|  |  |
| --- | --- |
| **字段摘要** | |
| static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**ONE**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ONE)            值为 1，标度为 0。 |
| static int | [**ROUND\_CEILING**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_CEILING)            接近正无穷大的舍入模式。 |
| static int | [**ROUND\_DOWN**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_DOWN)            接近零的舍入模式。 |
| static int | [**ROUND\_FLOOR**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_FLOOR)            接近负无穷大的舍入模式。 |
| static int | [**ROUND\_HALF\_DOWN**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_DOWN)            向“最接近的”数字舍入，如果与两个相邻数字的距离相等，则为上舍入的舍入模式。 |
| static int | [**ROUND\_HALF\_EVEN**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_EVEN)            向“最接近的”数字舍入，如果与两个相邻数字的距离相等，则向相邻的偶数舍入。 |
| static int | [**ROUND\_HALF\_UP**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_UP)            向“最接近的”数字舍入，如果与两个相邻数字的距离相等，则为向上舍入的舍入模式。 |
| static int | [**ROUND\_UNNECESSARY**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_UNNECESSARY)            断言请求的操作具有精确的结果，因此不需要舍入。 |
| static int | [**ROUND\_UP**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_UP)            舍入远离零的舍入模式。 |
| static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**TEN**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#TEN)            值为 10，标度为 0。 |
| static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**ZERO**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ZERO)            值为 0，标度为 0。 |

|  |  |
| --- | --- |
| **构造方法摘要** | |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val)            将 BigInteger 转换为 BigDecimal。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.math.BigInteger,%20int))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) unscaledVal, int scale)            将 BigInteger 非标度值和 int 标度转换为 BigDecimal。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.math.BigInteger,%20int,%20java.math.MathContext))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) unscaledVal, int scale, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 BigInteger 非标度值和 int 标度转换为 BigDecimal（根据上下文设置进行舍入）。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.math.BigInteger,%20java.math.MathContext))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 BigInteger 转换为 BigDecimal（根据上下文设置进行舍入）。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(char[]))(char[] in)            将 BigDecimal 的字符数组表示形式转换为 BigDecimal，接受与 [BigDecimal(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 构造方法相同的字符序列。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(char[],%20int,%20int))(char[] in, int offset, int len)            将 BigDecimal 的字符数组表示形式转换为 BigDecimal，接受与 [BigDecimal(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 构造方法相同的字符序列，同时允许指定子数组。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(char[],%20int,%20int,%20java.math.MathContext))(char[] in, int offset, int len, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 BigDecimal 的字符数组表示形式转换为 BigDecimal，接受与 [BigDecimal(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 构造方法相同的字符序列，同时允许指定子数组，并根据上下文设置进行舍入。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(char[],%20java.math.MathContext))(char[] in, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 BigDecimal 的字符数组表示形式转换为 BigDecimal，接受与 [BigDecimal(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 构造方法相同的字符序列（根据上下文设置进行舍入）。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(double))(double val)            将 double 转换为 BigDecimal，后者是 double 的二进制浮点值准确的十进制表示形式。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(double,%20java.math.MathContext))(double val, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 double 转换为 BigDecimal（根据上下文设置进行舍入）。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(int))(int val)            将 int 转换为 BigDecimal。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(int,%20java.math.MathContext))(int val, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 int 转换为 BigDecimal（根据上下文设置进行舍入）。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(long))(long val)            将 long 转换为 BigDecimal。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(long,%20java.math.MathContext))(long val, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 long 转换为 BigDecimal（根据上下文设置进行舍入）。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String))([String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) val)            将 BigDecimal 的字符串表示形式转换为 BigDecimal。 |  |
| [**BigDecimal**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String,%20java.math.MathContext))([String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) val, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            将 BigDecimal 的字符串表示形式转换为 BigDecimal，接受与 [BigDecimal(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 构造方法相同的字符串（按照上下文设置进行舍入）。 |  |

