import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Date;

import java.util.Scanner;

class Employee{

private int id = 0;

private String name;

private String department;

private Date dateOfJoining;

private int age;

private int salary;

Employee() {

this.id += 1;

}

Employee(String name, String dept, Date date, int age, int salary)

{

this.name = name;

this.department = dept;

this.dateOfJoining = date;

this.age = age;

this.salary = salary;

}

void setName(String name) {

this.name = name;

}

void setDepartment(String dept) {

this.department= dept;

}

void setDateOfJoining(Date date) {

this.dateOfJoining = date;

}

void setAge(int age) {

this.age = age;

}

int getAge() {

return age;

}

int getId() {

return id;

}

String getName() {

return name;

}

String getDepartment() {

return department;

}

Date getDateOfJoining() {

return dateOfJoining;

}

int getSalary() {

return salary;

}

@Override

public String toString() {

String ans = String.format("%-15s %-30s %-30s %-10s %-10s\n", this.name, this.department, this.dateOfJoining, this.age, this.salary);

return ans;

}

}

class salaryComparator implements Comparator<Employee>{

public int compare(Employee o1, Employee o2) {

// TODO Auto-generated method stub

if(o1.getSalary() != o2.getSalary()) {

return o1.getSalary() - o2.getSalary();

}

return 0;

}

}

class AgeComparator implements Comparator<Employee>{

public int compare(Employee o1, Employee o2) {

// TODO Auto-generated method stub

if(o1.getAge() != o2.getAge()) {

return o1.getAge() + o2.getAge();

}

else {

return o1.getSalary() + o2.getSalary();

}

}

}

public class EmployeeBO

{

public static void main(String[] args) throws ParseException

{

Scanner in = new Scanner(System.in);

ArrayList<Employee> employees = new ArrayList<Employee>();

// System.out.println("Enter number of employees!");

// int n = in.nextInt();

// while(n-- > 0) {

// SimpleDateFormat obj = new SimpleDateFormat("dd-mm-yyyy");

// String name = in.nextLine();

// String dept = in.nextLine();

// Date doj = obj.parse(in.nextLine());

// int age = in.nextInt();

// int sal = in.nextInt();

// employees.add(new Employee(name, dept, doj, age, sal));

// }

SimpleDateFormat obj = new SimpleDateFormat("dd-mm-yyyy");

employees.add(new Employee("Nikitha", "DataAnalysis", obj.parse("02-04-2003"), 35, 95000));

employees.add(new Employee("Varun", "Production", obj.parse("06-02-2001"), 30, 92000));

employees.add(new Employee("Rahul", "Marketing", obj.parse("03-05-2005"), 45, 56000));

System.out.println("1.Sort employees by salary");

System.out.println("2.Sort employees by age and by date of\r\n" + "joining");

System.out.println("Enter Your Choice");

int choice = in.nextInt();

switch(choice)

{

case 1: {

Collections.sort(employees, new salaryComparator());

System.out.println(String.format("%-15s %-30s %-30s %-10s %-10s\n", "Employee Name", "Department", "DOJ", "Age", "Salary"));

for (int i = 0; i < employees.size(); i++)

{

System.out.println(employees.get(i));

}

break;

}

case 2: {

Collections.sort(employees, new AgeComparator());

System.out.println(String.format("%-15s %-30s %-30s %-10s %-10s\n", "Employee Name", "Department", "DOJ", "Age", "Salary"));

for (int i = 0; i < employees.size(); i++)

{

System.out.println(employees.get(i));

}

break;

}

default:

break;

}

}

}



