrar çizdirdik.

Panel programında

```
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,RenderingHints.VALUE_ANTIALIAS_ON);  
komutu kullanıldığından dairenin oldukça düzgün çizilmiş olduğuna dikkatinizi çekeriz.
```

7.13 FLOWLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI

Grafik arayüz programlarının daha düzgün bir şekilde gösterilmesi amacıyla **Layout** sınıfları kullanılır. **awt** grafik interface’inde default değer olarak kullanılan Layout sınıfı **FlowLayout** sınıfıdır. Swing arayüz programlarında ise **BorderLayout** sınıfı default olarak kullanılır.

Flow layout sınıfı grafik ortamına (Panel, Applet, Frame, JPanel, JApplet, JFrame vs.) alt elemanları yerleştirirken soldan sağa doğru akleme (add) sırasıyla yerleştirir. Penceredeki yer bittiye bir alt satırda geçer.

Kurucu metodu şu şekillerde tanımlanır :

```
public FlowLayout()  
public FlowLayout(int formatbolgesi)
```

format bölgesi değişkeni

```
FlowLayout.RIGHT,  
FlowLayout.CENTER veya  
FlowLayout.LEFT
```

değerlerini alabilir. Bu değerlere göre içeriği elemanları sağa ortaya veya sola yerleştirir.

```
public FlowLayout(int formatbolgesi,int yatay_bosluk,int dikey_bosluk);
```

format bölgesi **FlowLayout.RIGHT**, **FlowLayout.CENTER** veya **FlowLayout.LEFT** değerlerini alabilir.

Yatay_bosluk pixel olarak yatay posisyonda elemanlar arasında kalan boşluğu gösterir. Dikey bosluk ise dikey doğrultudaki pixel olarak boşluktur.

Küçük bir örnek probleme **FlowLayout** kullanımını daha açık hale getirebiliriz.

Program 7.36 : FlowLayout.java programı

```
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



Şekil 7.33 FlowLayout sınıfı ve sonuçlarının Appletde görülmesi

Aynı programın bir de swing JFrame versiyonunu inceleyelim.

Program 7.37 : FlowLayoutTestiSWF_2000.java programı

```

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 sınıfı ve sonuçlarının JFrame penceresinde görülmesi

7.14 BORDERLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI

BorderLayout sınıfı GUI elemanlarını beş bolmeye ayıracak guruplar, kuzey(NORTH), güney(SOUTH), doğu(EAST), Batı(WEST), orta(CENTER), Kuzey Appletin üst ısmi anlamına gelir. Bu sınıf swing grafik kütüphanesinde default formatlama sınıfı olarak seçilmişdir. BorderLayout sınıfının kurucu metotları şunlardır :

```

public BorderLayout()
public BorderLayout(int yatay_bosluk,int dikey_bosluk);
yatay_bosluk ve dikey_basluk pixel olarak alınır. GUI elemanlarının arasındaki boşluk miktarını
tanımlar.

```

Şimdi bir örnekle sınıfı açıklamaya çalışalım.

Program 7.38 : BorderLayoutTesti.java Programı

```

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 JButton(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();
}
}

```

Şekil 7.11.1 BorderLayoutTesti.java programı

07035.JPG



Şekil 7.35 BorderLayoutTesti.java programının sonuçlarının applette görülmesi

Program 7.39 : BorderLayoutTestiSWF_2000.java Programı

```

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","Doğuyu Gizle",
"Batiyi Gizle","Ortayı 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



Şekil 7.36 BorderLayoutTestiSWF_2000.java programının sonuçlarının JFrame penceresinde görülmesi

7.15 GRIDLAYOUT SINIFI KULLANARAK GUI FORMATLANMASI

GridLayout sınıfı pencereyi (Applet,Japplet,Panel,JPAnel,Frame,JFrame vs.) eşit boyutta gridlere böler ve her grid elemanının içine bir GUI yerleştirilebilmesine olanak sağlar. GridLayout sınıfının kurucu metotları şöyledir :

```

public GridLayout(int satırlar,int sütunlar)
publicGridLayout(int satırlar,int sütunlar,int yatay_bosluk,int dikey_bosluk);
yatay_bosluk ve dikey_basluk pixel olarak alınır. GUI elemanlarının arasındaki bosluk miktarını
tanımlar.

```

Şimdi bir örneklerle bu sınıfı açıklamaya çalışalım.

Program 7.40 GridLayoutTesti.java programı

```

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()
{

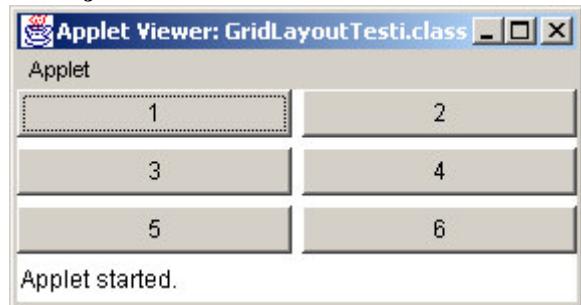
```

```

setLayout(new GridLayout(3,2,5,5));
B=new JButton[6];
for(int i=0;i<B.length;i++)
{
B[i]=new JButton(S[i]);
add(B[i]);
}
}
}
}

```

07037.JPG



Şekil 7.17 GridLayoutTesti.java programı sonuçlarının applette Görülmesi

Programın birde JFrame eşdeğerini inceleyelim.

Program 7.41 : GridLayoutTestiSWF_2000.java Programı

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class GridLayoutTestiSWF_2000 extends JFrame implements ActionListener
{
private JButton B[];
private String S[]={ "Düğme 1","Düğme 2","Düğme 3",
"Düğme 4","Düğme 5","Düğme 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);
}
}

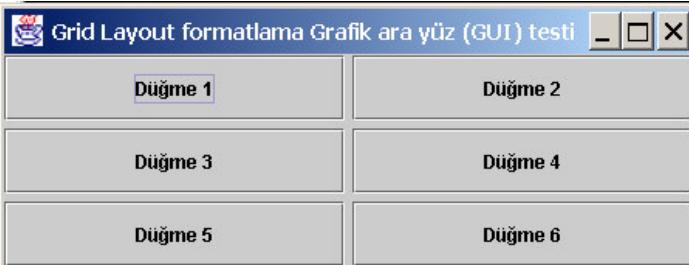
```

```

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



Şekil 7.38 GridLayoutTestiSWF_2000.java programı sonuçlarının JFRAme'de görülmesi

7.16 JTABBEDPANE SINIFI KULLANILARAK FORMATLAMA

JTabbedPane sadece swing sınıfında kullanılabilir. Değişik sayfalar halinde bir grafik arayüzü çağırır. Bu sayfalardan istenilen seçenekler kullanılabılır. Her sayfa bağımsız bir grafik ortamı oluşturur.

Program 7.42 : JTabbedPaneTestiSWF_2000.java Programı

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class JTabbedPaneTestiSWF_2000 extends JFrame implements ActionListener
{
private JButton B[];
private String S[]={ "Düğme 1","Düğme 2","Düğme 3",
"Düğme 4","Düğme 5","Düğme 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]+" düğmeye basıldı");
}
}

```

```

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



Şekil 7.39 JTabbedPaneTestiSWF_2000.java programı sonuçlarının JFrame'de görülmesi

7.17 JSPLITPANE SINIFI KULLANILARAK FORMATLAMA

JSplitPane ekranı ikiye böler. Bu ikiye bölme işlemlerini yineleyerek istediğimiz alt parçaları oluşturabiliriz. JSplitPane ile bölünen alt parçalar fare ile çekilerek büyütülp 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 parçacığında ekranı önce B[0] ve B[1] düğmelerini yerleştirdiğimiz yatay iki parça, sonra B[2] ve B[3] düğmelerini yerleştirdiğimiz iki yatay parça böldükten sonra bu parçaları (jsp[0] ve jsp[1]) iki düşey parça ayıriyoruz. Böylece birbirinden bağımsız dört ayrı alt parça (split) oluşturuyoruz.

```

Program 7.43 : JSplitPaneTestiSWF_2000.java Programı

```

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

public class JSplitPaneTestiSWF_2000 extends JFrame implements ActionListener
{
private JButton B[];
private String S[]={ "Düğme 1","Düğme 2","Düğme 3","Düğme 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];

```

```

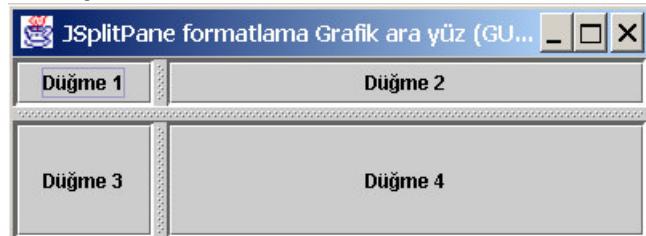
for(int i=0;i<B.length;i++)
{
B[i]=new JButton(S[i]);
B[i].addActionListener(this);
}
jsp[0]=new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,true,B[0],B[1]);
jsp[1]=new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,true,B[2],B[3]);
jsp[2]=new JSplitPane(JSplitPane.VERTICAL_SPLIT,true,jsp[0],jsp[1]);
c.add(jsp[2]);
}

public void actionPerformed(ActionEvent e)
{
for(int i=0;i<B.length;i++)
{
if(e.getSource()==B[i])
B[i].setText(S[i]+" düğmeye basıldı");
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



Şekil 7.40 JSplitPaneTestiSWF_2000.java programı sonuçlarının JFrame'de görülmesi

7.18 JSCROLLPANE SINIFI KULLANILARAK FORMATLAMA

JScrollPane sınıfını Jlist sınıfını incelerken görmüştük. Bu formatlama sınıfı Scrollbar adını verdigimiz yatay ve düşey kaydırma çubukları yardımıyla küçük bir pencereye sığmayacak kadar büyükolan grafiklerin kullanılmasını sağlar.

Program 7.44 : JScrollPaneTestiSWF_2000.java Programı

```

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

public class JScrollPaneTestiSWF_2000 extends JFrame implements ActionListener
{
private JButton B[];
private String S[]={ "Düğme 1","Düğme 2","Düğme 3","Düğme 4",
"Düğme 5","Düğme 6","Düğme 7","Düğme 8","Düğme 9","Düğme 10","Düğme 11",
"Düğme 12","Düğme 13","Düğme 14","Düğme 15","Düğme 16"};
private Container c;
JScrollPane jscroll;

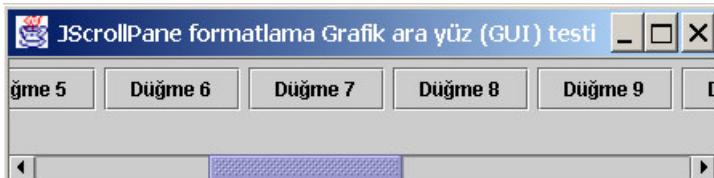
```

```

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]+" düğmeye basıldı");
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



Şekil 7.41 JScrollPaneTestiSWF_2000.java programı sonuçlarının JFrame'de görülmesi

7.19 PANEL VE JPANEL SINIFI

Daha kompleks formatlamalara gereksinme duyulduğunda panel sınıfını kullanabiliriz. Panel sınıfı applet yerleştirebileceğimiz alt bolgeler oluşturur. Hemen bir ornekle paneli nasıl kulandığımızı açıklamaya çalışalım.

Program 7.45 PanelTesti.java programı

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Paneltesti extends Applet
{

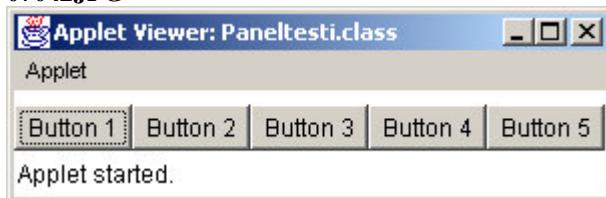
```

```

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);
}
}

```

07042.JPG



Şekil 7.42[PanelTesti.java](#) programı ve sonuçlarının applette görülmesi

JPanel sınıfını daha önce çeşitli alıştırmalarda zaten kullanmıştık. JPanel'i formatlama işleminde aynı Panel sınıfında olduğu gibi bir alt formatlama elemanı olarak kullanabiliriz. Zaten bütün swing programlarının temeli de JPanel sınıfıdır. Swing gurubunda bir üstteki problemin aynısını JApplet olarak yineledik. Awt penceresi ve swing penceresi arasındaki en önemli farkın awt de add terimini appletin bir alt terimi gibi kullanabilirken, swingde getContentPane() deyimiyle ana JPanel'i çağrırmak ve çizimleri bunun üzerine yapmak olduğunu bir kere daha yineleyelim.

Program 7.46 : PanelTestiSWA.java Programı

```

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("Düğme "+(i+1));
    buttonPanel.add(buttons[i]);
}
Container c=getContentPane();
c.setLayout(new BorderLayout());
c.add(buttonPanel,BorderLayout.SOUTH);
}

```

}

07043JPG



Şekil 7.43 PanelTestiSWA.java programı ve sonuçlarının JAppletde appletviewer ile görülmesi

Şu ana kadarki format örneklerinde anlatımı basitleştirmek amacıyla hep düğmeler (button) kullandık. Şimdi de başka bir problemde Panel sınıfının kullanılmasına bakalım :

Program 7.47 : PolinomunKokleri.java programı

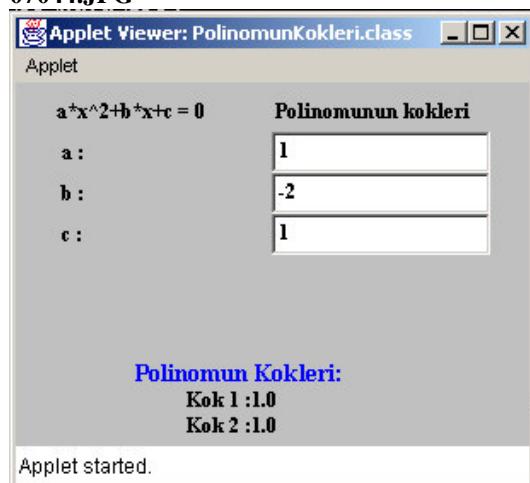
```
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
public class PolinomunKokleri extends Applet implements ActionListener
{
// polinom a*x^2+b*x+c=0
private Panel YaziPaneli;
Label prompt1;
Label prompt2;
Label prompta;
TextField inputa; //
Label promptb; //
TextField inputb; //
Label promptc; //
TextField inputc; //
double a,b,c; //polinomun katsayıları
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(delta>0)
{
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



Şekil 7.44 İkinci dereceden denklemin köklerini hesaplayan [PolinomunKokleri.java](#) programı ve sonuçlarının applette görülmesi

Şimdi aynı problemi JPanel kullanarak Japplet içinde gerçekleştirelim.

Program 7.48 : PolinomunKokleriSWA.java programı

```
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 katsayıları
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()
{
```

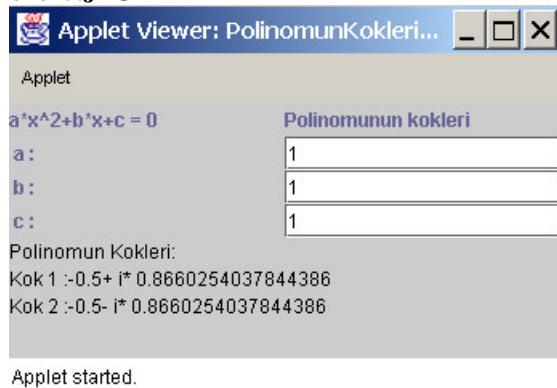
```

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



Şekil 7.45 İkinci dereceden denklemin köklerini hesaplayan [PolinomunKokleriSWA.java](#) programı ve sonuçlarının applette görülmesi

7.20 JOPTIONPANE SINIFI

Şu ana kadar gördüğümüz sınıflar grafik arayüzü dizaynı yapmaya yarayan sınıflardı. Java swing gurubunda aynı zamanda kullanılmaya hazır bir çok grafik arayüzleri tanımlanmıştır. Bunların en çok kullanılanı, ve bu kitapta da şu ana kadar yoğun olarak kullanageldiğimiz JOptionPane sınıfıdır. Bu sınıfın ana görevi pop-up mesaj ve girdi ortamı sağlamaktr. JOptionPane sınıfını programımızda kurucu metod kullanarak kurmamız gerekmekz. Temel olarak direk statik metodlarını çağırırız. JOptionPane panelinde genellikle bir resim,bir yazı alanı, bir girdi alanı (JTextField alanı), birde Seçilebilen düğmeler bulunur.

JoptionPane'in girdi/çıktı olarak kullanılabilen başlıca metodları şunlardır :

```
public static String showInputDialog(Object mesaj)
public static String showInputDialog(Component anapencere, Object mesaj)
public static String showInputDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü)
public static String showInputDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü, Icon resim, Object[] listeseçmedeğerleri, Object girişdeğeri)
public static String showInternalInputDialog(Component anapencere, Object mesaj)
public static String showInternalInputDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü)
public static String showInternalInputDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü, Icon resim, Object[] listeseçme değerleri, Object girişdeğeri)
public static void showMessageDialog(Component anapencere, Object mesaj);
public static void showMessageDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü);
public static void showMessageDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü, Icon resim);
public static void showInternalMessageDialog(Component anapencere, Object mesaj);
public static void showInternalMessageDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü);
public static void showInternalMessageDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü, Icon resim);
public static int showConfirmDialog(Component anapencere, Object mesaj);
public static int showConfirmDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü);
public static int showConfirmDialog(Component anapencere, Object mesaj, String başlık, int opsiyonürü, int mesajtürü);
public static int showConfirmDialog(Component anapencere, Object mesaj, String başlık, int opsiyonürü, int mesajtürü, Icon resim);
public static int showInternalConfirmDialog(Component anapencere, Object mesaj);
public static int showInternalConfirmDialog(Component anapencere, Object mesaj, String başlık, int mesajtürü);
public static int showInternalConfirmDialog(Component anapencere, Object mesaj, String başlık, int opsiyonürü, int mesajtürü);
public static int showInternalConfirmDialog(Component anapencere, Object mesaj, String başlık, int opsiyonürü, int mesajtürü, Icon resim);
public static int showOptionDialog(Component anapencere, Object mesaj, String başlık, int opsiyonürü, int mesajtürü, Icon resim, Object[] listeseçmedeğerleri, Object listedekiilkdeğer);
public static int showInternalOptionDialog(Component anapencere, Object mesaj, String başlık, int opsiyonürü, int mesajtürü, Icon resim, Object[] listeseçmedeğerleri, Object listedekiilkdeğer);
```

mesajtürü değişkeni (int) alabildiği değerler şunlardır :

JOptionPane.ERROR_MESSAGE	
JOptionPane.INFORMATION_MESSAGE	
JOptionPane.PLAIN_MESSAGE	
JOptionPane.QUESTION_MESSAGE	
JOptionPane.WARNING_MESSAGE	
Opsiyon türü değişkeni	
JOptionPane.DEFAULT_OPTION	: Sadece OK düğmesi verir
JOptionPane.OK_CANCEL_OPTION	: Ok ve Cancel(iptal et) düğmesi verir
JOptionPane.YES_NO_CANCEL_OPTION	: "Yes (evet) ve No(hayır) ve Cancel(İptal et) düğmeleri verir
JOptionPane.YES_NO_OPTION	: "Yes (evet) ve No(hayır) düğmeleri verir
OptionDialog kullanıldığından Metod bir integer değer gönderir	
JOptionPane.CANCEL_OPTION	
JOptionPane.CLOSED_OPTION	
JOptionPane.NO_OPTION	
JOptionPane.OK_OPTION	
JOptionPane.YES_OPTION	
JOptionPane.UNINITIALIZED_VALUE dur.	

Program 7.49 : JoptionPanel1.java Programı

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

```

public class JOptionPane1
{
    public static void main(String[] args)
    {
        Object[] o=new Object[]{"Turhan","Meral","İrfan","Hatice","Osman",
        "Nurhan","Birsen","Ali","Veli","Mustafa","Ahmet","Mehmet","Ayhan"};
        ImageIcon resim=new ImageIcon("images/Kuş.gif");
        JOptionPane.showInputDialog(null,"Lütfen bir isim seçin : ",
        "JOptionPane örnek 1",JOptionPane.QUESTION_MESSAGE,resim,o,"Turhan");
        System.exit(0);
    }
}

```

07046.JPG



Şekil 7.46 JOptionPane1.java programı sonucunun JOptionPane çıktısında görülmesi

7.21 SWING, JCOLORCHOOSEN SINIFI

JcolorChooser'da JOptionPane gibi kullanılmaya hazır bir dialog sınıfıdır. Renk seçmeye yarar ve renk seçilmesi gereken her yerde kullanılabilir. Bir önceki bölümdeki bir alıştırmada bu özelliği kullanmıştık. İkinci bir örnekle pekiştirelim :

Program 7.50 : ColorChooserDemo.java Programı

```

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 açalım
        final JLabel etiket = new JLabel("Java Programlama diline Hoş 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 sınıfını çağırıp rengi değiştir
    }
}

```

```

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(
    "Yazı 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



Şekil 7.47 ColorChooserDemo.java programı sonucunun JFrame çıktısında görülmesi

7.22 JFILECHOOSER SINIFI

JFileChooser'da JcolorChooser gibi hazır bir program demetidir. Bilgisayardan bir dosyayı seçme ileminde yararlanılır. Dosya prosesleri konusunda Java Girdi çıktı programlanması bölümune bakınız.

Program 7.51 : FileChooserDemo.java Programı

```

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 örneği");
        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 düğmesi 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 dosyanın açılma yeri.
                    log.append("Dosya açılıyor: " + file.getName() + "." + newline);
                } else {
                    log.append("Dosyanın açılması kullanıcı tarafından iptal edildi." + newline);
                }
            }
        });
        //kaydet düğmesi yarat
        ImageIcon saveIcon = new ImageIcon("images/save.gif");
        JButton kayıtDugmesi = new JButton("Dosya kaydet", saveIcon);
        kayıtDugmesi.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int returnVal = fc.showSaveDialog(FileChooserDemo.this);

                if (returnVal == JFileChooser.APPROVE_OPTION) {
                    File file = fc.getSelectedFile();
                    //gerçek dosyanın kaydedilme yeri..
                    log.append("Kaydediliyor : " + file.getName() + "." + newline);
                } else {
                    log.append("Dosyanın kaydedilmesi kullanıcı tarafından iptal edildi" + newline);
                }
            }
        });
        //Güzel bir görünüm için aç ve kapa düğmelerini ayrı bir panele yerleştir.
        JPanel buttonPanel = new JPanel();
        buttonPanel.add(acmaDugmesi);
        buttonPanel.add(kayıtDugmesi);

        //Düğmeleri pencereye ekle
        Container c = getContentPane();
        c.add(buttonPanel, BorderLayout.NORTH);
        c.add(logScrollPane, BorderLayout.CENTER);
    }
}

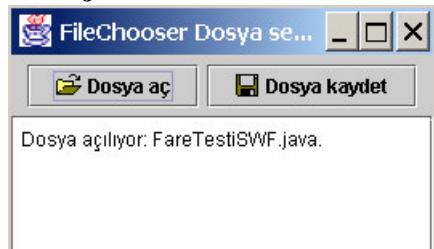
```

```

public static void main(String s[]) {
    JFrame frame = new FileChooserDemo();
    frame.addWindowListener(new BasicWindowMonitor());
    frame.pack();
    frame.setVisible(true);
}
}

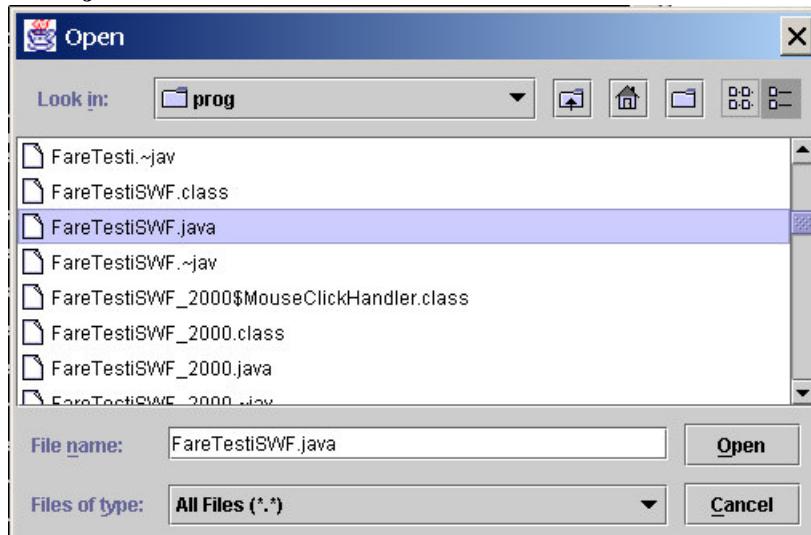
```

07048.JPG



Şekil 7.48 FileChooserDemo.java programı sonucunun JFrame çıktısında görülmesi

07049.JPG



Şekil 7.49 FileChooserDemo.java programı dosya açıcısının JFileChooser çıktısında görülmesi

7.23 JAVA JAR (JAVA ARCHIVES - JAVA ARŞİVİ) YAPILARININ KULLANIMI

JAR çeşitli sınıf ve ilgili dosyaları bir arada bir pakete dönüştürmek için geliştirilmiş bir teknolojidir. JAR teknolojisi programların taşınmasını ve yeniden yüklenmesini daha kolay bir hale getirir. Aynı zamanda JAR dosyasının içindeki elementler sıkıştırılmış halde bulunduklarından daha az yer kaplarlar. Bu yüzden bir JAR dosyasının internet üzerinden aktarılması, açık dosyaların tek tek aktarılmasından daha kısa süren bir prosedürdir.

Yeni JAR dosyalarını oluşturmak için :

eger **isim.jar** dosyasını oluşturmak ve bu dosyaya isim.class, program1.class ve resim.gif dosyalarını aktarmak istersek :

```
jar cf isim.jar isim.class program1.class resim.gif
```

Alıştırma 5deki H7AL1 örneği için JAR dosyası oluşturmak istersek :

```
jar cf H7AL1.jar H7AL1.class yıldız.class renk.class polar.class
```

komutunu kullanabiliriz. Problemi incelediğinizde JAR dosyasının içine ilave edilen tüm sınıfların bu probleme direkt olarak ilgili olduğunu görebilirsiniz.

jar dosyasının içindeki dosyalara bakmak için :

jar tf isim.jar

jar dosyası içindeki dosyaların listesini almak için :

jar xf isim.jar

jar dosyasını applet içinden çalıştırma için html içinde :

```
<applet code=AppletClassIsimi.class archive="JarDosyasiIsmi.jar"
width=width height=height
</applet
```

komutu kullanılır. Örnek olarak yine alıştırma 5 de kullanılan jar dosyasının html dosyasını verebiliriz.

```
<HTML
<HEAD
<TITLE yildiz cizim ornegi</TITLE
</HEAD
<BODY
<H3>yildiz cizim ornegi </H3
<applet code=H7AL1.class archive="H7AL1.jar" width=600 height=500
</applet
</BODY
</HTML
```

Jar komutuyla kullanılan kontrol karakterleri Tablo 7.1 de verilmiştir.

Tablo 7.1 Jar kontrol komut karakterleri ve anlamları

c	Yeni arşiv yarat
f	Listedeki ilk ismi arşiv dosyasının ismi olarak tanımla
m	Listedeki ikinci ismi arşiv dosyasının ismi olarak tanımla
t	arşiv içeriğini listele
x	dosyalar arşivden geri çağrılmacaktır. eğer listede sadece bir isim varsa bunu jar dosyası olarak alır ve içindeki tüm dosyaları listeler, eğer ilk simden sonra komutta başka isimler yer aliyorsa, sadece bu isimleri (eğer jarda mevcutsa) listeler.
0	sıkıştırma kullanma

7.24 ALIŞTIRMALAR

1. FareTesti.java programını ve bir önceki bölümde incelediğimiz cizgiciz programını birleştirerek fareyi kullanarak ekrana çizgi çizdiriniz (cizginin baslangic ve bitis noktaları fare tarafından belirlenecektir)
2. FareTesti.java programını ve gecen hafta incelediğimiz drawRect metotunu birleştirerek fareyi kallanarak ekrana dikdortgen çizdiriniz (dikdortgenin baslangic ve bitis noktaları fare tarafından belirlenecektir)
6. FareTesti.java programını ve gecen hafta incelediğimiz drawOval metotunu birleştirerek fareyi kallanarak ekrana daire çizdiriniz (dairenin merkezi ve capı fare tarafından belirlenecektir)

Resimli bir Jcheckbox örneği. Java döküman kütüphanesinden alınıp adapte edilen bu programda 16 resimden birisi JCheckBox kullanılarak yapılan seçime göre JLabel kullanılarak gösteriliyor.

Program 7.52 : CheckBoxDemo.java Programı

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

```

public class CheckBoxDemo extends JPanel {
    JCheckBox chinButton;
    JCheckBox glassesButton;
    JCheckBox hairButton;
    JCheckBox teethButton;

    /*
     * Dört değişik seçim imkanı tanımlanmıştır.
     * Bu seçimler 16 değişik dosyadaki resimlerden birini çağrımaktadır.
     * resim isimleri "geek-XXXX.gif" şeklinde olup
     * buradaki XXXX 16 seçimden birini gösterir.
     * seçim olasılıkları
     * ---- // hiçbirşey seçilmemiş
     *
     * 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 anlamları :
     * c : çene
     * g : gözlük
     * h : saç
     * t : diş
     */
}

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("Diş");
    teethButton.setMnemonic('D');
}

```

```

teethButton.setSelected(true);

// listener programlarını çağır
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 oluştur
pictureLabel = new JLabel(new ImageIcon(
    "images/geek/geek-"
    + choices.toString()
    + ".gif"));
pictureLabel.setToolTipText(choices.toString());

// CheckBoxları JPanel'e yerleştir.
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(

```

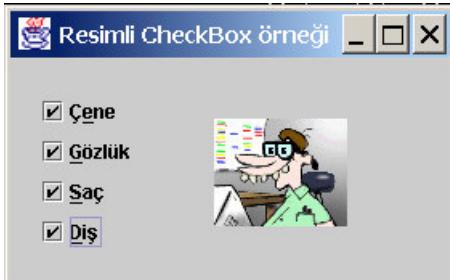
```

        "images/geek/geek-"
        + choices.toString()
        + ".gif"));
    pictureLabel.setToolTipText(choices.toString());
}
}

public static void main(String s[])
{
    JFrame frame = new JFrame("Resimli CheckBox örneği");
    frame.addWindowListener(new BasicWindowMonitor());
    frame.setContentPane(new CheckBoxDemo());
    frame.pack();
    frame.setVisible(true);
}
}

```

07050.JPG



Şekil 7.50 CheckBoxDemo.java çıktısının JFrame'de görülmesi

7. Şirin bir JComboBox örneği : Listeden yapılan seçime göre resim değişecektir. Bu programda java döküman kütüphanesinden alınıp türkçeye adapte edilmiştir.

Program 7.53 : ComboBoxDemo.java Programı

```

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

public class ComboBoxDemo extends JPanel {
    JLabel picture;

    public ComboBoxDemo() {
        String[] petStrings = { "Kuş", "Kedi", "Köpek", "Tavşan", "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 değer daha sonra program tarafından değiştirilecektir.  
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 çıktısının JFrame'de görülmESİ

5. Appletin alt bölgesinde(SOUTH) iki sıra, her sırada beşer düğmeden on düğme bulunan bir applet oluşturunuz.
Appletin tepesinde bir TextField oluşturunuz. Her düğmeye basıldığında textFieldde değişik bir yazı görünsin. Yazılıarı 2x5 lik bir String değişkeninden okuyarak Text fieldde yükleyiniz.

Program 7.16 Panel10.java program

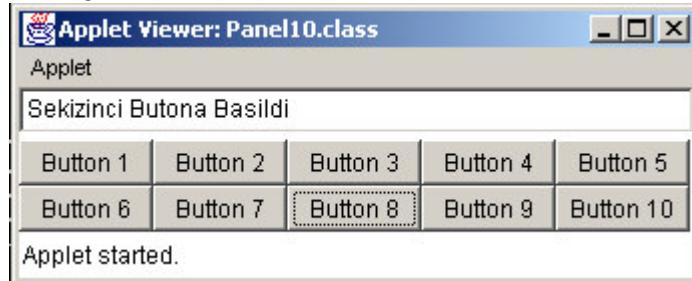
```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Panel10 extends Applet implements ActionListener
{
    private Panel buttonPanel;
    private Button buttons[];
    String s[]=
    {"Birinci Butona Basildi",
     "Ikinci Butona Basildi",
     "Ucuncu Butona Basildi",
     "Dorduncu Butona Basildi",
     "Besinci Butona Basildi",
     "Altinci Butona Basildi",
     "Yedinci Butona Basildi",
     "Sekizinci Butona Basildi".
```

```

"Dokuzuncu 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



Şekil 7.52 Panel10.html çıktısı

4. H7AL1.java programını incele ve çalıştır.

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("acyi 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(arcaPlanRenkListesi);
//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);
arcaPlanRenkListesi.addActionListener(this);
renkListesi.addItemListener(this);
arcaPlanRenkListesi.addItemListener(this);
for(int i=0;i<r.length;i++)
{
renkListesi.add(Renkler[i]);
arcaPlanRenkListesi.add(Renkler[i]);
}
add(p,BorderLayout.NORTH);
add(p1,BorderLayout.NORTH);
}

```

```

public void actionPerformed(ActionEvent e)
{
int indeks=arcaPlanRenkListesi.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: "+
arcaPlanRenkListesi.getSelectedItem());
}

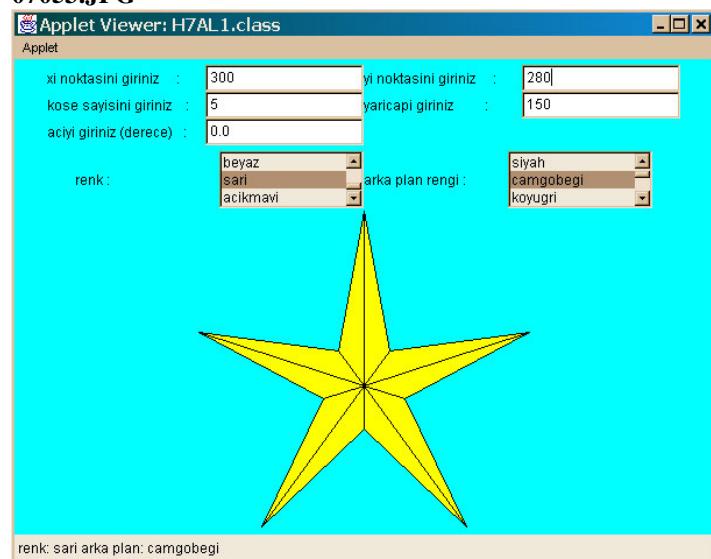
```

```

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 çıktısı

5. H7AL1_2000.java programını incele ve çalıştır.

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",
    "yeşil","mavimsi","sarı","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 değişkenlerin ilk değerleri verilmelidir.
c=getContentPane();
setBackground(renk.beyaz);
p=new JPanel();
p.setLayout(new GridLayout(3,4));
p.setBackground(renk.beyaz);
//3*4 tablo p paneli hazırlandı
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 noktasını giriniz :");
p.add(kutu1);
kutugir1=new JTextField(3);
p.add(kutugir1);
kutu2=new JLabel("yi noktasını giriniz :");
p.add(kutu2);
kutugir2=new JTextField(3);
p.add(kutugir2);
kutu3=new JLabel("kose sayısını 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 çıktısı

6. H7AL2.java programını incele ve çalıştır. Bu program fare kullanarak yıldızı çizdirmektedir. Fare'yi sol klik ettiğimiz noktayı merkez olarak almakta, fareyi sürüduğumuz nokta ve açıyı çizdirme açısı olarak almakta fare düğmesini bıraktığımız noktada yıldız çapını almakta ve yıldızı çizmektedir. Renkler girdi listelerinden değiştirilebilir.

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 tı̄m dēeri Ÿkenlerin ilk dēerleri verilmelidir.
setBackground(renk.beyaz);
p=new Panel();
p.setLayout(new GridLayout(3,4));
p.setBackground(renk.beyaz);
//3*4 tablo p paneli hazırlanır
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 noktasını giriniz   :");
p.add(kutu1);
kutugir1=new TextField(3);
p.add(kutugir1);
kutu2=new Label("yi noktasını giriniz   :");
p.add(kutu2);
kutugir2=new TextField(3);
p.add(kutugir2);
kutu3=new Label("kose sayısını giriniz   :");
p.add(kutu3);
kutugir3=new TextField(3);
p.add(kutugir3);

kutu4=new Label("yaricapı 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)
{
}

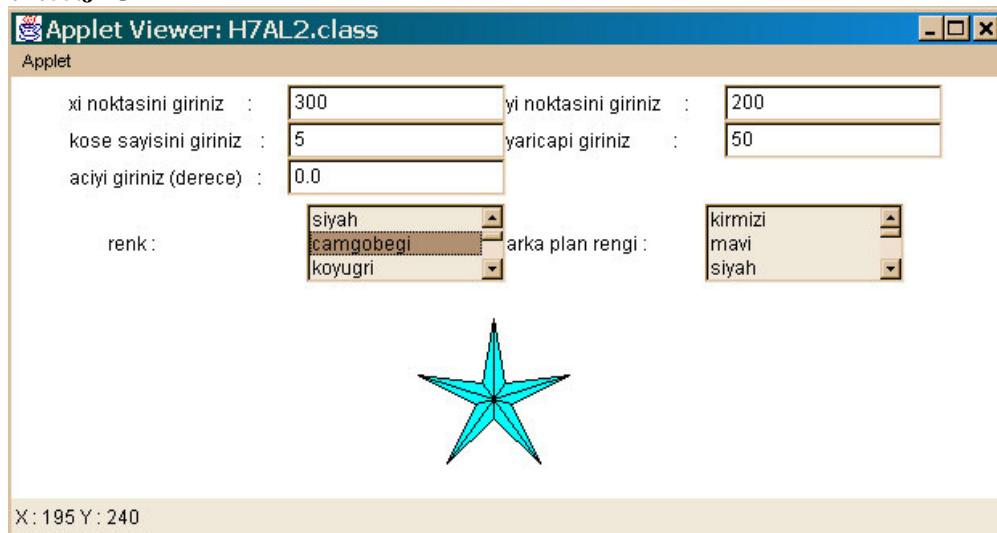
```

```

        showStatus(" X : "+ e.getX()+" Y : "+e.getY());
    }
}

```

07055.JPG



Şekil 7.55 H7AL2.java çıktısı

7. **H7AL3.java** programını incele bu program yıldız çizdirmektedir. Yıldızın rengi scrollbar kullanılarak değiştirilmektedir. Appleti çalıştır.

Program 7.56 H7AL3.java programı.

```

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,x1,y1,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 t̄m de§iÝkenlerin ilk de§erleri verilmelidir.
Panel p=new Panel();
p.setLayout(new GridLayout(5,4));
//5*4 tablo p paneli haz̄rlandı
kutu1=new Label("xi noktasını giriniz :");
p.add(kutu1);
kutugir1=new TextField(3);
p.add(kutugir1);

kutu2=new Label("yi noktasını giriniz :");
p.add(kutu2);
kutugir2=new TextField(3);
p.add(kutugir2);

kutu3=new Label("kose sayısını giriniz :");
p.add(kutu3);
kutugir3=new TextField(3);
p.add(kutugir3);

kutu4=new Label("yaricapı 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.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 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();
}

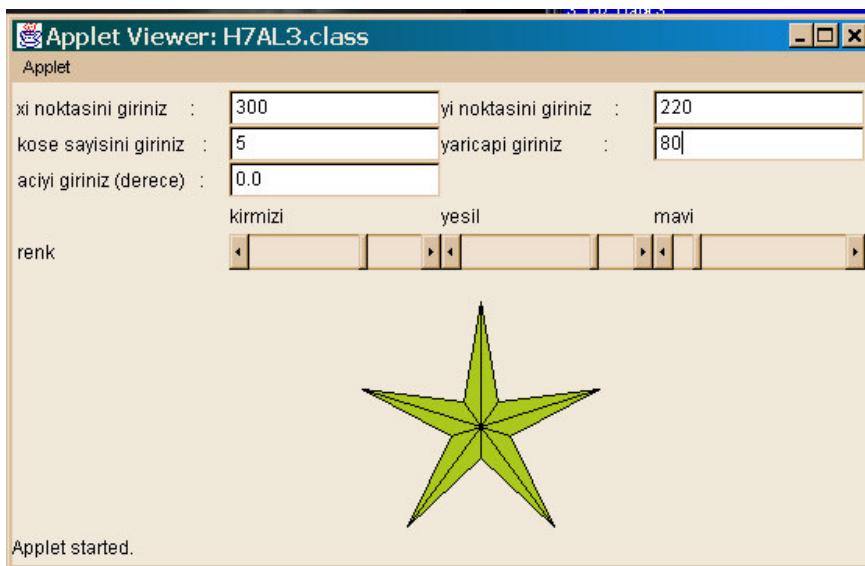
```

```

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



Şekil 7.56 Yıldız çizdiren ve scrollbar kullanarak renk kontrolu yapan H7AL3.html çıktısı

8. **H7AL4.java** programı cardLayout layout düzenleyicisini kullanmaktadır. Bu düzenleyici appleti çeşitli sayfalar şeklinde düzenlemektedir. Programı ve cardLayout layout düzenleme sistemini inceleyiniz.

Program 7.57 : H7AL4.java, cardLayout sistemini kullanan yıldız 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 manager Layout kullanılmıştır
    private CardLayout c;
    private Panel p,p1,p2,p3,p4;
    int n1,xi1,yi1,yaricap1;
    double aci1;
    renk renk; //sınıfin adı renk, nesne adı 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 t̄m de§iÝkenlerin 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 hazırlanır
kutu1=new Label("xi noktasını giriniz :");
p1.add(kutu1);
kutugir1=new TextField(3);
p1.add(kutugir1);
kutu2=new Label("yi noktasını giriniz :");
p1.add(kutu2);
kutugir2=new TextField(3);
p1.add(kutugir2);
kutu3=new Label("kose sayısını giriniz :");
p2.add(kutu3);
kutugir3=new TextField(3);
p2.add(kutugir3);
kutu4=new Label("yaricapı 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.addActionListener(this);
kutumavi.addActionListener(this);
kutuyesil.addActionListener(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();

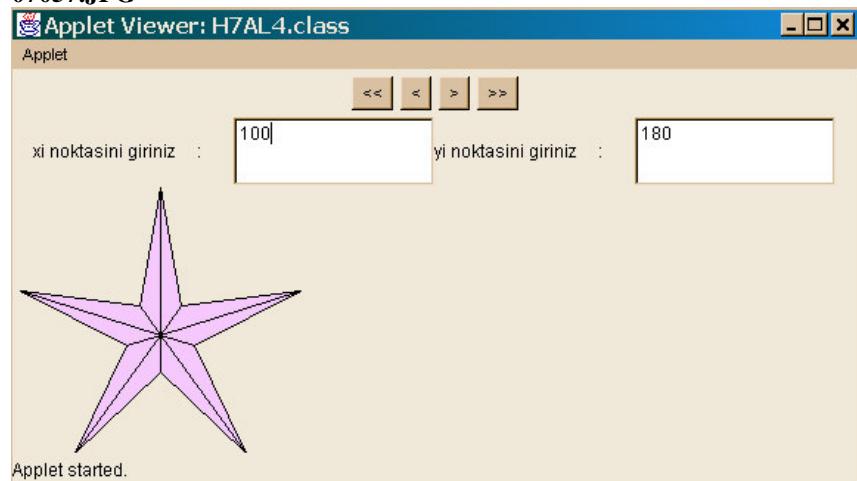
}

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



Şekil 7.57 H7AL4.html, cardLayout sistemini kullanan yıldız çizme appleti

9. Appletde fare kullanarak çizgi çizdiren H7AL5.java programını 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 t̄m de§i§enlerin 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 hazırlanır
        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)
{
```

```

}

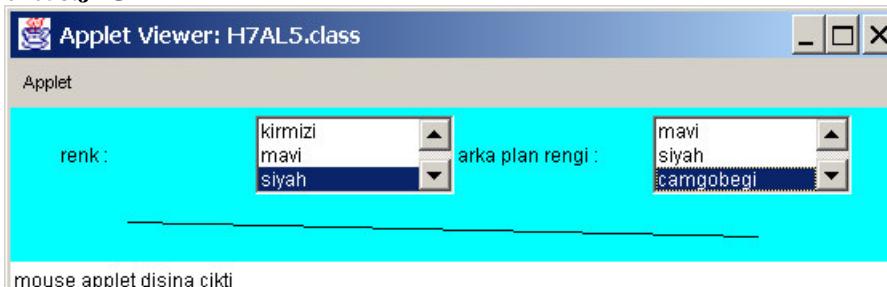
public void mouseExited(MouseEvent e)
{
    showStatus("mouse applet disina ciktig");
}

public void mouseMoved(MouseEvent e)
{
}

public void mouseDragged(MouseEvent e)
{
}

```

07058.JPG



Şekil 7.58 Appletde fare kullanarak şekil çizdiren H7AL5.html applet görüntüsü

- H7O1.java**, Alıştırma H7AL2'de yıldız çizilmiş ve mouse'u basıp çekip bırakarak yeni yıldız oluşturulmuştur. Aynı işlemi yapan, fakat eşkenar çokken çizen programı oluşturunuz.

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 t̄m de§i§kenlerin 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 hazırlandı
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 noktasını giriniz :");
p.add(kutu1);
kutugir1=new TextField(3);
p.add(kutugir1);
kutu2=new Label("yi noktasını giriniz :");
p.add(kutu2);
kutugir2=new TextField(3);
p.add(kutugir2);
kutu3=new Label("kose sayısını giriniz :");
p.add(kutu3);
kutugir3=new TextField(3);
p.add(kutugir3);

kutu4=new Label("yaricapı 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)
{
}

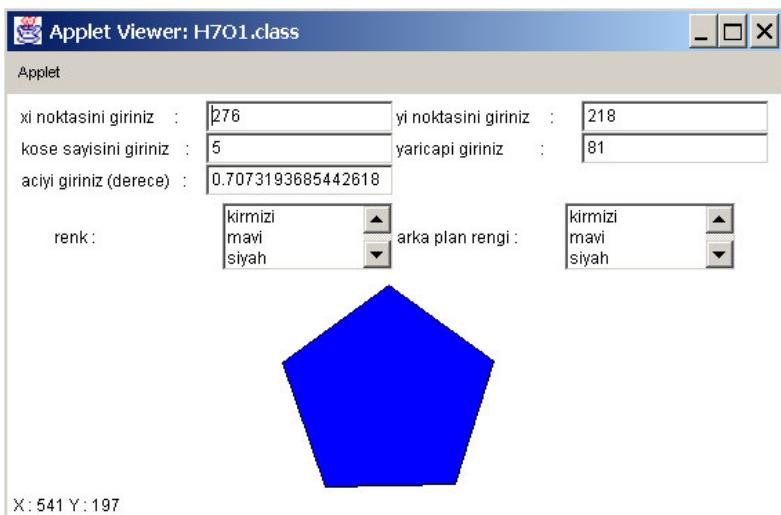
public void mouseExited(MouseEvent e)
{
}

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



Şekil 7.59 Appletde şekil çokgen çizdiren H7O1.html applet çıktı

11. H7O2.java, Choice kullanarak eskenarcokken veya yıldız çizebilen applet programını oluşturunuz.

Program 7.60 H7O2.java, choice kullanımı

```

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 t̄m de§iÝkenlerin 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 hazırlandı
kutu1=new Label("xi noktasını giriniz :");
p.add(kutu1);
kutugir1=new TextField(3);
p.add(kutugir1);

kutu2=new Label("yi noktasını giriniz :");
p.add(kutu2);
kutugir2=new TextField(3);
p.add(kutugir2);

kutu3=new Label("kose sayısını giriniz :");
p.add(kutu3);
kutugir3=new TextField(3);
p.add(kutugir3);

kutu4=new Label("yaricapı 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.addActionListener(this);
kutumavi.addActionListener(this);
kutuyesil.addActionListener(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§iÝirildiginde
    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§iÝirildi§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);
}
}

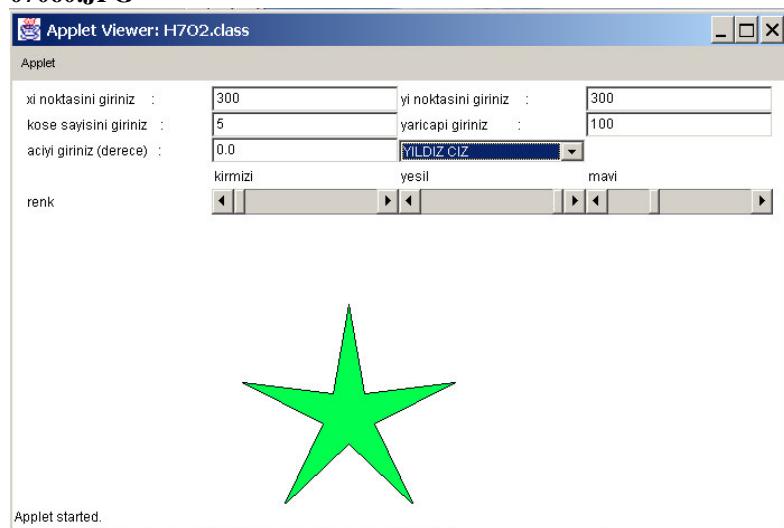
```

```

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



Şekil 7.60 Appletde choice kullanarak çokgen veya yıldız çizdiren H7O2.html applet çıktısı

12. H7O3.java, Checkbox kullanarak eskenarcokken veya yıldız veya herikisinide bir arada çizebilen (yıldızın köşelerinden geçen bir eskenar cokken) applet programını oluşturunuz.

