

# - username: String - password: String + Person(username: String, password: String) - getUsername(): String - changePassword(newPassword: String): void + displayOptions(): void + action(option: String): void

### Customer

- ~ chequingAccounts: ArrayList<Chequing>
- ~ savingAccounts: ArrayList<Saving>
- ~ linesCreditAccounts: ArrayList<Line>
- ~ creditAccounts: ArrayList<Credit>
- ~ primaryChequing: Chequing
- + Customer(username: String, password: String)
- ~ setPrimary(primary: Chequing): void
- requestAccount(AccountType: String): void
- getNetTotal(): double
- ~ rewriteWholeFile(): void
- getChequingInfo(): void
- getSavingInfo(): void
- getLineInfo(): void
- getCreditInfo(): voidgetUserInfo(): void
- + displayOptions(): void
- + action(option: String): void

# Manager

- chequingNum: int
- lineNum: int
- savingNum: int
- creditNum: int
- manager: Manager
- ~ requests: ArrayList<String>
- Manager(username: String, password: String)
- ~ getManager(username: String, password: String): void
- ~ setChequingNum(chequingNum: int): void
- ~ setLineNum(lineNum: int): void
- ~ setSavingNum(savingNum: int): void
- ~ setCreditNum(creditNum: int): void
- addCustomer(username: String): void
- addChequingAccount(username: String, type: String): void
- addSavingAccount(username: String, type: String): void
- addLineAccount(username: String, type: String): void
- addCreditAccount(username: String, type: String): void
- addAccount(username: String, type: String): void
- undoTransaction(transaction: Transaction): void
- ~ writeFile(): void
- addMoneyToATM(c5: int, c10: int, c20: int, c50: int): void
- verifyRequest(index: int): void
- + displayOptions(): void
- + action(option: String): void

# Transaction

- money: double
- receivingAccount: Account
- sendingAccount: Account
- final transactionTime: Date
- ~ isUndo: boolean
- + Transaction(money: double, receiving: Account, sending: Account, date: Date)
- + Transaction(money: double, Account: Account, isDeposit: boolean, date: Date)
- ~ undo(): void
- ~ getMoney(): double
- ~ getReceivingAccount(): Account
- ~ getSendingAccount(): Account
- + toString(): String
- ~ show(): String

## ATM

- $\sim \underline{cash5}$ : int
- $\sim \underline{cash10}$ : int
- $\sim \underline{cash\ 20}$ : int
- $\sim \underline{cash\ 50}$ : int
- $\sim \frac{1}{\text{total}}$ : int
- $\sim \overline{\text{date}}$ : Date
- ~ format: SimpleDateFormat
- <u>ATM</u>: ATM
- ATM()
- + getATM(): ATM
- ~ checkATMBalance(): boolean
- ~ writeFile(): void
- withdrawCash(money: int): int[]
- ~ withdraw(money: int): void
- ~ deposit(cashFive: int, cashTen: int, cashTwenty: int, cashFifty: int): void

# Main Class

# Simulator

- setDate(): void
- setSystemDate(): void
- initialize(): void
- <u>login()</u>: Person
- writeFile(): void
- + \_\_main\_\_(args: String[]): void

# Helper Class

# ReadFile

- ~ readFile (filename: String): ArrayList<String>
- ~ readFolder (folderName: String): ArrayList<ArrayList<String>>

# WriteFile

- ~ write(content: String, filename: String): void
- ~ clearInfo(fileName: String): void

# Login

~ <u>loginPass</u>(username: String, password: String): Person

# **Data Class**

## Accounts

- ~ accounts: ArrayList<Account>
- addAccount(a: Account): void
- ~ constructAccount(accountInfo: String[], owner: Customer): void
- ~ accountFinder(accNum: String): Account
- ~ <u>updateTransaction()</u>: void
- checkDuplication(acc: Account): boolean
- ~ depositAll(): void

# Users

- ~ manager: Manager
- ~ <u>folder</u>: ArrayList<ArrayList<String>>
- ~ <u>customers</u>: ArrayList<Customer>
- ~ constructManager(): void
- <a href="mailto:constructCustomer">constructCustomer</a>(fileContent: ArrayList<String>): void
- addCustomer(customer: Customer): void
- ~ getFolder(): ArrayList<ArrayList<String>>
- ~ getCustomerList(): void
- ~ customerFinder(username: String): Customer