SE116 – LAB#4 2023-2024 SPRING

Aim: Use of Java Collections, Arrays class, static methods in Java Collections and Arrays class.

1. Implement a program to simulate a product line in a factory. Your project should contain the following four classes: `Product`, `Item`, `Order` and `Test`.

The class `Product` will have the following `private` data members: `int ID`, `String name`, `double price`, `String color` and `String description`. As always, please implement appropriate set/get methods. Add non-parameterized and parameterized constructors. Moreover, implement a method to print out the information of a `Product` object.

The class `Item` will have the following `private` data members: `int itemID`, `Product product`, `int quantity`. The data member `product` refers to a `Product` instance. The **"has-a" relationship** between the class `Product` and the class `Item` may be explained in the following manner: "An instance of class `Item` **has a** `product`". As always, implement appropriate set/get methods. Add non-parameterized and parameterized constructors. Implement a method to print out the information of an `Item` object. Implement another method to calculate the price of products in an item using quantity information.

The class `Order` will have the following `private` data members: `int orderID`, `ArrayList<Item> itemList`, `double totalPrice` and `int instalmentCount`. The data member `itemList` refers to an `ArrayList` that will store `Item` instances. The **"has-a" relationship** between the class `Order` and the class `Item` may be explained in the following manner: "An instance of class `Order` **has an** `itemList`". As always, implement appropriate set/get methods. Add non-parameterized and parameterized constructors. Implement a method to print out the information of an `Order` object. Use the get method of `totalPrice` to calculate the total price of all items in an order.

The class `Test` will only contain the `main` method. In your `main`, use 3 orders each with 5 items that consists different quantities of different products. You are free in your design to fill the information of the objects.

2. Modify your class `Order` to add the following data member: `static int numberOfOrders`. The class member `numberOfOrders` will count the total number of orders instantiated in the program. Do not forget to modify the constructors to be able to increment the corresponding number of orders automatically when a new `order` object is instantiated. Also, add an appropriate get method to return the number of orders.

Modify the method `main` of class `Test` in the following manner: Use a few orders and add some items to them. Meanwhile, print out the total number of orders a few times in the program. Then, for each order, print out the information of the most valuable items.

3. Modify the method `main` of class `Test`. In `main`, use an `ArrayList<Order>` reference named as `orderList`. Add new orders with items into `orderList`. Then, for each order in `orderList`, print out the information of (1) the product with the highest price, (2) the item with the lowest quantity and (3) the order with the highest number of installments. Finally, print out the average total price of orders. You can use `max()`, `min()`, `reverse()`, `sort()` or other similar methods from `Collections` class to get the correct results for this part.