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 metodunda tı̄m de§iÜkenlerin 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 hazırlandı
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 noktasını giriniz   :");
p.add(kutu1);
kutugir1=new TextField(3);
p.add(kutugir1);
kutu2=new Label("yi noktasını giriniz   :");
p.add(kutu2);
kutugir2=new TextField(3);
p.add(kutugir2);
kutu3=new Label("kose sayısını 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]);
}
}
```

```

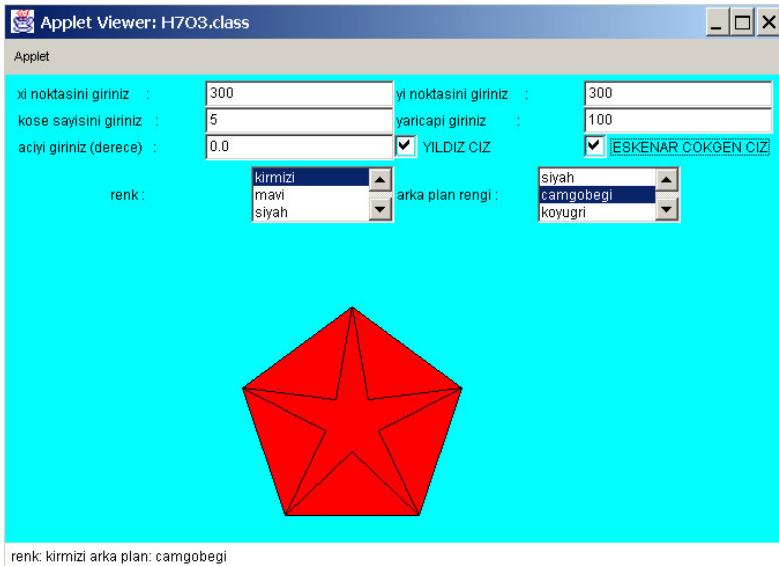
Integer xi1i=new Integer(kutugir1.getText());
xi1=xi1i.intValue();
Integer yi1i=new Integer(kutugir2.getText());
yi1=yi1i.intValue();
Integer n1i=new Integer(kutugir3.getText());
n1=n1i.intValue();
Integer yaricapi=new Integer(kutugir4.getText());
yaricap1=yaricapi.intValue();
Double aci1i=new Double(kutugir5.getText());
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 yıldız çizdiren H7O3.html applet çıktısı

13. H7O4.java, H7O3.java(alıştırma 12) programının aynısını oluşturunuz, ancak renk seçimini Scrollbar ile yapınız.

14. H7OD3_2000

JSlider kullanarak çizdiğiniz bir yıldızın çapını büyütüp küçültlen **H7OD3_2000** programını yazınız.
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

```

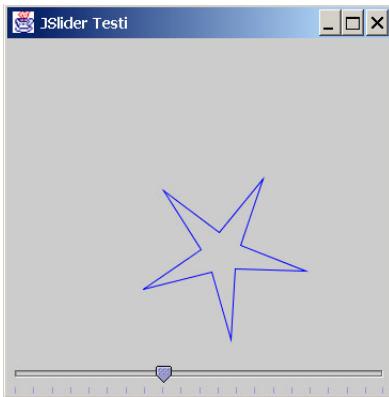
import javax.swing.*;      // java swing sinifini cagir
import java.awt.*;        // java pencere kullanma sinifini cagir
import java.awt.event.*;   // java pencereyi dinleme sinifini cagir
import BasicWindowMonitor;
import ovalPanelSWP;
import javax.swing.event.*;

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



Şekil 7.62 H7OD3_2000.java program çıktısının JFrame penceresinde görünümü

15. H7OD4_2000

Düğme (Button) kontrollü ile yıldız vetya oval çizdirecek bir program yazınız. Yıldız düğmesine basılıncı ekrana yıldız gelsin, oval düğmesine basılıncı 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("düğme - yıldız - oval seçim testi");
c= getContentPane();
JPanel p1=new JPanel();
p1.setLayout(new GridLayout(1,2));
B1=new JButton("yıldız");
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



Şekil 7.63 Şekil -7.64 H7OD4_2000.java program çıktısının JFrame penceresinde görünümü

17 swing ve awt karşılaştırmalı örnekler 1

Su ana kadar çeşitli örneklerde hem swing hem de awt olarak verdigimiz örneklerle iki grafik sistemindeli farklılık ve benzerlikleri belirtmeye çalıştık. Şimdi özellikle bu karşılaştırmayı çeşitli uygulama problemleri olarak göreceğiz. Bu örnekteki problemler temel olarak java döküman kütüphanesinden alınmıştır. İlk örneğimiz TextEventDemo awt de yazılmış bir applet programı. TextArea penceresine giren her harfte TextEvent yardımıyla kontrollar yapıyoruz.

Program 7.65 : TextEventDemo (awt)

```

/*
 * awt applet programı
 */

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", displayArea);

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
            + " yazı değişti."
            + 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 yazıyı döndürür */
}

```

```

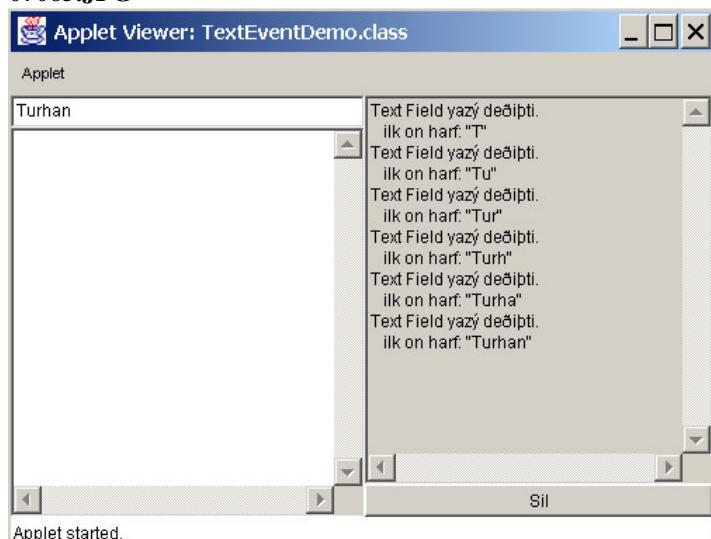
public void actionPerformed(ActionEvent e) {
    int selStart = textArea.getSelectionStart();
    int selEnd = textArea.getSelectionEnd();

    textArea.replaceRange(textField.getText(),
                          selStart, selEnd);
    textField.selectAll();
}
}

/** düğmeye basılınca... */
public void actionPerformed(ActionEvent e) {
    displayArea.setText("");
    textField.requestFocus();
}
}

```

07065.JPG



Şekil 7.65 TextEventDemo.java sınıfının çıktı programının awt applet'te görülmesi

18 swing ve awt karşılaştırmalı örnekler 2 : TextEventDemoSW.java

Program 7.66 TextEventDemo.java (swing) test programı

```

/*
 * 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
            + " yazı değiştirildi."
    }
}

```

```

+ 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



Şekil 7.66 TextEventDemo.java sınıfının çıktı programının swing JApplet'te görülmesi

18 swing ve awt karşılaştırmalı örnekler 3 : Birim dönüştürme örneği : **TextEventDemoSW.java**

Birim dönüştürme örneği :

Tablo AWT ve Swing birim dönüştürme programları

AWT Converter (birim dönüştürücü)	Swing Converter (birim dönüştürücü)
Converter.java ConversionPanel.java Unit.java	Converter.java ConversionPanel.java Unit.java ConverterRangeModel.java FollowerRangeModel.java DecimalField.java FormattedDocument.java

Program 7.67 Converter.java AWT programı

```
/*
 * 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 program

```

/*
 * 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.gridx = 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.gridxwidth = 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.gridxwidth = 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.gridxwidth = 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 program

```

/*
 * 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 program

```

/*
 * 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 program

```

/*
 * 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 program

```

/*
 * 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 program

```

/*
 * 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 program

```

/*
 * 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 program

```

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 program

```

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 programı

```

/*
 * 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());
}

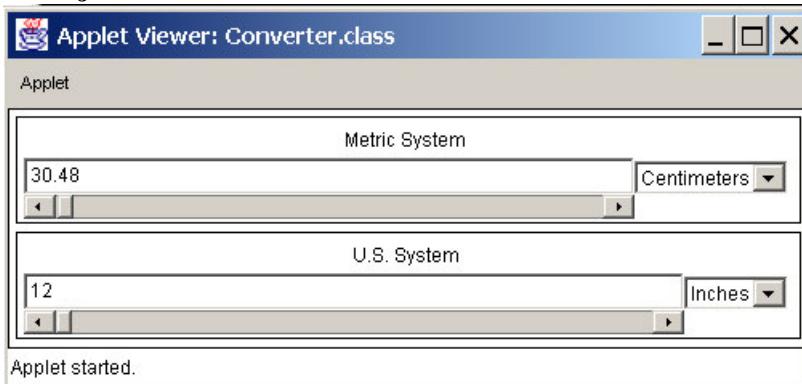
public int getExtent() {
    return super.getExtent();
}

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);
}
}

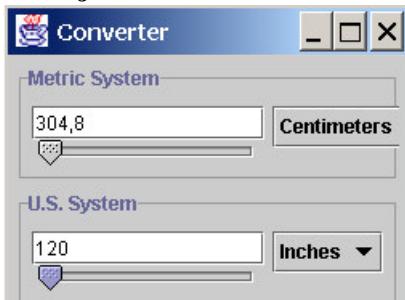
```

07067.JPG



Şekil 7.67 AWT Converter (birim dönüştürücü)

07068.JPG



Şekil 7.68 Swing Converter (birim dönüştürücü)

19 awt –swing karşılaştırmalı örnekler 4 : awt List örneği : ListDemo.java

bu programın swing eşdeğeri ana metinde verilmiştir. Şimdi de swing versiyonuna göz atalım

Program 7.78 ListDemo.java AWT programı

```
/*
 * 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("altı");
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.
GridLayout gridBag = new GridLayout();
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 oluştu "
        + list.getSelectedItem() + "\n"
        + language + " dilinde." + newline);
}

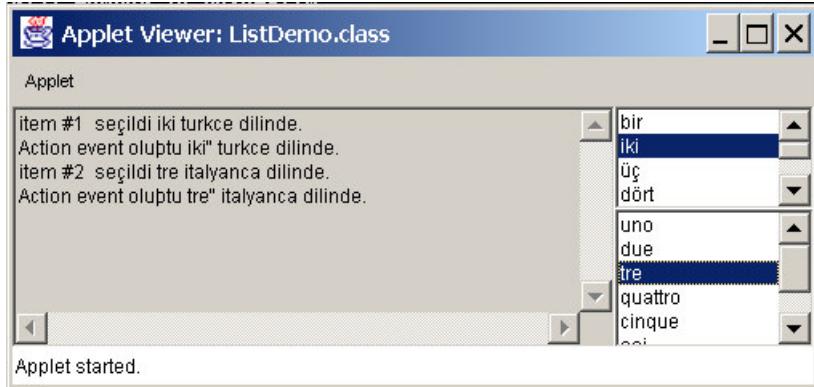
public void itemStateChanged(ItemEvent e) {
    List list = (List)(e.getItemSelectable());
    String language = (list == turkce) ?
        "turkce" : "italyanca";

    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



Şekil 7.69 ListDemo.java (awt)

20. bu program ListDialogu tanımlamakta ve kullanmaktadır.

Program 7.79 ListDialog.java AWT program

```
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 hatası");
        }
        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("İptal 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 kısmı
    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", "Yağmur", "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, "İsim seçme dialoğu",
        "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(Component.CENTER_ALIGNMENT);
    name.setAlignmentX(Component.CENTER_ALIGNMENT);
    button.setAlignmentX(Component.CENTER_ALIGNMENT);
    f.pack();
    f.setVisible(true);
}
}

```

07070.JPG



Şekil 7.70 ListDialog.java

20. bu program işlem takip çubuk grafiği olan progressMonitor tanımlamakta ve kullanmaktadır.

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;

```

```

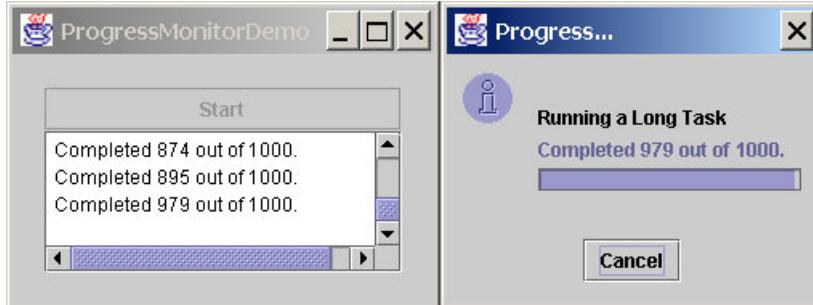
private ProgressMonitor progressMonitor;
private Timer timer;
private JButton startButton;
private LongTask task;
private JTextArea taskOutput;
private String newline;
public ProgressMonitorDemo() {
    super("ProgressMonitorDemo");
    newline = System.getProperty("line.separator");
    task = new LongTask();
    //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);
    }
}

```



21. bu program detaylı bir dosya seçme sistemi tanımlamaktadır.

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);

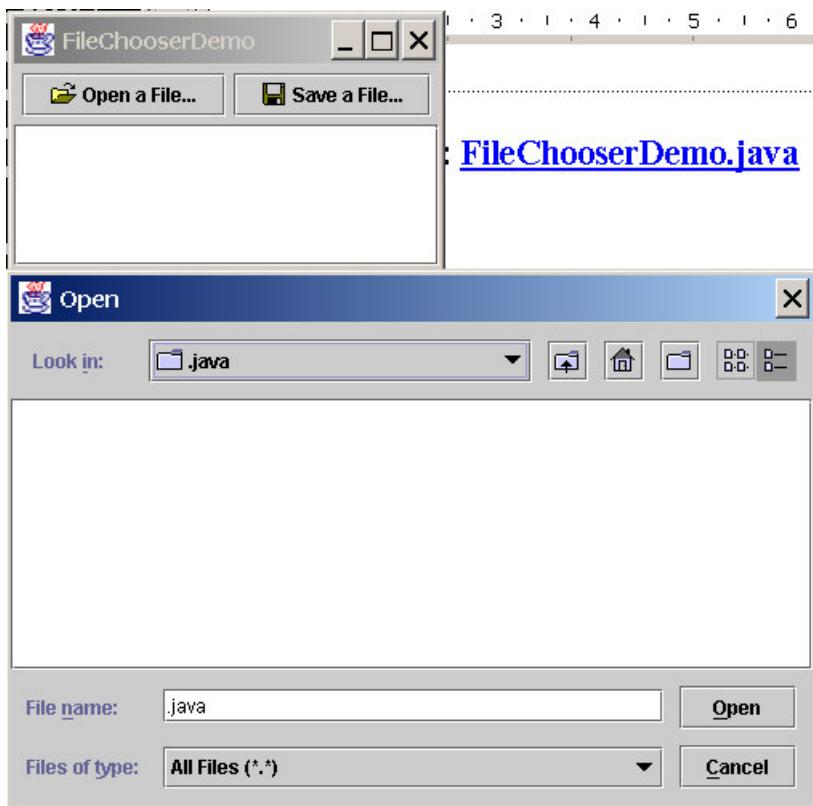
//Add the buttons and the log to the frame
Container contentPane = getContentPane();
contentPane.add(buttonPanel, BorderLayout.NORTH);
contentPane.add(logScrollPane, BorderLayout.CENTER);
}

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);
}
}

```



22. bu program detaylı bir dosya seçme sistemi tanımlamaktadır.

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 ImageFileDialog());
                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);
        }
    }
});

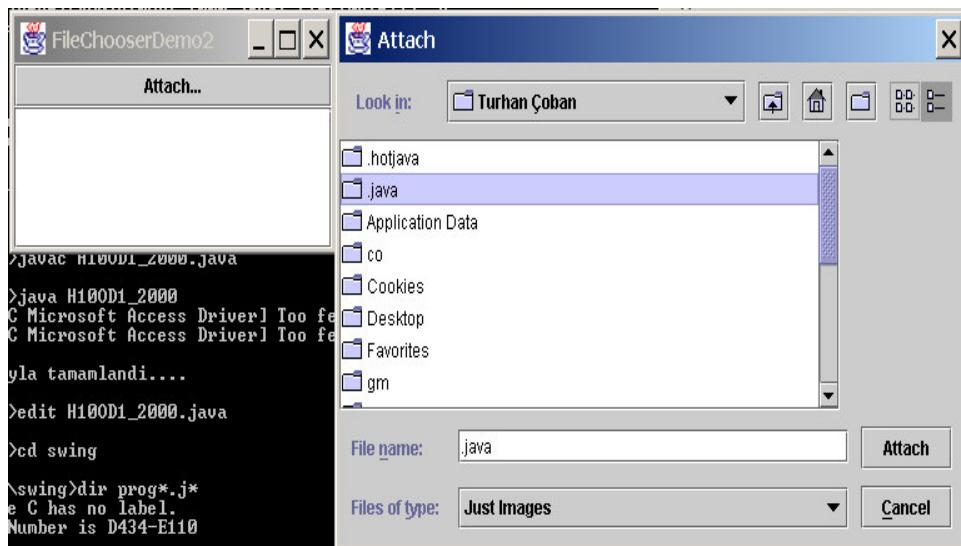
Container contentPane = getContentPane();
contentPane.add(sendButton, BorderLayout.NORTH);
contentPane.add(logScrollPane, BorderLayout.CENTER);
}

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);
}
}

```



23. bu program detaylı bir dosya seçme sistemi tanımlamaktadır.

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);

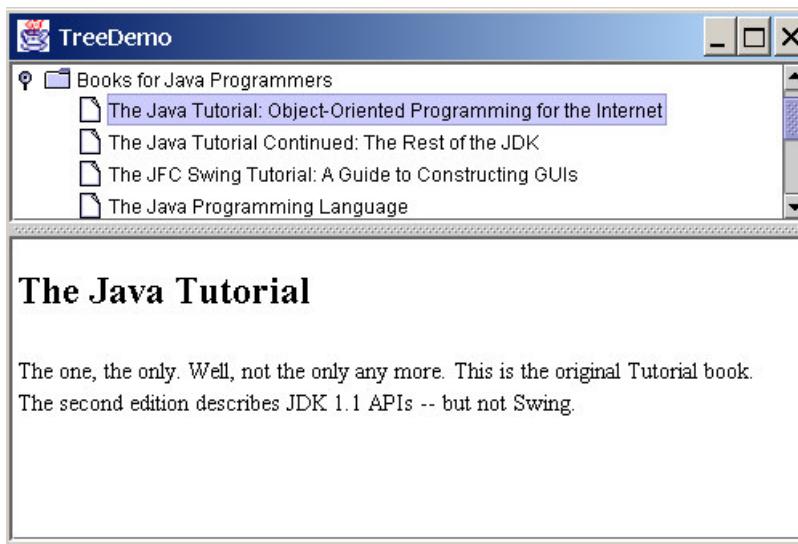
//Language Spec
book = new DefaultMutableTreeNode(new BookInfo
        ("The Java Language Specification",
         "jls.html"));
category.add(book);
}

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);
}
}

```



- 24.** Şimdi bir grafik çizim programına bakalım. Swing ve Graph2D olarak yazılmış olan bu program bir dosyadan (Plot.txt) okuduğu veriye göre grafik çizmektedir.
Örneğin Plot.txt dosyasında

```

Başlık
x ekseni
y ekseni
2
in.txt 20 0 0 0
out.txt 3 0 0 255

```

değerleri varsa Plot başlığı olarak “Başlık”, x ekseni yazısı olarak “x ekseni”, y ekseni yazısı başlığı olarak “y ekseni”, toplam veri dosyası olarak 2, veri dosyası isimleri “in.txt” ve “out.txt”, çizim şekilleri 20 ve 3 ve renkler 00 255 ve 0 0 0 kullanmaktadır.

Once Plot şekillerini tanımlayan PlotShapesSW.java programına göz atalım

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 x maxi,
                        double ymini,double y maxi)
    {
        // xabsmin : absolute 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=fon ts[2].deriveFont(Font.BOLD,12);
        //g.setFont(f);

        xabsmin=xabsmini;
        yabsmin=yabsmini;
        absheight=absheighti;
        abswidth=abswidthi;
        xmin=xmini;
        xmax=x maxi;
        ymin=ymini;
        ymax=y maxi;
    }

    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 gridlines |||||
    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)/100.0);
        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,x[i][j],y[i][j],(xmin-a)/b,(a+b*xmin));
                }
            }
        }
    }
}

```

```

        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 tanımlanan şekil, çizgi vs gerçek plot koordinat sistemini kullanmakta ve pencere koordinat sistemiyle gerçek koordinat sistemleri arasında otomatik dönüşüm yapmaktadır.

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("IOException in opening plot");}
    }

    public void yenidanPlotDatasıOku()
    {
        try{
            p1=new Plot();
        } catch(IOException ioe) {System.err.println("IOException in 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.xtic);
    if(p1.ygridon!=0)
        ps.drawYGrid(p1.ytic);
    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.xtic,(p1.ymax-p1.ymin)/80.0,0);
    ps.drawYTic(p1.ytic,(p1.xmax-p1.xmin)/80.0,0);
    ps.drawXNumbers(p1.xtic);
    ps.drawYNumbers(p1.xtic);
    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 şekillerini Plot oluşturacak şekilde bir araya koymaktadır.

Program 7.85 : [PlotWindowSWF_2000.java](#)

```

//=====
// Numerical Analysis package in java
// Plot (Çizim ) programı
// 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 sayfası";
    final static String PLOTPANEL = "Plot sayfası ";
    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 başlığı : ");
        promptXLabel=new JLabel("      x ekseni başlığı : ");

```

```
promptYLabel=new JLabel("      y ekseni başlığı : ");
inputEmail=new JTextField(30);
inputXLabel=new JTextField(30);
inputYLabel=new JTextField(30);
//*****
promptXntic=new JLabel("X tik no");
inputEmail=new JTextField(5);
promptYntic=new JLabel("Y tik no");
inputEmail=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 getPreferredSize()
{
    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);
//*****
inputEmail.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;
}
}
}

```

```

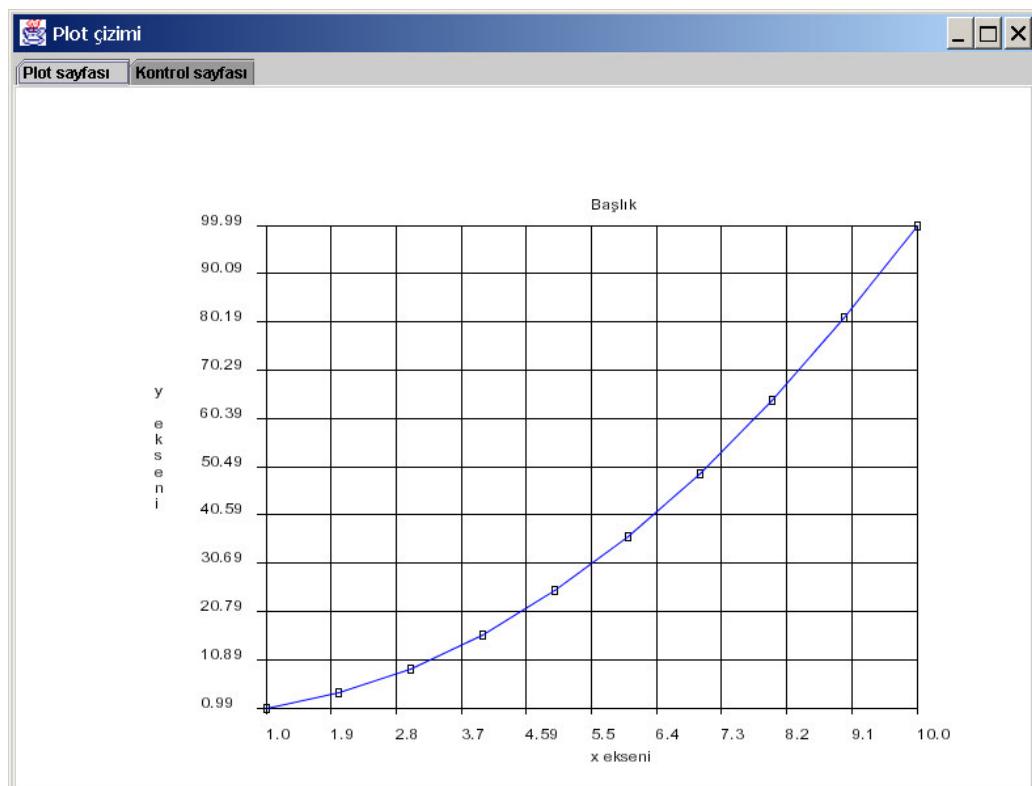
inputLabel.setText(jta.p1.label);
inputXLabel.setText(jta.p1.xlabel);
inputYLabel.setText(jta.p1.ylabel);
jta.yenidenciz();
}

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.p1ymax=valYmax.doubleValue();
jta.p1label=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);
}
}

```

Asıl Plot'ı çizer bu programdadır.



Plot çizimi

Plot sayfası **Kontrol sayfası**

Xmin	1.0	Xmax	10.0	Ymin	9999432	Ymax	9999432
Y tipli no	10	X grid (küçük kare)	<input checked="" type="checkbox"/>	Y grid (küçük kare)	<input checked="" type="checkbox"/>		
Plot başlığı :	Başlık						
x eksen'i başlığı :	x eksen'i						
y eksen'i başlığı :	y eksen'i						
Yazdır							

Plot başlığı :

x eksen'i başlığı :

y eksen'i başlığı :

Başlık

x eksen'i

y eksen'i

Yazdır

1.0

10.0

9999432

9999432

10

X grid (küçük kare)

Y grid (küçük kare)

8. JAVADA HATA ANALİZİ VE YAKALANMASI

8.1 HATA ANALİZİ

Javada diğer dillerden farklı olarak dil yapısının içinde hataları yakalamak için oldukça gelişmiş bir sistem mevcuttur. Eğer javada kontrol edilmemiş bir hata oluşursa program durur ve hatayı yazar. Biz hata oluştuğunda hata kontrol sisteminde hatayı kontrol edip programı çalışmaya devam edebiliriz. Javadaki hata kontrol blogu şu şekilde yazılır :

```
try { java deyimleri;
java kaynaklarından herhangi birini kullanan deyimler
}
catch( herhangibirtürException e1)
{ exception kontrol deyimleri }
catch( diğerherhangibirtürException e1)
{ exception kontrol deyimleri }
finally
{
deyimler
java kaynağını serbest bırakma deyimleri
}
```

Javadaki Exception türleri aşağıdaki tabloda verilmiştir :

Tablo 8.1 : Java hata ve exception listesi

Java.lang hata listesi

```
AbstractMethodError
ClassCircularityError
ClassFormatError
Error
ExceptionInInitializerError
IllegalAccessError
IncompatibleClassChangeError
InstantiationException
InternalError
LinkageError
NoClassDefFoundError
NoSuchFieldError
NoSuchMethodError
OutOfMemoryError
StackOverflowError
ThreadDeath
UnknownError
UnsatisfiedLinkError
VerifyError
VirtualMachineError
```

Java.lang Exception listesi

```
ArithmaticException
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 oluştuğunda hatanın olduğu metotun girişinde kullanılan **throws** kelimesini takip eden Exception sınıfıyla aynı exception gurubunu taşıyan **try-catch** gurubuna aktarır.

Bu metotun içinde gerekli noktada (hatanın olacağı şartda) **throw** deyimi kullanılarak hata olduğu aktarılır ve bu aktarım **try-catch** gurubu tarafından yakalanır ve işleme alınır.

Hata kontrolunu daha iyi açıklayabilmek amacıyla bir örnek vermek istiyoruz. SifiraBölünmeException sınıfında yeni bir hata kontrol sınıfı yaratıyoruz. Bu sınıf ve java kütüphanesinde mevcut olan tamsayı format hatası (NumberFormatException) sifirebolmetesti programında test edilmiştir. Sonuçlar applet çıktılarında görülmektedir. Eğer bu hatalar kontrol edilmeseydi, java programı hata vererek çalışmasını durdurdu.

Programdaki hata kontrol deyimleri kalın harfle verilmiştir. Hata kontrol bloğu ise italic olarak verilmiştir.

Problem 8.1 sifiraBolunmeException.java programı, sısra bolme hatası tanımlar

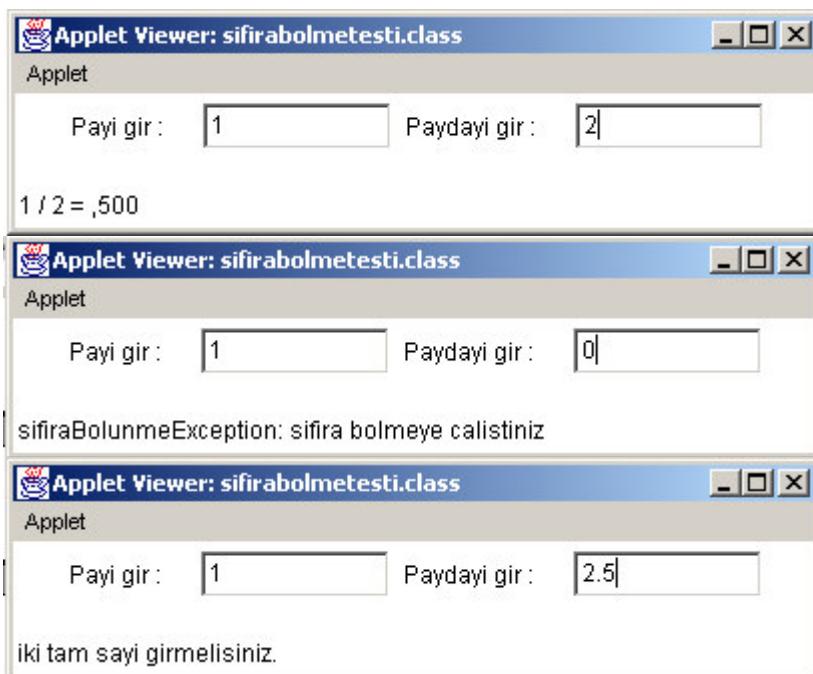
```
public class sifiraBolunmeException  
extends ArithmeticException  
{  
public sifiraBolunmeException()  
{
```

```
super("sifira bolmeye calistiniz ");
}
}
```

Problem 8.2 Bizim tarafımızdan oluşturulan **sifiraBolunmeException.java** hata kontrolü ve sayı formatı hata kontrolü (**NumberFormatException**) yapan [sifirabolmetesti.java](#) programı

```
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



Şekil 8.1-3 sifirabolmetesti.html appletinde sıfıra bolme ve tamsayı hata mesajlarının görünümü

Programlarda da görüldüğü gibi hata programın çalışmasına engel olmaksızın kullanıcıya bildirilmektedir. Bu program gurubunda önce ArithmeticException sınıfından sifiraBolunmeException sınıfı türetilmiştir. Sonra programın içindeki

```
public double bol(int s1,int s2) throws sifiraBolunmeException
{
if(s2==0) throw new sifiraBolunmeException();
return (double) s1/s2;
}
```

metotunda metotun önce throws sifiraBolunmeException deyi̇miyle exception gönderceği bildirilmiş, ve metotun içinde

```
if(s2==0) throw new sifiraBolunmeException();
deyi̇miyle exception ( hata değişkeni) gönderilmiştir. Gönderilen hata değişkeni (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 sayı girmelisiniz.");
catch(sifiraBolunmeException sbh)
{showStatus(sbh.toString());
}
}
```

gurubu tarafından yakalanmış, ve eğer sıfır hatası varsa

catch(sifiraBolunmeException sbh) deyi̇mi tarafından yakalanmış ve showStatus("iki tam sayı girmelisiniz."); işlemini çağrı̇mi̇ bu işlem sonunda applet sifiraBolunmeException sınıfının toString metotunu kullanarak

sifiraBolunmeException:sifira bolmeye calistiniz

mesajını vermiştir. Ayrıca yukarıdaki program parçasında da görüldüğü gibi programımız aynı zamanda **NumberFormatException nfe** terimiyle integer olan rakamımızın integer olarak gönderilip gönderilmemiğini kontrol etmektedir. Bu exception java kütüphanesindeki **parsInt** metodu tarafından gönderilmektedir.

8.2 ALIŞTIRMALAR

1. H6O1.java Hata (Exception) kontrolu programı. **gercekSayiException** ve **bayagikesirException** tanımlanmıştır.

[gercekSayiException.java](#) programının **gercek sayı girildiginde hata(Exception) vermek için** **gerceksayiException** tanımlanmıştır. [bayagikesir1.java](#) programındaki

```
public bayagikesir(double npay,double npayda) throws gercekSayiException
{
    //*** tanimlayiniz
}
```

kurucu metotunu bu exception'i kullanacak şekilde oluşturunuz. [H5O1.java](#) programının benzeri [H6O1.java](#) programında bayagikesir yerine bayagikesir1 kullanarak exception yapınızı kontrol ediniz.

Problem 8.3 : gercekSayiException, gercek sayı hata tanım programı

```
public class gercekSayiException extends NumberFormatException
{
    public gercekSayiException()
    {
        super("gercek sayı girdiniz tam sayı girmelisiniz");
    }
}
```

Problem 8.4 : bayagikesirException, bayağı kesir sıfır bölme hata tanım programı

```
class bayagikesirException extends Exception
{
    public bayagikesirException()
    {
        super("Bayağı kesir paydasinin degeri 0.");
    }
}
```

Problem 8.5 : bayagikesir1.java, hata kontrollü bayagikesir sınıfı

```
=====
// bayagikesir sınıfı tanımı
// Dr. Turhan Coban
=====
import java.io.*;
import bayagikesirException;
import gercekSayiException;

// sınıf bayagikesir
// bu sınıf bayagikesir sayıların matematik
// işlemlerini tanımlar
//

class bayagikesir {
    // sınıf değişkenleri
    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();
}
// giri  -   k    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 sayı 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 sayıyla 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 sayıyla topla
pay = pay - sag*payda;
sadelestir();
}

public void carp(bayagikesir sag ) throws bayagikesirException
{
```

```

//bir bayagikesir sayıyla carp
pay = pay*sag.Pay();
payda = payda*sag.Payda();
sadelestir();
}

public void carp(int sag ) throws bayagikesirException
{
//bir gerekci sayıyla 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 sayın toplamını return deyimiyle bayagikesir olarak
// aktarır
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 sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
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 sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
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'linmelerini 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 sayi yada double sayi 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 hazır bayagikesir formda String deşiskeni 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, bayağı kesir sıfıra bölme hata test programı
H6O1.java**

```

/bu sınıf sınıf değişkeni kompleksi boyutlu olarak çağırılır.
//not buradaki bayagikesir sınıfı bayagikesir1.java programı
//tarafından yaratılmıştır.

```

```

import java.io.*; //java girdi çıktı sınıfını çağır
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 sayısı, 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());
}
}

```

2. H8AL1_2000, Bayagikesir hata kontrolu programı çıktısı, JOptionPane kullanıyor.

```

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 Sayı Hatası",
JOptionPane.ERROR_MESSAGE);
i--;
}
catch (bayagikesirException e)
{
JOptionPane.showMessageDialog(null,e.getMessage(),"Bayağı kesir Hatası",
JOptionPane.ERROR_MESSAGE);
}
}

for(i=0;i<sayi.length;i++)
{
try{
toplams.topla(sayi[i]);
}
catch (bayagikesirException e)
{
JOptionPane.showMessageDialog(null,e.getMessage(),"Bayağı kesir Hatası",
JOptionPane.ERROR_MESSAGE);
}
}
s+="Ortalama = "+toplams.toString();

try{
toplams.bol(sayi.length);
}
catch (bayagikesirException e)
{
JOptionPane.showMessageDialog(null,e.getMessage(),"Bayağı kesir Hatası",
JOptionPane.ERROR_MESSAGE);
}
s+="Ortalama = "+toplams.toString();
JOptionPane.showMessageDialog(null,s,
"Bayağı kesir Ortalaması",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```

9. PARALEL KULLANIM (MULTITHREADING) , GERÇEK ZAMAN UYGULAMALARI, ANİMASYON

9.1 PARALEL KULLANIM(MULTITHREADING) VE GERÇEK ZAMAN PROGRAMLANMASI

İşlemleri tek tek sırayla ve iyi bir şekilde yapabilseydik belki de çok mükemmel sonuçlar alabilirdik. Fakat Dünyadaki hemen hemen hiçbirşey tek tek sırayla çalışmaz. İnsan vücudu da dahil bir çok doğal sistem paralel olarak çalışır. Yani bir işlem yapılrken parelelinde diğer bir işlem yapılmaya devam eder. Bizim şu anda kullandığımız bilgisayarların büyük bir bölümü tek bir işlemciye(processor) sahiptir. Birden fazla işlemciye sahip paralel bilgisayarlarda, her bir işlemcide bir iş tanımlayarak paralel kullanım mümkündür. Fakat tek işlemcili sistemler için(seri bilgisayar) yazılmış olan program dilleri genel olarak birden fazla işlemin aynı anda yapılmasına izin vermezler. Modern bilgisayar dilleri arasında sadece **Ada** ve **Java** dilleri bir işlemcide birden fazla işlemin (programın) aynı anda kullanılmasına izin verir. C++ dili birden fazla programın aynı anda kullanılmasına izin vermez. Bu yüzden tek kullanımlı bir programlama dilidir.

Parelel Kullanım(Multithreading) Tek bir işlemcinin birden fazla program tarafından kullanılmasıdır. Bu kullanım CPU (merkez program işlemcisi) adını verdigimiz temel bilgisayar işlem ünitesinin çeşitli programlar tarafından ortaklaşa kullanılması prensibine göre yapılır. Bilgisayar CPU zamanı küçük parçalara bölünerek her parçada ayrı bir bilgisayar programının çalışması ve diğer bilgisayar programlarının beklemesi sağlanır. Programlar küçük paketçikler halinde program işlemcisine gönderilir ve sonuçlar tekrar hafızaya aktarılır. Sonra ikinci bir program parçacığı işlemci belleğine gönderilir. Modern seri bilgisayarlarda cache adı verilen bu program parçacıkları çalışırken diğer program parçacıklarını tutabilen hızlı hafıza birimleri de bulunur.Tüm bu kontrolleri ve birden fazla programın tek işlemcide çalışma işlemini java dilinde doğrudan programlayabiliriz.

CPU nun Parelel kullanımı Runnable interface'i veya abstract Thread sınıfını kullanarak gerçekleştirilir. Aşağıda Bu iki yolu kullanarak bir digital saat programı oluşturulmuştur. Bu programlarda saat sadece tek bir program olmasına rağmen bilgisayarın belleğinde eğer devamlı çalıştırılacak olursa başka bir hibit programın çalıştırılamayacağı düşünülerek paralel kullanımlı olarak verilmiştir.

Zaman içinde programları bilgisayarda kontrollu yapıda kullanmak aynı zamanda bize gerçek zaman programlama olasılıklarını da sunar. Paralel ve gerçek zama kullanımı javayı kuvvetli bir modern programlama dili yapan en önemli özelliklerden biridir.

Program 9.1 Saat.java sınıfının thread sınıfından türemiş olarak yazılımı.

