*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

***«Московский государственный технический университет имени Н.Э. Баумана»   
(МГТУ им. Н.Э. Баумана)***

***Кафедра ИУ3, 3 курс, 6 семестр.***

**Отчёт**

**по лабораторной работе №3**

# “[Работа с плагинами и Apache Jena](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=54001702)”

**по курсу**

**“Разработка программного обеспечения”**

Выполнил:

Гольдштейн М.

Группа:

ИУ3-61

Проверил:

Иванов А.М.

**Москва, 2014**

# Цели лабораторной работы

* Углубление навыков работы с системой контроля версий
* Ознакомление на практике с основами графовых БД и технологий Semantic Web

# Задание

Создать тестовый Eclipse-проект, в котором происходит формирование информационной модели данных диаграммы и работа с данными в соответствии со своим вариантом задания и требованиями.

Добавить созданные проект в репозитарий системы контроля версий.

# План действий

1. Скачать Apache Jena

 2. Настроить Eclipse в соответствии с

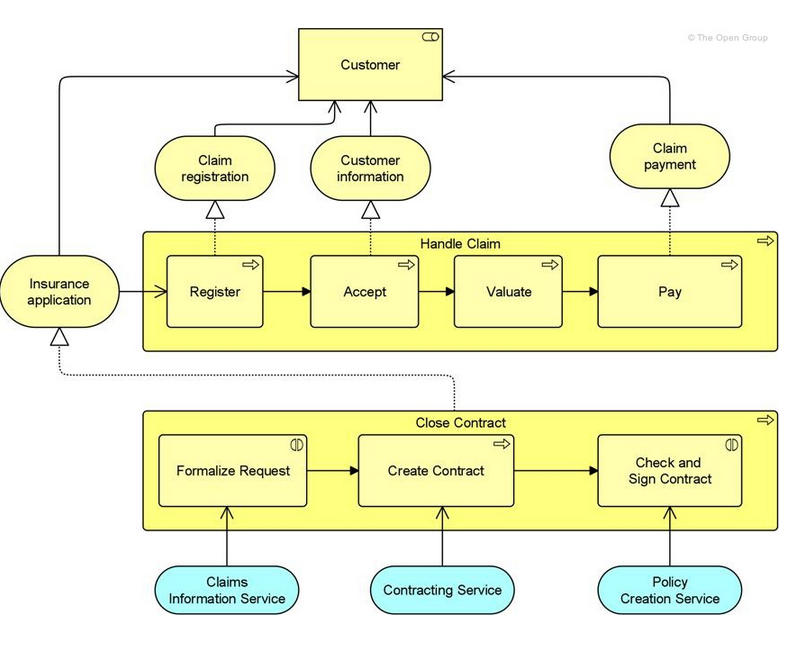
3. Сделать редакторе Protege модель классов и свойств заданной диаграммы в соответствии со своим вариантом

* По-сути создать "схему данных", состоящую из типов ее элементов и типов ее связей-стрелок
* [Про редактор Protege](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=56098839) ([**скачать нашу сборку**](ftp://maslect666:jtCWSxLb@agentlab.ru/distr/protege/Protege_5.0_beta-15_agentlab.zip))
* Сохранять файл в формате XML/RDF

4. Наполнить модель тестовыми данными

5. Создать 3 запроса к модели

Задание:



Листинг:

package JenaPruebas;

import java.io.FileWriter;

import java.io.IOException;

import com.hp.hpl.jena.ontology.Individual;

import com.hp.hpl.jena.ontology.ObjectProperty;

import com.hp.hpl.jena.ontology.OntClass;

import com.hp.hpl.jena.ontology.OntModel;

import com.hp.hpl.jena.rdf.model.ModelFactory;

import com.hp.hpl.jena.rdf.model.SimpleSelector;

import com.hp.hpl.jena.rdf.model.StmtIterator;

public class lab3M {

static final String SCHEMA = "http://www.agentlab.ru/jfxed/onto/strategicrationale";

static final String NS = SCHEMA + "#";

public static void main (String[] args) {

OntModel m = ModelFactory.createOntologyModel();

//create superclasses

OntClass concept = m.createClass(NS + "Concept");

OntClass relation = m.createClass(NS + "Relation");

//create objects

//OntClass bigYellowRectClass = m.createClass(NS + "bigYellowRect");

//bigYellowRectClass.addSuperClass(concept);

//OntClass bigYellowRectClass = m.createClass(NS + "bigYellowRect");

//bigYellowRectClass.addSuperClass(concept);

//OntClass yellowOvalClass = m.createClass(NS + "yellowOval");

//yellowOvalClass.addSuperClass(concept);

//OntClass yellowRectClass = m.createClass(NS + "yellowRect");

//yellowRectClass.addSuperClass(concept);

OntClass BusinessProcessClass = m.createClass(NS + "BusinessProcess");

BusinessProcessClass.addSuperClass(concept);

OntClass BusinessInteractionClass = m.createClass(NS + "BusinessInteraction");

BusinessInteractionClass.addSuperClass(concept);

