BigInteger

BigInteger sınıfı, mevcut tüm ilkel veri türlerinin sınırının dışında kalan çok büyük tamsayı hesaplamalarını içeren matematiksel işlem için kullanılır.

Bu sayede BigInteger sınıfı, geniş yöntem kütüphanesi nedeniyle kullanımı çok kullanışlıdır ve rekabetçi programlamada da çok kullanılmaktadır. Şimdi aşağıda ilkel aritmetikteki basit ifadelerin bir listesi ve BigInteger nesneleri açısından benzer ifadesi verilmiştir.

Örnek:

```
int a, b;

BigInteger A, B;
```

Başlatma aşağıdaki gibidir**:**

```
a = 54;

b = 23;

A  = BigInteger.valueOf(54);

B  = BigInteger.valueOf(37);
```

Dize olarak kullanılabilen Tamsayılar için bunları aşağıdaki gibi başlatabilirsiniz:

```
A  = new BigInteger(“54”);

B  = new BigInteger(“123456789123456789”);
```

Bazı sabitler, başlatma kolaylığı için BigInteger sınıfında aşağıdaki gibi tanımlanmıştır:

```
A = BigInteger.ONE;

// Other than this, available constant are BigInteger.ZERO

// and BigInteger.TEN
```

Matematiksel işlemler aşağıdaki gibidir:

```
int c = a + b;

BigInteger C = A.add(B);
```

Diğer benzer işlevler subtract(), multiply(), divide(), remainder(), ancak tüm bu işlevler bağımsız değişken olarak BigInteger'ı alır, bu nedenle tamsayılarla veya dizelerle bu işlemi istiyorsak, aşağıda gösterildiği gibi işlevlere geçirmeden önce bunları BigInteger'a dönüştürün:

```
String str = “123456789”;

BigInteger C = A.add(new BigInteger(str));

int val  = 123456789;

BigInteger C = A.add(BigInteger.valueOf(val));
```

BigInteger'dan değer çıkarma aşağıdaki gibidir:

```
int x   =  A.intValue();   // value should be in limit of int x
long y   = A.longValue();  // value should be in limit of long y
String z = A.toString();
```

Karşılaştırma

```
if (a < b) {}         // For primitive int
if (A.compareTo(B) < 0)  {} // For BigInteger
```

Aslında compareTo, değerlere göre -1 (daha az), 0 (Eşit), 1 (daha büyük) döndürür. Eşitlik için şunları da kullanabiliriz:

```
if (A.equals(B)) {}  // A is equal to B
```

BigInteger Sınıfı Yöntemleri

Yöntem	Gerçekleştirilen Eylem
add(BigInteger val)	Değeri (bu + val) olan bir BigInteger döndürür.
abs()	Değeri bu BigInteger'ın mutlak değeri olan bir BigInteger döndürür.
ve(BigInteger val)	Değeri (this & val) olan bir BigInteger döndürür.
andNot(BigInteger val)	Değeri (this & ~val) olan bir BigInteger döndürür.
bitCount()	Bu BigInteger'ın ikilinin tamamlayıcı gösteriminde, işaret bitinden farklı olan bit sayısını döndürür.
bitLength()	Bir işaret biti hariç, bu BigInteger'ın en az iki tamamlayıcı gösterimindeki bit sayısını döndürür.
byteValueExact()	Bu BigInteger'ı bir bayta dönüştürerek kayıp bilgileri denetler.
clearBit(int n)	Değeri bu BigInteger'a eşdeğer olan ve belirlenen bit temizlenmiş bir BigInteger döndürür.

Yöntem	Gerçekleştirilen Eylem
compareTo(BigInteger val)	Bu BigInteger'ı belirtilen BigInteger ile karşılaştırır.
böl (BigInteger val)	Değeri (this / val) olan bir BigInteger döndürür.
divideAndRemainder(BigInteger val)	(Bu / val) ve ardından (bu % val) içeren iki BigIntegers dizisi döndürür.
doubleValue()	Bu BigInteger'ı çifte dönüştürür.
eşittir(Nesne x)	Bu BigInteger'ı eşitlik için belirtilen Object ile karşılaştırır.
flipBit(int n)	Değeri bu BigInteger'a eşdeğer olan ve belirtilen bit çevrilmiş bir BigInteger döndürür.
floatValue()	Bu BigInteger'ı float'a dönüştürür.
gcd(BigInteger val)	Değeri abs(this) ve abs(val)'in en büyük ortak böleni olan bir BigInteger döndürür.
getLowestSetBit()	Bu BigInteger'da en sağdaki (en düşük sıralı) bir bitin dizinini döndürür (en sağdaki bir bitin sağındaki sıfır bit sayısı).
hashCode()	Bu BigInteger için karma kodu döndürür.
intValue()	Bu BigInteger'ı int'ye dönüştürür.
intValueExact()	Bu BigInteger'ı int'ye dönüştürerek kayıp bilgileri denetler.
isProbablePrime(int certainty)	Bu BigInteger muhtemelen asal ise true, kesinlikle bileşikse false değerini döndürür.
longValue()	Bu BigInteger'ı uzuna dönüştürür.

