Aritmetik Operatörleri

Hemen her programlama dilinde olduğu gibi Java dilinde de aritmetik işlemler yaparken aşağıdaki operatörleri kullanırız:

Operatör	***Açıklama***
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Modulo
++	1 artırma
--	1 eksiltme

Bu operatörlerin kullanılışları aşağıdaki örneklerle gösterilecektir.

Aritmetik.java

```
package Bölüm06;

public class Aritmetik {
    public static void main(String[] args) {
        int x = 10;
        int y = 3;

        System.out.println("x + y = " + (x + y) );
        System.out.println("x - y = " + (x - y));
```

```
        System.out.println("x * y = " + x * y);
        System.out.println("x / y = " + x / y);

        System.out.println("x % y = " + x % y);

        System.out.println("x++   = " + x++ );
        System.out.println("y--   = " + y-- );
    }
}
```

Çıktı:

```
x + y = 13
x - y = 7
x * y = 30
x / y = 3
x % y = 1
x++   = 10
y--   = 3
```

Uyarı

1. `Bu` çıktılarda, eşitliğin solundakiler string, sağındakiler ise sayısal işlem sonuçlarıdır. `string` tipi verilerle sayısal veriler bir arada olduğunda, artı (+) operatörü sayısal toplama işleminden fazlasını yapar. İki metni birleştirir, bir metinle bir sayıyı birleştirir. Buna java dilinde "*operator overloading*" denilir. Anlamı, (+) operatörünün, asıl işlevi dışında başka bir iş yapmasıdır. Buna "*aşkın operatör*" diyeceğiz. Daha genel olarak "*method aşımı – method overloading*" konusunu ileride ele alacağız.
2. 10 / 3 bölme işleminin sonucu kesirli olması gerekirken, çıktıda 10/3 = 3 tamsayı sonuç yazılmıştır. Çünkü, java, iki tamsayının bölümünü, bölümün tamsayı kısmı olarak verir. Tabii, bölümün kesir kısmını da istiyorsak, java'ya ne istediğimizi söylemek yetecektir. Eğer işleme giren sayılardan birisi `float` ya da `double` ise, Java bize, işlem sonucunu kesirli olarak verecektir..

KesirliİşlemJava

```
package Bölüm06;

public class KesirliİşlemJava {

    public static void main(String[] args) {
        int x = 10;
        float y = 3.0f;
        double z = 3.0;

        System.out.println("x + y = " + (x + y) );
        System.out.println("x - y = " + (x - y));
```

```
        System.out.println("x * y = " + x * y);
        System.out.println("x / y = " + x / y);

        System.out.println();

        System.out.println("x + z = " + (x + z) );
        System.out.println("x - z = " + (x - z));

        System.out.println("x * z = " + x * z);
        System.out.println("x / z = " + x / z);
    }
}
```

Çıktı

```
x + y = 13.0
x - y = 7.0
x * y = 30.0
x / y = 3.3333333

x + z = 13.0
x - z = 7.0
x * z = 30.0
x / z = 3.3333333333333335
```

Bölme işleminde `float` tipin 9 haneli, double tipin 18 haneli olarak çıktığına dikkat ediniz.

Şimdi veri tipleri arasında dönüşüm (casting) yapalım:

Casting01.java

```
package Bölüm06;

public class Casting01 {

    public static void main(String[] args) {
        int x = 10;
        int y =3;

        System.out.println("x / y           = " + (x / y) );
        System.out.println("(int)(x / y)    = " + (int)(x / y));
        System.out.println("(float)(x / y)  = " + (float)(x / y) );
        System.out.println("(double)(x / y) = " + (double)(x / y));
```

```
        System.out.println("(float)x / y)    = " + ((float)x / y));
        System.out.println("(double)x / y)   = " + ((double)x / y));
    }
}
```

Çıktı:

```
x / y             = 3
(int)(x / y)      = 3
(float)(x / y)    = 3.0
(double)(x / y)   = 3.0
(float)x / y)     = 3.3333333
(double)x / y)    = 3.3333333333333335
```