```
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 sınıfının thread sınıfından türemiş olarak yazılımı.

```

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 (Greenwich zamanına 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 saatSWF.java digital saat JFFrame çıktısı

Program 9.3 Saat.java sınıfının Runnable interface'sı ile yazılımı ve applet çıktısı

```

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



Şekil 9.2 saat.html appletinde saat theradinin işletilmesi

Program 9.4 saatRunnableSWF sınıfının Runnable interface'sı ile yazılımı

```

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 zamanına 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 görüldüğü gibi thread adı verilen ve bilgisayar belleğinde diğer programlarla beraber çalışmasını istediğimiz program parçacıklarını java programının içinde başlatabiliyoruz(start), kontrol edebiliyoruz veya durdurabiliyoruz(stop). Program parçacığının çalışmadan bekleyeceği sureyi de sleep metodunu kullanarak ayarlıyoruz.

9.2 PARALEL PROGRAM KULLANIMIDA HAFIZA SİNKRONİZASYONU

İkinci olarak biraz daha karışık bir örneğe göz atalım. Bu örnekte, iki tane program bilgisayar belleğinde aynı anda çalışacaktır. Birinci program ikinci programla ortak olarak kullanılan bir bilgisayar int hafıza noktasına sırasıyla birden ona kadar değişen bir rakam yazacak, ikinci program da aynı noktadan (OrtakHucreden) yazılın değeri okuyacaktır.

Program 9.5 OrtakHucre.java programı

```

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+" değerini üretti");
try{
sleep( (int) (Math.random()*5000 ) );
}
catch(InterruptedException e)
{
System.err.println("Exception "+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 çıktı :

```

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’ı ile çalışan [OrtakHucreR.java](#) sınıfı

```

import java.util.*;
import java.awt.*;
import java.io.*;/runnable interfacei kullanarak

class UretInteger implements Runnable
{

```

Program 9.7 Buffer ile uyum sağlanmış OrtakHucreBuf.java sınıfı

```
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 Kullanicilan "+intdegeri+" Okundugu Hucre "+YerelOku);
YerelOku = ++YerelOku % 5;
cikti.append("\t yazilan hucre "+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 sağlanmış [OrtakHucreBuf.java](#) sınıfı çıktısı

**Uretilen 0 Yazilan 0 yazilan hucre 1 okunan hucre 0 depo: 0 9 9 9 9
Kullanicilan 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
Kullanicilan 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
Kullanicilan 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
Kullanicilan 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
Kullanicilan 4 Okundugu Hucre 4 yazilan hucre 3 okunan hucre 0 depo: 5 6 7 3 4
Kullanicilan 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
Kullanicilan 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
Kullanicilan 7 Okundugu Hucre 2 yazilan hucre 0 okunan hucre 3 depo: 5 6 7 8 9
Kullanicilan 8 Okundugu Hucre 3 yazilan hucre 0 okunan hucre 4 depo: 5 6 7 8 9
Kullanicilan 9 Okundugu Hucre 4 yazilan hucre 0 okunan hucre 0 depo: 5 6 7 8 9
INTEGER DEPOSU BOS**

Problem 9.8 Uyum sağlanmış [OrtakHucreSync.java](#) sınıfı

```

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+"olarka 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+" yakalandı");
        }
    }
}
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 sağlanmış [OrtakHucreSync.java](#) sınıfı sonuçları

UretInteger OrtakInteger degerini 0 olarak olusturdu
KullanInteger 0 yakaladı
UretInteger OrtakInteger degerini 1 olarak olusturdu
KullanInteger 1 yakaladı
UretInteger OrtakInteger degerini 2 olarak olusturdu
KullanInteger 2 yakaladı
UretInteger OrtakInteger degerini 3 olarak olusturdu
KullanInteger 3 yakaladı
UretInteger OrtakInteger degerini 4 olarak olusturdu
KullanInteger 4 yakaladı
UretInteger OrtakInteger degerini 5 olarak olusturdu
KullanInteger 5 yakaladı
UretInteger OrtakInteger degerini 6 olarak olusturdu
KullanInteger 6 yakaladı
UretInteger OrtakInteger degerini 7 olarak olusturdu
KullanInteger 7 yakaladı
UretInteger OrtakInteger degerini 8 olarak olusturdu
KullanInteger 8 yakaladı
UretInteger OrtakInteger degerini 9 olarak olusturdu
KullanInteger 9 yakaladı

Programlardan da görüleceği gibi UretInteger sınıfı (thread'i) TutInteger sınıfını kullanarak yeni bir integer değeri yaratmakta ve KullanInteger sınıfı da bu değeri ortak hafızadan okumaktadır. Burada orijinal programlarda iki thread'in çalışma sıralarının sırayla gitmemesi problemi çıkmıştır. Bazen UretInteger programı birden fazla çalışmakta bazende KullanInteger programı birden fazla çalışmaktadır. Bunu onlemek için OrtakHucreSync programında TutInteger sınıfının içinde Uyum kontrolü yapılmıştır. boolean yazılabilir değişkenini bir trafik işaretи gibi kullanıp UretInteger ve TutInteger threat programlarının sırasıyla işleme girmesi, bu arada diğerinin beklemesi sağlanmıştır. Buradaki zorluk. Bazen bir programın çok fazla

bekleyebilmesi olasılığıdır. Bunu engellemek içinde OrtakHucreBuf programında yazmak için bir yerine beş integer yeri tanımlanmış, ve okutma ve yazma işlemlerinde iki ayrı kontrol boolean deyimi kullanılmıştır.

9.3 ALIŞTIRMALAR

1. daha önce tanımladığımız yildiz.java (problem 6.19) programını da kullanarak burada analog bir saat tanımı yapılmıştır. Analog saat her saniyede bir çalışmaktır ve kendini tekrar çizmektedir. Programı inceleyiniz.

Program 9.9 : analogSaat.java programı, yildizCiz2Thread ve analogSaat sınıfları programı

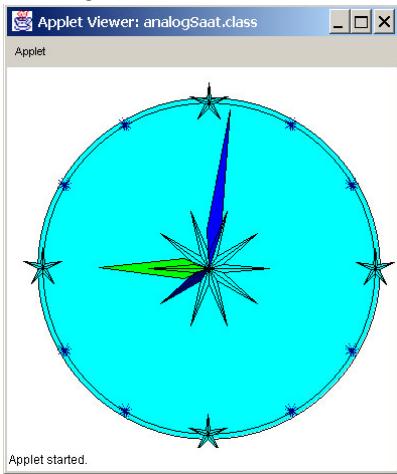
```
import java.applet.Applet;
import java.util.*;
import java.awt.*;
import yildiz;
import polar;
import java.awt.event.*;

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());}
        }
    }
} //yıldızciz2Thread sınıfının 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



Şekil 9.3 analogSaat.html appleti

Aynı programın Swing JFrame versiyonu :

Program 9.10 : analogSaatSWF.java JFrame programı, analog saatı swing Frame olarak tanımlar

```
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());}
}
}
}

} //yldizCiz2Thread sınıfı nın 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 zamanına 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);
}
}

```

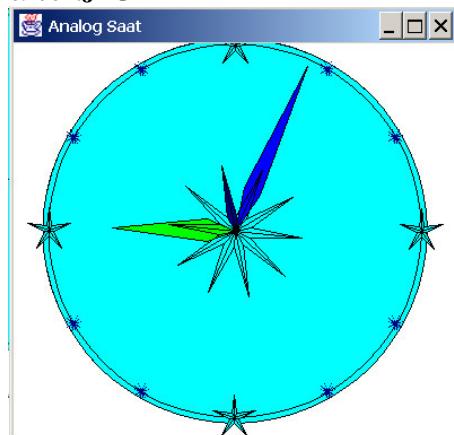
```

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



Şekil 9.4 analogSaatSWF.java JFrame çıktısı

Aynı programın swing Japplet versiyonu :

Program 9.11 : analogSaatSWA.java JApplet programı, analog saatı swing Applet olarak tanımlar

```
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());}
}
}

} //yildizCiz2Thread sınıfının sonu

public class analogSaatSWA extends JApplet
{
yildizCiz2ThreadSWA y;
double aci1,aci2,aci3;
Date c;
polar p;

public void init()
{
// Türkiye standart (Greenwich zamanına 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){}
}
}

```

2. [trafikIsigi.java](#) programını inceleyiniz.

Program 9.12 : trafikIsigi.java programı

```

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 programı, trafik ışığı panelini tanımlar

```

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';
        g2.setColor(Color.lightGray);
        g2.fill(elips2);
        g2.fill(elips3);
        g2.setColor(Color.red);
        g2.fill(elips1);
        break;
    case 'g':
        which_light='y';
        g2.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 programı, trafik ışığı tanımlar ve çalıştırır

```
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=ContentPane();
        JPanel p1=new JPanel();
        p1.setLayout(new GridLayout(2,4));
        yik=new isikKontroluThread1(this);
        yik.start();
        ti1=new trafikIsigi1SWP(yik);
        light_switch = new JButton("Işığı değiştir");
        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 ışık yanma zamanı :  ");
        l2=new JLabel("saniye          ");
        l3=new JLabel("kirmizi ışık yanma zamanı : ");
        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)
```

```

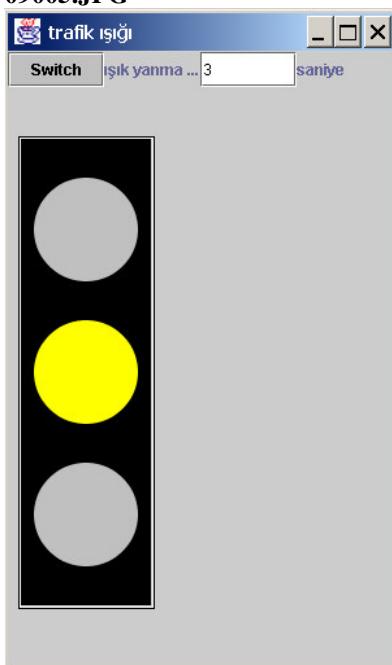
{
    Double i1=new Double(t1.getText());
    yik.setKirmiziZaman(i1.doubleValue());
}
}

public static void main(String[] args)
{
    trafikIsigi1SWF pencere= new trafikIsigi1SWF();
    pencere.addWindowListener(new BasicWindowMonitor());
    pencere.setSize(300,600);
    pencere.setVisible(true);
}

}

```

09005.JPG



Şekil 9.5 trafikIsigiSWF.java çıktısı

4. H8O1

trafikIsigi programını inceleyiniz. Bu programda yeşil ve kırmızının aynı sürede yanlığını göreceksiniz. (sarı 1 saniye yanmaktadır.) Yeşil ve kırmızının ayrı ayrı sürelerde yanmasını sağlayınız. Yanma sürelerini JTextField alanından kontrol ediniz.

Program 9.15 H8O1.java programı trafik ışığı 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 ışığı 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 ışığı programı

```

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);

```

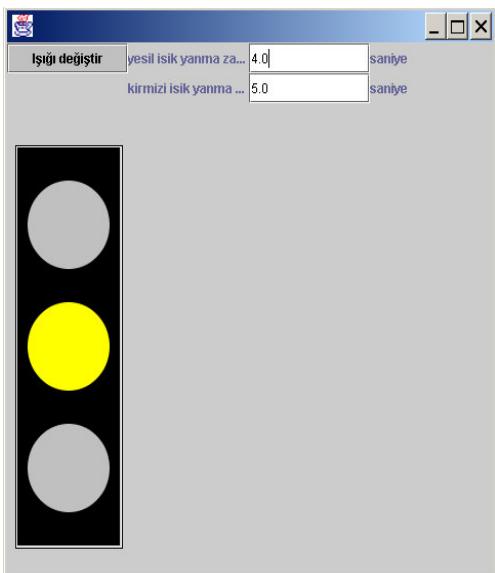
```

light_switch = new JButton("İşığı değiştir");
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



Şekil 9.6 trafikIsigi1.java trafik ışığı kontrol programı JFrame çıktısı

3. H8O2

Birbirinden bağımsız çalışan iki trafik ışığını iki ayrı ışık olarak simule edin. Bu ışıklardan birincisinde yesil veya sarı yanarken ikincisinde kırmızı yansın, ikincisinde yeşil veya sarı yanarken birincisinde kırmızı yansın. Kırmızı ve yeşil süreleri aynı olmayabilir, fakat birbirinin tersi olacaktır. Örneğin birinci ışıkta 6 saniye kırmızı, 10 saniye yeşil, 1 saniye sarı yanıyorsa, ikinci ışıkta 11 saniye kırmızı, 5 saniye yeşil, 1 saniye sarı yanacaktır. Birinci ışık sarıdan kırmızıya geçtiği anda ikinci ışık yeşil yanacaktır.

Program 9.18 : [IkiTrafikIsigi.java](#) programı

```

>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 tarafından paint metodu
// 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 ışık yanma zamanı : ");
        l2 = new Label("saniye");
        l3 = new Label("yesil ışık yanma zamanı : ");
        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](#) programı

```

//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';

```

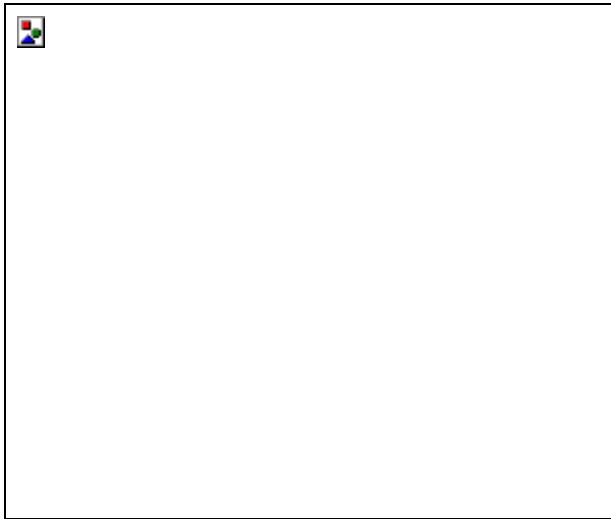
```

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



Şekil 9.7 sinkronize olarak çalışan iki trafik ışığı

4. H8OD2_2000

10 m³ boyutunda bir depomuz var. Bu depoya birinci pompa su doldurmakta, ikinci pompa is suyu boşaltmaktadır. Pompaların debileri değiştirilebilmektedir, ve kullanıcı tarafından gerektiğinde elle açılıp kapatılabilir. Depo tamamen boşalırsa boşaltan pompa otomatik olarak durdurulacaktır. Depo tamamen dolarsa dolduran pompa otomatik olarak durdurulacaktır. Bu prosesi gerçek zamanda simule eden iki pompa ve bir depodan oluşan bir sistem oluşturunuz, sonuçları belli zaman aralıklarıyla ekrana yazdırınız (pompa1, pompa2 debileri ve depodaki su miktarı ve başlangıçtan itibaren geçen zaman) veya bir grafik çıktı üzerinden simulasyon çıktısını oluşturunuz. (aynı değerleri grafik çıktıda gösteriniz) Depodaki su miktarını bir ortak double değeri olarak düşünebiliriz.

Program 9.20 : H8OD2_2000.java programı

```
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;
    }
    else
        System.out.println("Pompa1'in debisi " + suMiktari + " dir. Depo daki su miktarı :" +
OrtakSu);

        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;
        System.out.println("Pompa2'in debisi " + suMiktari + " dir. Depo Bos!!! (0).");
        OrtakSu = 0;
    }
    else
        System.out.println("Pompa2'in debisi " + suMiktari + " dir. Depo daki su miktarı :" +
OrtakSu);
        pompa1 = true;
        notify();
    }
}

```

9. AWT Animasyon örneği

Bu problemde animasyon'un thread kullanarak nasıl yapılacağını göreceğiz. Aslında kavram yeni değil aslında şu ana kadar bu bölümde yaptığımız tüm uygulamalar bir animasyon. Bu program java döküman kütüphanesinden alınmıştır.

Program 9.21 : AnimatorApplication.java programı

```

/*
 * awt animasyon
 */

import java.awt.*;
import java.awt.event.*;

/*
 * Arthur van Hoff'un animasyon örneği.
 * Bu örnek tüm animasyonlar için örnek olarak kullanılabilir.
 */
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 şey yapmanız gerekmek
        //
    } else {
        //Animasyon başlıyor...
        if (animatorThread == null) {
            animatorThread = new Thread(this);
        }
        animatorThread.start();
    }
}

public void stopAnimation() {
    //Animasyon thread'ini durdur
    animatorThread = null;
}

public void run() {
    //Önce bu thread'in önceliğini düşürüyoruz,
    // böylece öbür programlara engel olmaz
    Thread.currentThread().setPriority(Thread.MIN_PRIORITY);

    //Başlama zamanını hatırla
    long startTime = System.currentTimeMillis();
}

```

```

//hangi thread olduğumuzu hatırla
Thread currentThread = Thread.currentThread();

//gerçek zaman animasyon döngüsü
while (currentThread == animatorThread) {
    //Animasyon ekranındaki rakamı artır
    frameNumber++;

    //rakamı göster
    repaint();

    //beklemeye al
    try {
        startTime += delay;
        Thread.sleep(Math.max(0,
            startTime-System.currentTimeMillis()));
    } catch (InterruptedException e) {
        break;
    }
}

//animasyon ekranını 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 sayısını 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



Şekil 9.8 awt animasyon örneği

10. swing Animasyon örneği

Bu örnek problemde bir öncekinin temel olarak aynısı, sadece swing kullanılmış hali

Program 9.22 : AnimatorApplicationSW.java programı

```

/*
 * Swing versiyonu
 */

```

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

/*
 * Arthur van Hoff'un animasyon örneği.
 * Bu örnek tüm animasyonlar için örnek olarak kullanılabilir.
 */
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 başlat
        if (animatorThread == null) {
            animatorThread = new Thread(this);
        }
        animatorThread.start();
    }
}
```

```

public void stopAnimation() {
    //animasyon treadini durdur.
    animatorThread = null;
}

public void run() {
    //önceliği düşür.
    Thread.currentThread().setPriority(Thread.MIN_PRIORITY);

    //başlama zamanı
    long startTime = System.currentTimeMillis();

    //thread tanımı
    Thread currentThread = Thread.currentThread();

    //animasyon döngüsü
    while (currentThread == animatorThread) {
        //animasyon çerçevesini bir artır
        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



Şekil 9.9 swing animasyon örneği

11. zaman ayarlı swing Animasyon örneği

Program 9.23 : AnimatorApplicationTimer.java programı

```
/*
 * Tam anlamıyla swing'e dönüştürülmüş animasyon
 */

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

/*
 * Arthur van Hoff'un animasyon örneğinden yazılmıştır.
 */
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



Şekil 9.10 zaman ayarlı swing animasyon örneği

BÖLÜM 10 JAVA GİRDİ - ÇIKTI PROGRAMLANMASI

10.1 JAVANIN I/O (GİRDİ ÇIKTI) KÜTÜPHANESİNDE YER ALAN SINIFLAR VE INTERFACE'LER

Javadaki tüm girdi çıktı programlaması veri akış (stream) kavramına dayanır. Veri akışını bir su akışı veya bir kablolu sistem üzerinden bilgi akışı gibi düşünülebiliriz. Javada sadece dosya değil yazıcı, scanner gibi herhangi bir kaynağa veri iletmek istediğimizde bu veri akış sistemini kullanırız. Javada oldukça geniş bir veri akış kütüphanesi bulunmaktadır.

java.io kütüphanesinde yer alan girdi çıktı programlama sınıfları şunlardır :

```
InputStream
    FileInputStream
        LineNumberInputStream
        BufferedInputStream
        DataInputStream
        PushbackInputStream
    ByteArrayInputStream
    FileInputStream
    ObjectInputStream
    PipedInputStream
    SequenceInputStream
    StringBufferInputStream
OutputStream
    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 sınıfların hepsinin genel görevi javaya girdi sağlamak ve çıktı almaktır. Burada bunlardan bir kısmının işlevlerini ve nasıl çalışıklarını inceleyeceğiz. Önce temel ana sınıfların bazlarının tanımlarına daha detaylı

bakalım. InputStream sınıfı byte türü yazı okuma sınıfıdır. Tüm byte türü bilgi akış kanalı sınıflarının ana sınıfıdır. Tanımı :

```
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 sınıfının byte türü yazma pareleli olan OutputStream sınıfının tanımı 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 başlamanoktası,int uzunluk) throws IOException;
}
```

Abstract sınıf Reader, InputStream sınıfına oldukça yakındır, yalnız data transferinde byte yerine char kullanır, böylece girdi çıktıda unicode stream kullanabilme olasılığı 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 başlamanoktası,int uzunluk) throws IOException;
    public boolean ready() throws IOException;
    public void reset() throws IOException;
    public long skip(long n) throws IOException;
}
```

Abstract sınıf Writer da Output straem sınıfına eşdeğerdir. Fakat aynı readerda olduğu gibi bu sınıf da veri akışında char ve string kullanır.

```
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 başlamanoktası,int uzunluk) throws IOException;
    public void write(String s) throws IOException;
    public void write(String s,int başlamanoktası,int uzunluk) throws IOException;
}
```

Bu sınıflar temel sınıflardır. Bunların altında yer alan çeşitli sınıflara aşağıdaki bölümlerde daha detaylı bakacağız.

10.2 FILE (DOSYA) SINIFI

File sınıfı girdi çıktı işlemleri saflarken, dosya isimleri ve bulundukları dizin ile ilgili bilgi verir.

File sınıfının tanımı :

Public class File extends Object implements Serializable, Comparable

{

```
    public static final String pathSeparator;
    public static final char pathSeparatorChar;
    public static final String separator;
    public static final char separatorChar;
    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 başkabirdosya);
    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 getAbsoluteFile();
    public File getAbsolutePath();
    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 filtrefonksiyonu);
    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 metodların çağırılmasına örnek olarak :

```
File f1=new File("/");
File f2=new File("/", "autoexec.bat");
File f3=new File(f1, "autoexec.bat");
```

tanımlarını verebiliriz. File sınıfında tanımlanmış olan Metotlardan bazılarının tanımları şunlardır.

canRead()

Tanımlanan dosyadan bilgi okunabilirliğini test eder.

canWrite()

Tanımlanan dosyaya bilgi yazılabilirliğini test eder.

delete()
 Dosyayı siler

equals(Object)
 Dosya ismini Object te verilen dosya ismiyle karşılaştırır.

exists()
 Dosyanın mevcut olup olmadığını kontrol eder.

getAbsolutePath()
 Dosyanın tam dizin ismini ve dosya ismini verir(co/java/prog/Hosgeldiniz.java).

getName()
 Dosyanın ismini verir.(Hosgeldiniz.java)

getParent()
 Dosyanın içinde bulunduğu dizinin ismini verir.(prog)

getPath()
 dosyanın ismini ve içinde bulunduğu dizinin ismini verir.(prog/Hosgeldiniz.java)

isAbsolute()
 Dosya isminin mutlak isim olup olmadığını kontrol eder.(Eğer dosya ismi co/java/prog/Hosgeldiniz.java olarak verilmişse true değeri verir.)

isDirectory()
 verilen isimin bir dizin ismi olup olmadığını kontrol eder.

isFile()
 verilen isimin birdosya isimi olup olmadığını kontrol eder.

lastModified()
 Dosyanın en son değiştirildiği tarihi verir

length()
 Dosyanın boyutunu Byte olarak verir.

list()
 Verilen dizinin içindeki dosyaların listesini verir.

list(FilenameFiltre)
 Verilen dizinin içindeki Filtre nesnesindeki tanıma uyan dosyaların listesini verir.

mkdir()
 Yeni bir dizin oluşturur.

mkdirs()
 O anda tanımlı olan dizinin içine bir alt dizin oluşturur.

renameTo(File)
 Dosyanın ismini değiştirir.

toString()
 Dosya ve dizin isimleri topluluğunun String değişkeni eşdeğerini verir.

File(dosya) sınıfındaki metodlardan da görüleceği gibi bu sınıf dosyadan okuyup yazma gibi bir işlem yürütmez. File sınıfının temel işlevi girdi çıktı dosyasının adlandırılması ve isimle ilgili fonksiyonlara ulaşılmasıdır. File sınıfının bazı metodlarını ufak bir program içinde kullanalım.

Program 10.1 . FileTestiSW.java programı

```
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.*;
```

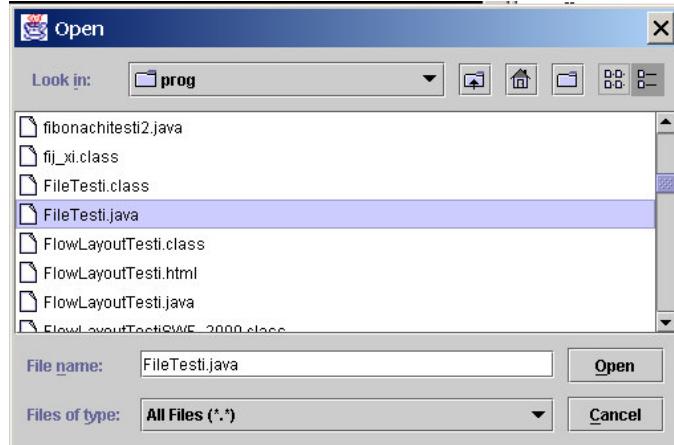
```
public class FileTestiSW
{
    public static void main(String args[])
    {
        //önce JFileChooser kullanarak dosyayı seç, sonra dosya özelliklerini göster.
        File f=new File("Hosgeldiniz.java");
        JFileChooser dosyasec=new JFileChooser();
        int onay = dosyasec.showOpenDialog(null);
        //veya kayıt 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 değil "+"\n";
s+=f.canWrite() ? "Yazılabilir " : " Yazılamaz)+"\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 değil)+"\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);
}
}

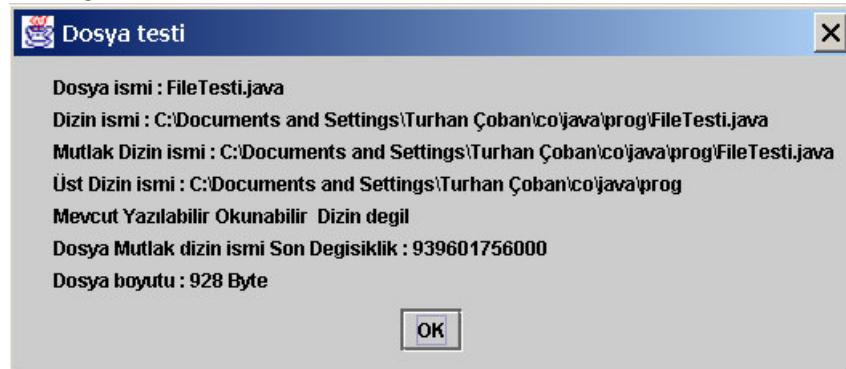
```

10001.JPG



Şekil 10.2 FileTesti.java programının içindeki dosya seçim penceresi

10002.JPG



Şekil 10.2 FileTesti.java programının sonucları

Burada swing hazır dosya seçme programı JfileChooser kullanılarak dosya ismi seçilmiştir, sonar bu dosya ile ilgili çeşitli bilgiler JOptionPane grafik ortamında aktarılmıştır.

10.3 ARDIŞIK (SEQUENTIAL) DOSYA YARATILMASI

Java bilgi dosyalarını birbiri ardısırı gelen byte veya char (unicode) akışı olarak görür. Her dosya dosya-bitiş işaretile sonlanır. Yeni bir dosya açıldığında bu dosyayla ilgili bir nesne oluşturulur. Yeni bir java programı açıldığında üç akış nesnesi otomatik olarak açılır. Bunlar **System.in** , **System.out**, **System.err** nesneleridir. Eğer kendimiz bir girdi veya çıktı akış nesnesi oluşturmak istersek **FileInputStream** veya **OutputStream** sınıfı bir nesne tanımlayabiliriz. Akısın içindeki değişkenleri sadece byte olarak okumak yerine direkt olarak double veya integer gibi değişkenler türünden okumak istersek **DataInputStream** ve **DataOutputStream** sınıflarını kullanabiliriz. Genel olarak OutputStream ve DataOutputStream sınıflarını aşağıdaki gibi bir arada kullanabiliriz:

```
DataStream ciktigimi;
try{ ciktigimi=new DataOutputStream(new FileOutputStream("isim.dat"));
    catch(IOException e) {System.out.println("Dosya acilamadi\n"+e.toString());}
```

Bu deym bize isim.dat isimli dosyadan bir fileOutputStream aracılığıyla açılan dosyayı DataOutputStream'e çevirip gerçek değişkenler cinsinden akış kanalı açar. DataOutputStream ve OutputStream sınıflarının tanımları :

```
public class DataOutputStream extends FilterOutputStream implements DataOutput
{
    public DataOutputStream(OutputStream o);
    protected int written
    public void flush()
    public final void size()
    public synchronized void write(byte[],int baslangicnoktası,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 sınıfı File ile DataOutputStream sınıfı arasındaki gerekli tanımları sağlar

```
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[])
    public void write(byte[], int, int) throws IOException;
    public void write(int) throws IOException;
    public final FileDescriptor getFD() throws IOException;
}
```

Kanal açıldıktan sonra **DataOutputStream** sınıfının metodlarını kullanarak dosyaya yazı yazabiliz. Yazma metodlarının işlevlerine bir göz atarsak :

write(byte B[])
byte tipi boyutludeğişkeni açılmış olan DataOutputStream kanalına yazar.
write(byte B[], int baslangicindeksi, int indeksboyutu)
byte tipi boyutludeğişkeni açılmış olan DataOutputStream kanalına yazar.

Baslangıç indeksinden baslar ve indeksboyutu uzunluğundaki kısmı yazar.
write(int B) int datayı data output stream'e yazar.
writeBoolean(boolean B) Boolean değişkeni DataOutputStream'e yazar.
writeByte(int) int değişkeni byte olarak DataOutputStream'e yazar.
writeBytes(String) String değişkeni byte boyutlu değişken olarak DataOutputStream'e yazar.
writeChar(int) int değişkeni char değişken olarak DataOutputStream'e yazar.
writeChars(String) String değişkeni Char değişken olarak DataOutputStream'e yazar.
writeDouble(double) double değişkeni double değişken olarak DataOutputStream'e yazar.
writeFloat(float) float değişkeni float değişken olarak DataOutputStream'e yazar.
writeInt(int) float değişkeni float değişken olarak DataOutputStream'e yazar.
writeLong(long) long değişkeni long değişken olarak DataOutputStream'e yazar.
writeShort(int) int değişkeni short (short integer) değişken olarak DataOutputStream'e yazar.
writeUTF(String) String değişkeni unicode UTF-8 formatı karekter boyutu olarak yazar.

Bu şekilde yazdığımız bir veri akışın 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);
}
```

tanımını kullanabiliriz. Buradaki DataInputStream sınıfının tanımı :

```
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 başlangıçdeğeri, int uzunluk) throws IOException;
    public final void readFully(byte b[]) throws IOException;
    public final void readFully(byte b[],int başlangıçdeğeri, 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;
}
```

şeklindedir. 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 başlangıçnoktası,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;
}
```

şeklinde tanımlıdır. **DataInputStream** metdunun alt okuma metodlarına biraz daha detaylı göz atarsak:

read(byte[])

byte veri akışını direk okur.

read(byte[], int başlangıçnoktası, int boyut)

byte veri akışını başlangıçnoktası indeksinden boyut uzunluğuna kadar okurdirek okur.

readBoolean()

boolean değişken (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 değişkenini, byte[] boyutlu girdi değişkeninin boyutuna göre okur.

readFully(byte[], int, int)

readInt()

Integer okur

readLine()

Bir satırı satırbaşı yap komutuna kadar okur. Java 1.0 komutudur. 1.1 ve üzerinde bazı okuma hataları yaptığı görüldüğünden kullanılmaması tavsiye edilir.

readLong()

Long tamsayı değişkeni okur.

readShort()

Short tamsayı değişkeni okur

readUnsignedByte()

8-bitlik işaretsiz tamsayı değişkeni okur.

readUnsignedShort()

16-bitlik işaretsiz tamsayı değişkeni okur.

readUTF()

Unicode karekteri okur.

readUTF(DataInput)

Unicode Karakterini (Character) DataInput girdi akışından okur.

skipBytes(int n)

n byte değeri okumadan atlar.

Şimdi bir örnek problem ile bu yazma ve okuma işlemine daha yakından bir göz atalım. Program 10.2 de yeni bir dosya açılmakta, ve dosyaya veri girilmektedir. Program 10.3 de aynı programın swing kullanılarak yazılmış versiyonu mevcuttur. Program 10.4 de ise yaratılan dosyaya girilen veriler okunmaktadır.

Program 10.2 ardisikDosyaYarat.java programı

```
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 ciktig;
    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");
}
}
```

```

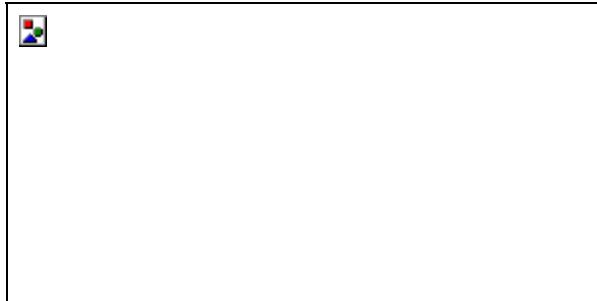
        catch(IOException io)
    {
        System.err.println("Dosyaya yazarken hata olustu\n"+io.toString());
        System.exit(1);
    }
}
}

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



Şekil 10.3 ArdisikDosyaYarat.java programı ve sonuclarının Frame çıktısında görülmesi

Program 10.3 ardisikDosyaYaratSWF_200.java programı

```

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("Müşteri dosyası Açı");
        c=getContentPane();
        try{
            cikti=new DataOutputStream(new FileOutputStream("musteri.txt"));
        } catch(IOException e)
    }
}

```

```
{ System.err.println("Dosya doğru açılamadı\n"+e.toString());
System.exit(1);
}
c.setLayout(new GridLayout(5,2));
H=new JLabel("Hesap numarası :");
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("Çıkış");
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 numarası tam sayı değil olmalıdır");
}
catch(IOException io)
{
System.err.println("Dosyaya yazarken hata oluştu\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



Şekil 10.4 [ArdisikDosyaYaratSWF.java](#) programı ve sonuçlarının Frame çıktısında görülmesi

Program 10.4 ardisikDosyaOku.java programı

```

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 :"));

```

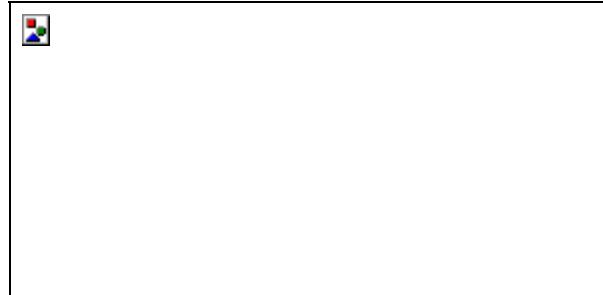
```
firstNameField=new TextField(20);
add(firstNameField);
add(new Label("soyisim : "));
lastNameField=new TextField(20);
add(lastNameField);
add(new Label("Hesaptaki para : "));
balanceField=new TextField(20);
add(balanceField);
next=new Button("bir sonraki hesap");
next.addActionListener(this);
add(next);
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\x09 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 Hatası olustu\n"+e.toString());
System.exit(1);
}
```

```

        }
    }
public static void main(String args[])
{
new ardisikDosyaOku();
}
}

```

10004.JPG



Şekil 10.4 [ArdisikDosyaOku.java](#) programı ve sonuçlarının Frame çıktısında görülmesi

ArdisikDosyaYarat.java programında **musteri.txt** dosyası

```

try{
cikti=new DataOutputStream(new FileOutputStream("musteri.txt"));
}
catch(IOException e)
{
System.err.println("Dosya doğru açılamadı\n"+e.toString());
System.exit(1);
}

```

gurubunu kullanarak açılmıştır. Bu gurubu File sınıfını da kullanarak

```

File f=new File("musteri.txt");
f=f.getAbsolutePath();
try{
cikti=new DataOutputStream(new FileOutputStream(f));
}
catch(IOException e)
{ System.err.println("Dosya doğru açılamadı\n"+e.toString());
System.exit(1);
}

```

şeklinde de yaratabildik. İkinci şekilde dosyayı tanımlarken tam dizin ismini de otomatik olarak tanımlamış olduk.

Dosyanın okunması içinse

```

try{
input=new DataInputStream(new FileInputStream("musteri.dat"));
}
catch(IOException e)
{
System.err.println("Dosya açılamadı\n"+e.toString());
System.exit(1);
}

```

gurubunu kullandık. Eğer dosya ismi de değişkende dışarıdan okuduktan sonra ismi File sınıfına yükleyip sonra akış kanalı açabiliriz.

Dosya ile işlemler bittikten sonra dosya kapanır.

```
private void closeFile()
{
try{
input.close();
System.exit(0);
}
catch(IOException e)
{
System.err.println("Dosya Kapama Hatası oluştu\n"+e.toString());
System.exit(1);
}
}
```

closeFile() metodu dosya akışını kapatmak amacıyla oluşturulmuştur.

Bu örnekte kullandığımız, **DataOutputStream**, **FileOutputStream**, **DataInputStream**, **FileInputStream** girdi çıktı akış kontrolleri temel olarak daha önce tanımlarını vermiş olduğumuz **OutputStream** ve **InputStream** sınıflarından türetilmiş sınıflardır. Ve temel olarak byte değişken türü üzerinden veri akışını sağlamaktadır.

Yine yukarıda tanımlanmış olan Reader ve Writer sınıfları ise aynı işlemi char sınıfı veri akışı üzerinden yapmaktadır ve char sınıfı direk olarak unicode olarak tanımlandığından herhangibir hataya sebep vermeden daha iyi bir veri akışı sağlar. Bu sınıfın alt sınıfları olarak tanımlanan ve yazma işleminde kullanılan sınıfların kullanılmasına bir göz atalım.

Genel olarak Writer sınıfından türetilen ve en çok kullanılan Yazma kanalı **PrintWriter**, **BufferedWriter**, **FileWriter** veya **OutputStreamWriter** sınıflarının beraber kullanılmasıyla gerçekleştirilebilir. Bu sınıfların birlikte kullanılalarını şu deyimlerle örnekleyebiliriz :

dosya "a.txt" e yazdırma için :

```
PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt")));
```

tanımı kullanılabilir.

PrintWriter sınıfının tanımı :

```
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 baslamindeksi,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 sınıfının mevcut metodlarının işlevleri şunlardır :

checkError()	Tüm veriyi gönderir ve hata kontrolü yapar.
close()	veri akış kanalını kapatır.
flush()	veri akış kanalındaki tüm veriyi gönderir (ve bufferi yeni veri için boşaltır).
print (boolean)	boolean değişken yazar.
print (char)	character değişken yazar.
print (char[])	boyutlu character değişkenleri yazar.
print (double)	double yazar.
print (float)	float yazar.
print (int)	integer yazar.
print (long)	long yazar.
print (Object)	object sınıfının tanımladığı çıktıyı (object türüne göre değişebilir) yazar.
print(String)	String yazar
println ()	satır sonu yapar alt satırı geçer.
println (boolean)	boolean yazar ve satır sonu yapar alt satırı geçer.
println (char)	character yazar ve satır sonu yapar alt satırı geçer.
println (char[])	boyutlu character değişkenleri yazar ve satır sonu yapar alt satırı geçer.
println (double)	double yazar ve satır sonu yapar alt satırı geçer.
println (float)	float yazar ve satır sonu yapar alt satırı geçer.
println (int)	Print an integer, and then finish the line.
println (long)	long yazar ve satır sonu yapar alt satırı geçer.
println (Object)	object sınıfının tanımladığı çıktıyı (object türüne göre değişebilir) yazar ve satır sonu yapar alt satırı geçer.
println(String)	String yazar ve satır sonu yapar alt satırı geçer.
setError()	bir hata olduğunu gösterir
write (char[])	boyutlu character değişkenlerini yazar
write (char[], int, int)	boyutlu character değişkenlerini birinci integer(tamsayı) boyutundan ikinci integer boyutuna kadar yazar.
write (int)	tek bir character yazar.
write (String)	string yazar
write(String, int, int)	string değişkenini birinci integer(tamsayı) boyutundan ikinci integer boyutuna kadar yazar.

BufferedWriter sınıfının tanımı :

```

public class BufferedWriter extends Writer
{
    public BufferedWriter(Writer ciktig);
    public BufferedWriter(Writer ciktig,int boyut);
    public void write(int c) throws IOException;
    public void write(char c[],int baslamanindeksi,int uzunluk) throws IOException;
    public void write(String s, int baslamanindeksi,int uzunluk) throws IOException;
    public void newLine() throws IOException;
    public void flush() throws IOException;
    public void close() throws IOException;
}

```

şeklindedir. Yazma verimini artırmak için akışa ilave edilmektedir.

FileWriter sınıfı ise **File** sınıfı ile bağlantımızı kuran (dosyayı tanımlayan sınıfımızdır.). **Writer** sınıfının altında yer alan **OutputStreamWriter** Sınıfının alt sınıfıdır. tanımı :

```
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 sınıfının tanımı :

```
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 başlamaindeksi,int uzunluk) throws IOException;
    public void write(String str,int başlamaindeksi,int uzunluk) throws IOException;
    public void flush() throws IOException;
    public void close() throws IOException;
}
```

Bu sınıfları kullanarak bir yazma kanalı açmak istersek :

```
PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("a.txt")));
cfout.println("hello");
```

Komutunu kullanabiliriz. Bu komut

hello string değişkenini a.txt dosyasına yazdırır.

Dosya veya ekrandan okumak için ise paralel olarak **BufferedReader**, **FileReader** ve **InputStreamReader** sınıflarını kullanabiliriz. BufferedReader sınıfının tanımı :

```
public class BufferedReader extends Reader
{
    public BufferedReader(Reader giriş,int boyut);
    public BufferedReader(Reader giriş);
    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 sınıflarını birarada kullanarak bir okuma kanalı açabiliriz. örneğin a.txt dosyasından 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 programı

```

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 çıktısı :

```

s1 = Merhaba
s2 = isminiz nedir
s3 = sizinle tanistigima memnun oldum

```

dosya a.txt :

```

Merhaba
isminiz nedir
sizinle tanistigima memnun oldum

```

Şeklinde olacaktır. Programımız a.txt dosyasını oluşturup, üç string değerini yazdıktan sonra okuma kanalı açarak bu veriyi okudu.

Benzer bir okuma sınıfı olarak şu ana kadar kullandığımız Text sınıfını da verebiliriz. Aslında Text sınıfının içeriğine baktığımızda yukarıda verilen sınımların organize bir şekilde kullanılmasından ibaret olduğunu rahatlıkla görebiliriz. Text sınıfının listesi :

Program 10.6 Text.java programı

```

//=====
// 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 command 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 sınıfını kullanarak sequential bilgiyi dosyadan okuyan YazOku1.java programı alta verilmiştir.

Program 10.7 YazOku1.java programı

```

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 programın çıktısı :

```

s1 = Merhaba
s2 = isminiz
s3 = nedir
s4 = Hosgeldiniz

```

şeklindedir. Burada görüldüğü gibi her kelime ayrı bir string olarak algılanmıştır. Bu yapıyı gerçekleştiren Text sınıfı içerisinde kullanılan StringTokenizer sınıfıdır.

String item = T.nextToken();

ifadesiyle okunmakta olan Stringi cümlelere böler.

Ardışık dosya okuma örneği olarak Atom.java, AtomTest.java, Atom.txt, gurubunu verebiliriz. Atom.java Atomların bazı fiziksel özelliklerini tanımlayan bir programdır. Program atomlarla ilgili veriyi Atom.txt dosyasından okumaktadır.

Program 10.8 Atom.java programı

```

=====
// 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
// -----
// 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 programı

```

//=====
// 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 dosyası

1 Hydrogen	H	1.00794	16 Sulphur	S	32.066	31 Gallium	Ga	69.723
2 Helium	He	4.002602	17 Chlorine	Cl	35.4527	32 Germanium	Ge	72.61
3 Lithium	Li	6.941	18 Argon	Ar	39.938	33 Arsenic	As	74.92159
4 Berylium	Be	9.012182	19 Potassium	K	39.0983	34 Selenium	Si	78.96
5 Boron	B	10.811	20 Calcium	Ca	40.078	35 Bromine	Br	79.904
6 Carbon	C	12.011	21 Scandium	Sc	44.95591	36 Krypton	Kr	83.8
7 Nitrogen	N	14.00674	22 Titanium	Ti	47.88	37 Rubidium	Ru	85.4678
8 Oxygen	O	15.9994	23 Vanadium	V	50.9415	38 Strontium	Sr	87.62
9 Florine	F	18.9984032	24 Chromium	Cr	51.9961	39 Yttrium	Yt	88.90585
10 Neon	Ne	20.1797	25 Manganese	Mn	54.93805	40 Zirconium	Zr	91.224
11 Sodium	Na	22.989768	26 Iron	Fe	55.587	41 Niobium	Nb	92.90638
12 Magnesium	Mg	24.305	27 Cobalt	Co	58.9332	42 Molybdenium	Mo	95.94
13 Aliminium	Al	26.981539	28 Nickel	Ni	58.69	43 Technetium	Tc	98
14 Silicon	Ag	28.0855	29 Copper	Cu	63.546	44 Ruthenium	Sm	101.007
15 Phosphourus	P	30.973762	30 Zink	Zn	65.39	45 Rhodium	Rh	102.9055

46 Palladium Pd 106.42	66 Dysprosium Dy 162.5	86 Radon Rn 222
47 Silver Ag 107.8682	67 Holmium Ho 164.93032	87 Francium Fr 223
48 Cadmium Cd 112.411	68 Erbium Er 167.26	88 Radium Ra 226
49 Indium In 114.82	69 Thulium Tm 168.93421	89 Actinium Ac 227
50 Tin Sn 118.71	70 Ytterbium Yb 173.04	90 Thorium Th 232.0381
51 Anthimony Sb 121.75	71 Lutetium Lu 174.967	91 Protactinium Pa 231
52 Tellerium Te 127.6	72 Hafnium Hf 178.49	92 Uranium U 238.0289
53 Iodine I 126.90447	73 Tantalum Ta 180.9479	93 Neptunium Np 237
54 Xenon Xe 131.29	74 Tungsten W 183.85	94 Plutonium Pu 244
55 Cesium Cs 132.90543	75 Rhenium Re 186.207	95 Americium Am 243
56 Barium Ba 137.327	76 Osmium Os 190.2	96 Curium Cm 247
57 Lanthanum La 138.9055	77 Iridium Ir 192.22	97 Berkelium Bk 247
58 Cerium Ce 140.115	78 Platinum Pt 195.08	98 Californium Cf 251
59 Praseodymium Pr 140.90765	79 Gold Au 196.96654	99 Einsteinium Es 252
60 Neodymium Nd 144.24	80 Mercury Hg 200.59	100 Fermium Fm 257
61 Promethium Pm 145	81 Thallium Tl 204.3833	101 Mendelevium Md 258
62 Samarium Sc 150.36	82 Lead Pb 207.2	102 Nobelium No 259
63 Europium Eu 151.965	83 Bismuth Bi 208.98037	103 Lawrencium Lr 260
64 Gadolinium Gd 157.25	84 Polonium Po 209	
65 Terbium Tb 158.92534	85 Astatine At 210	

Program AtomText.java çıktısı :

N2
O8

Burada hemen şunu kaydedelim. Programdan da anlaşılacağı gibi Atom.java java'yı öğretme amacıyla hazırlanmamıştır. Gazların termodinamik özelliklerini hesaplayan bir program gurubunun parçası olarak hazırlanmıştır.

10.4 RASLANTISAL ULAŞIM DOSYASI OKU (RANDOM ACCESS FILE)

Raslantısal Ulaşım Dosyasının (Random Access File) Ardışık Dosya sisteminden temel farkı, Teyp kaydıyla CD kaydı arasındaki fark gibidir. Ardışık dosyalar Teyp kaydı şeklinde birbiri ardı sıra gelen bilgilerden oluşur. Raslantısal Ulaşım sisteminde ise CD gibi herhangi bir kayıtına kayıt numarası kullanılarak doğrudan ulaşmak mümkündür. Raslantısal Ulaşım dosyası işlemleri RandomAccessFile sınıfı üzerinden yapılır. Bu sınıfın tanımı :

```
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 tanımlanan metodların görevlerine kısaca bir bakacak olursak :

getFD() : dosya (File) tanımını verir

getFilePointer() : Dosyanın (File) o anda hangi dosya referansını gösterdiğini belirtir.

length() : Dosyadaki toplam referans sayısını verir.

read() : Byte (char değişkeni karşılığı) olarak dosyadan bilgiyi okur (bir byte).

read(byte[]) : Byte (char değişkeni karşılığı) olarak dosyadan bilgiyi okur (byte[] değişkeninin boyu kadar) ve byte değişkenine aktarır.

read(byte[], int baslangic_indisi, int indis_boyutu) : Byte (char değişkeni karşılığı) olarak dosyadan bilgiyi okur (byte[] değişkeninin boyu kadar, başlangıç indisinden başlayarak, indis_boyutu uzunlığında) ve byte değişkenine aktarır.

readBoolean() : boolean değişken okur

readByte() : dosyadan integer tipi byte değişken (işaretli 8-bit) okur.

readChar() : Dosyadan Unicode karakter (character) okur.

readDouble() : Dosyadan double değişkeni okur.

readFloat() : Dosyadan float değişkeni okur.

readFully(byte[] b) : dosyadan byte olarak okur

readFully(byte[] b, int baslangic_indisi, int toplam_boy) : dosyadan bte olarak sadece byte boyutlu değişkeninin başlangic_indisi'nden başlamak üzere toplam_boy kadar kısmını okur.

readInt() : Dosyadan int değişkeni okur.

readLine() : Dosyadan “\n” = yeni satır işaretini veya “\r” satırbaşı işaretini veya herikisini birden arka arkaya görene kadar yazılan her şeyi bütün bir satır olarak okur.

readLong() : Dosyadan Long (Long integer) değişkeni okur.

readShort() : Dosyadan Short (Short integer) değişkeni okur.

readUnsignedByte() : İşaretsiz Byte değeri okur.

readUnsignedShort() : İşaretsiz 16 bitlik Short integer değeri okur.

readUTF() : UTF stringi okur

seek(long) : indisini herhangi bir bilgi gurubu indisine ayarlar.

skipBytes(int n) : n bit input değerini okumadan atlar

Aşağıdaki tanımlar read tanımlarına parel olan write terimleridir.

write(byte[]) : byte boyutlu değişkenini yazar.

write(byte[],int baslangic_indisi, int toplam_boy) : dosyaya byte olarak sadece byte boyutlu değişkeninin başlangic_indisi'nden başlamak üzere toplam_boy kadar kısmını yazar.

write(int) : dosyaya byte boyutlu değişkenini yazar

writeBoolean(boolean) : dosyaya boolean boyutlu değişkenini yazar

writeByte(int) : dosyaya int girdisini byte olarak yazar

writeBytes(String) : dosyaya String girdisini byte boyutlu değişkeni olarak yazar

writeChar(int) : dosyaya int girdisini Char olarak yazar

writeChars(String) : dosyaya string girdisini char boyutlu değişkeni 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 sınıfının tanımından da görüldüğü gibi b sınıf DataOutput ve DataInput sınıflarını implement eder. DataOutput ve DataInput sınıflarını tanımi :

```
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 ulaşım olanakları yaratır. Raslantısal Ulaşım dosyası örneği olarak aşağıdaki paketi veriyoruz. Kayıt sınıfı tek bir kayıtın yazılıp okunabilmesi için gerekli olan bilgileri içeriyor. Kayıtların hesap numarası, isim, soyisim ve hesaptaki paradan oluştuğunu kabul ediyoruz. Bütün bu kayıtları bir arada yapmak için oku ve yaz metotları bu sınıfın içindir tanımlanmıştır. Raslantısal ulaşım kanalının dosyaya açılması için :

```
try{
    girdi=new RandomAccessFile("musteri1.dat","rw");
} catch(IOException e)
{
    System.err.println("Dosya acma hatası\n"+e.toString());
    System.exit(1);
}
```

yapısı kullanılır. Buradaki “rw” yapısı dosyaya hem yazı yazılıp hem okunabileceğini belirtir. “r” sadece okumak için “w” sadece yazmak için kullanılır.

