JAVA

1.1 Giriş

Internet kullanıcılarının hayatını kolaylaştıran araçlar uzun süre değişime uğramadı.Uzun yıllar Telnet, FTP, e-posta, Usenet haberleri gibi servisler kullanıcıların temel ihtiyaçlarını tatmin etmişti. Ancak bilgisayar teknolojisindeki hızlı gelişme, grafik temelli programlar, ağ hızlarının artması ve bilgisayarların çok ucuzlayıp evlere kadar girmesi ile yepyeni Internet servisleri ortaya çıktı. Bunlardan en olanı önceki bölümde gördüğümüz WWW'dir. Ancak WWW, Internet servislerindeki son nokta değildir. Zira WWW'de bulunan bazı eksikliklerden dolayı yepyeni servislere gereksinim doğmuş ve 1996 yılından itibaren bu yeni servisler Internet'te hızla yaygınlaşmaya başlamıştır.En popüler olanlarından biri de java dır.

Java, WWW sayfalarına nesne temelli programlama yeteneğinin eklenmesini sağlayan ilk dil olmuştur. Java aynı zamanda kullanıcı ile WWW sayfası arasında bir etkileşimin olmasını da sağlar. Kullanıcıların Java ile, WWW üzerinde hazırlanmış olan formların doldurulmasının ve sayfa içindeki bilgilerin okunmasının yanı sıra, oyunlar oynaması, hesap makinasını kullanması, sürekli olarak en son değişiklikleri ile birlikte çeşitli verilen elde etmesi mümkündür. Aşağıda Java ile yapılması mümkün olan işlemlerden bazıları sıralanmıştır:

- Kullanıcı sayfayı çağırdığında gerçek zamanlı olarak çalışan ses bilgisi,
- Sayfanın arka planında çalan müzik,
- Animasyonlar,
- Gerçek zamanlı video görüntüleri,
- Birden çok kişinin oynayabildiği etkileşimli oyunlar.

Java birçok özelliği ile basit bir WWW göz gezdiricisinden çok farklıdır. Bu özelliklerin hepsi çeşitli şekillerde göz gezdiricilere kazandırılabilir.

1.2 Tarihçe

Java konusundaki ilk çalışmalar oldukça eski tarihlere dayanmaktadır. 1970'li yılların

sonunda Bill Joy, bugün kullanılan Java'nın temellerini atmaya başlamıştı. 1990 yılında Bill Joy yayınladığı "Further" başlıklı bir makale

de C++ temelli bir nesne ortamının yaratılmasına yönelik fikirlerini belirtmişti.Bugünkü anlamıyla Java'nın yaratılmasını Nisan 1991 'de kurulan 'Green Proje' adı verilen grup sağlamıştır. Ekim 1992 tarihinde bu grup Sun firma

sının sahibi Olduğu First Person Inc. adındaki firma altında buluştu. 1994 yılına kadar bu firma tarafından yapılan ve başarısızlıkla sonuçlanan çeşitli girişimlerden sonra aynı ekipten Patrick Naughton'nun bir hafta sonu çalışa

rak yarattığı WebRunner isimli prototip göz gezdirici (browser) bu konudaki dönüm noktasını oluşturdu. Ekibin diğer üyelerinin de katkılan ile WebRunner isimli yazılım kısa sürede HotJava'ya dönüştürüldü.

Şekil 1 : Javasoft web sayfası.

1.3 Javanın Özellikleri

Java, dağınık uygulamaların gerçekleştirilmesinde kullanılan bir programlama dilidir. Kullanıcının, sayfanın içeriği ile etkileşimi sağlaması nedeniyle diğer gözgezdiricilerden ayrılır. Java ile gözgezdiricilere, içeriği ve bu içeriğin görüntülenebilmesi için gerekli olan programları aynı zamanda göndermek mümkündür. Böylece göz gezdiricinin , ilgili içeriği görüntülemek için, gerekli özellikleri kendi bünyesinde bulundurmasına gerek kalmaz. Bu da kullanıcıların, ilgili göz gezdiricilerin içeriği desteklemesini beklemeden veya sürekli olarak göz gezdiricisini değiştirmek zorunda kalmadan farklı içerikli bilgileri görüntüleyebilmesini sağlar. Örneğin, HTTP'yi desteklemeyen bir ana bilgisayarda bulunan veri tabanı, WWW üzerinde kullanıma açılmak istendiğinde, ilgili göz gezdiricinin bu özel sistemi desteklemesini beklemekten başka çözüm yoktur. Ancak Java kullanılarak bu desteğin göz gezdirici tarafından verilmesine gerek kalmadan WWW üzerinde uygulamaların gerçekleştirilmesi mümkün olur.Java, aynı zamanda belirli ortamlara bağımlı olarak çalışmaz.

Herhangi bir Java programı, Java desteği bulunan herhangi bir göz gezdirici ile herhangi bir ortamda çalışabilir. Netscape Navigator 2.0 ile Windows 95, Windows NT, MacOS, ve birçok UNIX ortamında Java'nın kullanılması mümkün olmaktadır.

