

Assignment 6

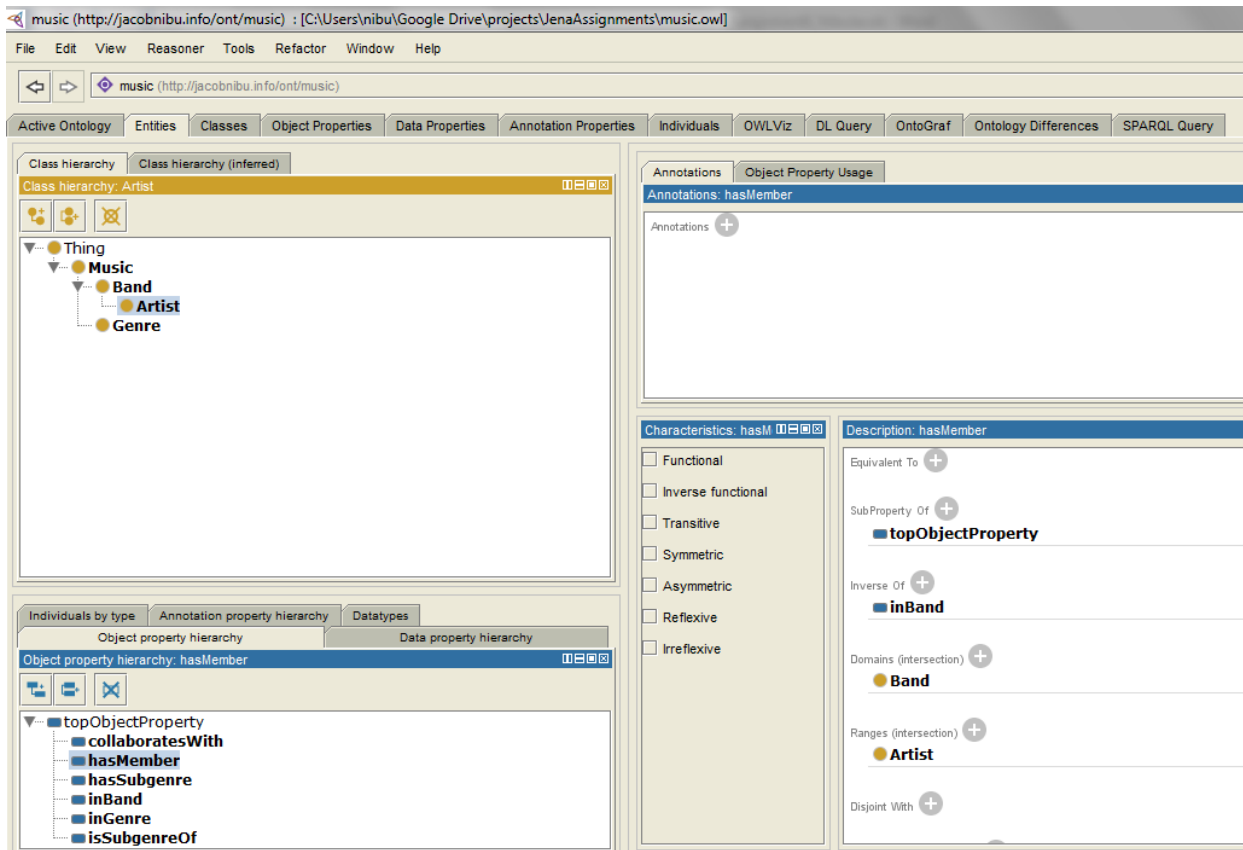
Task: Java code on using reasoner and sparql queries, submit codes and screenshots

Major steps in implementation

1. Create an OWL ontology in Protégé and store in file
2. Create an RDF data model using Jena and store in file
3. Create Java class in Eclipse for loading persistent ontology and data into an Inference model
4. Validate the inference model using OWL reasoner
5. Create Java class in Eclipse to query this model using sparql and output results on console

