*Public Interface for Transaction class*

|  |  |  |
| --- | --- | --- |
| *Task* | *Method* | *Return Value* |
| *Constructs transaction* | *\_\_init\_\_(tNumber, tType = “”, amount = 0.0 )* | *Transaction object* |
| *“==”* | *\_\_eq\_\_(other)* | *boolean* |
| *“!=”* | *\_\_ne\_\_(other)* | *boolean* |
| *“+”* | *\_\_add\_\_(other)* | *sum* |
| *“-”* | *\_\_sub\_\_(other)* | *diffrence* |
| *sum( )* | *\_\_radd\_\_(other)* | *sum* |
| *Get the amount from the transaction* | *getAmount()* | *amount* |
| *Get the date the transaction was made* | *getDate( )* | *date* |
| *Get the transaction’s number* | *getTNumber( )* | *tNumber* |
| *Get the transaction’s type* | *getTType( )* | *tType* |
| *Prints the transaction’s values* | *printTransaction( )* |  |
| *Converts the transaction into a human readable format* | *\_\_str\_**\_( )* | *string* |
| *Converts the transaction into a machine readable format* | *\_\_repr\_**\_( )* | *string* |
| *Calls the system to get the current date and sets the date for the transaction* | *\_setDate( )* | *None* |

*Class responsibilities for Transaction class:*

* Know
  + Transaction number
  + Transaction type
  + Transaction amount
  + Date of the transaction
    - Day
    - Month
    - Year
* Do
  + Compare if equal or not equal
  + Add or subtract
  + Find sum
  + Get transaction values
    - Amount
    - Date
    - Transaction type
    - Transaction number
  + Print transaction values
  + Create human readable string
  + Create machine readable string
* Notes
  + \_typeSet: private list of possible transaction types
  + \_setDate(): private helper function

**Class Definitions**

Class Name: *Transaction*

Data:

tNumber int – Transaction number

tType str – Transaction type (“deposit”, “withdrawal”, “interest”, “transfer”, “penalty”)

amount double – amount of the transaction

day int – day of the month (1-31)

month int – month of the year (1-12)

year int – year (>= 2022)

Queries (Accessor):

getAmount() returns the transaction’s amount

getDate() returns the transaction’s date

getTNumber() returns the transactions’s number

getTType() returns the transaction’s type

printTransaction() prints the transaction’s day, month, year, amount, type and number as a single string

Commands (Mutator):

**None**

*Public Interface for Bank Account class*

|  |  |  |
| --- | --- | --- |
| *Task* | *Method* | *Return Value* |
| *Creates a bank account* | *\_\_init\_\_(first: str = “”, last: str = “”)* | *BankAccount* |
| *Gets the first name* | *getFirst():* | *str* |
| *Gets the last name* | *getLast():* | *str* |
| *Get the current balance* | *getBalance():* | *float* |
| *Get the account number* | *getAccountNumber()* | *int* |
| *Get the list of transactions* | *getTransactions()* | *list<Transaction>* |
| *Get the amount of times overdrawn* | *getTimesOverdrawn()* | *int* |
| *Print the account details* | *printAccount()* | *None* |
| *Print List of Transactions* | *printTransactions()* | *None* |
| *Deposit into an account* | *deposit(amount)* | *None* |
| *Add interest into account* | *addInterest()* | *None* |
| *Withdraw money from an account* | *withdraw(amount)* | *bool* |
| *Transfer money from another into account into this account* | *transfer(other, amount)* | *bool* |
| *Create human readable string representation of a bank account* | *\_\_str\_\_* | *str* |
| *Create machine readable string representation of a bank account* | *\_\_repr\_\_* | *str* |
| *Determine if two bank accounts are equal to each other* | *\_\_eq\_\_* | *bool* |
| *Determines if two bank accounts are less than to each other* | *\_\_lt\_\_* | *bool* |
| *Determines if a bank account is greater than another based on account number* | *\_\_gt\_\_* | *bool* |
| *Determines if a bank account is less than or equal to another bank account* | *\_\_le\_\_* | *bool* |
| *Determines if a bank account is greater than or equal to another based on account number* | *\_\_ge\_\_* | *bool* |
| *Sets the first name for the account* | *\_setFirstName* | *None* |
| *Sets the last name for the account* | *\_setLastName* | *None* |

*Class responsibilities for Bank Account class:*

* Know
  + Class overdraft fee
  + Class interest rate
  + Class next account number
  + First Name
  + Last Name
  + Account Number
  + List of transactions
  + Times overdrawn
* Do
  + Accessor Methods for all instance variables
  + Mutator Methods for all mutable instance variables
  + Print the details of an account
  + Print the list of transactions
  + Deposit money into accounts
  + Add interest to accounts
  + Withdraw money from accounts
  + Transfer money between accounts
  + Make string representations for humans and machines
  + Compare by equality, less than, greater than, less than or equal to, and greater than or equal to

**Class Definitions**

Class Name: *Bank Account*

Data:

Interest Rate: .0075 (Private and immutable)

Overdraft Fee: 20.00 (Private and immutable)

\_nextAccountNumber: 1000 (Private)

first: str - first name on the bank account (1-25 characters, no special)

last: str - last name on the bank account (1-40 characters, no special)

accountNumber: int – unique number of the account

transactions : list of transactions – list of the transactions the account has

timesOverdrawn: int – amount of times the account has been overdrawn

Queries (Accessor):

getFirst(): returns the first name

getLast(): returns the last name

getBalance(): returns the balance

getAccountNumber(): returns the account number

getTimesOverdrawn(): returns the number of times overdrawn

Commands (Mutator):

\_setFirstName(): Updates the first name of the account

\_setLastName(): Updates the last name of the account

\_incrementOverdraft(): adds 1 to the timesOverdrawn counter variable

deposit(): Deposits money into the account via creating a deposit transaction

addInterest(): Calculates and adds interest into the account via creating an interest transaction

withdraw(): Withdraws money from the account via creating a withdraw transaction

transfer(): Withdraws money from the other account and deposits it into self