|  |  |
| --- | --- |
| **方法摘要** | |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**abs**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#abs())()            返回 BigDecimal，其值为此 BigDecimal 的绝对值，其标度为 this.scale()。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**abs**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#abs(java.math.MathContext))([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为此 BigDecimal 绝对值的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**add**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#add(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) augend)            返回一个 BigDecimal，其值为 (this + augend)，其标度为 max(this.scale(), augend.scale())。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**add**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#add(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) augend, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (this + augend) 的 BigDecimal（根据上下文设置进行舍入）。 |
| byte | [**byteValueExact**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#byteValueExact())()            将此 BigDecimal 转换为 byte，以检查丢失的信息。 |
| int | [**compareTo**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) val)            将此 BigDecimal 与指定的 BigDecimal 比较。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)            返回一个 BigDecimal，其值为 (this / divisor)，其首选标度为 (this.scale() - divisor.scale())；如果无法表示准确的商值（因为它有无穷的十进制扩展），则抛出 ArithmeticException。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20int))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, int roundingMode)            返回一个 BigDecimal，其值为 (this / divisor)，其标度为 this.scale()。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20int,%20int))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, int scale, int roundingMode)            返回一个 BigDecimal，其值为 (this / divisor)，其标度为指定标度。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20int,%20java.math.RoundingMode))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, int scale, [RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html) roundingMode)            返回一个 BigDecimal，其值为 (this / divisor)，其标度为指定标度。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (this / divisor) 的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20java.math.RoundingMode))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, [RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html) roundingMode)            返回一个 BigDecimal，其值为 (this / divisor)，其标度为 this.scale()。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html)[] | [**divideAndRemainder**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideAndRemainder(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)            返回由两个元素组成的 BigDecimal 数组，该数组包含 divideToIntegralValue 的结果，后跟对两个操作数计算所得到的 remainder。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html)[] | [**divideAndRemainder**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideAndRemainder(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回由两个元素组成的 BigDecimal 数组，该数组包含 divideToIntegralValue 的结果，后跟根据上下文设置对两个操作数进行舍入计算所得到的 remainder的结果。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divideToIntegralValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideToIntegralValue(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)            返回 BigDecimal，其值为向下舍入所得商值 (this / divisor) 的整数部分。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**divideToIntegralValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideToIntegralValue(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回 BigDecimal，其值为 (this / divisor) 的整数部分。 |
| double | [**doubleValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#doubleValue())()            将此 BigDecimal 转换为 double。 |
| boolean | [**equals**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#equals(java.lang.Object))([Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) x)            比较此 BigDecimal 与指定的 Object 的相等性。 |
| float | [**floatValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#floatValue())()            将此 BigDecimal 转换为 float。 |
| int | [**hashCode**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#hashCode())()            返回此 BigDecimal 的哈希码。 |
| int | [**intValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#intValue())()            将此 BigDecimal 转换为 int。 |
| int | [**intValueExact**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#intValueExact())()            将此 BigDecimal 转换为 int，以检查丢失的信息。 |
| long | [**longValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#longValue())()            将此 BigDecimal 转换为 long。 |
| long | [**longValueExact**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#longValueExact())()            将此 BigDecimal 转换为 long，以检查丢失的信息。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**max**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#max(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) val)            返回此 BigDecimal 和 val 的最大值。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**min**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#min(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) val)            返回此 BigDecimal 和 val 的最小值。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**movePointLeft**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#movePointLeft(int))(int n)            返回一个 BigDecimal，它等效于将该值的小数点向左移动 n 位。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**movePointRight**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#movePointRight(int))(int n)            返回一个 BigDecimal，它等效于将该值的小数点向右移动 n 位。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**multiply**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#multiply(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) multiplicand)            返回一个 BigDecimal，其值为 (this × multiplicand)，其标度为 (this.scale() + multiplicand.scale())。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**multiply**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#multiply(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) multiplicand, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (this × multiplicand) 的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**negate**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#negate())()            返回 BigDecimal，其值为 (-this)，其标度为 this.scale()。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**negate**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#negate(java.math.MathContext))([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (-this) 的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**plus**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#plus())()            返回 BigDecimal，其值为 (+this)，其标度为 this.scale()。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**plus**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#plus(java.math.MathContext))([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (+this) 的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**pow**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#pow(int))(int n)            返回其值为 (thisn) 的 BigDecimal，准确计算该幂，使其具有无限精度。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**pow**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#pow(int,%20java.math.MathContext))(int n, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (thisn) 的 BigDecimal。 |
| int | [**precision**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#precision())()            返回此 BigDecimal 的*精度*。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**remainder**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#remainder(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)            返回其值为 (this % divisor) 的 BigDecimal。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**remainder**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#remainder(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (this % divisor) 的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**round**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#round(java.math.MathContext))([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回根据 MathContext 设置进行舍入后的 BigDecimal。 |
| int | [**scale**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#scale())()            返回此 BigDecimal 的*标度*。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**scaleByPowerOfTen**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#scaleByPowerOfTen(int))(int n)            返回其数值等于 (this \* 10n) 的 BigDecimal。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**setScale**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int))(int newScale)            返回一个 BigDecimal，其标度为指定值，其值在数值上等于此 BigDecimal 的值。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**setScale**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int,%20int))(int newScale, int roundingMode)            返回一个 BigDecimal，其标度为指定值，其非标度值通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定，以维护其总值。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**setScale**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int,%20java.math.RoundingMode))(int newScale, [RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html) roundingMode)            返回 BigDecimal，其标度为指定值，其非标度值通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定，以维护其总值。 |
| short | [**shortValueExact**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#shortValueExact())()            将此 BigDecimal 转换为 short，以检查丢失的信息。 |
| int | [**signum**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#signum())()            返回此 BigDecimal 的正负号函数。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**stripTrailingZeros**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#stripTrailingZeros())()            返回数值上等于此小数，但从该表示形式移除所有尾部零的 BigDecimal。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**subtract**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#subtract(java.math.BigDecimal))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) subtrahend)            返回一个 BigDecimal，其值为 (this - subtrahend)，其标度为 max(this.scale(), subtrahend.scale())。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**subtract**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#subtract(java.math.BigDecimal,%20java.math.MathContext))([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) subtrahend, [MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)            返回其值为 (this - subtrahend) 的 BigDecimal（根据上下文设置进行舍入）。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) | [**toBigInteger**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toBigInteger())()            将此 BigDecimal 转换为 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) | [**toBigIntegerExact**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toBigIntegerExact())()            将此 BigDecimal 转换为 BigInteger，以检查丢失的信息。 |
| [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) | [**toEngineeringString**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toEngineeringString())()            返回此 BigDecimal 的字符串表示形式，需要指数时，则使用工程计数法。 |
| [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) | [**toPlainString**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toPlainString())()            返回不带指数字段的此 BigDecimal 的字符串表示形式。 |
| [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) | [**toString**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toString())()            返回此 BigDecimal 的字符串表示形式，如果需要指数，则使用科学记数法。 |
| [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**ulp**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ulp())()            返回此 BigDecimal 的 ulp（最后一位的单位）的大小。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) | [**unscaledValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#unscaledValue())()            返回其值为此 BigDecimal 的*非标度值* 的 BigInteger。 |
| static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**valueOf**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#valueOf(double))(double val)            使用 [Double.toString(double)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html#toString(double)) 方法提供的 double 规范的字符串表示形式将 double 转换为 BigDecimal。 |
| static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**valueOf**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#valueOf(long))(long val)            将 long 值转换为具有零标度的 BigDecimal。 |
| static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) | [**valueOf**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#valueOf(long,%20int))(long unscaledVal, int scale)            将 long 非标度值和 int 标度转换为 BigDecimal。 |

**ROUND\_UP**

public static final int **ROUND\_UP**

舍入远离零的舍入模式。在丢弃非零部分之前始终增加数字。注意，此舍入模式始终不会减少计算值的大小。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_UP)

**ROUND\_DOWN**

public static final int **ROUND\_DOWN**

接近零的舍入模式。在丢弃某部分之前始终不增加数字（即截短）。注意，此舍入模式始终不会增加计算值的大小。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_DOWN)

**ROUND\_CEILING**

public static final int **ROUND\_CEILING**

接近正无穷大的舍入模式。如果 BigDecimal 为正，则舍入行为与 ROUND\_UP 相同；如果为负，则舍入行为与 ROUND\_DOWN 相同。注意，此舍入模式始终不会减少计算值。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_CEILING)

