Object Oriented Programming Project: Banking System

Project Phase 1

Last Update: April 17, 2015

by

Theral Jessop

Readme (First Draft)

**Updates and code descriptions**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Name | Description | Started / Completed |
|  |  | ToDO List: I need to change all the Classes to the Payroll system. |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Date | Name | Description | Started / 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