

Introduction

The **java.lang.Double** class wraps a value of the primitive type double in an object. An object of type Double contains a single field whose type is double.

Class declaration

Following is the declaration for **java.lang.Double** class:

```
public final class Double
    extends Number
        implements Comparable<Double>
```

Field

Following are the fields for **java.lang.Double** class:

- **static int MAX_EXPONENT** -- This is the maximum exponent a finite double variable may have.
- **static double MAX_VALUE** -- This is the constant holding the largest positive finite value of type double, $(2-2^{-52}) \times 2^{1023}$.
- **static int MIN_EXPONENT** -- This is the minimum exponent a normalized double variable may have.
- **static double MIN_NORMAL** -- This is the constant holding the smallest positive normal value of type double, 2^{-1022} .
- **static double MIN_VALUE** -- This is the constant holding the smallest positive nonzero value of type double, 2^{-1074} .
- **static double NaN** -- This is the constant holding a Not-a-Number *NaN* value of type double.
- **static double NEGATIVE_INFINITY** -- This is the constant holding the negative infinity of type double.
- **static double POSITIVE_INFINITY** -- This is the constant holding the positive infinity of type double.
- **static int SIZE** -- This is the number of bits used to represent a double value.
- **static Class<Double> TYPE** -- This is the class instance representing the primitive type double

Class constructors

S.N. Constructor & Description

- | | |
|---|---|
| 1 | Double(double value)

This constructs a newly allocated Double object that represents the primitive double argument. |
| 2 | Double(String s) |

This constructs a newly allocated Double object that represents the floating-point value of type double represented by the string.

Class methods

S.N.	Method & Description
1	<u>byte byteValue</u> This method returns the value of this Double as a byte <i>broadcastingtoabyte</i> .
2	<u>static int comparedouble1, double2</u> This method compares the two specified double values.
3	<u>int compareToDoubleanotherDouble</u> This method compares the two specified double values.
4	<u>static long doubleToLongBitsdoublevalue</u> This method returns a representation of the specified floating-point value according to the IEEE 754 floating-point "double format" bit layout.
5	<u>static long doubleToRawLongBitsdoublevalue</u> This method returns a representation of the specified floating-point value according to the IEEE 754 floating-point "double format" bit layout, preserving Not-a-Number <i>NaN</i> values.
6	<u>double doubleValue</u> This method returns a representation of the specified floating-point value according to the IEEE 754 floating-point "double format" bit layout, preserving Not-a-Number <i>NaN</i> values.
7	<u>boolean equalsObjectobj</u> This method compares this object against the specified object.
8	<u>float floatValue</u> This method returns the float value of this Double object.
9	<u>int hashCode</u> This method returns a hash code for this Double object.

10

[int intValue](#)

This method returns the value of this Double as an int *bycasting to type int*.

11

[boolean isInfinite](#)

This method returns true if this Double value is infinitely large in magnitude, false otherwise.

12

[static boolean isInfiniteDouble](#)

This method returns true if the specified number is infinitely large in magnitude, false otherwise.

13

[boolean isNaN](#)

This method returns true if this Double value is a Not-a-Number *NaN*, false otherwise.

14

[static boolean isNaNDouble](#)

This method returns true if the specified number is a Not-a-Number *NaN* value, false otherwise.

15

[static double longBitsToDoubleLongbits](#)

This method returns the double value corresponding to a given bit representation.

16

[long longValue](#)

This method returns the value of this Double as a long *bycasting to type long*.

17

[static double parseDoubleStrings](#)

This method returns a new double initialized to the value represented by the specified String, as performed by the `valueOf` method of class Double.

18

[short shortValue](#)

This method returns the value of this Double as a short *bycasting to a short*.

19

[static String toHexStringdouble](#)

This method returns a hexadecimal string representation of the double argument.

20

[String toString](#)

This method returns a string representation of this Double object.

21

[static String toString\(*doubled*\)](#)

This method returns a string representation of the double argument.

22

[static Double valueOf\(*doubled*\)](#)

This method returns a Double instance representing the specified double value.

23

[static Double valueOf\(*Strings*\)](#)

This method returns a Double object holding the double value represented by the argument string s.

Methods inherited

This class inherits methods from the following classes:

• [java.lang.Object](#)

Loading [MathJax]/jax/output/HTML-CSS/jax.js