**ROUND\_FLOOR**

public static final int **ROUND\_FLOOR**

接近负无穷大的舍入模式。如果 BigDecimal 为正，则舍入行为与 ROUND\_DOWN 相同；如果为负，则舍入行为与 ROUND\_UP 相同。注意，此舍入模式始终不会增加计算值。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_FLOOR)

**ROUND\_HALF\_UP**

public static final int **ROUND\_HALF\_UP**

向“最接近的”数字舍入，如果与两个相邻数字的距离相等，则为向上舍入的舍入模式。如果舍弃部分 >= 0.5，则舍入行为与 ROUND\_UP 相同；否则舍入行为与 ROUND\_DOWN相同。注意，这是我们大多数人在小学时就学过的舍入模式。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_HALF_UP)

**ROUND\_HALF\_DOWN**

public static final int **ROUND\_HALF\_DOWN**

向“最接近的”数字舍入，如果与两个相邻数字的距离相等，则为上舍入的舍入模式。如果舍弃部分 > 0.5，则舍入行为与 ROUND\_UP 相同；否则舍入行为与 ROUND\_DOWN 相同。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_HALF_DOWN)

**ROUND\_HALF\_EVEN**

public static final int **ROUND\_HALF\_EVEN**

向“最接近的”数字舍入，如果与两个相邻数字的距离相等，则向相邻的偶数舍入。如果舍弃部分左边的数字为奇数，则舍入行为与 ROUND\_HALF\_UP 相同；如果为偶数，则舍入行为与 ROUND\_HALF\_DOWN 相同。注意，在重复进行一系列计算时，此舍入模式可以将累加错误减到最小。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_HALF_EVEN)

**ROUND\_UNNECESSARY**

public static final int **ROUND\_UNNECESSARY**

断言请求的操作具有精确的结果，因此不需要舍入。如果对获得精确结果的操作指定此舍入模式，则抛出 ArithmeticException。

**另请参见：**

[常量字段值](http://tool.oschina.net/uploads/apidocs/jdk-zh/constant-values.html#java.math.BigDecimal.ROUND_UNNECESSARY)

|  |
| --- |
| **方法详细信息** |

**valueOf**

public static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **valueOf**(long unscaledVal,

int scale)

将 long 非标度值和 int 标度转换为 BigDecimal。提供的此“静态工厂方法”优先于 ( long, int) 构造方法，因为前者允许重用经常使用的 BigDecimal 值。

**参数：**

unscaledVal - BigDecimal 的非标度值。

scale - BigDecimal 的标度。

**返回：**

其值为 (unscaledVal × 10-scale) 的 BigDecimal。

**valueOf**

public static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **valueOf**(long val)

将 long 值转换为具有零标度的 BigDecimal。提供的此“静态工厂方法”优先于 ( long) 构造方法，因为前者允许重用经常使用的 BigDecimal 值。

**参数：**

val - BigDecimal 的值。

**返回：**

其值为 val 的 BigDecimal。

**valueOf**

public static [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **valueOf**(double val)

使用 [Double.toString(double)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html#toString(double)) 方法提供的 double 规范的字符串表示形式将 double 转换为 BigDecimal。

**注：**这通常是将 double（或 float）转化为 BigDecimal 的首选方法，因为返回的值等于从构造 BigDecimal（使用 [Double.toString(double)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html#toString(double)) 得到的结果）得到的值。

**参数：**

val - 要转换为 BigDecimal 的 double。

**返回：**

其值等于或约等于 val 值的 BigDecimal。

**抛出：**

[NumberFormatException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/NumberFormatException.html) - 如果 val 为无穷大或 NaN。

**从以下版本开始：**

1.5

**add**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **add**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) augend)

返回一个 BigDecimal，其值为 (this + augend)，其标度为 max(this.scale(), augend.scale())。

**参数：**

augend - 将添加到此 BigDecimal 中的值。

**返回：**

this + augend

**add**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **add**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) augend,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (this + augend) 的 BigDecimal（根据上下文设置进行舍入）。 如果任一数字为零，并且精度设置为非零，则其他数字（必要时进行舍入）可以作为结果。

**参数：**

augend - 将添加到此 BigDecimal 中的值。

mc - 要使用的上下文。

**返回：**

this + augend，必要时进行舍入。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY。

**从以下版本开始：**

1.5

**subtract**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **subtract**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) subtrahend)

返回一个 BigDecimal，其值为 (this - subtrahend)，其标度为 max(this.scale(), subtrahend.scale())。

**参数：**

subtrahend - 从此 BigDecimal 减去的值。

**返回：**

this - subtrahend

**subtract**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **subtract**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) subtrahend,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (this - subtrahend) 的 BigDecimal（根据上下文设置进行舍入）。 如果 subtrahend 为零，则将其（必要时进行舍入）作为结果。如果为零，则该结果是subtrahend.negate(mc)。

**参数：**

subtrahend - 从此 BigDecimal 减去的值。

mc - 要使用的上下文。

**返回：**

this - subtrahend，必要时进行舍入。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY。

**从以下版本开始：**

1.5

**multiply**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **multiply**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) multiplicand)

返回一个 BigDecimal，其值为 (this × multiplicand)，其标度为 (this.scale() + multiplicand.scale())。

**参数：**

multiplicand - 乘以此 BigDecimal 的值。

**返回：**

this \* multiplicand

**multiply**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **multiply**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) multiplicand,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (this × multiplicand) 的 BigDecimal（根据上下文设置进行舍入）。

**参数：**

multiplicand - 乘以此 BigDecimal 的值。

mc - 要使用的上下文。

**返回：**

必要时舍入的 this \* multiplicand。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY。

**从以下版本开始：**

1.5

**divide**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divide**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

int scale,

int roundingMode)

返回一个 BigDecimal，其值为 (this / divisor)，其标度为指定标度。如果必须执行舍入，以生成具有指定标度的结果，则应用指定的舍入模式。

相对于此遗留方法，应优先使用新的 [divide(BigDecimal, int, RoundingMode)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20int,%20java.math.RoundingMode)) 方法。

**参数：**

divisor - 此 BigDecimal 要除以的值。

scale - 要返回的 BigDecimal 商的标度。

roundingMode - 要应用的舍入模式。

**返回：**

this / divisor

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor 为零，则 roundingMode==ROUND\_UNNECESSARY 和指定的标度不足以准确地表示相除的结果。

