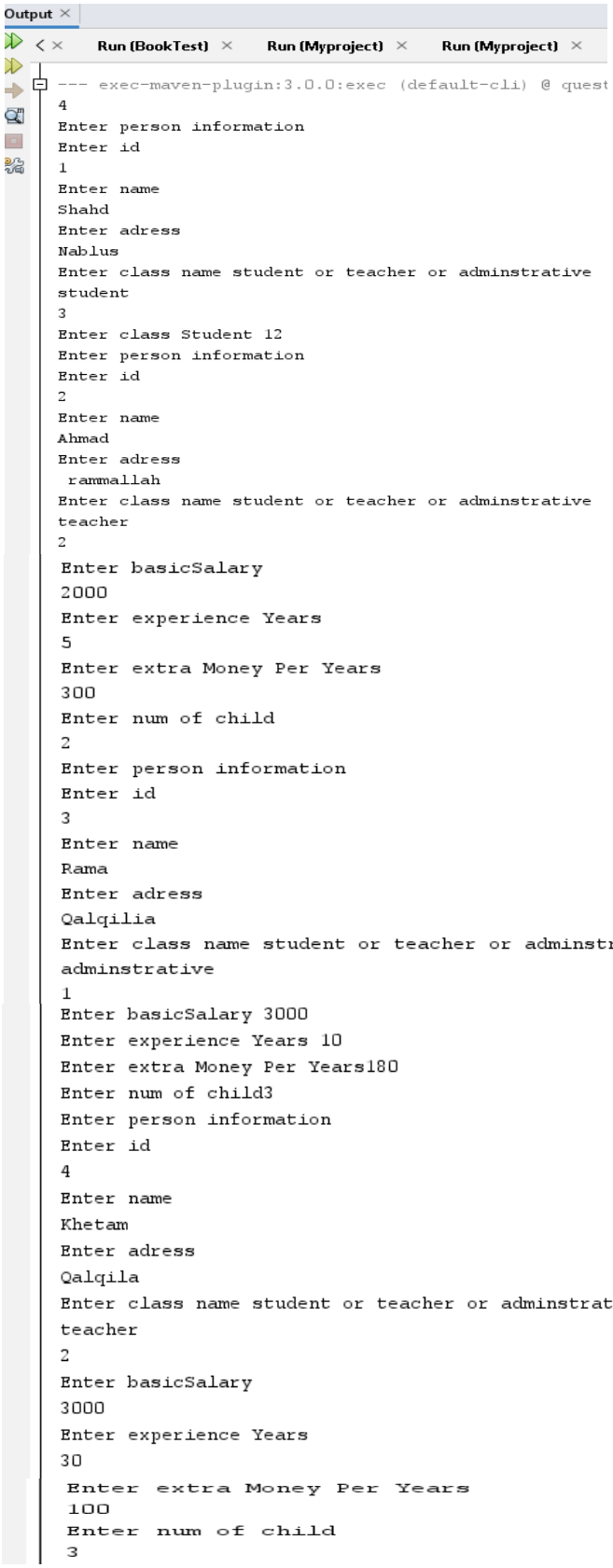
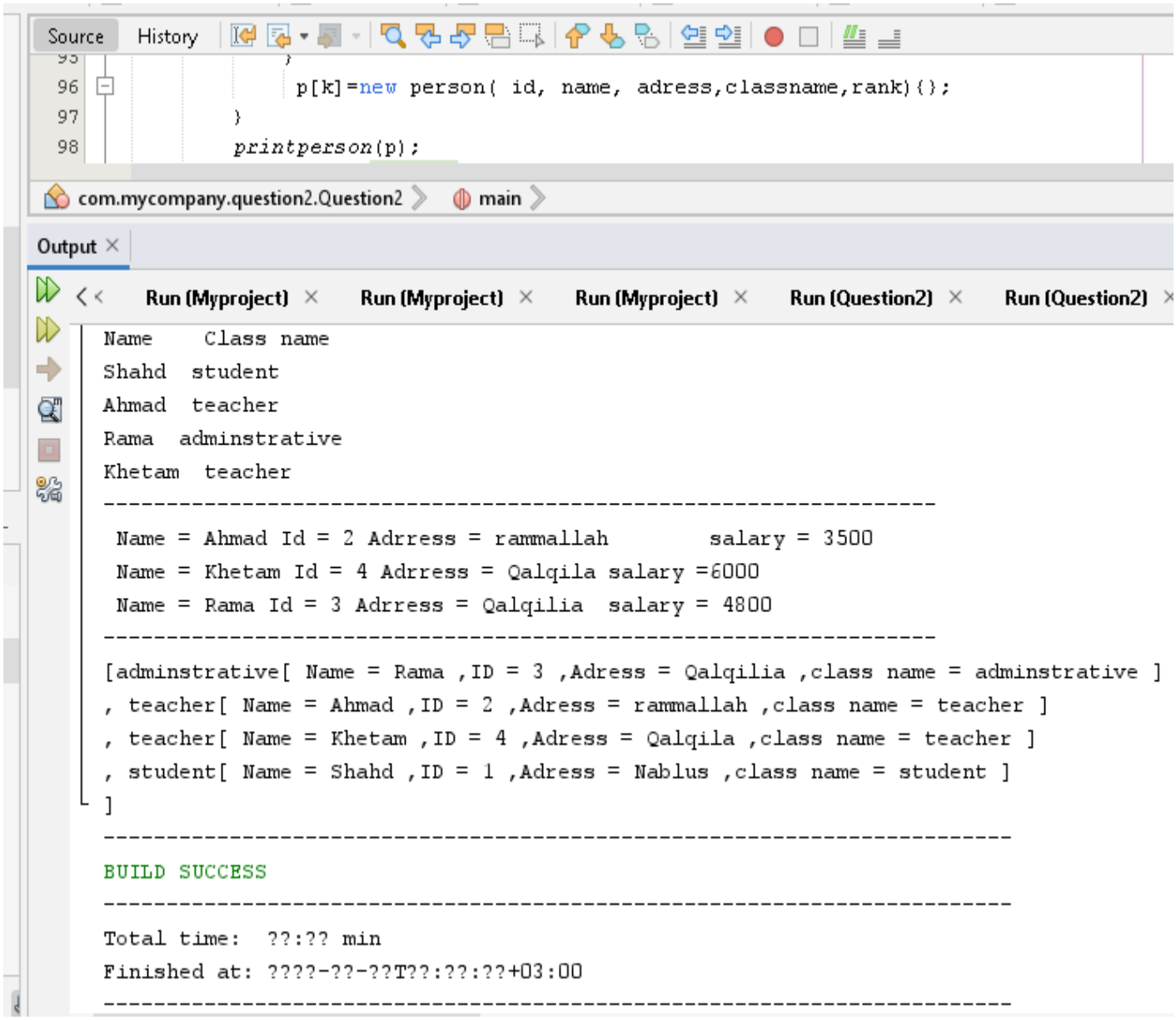
Question 2:

Input :



Output:


Source
History
96 p[k]=new person(id, name, adress,classname,rank){};
97 }
98 printperson(p);
com.mycompany.question2.Question2
main
Output
Run (Myproject)
Run (Myproject)
Run (Myproject)
Run (Question2)
Run (Question2)
Name Class name
Shahd student
Ahmad teacher
Rama adminstrative
Khetam teacher
Name = Ahmad Id = 2 Adrress = rammallah salary = 3500
Name = Khetam Id = 4 Adrress = Qalqila salary =6000
Name = Rama Id = 3 Adrress = Qalqilia salary = 4800
[adminstrative[Name = Rama ,ID = 3 ,Adress = Qalqilia ,class name = adminstrative]
, teacher[Name = Ahmad ,ID = 2 ,Adress = rammallah ,class name = teacher]
, teacher[Name = Khetam ,ID = 4 ,Adress = Qalqila ,class name = teacher]
, student[Name = Shahd ,ID = 1 ,Adress = Nablus ,class name = student]
]
BUILD SUCCESS
Total time: ??:?? min
Finished at: ????-??-??T??:??:??+03:00

```
import java.util.Arrays;
import java.util.Scanner;



public class Question2{
   public static int countt=0;
   public static int counta=0;
    public static int counts=0;
 public static void  printperson(person []arr) {

    System.out.println("Name \t"+"Class name");
    for (int i=0 ;i<=arr.length-1;i++){
     System.out.println(arr[i].getName()+"  "+arr[i].getClassname());
    }

 }
  public static void printsalary(paid []a,paid []t) {



   for (int i=0 ;i<=countt-1;i++){
      System.out.println ( t[i].printinfo()+"\t"+"salary "+t[i].salary()  );


    }
   for (int j=0 ;j<=2;j++){
      System.out.println ( a[j].printinfo()+"\t"+ "salary "+a[j].salary() );
   }
```

```
}


  public static void sortacadmic (person[]arr) {
   Arrays.sort(arr);
  System.out.println(Arrays.toString(arr));

  }
  public static void main(String[] args) {
     Scanner in=new Scanner (System.in);
     int num=in.nextInt();

   person []p  =new person[num];
   Student []s=new Student[num];
   paid [] t=new teacher[num];
    paid [] a=new Administrative[num];
   Administrative[] admin=new Administrative[num];
    teacher []teach=new teacher [num];


     for (int  k=0 ;k<=num-1;k++){
        System.out.println ("Enter person information ");
        System.out.println("Enter id ");
        int id=in.nextInt();
        System.out.println("Enter name ");
        String name=in.next();
        System.out.print("Enter adress ");
        String adress=in.next();
        System.out.print("Enter class name student or teacher or adminstrative ");
        String classname=in.next();
```

```
  int rank=in.nextInt();
        if (classname.equals("student")){
          counts++;
           System.out.print("Enter class Student");
           int levelclassStudent=in.nextInt();

           s[k]=new Student(levelclassStudent ,id, name, adress,"student",rank);
        }
