Implementing Custom Sort with Comparator

Bước 1 . Mở ứng dụng NetBean , New Project | Java | Java Application , nhấp Next , trên hộp thoại "New Java Application " điền các thông tin trong hộp thoại như sau :

Project Name : LabCustomSortComparator; project Location : "chọn đường dẫn lưu ứng dụng" , nhấn Finish để thúc .

1.1 Nhấp phải chuột trên <default package> chọn New | Java Class , đặt tên lớp : Employee, nhấp Finish để kế thúc .

Viết code cho tập tin Employee.java như sau :

```
public class Employee implements Comparable{
    private String id;
    private String name;
    private  int salary;
    private Date dateOfBirth;
    public Employee(String id, String name, int salary, Date dob )
        this.name = name;
        this.id = id;
        this.salary = salary;
        this.dateOfBirth = dob;
    }
    public String getName() {...3 lines }
    public void setName(String name) {...3 lines }
    public String getId() {...3 lines }
    public int getSalary() {...3 lines }
    public void setSalary(int salary) {...3 lines }
    public Date getDateOfBirth() {...3 lines }
    public void setDateOfBirth(Date dateOfBirth) {...3 lines }
```

```
    @Override
    public String toString() {
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        String dob = sdf.format(dateOfBirth);
        return "[ ID=" + id + ", name=" + name + ", salary=" + salary+",DoB="+ dob + "]";
    }
    public int compareTo(Object o) {
        return id.compareTo(((Employee)o).id);
    }
    public static Comparator compareBySalary = new Comparator() {
        @Override
        public int compare(Object o1, Object o2) {
            Employee e1 = (Employee)o1;
            Employee e2 = (Employee)o2;
            //int d = e1.salary-  e2.salary;
            int d =  Integer.compare(e1.salary, e2.salary);
            if(d>0) return -1;
            if(d==0) return e1.id.compareTo(e2.id);
            return 1;
        }
    };

    public static Comparator compareByDateOfBirth = new Comparator() {
        @Override
        public int compare(Object o1, Object o2) {
            Employee e1 = (Employee)o1;
            Employee e2 = (Employee)o2;
            //int d = e1.salary-  e2.salary;
            //the value 0 if the argument Date is equal tothis Date;
            //a value less than 0 if this Date is before the Date argument;
            //and a value greater than 0 if this Date is after the Date argument
            int d =  e1.dateOfBirth.compareTo(e2.dateOfBirth);
            if(d>0) return -1; //sap giảm dan (khi e1 sau e2)
            if(d==0) return 0;
            return 1;
        }
    };

     public static Comparator compareByName = new Comparator() {
         @Override
         public int compare(Object o1, Object o2) {
             Employee e1 = (Employee)o1;
             Employee e2 = (Employee)o2;
             //int d = e1.salary-  e2.salary;
             int d =  e1.name.compareTo(e2.name);
             if(d>0) return 1;
             if(d==0) return 0;
             return -1;
         }
     };
}//end class
```

1.2 Nhấp phải chuột trên <default package> chọn New | Java Class , đặt tên lớp : DemoComparator, nhấp Finish để kế thúc .

Viết code cho tập tin DemoComparator.java như sau :

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;
import java.util.List;

public class DemoComparator {
    static void printList(List<Employee> l){
       for(Employee emp: l){
            System.out.println(emp);
       }
    }//
    static Date getDateFromString(String date){
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        Date d = null;
        try {
            d = sdf.parse(date);
        } catch (ParseException ex) {
            System.out.println(ex.getMessage());
        }
        return d;
    }//
   public static void main(String[] args) {
        ArrayList<Employee> arraylist = new ArrayList();
        arraylist.add(new Employee("ID003", "Chaitanya", 25, getDateFromString("12/10/1990")));
        arraylist.add(new Employee("ID002", "Rahul", 28, getDateFromString("20/8/1990")));
        arraylist.add(new Employee("ID001", "Ajeet", 25, getDateFromString("11/07/1990")));
        System.out.println("Sorting on ID ascending");
        Collections.sort(arraylist);
        printList(arraylist);
        System.out.println("Sorting on descending by DateOfBirth ");
        Collections.sort(arraylist, Employee.compareByDateOfBirth);
        printList(arraylist);
        System.out.println("Sorting on ascending by Name");
        Collections.sort(arraylist,Employee.compareByName);
        printList(arraylist);
        System.out.println("Sorting on descending by Salary using Lambda");
        Collections.sort(arraylist, (Employee e1, Employee e2) -> {
            return e2.getSalary()-(e1.getSalary());
        });
        printList(arraylist);
    }
```

Bước 2 : Chạy ứng dụng và kiểm tra kết quả

```
Sorting on ID ascending
[ ID=ID001, name=Ajeet, salary=25,DoB=11/07/1990]
[ ID=ID002, name=Rahul, salary=28,DoB=20/08/1990]
[ ID=ID003, name=Chaitanya, salary=25,DoB=12/10/1990]
Sorting on descending by DateOfBirth
[ ID=ID003, name=Chaitanya, salary=25,DoB=12/10/1990]
[ ID=ID002, name=Rahul, salary=28,DoB=20/08/1990]
[ ID=ID001, name=Ajeet, salary=25,DoB=11/07/1990]
Sorting on ascending by Name
[ ID=ID001, name=Ajeet, salary=25,DoB=11/07/1990]
[ ID=ID003, name=Chaitanya, salary=25,DoB=12/10/1990]
[ ID=ID002, name=Rahul, salary=28,DoB=20/08/1990]
Sorting on descending by Salary using Lambda
[ ID=ID002, name=Rahul, salary=28,DoB=20/08/1990]
[ ID=ID001, name=Ajeet, salary=25,DoB=11/07/1990]
[ ID=ID003, name=Chaitanya, salary=25,DoB=12/10/1990]
```