

JAVA.UTIL.TREESET.LOWER METHOD

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

Copyright © tutorialspoint.com

Description

The **lower** method is used to return the greatest element in this set strictly less than the given element, or null if there is no such element.

Declaration

Following is the declaration for **java.util.TreeSet.lower** method.

```
public E lower(E e)
```

Parameters

- **e** -- This is the value to match.

Return Value

The method call returns the greatest element less than e, or null if there is no such element.

Exception

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

Example

The following example shows the usage of java.util.TreeSet.lower method.

```
package com.tutorialspoint;

import java.util.TreeSet;

public class TreeSetDemo {
    public static void main(String[] args) {
        // creating a TreeSet
        TreeSet<Integer>treeadd = new TreeSet<Integer>();

        // adding in the tree set
        treeadd.add(1);
        treeadd.add(13);
        treeadd.add(17);
        treeadd.add(2);

        // displaying the greatest element < 17
        System.out.println("Greatest element less than 17 is: "+treeadd.lower(17));
    }
}
```

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

```
Greatest element less than 17 is: 13
```

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