COMP102- Object Oriented Programming

Spring 2017

HOMEWORK2

Deadline: 18.02.2017

**1. Definition of the Program**

In this homework, you are asked to write an application for a supermarket. The rules for the supermarket are below:

* In the application, there are **Customer**s and **Item**s.
* Each customer has a **name** field. If the name of the customer is not given, the application defines his name as “*unknown*”.
* Customers have a **promotion** value between 0 and 100. For instance, if a customer has a promotion of 70, it means he pays only the 70% of his shopping. If an incorrect promotion value is given, (a negative value or a value grater than 100), the application automatically sets it as 100. Also, the promotion value of a customer is defined 100 at the initialization of the customer unless it is defined otherwise.
* For each customer, there are **calculatedPayment** and **discountedPayment** values. These values are set to 0 at the time the customer is initialized.
* The application records the items bought by each customer, called **basket.** To ease the job of the programmer, the size of the basket is defined as 30 elements. The user will NOT attempt to add more than 30 items to any customer. (Assume that the hidden goal of the application is to understand the favorite product of each customer).
* An item has a **name** and a **price** field. Note that price can never be a negative value. If the price is intended to set a negative value, the application sets the price of the item as zero.
* Until the termination of the program, any customer can add items to his/her basket. Thus, the customer class should have the **addItem** method. In the addItem method, the calculatedPayment and discountedPayment values should be updated according to the price of the Item. (To ease the job of the programmer, the quantity of the added item is always assumed 1 unit).
* Also, until the termination of the program, any customer can remove some items from his/her basket. Thus, the customer class should have **deleteItem** method. In the deleteItem method, the calculatedPayment and discountedPayment values should be updated according to the price of the Item
* At any time of the program, the program can list the items bought by a selected customer. Thus, the customer class should have **listItems** method.
* In this example, to ease the job of the programmer, assume that supermarket has only 3 customers.

**2. Programming Definitions**

The programmer has to write the classes below:

**Customer**

-name: String

-promotion: int

-basket: Item[30]

Customer()

Customer(name:String)

Customer(name:String, promotion:int)

setName(name: String)

getName(): String

setPromotion(promotion:int)

getPromotion(): int

addItem(Item i)

deleteItem (String itemName)

listItem()

**Item**

-name: string

-price: int

Item()

Item(String name, int price)

setName(customer: Customer)

getName(): Customer

setPrice(day:int)

getPrice(): int

In the main function, the application defines 3 customers as following:

Customer c1 = new Customer("Ali",90);

Customer c2 = new Customer("Ayse");

Customer c3 = new Customer();

c3.setPromotion(70);

Here are the steps of the application running with the commands:

* In each iteration, the program lists the name of the customers.

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

* The user selects the customer. (Example: enters 2)

(1) AddItem

(2) DeleteItem

(3) ListItems

* If the user selects 1, the program asks the following information and adds the defined item to the basket of the selected customer.

Enter the name of the item:

Enter the price of the item:

* If the user selects 2, the program asks the name of the item and deletes the item from the basket. Note that deleteItem method should search the basket of the customer to find the item with the given name. Note that the program is error free that user will not attempt to delete an item which does not exist in the basket.

Enter the name of the item:

* If the user selects 3, the program lists the item in the basket of the customer in the following format.

1 name\_1 price\_1

2 name\_2 price\_2

…..

Total price: XXX

Discounted Price: YYY

* If the user selects 4, the application terminates.

**Naming and Submission Rules:**

Name of the project: HW2

Name of the java files: Customer.java Item.java and Main.java

You can use package

Given messages to the console:

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

(1) AddItem

(2) DeleteItem

(3) ListItems

Enter the name of the item:

Enter the price of the item:

1 name\_1 price\_1

2 name\_2 price\_2

…..

Total price: XXX

Discounted Price: YYY

**Submission:**

- Make a directory and name it as name\_surname\_number.

- Put Customer.java Item.java and Main.java file under it.

- Zip your directory.

- e-mail it to: Huseyin Emre Erdem <huseyin.erdem@agu.edu.tr>

Here is an example output of the application

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

1

(1) AddItem

(2) DeleteItem

(3) ListItems

1

Enter the name of the item:

Computer

Enter the price of the item:

3000

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

2

(1) AddItem

(2) DeleteItem

(3) ListItems

1

Enter the name of the item:

Skirt

Enter the price of the item:

250

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

1

(1) AddItem

(2) DeleteItem

(3) ListItems

1

Enter the name of the item:

rose

Enter the price of the item:

20

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

1

(1) AddItem

(2) DeleteItem

(3) ListItems

3

1 Computer 3000

2 rose 20

Total price:3020

Discounted Price:271800

(1) Ali

(2) Ayse

(3) unknown

(4) Terminate

4