JAVA.MATH.BIGDECIMAL CLASS

Introduction

The **java.math.BigDecimal** class provides operations for arithmetic, scale manipulation, rounding, comparison, hashing, and format conversion.

The toString method provides a canonical representation of a BigDecimal. It gives the user complete control over rounding behavior.

Two types of operations are provided for manipulating the scale of a BigDecimal:

- scaling/rounding operations
- decimal point motion operations.

This class and its iterator implement all of the optional methods of the **Comparable** interfaces.

Class declaration

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

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

Field

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

- **static BigDecimal ONE** -- The value 1, with a scale of 0.
- **static int ROUND_CEILING** -- Rounding mode to round towards positive infinity.
- **static int ROUND_DOWN** -- Rounding mode to round towards zero.
- **static int ROUND_FLOOR** -- Rounding mode to round towards negative infinity.
- **static int ROUND_HALF_DOWN** -- Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round down.
- **static int ROUND_HALF_EVEN** -- Rounding mode to round towards the "nearest neighbor" unless both neighbors are equidistant, in which case, round towards the even neighbor.
- **static int ROUND_HALF_UP** --Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round up.
- **static int ROUND_UNNECESSARY** --Rounding mode to assert that the requested operation has an exact result, hence no rounding is necessary.
- **static int ROUND_UP** --Rounding mode to round away from zero.
- **static BigDecimal TEN** --The value 10, with a scale of 0.
- **static BigDecimal ZERO** --The value 0, with a scale of 0.

Class constructors

S.N.	Constructor & Description
1	**BigDecimal***BigIntegerval*

This constructor is used to translates a BigInteger into a BigDecimal.

2 **BigDecimal***BigIntegerunscaledVal, intscale*

This constructor is used to translate a BigInteger unscaled value and an int scale into a BigDecimal.

3 **BigDecimal***BigIntegerunscaledVal, intscale, MathContextmc*

This constructor is used to translate a BigInteger unscaled value and an int scale into a BigDecimal, with rounding according to the context settings.

4 **BigDecimal***BigIntegerval, MathContextmc*

This constructor is used to translate a BigInteger into a BigDecimal rounding according to the context settings.

5 **BigDecimal***char[]in*

This constructor is used to translate a character array representation of a BigDecimal into a BigDecimal, accepting the same sequence of characters as the BigDecimal*String* constructor.

6 **BigDecimal***char[]in, intoffset, intlen*

This constructor is used to translate a character array representation of a BigDecimal into a BigDecimal, accepting the same sequence of characters as the BigDecimal*String* constructor, while allowing a sub-array to be specified.

7 **BigDecimal***char[]in, intoffset, intlen, MathContextmc*

This constructor is used to translate a character array representation of a BigDecimal into a BigDecimal, accepting the same sequence of characters as the BigDecimal*String* constructor, while allowing a sub-array to be specified and with rounding according to the context settings.

8 **BigDecimal***char[]in, MathContextmc*

This constructor is used to translate a character array representation of a BigDecimal into a BigDecimal, accepting the same sequence of characters as the BigDecimal*String* constructor and with rounding according to the context settings.

9 **BigDecimal***doubleval*

This constructor is used to translates a double into a BigDecimal which is the exact decimal representation of the double's binary floating-point value.

10 **BigDecimal***doubleval, MathContextmc*

This constructor is used to translates a double into a BigDecimal, with rounding according to the context settings.

11 **BigDecimal***intval*

This constructor is used to translates an int into a BigDecimal.

12 **BigDecimal***intval, MathContextmc*

This constructor is used to translates an int into a BigDecimal, with rounding according to the context settings.

13 **BigDecimal*longval***

This constructor is used to translate a long into a BigDecimal.

14 **BigDecimal*longval, MathContextmc***

This constructor is used to translates a BigInteger into a BigDecimal.

15 **BigDecimal*Stringval***

This constructor is used to the string representation of a BigDecimal into a BigDecimal.