Java sadece WWW için kullanılmaz. Java ile, Fortran, C++ gibi diğer bir programlama dilleri ile yapılabilen hemen hemen her şeyin gerçekleştirilmesi mümkündür. Üstelik Java. bu dillere göre daha basit ve açıktır. Java'nın özelliklerini aşağıdaki gibi özetlemek mümkündür:

Basit

Java içinde gereksiz özelliklerin bulunmamasına ve gerekli olan tüm fonksiyonların da kullanımı kolay bir şekilde bulundurulmasına özen gösterilmiştir.

Object-Oriented

Java içindeki hemen hemen her şey ya bir sınıf, metod ya da bir objedir. Sadece en basit operasyonlar ve veri tipleri (ini, for, while vb) alt-obje seviyesinde bulunur.

Ortam Bağımsız

Java programları, byte kod formatında derlenirler. Yani, interpretenlar yardımıyla Windows 95, Windows NT ve Solanis 2.3 gibi bir çok ortamda okunması ve çalışması mümkündür.

Güvenli

Java kodu virüs tehlikesinden ve dosyaların bozulmasından uzak ortamlarda çalışabilir.

Yüksek Performans

Java'nın hızının C++'ın hızı ile yarışması beklenmektedir.

Multi-Threaded

Basit bir Java programı, birbirinden bağımsız olarak ve sürekli çalışan birçok işleme sahip olabilir.

Server üzerine az yük

Yüksek CPU gücü gerektiren WWW uygulamaları (CGI temelli programlar) Web servenlerini aşırı yüklerler, ama Java temelli uygulamalarda CPU gücü gerektiren hesaplamalar uç bilgisayarda yapıldığı için böyle bir sorun yaratmaz.

Javanın kullanılması için üç şeye ihtiyaç vardır. Bunlar: Java desteği olan bir WWW göz gezdiricisi, kaynak kodu byte koda çeviren Java derleyicisi ve programların çalışmasını sağlayan Java interpreterdir. Bunların dışındaki araçlar kullanılması zorunlu olmayan ancak çalışmaları kolaylaştıran araçlardır.[1]

Java, tümüyle nesneye dayalı olarak tasarlanmış bir programlama dilidir. Yaratılmasındaki temel gaye ortamdan bağımsız, her bilgisayarda çalışabilecek programlar geliştirebilmektir. Özellikle internet gibi farklı mimarilere ve işletim sistemlerine sahip bilgisayarlardan oluşan ağ ortamlarında böyle bir gereksinim vardır.

Derlenmiş Java progrramları, "byte code" adı verilen özel bir formata sahiptirler. Bu formatın işlemci tarafından doğrudan yürütülebilecek makina kodundan farkı bu kodun Java yorumlayıcısı tarafından çalıştırılabilmesidir. Yorumlayıcı yardımıyla çalıştırılabilir olması Java'yı ortamdan bağımsız kılmaktadır. Her ortam için ayrı bir Java yorumlayıcısı mevcuttur ve bu yorumlayıcılar Java kodunu o ortama uygun makina koduna dönüştürür.

Java uygulamaları, WEB hizmeti aracılığıyla sunucu makinalardan istemci makinalara aktarılabilmekte ve Java desteği olan WEB tarayıcıları tarafından çalıştırılabilmektedir. Böylece, Web tarayıcısı olan her bilgisayar internet üzerinde herhangi bir WEB sunucusu altında bulunan Java uygulamasını çalıştırabilmektedir. Bu yöntemle yazılacak istemci / sunucu programlarında istemcide standart bir WEB tarayıcısının olması yeterlidir. WEB tarayıcısına sunucuya ait URL bilgisi girildiğinde Java kodu kendiliğinden çalıştırılmaktadır. Böylece uygulama programlarında bir değişiklik yapılması gerektiğinde değişikliğin sadece sunucu bilgisayarda yapılması yeterli olacaktır.

Java'yı tasarlayanlar C++'dan büyük ölçüde etkilenmişlerdir. Bu etkilenme Java'nın sentaksında da kendini göstermektedir. Ancak iki dil arasında çok temel bazı farklar vardır. Bunların başında Java'nın tümüyle nesneye dayanması gelmektedir. C++ nesneye tabnalı modellemeyi desteklemekle beraber uzantısı olduğu C dilini desteklediği için melez bir yapıdadır.

[1] ÇAĞILTAY,Kürşat,İnternet,sayfa 155-159

Java programları çalıştırılma şekillerine göre 2'ye ayrılır:

- Applet
- Application (uygulama)

Java appletleri WEB tarayıcıları üzerinden yüklenerek çalışırlar. Sunucu bilgisayardan istemciye yüklenerek çalıştırıldıklarından istemci bilgisayarın kaynaklarına erişimleri sınırlandırılmıştır. İstemcinin sabit diskine, yazıcısına .. vs. erişemezler. Bu sınırlama güvenlik nedeniyle oluşturulmuştur.

