

DOCUMENTATION FOR BANK ACCOUNT MANAGEMENT SYSTEM

- Written by Student ID: 24110073 –

I. OOA Analysis

Classes / nouns:

- Transaction
- Account
- SpecializedAccount
- Customer

Object's attributes / Descriptive nouns:

- Transaction:
 1. Amount
 2. Date
 3. Type
 4. getCurrentTime
- Account:
 1. accountNumber
 2. balance
 3. ownerName
 4. transactionHistory (vector)
- SpecializedAccounts (inherited from account class with 3 new attributes):
 1. interestRate
 2. withdrawalCount
 3. withdrawalLimit
- Customer:
 1. name
 2. id
 3. accounts (vector)

Methods (Verbs):

- Transaction:
 1. getAmount
 2. getType
 3. getDate (get transaction date/ time command was issued on your local time)
- Account:
 1. Deposit / Withdrawal operators overload (operator+=/=-)

- 2. Additional virtual Withdrawal method (later overridden by SpecializedAccount)
 - 3. getAccountNumber
 - 4. getBalance
 - 5. getOwnerName
 - 6. printHistory (prints out transaction history)
- SpecializedAccounts:
 - 1. countInterest (adds interest to balance)
 - 2. Withdrawal (overrides the method in account class)
- Customer:
 - 1. addAccount (assign account)
 - 2. printAccount (prints out list of owned accounts)
 - 3. getTotalBalance (calculate total balance across accounts)
 - 4. displayInfo (self explanatory)

II. Inheritance and operator overload

- SpecializedAccounts inherited attributes from the Accounts class (accountNumber, balance, ownerName, transactionHistory) with 3 extra attributes (interestRate, withdrawalCount, withdrawalLimit). The class overrides the Withdrawal method displaying different withdrawal info.
- Operator overloading was used in Account class as deposit/withdrawal methods making the code feel more “natural” and readable compared to the standard method.

III. Code walkthrough

1. Transaction

- Represents transaction info.
- Has basic attributes (type, date and getCurrentTime).

2. Account

- Represents regular bank accounts.
- Contains basic attributes (accountNumber, balance, ownerName, transactionHistory)
- Methods allowing account owner to withdraw and deposit money and saves transaction history into a list.

3. SpecializedAccounts

- Represents saving accounts with additional interest.
- Inherit attributes from the account class.

- Overrides Withdrawal method.

4. Customer

- Customer using the service.

- Basic attributes (name, id, account list).

- Allows user to create an account, display accounts owned and total balance across them.

Key parts of the code:

- Transaction stores the transaction type and info.
- Account class carry out simple actions to deposit and withdraw.
- SpecializedAccounts includes interest rate and withdraw limit.
- Customer has their basic info and owned accounts info.

IV. Test results

The results correctly displayed the customer's basic info and owned accounts info, it also allowed to them to create and assign accounts to the respective owner while also correctly performed the withdraw/deposit functions and printing out transaction history list as a list. Unfortunately, I wasn't able to figure out printing out transaction date/time.

```
Customer: Hugo Vlad, ID: C1001
Accounts owned:
Account: HUG01, Owner: Hugo, Balance: 0
Account: HUG02, Owner: Hugo, Balance: 0
Account: HUGOSP1, Owner: Hugo, Balance: 0
Total Balance: 0
Customer: Vivian Banshee, ID: C1002
Accounts owned:
Account: VIVIAN1, Owner: Vivian, Balance: 0
Total Balance: 0
Deposited: 300, Current balance: 300
Deposited: 200, Current balance: 200
Withdrawn: 100, Current balance: 200
Deposited: 400, Current balance: 400
HUGOSP1 transaction history:
Deposit: 400
80 of interest applied. Balance: 480
Final Balance:200
Final Balance:200
Final Balance:480
Deposited: 400, Current balance: 400
Withdrawn: 50, Current balance: 350
VIVIAN1 transaction history:
Deposit: 400
Withdrawal: 50
Final Balance:350
Customer: Hugo Vlad, ID: C1001
Accounts owned:
Account: HUG01, Owner: Hugo, Balance: 200
Account: HUG02, Owner: Hugo, Balance: 200
Account: HUGOSP1, Owner: Hugo, Balance: 480
Total Balance: 880
Customer: Vivian Banshee, ID: C1002
Accounts owned:
Account: VIVIAN1, Owner: Vivian, Balance: 350
Total Balance: 350
```

V. LLM Usage

The majority of the code was handwritten. However, some of the functions was made possible with the help of chatGPT. I used chatGPT to assist with getting transaction time by prompting "Suggest implementations for the time the transaction was made" and transaction history by prompting "Suggest me ideas of how to store transaction history in a list and print it out" along with creating multiple bank accounts by prompting "Give out main ideas of how do I implement a list of owned accounts to the Customer class and print total balance out?".