[IllegalArgumentException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/IllegalArgumentException.html) - 如果 roundingMode 不表示一个有效的舍入模式。

**另请参见：**

[ROUND\_UP](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_UP), [ROUND\_DOWN](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_DOWN), [ROUND\_CEILING](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_CEILING), [ROUND\_FLOOR](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_FLOOR), [ROUND\_HALF\_UP](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_UP), [ROUND\_HALF\_DOWN](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_DOWN), [ROUND\_HALF\_EVEN](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_EVEN), [ROUND\_UNNECESSARY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_UNNECESSARY)

**divide**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divide**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

int scale,

[RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html) roundingMode)

返回一个 BigDecimal，其值为 (this / divisor)，其标度为指定标度。如果必须执行舍入，以生成具有指定标度的结果，则应用指定的舍入模式。

**参数：**

divisor - 此 BigDecimal 要除以的值。

scale - 要返回的 BigDecimal 商的标度。

roundingMode - 要应用的舍入模式。

**返回：**

this / divisor

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor 为零，则 roundingMode==RoundingMode.UNNECESSARY 和指定的标度不足以准确地表示相除的结果。

**从以下版本开始：**

1.5

**divide**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divide**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

int roundingMode)

返回一个 BigDecimal，其值为 (this / divisor)，其标度为 this.scale()。如果必须执行舍入，以生成具有给定标度的结果，则应用指定的舍入模式。

相对于此遗留方法，应优先使用新的 [divide(BigDecimal, RoundingMode)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divide(java.math.BigDecimal,%20java.math.RoundingMode)) 方法。

**参数：**

divisor - 此 BigDecimal 要除以的值。

roundingMode - 要应用的舍入模式。

**返回：**

this / divisor

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0 或者 roundingMode==ROUND\_UNNECESSARY 和 this.scale() 不足以准确地表示相除的结果。

[IllegalArgumentException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/IllegalArgumentException.html) - 如果 roundingMode 不表示一个有效的舍入模式。

**divide**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divide**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

[RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html) roundingMode)

返回一个 BigDecimal，其值为 (this / divisor)，其标度为 this.scale()。如果必须执行舍入，以生成具有给定标度的结果，则应用指定的舍入模式。

**参数：**

divisor - 此 BigDecimal 要除以的值。

roundingMode - 要应用的舍入模式。

**返回：**

this / divisor

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0 或者 roundingMode==RoundingMode.UNNECESSARY 和 this.scale() 不足以准确地表示相除的结果。

**divide**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divide**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)

返回一个 BigDecimal，其值为 (this / divisor)，其首选标度为 (this.scale() - divisor.scale())；如果无法表示准确的商值（因为它有无穷的十进制扩展），则抛出ArithmeticException。

**参数：**

divisor - 此 BigDecimal 要相除的值。

**返回：**

this / divisor

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果准确的商值没有无穷的十进制扩展

**从以下版本开始：**

1.5

**divide**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divide**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (this / divisor) 的 BigDecimal（根据上下文设置进行舍入）。

**参数：**

divisor - 此 BigDecimal 要除以的值。

mc - 要使用的上下文。

**返回：**

this / divisor，必要时进行舍入。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY 或 mc.precision == 0，并且商值具有无穷的十进制扩展。

**从以下版本开始：**

1.5

**divideToIntegralValue**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divideToIntegralValue**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)

返回 BigDecimal，其值为向下舍入所得商值 (this / divisor) 的整数部分。该结果的首选标度为 (this.scale() - divisor.scale())。

**参数：**

divisor - 此 BigDecimal 要除以的值。

**返回：**

this / divisor 的整数部分。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0

**从以下版本开始：**

1.5

**divideToIntegralValue**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **divideToIntegralValue**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回 BigDecimal，其值为 (this / divisor) 的整数部分。因为准确商值的整数部分与舍入模式无关，所以舍入模式不影响此方法返回的值。该结果的首选标度是(this.scale() - divisor.scale())。如果准确商值的整数部分需要的位数多于 mc.precision，则抛出 ArithmeticException。

**参数：**

divisor - 此 BigDecimal 要相除的值。

mc - 要使用的上下文。

**返回：**

this / divisor 的整数部分。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 mc.precision > 0，并且该结果需要的精度大于 mc.precision。

**从以下版本开始：**

1.5

**remainder**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **remainder**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)

返回其值为 (this % divisor) 的 BigDecimal。

余数由 this.subtract(this.divideToIntegralValue(divisor).multiply(divisor)) 给出。注意，这不是模操作（结果可以为负）。

**参数：**

divisor - 此 BigDecimal 要除以的值。

**返回：**

this % divisor。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0

**从以下版本开始：**

1.5

**remainder**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **remainder**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (this % divisor) 的 BigDecimal（根据上下文设置进行舍入）。 MathContext 设置会影响用于计算余数的隐式除法。余数计算本身要进行准确的定义。因此，余数包含的数字个数可能多于 mc.getPrecision()。

余数由 this.subtract(this.divideToIntegralValue(divisor, mc).multiply(divisor)) 给出。注意，这不是模操作（结果可以为负）。

**参数：**

divisor - 此 BigDecimal 要相除的值。

mc - 要使用的上下文。

**返回：**

this % divisor，必要时进行舍入。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但舍入模式为 UNNECESSARY，或者 mc.precision > 0，并且 this.divideToIntgralValue(divisor) 的结果要求的精度大于 mc.precision。

**从以下版本开始：**

1.5

**另请参见：**

[divideToIntegralValue(java.math.BigDecimal, java.math.MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideToIntegralValue(java.math.BigDecimal,%20java.math.MathContext))

**divideAndRemainder**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html)[] **divideAndRemainder**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor)

返回由两个元素组成的 BigDecimal 数组，该数组包含 divideToIntegralValue 的结果，后跟对两个操作数计算所得到的 remainder。

注意，如果同时需要整数商和余数，则此方法比分别使用 divideToIntegralValue 和 remainder 方法更快速，因为相除仅需执行一次。

**参数：**

divisor - 此 BigDecimal 要相除的值和计算的余数。

**返回：**

由两个元素组成的 BigDecimal 数组：商值（ divideToIntegralValue 的结果）是初始元素，余数是最终元素。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0

