Sagyou Jikan
作業時間
Banking Application Backend Design Document

The program takes in two inputs, the name of the master file as the first argument, and the name of the transaction file as the second argument.
The program outputs a new master file at the end.

Transaction

Method	Description
Transaction(transCode : byte,transName : String, transId: int, value: float, transName: String)	Default constructor of the transaction class. Initializes all member fields of the transaction objects with the parameters in the function signature.
getTransCode(): byte	Gets the transaction code from the transaction
getTransName(): String	Gets the name of the account holder involved in the transaction
GetTransId(): int	Gets the id of the account involved in the transaction
GetValue(): float	Gets the funds involved in the transaction
GetMisc(): String	Gets and miscallaneous information involved in the transaction

TransactionIO

Method	Description
TransactionIO(filename: String)	Default constructor of the TransactionIO class. Sets the transaction file to read transactions from.
ReadFile(): Vector<Transaction>	Reads the file containing the day's transactions and returns a vector containing the transactions in the order that they were executed.

Account

Method	Description
Account(id: int, name: String, balance: float, enabled: boolean, student: boolean, count: int)	Default constructor of the Account class. Initializes all the field members of the account object with the parameters in the function signature.
GetId(): int	Get the id of the account
GetName(): String	Get the name of the account holder
GetBalance(): float	Get the balance on the account
SetBalance(balance: float): void	Change the balance of the account
IsEnabled(): boolean	Check if the account is enabled
SetEnabled(enabled: boolean): void	Set the flag of the account being enabled
IsStudent(): boolean	Check if the account has a student plan
SetStudent(student: boolean): void	Set the flag of the account having a student plan

GetCount(): int	Get the ammount of total transactions on the file
IncCount(): void	Increment the transactions on the file

AccountIO

Method	Description
AccountIO(filename: String)	Default constructor of the AccountIO class. Sets the master bank account file to read from.
ReadFile(): Map<int,Account>	Reads in the master bank account file
WriteFile(accounts: Map<int, Account>): void	Writes modifed accounts to the master and current accounts file

TransactionProcessor

Method	Description
TransactionProcessor(masterFile:String, transactionFile:String):void	Default constructor of the TransactionProcessor class. Initializes accounts and transactions
Process():void	Processes transactions and applies them to accounts
changeBalance(increase:boolean, accountId:int , value:float):void	Changes the balance of an account
transfer(accountId1:int, accountId2:int,value: float):void	Transfer funds from the first account to the second account
paybill(accountId:int, company:String,value: float):void	Checks if company being paid to is valid, if so pays bill
create(trans:Transaction):void	Creates an account
delete(accountId:int):void	Deletes an account
enable(willEnable:boolean,accountId: int):void	Enables or disables an account depending on the boolean
changeplan(accountId:int):void	Changes the plan of an account

Bank

AccountIO

- filename: String

+ accounts: Vector

+ readFile(): Vector<Account>

+ writeFile(Accounts: map): void

TransactionsIO

- filename: String

+ transactions: Vector

+ readFile(): Vector<Transaction>

Transaction

-transCode: Byte

-transName: String

-transId: Integer

-value: Float

-misc: String

+getTransCode(): Byte

+getTransName(): String

+getTransId(): Integer

+getValue(): Float

+getMisc(): String

Account

- balance: float

- enabled: boolean

- student: boolean

- name: string

- id: int

-count: int

+getId(): int

+getName(): String

+getBalance(): Float

+setBalance(balance: float) : float

+isStudent(): Boolean

+setStudent(student: Boolean): void

+isEnabled(): Boolean

+setEnabled(enabled: Boolean): void

+getCount(): int

+incCount(): void

TransactionProcessor

- accounts: Map<Int,Account>

- transactions: Vector<Transaction>

- admin: boolean

- process(): void

- changeBalance(boolean, int, float) : void

- transfer(int, int, float):void

- paybill(int, string, float): void

- create(Transaction): void

- remove(int): void

- enable(boolean, int) : void

- changeplan(int):void