

Java Comparable interface

Java Comparable interface is used to order the objects of user-defined class. This interface is found in java.lang package and contains only one method named `compareTo(Object)`. It provides single sorting sequence only i.e. you can sort the elements on the basis of a single data member only. For example, it may be rollno, name, age or anything else.

`compareTo(Object obj)` method

`public int compareTo(Object obj)`: is used to compare the current object with the specified object.

We can sort the elements of:

1. String objects
2. Wrapper class objects
3. User-defined class objects

Collections class

Collections class provides static methods for sorting the elements of collections. If collection elements are of Set or Map, we can use `TreeSet` or `TreeMap`. But we cannot sort the elements of List. Collections class provides methods for sorting the elements of List type elements.

Method of Collections class for sorting List elements

`public void sort(List list)`: is used to sort the elements of List. List elements must be of Comparable type.

Note: String class and Wrapper classes implements Comparable interface by default. So if you store the objects of string or wrapper classes in list, set or map, it will be Comparable by default.

Java Comparable Example

Let's see the example of Comparable interface that sorts the list elements on the basis of age.

File: Student.java

```
class Student implements Comparable<Student>{  
    int rollno;  
    String name;  
    int age;  
    Student(int rollno,String name,int age){  
        this.rollno=rollno;  
        this.name=name;  
        this.age=age;  
    }  
  
    public int compareTo(Student st){  
        if(age==st.age)  
        return 0;
```

```
else if(age>st.age)
return 1;
else
return -1;
}
}
```

File: TestSort3.java

```
import java.util.*;
import java.io.*;
public class TestSort3{
public static void main(String args[]){
ArrayList<Student> al=new ArrayList<Student>();
al.add(new Student(101,"Vijay",23));
al.add(new Student(106,"Ajay",27));
al.add(new Student(105,"Jai",21));

Collections.sort(al);
for(Student st:al){
System.out.println(st.rollno+" "+st.name+" "+st.age);
}
}
}
```

Test it Now

```
Output:105 Jai 21
       101 Vijay 23
       106 Ajay 27
```

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

Share this page



Latest 4 Tutorials



CouchDB



Docker



Rails



RichFaces