Java uygulamaları bağımsız olarak çalışan programlardır. Çalıştırılabilmeleri için yerel diske önceden yüklenmiş olmaları gerekir. Buna karşılık Java uygulamaları için hiç bir güvenlik sınırlandırılması yoktur, tüm yerel kaynaklara erişebilirler.

JAVA'nın bazı özellikleri:

1-) Platformdan bağımsız olmasını sağlayan run-time kütüphaneleri sayesinde Windows, Unix ve Macintosh ortamında aynı kod kullanılabilmektedir. Bu özellikle internet programlarında gereklidir.

2-) Java'nın sentaksı C++'nın sentaksına benzemektedir.

3-) Java tam anlamıyla nesneye dayalı bir programlama dilidir. Java'da herşey, sadece bazı temel tipler haricinde, nesne olarak tanımlanır.

4-) C++ ya göre daha kolay hata duzeltilebilir bir dildir. Çünkü Java dili tasarlanırken C++' da sık sık hataya sebep olan durumlar belirlenip, ona göre bu hatalara yer vermeyen bir dil oluşturulmuştur. Java'nın C++'ya göre farklarından önemlileri aşağıda verilmiştir:

i) programcının bellekten yer ayırma (memory allocation) ve geri verme (deallocation) işlemleri yapmasına gerek yoktur. Java' nın önemli bir özelliği olarak bellek otomatikman geri verilir. Bu özelliğe "garbage collection" denir. Programcı bellek kaybını düşünmekten kurtulur.

ii) işaretçi aritmetiği ortadan kaldırılmıştır; gerçek diziler üzerinde işlem yapılması saglanmıştır.

iii) atama işlemi ile koşullardaki test işlemi arasındaki karışıklık ortadan kaldırılmıştır. Bu C/C++'da önemli bir sorundu. Java'da programcı

if (index = 3) .... şeklinde bir komut giremez.

iv) çoklu miras alma (multiple inheritence) kaldırılmıştır, onun yerine "interface" olarak adlandırılan yeni bir yapı tanıtılmıştır.

1.4 JAVA programlama ortamı ve kullanılan araçlar

Java ortamları,

- Java kaynak kodunu "byte code" una dönüştüren bir Java derleyicisi,
- Java programlarını çalıştıran bir Java yorumlayıcısı,
- Java appletlerini çalıştıran bir WEB tarayıcısından oluşur.

Ayrıca program yazmak için herhangi bir text editörune ihtiyaç vardir. Bu editör, örneğin DOS'taki Edit, Windows'daki Notepad vb. editör olabileceği gibi Java için geliştirilmiş özel bir editör (örn. TextPad, WinEdi vb..) de olabilir. Jbuilder, VJ++ gibi Java derleyicilerinin editörleri kullanıcıya kolaylık sağlamaktadır.

Çeşitli Java derleyicileri mevcuttur: SUN'in JDK (Java Development Kit), Microsoft'un VJ++, Borland'in Jbuilder,.. vb. Burada temel olması bakımından JDK 1.1.2 anlatılacaktır.

Tarayıcı olarak Java'nın ilgili sürümünü destekleyen bir WEB tarayıcısı kullanılmalıdır. Örneğin Netscape 4.x, IE 4.x, Sun' ın Hotjava 1.1 gibi.

Kaynak kodlari .java uzantılıdır. Bu kodun derlenmesi sonucu .class uzantili "byte code" lar elde edilir. Applet uygulamalarında Java' yı destekleyen tarayıcının açacağı .html uzantılı dosyada "applet" etiketinin bulunması gerekir. Burada tarayıcı, applet etiketinde belirtilen .class uzantılı "byte code" unu çalıştırmaktadır.

Java programları uzun dosya adlarına sahiptir, örneğin WelcomeApplet.java gibi. Eğer Jdk dizini kullanılıyorsa, Jdk\bin dizini PATH 'de ve Jdk\lib dizini de CLASSPATH 'de tanımlanmış olmalıdır.

Program komut satırından veya editörden çalıştırılabilir. Edit veya Notebook gibi genel amaçlı bir editörde Merhaba.java programı yazıldığında programı DOS komut satırında derlemek için

javac Merhaba.java

girilir, daha sonra çalıştırmak için

java Merhaba.class

girilir.

Javac, Java derleyicisi, .java uzantılı dosyayı derleyip .class uzantılı “byte code” oluşturur. Daha sonra, eğer hata yoksa java, Java yorumlayıcısı, bu “byte code” u çalıştırır.

Özel editörlerde ,örneğin TextPad 'de, program editörün içindeyken derlenir ve çalıştırılır. TextPad editöründe Tools menüsünden Java Compiler seçilirse (veya alt-0 'a basılırsa) o andaki java dosyası derlenir; Run Java seçildiğinde de (alt-1) bu derlenen program çalıştırılır.

Appletviewer, applet'leri test etmek için kullanılan bir programdır. Hazırlanan applet’in nasıl olduğunu görmek için DOS komut satırında şu iki satırın girilmesi yeterlidir:

javac Merhaba.java

appletviewer Merhaba.html

Appletviewer Merhaba.html dosyasında “applet” etiketini arar ve orda belirtilen applet’i çalıştırır.

