Collaboration Statement

The last page of the course syllabus contains a statement of the department collaboration policy with regard to projects. Violation of the collaboration policy can result in failure of the course.

Reminder: Project Hand in Rules

Read and follow the instructions!

1. All project due dates are specified on Brightspace.
2. All projects handed in must be NetBeans projects.
3. Each student starts the semester with 7 free late days. When handing in a project late you must add a comment telling me that you are using some number of late days for the project.
4. If you hand in a project late, you are not allowed to correct and resubmit. This is the only penalty for late submissions.
5. For projects, you will hand in what is specified by individual assignments. Most of the time it will be an entire zipped project folder. Name the archive with ONLY your last name. Example: Applin.zip ONLY zip archives will be accepted.

BACKGROUND

Publicly traded stocks are one of the many investment instruments available to people with surplus income. Although it can be very volatile in a short period, over periods of many years the stock market will generally outperform many other alternative investments, as measured by the percentage increase in value.

For this project you will read in a data file that contains a number of stock price quotations for a collection of companies, calculate a number of statistics for that data, and display the results in an attractive tabular form.

PROJECT KIT You will need to create your own project for this one. The project kit files (all of them) should be placed inside the project folder on the same level with the src folder. Your program must run correctly with each of the 3 input files (test0.txt, test1.txt, and test2.txt) the correct output is shown in three other text files (out0.txt, out1.txt, and out2.txt). You only need to test each file until it is successful. Please don't run it with all 3 files at once. That will make my testing more difficult. Print your output to the console not to a file.

INPUT DATA FILE FORMAT You must structure your program to read the name of the input file from the String[] args using the Run Arguments from the Project Properties.

The input file format will be a sequence of paired lines, the first line holding the name of the company (which may have embedded blanks, as in "Apple Inc.") and the second line

holding a sequence of positive real values representing the share price quotations(pq_n) over some number of periods. For example,

```
Apple Inc.
114.5 120.6 130.2 128.1 126.7 129.3
```

So the generic structure of the data for a single company is

```
<company name>
<pq0> <pq1> ... <pqn-1>
```

where there are n price quotations.

You should write your program so that if the file is empty, it fails gracefully, that is, without throwing an exception. Just have it report that there is no data in the file. (test with test0.txt) How to do this was coverei in the Bookstore Lab.

The number of prices given can vary from file to file, but within a file it will be the same for each company and will be at least one price. All the prices will be positive.

You will need to design and implement a Stock class and a Portfolio class. The Stock class will model a single Stock and the Portfolio class will contain an ArrayList of Stocks. Some of the statistics below are done for a Stock and should be part of that class, the extra credit statistics are all done on the entire Portfolio, so it should be responsible for those calculations. You should not calculate anything twice. If you need it in two places, it should be a property and should be stored.

STATISTICS TO CALCULATE For each company, you should calculate the following statistics. Each of these should be a property. Each calculation that requires more than one line, should be calculated in a private method that is called from the constructor.

- Low price: the minimum price given for the company's shares, 114.5 for our example above
- High price: the maximum price given for the company's shares, 130.2 in our example
- Net change over the whole sequence: the difference between the last price and the first price, 14.8 for our example
- Average price for the whole sequence: the sum of the prices divided by the number of quotations, 124.9 for our example above
- Standard deviation for prices: this is defined as the square root of the average of the squares of the differences between each price quotation and the average. For our example, this would be the square root of
  $\sqrt{((114.5 - 124.9)^2 + (120.6 - 124.9)^2 + \ldots + (129.3 - 124.9)^2)/6}$
  which turns out to be 5.59 when rounded to two decimal places.

  You should run this much with the Unit Test provided. If everything is correctly

running and calculating at this point, you have earned a C. Please check the grading criteria, the next two statistics are for the A. You should write Portfolio and get it working correctly before trying to tackle these.

