TCSS 435 — Artificial Intelligence & Knowledge Acquisition
Autumn 2014 — Homework Assignment 5

Due Date: Friday, Dec. 5

Guidelines

Homework should be electronically submitted to the instructor by the end of the day on the due date. A submission link is provided on the course Canvas page for this assignment.

Assignment Description

In this assignment you'll implement an automated stock trader. Like many automated trading systems before it, the goal of the trader is to maximize its return on investment over a span of (simulated) trading days. Unlike previous assignments that each focused on a specific A.I. technique, you are free to use any applicable A.I. technique to implement your trader. However, a hand-coded trader, i.e., manually designed trading rules, is not acceptable.

Implementation Specifications

The automated trader has two phases of operation. In the first phase, the trader is given historical information about a single stock. For a span of consecutive trading days the trader is proved the stock's opening price, closing price, high price, low price and trading volume for each day. The trader can use the historical information in any way it sees fit in order to prepare for the second phase.

In the second phase, the trader participates in a simple stock trading simulation. At the end of each simulated day, the trader receives the stock's information (opening, closing, high, low, volume). The trader then decides whether it wishes to be invested in the stock (own shares) on the following day. If the trader indicates that it wishes to invest and it currently doesn't own any shares of the stock, the simulator will purchase as many shares as the trader can afford for the trader at the following day's opening price. If the trader indicates that it doesn't wish to invest and it currently owns shares of the stock, the simulator will sell all the trader's shares at the following day's opening price. This simplified trading behavior forces buy and sell actions to alternate, the trader cannot buy (or sell) shares multiple times in a row. Note that the trader is not directly indicating whether it wishes to buy or sell shares, it's indicating its preference to own shares. For the most part, when the trader's preference changes from "don't invest" to "invest" shares will be purchased (provided that the trader has sufficient funds), and when the trader's preference changes from "invest" to "don't invest" shares will be sold. The trader begins with $1,000 and there is a fixed commission (cost) of $5 on every trade. The trader's goal is to maximize the amount of money it has at the end of the simulation.

The provided source code includes a class that bundles together daily stock information, `DailyStockInfo`, a skeleton `StockTrader` class, and a test driver for the trading simulation `Main.java`. As provided, the test driver runs simulations on 16 different stocks and displays the profit the trader achieved for each. Your primary task is to complete the `StockTrader` class. That class contains two methods, one for each of the two phases of operation. Refer to the Javadoc documentation for further details.

Deliverables

The following items should be submitted to the instructor. Failure to correctly submit assignment files will result in a 10% grade penalty.

1) Source code for your completed `StockTrader.java` file.
2) Any necessary supporting files, e.g., source code for additional classes.

Do not submit unmodified provided files, e.g., `DailyStockInfo.java`. Do not include any extraneous files, such as Eclipse IDE files, bytecode files, or subversion files.