

JAVA.MATH.BIGINTEGER CLASS

http://www.tutorialspoint.com/java/math/java_math_biginteger.htm

Copyright © tutorialspoint.com

Introduction

The **java.math.BigInteger** class provides operations analogues to all of Java's primitive integer operators and for all relevant methods from java.lang.Math.

It also provides operations for modular arithmetic, GCD calculation, primality testing, prime generation, bit manipulation, and a few other miscellaneous operations. All operations behave as if BigIntegers were represented in two's-complement notation.

Semantics of arithmetic operations and bitwise logical operations are similar to those of Java's integer arithmetic operators and Java's bitwise integer operators respectively. Semantics of shift operations extend those of Java's shift operators to allow for negative shift distances.

Comparison operations perform signed integer comparisons. Modular arithmetic operations are provided to compute residues, perform exponentiation, and compute multiplicative inverses. Bit operations operate on a single bit of the two's-complement representation of their operand.

All methods and constructors in this class throw NullPointerException when passed a null object reference for any input parameter.

Class declaration

Following is the declaration for **java.math.BigInteger** class:

```
public class BigInteger
    extends Number
        implements Comparable<BigInteger>
```

Field

Following are the fields for **java.math.BigInteger** class:

- **static BigInteger ONE** -- The BigInteger constant one.
- **static BigInteger TEN** -- The BigInteger constant ten.
- **static BigInteger ZERO** -- The BigInteger constant zero.

Class constructors

S.N.	Constructor & Description
1	BigInteger <i>byte[]val</i> This constructor is used to translate a byte array containing the two's-complement binary representation of a BigInteger into a BigInteger.
2	BigInteger <i>intsignum, byte[]magnitude</i> This constructor is used to translate the sign-magnitude representation of a BigInteger into a BigInteger.
3	BigInteger <i>intbitLength, intcertainty, Randomrnd</i> This constructor is used to construct a randomly generated positive BigInteger that is probably prime, with the specified bitLength.

4 **BigInteger***intnumBits, Randomrnd*

This constructor is used to construct a randomly generated BigInteger, uniformly distributed over the range 0 to ($2^{\text{numBits}} - 1$), inclusive.

5 **BigInteger***Stringval*

This constructor is used to translate the decimal String representation of a BigInteger into a BigInteger.

6 **BigInteger***Stringval, intradix*

This constructor is used to translate the String representation of a BigInteger in the specified radix into a BigInteger.

Class methods

S.N.	Method & Description
1	BigInteger abs This method returns a BigInteger whose value is the absolute value of this BigInteger.
2	BigInteger addBigInteger val This method returns a BigInteger whose value is <i>this</i> + <i>val</i> .
3	BigInteger andBigInteger val This method returns a BigInteger whose value is <i>this</i> & <i>val</i> .
4	BigInteger andNotBigInteger val This method returns a BigInteger whose value is <i>this</i> & ~<i>val</i> .
5	int bitCount This method returns the number of bits in the two's complement representation of this BigInteger that differ from its sign bit.
6	int bitLength This method returns the number of bits in the minimal two's-complement representation of this BigInteger, excluding a sign bit.
7	BigInteger clearBitint n This method returns a BigInteger whose value is equivalent to this BigInteger with the designated bit cleared.

8

[int compareToBigIntegerval](#)

This method compares this BigInteger with the specified BigInteger.

9

[BigInteger divideBigIntegerval](#)

This method returns a BigInteger whose value is *this/val*.

10

[BigInteger\[\] divideAndRemainderBigIntegerval](#)

This method returns an array of two BigIntegers containing *this/val* followed by *this*.

11

[double doubleValue](#)

This method converts this BigInteger to a double.

12

[boolean equalsObjectx](#)

This method compares this BigInteger with the specified Object for equality.

13

[BigInteger flipBitintn](#)

This method returns a BigInteger whose value is equivalent to this BigInteger with the designated bit flipped.

14

[float floatValue](#)

This method converts this BigInteger to a float.

15

[BigInteger gcdBigIntegerval](#)

This method returns a BigInteger whose value is the greatest common divisor of *absthis* and *absval*.

16

[int getLowestSetBit](#)

This method returns the index of the rightmost *lowest – order* one bit in this BigInteger *thenumberofzerobits to the right of the rightmost one bit*.

17

[int hashCode](#)

This method returns the hash code for this BigInteger.

18

[int intValue](#)

This method converts this BigInteger to an int.

19

[boolean isProbablePrimeintcertainty](#)

This method returns true if this BigInteger is probably prime, false if it's definitely composite.

20

[long longValue](#)

This method converts this BigInteger to a long.

21

[BigInteger maxBigIntegerVal](#)

This method returns the maximum of this BigInteger and val.

22

[BigInteger minBigIntegerVal](#)

This method returns the minimum of this BigInteger and val.

23

[BigInteger modBigInteger m](#)

This method returns a BigInteger whose value is *this* mod *m*.

24

[BigInteger modInverseBigInteger m](#)

This method returns a BigInteger whose value is (*this*⁻¹ mod *m*).

25

[BigInteger modPowBigInteger exponent, BigInteger m](#)

This method returns a BigInteger whose value is (*this*^{exponent} mod *m*).

26

[BigInteger multiplyBigInteger val](#)

This method returns a BigInteger whose value is *this* * *val*.

27

[BigInteger negate](#)

This method returns a BigInteger whose value is -*this*.

28

[BigInteger nextProbablePrime](#)

This method returns the first integer greater than this BigInteger that is probably prime.

29

[BigInteger not](#)

This method returns a BigInteger whose value is *this*.

30

[BigInteger orBigInteger val](#)

This method returns a BigInteger whose value is *this* | *val*.

31

[BigInteger pow\(int exponent\)](#)

This method returns a BigInteger whose value is $(this^{exponent})$.

32

[static BigInteger probablePrime\(int bitLength, Random rnd\)](#)

This method returns a positive BigInteger that is probably prime, with the specified bitLength.

33

[BigInteger remainder\(BigInteger val\)](#)

This method returns a BigInteger whose value is *this*.

34

[BigInteger setBit\(int n\)](#)

This method returns a BigInteger whose value is equivalent to this BigInteger with the designated bit set.

35

[BigInteger shiftLeft\(int n\)](#)

This method returns a BigInteger whose value is $this \ll n$.

36

[BigInteger shiftRight\(int n\)](#)

This method returns a BigInteger whose value is $this \gg n$.

37

[int signum](#)

This method returns the signum function of this BigInteger.

38

[BigInteger subtract\(BigInteger val\)](#)

This method returns a BigInteger whose value is $this - val$.

39

[boolean testBit\(int n\)](#)

This method returns true if and only if the designated bit is set.

40

[byte\[\] toByteArray](#)

This method returns a byte array containing the two's-complement representation of this BigInteger.

41

[String toString](#)

This method returns the decimal String representation of this BigInteger.

42

[String toStringinradix](#)

This method returns the String representation of this BigInteger in the given radix.

43

[static BigInteger valueOflongval](#)

This method returns a BigInteger whose value is equal to that of the specified long.

44

[BigInteger xorBigIntegerval](#)

This method returns a BigInteger whose value is *this*^{val}.

Processing math: 100%