AP Computer Science A@Beijing National Day School

Problem Set 13: `Stock Broker`

Due date: Monday, March 4, 2019
Instructor: Mr. Alwin Tareen
Total Points: 15

Task Overview

- Place an information box at the beginning of your `Java` program, **which includes your name**. This formally confers your ownership of the code.
- Implement a program that manages the value of an investor's stock portfolio.

Background

- Stocks are a type of security that gives stockholders a share of ownership in a company. They are issued by companies in order to raise capital, so that they may grow their business, or undertake new projects.
- Investors can buy and sell stocks on an open exchange called the stock market. It contains an indexed list of every company that has issued stock. It also provides the price of every company's stock at the current time, which is the stock's market value.
- The price of a stock can fluctuate dramatically according to market forces, and the concept of supply and demand. If a company is perfoming well, then the price of its stock tends to rise. However, if a company is doing poorly, then its stock price can drop.
- Investors can make money on the stock market by purchasing stock at a certain price, and then selling it later at a higher price, in which case they have earned a profit. However, if an investor sells a stock for a lower price than they have purchased it, then they shall incur a loss.

Specification

The Information Box Which Includes Your Name [5 points]

- **You are responsible** for typing your own name into the `Author` field, where it says: `YOUR NAME HERE`. Include both your English and Pinyin names.
- If the `Author` field is left blank, or if it contains **someone else's name**, then these 5 points will not be awarded to you.

Managing a Stock Portfolio [10 points]

- Write a `Java` program in the file `StockBroker.java` that keeps track of an investor's purchases of stock in a particular company. The price of a stock changes frequently, so stocks bought at different times are usually purchased at different prices.
- In order to simplify matters, we will assume that the investor is only purchasing shares in a single company. Also, we will only deal with the purchasing of shares, not the selling of them.
- You will write your solution in a class called `public class StockBroker`, right below the place where it says: `YOUR CODE HERE`.
- Your `StockBroker` class should have the following properties:

- Instance variables:
  - `private ArrayList<Integer> shares;`
    This data structure stores the *quantity of shares* that a investor has purchased, and each purchase is recorded as a separate event.
  - `private ArrayList<Double> cost;`
    This data structure stores the *price per share* for each transaction that an investor has engaged in.
  - In general, these two `ArrayLists` form a transaction history for the investor, as they store the quantity of shares purchased and the price per share for every purchase event, which occurs at every index.
- Constructor:
  - `public StockBroker()`
    This is a default constructor, as it has no parameters. It is responsible for initializing the two `ArrayLists`, `shares` and `cost`.
- Accessor method:
  - `public double getProfit(double currentPrice)`
    This method indicates the amount of money that has been gained or lost on the investor's stock holdings. The method has one parameter, which is the current price per share of the stock, which is a `double`.
  - If the investor has earned money on his stock investments(market value is greater than total cost), then the result will be a positive value. However, if the investor has lost money on the stock(market value is smaller than total cost), then the result will be a negative value.
  - The profit is computed in the following manner:
    - Compute the current *market value* of all the shares of the stock(which is the total number of shares purchased so far, times the current price passed to the method).
    - Subtract the *total cost* of all the shares of the stock(the total amount of money that the person has paid for all the shares of stock purchased so far).
- Mutator method:
  - `public void purchase(int amount, double price)`
    This method executes a purchase transaction, in which an investor buys some shares of stock at a given price.
  - The method has two parameters: `amount`, which is the *number of shares* as an `int`, and `price`, which is the *price per share* as a `double`.
- Make sure that you run your `Java` program, and ensure that it is free of errors. The program `StockBrokerTest` includes method calls to `purchase()` and `getProfit()`. It depicts an investor making a net profit of $30.00. This is because the investor buys a total of 30 shares(20 + 10) for a total of $90.00 cost($70.00 + $20.00), but the new current share price is $4.00, meaning that the shares are now worth $120.00. The output should look like the following:

```
Profit earned = 30.0
```

Testing

- In order to run the JUnit test bench, simply click on the Run Tests button. If you are missing this button, then right-click on the module StockBrokerJUnitTest and select All Tests. Then, you should see a BlueJ: Test Results window appear.
- If your JUnit test is successful, you should see a green bar appear. Also, each of the specific testing functions should have a green checkmark in front of them. If your test is unsuccessful, then a red bar will appear, and you will need to correct the errors.

Submission

- Submit your Java program by uploading it to the Web-CAT automated grading platform. Click on the following link:
  http://ec2-54-65-207-33.ap-northeast-1.compute.amazonaws.com:8080/Web-CAT/WebObjects/Web-CAT.woa