



CLARUSWAY

WAY TO REINVENT YOURSELF



clarusway



clarusway



clarusway



clarusway



clarusway

1

Logical Operators (Mantıksal Operatörler)

Lesson:
JAVA Chapter 07

Logical (Mantıksal) Operatörler

Mantıksal Operatörler

Mantıksal operatörler adından da anlaşılacağı gibi elde edilen verilerin mantıksal olarak test edilmesinde kullanılan işlemcilerdir. Mantıksal operatörler **doğru (true)** ve **yanlış (false)** gibi karşılaştırmalar yapmak için kullanılan operatörlerdir.

Operatör	İsim	Açıklama	Örnek
&&	Ve (AND)	Her iki ifade de doğruysa true döndürür	$x < 10 \ \&\& \ x < 25$
	Veya (OR)	İfadelerden biri doğruysa true döndürür	$x < 5 \ \ x < 4$
!	Ters İşlem	Sonucu tersine çevirir	$!(x < 5 \ \&\& \ x < 10)$



Logical (Mantıksal) Operatörler

- **&& Koşullu VE(AND) işlemi yapılmaktadır. Bütün sonuçlar kendi içinde true ise sonuç true, en az birisi bile false ise sonuç false olur.**

```
1 System.out.println(10==10 && 5<10); //Verdiğimiz örnekte iki karşılaştırmada true olduğu için çıktıda true olur
2 System.out.println(10==10 && 10<5); //Verdiğimiz örnekte 5 büyüktür 10 karşılaştırması yanlış olduğu için false olur
```

Ekran Çıktısı

```
1 true
2 false
```



& ile && Arasındaki Fark

& isareti kullanildiginda Java isaretin iki yanindaki mantiksal ifadelerin ikisini de kontrol eder. Bu islem kodumuzu yavaslatir

40<30 & 50==50 & 60>50

ilk karsilastirma yanlis olmasina ragmen Java tum karsilastirmalari kontrol etmeye devam eder.

&& isareti kullanildiginda ise Java en bastan kontrol etmeye baslar, mantiksal ifadelerin birinde yanlisi bulursa sonrakileri kontrol etme ihtiyaci duymaz. Bu islem kodumuzu hizlandirir

40<30 && 50==50 && 60>50

ilk karsilastirma yanlis oldugunu gorunce Java diger karsilastirmalari kontrol etmeden alt satira gecir.



Logical (Mantıksal) Operatörler

- **||** Koşullu VEYA işlemi yapar. Karşılaştırılan iki değerden en az biri true olduğu sürece, true döndürmektedir.

```
1 System.out.println(10==10 || 10<5); //10==10 true değerini döndürdüğü için sonuç true olarak çıkmaktadır.
```

Ekran Çıktısı

```
1 true
```



Logical (Mantıksal) Operatörler

- ! NOT (DEĞİL) işlemi yapar. Sonucu true ise false, false ise true yapmaktadır.

```
1 System.out.println(!(10==10));
```

Ekran Çıktısı

```
1 false
```

!= Esit değildir isareti

boolean sonuc1= 5+2 != 7;

System.out.println(5*2 != 15);

sonuc1 degeri **false** olur

true yazdirir



Logical (Mantıksal) Operatörler

```

genel sınıf OperatorÖrneği{
public static void main(String args[]){
    int a= 10 ;
    int b= 5 ;
    int c= 20 ;

    System.out.println(a>b||a<c); //doğru || doğru = doğru
    System.out.println(a>b|a<c); //doğru | doğru = doğru
    //|| vs |
    System.out.println(a>b||a++<c); //doğru || doğru = doğru
    System.out.println(a); //10 çünkü ikinci koşul kontrol edilmedi
    System.out.println(a>b|a++<c); //doğru | doğru = doğru
    System.out.println(a); //11 çünkü ikinci koşul kontrol edildi
}}

```

```

genel sınıf OperatorÖrneği{
public static void main(String args[]){
    int a= 10 ;
    int b= 5 ;
    int c= 20 ;

    System.out.println(a<b&&a++<c); //yanlış && doğru = yanlış
    System.out.println(a); //10 çünkü ikinci koşul kontrol edilmedi
    System.out.println(a<b&a++<c); //yanlış && doğru = yanlış
    System.out.println(a); //11 çünkü ikinci koşul kontrol edildi
}}

```

```

genel sınıf OperatorÖrneği{
public static void main(String args[]){
    int a= 10 ;
    int b= 5 ;
    int c= 20 ;

    System.out.println(a<b&&a<c); //yanlış && doğru = yanlış
    System.out.println(a<b&a<c); //yanlış ve doğru = yanlış
}}

```



Concatenation (String Variable Birleştirme)

Birden çok String variable'a " + " işlemi yapılırsa JAVA String variable'ları concatenation (birleştirme) yaparak yeni bir String oluşturur.

```
String a = "Selam";  
String b = "Javacan'lar";
```

```
System.out.println(a + b); // -> SelamJavacan'lar
```

```
System.out.println(a + " " + b); // -> Selam Javacan'lar
```



Concatenation (String Variable Birleştirme)

* **TRICK :) Bir aritmetik işlem içinde String kullanılırsa, aritmetik işlem önceliğine göre işlem yapıp String variable'a Concatenation** yapılır.

```
String a="Clarusway";  
int b=14;  
int c=53;  
  
System.out.println(a+b+c);// -> Clarusway1453  
  
System.out.println(c+b+a);// -> 67Clarusway  
  
System.out.println(a+(b+c));// -> Clarusway67  
  
System.out.println(a+b*c);// -> Clarusway742
```



THANKS!

Any questions?

Garry T. - Full Stack Automation Engineer

Haluk B. - JAVA Backend Developer

