

java.math

## Class BigInteger

See Also:

### Field Summary

static <a href="#">BigInteger</a>	<a href="#">ONE</a>  The BigInteger constant one.
static <a href="#">BigInteger</a>	<a href="#">TEN</a>  The BigInteger constant ten.
static <a href="#">BigInteger</a>	<a href="#">ZERO</a>  The BigInteger constant zero.

### Constructor Summary

[BigInteger](#)(byte[] val)

Translates a byte array containing the two's-complement binary representation of a BigInteger into a BigInteger.

[BigInteger](#)(int signum, byte[] magnitude)

Translates the sign-magnitude representation of a BigInteger into a BigInteger.

[BigInteger](#)(int bitLength, int certainty, [Random](#) rnd)

Constructs a randomly generated positive BigInteger that is probably prime, with the specified bitLength.

[BigInteger](#)(int numBits, [Random](#) rnd)

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

[BigInteger](#)([String](#) val)

Translates the decimal String representation of a BigInteger into a BigInteger.

[BigInteger](#)([String](#) val, int radix)

Translates the String representation of a BigInteger in the specified radix into a BigInteger.

## Method Summary

<a href="#">BigInteger</a>	<code>abs()</code>  Returns a BigInteger whose value is the absolute value of this BigInteger.
<a href="#">BigInteger</a>	<code>add(BigInteger val)</code>  Returns a BigInteger whose value is (this + val).
<a href="#">BigInteger</a>	<code>and(BigInteger val)</code>  Returns a BigInteger whose value is (this & val).
<a href="#">BigInteger</a>	<code>andNot(BigInteger val)</code>  Returns a BigInteger whose value is (this & ~val).
int	<code>bitCount()</code>  Returns the number of bits in the two's complement representation of this BigInteger that differ from its sign bit.
int	<code>bitLength()</code>  Returns the number of bits in the minimal two's-

	complement representation of this <code>BigInteger</code> , <i>excluding</i> a sign bit.
<a href="#"><code>BigInteger</code></a>	<code><a href="#">clearBit</a>(int n)</code>  Returns a <code>BigInteger</code> whose value is equivalent to this <code>BigInteger</code> with the designated bit cleared.
int	<code><a href="#">compareTo</a>(<a href="#">BigInteger</a> val)</code>  Compares this <code>BigInteger</code> with the specified <code>BigInteger</code> .
<a href="#"><code>BigInteger</code></a>	<code><a href="#">divide</a>(<a href="#">BigInteger</a> val)</code>  Returns a <code>BigInteger</code> whose value is (this / val).
<a href="#"><code>BigInteger</code></a> []	<code><a href="#">divideAndRemainder</a>(<a href="#">BigInteger</a> val)</code>  Returns an array of two <code>BigIntegers</code> containing (this / val) followed by (this % val).
double	<code><a href="#">doubleValue</a>()</code>  Converts this <code>BigInteger</code> to a double.

boolean	<a href="#"><code>equals(Object x)</code></a>  Compares this BigInteger with the specified Object for equality.
<a href="#"><code>BigInteger</code></a>	<a href="#"><code>flipBit(int n)</code></a>  Returns a BigInteger whose value is equivalent to this BigInteger with the designated bit flipped.
float	<a href="#"><code>floatValue()</code></a>  Converts this BigInteger to a float.
<a href="#"><code>BigInteger</code></a>	<a href="#"><code>gcd(BigInteger val)</code></a>  Returns a BigInteger whose value is the greatest common divisor of <code>abs(this)</code> and <code>abs(val)</code> .
int	<a href="#"><code>getLowestSetBit()</code></a>  Returns the index of the rightmost (lowest-order) one bit in this BigInteger (the number of zero

	bits to the right of the rightmost one bit).
int	<a href="#">hashCode()</a> Returns the hash code for this BigInteger.
int	<a href="#">intValue()</a> Converts this BigInteger to an int.
boolean	<a href="#">isProbablePrime</a> (int certainty) Returns true if this BigInteger is probably prime, false if it's definitely composite.
long	<a href="#">longValue()</a> Converts this BigInteger to a long.
<a href="#">BigInteger</a>	<a href="#">max</a> ( <a href="#">BigInteger</a> val) Returns the maximum of this BigInteger and val.