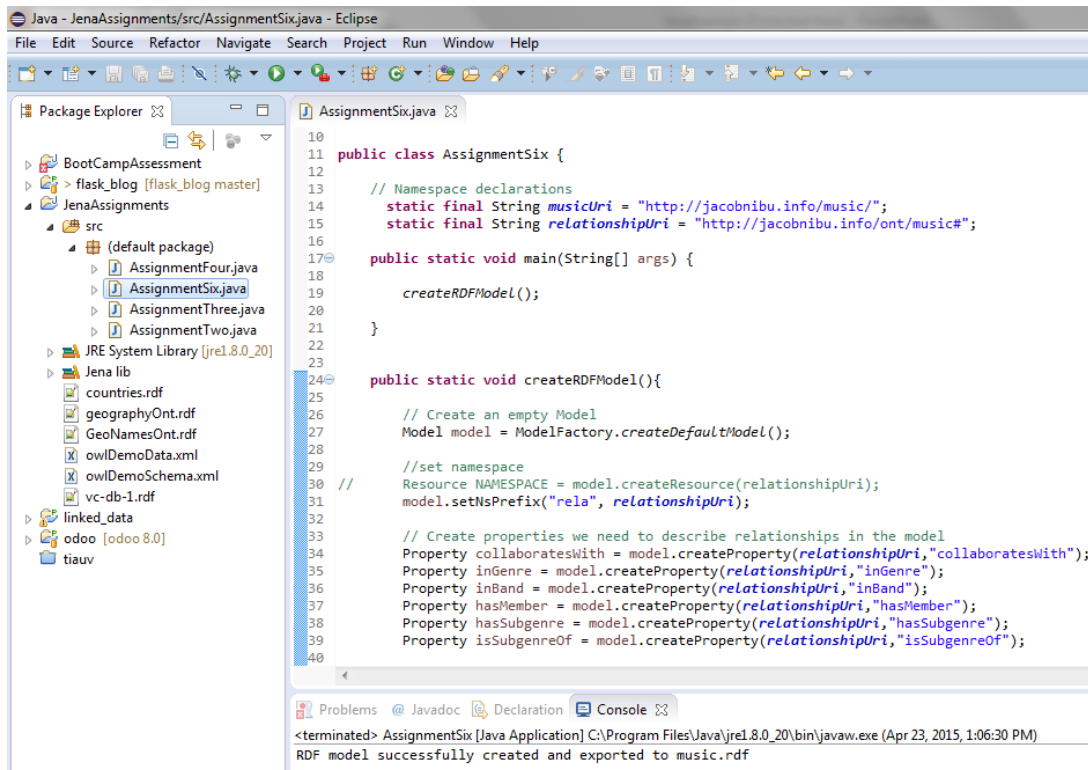
Step 1

Using protégé, created a simple ontology for music. This will have three classes: Artist, Genre, and Band (the group the artist belongs to). It will have the properties: collaboratesWith, inGenre, inBand, hasMember, hasSubgenre, and isSubgenreOf.



Step 2

Using Eclipse, created a java class named AssignmentSix.java that has a method that makes an RDF model and stores it on the disk:



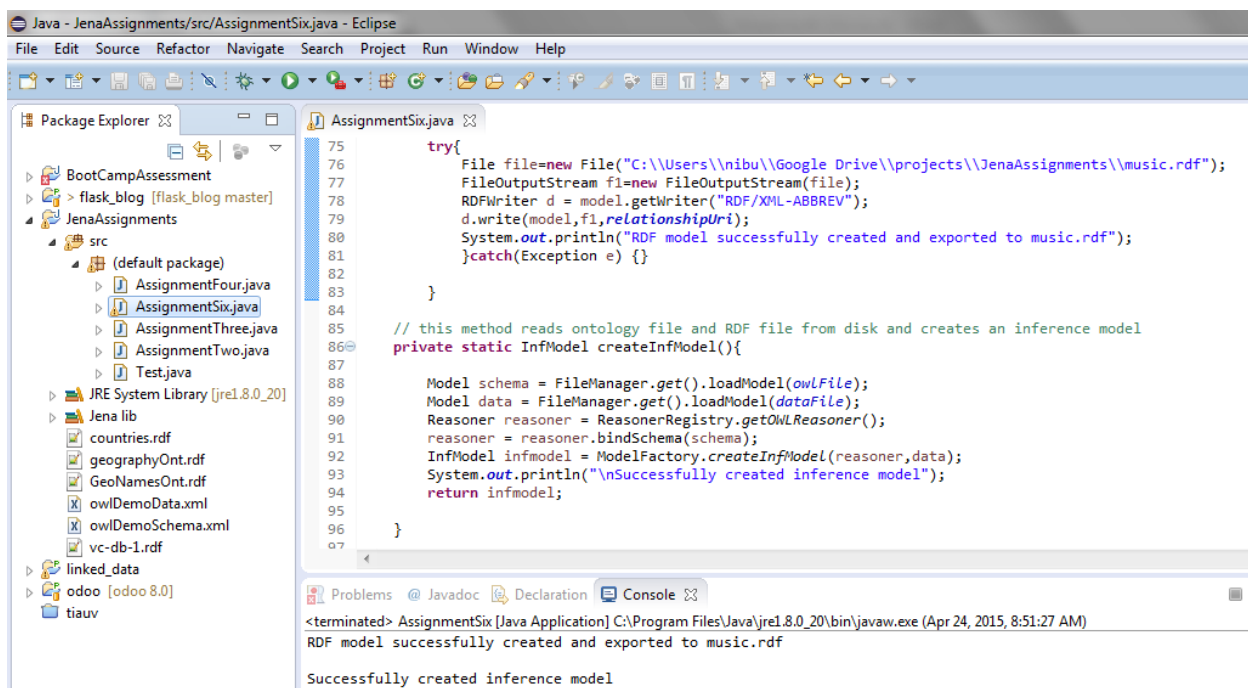
```
10
11 public class AssignmentSix {
12
13     // Namespace declarations
14     static final String musicUri = "http://jacobnibu.info/music/";
15     static final String relationshipUri = "http://jacobnibu.info/ont/music#";
16
17     public static void main(String[] args) {
18
19         createRDFModel();
20
21     }
22
23
24     public static void createRDFModel(){
25
26         // Create an empty Model
27         Model model = ModelFactory.createDefaultModel();
28
29         //set namespace
30         Resource NAMESPACE = model.createResource(relationshipUri);
31         model.setNsPrefix("rela", relationshipUri);
32
33         // Create properties we need to describe relationships in the model
34         Property collaboratesWith = model.createProperty(relationshipUri, "collaboratesWith");
35         Property inGenre = model.createProperty(relationshipUri, "inGenre");
36         Property inBand = model.createProperty(relationshipUri, "inBand");
37         Property hasMember = model.createProperty(relationshipUri, "hasMember");
38         Property hasSubgenre = model.createProperty(relationshipUri, "hasSubgenre");
39         Property isSubgenreOf = model.createProperty(relationshipUri, "isSubgenreOf");
40
41     }
42 }
```

Problems @ Javadoc Declaration Console

<terminated> AssignmentSix [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (Apr 23, 2015, 1:06:30 PM)
RDF model successfully created and exported to music.rdf

Step 3

Added a new method in the java class that loads the owl ontology file and the RDF data file from disk and builds an inference model using OWL reasoner:



