JAVA.UTIL.TREESET.HEADSET METHOD

Description

The **headSet*EtoElement, booleaninclusive*** method is used to return a view of the portion of this set whose elements are less than *orequalto, ifinclusiveistrue* toElement. The returned set is backed by this set, so changes in the returned set are reflected in this set, and vice-versa.

Declaration

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

```
public NavigableSet<E> headSet(E toElement,boolean inclusive)
```

Parameters

- **toElement** -- This is the high endpoint of the returned set.
- **inclusive** -- This is true if the high endpoint is to be included in the returned view.

Return Value

The method call returns a view of the portion of this set whose elements are less than *orequalto, ifinclusiveistrue* toElement.

Exception

- **ClassCastException** -- This is thrown if toElement is not compatible with this set's comparator.
- **NullPointerException** -- This is thrown if toElement is null and this set uses natural ordering, or its comparator does not permit null elements.
- **IllegalArgumentException** -- This is thrown if this set itself has a restricted range, and toElement lies outside the bounds of the range.

Example

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

```
package com.tutorialspoint;

import java.util.Iterator;
import java.util.TreeSet;

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

   // adding in the tree
   tree.add(12);
   tree.add(13);
   tree.add(14);
   tree.add(15);
   tree.add(16);
   tree.add(17);

   // getting values for 15 inclusive true
   treeheadsetincl = (TreeSet)tree.headSet(15, true);

   // creating iterator
```

```
   Iterator iterator;
   iterator = treeheadsetincl.iterator();

   //Displaying the tree set data
   System.out.println("Tree set data for '15' inclusive TRUE: ");
   while (iterator.hasNext()){
   System.out.println(iterator.next() + " ");
   }

   // getting values for 15 inclusive false
   treeheadsetincl = (TreeSet)tree.headSet(15, false);

   // creating iterator
   iterator = treeheadsetincl.iterator();

   //Displaying the tree set data
   System.out.println("Tree set data for '15' inclusive FALSE: ");
   while (iterator.hasNext()){
   System.out.println(iterator.next() + " ");
   }
   }
}
```

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

```
Tree set data for '15' inclusive TRUE:
12
13
14
15
Tree set data for '15' inclusive FALSE:
12
13
14
```