        else if(classname.equals("teacher")){

           System.out.println("Enter basicSalary ");
          double  basicSalary=in.nextDouble();
          System.out.println("Enter experience Years");
          double experienceYears=in.nextDouble();
          System.out.println("Enter extra Money Per Years");
          double extraMoneyPerYears=in.nextDouble();
           System.out.println("Enter num of child");
           int numofchild=in.nextInt();
           countt++;
            t[k]=new teacher(experienceYears,basicSalary,extraMoneyPerYears, numofchild,id, name,
adress,"teacher",rank);
        }
        else if(classname.equals("adminstrative")){
            System.out.print("Enter basicSalary ");
          double  basicSalary=in.nextDouble();
          System.out.print("Enter experience Years");
          double experienceYears=in.nextDouble();
          System.out.print("Enter extra Money Per Years");
          double extraMoneyPerYears=in.nextDouble();
           System.out.print("Enter num of child");
```

```
   int numofchild=in.nextInt();
                counta++;
              a[k]=new Administrative(experienceYears,basicSalary,extraMoneyPerYears, numofchild,id, name,
adress,"adminstrative",rank);
             }
              p[k]=new person( id, name, adress,classname,rank){};
          }
          printperson(p);
          System.out.println("----------------------------------------------------------------");
           printsalary(a,t);

          System.out.println("----------------------------------------------------------------");
          sortacadmic(p);

      }

  }
```

```
interface Academic  {
 public String materialName();
public void markAttendene();

}
interface paid{
   public double salary ();
   public String printinfo();
}

abstract class person implements Comparable<person> {
   protected int id;
   protected String name;
   protected String adress;
   protected String classname;
   protected int rank;
   public person() {
   }




   public person(int id, String name, String adress, String classname, int rank) {
      this.id = id;
      this.name = name;
      this.adress = adress;
      this.classname = classname;

      this.rank = rank;
```

```
}



public int getRank() {
   return rank;
}

public void setRank(int rank) {
   this.rank = rank;
}



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 String getAdress() {
        return adress;
    }

    public void setAdress(String adress) {
        this.adress = adress;
    }

    public String getClassname() {
        return classname;
    }

    public void setClassname(String classname) {
        this.classname = classname;
    }
     @Override
    public int compareTo(person o) {
        return this.rank -o.rank;
    }
    @Override
    public  String toString () {
