**COMPARABLE AND COMPARATOR**

|  |  |
| --- | --- |
| **COMPARABLE** | **COMPARATOR** |
| It sorts the single sorting objects either id, name or age | It sorts the multiple sorting sequences |
| It uses compare To method | It uses compare method |
| It affects the original class | It does not affect the original class |
| It is java.lang package | It is in java.util package |
| E.g. : Collections.sort(list); | E.g. : Collections.sort(List, comparator); |

**1 . COMPARABLE PROGRAM**

**Class Name :** EmployeeComparableProgram

**package** com.example.demo.samplePrograms;

**public** **class** EmployeeComparableProgram **implements** Comparable<EmployeeComparableProgram>{

**private** **int** id;

**private** String name;

**private** **int** age;

**private** **long** salary;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **long** getSalary() {

**return** salary;

}

**public** **void** setSalary(**long** salary) {

**this**.salary = salary;

}

**public** EmployeeComparableProgram(**int** id, String name, **int** age, **long** salary) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.age = age;

**this**.salary = salary;

}

**public** EmployeeComparableProgram() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** **int** compareTo(EmployeeComparableProgram employeeComparableProgram) {

//return Integer.compare(age, employeeComparableProgram.age); //line 50 and line 64 are same meaning

//return Integer.this.age.compareTo(employeeComparableProgram.age); this line gives a compilation error

// if(this.age==employeeComparableProgram.age) {

// return 0;

//

// }

// else if(this.age<employeeComparableProgram.age) {

// return -1;

//

// }else {

// return 1;

//

// }

//

**return**(**this**.age==employeeComparableProgram.age) ? 0:

((**this**.age<employeeComparableProgram.age)?-1:1);

}

//To get the value for each attribute instead of object value , the below method can be uncommented or in the main class we can use by object.getAge()

//@Override

//public String toString() {

// return "EmployeeComparableProgram [id=" + id + ", name=" + name + ", age=" + age + ", salary=" + salary + "]";

//}

}

**2 . COMPARATOR PROGRAM**

**Class Name :** StudentComparatorProgram

**package** com.example.demo.samplePrograms;

**public** **class** StudentComparatorProgram {

**int** srollno;

String sname;

**int** sage;

**public** **int** getSrollno() {

**return** srollno;

}

**public** **void** setSrollno(**int** srollno) {

**this**.srollno = srollno;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **int** getSage() {

**return** sage;

}

**public** **void** setSage(**int** sage) {

**this**.sage = sage;

}

**public** StudentComparatorProgram(**int** srollno, String sname, **int** sage) {

**super**();

**this**.srollno = srollno;

**this**.sname = sname;

**this**.sage = sage;

}

}

**Class Name :** StudentAgeComparator

**package** com.example.demo.samplePrograms;

**import** java.util.Comparator;

**public** **class** StudentAgeComparator **implements** Comparator<StudentComparatorProgram>{

@Override

**public** **int** compare(StudentComparatorProgram o1, StudentComparatorProgram o2) {

//return Integer.compare(o1.getSage(), o2.getSage());

// **TODO** Auto-generated method stub

**return** Integer.*compare*(o1.sage, o2.sage);

//return Integer.compare(s1.getSage(), s2.getSage());

}

// @Override

// public int compare(StudentComparatorProgram s1, StudentComparatorProgram s2) {

// //return s1.getSage().compareTo(s2.getSage());

// //return(s1.sage==s2.sage) ? 0: ((s1.sage<s2.sage)?-1:1);

// //return s1.sname.compareTo(s2.sname);

//

// return Integer.compare(s1.getSage(), s2.getSage());

// }

}

**Class Name :** StudentNameComparator

**package** com.example.demo.samplePrograms;

**import** java.util.Comparator;

**public** **class** StudentNameComparator **implements** Comparator<StudentComparatorProgram>{

@Override

**public** **int** compare(StudentComparatorProgram o1, StudentComparatorProgram o2) {

// **TODO** Auto-generated method stub

//return .compare(o1.getSname(), o2.getSname());

**return** o1.sname.compareTo(o2.sname);

}

}

**Class Name :** BasicExamplesProgramApplication (Main-Class)

**package** com.example.demo;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** com.example.demo.samplePrograms.EmployeeComparableProgram;