**从以下版本开始：**

1.5

**另请参见：**

[divideToIntegralValue(java.math.BigDecimal, java.math.MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideToIntegralValue(java.math.BigDecimal,%20java.math.MathContext)), [remainder(java.math.BigDecimal, java.math.MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#remainder(java.math.BigDecimal,%20java.math.MathContext))

**divideAndRemainder**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html)[] **divideAndRemainder**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) divisor,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回由两个元素组成的 BigDecimal 数组，该数组包含 divideToIntegralValue 的结果，后跟根据上下文设置对两个操作数进行舍入计算所得到的 remainder 的结果。

注意，如果同时需要整数商和余数，则此方法比分别使用 divideToIntegralValue 和 remainder 方法更快速，因为相除仅需执行一次。

**参数：**

divisor - 此 BigDecimal 要相除的值和计算的余数。

mc - 要使用的上下文。

**返回：**

由两个元素组成的 BigDecimal 数组：商值（ divideToIntegralValue 的结果）是初始元素，余数是最终元素。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 divisor==0

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但舍入模式为 UNNECESSARY，或者 mc.precision > 0，并且 this.divideToIntgralValue(divisor) 的结果要求的精度大于 mc.precision。

**从以下版本开始：**

1.5

**另请参见：**

[divideToIntegralValue(java.math.BigDecimal, java.math.MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#divideToIntegralValue(java.math.BigDecimal,%20java.math.MathContext)), [remainder(java.math.BigDecimal, java.math.MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#remainder(java.math.BigDecimal,%20java.math.MathContext))

**pow**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **pow**(int n)

返回其值为 (thisn) 的 BigDecimal，准确计算该幂，使其具有无限精度。

参数 n 必须在 0 到 999999999（包括）之间。ZERO.pow(0) 返回 [ONE](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ONE)。 注意，未来版本可能会扩展此方法允许的指数范围。

**参数：**

n - 此 BigDecimal 的幂。

**返回：**

thisn

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 n 超出范围。

**pow**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **pow**(int n,

[MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (thisn) 的 BigDecimal。当前实现使用的是 ANSI 标准 X3.274-1996 中定义的核心算法（根据上下文设置进行舍入）。一般情况下，返回的数值在具有选择精度的精确数值的两个 ulp 中。注意，未来版本可以使用不同的算法，以减少允许的错误范围，并增加允许的指数范围。

X3.274-1996 算法为：

* 如果出现以下情况，则抛出 ArithmeticException 异常
  + abs(n) > 999999999
  + mc.precision == 0 且 n < 0
  + mc.precision > 0 且 n 大于 mc.precision
* 如果 n 为零，则返回 [ONE](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ONE)（即使 this 为零），否则
  + 如果 n 为正数，则通过重复的平方技术将此结果计算到一个累加器中。每个附带累加器的乘法都使用与 mc 中相同的数学上下文设置，增加到 mc.precision + elength + 1 的精度除外，其中 elength 表示 n 中的十进制数字的个数。
  + 如果 n 为负数，则将 n 视为正数计算其结果；然后使用上面指定的工作精度，用一除以此值。
  + 然后将正数或负数的最终值舍入为目标精度。

**参数：**

n - 此 BigDecimal 的幂。

mc - 要使用的上下文。

**返回：**

使用 ANSI 标准 X3.274-1996 算法的 thisn

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果该结果不准确，但是舍入模式为 UNNECESSARY，或者 n 超出范围。

**从以下版本开始：**

1.5

**abs**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **abs**()

返回 BigDecimal，其值为此 BigDecimal 的绝对值，其标度为 this.scale()。

**返回：**

abs(this)

**abs**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **abs**([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为此 BigDecimal 绝对值的 BigDecimal（根据上下文设置进行舍入）。

**参数：**

mc - 要使用的上下文。

**返回：**

abs(this)，必要时进行舍入。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY。

**从以下版本开始：**

1.5

**negate**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **negate**()

返回 BigDecimal，其值为 (-this)，其标度为 this.scale()。

**返回：**

-this。

**negate**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **negate**([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (-this) 的 BigDecimal（根据上下文设置进行舍入）。

**参数：**

mc - 要使用的上下文。

**返回：**

abs(this)，必要时进行舍入。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY。

**从以下版本开始：**

1.5

**plus**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **plus**()

返回 BigDecimal，其值为 (+this)，其标度为 this.scale()。

此方法仅返回此 BigDecimal，该方法与一元减方法 [negate()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#negate()) 对称。

**返回：**

this。

**plus**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **plus**([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回其值为 (+this) 的 BigDecimal（根据上下文设置进行舍入）。

此方法的效果与 [round(MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#round(java.math.MathContext)) 方法的效果相同。

**参数：**

mc - 要使用的上下文。

**返回：**

this，必要时进行舍入。零结果具有的标度为 0。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果结果不准确，但是舍入模式为 UNNECESSARY。

**signum**

public int **signum**()

返回此 BigDecimal 的正负号函数。

**返回：**

当此 BigDecimal 的值为负、零或正时，返回 -1、0 或 1。

**scale**

public int **scale**()

返回此 BigDecimal 的 *标度*。如果为零或正数，则标度是小数点后的位数。如果为负数，则将该数的非标度值乘以 10 的负 scale 次幂。例如， -3 标度是指非标度值乘以 1000。

**返回：**

此 BigDecimal 的标度。

**precision**

public int **precision**()

返回此 BigDecimal 的 *精度*。（精度是非标度值的数字个数。）

零值的精度是 1。

**返回：**

此 BigDecimal 的精度。

**unscaledValue**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **unscaledValue**()

返回其值为此 BigDecimal 的 *非标度值* 的 BigInteger。（计算 (this \* 10this.scale())。）

**返回：**

此 BigDecimal 的非标度值。

**round**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **round**([MathContext](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/MathContext.html) mc)

返回根据 MathContext 设置进行舍入后的 BigDecimal。如果精度设置为 0，则不进行任何舍入操作。

此方法的效果与 [plus(MathContext)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#plus(java.math.MathContext)) 方法的效果相同。

**参数：**

mc - 要使用的上下文。

**返回：**

根据 MathContext 设置舍入后的 BigDecimal。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果舍入模式为 UNNECESSARY，并且 BigDecimal 操作要求进行舍入。

