CSCI-UA.0102-006/008 Data Structures - Recitation

Problem 1 of Lab 2 (Part 1)

By: Qi Feng

In this exercise, you will create a simulation of a bank and an ATM machine. The assignment is divided into three parts. You should be able to complete Parts 1 and 2 during this lab session. You should submit both Part 1 and 2 by Tuesday February 7th before midnight packaged in one folder and zipped before submission. We will do part 3 next week.

1. Download the lab2 package which are provided with this assignment.
2. Have a quick look at Program.java. This program is what you will use eventually to test your code. Make sure the rest of your code is consistent with it.
3. Examine the Account class provided with this assignment – Account.java.
4. Add another constructor that initializes the account with a starting deposit.
5. Add two classes called **SavingsAccount** and **CheckingAccount**. These classes should inherent all the attributes of the **Account** class.
6. A SavingsAccount object, in addition to the attributes of an Account object, should have an **interest** variable and a method which **adds interest** to the account. The constructor should initialize the interest value using a parameter.
7. A CheckingAccount object, in addition to the attributes of an Account object, should have an **overdraft limit** variable. The constructor should initialize this variable using a parameter. Ensure that you have overridden methods of the Account class as necessary in both derived classes (in a CheckingAccount, the withdraw() method should also check that the overdraft limit is not exceeded)

CSCI-UA.0102-006/008 Data Structures - Recitation

Problem 2 of Lab 2 (Part 2)

By: Qi Feng

1. Create a Bank class, which contains an ArrayList of Account objects and a name variable. Add a constructor that initializes the Bank's name and initialize the ArrayList.

2. The Bank class requires methods for opening and closing accounts. Implement a consistent way of assigning account numbers. For example, when you open a new account, you can assign to it an account number according to the following formula

$$\text{size of your accounts ArrayList} + 10,000$$

3. You also need to add an update method in the bank class: It should iterate through each account, updating it in the following ways: Savings accounts get interest added (via the method you already wrote); CheckingAccounts get a letter sent if they are in overdraft (you can implement this by printing out some text)

4. Add a getAccount(int accountNum) method to your Bank class. This method should return an Account object from your arrayList.

CSCI-UA.0102-006/008 Data Structures - Recitation

Problem 3 of Lab 2 (Part 3 (May change))

By: Qi Feng

1. Add a class called ATM, which contains an ArrayList of banks. Add the required constructor which should initialize the ArrayList.
2. Add a method to add a bank to the array list.
3. Add a login method: prompt the user to enter a Bank name followed by an account number. Use this information to retrieve the Account object associated with this information. Return the Account object. You will need to iterate through your ArrayLists to get the correct object.
4. Add a run method: First call the login method, then output a menu of ATM options. Your ATM options should repeat until Exist is selected.

Here is an example of a menu: (You don't have to follow this exactly)

```
Enter bank name
Best Bank
Enter Account Number
1001
Login Successful ....
Deposit
Enter Amount
1050
Deposit Successful ...
Deposit
Enter Ammount
100
Withdrawal Successful ...
Exist
Thank you for banking with us
```