Atamalar

```
int a ;
a=4 ;    // doğru bir atama
4=a ;    // yanlış bir atama!
```

Temel (Primitive) Tiplerde Atama

```
int a, b ;
a=4 ;
b=5 ;
a=b ;
```

Sonuç : a=5, b=5

Dosya İsimleri

- Fiziksel dosya ismi ile **`public`** sınıfın ismi aynı olmalı.

java Operatörleri

- Yordamlar parametre alırlar.
- Alınan bu paremetreler ile yordam içerisinde işlemler gerçekleşir.
- Operatörler programlama dillerinin en temel işlem yapma yeteneğine sahip simgesel isimlerdir.
  - Aritmetik Operatör
  - İlişkisel Operatör
  - Mantıksal Operatörler
  - Bit düzeyinde (*bitwise*) Operatörler

- Operatörler bir veya daha fazla değişken üzerinden işlemler gerceklestirirler.
  - İşlem gerçekleştirmek için tek bir değişkene ihtiyaç duyan operatörlere tekli operatör (unary operator)
  - İşlem gerçekleştirmek için iki değişkene ihtiyaç duyan operatörlere ikili operatör (binary operator)
  - İşlem gerçekleştirmek için üç adet değişkene ihtiyaç duyan operatörlere ise üçlü operatör (ternary operator) denir (bir adet var).

Aritmetik Operatörler

Operatör	Kullanılış	Açıklama
+	`değişken1 + değişken2`	değişken1 ile değişken2 yi toplar
-	`değişken1 - değişken2`	değişken1 ile değişken2 yi çıkarır
*	`değişken1 * değişken2`	değişken1 ile değişken2 yi çarpar
/	`değişken1 / değişken2`	değişken1 ,değişken2 tarafından bölünür
%	`değişken1 % değişken2`	değişken1 in değişken2 tarafından bölümünden kalan hesaplanır.

"+" ve "–" Operatörleri

Operatör	Kullanılış Şekli	Açıklama
+	+ değişken	Eğer değişken *char*, *sekizli* (*byte*) veya *short* tipinde ise *int* tipine dönüştürür.
-	- değişken	Değişkenin değerini negatif yapar (-1 ile çarpar).

Dönüştürme *(Casting)* İşlemi

- Bir temel (*primitive*) tip, diğer bir temel tipe dönüştürülebilir, fakat oluşacak değer kayıplarından kodu yazan kişi sorumludur .

```
1  public class IlkelDonusum {
2
3      public static void main(String args[]) {
4      int a = 5 ;
5      double b = (double) a ;
6
7      double x = 4.15 ;
8      int y = (int) x ;
9      long z = (long) y ;
10
11     System.out.println("b = " + b + " y = " + y + " z = " + z);
12     }
13 }
```

String (+) Operatörü

- "+" operatörü **String** tiplerde birleştirme görevi görür.
- Eğer bir ifade **String** ile başlarsa , onu takip eden tiplerde otomatik olarak String nesnesine dönüştürülür.

```
1  public class OperatorTest {
2
3      public static void main(String args[] ) {
4      char  a = 'a' ;
5      int b = +a ; //otomatik olarak int ilkel tipine cevrildi
6      int c = -b ; // degeri negatif yapti
7      System.out.println("a = " + a );
8      System.out.println("b = " + b );
9      System.out.println("c = " + c );
10     }
11 }
```

```
1  public class OtamatikCevirim {
2
3      public static void main(String args[]) {
4      int x = 0, y = 1, z = 2;
5      System.out.println("Sonuc =" + x + y + z);
6      }
7  }
```

- Sonuc = 012

- **String** bir ifadeden sonra gelen tamsayılar görüldüğü üzere toplanmadı.

- Direk **String** nesnesine çevrilip ekrana çıktı olarak gönderildiler.

Bir Arttırma ve Azaltma

- Java dilinde C dilinde olduğu gibi birçok kısaltmalar vardır.
- Bu kısaltmalar hayatı bazen daha güzel bazen ise çekilmez kılabilir.

Bir Arttırma ve Azaltma Tablosu

Operatör	Kullanılış Şekli	Açıklama
++	`değişken++`	Önce değişkenin değerini hesaplar sonra değişkenin değerini bir arttırır.
++	`++değişken`	Önce değişkenin değerini arttırır sonra değişkenin değerini hesaplar.
--	`değişken--`	Önce değişkenin değerini hesaplar sonra değişkenin değerini bir azaltır.
--	`--değişken`	Önce değişkenin değerini azaltır sonra değişkenin değerini hesaplar.

```
1  public class OtomatikArtveAz {
2
3      static void ekranaYaz(String s) {
4          System.out.println(s);
5      }
6
7      public static void main(String[] args) {
8          int i = 1;
9          ekranaYaz("i : " + i);
10         ekranaYaz("++i : " + ++i); // onek arttirim
11         ekranaYaz("i++ : " + i++); // sonek -arttirim
12         ekranaYaz("i : " + i);
13         ekranaYaz("--i : " + --i); // onek-azaltim
14         ekranaYaz("i-- : " + i--); // sonek-azaltim
15         ekranaYaz("i : " + i);
16     }
17 }
```

```
i : 1
++i : 2
i++ : 2
i : 3
--i : 2
i-- : 2
i : 1
Press any key to continue...
```

İlişkisel Operatörler

- İlişkisel operatörler iki değeri karşılaştırarak bu değerler arasındaki mantıksal ilişkiyi hesaplarlar.
- Örneğin iki değer birbirine eşit değilse "`5==8`"
- Bu ilişki çerçevesinde hesaplanan değer ***false*** olacaktır.

