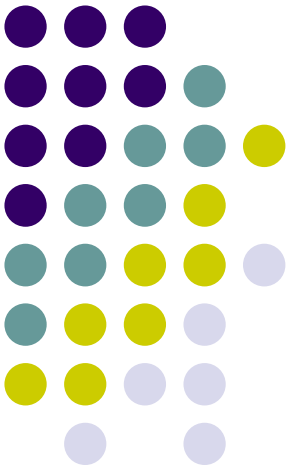
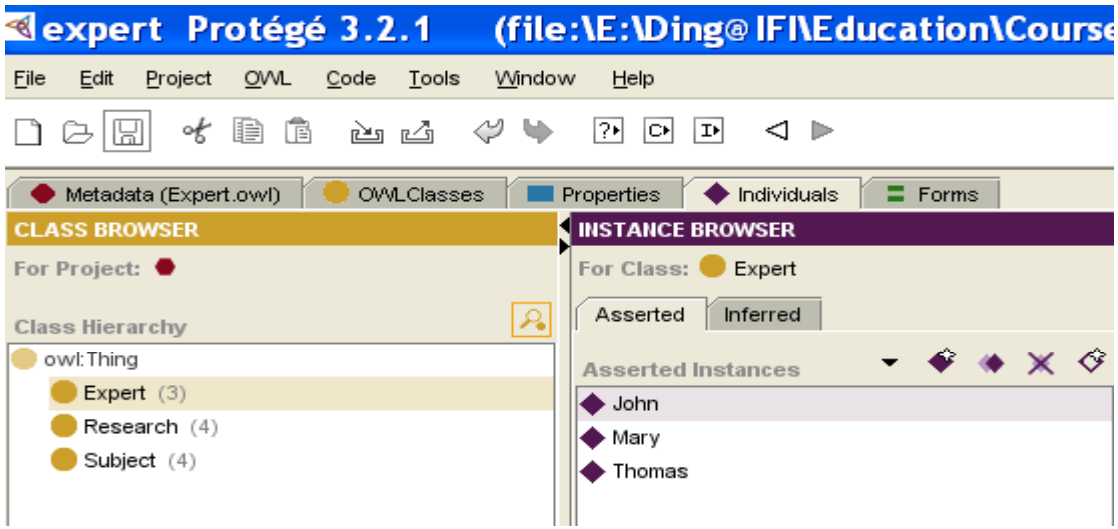
Expert

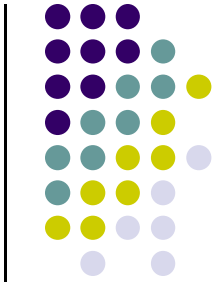


Expert

- Using Protege to build up expert.owl:
  - Three classes: Expert, Research, Subject
  - Properties: hasResearch: expert can have many researches; research is associated with many subjects
  - Create some instances.
  - Output this ontology as expert.owl (for example: C:\Jena\Tutorial\expert\expert.owl)

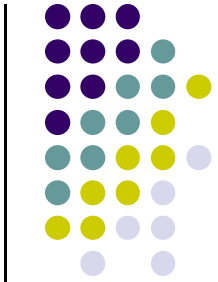


ontologyDB



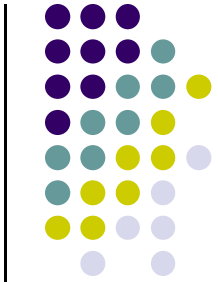
- Introduce some methods to handle store, read ontology in/from persistent database (here takes MySQL as example):
  - connectDB
  - createDBModelFromFile
  - getModelFromDB
  - getModelSpec

ontologyDB



```
import java.util.*;
import com.hp.hpl.jena.db.*;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.*;

public class ontologyDB {

/* link database */
public static IDBConnection connectDB(String DB_URL, String DB_USER, String
DB_PASSWD, String DB_NAME) {
        return new DBConnection(DB_URL, DB_USER, DB_PASSWD, DB_NAME);}

/* Read ontology from filesystem and store it into database */
public static OntModel createDBModelFromFile(IDBConnection con, String name,
String filePath) {
        ModelMaker maker = ModelFactory.createModelRDBMaker(con);
        Model base = maker.createModel(name);
        OntModel newmodel =
        ModelFactory.createOntologyModel( getModelSpec(maker), base );
        newmodel.read(filePath);
        return newmodel; }
```

