

JAVA.UTIL.TREEMAP.CEILINGKEY METHOD

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

Copyright © tutorialspoint.com

Description

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

Declaration

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

```
public K ceilingKey(K key)
```

Parameters

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

Return Value

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

Exception

- **ClassCastException** -- It throws the exception if the specified key cannot be compared with the keys currently in the map.
- **NullPointerException** -- It throws the exception 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.ceilingKey** method.

```
package com.tutorialspoint;

import java.util.*;

public class TreeMapDemo {
    public static void main(String[] args) {
        // creating tree map
        NavigableMap<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("Ceiling key entry for 4: "+ treemap.ceilingKey(4));
        System.out.println("Ceiling key entry for 5: "+ treemap.ceilingKey(5));
        System.out.println("Ceiling key entry for 7: "+ treemap.ceilingKey(7));
    }
}
```

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

```
Ceiling key entry for 4: 5
Ceiling key entry for 5: 5
Ceiling key entry for 7: null
```

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