A Simple Stock Data Analyzer

Goals:

- Learn how to choose a data structure for the desired performance.
- Practice with more Java collections, like HashMap and TreeSet.
- Learn how to customize a comparable data structure.
- Learn how to build an in-memory database for data analysis.

This project can be done with a single partner.

Tasks:

In this project, we would create a tool to conduct simple analysis with real-time stock market data. With the symbol of the New York Stock Exchange stock, one can get its current trading price by extracting the information from a web page. For example, the current stock price of Google (NYSE symbol = GOOG) can be extracted from the page: http://finance.yahoo.com/quote/GOOG.

We would use a Java API, called *YahooFinance*, to obtain the stock price based on this idea. The following code snippet shows how to retrieve the current price of GOOG using the API: (More information about the API can be found here https://financequotes-api.com/ and the library file is included in the starter code).

```java
Stock stock = YahooFinance.get("GOOG");
BigDecimal price = stock.getQuote().getPrice();
```

The tool implements an in-memory database that stores real-time market data in the memory to allow fast retrieval. Each record stored in the database includes the key (i.e., stock symbol) and the value (i.e., company name and current price).

You would create a *myStock* class to implement the database and provide the following functionalities such that users can conduct simple stock data analysis.

- Users can quickly retrieve the current price and company name using the stock symbol.
- Users can quickly insert and update the records.
- Users can quickly retrieve the stock record with top k prices

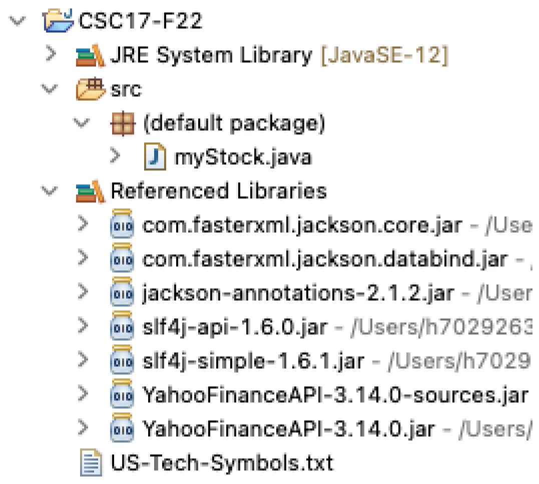
To achieve these functionalities, your task is to implement the following methods in myStock.java which is provided in the starter code. :

```
myStock() // initialize the in-memory database with proper data structures
stockInfo get(String symbol) // get the record using the symbol; runtime is O(1)
void insertOrUpdate(String symbol, stockInfo stock) // insert or update one
record; runtime is at least O(log(n))
List<Map.Entry<String, stockInfo>> top(int k) //returns a list of records with
top k prices; runtime is O(k)
```

A list of stocks is provided in the file *US-Tech-Symbols.txt* where each line contains a stock symbol and the company name. The real-time price for each stock in the file would be retrieved using the YahooFinance API, which would be used by you to initialize the database. According to the test code in myStock.java, the following output is expected (the actual price of the stock may vary).

```
===========Top 10 stocks===========
[1]CCUR CCUR Holdings, Inc. 5250.0
[2]CABO Cable One, Inc. 718.15
[3]EQIX Equinix, Inc. (REIT) 652.86
[4]FICO Fair Isaac Corporation 611.645
[5]AVGO Broadcom Inc. 513.79
[6]MSCI MSCI Inc. 499.68
[7]LRCX Lam Research Corporation 454.71
[8]FDS FactSet Research Systems Inc. 432.77
[9]TDY Teledyne Technologies Incorporated 413.195
[10]INTU Intuit Inc. 399.135
===========Stock info retrieval===========
VMW VMware, Inc. 115.34
BIDU Baidu, Inc. 96.12
```

Please closely follow the comments and hints in myStock.java for details instruction when you are implementing the tool. Please make sure to include the library files for YahooFinance API in your project's build path to allow successful compilation. To do that, right-click your project and choose “Build Path -> Add External Archives”, and select all library files (i.e., the jar files) from your computer. The project structure should look like the following one if everything is added correctly:



```
CSC17-F22
  JRE System Library [JavaSE-12]
  src
    (default package)
      myStock.java
  Referenced Libraries
    com.fasterxml.jackson.core.jar - /Use
    com.fasterxml.jackson.databind.jar - ,
    jackson-annotations-2.1.2.jar - /User
    slf4j-api-1.6.0.jar - /Users/h7029263
    slf4j-simple-1.6.1.jar - /Users/h7029
    YahooFinanceAPI-3.14.0-sources.jar
    YahooFinanceAPI-3.14.0.jar - /Users/
  US-Tech-Symbols.txt
```

Submission:

Please submit your completed myStock.java and a screenshot of your output to Blackboard. You won’t receive full credits if you fail to submit the screenshot.