İlişkisel Operatörler Tablosu

Operatör	Kullanılışı	true değeri döner eğer ki...
>	`değişken1 > değişken2`	değişken1 , değişken2'den büyükse
>=	`değişken1 >= değişken2`	değişken1 , değişken2'den büyükse veya eşitse
<	`değişken1 < değişken2`	değişken1 , değişken2'den küçükse
<=	`değişken1 <= değişken2`	değişken1 , değişken2'den küçükse veya eşitse
==	`değişken1 == değişken2`	değişken1 , değişken2'ye eşitse
!=	`değişken1 != değişken2`	değişken1 , değişken2'ye eşit değilse

```java
public class IliskiselDeneme {

    public static void main(String[] args) {

        //bir kac sayi
        int i = 37;
        int j = 42;
        int k = 42;
        System.out.println("Degisken degerleri...");
        System.out.println("    i = " + i);
        System.out.println("    j = " + j);
        System.out.println("    k = " + k);

    //Buyuktur
        System.out.println("Buyuktur...");
        System.out.println("    i > j = " + (i > j));      //false - i , j den kucuktur
        System.out.println("    j > i = " + (j > i));      //true  - j , i den buyuktur
        System.out.println("    k > j = " + (k > j));     //false - k , j 'ye esit

    //Buyuktur veya esittir
        System.out.println("Buyuktur veya esittir...");
        System.out.println("    i >= j = " + (i >= j));    //false - i , j den kucuktur
        System.out.println("    j >= i = " + (j >= i));    //true  - j , i den buyuktur
        System.out.println("    k >= j = " + (k >= j));    //true - k , j 'ye esit

    //Kucuktur
        System.out.println("Kucuktur...");
        System.out.println("    i < j = " + (i < j));      //true - i , j den kucuktur
        System.out.println("    j < i = " + (j < i));      //false - j , i den buyuktur
        System.out.println("    k < j = " + (k < j));      //false - k , j 'ye esit

```

```
32      //Kucuktur veya esittir
33          System.out.println("Kucuktur veya esittir...");
34          System.out.println("    i <= j = " + (i <= j));    //true - i , j den kucuktur
35          System.out.println("    j <= i = " + (j <= i));    //false - j , i den buyuktur
36          System.out.println("    k <= j = " + (k <= j));    //true - k , j 'ye esit
37
38      //Esittir
39          System.out.println("Esittir...");
40          System.out.println("    i == j = " + (i == j));    //false - i , j den kucuktur
41          System.out.println("    k == j = " + (k == j));    //true - k , j 'ye esit
42
43      //Esit degil
44          System.out.println("Esit degil...");
45          System.out.println("    i != j = " + (i != j));    //true - i , j den kucuktur
46          System.out.println("    k != j = " + (k != j));    //false - k , j 'ye esit
47
48      }
49  }
```

```
Degisken degerleri...
    i = 37
    j = 42
    k = 42
Buyuktur...
    i > j = false
    j > i = true
    k > j = false
Buyuktur veya esittir...
    i >= j = false
    j >= i = true
    k >= j = true
Kucuktur...
    i < j = true
    j < i = false
    k < j = false
Kucuktur veya esittir...
    i <= j = true
    j <= i = false
    k <= j = true
Esittir...
    i == j = false
    k == j = true
Esit degil...
    i != j = true
    k != j = false
Press any key to continue...
```

Nesnelerin Karşılaştırılması

- Nesnelerin eşit olup olmadığı (= =) veya (! =) operatörleri ile test edilebilir mi ?

```
1  public class Denklik {
2    public static void main(String[] args)
3    {
4        Integer s1 = new Integer(47);
5        Integer s2 = new Integer(47);
6        System.out.println(s1 == s2);
7        System.out.println(s1 != s2);
8        }
9    }
```

```
false
true
Press any key to continue...
```

Uygulama

- Peki bir önceki örneği **Integer** nesneleri yerine temel tip olan **int** tipini kullansaydık sonuç nasıl olurdu?

```
1  public class IntIcinDenklik {
2      public static void main(String[] args) {
3          int s1 = 47;
4          int s2 = 47;
5          System.out.println(s1 == s2);
6          System.out.println(s1 != s2);
7      }
8  }
```

```
true
false
Press any key to continue...
```

Mantıksal Operatörler

- Mantıksal operatörler birden çok karşılaştırma işleminin birleştirip tek bir koşul ifadesi haline getirilmesi için kullanılır.

Mantıksal Operatörler Tablosu

Operatör	Kullanılış Şekli	true değeri döner eğer ki.....
&&	değişken1 && değişken2	Eğer hem değişken1 hemde değişken2 true ise ; (değişken2'yi duruma göre hesaplar*)
||	değişken1 || değişken2	değişken1'in veya değişken2'in true olması ;(değişken2'yi duruma göre hesaplar*)
!	! değişken	Eğer değişken false ise
&	değişken1 & değişken2	Eğer hem değişken1 hemde değişken2 true ise ;
|	değişken1 | değişken2	değişken1'in veya değişken2'in true olması ;
^	değişken1 ^ değişken2	Eğer değişken1 ve değişken2 birbirlerinden farklı ise; ör: değişken1 true ,değişken2 false ise*

