UML Class Diagram (Bank of 3331)

<<interface>> Printable

+ print(ArrayList<Customer>:custList):void
+ reportLogin(String:firstName,String:lastName):void
+ reportLogout(String:firstName,String:lastName):void

<> Person

- firstName:String
- lastName:String
- dateOfBirth:String
- address:String
- phoneNumber:String

+ getFirstName():String
+setFirstName(String:firstName):void
+getLastName():String
+setLastName(String:lastName):void
+ getDateOfBirth():String
+setDateOfBirth(String:dateOfBirth):void
+getAddress():String
+setAddress(String:address):void
+getPhoneNumber():String
+setPhoneNumer(String:phoneNumber):void

Customer

- idNumber:int
- checkingAccount:Checking
- savingsAccount:Savings
- creditAccount:Credit
- password:String

+ getIdNumber():int
+ setIdNumber(int:idNumber):void
+ getCheckingAccount():Checking
+ setCheckingAccount(Checking:checkingAccount):void
+getSavingsAccount():Checking
+setSavingsAccount(Savings: savingsAccount):void
+getCreditAccount():Credit
+setCreditAccount(Credit: creditAccount)
+getPassword():String
+setPassword(String: password)

<> Account

- accountNumber:int
- startingBalance:double

+ getAccountNumber():int
+setAccountNumber(int:accountNumber):void
+ getStartingBalance():double
+setStartingBalance(double:startingBalance):void
+transferFunction(double:money, Account:receiver):void
+withdrawFunction(double:money):void
+depositFunction(double:money):void

Checking

Savings

Credit

- creditMax():int

+depositFunction(double:money):void
+withdrawFunction(double:money):void
+transferFunction(double:money, Account:receiver):void
+setCredMax(int:credMax):void
+getCredMax():int

Menu

+ printBankIntro():void
+ printUserChoice():void
+ printMainDirectory():void
+ printManagerMainDirectory():void
+ printAccountSelectionAll():void
+ printAccountSelectionNoAll():void
+ printAccountSelectionNoAllNoCredit():void
+ printAccountSelectionNoAllNoChecking():void
+ printAccountSelectionNoAllNoSavings():void
+ printManagerInquirySelection():void
+ printManagerUserCreationSelection():void
+ reportLogin():void
+ reportLogout():void
+ print(ArrayList<Customer>:custList):void

Logger

+ print(ArrayList<Customer>:custList):void
+ exitProgramLog():void
+ reportLogin(String:firstName, String:lastName):void
+ reportLogout(String:firstName, String:lastName):void
+ inquireBalanceLog(Customer:customerObject, String:accountFlag):void
+ depositLog(Customer:customerObject, String:accountFlag):void
+ withdrawLog(Customer:customerObject, String:accountFlag, double:withdrawFunds,double:initial):void
+ transferLogFromChecking(Customer:customerObject, String:accountFlag,double:initialBal1, double initialBal2, double:transferAmount):void
+ transferLogFromSavings(Customer:customerObject, String:accountFlag, double:initialBal1, double:initialBal2, double:transferAmount):void
+ payLog(Customer:customerObject1, Customer:customerObject2, String:accountFlag, double:initialBal1, double:initialBal2, double:payAmount):void

UserConsole

+ inquireBalanceChoice(Customer:customerObject):void
+ depositChoice(Customer:customerObject):void
+ withdrawChoice(Customer:customerObject):void
+ transferChoice(Customer:customerObject):void
+ transferChoiceNest1(Customer:customerObject):void
+ transferChoiceNest2(Customer:customerObject):void
+ payNest(Customer:customerObject1, Customer:customerObject2):boolean
+ customerChoice(ArrayList<Customer>:customerObject):void
+ managerChoice(ArrayList<Customer>:customerObject):void
+ managerChoiceNest1(ArrayList<Customer>:customerObject):void
+ managerChoiceNest2(ArrayList<Customer>:customerObject):void
+ managerChoiceInquiryByAccNest(ArrayList<Customer>:customerObject):void
+ findMaxChecking(ArrayList<Customer>:bank):int
+ findMaxSavings(ArrayList<Customer>:bank):int
+ findMaxCredit(ArrayList<Customer>:bank):int
+ findMaxId(ArrayList<Customer>:bank):int
+ findIndex(String[]:titles,String:input):int
+ transactionActions(String:fileName,ArrayList<Customer>:bank):
+ findAccountByName(ArrayList<Customer>:custList,String:custName):Customer
+ findAccountById(ArrayList<Customer>:custList,int:idNum):Customer
+ findIndex(String[]:titles,String:input):int
+ fileRead(String:file):ArrayList<Customer>
+ generateUpdatedBalanceSheet(ArrayList<Customer>:customers):void

BankStatement

1...*

+ writeStatement():void

1...* manager creates statement

1

1