CS F213 : OOP Lab 2 - Classes and Objects

Shreyansh Joshi, Santosh Srivatsan

February 23, 2021

The objective of this lab is to test your understanding of classes and objects in Java. In this lab you will simulate the working of a mall. For this task, you are required to create 3 classes - Mall, Shop, and Customer.

General Instructions -

Read the following instructions very carefully before starting your work -

1. Make sure that the source code is written in the default package.

2. Ensure that the name of each file is the same as the corresponding class name.

3. Ensure that the name of methods is the same as those given in Javadoc. Failure to comply may cause deduction in credit even if the implementation is correct.

4. **You won't be provided with a template file (helper code) this time**. You have to implement all the classes by yourself (including their respective constructors).

5. Do not copy directly the names of methods from Javadoc. Write them out yourselves.

6. You are required to upload all three classes at once. It is not possible to test individual classes on Codepost.

7. A brief description of the classes and methods to be used is given further in the problem statement but **you must refer to the Javadoc for the complete description.**

8. Be sure to check your code on the test cases provided locally (as jar file), before submitting on Codepost. Remember that Codepost takes only the latest submission and not the best submission into consideration.

9. The advisable order of writing the code:
    a) Customer
    b) Shop
    c) Mall

Note: The below description of classes is extremely brief. Only the names of fields and methods have been written (along with a 1-liner description wherever required). Students MUST refer to the javadoc for the complete details (use, return types, arguments, etc.) of all methods and fields .

Class Descriptions -

1. **Customer -**

   Fields -

   - **customerID -** ID of a particular customer.
   - **numProducts -** Number of products that the customer wants to buy.
   - **wantToBuy** - Denotes whether the customer wants to buy a product from a shop or not.

   Methods -

   - **getCustomerID()**
   - **getNumProducts()**
   - **getWantToBuy()**

2. **Shop -**

   Fields -

   - **capacity -** Maximum customers permitted in a shop at any time.
   - **currentCustomerCount -** No. of customers in the shop now.
   - **customerCountToday -** No. of customers that visited the shop today (till this time).
   - **customerList -** An array of 'Customer' objects that stores each customer that is currently present in the shop.
   - **MAX_PRODUCTS -** Maximum products that any shop can have. **Must be initialized to 10.**
   - **priceOfProduct -** Price of all the products in a shop.
   - **productCount -** No. of products available for purchase in the shop.
   - **productList -** An array of strings containing the names of all the products in the shop.
   - **revenueToday -** Total revenue the shop has made today (till this time).

   Methods -

   - **addProduct() -** Adds a product to the end of the 'productList'.

- **checkIn() -** Used to check-in a customer in the shop.
- **checkOut() -** Used to check-out a customer from the shop.
- **getCurrentCustomerCount()**
- **getCustomerCountToday()**
- **getProductCount()**
- **getProductList() -** Returns the list of products that are available for purchase in the shop.
- **getRevenueToday()**

3. **Mall -**

Fields -

- **MAX_SHOPS** - Maximum number of shops that a mall can have. **Must be initialized to 6.**
- **shopCount** - No. of shops in the mall.
- **shopList** - An array of 'Shop' objects that stores all the shops there in the mall.

Methods -

- **addProduct() -** Adds a new product to a particular shop in the mall.
- **addShop () -** Adds a new shop to the end of 'shopList'.
- **checkIn() -** Checkin a customer into a particular shop in the mall.
- **checkOut() -** Checkout a customer from a particular shop in the mall.
- **getCurrentCustomerCount() -** Returns no. of customers present in the mall currently.
- **getCustomerCountToday() -** Returns total customers that visited the mall today.
- **getProductList() -** Returns the list of products that are available for purchase in the shop specified by shopNumber.
- **getRevenueToday() -** Total revenue the mall has made today (till this time).
- **getTotalProductCount()**

Salient Points -

1. Use 0-based indexing everywhere. For instance, if a shop is added in the mall (that has currently no shops), 0 would denote that shop.

2. All products in a given shop have the same price and all shops have the same value of MAX_PRODUCTS. Likewise, all malls have the same value of MAX_SHOPS.

3. A customer buys product(s) (if he wants to) only while exiting a particular shop and not while entering. So update the corresponding variables accordingly.

4. Any list (array) be it *customerList , productList or shopList* adds new entries to the end of the array (as it is done normally). However, when it comes to removing elements from the list -
    - *productList* is implemented in FIFO (First In First Out) manner i.e whatever a customer buys, is removed from the front of the list. Suppose a shop has 4 products in the order - 'A' , 'B', 'C', 'D' in the queue, and a customer wants to buy 2 products, then he/she will have to buy 'A' and 'B'. So, the productList must be readjusted accordingly. Note that only the remaining products would be checked in the test cases. So, in the above example, only C and D are checked whether they are present and appear in the correct order or not.

    - *customerList* is implemented in LIFO (Last In First Out) manner. This simply means that the customer that entered a particular shop last, will be the first one to checkout.

5. Methods in 'Mall' require the corresponding methods in 'Shop' to be implemented correctly. So the 'Shop' class must be implemented correctly to get full credit. Some methods in 'Mall' and 'Shop' also require 'Customer' to be implemented correctly.

Marking Scheme -

Method tested	**Marks awarded**
Customer - getNumProducts, getWantToBuy, getCustomerID	0.5
Shop - checkIn	1
Mall - addShop	0.5
Mall - addProduct	0.5
Mall - getTotalProductCount	1
Mall - checkOut	2
Mall - getRevenueToday	1.5
Total	**7**