Some homework

These are some fairly simple questions that would give us a good idea of your thoughts and programming style. Please answer them as if it was a real development task at work.

Question 1

Your goal is to implement a method finding the most popular ticker (the symbol identifying a stock).

Read the javadoc of `TransactionStatistics` to understand what we mean by popular. Also read the javadoc of `Transaction` to understand the numbers you will be using.
Please do two implementations. One using a single-thread, the other performing the calculation in parallel. If you look at performance, assume a list of 1M entries and 40 different tickers.

Question 2

We have an `Employee` class extending a `Person` class. The task is to write appropriate `hashCode()` and `equals()` methods for both classes.

- Nothing from the source should be modified but additional fields and methods can be added.
- Instances of `Person` and `Employee` should never be equal to one another.
- An `Employee` is equal to another `Employee` if the role, age, and name are all equal between instances.
- A `Person` is equal to another `Person` instance if the age and name are both equal between instances.

Question 3

Implement the `Statistic` interface in other to provide an implementation that calculate statistics for an infinite flow of events. You can do many implementations.
The rules:

- Implementations of this class must be 'threadsafe'. The definition of the term 'threadsafe' is left up to you.
- Implementations may choose to prioritize (or not) various aspects of the performance and or behavior of their instances this is acceptable as long as the compromise can be explained and justified.
- You are free to choose appropriate behavior for any corner cases but are expected to be able to justify those behaviors.
- If you want to optimize for ‘performance’ (whatever you take performance to mean) here at the expense of code readability then that is fine, but you’ll need to be able to explain the code to us. If something appears counterintuitive or overly complex then code comments can be useful.