Object Oriented Programming Project: Banking System

Project Phase 1

Last Update: April 2, 2015

Group Members INFO-I 211 (Spring 2015)

Jeremiah McKinney Dr. Awny Alnusair

Ian Holston Indiana University Kokomo

Theral Jessop

Readme (First Draft)

**Customer Class (concrete)**

Constructor

**Non-Default constructor**

+Customer(String customerID, String customerName)

**Fields**

#customerID (String)

#customerName (String)

**Methods**

+getCustomerID() (return String)

+getCustomerName() (return String)

+setCusomerID(String customerID) (void)

+setCusomerName(String customerName) (void)

+toString() (return String)

**Account Class (abstract)**

Constructor

**Non-Default constructor**

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

**Fields**

#accountNumber (String)

#balance (double)

#customer (Customer)

**Methods**

+getAccountNumber() (return String)

+getBalance() (return double)

+getCustomer() (return Customer)

+setAccountNumber(String accountNumber) (void)

-setBalance(double balance) (void)

+setCustomer(Customer customer) (void)

+deposit() (abstract)

+withdrawal() (abstract)

**Transaction Class (concrete)**

Constructor

**Non-Default Constructor**

+Transaction(String transactionDescription, int transactionNumber, Account account)

**Fields**

#transactionDescription (String)

#transactionNumber (int) = 1

#account (Account)

**Methods**

+getTransactionDescription() (return String)

+getTranactionNumber() (return String)

+getAccount() (return Account)

+setTransactionDescription(String transactionDescription) (void)

+setTransactionNumber(int transactionNumber) (void)

+setAccount(Account account) (void)

-generateUniqueTranNumber() (returns int)

+toString() (return String)

**Checking Class (concrete) extends Account**

Constructor

**Non-Default constructor**

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

Super(accountNumber, balance, customer)

**Fields**

#transactionFees (double)

#numberOfTransactions = 1

**Methods**

+getTransactionFees() (return String)

+setTransactionFees(double transactionFees) (void)

First two transactions are free then there is a $3 fee for every one after that

+deposit() (return String)

If numberOfTransactions > 2 add $3 fee

Deposit funds

+withdrawal() return String)

If numberOfTransactions > 2 add $3 fee

make withdrawal of available funds

+toString() (return String)

**Gold Class (concrete) extends Account**

Constructor

**Non-Default constructor**

+Gold(double interest, double interestRate, String accountNumber, double balance,

Customer customer)

Super(accountNumber, balance, customer)

**Fields**

#interest (double)

#interestRate (double) = 5%

**Methods**

+getInterest() (return String)

+getInterestRate() (return String)

#setInterest() (return String)

#setInterestRate() (return String)

+deposit() (returns String)

Deposit funds

+withdrawal() (returns String)

Withdrawal funds no matter what the balance

+toString() (return String)

**Regular Class (concrete) extends Account**

constructor

**Non-Default constructor**

Regular(double interest, double interestRate, String accountNumber, double balance,

Customer customer)

Super(accountNumber, balance, customer)

**Fields**

#interest (double)

#interestRate (double) = 6%

**Methods**

+getInterest() (return String)

+getInterestRate() (return String)

#setInterest() (return String)

#setInterestRate() (return String)

+deposit() (returns String)

Deposit funds

+withdrawal() (returns String)

Withdrawal funds no matter what the balance

+toString() (return String)