

# Aritmetik Operatörler

Emre Altunbilek Java Dersleri

Matematiksel işlemleri yaparken kullandığımız + - \* / ve % ifadelerinin javadaki versiyonlarıdır.

Burada % operatörü mod alma işlemine yaramaktadır.

```
int sonuc = 5 % 2; //5 sayısının 2 sayısı ile modunu hesaplar. Sonuc = 1
```

Diğer operatörler matematikte bildiğiniz işlemler için kullanılır.

## ÖDEV:

sayi1 ve sayi2 diye iki integer değişken oluşturun ve bunlara sırasıyla 20 ve 40 değerlerini atayın. Bu değişkenleri yukarıda bahsettiğimiz operatörler ile işleme sokup sonuçlarını ekrana yazdırın.

## ÇÖZÜM :

?????

# Arttırma ve Azaltma Operatörleri

Emre Altunbilek Java Dersleri

`++` ve `--` operatörleri değişkende bulunan değeri 1 arttırmaya veya azaltmaya yarar.

Arttırma veya azaltma yapılacak değişkenin başına veya sonuna yazılabilir ve yazıldığı yere göre farklı davranışlar sergiler. Tabi şunu da söylemek gerekir ki tek başlarına kullanıldıklarında başta veya sonda olması fark yaratmaz.

Bu operatörler bilinen toplama ve çıkarma işlemine göre doğrudan işlemci tarafından desteklendiği için çok daha hızlı çalışırlar.

```
int sayi = 10;
sayi++; // ++sayi da yazsaydık bir önemi olmayacaktı.
```

```
int sonuc = ++sayi + 5 // sayı değişkenini 1 arttır ve sonra 5 ile topla
int sonuc = 5 + sayi++ // 5 sayısı ile sayı değişkenini topla, sonra sayıyı
bir arttır.
```

## ÖDEV:

Aşağıdaki ifadenin sonucunu hesaplayınız.

```
int sayi1 = 10;
int sayi2 = ++sayi1;
int sayi3 = sayi1--;
```

```
System.out.println("Sonuc : " + (sayi1)+(--sayi2)+(sayi3++));
```

## CEVAP :

????

# Aritmetik Atama Operatörleri

Emre Altunbilek Java Dersleri

Javada 5 çeşit atama operatörü bulunmaktadır, += -= \*= /= ve %=

```
int sayi1 = 10;
int sayi2 = 20;

int sonuc = 0;
sonuc += sayi1; // sonuc = sonuc + sayi1;
sonuc -= sayi2; // sonuc = sonuc - sayi2;
```

Diğer aritmetik atama operatörlerini de siz deneyip sonuçlarını yorumlayınız.

## ÖDEV :

Aşağıdaki ifadenin vereceği çıktıyı hesaplayınız.

```
double ondalikliSayi=6.50;
double sonuc = 0;
```

```
sonuc++;
ondalikliSayi *= sonuc;
```

```
System.out.println("Sonuç : " + sonuc);
```

## ÇÖZÜM :

???

# İlişkisel Karşılaştırma Operatörleri

Emre Altunbilek Java Dersleri

Javada 6 adet ilişkisel operatör bulunmaktadır, bunların en temel amacı değişkenleri karşılaştırmaktır.

Bu karşılaştırmaların sonucunda da true veya false sonuç verirler.

```
int s1=10, s2=20;
```

```
s1 == s2 // s1 ve s2 yi karşılaştırır, eşitse true eşit değilse false döndürür
s1 < s2 // s1 ve s2 yi karşılaştırır, eğer s1 s2den küçükse true döndürür.
s1 > s2 // s1 ve s2 yi karşılaştırır, eğer s1 s2den büyükse true döndürür.
s1 <= s2 // s1 ve s2 yi karşılaştırır, eğer s1 s2den küçük veya eşitse true döndürür.
s1 >= s2 // s1 ve s2 yi karşılaştırır, eğer s1 s2den büyük veya eşitse true döndürür.
s1 != s2 // s1 ve s2 yi karşılaştırır, eğer s1 s2ye eşit değilse true döndürür
```

Geriye döndürülen true veya false değerleri ile programımızı kontrol etmek, akışına yön vermek için if anahtar sözcüğü ile koşullar yazılabilir.

```
if(s1 == s2){
    System.out.println("s1 ve s2 eşittir");
    //detayları sonraki bölümlerde işlenecektir.
}
```

# Mantıksal Operatörler

Emre Altunbilek Java Dersleri

Matematikte kullandığımız ve, veya, değil ifadelerinin javadaki karşılıklarıdır.

Bu operatörler de sonuc olarak true veya false değerini döndürür.

`&&` ve `=>` true olması için iki değişkenin değerinin de true olması gerekir. Diğer durumlarda false değer döndürür.

`||` veya `=>` false olması için iki değişkenin değerinin de false olması gerekir. Diğer durumlarda true değer döndürür.

`!` değil `=>` önündeki ifadenin tersini verir.

```
boolean deger1 = true;  
boolean deger2 = false;
```

```
deger1 && deger2 => false;  
deger1 || deger2 => true;  
!deger1 => false;  
!deger2 => true;
```

# İşlem Önceliği

Emre Altunbilek Java Dersleri

Verdiğimiz operatörler çalıştırılırken, yanyana yazılmış değişkenler bu operatörlerin işlem önceliği kuralı baz alınarak hesaplamalar yapılır.

İşlem önceliği sırası büyükten küçüğe doğru şu şekildedir:

1. () önce parantez içi hesaplanır
2. ++ ve -- (eğer değişkenden önce ise)
3. çarpma ve bölme
4. toplama ve çıkarma
5. = atama işlemi
6. ++ ve -- (eğer değişkenden sonra ise)

## ÖDEV

Aşağıdaki sonuc degiskenine atanacak değerleri sırasıyla hesaplayınız.

```
int sayi1=15;  
int sayi2 = 5;  
int sonuc = 0;
```

```
sonuc = (sayi1 + sayi2*2 - sayi2) + sayi2 - sayi1 *4 + sayi1  
sonuc = (sayi1 * sayi2 + 4 / 2)+ sayi1++ * sayi2 + sayi1
```

## Ödev 1

Bir integer değişken oluşturun ve bu değişkende saniye değerini tutun.

Sonra bu saniye değerinin kaç dakika ve saniyeye denk geldiğini yazdırın  
Örnek 1200 değeri için 20 dakika 0 saniye yazmalıdır.

## Ödev 2

Bir double değişken oluşturun ve bu değişkende fahrenheit cinsinden sıcaklığı saklayın.

Sonrasında bu değerin kaç Celcius olduğunu hesaplayın.

100 değeri için 37.777778 gibi bir değer elde etmelisiniz.

Hesaplama formülü :  $\text{celcius} = 5/9 * (\text{fahrenheit} - 32)$

## Ödev 3

Bir yıl değerinin artık yıl olup olmadığını hesaplayın

Bir yılın artık olması için ya 400 ile modu 0 olmalı yada 4 ile modu 0 olmalı  
e 100 ile modu 0 olmamalı

4100 false değer vermeli.

Çözümler video derste mevcuttur.