**News Class Fields**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Type** | **Access Modifier** | **Static** |
| companyIndex | int | public | true |
| newsIndex | int | public | false |
| lastCompanyIndex | int | private | false |
| lastNewsIndex | int | private | false |
| newsSelect | int | public | true |

**News Class Fields Functions**

companyIndex – Stores the index of the randomly generated news company.

newsIndex – Stores the index of the randomly generated news.

lastCompanyIndex – Stores the value of the last companyIndex to prevent a repeat of the same news.

lastNewsIndex – Stores the value of the last newsIndex to prevent a repeat of the same news.

newsSelect – Used to determine whether to choose a positive news or a negative news, 0 means to pick a negative news dialogue and 1 means to pick a positive news dialogue.

**News Class Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Return Type** | **Access Modifier** | **Parameters** |
| newsSelect | String | public | N/A |
| randomNegativeNews | String | public | N/A |
| randomPositiveNews | String | public | N/A |
| resetNews | void | public | N/A |
| isPositive | Boolean | public | N/A |
| companyIndex | int | public | N/A |
| newsPicture | String | public | N/A |

**News Class Methods Functions**

newsSelect – Randomly chooses to either pick a positive news by calling the randomPositiveNews method or pick a negative news by calling the randomNegativeNews method.

randomNegativeNews – Randomly picks a company and one of its negative news, and outputs the news dialogue from a file.

randomPositiveNews – Randomly picks a company and one of its positive news, and outputs the news dialogue from a file.

resetNews – Reinitializes all class variables to their initial values, since all the variables are indices, they are all changed to the value of -1.

isPositive – Determines if the last news dialogue randomly selected is a positive news or a negative news, returns true if the news is positive and returns false if the news is negative.

companyIndex – Determines the company index of the last news dialogue selected, 0 = Car Company, 1 = Computer Company, 2 = Fast Food Company, 3 = Grocery Store Company, 4 = Bank Company.

newsPicture – Outputs the name of the visual file of the news selected.

**User Class Field**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Type** | **Access Modifier** | **Static** |
| username | String | private | false |
| firstname | String | private | false |
| lastname | String | private | false |
| loginpass | String | private | false |
| password | String | private | false |
| balance | double | private | false |
| list | UserStock | private | false |

**User Class fields functions**

Username, firstname, lastname, loginpass, password: user’s general info and code required to have accessed to their account.

balance: user’s balance left that can be used to buy stock

list: user’s stock list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Return Type** | **Access Modifier** | **Parameters** | **Static** |
| resetBalance | void | public | N/A | false |
| addBalance | void | public | double | false |
| deductBalance | void | public | double | false |
| saveUser | void | public | N/A | false |
| loadUser | void | public | N/A | false |

**User Class method functions**

resetBalance: set the user’s balance to 0

addBalance: add certain amount of balance to user’s account

deductBalance: deduct certain amount of balance when a user buys stock

saveUser: save user’s information to a file temporarily

loadUser: load user’s information from a file

**Company Class Field**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Type** | **Access Modifier** | **Static** |
| companyName | String | protected | false |

companyName: the name of a company

**Stock Class Field**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Type** | **Access Modifier** | **Static** |
| stockCode | int | private | false |
| updatedPercentage | double | private | false |
| trend | boolean | private | false |
| tradeDays | int | private | true |
| stockQty | int | private | false |
| stockPrice | double | private | false |
| locationX | double | private | false |
| locationY | double | private | false |

stockCode: unique code for each company’s stock

updatedPercentage: change of stock’s price in a day

trend: current trend of a stock, incline or decline

tradeDays: how many days of trading has been done

stockQty: quantities of stock has been bought , differ based on the owner of the stock

stockPrice: the price of stock currently

locationX: x-coordinate of the stock reflected on the line graph that shows the trend of stock

locationY: y-coordinate of the stock reflected on the line graph that shows the trend of stock

