

JAVA.LANG.INTEGER.NUMBEROFLEADINGZEROS METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.Integer.numberOfLeadingZeros** method returns the number of zero bits preceding the highest-order " *leftmost* " one-bit in the two's complement binary representation of the specified int value.

It returns 32 if the specified value has no one-bits in its two's complement representation, in other words if it is equal to zero.

Declaration

Following is the declaration for **java.lang.Integer.numberOfLeadingZeros** method

```
public static int numberOfLeadingZeros(int i)
```

Parameters

- **i** -- This is the int value.

Return Value

This method returns the number of zero bits preceding the highest-order " *leftmost* " one-bit in the two's complement binary representation of the specified int value, or 32 if the value is equal to zero.

Exception

- **NA**

Example

The following example shows the usage of java.lang.Integer.numberOfLeadingZeros method.

```
package com.tutorialspoint;

import java.lang.*;

public class IntegerDemo {

    public static void main(String[] args) {

        int i = 170;
        System.out.println("Number = " + i);

        /* returns the string representation of the unsigned integer value
        represented by the argument in binary (base 2) */
        System.out.println("Binary = " + Integer.toBinaryString(i));

        // returns the number of one-bits
        System.out.println("Number of one bits = " + Integer.bitCount(i));

        /* returns an int value with at most a single one-bit, in the position
        of the highest-order ("leftmost") one-bit in the specified int value */
        System.out.println("Highest one bit = " + Integer.highestOneBit(i));

        /* returns an int value with at most a single one-bit, in the position
        of the lowest-order ("rightmost") one-bit in the specified int value.*/
        System.out.println("Lowest one bit = " + Integer.lowestOneBit(i));

        /*returns the number of zero bits preceding the highest-order
        ("leftmost")one-bit */
    }
}
```

```
System.out.print("Number of leading zeros = ");  
System.out.println(Integer.numberOfLeadingZeros(i));  
}  
}
```

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

```
Number = 170  
Binary = 10101010  
Number of one bits = 4  
Highest one bit = 128  
Lowest one bit = 2  
Number of leading zeros = 24
```

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