Java küçük büyük harflere duyarlı bir programlama dilidir; bu nedenle programın adı, derlenirken ve çalıştırılırken doğru olarak yazılmalıdır.

Java derleyicisi 16 MB ‘dan daha az bir bellekte iyi çalışmayabilir, bu durumda bellek yetersiz cevabı verebilir.

1.4.1 Application Örneği: Merhaba.java

```
public class Merhaba
{  public static void main(String[] args)
   {  String greeting[] = new String[2];
      greeting[0] = "Merhaba !";
      greeting[1] = "Pinar Kayaci";
         int i;
      for (i = 0; i < greeting.length; i++)
         System.out.println(greeting[i]);
   }
}
```

TextPad editöründe Java Compiler ile derlendikten sonra, Run Java ile çalıştırıldığında DOS komut ekranında aşağıdaki satırlar gelir:

```
Merhaba !
Pinar Kayaci
Press any key to continue ….
```

1.4.2 Applet Örneği: MerhabaApplet.java

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;
public class MerhabaApplet extends Applet
   implements ActionListener
{  public void start()
   {
    setLayout(new BorderLayout());
```

```
    Label l = new Label(getParameter("greeting"),Label.CENTER);
    l.setFont(new Font("Times", Font.BOLD, 18));
    add(l, "Center");
    Panel p = new Panel();
    Button b1 = new Button("Pinar Kayaci");
    b1.addActionListener(this);
    p.add(b1);
    add(p, "South");
   }
   public void actionPerformed(ActionEvent evt)
   {  String arg = evt.getActionCommand();
      String uName;
      URL u;
      if (arg.equals("Pinar Kayaci"))
         uName = "mailto:pinar@itu.edu.tr";
      else return;
      try
      {  u = new URL(uName);
         getAppletContext().showDocument(u);
      }
      catch(Exception e)
      {  showStatus("Error " + e);
      }
   }
}
```

MerhabaApplet.html dosyası:

```
<HTML>
```

```
<TITLE>WelcomeApplet</TITLE>
<BODY>
<HR>
<APPLET CODE=WelcomeApplet.class WIDTH=500 HEIGHT=200>
<PARAM NAME=greeting VALUE="MERHABA!">
</APPLET>
<HR>
</BODY>
</HTML>
```

Hotjava ’da MerhabaApplet’in görüntüsü:

1.5 JAVA’da Programlama

1.5.1 JAVA’nın sentaksı:

/* Basit bir Java programı */

```
public class Ornek1
{  public static void main(String[] args)
   {  System.out.println("Ilk Java programi! ");
   }
}
```

Bu program basit bir şekilde “ ” işaretleri arasındaki yazıyı DOS ekranında gösterir.

Java büyük küçük harfe duyarlıdır, yani örneğin main yerine Main girilirse derleyici hata verir. Bu konuda dikkat eidlmelidir!

Örnekteki public terimi, bu kodu diğer kısımlardaki hangi kodların kullanabileceğini belirleyen bir erişim niteleyicisidir. public bu sınıf paketi dışındaki tüm programların da bu kodu kullanmalarına izin verir.

Java’da herşey sınıf olarak tanımlandığından class terimi her zaman kulanılacaktır. Bu terimden sonra sınıfın adı gelmektedir. .java uzantılı bu dosyanın adı tanımlanan sınıf adı ile aynı olmalıdır. Bu nedenle örnekteki dosyanın adı Ornek1.java, derlenen dosyanın adı da Ornek1.class ’dır.

Burada “{“ ve ”}” karakterleri C/C++ ‘de olduğu gibi blok başı ve sonunu göstermektedir. ,

Java uygulamalarında (application), örnekteki gibi main metodunun mutlaka bulunması gerekir.

Java ve C++ sınıfkları benzer olmalarına rağmen bazı önemli farklar mevcuttur: Java’da bütün fonksiyonlar bir sınıfın üyesidir, üye fonksiyonları yerine metod olarak geçerler. Bu nedenle her main fonksiyonu bir sınıfın içinde olmalıdır. C++2Daki statik üye fonksiyonları nesneler üzerinde işlem yapmayan fonksiyonlardır, Java’da main her zaman static olarak tanımlanır. C/C++’da olduğu gibi void terimi fonksiyonun geriye değer döndürmediğini ifade eder.

Java’da her ifadenin sonunda “;” ve ancak bu işaret ile ifade sonlanır.

Bu örnekte biz System.out nesnesini ve onun println metodunu kullanıyoruz.

Java'da metodların 0, 1 veya daha fazla parametresi olabilir. Metodun hiç parametresi olmasa bile parantez yazılmalıdır.

1.5.2 Açıklamalar

C/C++ ile benzer şekilde bir satırlık açıklama için satırın başında // kullanılabileceği gibi, birden fazla satır süren açıklamaların başına /* ve sonuna */ yazılır.

Java'da bunlara ek olarak /** ile başlayan ve */ ile biten bir açıklama türü geliştirilmiştir. Bu iki karakter katarı arasındaki açıklamalar o progamın dokümantasyonda otomatik olarak yer alır.

1.6 Veri Yapıları

Java'da 8 veri tipi vardır. Bunun altısı sayısal veri tipleridir:

Tipi	Büyüklüğü
int	4 byte
short	2 byte
long	8 byte
byte	1 byte
float	4 byte
double	8 byte

16 tabanındaki tamsayıların başına 0x gelir (örn. 0xCAFE). long integer olarak tanımlanan sayıların sonuna da L gelir (örn. 4000000000L).

Kayan noktalı sayılarda ise sayının sonuna F gelirse float tipinden, gelmezse double tipinden bir veridir. Örneğin 4.201F float, 4.201 ise double 'dır.

1.6.1 char (karakter tipi)

Karakter tipi verilerin başında ve sonuda " ' " işareti vardır. Karakterler Unicode karakter setinden seçilir. Unicode uluslarası dillerin tüm karakterlerini içeren ve her

karakter için 2 byte uzunluğundaki bir kodlama yöntemidir. \u ile bir karakterin “unicode” u yazıldığında o karakter elde edilir. ASCII karakter seti Unicode'un ilk 255 karakterine karşıý düşer.

Bazı tuşlar için tanımlanmış kısa ifadeler mevcuttur, örneğin:

\b	backspace	\u0008
\t	tab	\u0009
\n	satır sonu	\u000a
\r	enter	\u000d
\"	çift tırnak	\u0022
\'	tek tırnak	\u0027
\\	backslash	\u005c

1.6.2 boolean (mantıksal tip)

Mantıksal veriler false veya true değeri alırlar. C’de olduğu 0 false, 0 harici true gibi bir dönüşüm yoktur.

1.6.3 Değişkenler

Java'da değiiken isimleri sınırlı değildir. Değişken tanımlanması şu şekilde olur:

değişken_tipi değişken_adı

Örneğin:

