**Bank Account Program**

­­

Stores the starting time.

Stores the ending time.

Creates StopWatch object and starts stop watch.

Starts stop watch.

Stops stop watch.

Returns the elapsed time (it invokes after stop ()).

Returns the elapsed time (it can invoke before stop () to check elapsed time)

1

1

1..\*

1

Date of transaction.

Type of transaction

Withdraw/deposit amount.

Balance after transaction.

Creates Transaction object with specified.

Returns withdraw amount if type is ‘W’.

Returns deposit amount if type is ‘D’.

Returns the all details of transaction.

Prints all transactions detail of an account.

Prints the horizontal dashes (-).

**Transaction**

- date: Date

- type: char

- amount: double

- balance: double

Transaction (type: char, amount: double, balance: double)

- getWithdraw (): double

- getDeposit (): double

+ toString (): String

+ printTransactionsDetail (list: ArrayList<Transacation>): void

- printHorizontadashes (withLen: int, depLen: int, balLen: int): void

\* Saving Account has some different features, when we create account with certain balance, balance is increased by 5% of current balance after every 30 seconds until program runs. To increment balance after 30 seconds duration a **StopWatch** object is used.

**StopWatch**

- startTime: long

- endTime: long

StopWatch ()

+ start (): void

+ stop (): void

+ elapsedTime (): double

+ checkElapsedTime (): double

\* Checking Account have overdraft feature.

Creates CheckingAccount object with specified id and balance.

Withdraws amount from balance (overdraft of 30% is possible).

**CheckingAccount**

CheckingAccount (balance: double, name: String)

+ withdraw (amount: double): void <<override>>

**SavingAccount**

- timer: StopWatch

- timeInterval: double

SavingAccount (balance: double, name: String)

+ getBalance (): double <<override>>

+ withdraw (amount: double): void <<override>>

+ deposit (amount: double): void <<override>>

- savingAccountInterest (): void

StopWatch’s object to check elapsed time

Contains time interval for increment

Creates SavingAccount object with specified id and balance.

Returns balance.

Withdraws amount from balance (withdraw limit is 20%).

Deposits amount from balance.

Increments by 5% of current balance every after 30 second

The id of account.

The Balance in account.

Name of Account holder.

Holds all transaction made by account holder.

Array which holds all type of accounts.

Counter of created accounts.

Creates Account object with specified id and name.

Returns balance.

Withdraws amount from balance.

Deposits amount from balance.

Returns the account holder’s name, id and balance.

Return boolean but its actual task is to print accounts’ menu in which all types of accounts are shown and creates user desired account.

Add newly created account in **accounts** array.

Returns **accounts** array.

**Account**

# id: int: {read only}

# balance: double

# name: String

# transaction: Transaction []

- accounts: Account []

- numberOfAccounts: int

Account (balance: double, name: String)

+ getBalance (): double

+ withdraw (amount: double): void

+ deposit (amount: double): void

+ toString (): String

+ accountsMenu (): boolean

+ addAccount (a: Account): void

+ getAccountsArray (): Account []