|  |  |  |  |
| --- | --- | --- | --- |
| Title/Reference | Super Simple Stocks | Author | Savant Singh Dhindsa |

# Super Simple Stocks

### Requirements

1. Provide working source code that will :-
   1. For a given stock,
      1. Given a market price as input, calculate the dividend yield
      2. Given a market price as input, calculate the P/E Ratio
      3. Record a trade, with timestamp, quantity of shares, buy or sell indicator and trade price
      4. Calculate Volume Weighted Stock Price based on trades in past 15 minutes
   2. Calculate the GBCE All Share Index using the geometric mean of prices for all stocks

### Constraints & Notes

1. Written in one of these languages:
   * Java, C#, C++, Python
2. No database or GUI is required, all data need only be held in memory
3. Formulas and data provided are simplified representations for the purpose of this exercise

### Table1. Sample data from the Global Beverage Corporation Exchange

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stock Symbol | Type | Last Dividend | Fixed Dividend | Par Value |  |
| TEA | Common | 0 |  | 100 |  |
| POP | Common | 8 |  | 100 |  |
| ALE | Common | 23 |  | 60 |  |
| GIN | Preferred | 8 | 2% | 100 |  |
| JOE | Common | 13 |  | 250 |  |

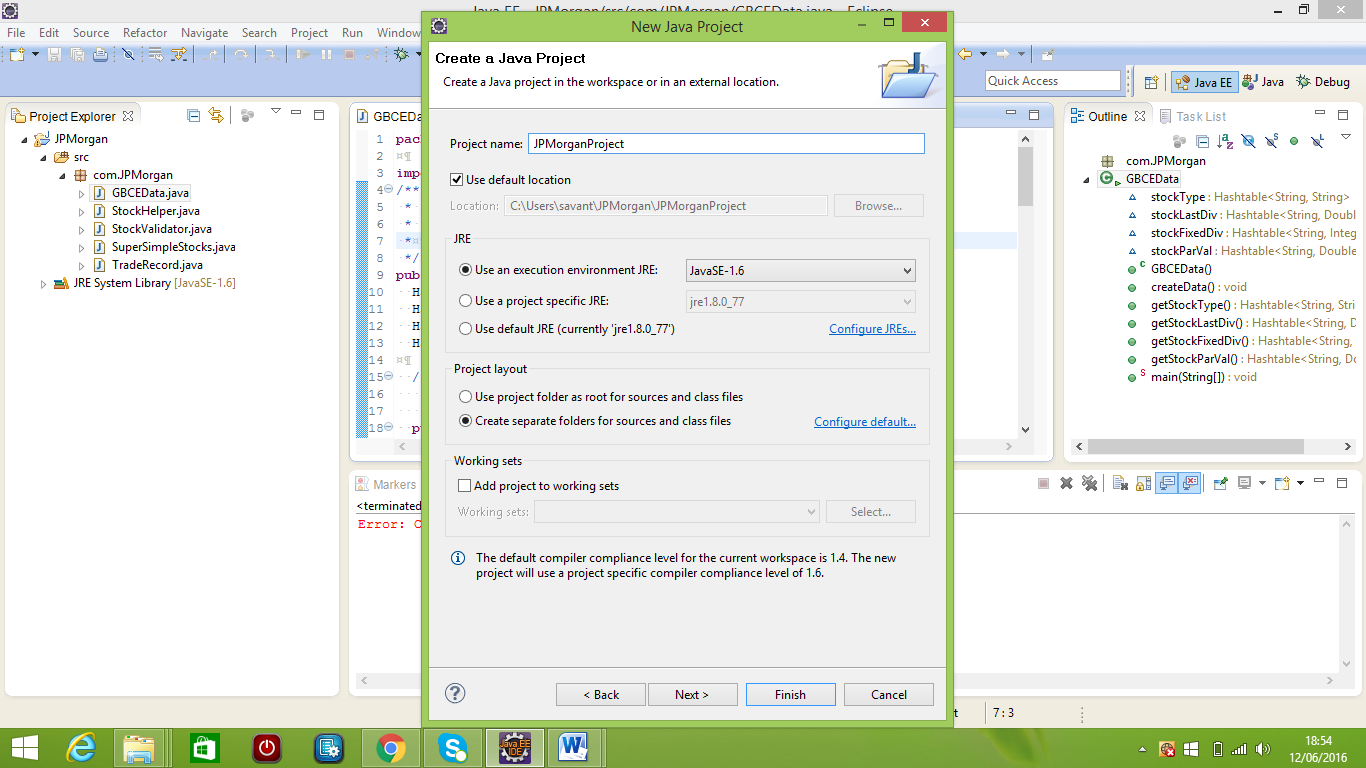
*All number values in pennies*

### Table 2. Formula

|  |  |  |
| --- | --- | --- |
|  | Common | Preferred |
| Dividend Yield |  |  |
| P/E Ratio |  | |
| Geometric Mean |  | |
| Volume Weighted Stock Price |  | |