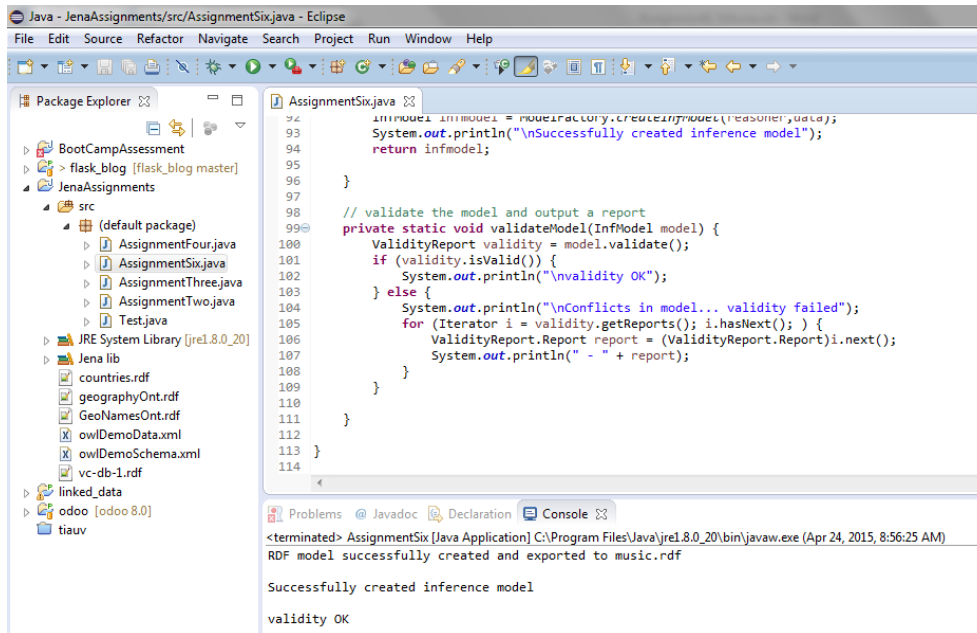
```
75     try{
76         File file=new File("C:\\Users\\nibu\\Google Drive\\projects\\JenaAssignments\\music.rdf");
77         FileOutputStream f1=new FileOutputStream(file);
78         RDFWriter d = model.getWriter("RDF/XML-ABBREV");
79         d.write(model,f1,relationshipUri);
80         System.out.println("RDF model successfully created and exported to music.rdf");
81     }catch(Exception e) {}
82
83
84
85     // this method reads ontology file and RDF file from disk and creates an inference model
86     private static InfModel createInfModel(){
87
88         Model schema = FileManager.get().loadModel(owlFile);
89         Model data = FileManager.get().loadModel(dataFile);
90         Reasoner reasoner = ReasonerRegistry.getOWLReasoner();
91         reasoner = reasoner.bindSchema(schema);
92         InfModel infmodel = ModelFactory.createInfModel(reasoner,data);
93         System.out.println("\nSuccessfully created inference model");
94         return infmodel;
95     }
96
97 }
```

Problems @ Javadoc Declaration Console

<terminated> AssignmentSix [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (Apr 24, 2015, 8:51:27 AM)
RDF model successfully created and exported to music.rdf
Successfully created inference model

