Project 2
Bank Accounts*

Jiashuo Wang
5100309436

April 4, 2013

Instructor: Ling Gong

I. Objective

Suppose you are opening your own tiny bank, and want to manage the bank accounts in your bank. In your bank, there are only two kinds of account: checking and saving. Both of them should have an account number, and should be able to perform deposit, withdraw and check balance operations. People can open a new account with some money deposited, or no money at all. Checking account and saving account perform differently in some aspects, and there are some requirement to meet about the bank accounts:

Checking account need a minimum balance of $50 to avoid the monthly fee $2. Saving account can have at most three transactions (deposit or withdraw) in a month, otherwise there will be monthly fee $5. Saving account has no minimum balance requirement, and checking account has no transaction times limit. The monthly fee is based on the balance at the end of a month. That means you can avoid the fee by putting some money in your checking account just the day before the end of month. At the end of each month, saving account can have some interest. The default interest rate is 5% for a month (wow, I wish my citibank can have so high interest rate!) , and based on the ending balance of that month. The interest computation is before any monthly fee deduction (I'm nice again). Adding the interest and deducting monthly fee are not considered as a transaction. They are internal thing. The interest computation and monthly deduction can only happen at the end of a month You don't need to consider when is the end of a month. Whenever you call endMonth(...), that means we are at the end of a month now. Finally, there is a requirement about the account number. No two accounts, no matter checking or saving, can have the same account number. And the account number should keep increasing when you create a new account. Even some account get closed, the account number cannot be reused. Hint* see below.

You can add your own to make your own little bank more complete. Also, the test code doesn't cover every requirement I listed above. So, please don't use it as a fully judgment. You'd better write some test code yourself to fully test your implementation.

One more thing requirement is: the bank ONLY allows checking or saving account! (This means ..... If you can't get it, email me. :-))

*Designed by LaTeX

II. Algorithm

II.1 Accounts

There are three different classes about accounts here, *BankAccount*, *SavingAccount*, *CheckingAccount*. *SavingAccount* and *CheckingAccount* are extended from *BankAccount*.

- **BankAccount**
  It fulfils some basic functions. It can help to make an account number for each user, which is private. It can store the balance of each user, which is public. And it defines some basic abstract methods, deposit, withdraw or end-of-month, to be implement in the extended classes.
- **SavingAccount**
  It extends from the *BankAccount*, which means it implement the undone methods defined in the *BankAccount* according to the rule.
- **CheckingAccount**
  It also extends form the *BankAccount*. The differences from the *SavingAccount* are the end-of-month operation and little changes in deposit and withdraw.

II.2 What's New

After doing the basic task, I think more about this bank system. However, it looks more like a paper than a robot that can interact with the users. So I add some functions in it, which will make it better. As I am a new Java programmer, I will add some visual interface later.

- When withdrawing from a *CheckingAccount*, you have to think more because there is a rule here, Checking account need a minimum balance of $50 to avoid the monthly fee $2. So for example, if the balance of the account is only $38, one should not withdraw $38 or $37 because the balance will be minus after the end of the month. So I add a If statement to judge this kind of situation. That is to say, if the balance is only $38, one can withdraw $36 at most. So is the saving account when considering about the transaction.
- The command line for bank system need to be comprehensive. So I make a big change to it by adding some interaction. People can choose to deposit or withdraw, and when the user stop the operation to return the previous menu, I define that it will be the end of month because it is not clear as there's no clock in the system. You can see the interface in the screenshot of the result.
- Considering that the money can only have up to two decimal places, I use the round-off to print the money in a better form by using the class *BigDecimal* and method *setScale*, which is in the java.math.*.

II.3 More details

The format of the input of money need to be considered, which should has up to two decimal places. And the choose of menu should also be considered. All of these can be solved by the method, *matches*, which was shown by our TA last Friday. It's quite a easy way to do so.

III. Results and Conclusions

III.1 Environment

- Windows 8
- NetBeans IDE 7.3

III.2 Screenshots of the result

Use Command Line to compile and execute the program in Figure 1 and Figure 2.