Yöntem	Gerçekleştirilen Eylem
longValueExact()	Bu BigInteger'ı kayıp bilgileri kontrol eden uzun, uzun bir tamsayıya dönüştürür.
max(BigInteger val)	Bu BigInteger ve val'in maksimum değerini döndürür.
min(BigInteger val)	Bu BigInteger ve val'in minimumunu döndürür.
mod(BigInteger m	Değeri (bu mod m) olan bir BigInteger döndürür.
modInverse(BigInteger m)	Değeri (this-1 mod m) olan bir BigInteger döndürür.
modPow(BigInteger exponent, BigInteger m	Değeri (thisexponent mod m) olan bir BigInteger döndürür.
multiply(BigInteger val)	Değeri (bu * val) olan bir BigInteger döndürür.
negate()	Değeri (-this) olan bir BigInteger döndürür.
nextProbablePrime()	Muhtemelen asal olan bu BigInteger'dan daha büyük olan ilk tamsayıyı döndürür.
not()	Değeri (~this) olan bir BigInteger döndürür.
veya(BigInteger val)	Değeri (bu | val) olan bir BigInteger döndürür.
pow(int exponent)	Değeri (thisexponent) olan bir BigInteger döndürür.
probablePrime(int bitLength, Rastgele rnd)	Belirtilen bitLength ile muhtemelen asal olan pozitif bir BigInteger döndürür.
kalan(BigInteger val)	Değeri (bu % val) olan bir BigInteger döndürür.

Yöntem	Gerçekleştirilen Eylem
setBit(int n)	Değeri belirtilen bit kümesine sahip bu BigInteger'a eşdeğer olan bir BigInteger döndürür.
shiftLeft(int n)	Değeri (bu << n) olan bir BigInteger döndürür.
shiftRight(int n)	Değeri (bu >> n) olan bir BigInteger döndürür.
shortValueExact()	Bu BigInteger'ı kayıp bilgileri kontrol eden kısa bir süreye dönüştürür.
signum()	Bu BigInteger'ın signum işlevini döndürür.
sqrt()	Bu BigInteger'ın tamsayı karekökünü döndürür.
sqrtAndRemainder()	Bunun tamsayı kareköklerini ve geri kalanı olan bu – s*s'yi içeren iki BigIntegers dizisini döndürür.
çıkar(BigInteger val)	Değeri (this – val) olan bir BigInteger döndürür.
testBit(int n)	Yalnızca ve yalnızca belirtilen bit ayarlanmışsa true değerini döndürür.
toByteArray()	Bu BigInteger'ın iki-tamamlayıcı gösterimini içeren bir bayt dizisi döndürür.
toString()	Bu BigInteger'ın ondalık Dize gösterimini döndürür.
toString(int radix)	Bu BigInteger'ın dize gösterimini verilen radix'te döndürür.
valueOf(uzun val)	Değeri belirtilen uzununkine eşit olan bir BigInteger döndürür.
xor(BigInteger val)	Değeri (bu ^ val) olan bir BigInteger döndürür.

Örnek:

- Java

```
import java.math.BigInteger;

import java.util.Scanner;

 public class Example

{

    static BigInteger  factorial(int N)

    {

        // Initialize result

        BigInteger f = new BigInteger("1"); // Or BigInteger.ONE

         // Multiply f with 2, 3, ...N

        for (int i = 2; i <= N; i++)

            f = f.multiply(BigInteger.valueOf(i));

         return f;

    }

    public static void main(String  args[])

    {

        int N = 20;

        System.out.println(factorial(N));

    }

}
```

Çıktı:

```
2432902008176640000
```

StringBuffer ve StringBuilder Arasındaki Fark Nedir?

Java'da String **immutable** yani sabit/değiştirilemez bir class'tır. String sonuna ekleme yapmak, karakterleri değiştirmek gibi işlemlerde yeni bir string nesnesi yaratılır. Yeni nesne üretmek de maliyetli bir iştir.

StringBuffer

Java 1.0'dan beri mevcut olan StringBuffer class'ı mutable (değiştirilebilir) stringler yaratmak için kullanılır. Karakter dizisi içeren buffer'ı (ara bellek/tampon bellek) oluşturmak ve modifiye etmek için çeşitli methodları vardır. StringBuffer **thread-safe**'tir yani birden fazla thread aynı anda buffer'a erişemez!

```
String str = "hello";

str.concat("world");

System.out.println(str); // Output: hello

StringBuffer strBuffer = new StringBuffer("hello");

strBuffer.append("world");

System.out.println(strBuffer); // Output: helloworld
```

StringBuilder

StringBuilder class'ı StringBuffer'ın performans sorunlarını gidermek için Java 5'te eklenmiştir. İki class için de API aynıdır. StringBuilder'ın farkı StringBuffer'ın aksine senkronize olmamasıdır. **(non-synchronized).**

Metinsel ifadeleri birleştirmek için kullanılır. String sınıfında “+” ifadesiyle birleştirme yapıldığında her seferinde yeni bir string nesnesi tanımlanır.

Bu yönden stringten performans üstünlüğü vardır. Daha hızlıdır.

```
int one = 1;

String color = "red";

StringBuilder sb = new StringBuilder();

sb.append("One=").append(one).append(", Color=").append(color).append('\n');

System.out.print(sb); // Prints "One=1, Color=red"
```

Diğer örnek kodlar e kampüste diğer dosyalardadır.