Uygulama

```
1  public class KosulOp {
2
3      public static void main( String args[] ) {
4
5        int a = 2 ;
6        int b = 3 ;
7        int c = 6 ;
8        int d = 1 ;
9        /*
10       (a < b) = bu ifadenin true oldugunu biliyoruz
11       (c < d) = bu ifadenin false oldugunu biliyoruz
12       */
13       System.out.println(" (a < b) && (c < d) --> " + ( (a < b) && (c < d) ) );
14       System.out.println(" (a < b) || (c < d) --> " + ( (a < b) || (c < d) ) );
15       System.out.println(" ! (a < b)  --> " + (   ! (a < b)  ) );
16       System.out.println(" (a < b) & (c < d) --> " + ( (a < b) & (c < d) ) );
17       System.out.println(" (a < b) | (c < d) --> " + ( (a < b) | (c < d) ) );
18       System.out.println(" (a < b) ^ (c < d) --> " + ( (a < b) ^ (c < d) ) );
19     }
20 }
```

```
 (a < b) && (c < d) --> false
 (a < b) || (c < d) --> true
 ! (a < b)  --> false
 (a < b) & (c < d) --> false
 (a < b) | (c < d) --> true
 (a < b) ^ (c < d) --> true
Press any key to continue...
```

```java
import java.awt.*;
import java.io.*;
import java.lang.*;
import java.*;

class faktoriyel
    {
public static void main(String args[])
    throws java.io.IOException
    {
        BufferedReader konsol = new BufferedReader(
            new InputStreamReader(System.in));

        System.out.print(" Sayi giriniz:");

        String giris=konsol.readLine();

        Double d=new Double(giris);
        long sayi=d.longValue();// uzun tam sayı int ten buyuk
                                // tam sayıla için kullnılır.
        double fak=1;
        for (int i=1; i<=sayi; i++){
            fak=fak*i;
        }
    System.out.println("Sayinin faktoriyeli:"+fak);
    }}
```

Kısa Yollar

- `i = i + 1 ;` yerine.
- `i += 1 ;` kullanılabilir.

- `i = i * 1 ;` yerine
- `i *= 1 ;` kullanılabilir.

Kontrol İfadeleri

- Kontrol ifadeleri bir uygulamanın hangi durumlarda ne yapması gerektiğini belirtir.
- Java programlama dilinde toplam 4 adet kontrol ifade çeşidi bulunur.

Kontrol İfadeleri Tablosu

İfade Tipi	Anahtar Kelime
Döngü	`while, do-while , for`
Karar verme	`if-else, switch-case`
Dallandırma	`break, continue, label, return`
İstisna yakalama	`try-catch-finally, throw`

false

Uygulama

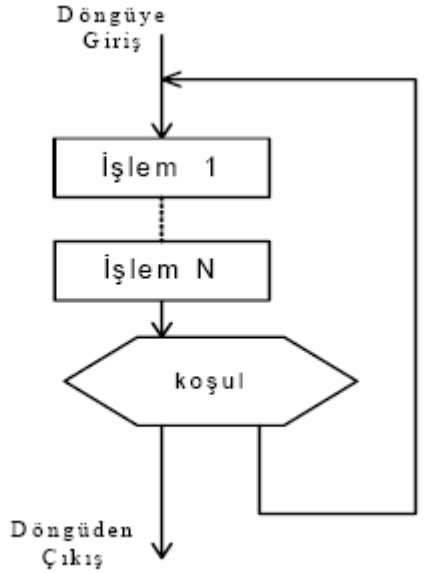
WhileOrnek.java

```
1  public class WhileOrnek {
2
3      public static void main(String args[]) {
4
5          int i = 0 ;     //döngü kontrol degiskeni
6          while (i < 10 )
7           {
8              System.out.println("i = " + i);
9              i++ ;
10         }
11         System.out.println("Sayma islemi tamamlandi.");
12     }
13
14 }
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
Sayma islemi tamamlandi.
Press any key to continue...
```

Döngüleme – `do while`

- ***`do-while`*** ifadesi, koşulu en yukarıda değil de en aşağıda hesaplar.
- Böylece ***`do-while`*** ifadesinde durum ***false*** olsa bile çalışması istenen kod bloğuna en az bir kere girilir.



Uygulama

WhileDoOrnek.java

```
1  public class WhileDoOrnek {
2
3      public static void main(String args[])
4      {
5          int i = 0 ;     //döngü kontrol degiskeni
6          do {
7              System.out.println("i = " + i);
8              i++ ;
9          } while ( i < 0 );
10
11         System.out.println("Sayma islemi tamamlandi.");
12     }
13 }
```

```
i = 0
Sayma islemi tamamlandi.
Press any key to continue...
```

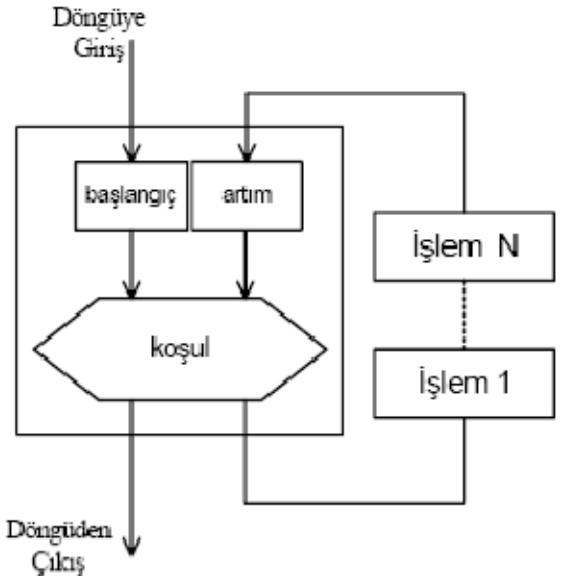
`while` Döngüsü Kullanırken Dikkat Edilmesi Gereken Hususlar

1. Döngü kontrol değişkenine uygun bir şekilde değer atandığına dikkat edilmeli.
2. Döngü durumunun ***true*** ile başlamısına dikkat edilmeli.
3. Döngü kontrol değişkeninin uygun bir şekilde güncellendiğinden emin olunması gerekir (sonsuz döngüye girmemesi için) .

