Lab Assignment 1: Semantic Web

Goal: *Be able to create in-memory and persisted models with Jena.*

Download the latest release of Jena:
https://jena.apache.org/download/index.cgi
and extract to a convenient directory such as c:\extra\ (make the c:\extra directory if you don't already have one) or your documents or project directory. The extraction makes a subdirectory apache-jena-x.x.x (where x.x.x is the version).

1. Read through and understand the entire Jena RDF Tutorial:
https://jena.apache.org/tutorials/rdf_api.html

- Execute tutorial 3 (in the Statements section) in the Eclipse debugger (copy source code from the web pages – select Raw first). You will need to have your Eclipse project point to the jars in the /lib directory below /apache-jena-2.12.1 and add to the appropriate package.

- If your compiler doesn't like the 'extends Object' phrase, delete it—it is implicit in Java.

- If you get complaints that log4j isn't initialized, add an initialization line near the top of main:

```
org.apache.log4j.Logger.getRootLogger().
     setLevel(org.apache.log4j.Level.OFF);
```

- Confirm answer is correct.

2. Create an in-memory Jena model (the default model).

- Use vCard RDF (http://www.w3.org/TR/vcard-rdf) to represent (hardcoded) that "Dr. Viera Chandler" is a vCard individual with the title "South West Division President" of the company "Allied Semantics, U.S." who was born on "January 15, 1964". She has e-mail address vechandler@alliedsem.com. Viera should be represented as a resource using "http://utdallas/semclass" as the namespace and 534772 as the resource ID.

  Jena has some built-in vCard vocabulary you may use as described at:
  https://jena.apache.org/documentation/javadoc/jena/com/hp/hpl/jena/vocabulary/VCARD.html
  and example usage at:

http://jena.apache.org/tutorials/rdf_api.html

- Output the model in the following three formats to the indicated filename in the default execution directory:

Output Format	Output filename*
1) RDF/XML	Lab1p2_<YourID>.xml
2) N-TRIPLE	Lab1p2_<YourID>.ntp
3) N3	Lab1p2_<YourID>.n3

* <YourID> should be replaced with your UTD NetID.

NOTE: Your Eclipse project's default execution directory is the root of your project directory such as "…/Documents/workspace/Lab1" where "Lab1" is the Eclipse project directory.

Be certain to open the .xml file (with an editor or within eclipse) to see and understand the automated tree structure it provides for xml data.

The above parts of the lab (i.e. parts 1 and 2) are not submitted for grading. It is, however, the foundation for part 4, below.

3. You must create your own Friend-Of-A-Friend (FOAF) profile using FOAF-a-matic. Here are the steps on how to do this:

a. First step, create your own FOAF profile using FOAF-a-matic, located here:
http://www.ldodds.com/foaf/foaf-a-matic
Fill out all fields. For the homepage field, use your UTD home page. For the picture, just make up a file name for a picture (e.g: JohnDoe.jpg) or select some small, suitable JPG file, copy, and rename. For phone number, you don't have to supply your actual phone number—use a fake 555 number. The important thing is to fill out this field.

For the work fields, fill out the work homepage and page describing what you do. For school homepage, use:
http://www.utdallas.edu

For people you know, add any 3 names. They don't have to be your actual friends—they can be fake names. The important part for your "friends" is the email fields—make at least one of the friend emails be a ".gmail.com" account and another be a ".utdallas.edu" account.

The "See also" field could be any URL, real or fake.

b. When you have filled out all fields, be sure to **uncheck** the checkbox with the label "**Protect email addresses from spammers**". If you fail to do this, reset the box and do it again. Otherwise, your generated FOAF file (see step c below) will generate the property tags:
<foaf:mbox_sha1sum> </foaf:mbox_sha1sum>
instead of:
<foaf:mbox> </foaf:mbox>
You will need the "foaf:mbox" tags to properly execute your code.

c. Press the "**FOAF me!**" button. The corresponding FOAF profile given your answers in the form will appear in the text area.

d. Copy the FOAF code to a text file and name it:
<YourID>_FOAFFriends.rdf
where <YourID> is your UTD NetID. This file will be used in part 4 below.

e. Copy your FOAF file into your Eclipse project at the top-level execution directory.

4. Modify the code for Part 2 (the Jena model above) to use a TDB persistence model and add your FOAF profile to the previous data. TDB is a persistence file store built into the current version of Jena.

a. Modify the code you wrote for Part 2 to use an un-named TDB model rather than the in-memory model.

Documentation on the use of a TDB store is described at:
https://jena.apache.org/documentation/tdb/java_api.html

Here are some useful Jena definitions (or synonyms):
Dataset: a collection of one, unnamed, default graph and zero, or more named graphs
https://jena.apache.org/documentation/tdb/datasets.html
A Dataset consists of:
- The node table
- Triple and Quad indexes
- The prefixes table

https://jena.apache.org/documentation/tdb/architecture.html

In configuring TDB, use the "directory name approach" where the database directory is "MyDatabases/Dataset1" (i.e., same as the TDB examples and relative to the execution directory). You need to manually create the MyDatabases/ directory at the proper location.

NOTE: You need to put the transaction mechanisms `dataset.begin()` and `dataset.end()` method calls before and after all the model activities. If you do an `end()` somewhere in the midde, the next time you do a `begin()` you must get the model again. You can do a `commit()` without having to get the model again. See the TDB sample code for hints on how your code should be re-structured.

b. Add code to import your FOAF profile (created in part 3 above). As an example, the "Hello Semantic Web Demonstration" (Class Demo 1) program covered in class imports data into a model near the top of its code.

c. Modify the code further to use a named graph ("myrdf") stored in the TDB model rather than the un-named model. This involves changing the `getDefaultModel()` method call to `getNamedModel()`. These calls are documented in the Javadoc at:
http://jena.apache.org/documentation/javadoc/arq/org/apache/jena/query/Dataset.html

d. As before, output the model in the following three formats to the indicated filename in the default execution directory:

Output Format	Output filename*
1) RDF/XML	Lab1p4_<YourID>.xml
2) N-TRIPLE	Lab1p4_<YourID>.ntp
3) N3	Lab1p4_<YourID>.n3

* <YourID> should be replaced with your UTD NetID.

Submission:

Please compress your final Eclipse project for <u>Part 4</u> into one zip file and submit the zip file on eLearning. Name your main java file "Lab1p4.java". Include any files you created for compiling and running your program. Include the final three model files as well.

Grading (100 points):

Doesn't Run: -10
Has Log4j runtime complaint: -3
No Summary Report: -10
Lacking vCard data in (program or) result file: -10
Lacking FOAF data in (program or) result file: -10
Incorrect output filenames: -5
Incorrect source filenames: -5
Lacking use of TDB: -10
Nothing submitted: -100