Assignment 2

This assignment builds on the work we've done in the lectures. Your goal is to create and run two test scenarios to demonstrate the functionality of the shopping cart transaction classes.

1. The first scenario is the scenario covered in the lecture session.
2. The second scenario is a slight variation of the first:
    a. The user adds three items
    b. Requests a display of the shopping cart items and total
    c. Removes one item
    d. Confirms the cart and makes an order
    e. The user submits a payment, however, the payment is not valid
    f. The user is sent a regret email notifying them that the order was unsuccessful

Each scenario should be contained in its own test method. See the lecture notes on this. Your starting point will be the Java code that I supplied in the lecture.

To complete this assignment, you are expected to proceed according to the development process that I demonstrated in last week's lecture.

Your solution for each scenario is only allowed to use the following classes. These are the classes we identified together in the lecture.

- Customer
- ShoppingCart
- Item
- Order
- Address
- Payment
- Email
- TransactionTest

Procedure

- The first thing you should do is get the basic ShoppingCart class compiling. In the lecture, we discussed the fields and methods it should contain. It will be useful for you to look at the notes again.
- Then you should move to the next line of the scenario, which involves adding items to the ShoppingCart. Make sure your ShoppingCart can add/remove Items.
- Then proceed one or two lines at a time, adding just as much functionality as is required to get the code compiling. In the end, you will have a programme that compiles but may still lack full functionality.
- Start again at the top. Add additional functionality in small parts - compiling and running your programme as you go.
- Do this until your program provides the output that you expect.

ShoppingCart:

It should be clear how to implement all the methods needed here. Some are 'getter' methods, others are add/remove methods for adding/removing items from the ShoppingCart's ArrayList. The one method that might not be clear is the one called 'close()'. When the close() method is called, items cannot be added or removed from the ShoppingCart. If you try to call the ShoppingCart's add(Item item) method after the close method is callled, it will printout an error : "*Sorry the shopping cart is closed*".

Hint: you should use a boolean variable to implement this. If its value is false, you can add/remove items; if its value is true, you cannot add/remove items

Item:

You already have the code for the Item class

Order:

The role of the Order class is to take a ShoppingCart object and transfer its items one by one into itself. It should also take the information about the Customer. Once this is done, the ShoppingCart should be empty. i.e. its Arraylist should be empty.

Order is an important class in this programme. Look over the lecture notes to see the list of methods that we provisionally assigned to it. It will have relationships with several other objects such as ShoppingCart, Payment, Address and Email.

Address:

The role of the Address class is to hold the address fields of the Customer's address: e.g. street, city, zip, country. A customer may have two Address objects associated with them - a billing address and a delivery address. You will need methods to set and get the information in each Address object.

Payment:

The Payment class holds the following pieces of information

- customer;
- credit card type;
- credit card number;
- date;
- address;
- credit card bank name

As an object that is in charge of verifying its own information, the Payment class also has code for validating the data submitted. For example, a new Payment object might be created as follows:

Payment payment1 = new Payment(customer1, address1, "NasterCard", 16534, 26.66, "05/10/2018") ;

or since the Order object contains most of the information you need, you could have an alternative constructor for Payment that takes an order object as an input parameter.

Payment payment1 = new Payment(order, "NasterCard", "05/10/2018") ;

You will need to write some code to make sure that the Card type is either "MasterCard" or "Visa". Clearly, "NasterCard" is not valid.

A bonus challenge would be that you check that the card number has the correct number of digits (not like 16534 above).

A bonus++ challenge is to check if the date is in the future. However, the minimum you need to do is the Card Type check.

[*Only do the bonus bits if you've got time on your hands*.]

If the input information is correct, then the Payment object should set a boolean field called 'valid'. This will allow you to define a method called isValid() which returns the value of the field. If it is true, it will run otherwise it will not run the body of the ‘if’ statement.

You can then write the following in your test code

```
if(payment1.isValid()){
//do something
}
else{
        // do something (print a message that the payment is not valid)
}
```

Email:

The role of the Email object is to send (you are required to just printout on the screen) an email message to the customer. If the Payment has been successful, then it will be a positive message giving the order number, the order details, the delivery and billing addresses. If the Payment has been unsuccessful, then the message explains that the order has not been made. In either case, the customer must be addressed by their first name and the email address is their email address (from the Customer object).

TransactionTest:

As illustrated in the lectures, this is a test class. It should have two methods - one for each scenario. They are executed from the main method, which is contained by the TransactionTest class.

What to submit:

1. A PDF document
2. A zip file containing your .java files.

The PDF document should contain (in this order):

- A text description of your solution to demonstrate your understanding
- Your output for scenario one
- Your output for scenario two

Your code should be copied and pasted into the PDF in this order:

- TransactionTest
- ShoppingCart
- Order
- Address
- Payment
- Email

This assignment is due on the 6th of November @ 5pm.