CSCI 2073 – Programming Assignment 4 – Fall 2021

You have been hired to develop a Java program to keep track of customers' stock holdings. As customers buy and sell shares of various company stocks, the program is to keep track of the value of their portfolio of stocks. Part of your task is to select the appropriate classes, methods, and data structures to be used when storing and manipulating the information. While you are free to develop any additional classes and/or methods as part of your solution, you are required to implement the **Stocks.java** class with the following public methods:

`Stocks(String dataFile)`: constructor to create an empty stock portfolio for a customer. Information about all available stocks is stored in the file given as argument. Each line of the file contains the following information about a company, with items being separated by commas:

```
Symbol,Title,Sector
```

You are free to store the stock information from the data file in any suitable data structure of your choice, but keep in mind that the actual file may be of a size different from the one provided for testing. Also note that this file contains all of the possible stocks that can be acquired by your customers, whether they end up being part of a portfolio or not.

`String transactions(String transFile)`: method to process a file containing the actual transactions carried out by a customer. The file contains one transaction per line. The method returns a single string containing one line of output corresponding to each transaction. The possible transactions are:

```
BUY symbol #shares sharePrice
SELL symbol #shares sharePrice
SUMMARY
SECTOR name
```

The functionality expected when processing each transaction is as follows:

- BUY: display number of shares, the name of the company, as well as the resulting total shares and average share price for that company stock, using the format below.
  `BOUGHT XX shares of YY (holding ZZ shares at $DD.CC per share)`
- SELL: display number of shares, the name of the company, and the resulting profit or loss, using the format:
  `SOLD XX shares of YY at a profit/loss of $DD.CC`
- SUMMARY: display total cost of all stocks as well as total profit or loss from all stock sales, using the format:
  `TOTAL ASSETS: $DD.CC, TOTAL PROFIT/LOSS: $DD.CC`
- SECTOR: display the dollar amount and percentage of stock holdings in a particular sector, using the format:
  `XYZ: $DD.CC (AA.BB pct)`

Any BUY/SELL transaction involving an invalid stock symbol should be ignored and result in an INVALID RANSACTION message, as should any SELL transaction involving more shares than the number of shares being held. Sample input file contents and corresponding output:

BUY F 100 59.75	BOUGHT 100 shares of FORD MOTOR CO (holding 100 shares at $59.75 per share)
BUY KR 200 39.25	BOUGHT 200 shares of KROGER CO (holding 200 shares at $39.25 per share)
BUY BA 100 754.25	BOUGHT 100 shares of BOEING CO (holding 100 shares at $754.25 per share)
BUY CVS 200 19.95	BOUGHT 200 shares of CVS HEALTH CORP (holding 200 shares at $19.95 per share)
SELL KR 100 40.00	SOLD 100 shares of KROGER CO at a profit/loss of $75.00
BUY CVS 100 20.25	BOUGHT 100 shares of CVS HEALTH CORP (holding 300 shares at $20.05 per share)
BUY BA 100 750.25	BOUGHT 100 shares of BOEING CO (holding 200 shares at $752.25 per share)
SELL KR 100 38.00	SOLD 100 shares of KROGER CO at a profit/loss of $-125.00
SELL APL 100 45.00	INVALID TRANSACTION
SECTOR Industrials	Industrials: $150450.00 (92.62 pct)
SUMMARY	TOTAL ASSETS: $162440.00, TOTAL PROFIT/LOSS: $-50.00

Test cases for this assignment are based on a comparison of the string returned by **transactions** with expected output. Make sure that your program's output matches the expected output EXACTLY. A sample test program and data files have been provided for basic testing purposes. For final testing, submit **Stocks.java** (and any other .java files you may develop) to the Mimir assignment system.