16 **BigDecimal*Stringval, MathContextmc***

This constructor is used to translates the string representation of a BigDecimal into a BigDecimal, accepting the same strings as the BigDecimal*String* constructor, with rounding according to the context settings.

Class methods

S.N.	Method & Description
1	BigDecimal abs This method returns a BigDecimal whose value is the absolute value of this BigDecimal, and whose scale is this.scale.
2	BigDecimal abs*MathContextmc* This method returns a BigDecimal whose value is the absolute value of this BigDecimal, with rounding according to the context settings.
3	BigDecimal add*BigDecimalaugend* This method returns a BigDecimal whose value is *this* + *augend*, and whose scale is max *this. scale*(, augend.scale).
4	BigDecimal add*BigDecimalaugend, MathContextmc* This method returns a BigDecimal whose value is *this* + *augend*, with rounding according to the context settings.
5	byte byteValueExact This method converts the BigDecimal to a byte, checking for lost information.
6	int compareTo*BigDecimalval*

This method compares the BigDecimal with the specified BigDecimal.

7

BigDecimal divide*BigDecimaldivisor*

This method returns a BigDecimal whose value is *this*/*divisor*, and whose preferred scale is *this. scale*(- divisor.scale); if the exact quotient cannot be represented *becauseithasanon − terminatingdecimalexpansion* an ArithmeticException is thrown.

8

BigDecimal divide*BigDecimaldivisor, introundingMode*

This method returns a BigDecimal whose value is *this*/*divisor*, and whose scale is this.scale.

9

BigDecimal divide*BigDecimaldivisor, intscale, introundingMode*

This method returns a BigDecimal whose value is *this*/*divisor*, and whose scale is as specified.

10

BigDecimal divide*BigDecimaldivisor, intscale, RoundingModeroundingMode*

This method returns a BigDecimal whose value is *this*/*divisor*, and whose scale is as specified.

11

BigDecimal divide*BigDecimaldivisor, MathContextmc*

This method returns a BigDecimal whose value is *this*/*divisor*, with rounding according to the context settings.

12

BigDecimal divide*BigDecimaldivisor, RoundingModeroundingMode*

This method returns a BigDecimal whose value is *this*/*divisor*, and whose scale is this.scale.

13

BigDecimal[ ] divideAndRemainder*BigDecimaldivisor*

This method returns a two-element BigDecimal array containing the result of divideToIntegralValue followed by the result of remainder on the two operands.

14

BigDecimal[ ] divideAndRemainder*BigDecimaldivisor, MathContextmc*

This method returns a two-element BigDecimal array containing the result of divideToIntegralValue followed by the result of remainder on the two operands calculated with rounding according to the context settings.

15

BigDecimal divideToIntegralValue*BigDecimaldivisor*

This method returns a BigDecimal whose value is the integer part of the quotient *this*/*divisor* rounded down.

16

BigDecimal divideToIntegralValue*BigDecimaldivisor, MathContextmc*

This method returns a BigDecimal whose value is the integer part of *this*/*divisor*.

17

double doubleValue

This method converts the BigDecimal to a double.

18

boolean equals*Objectx*

This method compares the BigDecimal with the specified Object for equality.

19

float floatValue

This method converts the BigDecimal to a float.

20

int hashCode

This method returns the hash code for this BigDecimal.

21

int intValue

This method converts the BigDecimal to an int.

22

int intValueExact

This method converts the BigDecimal to an int, checking for lost information.

23

long longValue

This method converts the BigDecimal to a long.

24

long longValueExact

This method converts the BigDecimal to a long, checking for lost information.

25

BigDecimal max*BigDecimalval*

This method returns the maximum of this BigDecimal and val.

26

BigDecimal min*BigDecimalval*

This method returns the minimum of this BigDecimal and val.

27

BigDecimal movePointLeft*intn*

This method returns a BigDecimal which is equivalent to this one with the decimal point moved n places to the left.

