Reader Discretion

The contents of this document only reveal the documentation string(s) of attributes and methods that serve great important in the completion of ***Project 1: My Banking System.***

For the Main package, please refer to the files inside the Main package. They have their own set of documentation strings for you to read and understand.

Created by

Neil John J. Jomaya

DocString Table of Contents

Account Class

An abstract account class that has comparators to compare itself with different account objects.

Attributes

Data Type and Name	***Description***
Bank bank	A constant bank object associated to this account.
String accountNumber	Account number of this account object. Cannot be modified once set.
ArrayList<Transaction> transactions	Transactions refer to the transaction logs recorded in this account. A transaction is logged based on the following: 1. A successful withdrawal. 2. A successful deposit. 3. A successful payment. 4. A successful fund transfer.

Methods

Method Name	***Description***
addNewTransaction(String, Transaction.Transactions, String)	Add a new transaction log to this account. Params: account – Account number of source account that triggered this transaction. type – Type of transaction triggered. description – Description of the transaction.
getTransactionsInfo()	Get all information for every transaction that has been logged into this account.

Savings Account Class

Methods

Method Name	*Description*
hasEnoughBalance(double)	Validates whether this savings account has enough balance to proceed with such a transaction based on the amount that is to be adjusted. Params: amount – Amount of money to be supposedly adjusted from this account's balance. Returns: Flag if transaction can proceed by adjusting the account balance by the amount to be changed.
transfer(Account, double)	Transfers an amount of money from this account to another savings account. Is extensively used by the other transfer() method. Params: account – Account number of recipient. amount – Amount of money to be supposedly adjusted from this account's balance. Returns: Flag if fund transfer transaction is successful or not. Throws: IllegalAccountType – Cannot fund transfer when the other account is of type CreditAccount.
transfer(Bank, Account, double)	Transfers an amount of money from this account to another savings account. Should be used when transferring to other banks. Params: bank – Bank object of the recipient. account – Account number of recipient. amount – Amount of money to be supposedly adjusted from this account's balance. Returns: Flag if fund transfer transaction is successful or not. Throws: IllegalAccountType – Cannot fund transfer when the other account is of type CreditAccount.
cashDeposit(double)	Deposit some cash into this account. Cannot be greater than the bank's deposit limit.

	Params: amount – Amount of money to be deposited.
withdrawal(double)	Withdraw an amount of money from this savings account. Cannot proceed if account does not have sufficient balance. Params: amount – Amount of money to be withdrawn.
getAccountBalanceStatement()	Get the account balance statement of this savings account. Returns: String balance statement.
insufficientBalance()	Warns the account owner that their balance is not enough for the transaction to proceed successfully.
adjustAccountBalance(double)	Adjust the account balance of this savings account based on the amount to be adjusted. If it results to the account balance going less than 0.0, then it is forcibly reset to 0.0. Params: amount – Amount to be added or subtracted from the account balance.

Credit Account Class

Methods

Method Name	*Description*
getAccountBalanceStatement()	Loan statement of this credit account. Returns: String loan statement.
canCredit(double)	Checks if this credit account can do additional credit transactions if the amount to credit will not exceeded the credit limit set by the bank associated to this Credit Account. Params: amountAdjustment – The amount of credit to be adjusted once the said transaction is processed. Returns: Flag if this account can continue with the credit transaction.
adjustLoanAmount(double)	Adjust the owner's current loan. Result of adjustment cannot be less than 0. Params: amountAdjustment – Amount to be adjusted to the loan of this credit account.
pay(Account, double)	Pay an amount of money to a selected account. Such an account cannot be of type CreditAccount. Params: account – Target account to pay money into. Returns: True if pay transaction was successful. False otherwise. Throws: IlegalAccountType – Credit Accounts cannot pay to other Credit Accounts as they do not hold account money balance but rather, credits. As such, in this scenario, only Savings Accounts can receive payment from Credit Accounts.
recompense(double)	Recompense some amount of money to the bank and reduce the value of loan recorded in this account. Must not be greater than the current credit. Params: amount – Amount of money to be recompensed. Returns: Flag if compensation was successful.

