SOFT7004 – OOP- Project

(Third Assessment)

You are to create an inventory/shopping application as outlined below. Test program thoroughly so that it can handle the requirements. Please use only one Java package for the entire project.

PART 1

You are to write the following java classes:

1.

Product: This is a class for products that a company may sell.

This class has the following fields:

1. productID
2. name
3. description
4. price

Use appropriate types for the above fields.

Write the following methods:

1. getter and setter for description
2. getter for productID
3. getter and setter for price
4. A method called print() with no implementation.

Note that no two Product objects should have the same productid, apply a strategy that the productid is assigned automatically.

productID must not be used as an input argument in the constructor.

2.

ProductDB:
This class manages a HashMap of Products (This HashMap is the productDB field). The keys of this HashMap is the prodcutID.

The class provides methods to add, remove and find a product using the keys and returning the whole collection of products. Note that the HashMap should not be manipulated directly.

3.

Phone:

Phone **is a** product.

This class has the following fields:

1. make: **Make** can only be either **Apple**, **Samsung** or **Nokia**.
2. model
3. storage

Write the following methods:

1. getter and setter for make
2. getter and setter for model
3. getter and setter for storage
4. getter and setter for **name** from the **Product** class.
5. Override print() method

Note: There should be a mechanism in the project so every time a new Phone object is created, the object is automatically added into the ProductDB class.

4.

TV:

TV **is a** product.

This class has the following fields:

1. make
2. model

3. type: Type can only be either **LED** or **LCD**

Write the following methods:

1. getter and setter for make
2. getter and setter for model
3. getter and setter for type
4. getter and setter for **name** from the **Product** class.
5. Override print() method

Note: There should be a mechanism in the project so every time a new TV object is created, the object is automatically added into the ProductDB class.

5.

OrderDetails:

This class has the following fields:

1. **Product**
2. **quantity**

Provide setters and getters for **product** and **quantity**.

6.

Order:

This class has one field:

An ArrayList of OrderDetails.

The class provides methods to add, remove, find an OrderDetails and return the entire arrayList. Note that the ArrayList should not be manipulated directly.

7.

Customer:

This class has the following fields:

1. Name
2. Address
3. Arraylist of Order

Write the following methods:

1. getter and setter for name.
2. getter and setter for address.
3. Add, remove, get, showAll methods for ArralyList of Order.

Note1: You are responsible to decide the correct relations between these classes (inheritance, association, aggregation etc).

Note2: No extra method or field (except those mentioned above) are allowed to be inserted in the classes.

PART 2

You are to create a UML class diagram which details all the java classes used in your application.

PART 3

Create a Test class and perform the following scenario in the main method:

1. Create an empty object of ProductDB.
2. Create 4 phone objects (p1, p2, p3, p4) with your desired data. (Every time a new object of Phone is created, it needs to be added automatically in the ProductDB)
3. Create 3 TV objects (T1, T2, T3) with your desired data. (Every time a new object of TV is created, it needs to be added automatically in the ProductDB)
4. Create 3 OrderDetails as follows
    a. 5 phones (P2)
    b. 2 TVs (T1)
    c. 1 TV (T3)
5. Create an Order that contains the above OrderDetails.
6. Create a Customer with the above Order.
7. Display all products in ProductDB

Note: Appropriate error and exception handling need to be considered wherever required.

Rubric

This rubric is subject to change.

1. Correct task implementation (E.g., Correct use of OOP, Correct relation between components). (100\%)
2. Relatively correct task implementation (E.g., use of OOP, relation between components etc). (70\%)
3. Partly correct task implementation (E.g., use of OOP, relation between components etc). (40\%)
4. Wrong task implementation. (0\%)

Submission

Type your name and student ID and your cohort's name (e.g., SD2, SDH2 CS2) as a comment at the top of EVERY single Java file.

Please zip the entire project and submit it via Canvas. Please do not email your submission to me as it is getting rejected.

The deadline for this project is 15th Dec 2021. One-week late submission with 10 marks penalty would be accepted and the deadline would be 22nd Dec 2021. Two weeks late submission with 20 marks penalty would be accepted and the deadline would be 29th Dec 2021.

Any question regarding the project should be communicated with farshad.toosi@mtu.ie