```
int a, b;

char ch;
```

1.6.4 Atamalar ve Başlangıç değerinin atanması

```
char ch;

ch = 'a';
```

int i = 10 ; // C/C++'dan gelen bir özellik, değişkene başlangıç değerinin atanması

1.6.5 Sayısal tipler arasında dönüşüm

İkili işlemlerdeki değişkenler için değişkenlerden birinin tipi diğerinin tipini kapsıyorsa kapsayan tip kabul edilir. Örneğin biri double ise diğeri de double kabul edilir, aksi halde biri float ise diğeri de float kabul edilir. Biri long ise diğeri de long kabul edilir .. vs.

C'deki tip zorlaması (type casting) Java'da da vardır. Örneğin:
double x = 2.333;
int nx = (int) x;

1.6.6 Sabitler

Java 'da yerel sabit yoktur, global sabitler şu de şekilde tanımlanabilir:
public static final double max = 100;

Yalnız bu satır sınıf içinde olmalı; fakat hiç bir metodun içinde olmamalı. O zaman o sınıf içindeki büytün metodlar bu sabiti kullanablir.

1.6.7 Operatörler

+, -, *, / kullanılır.

(ikili operatör)= şeklinde bir kısaltma da kullanılabilir:

x = x + 4 ; yerine

x += 4;

1.6.8 Üsler

Math sınıfına ait pow metodu kullanılır.

double y = Math.pow(x,a); // y= x^a dır.

1 azaltma veya 1 arttırma

x değişkenini 1 arttırmak için x++, 1 azaltmak için x-- ifadesi kullanılır.

Aslında bu operatörlerin iki formu vardır. Değişkenden sonra yazılan ve önce yazılan.

int x = 1;

int y = x++; // y = 1 ve x = 2 dir, burda önce x'in değeri atanır, sonra x 1 arttırılır

int y = ++x; // y = 2 ve x = 2 dir, burda x'in değeri 1 arttırılır sonra değeri atanır

1.6.9 Ilişkisel ve mantıksal operatörler

İlişkisel operatörler: ==, !=, <, >, <=, >=
C++ gibi Java da mantıksal ve için && , veya için || kullanmaktadır.

1.6.10 Bit düzeyindeki operatörler

Maskelemek için kullanılan operatörlerdir.

& ve

| veya

^ darveya (xor)

~ değil

>> sağa öteleme

<< sola öteleme

>>> yüksek anlamlı bitleri 0 ile doldurur.

1.6.11 Operatör Hiyerarşisi (soldan sağa doğru)

[] . () (method çağrısı)

new (tip zorlama) () - + -- ++ ~ !

* / %

+ -

<< >> >>>

< <= > >=

== !=

&

^

|

&&

||

?:

= += -= *= /= %= &= |= ^= <<= >>= >>>=

1.6.12 Karakter Katarları

string anahtar kelimesi kullanılır.

+ operatörü ile katarın sonuna ekleme yapılabilir:

String message = "merhaba " + "!";

Katarın içinde bir alt katar seçmek için substring metodu kullanılır.

String greeting = “Hello”;

String s = greeting.substring(0,4); // s=”Hell” dir

Ilk parametre başlangıç pozisyonunu, ikince parametre kopyalamak istenmeyen bitiş pozisyonunu belirtir. Katarın ilk harfinin pozisyonu 0 dır.

1.6.13 Katarların karşılaştırılması

Bunun için equals metodu kullanılır.

s.equals(t)

dönüş değeri olarak true veya false gelir.

== operandının kullanılması yanlış olur; çünkü bu operand sadece iki katarın aynı yerde olup olmadığına bakar.

Mantıksal ile sayısal değerler arasında işlem yapmayın!

1.6.14 Katar Fonksiyonları

char charAt(int index)

int compareTo(String anotherString)

boolean endsWith(String suffix)

boolean equals(Object anObject)

boolean equalsIgnoreCase(String anotherString)

int indexOf(String str)

int indexOf(String str, int fromIndex)

int lastIndexOf(String str)

int lastIndexOf(String str, int fromIndex)

int length()

String replace(char oldChar, char newChar)

boolean startsWith(String prefix)

String subsrting(int beginIndex)

String subsrting(int beginIndex, int endIndex)

String toLowerCase()

String toUpperCase()

String trim()

1.6.15 Veri okumak

ReadDoubleTest.java örneğinde double bir sayının nasıl okunacağı gösterilmiştir.

Örnek: ReadDoubleTest.java

