



ADNAN MENDERES UNIVERSITY

CSE 203 Object-Oriented Programming

Lab 04 Encapsulation- Inheritance

- You should submit **ONLY solutions of Homework**. Submit **whole java project file in zipped format** and named project file as *NameSurname_studentNo_Lab4*.
- Do your homework in **ECLIPSE IDE**.
- Do not use Turkish Characters(ç,ğ,ı,ö,ş,ü) for naming project, methods, classes.
- Late submissions are not allowed.
- You should do homework **YOURSELF**. Group working is not allowed.
- Copy homework will be evaluated as 0.
- Use Google Classroom for your questions.

NOTE: **DO NOT TAKE ANY INPUT FROM USER!!!** Use true encapsulation (design all data fields as private and reach them just using get/set methods) and inheritance practice in your implementation. Your project have to consist of 3 classes: *CommisionEmployee*, *BasePlusCommisionEmployee*, *InheritanceTest*

HOMEWORK

There are 2 separate class given below, create those accordingly (*CommisionEmployee*, *BasePlusCommisionEmployee*, *Test*).

1- Design a class named *CommisionEmployee* that contains:

- A private *String* data field named *firstName* that stores first name of employee
- A private *String* data field named *lastName* that stores last name of employee
- A private *String* data field named *socialSecurityNumber* that stores SSN of employee
- A private *double* data field named *grossSales*.
- A private *double* data field named *commisionRate*.
- A five argument constructor that creates a *CommisionEmployee* with the specified **firstName, lastName, socialSecurityNumber, grossSales, commisionRate**.
 - Validate **grossSales** and **commisionRate** (grossSales must be greater than or equal to 0, commission rate must be greater than 0 and less than 1). If these values are valid, assign arguments to class's data fields. Otherwise, print warning message then terminate the program or you may throw an exception
- The accessor (getter) method for all data fields.
- The mutator (setter) method for all data fields. Apply validation in grossSales and commisionRate's setter methods like stated in the constructor method.
- A method named *earnings()* that returns the earning of employee which is calculated by multiplying grossSales and commisionRate.
- Create *toString()* method that returns object information as a String.



ADNAN MENDERES UNIVERSITY

CSE 203 Object-Oriented Programming

2- Design a class named **BasePlusCommisionEmployee** that contains:

- A private **String** data field named **firstName** that stores first name of employee
- A private **String** data field named **lastName** that stores last name of employee
- A private **String** data field named **socialSecurityNumber** that stores SSN of employee
- A private **double** data field named **grossSales**.
- A private **double** data field named **commisionRate**.
- A private **double** data field named **baseSalary**.
- A six argument constructor that creates a **BasePlusCommisionEmployee** with the specified **firstName**, **lastName**, **socialSecurityNumber**, **grossSales**, **commisionRate**, **baseSalary**.
 - Validate **grossSales**, **commisionRate** and **baseSalary** (**grossSales** must be greater than or equal to 0, **commissionRate** must be greater than 0 and less than 1, **baseSalary** must be greater than or equal to 0). If these values are valid, assign arguments to class's data fields. Otherwise, print warning message then terminate the program or you may throw an exception.
- The accessor (getter) method for all data fields.
- The mutator (setter) method for all data fields. Apply validation in **grossSales**, **commisionRate** and **baseSalary**'s setter methods like stated in the constructor method.
- A method named **earnings()** that returns the earning of employee which is calculated by adding base salary to the multiplication of **grossSales** and **commisionRate**.
- Create **toString()** method that returns object information as a **String**.

Apply Following Changes to Provide Inheritance

You created 2 separate class and their properties as given above. Now, **Inheritance** relationship should be applied between them. There are same data fields, constructor parameters and methods in both classes. According to inheritance; data fields-constructor-other methods should be arranged so that redundant data/code in these fields minimizes.

Thus, make necessary changes in the given classes below so that:

- **CommissionEmployee** will be superclass of inheritance hierarchy
- **BasePlusCommissionEmployee** will be a subclass of **ComissionEmployee**.

Some necessary changes are given below

Changes in “**CommisionEmployee**” Class

- You may use “this” keyword
- **CommissionEmployee** is **superclass** of inheritance hierarchy
- Make data fields **private**.

Changes in “**BasePlusCommisionEmployee**” Class

- Design **BasePlusCommissionEmployee** as a **subclass of** **ComissionEmployee**.



ADNAN MENDERES UNIVERSITY

CSE 203 Object-Oriented Programming

- Make data fields **private**
- Make an **explicit call to superclass's constructor** in six argument constructor of BasePlusCommissionEmployee (**super()**). Provide necessary validations (ex: BaseSalary).
- Make getter/setter for data field(s) which are not inherited from superclass. Provide necessary validations.
- Eliminate redundancies in **data fields, constructor, getter/setter methods, earnings(), toString()**

Create "InheritanceTest" Class

- Create a BasePlusCommissionEmployee object with a first name, last name, SSN, gross sales, commission rate and base salary.
- Print the object's information using getter and related methods.
- Update the object with setting new value to base salary.
- Print the object's updated information using toString() method and related methods.

You may use your own preferred data field values while creating and updating objects.

Sample Run:

Employee information obtained by get methods and earnings:

```
First name: Bob
Last name: Lewis
Social security number: 333-33-3333
Gross sales: 5000,00
Commission rate: 0,04
Base salary: 300,00
Earnings: 500,00
```

Updated employee information obtained by toString and earnings:

```
First name: Bob
Last name: Lewis
Social security number: 333-33-3333
Gross sales: 5000,00
Commission rate: 0,04
Base salary: 1000,00
Earnings: 1200,00
```