Step 4

Created a method in the class to validate this inference model:



```
Java - JenaAssignments/src/AssignmentSix.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  BootCampAssessment
  flask_blog [flask_blog master]
  JenaAssignments
    src
      (default package)
        AssignmentFour.java
        AssignmentSix.java
        AssignmentThree.java
        AssignmentTwo.java
        Test.java
  JRE System Library [jre1.8.0_20]
  Jena lib
    countries.rdf
    geographyOnt.rdf
    GeoNamesOnt.rdf
    owlDemoData.xml
    owlDemoSchema.xml
    vc-db-1.rdf
  linked_data
  odoo [odoo 8.0]
  tiavv

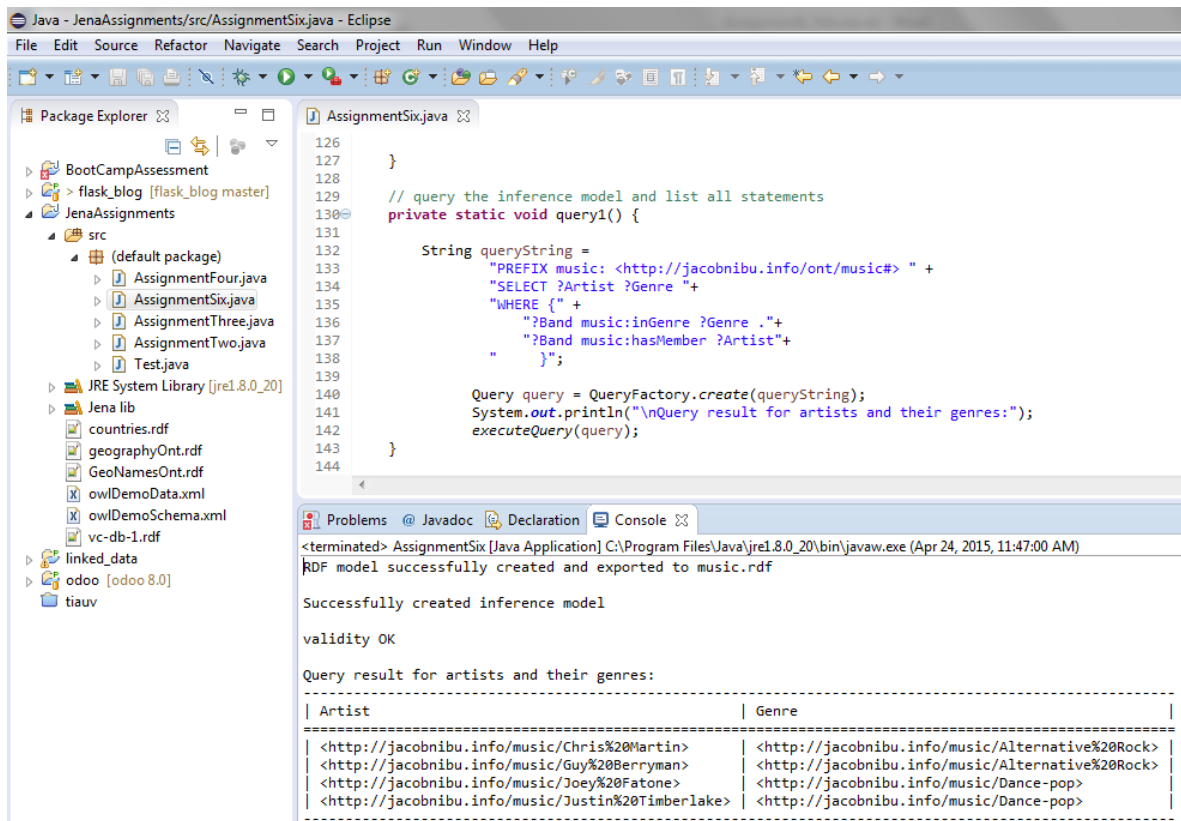
AssignmentSix.java
93 infmodel = modelFactory.createInferenceModel(reasoner, data);
94 System.out.println("\nSuccessfully created inference model");
95 return infmodel;
96 }
97
98 // validate the model and output a report
99 private static void validateModel(InfModel model) {
100     ValidityReport validity = model.validate();
101     if (validity.isValid()) {
102         System.out.println("\nValidity OK");
103     } else {
104         System.out.println("\nConflicts in model... validity failed");
105         for (Iterator i = validity.getReports(); i.hasNext(); ) {
106             ValidityReport.Report report = (ValidityReport.Report)i.next();
107             System.out.println(" - " + report);
108         }
109     }
110 }
111 }
112 }
113 }
114 }

Problems Javadoc Declaration Console
<terminated> AssignmentSix [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (Apr 24, 2015, 8:56:25 AM)
RDF model successfully created and exported to music.rdf

Successfully created inference model
validity OK
```