```
import java.io.*;
import java.text.*;
public class ReadDoubleTest
  // shows how to read a double the hard way
{  public static void main(String[] args)
  {  System.out.println
       ("Enter a number, I'll add two to it.");
     double x; // the number we wish to read
     try
     {  InputStreamReader isr
          = new InputStreamReader(System.in);
        BufferedReader br
          = new BufferedReader(isr);
        String s = br.readLine();
        DecimalFormat df = new DecimalFormat();
        Number n = df.parse(s);
        x = n.doubleValue();
     }
     catch(IOException e)
```

```
    {  x = 0;
    }
    catch(ParseException e)
    {  x = 0;
    }
    System.out.println(x + 2);
  }
}
```

Console sınıfının 3 tane metodu (fonksiyonu) vardır:

- ReadWord(String prompt)
- ReadInt(String prompt)
- ReadDouble(String prompt)

1.6.16 Çıkış formatı

Sayısal verilerin çıkış formayı için Java.text paketinin 3 metodu kullanılabilir.

NumberFormat.getNumberInstance() // sayı değeri

NumberFormat.getCurrencyInstance()// para birimi ile birlikte çıkış

NumberFormat.getPercentInstance() // 100'e bölünüp, % işareti ile çıkış

Çıkış almak için sayı karakter katarı olarak saklanır. Bunun için format metodu kullanılr.

Örnek:

```
double          x=10000.0/3.0;
NumberFormat nf=NumberFormat.getNumberInstance();
```

String fx=nf.format();

System.Out.println(fx);

U.S. veya Alman veya başka stamdartlara göre de sayılar formatlanabilir. Bunun için Locale tipi kullanılır. Örneğin Alman formatında elde etmek için

NumberFormat nf=NumberFormat.getNumberInstance(Locale.German);

Kendi formatınızı yaratmak için DecimalFormat tanımlanabilir:

DecimalFormat df=DecimalFormat("0.######");

System.out.printle(sdf.format(x));

Buradaki semboller ve anlamları aşağıda verilmiştir.

Sembol	Anlamı
0	Bir basamak
#	Bir basamak, gereksiz 0'ları göstermez
.	Basamak ayıracı
,	Gruplama ayıracı
;	Pozitif ve negatif sayılar için formatı ayırır
-	Negatif
%	100'e bölüp, yüzde olarak gösterir
Başka bir sembol	Çıkışta bu sembolü gösterir

Aşağıdaki Mortgage.java örneğinde CoreJava'nın console sınıfı kullanılmıştır.

Mortgage.java

import corejava.*;

```
import java.text.*;
public class Mortgage
{  public static void main(String[] args)
   {  double principal;
      double yearlyInterest;
      int years;
      principal = Console.readDouble
         ("Loan amount (no commas):");
      yearlyInterest = Console.readDouble
         ("Interest rate in % (ex: use 7.5 for 7.5%):")/100;
      years = Console.readInt("The number of years:");
     double monthlyInterest = yearlyInterest / 12;
      double payment = principal * monthlyInterest
         / (1 - (Math.pow(1/(1 + monthlyInterest),
            years * 12)));
      System.out.println("Your payment is ");
      NumberFormat nf = NumberFormat.getCurrencyInstance();
      System.out.println(nf.format(payment));
   }
}
```

1.6.17 Kontrol Akışı

C/C++ ‘daki kontrol akış terimleri ile aynıdır.

- if

```
if (koşul)
  {blok1}
else
  {blok2}
```

(Java'da 1/x gibi bir ifade x=0 ise hesaplanmaz dolayısıyla divide-by-zero hatası oluşmaz.)

- while

```
while (koşul)
      {blok}
```

1.6.18 Örnek :Retirement.java

```
import corejava.*;
public class Retirement
{  public static void main(String[] args)
  {  double goal;
     double interest;
     double payment;
     int years = 0;
     double balance = 0;
     goal = Console.readDouble
       ("How much money do you need to retire?");
     payment = Console.readDouble
```