      return classname+ "[ "+"Name = "+name+ " ,ID = "+id + " ,Adress = "+ adress+" ,class name = "+classname+ "
]\n";

    }



}
```

```
class Student extends person implements Academic {
  public int levelclassStudent;
  private String material;
  private String teacherName;

   public Student() {
   }

   public Student(int levelclassStudent) {
      this.levelclassStudent = levelclassStudent;
   }

   public Student(int levelclassStudent, int id, String name, String adress, String classname, int rank) {
      super(id, name, adress, classname, rank);
      this.levelclassStudent = levelclassStudent;
   }


   @Override
   public int getId() {
      return id;
   }

   @Override
   public void setId(int id) {
      this.id = id;
   }

   @Override
```

```
public String getName() {
    return name;
}

@Override
public void setName(String name) {
    this.name = name;
}

@Override
public String getAdress() {
    return adress;
}

@Override
public void setAdress(String adress) {
    this.adress = adress;
}

@Override
public int getRank() {
    return rank;
}

@Override
public void setRank(int rank) {
    this.rank = rank;
}
```

```
    public int getLevelclassStudent() {
        return levelclassStudent;
    }

    public void setLevelclassStudent(int levelclassStudent) {
        this.levelclassStudent = levelclassStudent;
    }

    @Override
    public String materialName() {
      return "Studen name" +this.name+"material that student study "+material +" "+"Teacher name "+teacherName;
    }

    @Override
    public void markAttendene() {
        teacher t=new teacher();
       System.out.print(t.materialName());
    }

}
```

```
abstract class employee extends person implements paid{
     public double experienceYears;
    public double basicSalary;
    public double extraMoneyPerYears;
   int numOfemployeechild;









    public employee() {
    }


    public employee(int numOfemployeechild) {
        this.numOfemployeechild = numOfemployeechild;
    }


    public employee(double experienceYears, double basicSalary, double extraMoneyPerYears, int
numOfemployeechild) {
        this.experienceYears = experienceYears;
        this.basicSalary = basicSalary;
        this.extraMoneyPerYears = extraMoneyPerYears;
        this.numOfemployeechild = numOfemployeechild;
    }


    public employee(double experienceYears, double basicSalary, double extraMoneyPerYears, int
numOfemployeechild, int id, String name, String adress, String classname, int rank) {
        super(id, name, adress, classname, rank);
        this.experienceYears = experienceYears;
```

```
this.basicSalary = basicSalary;
    this.extraMoneyPerYears = extraMoneyPerYears;
    this.numOfemployeechild = numOfemployeechild;
  }

  public double getExperienceYears() {
    return experienceYears;
  }

  public void setExperienceYears(double experienceYears) {
