SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

DEPARTMENTOFINFORMATIONTECHNOLOGY

COURSE CODE:DJ19ITL801	**DATE:** 03/04/2024
COURSE NAME: Semantic Web Technology Laboratory	**CLASS**: Final Year B.Tech IT I-1-2
NAME: Rishikesh Sharma	**SAP ID**: 60003200102

EXPERIMENTNO. 10

CO/LO: Apply Semantic web technologies in real world applications.

AIM/ OBJECTIVE: Ontology Programming with Jena API

THEORY: Ontology programming with the Jena API involves using Java to create, manipulate, and reason about ontologies represented in RDF, OWL, or other semantic web languages, providing a powerful tool for knowledge representation and management.

DESCRIPTION:

Install Jena in your computer and create programmatically an OWL ontology describing paintersandtheirpaintings.Theontologyshouldbeabletorepresentthefollowingstatements:“PainterX has painted the painting Y”, “Painter X was born in W”, and “Painting Y was painted in year Z”.CreateseveralindividualsforPainters’ontology.Forexample:PaulCezanne,born1839,Aix-en-Provence,France,painted“Lepaysan”and“LeVaseBleu”;LeonardodaVinci,born1452,Vinci,Florence, painted “Mona Lisa” and “The Last Supper”; Michelangelo Buonaroti, born 1475,Florence,painted“Sybille de Cummes”and“DelphesSylphide”.

PROCEDURE:

Installing Jena:

1. Go to the Apache Jena download page: https://jena.apache.org/download/
2. Download the latest version of Apache Jena, which is currently4.0.1.
3. Extract the downloaded zip file to a directory on your computer.
4. Set the JENA_HOMEenvironmentvariabletothedirectorywhereyouextractedthezipfile.
5. Add theJENA_HOME/bin directory to your PATH environment variable.

Creating an OWL ontology:

1. Openatexteditororanintegrateddevelopmentenvironment(IDE)thatsupportsJavaprogramming.
2. Create a new Java poject.
3. Add the Jena libraries to your project's class path.
4. Create a new Java class and name it "Painter Ontology".
5. Import the necessary Jena classes:

```
import
org.apache.jena.ontology.OntModel;import
org.apache.jena.rdf.model.ModelFactory;import
org.apache.jena.rdf.model.Resource;importorg.a
pache.jena.rdf.model.Property;
```

6. CreateanewOWL ontologymodel:

```
OntModelmodel=ModelFactory.createOntologyModel();
```

7. Createanamespacefor theontology

```
Stringns="http://example.com/painters#";
```

8. Definetheontologyclasses:

```
Resource painterClass = model.createClass(ns +
"Painter");ResourcepaintingClass=
```

9. Definethe ontology'sproperties:

```
PropertyhasPaintedProperty=model.createProperty(ns+"hasPainted");Pr
operty bornInProperty = model.createProperty(ns +
"bornIn");PropertypaintedInProperty=model.createProperty(ns+
```

10. Definethedomainsand rangesoftheproperties:

```
hasPaintedProperty.addDomain(painterClass);ha
sPaintedProperty.addRange(paintingClass);

bornInProperty.addDomain(painterClass);bornInProperty.addRange(XSDDateTime.XS
Ddate);

paintedInProperty.addDomain(paintingClass);paintedInProperty.addRange(XSDDateTi
me.XSDdate);
```

11. Createindividualsforthe paintersandtheirpaintings:

```
Resourcecezanne=model.createResource(ns+"Cezanne");Re
sourcedaVinci=model.createResource(ns +"DaVinci");
Resourcemichelangelo=model.createResource(ns+"Michelangelo");

Resource lePaysan = model.createResource(ns +
"LePaysan");ResourceleVaseBleu=model.createResource(ns+"LeV
aseBleu");ResourcemonaLisa =model.createResource(ns +
"MonaLisa");
Resource theLastSupper = model.createResource(ns +
"TheLastSupper");ResourcesybilleDeCummes=model.createResource(ns+"Sybil
```