Step 5

Created methods in the class to query the model using SPARQL. The RDF data model created by the class had 10 RDF statements and none of them state the genres of artists. The first query fetches the artists and their genres inferred by the reasoner:



```
Java - JenaAssignments/src/AssignmentSix.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  BootCampAssessment
  flask_blog [flask_blog master]
  JenaAssignments
    src
      (default package)
        AssignmentFour.java
        AssignmentSix.java
        AssignmentThree.java
        AssignmentTwo.java
        Test.java
  JRE System Library [jre1.8.0_20]
  Jena lib
    countries.rdf
    geographyOnt.rdf
    GeoNamesOnt.rdf
    owlDemoData.xml
    owlDemoSchema.xml
    vc-db-1.rdf
  linked_data
  odoo [odoo 8.0]
  tiavv

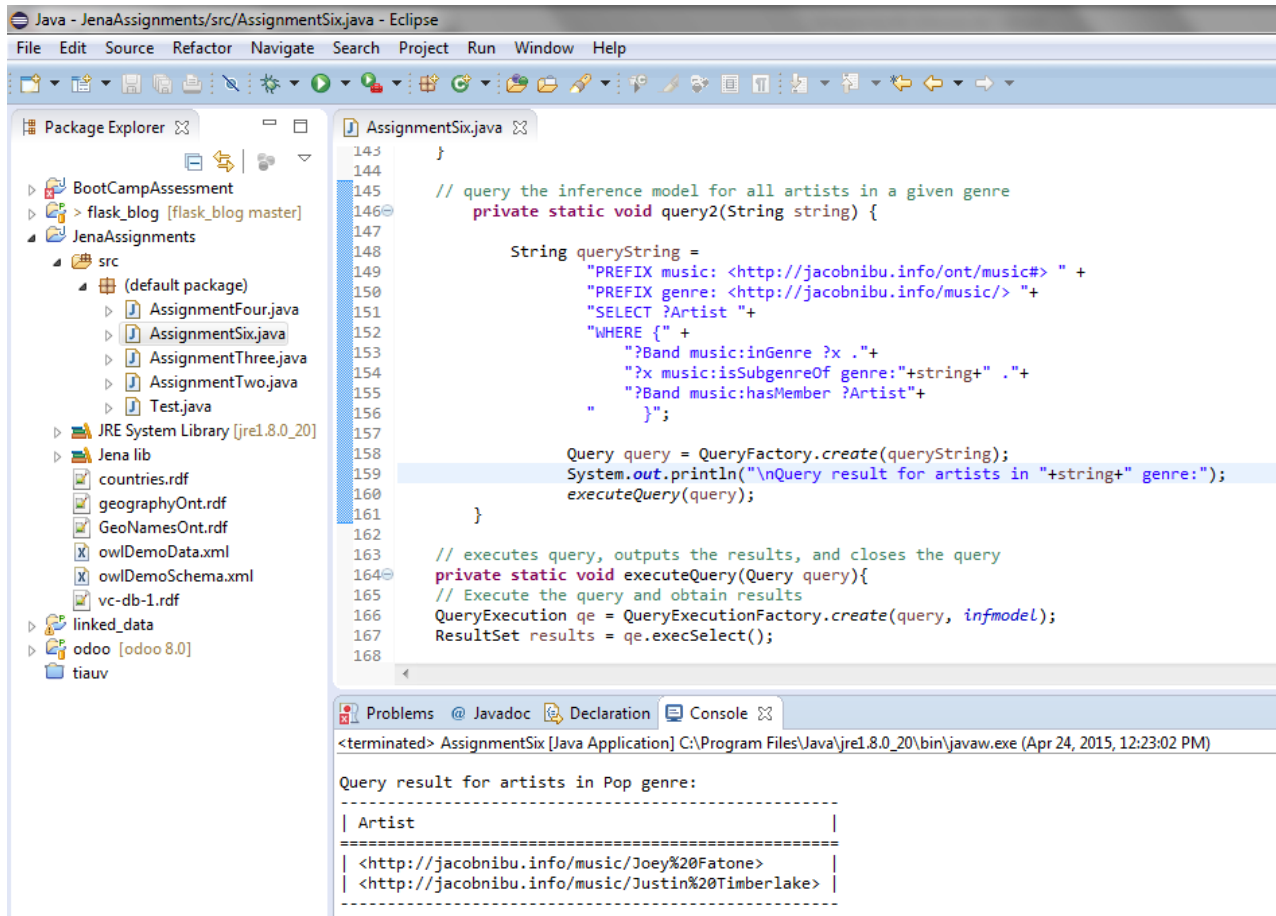
AssignmentSix.java
126 }
127 }
128 }
129 // query the inference model and list all statements
130 private static void query1() {
131
132     String queryString =
133         "PREFIX music: <http://jacobnibu.info/ont/music#> " +
134         "SELECT ?Artist ?Genre " +
135         "WHERE {" +
136             "?Band music:inGenre ?Genre ." +
137             "?Band music:hasMember ?Artist"+
138         " }";
139
140     Query query = QueryFactory.create(queryString);
141     System.out.println("\nQuery result for artists and their genres:");
142     executeQuery(query);
143 }
144 }

Problems Javadoc Declaration Console
<terminated> AssignmentSix [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (Apr 24, 2015, 11:47:00 AM)
RDF model successfully created and exported to music.rdf

Successfully created inference model
validity OK

Query result for artists and their genres:
-----|-----
| Artist | Genre |
-----|-----
| <http://jacobnibu.info/music/Chris%20Martin> | <http://jacobnibu.info/music/Alternative%20Rock> |
| <http://jacobnibu.info/music/Guy%20Berryman> | <http://jacobnibu.info/music/Alternative%20Rock> |
| <http://jacobnibu.info/music/Joey%20Fatone> | <http://jacobnibu.info/music/Dance-pop> |
| <http://jacobnibu.info/music/Justin%20Timberlake> | <http://jacobnibu.info/music/Dance-pop> |
-----|-----
```