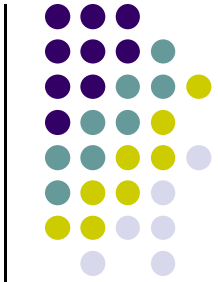
ontologyDB



```
/* Get ontology from database */
public static OntModel getModelFromDB(IDBConnection con, String name) {
        ModelMaker maker = ModelFactory.createModelRDBMaker(con);
        Model base = maker.getModel(name);
        OntModel newmodel =
        ModelFactory.createOntologyModel( getModelSpec(maker), base);
        return newmodel; }

public static OntModelSpec getModelSpec(ModelMaker maker) {
        OntModelSpec spec = new OntModelSpec(OntModelSpec.OWL_MEM);
        spec.setImportModelMaker(maker);
        return spec;}
}
```

C:\Jena\Tutorial\expert\ontologyDB.java

Main.java



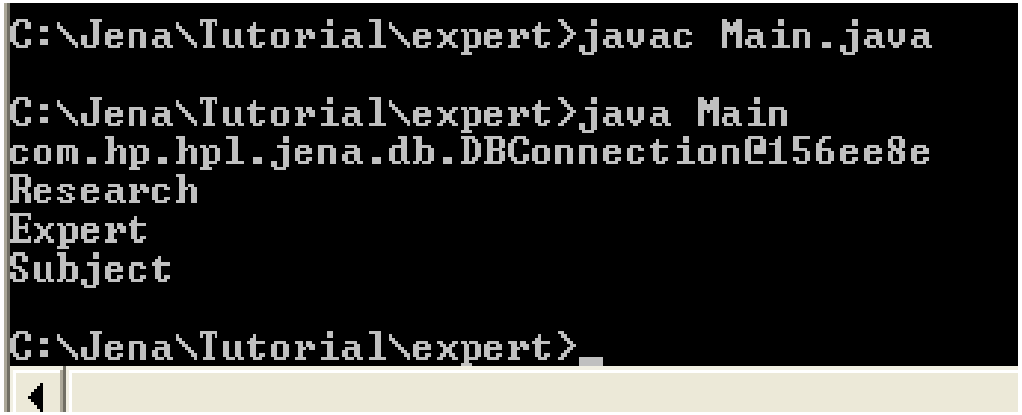
- Store expert.owl into MySQL database,
- Read expert.owl from MySQL database,
- List the classes of expert.owl

C:\Jena\Tutorial\expert\Main.java

```
import java.io.*;
import java.util.*;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.ModelMaker;
import com.hp.hpl.jena.util.FileManager;
import com.hp.hpl.jena.util.*;
import com.hp.hpl.jena.db.*;

public class Main  {
    public static final String DB_URL = "jdbc:mysql://localhost/expert";
    public static final String DB_USER = "root";
    public static final String DB_PASSWD = "111";
    public static final String DB = "MySQL";
    public static final String DB_DRIVER = "com.mysql.jdbc.Driver";

  public static void main (String args[]) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
          } catch (ClassNotFoundException e) {
           e.printStackTrace();     }

    String filePath = "file:///C:/Jena/Tutorial/expert/expert.owl";
    IDBConnection con = ontologyDB.connectDB(DB_URL,DB_USER, DB_PASSWD, DB);
    System.out.println(con);

    ontologyDB.createDBModelFromFile(con, "expert", filePath);
    OntModel model = ontologyDB.getModelFromDB(con, "expert");

    for (Iterator i = model.listClasses(); i.hasNext();) {
          OntClass c = (OntClass) i.next();
          System.out.println(c.getLocalName());   }
          }
}
```

Expert

- Open MySQL
  - Mysql>mysql –uroot –p111
- Create expert database in MySQL
  - Mysql>create database expert;
- Parse ontologyDB.java and Main.java
- Run Main.java

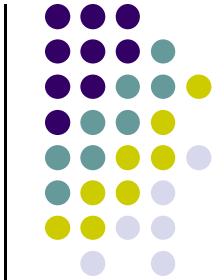


Notes:

Make sure all classes are owl classes because OWL_MEM:

```
<owl:Class rdf:ID="Research"/>
  <owl:Class rdf:ID="Expert"/>
  <owl:Class rdf:ID="Subject"/>
```

Expert - Class



- Main.java
  - Not using method to output class of the ontology
- Main1.java
  - Using defined SimpleReadOntology method to output class of the ontology

```
public static void SimpleReadOntology(OntModel model) {
    for (Iterator i = model.listClasses(); i.hasNext();) {
        OntClass c = (OntClass) i.next();
        System.out.println(c.getLocalName());
       }
}
```

Expert - individual

- MainIndividual.java
  - C:\Jena\Tutorial\expert\MainIndividual.java, defined prix is default namespace from expert.owl, using getInstance method.
  - But you have to compile ontologyDB.java, as all the Main* files are calling the methods defined in ontologyDB.class

```
public static void getInstance(OntModel model){
    String prix = "http://www.owl-ontologies.com/Expert.owl#";
    /*get Expert class from the onotlogy*/
    OntClass expert = model.getOntClass(prix + "Expert");
    //print out the name of the Expert class
    System.out.println(expert.getLocalName());

    //get instances of the Expert class
    Iterator it = expert.listInstances();
    //print out the instances of the Expert class
    while(it.hasNext()){
        Individual oi = (Individual)it.next();
        System.out.println(oi.getLocalName());
    }
}
```

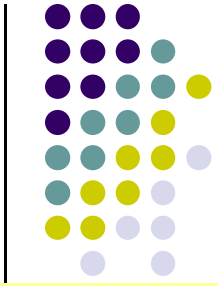
```
C:\Jena\Tutorial\expert>javac MainIndividual.java

C:\Jena\Tutorial\expert>java MainIndividual
com.hp.hpl.jena.db.DBConnection@156ee8e
Expert
John
Mary
Thomas

C:\Jena\Tutorial\expert>_
```

Expert - property

- MainProperty.java
  - C:\Jena\Tutorial\expert\MainProperty.java
  - Using getProperty method
  - Compile ontologyDB.java first

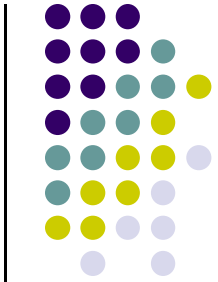
```
C:\Jena\Tutorial\expert>javac MainProperty.java

C:\Jena\Tutorial\expert>java MainProperty
com.hp.hpl.jena.db.DBConnection@e0e1c6
Expert
John
hasResearch
http://www.owl-ontologies.com/Expert.owl#SemanticWeb
http://www.owl-ontologies.com/Expert.owl#DigitalLibrary
Mary
hasResearch
http://www.owl-ontologies.com/Expert.owl#DataMining
http://www.owl-ontologies.com/Expert.owl#ECommerce
Thomas
hasResearch
http://www.owl-ontologies.com/Expert.owl#SemanticWeb
http://www.owl-ontologies.com/Expert.owl#DigitalLibrary
http://www.owl-ontologies.com/Expert.owl#DataMining

C:\Jena\Tutorial\expert>_
```

Expert – getProperty method



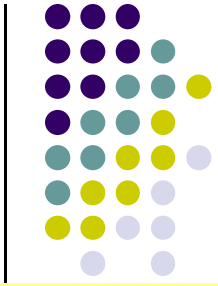
```
public static void getProperty(OntModel model) {
  String NS = "http://www.owl-ontologies.com/Expert.owl#";
  /* get the Expert class */
  OntClass expert = model.getOntClass(NS + "Expert");
  // print out the name of the Expert class
  System.out.println(expert.getLocalName());

  // get the instances of the Expert class
  Iterator it = expert.listInstances();
  // print out the instances of the Expert class
  while (it.hasNext()) {
   Individual oi = (Individual) it.next();
   System.out.println(oi.getLocalName());

   //get the properties of the instances of the Expert class
   for (Iterator ipp = expert.listDeclaredProperties(); ipp.hasNext();) {
        OntProperty p = (OntProperty) ipp.next();
       //print out property name and its values
       System.out.println( p.getLocalName());

        for (Iterator ivv = oi.listPropertyValues(p); ivv.hasNext();) {
          String valuename = ivv.next().toString();
          System.out.println(valuename); }        }
    }
}
```

