| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RoundingMode.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/MathContext.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?java/math/RoundingMode.html)    [**NO FRAMES**](http://docs.google.com/RoundingMode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [ENUM CONSTANTS](#3znysh7) | FIELD | [METHOD](#2et92p0) | DETAIL: [ENUM CONSTANTS](#1t3h5sf) | FIELD | [METHOD](#44sinio) |

## **java.math**

Enum RoundingMode

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.lang.Enum](http://docs.google.com/java/lang/Enum.html)<[RoundingMode](http://docs.google.com/java/math/RoundingMode.html)>  
 **java.math.RoundingMode**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[RoundingMode](http://docs.google.com/java/math/RoundingMode.html)>

public enum **RoundingMode**extends [Enum](http://docs.google.com/java/lang/Enum.html)<[RoundingMode](http://docs.google.com/java/math/RoundingMode.html)>

Specifies a *rounding behavior* for numerical operations capable of discarding precision. Each rounding mode indicates how the least significant returned digit of a rounded result is to be calculated. If fewer digits are returned than the digits needed to represent the exact numerical result, the discarded digits will be referred to as the *discarded fraction* regardless the digits' contribution to the value of the number. In other words, considered as a numerical value, the discarded fraction could have an absolute value greater than one.

Each rounding mode description includes a table listing how different two-digit decimal values would round to a one digit decimal value under the rounding mode in question. The result column in the tables could be gotten by creating a BigDecimal number with the specified value, forming a [MathContext](http://docs.google.com/java/math/MathContext.html) object with the proper settings (precision set to 1, and the roundingMode set to the rounding mode in question), and calling [round](http://docs.google.com/java/math/BigDecimal.html#round(java.math.MathContext)) on this number with the proper MathContext. A summary table showing the results of these rounding operations for all rounding modes appears below.

Summary of Rounding Operations Under Different Rounding Modes

|  | Result of rounding input to one digit with the given rounding mode | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input Number | UP | DOWN | CEILING | FLOOR | HALF\_UP | HALF\_DOWN | HALF\_EVEN | UNNECESSARY |
| 5.5 | 6 | 5 | 6 | 5 | 6 | 5 | 6 | throw ArithmeticException |
| 2.5 | 3 | 2 | 3 | 2 | 3 | 2 | 2 | throw ArithmeticException |
| 1.6 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | throw ArithmeticException |
| 1.1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | throw ArithmeticException |
| 1.0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| -1.0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| -1.1 | -2 | -1 | -1 | -2 | -1 | -1 | -1 | throw ArithmeticException |
| -1.6 | -2 | -1 | -1 | -2 | -2 | -2 | -2 | throw ArithmeticException |
| -2.5 | -3 | -2 | -2 | -3 | -3 | -2 | -2 | throw ArithmeticException |
| -5.5 | -6 | -5 | -5 | -6 | -6 | -5 | -6 | throw ArithmeticException |

This enum is intended to replace the integer-based enumeration of rounding mode constants in [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) ([BigDecimal.ROUND\_UP](http://docs.google.com/java/math/BigDecimal.html#ROUND_UP), [BigDecimal.ROUND\_DOWN](http://docs.google.com/java/math/BigDecimal.html#ROUND_DOWN), etc. ).

**Since:** 1.5 **See Also:**[BigDecimal](http://docs.google.com/java/math/BigDecimal.html), [MathContext](http://docs.google.com/java/math/MathContext.html)

| **Enum Constant Summary** | |
| --- | --- |
| [**CEILING**](http://docs.google.com/java/math/RoundingMode.html#CEILING)            Rounding mode to round towards positive infinity. |
| [**DOWN**](http://docs.google.com/java/math/RoundingMode.html#DOWN)            Rounding mode to round towards zero. |
| [**FLOOR**](http://docs.google.com/java/math/RoundingMode.html#FLOOR)            Rounding mode to round towards negative infinity. |
| [**HALF\_DOWN**](http://docs.google.com/java/math/RoundingMode.html#HALF_DOWN)            Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round down. |
| [**HALF\_EVEN**](http://docs.google.com/java/math/RoundingMode.html#HALF_EVEN)            Rounding mode to round towards the "nearest neighbor" unless both neighbors are equidistant, in which case, round towards the even neighbor. |
| [**HALF\_UP**](http://docs.google.com/java/math/RoundingMode.html#HALF_UP)            Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round up. |
| [**UNNECESSARY**](http://docs.google.com/java/math/RoundingMode.html#UNNECESSARY)            Rounding mode to assert that the requested operation has an exact result, hence no rounding is necessary. |
| [**UP**](http://docs.google.com/java/math/RoundingMode.html#UP)            Rounding mode to round away from zero. |

