| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MathContext.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/math/BigInteger.html)   [**NEXT CLASS**](http://docs.google.com/java/math/RoundingMode.html) | [**FRAMES**](http://docs.google.com/index.html?java/math/MathContext.html)    [**NO FRAMES**](http://docs.google.com/MathContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#44sinio) |

## **java.math**

Class MathContext

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.math.MathContext**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

public final class **MathContext**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

Immutable objects which encapsulate the context settings which describe certain rules for numerical operators, such as those implemented by the [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) class.

The base-independent settings are:

1. precision: the number of digits to be used for an operation; results are rounded to this precision
2. roundingMode: a [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) object which specifies the algorithm to be used for rounding.

**Since:** 1.5 **See Also:**[BigDecimal](http://docs.google.com/java/math/BigDecimal.html), [RoundingMode](http://docs.google.com/java/math/RoundingMode.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.math.MathContext)

| **Field Summary** | |
| --- | --- |
| static [MathContext](http://docs.google.com/java/math/MathContext.html) | [**DECIMAL128**](http://docs.google.com/java/math/MathContext.html#DECIMAL128)            A MathContext object with a precision setting matching the IEEE 754R Decimal128 format, 34 digits, and a rounding mode of [HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), the IEEE 754R default. |
| static [MathContext](http://docs.google.com/java/math/MathContext.html) | [**DECIMAL32**](http://docs.google.com/java/math/MathContext.html#DECIMAL32)            A MathContext object with a precision setting matching the IEEE 754R Decimal32 format, 7 digits, and a rounding mode of [HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), the IEEE 754R default. |
| static [MathContext](http://docs.google.com/java/math/MathContext.html) | [**DECIMAL64**](http://docs.google.com/java/math/MathContext.html#DECIMAL64)            A MathContext object with a precision setting matching the IEEE 754R Decimal64 format, 16 digits, and a rounding mode of [HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), the IEEE 754R default. |
| static [MathContext](http://docs.google.com/java/math/MathContext.html) | [**UNLIMITED**](http://docs.google.com/java/math/MathContext.html#UNLIMITED)            A MathContext object whose settings have the values required for unlimited precision arithmetic. |

| **Constructor Summary** | |
| --- | --- |
| [**MathContext**](http://docs.google.com/java/math/MathContext.html#MathContext(int))(int setPrecision)            Constructs a new MathContext with the specified precision and the [HALF\_UP](http://docs.google.com/java/math/RoundingMode.html#HALF_UP) rounding mode. |
| [**MathContext**](http://docs.google.com/java/math/MathContext.html#MathContext(int,%20java.math.RoundingMode))(int setPrecision, [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) setRoundingMode)            Constructs a new MathContext with a specified precision and rounding mode. |
| [**MathContext**](http://docs.google.com/java/math/MathContext.html#MathContext(java.lang.String))([String](http://docs.google.com/java/lang/String.html) val)            Constructs a new MathContext from a string. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/math/MathContext.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) x)            Compares this MathContext with the specified Object for equality. |
| int | [**getPrecision**](http://docs.google.com/java/math/MathContext.html#getPrecision())()            Returns the precision setting. |
| [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) | [**getRoundingMode**](http://docs.google.com/java/math/MathContext.html#getRoundingMode())()            Returns the roundingMode setting. |
| int | [**hashCode**](http://docs.google.com/java/math/MathContext.html#hashCode())()            Returns the hash code for this MathContext. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/math/MathContext.html#toString())()            Returns the string representation of this MathContext. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### UNLIMITED

public static final [MathContext](http://docs.google.com/java/math/MathContext.html) **UNLIMITED**

A MathContext object whose settings have the values required for unlimited precision arithmetic. The values of the settings are: precision=0 roundingMode=HALF\_UP

### DECIMAL32

public static final [MathContext](http://docs.google.com/java/math/MathContext.html) **DECIMAL32**

A MathContext object with a precision setting matching the IEEE 754R Decimal32 format, 7 digits, and a rounding mode of [HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), the IEEE 754R default.