Döngüleme – `for` ifadesi

- Döngünün ne zaman başlayacağı ve ne zaman biteceği en başta belirtilmiştir.

```
for (başlangıç; koşul; artış) {
   çalışması istenen kod bloğu
}
```



Uygulama

ForOrnek.java

```
1  public class ForOrnek {
2
3      public static void main(String args[]) {
4
5          for (int i=0 ; i < 10 ; i ++) {
6              System.out.println("i = " + i);
7          }
8
9      }
10
11 }
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
Press any key to continue...
```

for İle Sonsuz Döngü

```
for ( ; ; ) {     // sonsuz döngü
     ...
}
```

for - Çoklu Değişken

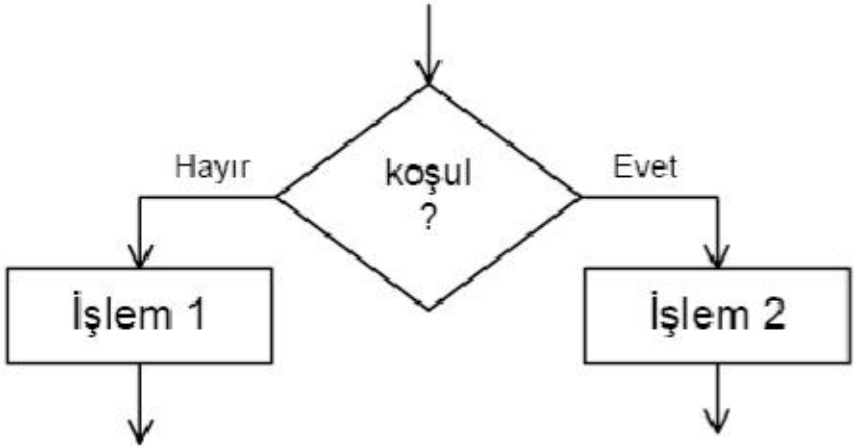
```
public class ForOrnekVersiyon2 {

  public static void main(String args[]) {

    for ( int i = 0 , j = 0 ; i < 20 ; i++ , j++ )
    {
       i *= j ;
       System.out.println("i = " + i + " j = " + j);
    }
  }
}
```

Karar Verme - `if`

```
if (koşul) {
  durum true olduğunda çalışması istenen kod bloğu
} else {
  durum false olduğunda çalışması istenen kod bloğu
}
```



Uygulama

IfElseTest.java

```
 1 public class IfElseTest {
 2     public static void main(String[] args) {
 3
 4         int puan = 76;
 5         char sonuc;
 6
 7         if (puan >= 90) {
 8             sonuc = 'A';
 9         } else if (puan >= 80) {
10             sonuc = 'B';
11         } else if (puan >= 70) {
12             sonuc = 'C';
13         } else if (puan >= 60) {
14             sonuc = 'D';
15         } else {
16             sonuc = 'F';
17         }
18         System.out.println("Sonuc = " + sonuc);
19     }
20 }
```

```
Sonuc = C
Press any key to continue...
```

Üçlü `if-else`

```
boolean-ifade ? deger0 : deger1
```

- Eğer *boolean* ifade *true* ise **değer0** hesaplanır , eğer *boolean* ifade false ise **deger1** hesaplanır.

Kısa Devre

- **if** ifadesinde eğer **VE(&&)** işlemi kullanılmış ise ve ilk değerden **false** dönmüş ise ikinci değer kesinlikle hesaplanmaz çünkü bu iki değerin sonucunun **VE(And)** işlemine göre **true** dönmesi imkansızdır.

- Kısa devre özelliği sayesinde uygulamalar gereksiz hesaplamalardan kurtulmuş olur.

KisaDevre.java *

Uygulama

```
1  public class KisaDevre {
2
3      public static boolean hesaplaBir(int a) {
4          System.out.println("hesaplaBir metoduna girildi");
5          return a > 1 ;
6      }
7
8      public static boolean hesaplaIki(int a) {
9          System.out.println("hesaplaIki metoduna girildi");
10         return a > 2 ;
11     }
12
13     public static void main(String[] args) {
14         System.out.println("Baslangic");
15         //hesaplaBir(0) --> false deger doner
16         //hesaplaIki(3) --> true deger doner
17
18         System.out.println("hesaplaBir(0) && hesaplaIki(3)");
19         if ( hesaplaBir(0) && hesaplaIki(3) ) {
20             System.out.println(" 1 -true ");
21         } else {
22             System.out.println(" 1 -false ");
23         }
24         System.out.println("--------");
25         System.out.println("hesaplaBir(0) || hesaplaIki(3)");
26         if (hesaplaBir(0) || hesaplaIki(3)) {
27             System.out.println(" 2 -true ");
28         } else {
29             System.out.println(" 2 -false ");
30         }
```

```
30              }
31              System.out.println("--------");
32              System.out.println("hesaplaBir(0) & hesaplaIki(3)");
33              if (hesaplaBir(0) & hesaplaIki(3)) {
34                  System.out.println(" 3 -true ");
35              } else {
36                  System.out.println(" 3 -false ");
37              }
38              System.out.println("--------");
39              System.out.println("hesaplaBir(0) | hesaplaIki(3)");
40              if (hesaplaBir(0) |  hesaplaIki(3)) {
41                  System.out.println(" 4 -true ");
42              } else {
43                  System.out.println(" 4 -false ");
44              }
45              System.out.println("--------");
46              System.out.println("hesaplaBir(0) ^ hesaplaIki(3)");
47              if (hesaplaBir(0) ^  hesaplaIki(3)) {
48                  System.out.println(" 5 -true ");
49              } else {
50                  System.out.println(" 5 -true ");
51              }
52              System.out.println("Son..");
53
54          }
55
56      }
```

```
Baslangic
hesaplaBir(0) && hesaplaIki(3)
hesaplaBir metoduna girildi
 1 -false
--------
hesaplaBir(0) || hesaplaIki(3)
hesaplaBir metoduna girildi
hesaplaIki metoduna girildi
 2 -true
--------
hesaplaBir(0) & hesaplaIki(3)
hesaplaBir metoduna girildi
hesaplaIki metoduna girildi
 3 -false
--------
hesaplaBir(0) | hesaplaIki(3)
hesaplaBir metoduna girildi
hesaplaIki metoduna girildi
 4 -true
--------
hesaplaBir(0) ^ hesaplaIki(3)
hesaplaBir metoduna girildi
hesaplaIki metoduna girildi
 5 -true
Son..
Press any key to continue...
```

