

Koşullar

Uygulama akışına müdahale eden yapılardan biri koşullardır. Java' da C/C++ söz dizimi yapısından dolayı gelen iki koşul vardır. Bunlar if-else if-else ve switch-case koşullarıdır.

if – else if - else koşul yapısı

if

if koşulu Java içerisinde durum kontrolü yapıp durumun true dönmesi halinde ilgili kod bloğunun çalıştırılmasını sağlar. Bununla ilgili örnek yapı aşağıdadır.

```
if(koşul){  
    //çalıştırılacak kod bloğu  
}
```

Bu yapıya göre ilgili if bloğunun içerisine girilmesi için koşulun true dönmesi gerekmektedir. Aşağıdaki örnekte `x == 4` sonucu true döneceği için ekrana "if bloğu çalıştı" yazacaktır.

```
public class Ornek {  
    public static void main(String[] args) {  
        int x = 4;  
        if (x == 4){  
            System.out.println("if bloğu çalıştı");  
        }  
    }  
}
```

if koşulu içerisindeki blok tanımları süslü parantezler ile belirtilir. Eğer bu belirtimi yapmazsak bir sonraki satırın sonuna kadar olanı if için geçerli kod bloğu olarak kabul eder. Yani yukarıdaki kodu aşağıdaki gibi de yazabilirdik. Ancak if satırından sonraki ; yani noktalı virgülü gördüğü alana kadarki satırı blok kabul ederdi.

```
public class Ornek {  
    public static void main(String[] args) {  
        int x = 4;  
        if (x == 4)  
            System.out.println("if bloğu çalıştı");  
    }  
}
```

Dikkat : if bloğu ise anlamına gelir örneğin `x` değeri 4' e eşitse gibi.

else

if ilgili koşulun olumlu olmasını yani true dönmesini bekler peki koşulun uymadığı durumlarda alternatif bir blok işletmek istersek ne yapmalıyız. Bu durumda else kullanılabilir. Aşağıdaki örnekte else bloğu işletilecektir.

```
public class Ornek {
    public static void main(String[] args) {
        int x = 20;
        if (x < 4) {
            System.out.println("if bloğu çalıştı");
        }
        else {
            System.out.println("else bloğu çalıştı");
        }
    }
}
```

Not : else cümlecisi if ile bloğu ile birlikte kullanılır. Tek başına bir anlam ifade etmez.

Dikkat : else bloğu değilse anlamına gelir. if koşulundaki karşılanmıyorsa yani x 4' eşitse – değilse gibi.

else if

Birbirine bağlı koşul yapısı kurmak istiyorsak if arkasından bunun devamını belirtmek için "else if" bloğu kullanılır. Buradaki önemli nokta if-else if-else bloğunun tek bir alan olarak görünmesidir.

Aşağıdaki kod içerisinde if ve else if ler tüm koşulları sağladığından x değeri kaç olursa olsun else bloğu hiç bir durumda çalışmayacaktır.

```
public class Ornek {
    public static void main(String[] args) {
        int x = 20;
        if (x < 0) {
            System.out.println("x negatiftir");
        }
        else if(x == 0){
            System.out.println("x sıfırdır");
        }
        else if(x > 0){
            System.out.println("x pozitif tam sayıdır");
        }
        else {
            System.out.println("çalışmayacak blok");
        }
    }
}
```

Not : else if blokları arasında koşullardan birini sağlanırsa diğer koşullar kontrol edilmez. Buna göre diğer else if içerisindeki koşullar uysa bile bu bloklara uğramaz.

switch case

Java içerisindeki diğer koşul operatörü switch-case dir. Çalışma tarzı if bloklarına benzese de farklı olarak boolean değer beklemek yerine anahtar değerın karşılığını bulmaya çalışır. Yazım tarzı aşağıdaki gibidir.

```
switch (anahtar) {  
    case değer-1:  
        //işletilecek kod bloğu  
        break;  
    case değer-2:  
        //işletilecek kod bloğu  
        break;  
    default:  
        //işletilecek kod bloğu  
        break;  
}
```

Burada yer alan switch deyimi içerisinde aldığı anahtar değer ile koşul yapısını başlatıyor. Ardından case ler uyuşan değerlere göre kontrol ediliyor. if ten farklı olarak case lerden biri sağlanırsa diğer case lerde kontrol ediliyor. Bunu engellemek istiyorsak her case sonuna break değimi ile koşulu sonlandırabiliriz. En sonda yer alan default aynen else değimi gibi herhangi bir case yakalanmazsa uğranacak bloğu ifade ediyor.

Dikkat: Eğer yakalanan case ler içerisinde break yoksa her durumda default çalıştırılır.

Aşağıda switch case ile ilgili örnek bir kullanım bulunmaktadır. Buna göre sadece 2 değerine sahip case çalıştırılacaktır.

```
public class Ornek {  
    public static void main(String[] args) {  
        int x = 2;  
        switch (x) {  
            case 1:  
                System.out.println("değer bir");  
                break;  
            case 2:  
                System.out.println("değer iki");  
                break;  
            default:  
                System.out.println("değer bilinmeyen bir sayı");  
                break;  
        }  
    }  
}
```

Not: switch case deyimi ile birlikte sadece ilkel tipler ve enum' lar kullanılabilir ancak Java SE 7 ile birlikte buna String değerleride dahil edilmiştir. Buna göre Java 7' den itibaren aşağıdaki deyimlerde kullanılabilir.

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

```
String gun = "Pazartesi";
switch (gun) {
case "Pazartesi":
    System.out.println("1. Gün");
    break;
case "Salı":
    System.out.println("2. Gün");
    break;
default:
    System.out.println("diğer günler");
    break;
}
}
```