RandomAccessFile içinde yer alan

close() metodu RandomAccessFile (Raslantısal Ulaşım Dosyası) kanalını ve ilgili tüm sınıfları kapatır. örnək olaraq :

```
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.

Şimdi de örnek programın listelerini verelim:

Program 10.10 Kayit.java raslantısal ulaşım dosyası kayıt programı

```
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 = Raslantısal Ulasım dosyası
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 programı

```

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 programı

```

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("Tesanufi 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 programı

```

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");

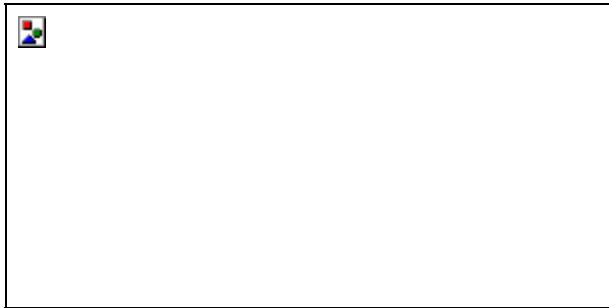
```

```

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("Dosyayı 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



Şekil 10.3 RaslantisalUlasimDosyasiOku.java Frame çıktısı

Yukardaki programın ilginç bir yönü de **Frame** sınıfını kullanmış olmasıdır. Frame ve swing eşiti JFrame konsol ortamında kullanılabilen grafik ortamı programlamasıdır. Temel olarak applet yapılarında kullanılan her yerde Frame de kullanılabilir. Frame **main** programdan başlatılır. Html dosyasından başlatılmaz.

10.5 DOSYA SIKIŞTIRILMASI (GZIP,GUNZIP,ZIP,UNZIP)

java.util.zip paketi veri sıkıştırma yarayan dosyaları içerir. Günümüz bilgisayar kullanımında GZIP ve ZIP adı verilen bu yapılar ZLIB sıkıştırma algoritmasında tanımlanmıştır. Bu algoritım RFC 1950, RFC 1951 ve RFC 1952 dökümanlarında tanımlanmıştır. Bu dökümanlara <http://www.faqs.org/rfcs> adresinden ulaşılabilir. Kullanma açısından bu dökümanlara ihtiyacınız yoktur. Gerekli programlar java zip paketinde bulunmaktadır.

GZIP tek bir dosyayı sıkıştırarak isim.gz ismiyle sıkıştırılmış dosya oluşturur. Bunun için GZIPOutputStream dosyasına bir FileOutputStream açmamız kafidir. Program 10.4 den de görüleceği gibi

```
GZIPOutputStream zipout;
try {
    FileOutputStream out = new FileOutputStream(zipname);
    zipout = new GZIPOutputStream(out);
}
catch (IOException e) {
    JOptionPane.showInputDialog(null,"Dosya oluşturma hatası : " + zipname + ".");
    return;
}
deyimi dosyanın gzip olarak yazılması için gerekli kanalı oluşturur.
```

Program 10.14 Gzip.java dosya sıkıştırma programı

```
//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;
        }
        // çıktı doyası oluştur.
        String zipname = args[0] + ".gz";
        GZIPOutputStream zipout;
        try {
            FileOutputStream out = new FileOutputStream(zipname);
            zipout = new GZIPOutputStream(out);
        }
        catch (IOException e) {
```

```

JOptionPane.showInputDialog(null,"Dosya oluşturma hatası : " + zipname + ".");
return;
}
byte[] buffer = new byte[sChunk];
// dosyayı sıkıştır
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 sıkıştırma hatası : " + args[0] + ".");
}
try { zipout.close( ); }
catch (IOException e) {}
}
}

```

GZIP işleminin tersini yapmak içinse parellel olarak :

```

GZIPInputStream zipin;
try {
    FileInputStream in = new FileInputStream(zipname);
    zipin = new GZIPInputStream(in);
}
catch (IOException e) {
    JOptionPane.showInputDialog(null,"Dosya açma hatası : " + zipname + ".");
    return;
}

```

yapısı kullanılır.

Programların çalışması 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 sıkıştırılmış dosyayı açma programı

```

//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 dosyasını 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 hatası : " + zipname + ".");
    return;
}
byte[] buffer = new byte[sChunk];
// dosyayı 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 sıkıştırılması açılma hatası : " + args[0] + ".");
}
try { zipin.close(); }
catch (IOException e) {}
}
}

```

ZIP ve UNZIP biraz daha kompleks bir yapıya sahiptir. Çünkü bu prosesde direktory içindeki dosyaların açılma işlemi söz konusudur.

Prosesi izah etmek için önce ZIP işlemi yapan bir programa göz atalım :

Program 10.16 zip.java dosya sıkıştırma programı

```

// 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="";
OutputStream out;
ZipOutputStream zout;
String zipname, source;
String s="KULLANIM: \n"+
"java zip zipdosyası dosya(veyadirectory)ismi1 dosya(veyadirectory)ismi1... \n"+
" zip dosyası yarat ve dosya(veya directorylerdeki dosyaları) sıkıştır "+
" alt direktoryler atlanmaktadır tekrar tanımla";
// anahtarlı kontrol et
if(args.length<1 )
{
String s1="zipdosyası isim.zip tanımlanmadı lütfen argümanları giriniz\n"+s;
JOptionPane.showMessageDialog(null,s1);
System.exit(0);
return;
} // girdi dosyasını incele
else if(args.length<2)
{
String s1="sıkıştırılacak dosya(veya direktory) isimleri tanımlanmadı\n"+
" lütfen argümanları 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 kanalını tanımla
try {
out = new FileOutputStream(zipname);
zout=new ZipOutputStream(out);
}
catch (IOException e) {
JOptionPane.showMessageDialog(null,"Dosya oluşturma hatası : " + zipname + ".");
return;
}

```

```

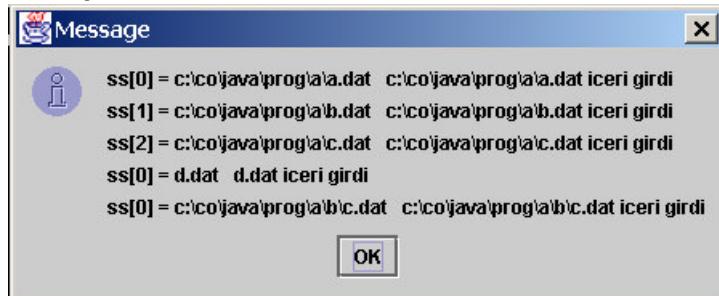
}
int k=1;
while(k<args.length)
{
try
{
String ss[];
ss=dosyalistesi(args[k]);
int j=0;
byte[] buffer=new byte[sChunk];
while(j<ss.length)
{
ZipEntry giris=new ZipEntry(ss[j]);
zout.putNextEntry(giris);
try {
    FileInputStream in=new FileInputStream(ss[j]);
    int length;
    while((length=in.read(buffer,0,sChunk))!=-1)
        zout.write(buffer,0,length);
    in.close();
} //try sonu
catch (IOException e)
{
    JOptionPane.showMessageDialog(null,"Dosya sıkıştırma hatası : " +ss[j] );
} //catch sonu
str+="ss["+j+"] = "+ss[j]+" "+giris+" içeri 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



Şekil 10.3 zip.java programını çıktısı

şimdi de ZIP yapılmış bir dosyayı açan Unzip.java programına bir göz atalım :

Program 10.16 Unzip.java dosya sıkıştırılmış dosyayı açma programı

```
//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);
        }
        // girdi dosyasını 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 dosyası : "+zipname+" içindeki sıkıştırılmış \n";
        try {
            FileInputStream in = new FileInputStream(zipname);
            ZipInputStream zin=new ZipInputStream(in);
            byte[] buffer = new byte[sChunk];
            // dosyayı 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 sıkıştırma hatası : " + args[0] + ".");
                        System.exit(0);
                    }
                    s+=" " +z.getName()+" \n";
                }while(z!=null);
            } catch(NullPointerException npe)
            {
                s+="dosyaları açıldı";JOptionPane.showMessageDialog(null,s);
                System.exit(0);
            }
            try { zin.close(); } catch (IOException e) {}
        }
        catch(IOException e)
        {
        }
```

```

        JOptionPane.showMessageDialog(null,"Dosya açılma hatası : " + zipname);
        System.exit(0);
    }
}

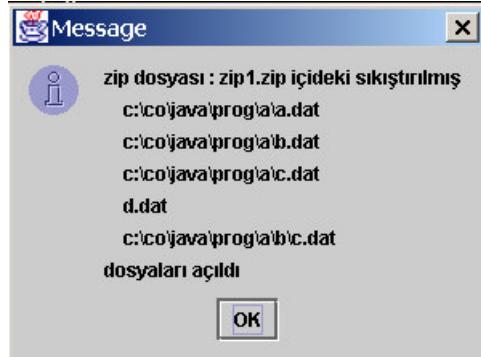
```

programı

java unzip zip1.zip

komutu kullanarak çalıştırırsak :

10005.JPG



Şekil 10.4 unzip.java programını çıktısı

Çıktısını elde ederiz.

Burada zip dosyasına ulaşmak için :

```

FileInputStream in = new FileInputStream(zipname);
ZipInputStream zin=new ZipInputStream(in);

```

Deyimini kullandık. Zipli dosayı 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 kullandık. Burada temel olarak iki yeni sınıf kullanıldı. Bunlardan birincisi ZipInputStream, diğer ise ZipEntry sınıfları idi. Şimdi bu iki sınıfın tanımlarına göz atalım :

```

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 tanımı 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)
}

```

şeklindedir. Ayrıca zip dosyasının içindeki elemanlar ZipEntry sınıfında tanımlanmaktadır. ZipEntry sınıfı :

```

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()
}

```

şeklinde tanımlanmıştır. Burada şu ana kadar verdığımız sınıf tanımları ile ilgili şunu söyleyelim. Tanımlamalar sadece sınıf ve değişken adlarını vermektedir, gerçek programı açtığınızda göreceğiniz koda benzemezler sadece metod isimlerini taşırlar.

10.6 ALIŞTIRMALAR

1. H9O1.java programını inceleyiniz. Bu programda sayılar dosyadan okunup ortalamaları hesaplanmaktadır.

Program 10.14 H9O1.java programı

```

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);
toplum+=i[j];
++j;
}
catch (EOFException e)
{
b.close();
EOF=true;
}
} //while

System.out.println("Dosyanın içindeki sayıların adedi: "+j);
System.out.println("Ortalama: "+(double)toplum/20);

} //main

} //class

```

2. Karsilastir.java verilen iki dosyayı karşılaştırmaktadır. Dosya isimleri program kullanımı sırasında verilecektir. Örnek :

java Karsilastir a.dat a1.dat

Program 10.15 Karsilastir.java programı

```

import java.io.*;
public class Karsilastir {

//_____
// RandomAccessFile örneği
// program main()
//_____

// dışarıdan iki tane dosya ismi giriniz
// dosyaları açar ve içindeleri yükler
// dosyaları değerlendirir (karşılaştırır)
// dosyaları kapatır
//_____
public static void main (String args[]) {
    RandomAccessFile fh1 = null;
    RandomAccessFile fh2 = null;

    int bufsize;      // en küçük dosyanın byutu
    long filesize1 = -1;
    long filesize2 = -1;
    byte buffer1[];
    byte buffer2[];

    // dışarıdan okunan veriyi kontrol et

    if (args.length == 0 || args[0].equals("?") ) {
        System.err.println ("Random Acess dosya Karşılaştırma : ");
        System.err.println ("-----");
        System.err.println ("KULLANIM : java Karsilastir <dosya1> <dosya2> | ?");
        System.err.println ();
        System.exit(0);
    }
}

```

```

// birinci dosyayı 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 dosyayı 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 boyutları aynı değil! ");
    System.out.println ("Dosya 1 : " + args[0] + "boyutu " + filesize1 + " bytes");
    System.out.println ("Dosya 2 : " + args[1] + "boyutu " + filesize2 + " bytes");
}

// iki dosyanında içeriğini alabilecek bir yer ayır

bufsize = (int) Math.min(filesize1, filesize2);
buffer1 = new byte [bufsize];
buffer2 = new byte [bufsize];
// dosyaları toptan oku
try {
    fh1.readFully (buffer1, 0, bufsize);
    fh2.readFully (buffer2, 0, bufsize);

    // şimdi gelelim asıl işimize...

    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: dosyaları icelerken bir hata oluştu");
    System.err.println (ioErr.toString());
} finally {
    try {
        fh1.close();
        fh2.close();
    } catch (IOException ignored) {}
}
}

```

3. File sınıfı uygulaması : dir.java programı istenilen directory ve alt directory dosyalarını sırayla listeler.

Program 10.16 dir.java programı

```
//-----
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'deki dosyaların listesi

        indentLevel++;           // sayılıyor...

        // bu directory'deki dosyaların listesini hazırla
        files = path.listFiles();

        // Dosya isimlerini sıraya 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
C:\Drive\hpscanner\disk1\@631f.adf
C:\Drive\hpscanner\disk1\_inst32i.exe
C:\Drive\hpscanner\disk1\_isdel.exe
```

4. zip.java sınıfı içerisindeki `public static String[] dosyalistesi(String p)` metodu alt direktoryleri okumadan atlamaktadır. Alt direktoryleri de okuyup zipteyecek yeni bir zip1.java programı yazınız.

5. Bu problemde gerçek bir bilgisayar programlama örneğine göz atacağız. Program 10.8 de Atom.java programına bakmıştık, bu problemde Atom.java programı kullanılarak yazılmış olan Gas.java programına yer vereceğiz. Bu program ideal gazların termodinamik özelliklerin hesaplamak amacıyla yazılmıştır. Program ideal

gazların verilerini Gas.txt dosyasından akumaktadır. Gaz isimleri kurucu metod tarafından aranan gaz isimlerine uyum sağladığında program geriye kalan veriyi okumaktadır. Gazlar atomlardan yapıldığından veride okunan Atom isimlerine göre Atom özellikleri de bu sınıfın kurucu metodunda Atom.txt dosyasından okunmakta ve değerlendirilmektedir. Program çıktıtı almak için GasPropertySWF.java programı da ayrıca verilmiştir. Burada verilen kod özellikle bu kitap için yazılmamıştır. Uluslararası projelerde kullanıldığından açıklamaları ingilizcedir.

Program 10.16 Gas.java programı

```
//=====
// Thermodynamic Package in java
// Class Gas Properties of perfect gases
// Dr. Turhan Coban
// TUBİTAK Marmara Research Center
// Energy Systems and Environmental Research Institute
// email : 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() : pressure 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 Timothy 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;
// definitions 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;
aN=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,xdii,tli,thi;
int n_vis,n_k;
double xvsi,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);
                xdii=Text.readDouble(fin);
                tli=Text.readDouble(fin);
                thi=Text.readDouble(fin);
            }
            n_vis=Text.readInt(fin);
            for(i=0;i<n_vis;i++)
            {
                xvsi=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 or 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 or 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 aldığı bilgi dosyası Gas.txt (sadece iki örnek gaz ch4 ve c2h6 için veri listelenmiştir.) :
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 1000.0
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
-.317434898401931E+01
.455271007474814E+00
-.273399388607163E-03
.895312831114117E-07
4
.399296677905716E+01
.551053536678452E-01
.172999763063988E-03
-.657213363465470E-07
c2h6
2
C 2
H 6
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 çıktısının bir parçasıdır, çıktıyı tablo formunda elde etmemiz amacıyla kullanılmıştır.

```

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 enthalpy", "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
// =====

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, TUBİTAK 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();
}

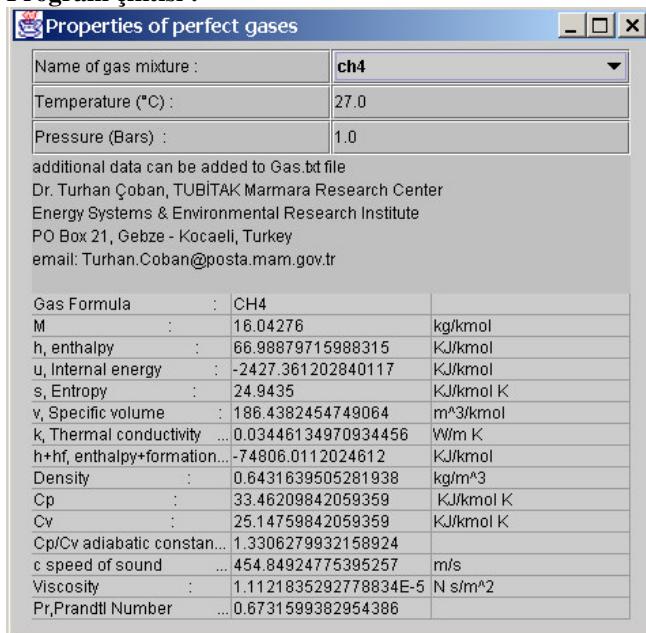
```

```

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 çıktısı :



6. Bu problemde bir önceki problemde verilen uygulamanın bir kademe daha ilerine gideceğiz. Burada Gazları karıştırıp bir gaz karışımı elde edecek ve bu gaz karışımının termodinamik özelliklerine bakacağız.
 Programımızın ismi Gmix.java, gazları karıştırdığımız veri dosyası Gmix.txt ve çıktı dosyalarımız GmixModel.java ve GmixPropertySWF.java

Program 10.19 Gmix.java

```

// =====
// File Name  : Gmix.java
// Author     : Dr. Turhan Coban
// TUBİTAK Marmara Research Center
// Energy Systems and Environmental Research Institute
// email : 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()   : pressure 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 :
//=====definitions 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))
        {
            Gas newGas[];
            newGas=new Gas[ngas];
            M=0;
            N=0;
            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);

```

```

        vratio=Math.sqrt(c1);
    }
else
    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 or 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 dosyası Gmix.txt (kısmi 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
// =====
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 Coban, 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";
    //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();
}

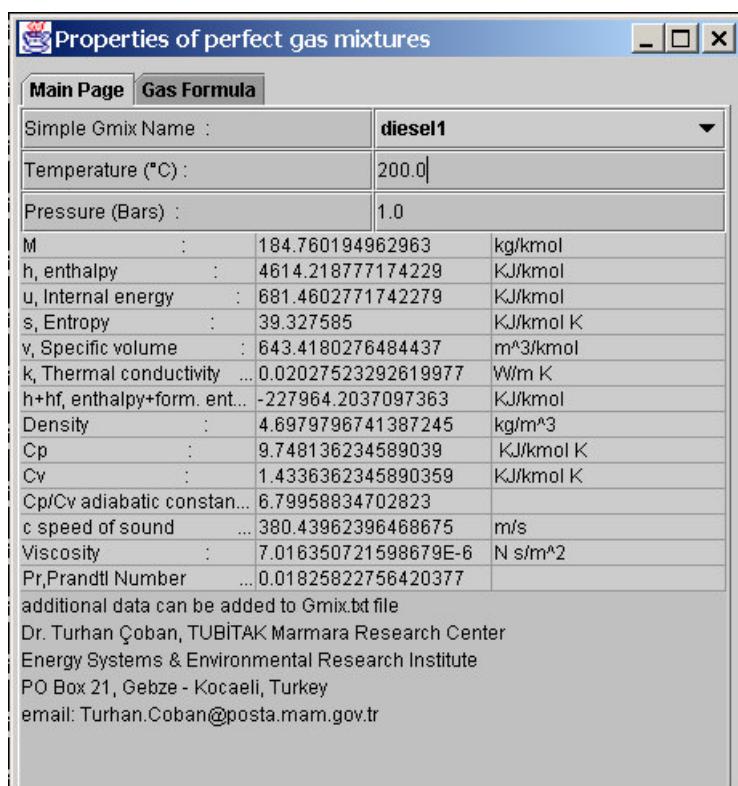
```

```

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 çıktısı :



Properties of perfect gas mixtures

Main Page Gas Formula

Gas mixture formula :

```
=====
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
```

Gas mixture composition :

```
=====
C13.244544544544542
H25.47767767767768
```

Burada hemen bir noktaya dikkat çekelim. Gerçek programlamalarda asıl veriyi işleyen sınıflar ve insan grafik arayüzü programları genelde ayrı tutulur. Bu problem seti aynı zamanda sınıfların birbiri üzerine inşasının da güzel bir örneğini teşkil etmektedir.

6. Bu problemden bir önceki problemden verilen uygulamanın bir kademe daha ilerine gideceğiz. Burada Gaz karışımlarını bir borudan akıtacağız. Borunun basınç düşümü ısı transferi gibi çeşitli özelliklerini bu program yardımıyla değerlendireceğiz. Boruların özellikleri Pipe.txt dosyasında tanımlanmıştır. Gaz karışımı için Boru basınç düşümü ve ısı transferini hesaplayan programımızın ismi Pipe.java'dır.

Program 10.22 Pipe.java

```
=====
// Thermodynamic package in java
// File Name : pipe.java
// Author : Turhan Coban
// TUBİTAK Marmara Research Center
// Energy Systems and Environmental Research Institute
// email : 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 roughness 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 internal 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 dosyası Pipe.txt (sadece birkaç örnek durum listelenmiştir.)

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 dosyası boru ismi, boru şekli(circular, rectangular..), gazkarışımının adı(Gmix.txt dosyasında tanımlanmış olmalıdır), boru çapı veya boru en ve boyu ve akış debisi(kg/s) bilgilerini içerir. Bu programın kullanıldığı çıktı veren bir insan arayüzü programına göz atalım :

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.fT(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),250,310);
    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());
}

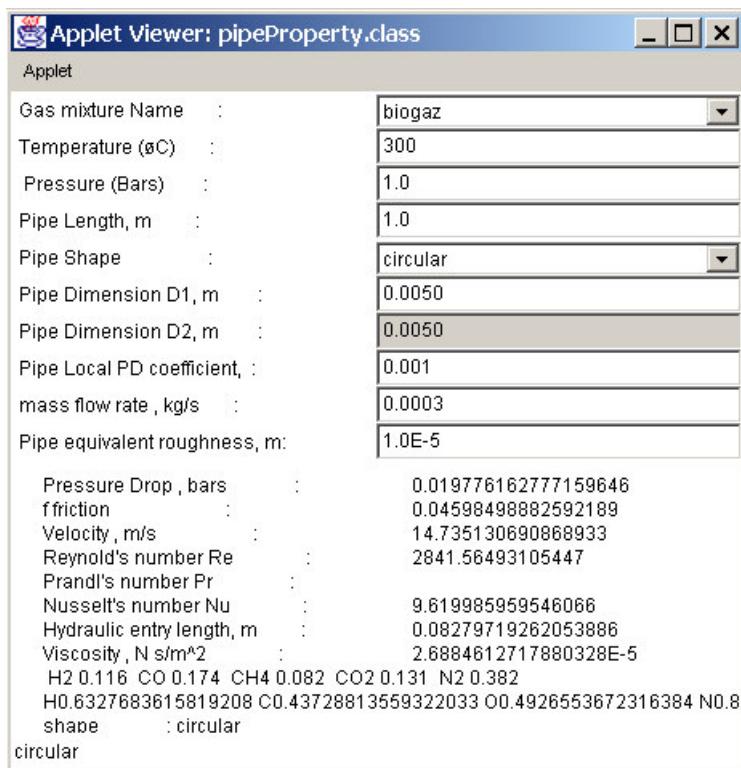
```

```

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 çıktıtı:



7. Thermodinamik özellikler serisinden bir adım daha ileri gidip gazları kimyasal reaksiyonlara sokabiliriz. Bu sınıfımız Reaction.java ismini taşiyor. Gazların termodinamik reaksiyona girdiklerinde enerji dengelerini, adyabatik alev sıcaklıklarını ve reaksiyon sabitlerini (k) hesaplama kapasitesinde bir program. Program veridosyası Reaction.txt adında. Bu dosyada reaksiyonun ismi, toplam gaz sayısı, bu gazın reaksiyona giren ve reaksiyondan çıkan moleküllerinin sayısı liste olarak veriliyor. Reaksiyon kütle denklik hesabı program tarafından yapılmamaktadır. Doğru molekül sayılarının verilmesi kullanıcıya bırakılmıştır.

Bu dosyadan birkaç örnek veri setine göz atalım:

```

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

```

Programımızın 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          //

```

```

//=====================================================================
// 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;
    }
}
```

programımızı 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 çıktısı :

```
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
```

8. Termodinamik serisinden bir diğer program. Bu program gerçek gazların termodinamik özelliklerini hesaplıyor. Programın ismi LeeKesler.java, veri dosyası RealGas.txt
Program Listesi :

```
=====
// Thermodynamic Package in java
// Class Lee Kesler Properties of real gases
// Dr. Turhan Coban
// TUBİTAK Marmara Research Center
// Energy Systems and Environmental Research Institute
// email : 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, structural information
// natom     : number of atoms
// M         : Molecular weight as calculated from atomic structure
//             through class atom and atomList
// MolWt    : Molecular Weight 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 compressibility factor PcVc/RTc
// Omega     : Pitzer's acentric 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. Poling
// 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)
+ 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)
+ 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)/(1-x));
break;
}
case 2:
{
cPvp=VPA-VPB/T+VPC*Math.log(T)+VPD
break;
}
case 3:
{
break;
}
return cPvp;
}
*/
}

```

Programın veri dosyasından bazı gazların 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

Sanırım son birkaç örnek gerçek programlama ortamında veri dosyalarının kullanılması hakkında daha iyi bir fikir vermiştir. Buradaki tüm örnekler sıralı (sequential) tip veri dosyası işlemektedir. Aynı işlemler veri tabanı kullanılarak da yapılabilirdi.

BÖLÜM 11 GELİŞMİŞ JAVA BİLGİ İŞLEME YAPILARI

11.0 GİRİŞ

Bu bölümde çeşitli ileri bilgi işleme ve boyut oluşturma kavramlarına göz atacağız.

11.1 STRINGTOKENIZER SINIFI

Biz bir cümle okuduğumuzda beynimiz cümleyi otomatik olarak kelimelerine ayırır. Java Dilinde bu görevi StringTokenizer sınıfı yerine getirir. StringTokenizer sınıfının yapısı :

```
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 sınıfı açıklamak için önce küçük bir program verelim :

Program 11.1 : StringTokenizerTest.java programı

```
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 sayısı : "+t.countTokens());  
        int i=0;  
        while(t.hasMoreTokens())  
        {  
            System.out.println("kelime indeksi : "+(i++)+" kelime : "+t.nextToken());  
        }  
    }  
}
```

Programdan da görüleceği gibi StringTokenizer sınıfı

StringTokenizer t=new StringTokenizer(s);

deyiğiyle tanımlanmıştır. Ayrıca stringin içindeki cümle sayısı **t.countTokens()**; metoduyla çağrılmış, **t.hasMoreTokens()** sınıfı ise daha fazla kelime kalıp kalmadığının boolean kontrolu metodu olarak çağrılmıştır. **t.nextToken()** metodu o andaki kelimeyi dışarıya (String cinsinden) aktarırken indeksi de (currentPosition) bir sonraki kelimeye kaydırır.

Program çıktısı :

```
cumle : Ali veli 49 elli
kelime sayısı : 4
kelime indeksi : 0 kelime : Ali
kelime indeksi : 1 kelime : veli
kelime indeksi : 2 kelime : 49
kelime indeksi : 3 kelime : elli
olacaktır.
```

StringTokenizer sınıfının kullanılmasıyla ilgili bir örnek daha verelim. Bu örnekte ninci dereceden bir polinomun kökleri Matrix.java kütüphanesi (detaylar için bakınız bölüm 12) kullanılarak çözülür. girdi alanında tüm polinom katsayıları bir alan içinde tanımlanmakta ve sonra bu bileşik alandan StringTokenizer sınıfı yardımı ile katsayılar tek tek çekilmektedir.

Program 11.2 kokN.java programı, n'inci dereceden polinomun köklerini hesaplar. Tek bir string olarak girilen katsayılar matrisinden katsayıları StringTokenizer sınıfını kullanarak çeker.

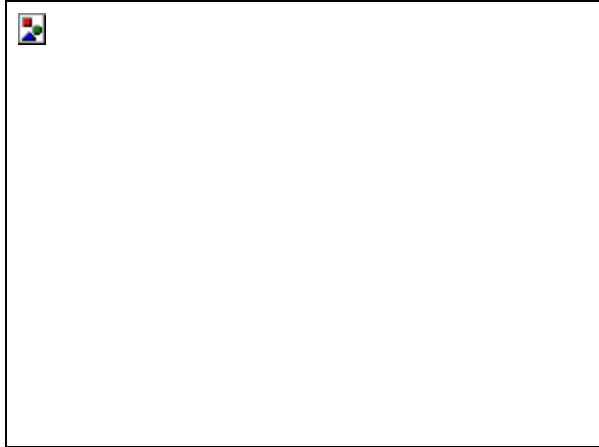
```
import java.util.*;
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
import Matrix;
public class kokN extends Applet implements ActionListener
{
    private Label prompt1,prompt2;
    private TextField input;
    TextArea t;
    Panel YaziPaneli;
    int n;
    String s;
    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 katsayılarını 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



Şekil 11.1 StringTokenizer sınıfını ve matrix sınıfını kullanarak ninci dereceden bir polinomun köklerini hesaplayan kokN.java programının applet çıktısı

Aynı programın swinf JFrame versiyonu :

Program 11.3 kokNSWF_2000.java programı, n'inci dereceden polinomun köklerini hesaplar. Tek bir string olarak girilen katsayılar matrisinden katsayıları StringTokenizer sınıfını 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();

```

```

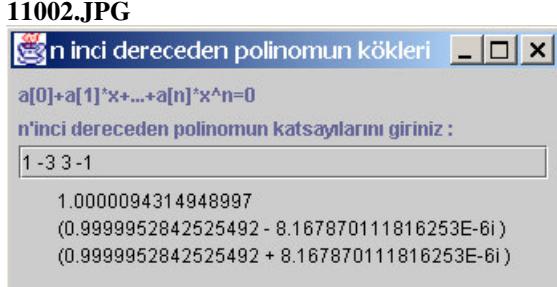
YaziPaneli.setFont(new Font("Serif",Font.BOLD,12));
YaziPaneli.setLayout( new GridLayout(3,1) );
t=new JTextArea();
t.setBackground(c.getBackground());
prompt1= new JLabel("a[0]+a[1]*x+...+a[n]*x^n=0");
prompt2= new JLabel("n'inci dereceden polinomun katsayılarını giriniz : ");
input = new JTextField(30);
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



Şekil 11.2 StringTokenizer sınıfını ve matrix sınıfını kullanarak ninci dereceden bir polinomun koklerini hesaplayan kokNSWF_2000.java programının JFrame çıktısı

StringTokenizer sınıfının alt değişkenleri :

nval : eğer seçilmiş olan token bir sayı ise nval bu sayının değerini içerir.

sval

eger seçilmiş olan token bir kelime(String) ise sval bu String değişkenini içerir

TT_EOF

girdi stringinin sonuna ulaşıldığını gösterir

TT_EOL

satır sonu okunduğunu bildirir

TT_NUMBER

bir sayı tokeni okunduğunu bildirir

TT_WORD

bir kelime(String) token'i okunduğunu bildirir..

ttype

nextToken metodu çağrıldıktan sonra bu değişken en son okunan token (kelimenin) türünü bildiri

StringTokenizer sınıfının metodları :

kurucu metot :

StreamTokenizer(Reader)

eolIsSignificant(boolean)

satır sonu kumutunun yeni kelime tanımlayıp tanımlamadığını belirler.

lineno()

satır sayısını verir.

lowerCaseMode(boolean)

kelimelerin otomatik olarak küçük harfe çevrilip çevrilmeyeceğini tanımlar

nextToken()

giriş Stringinin içindeki bir sonraki kelimeyi bulur.

ordinaryChar(int)

Tokenizer'in normalde kontrol için kullandığı coşluk,satır sonu gibi özel kontrol karakterlerinin normal karakter olarak okunmasını sağlar

parseNumbers()

Direk olarak bir sayı dizininin geldiğini haber verir

pushBack()

bir sonraki nextToken çağırılmasında kelimenin şu andaki kelime olarak kalmasını sağlar.

quoteChar(int)

eger " karecteri varsa bunun içindeki karakterleri bir String değişkeni olarak algılar.

slashSlashComments(boolean)

Tokenizer'in C++-tipi (//.....)comment deyimlerini comment olarak tanıyorum tanımlayacağını belirler.

slashStarComments(boolean)

Tokenizer'in C-tipi comment /*...*/ deyimlerini comment olarak tanıyıp tanımlayacağımı belirler.

toString()

o anda seçilmiş olan klimeyi (token) gönderir.

whitespaceChars(int, int)

birinciinteger <= c <= ikinciinteger arasındaki karakterleri boşluk olarak tanımlar

wordChars(int, int)

birinciinteger <= c <= ikinciinteger arasındaki karakterleri tek kelime olarak tanımlar

11.2 STRING BUFFER SINIFI

Yeri gelmişken **String** sınıfının kardeş sınıfı, **StringBuffer** sınıfından da bahsedelim. StringBuffer genel olarak String sınıfının ayındır, fakat genellikle dinamik olarak program içinde değeri sıkça değişen Stringler için kullanılır. String ise genellikle statik olarak verilen String değişkenleri içindir. StringBuffer sınıfına Append metotuya dinamik olarak ilave yapabiliriz.

StringBuffer sınıfının metod yapısı şöyledir :

```
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
}

```

Şimdi küçük bir test programında StringBuffer sınıfını kullanalım :

Program 11.4 TestString.java, giren String değişkenini 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();
    }
}

```

diğer bir StringBuffer metodu **insert** metotudur. Bu metot iinci elementten sonra verilen yeni parçayı ilave edecektir.

Program 11.5 javaKahvesi.java, StringBuffer, insert metodu örneği

```
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 programı çalıştırduğumızda
Java kahvesini sıcak iç!
mesajını verecektir.

11.3 VECTOR SINIFI

Daha önce boyutlu değişkenleri görmüştük. Boyutlu değişkenlerin boyutları bilgisayar belleğinde oluşturulurken birbiri ardına sırayla oluşturulur.
Örneğin **double b[]={new double[3];**
Deyimi bilgisayar belleğinde

Tablo 11.1 boyutlu değişkenlerin bilgisayar belleğinde yerleşimi

1023400	b[0]
1023464	b[1]
1023528	b[2]

yerlerini birbirini izleyen adreslerde ayırrı.

Buradaki sorun eğer bilgisayar belleğindeki toplam yer değiştirilmek istenirse bunun zor olmasıdır. Aynı zamanda bilgisayar boyutlu değişkenin adreslerini bilgisayar hafızasında bir referans tablosu olarak tuttuğundan boyut değiştirme bilgisayar hafızasını da zorlayan bir işlemidir. Pratik olarak yapılan uygulama yeni bir boyutlu değişken oluşturup adresleri değiştirmektir.

```
double c[]={new double[5];
for(int i=0,i<b.length;i++)
{ c[i]=b[i];}
b=c;
```

Daha gelişmiş bir bilgi yapısı olarak Vector sınıfını kullanabiliriz. Vector sınıfı **java.util** kütüphanesinde tanımlanmıştır. Vector sınıfında her bilgi elemanı aynı zamanda bir sonraki bilgi elemanın adresini taşıır. Böylece istenildiğinde dizinin son elemanına yeni bir adres ilave edilerek kolayca yeni elemanlar ilave edilebilir.

11003.JPG



Sekil 11.3 Vectör sınıfının hafızada yerleşimi

Dizinin arasından herhangibir elemanı da kalaylıkla sadece referans adresini değiştirerek silebiliriz.

Vector sınıfının tanımı :

```
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 sınıfında tanımlanan metodların bazlarının anlamları şunlardır :

addElement(Object)

Vektöre object türü (herhangi bir sınıf olabilir, vectörün hepsi bu sınıfta tanımlanmış olmalıdır.) bir eleman ekler.

capacity()

vektörün toplam kapasitesini verir.

clone()

vectorun bir kopye vectörünü verir.

contains(Object)

Vectörün içinde sorgulanınan elemanın olup olmadığını kontrol eder.

copyInto(Object[])

vectorü boyutlu değişkene aktarır.

elementAt(int indeks)
Verilen indeksteki elemanı verir..

elements()
Tüm vektörü enumeration sınıfına aktarır.

ensureCapacity(int yeni_kapasite)
eğer yeni_kapasite vector kapasitesinin üstündeyse vector yeni_kapasite'ye aktarılır.

firstElement()
Vectörün ilk elemanını verir.

indexOf(Object)
Object'in indeks değerini verir

indexOf(Object, int indeks)
Object'in indeks değerini verir, arama indeks değişkeninden başlayarak yapılır.

insertElementAt(Object, int indeks)
Vectörün indeks noktasına Object ilave edilir.

isEmpty()
Vectörün boş olup olmadığı kontrol edilir.

lastElement()
vectörün son elemanın indeksini verir.

lastIndexOf(Object)
Object'in vectördeki son tekrarının indeksini verir.

lastIndexOf(Object, int indeks)
indeks ten başlayarak geriye doğru Objectin ilk tekrarlandığı yeri bulur.

removeAllElements()
Vectördeki bütün elemanları siler.

removeElement(Object)
Vectördeki Object elemanını siler.

removeElementAt(int indeks)
indeksle verilen elemani siler.

setElementAt(Object, int indeks)
Objectin indeksini indeks olarak belirler.

setSize(int)
Sets the size of this vector.

size()
Vectördeki toplam eleman sayısını verir.

toString()
Vectörün String eşitini verir.

trimToSize()
Vectörün boyutunu dolu olan elemanlarının boyutuna kısaltır.

Aşağıda bu metodların önemlilerinden bazılarını kullanan bir örnek program verilmiştir.

Program 11.6 : VectorTesti.java programı, vector sınıfının kullanımı

```
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();
```

```

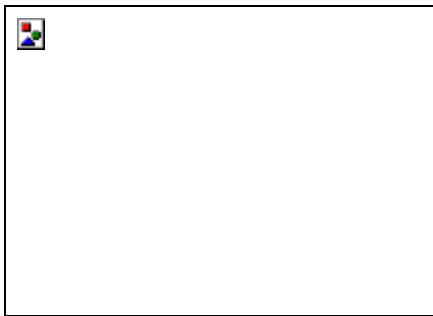
YaziPaneli.setFont(new Font("Serif",Font.BOLD,12));
YaziPaneli.setLayout( new GridLayout(5,2) );
v = new Vector(1);
prompt= new Label("Bir String degiskeni giriniz ");
input = new TextField(10);
addBtn = new Button("Ekle");
removeBtn = new Button("Cikar");
firstBtn = new Button("Ilk");
lastBtn = new 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)
    {
}
}

```

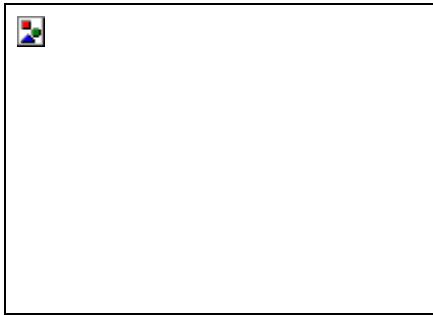
```

        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\u0131 " +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\u0131salt\u0131ld\u0131");
}
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



Şekil 11.4-11.5 [VectorTesti.java](#) programı applet çıktısı

Bundan sonraki program iki kümenin bileşim ve kesişim kümelerini hesaplamaktadır. Kümelerin kaç elemandanoluştuğu tanımlanmamıştır. İşlemler vector sınıfını kullanarak yapılmaktadır. Ayrıca bu yapıda vector sınıfını bir döngü içinde kullanmak için vector sınıfı **Enumeration** sınıfına yüklenmektedir. Enumeration sınıfının görevi, StringTokenizer sınıfının görevini andırır. Vector yapısının içinden her vector elemanını ayrı ayrı çağrıma ve indeksleme görevi görür. Enumeration interface'sinin tanımı :

```
public interface Enumeration {  
    boolean hasMoreElements();  
    Object nextElement();  
}
```

şeklindedir.

```
Enumeration n1=list1.elements();  
Enumeration n2=list2.elements();  
while(n1.hasMoreElements())  
{  
    s1=(String)n1.nextElement();  
    bilesimVectoru.addElement(s1);  
}
```

kod parçasığında vector sınıfının elements metodu,Enumeration sınıfı ve Enumeration sınıfının **hasMoreElements** ve **nextElement** metotları kullanılarak nasıl döngü oluşturulduğu görülmektedir.
Aşağıdaki örnekte setA sınıfı verilmiştir. Bu örnekte bir kümelerin bileşim ve kesişim kümelerinin vector metodunu kullanarak nasıl oluşturulabileceği gösterilmektedir.

Program 11.7 setA.java setTest.java sınıfları, bir kümenin bileşim ve kesişim kümelerini hesaplar.

```
// Bu program vector ve Enumeration sınıflarını  
// Kullanmaktadır. Bileşim, kesişim kümelerini hesaplar  
// Aynı zamanda StringTokenizer sınıfını kullanır.  
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();
    // Sonuclarini 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)
    {
        // StringTokenizer 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 kesim k□melerini hesapla.
        ansUn = set.bilesim(list1,list2);
        ansInt = set.kesisim(list1,list2);
        // sonucları cikti alanına gönder
        cikti.append("\nIki listenin bilesim kumesi : \n");
        ciktiA(ansUn);
        cikti.append("\nIki listenin kesim kumesi : \n");
        ciktiA(ansInt);
    }
} //actionPerformed metotunun sonu
}

```

Bu programda aynı zamanda StringTokenizer sınıfı kullanılmıştır. StringTokenizer String değişkenini boşluk kullanarak alt değişkenlere ayırmaya yarayan Enumeration türü bir sınıfır.

11006.JPG



Şekil 11.6 Vector sınıfını kullanarak iki kumenin bileşim ve kesişim kümelerini hesaplayan [SetTest.java](#) programı applet çıktısı

Aynı programın Swing versiyonu [SetTestSWF_2000.java](#) aşağıda verilmiştir :

```
// Bu program vector ve Enumeration sınıflarını  
// Kullanmaktadır. Bileşim ve kesişim kümelerini hesaplar  
// Aynı zamanda StringTokenizer sınıfını kullanır.
```

Program 11.8 setTestSWF_2000.java, bir kümenin bileşim ve kesişim 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 sınıfı Vector'u kullanarak  
    // iki vector listesi yarat  
    Vector list1 = new Vector();  
    Vector list2 = new Vector();  
  
    // Sonuçları yine vektor cinsinden de§iskenler  
    // kullanarak aktar  
    Vector ansUn;  
    Vector ansInt;  
    Container c;  
  
    public SetTestSWF_2000()  
    {  
        super("küme testi bileşim ve kesişim kümeleri");  
        c=getContentPane();  
        c.setLayout(new FlowLayout());  
        P1 = new JLabel("Birinci listeyi giriniz : ");  
        T1 = new JTextField(30);  
        T1.setBackground(c.getBackground());  
        P2 = new JLabel("İkinci 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)
{
    // StringTokenizer sinifi String degiskenleri icin
    // Enumeration sinifinin yaptigina paralel gorev
    // gorur String'i soraya sokarak degerlerini sorayla 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 kesim kümelerini hesapla.
    ansUn = set.bilesim(list1,list2);
    ansInt = set.kesisim(list1,list2);
    // sonucları cikti alanina gonder
    cikti.setText("");
    cikti.append("İki listenin bileşim kümesi : \n");
    ciktiA(ansUn);
    cikti.append("İki listenin kesişim kümesi : \n");
    ciktiA(ansInt);
}

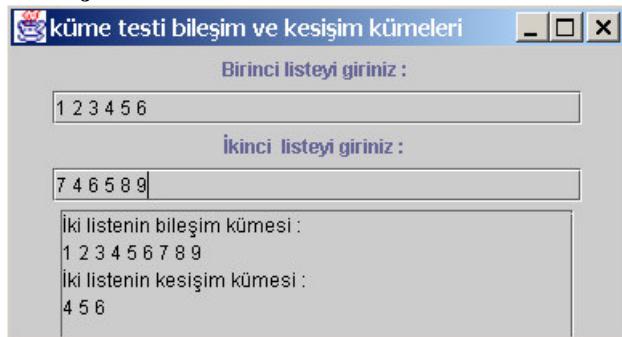
```

```

} //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



Şekil 11.7 Vector sınıfını kullanarak iki kumenin bileşim ve kesişim kümelerini hesaplayan SetTestSWF_2000 programı JFrame çıktısı

11.4 LIST(LİSTE) SINIFI

Yukarıdaki vektor sınıfında ilk defa dinamik hafiza kullanabilen (vektor boyutunu program çalışırken değiştirebilen) Vector sınıfını inceledik. Eğer vektor boyutları program çalışırken azalıyor veya çoğalıyorsa, boyutlu değişkenler yerine dinamik hafiza kullanımı toplam bilgisayar hafızası kullanımı açısından çok daha verimlidir. Dinamik data yapıları sadece vektorden ibaret değildir. List(liste), Stack(dizin) ve Queue (sıra) ve Tree(ağaç) yapıları da oldukça sık kullanıcılar dinamik yapıları teşkil eder. Bunlardan List yapısının özellikleri şunlardır :

- Listenin ilk elemanından önce veya son elemanından sonra listeye dinamik eleman ilavesi yapılabilir.
- Listenin herhangibir ara noktasına yeni eleman ilave edilemez.
- Listenin ilk elemanı veya son elemanı listeden çekilebilir.
- Listenin herhangi bir ara noktasından eleman çekilemez.

Burada List kavramının daha iyi anlaşılabilmesi için önce kendi List sınıfımız oluşturulmuş ve bir örnek problemde kullanılmıştır. Bu program örneklerine bir göz atalım.

Program 11.9 List.java programı. Bu program List yapısını tanımlayan ListNode ve List sınıflarını 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 eşdeğer 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 sınıfının [EmptyListException](#) dosyasındaki tanımı

```

public class EmptyListException extends RuntimeException
{
public EmptyListException(String name)
{
super(" "+name+" bos ");
}
}

```

Program 11.11 : List yapısını 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 programının sonucu :

List yapısını kullanan örnek program [ListTest.java](#) nin çıktısı

```

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 sınıfının girdisinin Object sınıfı olduğun dikkatinizi çekelim. Object sınıfı yine dinamik object değişken yapıları olarak tanımlanan **Double**, **Byte**, **String**, **Integer**, **Boolean**, **Float**, **Short**, **Long** gibi değişken türleri için kullanılan sınıfların abstract sınıfını teşkil eder. Bu yüzden bu sınıfların hepsini temsil edebilir ve yerlerine kullanılabilir (genel bir sınıfır.). Bu sınıflar normal değişken türlerine dönüştürülebilirler. (daha önce de kullandığımız intValue(), doubleValue() gibi metodları kullanarak)

Object sınıfının temel tanımı (metodtanımları olmadan) şu şekildedir.

```

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 sözcüğünün anlamak istiyorsanız, 14 üncü bölümü inceleyebilirsiniz. Bu terim metodun anadilde (native) yazıldığını belirtir.

Java.util paketinde List sınıfı tanımlanmıştır. Yukarıda kendi tanımladığımız List ile yapabildiklerimizi ve daha fazlasını tanımlı List paketiyle a gerçekleştirebiliriz.

```

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 sınıfıyla birlikte kullanılmak üzere ListIterator interface'sı tanımlanmıştır.

```

/*
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);
}

```

Şimdi bu iki sınıfın kullanımını bir örnekle gösterelim.

Program 11.12 : List yapısını 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++) {

```

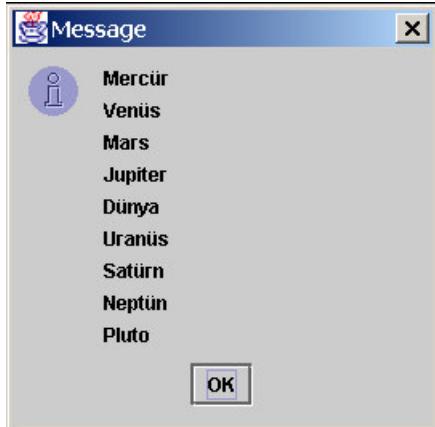
```

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



Şekil 11.08 List sınıfı, MovingPlanets çıktısı

List sınıfının bir alt sınıfı da LinkedList sınıfıdır. LinkedList sınıfı List sınıfında olmayan ilave metodlar tanımlamıştır.

```

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 sınıfı kullanan küçük bir örnek problemi inceleyelim :

Program 11.13 : LinkedList yapısını 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)
        {

```

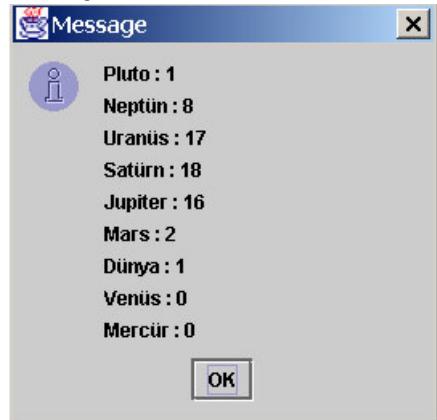
```

        Gezegen p = (Gezegen)(Gezegens.get (i));
        ss+=p.getName() + " : " + p.getaySayisi()+"\n";
    }
    JOptionPane.showMessageDialog(null,ss);
    System.exit(0);
}
}

```

Program çıktısı

1109.JPG



Şekil 11.09 Linkedlist sınıfı, Gezegenler Ve Aylar Listesi çıktısı

LinkList Bölüm 11.4'de incelediğimiz Stack(dizi) yapısı ve bölüm 11.5 de incelediğimiz queue(sıra) yapısı yerine kullanılabilir.

