

Lab 12

Refactoring, File Input and Output

The following exercises are to be completed during lab class. If you do not have time to finish during lab, they must be completed before the beginning of the following lab session.

Set-Up

- Start files at T:/it168/Labs/Lab12
 - *Transactions.txt*
 - *BankAccount.java*
- Create a new project in your Eclipse workspace named: **Lab12**
- In the *src* folder, create a package named: **edu.ilstu**
- Import the *BankAccount.java* file into the package in your *src* folder.
- Import the input file, *Transactions.txt*, into the root of your project.

Problem

At the end of the day, all of the transactions that a bank customer has had for a day are processed. This is called a batch process - transactions are accumulated and processed at one time, often overnight when the computer is not being used for online transactions.

Some requirement changes have been requested, so your task is to refactor (modify) the existing `BankAccount.java` to meet the new requirements.

The data to process is to be read from a file called `Transactions.txt`. The file will have the account number, first name, and last name on the first three lines and the balance on the next line. That will be followed by each transaction on its own line. A transaction will have an integer code: **1** for withdrawal or **2** for deposit, and the amount (a double).

Example input file:

```
BJS15923
Barbara
Smith
2000.00
1  20.00
2  120.00
. . . (unknown number of transactions)
```

There are two types of output. Statements to the screen giving the user immediate feedback as to what processing has taken place. At the end of the program, the account number, first name, last name, and balance need to be written to a new file called "AccountRecord.txt". This output should be the same as the first four lines of the input file.

Sample output file

BJ515923

Barbara

Smith

500.0 (**Note:** *this is not the correct balance for the given input file. Part of a programmer's job is to verify the program produces the correct results.*)

Note: The balance should not have any formatting.

BankAccount class requirement changes

- processWithdrawal and validateWithdrawalAmount
 - Old requirements:
 - When a withdrawal is requested that results in an overdraft of the account, the transaction is denied.
 - Balance does not change
 - Message to user:
You do not have sufficient funds to withdraw \$2000.00
Request denied!
 - New requirements:
 - A withdrawal is always allowed.
 - Balance may become a negative number.
 - If the withdrawal causes an overdraft, an extra \$30 is deducted from the balance for the overdraft fee.
 - Message to user should now be:
You do not have sufficient funds to withdraw \$2000.00
\$30 overdraft charge has been applied
- Modifications will need to be made to the processWithdrawal and possibly the validateWithdrawalAmount method. The rest of the existing methods do not require any changes.
- Modify the existing constructor to accept all four instance variables and remove the setter methods.
- Add the constant for the overdraft charge.
- Add a toString method.
 - Format the toString method to the requirements for how to write to the output file.

BankAccount class diagram reflecting new requirements

BankAccount
<ul style="list-style-type: none">- OVERDRAFT_CHARGE:double- accountNumber:String- firstName:String- lastName:String- balance:double
<ul style="list-style-type: none">+ BankAccount(String accountNumber, String firstName, String lastName, double balance)+ processWithdrawal(double withdrawal):void- validateWithdrawalAmount(double withdrawal):boolean+ processDeposit(double deposit):void+ getAccountNumber():String+ getFirstName():String+ getLastName():String+ getBalance():double+ toString():String

BankDriver

Implement the BankDriver class you designed in pre-lab. Manually calculate the final balance using the data in the input file. Confirm the output written in the output file matches the result from your manual calculation.

To Be Submitted

The following files should be zipped together into a file called Lab12.zip and submitted to ReggieNet by the beginning of your next lab.

- BankAccount.java
- BankDriver.java