

JAVA.LANG.STRICTMATH.ULP METHOD

http://www.tutorialspoint.com/java/lang/strictmath_ulp_double.htm

Copyright © tutorialspoint.com

Description

The **java.lang.StrictMath.ulp***doubled* method returns the size of an ulp of the argument. An ulp of a double value is the positive distance between this floating-point value and the double value next larger in magnitude. For non-NaN x, **ulp**-x == **ulp**x. It includes these cases:

- If the argument is NaN, then the result is NaN.
- If the argument is positive or negative infinity, then the result is positive infinity.
- If the argument is positive or negative zero, then the result is Double.MIN_VALUE.
- If the argument is $\pm$ Double.MAX_VALUE, then the result is equal to 2^{971} .

Declaration

Following is the declaration for **java.lang.StrictMath.ulp** method

```
public static double ulp(double d)
```

Parameters

- **d** -- This is the floating-point value whose ulp is to be returned.

Return Value

This method returns the size of an ulp of the argument.

Exception

- NA

Example

The following example shows the usage of java.lang.StrictMath.ulp method.

```
package com.tutorialspoint;

import java.lang.*;

public class StrictMathDemo {

    public static void main(String[] args) {

        double d1 = 5.5 , d2 = -2.9, d3 = 0.0;

        // returns the size of an ulp of the argument
        double ulpValue = StrictMath.ulp(d1);
        System.out.println("Size of ulp of " + d1 + " = " + ulpValue);

        ulpValue = StrictMath.ulp(d2);
        System.out.println("Size of ulp of " + d2 + " = " + ulpValue);

        ulpValue = StrictMath.ulp(d3);
        System.out.println("Size of ulp of " + d3 + " = " + ulpValue);
    }
}
```

Let us compile and run the above program, this will produce the following result:

```
Size of ulp of 5.5 = 8.881784197001252E-16
```

Size of ulp of -2.9 = $4.440892098500626E-16$

Size of ulp of 0.0 = $4.9E-324$

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