Karar Verme - switch

```
switch(tamsayı) {
  case uygun-tamsayı-deger1 : çalışması istenen kod bloğu; break;
  case uygun-tamsayı-deger2 : çalışması istenen kod bloğu; break;
  case uygun-tamsayı-deger3 : çalışması istenen kod bloğu; break;
  case uygun-tamsayı-deger4 : çalışması istenen kod bloğu; break;
  case uygun-tamsayı-deger5 : çalışması istenen kod bloğu; break;
          // ...
  default: çalışması istenen kod bloğu ;
}
```

Uygulama 1

```
public class AylarSwitchTest {

    public static void main(String[] args) {

        int ay = 8;
        switch (ay) {
            case 1:  System.out.println("Ocak"); break;
            case 2:  System.out.println("Subat"); break;
            case 3:  System.out.println("Mart"); break;
            case 4:  System.out.println("Nisan"); break;
            case 5:  System.out.println("Mayis"); break;
            case 6:  System.out.println("Haziran"); break;
            case 7:  System.out.println("Temmuz"); break;
            case 8:  System.out.println("Agustos"); break;
            case 9:  System.out.println("Eylul"); break;
            case 10: System.out.println("Ekim"); break;
            case 11: System.out.println("Kasim"); break;
            case 12: System.out.println("Aralik"); break;
        }
    }
}
```

```
Agustos
Press any key to continue...
```

Uygulama 2

```
public class AylarSwitchTestNoBreak {

    public static void main(String[] args) {

        int ay = 8;
        switch (ay) {
            case 1:  System.out.println("Ocak");
            case 2:  System.out.println("Subat");
            case 3:  System.out.println("Mart");
            case 4:  System.out.println("Nisan");
            case 5:  System.out.println("Mayis");
            case 6:  System.out.println("Haziran");
            case 7:  System.out.println("Temmuz");
            case 8:  System.out.println("Agustos");
            case 9:  System.out.println("Eylul");
            case 10: System.out.println("Ekim");
            case 11: System.out.println("Kasim");
            case 12: System.out.println("Aralik");
        }
    }
}
```

```
Agustos
Eylul
Ekim
Kasim
Aralik
Press any key to continue...
```

Uygulama 3

```
public class AylarSwitchDefaultTest {

  public static void main(String[] args) {

   int ay = 25;
   switch (ay) {
     case 1:  System.out.println("Ocak"); break;
     case 2:  System.out.println("Subat"); break;
     case 3:  System.out.println("Mart"); break;
     case 4:  System.out.println("Nisan"); break;
     case 5:  System.out.println("Mayis"); break;
     case 6:  System.out.println("Haziran"); break;
     case 7:  System.out.println("Temmuz"); break;
     case 8:  System.out.println("Agustos"); break;
     case 9:  System.out.println("Eylul"); break;
     case 10: System.out.println("Ekim"); break;
     case 11: System.out.println("Kasim"); break;
     case 12: System.out.println("Aralik"); break;
     default: System.out.println("Heyoo,Aranilan Kosul" +
                                 "Bulunamadi!!");
    }
  }
}
```

```
Aranilan Kosul Bulunamadi !!
Press any key to continue...
```

Dallandırma İfadeleri

- Java programlama dilinde dallandırma ifadeleri toplam 3 adettir.
  - **`break`** ifadesi
  - **`continue`** ifadesi
  - **`return`** ifadesi

Uygulama

BreakTest.java

```
1  public class BreakTest {
2      public static void main(String[] args) {
3          for ( int i = 0; i < 100; i++ ) {
4              if ( i ==9 ) {// for dongusunu kiriyor
5                  break;
6              }
7              System.out.println("i =" +i);
8          }
9          System.out.println("Donguden cikti");
10     }
11 }
```

