TreeSet class in Java with Examples

java.util.TreeSet is implementation class of SortedSet Interface. TreeSet has following important properties.

1. TreeSet implements the SortedSet interface so duplicate values are not allowed.
2. TreeSet does not preserve the insertion order of elements but elements are sorted by keys.
3. TreeSet does not allow to insert Heterogeneous objects. It will throw classCastException at Runtime if trying to add hetrogeneous objects.
4. TreeSet allows to insert null but just once.
5. TreeSet is basically implementation of a self-balancing binary search tree like Red-Black Tree. Therefore operations like add, remove and search take O(Log n) time. And operations like printing n elements in sorted order takes O(n) time.

Constructors:

Following are the four constructors in TreeSet class.

1. **TreeSet t = new TreeSet();**
   This will create empty TreeSet object in which elements will get stored in default natural sorting order.
2. **TreeSet t = new TreeSet(Comparator comp);**
   This constructor is used when you externally wants to specify sorting order of elements getting stored.
3. **TreeSet t = new TreeSet(Collection col);**
   This constructor is used when we want to convert any Collection object to TreeSet object.
4. **TreeSet t = new TreeSet(SortedSet s)**;
   This constructor is used to convert SortedSet object to TreeSet Object.

Synchronized TreeSet:

Implementation of TreeSet class is not synchronized. If there is need of synchronized version of TreeSet, it can be done externally using Collections.synchronizedSet() method.

```
TreeSet ts = new TreeSet();
Set syncSet = Collections.synchronziedSet(ts);
```

Adding (or inserting) Elements to TreeSet:

TreeSet supports add() method to insert elements to it.

```
// Java program to demonstrate insertions in TreeSet
import java.util.*;

class TreeSetDemo
{
    public static void main (String[] args)
    {
        TreeSet ts1= new TreeSet();
        ts1.add("A");
        ts1.add("B");
        ts1.add("C");

        // Duplicates will not get insert
        ts1.add("C");

        // Elements get stored in default natural
        // Sorting Order(Ascending)
        System.out.println(ts1);  // [A,B,C]

        // ts1.add(2) ; will throw ClassCastException
        //              at run time
    }
}
```

Run on IDE

Output :

```
[A, B, C]
```

Null Insertion:

If we insert null in non empty TreeSet, it throws NullPointerException because while inserting null it will get compared to existing elements and null can not be compared to any value.

```
// Java program to demonstrate null insertion
// in TreeSet
import java.util.*;

class TreeSetDemo
{
    public static void main (String[] args)
    {
        TreeSet ts2= new TreeSet();
        ts2.add("A");
        ts2.add("B");
        ts2.add("C");
        ts2.add(null); // Throws NullPointerException
    }
}
```

Run on IDE

Output :

```
Exception in thread "main" java.lang.NullPointerException
        at java.util.TreeMap.put(TreeMap.java:563)
        at java.util.TreeSet.add(TreeSet.java:255)
        at TreeSetDemo.main(File.java:13)
```

We can insert first value as null, but if we insert any more value in TreeSet, it will also throw NullPointerException.

```
// Java program to demonstrate insertion
// in a TreeSet with null
import java.util.*;

class TreeSetDemo
{
    public static void main (String[] args)
    {
        TreeSet ts3 = new TreeSet();
        ts3.add(null);
        ts3.add("A");  // Throws NullPointerException
    }
}
```

Run on IDE

Output :

```
Exception in thread "main" java.lang.NullPointerException
        at java.util.TreeMap.compare(TreeMap.java:1294)
        at java.util.TreeMap.put(TreeMap.java:538)
        at java.util.TreeSet.add(TreeSet.java:255)
        at TreeSetDemo.main(File.java:10)
```

Methods:

TreeSet implements SortedSet so it has availability of all methods in Collection, Set and SortedSet interfaces. Following are the methods in Treeset interface.

1. **void add(Object o):** This method will add specified element according to some sorting order in TreeSet. Duplicate entires will not get added.
2. **boolean addAll(Collection c):** This method will add all elements of specified Collection to the set. Elements in Collection should be homogeneous otherwise ClassCastException will be thrown.Duplicate Entries of Collection will not be added to TreeSet.

```
// Java program to demonstrate TreeSet creation from
// ArrayList
import java.util.*;

class TreeSetDemo
{
```

```
    public static void main (String[] args)
    {
        ArrayList al = new ArrayList();
        al.add("GeeksforGeeks");
        al.add("GeeksQuiz");
        al.add("Practice");
        al.add("Compiler");
        al.add("Compiler"); //will not be added

        // Creating a TreeSet object from ArrayList
        TreeSet ts4 = new TreeSet(al);

        // [Compiler,GeeksQuiz,GeeksforGeeks,Practice]
        System.out.println(ts4);
    }
}
```

Run on IDE

Output :

```
[Compiler, GeeksQuiz, GeeksforGeeks, Practice]
```

3. **void clear() :** This method will remove all the elements.
4. **Comparator comparator():** This method will return Comparator used to sort elements in TreeSet or it will return null if default natural sorting order is used.
5. **boolean contains(Object o):** This method will return true if given element is present in TreeSet else it will return false.
6. **Object first() :** This method will return first element in TreeSet if TreeSet is not null else it will throw NoSuchElementException.
7. **Object last():** This method will return last element in TreeSet if TreeSet is not null else it will throw NoSuchElementException.
8. **SortedSet headSet(Object toElement):** This method will return elements of TreeSet which are less than the specified element.
9. **SortedSet tailSet(Object fromElement):** This method will return elements of TreeSet which are greater than or equal to the specified element.
10. **SortedSet subSet(Object fromElement, Object toElement):** This method will return elements ranging from fromElement to toElement. fromElement is inclusive and toElement is exclusive.

```
// Java program to demonstrate TreeSet creation from
// ArrayList
import java.util.*;

class TreeSetDemo
{
```