```
      ("How much money will you contribute every year?");
    interest = Console.readDouble
      ("Interest rate in % (ex: use 7.5 for 7.5%):") / 100;
    while (balance < goal)
    {  balance = (balance + payment) * (1 + interest);
      years++;
    }
    System.out.println
      ("Your can retire in " + years + " years.");
  }
}
```

- do

```
do {blok} while (koşul)
```

1.6.19 Örnek: SquareRoot.java

```
import corejava.*;
public class SquareRoot
{  public static void main(String[] args)
  {  double a = Console.readDouble("Please enter a number:");
    double xnew = a / 2;
    double xold;
    do
    {  xold = xnew;
```

```
        xnew = (xold + a / xold) / 2;
        System.out.println(xnew);
      }
      while (Math.abs(xnew - xold) > 1E-4);
   }
}
```

- for

```
for (ifade; deyim1; deyim2)
      {blok}
```

ile aşağıdaki aynıdır:

```
{ ifade;
  while(deyim1)
  { blok;
    deyim2; }
  }
```

1.6.20 Örnek : MortgageLoop.java

```
import corejava.*;
public class MortgageLoop
{  public static void main(String[] args)
   {  double principal;
      double yearlyInterest;
      int years;
```

```
      principal = Console.readDouble
        ("Loan amount (no commas):");
      yearlyInterest = Console.readDouble
        ("Interest rate in % (ex: use 7.5 for 7.5%):") / 100;
      years = Console.readInt
        ("The number of years:");
      double y;
      for (y = yearlyInterest - 0.01;
        y <= yearlyInterest + 0.01; y += 0.00125)
      {  double monthlyInterest = y / 12;
        double payment = principal * monthlyInterest
          / (1 - (Math.pow(1/(1 + monthlyInterest),
            years * 12)));
        Format.print(System.out,
          "With rate %6.3f", 100 * y);
        Format.print(System.out,
          "%%, your monthly payment is $%10.2f\n", payment);
      }
    }
}
```

1.6.21 switch

Bir değişkenin belirli değerleri için farklı işlemler yapılacaksa switch kullanılır.

```
switch (seçim)
 { case 1:
   ……
  break;
  case 2:
  ……
  break;
  default:  // isteğe bağlı
  ……
  break;
}
```

1.6.22 Etiketler

Java'da c/C++'da olmayan bir özellik olarak etiket kavramı tanımlanmıştır. Bunu için etiketin sonuna ":" işareti gelir, ve break anahtar kelimesi ile bu etikete dallanılabilir. Örneğin:

```
Basla:
while ( ……)
{ for (……)
 { if (……)
    {  ……
       break basla;
    }
```

```
  }
}
```

1.6.23 Class Metodları – Kullanıcı tarafından tanımlanan fonksiyonlar

Bir metodun tanımı mutlaka bir sınıfın içinde olmalıdır. Java'da global fonksiyon kavramı yoktur!

Şimdilik public static metodları göreceğiz; aslında çok çeşitli metod tipleri vardır.

Örnek olarak LotteryOdds.java programını inceleyelim. Diyelim ki biz 1'den n'e kadar sayılar arasından m tane sayı seçeceğiz ve bizim aradığımız sayıları bulma şansımızı hesaplamak için örnekteki lotteryOdds metodu kullanılabilir.

Sayısal bir örnek verirsek, 1 den 50'ye kadar sayı arasından belirli 6 tane seçilirse (50.49.48.47.46.45)//(1.2.3.4.5.6) seçenek vardır.

1.6.24 Örnek: LotteryOdds.java

```
import corejava.*;

public class LotteryOdds

{  public static long lotteryOdds(int high, int number)

   {  long r = 1;

      int i;

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

      {  r = r * high / i;

         high--;

      }

      return r;