11.5 DİZİ (STACK) SINIFI

Diziler veya ingilizce adıyla Stack aslında üstte tanımladığımız List sınıfının bir alt sınıfıdır. dizin listeye göre daha sınırlı olan şu koşullara sahiptir :

- Dizide son elemandan sonra listeye dinamik eleman ilavesi yapılabilir.
- Dizinin herhangibir ara noktasına yeni eleman ilave edilemez.
- Dizinin sadece son elemanı listeden çekilebilir.
- Listenin herhangi bir ara noktasından eleman çekilemez.

Yukardaki kendi tanımladığımız List (Program 11.5,11.6) kullanarak oluşturduğumuz dizi programı ve diziyi kullanın bir test programı aşağıda verilmiştir.

Program 11.14 : dizi sınıfının [dizi.java](#) dosyasındaki tanımları

```

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 sınıfının [diziTesti.java](#) dosyasındaki örnek kullanımı

```
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 yapısını kullanan örnek program diziTesti.java'nın çıktısı

```
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
```

yukarıda tanımlanan **LinkedList** sınıfının addFirst()/removeFirst() metodlarını kullanarak da stack oluşturabiliriz.

java.util paketinde **vector** sınıfının yanı sıra **Stack** sınıfı da tanımlanmıştır. Java Stack sınıfında push, pop ve peek metotları vardır. push ve pop dizin programını aynısıdır. peek ise listenin en sonundaki

elemanın değerini bu elemanı listeden uzaklaştırmadan alır. (pop ile push beraber kullanılarak da aynı işi yapabilirler)

Bu sınıfın tanımı :

```
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 sınıfının metodlarının ne olduğunu daha detaylı inceleyeceğiz:

push

public Object push(Object item)

stack'in üstüne yeni eleman ilave eder

pop

public synchronized Object pop()

Stack'in tepesindeki elemanı çeker (Stack dizisinden kaldırır ve değerini return eder)

Stack'in boş olması halinde **EmptyStackException** gönderir.

peek

public synchronized Object peek()

Stack'in en tepesindeki elemanın değerini okur (Stack dizisinden kaldırılmaz ve sadece değerini return eder)

Stack'in boş olması halinde **EmptyStackException** gönderir.

empty

public boolean empty()

Stack'in boş olup olmadığını kontrol eder, boş ise true, dolu ise false değeri gönderir.

Tests if this stack is empty.

search

public synchronized int search(Object o)

Stack'in içindeki istenen Object'in yerini verir. Eğer object Stack'de yok ise -1 değeri verir

Burada hemen Stack sınıfının Vector sınıfının alt sınıfı olduğunu ve Vector sınıfında mevcut olan tüm metodları Stack metodunda da geçerli olduğunu da hatırlatalım.

Şimdi bir örnek probleme bu sınıfın nasıl kullanıldığına bakalım :

Program 11.16 : java.util kütüphanesinde yer alan Stack yapısını 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()+" deşerine bakildi");
st.print(d);
cekilen=d.pop();
System.out.println(cekilen.toString()+" deşeri 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 yapısını kullanan örnek program [StackTesti.java](#) nin çıktısı

```

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
-----
```

11.6 SIRA (QUEUE) SINIFI

Sıra veya ingilizce adıyla queue aslında üstte tanımladığımız List sınıfının bir alt sınıfıdır. sıra listeye göre daha sınırlı olan şu koşullara sahiptir :

- Sırada son elemandan sonra listeye dinamik eleman ilavesi yapılabilir.

- Sıranın herhangibir ara noktasına yeni eleman ilave edilemez.
- Sıranın sadece ilk elemanı listeden çekilebilir.
- Sıranın herhangi bir ara noktasından eleman çekilemez.

Sıra kavramının oldukça yaygın olarak kullanıldığı ülkemizde oldukça yararlı olabilecek bir sınıfı değil mi? Yukardaki List tanımını kullanarak oluşturduğumuz sıra programı ve sırayı kullanan bir test programı aşağıda verilmiştir.

Program 11.17 : sıra sınıfının [sira.java](#) dosyasındaki tanımları

```
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 : sıra sınıfının [siraTesti.java](#) dosyasındaki test programı

```
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 ciktig");
                s.print();
            }
        }
        catch(EmptyListException e)
        {System.err.println("\n"+e.toString());}
    }
}
```

sıra sınıfını kullanan [siraTesti.java](#) dosyasının çıktısı

```
sira : true
sira : true $
sira : true $ 34567
```

```

sira : true $ 34567 merhaba
true sirasi geldi onden ciktig
sira : $ 34567 merhaba
$ sirasi geldi onden ciktig
sira : 34567 merhaba
34567 sirasi geldi onden ciktig
sira : merhaba
merhaba sirasi geldi onden ciktig
Bos sira

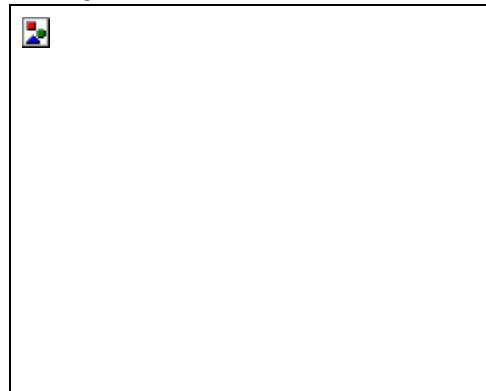
```

Queue sınıfı Java'da direk olarak tanımlanmamıştır, ancak üstteki sıra örneğinin bir benzeri Vector sınıfından bizim tarafımızdan rahatlıkla yazılabilir, veya yine LinkedList sınıfının addFirst()/removeLast() metodlarını kullanarak oluşturulabilir.

11.7 TREE(AGAÇ) SINIFI

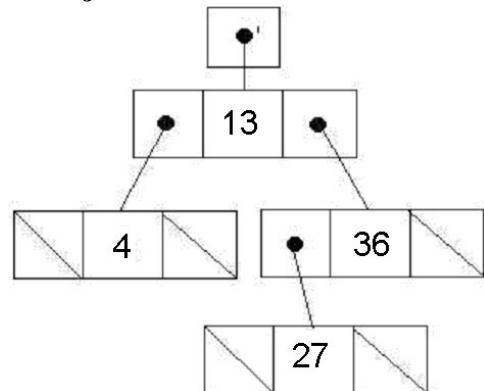
Elimizdeki bir gurup boyutlu değişkenin belli bir yapıda guruplandırılması veya aranması gerektiğinde bu standart boyut kavramında oldukça kompleks bir yapı arzeder. Bilgi guruplandırma işlemini çok daha basit bir şekilde yapabilmek için Tree yapıları öngörmektedir. Tree temel olarak Vector gibi bir yapıdır. fakat burada her Tree noktası (Nod'u) iki değişik adrese işaret eder bu adreslerden birisi o nodedaki objenin kokdekinden daha küçük olduğu bir değeri, diğeri ise daha büyük olduğu bir değeri taşır. hiç bir değer taşıımıyorsa da null değeriyle o Tree-ağaç dalı sona erer. Bu yapıyı kullanarak sayıları büyükten küçüğe veya küçükten büyüğe veya daha değişik mantıklarla direkt olarak sıralamak mümkündür.

11010.JPG



Şekil 11.10 : Binary search Tree (ikili arama ağıacı) yapısının grafiksel gösterimi

11011.JPG



Şekil 11.11 Binary search Tree (ikili arama ağıacı) yapısının bir rakam setine uygulanmasının grafiksel gösterimi

Şekil 11.10 de ikili arama ağıacı yapısının seti görülmektedir. Şekil 11.11 de ise sayısal bir örnekle yapı gösterilmiştir. Her bir kökü kendinden sonra gelen rakamı küçükse sol dala büyükse sağ dala göndermektedir. ve her yeni veri yeni bir dallanma oluşturmaktadır. Şimdi Tree yapısını oluşturacak bir örnek

sınıfı ve onun kullanıldığı bir örnek problem sınıfını oluşturarak kavramı biraz daha açmaya çalışalım :

Program 11.19 : Binary search Tree (ikili arama ağacı) yapısının programlandığı TreeNode ve Tree sınıflarının ter aldığı [Tree.java](#) programı

```
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+" ");
soldal_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);
soldal_node_soldal_siralayici(node.sol);
System.out.print(node.data+" ");
}
}

```

Program 11.20 : Binary search Tree (ikili arama ağacı) yapısının programlandığı TreeNode ve Tree sınıflarının kullanılmasını örnekleyen [TreeTest.java](#) programı

```

import Tree;

public class TreeTest
{
    public static void main(String args[])
    {
        Tree agac=new Tree();
        int sayı;
        System.out.println("orijinal olarak Tree'ye girilen sayı dizini");
        for(int i=0;i<=6;i++)
        {
            sayı=(int)(Math.random()*45)+1;
            System.out.print(sayı+" ");
            agac.gir(sayı);
        }
        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 ağıacı) yapısının programlandığı TreeNode ve Tree sınıflarının kullanılmasını örnekleyen [TreeTest.java](#) programının sonuçları. renkli olarak işaretlenen satırların sayı dizisini küçükten büyüğe ve büyükten küçüğe sıraladığını dikkat ediniz.

orijinal olarak Tree'ye girilen sayı dizini

13 11 25 34 21 26 14

Sol dal - kok - sağ dal sıralama :

11 13 14 21 25 26 34

kok -Sol dal - sağ dal sıralama :

13 11 25 21 14 34 26

Sol dal - sağ dal - kok sıralama :

11 14 21 25 26 34 13

Sağ dal - kok - sol dal sıralama :

34 26 25 21 14 13 11

kok -Sag dal - sol dal sıralama :

13 25 34 26 21 14 11

Sağ dal - sol dal - kok sıralama :

34 26 25 21 14 11 13

11.8 DICTIONARY ve HASHTABLE SINIFLARI

Şu ana kadar bu konuda gördüğümüz elemanlarda belli bir bağlantı kuralı kullanarak elemanları birbirine bağlamıştık. Bu hafiza açısından yoğun bir işlemidir. Her seferinde hafiza pozisyonunun araştırılmasını gerektirir. Java anahtar referans sağlayan daha değişik bir yapıyı da içerir. Burada her elemanın kendimiz tarafından verilen bir referans değeri mevcuttur, ve elemani bulmak için bu referansı kullanabiliriz. Bu gurubun abstract sınıfı Dictionary sınıfıdır. Dictionary sınıfının altında Hashtable sınıfı yer alır. Dictionary sınıfının tanımı :

```
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 sınıfının tanımı :

```
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 entrySet()
public Set keySet()
public Collection values()
public synchronized int hashCode()
}

```

Hashtables sınıfı temel olarak Dictionary ve Map sınıflarından türetilmiştir. Map sınıfı tanımı

```

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();
    }
}

```

şeklindedir.

En çok kullanacağımız Hashtable metodlarının anlamlarına bir göz atalım :

Bir referans çifti girmek için : put(Object anahtar, Object değer)

Anahtarı vererek referans değerini çağırmak için : get(Object anahtar)

Bir elemanı silmek için : remove((Object anahtar))

Boyutu kontrol için : size()

Bos olup olmadığına kontralı için empty()

Tüm anahtarların listesini almak için : keys() veya keySet()

Tüm değerlerin listesini almak için : elements() veya entrySet()

Program 11.21 : Hashtable yapısının programlandığı GezegenCaplari.java programı

```

import java.util.Enumeration;
import java.util.Hashtable;
import javax.swing.JOptionPane;

public class GezegenCaplari {

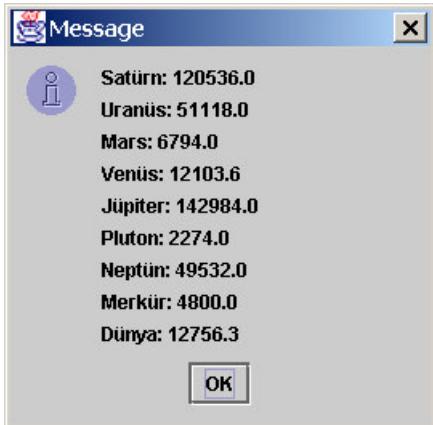
```

```

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



Şekil 11.12 HashTable sınıfını kullanan GezegenCapları programı

Programdan da görüldüğü gibi çıktı hiçbir mantıksal sıralama düzeni içermemektedir, fakat burada önemli olan gezegenlerin ve çaplarının birbirleriyle olan ilişkilerinin listelenmesidir. Hashtable bu tür işlevleri en iyi bir şekilde yerine getirir.

11.9 ARRAYS SINIFI VE SIRALAMA

Arrays sınıfı boyutlu değişkenleri sıralamaya yarıyan bir java sınıfıdır. Metodları:

```

public static List asList(object array[]);
b değerinin a değerinin hangi elemanında olduğunu bulan :
public static int binarySearch(<değişkentürü> a[],<değişkentürü>b);
<değişkentürü> char,double,float,int,Object,long,short değerleri alabilir, elbette Object üzerinden Objectin alt sınıfları olan String, Double, Integer gibi sınıfları da kabul eder.
İki boyutlu değişkenin birbirine eşit olup olmadığını denetleyen :
public static boolean equals(<değişkentürü> a[],<değişkentürü>b[]);
boyutlu değişkenin tüm değerlerini veya tanımlanan bir bölümünü bir bölümünü aynı tür bir değişken atayan
public static void fill(<değişkentürü> a[],<değişkentürü>b);
public static void fill(<değişkentürü> a[],int başlangışındisi,int bitişindisi,<değişkentürü>b);
boyutlu değişkeni sıraya sokan :
public static void sort(<değişkentürü> a[]);
public static void sort(Object a[],Comparator comp);

```

listeden de görüldüğü gibi tüm metodlar static metodlardır.

Program 11.22 : Arrays yapısının kullanıldığı Gezegensirasi.java programı

```
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



Şekil 11.13 Arrays sınıfını kullanan Gezegensirasi programı

programda liste ikinci kere Comparator sınıfı InsensitiveComp() kullanarak büyük ve küçük harfi göz önüne alınmadan sıraya sokulmuştur.

11.10 ALIŞTIRMALAR

1. [H10AL1.java](#) : Vector sınıfı örneği, H9O1 de boyutlu değişkenle çözülen dosyadan okunan rakamların ortalamasının bulunması.

Program 11.23 : H10Al1.java programı, vector ve dosya girdi örneği

```

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
dosyanın ve yuklendiği vektorun içindeki sayıların adedi: 20
Ortalama: 10.5
```

2. [H10AL2.java](#) : List sınıfı örneği, H9O1 de boyutlu değişkenle çözülen dosyadan okunan rakamların ortalamasının bulunması.

Program 11.24 : List sınıfı ve dosyadan okuma örneği 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("Lütfen 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("dosyanın ve yükleniği listenin içindeki sayıların adedi: "+j);
        System.out.println("Ortalama: "+(double)toplam/j);
    } //main
} //class
```

```
Lütfen 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
dosyanın ve yuklendiği listenin içindeki sayıların adedi: 20
Ortalama: 10.5
```

3. [H10AL3.java](#) : **dizi** sınıfı örneği, H9O1 de boyutlu değişkenle çözülen dosyadan okunan rakamların ortalamasının bulunması.

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

12. H10AL4.java : **dizi** sınıfı örneği, H9O1 de boyutlu değişkenle çözülen dosyadan okunan rakamların ortalamasının bulunması.

Problem 11.26 : H10AL4.java dizi sınıfı örneği

```

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

13. [H10Al5.java](#) : Stack sınıfı örneği, H9O1 de boyutlu değişkenle çözülen dosyadan okunan rakamların ortalamasının bulunması.

Problem 11.27 [H10Al5.java](#) Stack sınıfı örneği

```

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 yuklendiği listenin içindeki sayıların 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

dosyanın ve yükleniği listenin içindeki sayıların adedi: 20

Ortalama: 10.5

Alıştırmalarda kullanılan c.txt dosyası :

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

6. SetTest.java programını inceleyiniz. Kesişim kumesiyle bileşim kumesinin farkını oluşturan kümeyi hesaplayan bir metot yazınız ve test programına da ekleyerek çalıştırınız.

17. H1OD1.java

veri dosyası e.txt'e e dizisini giriniz. d.txt'e d dizisini giriniz. program setText'ti inceleyiniz. iki dizinin bileşim kümесini ed.txt dosyasına yazdırınız.

Program 11.28 H1OD1.java iki dizinin bileşim kümесini vektor sınıfını kullanarak hesaplayan program

```
import java.io.*;
import Text;
import java.util.*;
public class H1OD1
{
    public void girdiA(String string, Vector list)
    {
        // StringTokenizer sınıfı String değişkenleri için
        // Enumeration sınıfının yaptığına paralel görev
        // gorur String'i sıraya sokarak değerlerini sırayla 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

```

8. H10OD2

alıştırma 7 deki sonuçları Tree sınıfını kullanarak küçükten büyüğe doğru sıralanmış olarak ed.txt dosyasına yazdırınız.

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)
    {
        // StringTokenizer sinifi String degiskenleri icin
        // Enumeration sinifinin yaptigina paralel gorev
        // gorur String'i soraya sokarak degerlerini sorayla 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 ayin 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;
    }
}

```

18. **H10OD3**

Ekrandan bir rakam dizisini tek bir String olarak giriniz. Girdiginiz diziyi StringTokenizer sınıfı bir gerçek sayı rakam dizisine dönüştürünüz ve vector sınıfına yükleyiniz, daha sonra vector elemanlarının karesinin ortalamasını çıktı 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.readLine();
        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
```

19. [H10OD4](#)

Ekrandan bir rakam dizisini tek bir String olarak giriniz. Girdiginiz diziyi StringTokenizer sınıfı bir gerçek sayı rakam dizisine dönüştürünüz ve Stack sınıfına yükleyiniz. daha sonra Stack elemanlarının karesinin ortalamasını çıktı 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 "zelliğten yararlan❑lm❑Ýt❑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("tüm rakamları giriniz : ");
        s1 = cin.readLine();
        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 :
tüm rakamları 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 (aynı program 11.20'nin değişik yazılışı) : [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.readLine();
        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

```

20. Java Vector uygulaması Gezegenler.java

Program 11.34 . Gezegenler.java vectör test programı

```

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));
}
}
}
}

```

21. GezegenlerveAylar.java, Vector uygulaması

Program 11.35 . GezegenlerveAylar.java vector test programı

```

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 ANALİZ

12.1 SAYISAL ANALİZE GİRİŞ

Bu bölümümüzde bazı pratik matematik problemlerinin java ile nasıl çözülebileceğine göz atacağız. Sayısal analiz işlemlerini yapmak amacıyla iki sınıf oluşturduk bunlar Matrix ve Numeric sınıflarıdır. Bu serinin devamı olan java ile numerik analiz kitabında bu sınıflardaki metodları tek tek incelemeye çalışacağım. Burada ise sadece program listeleri ve örnek problemler vermekle yetineceğiz. Buradaki metodları uluslararası projelerde kullanıldığından metot isim ve tanımlamaları ingilizcedir. Sadece bu kitabın kapsamı için değil gerçek nümerik problemlerde kullanılmak amacıyla geliştirilmiştir. Burada ayrıca hemen şunu da belirtelim: sayısal metodların kullanımı bu metodlar, sayısal analiz ve bilgisayar değişkenlerinin değişimi ile ilgili iyi bilgi gerektirmektedir. Bu sınıf ve metodları ciddi bir uygulamada kullanmadan önce verilen metodlar hakkında nümerik analiz kitaplarından yeterince bilgi sahibi olmanızı ve test fonksiyonları kullanarak sizin istediğiniz görevi yapıp yapamayacaklarını iyice kontrol etmenizi tavsiye ederim. Türkiyedeki nümerik analizle ilgilenen arkadaşlar eğer ilgilenirlerse benim kendi projelerimde kullanmak için geliştirdiğim bu gurubu daha da geliştirerek genel bir nümerik analiz kütüphanesine dönüştürmemiz mümkün olabilir.

12.2 MATRIX SINIFI

Matrix sınıfı double ve complex vektör (bir boyutlu değişken) ve matris (iki boyutlu değişken) değişkenlerinin matematik işlemlerini yapmak amacıyla tanımlanmıştır. complex (kompleks değişken sınıfı) daha önce türkçe olarak tanımlanan kompleks sınıfının aynıdır, fakat metot isimleri burada ingilizce olduğundan bu sınıfın tanımını tekrar verelim :

Program 12.1 complex.java, ingilizce metot isimleriyle kompleks değişkenler sınıfı.

```
//=====
// 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 subtract(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 subtract(complex left, complex right)
{ // return subtraction 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 definitions

```

Şimdi Matrix metoduna daha detaylı bakabiliriz. İlk önce Matrix sınıfının metod başlıklarını verelim :

toString Metotları

```

public static String toString(double left)
    left double değişkeninin String eşdeğerini yazma hazır hale getirir.
public static String toString(double[][] left)
    left double matrisini String değişkeni olarak yazma hazır hale getirir.
public static String toString(complex[][] left)
    left complex matrisini String değişkeni olarak yazma hazır hale getirir.
public static String toStringT(complex[] left)
    left complex vektörünü String değişkeni sütun olarak yazma hazır hale getirir.
public static String toString(complex[] left)
    left complex vektörünü String değişkeni satır olarak yazma hazır hale getirir.
public static String toStringT(double[] left)
    left double vektörünü String değişkeni sütun olarak yazma hazır hale getirir.
public static String toString(double[] left)
    left complex vektörünü String değişkeni satır olarak yazma hazır hale getirir.

```

inverse matris ve denklem sistemi çözüm hesapları

Denklem çözümlerinde temel olarak iki sayısal analiz tekniğinden yararlanılmıştır. Birinci metod tam pivotlu gauss eliminasyon yöntem,, diğeri ise LU ayrıştırma yöntemidir. Gauss eliminasyon yöntemi matrisi eliminasyon yöntemi kullanarak bir üst üçgen matris haline dönüştürür, yani diagonal elementlerin altında kalan elemanları sıfırlar. Böylece denklem sistemi tek bilinmiyenli bir lineer denklemden başayarak iki bilinmiyenli, üç bilinmiyenli vs bir denklem sistemine dönüşür. Bu denklem sistemi bir bilinmiyenli denklemin çözülüp değerinin iki bilinmiyenli denklemde yerine konulmasıyla ikincinin çözümüne ve aynı işlemin seri halde devamıyla tüm bilinmeyenler çözülür. LU ayrıştırma metodunda ise matris önce bir üst üçgen matris(U) ve bir alt üçgen matris (L) ye ayrılır. Sistem L ve U matrislerini kullanarak çözülür. İverse (ters) matris hesabı da yine bu iki yöntemin kullanılmasıyla gerçekleştirilir. Eğer sadece bir matris çözülecekse gauss eliminasyon ve LU yöntemleri arasında hesaplama verimi farkı yoktur, ancak sadece denklemin sağ tarafı değişiyorsa LU daha

verimli olarak hesap yapar. Hesaplama hassasiyetini artırmak için pivoting adı verilen bir metod kullanılmıştır. Pivoting matrisin satır ve sütunlarının yerini değiştirerek yuvarlama hatalarının 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 metodу yardımıyla çözer.
public static double[] AXB(double a[][],double b[])
    lineer denklem sistemini tam pivotlu gauss eliminasyon metodu yardımıyla çözer.
```

Determinant hesapları

Determinant özel bir matris çarpım prosesidir. Tek bir sayıyla ifade edilir. Determinant hesabı için LU metodu daha uygun olduğundan bu metod kullanılmıştır. Aşağıdaki metodlar aslında tek bir metod olup, değişik isimleri verebilmek amacıyla üç ayrı metod olarak sunulmuştur.

```
public static double det(double a[][])
    LU metodu yardımıyla determinant hesaplar
public static double determinant(double a[][])
    LU metodu yardımıyla determinant hesaplar
public static double D(double a[][])
    LU metodu yardımıyla determinant hesaplar
```

Matrislerin dört işlemi

Burada matrislerin çarpımı (multiply) metodlarında vektörel matris çarpımlarından bahsediyoruz.

```
public static double[][] multiply(double[][] left,double[][] right)
    iki double matris çarpımı
public static double[] multiply(double[][] left,double[] right)
    double matris - vektör çarpımı
public static double[] multiply(double[] left,double[][] right)
    double vektör - matris çarpımı
public static double[][] multiply(double left,double[][] right)
    double sayı - matris çarpımı
public static double[][] multiply(double[][] left,double right)
    double matris - sayı çarpımı
public static double[] multiply(double left,double[] right)
    double sayı - vektör çarpımı
public static double[] multiply(double[] left,double right)
    double sayı - vektör çarpımı
```

Matrislerin toplamı(add) aynı konumda olan elemanların toplamıdır. Fark işlemi de (subtract) aynı konumda olan iki matrisden ilk verilenden ikinci verilenin işaretinin değiştirilerek toplanmasıdır.

```
public static double[][] add(double[][] left,double[][] right)
    iki double matris toplamı
public static double[] add(double[] left,double[] right)
    iki double vektör toplamı
public static double[][] subtract(double[][] left,double[][] right)
    iki double matrisin farkı
public static double[] subtract(double[] left,double[] right)
    iki double vektörün farkı
```

vectorlerin bölümüyle birinci vectorün tersinin (inverse) alınarak ikinci matrisle çarpılması anlaşılır.

$A/B = B^{-1}A$

```
A/B = B*inv(A)
public static double[][] divide(double[][] left,double[][] right)
    iki matrisin bölümü (birinci matrisin ikinci matrisin inversiyle çarpımı)
public static double[][] LUDivide(double[][] left,double[][] right)
    iki matrisin LU metodу kullanılarak bölümü (birinci matrisin ikinci matrisin inversiyle çarpımı)
public static double[] LUDivide(double[] left,double[] right)
    vektörün matrise bölümü (LU metodу)
public static double[] divide(double[] left,double[] right)
    vektörün matrise bölümü
public static double[] divide(double[] left,double[][] right)
```

vektörün matrise bölümü

Üst işlemi : üst işleminden anlaşılan matristeki her elemanın ayrı ayrı üssünü almaktır

public static double[][] pow(double[][] right,double left)

matrisin double kuvveti (her elemanın ayrı ayrı kuvveti)

public static double[] pow(double[] right,double left)

vektörün double kuvveti (her elemanın ayrı ayrı kuvveti)

Absolute matris : absolute matris, matris elemanlarının tümünün değerini positife çevirir.

public static double abs(double[][] left)

matrisin mutlak değeri

public static double abs(double[] left)

vektörün mutlak değeri

Transpose matris : transpose matris matrisin satır ve sütünlarının yerini değiştirir, yani satırlar sütun, sütunlar satır haline dönüşür. $(a_{ij})^T = a_{ji}$

public static double[][] Transpose(double [][] left)

Transpose matrix (satır ve sütunların yer değiştirmiş hali)

public static double[][] T(double [][] left)

Transpose matrix (satır ve sütunların yer değiştirmiş hali)

public static complex[][] T(complex [][] left)

Transpose matrix (satır ve sütunların yer değiştirmiş hali)

Birim matris : birim matris diagonal elemanında (a_{ii}) 1 değeri olan diğer elemanları ise 0 taşıyan ($a_{ij,j \neq i} = 0$) matristir.

public static double[][] I(int n)

Birim matris

public static double[] one(int n)

Birim vektör (tüm değerler 1'e eşit)

EigenValue Problemi : EigenValue, veya sınır değer problemi matris işlemlerindeki ve matematikte oldukça önemli bir işlemidir. Sınır değer işlemine kısaca bir denklem sisteminin köklerini bulma da diyebiliriz. Genel tanım olarak $[A]$ matrisinin sınır değeri

$[A]\{X\} - \lambda[I] = 0$ olarak tanımlanır. Burada $\{X\}$ denklemi sağlayan vektör setidir, eigen vektörü (sınır değer vektörü) olarak adlandırılır. λ değeri ise EigenValue(sınır değeri) adını alır. Buradaki $[I]$ birim matrisdir.

Şüphesiz, verilen bir matris için birden fazla eigenvektör ve eigenvalue bulunabilir.

public static double[][] balance(double b[][])

balance işlemi eigenvalue hesaplamalarını kolaylaştırın bir ilk iterasyon prosesidir. Bazı durumlarda sonuca ulaşamayabilir. Bu yüzden dikkatli kullanılmalıdır.

public static double[][] Hessenberg(double b[][])

Hessenberg matris dönüşümü, eigenvalue hesaplamalarında kullanılır.

public static double[][] QR(double b[][])

Eigenvalue (sınır-değer hesaplama metodu, simetrik olmayan matrlslere de uygulanabilir.)

sonuçlar n^2 (gerçek ve imajineri) matris olarak aktarılır.

public static double[][] eigenValue(double b[][])

QR metodunun aynıdır.

public static complex[] eigenValueC(double b[][])

QR metodunun aynıdır tek farkı sonuçları kompleks vektör olarak vermesidir.

public static double[][] balancedEigenValue(double b[][])

QR metodu uygulanmadan önce balance metodunu uygular.

public static double[][] eigenQL(double a[][])

Eigenvalue (sınır-değer hesaplama metodu, sadece simetrik matrlslere uygulanabilir, simetrik eigenvalue değerleri gerçekdir, kompleks kökler vermez)

Polinomun kökleri

Polinomun köklerinin bulunması aslında bir eigenvalue (sınırdeğer) problemidir. Herhangi bir dereceden polinomun katsayıları verildiğinde karekteristik matris adı verilen bir matris eşdeğeri oluşturulabilir. Bu matrisin eigenvalue değerleri verilen polinomun kökleridir.

public static double[][] poly_roots(double c[])

QR Eigenvalue (sınır-değer hesaplama metodu, simetrik olmayan matrlslere de uygulanabilir.)

metodu kullanarak polinomun köklerini hesaplar. polinom katsayılarını önce karektteristik matrise çevirir. sonuçlar n*2 (gerçek ve imajineri) matris olarak aktarılır.

public static complex[] poly_rootsC(double c[])

Yukarıdakının aynıdır. ancak sonuçlar kompleks vektör olarak aktarılır.

12.1.1 Matrix Sınıfının Listesi :

Problem 12.2 [Matrix.java](#), Matris proseslerini tanımlayan Matrix sınıfı

```
//=====
// 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 idxc[];
int idxr[];
double ipiv[];
idxc=new int[n];
idxr=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;
        }
    idxr[i]=irow;
    idxc[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(l=0;l<n;l++)

```

```

if(l!=icol)
{
    dum=b[l][icol];
    b[l][icol]=0.0;
    for(l=0;l<n;l++) b[l][l]-= b[icol][l]*dum;
}
for(l=n-1;l=0;l--)
{
    if(indxr[l] != idxc[l])
        for(k=0;k<n;k++)
        {
            temp=b[k][idxc[l]];
            b[k][idxc[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 idx[] = new int[n];
int d[]={1};
double e;
double b[][]=new double[n][n];
b=Matrix.LU(a,idx,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]=left*b[i][j];
}
return b;
}

```

```

{
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 *****
===== definatoin 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
}
//=====================================================================
//*****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[][] subtract(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[] subtract(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(c>g)
{
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
}

```

12.3 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 iteratif metodlar daha az sayıda işlemle çözüme gidebilir. Bu metod hassas sonuç verebilme özelliğinden dolayı seçilmiştir. aynı zamanda LU (alt ve üst üçgen matris) parçalama metoduyla da çözübiliriz. Bu metodun avantajı lineer bir denklem sistemi çözerken eğer ikinci taraf sürekli değişiyorsa ortaya çıkar. tek bir çözüm için ilave bir avantaj getirmez.

Problem 12.3 [Matrix3.java](#), inverse matris problemi örneği

```

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 metotları 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-
0.1111111111111112
```

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
0.0
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 0.025000000000000000064 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 matrix problemı örneği

```
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 - Sınır değer problemi : matislerle ilgili belki de en ilginç proses eigen value - sınır değer problemidir. Sınır değer hesaplamaları iteratif bir problemdir. Simetrik matrislerde sonuçlar gerçek sayı olarak çıkar. bu tür uygulamalarda QL metodu kullanılabilir. Simetrik olmayan matrislerin sınır değerleri kompleks değerlerdir (kompleks ve gerçek değerlerin karışımı olabilir.) burada temel olarak QR formülü kullanılmıştır. Eigen value problemleriyle direkt olarak ilgili ilginç bir problem de ninci dereceden bir polinomun köklerinin hesaplanması problemdir. polinom karakteristik matris ismi verilen özel bir matrise dönüştürülebilir. Bu matrisin eigen-value değerleri polinomun da kökleridir.

Karakteristik matrisin tanımı

```

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 verilmiştir.

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}, {3,4,5,0,0,0},
               {-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
(2.0000000000000002 - 1.732050807568873i)
(2.0000000000000002 + 1.732050807568873i)
7.000000000000001
-8.99999999999998

Problem 12.6 [Matrix4.java](#) , ninci dereceden denklemi 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

Karakteristik matris :

8.0	-19.0	12.0
1.0	0.0	0.0
0.0	1.0	0.0

Eigen Value (sinir deger) :

3.999999999999964	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 çarpım, inverse ve determinant işlemleri

```
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. sınır değer vektörü QL metodu kullanılarak hesaplanmıştır. Simetrik matris olduğundan sınır değerler gerçek değerler olarak elde edilmişdir.

Problem 12.5 [Matrix6.java](#) , eigenvalue (sınır-değer) 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 2.999999999999996 1.0000000000000004  
1.0 1.0 1.0  
-1.0000000000000004 0.0 2.0000000000000004  
1.0000000000000007-0.999999999999997 1.0000000000000004
```

12.3 NUMERIC SINIFI (SAYISAL ANALIZ PAKETİ)

Numeric sınıfı genel bir sayısal analiz paketi olarak düşünülmüştür. Bu pakette benim kişisel ihtiyaçlarım için yazdığım yada başka dillerde daha önceden yazılmış olan kodlardan adepte ettiğim bazı programlar verilmiştir. Bir paketin genel olarak herhangi bir fonksiyonla kullanılabilmesi için herhangi bir fonksiyonla birlikte kullanılması gereklidir. Bu yüzden programlarda fonksiyonlar tanımlanmış abstract sınıflar içinde yer almıştır. Gerçek fonksiyonlar yazılrken abstract sınıflardan türetilmeleri gerekmektedir. Numeric paketindeki ve abstract sınıflar ve metodlar şu şekilde tanımlanmıştır :

Numerik sınıfı ve fonksiyon girdi abstract sınıfları ve metodları :

abstract class f_x

tek değişkenli tek fonksiyonları aktarmak için kullanılan abstract sınıfı. Bu sınıfla kullanılabilen fonksiyona örnek olarak $f=x^*x$ fonksiyonu verilebilir. Buradaki abstract fonksiyonun tanımı :
abstract double func(double x);
şeklindedir. Örnek verecek olursak :

```
class f1 extends f_x
{ double func(double x) { return x*x; } }
```

Bu sınıf dinamik olarak tanımlandığından kullanılıcagında tanımlanması gereklidir.

```
double y=Numeric.derivative(new f1(),x);
veya
f1 xkare=new f1();
double y=Numeric.derivative(xkare,x);
şeklinde çağrırlır.
```

abstract class f_xi

birden fazla fonksiyon birden fazla değişken ile kullanılacaksa f_xi abstract sınıfı kullanılır. Bu sınıfta tanımlanan func metoduna denklem_referansı indisinin değerine 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] değerini verir
func(x,0) f[0] değerini verir
abstract metodun tanımı :

```
abstract double func(double x[],int equation_ref);
şeklindedir.
```

Örnek problemin java kodu eşdeğeri :

```
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; } }
```

şeklinde verilebilir. Bu fonksiyonun asıl programda çağrılmaması :

```
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);
```

şeklinde gerçekleştirilir.

abstract class fi_xi

birden fazla fonksiyon birden fazla değişken ile kullanılacaksa fi_xi abstract sınıfı kullanılır. Bu sınıfta tanımlanan func metoduna tüm fonksiyon değerlerini 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] değerlerini verir.
abstract double[] func(double x[]);

Örnek problemin java kodu eşdeğeri :

```
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; }
}

```

şeklinde verilebilir. Bu fonksiyonun asıl programda çağırılması :

```

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);

```

şeklinde gerçekleştirilir.

abstract class fij_xi

```
{

```

iki boyutlu fonksiyon matrisi ve bir boyutlu değişken vectörü verilmişse fij_xi sınıfını kullanabiliriz. Bu tür fonksiyonlar özellikle iki boyutlu fonksiyon setinin türev fonksiyonlarını hesaplarken oluşabilir. Çıktı matris olarak tüm fonksiyonları 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]	

değerlerin iki boyutlu matrix olarak verir.

abstract double[][] func(double x[]);
örneğin java kodu eşdeğeri :

```

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 sınıfın java programında kullanımı :

```

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 sınıfındaki metodlar genelde statik metodlardır. Bu yüzden bağımsız olarak kullanılabilirler. Şimdi bu metodları daha detaylı inceleyelim :

Türev : bu metodlar gurubu yukarıda tanımlanan double sayı, vektor ve matris olarak verilen bir fonksiyonun türevlerini alır.

public static double derivative(f_x f_deriv,double x)

df/dx

tek fonksiyon, tek değişken türevi, örnek fonksiyon $f=x*x$

fonksiyon abstract sınıfı : f_x

türevin alınacağı noktası : double x

**public static double derivative(f_xi f_deriv,int equation_ref,
double x[],int x_ref)**

$\delta f_i / \delta x_j$

birden fazla fonksiyon birden fazla değişken türevi

denklem equation_ref indisinin ve x_ref x indisindeğerine göre sadece tanımlanan fonksiyonun tanımlanan 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 sınıfı : f_xi

yukarıdaki örnek için türevin alınacağı noktası : double x[] ile belirlenmiştir

**public static double derivative(f_i_xi f_deriv,
int equation_ref,double x[],int x_ref)**

$\delta f_i / \delta x_j$

birden fazla fonksiyon birden fazla değişken türevi

denklem equation_ref indisinin ve x_ref x indisindeğerine göre sadece tanımlanan fonksiyonun tanımlanan 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 sınıfı : f_xi

yukarıdaki örnek için türevin alınacağı noktası : double x[] ile belirlenmiştir

not1 : bir üsteki türev formülüyle bu türev formülünün temel farkı, burada tanımlanan fonksiyonun tüm vektörü çıktı olarak vermesidir. sonuçlar her ikisinde de aynıdır.

not2: denklemlerin hassasiyeti metodların içerisinde verilen h_0 ve n değişkenlerinin değiştirilmesiyle artırılabilir. Şu andaki değerleri optimize edilmişdir ve denenen tüm fonksiyonlar için hassas değerler vermiştir. Nümerik türevlerle çözülmesi oldukça zor olan boyutlu newton-raphson tekniğinde bile başarılı sonuçlar vermiştir. Daha küçük değerler kullanırken önce türevini bildiğiniz fonksiyonlarla sonuçları kontrol etmeniz tavsiye edilir.

Integral: Sayısal olarak integral almak için çok çeşitli integrasyon metodları bulunmaktadır. Burada verilen metodlarda ana kriteria sonuçların hassaslığıdır. Denklemler nümerik olarak doğru sonuç çıkarması oldukça zor olan siyah cisim radyasyon fonksiyonu ve istatistik fonksiyonlarda bile tam hassaslık vermiştir. integral 30 noktalı Gauss - Legendre integral formülü kullanmaktadır. trap, trapezoidal integral formülü romberg integrasyonun temelini teşkil etmek için hazırlanmıştır, tekbaşına da kullanılabilir. Çok hassasiyet gereken uygulamalarınız için kullanmanızı tavsiye etmiyoruz. Romberg integrasyonu iteratif bir integrasyon tekniğidir. Trapezoidal integrasyonla hesaplanan kök değerlerini alarak iterasyonla hata miktarını azaltır.

public static double integral (f_x f_xnt,double a,double b)

f(x) fonksiyonunun double a ve double b sınırları arasında integrali, Gauss-Legendre metodu

örnek fonksiyon $f=x*x$

fonksiyon abstract sınıfı : f_x

örnek çağrıma : 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 sınırları arasında integrali, trapezoidal metodu

örnek fonksiyon f=x*x

fonksiyon abstract sınıfı : f_x

örnek çağrıma : 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 sınırları arasında integrali, Romberg integrasyon metodu

örnek fonksiyon f=x*x

fonksiyon abstract sınıfı : f_x

örnek çağrıma : double y=Numeric.integral_romberg(new fx(),a,b);

Non-linear denklem kökleri : non linear denklem veya denklem sistemlerinin çözümü nümerik analizde en çok karşıımıza çıkan problemlerden biridir. Genelde tüm metodlar iteratif yöntemler kullanırlar. Bu metodlar içinde en verimli olarak hesap yapan metodun newton metodu olduğu söylenebilir, fakat newton metodu fonksiyonun türevinin de hesaplanması gereklidir. buradaki metodların bir kısmında türev girdi olarak alınırken, bir kısmında da yine nümerik metodlarla hesaplanmıştır. denklemin ikinci türevini de göz önüne alan newton_2nd_derivative gibi metodlar da verilmiştir. ayrıca secant, bisection gibi türev hesabı gerektirmeyen metodlar da mevcuttur. Bisection metodу temelde bir arama metodudur ve verimi en düşük metoddur, fakat verilen bölgeyi tam olarak taradığı ve bir kök varsa muhakkak bulabildiği için verimsiz bir metod için oldukça sık 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 sınıfı : f_x

örnek çağrıma : 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 hesaplanmaktadır

örnek fonksiyon y=x*x-2

fonksiyon abstract sınıfı : f_x

örnek çağrıma : 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 sınıfı : f_x

örnek çağrıma : 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, 2ncı türev formülü, dy(x)/dx fonksiyonu nümerik olarak hesaplanmaktadır

örnek fonksiyon y=x*x-2

fonksiyon abstract sınıfı : f_x

örnek çağrıma : 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, 2ncı türev formülü, secant metodу kullanılmıştır (türev formülü gerekmekz)

örnek fonksiyon y=x*x-2

fonksiyon abstract sınıfı : f_x

örnek çağrıma : 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, 2ncı türev formülü, bisection metodу kullanılmıştır

(türev formülü gerekmekz, temel olarak bir arama metodudur.)

örnek fonksiyon y=x*x-2

fonksiyon abstract sınıfı : f_x
örnek çağrıma : double y=Numeric.bisection(xl,xu,new fx());

public static double[] newton(double x[],fi_xi y,fij_xi dy)
 $y_i(x_i)$ fonksiyon sisteminin kökleri kökleri, $dy_i(x)/dx_j$ fonksiyonu da matrix fonksiyonu olarak verilmektedir.
örnek fonksiyon
 $y_1=2*x_1*x_1*x_2-3*x_1$
 $y_2=2*x_1+2*x_2$
örnek türev fonksiyonu
 $dy_1/dx_1=4*x_1*x_2-3$ $dy_1/dx_2=2*x_1$
 $dy_2/dx_1=2$ $dy_2/dx_2=2$
fonksiyon abstract sınıfı : fi_xi
türev fonksiyonu abstract sınıfı . fi_xij
örnek çağrıma : double y[]=Numeric.newton(x,new fixi(),new dfixi());

public static double[] newton(double x[],fij_xi y)
 $y_i(x_i)$ fonksiyon sisteminin kökleri, $dy_i(x)/dx_j$ fonksiyonu nümerik olarak hesaplanmaktadır, girdi
değildir.
örnek fonksiyon
 $y_1=2*x_1*x_1*x_2-3*x_1$
 $y_2=2*x_1+2*x_2$
fonksiyon abstract sınıfı : fi_xi
örnek çağrıma : 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, çıktı olarak polinomun katsayılarını
verir
örnek çağrıma : 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 katsayılarını kullanarak
fonksiyon değerini hesaplar
örnek çağrıma : 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 katsayılarını kullanarak
fonksiyon değerini hesaplar ve toplam hata miktarını bulur.
örnek çağrıma : double y=Numeric. error(x,y,e);

public static double[][] orthogonal_polynomial_least_square (double xi[],double fi[],int m)
ortogonal polynom least square (en küçük kareler metodu) polinom uydurma, çıktı olarak polinomun
katsayılarını verir. hesaplama yöntemi matrislere dayanmadığından matris çözümlerine dayanan hataları
yapmaz. yalnız polinom formülü daha kompleksdir
örnek çağrıma : 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 katsayılarını
kullanarak fonksiyon değerini hesaplar
örnek çağrıma : 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 diferansiyel denklem çözüm metodu, tek bir denklemi çözmek için
örnek çağrıma : 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 denklemeleri çözmek için kullanılır.
örnek çağrıma : double y[][]=RK4 (new fxi(),x0,xn,f0,N);

public static double[][] RK45(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 denklemeleri çözmek için kullanılır.

örnek çağrıma : double y[][]=RK45 (new fxi(),x0,xn,f0,N);

Problem 12.6 Numeric sınıfı 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 indexed 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;
}
}

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((ea>es)&&(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,fi_j_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_orthogonal_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(s>1.5 && 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
}

```

12.4 NUMERIC SINIFI (SAYISAL ANALIZ PAKETİ) ÖRNEK PROBLEMLERİ

Non linear denklem çözümleri :

Problem 12.7 non lineer denklem sistemi çözümü örneği , [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 not required)");
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
 $y_i = f_i(x_i) = 0, i=0, n$
 $f_i(x_i), i=1, n$ is given (derivative matrix is not required)
 1.0673460858066897
 0.1392276668868614

İntegral -türev : İntegral ve türev örneği

Problem 12.8 integral ve türev örneği, 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 : 0.33333333333329
romberg integral of class f1 : 0.33333333333333
integral of class f2 : 0.4999999999999384
Derivative of class f1 : 1.9999999999994864
Derivative of class f2 : 0.9999999999998606
Derivative of fm1,0 : 3.00000000000028932
Derivative of fm2,0 : 3.00000000000028932
Derivative of fm1,0 : 3.00000000000028932
Derivative of fm2,0 : 3.00000000000028932

```

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 işlevini biraz daha kullanıcıya cazip hale getirmek için grafik formatındaki leastSW.java programı geliştirilmiştir.