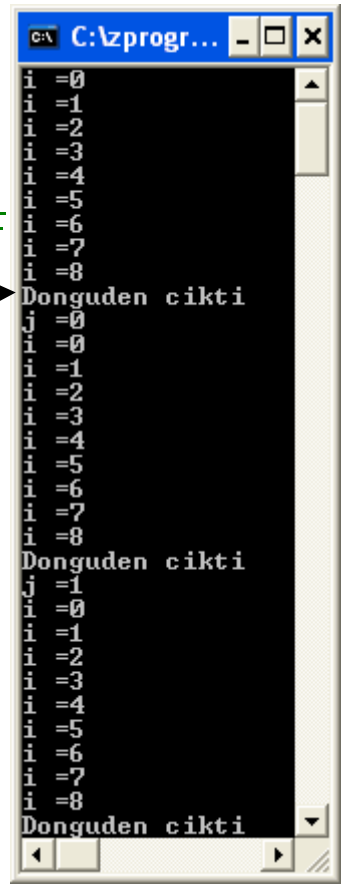
```
i =0
i =1
i =2
i =3
i =4
i =5
i =6
i =7
i =8
Donguden cikti
Press any key to continue...
```

break İfadesi - Etiketli

BreakTestEtiketli.java

```
1  public class BreakTestEtiketli {
2      public static void main(String[] args) {
3
4          kiril :
5          for ( int j = 0 ; j < 10 ; j ++ ) {
6
7              for ( int i = 0; i < 100; i++ ) {
8                  if ( i ==9 ) {// for dongusunu kir
9                      break kiril;
10                 }
11                 System.out.println("i =" +i);
12             }
13             System.out.println("Donguden cikti");
14             System.out.println("j =" +j);
15
16         }
17     }
18 }
```

```
i =0
i =1
i =2
i =3
i =4
i =5
i =6
i =7
i =8
Press any key to continue...
```



continue İfadesi - Etiketsiz

ContinueTest.java

```
1  public class ContinueTest {
2      public static void main(String[] args) {
3          for ( int i = 0; i < 30; i++ ) {
4              if ( i ==9 ) {// for dongusunu kiriyor
5                  continue;
6              }
7              System.out.println("i =" +i);
8          }
9          System.out.println("Donguden cikti");
10     }
11 }
```

```
i =0
i =1
i =2
i =3
i =4
i =5
i =6
i =7
i =8 → 9 yok
i =10
i =11
i =12
i =13
i =14
i =15
i =16
i =17
i =18
i =19
i =20
i =21
i =22
i =23
i =24
i =25
i =26
i =27
i =28
i =29
Donguden cikti
```

continue İfadesi - Etiketli

ContinueTestEtiketli.java

```
1  public class ContinueTestEtiketli {
2      public static void main(String[] args) {
3
4          pas :
5          for ( int j = 0 ; j < 6 ; j ++ ) {
6
7              for ( int i = 0; i < 5; i++ ) {
8                  if ( i ==3 ) {// for dongusunu kiriyor
9                      continue pas;
10                 }
11                 System.out.println("i =" +i);
12             }
13             System.out.println("Dongu den cikti");
14             System.out.println("j =" +j);
15
16         }
17     }
18 }
```

```
i =0
i =1
i =2
i =0
i =1
i =2
i =0
i =1
i =2
i =0
i =1
i =2
i =0
i =1
i =2
i =0
i =1
i =2
```

`return` İfadesi - Etiketli

- Sadece **`return`** anahtar kelimesi kullanarak yordamların içerisini tavizsiz bir şekilde terk edelebilir.

dizi.java *

```
1  public class dizi{
2      public static void main(String[] args){
3          int dizi1[]=new int[10];
4          for(int i=0;i<dizi1.length;++i){
5              dizi1[i]=i*i;
6              System.out.println((i+1)+". eleman="+dizi1[i]);
7          }
8      }
9  }
```

```
1. eleman=0
2. eleman=1
3. eleman=4
4. eleman=9
5. eleman=16
6. eleman=25
7. eleman=36
8. eleman=49
9. eleman=64
10. eleman=81
Press any key to continue..
```

dizi2.java *

```
1  public class dizi2{
2      public static void main(String[] args){
3          int[] dizi={1,113,5,7,9,11,13,15};
4          for(int i=0;i<dizi.length;++i){
5              System.out.println((i)+". eleman="+dizi[i]);
6          }
7      }
8  }
```

```
0. eleman=1
1. eleman=113
2. eleman=5
3. eleman=7
4. eleman=9
5. eleman=11
6. eleman=13
7. eleman=15
Press any key to continue.
```

dizi3.java *

```
1  /*Peki ya bir diziye tanımlandığı veri tipinden başka bir
2    tipte eleman eklemeye kalkarsak ne gibi sonuçlarla
3    karşılaşabiliriz. Doğrusu bunu merak ediyorum. Bu nedenle
4    yukarıdaki, örnekte, integer tipteki dizimize, başka bir
5    veri tipinden eleman ekleyelim. Bu amaçla, 'A' karakterini
6    "sincizce" dizi elemanları arasına yerleştirelim. */
7
8  public class dizi3 {
9      public static void main(String[] args){
10         int[] dizi={1,3,5,7,9,11,'A',15};
11         for(int i=0;i<dizi.length;++i){
12             System.out.println((i+1)+". eleman="+dizi[i]);
13         }
14     }
15 }
```

```
1. eleman=1
2. eleman=3
3. eleman=5
4. eleman=7
5. eleman=9
6. eleman=11
7. eleman=65
8. eleman=15
Press any key to continue.
```

dizi4.java

```
1  public class dizi4
2  {
3      public static void main(String[] args)
4      {
5          int[] dizi={1,2,3,4,5};
6          for(int i=0;i<dizi.length;++i)
7          {
8              System.out.println((i+1)+". eleman="+dizi[i]);
9          }
10         dizi[16]=7; //olmayan bir dizi elemanına değer atayalım bakalım ne
11     }               // hata verecek göreceğiniz gibi derleme hatası vermeyecek !!!
12 }
```

```
1. eleman=1
2. eleman=2
3. eleman=3
4. eleman=4
5. eleman=5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 16
        at dizi4.main(dizi4.java:10)
Press any key to continue...
```