12. Addpropertiesto theindividuals:

```
cezanne.addProperty(bornInProperty,model.createTypedLiteral("1839-01-
19",XSDDateTime.XSDdate));
cezanne.addProperty(hasPaintedProperty,
lePaysan);cezanne.addProperty(hasPaintedProperty,leVaseBleu);
```

```
daVinci.addProperty(bornInProperty,            model.createTypedLiteral("1452-04-
15",XSDDateTime.XSDdate));
daVinci.addProperty(hasPaintedProperty,
monaLisa);daVinci.addProperty(hasPaintedProperty,
theLastSupper);michelangelo.addProperty(bornInProperty,
                                               model.createTypedLiteral("1475-03-
06",XSDDateTime.XSDdate));
michelangelo.addProperty(hasPaintedProperty,
sybilleDeCummes);michelangelo.addProperty(hasPaintedProperty,
delphesSylphide);lePaysan.addProperty(paintedInProperty,
                                                    model.createTypedLiteral("1878",
XSDDateTime.XSDgYear));
leVaseBleu.addProperty(paintedInProperty,
                                                    model.createTypedLiteral("1893",
XSDDateTime.XSDgYear));
monaLisa.addProperty(paintedInProperty,     model.createTypedLiteral("1503-
1506",XSDDateTime.XSDgYear));
theLastSupper.addProperty(paintedInProperty, model.createTypedLiteral("1495-
1498",XSDDateTime.XSDgYear));
sybilleDeCummes.addProperty(paintedInProperty,
                                                    model.createTypedLiteral("1509",
XSDDateTime.XSDgYear));
delphesSylphide.addProperty(paintedInProperty,     model.createTypedLiteral("1501-
1504",XSDDateTime.XSDgYear));
```

13. Savethe ontology:

```
model.write(System.out,"RDF/XML-ABBREV");
```

14. Runtheprogram andcheck theoutput.

leDeCummes");

CODE:

```
import
org.apache.jena.ontology.OntModel;import
org.apache.jena.rdf.model.ModelFactory;import
org.apache.jena.rdf.model.Resource;importorg.a
pache.jena.rdf.model.Property;
importorg.apache.jena.datatypes.xsd.XSDDateTime;
publicclass PainterOntology {
    publicstatic void main(String[]args){
        Stringns="http://example.com/painters#";
        OntModel model =
        ModelFactory.createOntologyModel();Resource   painterClass
        = model.createClass(ns +
        "Painter");ResourcepaintingClass=model.createClass(ns+"Pai
        nting");
        Property hasPaintedProperty = model.createProperty(ns +
        "hasPainted");Property bornInProperty = model.createProperty(ns +
        "bornIn");Property paintedInProperty = model.createProperty(ns +
        "paintedIn");hasPaintedProperty.addDomain(painterClass);hasPaintedPr
        operty.addRange(paintingClass);bornInProperty.addDomain(painterClas
        s);bornInProperty.addRange(XSDDateTime.XSDdate);paintedInPropert
        y.addDomain(paintingClass);paintedInProperty.addRange(XSDDateTim
        e.XSDdate);
        Resourcecezanne=model.createResource(ns+"Cezanne");Re
        sourcedaVinci=model.createResource(ns+"DaVinci");
        Resourcemichelangelo=model.createResource(ns+"Michelangelo");Re
        source lePaysan = model.createResource(ns + "LePaysan");Resource
        leVaseBleu = model.createResource(ns +
        "LeVaseBleu");ResourcemonaLisa =model.createResource(ns+
        "MonaLisa");
        Resource theLastSupper = model.createResource(ns +
        "TheLastSupper");ResourcesybilleDeCummes=model.createResource(ns+"Sybil
```