OntClass ApplicationServiceClass = m.createClass(NS + "ApplicationService");

ApplicationServiceClass.addSuperClass(concept);

OntClass BusinessServiceClass = m.createClass(NS + "BusinessService");

BusinessServiceClass.addSuperClass(concept);

OntClass BusinessRoleClass = m.createClass(NS + "BusinessRole");

BusinessRoleClass.addSuperClass(concept);

//create arrows

OntClass UsedbyClass = m.createClass(NS + "Usedby");

UsedbyClass.addSuperClass(relation);

OntClass RealizationClass = m.createClass(NS + "Realization");

RealizationClass.addSuperClass(relation);

OntClass TriggeringClass = m.createClass(NS + "Triggering");

TriggeringClass.addSuperClass(relation);

OntClass CompositionClass = m.createClass(NS + "Composition");

CompositionClass.addSuperClass(relation);

//create properties

//for Usedby

ObjectProperty propUsedby = m.createObjectProperty(NS + "Usedby");

propUsedby.addDomain(BusinessServiceClass);

propUsedby.addRange(BusinessRoleClass);

ObjectProperty propUsedby1 = m.createObjectProperty(NS + "Usedby1");

propUsedby1.addDomain(ApplicationServiceClass);

propUsedby1.addRange(BusinessProcessClass);

ObjectProperty propUsedby2 = m.createObjectProperty(NS + "Usedby2");

propUsedby1.addDomain(ApplicationServiceClass);

propUsedby1.addRange(BusinessInteractionClass);

//for Realization

ObjectProperty propRealization = m.createObjectProperty(NS + "Realization");

propRealization.addDomain(BusinessProcessClass);

propRealization.addRange(BusinessServiceClass);

// ObjectProperty propRealization1 = m.createObjectProperty(NS + "Realization1");

// propRealization1.addDomain(bigYellowRectClass);

// propRealization1.addRange(yellowOvalClass);

//for Triggering

ObjectProperty propTriggering = m.createObjectProperty(NS + "Triggering");

propTriggering.addDomain(BusinessProcessClass);

propTriggering.addRange(BusinessProcessClass);

ObjectProperty propTriggering1 = m.createObjectProperty(NS + "Triggering");

propTriggering.addDomain(BusinessInteractionClass);

propTriggering.addRange(BusinessProcessClass);

ObjectProperty propTriggering2 = m.createObjectProperty(NS + "Triggering");

propTriggering.addDomain(BusinessProcessClass);

propTriggering.addRange(BusinessInteractionClass);

//for Composition

ObjectProperty propComposition = m.createObjectProperty(NS + "Composition");

propComposition.addDomain(BusinessProcessClass);

propComposition.addRange(BusinessProcessClass);

ObjectProperty propComposition1 = m.createObjectProperty(NS + "Composition");

propComposition.addDomain(BusinessInteractionClass);

propComposition.addRange(BusinessProcessClass);

//create individuals

Individual cust = m.createIndividual(NS + "Customer", BusinessRoleClass);

Individual claimReg = m.createIndividual(NS + "ClaimRegistration", BusinessServiceClass);

Individual custInf = m.createIndividual(NS + "customerInformation", BusinessServiceClass);

Individual claimPay = m.createIndividual(NS + "claimpayment", BusinessServiceClass);

Individual insuranceApl = m.createIndividual(NS + "insuranceApplication", BusinessServiceClass);

Individual reg = m.createIndividual(NS + "register", BusinessProcessClass);

Individual accept = m.createIndividual(NS + "accept", BusinessProcessClass);

Individual valuate = m.createIndividual(NS + "valuate", BusinessProcessClass);

Individual pay = m.createIndividual(NS + "pay", BusinessProcessClass);

Individual hanCl = m.createIndividual(NS + "handleClaim", BusinessProcessClass);

Individual closeCon = m.createIndividual(NS + "closeContract", BusinessProcessClass);

Individual formReq = m.createIndividual(NS + "FormalizeRequest", BusinessInteractionClass);

Individual createCon = m.createIndividual(NS + "createContract", BusinessProcessClass);

Individual checkSignCon = m.createIndividual(NS + "checkAndSignContract", BusinessInteractionClass);

Individual claimsInfServ = m.createIndividual(NS + "claimsInformationService", ApplicationServiceClass);

Individual contrServ = m.createIndividual(NS + "ContractingService", ApplicationServiceClass);

Individual policyCreServ = m.createIndividual(NS + "policyCreationService", ApplicationServiceClass);

//\*\*\*\*

//create ind links

//\*\*\*\*

//create all 1waydots links

//create 1 links

Individual link11 = m.createIndividual(NS + "insuranceApplication", UsedbyClass);

link11.addProperty(propUsedby, cust);

Individual link12 = m.createIndividual(NS + "insuranceApplication", UsedbyClass);

link12.addProperty(propUsedby, reg);

Individual link13 = m.createIndividual(NS + "ClaimRegistration", UsedbyClass);

link13.addProperty(propUsedby, cust);

