

JAVA.LANG.STRICTMATH.NEXTAFTER METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.StrictMath.nextAfter***float**start, doubledirection* method returns the floating-point number adjacent to the first argument in the direction of the second argument. If both arguments compare as equal a value equivalent to the second argument is returned. It includes these cases:

- If either argument is a NaN, then NaN is returned.
- If both arguments are signed zeros, *direction* is returned unchanged.
- If start is \pm Float.MIN_VALUE and direction has a value such that the result should have a smaller magnitude, then a zero with the same sign as start is returned.
- If start is infinite and direction has a value such that the result should have a smaller magnitude, Float.MAX_VALUE with the same sign as start is returned.
- If start is equal to \pm Float.MAX_VALUE and direction has a value such that the result should have a larger magnitude, an infinity with same sign as start is returned.

Declaration

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

```
public static float nextAfter(float start, double direction)
```

Parameters

- **start** -- This is the starting floating-point value
- **direction** -- This is the value indicating which of start's neighbors or start should be returned

Return Value

This method returns the floating-point number adjacent to start in the direction of direction.

Exception

- **NA**

Example

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

```
package com.tutorialspoint;

import java.lang.*;

public class StrictMathDemo {

    public static void main(String[] args) {

        float f1 = 90.2f, f2 = 0.0f;

        /* returns the floating-point number adjacent to the first argument in the
        direction of the second argument */
        float retval = StrictMath.nextAfter(f1, 9.2d);
        System.out.println("NextAfter = " + retval);

        /* returns the floating-point number adjacent to the first argument in the
        direction of the second argument */
        retval = StrictMath.nextAfter(f2, 9.2d);
        System.out.println("NextAfter = " + retval);
    }
}
```

```
// returns 0 if both arguments is zero
retval = StrictMath.nextAfter(f2, 0.0d);
System.out.println("NextAfter = " + retval);
}
```

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

```
NextAfter = 90.19999
NextAfter = 1.4E-45
NextAfter = 0.0
```

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