

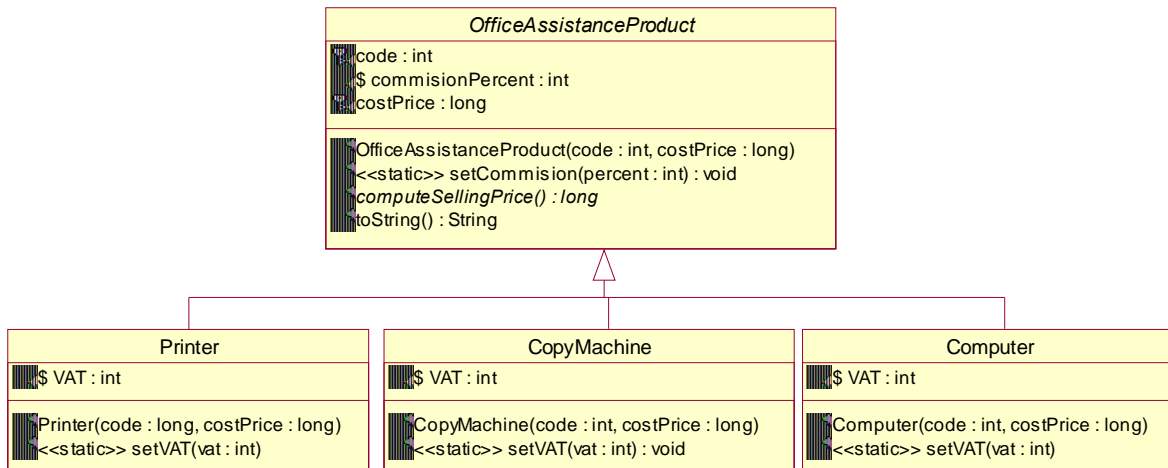
Lab 13. Recapitulative Exercises (I)

Exercise 1. A shop for office furniture and products sells printers, copy machines, and computers. Each product has a numerical code and a cost price. For each product the shop adds a 10% commission. Moreover the shop collects a VAT differentiated by the product type. Let assume that the VAT list is stored in the file “vat.txt” with the following content:

```
printer 15%
computer 10%
copy machine 20%
```

Implementing the following hierarchy classes:

As you may observe, the class `OfficeAssistanceProduct` is abstract, having the abstract method

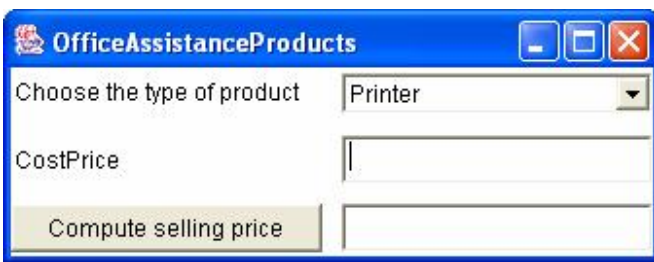


`computeSellingPrice`.

Write a class `Test` containing a `main()` method which receives a series of products (maximum 10 products) computes and displays their selling prices. By example, if a printer with code 1245 comes from a supplier with a cost price of 3,000,000 lei, your program should print the following message:

Printer, code: 1245, selling price: 3,750,000 lei.

To improve the application, create a graphical interface like the following figure:



The user will choose a kind of product and he will insert its cost price. When the user click on the “Compute selling price” button, your program should display the selling price computed with the method `computeSellingPrice()`.

Exercise 2. Using the classes from the previous problem, write a class named Shop, which contains variables to store information about the following properties: shop's name (a string of characters), owner's name (a string of characters), closing hour (an integer number) and opening hour (an integer number). In addition, the class will have the following methods:

- two constructors: one which initialize all the properties of the class with the information given through the parameters of the constructor and the other one will initialize only the name of the shop and owner, respectively, imposing the timetable: 10-18.
- rewrite the method toString() of the class Object to return complete information about a certain shop. For instance:
Name shop: RTC Office Express
Owner: Ionescu Pop
Timetable: 9:00-19:00
- a predicate-method isOpened() which returns a boolean value corresponding with the shop is opened or not.
- A method getMessage() which returns a string to say if the shop is opened ("Open") or closed ("Close") in the moment that you call the method isOpened().

Use this class to create a test class that contains the method main() and:

- through the static method showInputDialog(...) of the class JOptionPane ask the user to insert information to create two objects of the class Shop,
- creates two objects of the Shop class, one with the first constructor and the other using the second constructor,
- displays the information memorized in the objects through the method toString(),
- displays the timetable of the second shop,
- displays if the second shop is opened now.

By example:

Name shop: RTC Office Express
Owner: Ionescu Pop
Timetable: 9:00-19:00
Open