Take Home Java Coding Project

Overview

This is a Java software development problem which you should complete by yourself at home as part of your interview at Raytheon BBN Technologies. We'll build and run your code, then meet again to ask questions and discuss alternate approaches.

The Goal

Our team at BBN often deals with open-ended and difficult research problems which need practical solutions on a short timeline. The challenge below is supposed to gauge your ability to code in that environment by translating ambiguous requirements into a functioning prototype. You need to provide the correct answer, but it's equally important to document and defend your decisions while demonstrating that you've considered alternatives. It's also okay to show off your expertise when the problem calls for it!

Guidelines

- We believe this should require less than five hours. Please **keep track of and report how much time you actually spend** (less is perfectly okay).
- Please **don't let this exercise impact your personal life**. If you see any issue finding the time, let us know and we'll work with you to arrange something else. Seriously.
- Consult any source of information that you like, but **cite all sources** - the overall product must be your own. You may be able to find an exact solution on the internet; please don't copy.
- You must **use Java**, but you may use any development environment, language features, software libraries, or existing code that you own. Remember: the point is to demonstrate software proficiency and understanding of the problem.
- Remember that good code is clean, reusable, extensible, testable, and documented. **Consider how your code might change** for a new use case or different inputs.
- If possible, **place your code in a GitHub repository** so we can access it and run it during the in-person interview. If not, just email it to us beforehand.

The Problem

Imagine that you're part of a software team building a basic Calendar/Scheduling application. While developing a portion of the "Create Meeting" subsystem, you learn that the user has documented their needs as follows:

> *"I need help counting meeting occurrences. If I schedule on Wednesdays, how many meetings will we have between today and the end of 2019? If we move to Thursdays next year, how many meetings will we have in total? We might have to miss certain dates for vacation or holidays."*

Task 1: Design

Provide a Java Interface which defines a reusable software component to help meet the user's needs above. Document how other developers would interact with your interface and give examples.

Task 2: Implementation

Provide two implementations for the Interface from Task 1 which could help answer the user's questions. One implementation should be "quick and dirty" for the MVP (minimum viable product); but the second implementation gives you a chance to show off to your team. Make sure to provide documentation for when users should select a particular implementation. If there's spare time feel free to add extra functionality or more options but be prepared to justify the extra effort.

Task 3: Delivery

Add tests or examples as necessary to actually provide answers to the user's questions using the code you developed on Task 2.

Document differences between your two approaches. Have you covered other potential inputs or configuration? Have you anticipated future use cases?

Input

Your program should be able to process the following file:

```
# input.csv
# start, end, day of week
2018-05-02, 2018-12-31, Wednesday
2019-01-01, 2019-12-31, Thursday
```

In-Person

During your code review, we'll go over your product and usually ask questions in the following areas:

- Similarities to other projects you've worked on
- Design decisions
- Implementation choices
- Any defects or potential defects we find
- Reasoning about algorithms, data structures, or performance
- Necessary updates for new user needs
- Potential improvements
- Feedback on the problem itself

Feel free to include any notes or other reference materials for discussion, and please don't hesitate to ask if you have any questions!