Bu programın çıktısının neden böyle olduğunu anlamak kolaydır. Programdaki x=10 tamsayısı y=3 tamsayısına tam bölünemez. Gerçek bölüm 3.333… dir. İlk satırda x/y = 10/3 tamsayı bölme işlemi, bölümün kesirli kısmını atıyor, yalnızca tamsayı kısmını alıyor; dolayısıyla sonuç 3 çıkıyor. İkinci satır, zaten `int` tipinden olan 3 sayısını `int` tipine dönüştürdüğü için, sonuç gene 3 olmaktadır. Üçüncü satırda, parantezler işlem önceliğine sahip olduğu için, önce (x/y)=3 tamsayı bölme işlemi yapılmakta, sonra 3 tamsayısı `float` tipine dönüştürülmektedir. Bu dönüşüm 3 tamsayısını 3.0 `float` tipine dönüştürmektedir. Dördüncü satır, `float` yerine `double` koyarak aynı işi yinelemektedir. Beşinci satırda, x/y işlemi parantez içinde olmadığından (float)x işlemi öncelik alır. Dolayısıyla, önce x tamsayısı 10.0 `float` tipine dönüştürülmekte, sonra 10.0/y bölme işlemi yapılmaktadır. Bunun sonucu `float` tipidir ve sonucu 9 haneli olarak yazılmaktadır. Son satırda `float` yerine `double` konularak aynı iş yinelenmektedir. Bunun sonucu double tipidir ve 18 haneli olarak yazılmaktadır. Program koşturulurken bellekte oluşan değerler aşağıdaki tabloda gösterilmektedir:

x	10
y	3
x/y	3
(int)(x/y)	3
(float)(x/y)	3.0
(double)(x/y)	3.0
(float)x/y =	(float)10/3 = ((float)10) / 3 = 3.3333333
(double)x/y	(double)10/3 = ((double)10) / 3 = 3.3333333333333335

Aşağıdaki program, tamsayı bölme işleminden çıkan bölümü kesirli yazdırmanın başka bir yöntemini

vermektedir. Satır satır inceleyiniz ve her satırın ne iş yaptığını algılayınız.

Bölme.java

```
package Bölüm06;

public class Bölme {

    public static void main(String[] args) {
        int x, y, isonuç ;
        float fsonuç ;
        double dsonuç;

        x = 7;
        y = 5;

        isonuç = x/y;
        System.out.println("x / y = " + isonuç );

        fsonuç = (float) x / y;
        System.out.println("x / y = " + fsonuç );
        fsonuç = x / (float) y;
        System.out.println("x / y = " + fsonuç );
        fsonuç = (float)x / (float) y;
        System.out.println("x / y = " + fsonuç );

        dsonuç = (double)x/y;
        System.out.println("x / y = " + dsonuç );
        dsonuç = x/(double)y;
        System.out.println("x / y = " + dsonuç );
        dsonuç = x/(double)y;
        System.out.println("x / y = " + dsonuç );
    }
}
```

Çıktı:

```
x / y = 1
x / y = 1.4
x / y = 1.4
x / y = 1.4
x / y = 1.4
x / y = 1.4
x / y = 1.4
```

Birinci çıktı `isonuç = x / y = 7/5 = 1` bir tamsayı bölme işlemi olduğundan bölümün tamsayı kısmı alınmıştır.

İkinci çıktı `fsonuç = (float)x / y = (float)7 / 5` deyiminde, önce 7 sayısı float tipine dönüştürülüyor, sonra 5 sayısına bölünüyor. Bir float tipin bir tamsayıya bölümü gene float tipindendir. Dolayısıyla, (float)7 / 5 = 7.0 / 5 = 1.4 dür.

Üçüncü çıktı Bu çıktı ikinci çıktının simetriğidir. `fsonuç = x/(float)y = 7/(float)5` deyiminde, önce 5 sayısı float tipine dönüştürülüyor, sonra 7 tamsayısı 5.0 sayısına bölünüyor. Bir tamsayı tipin bir float tipine bölümü gene float tipindendir. Dolayısıyla, 7 / (float)5 = 1.4 dür.

Dördüncü çıktı İkinci ve üçüncü çıktının birleşidir. fsonuç = (float)x / (float)y = (float)7 / (float)5 deyiminde, önce 7 ve 5 sayılarının her ikisi de float tipine dönüşür. Sonra iki float tipin birbirine bölümü yapılır. Bu işlemin sonucu, doğal olarak bir float tipidir. Dolayısıyla, (float)7 / (float)5 = 1.4 dür.

Sondaki üç çıktı, float yerine double konularak önceki çıktılar gibi elde edilmiştir.

Modulo Operatörü: %

`a % b` modulo işlemi `a` sayısının `b` sayısına bölümünden arta (kalan) sayıyı verir. Aşağıda modulo işlemi yapan programları satır satır inceleyiniz; her satırın ne yaptığını açıklayınız.

Modulo1.java

```
package Bölüm06;
public class Modulo1 {
    public static void main(String[] args) {
        int x = 13;
        int y = 7;
        int sonuç = x % y;
        System.out.println("x % y = " + sonuç);
    }
}
```

Çıktı

```
x % y = 6
```

`x % y modulo` işlemi, x sayısının y sayısına bölümünden kalanı verir. 13 sayısı 7 sayısına bölünürse kalan 6 dır.

