CS2430 SPRING 2016

Program #6 (30 points)

Dr. Lily Chang

Due Dates

1. All Java Files and Javadoc due on K drive group folder: Monday, May 2 @10pm
2. **Grace date: Wednesday, May 4 @10pm, or 0 points.**
3. You MUST demo by Thursday, May 5 @5pm, or **0 points**. You will NOT be able to demo if ANY files are dated after May 4 @10pm.

Program Requirement

1. Program 6 is an extension of Program 5. You will be working with the same partner(s) from Prog5.
2. You MUST DEMO Prog6 to get the credit for this program. All group members must attend the demo. And you MUST have the whole project folder ready on the K drive group folder by the grace date. Demo no later than **Thursday, May 5 @5pm**. You will not be allowed to demo if ANY files are dated after **May 4 @10pm**.
3. Each member of the group must do a fair share of work. I will be asking questions when you demo. If it is obvious that one person has a significantly lower level of understanding than the other, that person can expect to receive a lower grade. There will be a standard demo steps for all groups. The demo schedule will be announced on D2L.
4. You MUST use **SE tools** to log your time individually. Up to **-5 points** if this is not done or you did not log all your time or you did not provide specific comments as to what you were working on.
5. You will get **0 points** if you submit Prog6 after **May 4, 10pm** OR you **failed the demo**. You MUST submit a runnable Prog6, which met all the requirements to pass this course.
6. You will **lose 3 points** if the Java Doc is missing.
7. You MUST follow the software development ground rules. Maximum **10 points off** for programming style!
8. If you need help from me, place your project in your group folder on K drive. So I can open it from my office when you stop by.
9. You are required to add the following features to the **`GUI`**.
   - **`JList`** – Add a Java JList to the GUI. The JList dynamically displays the list of accounts. Use an instance of `Vector` class to manage the JList. You may use the following methods provided by the Vector class: `add()`, `remove()`, and `set()`. You can use `setListdata()` method to publish the Vector on GUI. Note that the Vector is solely used for managing the JList. You still need to maintain a list of accounts with an instance of the BankDatabase class.
     You must modify the `ActionListener` for Open/Close account buttons. After opening an account, the JList must show the current list of accounts including the one just opened. The user can also select an account from the JList to close the account. **Up to -10 points** if the JList is missing or not working, or the user cannot select and close from the JList.
   - **`JButton`** to **`Load Accounts`** – read the input file "Prog6_1.in" and load the accounts to the BankDatabase. After you loaded the accounts, the accounts should be published on the JList. **-2 points** if accounts displayed incorrectly.
   - **`JButton`** to **`deposit`** and **`withdraw`** money – select an account from the JList and deposit/withdraw money to/from the account. Use the `JOptionPane.showInputDialog()` to enter the amount for the deposit/withdraw. You must try catch the `NumberFormatException` if the amount entered is not numeric. Up to **-5 points** if the amount is incorrect after deposit/withdraw, or the exceptions are not handled.
   - **`JButton`** to apply the monthly interest and fee. Up to **-5 points** if the resulting balance is incorrect.
   - **`JButton`** with a **`textField and textArea`** to list the accounts opened before a certain date. Up to **-5 points** if the result is incorrect.
10. Modify the following classes.
    - **`BankDatabase.java`** – you may need to add the following method.

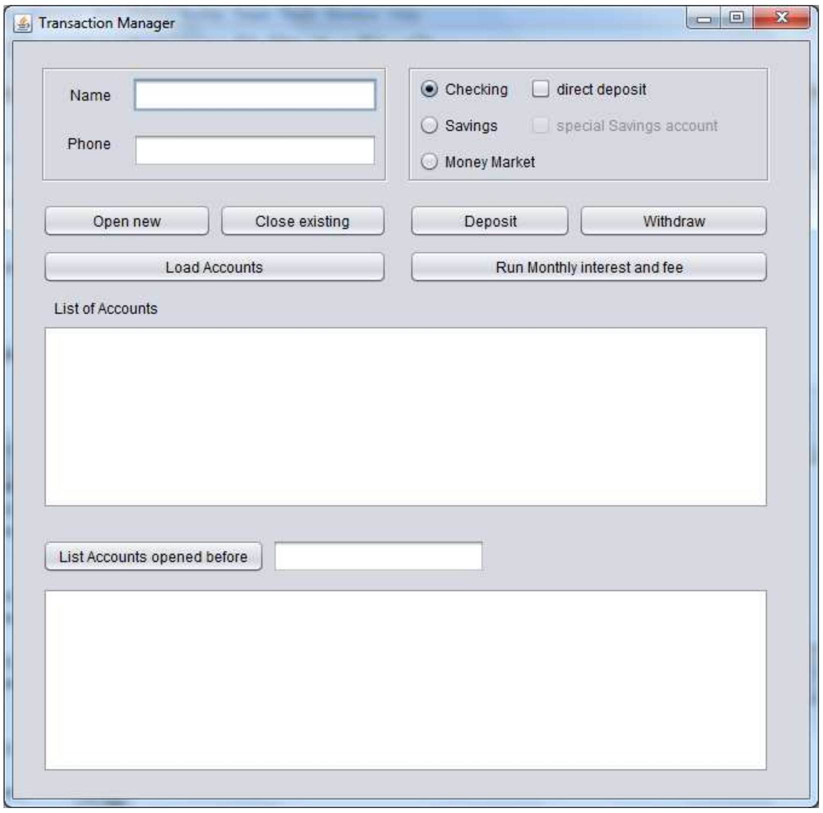
```
public boolean remove(int accno) //remove an account based on accno; move everybody up
public Account deposit(int accno, int amount)  //deposit money to the account
public Account withdraw(int accno, int amount) //withdraw money from the account
public void runInterest() //applyInterestAndFee() for everyone
```

- **`applyInterestAndFee()`** – implement this method in the `Checking`, `Savings` and the `MoneyMarket` `classes` based on the fee schedule in the table in Prog5. This method should calculate the monthly interest and fee based on current balance, and update the balance. For example, assume a Checking account has a balance of 1000, then the monthly interest earned will be `(1000 * 0.05%) / 12`. The new balance will be: `balance + interest – fee.` Remember, the fee is waived under certain conditions. Please refer to the conditions listed in the table of Prog5. Up to **-5 points** if the new balance is incorrect.

- **`Account.java`** – you must have the following methods.
```
public void debit(double amount)  //subtract the amount from the balance
public void credit(double amount) //add the amount to the balance
```

- **`Date.java`** – you must implement the `Comparable interface` and implement the `compareTo()` method. **-5 points** if this method is not implemented OR you have an incorrect implementation OR this method is not used.
```
public class Date implements Comparable
{
   ...
   /**
    Compare “this” with (Date) o; if “this” is before o, return -1; if “this” is equal
    to o return 0; if “this” is after o, return 1.
    */
   public int compareTo(Object o)
   ...
}
```

11. Sample GUI.


Transaction Manager
Name
Phone
Checking
direct deposit
Savings
special Savings account
Money Market
Open new
Close existing
Deposit
Withdraw
Load Accounts
Run Monthly interest and fee
List of Accounts
List Accounts opened before