

Java TreeSet class

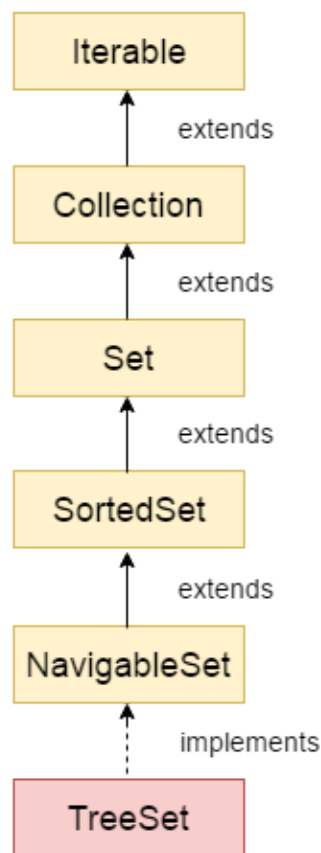
Java TreeSet class implements the Set interface that uses a tree for storage. It inherits AbstractSet class and implements NavigableSet interface. The objects of TreeSet class are stored in ascending order.

The important points about Java TreeSet class are:

- Contains unique elements only like HashSet.
- Access and retrieval times are quite fast.
- Maintains ascending order.

Hierarchy of TreeSet class

As shown in above diagram, Java TreeSet class implements NavigableSet interface. The NavigableSet interface extends SortedSet, Set, Collection and Iterable interfaces in hierarchical order.



TreeSet class declaration

Let's see the declaration for java.util.TreeSet class.

```
public class TreeSet<E> extends AbstractSet<E> implements NavigableSet<E>, Cloneable, Serializable
```

Constructors of Java TreeSet class

Constructor	Description
TreeSet()	It is used to construct an empty tree set that will be sorted in an ascending order according to the natural order of the tree set.
TreeSet(Collection c)	It is used to build a new tree set that contains the elements of the collection c.
TreeSet(Comparator comp)	It is used to construct an empty tree set that will be sorted according to given comparator.
TreeSet(SortedSet ss)	It is used to build a TreeSet that contains the elements of the given SortedSet.

Methods of Java TreeSet class

Method	Description
boolean addAll(Collection c)	It is used to add all of the elements in the specified collection to this set.
boolean contains(Object o)	It is used to return true if this set contains the specified element.
boolean isEmpty()	It is used to return true if this set contains no elements.
boolean remove(Object o)	It is used to remove the specified element from this set if it is present.
void add(Object o)	It is used to add the specified element to this set if it is not already present.
void clear()	It is used to remove all of the elements from this set.
Object clone()	It is used to return a shallow copy of this TreeSet instance.
Object first()	It is used to return the first (lowest) element currently in this sorted set.
Object last()	It is used to return the last (highest) element currently in this sorted set.
int size()	It is used to return the number of elements in this set.

Java TreeSet Example

```
import java.util.*;
class TestCollection11{
    public static void main(String args[]){
        //Creating and adding elements
        TreeSet<String> al=new TreeSet<String>();
        al.add("Ravi");
        al.add("Vijay");
```

```
al.add("Ravi");
al.add("Ajay");
//Traversing elements
Iterator<String> itr=al.iterator();
while(itr.hasNext()){
    System.out.println(itr.next());
}
}
```

Test it Now

Output:

```
Ajay
Ravi
Vijay
```

Java TreeSet Example: Book

Let's see a TreeSet example where we are adding books to set and printing all the books. The elements in TreeSet must be of Comparable type. String and Wrapper classes are Comparable by default. To add user-defined objects in TreeSet, you need to implement Comparable interface.

```
import java.util.*;
class Book implements Comparable<Book>{
    int id;
    String name,author,publisher;
    int quantity;
    public Book(int id, String name, String author, String publisher, int quantity) {
        this.id = id;
        this.name = name;
        this.author = author;
        this.publisher = publisher;
        this.quantity = quantity;
    }
    public int compareTo(Book b) {
        if(id>b.id){
            return 1;
        }else if(id<b.id){
```

```
        return -1;
    }else{
        return 0;
    }
}
}

public class TreeSetExample {
    public static void main(String[] args) {
        Set<Book> set=new TreeSet<Book>();
        //Creating Books
        Book b1=new Book(121,"Let us C","Yashwant Kanetkar","BPB",8);
        Book b2=new Book(233,"Operating System","Galvin","Wiley",6);
        Book b3=new Book(101,"Data Communications & Networking","Forouzan","Mc Graw Hill",4);
        //Adding Books to TreeSet
        set.add(b1);
        set.add(b2);
        set.add(b3);
        //Traversing TreeSet
        for(Book b:set){
            System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);
        }
    }
}
```

Output:

```
101 Data Communications & Networking Forouzan Mc Graw Hill 4
121 Let us C Yashwant Kanetkar BPB 8
233 Operating System Galvin Wiley 6
```

[< prev](#)[next >](#)

Share this page



Latest 4 Tutorials



CouchDB



Docker



Rails



RichFaces