java.math

Class MathContext

java.lang.Object
    java.math.MathContext

All Implemented Interfaces:

Serializable

```
public final class MathContext
extends Object
implements Serializable
```

Immutable objects which encapsulate the context settings which describe certain rules for numerical operators, such as those implemented by the `BigDecimal` 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` object which specifies the algorithm to be used for rounding.

Since:

1.5

See Also:

`BigDecimal`, `RoundingMode`, Serialized Form

Field Summary

Fields

Modifier and Type	Field and Description
`static MathContext`	**`DECIMAL128`** A `MathContext` object with a precision setting matching the IEEE 754R Decimal128 format, 34 digits, and a rounding mode of **`HALF_EVEN`**, the IEEE 754R default.
`static MathContext`	**`DECIMAL32`** A `MathContext` object with a precision setting matching the IEEE 754R Decimal32 format, 7 digits, and a rounding mode of **`HALF_EVEN`**, the IEEE 754R default.
`static MathContext`	**`DECIMAL64`** A `MathContext` object with a precision setting matching the IEEE 754R Decimal64 format, 16 digits, and a rounding mode of **`HALF_EVEN`**, the IEEE 754R default.
`static MathContext`	**`UNLIMITED`** A `MathContext` object whose settings have the values required for unlimited precision arithmetic.

Constructor Summary

Constructors

Constructor and Description
`MathContext(int setPrecision)` Constructs a new `MathContext` with the specified precision and the `HALF_UP` rounding mode.
`MathContext(int setPrecision, RoundingMode setRoundingMode)` Constructs a new `MathContext` with a specified precision and rounding mode.
`MathContext(String val)` Constructs a new `MathContext` from a string.

Method Summary

Methods

Modifier and Type	Method and Description
`boolean`	`equals(Object x)` Compares this `MathContext` with the specified `Object` for equality.
`int`	`getPrecision()` Returns the `precision` setting.
`RoundingMode`	`getRoundingMode()` Returns the roundingMode setting.
`int`	`hashCode()` Returns the hash code for this `MathContext`.
`String`	`toString()` Returns the string representation of this `MathContext`.

Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Field Detail

UNLIMITED

```
public static final MathContext 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 DECIMAL32
```

A `MathContext` object with a precision setting matching the IEEE 754R Decimal32 format, 7 digits, and a rounding mode of `HALF_EVEN`, the IEEE 754R default.

DECIMAL64

```
public static final MathContext DECIMAL64
```

A `MathContext` object with a precision setting matching the IEEE 754R Decimal64 format, 16 digits, and a rounding mode of `HALF_EVEN`, the IEEE 754R default.

DECIMAL128

```
public static final MathContext DECIMAL128
```

A MathContext object with a precision setting matching the IEEE 754R Decimal128 format, 34 digits, and a rounding mode of 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 rounding mode.

Parameters:

setPrecision - The non-negative int precision setting.

Throws:

IllegalArgumentException - if the setPrecision parameter is less than zero.

MathContext

```
public MathContext(int setPrecision,
           RoundingMode 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 - if the setPrecision parameter is less than zero.

NullPointerException - if the rounding mode argument is null

MathContext

```
public MathContext(String val)
```

Constructs a new MathContext from a string. The string must be in the same format as that produced by the 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() method.

Parameters:

val - The string to be parsed

Throws:

IllegalArgumentException - if the precision section is out of range or of incorrect format

NullPointerException - 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 getRoundingMode()
```

Returns the roundingMode setting. This will be one of `RoundingMode.CEILING`, `RoundingMode.DOWN`, `RoundingMode.FLOOR`, `RoundingMode.HALF_DOWN`, `RoundingMode.HALF_EVEN`, `RoundingMode.HALF_UP`, `RoundingMode.UNNECESSARY`, or `RoundingMode.UP`.

Returns:

a `RoundingMode` object which is the value of the `roundingMode` setting

equals

```
public boolean equals(Object x)
```

Compares this `MathContext` with the specified `Object` for equality.

Overrides:

`equals` in class `Object`

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()`, `HashMap`

hashCode

```
public int hashCode()
```

Returns the hash code for this `MathContext`.

Overrides:

`hashCode` in class `Object`

Returns:

hash code for this `MathContext`

See Also:

`Object.equals(java.lang.Object)`, `System.identityHashCode(java.lang.Object)`

toString

```
public String 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` 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` 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` in class `Object`

Returns:

a `String` representing the context settings

Overview Package Class Use Tree Deprecated Index Help

Java™ Platform
Standard Ed. 7

Prev Class **Next Class** Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Submit a bug or feature

For further API reference and developer documentation, see Java SE Documentation. That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.