**Main**

|  |  |
| --- | --- |
| **Variables** | **Description** |
| Files | Calls the Files Class |

**Files**

|  |  |
| --- | --- |
| **Variables** | **Description** |
| List<UserAccounts> : list | Stores a list of the UserAccounts |
| List<AvailableItems> : list | Stores a list of the Available Items |
| List<Transaction> : list | Stores a list of the Transaction Items |
| **Methods** |  |
| updateUserFile() | Update the User Files |
| updateAvailableItemsFile() | Update the Available Items File |
| updateTransactionList() | Update the Transaction File |
| create(Transaction transactions) | Create a new user with information from the transaction object |
| delete(Transaction transactions) | Delete a user with information from the transaction object |
| advertise(Transaction transactions) | Advertise an item from the transaction object |
| bid(Transaction transactions) | Bid on an item with information from the transaction object |
| refund(Transaction transactions) | Refund credit with information from the transaction object |
| addCredit(Transaction transactions) | Add credit to a used with information from the transaction object. |

**Parser**

|  |  |
| --- | --- |
| **Methods** | **Description** |
| FileIO() | Calls the file input and output function |
| parseUsers(string UserAccountFile, list<User> users) : list | Parses the User Account File and stores them in a list |
| parseItems(string itemFile, list<Item> items): list | Parses the Item File and stores them in a list |
| parsedTransactions(string transactionFile, list<Transaction> &transactions): list | Parses the Transaction File and stores them in a list |
| writeUseFile(list <User> users): void | Writes to the user file |
| writeItemFile(list <Item> items): void | Writes to the items file |
| toString() | Converts raw data to a string |

**AvailableItems**

|  |  |
| --- | --- |
| **Variables** | Description |
| itemName: string | Contains the item name |
| sellerName: string | Contains the seller name |
| currentWinningBidder: string | Contains the current winning bidder name |
| numberOfDaysLeft: int | Contains the number of days left that the item is up for auction |
| highestBid: BidDecimal | Contains the highest bid for the item |
| **Methods** |  |
| AvailableItems() | Constructor that creates a blank new AvailableItems item with no data |
| AvailableItem(string itemName, string sellerName, string currentWinningBidder, int numberOfDaysLeft, BigDecimal highestBid) | Creates an item object that contains all the information about the object |
| getName(): string | Gets the name of the item |
| setName(): string | Sets the name of the item |
| getSeller(): string | Gets the name of the seller |
| setSeller(): string | Sets the name of the seller |
| getCurrentWinningBidder(): string | Gets the name of the user that has the highest bid on the item |
| setCurrentWinningBidder(): string | Sets the name of the user that has the highest bid on the item |
| getDaysLeft(): int | Gets the number of days left that the item is up for auction |
| setDaysLeft(): int | Sets the number of days left that the item is up for auction |
| setHighestBid(): BigDecimal | Sets the highest bid on the item |
| setHighestBid(): BigDecimal | Gets the highest bid on the tiem |
| toString(): string | Converts the item to a string |

**UserAccounts**

|  |  |
| --- | --- |
| **Variables** | **Descriptions** |
| userName: string | Stores the user name of the account |
| userType: string | Stores the user type of the account |
| availableCredit: BigDecimal | Stores the available credit of the account |
| password: string | Stores the password of the account |
| **Methods** |  |
| UserAccounts() | Creates a blank UserAccounts instance with no data |
| UserAccounts(string userName, string userType, string password, BigDecimal availableCredit) | Creates a UserAccounts instance with all the data specified |
| getUserName(): string | Get the username from the userAccount |
| setUserName(): string | Set the username of the userAccount |
| getUserType(): string | Get the user type from the userAccount |
| setUserType(): string | Set the user type from the user account |
| getAvailableCredit(): Big Decimal | Get the available credit from the user account |
| setAvailableCredit(): Big Decimal | Set the available credit in the user account |
| getPassword(): string | Get the password from the user account |
| setPassword(): string | Set the password in the user account |
| toString(): string | Converts the raw data into a string |

**Transactions**

|  |  |
| --- | --- |
| **Variables** | **Description** |
| transactionCode: int | Contains the transaction code of a transaction |
| userName: string | Contains the user name in the transaction file |
| availableCredit: BigDecimal | Contains the available credit in the transaction file |
| userType: string | Contains the user type in the transaction file |
| buyerName: string | Contains the buyer name in the transaction file |
| sellerName: string | Contains the seller name in the transaction file |
| refundCredit: BigDecimal | Contains the refund credit in the transaction file |
| daysToAuction: int | Contains the number of days that |
| minBid: BigDecimal | Contains the minimum bid for the item |
| itemName: string | Contains the item name |
| newBid: BigDecimal | Contains the new bid value |
| currentWinningBidder: string | Contains the current winning bidder on an item |
| highestBid: BigDecimal | Contains the highest bid on an item |
| **Methods** |  |
| Transaction() | Creates an empty instance of transaction |
| Transaction(int transactionCode, string userName, string userType, BigDecimal availableCredit, string buyerName, string sellerName, BigDecimal refundCredit, in daysToAuction, BigDecimal newBid, BigDecimal minBid, string itemName, string currentWinningBidder, BigDecimal highestBid) | Creates an instance of transaction with data applied. |
| getTranscationCode(): int | Gets the transaction code from the transaction object |
| setTransactionCode(): int | Sets the transaction code in the transaction object |
| getUserName(): string | Gets the username from the transaction object |
| setUserName(): string | Sets the username in the transaction object |
| getUserType(): string | Gets the user type from the transaction object |
| setUserType(): string | Sets the user type in the transaction object |
| getAvailableCredit(): BIgDecimal | Gets the available credit in the transaction object |
| setAvailableCredit(): BIgDecimal | Sets the available credit in the transaction object |
| getBuyerName(): string | Gets the buyer name from the transaction object |
| setBuyerName(): string | Sets the buyer name from the transaction object |
| getSellerName(): string | Gets the seller name from the transaction object |
| setSellerName(): string | Sets the seller name from the transaction object |
| getRefundCredit(): BigDecimal | Gets the refund credit amount from the transaction object |
| setRefundCredit(): BigDecimal | Sets the refund credit amount in the transaction object |
| getDaysToAuction(): int | Gets the number of days left in the item auction from the transaction object |
| setDaysToAuction(): int | Sets the number of days left in the item auction in the transaction object |
| setMinBid(): BigDecimal | Sets the minimum bid of the item in the transaction object |
| getMinBid(): BIgDecimal | Gets the minimum bid from the transaction object |
| getItemName(): string | Gets the item name from the transaction object |
| setItemName(): string | Sets the item name in the transaction object |
| getCurrentWinningBidder(): string | Gets the winning bidder from the transaction object |
| setCurrentWInningBidder(): string | Sets the new winning bidder in the transaction object |
| getHighestBid(): BigDecimal | Gets the highest bid amount from the transaction object |
| setHighestBig(): BigDecimal | Sets the highest bid amount in the transaction object. |