

Assignment 8: Simple Algorithm Trader

CIS 210 – Fall 2016

Point: 100

Project Due: 11:59 PM on Thursday, 10/27/2015

Learning Objectives:

- ❖ To understand the use of Classes in Java
- ❖ To understand the concept of Class aggregation and class collaboration.
- ❖ To use a data structure such as Array List

Problem Statement

Assume that you are a software developer in an investment bank on Wall Street, New York. Your boss asks you to design a trading strategy and test it. You will design a set of classes that work together to simulate buying and selling a stock. The classes you will design are the following:

- The **Stock** Class: this class will represent a stock. You need to decide the member variables of this class based on the input file. This input file is given to you and it is a “comma-separated-value” file (.csv). You can open a .csv file using Microsoft Excel.
- The **AlgorithmTrader** Class: this class will simulate the stock trading activities. Its responsibilities are as follows:
 - Read stock minute-by-minute data from an input file, MSFT.csv.
 - Store stock minute-by-minute data in an Array List.
 - Decide when to buy and sell a stock using the simple algorithm.
 - **Entry position strategy:** if a stock’s share price increases **5** times consecutively, then you buy **10,000** shares based on the current price.
 - **Exit position strategy:** if your current position is gaining 0.12% profit or losing 0.12%, then you sell all the shares you are holding base on the current price.
 - **Special Exit strategy:** if you are still holding a stock at the last minute of a trading day, you **MUST** sell all of the shares you own. Because the risk of losing money is too big during the after hour trading period.
 - Simulate buying and selling stocks
 - To write a summary of Transactions, Profit/Loss for each minute. Please see the tradingSummary-sample.csv file on Blackboard.
 - **Note, the AlgorithmTrader class MUST use an array of objects of the Stock class.**
- Demonstrate the classes by creating instances of each class. Create the main class *SimpleAlgorithmTradingPlatform.java*. Simulate one day of trading stocks. This main method should ask user to enter an input file name, create an object of the **AlgorithmTrader** class, open an output file, and call the methods of the *AlgorithmTrader* object.

Note: your program must be user-friendly and intuitive. This is a part of your grade. In other words, even if your program does everything the problem statement states, your grade may be reduced because of difficulty to it.

Input

The name of the input file which stores a stock's minute-by-minute data. Please download the MSFT.csv file from Blackboard. This input is from the user's keyboard. If the file doesn't exist, please print an appropriate message and terminate the program.

Output

Your program should create an output file and store it on your computer disk. Your output file must be a .csv file. Please name the file *tradingSummary.csv*. A sample output file, *tradingSummary-sample.csv*, can be downloaded from BB.

Sample Output:

A	B	C	D	E	F	G	H	I
TIMESTAMP	CURRENT_PRICE	SHARES	P/L_PERCENT	PROFIT/LOSS	REALIZED_PROFIT/LOSS	HOLD/NONE	PURCHASE/SELL_PRICE	PURCHASE_COST
9:30	57.515	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:31	57.46	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:32	57.45	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:33	57.44	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:34	57.475	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:35	57.465	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:36	57.49	0	0	\$0.00	\$0.00	NONE	0	\$0.00
9:37	57.49	0	0	\$0.00	\$0.00	NONE	0	\$0.00

A	B	C	D	E	F	G	H	I
10:38	57.7455	0	0	\$0.00	\$0.00	NONE	0	\$0.00
10:39	57.75	0	0	\$0.00	\$0.00	NONE	0	\$0.00
10:40	57.76	10000	0	\$0.00	\$0.00	HOLD	57.76	\$577,600.00
10:41	57.785	10000	0.000433	\$250.00	\$0.00	HOLD	57.76	\$577,600.00
10:42	57.7632	10000	5.54E-05	\$32.00	\$0.00	HOLD	57.76	\$577,600.00
10:43	57.735	10000	-0.00043	(\$250.00)	\$0.00	HOLD	57.76	\$577,600.00
10:44	57.78	10000	0.000346	\$200.00	\$0.00	HOLD	57.76	\$577,600.00
10:45	57.805	10000	0.000779	\$450.00	\$0.00	HOLD	57.76	\$577,600.00
10:46	57.8154	10000	0.000959	\$554.00	\$0.00	HOLD	57.76	\$577,600.00
10:47	57.78	10000	0.000346	\$200.00	\$0.00	HOLD	57.76	\$577,600.00
10:48	57.78	10000	0.000346	\$200.00	\$0.00	HOLD	57.76	\$577,600.00
10:49	57.82	10000	0.001039	\$600.00	\$0.00	HOLD	57.76	\$577,600.00

Use of Methods, Parameters, Modularity, Design, etc.

Part of your grade on this and ALL future programming projects in this course will be determined by how well you use multiple functions and parameter passing appropriately and how well you design a modular and functionally cohesive program using the principles discussed in class. Large grade point penalties can be incurred for not setting up a modular, well designed program structure. This emphasizes good program structure, design, and fundamental software engineering principles.

Miscellaneous

Be sure to read and understand the sections in the Course Syllabus handout on general project requirements. Also be sure to study the style, documentation and formatting guidelines discussed in the *Programming Style Guidelines* handout and in the lecture.

What File To Turn In and How to Turn In Your Work using Blackboard

You must turn in your Java program source file in a zip file. The zip file must include a Stock.java file which defines the Stock class, a AlgorithmTrader.java file that defines the AlgorithmTrader class, and a SimpleAlgorithmTradingPlatform.java file which includes the main() method. The zip file's name must follow this format: hw8_SimpleAlgortihmTrader.zip