A second SPARQL query fetches all artists in the genre given as argument. The result for argument “Pop”, which was only stated as a superclass of Dance-pop in the data, is:



```
Java - JenaAssignments/src/AssignmentSix.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  BootCampAssessment
  flask_blog [flask_blog master]
  JenaAssignments
    src
      (default package)
        AssignmentFour.java
        AssignmentSix.java
        AssignmentThree.java
        AssignmentTwo.java
        Test.java
  JRE System Library [jre1.8.0_20]
  Jena lib
    countries.rdf
    geographyOnt.rdf
    GeoNamesOnt.rdf
    owlDemoData.xml
    owlDemoSchema.xml
    vc-db-1.rdf
  linked_data
  odoo [odoo 8.0]
  tiauv

AssignmentSix.java
143
144
145 // query the inference model for all artists in a given genre
146 private static void query2(String string) {
147
148     String queryString =
149         "PREFIX music: <http://jacobnibu.info/ont/music#> " +
150         "PREFIX genre: <http://jacobnibu.info/music/> "+
151         "SELECT ?Artist "+
152         "WHERE {" +
153             "?Band music:inGenre ?x ." +
154             "?x music:isSubgenreOf genre:"+string+" ." +
155             "?Band music:hasMember ?Artist"+
156         " }";
157
158     Query query = QueryFactory.create(queryString);
159     System.out.println("\nQuery result for artists in "+string+" genre:");
160     executeQuery(query);
161 }
162
163 // executes query, outputs the results, and closes the query
164 private static void executeQuery(Query query){
165     // Execute the query and obtain results
166     QueryExecution qe = QueryExecutionFactory.create(query, infmodel);
167     ResultSet results = qe.execSelect();
168 }

Problems @ Javadoc Declaration Console
<terminated> AssignmentSix [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (Apr 24, 2015, 12:23:02 PM)

Query result for artists in Pop genre:
-----
| Artist |
-----
| <http://jacobnibu.info/music/Joey%20Fatone> |
| <http://jacobnibu.info/music/Justin%20Timberlake> |
-----
```

Attachments

The java code (AssignmentSix.java), the OWL file (music.owl), and the RDF file (music.rdf) are attached along with this report.