### DECIMAL64

public static final [MathContext](http://docs.google.com/java/math/MathContext.html) **DECIMAL64**

A MathContext object with a precision setting matching the IEEE 754R Decimal64 format, 16 digits, and a rounding mode of [HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), the IEEE 754R default.

### DECIMAL128

public static final [MathContext](http://docs.google.com/java/math/MathContext.html) **DECIMAL128**

A MathContext object with a precision setting matching the IEEE 754R Decimal128 format, 34 digits, and a rounding mode of [HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), the IEEE 754R default.

| **Constructor Detail** |
| --- |

### MathContext

public **MathContext**(int setPrecision)

Constructs a new MathContext with the specified precision and the [HALF\_UP](http://docs.google.com/java/math/RoundingMode.html#HALF_UP) rounding mode.

**Parameters:**setPrecision - The non-negative int precision setting. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the setPrecision parameter is less than zero.

### MathContext

public **MathContext**(int setPrecision,  
 [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) setRoundingMode)

Constructs a new MathContext with a specified precision and rounding mode.

**Parameters:**setPrecision - The non-negative int precision setting.setRoundingMode - The rounding mode to use. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the setPrecision parameter is less than zero. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the rounding mode argument is null

### MathContext

public **MathContext**([String](http://docs.google.com/java/lang/String.html) val)

Constructs a new MathContext from a string. The string must be in the same format as that produced by the [toString()](http://docs.google.com/java/math/MathContext.html#toString()) method.

An IllegalArgumentException is thrown if the precision section of the string is out of range (< 0) or the string is not in the format created by the [toString()](http://docs.google.com/java/math/MathContext.html#toString()) method.

**Parameters:**val - The string to be parsed **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the precision section is out of range or of incorrect format [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the argument is null

| **Method Detail** |
| --- |

### getPrecision

public int **getPrecision**()

Returns the precision setting. This value is always non-negative.

**Returns:**an int which is the value of the precision setting

### getRoundingMode

public [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) **getRoundingMode**()

Returns the roundingMode setting. This will be one of [RoundingMode.CEILING](http://docs.google.com/java/math/RoundingMode.html#CEILING), [RoundingMode.DOWN](http://docs.google.com/java/math/RoundingMode.html#DOWN), [RoundingMode.FLOOR](http://docs.google.com/java/math/RoundingMode.html#FLOOR), [RoundingMode.HALF\_DOWN](http://docs.google.com/java/math/RoundingMode.html#HALF_DOWN), [RoundingMode.HALF\_EVEN](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN), [RoundingMode.HALF\_UP](http://docs.google.com/java/math/RoundingMode.html#HALF_UP), [RoundingMode.UNNECESSARY](http://docs.google.com/java/math/RoundingMode.html#UNNECESSARY), or [RoundingMode.UP](http://docs.google.com/java/math/RoundingMode.html#UP).

**Returns:**a RoundingMode object which is the value of the roundingMode setting

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) x)

Compares this MathContext with the specified Object for equality.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**x - Object to which this MathContext is to be compared. **Returns:**true if and only if the specified Object is a MathContext object which has exactly the same settings as this object**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns the hash code for this MathContext.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**hash code for this MathContext**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns the string representation of this MathContext. The String returned represents the settings of the MathContext object as two space-delimited words (separated by a single space character, '\u0020', and with no leading or trailing white space), as follows:

1. The string "precision=", immediately followed by the value of the precision setting as a numeric string as if generated by the [Integer.toString](http://docs.google.com/java/lang/Integer.html#toString(int)) method.
2. The string "roundingMode=", immediately followed by the value of the roundingMode setting as a word. This word will be the same as the name of the corresponding public constant in the [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) enum.

For example:

precision=9 roundingMode=HALF\_UP

Additional words may be appended to the result of toString in the future if more properties are added to this class.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a String representing the context settings

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MathContext.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/math/BigInteger.html)   [**NEXT CLASS**](http://docs.google.com/java/math/RoundingMode.html) | [**FRAMES**](http://docs.google.com/index.html?java/math/MathContext.html)    [**NO FRAMES**](http://docs.google.com/MathContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#44sinio) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).