Modulo2.java

```
package Bölüm06;
public class Modulo1 {
    public static void main(String[] args) {
        int x = 13;
        int y = 7;
```

```
            int sonuç = x % y;
            float fsonuç = x % y;
            double dsonuç = x%y;
            System.out.println("x % y = " + sonuç);
            System.out.println("x % y = " + fsonuç);
            System.out.println("x % y = " + dsonuç);
        }
    }
```

Çıktı

```
x % y = 6
x % y = 6.0
x % y = 6.0
```

`Modula` operatörü kesirli sayılar için de tanımlıdır. Bunu aşağıdaki örnekten görebiliriz.

Modulo3.java

```
package Bölüm06;
public class Modulo3 {
    public static void main(String[] args) {
        double x = 13.8;
        double y = 7.2;
        double sonuç = x % y;
        System.out.println("x % y = " + sonuç);
    }
}
```

Çıktı

```
x % y = 6.6000000000000005
```

++ ve -- Operatörleri

Sayısal değişkenlerin değerlerine 1 eklemek ya da 1 çıkarmak için `++` ve `--` operatörlerini kullanırız. x sayısal bir değişken ise

```
x++ = x+1       x-- = x-1
++x = x+1       --x = x-1
```

eşitlikleri vardır. Ancak bu iki operatör, bir sayıya 1 eklemek ya da çıkarmaktan daha fazlasını yapar. Takının, değişkenin önüne ya da sonuna gelmesi, işlem sonuçlarını bütünüyle etkiler. Bunu aşağıdaki örneklerden daha kolay anlayacağız.

Önel (Prefix) ve Sonal (Postfix) Takılar

```
int    x = 5;
float  y = 15.63 ;
```

bildirimleri yapılmış olsun.

```
x++ = 6   x-- = 4     ++x = 6     --x = 4
y++ = 16.63     y-- = 14.63 ++y = 16.63 -- y= 14.63
```

olur. Ancak, x ve y işleme giriyorsa, önel ve sonal operatörler farklı etkiler yaratır.

Önel operatör alan değişken değeri işleme girmeden önce değişir, işleme sonra girer.

Sonal operatör alan değişken değeri önce işleme girer, sonra değeri değişir.

Bunu şu işlemlerle daha kolay açıklayabiliriz.

```
x=5 iken  --x  *  --x  = 12       // [ 4*3  = 12]   ve    x = 3  olur
x=5 iken  ++x  *  ++x  = 42       // [6 * 7 = 42]   ve    x = 7  olur.
x=5 iken  x--  *  x--  = 20       // [5 * 4 = 20]   ve    x = 4  olur.
x=5 iken  x++  *  x++  = 30       // [5 * 6 = 30]   ve    x = 7  olur.
```

Şimdi bu operatörlerin işlevlerini program içinde görelim.

```
package Bölüm06;

public class Artım01 {
static int x = 5;
static int y = 3;
static int z = 8;
static int t = 11;
      public static void main(String[] args) {
            System.out.println("--x  *  --x    = " +  --x  *  --x) ;
            System.out.println("++y  *  ++y    = " +  ++y  *  ++y) ;

            System.out.println("z--  *  z--    = " +  z--  *  z--) ;
            System.out.println("t++  *  t++    = " +  t++  *  t++) ;
      }

}
```

Çıktı:

```
--x  *  --x   = 12
++y  *  ++y   = 20
z--  *  z--   = 56
t++  *  t++   = 132
```

Açıklama

```
\\ 4*3   = 12
\\ 4*5   = 20
\\ 8*7   = 56
\\ 11*12 = 132
```

Artım02.java

```
package Bölüm06;

public class Artım02 {

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

```
        int x = 5;
        System.out.println("x = " + x);
        System.out.println("++x değeri = " + ++x);
        System.out.println("x = " + x);
        System.out.println("x++ değeri = " + x++);
        System.out.println("x = " + x);
    }

}
```

Çıktı

```
x = 5
++x değeri = 6
x = 6
x++ değeri = 6
x = 7
```

Artım03.java

```
package Bölüm06;

public class Artım03 {
    public static void main(String[] args) {
    int n = 35;
   float x = 12.7f;

     System.out.println("n = " + n + " iken --n = " + --n + " ve n = " + n +
" olur");
        System.out.println("n = " + n + " iken   ++n = " + ++n  + " ve n =
" + n + " olur");
        System.out.println("n = " + n + " iken   --n = " + --n  + " ve n =
" + n + " olur");
        System.out.println("n = " + n + " iken   --n = " + --n  + " ve n =
" + n + " olur");

        System.out.println();