**setScale**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **setScale**(int newScale,

[RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html) roundingMode)

返回 BigDecimal，其标度为指定值，其非标度值通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定，以维护其总值。如果该操作减少标度，则非标度值必须被除（而不是乘），并且该值可以更改；在这种情况下，将指定的舍入模式应用到除法中。

**参数：**

newScale - 要返回的 BigDecimal 值的标度。

roundingMode - 要应用的舍入模式。

**返回：**

一个 BigDecimal，其标度为指定值，其非标度值可以通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定，以维护其总值。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 roundingMode==UNNECESSARY，并且指定的标度操作要求进行舍入。

**从以下版本开始：**

1.5

**另请参见：**

[RoundingMode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/RoundingMode.html)

**setScale**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **setScale**(int newScale,

int roundingMode)

返回一个 BigDecimal，其标度为指定值，其非标度值通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定，以维护其总值。如果该操作减少标度，则非标度值必须被除（而不是乘），并且该值可以更改；在这种情况下，将指定的舍入模式应用到除法中。

注意，由于 BigDecimal 对象是不可变的，此方法的调用*不会* 导致初始对象被修改，这与使用名为 set*X* 变异字段 *X* 方法的常规约定相反。相反，setScale 返回具有适当标度的对象；返回的对象不一定是新分配的。

相对于此遗留方法，应优先使用新的 [setScale(int, RoundingMode)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int,%20java.math.RoundingMode)) 方法。

**参数：**

newScale - 要返回的 BigDecimal 值的标度。

roundingMode - 要应用的舍入模式。

**返回：**

一个 BigDecimal，其标度为指定值，其非标度值可以通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 roundingMode==ROUND\_UNNECESSARY，并且指定的标度操作要求进行舍入。

[IllegalArgumentException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/IllegalArgumentException.html) - 如果 roundingMode 不表示一个有效的舍入模式。

**另请参见：**

[ROUND\_UP](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_UP), [ROUND\_DOWN](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_DOWN), [ROUND\_CEILING](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_CEILING), [ROUND\_FLOOR](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_FLOOR), [ROUND\_HALF\_UP](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_UP), [ROUND\_HALF\_DOWN](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_DOWN), [ROUND\_HALF\_EVEN](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_HALF_EVEN), [ROUND\_UNNECESSARY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#ROUND_UNNECESSARY)

**setScale**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **setScale**(int newScale)

返回一个 BigDecimal，其标度为指定值，其值在数值上等于此 BigDecimal 的值。如果这不可能，则抛出 ArithmeticException。

此调用通常用于增加标度，在这种情况下，可以保证存在指定标度和正确值的 BigDecimal。如果调用方知道 BigDecimal在其小数部分的结尾有足够多的零（即其整数值中的十的因子），则该调用也可用于减少标度，以允许重新标度，而不更改其值。

此方法返回与 setScale 的两个参数版本相同的结果，但是，为调用方省去了指定舍入模式的麻烦（舍入模式不影响结果）。

注意，由于 BigDecimal 对象是不可变的，因此此方法的调用*不会* 导致初始对象被修改，这与使用名为 set*X* 变异字段 *X* 方法的常规约定相反。相反，setScale 返回具有适当标度的对象；返回的对象不一定是新分配的。

**参数：**

newScale - 要返回的 BigDecimal 值的标度。

**返回：**

一个 BigDecimal，其标度为指定值，其非标度值可以通过此 BigDecimal 的非标度值乘以或除以十的适当次幂来确定，以维护其总值。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果指定的标度操作要求进行舍入。

**另请参见：**

[setScale(int, int)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int,%20int)), [setScale(int, RoundingMode)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int,%20java.math.RoundingMode))

**movePointLeft**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **movePointLeft**(int n)

返回一个 BigDecimal，它等效于将该值的小数点向左移动 n 位。如果 n 为非负数，则调用仅将 n 添加到该标度。如果 n 为负数，则该调用等效于 movePointRight(-n)。此调用返回的 BigDecimal 的值为 (this × 10-n)，标度为 max(this.scale()+n, 0)。

**参数：**

n - 将小数点向左移动的位数。

**返回：**

一个 BigDecimal，它等效于将该值的小数点向左移动 n 位。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果标度溢出。

**movePointRight**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **movePointRight**(int n)

返回一个 BigDecimal，它等效于将该值的小数点向右移动 n 位。如果 n 为非负数，则该调用仅从该标度减去 n。如果 n 为负，则该调用等效于 movePointLeft(-n)。此调用返回的 BigDecimal 的值为 (this × 10n)，标度为 max(this.scale()-n, 0)。

**参数：**

n - 将小数点向右移动的位数。

**返回：**

一个 BigDecimal，它等效于将该值的小数点向右移动 n 位。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果标度溢出。

**scaleByPowerOfTen**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **scaleByPowerOfTen**(int n)

返回其数值等于 ( this \* 10 n) 的 BigDecimal。该结果的标度为 (this.scale() - n)。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果该标度的范围超出 32 位整数的范围。

**从以下版本开始：**

1.5

**stripTrailingZeros**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **stripTrailingZeros**()

返回数值上等于此小数，但从该表示形式移除所有尾部零的 BigDecimal。例如，从 BigDecimal 值 600.0 中移除尾部零，该值具有的 [ BigInteger, scale] 组件等于 [6000, 1]，使用 [ BigInteger, scale] 组件生成的 6E2 等于 [6, -2]

**返回：**

数值上等于移除所有尾部零的 BigDecimal。

**从以下版本开始：**

1.5

**compareTo**

public int **compareTo**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) val)

将此 BigDecimal 与指定的 BigDecimal 比较。根据此方法，值相等但具有不同标度的两个 BigDecimal 对象（如，2.0 和 2.00）被认为是相等的。相对六个 boolean 比较运算符 (<, ==, >, >=, !=, <=) 中每一个运算符的各个方法，优先提供此方法。建议使用以下语句执行上述比较： (x.compareTo(y) < *op*> 0)，其中 < *op*> 是六个比较运算符之一。

**指定者：**

接口 [Comparable](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Comparable.html)<[BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html)> 中的 [compareTo](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Comparable.html#compareTo(T))