- Longest upward trend: if a quoted price is not LESS than its predecessor, then we say the stock is on an upward trend for that period(we could define this more strictly, but we will do it this way). A maximal length upward trend would be a sequence of price quotes $q_i, q_{i+1}, \ldots, q_j$such that:

  either i is 0 so pq_i is the first quotation given, or i >0 and $pq_{i-1} > pq_i$

  either j is n - 1 so pq_j is the last quotation given, or j <n and $pq_{j+1} < pq_j$

  The length of such a trend is one less than the number of quotations in it, j - i. Our example has two upward trends of positive length, one from 114.5 to 130.2, of length 2, and one from 126.7 to 129.3, of length 1. The longest upward trend is the largest length of all the maximal length upward trends, which is 2 in our example.
  Note, under this definition, if the stock prices declined over the entire represented interval, each individual quotation would be a maximal length upward trend of length 0.

- Best growth rate of upward trend: with a maximal length upward trend defined as above, we define the growth rate of the upward trend as $(pq_j - pq_i)/(j - i)$, provided j >i. So for our first trend, the growth rate is (130.2 - 114.5)/2, or 7.85, and the growth rate of the second is (129.3 - 126.7)/1, or 2.6. The best growth rate of an upward trend is the highest value for all of the upward trends, 7.85 for our example. This value represents the best average per period gain for the stock during its upward trends. It is undefined for a stock that declines over the entire represented interval, since such a sequence will not have positive length upward trends.

- Portfolio class can calculate a statistic for the entire portfolio. We can define the rate of increase(or decrease, if the stock declines in value) for a company from period i to period i + 1 as $pq_{i+1}/pq_i = r_i$. It represents the investment multiplier for that period. If a person invested X dollars in the company at period i, that money would purchase X/pq_i shares. At the next period, those shares would be worth $pq_{i+1} * X/pq_i$ dollars, which is $r_i * X$. Add two statistical columns to the output
  Best rate: the largest value of r_i for that company, for i ranging over 0, 1, ..., n-2. Note this statistic is undefined if n is 1, that is, when there is only one price quotation in the list.
  Period of best rate: the smallest i for which r_i achieves the best rate value, that is the earliest period in the list in which the best rate value is achieved. Again, this is undefined if there is only one quotation in the list.

OUTPUT DISPLAY After calculating the values for each listed company, you should display them in a table with labeled columns for the company name, and each of the seven numerical statistics described above. The values for each column should line up attractively. The company names should be left justified in their column, and the numerical values should be right justified in theirs. Double values should be rounded to the hundredths place.

You can use the String class's format method to construct appropriate strings. Company names will be no longer than 20 characters, all double values will be no longer than 8 characters, which includes the minus sign(when needed), the decimal point, and the two digits of precision after the decimal point. The integer value will need no more than 3 digits. Of course, there should be at least one space character between the value of a statistic and whatever data is on the same row to its left.

Column labels need not be as long as in our list above, but should be clear. If you use a long description string, such as "Best Upward Trend Rate", if the column itself is not long enough to accommodate all the characters of the label, you should break it up into more than one line, as in:

Best Upward
Trend Rate

The last statistic may be undefined for some companies. In such cases, just put "n/a" for the entry, right justified in the column. Refer to out2.txt for formatting and correct statistics. Warning, the data files change from semester to semester.

Grading Criteria:
Each requirement below is graded as all or nothing. Grading for this project will be done as follows:

Grade Earned	Requirements
D	main methods are the only static methods in any class. main method in the driver checks command line arguments Usage message is correct for the number of required arguments . main method in the driver creates an object of the class and calls a run method. All code handed in is syntactically correct - the program compiles and runs.
C	Everything above Plus Stock class is correctly written with methods in IPO order Correctly calculates the first 5 statistics. Stock class toString returns a format string using format desciptors with field widths and does NOT end with a linefeed. Stock works with the Unit Test provided.
B	Everything above plus Portfolio class is correctly written with methods in IPO order Portfolio toString returns the formatted output including column headers, and linefeeds to format the strings returned from calling the toString in Stock
A	Everything above plus Stock class correctly calculates the Longest Upward Trend Stock correctly calculates the best growth rate. Portfolio correctly calculates the period of best rate.

WHAT TO HAND IN Compress the entire project folder with the data files in the correct place and upload it to Brightspace. All input files should be where they belong in the project

folder. The Unit tests for stock and portfolio should be in those files and NOT commented out.