Problem 12.10 least square polinom uydurma, [leastSW.java](#) jFrame grafik programı

```

//=====
// 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 çıktı sayfasi";
    final static String KONTROLPanel = "PLOT Kontrol sayfasi";
    final static String PLOTPanel = "Plot sayfasi";
    Plot2D jta;
    StringTokenizer t;
    // least değişken 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;

// ***** çıktı 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 başlığı : ");
promptXLabel = new JLabel("x ekseni başlığı : ");
promptYLabel = new JLabel("y ekseni başlığı : ");
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(mpante,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 bağlantısı  

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 hatası : "+plotDosyasi);
}
}

public void sayfalarinyenile()
{
//en küçük kareler metoduna göre değişik
//sayfalardaki deşerleri 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 dosyası hatası : "+ girdiDosyasi);
}
catch(AccessControlException ace)
{
    System.err.println("least girdi access kontrol hatası : "+ 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 küçük kareler metodu girdileri verildiginde bu yeni action
// eventi oluşturur
if( e.getSource()==inputGirdi ||
    e.getSource()==inputPD || 
    e.getSource()==inputCikti ||
    e.getSource()==inputP1_2 || 
    e.getSource()==inputP1_3 || 
    e.getSource()==inputP1_4 )
{
    sayfalariyenile();
}

//grafigi yazıcıya gönderir
/*************
if (e.getSource()==printButton )
{
jta.yazdir();
}

//grafik min max değerlerini 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 sayısını 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 başlıklarını yeniden okur
jta.p1ymax=valYmax.doubleValue();

```

```

jta.p1.label=inputLabel.getText();
jta.p1.xlabel=inputXLabel.getText();
jta.p1.ylabel=inputYLabel.getText();
//Plot'ı (grafik) yeniden çizer
jta.yenidenciz();
}

//=====================================================================
//girdi dosyasından girdi değerlerini okur.

public void oku() throws IOException
{
//dosyadan bilgileri okur xi,yi boyutlu matrisine yükler.
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 alanına girdi deşerlerini 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 katsayılarını yazar
public String yaze()
{

String b="Polinom katsayıları : ";
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;
}

//=====
// cıktı dosyasına kayıt yapar ve cıktı alanına sonuçlar□ yazar
public String yazCikti()
{
String b=new String();
try{
ObjectOutputStream fout=new ObjectOutputStream(new FileOutputStream(ciktiDosyasi));
b="cıktı 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)
{
}
}

```

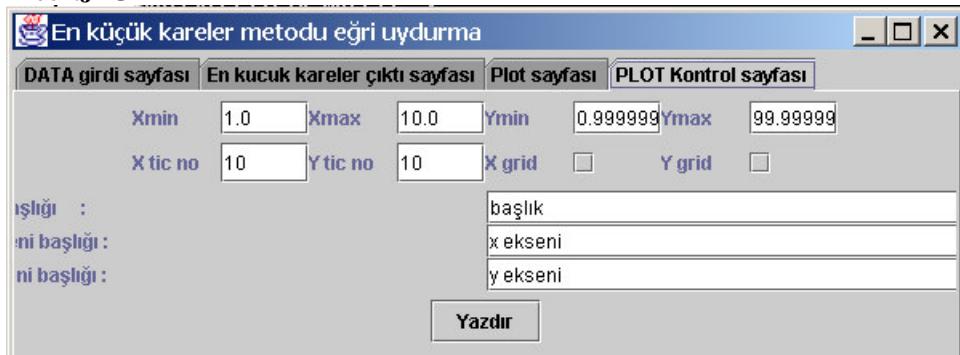
```

        System.err.println("yaz Cikti dosyasi"+ciktiDosyasi+ " excess kontrol");
    }
    return b;
}

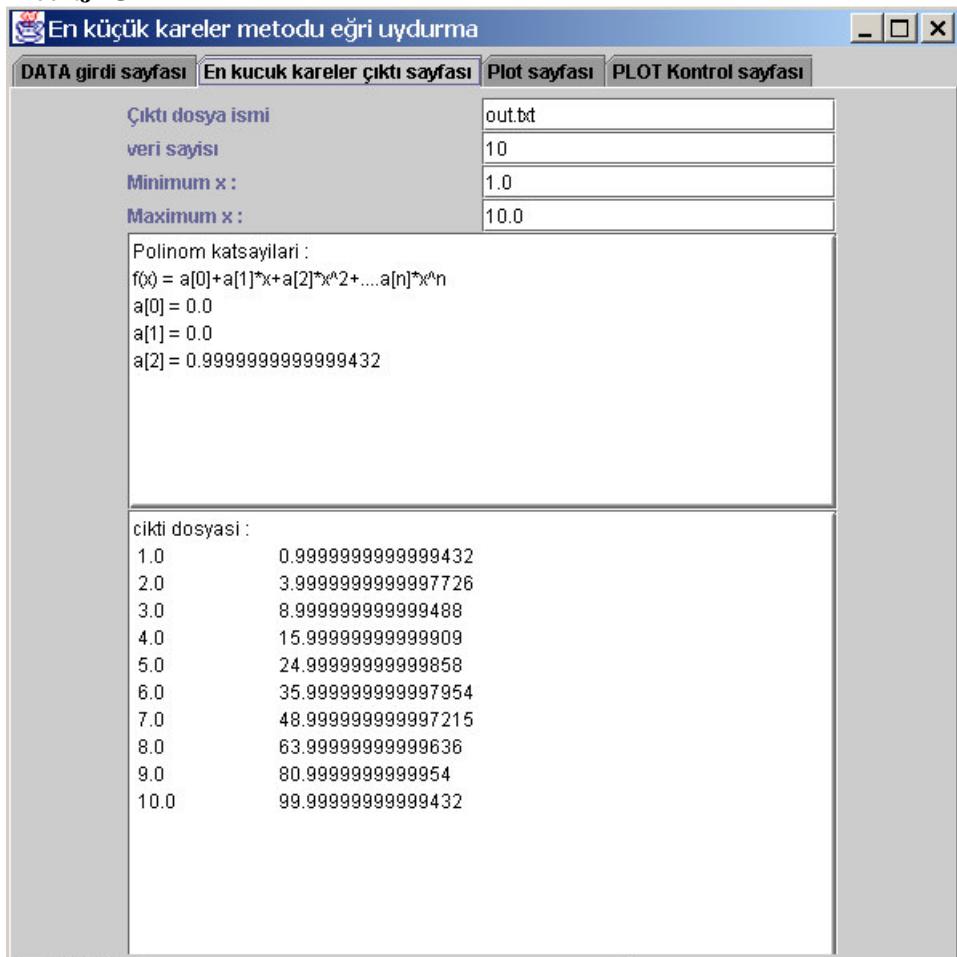
//=====
public static void main( String argv[] ) throws IOException {
    leastSW frame = new leastSW( "En küçük kareler metodu eğri uydurma" );
    frame.addWindowListener(new BasicWindowMonitor());
    frame.setSize( 600, 600 );
    frame.setVisible(true);
}
}

```

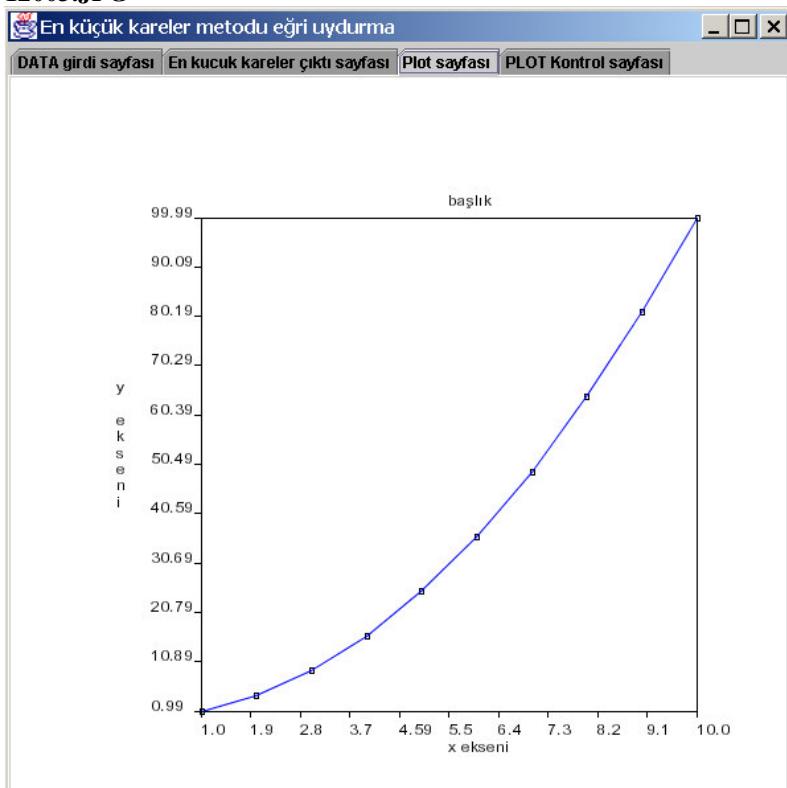
12001.JPG



12002.JPG



12003.JPG



12004.JPG



Şekil 12.1-4 en küçük kareler metoduyla polinom eğri uyduran leastSW.java sınıfı Jframe çıktılarının çeşitli sayfalarının görünümü.

Örnek çoktadaki girdi dosyaları :

Plot.txt

başlık

x ekseni

y ekseni

2

in.txt 20 0 0 0

out.txt 3 0 0 255

in.txt

1.0 0.999999999999432
2.0 3.999999999997726
3.0 8.99999999999488
4.0 15.9999999999909
5.0 24.9999999999858
6.0 35.9999999997954
7.0 48.9999999997215
8.0 63.9999999999636
9.0 80.999999999954
10.0 99.9999999999432

out.txt

cikti dosyasi :

1.0 0.999999999999432

2.0	3.999999999999726
3.0	8.99999999999488
4.0	15.9999999999909
5.0	24.9999999999858
6.0	35.99999999997954
7.0	48.99999999997215
8.0	63.9999999999636
9.0	80.999999999954
10.0	99.9999999999432

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 örneği, [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
0.04999999999999996-0.9536914203107341
0.0549999999999999-0.94945886933206
0.0599999999999999-0.9452972985683943
0.0649999999999999-0.9412063540290199
0.0699999999999999-0.9371856834886335
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
 0.19500000000000001-0.8585128324297647
 0.20000000000000001-0.856201234274326
 0.20500000000000001-0.8539510362070005
 0.21000000000000001-0.8517619320148359
 0.21500000000000001-0.849633617012017
 0.22000000000000001-0.8475657880322502
 0.225000000000000012-0.8455581434211851
 0.230000000000000012-0.843610383028874
 0.235000000000000013-0.8417222082022695
 0.240000000000000013-0.8398933217777589
 0.245000000000000013-0.8381234280737365
 0.25000000000000001-0.8364122328832122
 0.25500000000000001-0.8347594434664576
 0.26000000000000001-0.8331647685436885
 0.26500000000000001-0.8316279182877833
 0.270000000000000013-0.8301486043170393
 0.275000000000000013-0.8287265396879636
 0.280000000000000014-0.8273614388881004
 0.285000000000000014-0.8260530178288946
 0.290000000000000015-0.8248009938385901
 0.295000000000000015-0.8236050856551641
 0.300000000000000016-0.8224650134192967
 0.305000000000000016-0.8213804986673754
 0.310000000000000016-0.8203512643245341
 0.315000000000000017-0.8193770346977273
 0.32000000000000002-0.8184575354688391
 0.32500000000000002-0.8175924936878259
 0.33000000000000002-0.8167816377658933
 0.33500000000000002-0.8160246974687078
 0.34000000000000002-0.8153214039096416
 0.34500000000000002-0.8146714895430517
 0.35000000000000002-0.8140746881575915
 0.35500000000000002-0.8135307348695573
 0.36000000000000002-0.8130393661162659
 0.36500000000000002-0.812600319649468
 0.37000000000000002-0.8122133345287916
 0.37500000000000002-0.8118781511152201
 0.38000000000000002-0.8115945110646018
 0.385000000000000023-0.8113621573211925
 0.390000000000000024-0.8111808341112298
 0.395000000000000024-0.8110502869365394
 0.4 -0.8109702625681743

Problem 12.13 diferansiyel denklem çözüm örneği, 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

Şu ana kadarki örneklerimizin hepsi konsol programı olarak verildi. simdi grafik (applet) çıktısına da bir örnek olmak üzere ninci dereceden denklemin köklerini hesaplayan [kokN.java](#) programını verelim.

Problem 12.14 ninici dereceden polinomun köklerini bulma, kokN.java programı

```
import java.util.*;  
import java.awt.*;
```

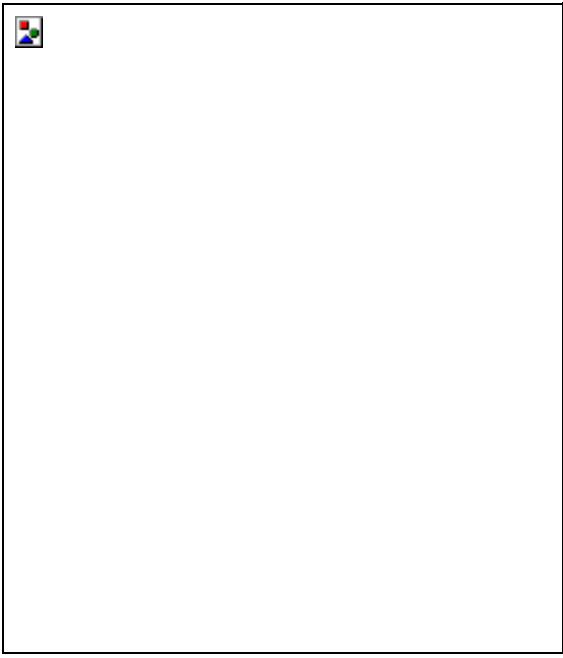
```

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("ninci 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 programın sonucu :

12005.JPG



Şekil 12.5 n inci dereceden polinomun köklerini bulan kokN programı, StringTokenizer ve Matrix sınıf örneği

12.5 ALIŞTIRMALAR

1. $f(x)=x*x*x-3$ denkleminin köklerini newton metodunu kullanarak çözünüz (3 ün küp kökünü hesaplayınız), Bunun için bir java programı yazıp çıktıyi 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 hesaplayınız), bunun için bir java programı yazıp çıktıyi hesaplatiniz.
4. $f(x)=\sin(x)$ denkleminin 0 ila pi arasındaki integralini hesaplayınız. Bunun için bir java programı yazıp çıktıyi hesaplatiniz.
5. $f(x)=\sin(x)$ denkleminin pi noktasındaki türevini alınız. Bunun için bir java programı yazıp çıktıyi hesaplatiniz.
6. $f(x,y)=x*x+y*y-R*R$ ($R=\text{sabit}$) denkleminin 0 ile 1 arasındaki türevini hesaplayınız.
7. $df(x)/dx=-f(x)$ denkleminin $x=1$ den 2 ye kadar çözümünü yapınız, $f(0)=1$, Bunun için bir java programı yazıp çıktıyi hesaplatiniz.

BÖLÜM 13 DATABASE (VERİ TABANI) PROGRAMLAMASINA GİRİŞ

13.1 TEMEL KAVRAMLAR

Database programlarına Java dili üzerinden erişmek için java JDBC API kullanılır. JDBC bize herhangi bir Data Base (veri Tabanı) yapısına java üzerinden ulaşmamızı ve Veri tabanı dili olan SQL (Structured Query Language) dilinde direk olarak programlama yapabilmemizi sağlar. JDBC ye geçmeden önce SQL dilinin bazı temel yapılarına göz atalım. Burada göreceğimiz oldukça kompleks işlemleri yapabileceğimiz üçüncü kuşak bir bilgisayar dildidir. Bu kitabın konusu Veri tabanları olmadığı için biz fazla detaya girmeyeceğiz. Dileyen arkadaşlar Veri Tabanı ile ilgili kitaplarda SQL ile ilgili çok daha detaylı bilgi bulabilirler.

Örnek olarak aşağıdaki Veri tabanı dosyasının olduğunu kabul edelim :

13001.JPG

The screenshot shows the Microsoft Access application window titled "Microsoft Access - [KAHVE : Table]". The menu bar includes File, Edit, View, Insert, Format, Records, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main area displays a table named "KAHVE". The table has five columns: KAHVE_ISMI, NUMARA, FIYAT, SATIS, and TOPLAM. The data rows are:

KAHVE_ISMI	NUMARA	FIYAT	SATIS	TOPLAM
kolombiya	101	7.99	120	0
fransız	49	8.99	130	0
kafeinsiz fransız	51	9.12	110	0
espresso	150	9.99	110	0
brezilya	89	7.99	120	0
kafeinsiz brezilya	91	8.15	120	0
*				

Şekil 13.1 Kahve veri tabanı dosyasının access veri tabanı programında görünüşü

Bu dosyayı 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 tabanı tablosu adı **KAHVE** ve Tablo sütun isimleri KAHVE_ISMI, NUMARA, FIYAT, SATIS ve TOPLAM olarak tanımlanmıştır. Değişken isminden sonra SQL değişken türleri verilmistir. Deşken türlerinin isimleri değişik SQL dillerinde değişebilir.

Bu tablodan belli değerleri girmek için INSERT INTO deyimi kullanılır. Örneğin tablodaki ilk satır

INSERT INTO KAHVE

VALUES('Fransız',49,8.99,130,0)

SQL deyimiyle girilebilir.

Tablodan belli değerleri geri çekmek için ise SELECT deyimi kullanılır. SELECT deyiminin en genel şekli

SELECT [DISTINCT|ALL] {*| liste}
FROM [tablo ismi][, ikinci tablo ismi]....
[WHERE şart]
[GROUP BY kolon listesi]
[HAVING şart]

[ORDER BY kolon listesi] şeklindedir. SELECT ve FROM her zaman için kullanılır, diğerleri istenirse kullanılır. SELECT komutundan sonra çıktıda görmek istediğimiz kolon adları bulunur. Tablonun tüm kolonlarını görmek istersek * kullanabiliriz. FROM komutundan sonra seçilen tablonun (veya tabloların) isimleri yer alır. Tablo

isimleri ‘,’ ile ayrılır. ALL veya DISTINCT anahtarları eğer tekrar varsa bunların verilip verilmeyeceğini bildirir. DISTINCT secildiğinde tüm tekrarlar elimine edilir. ALL seçildiğinde tüm girdiler tekrar da olsa verilir. WHERE deyimi javadaki if deyimi gibidir. Şart oluşturur. WHERE ile birlikte kullanılan bazı mantık işlemleri şunlardır :

AND : ve

OR : veya

NOT : olumsuzluk

BETWEEN : arasında (genellikle AND ile birlikte BETWEEN 20 AND 30 gibi kullanılır)

IN : içinde

LIKE : gibi, benzer(karecter değişkenleri için eşittir yerine geçen bir yapıdır, kullanılırken bazı kısımlar boş bırakılabilir LIKE '%ali%' içinde ali geçen her şeyi kapsar.)

NULL : boş

ANY : herhangibiri

ALL : hepsi

EXIST : varsa

Rakamsal karşılaştırmalarda

= : eşittir

<> : eşit değildir

> : büyüktür

< : küçüktür

>= : büyüktür ve eşittir.

<= : küçüktür ve eşittir.

Kullanılabilir.

ORDER BY sıraya sokar ASC kelmesiyle birlikte kullanıldığından küçükten büyüğe doğru, DESC kelimesiyle birlikte kullanıldığından büyükten küçüğe doğru sıralar.

GROUP BY belli bir kolona göre veriyi tasnif edebilir. HAVING deyimi GROUP by deyimiyle birlikte kullanılır ve GROUP By terimi için ilave şart oluşturur.

Select deyiminin kaç örnekle açalım:

```
SELECT KAHVE_ISMI, FIYAT  
FROM KAHVE  
WHERE KAHVE_ISMI like "Fransiz"  
deyimi
```

13002.JPG

KAHVE Query : Select Query		
	KAHVE_ISMI	FIYAT
▶	Fransiz	8,99
*		

Şekil 13.2 Query sonuçlarının access veri tabanında görünümü

sonucunu verir.

```
SELECT KAHVE_ISMI, FIYAT  
FROM KAHVE  
WHERE KAHVE_ISMI like "*Fransiz*"
```

13003.JPG

Query1 : Select Query		
	KAHVE_ISMI	FIYAT
▶	fransiz	8,99
	fransiz kafeinsiz	9,12
*		

Şekil 13.3 Query sonuçlarının access veri tabanında görünümü

sonucunu verir.

```
SELECT KAHVE_ISMI, FIYAT
FROM KAHVE
WHERE FIYAT > 8.5 AND FIYAT < 9.0
```

13003.JPG

	KAHVE_ISMI	FIYAT
▶	Fransız	8.99
	Kolombiya kafe	8.99
*		

Şekil 13.3 SELECT..WHERE querry sonuçlarının access veri tabanında görünümü

Fiyatı 8.5 ile 9.0 arasında olan kahvelerin listesi sonucunu verecektir.

```
UPDATE KAHVE
SET TOPLAM=FIYAT*SATIS;
```

Komutu kullanarak TOPLAM kolonunun değerlerini hesaplayabiliz. Tablonun yeni hali :

13004.JPG

	KAHVE_ISMI	NUMARA	FIYAT	SATIS	TOPLAM
▶	kolombiya	101	7.99	120	959
	fransız	49	8.99	130	1169
	kafeinsiz fransız	51	9.12	110	1003
	espresso	150	9.99	110	1099
	brezilya	89	7.99	120	959
	kafeinsiz brezilya	91	8.15	120	978
*					

Şekil 13.4 UPDATE querry sonuçlarının access veri tabanında görünümü

Sonucunu verir.

SQL de direk olarak hesap yapabilen fonksiyonlar da mevcuttur. Örneğin SUM(kolon ismi) değerleri toplar, MAX, maksimum değeri verir, MIN, minimum değeri verir, COUNT toplam sütun sayısını sayar. Bu fonksiyonlara örnek olarak SUM fonksiyonunu kullanalım :

```
SELECT SUM(TOPLAM)
FROM KAHVE;
```

13005.JPG

	Expr1000
▶	5167

Şekil 13.5 querry sonuçlarının access veri tabanında görünümü

Veri tabanını sıraya koymak için SELECT ..ORDER BY komutunu kullanabiliriz
ASC küçükten büyüğe DESC büyükten küçüğe sıraya koyar. Örneğin veri tabanı tablomuz kahveyi KAHVE_ISMI'ne göre sıralamak istersek

```
SELECT *
FROM KAHVE
ORDER BY KAHVE_ISMI ASC;
```

SQL komutunu kullanabiliriz. Sonuç :

13006.JPG

Query3 : Select Query					
	KAHVE_ISMI	NUMARA	FIYAT	SATIS	TOPLAM
▶	brezilya	89	7.99	120	959
	espresso	150	9.99	110	1099
	fransız	49	8.99	130	1169
	kafeinsiz brezilya	91	8.15	120	978
	kafeinsiz fransız	51	9.12	110	1003
	kolombiya	101	7.99	120	959
*					

Şekil 13.6 SELECT..ORDERBY query sonuçlarının access veri tabanında görünümü

13.2 JAVA VE SQL BAĞLANTISI

Şimdi bu işlemleri direk olarak java programlama dili üzerinden nasıl yapacağımızı inceleyelim.

Java dilinde data base tanımlamak için önce database ile bir bağlantı kurulmalıdır. Burada örnek olarak MS Access ile Window ortamında nasıl bağlantı tanımlayıcağımızı step-step verelim

Window (95-98) ortamında access database ile bağlantı kurulması:

prog (veya başka bir)isimli bir access db programını aç ve save et
başlat,ayarlar,denetim masası,32 bit ODBC

User DSN sayfasından MS Access Database seç ve Add düğmesine bas

Microsoft Access Driver(*.mbd) seç, Son düğmesini tikla

data source name olarak prog (veya başka bir isim) seç

Description alanına JDBC - odbc database yaz, select düğmesine bas

prog DB dosyasının olduğu directory 'i seç

OK düğmesine bas, Advanced düğmesinden login ve password da tanımlayabilirsiniz.

Windows 200-NT ortamında access database ile bağlantı kurulması (benim windows 2000nim ingilizce, türkçe olursa win 98'e parel isimler olmalı)

prog (veya başka bir)isimli bir access db programını aç ve save et
start,settings,control panel,Administrative Tools, Data Sources

ODBC Data Source Administrator, User DSN sayfasından MS Access Database seç ve Add düğmesine bas

Microsoft Access Driver(*.mbd) seç, Son düğmesini tikla

data source name olarak prog (veya başka bir isim) seç

Description alanına JDBC - odbc database yaz, select düğmesine bas

prog DB dosyasının olduğu directory 'i seç

OK düğmesine bas, Advanced düğmesinden login ve password da tanımlayabilirsiniz.

Program 13.1 kahve veri tabanı dosyasını tanımlayan CreateKahve.java programı

```
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 {
        con = DriverManager.getConnection(url, "Login", "Password");
        stmt = con.createStatement();
        stmt.executeUpdate(createString);

        stmt.close();
        con.close();
    } catch(SQLException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }
}

```

Buradaki

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

deyimi java JdbcOdbcDriver ile bağlantı açmaktadır. Eğer varsa kendi database driver'ınızı kullanabilirsiniz. Bunun size getireceği en büyük avantaj daha önce yazılmış bir database programına erişmeyi mümkün kılmıştır. JDBC driverlerinin en son listesi <http://java.sun.com/products/jdbc/jdbc.drivers.html> sitesinden öğrenilebilir.

**String url = "jdbc:odbc:prog";
Connection con = DriverManager.getConnection(url, "Login", "Password");**

deyimi ise daha önce ismini prog olarak tanımladığımız (ismi siz database kaynak kotunun ismine göre tanımlayabilirsiniz) prog database'ine ulaşmamızı sağlamaktadır.

Veri tabanı sonuçlarını almak için ResultSet kullanılır. ResultSet metodunun başlıca alt metodları :

ResultSet executeQuery(String sql) throws SQLException : SQL deyimini çalıştırır ve sonuçlarını ResultSET tipi Object olarak aktarır.

int executeUpdate(String sql) throws SQLException : bir sonuç tablosu vermeyen SQL terimlerini çalıştırır. Komuttan etkilenen satır sayısını iletir.

Boolean execute(String sql) throws SQLException : genel maksatlı SQL deyimi çalıştırıcı

ResultSet getResultSet() Throws SQLException:

int getUpdateCount() throws SQLException :

Boolean getMoreResults() : throws SQLException :

Kahve verisini veri tabanına yazdırın ve select deyimi kullanarak okutan program :

Program 13.2 kahve veri tabanı dosyasına veri girişi sağlayan InsertKahve.java programı

```

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 çıktısı :

```

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

```

Aynı programın şimdi de applet olarak yazılmış bir eşdeğerine göz atalım.

Program 13.3 KahveApplet.java programı InsertKahve programının applet eşdeğерidir. programın thread olarak yazıldığına 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 fiyatları : ", 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 programın çıktısı, bir önceki programda olduğu gibidir.

13007.JPG



Şekil 13.7 KahveApplet.html programının Netscape browserdaki çıktısı

Statement interface'sinin alt metodlarına biraz daha detaylı bakalım. Statement SQL komutlarını veritabanı programına gönderen ve sonuçlarını geri çağırın ana sınıfı. SQL komut istemi çeşitli metodlar üzerinden yapılabilir

ResultSet executeQuery(String sqldeyimi) throws SQLException

Bu metod çalıştırıldığında tek bir SQL query çalıştırılır ve sonuçları ResultSet sınıfından geri alınır.

int executeUpdate(String sqldeyimi) throws SQLException

Bu metod çalıştırıldığında tek bir SQL Query çalıştırılır, geriye sonuç iletmez, sadece kaç satır verinin bu deyimden etkilendiğini belirtir.

boolean execute(String sqldeyimi) throws SQLException

Bu metod çalıştırıldığında genel bir SQL Query deyimi çalıştırılabilir. Bilhassa çalıştırılan SQL deyiminin ne yapabileceği tam olarak bilinmiyorsa veya değişken sonuçlar elde ediliyorsa kullanılabilir. Bu deyim sonucu iletilen veriler GetResultSet(),getUpdateCount() ve getMoreResults() metodları kullanılarak alınabilir.

ResultSet getResultSet() throws SQLException

En son çıkmış olan sonuçları iletir. Eğer hiç bir sonuç mevcut değilse null sonucu verir. Bir kere okunduktan sonra sonuçlar silinir.

int getUpdateCount() throws SQLException

Update, insert ve delete deyimlerinin sonuçlarında aç satırın etkilendiğini verir. Eğer SQL tarafından bir sonuç seti verilmişse veya herhangi bir veri değiştirilmemişse -1 değerini verir.

int getMoreResults() throws SQLException

Birden fazla sonuç veya update sayımı yapılan durumlarda bir sonraki değer setine geçer. Bu metod aynı zamanda daha önce açılmış olan tüm ResultSet deyimi değerlerini kapatır.

ResultSet sınıfı temel olarak veritabanı dosyasındaki değerleri okumak için kullanılır. SQL veri türleri çeşitli SQL veritabanlarının varlığından dolayı oldukça çeşitlilik gösterir. Temel olak kullanılan metodlar veoların SQL veri ilişkilikleri şunlardır :

```

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

```

kalın harfle yazılı olan SQL tipleri bu metod için en uygun tiplerdır. Metodlar SQL tablo kolonuna kolon ismi veya indeks numarası kullanarak ulaşabilirler. Örneğin getDouble metodu

```

double getDouble(int sütunindeksi) throws SQLException;
double getDouble(String sütunismi) throws SQLException;

```

şeklinde tanımlanmıştır.
GetObject metodu her türlü değişken için kullanılabilir.

ResultSetMetaData sınıfı ResultSet'te tanımlanan bilgi ile ilgili ilave sütun bilgisi aktarmak için kullanılır.

```

ResultSet rs=stmt.executeQuery("SELECT Toplam FROM KAHVE");
ResultSetMetaData rsmd=rs.getMetaData();

```

Şeklinde tanımlanır. Önemli metodları :

```

int getColumnCount() throws SQLException : kaç sütunluk bir set olduğunu tanımlar
String getColumnLabel(int column) throws SQLException :
String getColumnName(int column) throws SQLException : veri tabanı sütun ismini verir
String getTableName(int column) throws SQLException : veri tabanı tablosunun ismini verir
Booleans isReadOnly(int sütun) throws SQLException : sütunun sadece okunabilir olarak izni olup olmadığını verir.

```

Sıra ana kadar geliştirdiğimiz veri tabanı programları uygulamaya özel idi. Şimdi biraz daha genel bir uygulama oluşturulalım. Bu uygulamada swing JTable sınıfını kullanacağız. Bu programda oluşturduğumuz ilk sınıf QuerryTableModel sınıfı tabloya girmek için query sonuçlarını tablo giriş formatında hazırlamaktadır. Bu metodun bilgi işlerken Vector sınıfını kullandığına dikkatinizi çekeriz. Vector sınıfı hakkında daha detaylı bilgi için ilgili bölümę bakınız.

Program 13.4 QueryTableModel.java programı

```
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 başlatılamadı");
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 kapatılamadı");
e.printStackTrace();
}
} //closeDB sonu
}

```

İkinci sınıfımız (Tabloyu gösterme sınıfı) olan DataBaseTest programı ise girdi alanından SQL komutunu ve veritabanı bağlantısını okumakta ve SQL sonuçlarını tabloda vermektedir.

Program 13.5 DatabaseTest.java programı

```

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()
{

```

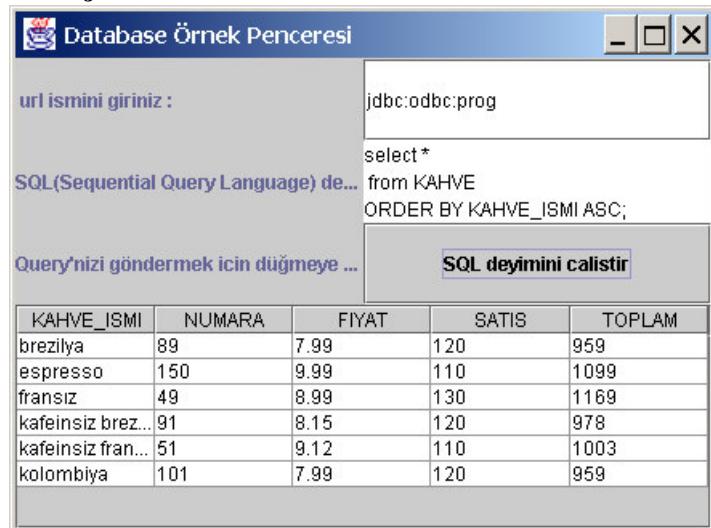
```

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 için düğmeye basınız:"));
 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



Şekil 13.8 DatabaseTest.java programının JFrame'de ve JTable kullanılarak oluşturulan çıktısı

13.3 ALIŞTIRMALAR

1. Program örneği database dosyasını girdi dosyası olarak kullanan ve içindeki sayıların ortalamasını alan H9O2.java programı : Bu programda sayılar dosyadan okunmakta, ortalaması hem querry dilinde (SQL) hemde java dilinde hesaplanmaktadır. Programı inceleyiniz.

Program 13.6 H9O2.java, database dosyasından alınan bilgilerle database dosyasının girdi dosyası olarak kullanılması

```
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.out.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.out.println(e.getMessage());
        } //catch
    } //main
} //class
```

program [H9O2.java](#) çıktısı :

```
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 = 2.484159873999995
metot 2 ortalamalar java dilinde hesaplandi
metot 2 ortalama : 2.484159873999995
=====

```

2. H9OD2_2000

prog.mdb access veri tabanında H9OD2_2000 veri tabanı tablosunu açınız. Bu dosyada x ve y ve y2 başlıklarında üç kolon yaratınız. x ve y kolonlarına

```

1   1
1   4
2   9
3   16
4   25
5   36

```

veri setini giriniz. H9OD2_2000.java programında veri tabanı doyasından okutup ekranda gösteriniz. Aynı zamanda y2 kolonuna x^2 değerlerini yazdırınız.

Program 13.7 H9O2_2000.java, database dosyasından alınan bilgilerle database dosyasının girdi dosyası olarak kullanılması

```

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.out.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  $y^2$  değerini veri tabanı dosyasına yazdır.
            int jj = stmt.executeUpdate(query2);
            System.out.println("jj"+"kolon değiştirildi");
        }
    }
}

```

```

stmt.close();
con.close();
}
catch (SQLException e)
{
System.err.println(e.getMessage());
} //catch
} //main
} //class

```

Program çıktısı :

```

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 veritabanı dosyası:

```

Program 13.8 H9O2_2000a.java, database dosyasından alınan bilgilerle database dosyasının girdi dosyası olarak kullanılması, sonuçların 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 sonuçları");
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

	x	y	y2
►	1	1	1
	2	4	4
	3	9	9
	4	16	16
	5	25	25
	6	36	36
	7	49	49
*	0	0	0

Şekil 13.9 H9O2_2000a.java çıktısı, database dosyasından alınan bilgilerle database dosyasının girdi dosyası olarak kullanılması, çıktı ortamı olarak tablo kullanan bir alternatif

Program 13.10 H9O2_2000b.java, database dosyasından alınan bilgilerle database dosyasının girdi dosyası olarak kullanılması, çıktı ortamı olarak QueryTableModel sınıfını kullanarak oluşturduğu tabloyu kullanan bir alternatif

```
import java.sql.*;  
import javax.swing.*;  
import java.awt.*;  
import QueryTableModel;  
// Vector (Querytable model'da ) kullanıyor  
public class H9OD2_2000b extends JFrame  
{  
    public H9OD2_2000b()  
    {  
        super("Data base sonuçları");  
        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

x	y	y2
1	1	1
2	4	4
3	9	9
4	16	16
5	25	25
6	36	36
7	49	49

Şekil 13.10 H9O2_2000b.java, database dosyasından alınan bilgilerle database dosyasının girdi dosyası olarak kullanılması, çıktı ortamı olarak QueryTableModel sınıfını kullanarak oluşturduğu tabloyu kullanan bir alternatif

=====

3. Daha önce de kullandığımız prog veritabanında e isimli bir tablo oluşturunuz, bu tablodaki e sütununa gerçek sayı veri giriniz. Diğer bir tablo olarak e tablosunda e sütununu oluşturunuz, bu tabloya da gerçek sayı değerler giriniz. İki veri tabanı tablosundan değerleri okuyup, bileşim kümelerini hesaplayınız ve yarattığınız ed tablosuna ed değerleri olarak yazdırınız.

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+="bileşim 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 sınıflarını
// Kullanmaktadır. Bileşim ve kesişim kümelerini hesaplar
// Aynı zamanda StringTokenizer sınıfını kullanır.
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 bilesimeksikesim(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;
}
}

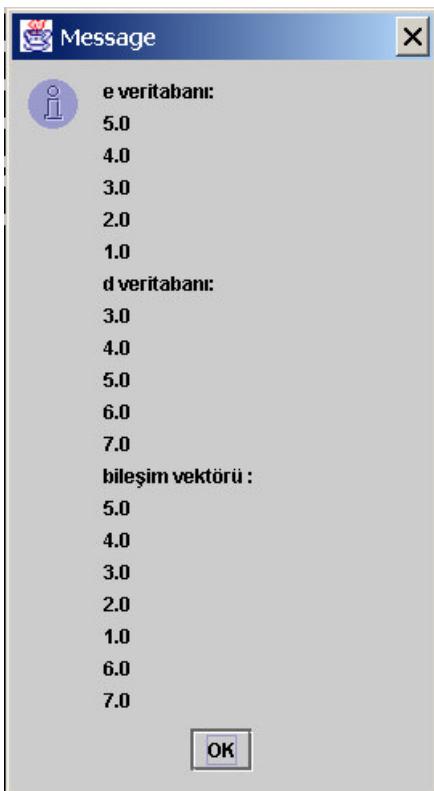
```

13011.JPG

H10OD1_2000 : Table		
	ID	ed
▶	1	5
	2	4
	3	3
	4	2
	5	1
	6	6
	7	7
*	(AutoNumber)	0

Sekil 13.11 H10OD1_2000.java, programının oluşturduğu H10OD1_2000 veritabanı dosyası

13012.JPG



Şekil 13.12 H10OD1_2000.java, programının sonuçlarının JOptionPane çıktıları olarak görüntülenmesi

14. Daha önce de kullandığımız prog veritabanında e isimli bir tablo oluşturunuz, bu tablodaki x sütununa gerçek sayı veri giriniz. Diğer bir tablo olarak e tablosunda x sütununu oluşturunuz, bu tabloya da gerçek sayı değerler giriniz. İki veri tabanı tablosundan değerleri okuyup, bileşim kümесini hesaplayınız ve yarattığınız H10OD1_2000 tablosuna x değerleri olarak yazdırınız. Proseste boyutlu değişkenleri vektör üzerinden taşıyınız

Program 13.12 H10OD1_2000.java

Bu program Gebze Yüksek teknoloji enstitüsü Java programlama dili dersi, 2000 Güz dönemi öğrencilerinden Deniz KAYA tarafından yazılmıştır.

```
// Deniz KAYA
// Hafta 10 Ödevi
// Tablolardan vektore, vektor bileşimininden tabloya
// yazdırma işlemi.
```

```
import java.io.*;
import java.util.*;
import java.sql.*;
```

```
class SetA
{
    public Vector tablodanIntYukle(String d)
    {
        // ismi d deðiskeni ile gelen tablodan
        // Rakamları Vektor İcine 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 değerlerinin yükliyor.
        Vector list2 = A.tablodanIntYukle("e"); // list2 'ye (e) tablosunun değerlerinin yükliyor.
        Vector bilesim = A.bilesim(list1,list2); // listeleri birleştirir.
        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

```

15. Program 4 de H10OD1_2000 veri tabanı dosyasına yazılan veriyi okuyunuz. Binary Tree yapılarını kullanarak sıraya koynuz ve H10OD2_2000 dosyasına yazdırınız.

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 sagdal_node_sirala(){sagdal_node_sirala();}
  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 öğrencilerinden Deniz KAYA tarafından yazılmıştır.

```

// Deniz KAYA
// H10OD2_2000
// Tablodan-Treeye, Tree-den tabloya(recursive) yazdırma işlemi.

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 sıralı bir sekilde tabloya atar.
// Tabloya yazdırma için veriyazdır metodunu kullanır.

public void soldal_node_sagdal_siralayici(TreeNode node)
{

```

```

if(node==null) return;
soldal_node_sagdal_siralayici(node.sol);
// veriyi tabloya yazdır.
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 sayı 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 büyüğe sıralı bir şekilde ekrana yazdırır.
agac.soldal_node_sagdal_sirala();

stmt.close();
con.close();

// H10OD2_2000 tablosuna verileri küçükten büyüğe sıralı yazdırır.
h.soldal_node_sagdal_siralayici(agac.kok);
System.out.println("\n H10OD2_2000 tablosuna veriler yazıldı.");
}
catch (SQLException e)
{
System.err.println(e.getMessage());
} //catch
}
}

```

16. Bu alıştırmada bir adres defteri oluşturacağız. Önce adres defterinde tanımlanan alanları verelim

ID anahtar alan
isim
soyisim
adres

ilce
 il
 postakodu
 ulke
 emailadresi
 telefonev
 telefonis
 telefoncep
 faks
 webadresi
 evadresi
 alanları tanımlanacaktır.

Temel access veri tabanı tanımlamasını access'de yaptık. veri tabanı dosyasının ismi adres olarak tanımlandı.
 Veri tabanının görünümü:

13013.JPG

adres : Table									
	ID	isim	soyisim	adres	ilce	il	postakodu	ulke	emailadresi
▶	4	Turhan	Çoban	TUBITAK Marm Gebze	Kocaeli	41470	Türkiye	Turhan.C...	
	5	Meral	Çoban	TUBITAK Marm Gebze	Kocaeli	41470	Türkiye	Meral.C...	
*	(AutoNumber)								

Şekil 13.13 adres.mpg veritabanı tablosu adres

şeklindedir. Bu programı oluşturabilmek için bir dizi program tanım yapıldı. Bunlar adresdefteri.java, adresgir.java, adresbul.java, adressil, adresdegistir.java, yardım.java, adrespanel.java ve ciktipaneli.java programlarıdır. Bu programların kodlarını verelim :

Program 13.15 adresdefteri.java

```

// adresdefteri ana programı
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 bağlantısını oluştur
    try {
      url = "jdbc:odbc:adres";
      Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
      connect = DriverManager.getConnection( url );
      output.append( "Bağlantı sağlandı\n" );
    }
  }
}

```

```

        catch ( ClassNotFoundException cnfex ) {
            cnfex.printStackTrace();
            output.append( "ba\u011flant\u00f6 sa\u011flanamad\u00f6\n" +
                cnfex.toString() );
        }
        catch ( SQLException sqlex ) {
            sqlex.printStackTrace();
            output.append( "ba\u011flant\u00f6 sa\u011flanamad\u00f6\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 gönderiliriyor: " +
connection.nativeSQL( query )
+ "\n");
int result = statement.executeUpdate( query );

if ( result == 1 )
    output.append( "\nQuery başarıyla çalıştı\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 azından isim ve soy isim girin ve sonra " +
"ekle dugmesine basın\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.isim.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 gönderiliriyor: " +
                              connection.nativeSQL( query ) +
                              "\n" );
                ResultSet rs = statement.executeQuery( query );
                display( rs );
                output.append( "\nisim + soyisim Query'si başarıyla sonuçlandı\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 gönderiliriyor: " +
                              connection.nativeSQL( query ) +
                              "\n" );
                ResultSet rs = statement.executeQuery( query );
                display( rs );
                output.append( "\nisim Query'si başarıyla sonuçlandı\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 gönderiliriyor: " +
                              connection.nativeSQL( query ) +
                              "\n" );
                ResultSet rs = statement.executeQuery( query );
                display( rs );
                output.append( "\nQuery başarıyla sonuçlandı\n" );
                statement.close();
            }
        }
    }
}

```

```

        else
            fields.soyisim.setText(
                "Burada soyismi girin ve bul düğmesine basın" );
    }
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( "\nKayıt bulunamadı\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( "\nkayıt başarıyla değiştirildi \n" );
            else {
                output.append( "\nkayıt değiştirme işlemi başarılımadı\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 kayıtları değiştirebilirsiniz\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 :", "iş 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( "DEĞİŞTİR" );
updateName.addActionListener(
    new adresdegistir( c, s, t ) );
add( updateName );

clear = new JButton( "SİL" );
clear.addActionListener( new adressil( s ) );
add( clear );

help = new JButton( "YARDIM" );
help.addActionListener( new yardim( t ) );
add( help );
}

}

```

Program 13.22 yardım.java

```

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

public class yardım implements ActionListener {
    private JTextArea output;

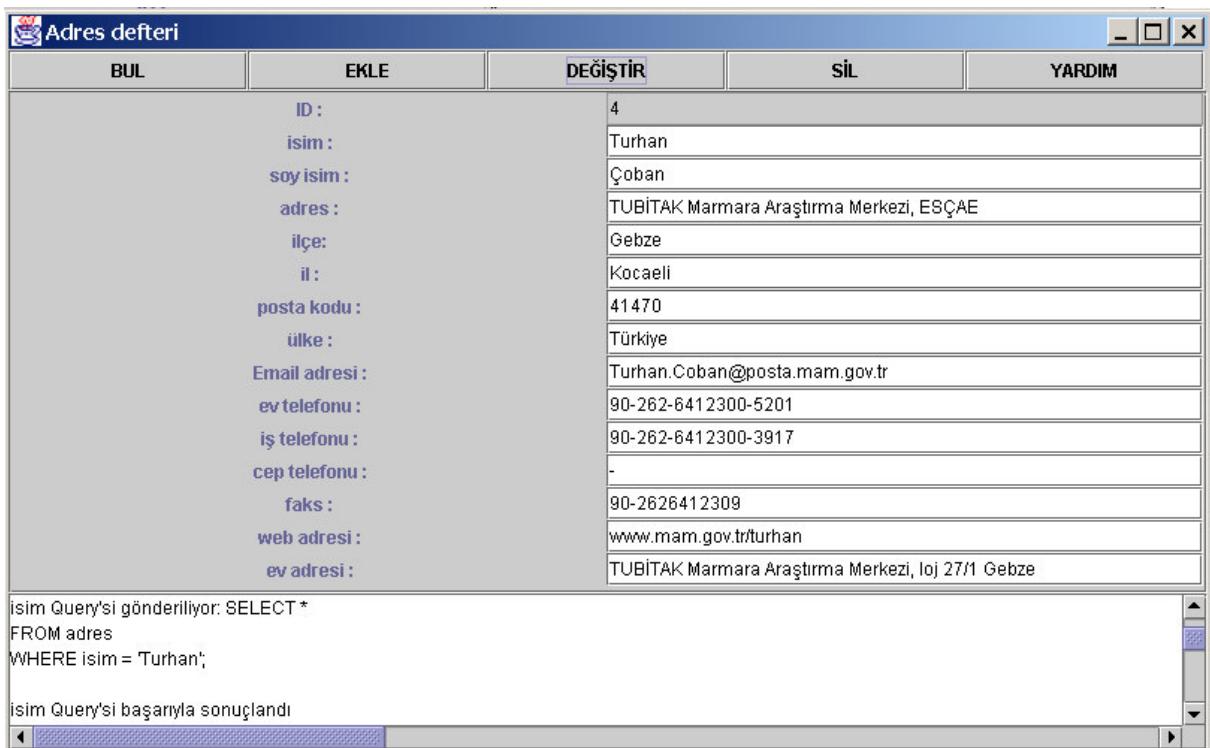
    public yardım( JTextArea o )
    {
        output = o;
    }

    public void actionPerformed( ActionEvent e )
    {
        output.append( "\nbir kayıt bulmak için isim ve/veya soyisim alanına yazıp BUL düğmesine basınız\n" +
                      "Yeni bir kayıt girmek için EKLE düğmesine basınız\n" +
                      "Kayıttaki değerleri değiştirmek için DEĞİŞTİR düğmesine basınız\n" +
                      "kayıdı silmek için SİL düğmesine basınız\n" );
    }
}

```

bu program setinin ana programı olan adresdefteri programını çalıştırduğumızda adres defterimiz açılır. Adres defterinde verilen bul, değiştir, ekle, sil gibi alanları kullanarak adres defterimizi kullanabiliriz. Belli bir komut verdiğimizde SQL Querry dilinde oluşan ve gönderilen komutlar alt pencereden izlenebilir. Ayrıca herhangi bir hata oluştu ise de bu pencerede hata mesajı verilecektir.

13014.JPG



Şekil 13.14 adres.mpg veritabanı tablosunu kullanan adresdefteri.java programı Jframe çıktısı.

BÖLÜM 14 JAVA ANADİL (NATIVE LANGUAGE) PROGRAMLAMASINA GİRİŞ C++ ve C PROGRAMLAMA DİLLERİNİN JAVA İLE BİRLİKTE KULLANIMI

14.1 TEMEL KAVRAMLAR

Java programlama dili C dilinde yazılmıştır. Bu yüzden C'den javaya geçiş oldukça kolay bir şekilde yapılabilir. Bir örnekle birden fazla programlama dilinin nasıl kullanılabildiğine bakmadan önce, bir temel kavramı iyi anlamamız gereklidir. Java güvenlik açısından maksimum güvenliği verebilen bir dildir. Öte taraftan C maksimum hızlara ulaşmayı sağlayabilen bir dildir. C programlarını java programlarıyla birlikte kullanmanızın temel nedeni hız gereken yerlerde bu hızı sağlamaktır. Ancak hız sağlarken güvenlikten olan kaybımızı her zaman hesaba katmamız ve ikisi arasında bir denge bulmamız biz programcılara bırakılmıştır. Bu yüzden iki dilli programlamada oldukça dikkatli bir şekilde çalışma gereksinimi oluşturur. Bu konu C++ dilini iyi bilen programcılar içindir. Burada C programlamasına girmeyeceğiz. Eğer bu dili bilmiyorsanız, bu konuya başlamadan önce öğrenmenizi veya java anadil programlamasını hiç kullanmamanızı tavsiye ederim. C++ dilini kullanabilmek için, hele iki dili bir arada kullanabilmek için iyi bir programcı olmanız gerekiyor, eğer kendinizi güveniyorsanız, bundan sonraki kısma geçebilirsiniz.

14.2 Örnek probleme step step iki dilli program kodunun hazırlanması ve çalıştırılması

Çok dilli programlama birden fazla basamağı gerektiren bir prosesdir. Bir örnek problemler bu basamakları inceleyelim :

C++ dili ile java dilinin beraber çalışacağı bir örnek oluşturalım. İlk step olarak java dilindeki programımızı yazalım :

Program 14.1 JavaCpp.java programı

```
class JavaCpp
{
    public native void cdenaktar();
    static
    {
        System.loadLibrary("jcpp");
    }

    public static void main(String[] args)
    {
        new JavaCpp().cdenaktar();
    }
}
```

JavaCpp programında iki temel metod görüyoruz. Birinci metod cdenaktar() metodudur. Bu metod biraz sonra C++ dilinde oluşturacağımız fonksiyonu Java'nın bir parçası olarak çağırır. İkinci metod ise main metodudur. Cdenaktar metoduna bakarken bunun java metodu olmadığını belirtlen native sözcüğü yer almaktadır. Metodun türü void'dır, yani hiçbir şey aktarılmıyacaktır. Programımızı, herhangi bir java programında olduğu gibi önce **java JavaCpp.java**

Komutunu kullanarak derleriz. İkinci stepte javah komutunu kullanarak

javah JavaCpp

JavaCpp.h dosyasını oluştururuz. Bu dosyanın içeriği :

Program 14.2 JavaCpp.h arabağlantı programı

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class JavaCpp */

#ifndef _Included_JavaCpp
#define _Included_JavaCpp
#endif /* __cplusplus
```

```

extern "C" {
#endif
/*
 * Class: JavaCpp
 * Method: cdenaktar
 * Signature: ()V
 */
JNIEXPORT void JNICALL Java_JavaCpp_cdenaktar
(JNIEnv *, jobject);

#ifndef __cplusplus
}
#endif
#endif

```

şeklinde olacaktır. Bundan sonraki stepte C++ dilinde aşağıdaki programı oluşturalım :

Program 14.3 JavaCpp_c.cpp C++ programı

```

#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 kullanıyoruz ";
    return;
}

```

c++ programını derlemek için elbette bir derleyiciye ihtiyaç vardır. Biz burada MS Visual C++ versiyon 6 derleyicisini kullandık, herhangi bir dll dosyası hazırlayan derleyici (MS Windows için) aynı işi görebilir. Bu derleyici ile derleme için ya direkt olarak dos'ta

```
cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LD %JavaCpp_c.cpp -Fejcpp.dll
```

komutunu kullanırız, yada bu uzun komutu her sefer yazmamak için bir isim.bat dosyası yaratıp bunu kullanabiliriz. Örneğin hazırladığımız pcomp.bat dosyası

Program 14.4 pcomp.bat MSdos programı

```
cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LD %1.cpp -Fe%2.dll
```

kullanılarak yukarıdaki komut
pcomp JavaCpp_c jcpp

şeklinde verilebilir. Bu işlemleri yaptığımızda programımız çalışmaya hazır olacaktır. Programı çalıştırmak için :

java JavaCpp

komutunu kullanmamız yeterli olacaktır. Program çıktısı :

Java-C++ programlama dillerini birlikte kullanıyoruz

Şeklinde olacaktır. Burada derlemenin jcpp.dll dosyasına yapıldığını hatırlatalım, ve bu dosya java programı üzerinden C++ metodunu çalıştırırken kullandığımız ana programdı.