**参数：**

val - 将此 BigDecimal 与之比较的 BigDecimal。

**返回：**

当此 BigDecimal 在数字上小于、等于或大于 val 时，返回 -1、0 或 1。

**equals**

public boolean **equals**([Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) x)

比较此 BigDecimal 与指定的 Object 的相等性。与 [compareTo](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal)) 不同，仅当两个 BigDecimal 对象的值和标度都相等时，此方法才认为它们相等（因此通过此方法进行比较时，2.0 不等于 2.00）。

**覆盖：**

类 [Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) 中的 [equals](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html#equals(java.lang.Object))

**参数：**

x - 将与此 BigDecimal 进行比较的 Object。

**返回：**

当且仅当指定的 Object 为 BigDecimal，并且其值和标度都等于此 BigDecimal 的值和标度时，返回 true。

**另请参见：**

[compareTo(java.math.BigDecimal)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal)), [hashCode()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#hashCode())

**min**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **min**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) val)

返回此 BigDecimal 和 val 的最小值。

**参数：**

val - 要计算最小值的值。

**返回：**

其值为 BigDecimal 和 val 中较小值的 BigDecimal。根据 [compareTo](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal)) 方法的定义，如果它们相等，则返回 this。

**另请参见：**

[compareTo(java.math.BigDecimal)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal))

**max**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **max**([BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) val)

返回此 BigDecimal 和 val 的最大值。

**参数：**

val - 要计算最大值的值。

**返回：**

其值为此 BigDecimal 和 val 中较大值的 BigDecimal。根据 [compareTo](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal)) 方法的定义，如果它们相等，则返回 this。

**另请参见：**

[compareTo(java.math.BigDecimal)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#compareTo(java.math.BigDecimal))

**hashCode**

public int **hashCode**()

返回此 BigDecimal 的哈希码。注意，数值上相等但标度不同的两个 BigDecimal 对象（如，2.0 和 2.00）通常 *没有* 相同的哈希码。

**覆盖：**

类 [Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) 中的 [hashCode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html#hashCode())

**返回：**

此 BigDecimal 的哈希码。

**另请参见：**

[equals(Object)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#equals(java.lang.Object))

**toString**

public [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) **toString**()

返回此 BigDecimal 的字符串表示形式，如果需要指数，则使用科学记数法。

根据以下步骤可创建 BigDecimal 的标准的规范化字符串形式：首先，使用不带前导零的字符 '0' 至 '9'（如果其值为零，则使用单个 '0' 字符）将 BigDecimal 的非标度值的绝对值转换为以十为基数的字符串。

其次，计算*调整的指数*；这是无效的标度，加上转换的非标度值中的字符数减 1。即：-scale+(ulength-1)，其中 ulength 是十进制数字中非标度值的绝对值的长度（其*精度*）。

如果该标度大于或等于零，并且调整的指数大于或等于 -6，则在不使用指数记数法的情况下将该数转换为字符形式。在这种情况下，如果标度为零，则不添加小数点，如果标度为正数，则插入小数点，且标度指定了小数点右边的字符个数。必要时，将字符 '0' 添加到转换的非标度值的左边。如果插入后小数点前面没有字符，则以传统字符 '0' 为前缀。

否则（即：如果标度为负数，或者调整的指数小于 -6），使用指数记数法将该数转换为字符形式。在这种情况下，如果转换后的 BigInteger 多于一位数，则小数点插入在第一个数字之后。然后，将字符形式的指数作为转换的非标度值（也许具有插入的小数点）的后缀；这包含字母 'E' 和直接跟在其后的转换为字符形式的调整指数。后者的基数为十，使用的字符是 '0' 到 '9'，没有前导零，并且，如果调整的指数为负数，则总是以符号字符 '-' ('\u002D') 为前缀，否则以 '+' ('\u002B') 为前缀。

最后，如果非标度值小于零，则整个字符串以减号 '-' ('\u002D') 为前缀。如果非标度值为零或正数，则不使用符号字符作为前缀。

**示例：**

对于左边的每个表示形式 [*unscaled value*, *scale*]，得到的字符串显示在右边。

[123,0] "123"

[-123,0] "-123"

[123,-1] "1.23E+3"

[123,-3] "1.23E+5"

[123,1] "12.3"

[123,5] "0.00123"

[123,10] "1.23E-8"

[-123,12] "-1.23E-10"

**注：**

1. 可区分的 BigDecimal 值和此转换的结果之间存在一对一的映射关系。即：每个可区分的 BigDecimal 值（非标度值和标度）都有唯一的字符串表示形式，作为使用 toString 的结果。如果使用 [BigDecimal(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 构造方法将该字符串表示形式转换为 BigDecimal，则将恢复初始值。
2. 给定的数产生的字符串总是相同的；它不受语言环境的影响。这意味着它可以用作交换十进制数据的规范化字符串表示形式，或用作 Hashtable 的关键字等等。[NumberFormat](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/text/NumberFormat.html) 类及其子类可以处理区分语言环境的数的格式化和分析。
3. [toEngineeringString()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toEngineeringString()) 方法可用于在工程计数法中表示带有指数的数。[setScale](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#setScale(int,%20java.math.RoundingMode)) 方法可用于对 BigDecimal 进行舍入，使其小数点后的位数为已知位数。
4. 使用 Character.forDigit 提供的数字到字符的映射关系。

**覆盖：**

类 [Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) 中的 [toString](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html#toString())

**返回：**

此 BigDecimal 的字符串表示形式。

**另请参见：**

[Character.forDigit(int, int)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Character.html#forDigit(int,%20int)), [BigDecimal(java.lang.String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String))

**toEngineeringString**

public [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) **toEngineeringString**()

返回此 BigDecimal 的字符串表示形式，需要指数时，则使用工程计数法。

返回如 [toString()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toString()) 方法中所描述的表示 BigDecimal 的字符串，不包括使用指数记数法的情况，将十的幂调整为三的倍数（工程记数法），这样，非零值的整数部分的范围是 1 到 999。如果对零值使用指数记数法，则使用小数点和小数的一（或二）个零数字，以便保留零值的标度。注意，与 [toString()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toString()) 的输出不同，如果使用 [string constructor](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String)) 将输出字符串转换为 BigDecimal，则此方法的输出*不* 保证恢复此 BigDecimal 的相同 [integer, scale] 对。此方法满足以下较弱约束，即产生的结果在数值上始终等于将字符串构造方法应用到方法的输出得到的结果。