28

BigDecimal movePointRight*intn*

This method returns a BigDecimal which is equivalent to this one with the decimal point moved n places to the right.

29

BigDecimal multiply*BigDecimalmultiplicand*

This method returns a BigDecimal whose value is $this \times multiplicand$, and whose scale is *this. scale*(+ multiplicand.scale).

30

BigDecimal multiply*BigDecimalmultiplicand, MathContextmc*

This method returns a BigDecimal whose value is $this \times multiplicand$, with rounding according to the context settings.

31

BigDecimal negate

This method returns a BigDecimal whose value is $+this$, and whose scale is this.scale.

32

BigDecimal negate*MathContextmc*

This method returns a BigDecimal whose value is $-this$, with rounding according to the context settings.

33

BigDecimal plus

This method returns a BigDecimal whose value is $+this$, and whose scale is this.scale.

34

BigDecimal plus*MathContextmc*

This method returns a BigDecimal whose value is $+this$, with rounding according to the context settings.

35

BigDecimal pow*intn*

This method returns a BigDecimal whose value is ($this^n$), The power is computed exactly, to unlimited precision.

36

BigDecimal pow*intn, MathContextmc*

This method returns a BigDecimal whose value is ($this^n$).

37

int precision

This method returns the precision of this BigDecimal.

38

BigDecimal remainder*BigDecimaldivisor*

This method converts this BigDecimal to a byte, checking for lost information.

39

BigDecimal remainder*BigDecimaldivisor, MathContextmc*

This method returns a BigDecimal whose value is *this*, with rounding according to the context settings.

40

BigDecimal round*MathContextmc*

This method returns a BigDecimal rounded according to the MathContext settings.

41

int scale

This method returns the scale of this BigDecimal.

42

BigDecimal scaleByPowerOfTen*intn*

This method returns a BigDecimal whose numerical value is equal to (this * 10^n).

43

BigDecimal setScale*intnewScale*

This method returns a BigDecimal whose scale is the specified value, and whose value is numerically equal to this BigDecimal's.

44

BigDecimal setScale*intnewScale, introundingMode*

This method 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.

45

BigDecimal setScale*intnewScale, RoundingModeroundingMode*

This method 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.

46

short shortValueExact

This method converts the BigDecimal to a short, checking for lost information.

47

int signum

This method returns the signum function of this BigDecimal.

48

BigDecimal stripTrailingZeros

This method returns a BigDecimal which is numerically equal to this one but with any trailing zeros removed from the representation.

49

BigDecimal subtract*BigDecimalsubtrahend*

This method returns a BigDecimal whose value is *this − subtrahend*, and whose scale is max

this. scale(, subtrahend.scale).

50

BigDecimal subtract*BigDecimalsubtrahend, MathContextmc*

This method returns a BigDecimal whose value is *this − subtrahend*, with rounding according to the context settings.

51

BigInteger toBigInteger

This method converts the BigDecimal to a BigInteger.

52

BigInteger toBigIntegerExact

This method converts the BigDecimal to a BigInteger, checking for lost information.

53

String toEngineeringString

This method returns a string representation of this BigDecimal, using engineering notation if an exponent is needed.

54

String toPlainString

This method returns a string representation of this BigDecimal without an exponent field.

55

String toString

This method returns the string representation of this BigDecimal, using scientific notation if an exponent is needed.

56

BigDecimal ulp

This method returns the size of an ulp, a unit in the last place, of this BigDecimal.

57

BigInteger unscaledValue

This method returns a BigInteger whose value is the unscaled value of this BigDecimal.

58

static BigDecimal valueOf*doubleval*

This method translates a double into a BigDecimal, using the double's canonical string representation provided by the Double.toString*double* method.

59

static BigDecimal valueOf*longval*

This method translates a long value into a BigDecimal with a scale of zero.

60

static BigDecimal valueOf*longunscaledVal, intscale*

This method translates a long unscaled value and an int scale into a BigDecimal.