

JAVA.LANG.STRICTMATH.ULP METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.StrictMath.ulp(float f)** method returns the size of an ulp of the argument. An ulp of a float value is the positive distance between this floating-point value and the float value next larger in magnitude. Note that 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 Float.MIN_VALUE.
- If the argument is $\pm$ Float.MAX_VALUE, then the result is equal to 2^{104} .

Declaration

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

```
public static float ulp(float f)
```

Parameters

- **f** -- 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) {

        float f1 = 3.71f , f2 = -4.3f, f3 = 0.0f;

        // returns the size of an ulp of the argument
        float ulpValue = StrictMath.ulp(f1);
        System.out.println("Size of ulp of " + f1 + " : " + ulpValue);

        ulpValue = StrictMath.ulp(f2);
        System.out.println("Size of ulp of " + f2 + " : " + ulpValue);

        ulpValue = StrictMath.ulp(f3);
        System.out.println("Size of ulp of " + f3 + " : " + ulpValue);
    }
}
```

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

```
Size of ulp of 3.71 = 2.3841858E-7
```

Size of ulp of $-4.3 = 4.7683716E-7$

Size of ulp of $0.0 = 1.4E-45$

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