    this.experienceYears = experienceYears;
  }

  public double getBasicSalary() {
    return basicSalary;
  }

  public void setBasicSalary(double basicSalary) {
    this.basicSalary = basicSalary;
  }

  public double getExtraMoneyPerYears() {
    return extraMoneyPerYears;
  }

  public void setExtraMoneyPerYears(double extraMoneyPerYears) {
    this.extraMoneyPerYears = extraMoneyPerYears;
  }

  public int getNumOfemployeechild() {
```

```
return numOfemployeechild;
 }

 public void setNumOfemployeechild(int numOfemployeechild) {
    this.numOfemployeechild = numOfemployeechild;
 }

  @Override
 public int getId() {
    return id;
 }

  @Override
 public void setId(int id) {
    this.id = id;
 }

  @Override
 public String getName() {
    return name;
 }

  @Override
 public void setName(String name) {
    this.name = name;
 }

  @Override
 public String getAdress() {
```

```
    return adress;
    }

    @Override
    public void setAdress(String adress) {
        this.adress = adress;
    }

    @Override
    public int getRank() {
        return rank;
    }

    @Override
    public void setRank(int rank) {
        this.rank = rank;
    }
    public int markAttendence;

}
```

```
class teacher extends employee implements Academic{


   private String levelclass;

   String studentname;

   String materialname;

   public int markAttendence;

   public teacher() {

   }


   public teacher(double experienceYears, double basicSalary, double extraMoneyPerYears, int
numOfemployeechild, int id, String name, String adress, String classname, int rank) {

      super(experienceYears, basicSalary, extraMoneyPerYears, numOfemployeechild, id, name, adress, classname,
rank);

   }



   @Override

   public double getExperienceYears() {

      return experienceYears;

   }


   @Override

   public void setExperienceYears(double experienceYears) {

      this.experienceYears = experienceYears;

   }


   @Override

   public double getBasicSalary() {

      return basicSalary;

   }
```

```
@Override
public void setBasicSalary(double basicSalary) {
   this.basicSalary = basicSalary;
}

@Override
public double getExtraMoneyPerYears() {
   return extraMoneyPerYears;
}

@Override
public void setExtraMoneyPerYears(double extraMoneyPerYears) {
   this.extraMoneyPerYears = extraMoneyPerYears;
}

@Override
public int getNumOfemployeechild() {
   return numOfemployeechild;
}

@Override
public void setNumOfemployeechild(int numOfemployeechild) {
   this.numOfemployeechild = numOfemployeechild;
}

@Override
public int getId() {
   return id;
}
```

```
@Override
public void setId(int id) {
   this.id = id;
}

@Override
public String getName() {
   return name;
}

@Override
public void setName(String name) {
   this.name = name;
}

@Override
public String getAdress() {