| **Method Summary** | |
| --- | --- |
| static [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) | [**valueOf**](http://docs.google.com/java/math/RoundingMode.html#valueOf(int))(int rm)            Returns the RoundingMode object corresponding to a legacy integer rounding mode constant in [BigDecimal](http://docs.google.com/java/math/BigDecimal.html). |
| static [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) | [**valueOf**](http://docs.google.com/java/math/RoundingMode.html#valueOf(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns the enum constant of this type with the specified name. |
| static [RoundingMode](http://docs.google.com/java/math/RoundingMode.html)[] | [**values**](http://docs.google.com/java/math/RoundingMode.html#values())()            Returns an array containing the constants of this enum type, in the order they are declared. |

| **Methods inherited from class java.lang.**[**Enum**](http://docs.google.com/java/lang/Enum.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Enum.html#clone()), [compareTo](http://docs.google.com/java/lang/Enum.html#compareTo(E)), [equals](http://docs.google.com/java/lang/Enum.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Enum.html#finalize()), [getDeclaringClass](http://docs.google.com/java/lang/Enum.html#getDeclaringClass()), [hashCode](http://docs.google.com/java/lang/Enum.html#hashCode()), [name](http://docs.google.com/java/lang/Enum.html#name()), [ordinal](http://docs.google.com/java/lang/Enum.html#ordinal()), [toString](http://docs.google.com/java/lang/Enum.html#toString()), [valueOf](http://docs.google.com/java/lang/Enum.html#valueOf(java.lang.Class,%20java.lang.String)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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)) |

| **Enum Constant Detail** |
| --- |

### UP

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

Rounding mode to round away from zero. Always increments the digit prior to a non-zero discarded fraction. Note that this rounding mode never decreases the magnitude of the calculated value.

Example:

| Input Number | Input rounded to one digit  with UP rounding |
| --- | --- |
| 5.5 | 6 |
| 2.5 | 3 |
| 1.6 | 2 |
| 1.1 | 2 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -2 |
| -1.6 | -2 |
| -2.5 | -3 |
| -5.5 | -6 |

### DOWN

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

Rounding mode to round towards zero. Never increments the digit prior to a discarded fraction (i.e., truncates). Note that this rounding mode never increases the magnitude of the calculated value.

Example:

| Input Number | Input rounded to one digit  with DOWN rounding |
| --- | --- |
| 5.5 | 5 |
| 2.5 | 2 |
| 1.6 | 1 |
| 1.1 | 1 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -1 |
| -1.6 | -1 |
| -2.5 | -2 |
| -5.5 | -5 |

### CEILING

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

Rounding mode to round towards positive infinity. If the result is positive, behaves as for RoundingMode.UP; if negative, behaves as for RoundingMode.DOWN. Note that this rounding mode never decreases the calculated value.

Example:

| Input Number | Input rounded to one digit  with CEILING rounding |
| --- | --- |
| 5.5 | 6 |
| 2.5 | 3 |
| 1.6 | 2 |
| 1.1 | 2 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -1 |
| -1.6 | -1 |
| -2.5 | -2 |
| -5.5 | -5 |

### FLOOR

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

Rounding mode to round towards negative infinity. If the result is positive, behave as for RoundingMode.DOWN; if negative, behave as for RoundingMode.UP. Note that this rounding mode never increases the calculated value.

