

## MF Hack Case

Below three tasks will be described in detail. Please complete them to the best of your ability. It is fine if you do not manage to complete them all completely, but make sure you commit and push the code anyways for our review with attached comments regarding your problems.

### Guidelines

- Code in general shall be self explanatory. Use proper naming and structure. The purpose of comments in your code shall be to make notes for other developers to grasp the context.
- The code should be compliant with Google Java Style
  - <https://google-styleguide.googlecode.com/svn/trunk/javaguide.html>
  - <http://code.google.com/p/google-styleguide/source/browse/trunk/eclipse-java-google-style.xml>
  - <http://code.google.com/p/google-styleguide/source/browse/trunk/intellij-java-google-style.xml>
- All dependencies, if necessary, shall be handled by maven.
- Each project should have junit test

### Task 1 - Tools

- Set up and get familiar with git. If you do not have an account at github.com, please create one.
- Create two projects, one for task 2 and one for task 3. Sharing is caring

### Task 2 - Design

- Create your own implementation the java 7 interface *java.util.Map*
  - <http://docs.oracle.com/javase/7/docs/api/java/util/Map.html>
- The internal structure for storing and retrieving key, element pairs shall be your own. You are not allowed to use other Java API constructs for this.
  - There are no specific complexity or time dependencies, so feel free to use something simple as a internal Linked List or Array List structure.
- You are allowed to use existing Java API constructs for the method calls *entrySet*, *keySet* and *values*. I.e. on request iterate over your internal structure and creating a Set or Collection

*hint*

*Don't over do it. We just want to see you use generics and get the idée of polymorphism.*

### Task 3 - Algorithm

The year is 2014 and we have just created a quantum entangled non causal algorithm that can precisely predict a stocks price movements one year ahead with a daily resolution. At great cost we have managed to generate three .csv files containing the stock prices predictions for three different companies. Now we want to make the most of it and cash in! We are however rather lazy and do not wish make that many trades, there are so many other things to do. Look at funny cat videos, floss our dogs teeth and so on. So in order to accommodate all this other activities we have chosen to only make 1 buy and 1 sell trade in each stock.

Your task is to create a software that finds the date and price for when to buy the stock and the date and price to sell the stock in order to give us the best return on our investment.

- The software shall takes a csv file as an inputstream or read the file from disk.
- After it solved the problem, it shall present the date and price for when to buy and sell on the screen as well as the return in percent.

We spent a few minutes pondering the problem and now think the naïve algorithm for finding the solution can be done with the time complexity  $O(n^2)$ . We are however bankers and all our computers are from the 1990s as well as extra slow from too much *bad-porn-site-consumption* and therefore would like to see a solution in  $O(n)$  if it is at all possible.

#### *hint*

- *Do not invent the wheel, there are libs for reading csv files,*  
<https://commons.apache.org/proper/commons-csv/index.html>
- *The return in percent is calculated as follows:*  
`returnInPercent = 100 * (sellPrice - buyPrice) / buyPrice`
- Visualize the data as a regular stock curve. Such can be found on [www.avanza.se](http://www.avanza.se)