<a href="#"><u>BigInteger</u></a>	<code>min(BigInteger val)</code>  Returns the minimum of this BigInteger and val.
<a href="#"><u>BigInteger</u></a>	<code>mod(BigInteger m)</code>  Returns a BigInteger whose value is (this mod m).
<a href="#"><u>BigInteger</u></a>	<code>modInverse(BigInteger m)</code>  Returns a BigInteger whose value is (this <sup>-1</sup> mod m).
<a href="#"><u>BigInteger</u></a>	<code>modPow(BigInteger exponent, BigInteger m)</code>  Returns a BigInteger whose value is (this <sup>exponent</sup> mod m).
<a href="#"><u>BigInteger</u></a>	<code>multiply(BigInteger val)</code>  Returns a BigInteger whose value is (this * val).
<a href="#"><u>BigInteger</u></a>	<code>negate()</code>  Returns a BigInteger whose value is (-this).
<a href="#"><u>BigInteger</u></a>	<code>nextProbablePrime()</code>  Returns the first integer

	greater than this BigInteger that is probably prime.
<a href="#">BigInteger</a>	<code>not()</code>  Returns a BigInteger whose value is (~this).
<a href="#">BigInteger</a>	<code>or(BigInteger val)</code>  Returns a BigInteger whose value is (this   val).
<a href="#">BigInteger</a>	<code>pow(int exponent)</code>  Returns a BigInteger whose value is (this <sup>exponent</sup> ).
static <a href="#">BigInteger</a>	<code>probablePrime(int bitLength, Random rnd)</code>  Returns a positive BigInteger that is probably prime, with the specified bitLength.
<a href="#">BigInteger</a>	<code>remainder(BigInteger val)</code>  Returns a BigInteger whose value is (this % val).
<a href="#">BigInteger</a>	<code>setBit(int n)</code>  Returns a BigInteger



	whose value is equivalent to this BigInteger with the designated bit set.
<a href="#">BigInteger</a>	<a href="#">shiftLeft</a> (int n)  Returns a BigInteger whose value is (this << n).
<a href="#">BigInteger</a>	<a href="#">shiftRight</a> (int n)  Returns a BigInteger whose value is (this >> n).
int	<a href="#">signum</a> ()  Returns the signum function of this BigInteger.
<a href="#">BigInteger</a>	<a href="#">subtract</a> ( <a href="#">BigInteger</a> val)  Returns a BigInteger whose value is (this - val).
boolean	<a href="#">testBit</a> (int n)  Returns true if and only if the designated bit is set.
byte[]	<a href="#">toByteArray</a> ()  Returns a byte array

	containing the two's-complement representation of this BigInteger.
<a href="#">String</a>	<a href="#">toString()</a>  Returns the decimal String representation of this BigInteger.
<a href="#">String</a>	<a href="#">toString(int radix)</a>  Returns the String representation of this BigInteger in the given radix.
static <a href="#">BigInteger</a>	<a href="#">valueOf(long val)</a>  Returns a BigInteger whose value is equal to that of the specified long.
<a href="#">BigInteger</a>	<a href="#">xor(BigInteger val)</a>  Returns a BigInteger whose value is (this ^ val).

## Class BigDecimal

Method Summary	
<a href="#">BigDecimal</a>	<b><a href="#">abs()</a></b>  Returns a BigDecimal whose value is the absolute value of this BigDecimal, and whose scale is this.scale().
<a href="#">BigDecimal</a>	<b><a href="#">abs(MathContext mc)</a></b>  Returns a BigDecimal whose value is the absolute value of this BigDecimal, with rounding according to the context settings.
<a href="#">BigDecimal</a>	<b><a href="#">add(BigDecimal augend)</a></b>  Returns a BigDecimal whose value is (this + augend), and whose scale is max(this.scale(), augend.scale()).
<a href="#">BigDecimal</a>	<b><a href="#">add(BigDecimal augend, MathContext mc)</a></b>  Returns a BigDecimal whose

	value is (this + augend), with rounding according to the context settings.
byte	<a href="#">byteValueExact()</a> Converts this BigDecimal to a byte, checking for lost information.
int	<a href="#">compareTo(BigDecimal val)</a> Compares this BigDecimal with the specified BigDecimal.
<a href="#">BigDecimal</a>	<a href="#">divide(BigDecimal divisor)</a> Returns a BigDecimal whose value is (this / divisor), and whose preferred scale is (this.scale() - divisor.scale()); if the exact quotient cannot be represented (because it has a non-terminating decimal expansion) an ArithmeticException is thrown.
<a href="#">BigDecimal</a>	<a href="#">divide(BigDecimal divisor, int roundingMode)</a> Returns a BigDecimal whose value is (this / divisor), and whose scale is this.scale().

