

JAVA.UTIL.BITSET.CLEAR METHOD

http://www.tutorialspoint.com/java/util/bitset_clear_index.htm

Copyright © tutorialspoint.com

Description

The **java.util.BitSet.clear** method sets the bits from the specified *fromIndex* *inclusive* to the specified *toIndex* *exclusive* to false.

Declaration

Following is the declaration for **java.util.BitSet.clear** method

```
public void clear(int fromIndex, int toIndex)
```

Parameters

- **fromIndex** -- index of the first bit to be cleared
- **toIndex** -- index of the last bit to be cleared

Return Value

This method does not return a value.

Exception

- **IndexOutOfBoundsException** -- if *fromIndex* is negative, or *toIndex* is negative, or *fromIndex* is larger than *toIndex*.

Example

The following example shows the usage of **java.util.BitSet.clear** method.

```
package com.tutorialspoint;

import java.util.*;

public class BitSetDemo {

    public static void main(String[] args) {

        // create 2 bitsets
        BitSet bitset1 = new BitSet(8);
        BitSet bitset2 = new BitSet(8);

        // assign values to bitset1
        bitset1.set(0);
        bitset1.set(1);
        bitset1.set(2);
        bitset1.set(3);
        bitset1.set(4);
        bitset1.set(5);

        // assign values to bitset2
        bitset2.set(2);
        bitset2.set(4);
        bitset2.set(6);
        bitset2.set(8);
        bitset2.set(10);

        // print the sets
        System.out.println("Bitset1:" + bitset1);
        System.out.println("Bitset2:" + bitset2);
    }
}
```

```
// clear from index 2 to index 4 in bitset1
bitset1.clear(1, 4);

// clear index 2 to index 8 in bitset2
bitset2.clear(2, 8);

// print new bitsets
System.out.println(" " + bitset1);
System.out.println(" " + bitset2);

}
}
```

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

```
Bitset1:{0, 1, 2, 3, 4, 5}
Bitset2:{2, 4, 6, 8, 10}
{0, 4, 5}
{8, 10}
```

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