**返回：**

此 BigDecimal 的字符串表示形式，如果需要指数，则使用工程记数法。

**从以下版本开始：**

1.5

**toPlainString**

public [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) **toPlainString**()

返回不带指数字段的此 BigDecimal 的字符串表示形式。对于具有正标度的值，小数点右边的数字个数用于指示标度。对于具有零或负标度的值，生成得到的字符串，好像将该值转换为在数值上等于具有零标度的值一样，并且好像零标度值的所有尾部零都出现在该结果中。 如果非标度值小于零，则整个字符串以减号 '-' ( '\u002D') 为前缀。如果非标度值为零或正数，则没有任何符号字符作为前缀。 注意，如果将此方法的结果传递到 [string constructor](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#BigDecimal(java.lang.String))，则只需要恢复此 BigDecimal 的数值；新的BigDecimal 的表示形式可以有不同的标度。尤其是，如果此 BigDecimal 具有负标度，则在由字符串构造方法进行处理时，此方法产生的字符串将具有零标度。 （此方法的作用类似于 1.4 和更早版本中的 toString 方法。）

**返回：**

不带指数字段的此 BigDecimal 的字符串表示形式。

**从以下版本开始：**

1.5

**另请参见：**

[toString()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toString()), [toEngineeringString()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toEngineeringString())

**toBigInteger**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **toBigInteger**()

将此 BigDecimal 转换为 BigInteger。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 long 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：将丢弃此 BigDecimal 的小数部分。注意，此转换会丢失关于 BigDecimal 值的精度信息。

要在转换不准确时（即当丢弃非零小数部分时）抛出异常，请使用 [toBigIntegerExact()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html#toBigIntegerExact()) 方法。

**返回：**

转换为 BigInteger 的此 BigDecimal。

**toBigIntegerExact**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **toBigIntegerExact**()

将此 BigDecimal 转换为 BigInteger，以检查丢失的信息。如果此 BigDecimal 具有非零小数部分，则抛出一个异常。

**返回：**

转换为 BigInteger 的此 BigDecimal。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 this 具有非零小数部分。

**从以下版本开始：**

1.5

**longValue**

public long **longValue**()

将此 BigDecimal 转换为 long。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 short 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：将丢弃此 BigDecimal 的小数部分，并且如果生成的 " BigInteger" 太大而不适合用 long 表示，则仅返回 64 位低位字节。注意，此转换会丢失关于此 BigDecimal 值的总大小和精度的信息，并返回带有相反符号的结果。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [longValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html#longValue())

**返回：**

转换为 long 的此 BigDecimal。

**longValueExact**

public long **longValueExact**()

将此 BigDecimal 转换为 long，以检查丢失的信息。如果此 BigDecimal 具有非零小数部分，或者超出 long 结果的可能范围，则抛出 ArithmeticException。

**返回：**

转换为 long 的此 BigDecimal。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 this 具有非零小数部分，或者不适合用 long 表示。

**intValue**

public int **intValue**()

将此 BigDecimal 转换为 int。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 short 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：将丢弃此 BigDecimal 的所有小数部分，并且如果生成的 " BigInteger" 太大而不适合用 int 表示，则仅返回 32 位低位字节。注意，此转换会丢失关于此 BigDecimal 值的总大小和精度的信息，并返回带有相反符号的结果。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [intValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html#intValue())

**返回：**

转换为 int 的此 BigDecimal。

**intValueExact**

public int **intValueExact**()

将此 BigDecimal 转换为 int，以检查丢失的信息。如果此 BigDecimal 具有非零小数部分，或者超出 int 结果的可能范围，则抛出 ArithmeticException。

**返回：**

转换为 int 的此 BigDecimal。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 this 具有非零小数部分，或者不适合用 int 表示。

**shortValueExact**

public short **shortValueExact**()

将此 BigDecimal 转换为 short，以检查丢失的信息。如果此 BigDecimal 具有非零小数部分，或者超出 short 结果的可能范围，则抛出 ArithmeticException。

**返回：**

转换为 short 的此 BigDecimal。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 this 具有非零小数部分，或者不适合用 short 表示。

**byteValueExact**

public byte **byteValueExact**()

将此 BigDecimal 转换为 byte，以检查丢失的信息。如果此 BigDecimal 具有非零小数部分，或者超出 byte 结果的可能范围，则抛出 ArithmeticException。

**返回：**

转换为 byte 的此 BigDecimal。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - 如果 this 具有非零小数部分，或者不适合用 byte 表示。

**从以下版本开始：**

1.5

**floatValue**

public float **floatValue**()

将此 BigDecimal 转换为 float。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 float 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：如此 BigDecimal 的值太大而不能表示为 float，则将其适当地转换为 [Float.NEGATIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Float.html#NEGATIVE_INFINITY) 或 [Float.POSITIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Float.html#POSITIVE_INFINITY)。注意，即使在返回值为有限值的情况下，此转换也可能丢失关于 BigDecimal 值精度的信息。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [floatValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html#floatValue())

**返回：**

转换为 float 的此 BigDecimal。

**doubleValue**

public double **doubleValue**()

将此 BigDecimal 转换为 double。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 float 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：如果此 BigDecimal 的数量太大而不能表示为double，则将其适当地转换为 [Double.NEGATIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html#NEGATIVE_INFINITY) 或 [Double.POSITIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html#POSITIVE_INFINITY)。注意，即使在返回值为有限值的情况下，此转换也可能丢失关于 BigDecimal 值精度的信息。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [doubleValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html#doubleValue())

**返回：**

转换为 double 的此 BigDecimal。

**ulp**

public [BigDecimal](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigDecimal.html) **ulp**()

返回此 BigDecimal 的 ulp（最后一位的单位）的大小。非零 BigDecimal 值的 ulp 是此值与下一个具有相同位数的较大 BigDecimal 值之间的正距离。零值的 ulp 在数值上等于具有 this 标度的 1。使用与 this 相同的标度存储该结果，这样，零和非零值的结果等于 [1, this.scale()]。

**返回：**

this 的 ulp 的大小

**从以下版本开始：**

1.5