# Java TreeSet and SortedSet Notes

## 1. SortedSet in Java

SortedSet is a subinterface of Set in Java. It stores elements in a sorted (ascending) order and does not allow duplicates. It is part of the java.util package and is most commonly implemented by TreeSet.

Key Features:

* - Sorted in natural or custom order
* - No duplicates allowed
* - Nulls not allowed
* - Not thread-safe

Common Implementing Class: TreeSet

Important Methods:

* • first() – Returns the first (lowest) element
* • last() – Returns the last (highest) element
* • headSet(E to) – Elements less than 'to'
* • tailSet(E from) – Elements greater than or equal to 'from'
* • subSet(E from, E to) – Elements between 'from' and 'to'

## 2. TreeSet in Java

TreeSet is a NavigableSet implementation that stores elements in sorted order. It uses a Red-Black Tree internally and is backed by a TreeMap.

Key Features:

* - Sorted order (ascending by default)
* - No duplicates
* - Not thread-safe
* - Null elements not allowed
* - Red-Black Tree internally (O(log n) operations)

### 2.1 Sorting Behavior

TreeSet can sort elements in two ways:

1. 1. Natural Sorting (Default):

* - Elements must implement Comparable

- Sorted in ascending order

1. 2. Custom Sorting using Comparator:

* - Pass Comparator to TreeSet constructor

- Example: new TreeSet<>(Comparator.reverseOrder())

### 2.2 Internal Working of TreeSet

TreeSet uses TreeMap internally, which is implemented using a Red-Black Tree. When elements are added, the tree structure is maintained in a balanced way.

Sorting is done using compareTo() or Comparator's compare().

### 2.3 Example Code

TreeSet<Integer> set = new TreeSet<>();  
set.add(50);  
set.add(10);  
set.add(30);  
System.out.println(set); // Output: [10, 30, 50]

### 2.4 TreeSet Summary

|  |  |
| --- | --- |
| Feature | TreeSet |
| Order | Sorted (natural or custom) |
| Duplicates | Not allowed |
| Null elements | Not allowed |
| Thread-safe | No |
| Backed by | TreeMap (Red-Black Tree) |
| Time complexity | O(log n) |