<a href="#">BigDecimal</a>	<div> <div> <div>divide(<a href="#">BigDecimal</a> divisor, int scale, int roundingMode)</div> <div>Returns a BigDecimal whose value is (this / divisor), and whose scale is as specified.</div> </div> </div>
<a href="#">BigDecimal</a>	<div> <div> <div>divide(<a href="#">BigDecimal</a> divisor, int scale, <a href="#">RoundingMode</a> roundingMode)</div> <div>Returns a BigDecimal whose value is (this / divisor), and whose scale is as specified.</div> </div> </div>
<a href="#">BigDecimal</a>	<div> <div> <div>divide(<a href="#">BigDecimal</a> divisor, <a href="#">MathContext</a> mc)</div> <div>Returns a BigDecimal whose value is (this / divisor), with rounding according to the context settings.</div> </div> </div>
<a href="#">BigDecimal</a>	<div> <div> <div>divide(<a href="#">BigDecimal</a> divisor, <a href="#">RoundingMode</a> roundingMode)</div> <div>Returns a BigDecimal whose value is (this / divisor), and whose scale is this.scale().</div> </div> </div>
<a href="#">BigDecimal</a> [ ]	<div> <div> <div>divideAndRemainder(<a href="#">BigDecimal</a> divisor)</div> <div>Returns a two-element BigDecimal array containing the</div> </div> </div>

	<p>result of <code>divideToIntegralValue</code> followed by the result of remainder on the two operands.</p>
<a href="#">BigDecimal</a> [ ]	<p><a href="#">divideAndRemainder</a>(<a href="#">BigDecimal</a> divisor, <a href="#">MathContext</a> mc)</p> <p>Returns a two-element <code>BigDecimal</code> array containing the result of <code>divideToIntegralValue</code> followed by the result of remainder on the two operands calculated with rounding according to the context settings.</p>
<a href="#">BigDecimal</a>	<p><a href="#">divideToIntegralValue</a>(<a href="#">BigDecimal</a> divisor)</p> <p>Returns a <code>BigDecimal</code> whose value is the integer part of the quotient (this / divisor) rounded down.</p>
<a href="#">BigDecimal</a>	<p><a href="#">divideToIntegralValue</a>(<a href="#">BigDecimal</a> divisor, <a href="#">MathContext</a> mc)</p> <p>Returns a <code>BigDecimal</code> whose value is the integer part of (this / divisor).</p>

double	<a href="#">doubleValue()</a> Converts this BigDecimal to a double.
boolean	<a href="#">equals(Object x)</a> Compares this BigDecimal with the specified Object for equality.
float	<a href="#">floatValue()</a> Converts this BigDecimal to a float.
int	<a href="#">hashCode()</a> Returns the hash code for this BigDecimal.
int	<a href="#">intValue()</a> Converts this BigDecimal to an int.
int	<a href="#">intValueExact()</a> Converts this BigDecimal to an int, checking for lost information.

<code>long</code>	<a href="#"><code>longValue()</code></a>  Converts this <code>BigDecimal</code> to a <code>long</code> .
<code>long</code>	<a href="#"><code>longValueExact()</code></a>  Converts this <code>BigDecimal</code> to a <code>long</code> , checking for lost information.
<a href="#"><code>BigDecimal</code></a>	<a href="#"><code>max(BigDecimal val)</code></a>  Returns the maximum of this <code>BigDecimal</code> and <code>val</code> .
<a href="#"><code>BigDecimal</code></a>	<a href="#"><code>min(BigDecimal val)</code></a>  Returns the minimum of this <code>BigDecimal</code> and <code>val</code> .
<a href="#"><code>BigDecimal</code></a>	<a href="#"><code>movePointLeft(int n)</code></a>  Returns a <code>BigDecimal</code> which is equivalent to this one with the decimal point moved <code>n</code> places to the left.
<a href="#"><code>BigDecimal</code></a>	<a href="#"><code>movePointRight(int n)</code></a>  Returns a <code>BigDecimal</code> which is equivalent to this one with the



	decimal point moved n places to the right.
<a href="#">BigDecimal</a>	<code>multiply(BigDecimal multiplicand)</code>  Returns a BigDecimal whose value is (this × multiplicand), and whose scale is (this.scale() + multiplicand.scale()).
<a href="#">BigDecimal</a>	<code>multiply(BigDecimal multiplicand, <a href="#">MathContext</a> mc)</code>  Returns a BigDecimal whose value is (this × multiplicand), with rounding according to the context settings.
<a href="#">BigDecimal</a>	<code>negate()</code>  Returns a BigDecimal whose value is (-this), and whose scale is this.scale().
<a href="#">BigDecimal</a>	<code>negate(<a href="#">MathContext</a> mc)</code>  Returns a BigDecimal whose value is (-this), with rounding according to the context settings.

