Account Management System

In this project, you will write a program to manage a bank account and a stock portfolio. The program will be written using inheritance structure for the classes. Starting with the abstract base class `Account`, write two derived classes `stockAccount` and `bankAccount`. All of the accounts should be linked together through the common `cashBalance` variable. The starting balance for the account will be $10000. The balance will change as you perform the transactions.

In the main menu, you can select to work with either stock account, bank account, or exit the program. The sub menu for each will be as following:

Stock account:

- Display current price for a stock symbol
- Buy stock
- Sell stock
- Display current portfolio
- Display transactions history
- Return to main menu

Bank account:

- Display current cash balance
- Deposit to account
- Withdraw from account
- Display transactions history
- Return to main menu

1. Stock Account

This program manage your stock portfolio. You can purchase or sell stocks. Use the stock information in `stock1.txt, stock2.txt, stock3.txt, stock4.txt` files for all of the transactions. When you initiate a transaction, you can randomly select one of the files for stock prices.

Display current price for a stock symbol: this function take the symbol of a stock and display its current price. The current price will be the price of one of the stock file selected randomly. The format as follow:

```
VNET       21Vianet Group Inc. ADR        5.48       12:48:53
```

If the stock is not available, display corresponding message.

Buy stock: you can purchase the stock by enter the symbol, the amount of share you want to buy, and the price you're willing to pay for each share. This function will first select a random stock file. This selected stock file will provide the symbol name and the price for each stock. Assume that the number of available shares is unlimited. If the price you're willing to pay for each stock is large or equal to the current price, the transaction will be performed (given that you have enough cash in your account balance). If the price you're willing to pay less than the current price then the transaction will not go through. Display a corresponding message (`“Cannot process transaction”`).

Update your account balance and stock portfolio after each transaction.

Sell stock: you can sell the stock by enter the symbol, the amount of shares you want to sell, and the price you're willing to sell for each share. This function will first check your stock portfolio to see if you have the stock and the shares you want to sell. Next, the function selects the random stock file for checking the current stock prices. If the price you're willing to sell for each stock is less than or equal to the current price, the transaction will be performed. If the price you're willing to pay larger than the current price then the transaction will not go through. Display a corresponding message (`“Cannot process transaction”`).

Update your account balance and stock portfolio after each transaction.

Display current portfolio: this function displays the current account balance and the stock portfolio. The function first select a random stock file for current prices. The format as follow:

```
Cash balance = $12034.54

Symbol     Company                    Number     Price      Total
VNET       21Vianet Group Inc. ADR    120        5.48       657.60
AGTK       Agritek Holdings Inc.      3000       0.02       60.00

Total portfolio value: $ 12752.14
```

Display transactions history: this function displays the stock transaction history. For every transaction, the information should be saved in `stock_transaction_history.txt` file. This function will read the `stock_transaction_history.txt` file to display all the the transaction history. The format as in following

```
Action     Symbol     Shares     Price      Time
Buy        VNET       20         5.48       13:30:45
Buy        AGTK       130        0.02       13:34:54
Sell       AKAM       100        58.88      14:34:23
```

Return to main menu: this function returns to the main menu

2. Bank Account

This program manage your bank account.

Display current cash balance: this function can reuse the **Display current portfolio** function. The format as in following:

```
Cash balance = $12034.54

Symbol     Company                   Number     Price      Total
VNET       21Vianet Group Inc. ADR   120        5.48       657.60
AGTK       Agritek Holdings Inc.     3000       0.02       60.00

Total portfolio value: $ 12752.14
```

Deposit to account: this function deposit money to your account. Enter the amount to be deposited and update your account. Display corresponding message if the amount is not correct, "`Invalid input`", (negative amount, not a number, etc.). You can reuse the **Display current portfolio** function to update the account. The format as in following:

```
Deposit $500.00 to bank account

Cash balance = $12534.54

Symbol     Company                   Number     Price      Total
VNET       21Vianet Group Inc. ADR   120        5.48       657.60
AGTK       Agritek Holdings Inc.     3000       0.02       60.00

Total portfolio value: $ 13252.14
```

Withdraw from account: this function withdraw cash from your account. Enter the amount to be withdrawn and update your account. Display corresponding message if the amount is not correct, "`Invalid input`", (negative amount, amount more than available cash, not a number, etc.). You can reuse the **Display current portfolio** function to update the account. The format as in following:

```
Deposit $500.00 to bank account

Cash balance = $12034.54

Symbol     Company                   Number     Price      Total
VNET       21Vianet Group Inc. ADR   120        5.48       657.60
AGTK       Agritek Holdings Inc.     3000       0.02       60.00
```

```
Total portfolio value: $ 12752.14
```

Display transaction history: this function display all of the bank transaction history. For every transaction, the information should be saved to `bank_transaction_history.txt` file. This function will read the `bank_transaction_history.txt` file to display all the the transaction history. The format as in following:

```
Action      Amount      Cash Balance     Date            Time
Deposit     500         12534.54         04/14/2017      01:48:19
Withdraw    500         12034.54         04/14/2017      01:48:33
```

A few points...

Use a vector to manage the current list of stocks. Sort the list for every stock transaction. The list is sorted by the stock symbol.

Reuse all the functions and variables as much as possible through inheritance and polymorphism.

Each function will check for the current stock prices by selecting a new random stock file if needed.

The program will read a text file for input and run. It will generate an output and save it to a separate text file. You should make your own input file for the more extensive test cases. For example, make the test cases where the amount of deposit or withdraw is invalid (negative number, letters, etc.).

The price of stock always be the current prices. That means, you always have to check for current prices (select random stock file) for every time you calculate the total portfolio value.

If you need any additional information, please let me know. I will update the input/output files as needed.

Good luck
Tuan Le