Example:

| Input Number | Input rounded to one digit  with FLOOR rounding |
| --- | --- |
| 5.5 | 5 |
| 2.5 | 2 |
| 1.6 | 1 |
| 1.1 | 1 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -2 |
| -1.6 | -2 |
| -2.5 | -3 |
| -5.5 | -6 |

### HALF\_UP

public static final [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) **HALF\_UP**

Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round up. Behaves as for RoundingMode.UP if the discarded fraction is >= 0.5; otherwise, behaves as for RoundingMode.DOWN. Note that this is the rounding mode commonly taught at school.

Example:

| Input Number | Input rounded to one digit  with HALF\_UP rounding |
| --- | --- |
| 5.5 | 6 |
| 2.5 | 3 |
| 1.6 | 2 |
| 1.1 | 1 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -1 |
| -1.6 | -2 |
| -2.5 | -3 |
| -5.5 | -6 |

### HALF\_DOWN

public static final [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) **HALF\_DOWN**

Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round down. Behaves as for RoundingMode.UP if the discarded fraction is > 0.5; otherwise, behaves as for RoundingMode.DOWN.

Example:

| Input Number | Input rounded to one digit  with HALF\_DOWN rounding |
| --- | --- |
| 5.5 | 5 |
| 2.5 | 2 |
| 1.6 | 2 |
| 1.1 | 1 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -1 |
| -1.6 | -2 |
| -2.5 | -2 |
| -5.5 | -5 |

### HALF\_EVEN

public static final [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) **HALF\_EVEN**

Rounding mode to round towards the "nearest neighbor" unless both neighbors are equidistant, in which case, round towards the even neighbor. Behaves as for RoundingMode.HALF\_UP if the digit to the left of the discarded fraction is odd; behaves as for RoundingMode.HALF\_DOWN if it's even. Note that this is the rounding mode that statistically minimizes cumulative error when applied repeatedly over a sequence of calculations. It is sometimes known as "Banker's rounding," and is chiefly used in the USA. This rounding mode is analogous to the rounding policy used for float and double arithmetic in Java.

Example:

| Input Number | Input rounded to one digit  with HALF\_EVEN rounding |
| --- | --- |
| 5.5 | 6 |
| 2.5 | 2 |
| 1.6 | 2 |
| 1.1 | 1 |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | -1 |
| -1.6 | -2 |
| -2.5 | -2 |
| -5.5 | -6 |

### UNNECESSARY

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

Rounding mode to assert that the requested operation has an exact result, hence no rounding is necessary. If this rounding mode is specified on an operation that yields an inexact result, an ArithmeticException is thrown.

Example:

| Input Number | Input rounded to one digit  with UNNECESSARY rounding |
| --- | --- |
| 5.5 | throw ArithmeticException |
| 2.5 | throw ArithmeticException |
| 1.6 | throw ArithmeticException |
| 1.1 | throw ArithmeticException |
| 1.0 | 1 |
| -1.0 | -1 |
| -1.1 | throw ArithmeticException |
| -1.6 | throw ArithmeticException |
| -2.5 | throw ArithmeticException |
| -5.5 | throw ArithmeticException |

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

### values

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

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

for (RoundingMode c : RoundingMode.values())  
  System.out.println(c);

**Returns:**an array containing the constants of this enum type, in the order they are declared

### valueOf

public static [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) **valueOf**([String](http://docs.google.com/java/lang/String.html) name)

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**name - the name of the enum constant to be returned. **Returns:**the enum constant with the specified name **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if this enum type has no constant with the specified name [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the argument is null

### valueOf

public static [RoundingMode](http://docs.google.com/java/math/RoundingMode.html) **valueOf**(int rm)

Returns the RoundingMode object corresponding to a legacy integer rounding mode constant in [BigDecimal](http://docs.google.com/java/math/BigDecimal.html).

**Parameters:**rm - legacy integer rounding mode to convert **Returns:**RoundingMode corresponding to the given integer. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - integer is out of range

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RoundingMode.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/MathContext.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?java/math/RoundingMode.html)    [**NO FRAMES**](http://docs.google.com/RoundingMode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [ENUM CONSTANTS](#3znysh7) | FIELD | [METHOD](#2et92p0) | DETAIL: [ENUM CONSTANTS](#1t3h5sf) | FIELD | [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).