//import com.example.demo.samplePrograms.EmployeeComparableProgram;

**import** com.example.demo.samplePrograms.StudentAgeComparator;

**import** com.example.demo.samplePrograms.StudentComparatorProgram;

**import** com.example.demo.samplePrograms.StudentNameComparator;

**@SpringBootApplication**

**public class** BasicExamplesProgramApplication {

**public static void main(String[] args)** {

SpringApplication.run(BasicExamplesProgramApplication.class, args);

ArrayList<EmployeeComparableProgram> object1 = new ArrayList<EmployeeComparableProgram>();

object1.add(new EmployeeComparableProgram(1, "pri", 79, 10000));

object1.add(new EmployeeComparableProgram(2, "divya", 21, 2000));

object1.add(new EmployeeComparableProgram(3, "moni", 80, 2000));

**System.out.println("-----------Comparable Program----------");**

**System.out.println("------Before Sorting--------");**

object1.forEach(employeeComparableProgram -> **System.out.println("Id " + employeeComparableProgram.getId()**

**+ " name " + employeeComparableProgram.getName() + " age " + employeeComparableProgram.getAge()));**

Collections.sort(object1);

**System.out.println("--------After Sorting--------");**

object1.forEach(employeeComparableProgram -> System.out.println("Id " + employeeComparableProgram.getId()

+ " name " + employeeComparableProgram.getName() + " age " + employeeComparableProgram.getAge()));

Collections.reverse(object1);

**System.out.println("------After Sorting in reverse order-------");**

object1.forEach(employeeComparableProgram -> System.out.println("Id " + employeeComparableProgram.getId()

+ " name " + employeeComparableProgram.getName() + " age " + employeeComparableProgram.getAge()));

// System.out.println("After Sorting in reverse order" + object1);

// for(EmployeeComparableProgram object2 :object1) { //

// System.out.println("Age" + object2.getAge()); // }

**System.out.println("--------Comparator Program-------------");**

ArrayList<StudentComparatorProgram> studentComparatorProgramObject = new ArrayList<StudentComparatorProgram>();

studentComparatorProgramObject.add(new StudentComparatorProgram(56, "priyanka", 77));

studentComparatorProgramObject.add(new StudentComparatorProgram(57, "monika", 22));

studentComparatorProgramObject.add(new StudentComparatorProgram(58, "divya", 61));

**System.out.println("--------Age comparator-----------");**

**System.out.println("---Before Sorting----");**

studentComparatorProgramObject.forEach(

l -> System.out.println("roll No " + l.getSrollno() + " name " + l.getSname() + " age " + l.getSage()));

**System.out.println("-----After Sorting-----");**

Collections.sort(studentComparatorProgramObject, new StudentAgeComparator());

studentComparatorProgramObject.forEach(

l -> System.out.println("roll No " + l.getSrollno() + " name " + l.getSname() + " age " + l.getSage()));

**System.out.println("------After Sorting in reverse order------");**

Comparator<StudentComparatorProgram> comparatorObject5 = Collections.reverseOrder(new StudentAgeComparator());

Collections.sort(studentComparatorProgramObject, comparatorObject5);

studentComparatorProgramObject.forEach(

l -> System.out.println("roll No " + l.getSrollno() + " name " + l.getSname() + " age " + l.getSage()));

**System.out.println("----------Name comparator----------");**

**System.out.println("-------Before Sorting--------");**

studentComparatorProgramObject.forEach(

l -> System.out.println("roll No " + l.getSrollno() + " name " + l.getSname() + " age " + l.getSage()));

**System.out.println("------After Sorting-------");**

Collections.sort(studentComparatorProgramObject, new StudentNameComparator());

studentComparatorProgramObject.forEach(

l -> System.out.println("roll No " + l.getSrollno() + " name " + l.getSname() + " age " + l.getSage()));

**System.out.println("-----After Sorting in reverse order-----");**

Comparator<StudentComparatorProgram> comparatorObject = Collections.reverseOrder(new StudentNameComparator());

Collections.sort(studentComparatorProgramObject, comparatorObject);

studentComparatorProgramObject.forEach(

l -> System.out.println("roll No " + l.getSrollno() + " name " + l.getSname() + " age " + l.getSage()));

}

}

**OUTPUT**