   return adress;
}

@Override
public void setAdress(String adress) {
   this.adress = adress;
}

@Override
public int getRank() {
   return rank;
}
```

```
    @Override
    public void setRank(int rank) {
        this.rank = rank;
    }

    @Override
    public double salary() {

      return  basicSalary+(experienceYears*extraMoneyPerYears);
    }

    @Override
    public  String printinfo() {
      return "Name = "+name+ "Id = "+id + "Adrress = "+ adress;

    }

    @Override
    public String materialName() {
        return "Level class that techer learn them = "+this.levelclass+"teacher name "+this.name+"material
name"+this.materialname;
    }
    @Override
    public void markAttendene() {
       System.out.print( "student name" +studentname + materialName()+"mark attendence"+ markAttendence);
    }

}
```

```
class Administrative extends employee{


    public Administrative() {

    }


    public Administrative(double experienceYears, double basicSalary, double extraMoneyPerYears, int
numOfemployeechild, int id, String name, String adress, String classname, int rank) {

        super(experienceYears, basicSalary, extraMoneyPerYears, numOfemployeechild, id, name, adress, classname,
rank);

    }


    @Override

    public double getExperienceYears() {

        return experienceYears;

    }


    @Override

    public void setExperienceYears(double experienceYears) {

        this.experienceYears = experienceYears;

    }


    @Override

    public double getBasicSalary() {

        return basicSalary;

    }


    @Override

    public void setBasicSalary(double basicSalary) {

        this.basicSalary = basicSalary;

    }
```

```
@Override
public double getExtraMoneyPerYears() {
   return extraMoneyPerYears;
}

@Override
public void setExtraMoneyPerYears(double extraMoneyPerYears) {
   this.extraMoneyPerYears = extraMoneyPerYears;
}

@Override
public int getNumOfemployeechild() {
   return numOfemployeechild;
}

@Override
public void setNumOfemployeechild(int numOfemployeechild) {
   this.numOfemployeechild = numOfemployeechild;
}

@Override
public int getId() {
   return id;
}

@Override
public void setId(int id) {
   this.id = id;
}
```

```
@Override
public String getName() {
    return name;
}

@Override
public void setName(String name) {
    this.name = name;
}

@Override
public String getAdress() {
    return adress;
}

@Override
public void setAdress(String adress) {
    this.adress = adress;
}

@Override
public int getRank() {
    return rank;
}

@Override
public void setRank(int rank) {
    this.rank = rank;
}
```

```
    @Override
    public double salary() {

        return  basicSalary +(experienceYears*extraMoneyPerYears);
    }

   @Override
    public  String printinfo () {
      return "the name = "+name+ "id = "+id + "adrress = "+ adress;

    }

}
```