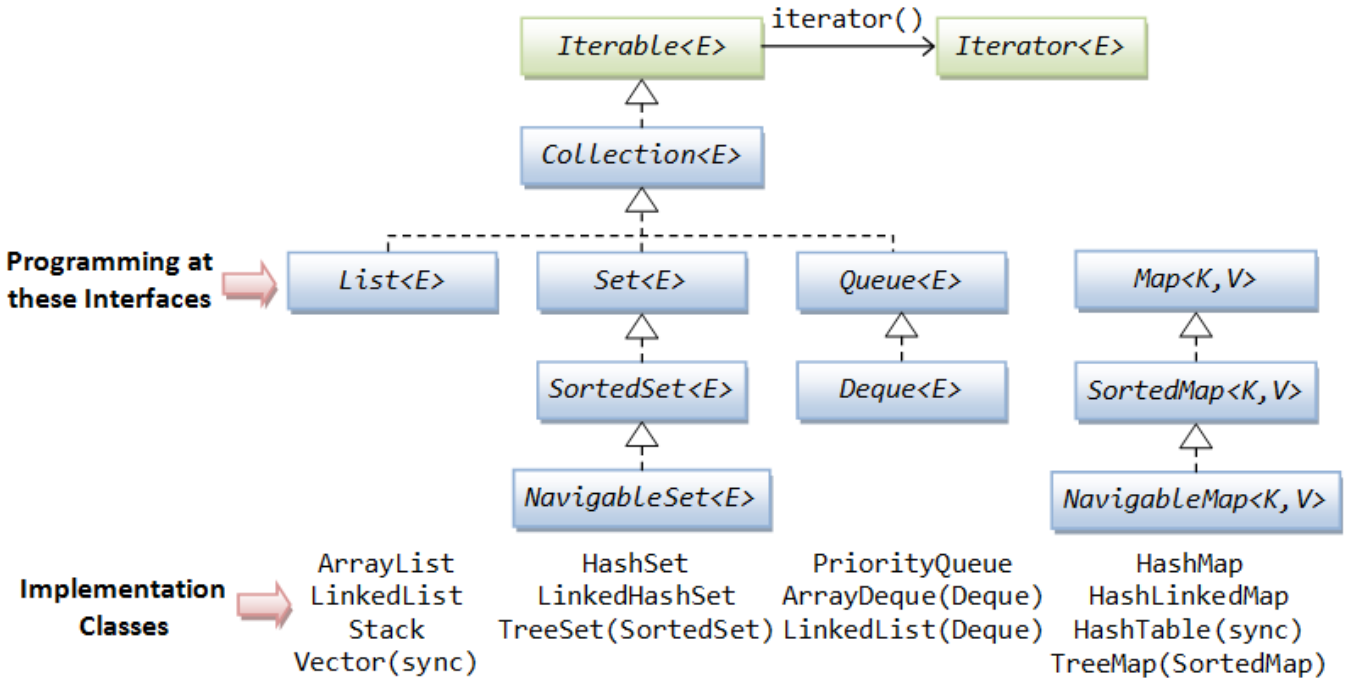
```
    public static void main (String[] args)
    {

        TreeSet ts5 = new TreeSet();

        // Uncommenting below  throws NoSuchElementException
        // System.out.println(ts5.first());

        // Uncommenting below throws NoSuchElementException
        // System.out.println(ts5.last());

        ts5.add("GeeksforGeeks");
        ts5.add("Compiler");
        ts5.add("practice");

        System.out.println(ts5.first()); // Compiler
        System.out.println(ts5.last()); //Practice

        // Elements less than O. It prints
        // [Compiler,GeeksforGeeks]
        System.out.println(ts5.headSet("O"));

        // Elements greater than or equal to G.
        // It prints [GeeksforGeeks, Practice]
        System.out.println(ts5.tailSet("G"));

        // Elements ranging from C to P
        // It prints [Compiler,GeeksforGeeks]
        System.out.println(ts5.subSet("C","P"));

        // Deletes all elements from ts5.
        ts5.clear();

        // Prints nothing
        System.out.println(ts5);
    }
}
```

Run on IDE

Output :

```
Compiler
practice
[Compiler, GeeksforGeeks]
[GeeksforGeeks, practice]
[Compiler, GeeksforGeeks]
[]
```

TreeSet in Collection Framework:



Link to the Image: https://www.ntu.edu.sg/home/ehchua/programming/java/J5c_Collection.html

This article is contributed by **Dharmesh Singh**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Java Java-Collections

Recommended Posts:

EnumSet class in Java with Example
ArrayList and LinkedList remove() methods in Java with Examples
HashSet in Java
Java.util.HashMap in Java
Map interface in Java with examples
Java.lang.String.compareTo()
Templates in C++ vs Generics in Java
Converting Text to Speech in Java
Difference between x++ and x=x+1 in Java