2018 Java Coding Competition

The Problem

As a new developer at SF Analysis, a project has been provided to assist in the implementation of an application using geographical data provided by www.openstreetmap.org - © OpenStreetMap contributors. We have pulled data from a Small Metro – Bloomington, IL and a Large Metro – Chicago, IL. The problem is separated in multiple pieces and allows you to build confidence as you pass the simple problem of parsing the data, complete data queries, etc. There are some basic requirements (JUnits) that must be completed before you are able to showcase your creativity in using the data.

StreetMapDataInterpreterTest

- Read the small file – small-metro.xml (canInterpretSmallMetro)
- Read the large file – large-metro.xml (canInterpretLargeMetro)
- Using search criteria search xml files

PointOfInterestParserTest

- Read latitude from xml file (canInterpretLatitudeFromDataSource)
- Read longitude from xml file (canInterpretLongitudeFromDataSource)
- Read tag information from xml file (canInterpretTagsWithinNodes)

JUnit tests are the highest authority associated to requirements and can be found by following the path *src/test/java*. Additional details/comments are found in the JUnits.

********* **Do not change anything in the JUnit tests!** **********

First Actions:

- Import the problem statement into your IDE.
- We have provided Maven dependency for JUnit 4. If you are not set up with the recommended IDE (Spring Tool Suite), you may need to add JUnit 4.
- If you identify any additional libraries you would like to use, please add them to the pom.xml file or copy the .jar files into the resources folder
- Run your JUnit tests, code, and repeat. “How to run JUnits”

When you are done:

- Update the feedback.txt file and include the following information:
  - Your team – name of each individual participating.
  - How many JUnits you were able to execute successfully.
  - Document and describe the additional “nice to have” features included, to help the judges properly grade your submission and explain how to properly execute new enhancements.

- Push your changes to one single branch for you and your teammate. Open a single pull request against the main State Farm Coding Competition repository before 11:59PM CST on October 14, 2018.
  - If you make any commits after midnight without prior approval from codingcompetition@statefarm.com, your submission will be disqualified.
  - If you so choose, you may open a pull request at any time during the competition and continue to update it as long as you do not make any commits after midnight.

2018 Java Coding Competition

Rules

- Contestants cannot seek help from individuals outside their team.
- Teams are expected to have the necessary tools and JARs preloaded on their machines **prior** to the competition.
- If you believe this document and the JUnit tests conflict, the JUnit tests are the highest authority.

How you will be Graded

Components of submission will be weighted as follows.

- 100% core requirements met, including:
  - Number of JUnits that pass using correct functionality in the program
  - Maintaining Object Oriented Programming principles
  - Code documentation
  - Code must compile and execute
- Do not complete any Bonus unless you have all the JUnit tests completed
  - Bonus credit awarded for any extra features added (up to 10%)

In the event of a tie, we will further judge your solution based on: code cleanliness, maintainability, and adherence to object-orientated principles.

<u>Bonus "Nice to have" features</u>:

- Bonus Credit – Do not complete any bonus features unless you have completed all the required functionality and all JUnits pass.
  - Create additional functionality and demonstrate through a Graphical User Interface (GUI), command line interface, JUnits, or web UI.
  - If you can think of any other useful features to add, we appreciate ingenuity and will gladly accept any useful enhancements. Be sure to document any bonus features in the feedback text file.

2018 Java Coding Competition

If you decide to create a GUI or print info from the data provided – you must give credit to www.openstreetmap.org - **© OpenStreetMap contributors.** This credit needs to appear in a place that is reasonable to the medium or means you are utilising. In other words, you should expect to credit OpenStreetMap in the same way and with the same prominence as would be expected by any other map supplier. Therefore:

- For a **printed map**, the credit should appear beside the map if that is where other such credits appear, and/or in the "acknowledgements" section of the publication (often at the start of a book or magazine).

- For a **browsable electronic map** (e.g. embedded in a web page or mobile phone application), the credit should typically appear in the corner of the map, as commonly seen with map APIs/libraries such as Google Maps.

- As an **electronic database or collection of databases, file, CD**, Place the text in the file itself or in an extra file.

- A **video or computer game -** on the credits page or in the game view.

- If you are producing library code that offers OpenStreetMap data or tiles, you should make sure library users are aware of these terms. We strongly recommend that you display this credit by default when your library is used.

See https://wiki.osmfoundation.org/wiki/Licence/Licence_and_Legal_FAQ for more details.