dizi5.java

```
1  /*Dizileri tanımlarken, aynı veri türünden elemanları grupladığımızdan bahsetmiştik.
2  Farklı bir durum daha vardır. Bu da bir diziyi Java dilinin en üst sınıf olan
3  (C# taki gibi) Object türünden tanımlamak ve elemanları birer nesne olarak
4  belirlemektir. Bu teknikte, dizinin tüm elemanları birer nesnedir.
5  Bu anlamda bellekteki dizilişte farklı olmaktadır. Öyleki, Object türünden
6  bir dizi tanıladığımızda, dizinin elemanları öbekte tutulurken her bir eleman nesne
7  olduğu için, yine öbekteki asıl alanlarına referans eden değerleri taşırlar.
8  Ancak dizi elemanlarına eriştiğimizde, bu nesnelerin referans ettikleri öbek
9  alanlarındaki verilere ulaşılır. */
10
11 public class dizi5
12 {
13     public static void main(String[] args)
14     {
15         Integer eleman1=new Integer(456);
16         String eleman2=new String("Ben bir stringim...");
17         Double eleman3=new Double(45.45687);
18         Boolean eleman4=new Boolean(false);
19
20         Object[] dizi={eleman1,eleman2,eleman3,eleman4};
21         for(int i=0;i<dizi.length;++i)
22         {
23             System.out.println((i+1)+". eleman="+dizi[i]);
24         }
25     }
26 }
```

```
1. eleman=456
2. eleman=Ben bir stringim...
3. eleman=45.45687
4. eleman=false
Press any key to continue...
```

dizi_gir_sirala.java

```
1  import java.io.*;
2  class dizi_gir_sirala {
3  public static void main (String args[])throws IOException{
4               BufferedReader konsol=new BufferedReader(
5               new InputStreamReader(System.in));
6         System.out.println("Kac sayi gireceksiniz");
7         String giris=konsol.readLine();
8         Integer d=new Integer(giris);
9         int n=d; //int n=d.intValue();
10        int sayi[]=new int[n];
11          for (int i=0; i<sayi.length; i++)   {//for başı
12               System.out.print((i+1)+"inci Sayiyi giriniz:");
13               String giris1=konsol.readLine();
14               Integer a=new Integer(giris1);
15               sayi[i]=a;                          } //for sonu
16               for (int i=0; i<sayi.length-1; i++) {//for başı
17               for(int j=i+1;j<sayi.length;j++)
18               if (sayi[j]<sayi[i]){int b=sayi[j];sayi[j]=sayi[i];sayi[i]=b;}
19                                                   } //for sonu
20               for (int i=0; i<sayi.length; i++) System.out.println(sayi[i]);
21            }//main sonu
22        } //class sonu
```

- Mesajı göstermek için **`JOptionPane`**' i kullanılır:

```
JOptionPane.showMessageDialog(null, "Hello,World!");
```

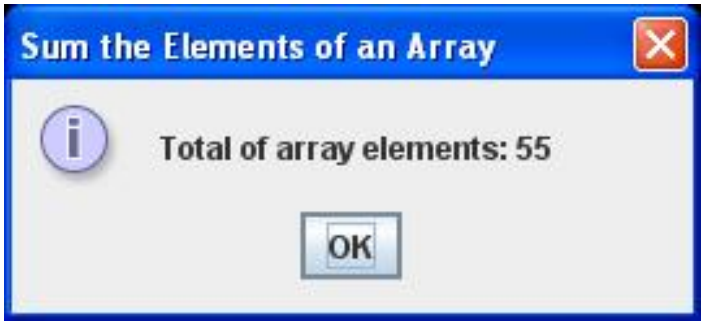
- İcon'un notu sol tarafta

- Keyfi bir resim dosyası belirlenebilir

```
JOptionPane.showMessageDialog( null,
  "Hello,World!",
  "Mesaj",
  JOptionPane.INFORMATION_MESSAGE,
  new ImageIcon("globe.gif"));
```

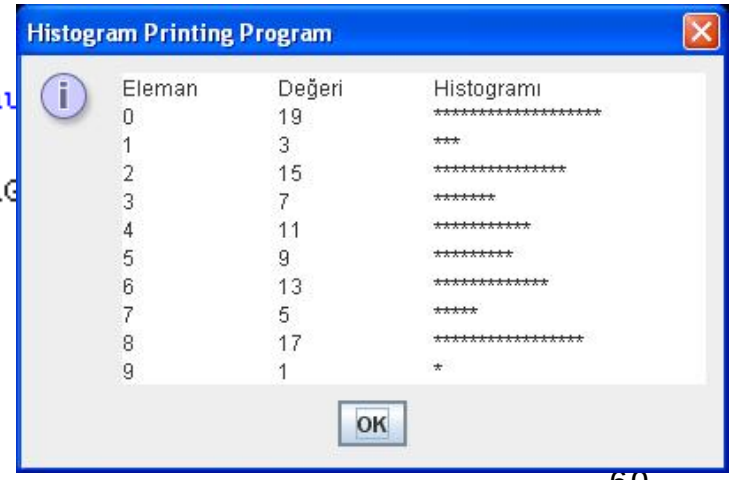
SumArray.java *

```
1  import javax.swing.*;
2
3  public class SumArray {
4     public static void main( String args[] )
5     {
6        int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
7        int total = 0;
8
9        for ( int i = 0; i < a.length; i++ )
10          total += a[ i ];
11
12       JOptionPane.showMessageDialog( null,
13          "Total of array elements: " + total,
14          "Sum the Elements of an Array",
15          JOptionPane.INFORMATION_MESSAGE );
16
17       System.exit( 0 );
18    }
19 }
```