Query Expert



- Using SPARQL and Jena Reasoner
- Add familiar_with property (domain: Expert, range: Subject) to expert.owl.
  - There is no instance for this property.
  - We will use reasoner to find its inferred instances
- Query: list the experts and their familiar_with subjects.
- Setting up rules for the reasoner:
  - ExpertA hasResearch ResearchB, ResearchB is associated with SubjectC → ExpertA is familiar_with SubjectC
- Using Sparql to query ExpertX (is familir_with) SubjectY
  - Select ?expert ?subject
  - Where { ?expert faimilar_with ?subject }

MainQuery.java



```
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.util.*;
import com.hp.hpl.jena.query.* ;
import com.hp.hpl.jena.sparql.*;
import com.hp.hpl.jena.reasoner.*;
import com.hp.hpl.jena.reasoner.rulesys.*;
import com.hp.hpl.jena.db.*;

public class MainQuery  {
    public static final String DB_URL = "jdbc:mysql://localhost/expert";
    public static final String DB_USER = "root";
    public static final String DB_PASSWD = "111";
    public static final String DB = "MySQL";
    public static final String DB_DRIVER = "com.mysql.jdbc.Driver";
  public static void main (String args[]) {
     try {
        Class.forName("com.mysql.jdbc.Driver");
          } catch (ClassNotFoundException e) {
          e.printStackTrace();     }
    String filePath = "file:///C:/Jena/Tutorial/expert/expert.owl";
    IDBConnection con = ontologyDB.connectDB(DB_URL,DB_USER, DB_PASSWD, DB);
    System.out.println(con);
    ontologyDB.createDBModelFromFile(con, "expert", filePath);
    OntModel model = ontologyDB.getModelFromDB(con, "expert");

    searchOnto(model);       }
```

MainQuery.java

```
public static void searchOnto(OntModel model){
     /*Setting up rules*/
     String rule = "[rule1:(?x http://www.owl-ontologies.com/Expert.owl#hasResearch ?y) " +
         "(?y http://www.owl-ontologies.com/Expert.owl#associate ?z) " +
         "->(?x http://www.owl-ontologies.com/Expert.owl#familiar_with ?z)]";

     /*query String*/
     String queryString = "PREFIX Expert:<http://www.owl-ontologies.com/Expert.owl#> " +
     "SELECT ?expert ?subject " +
     "WHERE {?expert Expert:familiar_with ?subject} ";

     /*set up reasoner*/
     Reasoner reasoner2 = new GenericRuleReasoner(Rule.parseRules(rule));

     InfModel inf = ModelFactory.createInfModel(reasoner2, model);
     Query query = QueryFactory.create(queryString);

     QueryExecution qe = QueryExecutionFactory.create(query, inf);
     ResultSet results = qe.execSelect();

     /*output result*/
     ResultSetFormatter.out(System.out, results, query);
     qe.close();    }
}
```

C:C:\Jena\Tutorial\expert\MainQuery.java

Query Expert

- MainQuery.java
  - C:\Jena\Tutorial\expert\MainQuery.java
  - Using searchOnto method
  - Compile ontologyDB.java first

```
C:\Jena\Tutorial\expert>javac MainQuery.java

C:\Jena\Tutorial\expert>java MainQuery
com.hp.hpl.jena.db.DBConnection@a59698
-----------------------------------------
| expert          | subject               |
=========================================
| Expert:Mary     | Expert:ComputerScience     |
| Expert:Thomas   | Expert:LibraryScience      |
| Expert:Mary     | Expert:Bussiness           |
| Expert:John     | Expert:InformationScience  |
| Expert:John     | Expert:ComputerScience     |
| Expert:Mary     | Expert:InformationScience  |
| Expert:John     | Expert:LibraryScience      |
| Expert:Thomas   | Expert:Bussiness           |
| Expert:Thomas   | Expert:ComputerScience     |
| Expert:Thomas   | Expert:InformationScience  |
-----------------------------------------

C:\Jena\Tutorial\expert>
```