# Java TreeSet class

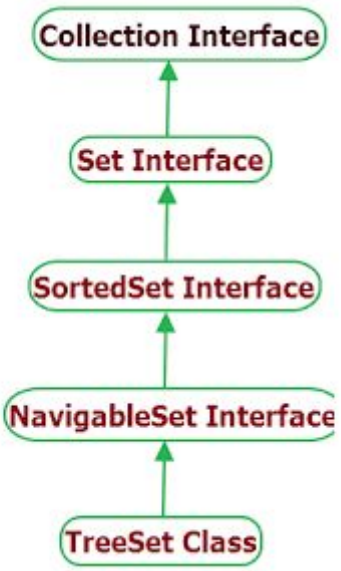
Java TreeSet class extends AbstractSet and `implements NavigableSet` interface. It is very similar to HashSet class, except it stores the element in sorted order.

The sort order is either natural order or by a Comparator provided at treeset creation time, depending on which constructor is used.

# 1. TreeSet Hierarchy

The TreeSet class extends `AbstractSet` class and implements `NavigableSet` interface. The NavigableSet interface extends `SortedSet` in hierarchical order.

```
class TreeSet<E> extends AbstractSet<E>
        implements NavigableSet<E>, Cloneable, Serializable
{
  //implementation
}
```




# 2. TreeSet Features

- It extends `AbstractSet` class which extends `AbstractCollection` class.

- It implements `NavigableSet` interface which extends `SortedSet` interface.
- Duplicate values are not allowed in TreeSet.
- NULL is not allowed in TreeSet.
- It is an ordered collection which store the elements in sorted order.
- Like HashSet, this class offers constant time performance for the basic operations(add, remove, contains and size).
- TreeSet does not allow to insert heterogeneous objects because it must compare objects to determine sort order.
- TreeSet is not synchronized. If multiple threads access a hash set concurrently, and at least one of the threads modifies the set, it must be synchronized externally.
- Use Collections.synchronizedSortedSet(new TreeSet()) method to get the synchronized TreeSet.
- The iterators returned by this class's iterator method are fail-fast and may throw `ConcurrentModificationException` if the set is modified at any time after the iterator is created, in any way except through the iterator's own `remove()` method.
- TreeSet also implements Searlizable and Cloneable interfaces.