Individual link14 = m.createIndividual(NS + "customerInformation", UsedbyClass);

link14.addProperty(propUsedby, cust);

Individual link15 = m.createIndividual(NS + "cliampayment", UsedbyClass);

link15.addProperty(propUsedby, cust);

Individual link16 = m.createIndividual(NS + "claimsInformationService", UsedbyClass);

link16.addProperty(propUsedby1, formReq);

Individual link17 = m.createIndividual(NS + "ContractingService", UsedbyClass);

link17.addProperty(propUsedby1, createCon);

Individual link18 = m.createIndividual(NS + "policyCreationService", UsedbyClass);

link18.addProperty(propUsedby1, checkSignCon);

//create all 1wayblackarrow links

//create 2 links

Individual link21 = m.createIndividual(NS + "register", TriggeringClass);

link21.addProperty(propTriggering, accept);

Individual link22 = m.createIndividual(NS + "accept", TriggeringClass);

link22.addProperty(propTriggering, valuate);

Individual link23 = m.createIndividual(NS + "valuate", TriggeringClass);

link23.addProperty(propTriggering, pay);

Individual link24 = m.createIndividual(NS + "FormalizeRequest", TriggeringClass);

link24.addProperty(propTriggering, createCon);

Individual link25 = m.createIndividual(NS + "createContract", TriggeringClass);

link25.addProperty(propTriggering, checkSignCon);

//create all background links

//create 3 links

Individual link31 = m.createIndividual(NS + "register", CompositionClass);

link31.addProperty(propComposition, hanCl);

Individual link32 = m.createIndividual(NS + "accept", CompositionClass);

link32.addProperty(propComposition, hanCl);

Individual link33 = m.createIndividual(NS + "valuate", CompositionClass);

link33.addProperty(propComposition, hanCl);

Individual link34 = m.createIndividual(NS + "pay", CompositionClass);

link34.addProperty(propComposition, hanCl);

Individual link35 = m.createIndividual(NS + "FormalizeRequest", CompositionClass);

link35.addProperty(propComposition, closeCon);

Individual link36 = m.createIndividual(NS + "createContract", CompositionClass);

link36.addProperty(propComposition, closeCon);

Individual link37 = m.createIndividual(NS + "checkAndSignContract", CompositionClass);

link37.addProperty(propComposition, closeCon);

//create all background links

//create 4 links

Individual link41 = m.createIndividual(NS + "register", RealizationClass);

link41.addProperty(propRealization, claimReg);

Individual link42 = m.createIndividual(NS + "accept", RealizationClass);

link42.addProperty(propRealization, custInf);

Individual link43 = m.createIndividual(NS + "pay", RealizationClass);

link43.addProperty(propRealization, claimPay);

Individual link44 = m.createIndividual(NS + "closeContract", RealizationClass);

link44.addProperty(propRealization, insuranceApl);

System.out.println("Request1\n");

for(StmtIterator it = link11.listProperties(); it.hasNext();)

{

System.out.println(it.nextStatement());

}

System.out.println("Request2\n");

for(StmtIterator it = reg.listProperties(); it.hasNext();)

{

System.out.println(it.nextStatement());

}

System.out.println("Request3\n");

//вывод всего, что связано с contrServ для всей модели

OntClass c = contrServ.getOntClass();

SimpleSelector s = new SimpleSelector(null, null, c);

for(StmtIterator it = m.listStatements(s); it.hasNext();)

{

System.out.println(it.nextStatement());

}

try {

m.write(new FileWriter("Mike.owl"), "RDF/XML");

} catch (IOException e) {

e.printStackTrace();

}

}

}

Результаты запросов:

Request1

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby, http://www.agentlab.ru/jfxed/onto/strategicrationale#register]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby, http://www.agentlab.ru/jfxed/onto/strategicrationale#Customer]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#BusinessService]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Relation]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/01/rdf-schema#Resource]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Concept]

Request2

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.agentlab.ru/jfxed/onto/strategicrationale#Realization, http://www.agentlab.ru/jfxed/onto/strategicrationale#ClaimRegistration]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Realization]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.agentlab.ru/jfxed/onto/strategicrationale#Composition, http://www.agentlab.ru/jfxed/onto/strategicrationale#handleClaim]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Composition]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.agentlab.ru/jfxed/onto/strategicrationale#Triggering, http://www.agentlab.ru/jfxed/onto/strategicrationale#accept]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Triggering]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#BusinessProcess]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#BusinessRole]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#BusinessInteraction]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Relation]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/01/rdf-schema#Resource]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#register, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Concept]

Request3

[http://www.agentlab.ru/jfxed/onto/strategicrationale#policyCreationService, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#ContractingService, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#claimsInformationService, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#cliampayment, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#customerInformation, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#ClaimRegistration, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#insuranceApplication, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

[http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby, http://www.w3.org/2000/01/rdf-schema#subClassOf, http://www.agentlab.ru/jfxed/onto/strategicrationale#Usedby]

Результаты моделирования в Protégé:

