

| Class | Outer Class | Method | Description | Parameters | Return |
|-----------------------|--------------------|--------------------------------|--|---|--|
| FrontEndInstance | N/A | constructor | Class to represent an instance of a front end that contains an accounts file, a user status, a transaction summary file, and an accounts list. | String: Accounts file name String: Transaction summary file name | FrontEndInstance: Instance |
| | | missing_user_state_for_command | A string that is output when a user attempts to use a command when not in the correct state | UserData: User state Command: Current transaction command | String: 'Error: ' + state + 'session required for ' + command + ' command' |
| | | successful_login | A string that is output when a user successfully logs in. | UserData: Session type (agent or atm) | String: 'Successfully logged in as ' + sessionType + '' |
| | | unrecognized_login_command | A string that is output when a user inputs an invalid login command. | String: Login command entered by user | String: 'Error unrecognized command: ' + command + '', login with either '\ + atm + ' or ' + agent |
| | | must_be_signed_in_for_command | A string that is output when a user inputs a command from an idle state. | Command: Command entered by user | String: 'Command: ' + command. value + '' requires a logged in session' |
| | | error_cents_less_than_or_equal | A string that is output when a user inputs a transaction amount greater than the maximum allowed amount. | String: Max transaction value | Error: must be less than or equal to ' + max + ' cents' |
| | | front_end_loop | Simulates a full day of banking. Starts a while loop that doesn't end until the program stops running, in which any number of users can login, make transactions, then logout. | None | None |
| | | login | Logs the user in (as long as they aren't already logged in), select their session type, and opens the account list file. | None | None |
| | | logout | Logs out a logged in user, writes the contents of the transaction summary object to the transaction summary file, and clears the transaction summary object and user state. | None | None |
| | | create_account | Allows authorized users to create an account – gets an account name and number and adds a row to the transaction summary. | None | None |
| | | delete_account | Allows authorized users to delete an account – gets an account name and number and adds a row to the transaction summary. | None | None |
| | | deposit | Allows users to deposit money to an account – gets a valid amount to deposit from user and adds a row to the transaction summary. | None | None |
| | | withdraw | Allows users to withdraw money - gets an account number and a withdrawal amount, removes the amount from the user's account, and adds a row to the transaction summary | None | None |
| | | transfer | Allows users to transfer money between accounts - gets account numbers and a transfer amount, performs the transfer and adds a row to the transaction summary | None | None |
| | | valid_account_name | Checks if the account name entered fits the required format of containing 3-30 digits, and not starting or ending with whitespace | String: Account name | Boolean: True if account name is valid, false otherwise |
| | | valid_account_number | Checks if the account number entered fits the required format of containing exactly seven digits, and not starting with 0 | String: Account number | Boolean: True if account number is valid, false otherwise |
| | | get_valid_account_number | Gets an account number as input from the user and checks if the number is valid. | None | If valid, Int: Account number else, String: Error message |
| | | get_account_number_in_list | Gets an account number as input from the user and checks if the account exists. | String: Prompt shown to user | If number exists, Int: Account number else, String: error message |
| | | get_valid_account_name | Gets an account name as input from the user and checks if the account name is valid. | None | If valid, String: Account name else None and print error |
| | | get_valid_numeric_amount | Gets a numeric input from the user and checks that the value entered is valid. | Int: Maximum valid value that can be given as input | If valid, Int: Value given as input else None and print error |
| UserData | FrontEndInstance | N/A | Class extending enum, specifies the possible user states (idle, atm, agent) | N/A | N/A |
| | | to string method | Gets a string representing the current user state. | None | Returns the string representation of the user state. |
| Commands | FrontEndInstance | N/A | Class extending enum, specifies the possible commands (login, logout, createacc, deleteacc, transfer, withdraw, deposit, cancel, help) | N/A | N/A |
| | | to string method | Gets a string representing the command. | None | Returns the string representation of the transaction type. |
| TransactionSummary | FrontEndInstance | constructor | Class to represent an instance of a transaction summary object, containing the information (list of rows) to be written to the transaction summary file. | String: Summary file name | TransactionSummary: Instance |
| | | add_row | Adds a TransactionSummaryRow object to the transaction summary rows list, containing the all the information for one transaction. | TransactionSummaryKey: Transaction type String: To account String: From account String: Account name, String: Transaction amount in cents | None |
| | | clear | Erases the current contents of the transaction summary. | None | None |
| | | total_within_daily_limit | For a particular account number, checks if transactions of the same type are below the daily limit. | TransactionSummaryKey: Output key String: Account number Int: Transaction amount Int: Daily transaction amount limit | Boolean: If the sum of all transactions of the same type on a particular account are within the allowed limits, returns true, else returns false |
| | | to_file | Writes the transaction summary file – adds the end of file line, creates folders if necessary to store log, and writes the list of transaction summary rows to the transaction summary file. | None | None |
| TransactionSummaryRow | TransactionSummary | constructor | Class to represent an instance of a row in the transaction summary file, containing all the relevant info (with to/from account, amount, transaction type, and account name). | TransactionSummaryKey: Transaction type String: To account String: Transaction amount in cents String: From account String: Account name | TransactionSummaryRow: Instance |
| | | to string method | Gets a string representing a row in the transaction summary file. | None | String: Representation of a row in the transaction summary file (with to/from account, amount, transaction type, and account name) |

| Class | Outer Class | Method | Description | Parameters | Return |
|------------------------|-----------------------|------------------|--|------------|-----------------------------------|
| TransactionSummaryKeys | TransactionSummaryRow | N/A | Class extending enum, specifies the possible codes representing certain commands to be logged to the summary file (DEP, WDR, XFR, NEW, DEL, EOS) | N/A | N/A |
| | | to string method | Gets a string representing the transaction summary code. | None | String: Transaction summary code. |