   }
```

```
  public static void main(String[] args)
  {  int numbers = Console.readInt
       ("How many numbers do you need to draw?");
     int topNumber = Console.readInt
       ("What is the highest number you can draw?");
     long oddsAre = lotteryOdds(topNumber, numbers);
     System.out.println
       ("Your odds are 1 in " + oddsAre + ". Good luck!");
  }
}
```

Burada

public static long lotteryOdds(int high, int number)

ifadesi kullanılmıştır; ifadedeki long metodun geri döndürdüğü değerin tipidir. Metod geriye değer döndürmezse void yazılır. Return metoddan çıkışı sağlar, yanındaki değişken veya değer ise metodun dönüş değeridir.

C++'daki fonksiyonlar ile Java'daki metodlar benzer yapıda olmalıran rağmen aralarında öenmli farlar vardır: Java'da fonksiyonların kullanılmadan önce tanımlanmaları gerekmez; Java'ad işaretçi ve referans parametreleri yoktur dolayısıyla bir değişkenin konumu (location) fonksiyon üzerinden aktarılamaz. C++'da olduğu gibi birden dazla fonksiyonun aynı isme sahip olması mümkündür.

Public sınıfları diğer sınıflar tarafından çağrılabilen sınıflardır.

1.6.25 Class değişkenleri

Bir sınıfın içinde kullanılan değişkenler main metodundan önce sınıfın içinde tanımlanır. Bu değişkenlere eğer başlangıç değeri verilmezse ototmatikman 0 değeri atanır. (Object için NULL, boolean içn false vb.)

1.6.26 Rekürsif çağırma

Bazı problemlerin çözümünde rekürsif metod çağırma kullanılabilir. Bu problemin basit parçalara bölünmesi ve önce basit parçaların çözülmesi sonra da bu çözümleri kullanarak genel problemin çözülmesi anlamındadır.

Daha açık olamsı için bir önceki probelim göz önünde bulundurabiliriz. Diyelim ki 50 sayıdan bir tane seçtik, öyleyse geriye 49 tane sayıdan 5 tane seçim kalır (= lotteryOdds(49,5)). Yani daha basit bir problem elde edilir. Bu sayıyı seçerken 50 türlü seçim yapılabileceğinden 6 sayı seçmek için toplam 50*lotteryOdds(49,5) tane seçim yapılabilir. Bu ndednle bu sonucu 6'ya bölmeliyiz.

high 50 ve number 6 olmak üzere bu problem rekürsif olarak aşağıdaki şekilde yazılabilir:

```
public static long lotteryOdds(int high, int number)

{

  if (number <=0 ) return 0;

  else if (number ==1)) return high;

  else return high * lotteryOdds( high – 1, number -1) / number;

}
```

1.6.27 Diziler

Java'da array ile tanımlanan dizilerin boyutları baştan tanımlanır ve sonradan değiştirilemez. Eğer program çalışırken dizinin boyutunun değiştirilmesi gerekirse değişik bir Java nesnesi olan vector tipi kullanılmalıdır.

main metodunda kulanılan bir dizi : String[] args

Dizilerin yaratılmasında new terimi kullanılmalıdır. Örneğin

```
int[] arrayofInt=new int[100]
```

Burada dizinin ilk elemanı arrayOfInt[0] ve son elemanı arrayOfInt[99] 'dur. Dizinin indisleri 0'dan başlar.

Diziler doğrudan değer verilerek yaratılabilr:

int[] smallPrimes={2,3,5,7,,11,13};

Java'da system sınıfınad dizi kopyalama için kullanışlı bir metod vardır:

System.arraycopy(kaynakDizi, kaynakpPzisyon, hedefDizi, hedefPozisyon, kopyalancakElemanSayısı);

Aşağıdaki örnekte parametre olarak aktarılan bir tamsayı dizisi sıralanmaktadır.

1.6.28 Ornek: ShellSort.java

```
public class ShellSort
{  public static void sort(int[] a)
  {  int n = a.length;
     int incr = n / 2;
     while (incr >= 1)
     {  for (int i = incr; i < n; i++)
        {  int temp = a[i];
           int j = i;
           while (j >= incr && temp < a[j - incr])
           {  a[j] = a[j - incr];
              j -= incr;
           }
           a[j] = temp;
        }
        incr /= 2;
```

```
    }
  }
  public static void main(String[] args)
  {  // make an array of ten integers
    int[] a = new int[10];
    int i;
    // fill the array with random values
    for (i = 0; i < a.length; i++)
      a[i] = (int)(Math.random() * 100);
    // sort the array
    sort(a);
    // print the sorted array
    for (i = 0; i < a.length; i++)
      System.out.println(a[i]);
  }
}
```

Metoddan geri dönüş değerinin dizi olması da oldukça kullanışlıdır, özellikle de metod bir dizi değer hesaplıyorsa. Aşağıdaki LotteryDrawing.java programı 1 den high'a kadar sayı arasından number tane sayının seçilmesini simule eder ve sonuçta elde edilen number tane elemanı ekrana yazdırır.

1.6.29 Örnek: LotteryDrawing.java

```
import corejava.*;

public class LotteryDrawing
```

```
{  public static int[] drawing(int high, int number)
   {  int i;
      int numbers[] = new int[high];
      int result[] = new int[number];
      // fill an arrays with numbers 1 2 3 . . . high
      for (i = 0; i < high; i++) numbers[i] = i + 1;
      for (i = 0; i < number; i++)
      {  int j = (int)(Math.random() * (high - i));
         result[i] = numbers[j];
         numbers[j] = numbers[high - 1 - i];
      }
      return result;
   }
   public static void main(String[] args)
   {  int numbers = Console.readInt
         ("How many numbers do you need to draw?");
      int topNumber = Console.readInt
      ("What is the highest number you can draw?");
         int[] a = drawing(topNumber, numbers);
      ShellSort.sort(a);
      System.out.println
```

```
("Bet the following combination. It'll make you rich!");
    int i;
    for (i = 0; i < a.length; i++)
      System.out.println(a[i]);
  }                                                                    } [2]
```

1.1.1.1 [2] KAYA,Pınar,JAVA EĞİTİM SEMİNERLERİ I ,1996.