

<<Comparable >> (default sorting)

Comparable is a predefined interface present in java.lang package

<<Comparable>> was introduced from JDK 1.2

Comparable is used for comparing Objects and sorting them either in ascending order or descending order

comparable has an abstract method called as compareTo()

syntax:

Comparable is used for default sorting and mutual comparison

Rules for comparing Objects using comparable interface

- 1) class should implement comparable interface
- 2) specify generics, indicating the object to be compared
- 3) Override compareTo() by specifying the logic of comparison

note: All wrapper classes and String class implements comparable interface and has overridden compareTo()

Therefore in order to compare String Objects double Objects we should always use compareTo() present in the respective classes

You are developing a system to manage a music library. Each song is represented by a Song class, which has attributes like title, artist, and duration. You decide to use a TreeSet to store the songs in a sorted order based on their titles. Assume the Song class has appropriate implementations of compareTo

Question:

Create a TreeSet named songSet and add the following songs to it:

Title: "Sundari", Artist: "Sanju Rathod ", Duration: 2:50

Title: "Maniac", Artist: "Yo Yo Honey Singh", Duration: 4:54

Title: "Shape of You", Artist: "Ed Sheeran", Duration: 3:54

Title: "Gulabi Sadi", Artist: "Sanju Rathod", Duration: 3:10

After adding the songs, print the songs in the sorted order of their titles.

<<Comparator>> (custom sorting)

Compare is a predefined interface present in java.util package

Comparator was introduced in java 1.2

Comparator is also used for Comparing Objects and sorting them either in ascending order or descending order

Comparator has an abstract method called as compare()

syntax: public int compare(Element e1, Element e2)

```
{  
//e1--Object to be inserted
```

```
//e2 already existing Object
```

```
}
```

Comparator is used for custom sorting and Unmutual Comparison

Rules for comparing objects using Comparator interface

- 1)create a new class which implements comparator interface and import it from java.util package
- 2)specify generics indicating the object to be compared
- 3)Override compare() by specifying the business logic of sorting
- 4)create an Object of the class where the sorting logic is defined and pass it to the constructor of TreeSet

Difference Between Comparable and Comparator

- ---Comparable

- 1)predefined interface present in java.lang package introduced from jdk 1.2
- 2)comparable has an abstract method called as compareTo()
- 3)syntax:
`public int compareTo(Element e);`
- 4)default sorting
- 4)mutual comparison
- 5)The java Source code and sorting logic is written in same class (dependent on each other)
- 6)Comparable provide single sorting sequence

- ---Comparator

- 1)predefined interface present in java.util package introduced from jdk 1.2
- 2)comparator has an abstract method called as compare()
- 3)syntax:
`public int compare(Element e1,Element e2);`
- 4)Custom sorting
- 4)Unmutual comparison
- 5)The java Source code and sorting logic is present in different class (Independent on each other)
- 6)Comparable provide multiple sorting sequence