        System.out.println("x = " + x + " iken   --x = " + --x  + " ve x =
" + x + " olur");
        System.out.println("x = " + x + " iken   ++x = " + ++x  + " ve x =
" + x + " olur");
        System.out.println("x = " + x + " iken   --x = " + --x  + " ve x =
" + x + " olur");
        System.out.println("x = " + x + " iken   --x = " + --x  + " ve x =
" + x + " olur");
    }
}
```

Çıktı

```
n = 35 iken      --n = 34 ve  n = 34 olur
n = 34 iken      ++n = 35 ve  n = 35 olur
n = 35 iken      --n = 34 ve  n = 34 olur
n = 34 iken      --n = 33 ve  n = 33 olur

x = 12.7 iken    --x = 11.7 ve x = 11.7 olur
x = 11.7 iken    ++x = 12.7 ve x = 12.7 olur
x = 12.7 iken    --x = 11.7 ve x = 11.7 olur
x = 11.7 iken    --x = 10.7 ve x = 10.7 olur
```

Birli (unary) Operatörler

UnaryOperatörler.java

```
package Bölüm06;

public class UnaryOperatörler {
    public static void main(String[] args) {
          int sayı = 0;
        int önelArtım;
        int önelEksim;
        int sonalArtım;
        int sonalEksim;
        int pozitif;
        int negatif;
        byte bitWiseNot;
        boolean logicalNot;

        önelArtım = ++sayı;
        System.out.println("önel_Artım:  "+ önelArtım);

        önelEksim = --sayı;
        System.out.println("önel_Eksim:  "+ önelEksim);

        sonalEksim = sayı--;
        System.out.println("Sonal_Eksim:  "+ sonalEksim);

        sonalArtım = sayı++;
        System.out.println("Sonal_Artım:  "+ sonalArtım);

        System.out.println("sayı nın son değeri:  "+ sayı);
```

```
        pozitif = -sonalArtım;
        System.out.println("Pozitif:  "+ pozitif);

        negatif = +sonalArtım;
        System.out.println("Negatif:  "+ negatif);

        bitWiseNot = 0;
        bitWiseNot = (byte)(~bitWiseNot);
        System.out.println("Bitwise Not:  "+ bitWiseNot);

        logicalNot = false;
        logicalNot = !logicalNot;
        System.out.println("Logical Not:  "+ logicalNot);
    }
}
```

Çıktı

```
önel_Eksim            :  0
Sonal_Eksim           :  0
Sonal_Artım           :  -1
sayı nın son değeri   :  0
Pozitif               :  1
Negatif               :  -1
Bitwise Not           :  -1
Logical Not           :  true
```

AritmetikDemo.java

```
package Bölüm06;

public class Aritmetikdemo {
            public static void main(String[] args) {
               //İşleme giren sayılar
               int i = 28;
               int j = 37;
               double x = 23.674;
               double y = 8.46;
               System.out.println("Sayıları yaz...");
               System.out.println(" i = " + i);
               System.out.println(" j = " + j);
               System.out.println(" x = " + x);
```

```
        System.out.println(" y = " + y);

        //Toplama işlemi
        System.out.println("Toplama...");
        System.out.println(" i + j = " + (i + j));
        System.out.println(" x + y = " + (x + y));

        //Çıkarma işlemi
        System.out.println("Çıkarma...");
        System.out.println(" i - j = " + (i - j));
        System.out.println(" x - y = " + (x - y));

        //Çarpma işlemi
        System.out.println("Çarpma...");
        System.out.println(" i * j = " + (i * j));
        System.out.println(" x * y = " + (x * y));

        //Bölme İşlemi
        System.out.println("Bölme...");
        System.out.println(" i / j = " + (i / j));
        System.out.println(" x / y = " + (x / y));

        //Modulo işlemi
        //Bölme işleminde kalan
        System.out.println("Kalanı bul...");
        System.out.println(" i % j = " + (i % j));
        System.out.println(" x % y = " + (x % y));

        //Karma işlemler
        System.out.println("Karma işlemler...");
        System.out.println(" j + y = " + (j + y));
        System.out.println(" i * x = " + (i * x));
    }
}
```

Çıktı:

```
Sayıları yaz...
 i = 28
 j = 37
```

```
 x = 23.674
 y = 8.46
Toplama...
 i + j = 65
 x + y = 32.134
Çıkarma...
 i - j = -9
 x - y = 15.213999999999999
Çarpma...
 i * j = 1036
 x * y = 200.28204000000002
Bölme...
 i / j = 0
 x / y = 2.7983451536643025
Kalanı bul...
 i % j = 28
 x % y = 6.753999999999998
Karma işlemler...
 j + y = 45.46
 i * x = 662.872
```