Bank Class

Attributes

Data Type and Name	***Description***
double depositLimit	The amount of money each Savings Account registered to this bank can deposit at every transaction. Defaults to 50,000.0
double withdrawLimit	The amount of money withdrawable / transferrable at once, restricted to every Savings Account registered to this bank. Defaults to 50,000.0
double creditLimit	Limits the amount of credit or loan that all Credit Accounts, registered on this bank, can handle all at once. Defaults to 100,000.0
double processingFee	Processing fee added when some transaction is involved with another bank. Cannot be lower than 0.0. Defaults to 10.00

Methods

Method Name	*Description*
showAccounts(Class<T>)	Show accounts based on option. Params: accountType – Type of account to be shown.
getBankAccount(Bank, String)	Get the Account object (if it exists) from a given bank. Params: bank - Bank to search from. accountNum – Account number of target account.
createNewAccount()	Handles the processing of inputting the basic information of the account. Returns: Array list of Field objects, which are the basic account information of the account user.
createNewCreditAccount()	Create a new credit account. Utilizes the *createNewAccount()* method. Returns: New credit account.
createNewSavingsAccount()	Create a new savings account. Utilizes the createNewAccount*()* method.
addNewAccount(Account)	Adds a new account to this bank, if the account number of the new account does not exist inside the bank. Params:

	account – Account object to be added into this bank.
accountExists(Bank, accountNum)	Checks if an account object exists into a given bank based on some account number. Params: bank – Bank to check if account exists. accountNum – Account number of target account to check.

Inner Classes

Inner Class Name	*Description*
BankComparator	A comparator that compares if two bank objects are the same.
BankIdComparator	A comparator that compares if two bank objects have the same bank id.
BankCredentialsComparator	A comparator that compares if two bank objects share the same set of credentials.

Bank Launcher

A class primarily used for interacting with the bank module.

Attributes

Data Type and Name	***Description***
ArrayList<Bank> banks	List of banks currently registered in this session.
Bank loggedBank	The Bank object currently logged in. Null by default, or when no bank is currently logged in.

Methods

Method Name	*Description*
bankInit()	Bank interaction initialization. Utilized only when logged in.
showAccounts()	Show the accounts registered to this bank. Must prompt the user to select which type of accounts to show: (1) Credit Accounts, (2) Savings Accounts, and (3) All.
newAccounts()	Bank interaction when creating new accounts for the currently logged in bank.
bankLogin()	Bank interaction when attempting to login to the banking module using a bank user's credentials.
setLogSession(Bank)	Creates a new login session for the logged in bank user. Sets up a new value for the *loggedBank* field. Params: b – Bank user that successfully logged in.
logout()	Destroys the login session for the current user.
createNewBank()	Creates a new bank record. Utilized separately from the rest of the methods of this class. Throws: NumberFormatException – May happen when inputting deposit, withdraw, and credit limit, and processing fee.
showBanksMenu()	Output a menu of all registered or created banks in this session.
addBank(Bank)	Adds new bank to array list of banks. Params: b – Bank object to be added.

getBank(Comparator<Bank>, Bank)	Checks if a bank exists based on some criteria. Params: bankComparator – Criteria for searching. bank – Bank object to be compared. Returns: Bank object if it passes the criteria. Null if none.
findAccount(String)	Finds the Account object based on some account number on all registered banks. Params: accountNumber – Account number of target Account. Returns: Account object if it exists. Null if not found.
bankSize()	Get the number of currently registered banks.

Account Launcher

A class primarily used for interacting with the account module.

Attributes

Data Type and Name	***Description***
Account loggedAccount	Account object of logged account user.
Bank assocBank	Selected associated bank when attempting to login in the account module.

Methods

Method Name	*Description*
isLoggedIn()	Verifies if some account is currently logged in.
accountLogin()	Login an account. Bank must be selected first before logging in. Account existence will depend on the selected bank.
selectBank()	Bank selection screen before the user is prompted to login. User is prompted for the Bank ID with corresponding bank name. Returns: Bank object based on selected ID.
setLogSession(Account)	Create a login session based on the logged user account. Params: account – Account that has successfully logged in.
destroyLogSession()	Destroy the log session of the previously logged user account.
checkCredentials(String, String)	Checks inputted credentials during account login. Params: accountNum – Account number. pin – 4-digit pin. Returns: Account object if it passes verification. Null if not.

Credit Account Launcher

Methods

Method Name	*Description*
creditAccountInit()	Method that deals with all things about credit accounts. Mainly utilized for showing the main menu after Credit Account users log in to the application.
creditPaymentProcess()	Method that is utilized to process the credit payment transaction.
creditRecompenseProcess()	Method that is utilized to process the credit compensation transaction.
getLoggedAccount()	Get the Credit Account instance of the currently logged account.

Savings Account Launcher

Methods

Method Name	*Description*
savingsAccountInit()	Method that deals with all things about savings accounts. Mainly utilized for showing the main menu after Savings Account users log in to the application.
depositProcess()	A method that deals with the deposit process transaction.
withdrawProcess()	A method that deals with the withdraw process transaction.
fundTransferProcess()	A method that deals with the fund transfer process transaction.
getLoggedAccount()	Get the Savings Account instance of the currently logged account.