14.2 DEĞİŞKENLERİN VE JAVA NESLERİNİN AKTARILMASI

İlk programımızda java ve C++ arasında herhangi bir değişken aktarılması olmamış, sadece void türü fonksiyon kullanılarak belli bir yazı yazan C++ fonksiyonu çağrılmıştır. İkinci örneğimizde java programımızdan String türü değişkeni C++ programlama diline aktaracak ve kullanacağız.

İlk java programımız Showmessage.java. bu programda ekrandan girilen bir string, yahut ekrandan girilen bir string değeri yoksa programın içinde tanımlanan bir stringle birlikte C++ programına aktarılıp burada ekrana yazılmaktadır.

Program 14.5 ShowMessage.java programı

```
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 yazıldı");  
        else  
            app.ShowMessage(args[0]);  
    }  
}
```

bu program **javac ShowMessage.java** ve **javadoc ShowMessage** komutlarıyla işlendikten sonra

Program 14.6 MsgImpl.cpp programı

```
//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 çıktı stringi : %s\n",msg);  
env->ReleaseStringUTFChars(jMsg, msg);  
}
```

c++ dilindeki MsgImpl.cpp programı hazırlanmış ve
üstte hazırladığımız pcomp.bat yardımıyla

pcomp MsgImpl MsgI

komutu kullanılarak derlenmiştir. Program çıktısı

java ShowMessage “Merhaba Dünya” komutu kullanılarak :

JNI çıktı stringi : Merhaba Dünya

veya

java ShowMessage komutu kullanılarak

JNI çıktı stringi : Bu mesaj javadan C++ ya gönderildi ve orada yazdırıldı

Çıktısı alınabilir. Programda veri aktarımı yapmak amacıyla java tarafında metod tanımı yapılırken

private native void ShowMessage(String msg);

tanımı verilmiştir. Buradaki String msg metodun dolayısıyla C++ fonksiyonunun girdisini teşkil etmektedir. C++ tarafında mesajı alırken,

```
Java_ShowMessage_ShowMessage(JNIEnv* env,  
jobject, jstring jMsg)  
{  
const char* msg=env->GetStringUTFChars(jMsg,0);
```

jstring jMsg tanımıyla aktarılmış ve const char* msg=env->GetStringUTFChars(jMsg,0); tanımlı c değişkenine dönüştürülmüştür.

Javada Stringler unicode (16 bit) olarak tanımlanmıştır. C stringleri ise 8 bitlik ascii char boyutlu değişkeni olarak tanımlanır. Bu yüzden string değişkenlerini javadan cye veya cden javaya aktarırken dönüşümler yapmak gereklidir. Aynı zamanda java string değişkenleri length (boyut) değişkenini de bünyelerinde taşırlar. Cde ise boyut tanımlanmamıştır. Ayrı bir değişken üzerinden tanımlanması gerekebilir. Bu dönüşümleri yapmak için bir dizi metod tanımlanmıştır.

Bu tanımlaların tamamı şöyledir :

```
GetStringChars  
GetStringLength  
GetStringUTFChars  
GetStringUTFLength  
NewString  
NewStringUTF  
ReleaseStringChars  
ReleaseStringUTFChars
```

Buradaki ikinci örneğimizde java programında c++ dilindeki bir fonksiyonun (metod) yaptığı hesap sonuçlarını aktaracağız. Programımız java'da çağrılan bir C++ kare metodunu kullanarak java dilinde girilen sayının karesini hesaplayacaktır.

Program 14.7 karesinifi.java programı

```
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 programı javac karesinifi.java deyimiyle derleyip, javah karesinifi deyimiyle anadil kütüphanesine çevirirsek

Program 14.8 karesinifi.h programı

```
/* 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 programındaki
private native double kare(double x);
metodunun c++ anadil (Native) karşılığının
JNIEXPORT jdouble JNICALL Java_karesinifi_kare
(JNIEnv *, jobject, jdouble);
olduğunu göstermektedir. Şimdi kare metodunun aslını C++ dilinde yazarsak :
```

Program 14.9 karesinifi.cpp C++ programı

```
#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** şeklini aldığığini görüyoruz. Double değişkeni de yerini jdouble değişkenine bırakmış durumda. Java anadilinde programları yazarken metod isimleri **Java_sınıfismi_metodismi** Şeklini alır. Basit java değişkenlerinin Anadil karşılıkları ise

Tablo 14.1 Java değişken tipleri ve ana dil (native) eşdeğerleri

Java değişken tipi	Anadil (Native) değişken eşdeğeri	Değişken bit değeri
Boolean	jboolean	8, işaretsız
Byte	jbyte	8
Char	jchar	16, işaretsız
Short	jshort	16
Int	jint	32
Long	jlong	64
Float	jfloat	32
Double	jdouble	64
Object	jobject	
Void	void	Yok

Şeklindedir. C++ programı
 pcomp karesinifi kareI
 deyimiyle veya açık yazılışıyla
 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
 şeklinde olacaktır.

Şimdi de bu işlemin tam tersini inceleyelim. Java'da yazılmış olan kare metodunu C++ ve C dillerinde çağıralım.

Java programımız karesinifi1.java aşağıdaki gibi tanımlanmıştır.

Program 14.10 karesinifi1.java java programı

```
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");
}
}

```

sınıfın alt sınıfı olan kare'de girilen sayının karesi hesaplanmaktadır. NativeMethod metodu da C (veya c++) dilleriyle bağlantı sağlamaktadır. Metodumuzu yine java karesinifi1.java ile derledikten sonra javah karesinifi1 komutuyla karesinifi1.h dosyasını oluştururuz. Bu dosya :

Program 14.11 karesinifi1.h programı

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class karesinifi1 */

#ifndef _Included_karesinifi1
#define _Included_karesinifi1
#ifndef __cplusplus
extern "C" {
#endif
/*
 * Class:     karesinifi1
 * Method:    nativeMethod
 * Signature: ()V
 */
JNIEXPORT void JNICALL Java_karesinifi1_nativeMethod
(JNIEnv *, jobject);

#endif /* __cplusplus
#endif
#endif

```

anadıl tarafında program yazılrken bize yardımcı olabilecek bir ara java programı javap de mevcuttur. Bu işlemi yapmak zorunda değiliz. Yapma sebebimizi c++ dili programını inceleyince daha iyi anlayacaksınız.

javap -s -p karesinifi1

Program 14.12 javap komutuyla oluşturulan 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 */
}

```

şimdi c++ koduna bakalım :

Program 14.13 karesinifi1.cpp programı

```
#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 sınıfının (karesinifi1) aktarılması **jobject obj** değişkeni ve **jclass cls=env->GetObjectClass(obj);** üzerinden yapılır. Burada sınıf adresi cls değişkenine yüklenmiş olur. Alt metod karenin tanımı için **jmethodID mid=env->GetMethodID(cls,"kare","(D)D");** tanımı kullanılmıştır. Buradaki "(D)D" tanımı program 14.12 de bize verilmiştir. Anlamı java programındaki kare metodunun double değişken girişi olduğu ve çıkışında double değişken istedigidir. "kare" deyimiyle metodun javadaki ismi verilmiştir. Cls ise üstte tanımladığımız karesinifi1 java sınıfının adresine belirtmektedir.

double y=env->CallDoubleMethod(obj,mid,x);
deyimiyle de metodu çağrıp sonuçlarını y değişkenine yüklemektedir.
Burada kullandığımız D işaretti java değişken türünün double olduğunu belirtiyordu. Java değişken tiplerinin işaret karşılıkları Tablo 14.2 de verilmiştir.

Tablo 14.2 Java tiplerinin eşdeğer işaretleri

İşaret	Java değişken tipi
Z	boolean
B	Byte
C	Char
S	Short
I	Int
J	Long
F	Float
D	Double
L sınıf ismi	sınıf ismi
[Değişken türü	Değişken türü[]

Bu programda sınıfı çağrırmak için GetObjectClass metodunu, metod tanımlamak içinde GetMethodID metodunu kullandık. CallDoubleMethod metodunu ile de metodu çalıştırıldı. Paralel işlemleri yapmak için mevcut olan metod listesi ve anlamları şöyledir :

GetObjectClass : sınıf adresini tanımlar
GetMethodID : dinamik metod adresini tanımlar
GetStaticMethodID : statik metod adresini tanımlar
Call<değişkentürü>Method : çeşitli metodları çağırır
 CallBooleanMethod
 CallByteMethod
 CallCharMethod
 CallDoubleMethod
 CallFloatMethod
 CallIntMethod
 CallLongMethod
 CallObjectMethod
 CallShortMethod
 CallVoidMethod
CallStatic< değişkentürü >Method : statik metodları çağırır, değişken türleri olarak yukarıdakinin aynı

türleri kabul eder.

CallNonvirtual<değişkentürü>Method

Call<değişkentürü>MethodV

Call<değişkentürü>MethodA

Aynı programın c dilinde yazılması küçük farklılıklar taşıır. C dilideki program :

Program 14.14 karesinifi1.c programı

```
#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 görüldüğü gibi C++ daki :

jclass cls=env->GetObjectClass(obj);

komutu

jclass cls=(*env)->GetObjectClass(env,obj);

halini almıştır.

jmethodID mid=env->GetMethodID(cls,"kare","(D)D");

komutu

jmethodID mid=(*env)->GetMethodID(env,cls,"kare","(D)D");

halini almıştır.

double y=env->CallDoubleMethod(obj,mid,x);

komutu ise

double y=(*env)->CallDoubleMethod(env,obj,mid,x);

halini almıştır. Sonuç olarak env yerine referans değişkeni (*env) gelmektedir ve env referans değişkeni aynı zamanda metod değişkeni olarak çağrılmaktadır. Bunun dışında işlemler arasında bir fark yoktur.

Program **java karesinifi1** deyiimile çalıştırıldığında

2nin karesi = 4

sonucunu alırız.

İkinci bir sınıf ulaşım örneği olarak UseObjects programını verelim. Burada sınıf, MyJavaClass içinde bir int değişken ve bir void metod mevcuttur. İkinci bir metod olan UseObjects metodу kullanılarak native metod changeObject'e ulaşılmakta ve bu metod c++ programında kullanılarak sonuçlar hesaplanmaktadır.

Program 14.15 UseObjects.java programı

```
//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 kullanılarak oluşturulan UseObjects.h programı

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class UseObjects */

#ifndef _Included_UseObjects
#define _Included_UseObjects
#ifndef __cplusplus
extern "C" {
#endif
/*
 * Class:  UseObjects
 * Method: changeObject
 * Signature: (LMyJavaClass;)V
 */
JNIEXPORT void JNICALL Java_UseObjects_changeObject
(JNIEnv *, jobject, jobject);

#ifndef __cplusplus
}
#endif
#endif

```

Program 14.17 UseObjImp.cpp programı

```

#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 tanımlanan MyJavaClass sınıfı bir int değişken, aValue, ve bir metod, divByTwo , içermektedir. Metod int de verilen değeri 2 ye bölmekte ve sonucu aktarmaktadır. Bu programda int sınıf alt değişkenini tanımlamak için

```

jfieldID fid = env->GetFieldID(cls, "aValue", "I");
deyimini kullandık. Void aValue metodu
jmethodID mid = env->GetMethodID(cls, "divByTwo", "()V");
deyimiyle tanımlandı. Ve
env->SetIntField(obj, fid, 6);
tanımıyla 6 değeri atandı.

```

Sınıfların alt değişkenlerine değer atamak (ve okumak)için

- GetFieldID
- GetStaticFieldID
- Get< değişkentürü >Field
 - GetBooleanField
 - GetByteField

```

GetCharField
GetDoubleField
GetFloatField
GetIntField
GetLongField
GetObjectField
GetShortField
Set< değişkentürü >Field
GetStatic< değişkentürü >Field
SetStatic< değişkentürü >Field

```

Metodları mevcuttur.

Şimdi de Çok boyutlu değişkenleri nasıl kullanabildiğimizi inceleyelim :

Program 14.18 IntArray.java programı

```

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 programı

```

#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 tanımlanan java programında boyutlu değişken arr tanımlanmış, ve boyutlu değişkenin toplamı c programında hesaplanmıştır. C metodunda boyutlu değişkeni tanımlamak için jintArray türü arr değişkeni kullanılmıştır. Boyutlu değişkenin boyutu jsize tipi len değişkenine

jsize len = (*env)->GetArrayLength(env, arr);

deyi̇mle aktarılmıştır.

arr değişkeninin değeri jint tipi body boyutlu değişkene, indeksi 0dan başlamak üzere

jint *body = (*env)->GetIntArrayElements(env, arr, 0);

metoduyla aktarılır.

arr boyutlu değişkeninin değeri (değerler değişmiş olabilir) tekrar kullanılabilmek üzere

(*env)->ReleaseIntArrayElements(env, arr, body, 0);

deyimi kullanılarak serbest bırakılır (javaya gönderilir). Bu işlemlerin sonucunda java programı :
sum = 45

sonucunu verecektir. JNI boyutlu değişken çağrıma fonksiyonları(metodları) şöyledir :

```
GetArrayLength
Get<değişkentürü>ArrayElements
    GetBooleanArrayElements
    GetByteArrayElements
    GetCharArrayElements
    GetDoubleArrayElements
    GetFloatArrayElements
    GetIntArrayElements
    GetLongArrayElements
    GetShortArrayElements
Release<değişkentürü>ArrayElements
Get<değişkentürü>ArrayRegion
Set<değişkentürü>ArrayRegion
GetObjectArrayElement
SetObjectArrayElement
```

Java dilinde tanımlanmış, fakat c dilinde direkt tanımlanmamış işlemlerden birisi de hata analiz işlemidir (Catch-throw exception). Ana dil programlaması kullanırken, bu işlemi de yapmak mümkündür. Aşağıdaki CatchThrow program setinde hata sisteminin kullanılmasını görüyoruz.

Program 14.20 CatchThrow.java programı

```
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 programı

```
#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 aslında c'de hata konusunda gerçekten bir şey yapmıyoruz,
sadece bir mesaj iletiyoruz, hata değişkenini 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 programı çalıştırdığımızda :

```

C:\co\java\prog> java CatchThrow
Exception in thread "main" java.lang.NullPointerException: thrown in CatchThrow.
callback
        at CatchThrow.callback(CatchThrow.java:4)
        at CatchThrow.catchThrow(Native Method)
        at CatchThrow.main(CatchThrow.java:9)
In Java:
        java.lang.IllegalArgumentException: thrown from C code

```

sonucunu alırız. Buradaki

In Java:

java.lang.IllegalArgumentException: thrown from C code
mesajı c tarafından oluşturulup gönderilmiştir.

JNI Hata fonksiyonları şunlardır :

```

ExceptionClear
ExceptionDescribe
ExceptionOccurred

```

Burada iki dilli programlama kavramına bir giriş yaptık. Buradaki kavramlar size biraz zor geldiyse lütfen umutsuzluğa kapılmayın, java ve C dillerini çok iyi öğrendikten sonra geri dönün. İki dilli programlama her zaman programlama dünyasının en zor işlerindendir, ayrıca jbuilder, Microsoft j++ gibi paketler java anadil (native) programlamasında işleri kolaylaştıran otomatik kod oluşturma sisyemleri sunarlar, bu tür programlama araçlarıyla işinizi daha kolay hale getirebilirsiniz.

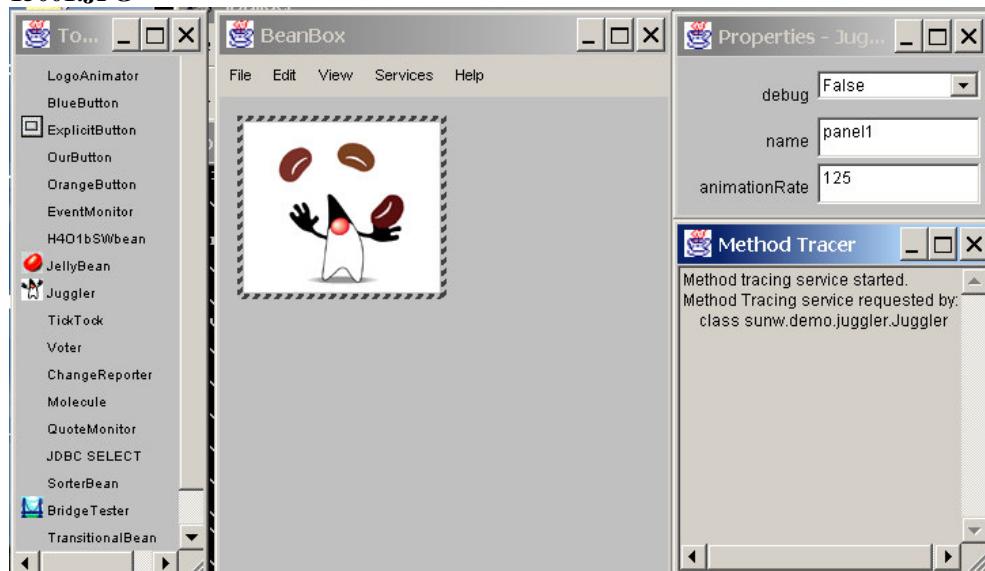
BÖLÜM 15 JAVA BEANS (JAVA FASULYELERİ) PROGRAMLAMA

15.1 TEMEL KAVRAMLAR

Java beans tekrar kullanılabilir paketler halinde java programlaması geliştiren, aynı zamanda hızlı yeni geliştirmelere de ön ayak olan bir programlama sistemidir. Daha önceden programlanmış hazır program paketçiklerinin ve apletlerinin bilgisayarlar arasında hızlı bir şekilde alışverişine yardımcı olan bir sistemdir. "Java beans" kavramı yeni bir programlama şeklini getirebilir. Bu kod yazma yerine program parçacıklarını birleştirerek programlama yapan bir programcı olabilir.

Java beans sistemini kullanmak için "JavaBeans development kit" Java bean kullanma seti'ne ihtiyacınız vardır. 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 kullanıldığını varsayıcağız.. Java Bean geliştirme setini kullanmak için önce programı kurmamız gereklidir. Paketi java dosyalarınızın olduğu ana dosyanın bir alt dosyası veya bağımsız bir dosya sisteminde tanımlayabilirsiniz. Bu dosyanın alt dosyası olarak **beanbox alt** dosyası mevcuttur. Bu dosyanın altında window sisteminde **run.bat**, unix sisteminde **run.sh** dosyasını bulacaksınız. Bu dosyayı açtığınızda beanbox açılacaktır. Beanbox programını açtığımızda, yanında Toolbox(aletkutusu), Properties (özellikler) ve Method tracer(metot izleyici) adı verilen programları da göreceğiz. ToolVox kutusunda jars alt dosyasında yer alan bean program paketlerinin listesini göreceğiz bunların birini seçip BeanBox kutusuna getirince bu program çalışmaya başlayacaktır. Örneğin Şekil 15.1 de Bean Box ve içinde çalışan Jogler programını görmekteyiz.

15001.JPG



Şekil 15.1 Juggler bean programının BeanBox'da görünümü

15.2 Bean programını geliştirme

Şimdi kendi bean programımızı nasıl yazabileceğimize bir göz atalım. Daha önceki bölümlerde yildiz.java programını geliştirmiştir ve bu programdan yola çıkararak analog saat ve çeşitli çizim programları yazmıştır. Önce yilsiz.java programının swing ve Graph2D olarak değiştirilmiş şeklinde bir göz atalım

Program 15.1 yildizSW.java programı, yıldız sınıfı

```
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 yolda 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 yıldız çizime  $\theta = \pi/2 + \text{aci}$  radyandan baslar
    // yıldızın  $i^{\text{inci}}$  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));
}

```

Şimdi ilk bean programımızı hazırlıyalım. Önce programımızı yazalım. program 15.2 de yildizciz2Dbean.java programı görülmektedir. Bu programda **Serializable** sınıfı implement edilmiştir. bunun dışında normal bir java programıdır. Programın JPanel olarak yazılmış olduğuna dikkat edelim. Bu yüzden

java yildizciz2Dbean

terimi kullanılarak çalıştırılabilir.

Program 15.2 yildicz2Dbean.java programı

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.net.*;
import java.io.*;

public class yildicz2Dbean 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 yildicz2Dbean()
    {
        setSize(getPreferredSize());
    }

    public void init()
    {
        setBackground(bg);
        setForeground(fg);
    }

    public Dimension getMinimumSize()
    {
        return getPreferredSize();
    }

    public Dimension getPreferredSize()
    {
        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.getPreferredSize().width+10,
              yildiz.getPreferredSize().height+30);
    f.show();
}
}

```

Bölüm 7.14 de bir tür program paketleme yolu olan jar deyimi ve kullanılmasından kısaca bahsedilmiştir. Burada programımızı bir jar dosyası haline getireceğiz. Fakat bunu yaparken aynı zamanda hem jarın içinde hangi dosyanın ana programı içerdiğini, hemde Java-bean olarak kullanılma izni veren Java_Bean komutunu veren **MANIFEST.MF** dosyasını oluşturacağız. Bunu yapmak için **manifest.tmp** isimli birdosya hazırlayacağız. Bu dosyanın içeriği :

Main-Class: yildiz2Dbean

Name: yildiz2Dbean.class

Java-Bean: True

şeklinde olacaktır. yildiz.jar isimli jar dosyasını oluşturmak 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 kullanılan polar, renk sınıflarının da jar gurubuna ilave edildiğine dikkat edelim. Bu komut yildiz.jar dosyasını yaratacaktır. Şimdi programımızı

java -jar yildiz.jar

komutunu kullanarak çalıştırabiliriz. Program

java -cp yildiz.jar yildiz2Dbean

komutuyla da çalıştırılabilir.

yildiz.jar dosyasının içeriğini 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

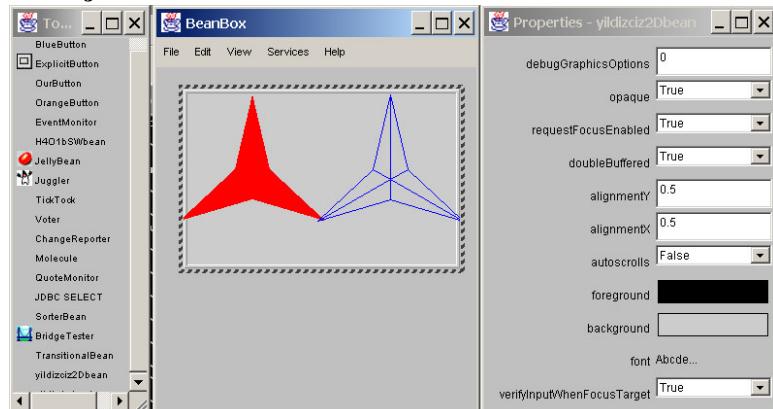
```

sonucunu verecektir. Tanımladığımız yıldız.jar dosyası aynı zamanda bir JavaBean dosyasıdır. Bu dosyayı jars alt direktörüsüne yataşındıktan sonra beanbox direktörüsündeki run.bat komutunu kullanarak beanbox programımızı açabilir ve bu ortamda gösterebiliriz. Şekil 15.2 de Java Bean haline gelmiş olan programımız görülmektedir.

İkinci bir örnek olarak aynı yıldız çizme prosesini amimasyon olarak yapacak yildizAnimasyon.java programını veriyoruz. Bu programda kullanılan stopAnimation() ve startAnimation() metodları animasyonu başlatma ve durdurma işlevlerini görmektedirler. Animasyon başladığında çizilen yıldız dönmektedir. Program 15.3 de bu programın listesi verilmiştir.

yildizAnimator programı bir önceki yildizciz2Dbean programına göre biraz daha detaylıdır. örneğin yıldızın kanat sayısını değiştirebilen metodlara sahiptir. Burada da bir önceki programla aynı yolları izleyerek yildizAnimator.jar dosyasını oluşturduk ve jars alt direktörüsüne yerleştirdik.

15002.JPG



Şekil 15.2 yildiz2Dbean programının BeanBox'da görünümü

Program 15.3 yildizAnimator.java programı

```

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 yildizkesesayisi;
    protected double aci;
    protected double piaci=Math.PI/360;

    public yildizAnimator()
    {
        setSize( getPreferredSize() );
        startAnimation();
        yildizkesesayisi=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 getAci()
{
    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 );
    }

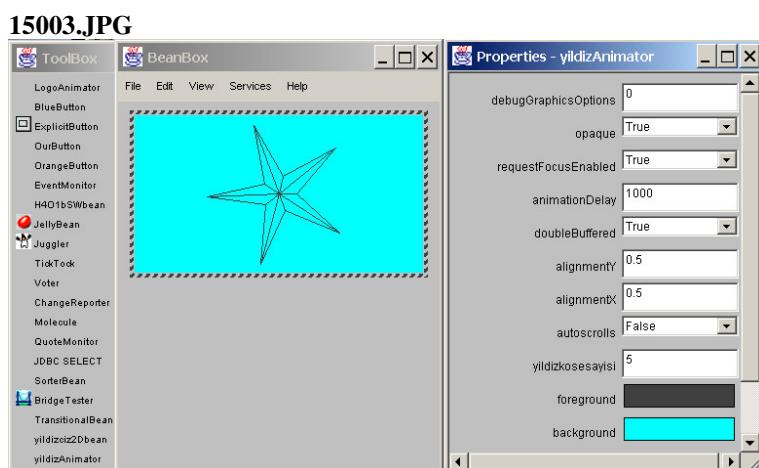
    public static void main( String args[] )
    {
        yildizAnimator anim = new yildizAnimator();

        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 dosyasının BeanBox'daki görüntüsü :

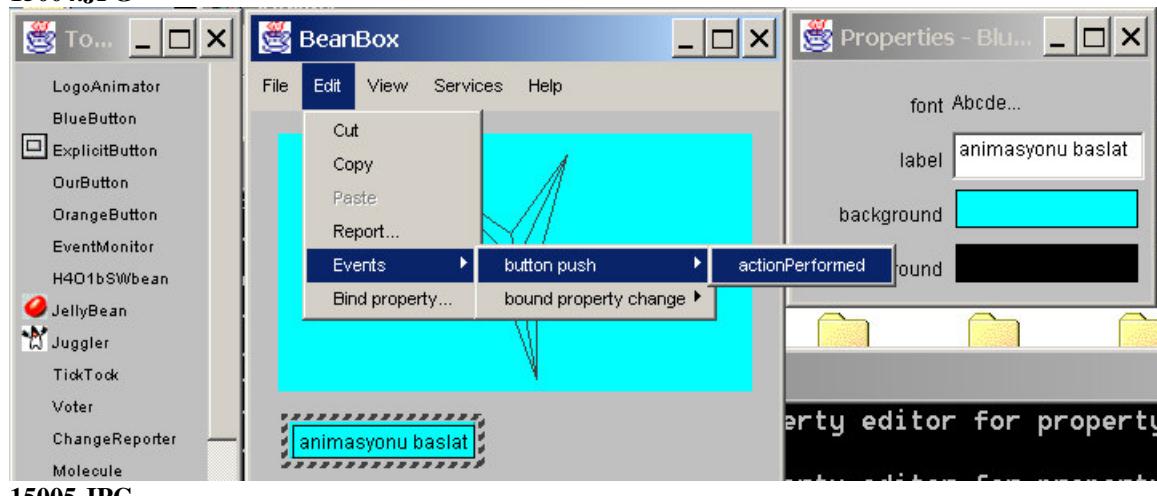


Sekil 15.3 yildizAnimator.jar programının BeanBox'ta görünümü

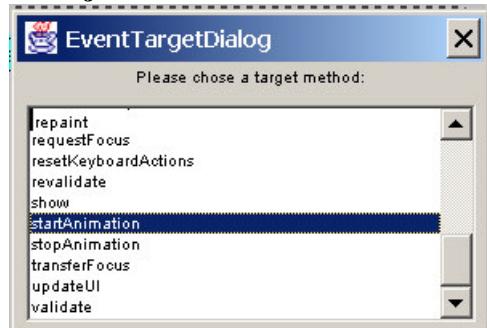
BeanBox Properties penceresini kullanarak program çalışırken özelliklerini değiştirebilirsiniz. Örneğin yıldızın ve arka planın çizim renklerini değiştirebilirsiniz.

Şu ana kadar yeni bir ortamda çalışmakla birlikte bu bölümün başında söylediğim gibi programlama olmadan çalıştığımızı söylemek pek mümkün değil. Fakat bundan sonraki kısımlarda programı bilgisayar oluşturacaktır. Örneğin yıldız döndüren yıldızAnimator programımıza animasyonu başlatacak ve durduracak komutlar ilave etmek isteyelim. Bunun için yeni bir program yazmak mümkün değildir. Fakat program yazmadan bu yeni düzenlemeyi BeanBox üzerinden gerçekleştirebiliriz. Bunu yapmak için önce beanBox'ta daha önce hazır olarak tanımlanmış bulunan **ExplicitButton** bean'ını seçelim ve bean box'a ilave edelim. Daha sonra properties penceresindeki , "press" yazısını "animasyonu baslat" olarak değiştirelim. Aynı işlemi tekrarlayarak "animasyonu durdur düğmesi" oluşturalım. Sonra Animasyonu başlat düğmesini seçelim. BeanBox da **Edit**, **Events**, **button push,actionPerformed** dizisini seçtikten sonra faremizin ucunda bir doğru oluşacaktır. Bu doğruya yıldızAnimasyon penceresinin içine çekerek sol mouse düğmesine basalım.

15004.JPG



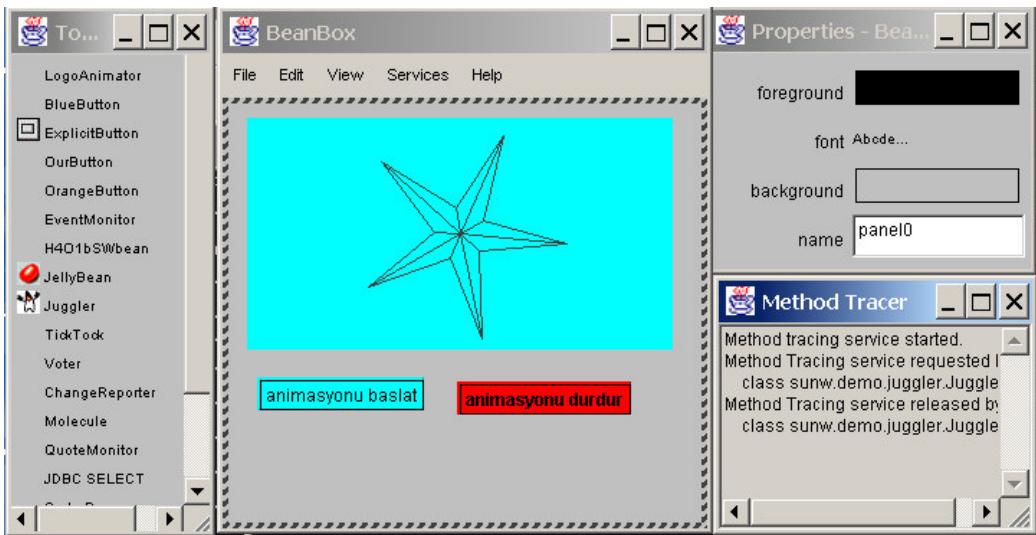
15005.JPG



Şekil 15.4-15.5 BeanBox'ta startAnimation metodunun "Animasyonu baslat" düğmesine actionPerformed olarak eklenmesi.

EventTargetDialog penceresi ekrana gelecektir. Bu pencerede startAnimation metodunu seçelim ve OK düğmesine basalım. Bu yeni program yeni bir kod oluşturacaktır. Aynı şeyi Animasyonu durdur düğmesi için de tekrarlıyalım. yalnız bu kez stopAnimation metodunu seçelim. Yeni programımız animasyonu durdurmuş ve tekrar çalıştırın düğmeleriyle birlikte hazır hale gelecektir. Düğmelere basarak çalışmasını deneyiniz.

15006.JPG



Şekil 15.6 BeanBox'ta startAnimation metodunun" tamamlanmış şekli

Oluşan yeni programı kaydedebiliriz. Bilgisayarın oluşturduğu java program kodu aşağıdaki gibidir.

Program 15.4 yildizAnimator.java programına BeanBox'da animasyon başlatma ve durdurma düğmelerinin eklenmesinden sonra otomatik olarak oluşturulan serizable MyApplet.java programı

```

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
// =====
}

```

Şüphesiz bu program kodunu biz çok daha kısa ve etkili ekilde yazabilirdik, ancak otomatik kod oluşturmaının zamansal olarak sağladığı avantaj ile bizim harcayacağımız zamanın bize ilave maliyeti hesaplanarak optimal çözümler oluşturulabilir.

15.2 ALIŞTIRMALAR

1. Alıştırma 3.13 ve alıştırma 14.10 da üzerinde çalıştığımız bilimsel hesap makinası programını 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 makinası

    JTextField sonucgirdisi;
    JTextField kutugirdisi;
    JButton Gir,Arti,Eksi,Carpı,Bolu,M,MR,MArtı,MEksi,Isaret,C;
    JButton Pow,Cos,Sin,Tan,RtoD,DtoR,Acos,Azin,Atan,Exp,Kok,Kare,birBoluX;
    JButton Ln,Log10,Oku,PI;
    //Button sinifi degiskenleri
    double sayi;
    bilimselhesapmakinası1 D;
    // pencereyi baslatma metodu

    public H4O1bSWbean()
    {
        D=new bilimselhesapmakinası1();
        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 ");
        Artı=new JButton(" + ");
        Eksi=new JButton(" - ");
        Carpı=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);
Carpı.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()==Carpı) 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());
}

```

```

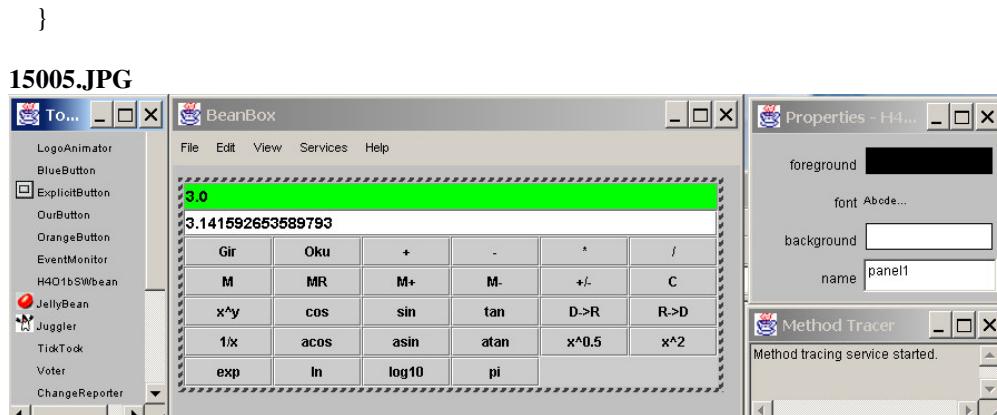
if(e.getSource()==MR || e.getSource()==M ||
e.getSource()==MArtille.getSource()==MEksi)
{
    kutugirdisi.setText(D.MtoString());
}
else if(e.getSource()==Cos ||
e.getSource()==Sinlle.getSource()==Tanlle.getSource()==RtoD ||
e.getSource()==DtoR || e.getSource()==Acos || e.getSource()==Ln ||
e.getSource()==Asinlle.getSource()==Atanlle.getSource()==Exp ||
e.getSource()==Log10lle.getSource()==Isaret lle.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 Makinası 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();
}
}

```



Şekil 15.5 H4O1bSWbean.java programının java BeanBox'ta görünümü

16. NETWORK PROGRAMLAMAYA GİRİŞ

16.1 TCP/IP PROTOKOLÜ

Java network protokolü olarak TCP/IP protokolünü kullanır. TCP ingilizce Transmission interface protokol (Veritransferi arayüz protokolü) IP ise Internet protokolü anlamına gelir.

IP kodu veriyi şu alt paketcikler olarak taşır :

- 0-10 : başlık
- 10-12 : kontrol
- 12-16 gönderici adresi
- 16-20 gideceği adres
- 20-24 : seçenekler
- 24....gönderilen bilgi

IP bilgi paketlerinin yerine ulaşma garantisiyoktur. TCP protokolü gitme garantili paketler oluşturur. IP paketini telgrafa benzetebiliriz. TCP paketleri ise telefon servisi gibidir. Karşılıklı iki tarafı bağlar. TCP paketinde alt birimler şunlardır :

- 0-2 gönderici adresi
- 2-4 gidecek yerin adresi
- 4-8 paket numarası
- 8-12 paket varış numarası
- 12-20 başlıklar
- 20+ veri

TCP kanallarından bazıları özel görevler görürler. Örneğin kanal 7 gönderilen bilgiyi geriye yansıtır. Kanal 13 verinin gittiği makinanın yerel zamanını bildirir. Kanal 20-21 ftp protokolü, kanal 23 telnet protokolü, kanal 80 http protokolü tarafından kullanılır. Javada internet adreslerini proses aden sınıfımız InetAddress sınıfıdır.

```
package java.net;
public final class InetAddress implements java.io.Serializable {
    InetAddress();
    InetAddress(String hostName, byte addr[]);
    public static InetAddress[] getAllByName(String Host) throws UnknownHostException;
    public static InetAddress getByName(String Host) throws UnknownHostException;
    public static synchronized InetAddress getLocalHost() throws UnknownHostException;
    public boolean isMulticastAddress();
    public byte[9] getAddress();
    public String getHostName();
    String getHostName(boolean check);
    public String getHostAddress();
    public int hashCode();
    public boolean equals(Object obj);
    public String toString();
}
```

Inet adres sınıfını kullanarak DSN server sitelerinin adresine ulaşabiliriz. Bunu yapan bir program kodu aşağıda verilmiştir.

```
Program 16.1 DNSadres.java programı
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 bilgisayarın adresi : "+InetAddress.getLocalHost());
}
}

```

program çalıştırılınca :

```

site 'www.mam.gov.tr' sitenin adresi : 193.140.76.10
Local bilgisayarın adresi : turhan-czvbccoqtf/193.140.79.15

```

sonucunu verecektir.

16.2 ALT SEVİYE İLETİŞİM : UDP KULLANIMI

network üzerinden kurabileceğimiz en basit iletişim yollarından biri olan UDP DatagramPacket ve DatagramSocket sınıflarını kullanarak gerçekleştirilebilir. DatagramPacket sınıfı dört kurucu metod içermektedir. Bu metodlardan ikisi veri almak, ikisi de veri göndermek amaçlıdır. Hem gönderirken hemde veri alırken byte olarak veriyi ve veri uzunluğunu bildirme zorunluğu mevcuttur. Bu sınıf asıl gönderme işlemlerini yapmaz, bu parelél sınıfı DatagramSocket tarafından gerçekleştiriliyor.

```

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 sınıfı bir örnek problemde kullanalım.

Program 16.2 GetDate.java programı

```

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 ("TÜBİTAK MAM'da şu anda saat: " + received);
        dgSocket.close();
    }
} // end of class GetDate

```

program çıktısı :

TÜBİTAK MAM'da şu anda saat: Sun Mar 11 16:14:20 2001

Program çıktısından da görüldüğü gibi bu program zaman port'undan mam.gov.tr bilgisayarının zamanını okumaktadır.

16.3 TCP BİLGİ İLETİMİ, SOCKET SINIFI

Direk olarak TCP üzerinden bilgi aktarma UDP üzerinden bilgi aktarımı gibidir. En önemli fark, TCP kullanırken veri miktarının önemli olmamasıdır. İstenilen miktarda veri TCP lanalı üzerinden aktarılabilir. TCP veri aktarımı için socket sınıfını kullanıyoruz. Socket sınıfının tanımı :

```
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
}
```

şeklindedir.

Program 16.3 email.java programı (TCP Socket sınıfı örneği)

```
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,"Doğru çağrıma: java email gönderilendosya gönderenin_adresi
gittiği_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 bağlantı sağlanamadı");
    }
}
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;
}

```

programı çalıştırınmak için :

```
>java email a.dat Turhan.Coban@posta.mam.gov.tr turhan@mam.gov.tr mam.gov.tr  
komutunu kullanabiliriz.
```

Program çıktısı :

```
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>  
250 OK - mail from <turhan@mam.gov.tr>  
RCPT TO:<Turhan.Coban@posta.mam.gov.tr>  
250 OK - Recipient <Turhan.Coban@posta.mam.gov.tr>  
DATA  
  
.354 Send data. End with CRLF.CRLF  
QUIT
```

Şeklinde oluşur.

16.4 İNTERNET SİTELERİYLE BİLGİ ALIŞVERİŞİ, URL SINIFI

Java'nın Internet ortamında programlama için oldukça iyi bir araç olduğundan kitabı başında bahsetmiştık. Bu konuda javanın çok iyi bir araç olması nazarlarının Java dilini sadece bir internet aracı olduğunu sanmalarına kadar gitmektedir. Java dilinde internet ve serverlerin haberleşmeleri için bir çok araç bulunmaktadır. Bu araçların en önemlilerinden biri de şüphesiz URL sınıfıdır. bu sınıf yardımcıla bir internet adresiyle (html adresiyle) rahatlıkla bağlantı kurulabilir. url sınıfının genel tanımı şöyledir :

```
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
}

```

İnternetten URL sınıfını kullanarak veri aktarımı konusunda iki örnek program verelim. Birinci program seçilen sitelere ulaşım yapacak bir internet sitesi seçme programıdır.

