

JAVA.UTIL.TREEMAP.HIGHERKEY METHOD

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

Copyright © tutorialspoint.com

Description

The **higherKey** method is used to return the least key strictly greater than the given key, or null if there is no such key.

Declaration

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

```
public K higherKey(K key)
```

Parameters

- **key** -- This is the key to be matched.

Return Value

The method call returns the least key greater than key, or null if there is no such 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.higherKey method.

```
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");

        // getting higher key for key 4
        System.out.println("Checking values of the map");
        System.out.println("Value is: " + treemap.higherKey(3));
    }
}
```

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

```
Checking values of the map
Value is: 5
```

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