### Tool used for development and testing:

1. Eclipse Keplier.
2. JavaSE-1.6.



### Summary of Changes

The files involved during the development of assignment and their basic functionality are as follow:

1. GBCEData.java: This class is used to hold the sample data of Global Beverage Corporation Exchange in the memory as a entries of HashTable<K,V>. In this class for each instance of HashTable<K,V>. Stock Symbol are used as the key and the different attributes like Type, Last Dividend ,Fixed dividend and Par Value are used as the value for that respective stock. Each instance has its respective getXXX() method which return the reference of each instance.
2. TradeRecord.java: This class is used to hold the value of different attribute like Stock Symbol, Timestamp, Quantity of shares, Buy or Sell indicator and Trade price for each record.
3. StockHelper.java: Implementation of all the method like getDividentYield(),getPERatio(),printStocks(),geoMetricMeanOfAllStocks() are defined in this class. This class is used to hold the business logic for this application.
4. StockValidator.java: This class is used to validate the values entered by the user.
5. SuperSimpleStocks.java: This is the main class where we initialize the objects of all other classes. In this class I have declared an instance variable allTrades of type Map<String, List<TradeRecord>> which is used to hold the details of all trade related to all stocks. In Map<String, List<TradeRecord>> allTrades the name of stock are used as key and value represent the list of all record related to that stock. When application start the user is prompted to provide the details of stock like Name of stock, market price of stock, quantity of share and the option indicator like whether you want to buy or sell share. If all these input passed the validation stage then these values are used to initialize the object of TradeRecord class.

Once the object is initialized the next step is to check whether that stock already exist in Map<key,value> allTrades. If it exist then we only add the object in the list of trades of that particular stock and if not we first create the entry of that stock in the Map and then add the record in its listArray. Now we have stock name as key and its list of trade record as its value for each stock in Map<key,value> allTrades. By using this Map and its key values we can calculate dividend yield, P/E ratio, Volume Weighted Stock Price and the geometric mean of prices for all stocks

### Output of Application

1. Run the main class (SuperSimpleStocks.java).

Console screen will ask you few details like

1. Enter the Stock name from (TEA,POP,ALE,GIN,JOE):

pop

1. Enter the Market Price of Stock:

44

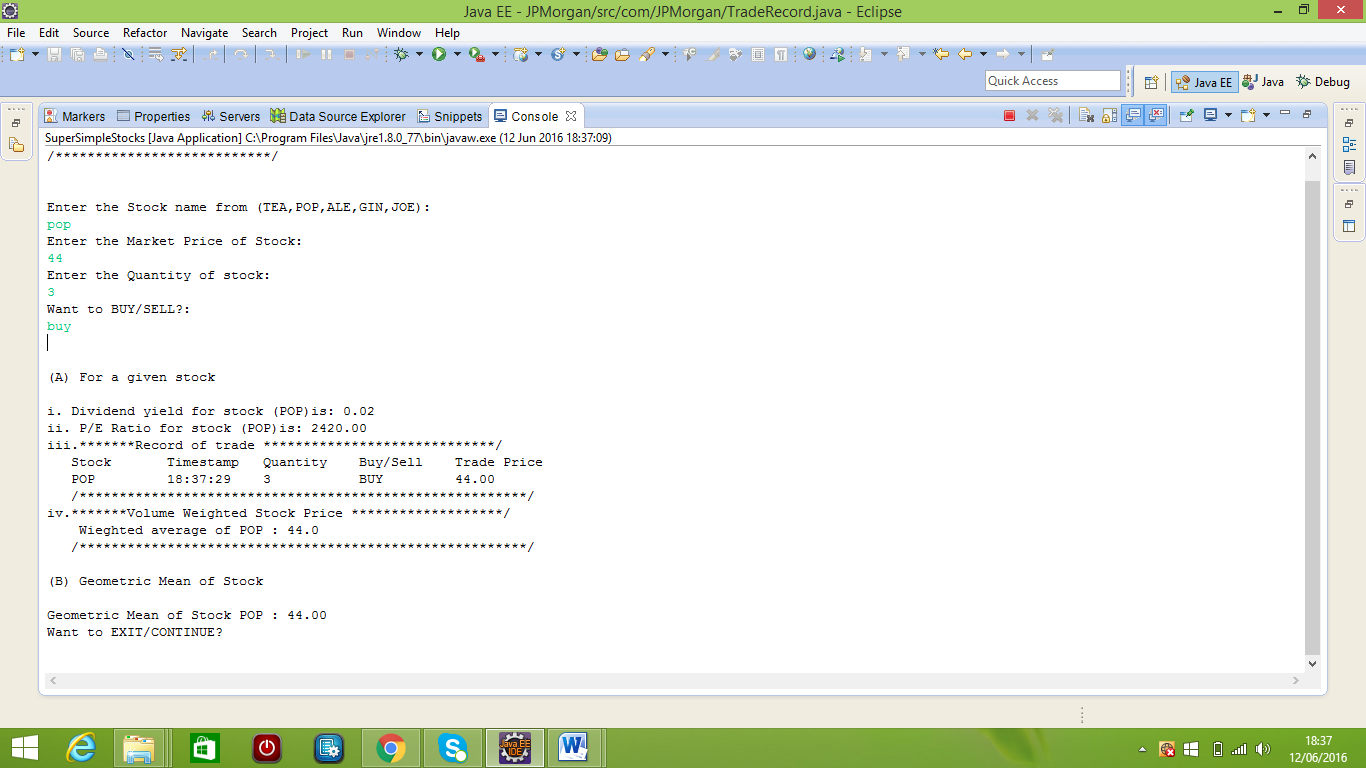
1. Enter the Quantity of stock:

3

1. Want to BUY/SELL?:

Buy

Console will show you the result as follow



1. Want to continue trading type continue and press continue

Same question will appear again

1. Enter the Stock name from (TEA,POP,ALE,GIN,JOE):

tea

1. Enter the Market Price of Stock:

22

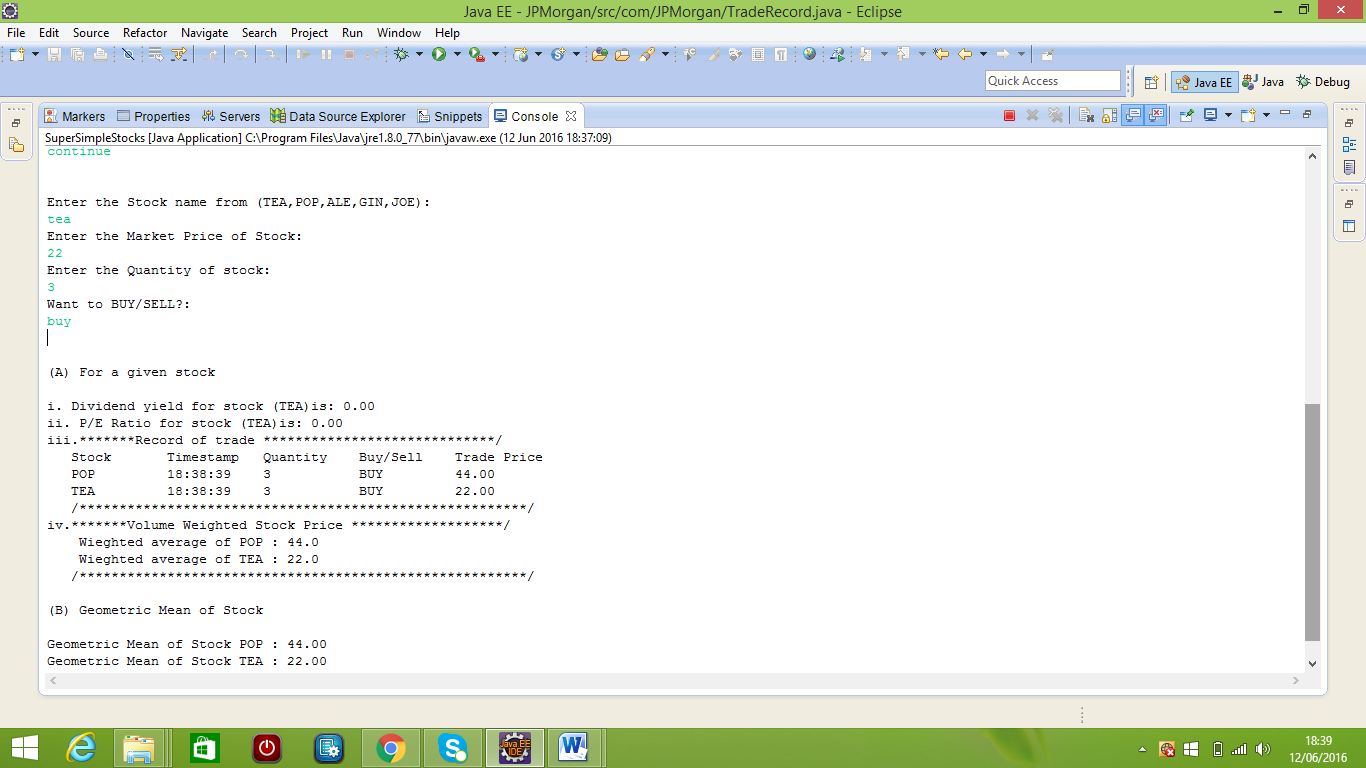
1. Enter the Quantity of stock:

3

1. Want to BUY/SELL ?

Buy

Console will show you the result as follow



1. Want to continue trading type continue and press continue

Same question will appear again

1. Enter the Stock name from (TEA,POP,ALE,GIN,JOE):

pop

1. Enter the Market Price of Stock:

33

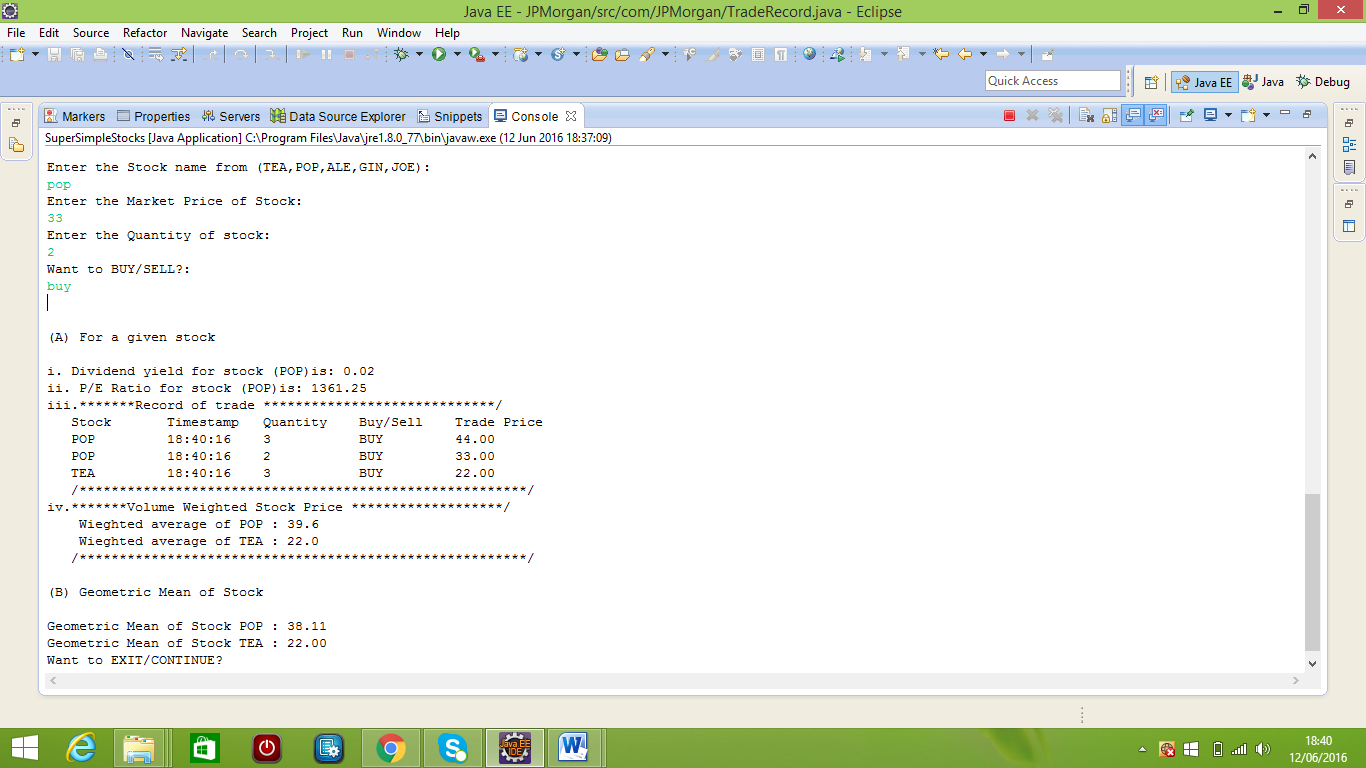
1. Enter the Quantity of stock:

2

1. Want to BUY/SELL?:

Buy

Console will show you the result as follow



1. Do not want to continue type exit and press enter

Console will show the message that Application terminated .If you want to do trading again you have run the main class (SuperSimpleStocks.java) again.

