

JAVA - THE SORTEDSET INTERFACE

http://www.tutorialspoint.com/java/java_sortedset_interface.htm

Copyright © tutorialspoint.com

The SortedSet interface extends Set and declares the behavior of a set sorted in ascending order. In addition to those methods defined by Set, the SortedSet interface declares the methods summarized in below Table:

Several methods throw a NoSuchElementException when no items are contained in the invoking set. A ClassCastException is thrown when an object is incompatible with the elements in a set.

A NullPointerException is thrown if an attempt is made to use a null object and null is not allowed in the set.

SN Methods with Description

1 Comparator comparator

Returns the invoking sorted set's comparator. If the natural ordering is used for this set, null is returned.

2 Object first

Returns the first element in the invoking sorted set.

3 SortedSet headSetObjectend

Returns a SortedSet containing those elements less than end that are contained in the invoking sorted set. Elements in the returned sorted set are also referenced by the invoking sorted set.

4 Object last

Returns the last element in the invoking sorted set.

5 SortedSet subSetObjectstart, Objectend

Returns a SortedSet that includes those elements between start and end. Elements in the returned collection are also referenced by the invoking object.

6 SortedSet tailSetObjectstart

Returns a SortedSet that contains those elements greater than or equal to start that are contained in the sorted set. Elements in the returned set are also referenced by the invoking object.

Example:

SortedSet have its implementation in various classes like TreeSet, Following is the example for a TreeSet class:

```
import java.util.*;

public class SortedSetTest {

    public static void main(String[] args) {

        // Create the sorted set
```

```
SortedSet set = new TreeSet();

// Add elements to the set
set.add("b");
set.add("c");
set.add("a");

// Iterating over the elements in the set
Iterator it = set.iterator();
while (it.hasNext()) {
    // Get element
    Object element = it.next();
    System.out.println(element.toString());
}
}
```

This would produce the following result:

```
a
b
c
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js