Histogram.java

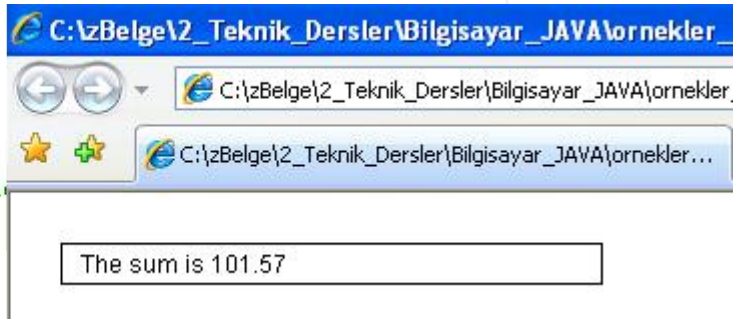
```
1  import javax.swing.*;
2  public class Histogram {
3     public static void main( String args[] )
4     {
5        int n[] = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
6        String output = "";
7
8        output += "Eleman\tDeğeri\tHistogramı";
9
10       for ( int i = 0; i < n.length; i++ ) {
11          output += "\n" + i + "\t" + n[ i ] + "\t";
12
13          for ( int j = 1; j <= n[ i ]; j++ ) // yıldızları yaz
14             output += "*";
15       }
16       JTextArea outputArea = new JTextArea( 11, 30 );
17       outputArea.setText( output );
18
19       JOptionPane.showMessageDialog( nu
20          "Histogram Printing Program",
21          JOptionPane.INFORMATION_MESSAG
22
23       System.exit( 0 );
24    }
25 }
```



klv_gir.java *

```
1  import java.awt.Graphics;    // import class Graphics
2  import javax.swing.*;        // import package javax.swing
3  public class klv_gir extends JApplet {
4     double sum;  // toplamda kullanılacak
5
6     public void init()
7     {
8        String firstNumber,secondNumber;  // gireceğimiz string değişkenleri
9        double number1,number2;       // stringten alınacak toplanacak değerler
10
11       // 1.sayıyı oku
12       firstNumber =JOptionPane.showInputDialog("floating-point 1. değeri girin" );
13       // 2.sayıyı oku
14       secondNumber =JOptionPane.showInputDialog("floating-point 2. değeri girin" );
15
16       //  String tipini  double tipine çevir
17       number1 = Double.parseDouble( firstNumber );
18       number2 = Double.parseDouble( secondNumber );
19       // sayıları opla
20       sum = number1 + number2;
21    }
22    public void paint( Graphics g )
23    {// sonuçları g.drawString kullanarak çiz
24       g.drawRect( 15, 10, 270, 20 );
25       g.drawString( "The sum is " + sum, 25, 25 );
26    }
27 }
```

```
<html>
<applet code="klv_gir.class" width=300 hei
</html>
```



RandomInt.java *

```
1  import javax.swing.JOptionPane;
2
3  public class RandomInt {
4     public static void main( String args[] )
5     {
6        int value;
7        String output = "";
8
9        for ( int i = 1; i <= 24; i++ ) {
10          value = 1 + (int) ( Math.random() * 49 );
11          output += value + "   ";
12
13          if ( i % 6 == 0 )
14             output += "\n";
15       }
16
17       JOptionPane.showMessageDialog( null, output,
18          "24 Rastgele sayi 1 den 49 kadar",
19          JOptionPane.INFORMATION_MESSAGE );
20
21       System.exit( 0 );
22    }
23 }
```

Sayi_Tahmini.java *

```
1  import java.io.*;
2  import java.lang.*;
3  import java.awt.*;
4  class Sayi_Tahmini{
5      public static void main(String args[])
6      throws IOException{
7      System.out.println("\tSAYI TAHMIN OYUNU\n");
8
9      BufferedReader konsol = new BufferedReader( new InputStreamReader(System.in));
10         //int sabit=56;
11         int sabit=1 + (int) ( Math.random() * 1000 );
12         int i=0;
13         while(2>1){
14
15         System.out.print("Sayiyi giriniz:");
16         String giris=konsol.readLine();
17         Integer a=new Integer(giris);
18         if (a==sabit){System.out.println("\nTebrikler Sayiyi bildiniz");
19         System.out.println("\nBeykenli olmak ayricaliktir...\n"); break; }
20         if (a!=sabit){
21             i=i+1;
22                 if (i==10){
23                 System.out.println("\nBilemediniz bugun Sanssiz Bir insansiniz");break;}
24
25             if (a<sabit) System.out.print("Sayiyi buyutun ve Tekrar deneyin");
26             if (a>sabit) System.out.print("Sayiyi kucultun ve Tekrar deneyin");
27 }}}}
```

```
C:\zprogram\java\JCREAT~1\GE2001.exe
        SAYI TAHMIN OYUNU

Sayiyi giriniz:34
Sayiyi buyutun ve Tekrar deneyinSayiyi giriniz:78
Sayiyi kucultun ve Tekrar deneyinSayiyi giriniz:12
Sayiyi buyutun ve Tekrar deneyinSayiyi giriniz:56

Tebrikler Sayiyi bildiniz

Beykenli olmak ayricaliktir...

Press any key to continue...
```