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Return Type** | **Access Modifier** | **Parameters** | **Static** |
| initializeTradeDays | void | public | N/A | false |
| generateTrend | void | public | N/A | false |
| generatePercent | double | public | double | false |
| updatePercent | void | public | double | false |
| updatePrice | void | public | double | false |
| initializePixel | void | public | N/A | false |
| updateLocation | void | public | N/A | false |
| recordHistory | void | public | N/A | false |

initializeTradeDays: initialize the number of days of trading when the trading of each stock starts in the first day

generateTrend: randomly determine the trend of a piece of stock in a day and update this value to the field of trend

generatePercent: based on the trend of the stock, randomly generate the change of stock price in percentage and return the value

updatePercent: based on the percent generated, update the field of updatedPercentage;

updatePrice: calculate the new price of one stock in a new days based on the updatedPercentage;

initializePixel: set locationX and locationY to the initial position of the line graph on the mainframe.

updateLocation: update the values of locationX and locationY after a day of trading

recordHistory: record days of trading, stock price, location x and y to a file.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Type** | **Access Modifier** | **Static** |
| stockList[] | Stock | public | false |
| purchase | double | public | false |
| withdrawal | double | public | false |
| Invested[] | double | public | false |
| defaultStockList[] | Stock | public | true |
| tradeTrend | ArrayList<String> | public | false |
| tradeQty | ArrayList<double> | public | false |
| stockName | ArrayList<String> | public | false |

stockList[]: a list of stock owned by an user

Purchase: total amount of money has been used to purchase

Withdrawal: total amount of money has withdrawn from the stock market

Invested[]: amount of money for each piece of stock

defaultStockList[]: a list of all stocks of all company

stockName: after a day of trading, record the trading company

tradeTrend: after a day of trading, record the newest trend of a stock bought by an user

tradeQty:after a day of trading, record the newest trend of a stock bought by an user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Return Type** | **Access Modifier** | **Parameters** | **Static** |
| loadStockList | void | public | N/A | true |
| saveStockList | void | public | N/A | true |
| updateStock | void | public | N/A | true |
| updateMyStock | void | public | N/A | false |
| initializeAmount | void | public | N/A | false |
| initializeinvestment | void | public | N/A | false |
| addStockToList | void | public | int | false |
| buyStock | void | public | Int,int | false |
| removeStockFromList | void | public | int | False |
| sellStock | void | public | int,int | false |
| rankStock | Stock[] | public | N/A | true |
| listStock | String[][] | public | N/A | true |
| allToString | String[] | Public | N/A | true |
| calculate | double | public | int, int | false |
| addTradeHistory | void | public | Int, boolean, int | false |
| myStockToString | ArrayList<String> | public | N/A | false |

loadStockList: add info for each stock from the file Company Information

saveStockList: record newest info of each stock to the file Company Information, override

updateStock: update the information of stock in the default stock list after a day of trading

updateMyStock: synchronize user’s stock list with the default stock list

initializeAmount: reset or initialize the value of purchase and withdrawal

initializeInvestment: reset or initialize the array of value of the amount invested for each stock

buyStock: take in the quantity user wants to buy and the stock’s index in the user’s stock list, consume amount of balance and buy stock

addStockToList: inside of buyStock, if the stock being bought has never been bought by the user before, add it to user’s stock list

sellStock: take in the quantity user wants to want and the stock’s index in the user’s stock list, return amount of balance and sell stock

removeStockFromList: inside of sellStock, if the stock being sold has 0 piece of stock left, remove it from user’s stock list

rankStock: return a new stock list that has the same content default stock list but different order, rank from highest price to lowest

listStock: return a 2d array that contains the company name, current trend and change of percentage of the stocks in stock list returned by rankStock

allToString: convert all the information of defaultStockList to string

Calculate: takes in the stock index and the quantity and return the amount of money required to buy certain amount of stock

addTradeHistory: everytime a series of trade is done, add it to the arraylist that record trading history

myStockToString: return an arraylist of string information that tells user’s stock information