leDeCummes");

```
        Resource delphesSylphide = model.createResource(ns +
        "DelphesSylphide");cezanne.addProperty(bornInProperty,
                                                        model.createTypedLiteral("1839-01-19",
XSDDateTime.XSDdate));cezanne.addProperty(hasPainted
        Property,
        lePaysan);cezanne.addProperty(hasPaintedProperty,le
        VaseBleu);
        daVinci.addProperty(bornInProperty,             model.createTypedLiteral("1452-04-
15",XSDDateTime.XSDdate));
        daVinci.addProperty(hasPaintedProperty,
        monaLisa);daVinci.addProperty(hasPaintedProperty,
        theLastSupper);michelangelo.addProperty(bornInProperty,
                                                        model.createTypedLiteral("1475-03-06",
XSDDateTime.XSDdate));michelangelo.addProperty(hasPaintedPropert
        y,sybilleDeCummes);michelangelo.addProperty(hasPaintedPropert
        y,delphesSylphide);
        lePaysan.addProperty(paintedInProperty,
                                                                model.createTypedLiteral("1878",
XSDDateTime.XSDgYear));
        leVaseBleu.addProperty(paintedInProperty,
                                                                model.createTypedLiteral("1893",
XSDDateTime.XSDgYear));
        monaLisa.addProperty(paintedInProperty,         model.createTypedLiteral("1503-
1506",XSDDateTime.XSDgYear));
        theLastSupper.addProperty(paintedInProperty,    model.createTypedLiteral("1495-
1498",XSDDateTime.XSDgYear));
        sybilleDeCummes.addProperty(paintedInProperty,
                                                                model.createTypedLiteral("1509",
XSDDateTime.XSDgYear));
        delphesSylphide.addProperty(paintedInProperty,model.createTypedLiteral("1501-
1504",XSDDateTime.XSDgYear));
    model.write(System.out,"RDF/XML-ABBREV");
}
```

```
}
```

OUTPUT:

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:painters="http://example.com/painters#">
<rdf:Descriptionrdf:about="http://example.com/painters#Cezanne">
<painters:bornIn rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1839-01-
19</painters:bornIn>
<painters:hasPaintedrdf:resource="http://example.com/painters#LeVaseBleu"/>
<painters:hasPaintedrdf:resource="http://example.com/painters#LePaysan"/>
<rdf:typerdf:resource="http://example.com/painters#Painter"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#DaVinci">
<painters:bornIn rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1452-04-
15</painters:bornIn>
<painters:hasPaintedrdf:resource="http://example.com/painters#TheLastSupper"/>
<painters:hasPaintedrdf:resource="http://example.com/painters#MonaLisa"/>
<rdf:typerdf:resource="http://example.com/painters#Painter"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#Michelangelo">
<painters:bornIn rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1475-03-
06</painters:bornIn>
<painters:hasPaintedrdf:resource="http://example.com/painters#SybilleDeCummes"/>
<painters:hasPaintedrdf:resource="http://example.com/painters#DelphesSylphide"/>
<rdf:typerdf:resource="http://example.com/painters#Painter"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#LeVaseBleu">
<painters:paintedIn
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1893</painters:paintedIn>
<rdf:typerdf:resource="http://example.com/painters#Painting"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#LePaysan">
<painters:paintedIn
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1878</painters:paintedIn>
<rdf:typerdf:resource="http://example.com/painters#Painting"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#SybilleDeCummes">
<painters:paintedIn
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1509</painters:paintedIn>
<rdf:typerdf:resource="http://example.com/painters#Painting"/>
</rdf:Description>
```

```
<rdf:Descriptionrdf:about="http://example.com/painters#DelphesSylphide">
<painters:paintedIn rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1501-1504</painters:paintedIn>
<rdf:typerdf:resource="http://example.com/painters#Painting"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#TheLastSupper">
<painters:paintedIn rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1495-1498</painters:paintedIn>
<rdf:typerdf:resource="http://example.com/painters#Painting"/>
</rdf:Description>
<rdf:Descriptionrdf:about="http://example.com/painters#MonaLisa">
<painters:paintedIn rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1503-1506</painters:paintedIn>
<rdf:typerdf:resource="http://example.com/painters#Painting"/>
</rdf:Description>
</rdf:RDF>
```

CONCLUSION:

This provides a simple example of how to create an OWL ontology programmatically using theJena API. With this knowledge, you can create more complex ontologies and extend them to suityourneeds.

REFERENCES:

[1] ApacheJenaDocumentation:https://jena.apache.org/documentation/

[2] OWLWebOntologyLanguageGuide:https://www.w3.org/TR/owl-guide/