<a href="#">BigDecimal</a>	<p><a href="#">plus</a>()</p> <p>Returns a BigDecimal whose value is (+this), and whose scale is this.scale().</p>
<a href="#">BigDecimal</a>	<p><a href="#">plus</a>(<a href="#">MathContext</a> mc)</p> <p>Returns a BigDecimal whose value is (+this), with rounding according to the context settings.</p>
<a href="#">BigDecimal</a>	<p><a href="#">pow</a>(int n)</p> <p>Returns a BigDecimal whose value is (this<sup>n</sup>), The power is computed exactly, to unlimited precision.</p>
<a href="#">BigDecimal</a>	<p><a href="#">pow</a>(int n, <a href="#">MathContext</a> mc)</p> <p>Returns a BigDecimal whose value is (this<sup>n</sup>).</p>
int	<p><a href="#">precision</a>()</p> <p>Returns the <i>precision</i> of this BigDecimal.</p>
<a href="#">BigDecimal</a>	<p><a href="#">remainder</a>(<a href="#">BigDecimal</a> divisor)</p> <p>Returns a BigDecimal whose value is (this % divisor).</p>

<a href="#">BigDecimal</a>	<code>remainder(BigDecimal divisor, MathContext mc)</code>  Returns a BigDecimal whose value is (this % divisor), with rounding according to the context settings.
<a href="#">BigDecimal</a>	<code>round(MathContext mc)</code>  Returns a BigDecimal rounded according to the MathContext settings.
int	<code>scale()</code>  Returns the <i>scale</i> of this BigDecimal.
<a href="#">BigDecimal</a>	<code>scaleByPowerOfTen(int n)</code>  Returns a BigDecimal whose numerical value is equal to (this * $10^n$ ).
<a href="#">BigDecimal</a>	<code>setScale(int newScale)</code>  Returns a BigDecimal whose scale is the specified value, and whose value is numerically equal to this BigDecimal's.

<a href="#">BigDecimal</a>	<a href="#">setScale</a> (int newScale, int roundingMode)  Returns a BigDecimal whose scale is the specified value, and whose unscaled value is determined by multiplying or dividing this BigDecimal's unscaled value by the appropriate power of ten to maintain its overall value.
<a href="#">BigDecimal</a>	<a href="#">setScale</a> (int newScale, <a href="#">RoundingMode</a> roundingMode)  Returns a BigDecimal whose scale is the specified value, and whose unscaled value is determined by multiplying or dividing this BigDecimal's unscaled value by the appropriate power of ten to maintain its overall value.
short	<a href="#">shortValueExact</a> ()  Converts this BigDecimal to a short, checking for lost information.

int	<a href="#">signum()</a>  Returns the signum function of this BigDecimal.
<a href="#">BigDecimal</a>	<a href="#">stripTrailingZeros()</a>  Returns a BigDecimal which is numerically equal to this one but with any trailing zeros removed from the representation.
<a href="#">BigDecimal</a>	<a href="#">subtract(BigDecimal subtrahend)</a>  Returns a BigDecimal whose value is (this - subtrahend), and whose scale is max(this.scale(), subtrahend.scale()).
<a href="#">BigDecimal</a>	<a href="#">subtract(BigDecimal subtrahend, MathContext mc)</a>  Returns a BigDecimal whose value is (this - subtrahend), with rounding according to the context settings.

<a href="#"><u>BigInteger</u></a>	<a href="#"><u>toBigInteger()</u></a>  Converts this BigDecimal to a BigInteger.
<a href="#"><u>BigInteger</u></a>	<a href="#"><u>toBigIntegerExact()</u></a>  Converts this BigDecimal to a BigInteger, checking for lost information.
<a href="#"><u>String</u></a>	<a href="#"><u>toEngineeringString()</u></a>  Returns a string representation of this BigDecimal, using engineering notation if an exponent is needed.
<a href="#"><u>String</u></a>	<a href="#"><u>toPlainString()</u></a>  Returns a string representation of this BigDecimal without an exponent field.
<a href="#"><u>String</u></a>	<a href="#"><u>toString()</u></a>  Returns the string representation of this BigDecimal,

	using scientific notation if an exponent is needed.
<a href="#">BigDecimal</a>	<a href="#">ulp()</a>  Returns the size of an ulp, a unit in the last place, of this BigDecimal.
<a href="#">BigInteger</a>	<a href="#">unscaledValue()</a>  Returns a BigInteger whose value is the <i>unscaled value</i> of this BigDecimal.
static <a href="#">BigDecimal</a>	<a href="#">valueOf(double val)</a>  Translates a double into a BigDecimal, using the double's canonical string representation provided by the <a href="#">Double.toString(double)</a> method.
static <a href="#">BigDecimal</a>	<a href="#">valueOf(long val)</a>  Translates a long value into a BigDecimal with a scale of zero.
static <a href="#">BigDecimal</a>	<a href="#">valueOf(long unscaledVal, int scale)</a>  Translates a long unscaled

	value and an int scale into a BigDecimal.
--	--