

# Bank Accounts

---

Li Chengyao

5140219288

May 28, 2017

## 1 PROBLEM

### 1.1 DESCRIPTION

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. :- )

## 1.2 SAMPLE

Some simple test code is provided.

```
class AccountTest{
    public static void main(String[] args){
        BankAccount[] bank = new BankAccount[10];
        for(int i=0;i<5;i++){
            bank[i]=new CheckingAccount(50);
        }
        for(int i=5;i<10;i++){
            bank[i]=new SavingAccount(100);
        }

        for(int i=0;i<10;i++){
            bank[i].withdraw(20);
            bank[i].deposit(10);
            //bank[i].withdraw(10);
        }

        for(int i=0;i<10;i++){
            bank[i].endMonth();
            bank[i].print();
        }
    }
}
```

The output should look similar to the following with the basic information about account type, number, and balance. But you can make it as fancy as you like.

```
>java AccountTest
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
```

## 2 ALGORITHM

### 2.1 ACCOUNTS

There are three different accounts designed here, Account, SavingAccount and CheckingAccount. SavingAccount and CheckingAccount are extended from BankAccount.

- Account

It realizes basic properties such as balance, id and transaction times. Basic operations like deposit, withdraw, calculate monthly fee and print account information. Some methods is to be implemented in the extended classes.

- SavingAccount

It is extended from Account. When the end of the month comes, if transaction time is smaller or equal to 3 the balance of the SavingAccount will be reduced by \$ 3.

- CheckingAccount

It is extended from Account as well. When the end of the month comes, the user can choose to withdraw some money to avoid the monthly fee.

### 2.2 MORE OPERATIONS

After I finished the basic task I try to design a more interesting bank system by realizing a simple user interface.

- I add a static variable in Account to record how many accounts have been created. In this way can I choose which account I am manipulating.
- A user can choose to create a new account, change current account, withdraw, deposit, check account balance, check the monthly fee and list all accounts created.

### 2.3 ROBUSTNESS CONSIDERATION

In order to improve the robustness of user input we use regular expression to catch correct input. When the user chooses operation the input is limited to [1-9]. And if user needs to input money the input is limited to a decimal. If the input is out of bound or characteristics other than numbers the command line window will display respective warning message.

## 3 RESULTS

### 3.1 ENVIRONMENT

- Windows 10
- Java Development Kit 1.8.0\_131
- Eclipse

### 3.2 SCREENSHOTS OF THE RESULT

We use command line to compile and execute the program. The result is shown in Fig. 3.1.

```
F:\OS Project\AdvancedBank\src\com\bank>JAVA AdvancedBank

Press 1 -- create a new account;
Press 2 -- log in;
Press 3 -- withdraw;
Press 4 -- deposit;
Press 5 -- check balance;
Press 6 -- check the month;
Press 7 -- list all accounts created;
Press 8 -- quit.
1

Press 1 to create a Checking Account;
Press 2 to create a Saving Account.
2

Press 1 -- create a new account;
Press 2 -- log in;
Press 3 -- withdraw;
Press 4 -- deposit;
Press 5 -- check balance;
Press 6 -- check the month;
Press 7 -- list all accounts created;
Press 8 -- quit.
4

How much do you want to deposit?30
My saving number is 1, and my balance is $30.0.

Press 1 -- create a new account;
Press 2 -- log in;
Press 3 -- withdraw;
Press 4 -- deposit;
Press 5 -- check balance;
Press 6 -- check the month;
Press 7 -- list all accounts created;
Press 8 -- quit.
6

There are 1 transactions this month.
My saving number is 1, and my balance is $31.5.
```

Figure 3.1: Screenshot of an Advanced Bank System

### 3.3 THOUGHTS

Although realizing the additional functions consumes lots of time I feel a sense of accomplishment when finishing it. When I was learning C++ dealing with characters frustrated me a lot. However when I try to use regular expression in Java some problems seems to be addressed.