

JAVA.UTIL.TREEMAP.REMOVE METHOD

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

Copyright © tutorialspoint.com

Description

The **removeObjectkey** method is used to remove the mapping for this key from this TreeMap if present.

Declaration

Following is the declaration for **java.util.TreeMap.remove** method.

```
public V remove(Object key)
```

Parameters

- **key** -- This is the key for which mapping should be removed.

Return Value

The method call returns the previous value associated with key, or null if there was no mapping for key.

Exception

- **ClassCastException** -- This exception is thrown if the specified key cannot be compared with the keys currently in the map.
- **NullPointerException** -- This exception is thrown if the specified key is null and this map uses natural ordering, or its comparator does not permit null keys.

Example

The following example shows the usage of java.util.TreeMap.remove

```
package com.tutorialspoint;

import java.util.*;

public class TreeMapDemo {
    public static void main(String[] args) {
        // creating tree map
        TreeMap<Integer, String> treemap = new TreeMap<Integer, String>();

        // populating tree map
        treemap.put(2, "two");
        treemap.put(1, "one");
        treemap.put(3, "three");
        treemap.put(6, "six");
        treemap.put(5, "five");

        System.out.println("Value before modification: "+ treemap);

        // removing value at key 5
        System.out.println("Removed value: "+treemap.remove(5));

        System.out.println("Value after modification: "+ treemap);
    }
}
```

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

```
Value before modification: {1=one, 2=two, 3=three, 5=five, 6=six}
```

Removed value: five

Value after modification: {1=one, 2=two, 3=three, 6=six}

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