Program 16.4 internetsitesisecici.java, bu program applet kullanarak choice deyimiyle listelenmiş (programın içinde) internet siteleri arasında seçim yapar ve sonucda bu internet sitesini browsere çağıır.

```

import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.util.Hashtable;
import java.applet.Applet;

public class internetsitesisecici 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";
            url=new URL(yer);
            site.put(baslik,url);
            siteChoice.add(baslik);
            baslik="Turhan Coban";
            yer="http://www.mam.gov.tr/~turhan";
            url=new URL(yer);
            site.put(baslik,url);
            siteChoice.add(baslik);
            baslik="TUBITAK Marmara Arastirma Merkezi network sitesi";
            yer="http://www.mam.net.tr";
            url=new URL(yer);
            site.put(baslik,url);
            siteChoice.add(baslik);
            baslik="Java sitesi";
            yer="http://java.sun.com";
            url=new URL(yer);
            site.put(baslik,url);
            siteChoice.add(baslik);
        }
    }
}

```

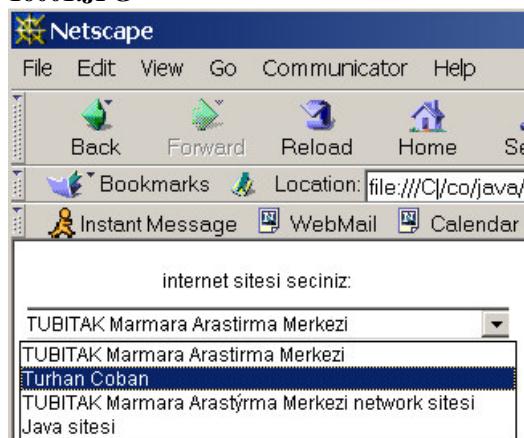
```

}
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.getSelectedItem();
    getAppletContext().showDocument(url);
}
}

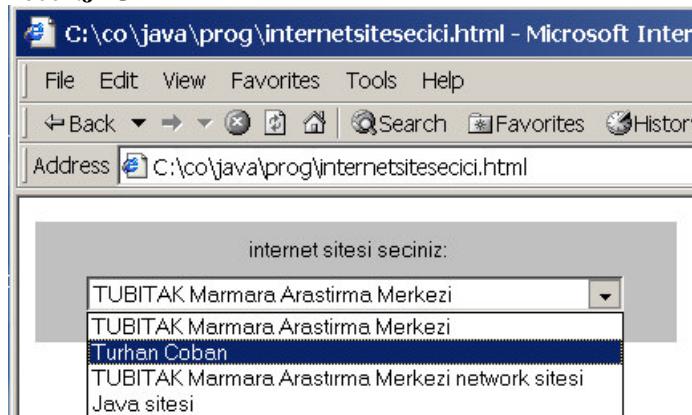
```

16001.JPG



Şekil 16.1 internetsitesecici programını kullanarak bir internet sitesinin seçilmesi (netscape)

16002.JPG



Şekil 16.2 internetsitesecici programını kullanarak bir internet sitesinin seçilmesi (Microsoft internet explorer)

16003.JPG



Şekil 16.3 internetsitesecici programını kullanarak seçilen internet sitesinin Netscape browsereyle gösterilmesi

Program 16.5 internetsitesiseciciSW.java, bu program Japplet (swing) kullanarak listelenmiş (programın dündə html dosyasında tanımlanmış) internet siteleri arasında seçim yapar ve sonucda bu internet sitesini browsere çağırır.

```
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 dökümanındaki 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 programın çıktısı da bir önceki programda olduğu gibidir. Bir sonraki programda internet üzerinden verilen adresle giderek dosyayı okumaktayız. Yani bir anlamda küçük bir browser programı yarattık.

Program 16.6 InternetDosyaOku.java, bu program küçük bir browser tanımlar. JTextField alanına yazılan URL adresindeki dosyayı JeditorPane kullanarak ekran'a getirir.

```

// Bu program JEditorPane kullanarak
// bir Web sayfasını ekran'da 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 ) );

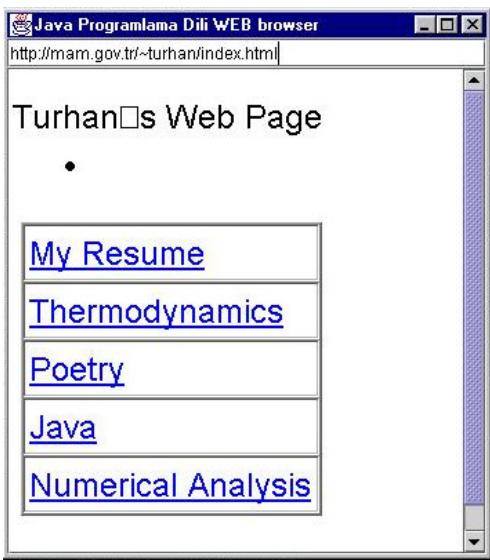
    try {
        contents.setPage( adres );
        gir.setText( adres );
    }
    catch ( IOException io ) {
        JOptionPane.showMessageDialog( this,
            "internet URL adresine erisilemedi",
            "URL adresinde hata var",
            JOptionPane.ERROR_MESSAGE );
    }

    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 );
            }
        );
}
}

```



Şekil 16.4 internetdosyaoku programını kullanarak seçilen internet sitesinin gösterilmesi

17. GÜVENLİK

Güvenlik denince bir programın girdi çıktı fonksiyonları aracılığıyla yapabileceği değişiklikler, bu değişikliklerin etkileri anlaşılmır. Javada güvenliği sağlamak için program girdi çıktıları kontrol edilir ve bir programın güvenlik riski taşıdığı düşünülüyorrsa girdi çıktı fonksiyonları sınırlanır.

17.1 APPLETLERDE GÜVENLİK

Appletler güvenlik açısından sınırlanmış programlardır. Genel olarak yerel bilgisayara yazma ve okuma iznine sahip degillerdir. Bunu bir örnekle açıklayalı. Program 17.1'de bir applet programı olan yazitesti.java verilmiştir. Bu program yazitesti.txt dosyasına "bu yazı deneme amacıyla dosya yazitesti.txt içine Appletden yazıldı" yazısını yazmak için oluşturulmuştur.

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 yazı deneme amacıyla dosya yazitesti.txt içine Appletden yazıldı\n");
            dos.close();
            g.drawString("Yazım işlemi başarı ile gerçekleşti : " + myFile + " -- dosyasını açıp bakabilirsiniz ", 10, 10);
        }
        catch (SecurityException e) {
            g.drawString("yazitesti: güvenlik kontrolu aşılamadı (güvenlik hatası) : " + e, 10, 10);
        }
        catch (IOException ioe) {
            g.drawString("yazitesti: giriş çıkış hatası", 10, 10);
        }
    }
}
```

Program çalıştırıldığında (appletviewer yazitesti.html)

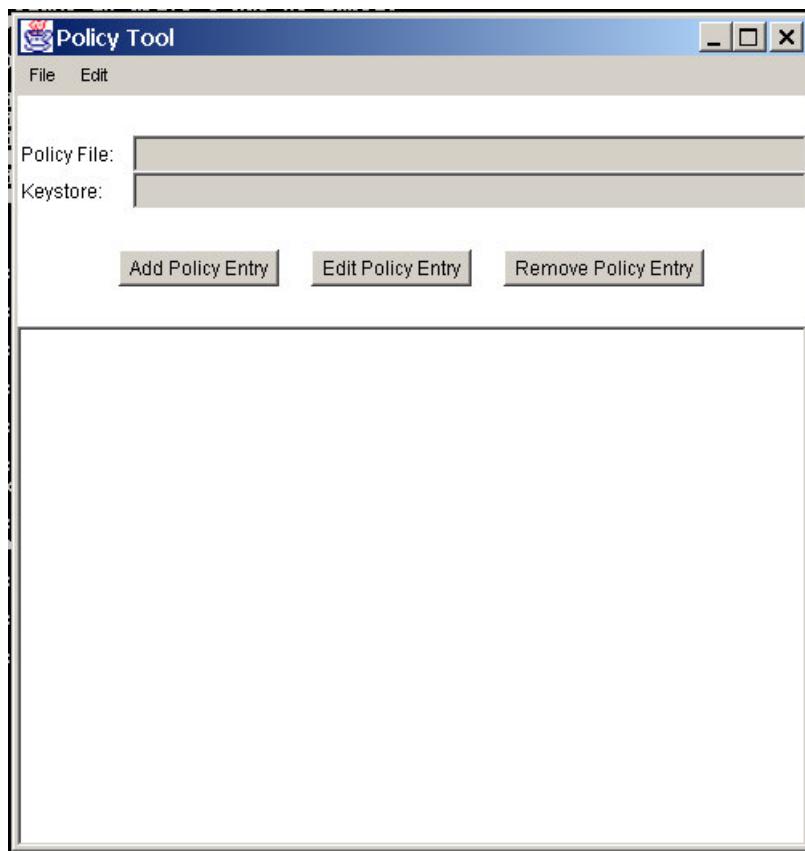
17001.JPG



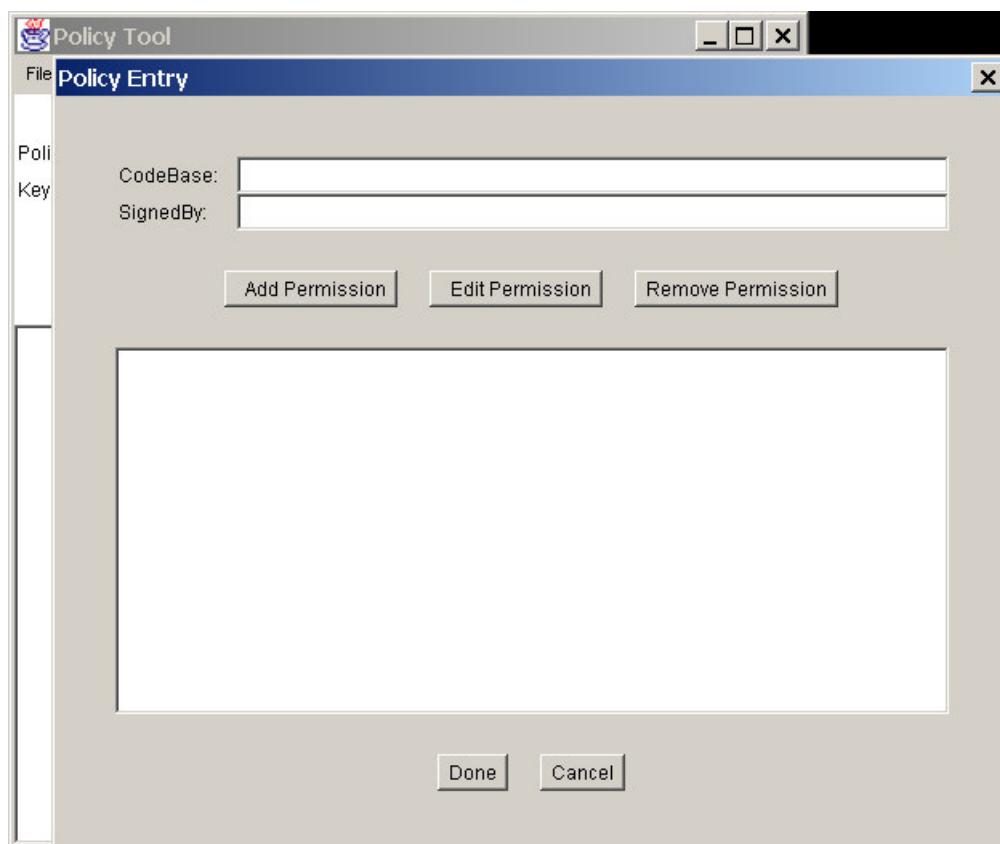
Figure 17.1 yazitesti.html applet çıktısı

Yazitesti:güvenlik kontrolu aşılamadı(güvenlik hatası): java security.AccessControlException:access denied(java.io.FilePermission yazitesti.txt write)

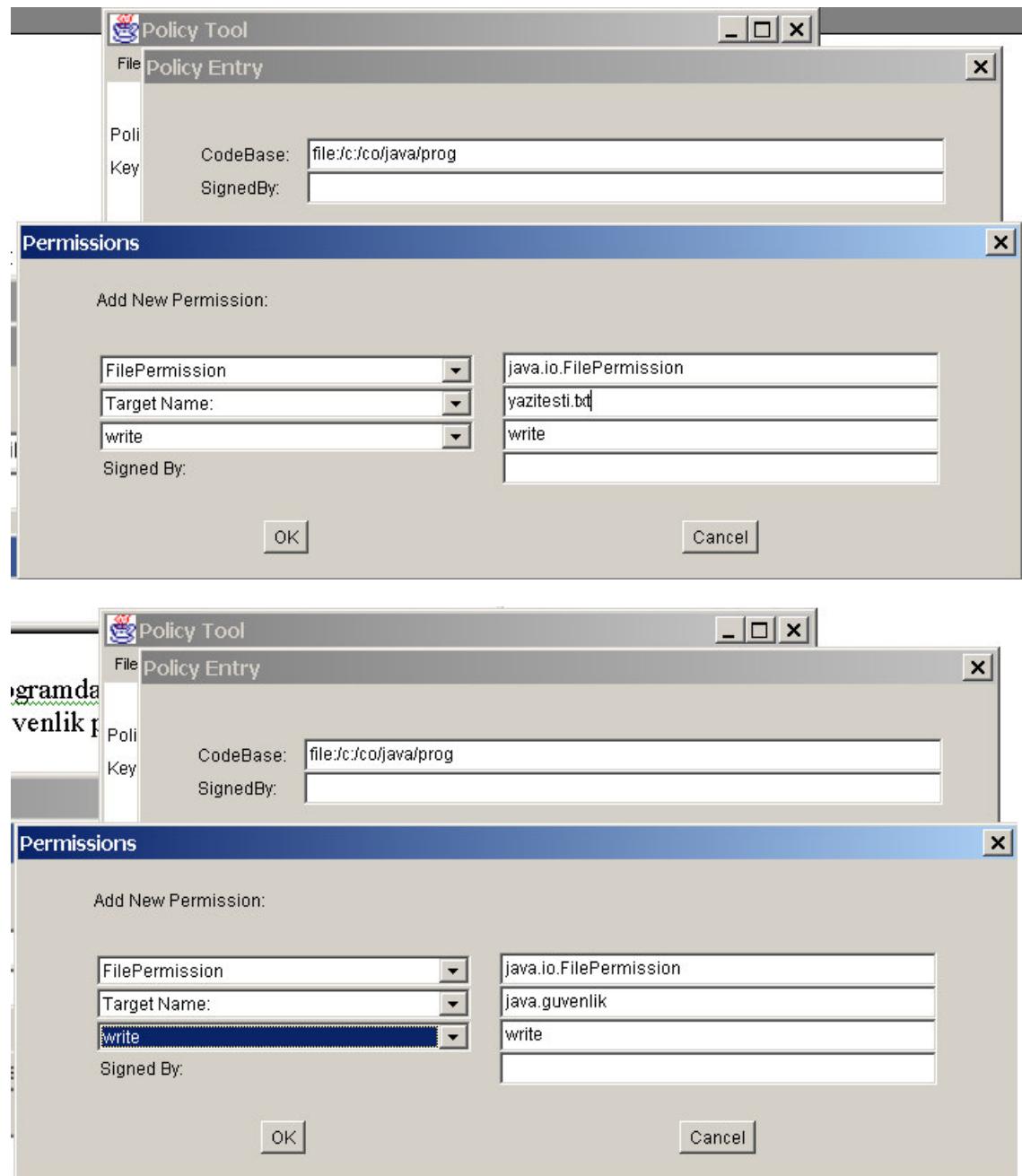
Mesajı verilmiştir. yazitesti.txt dosyası incelendiğinde de hiç bir şey yazılmadığı görülür. Bu applet programını kullanarak nasıl dosyaya yazı yazdırabiliz. Bunun için önce **policytool** isimli programı çalıştırıralım. Ekrana



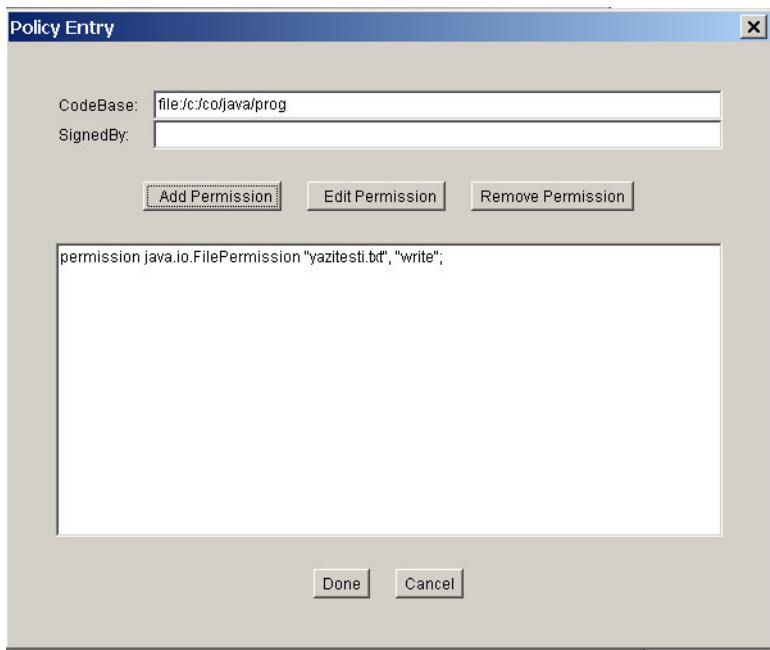
bir java programı gelecektir. Programdaki Add Policy Entry (güvenlik kurallı ekle) düğmesine basalım. Yeni bir ekran açılacaktır. Bu ekran da güvenlik programımızı içerecek dosya url adresini yazıp (örneğin file:/c:/co/java/prog veya html://www.mam.gov.tr/~turhan) Add Permission tuşuna basalım.



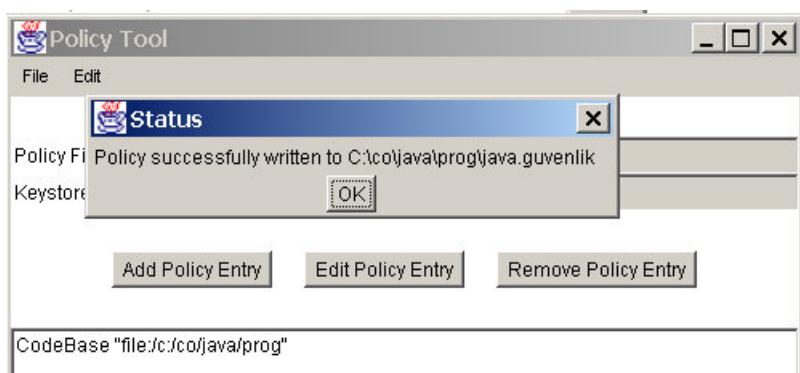
File permission (dosya güvenlik izni) penceresinden file permission (dosya izni) seçtikten sonra Target Name alanına yazma izni istediğimiz yazitesti.txt dosyasının ismini yazalım. Ve son alanda da yazma izni (“write”) seçelim ve ok düğmesine basalım.



son pencere kapanacak ve bir önceki pencere gelecektir. Burada da Done düğmesine basalım.



ilk pencerede file, save as seçeneklerini seçtikten sonra c:\co\java\prog (veya sizin çalışığınız dosyaya) java.guvenlik adıyla kaydedelim. Program policy successfully written to c:\co\java\prog\java.guvenlik mesajını verecektir.



şimdi policy tool programını exit deyimiyle kapatabiliriz. Policytool programı bize otomatik olarak **java.guvenlik** dosyasını yarattı. Bu dosyanın kapsamına 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 aynı komutu dosyaya elde de girebilirdik. Bu komutun anlamı c:\co\java\prog direktory'sindeki yazitesti.txt dosyasına yazma ("write") müsadesi ver demektir.

Şimdi appletviewer programını

appletviewer -J-Djava.security.policy=java.guvenlik yazitesti.html komutuyla çalıştırıldığımızda :



çıktısını alacağız. Yazitesti.txt dosyasını açtığımızda

bu yazı deneme amacıyla dosya yazitesti.txt içine Appletden yazıldı

dosyaya yazısının yazıldığını göreceğiz.

Burada yazımı gerçekleştirmek için **-J-Djava.security.policy=java.guvenlik** gibi özel bir anahtar kullandık. Aynı işi bu anahtarı kullanmadan yapmak için **java.security** dosyasına yeni bir komut ilave etmemiz gereklidir. Bu dosya sizin kullandığınız ana java directory'sinin altındaki lib\security de bulunacaktır. Örneğin benim dosyam c:\co\java\jre\lib\security directory'sinde yer almaktadır. Bu dosyanın 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

-----

```

şeklindedir. Bu dosyaya biz yazitesti.txt izni olan java.guvenlik dosyasını

policy.url.3=file:/C:/co/java/prog/java.guvenlik

komutunu kullanarak ekledik. 3 rakamının anlamı daha önce yazılan iki güvenlik kodunun olmasıdır. Şimdi sadece **appletviewer yazitesti.html** komutu kulanarak programımızı çalıştırıp yazma işlemini başarabiliriz.

17.2 KONSOL PROGRAMLARINDA GÜVENLİK

Applet dışı programlarda (konsol programlarında) java direkt olarak güvenlik kontrolü istemez. Yukardaki programın bir JFrame eşdeğerini oluşturalım

```

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("yazı testi swing frame konsol uygulaması");
    }

    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D)g;

        try {
            dos=new PrintWriter(new BufferedWriter(new FileWriter(myFile)));
            dos.println("bu yazı deneme amacıyla dosya yazitesti.txt içine Appletden yazıldı\n");
            dos.close();
            g2.drawString("Yazım işlemi başarı ile gerçekleşti : " + myFile + " -- dosyasını açıp bakabilirsiniz ", 10, 50);
        }
        catch (SecurityException e) {
            g2.drawString("yazitesti: güvenlik kontrolü aşılamadı (güvenlik hatası) : " + e, 10, 50);
        }
        catch (IOException ioe) {
            g2.drawString("yazitesti: giriş çıkış hatası", 10, 50);
        }
    }

    public static void main(String[] args)
    {
        yazitestiSWF pencere= new yazitestiSWF();
        pencere.addWindowListener(new BasicWindowMonitor());
        pencere.setSize(350,200);
        pencere.setVisible(true);
    }
}

```

```
}
```

bu programı javac yazitestiSWF.java deyimiyle derleyip
java yazitestiSWF ile çalıştırırsak



programın direk olarak hiçbir güvenlik işlemine gereksinim duymadan çalıştığını görürüz. Eğer bu güvenlik kontrolunu özel olarak arzularsak, programımızı **java -Djava.security.manager yazitestiSWF** komutuyla çağırabiliriz. Bu durumda daha yazitestiSWF.txt dosyasını güvenlik sisteminde tanımlamadığımız için tipki appletde olduğu gibi



sonucuyla karşılaşırız. Güvenlik sistemine bu yeni dosyayı da tanımlayarak konsol programını da güvenli moda çalıştırabiliriz.

Örneğin java.guvenlik dosyasının yazitestiSWF.txt dosyası içinde yazma izni almış 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 güvenliğini sağlayan komut

java -Djava.security.manager=Djava.security.policy=java.guvenlik yazitestiSWF

şeklinde olacaktır.

Şimdi size sistem özelliklerini veren bir program verelim.

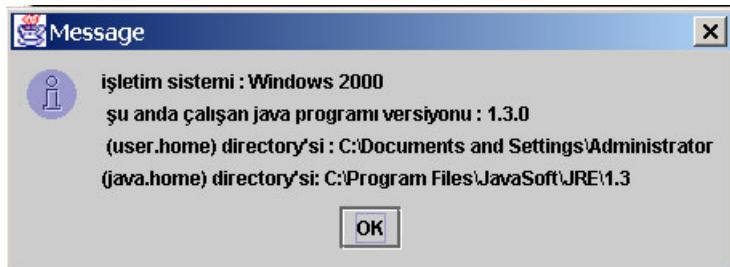
```
import java.lang.*;  
import java.security.*;  
import javax.swing.JOptionPane;  
  
class ozellikler {  
    public static void main(String[] args) {  
        String s,s1;  
        try {  
            s = System.getProperty("os.name", "not specified");  
            s1="işletim sistemi : " + s+"\n";  
            s = System.getProperty("java.version", "not specified");  
            s1+=" şu anda çalışan java programı 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 yakalandı " + e.toString();}
```

```

        JOptionPane.showMessageDialog(null,s1);
        System.exit(0);
    }
}

```

bu programı çalıştırduğımızda :



çıktısını verecektir. Şimdi programı güvenli moda çalıştıralım, çıktı :



olacaktır. 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 hatasız olarak iletecektir.

17.3 GÜVENLİK SERTİFİKALARI

Bilgisayarlardaki genel güvenlik sistemi genelde iki anahtarlı bir sistemdir. Bu anahtarlardan birincisi, özel anahtar, sadece sizin bildiğiniz bir anahtاردır. İkinci anahtar güvenli olarak iletişim kurmak istediğiniz her yere gönderilebilir. Sertifika almanın ilk aşamasını javada verilen keytool adlı programı kullanarak yapabiliriz. Bunun için

keytool –genkey

Komutunu kullanmamız ve sorulan sorulara (ingilizce) cevap vermemiz kafidir. Keytool –genkey şu soruları soracaktır : kalın ile belirttiğim cevaplar, parantez içerisindekiler türkçe karşılıklarıdır.

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? (iş yerinizin iş türü nedir?)

[unknown]: **research (arştırma)**

What is name of your organisation? (iş yerinizin adı nedir?)

[unknown]: **TUBITAK MRC**

What is the name of your City or Locality? (şehrinizin veya bulunduğuuz yerin adı nedir?)

[unknown]: **Gebze**

What is the name of your State or Province? (eyaletinizin veya bölgenizin adı 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 doğru mu?)

[no]: yes (evet)

bu bilgiler verildikten sonra program

Enter key password for <mykey>

(RETURN if same as keystrore password):

(mykey anahtarı için şifreyi giriniz daha önce girdiğiniz şifreyle aynıysa sadece gir tuşuna basınız) sorusunu soracak ve şifre isteyecektir. Burada sizin şifreniz mykey ismini almıştır.

Eğer ikinci bir kişi size şifre gönderirse bunu

keytool –import –file anahtarismi

komutuyla depolayabilirsiniz. Anahtarı oluşturuktan sonra anahtarınıza bir anahtar sertifikasyon kurumundan sertifika alabilirsiniz sertifika alabilirsiniz, bunun için önce sertifika onay belgesi oluşturabilirsiniz. Bunun için **keytool –certreq** komutu kullanabilirsiniz. Bu komut

-----BEGIN NEW CERTIFICATE REQUEST-----

```
MIIccjCCAi8CAQAwbDELMAkGA1UEBhMCVFIxEDAOBgNVBAgTB0tvY2FlbGkxDjAMBgNVBAcTBUDl  
YnpIMRQwEgYDVQQKEwtNUkNFVFCSVRSzEOMAwGA1UECxMFRVNFKxxFTATBqNVBAMTDFR1cmhh  
biBDb2JhbjCCAAbgwggEsBgcqhkjOOAQBMIIIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3UjzvRADD  
Hj+AtlEmaUvdQCJR+1k9jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7qdf+  
t8Yb+DtX58aophUPBPuD9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAJdgUI8VIwvM  
spK5gqLrhAvvWBz1AoGBAPfhIXWmz3ey7yrXDa4V7I5IK+7+jrqgvIXTAs9B4JnUVIXjrrUWU/m  
cQcQgYD0SRZxI+hMKBYTt88JMozIpuE8FnqLVHyNKOCjrh4rs6Z1kW6jfwb6ITVi8ftiegEkO8yk  
8b6oUZCJqlPf4VrlnwaSi2ZegHtVJWQBTDv+z0kqA4GFAAKBqQCGLh1t/Vo3fMkMFLXNu9dhueKG  
dac3vGCUmd1FzqBALncr+YEQw2DZqtixwqleivbXOoean2516S0cDBi33ArOGtK0pm/pHQ3Yyeh  
W/ePbV6ljh/Kbmua3ZSfAHKseuHBcMkS4SNC5DeTVifzT1QLRwGeAkP5xdL72EX/dt8KAAMAsG  
ByqGSM44BAMFAAMwADAtAhUAlm7LpK8FaOco1hgARAKCX0ik9CECFDqrUbnw9Fyp5CRUYs0+RCFy  
L+U9  
-----END NEW CERTIFICATE REQUEST-----
```

gibi bir çıktı oluşturacaktır. Bu çıktı anahtar sertifika otoritelerine direk olarak email ile gönderilebilir. Sertifikanın gayesi sizi tanımayan kişilerce bile programlarınızın güvenliğinin onaylanmasıdır. Genellikle sertifika kuruluşları bunu bir ücret karşılığı yapar. Örneğin www.baltimore.ie adresinde böyle bir kuruluşla ilgili detaylı bilgi bulabilirsiniz. Sertifika kuruluşundan onaylanmış olarak gelen sertifikانızı

keytool –import –file \directory\yenisertifika

komutuyla yeni bir isimle depolayabilirsiniz. Sertifikanızı başka birine aktarırken

keytool –export –rfc komutu ile önderme anahtarını oluşturup gönderebilirisiniz.

17.4 JAR DOSYALARININ GÜVENLİK KODUYLA İMZALANMASI

Jar dosyalarının hazırlanmasından daha önce bahsetmiştik. Java jar dosyaları jarsigner komutu kullanılarak sertifikalandırılabilir. Komut

jarsigner arsiv.jar mykey şeklindedir. Buradaki mykey sizin anahtarınızın adı, arsiv.jar ise kullandığımız jar dosyasıdır. Dosayı kullanmadan önce –verify ve –verbose terimleriyle açmamız gereklidir.

jarsigner –verify –verbose arsiv.jar

17.5 BROWSERLARDA GÜVENLİK

Browserlerin güvenlik sistemi burada anlattıklarımızla benzer olmakla beraber aynısı değildir. Microsoft ve netscape birbirinden ayrı güvenlik sistemlerine sahiptirler. Bunlarla ilgili bilgi

Microsoft için : <http://microsoft.com/java/security/default.htm>

Netscape için : <http://home.netscape.com/info/security-doc.html>

adreslerinden edinilebilir.

EK A TEXT SINIFI

Bu kitapta okuma işlemlerinde Text sınıfı oldukça yoğun olarak kullanılmıştır. 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 DERSİ SINAV SORULARI

Java derslerindeki bazı sınav soruları bu ekde verilmiştir.

JAVA PROGRAMLAMA DİLİ GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ GÜZ DÖNEMİ 2000 BİTİRME SINAVI SORULARI

İsim – soyisim	
Öğrenci numarası	
Master/doktora	
Email	
Telefon	
Not	

AÇIKLAMALAR

- Toplam süre 3 saatdir.
- Sizin için verilen boş alanlara soruları cevaplandırınız.
- Verilen yerler yetmezse ilave kağıt kullanabilirsiniz
- Her soru on puandır (toplam 100 puan)
- İyi şanslar

SORU 1

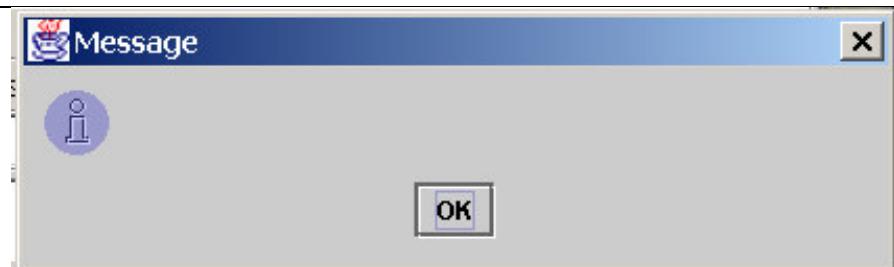
Aşağıda verilen programın girdisi :

ilk gerçek sayiyi giriniz : 2
ikinci ger‡ek sayiyi giriniz : 2

şeklindedir. Program çıktısını yazınız.

```
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

Aşağıda verilen kutuya girdi olarak 2 girildiğinde oluşacak program çıktısını yazınız

```

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; //yazı alanları sınıfı
JTextField t1; //girdi alanları (kutuları)
JTextArea ciktiAlani; //cıkıtı alanı
//pencereyi başlatma metodu
public void init()
{
    Container c=getContentPane();
    c.setLayout(new FlowLayout());
    Color renk=c.getBackground();
    s1=new JLabel("küp ortalaması 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 alanından sinyal bekle ve
sinyal gelince şu işlemleri yap

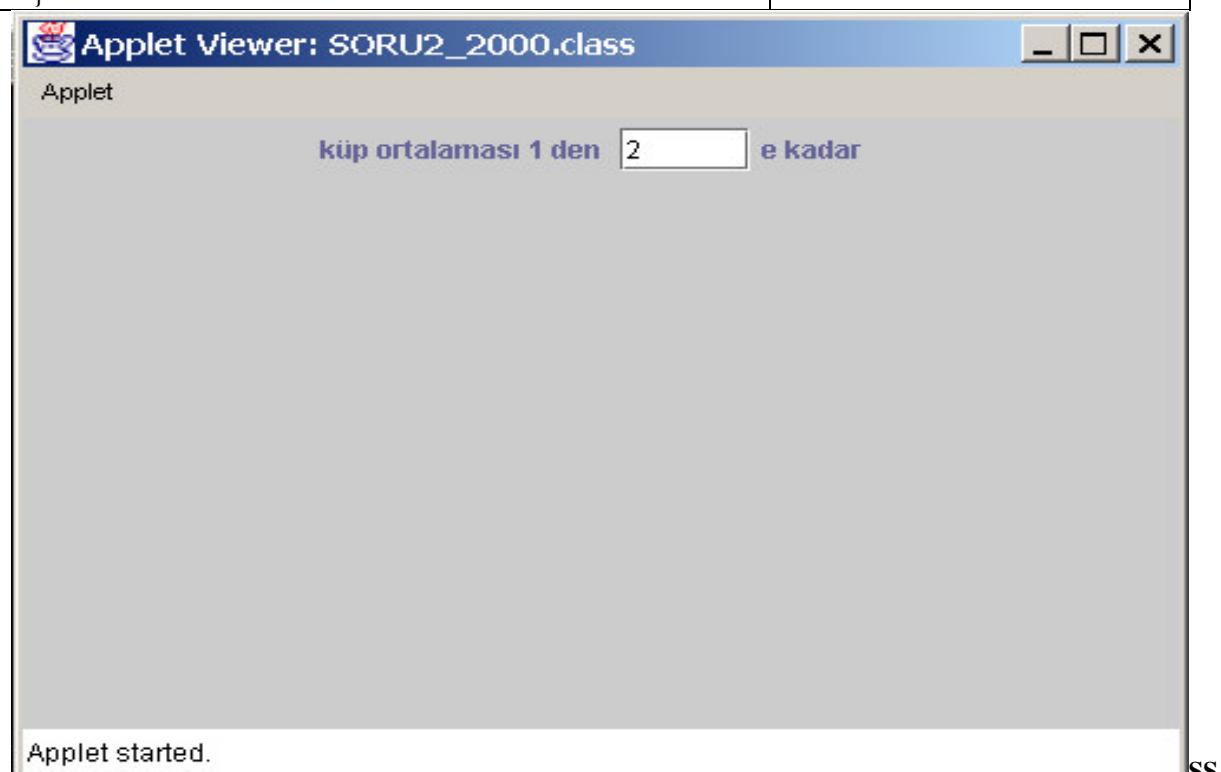
public void
actionPerformed(ActionEvent e)
{
    toplam=0; //sayi1 i kutulardan
oku

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+
kadar küp ortalaması :" +ort);
repaint();
}

//ActionPerfomed metotunun
kapanisi
} //H2OD3_2000 sınıfının kapanısı

```



SORU 3

Aşağıda verilen programdaki sayının kuvvetini hesaplayan
public static double pow(double taban,double kuvvet)

metodunu yazınız. Bu metod kendi kendini çağrıran bir program kodu veya bir döngü yapısı kullanan bir şekilde oluşturulabilir.

```
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 sayınızı giriniz : "));
        kuvvet=Double.parseDouble(JOptionPane.showInputDialog
        ("kuvveti giriniz : "));
        JOptionPane.showMessageDialog(null,taban+" nin "+kuvvet+" inci kuvveti: "
        +pow(taban,kuvvet),"H3OD3_2000.java programı",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Cevap :

```
public static double pow(double taban,double kuvvet)
{
    }
```

SORU 4

Aşağıda verilen kompleks2 programının içerisinde yer olması gereken ve bir kompleks sayının karesini hesaplayan kare metodunu yazınız. kompleks2 sınıfını kullanarak $(1+2i)$ kompleks sayısının karesini alan ve sonuçlarını konsol ortamında yazdırın bir küçük bir çıktı programı yazınız

```
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 sayının 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 sınıfının tanımının sonu

```

Kare metodu :

```

public static kompleks2 kare(kompleks sol)
{
}

```

SORU4_2000.java çıktı programı

SORU 5

Boyutlu değişken kullanımında en büyük sorunlardan biri boyutların sabit olması, ve değiştirilememesidir. Double tipi değişkenlerin boyutlarının değiştirilmesi amacıyla doubleBoyut sınıfı geliştirilmiştir. Fakat program dosyasına kaza sonucu dökülen kahve sonucu programın bir kısmı okunamamaktadır. Program kodu bilgisayara tekrar girilecektir. BoyutEkle alt metodundaki okunamayan yerin tekrar yazılmasında yardımcı olabilirsiniz.

<pre> public class doubleBoyut { //sınıf değişkenleri 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) </pre>	<pre> 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]; } } </pre>
--	--

<pre> { 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; } </pre>	<pre> 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; } </pre>
--	---

<pre> public void boyutEkle(int n) { if(n>0) { int z=a.length+n; a=b; length=a.length;}}} </pre>

SORU 6

OvalPanelSWP panel programı verilmiştir. Biz bu programı kullanarak bir elips çizdirmek istiyoruz. Bunun için yazılan JFrame programı SORU6_2000 nin kurucu metodu SORU6_2000metodunu tamamlayınız.

```

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)
=====
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());
}

}

{
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);
}

}

=====

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 çıktı tablosunu çiziniz ve oluşan değerleri yazınız.

```

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 çıktısı :

SORU 8

gir1.txt dosyası incelediğinde

- 1 2**
- 2 3**
- 3 4**
- 4 5**
- 5 6**

verisinin kaydedildiği gözlenmiştir. Aşağıdaki program çalıştırıldığında cik1.txt dosyasında bulunması gereken veriyi yazınız

```

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

- a) programın işlevini izah ediniz
- b) verilen girdi için çıktıyı yazınız

```

import java.util.*;
import java.awt.*;

```

```

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

```

```

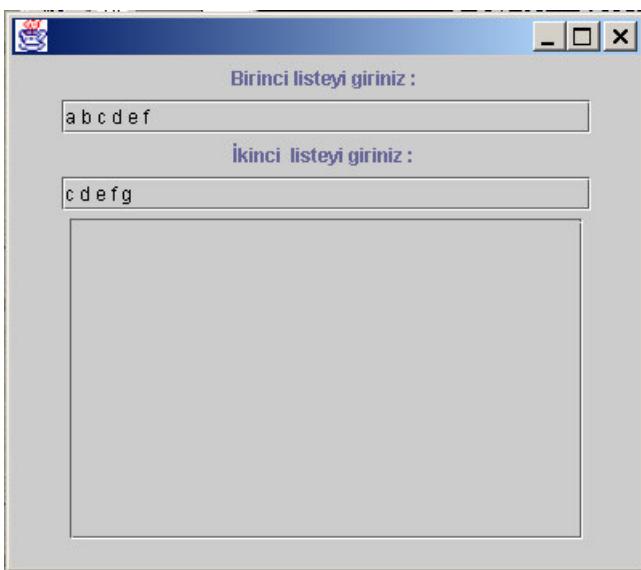
class SetA
{
    public Vector bilesim(Vector list1, Vector
list2)
    {
        Vector bilesimVectoru = new Vector();
        String s1,s2;
        Enumeration n1=list1.elements();
        Enumeration n2=list2.elements();
        while(n1.hasMoreElements())
        {
            s1=(String)n1.nextElement();
            bilesimVectoru.addElement(s1);
        }
        while(n2.hasMoreElements())
        {
            s2=(String)n2.nextElement();
            if(!bilesimVectoru.contains(s2))
                bilesimVectoru.addElement(s2);
        }
        return bilesimVectoru;
    }
}

public 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()
    {
        super("küme testi bileşim ve kesişim
kümeleri");
        c=getContentPane();
        c.setLayout(new FlowLayout());
        P1 = new JLabel("Birinci listeyi giriniz : ");
        T1 = new JTextField(30);
        P2 = new JLabel("İkinci 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));
        c.add(jp);
    }

    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("İki listenin bileşim kümesi :
\n");
            ciktiA(ansUn);
        }
    }
}

//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 sayıların ortalamasını bulan bir java bilgisayar programı yazınız.

JAVA PROGRAMLAMA DİLİ GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ 20 OCAK 2000 BITİRME SINAVI SORULARI

Dr. Turhan Çoban

isim soyisim	
ogrenci numarası	
email /telefon	
not	

AÇIKLAMALAR :

- Toplam süre 3 saatdir. (9.00 -12.00)
- Sizin için verilen boş alanlara soruları cevaplandırınız
- verilen yerler yetmezse ilave kağıt kullanabilirsiniz.
- İyi şanslar

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);
}
}
```

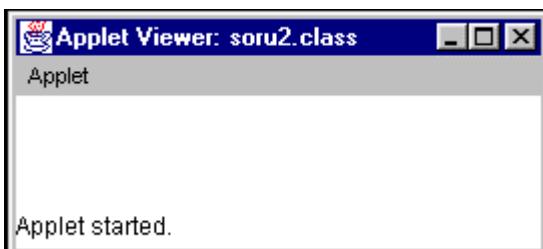
Verilen programın sonuçlarını yazınız :

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 programın sonuçlarını yazınız :

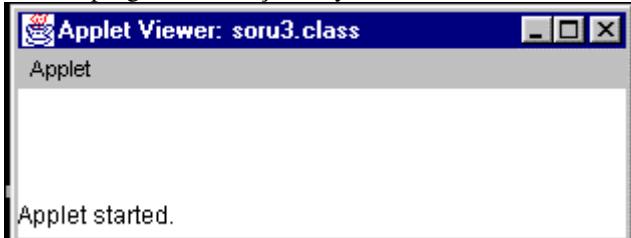


SORU 3

```
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 programın sonuçlarını yazınız :

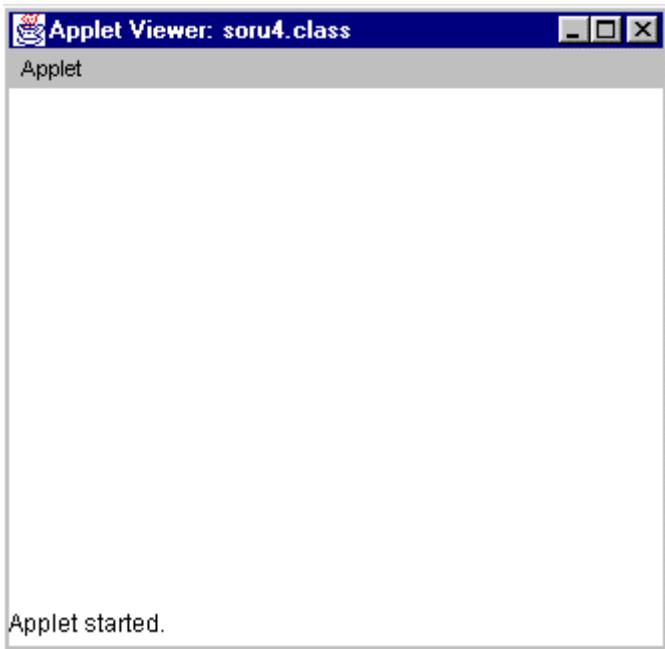


SORU 4

```
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 programın sonuçlarını çiziniz :



SORU 5

aşağıdaki program kodunda ***** ile verilen yerde kare metodunu kullanarak sonucu yazdıracak System.out.println(...) komutunu yeniden yazınız

```
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 yazınız.

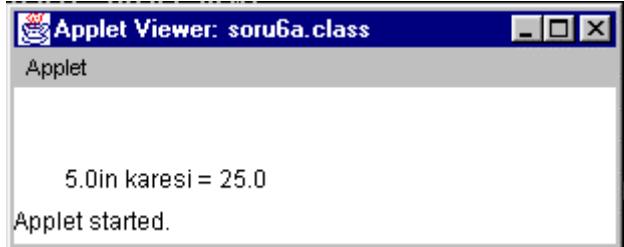
```
import java.applet.Applet; // java applet sinifini cagir
import java.awt.*; // java pencere kullanma sinifini cagir

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(sayı+"in karesi = "+nesne.kare(sayı),25,50);
}
}
```



SORU 7

Verilen programın çıktısını yazınız.

```
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

Aşağıda verilen noktaX sınıfında boş bırakılan
public noktaX(double a, double b)
kurucu metodunu yazınız.

```
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 sınıfını kullanan dogruX sınıfı verilmiştir.
public dogruX(nokta nokta1,nokta nokta2)
kurucu metodunu yazınız.

```
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

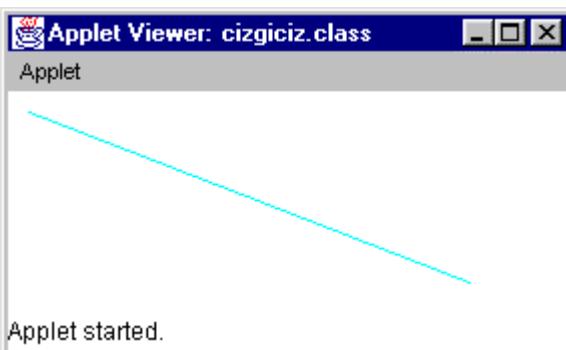
Aşağıdaki programın çıktısını yazınız

```
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 aşağıdaki çıktıyı alacak bir program yazınız. çizgi koordinatlarının ne olduğu önemli değildir.



SORU 11

drawRect metodu kullanarak aşağıdaki çıktıyı alacak bir program yazınız. dikdörtgen koordinatlarının ne olduğu önemli değildir.

SORU 12

Bu programın çıktısını yazınız :

```
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);
    }
}
```

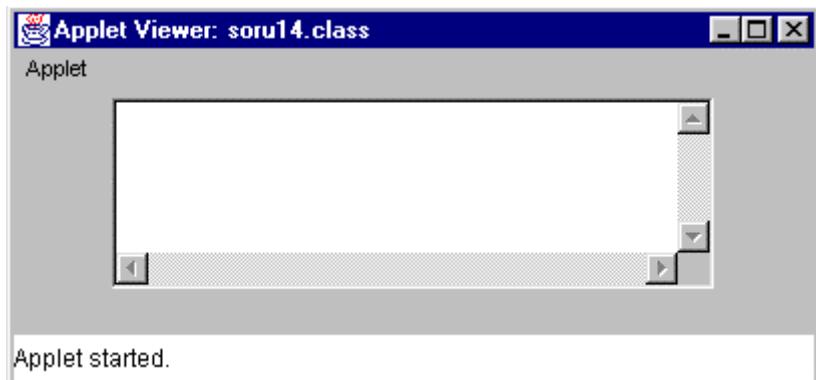
SORU 13

Aplete "Merhaba" yazdırın bir applet programı yazınız.

SORU 14

Aşağıdaki programın sonuçlarını (aplete) yazınız.

```
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\tsonrada 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 sayıların ortalamasını hesaplayan bir application (konsol) programı yazınız.

SORU 16

Aşağıdaki kodun çıktısı ne olacaktır :

```
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

a) Bu programın ne yaptığını izah ediniz

b) program çıktısını yazınız.

```
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 programın çıktısını yazınız.

```
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

Aşağıda verilen **bileşim** metodunun ne yaptığını 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

Aşağıda verilen Tree sınıfının ve
soldal_node_sagdal_sirala();
metodunun ne yaptığını 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 alınmıyacaktır
        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 LİSTESİ

1. Java How to Program, Deitel & Deitel, Second Edition, Prentice Hall, <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 Geliştirme klavuzu, uzmanlar için, Philip Heller, S. Robberts, P. Seymour, T. McGin, Türkçe basım editörü Dr. Cahit Akın, ISBN 975-322-011-1
9. Java, Hazırlayan: Gurup Java, Beta basın yayım dağıtım AŞ, İstanbul 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