Object Oriented Programming Project: Banking System

Project Phase 1

Last Update: April 19, 2015

Group Members INFO-I 211 (Spring 2015)

Jeremiah McKinney Dr. Awny Alnusair

Ian Holston Indiana University Kokomo

Theral Jessop

Readme (First Draft)

**Updates and code descriptions**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Name | Description | Started /Continued /Completed |
| April 4, 2015 | Theral Jessop | Finished the mockup of the UML and gave it to Ian to create the graphics of it. I also started this Readme file. | Completed |
|  | Theral Jessop | Talked to Professor about the design and verbally worked through some issues that I was having with the design phase. I was fiannaly able to see the clear picture of how it should look on paper. | Completed |
|  | Theral Jessop | Updated Ian and Jeremiah on the design phases and additions and subtractions that would go into this project | Completed |
|  | Theral Jessp | Got the general layout coded. Established all the Classes, child classes, etc | Completed |
|  | Theral Jessop | Worked on coding the Banker (use) class | Started |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Date | Name | Description | Started /Continued /Completed |
|  |  |  |  |
| April 17, 2015 | Theral Jessop | Updated Banker: when entering a new customer it now tells you that you created a new customer whether you are entering from the customer or from the account. Also added error checking for when entering the balance. If it is a negative value it will keep asking you for the balance until it is a positive value. No negative balances are allowed when creating a new account! | Continued |
| April 17, 2015 | Theral Jessop | When testing and entering test data, I found a bug in withdrawal transactions where it was adding the amount that was entered to the transaction table but the actual amount should have been what the remainder that was left in the account. Fixed and changed the way the withdrawals happen. Changed Account, Checking, Gold, and Regular classes to reflect this change. | Continued |
| April 17, 2015 | Theral Jessop | Verified that the Deposits and Withdrawals are working correctly and recording according to the guidelines. Validated the balance after 11 deposits and 7 withdrawals and the balance was exactly to the penny and the transactions fees according the guidelines are calculating properly. Also validated the transations are being processed correctly now | Complete |
| April 18, 2015 | Theral Jessop | Economized the code and combined a lot of methods into one with conditional operations based on which option the user selects from the menu. | Complete |
| April 18, 2015 | Theral Jessop | Extensive testing and modifying code to be more efficient and reusable. This is a process of becoming good and coding it once. Sometimes it takes several time before we are able to make it right the first time. Hence the re-write | Complete |
| April 19, 2015 | Theral Jessop | Re-Write the eom calculations method to make it more efficient and track with better results | Started |
| Date | Name | Description | Started /Continued /Completed |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**End of Updates and code descriptions**

**Customer Class (concrete)**

**Fields**

#customerID: String

#customerName: String

Constructor

**Non-Default constructor**

+Customer(String customerID, String customerName)

**Methods**

+getCustomerID(): String

+getCustomerName(): String

+setCusomerID(String): void

+setCusomerName(String): void

+toString(): String

**Account Class (abstract)**

**Fields**

#accountNumber: String

#accountBalance: double

#numberOfTransactions: int

#customer: Customer

Constructor

**Non-Default constructor**

+Account(String accountNumber, double accountBalance, Customer customer)

sets the numberOfTransactions to 0

**Methods**

+getAccountNumber(): String

+getAccountBalance(): String

+getCustomer(): Customer

+getNumberOfTransactions: String

#setAccountNumber(String): void

#setAccountBalance(double): void

#setCustomer(Customer): void

-resetNumberOfTransactions(): String

#addTransaction(double, String): void

+abstract makeDeposit(double): String

+abstract makeWithdrawal(double): String

**Checking Class (concrete) extends Account**

**Fields**

#checkingTransactionFee: double

#checkingTransactionFeeAmount: double

Constructor

**Non-Default constructor**

+Checking( String accountNumber, double balance, Customer customer)

Super(accountNumber, balance, customer)

checkingTransactionFee = 3.00

checkingTransactionAmout = 0.0

**Methods**

+getCheckingTransactionFee(): String

+getCheckingTransactionFeeAmount(): String

#setCheckingTransactionFee(double): String

+eomCalculations(): String

Note: A transaction is any activity that changes the balance of an account, therefore, the first **two** transactions are free, all transactions after will be charged a $3 fee.

end of note

+makeDeposit(double): String

Increment the numberOfTransactions by 1

Deposit funds

+makeWithdrawal(double): String

Increment the numberOfTransactions by 1

If funds are below the amount being asked for

make withdrawal of available funds

otherwise

make withdrawal

+toString(): String

**Gold Class (concrete) extends Account**

**Fields**

#goldInterestAmount: double

#goldInterestRate: double

Constructor

**Non-Default constructor**

+Gold(String accountNumber, double balance, Customer customer)

Super(accountNumber, balance, customer)

goldInterestAmount = 0

goldInterestRate = 5.0

**Methods**

+getGoldInterestAmount(): String

+getGoldInterestRate(): String

Validate rate as a % not as a decimal

#setGoldInterestRate(double): String

+eomCalculations(): String

+makeDeposit(double): String

Increment the numberOfTransactions by 1

Deposit funds

+makeWithdrawal(double): String

Increment the numberOfTransactions by 1

Withdrawal funds no matter what the balance

+toString(): String

**Regular Class (concrete) extends Account**

**Fields**

#regularInterestRate: double

#regularFixedCharge: double

constructor

**Non-Default constructor**

Regular(String accountNumber, double balance, Customer customer)

Super(accountNumber, balance, customer)

regularInterestRate = 6.0

regularFixedCharge = 10.0

**Methods**

+getRegularInterestRate(): String

Validate rate as a % not as a decimal

+getRegularFixedCharge(): String

#setRegularInterestRate(double): String

#setRegularFixedCharge(double): String

+eomCalculations(): String

+makeDeposit(double): String

Increment the numberOfTransactions by 1

Deposit funds

+makeWithdrawal(double): String

Increment the numberOfTransactions by 1

Withdrawal funds no matter what the balance

+toString(): String