III.3 Thoughts

Thinking more to add some new functions is interesting!

```
F:\Course\OS\Project2\src\project2>javac Project2.java

F:\Course\OS\Project2\src\project2>java Project2
my checking number is 1, my balance is 38.0
my checking number is 2, my balance is 38.0
my checking number is 3, my balance is 38.0
my checking number is 4, my balance is 38.0
my checking number is 5, my balance is 38.0
my saving number is 6, my balance is 94.5
my saving number is 7, my balance is 94.5
my saving number is 8, my balance is 94.5
my saving number is 9, my balance is 94.5
my saving number is 10, my balance is 94.5
Welcome to the bank system in the next month! Please choose what you want to do!

1: choose a bank account
0: exit
Your choice:1
Please input the bank account:4
What's next?
1: Deposit
2: Withdraw
3: Balance
0: back
Your choice:2
How much do you want to withdraw:37
Sorry! Your saving account don't have enough money!
What's next?
1: Deposit
2: Withdraw
3: Balance
0: back
Your choice:0
my checking number is 1, my balance is 36.0
my checking number is 2, my balance is 36.0
my checking number is 3, my balance is 36.0
my checking number is 4, my balance is 36.0
my checking number is 5, my balance is 36.0
my saving number is 6, my balance is 99.22
my saving number is 7, my balance is 99.22
my saving number is 8, my balance is 99.22
my saving number is 9, my balance is 99.22
my saving number is 10, my balance is 99.22
Welcome to the bank system in the next month! Please choose what you want to do!
```

Figure 1: *Screenshots of Bank Accounts(1)*

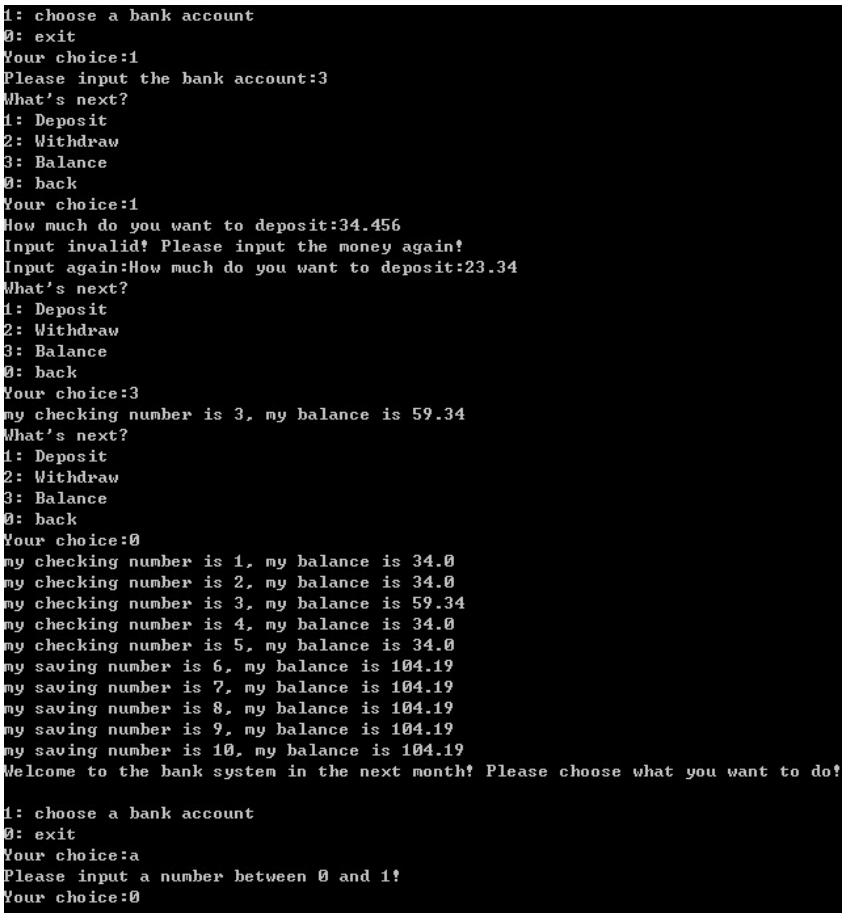

Figure 2: *Screenshots of Bank Accounts(2)*