

Koşul Operatörü

Koşul operatörü Java programlama dilinin üç operandlı (ternary) tek operatörüdür. Kullanımı şu şekildedir:

op1 ? op2 : op3

Koşul operatörü adeta if deyiminin operatörsel biçimidir. Operatörün operandları birer ifade olabilir. ? atomunun solundaki ifade boolean türden olmak zorundadır. Koşul operatörü şöyle çalışır: Önce operatörün solundaki ifadenin değeri hesaplanır. Bu değer true ise ? ile : arasındaki ifade yapılır ve operatörün ürettiği değer bu olur, eğer soru işaretinin solundaki ifade false ise bu kez : atomunun sağındaki ifade yapılır ve operatör bu değeri üretir. Örneğin:

```
package csd;

class App {

    public static void main(String[] args)
    {
        int a, b;

        a = 10;
        b = a > 0 ? 20 : -10;

        System.out.printf("b=%d\n", b);
    }
}
```

Koşul operatörünün ürettiği değer kullanılmalıdır. Aksi takdirde error oluşur. Örneğin:

```
package csd;

class App {

    public static void main(String[] args)
    {
        int a, b;

        a = 10;

        (a > 0) ? 20 : -10; //error: Ürettiği değer kullanılmıyor

    }
}
```

Koşul operatörü operatör öncelik tablosunda atama operatörünün hemen yukarısında ve sağdan sola önceliğe sahiptir:

Operatör	İlişkisi
()	Soldan Sağa
+ - ++ -- ! (tür)	Sağdan Sola
* / %	Soldan Sağa
+ -	Soldan Sağa
< > <= >=	Soldan Sağa
== !=	Soldan Sağa
&&	Soldan Sağa
	Soldan Sağa

?:	Sağdan Sola
=	Sağdan Sola

Aslında ? ve : nin solundaki ve sağındaki ifadeler koşul operatörünün operandlarını oluşturmaktadır. Örneğin:

```
result = a > 0 && b % 2 == 0 ? 100 : 200;
```

Bu ifadede aslında koşul operatörü ve atama operatörü olmak üzere toplam iki operatör vardır.

Koşul operatörüyle yapılan her şey aslında aslında if deyimiyle de yapılabilir. Fakat bazı durumlarda koşul operatörü koda okunabilirlik kazandırmaktadır.

Koşul operatörü tipik olarak üç durumda tercih edilir:

1. Karşılaştıma sonucunda elde edilen değerin bir değişkene atandığı durumda. Örneğin:

```
package csd;

class App {
    public static void main(String[] args)
    {
        java.util.Scanner kb = new java.util.Scanner(System.in);

        System.out.println("Birinci sayıyı giriniz");
        int a = Integer.parseInt(kb.nextLine());

        System.out.println("İkinci sayıyı giriniz");
        int b = Integer.parseInt(kb.nextLine());

        int max;

        max = a > b ? a : b;

        System.out.printf("max{%d, %d}=%d", a, b, max);

        kb.close();
    }
}
```

`max = a > b ? a : b;` deyiminin eşdeğer karşılığı şöyledir:

```
if (a > b)
    max = a;
else
    max = b;
```

2. return ifadelerinde. Örneğin:

```
class Util {
    public static int max(int a, int b)
    {
        return a > b ? a : b;
    }
}
```

`return a > b ? a : b;` deyiminin eşdeğer karşılığı şu şekildedir:

```
if (a > b)
    return a;
return b;
```

3. Bir metod çağrılırken argüman olarak. Örneğin:

```
package csd;

class App {

    public static void foo(int val)
    {
        System.out.printf("val=%d\n", val);
    }

    public static void main(String[] args)
    {
        java.util.Scanner kb = new java.util.Scanner(System.in);

        System.out.println("Birinci sayıyı giriniz");
        int a = Integer.parseInt(kb.nextLine());

        System.out.println("İkinci sayıyı giriniz");
        int b = Integer.parseInt(kb.nextLine());

        foo(a > b ? a : b);

        kb.close();
    }
}
```

foo(a > b ? a : b); deyiminin eşdeğer karşılığı şu şekildedir:

```
if (a > b)
    foo(a);
else
    foo(b);
```

Parantezler kullanılarak koşul operatörünün operandları diğer operatörlerden ayrıştırılabilir. Örneğin:

```
sum += (val % 2 == 0 ? 100 : 200) + val;
```

Koşul operatörü içiçe kullanılabilir. Örneğin:

```
package csd;

class App {
    public static void main(String[] args)
    {
        java.util.Scanner kb = new java.util.Scanner(System.in);

        System.out.println("Birinci Sayıyı giriniz");
        int a = Integer.parseInt(kb.nextLine());

        System.out.println("İkinci Sayıyı giriniz");
        int b = Integer.parseInt(kb.nextLine());

        System.out.println("Üçüncü Sayıyı giriniz");
        int c = Integer.parseInt(kb.nextLine());

        int max = a > b ? (a > c ? a : c) : (b > c ? b : c);
    }
}
```

```
        System.out.printf("max{%d, %d, %d}=%d\n", a, b, c, max);  
        kb.close();  
    }  
}  
  
int max = a > b ? (a > c ? a : c) : (b > c ? b : c);
```

deyiminde parantezler zorunlu olmamasına karşın okunabilirlik açısından konulması tavsiye edilir.