*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”**

Выполнила: Емельянова А.А.

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

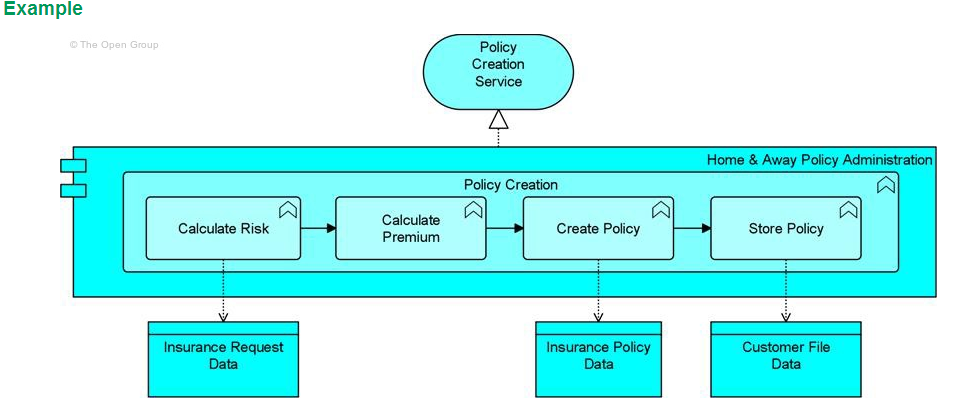
3. Сделать редакторе Protege модель классов и свойств заданной диаграммы в соответствии со своим вариантом

* 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 JenaProebas;**

**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 lab3N {**

**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 applicationserviceClass = m.createClass(NS + "applicationserviceClass");**

**applicationserviceClass.addSuperClass(concept);**

**OntClass applicationfunctionClass = m.createClass(NS + "applicationfunctionClass");**

**applicationfunctionClass.addSuperClass(concept);**

**//OntClass applicationfunctionClass = m.createClass(NS + "applicationfunctionClass");**

**//applicationfunctionClass.addSuperClass(concept);**

**OntClass applicationcomponentClass = m.createClass(NS + "applicationcomponentClass");**

**applicationcomponentClass.addSuperClass(concept);**

**OntClass dataobjectClass = m.createClass(NS + "dataobjectClass");**

**dataobjectClass.addSuperClass(concept);**

**//create arrows**

**OntClass accessClass = m.createClass(NS + "access");**

**accessClass.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 onewaywhitearrow**

**ObjectProperty proprealizationealization = m.createObjectProperty(NS + "realizationealization");**

**proprealizationealization.addDomain(applicationcomponentClass);**

**proprealizationealization.addRange(applicationserviceClass);**

**//!for onewayblackarrow**

**ObjectProperty proptriggering = m.createObjectProperty(NS + "triggering");**

**proptriggering.addDomain(applicationfunctionClass);**

**proptriggering.addRange(applicationfunctionClass);**

**//!for onewaydotsarrow**

**ObjectProperty propaccess = m.createObjectProperty(NS + "access");**

**propaccess.addDomain(applicationfunctionClass);**

**propaccess.addRange(dataobjectClass);**

**//!for backgroundConnect**

**ObjectProperty propcomposition = m.createObjectProperty(NS + "composition");**

**propcomposition.addDomain(applicationfunctionClass);**

**propcomposition.addRange(applicationfunctionClass);**

**//ObjectProperty propcomposition = m.createObjectProperty(NS + "composition");**

**//propcomposition.addDomain(applicationfunctionClass);**

**// propcomposition.addRange(applicationcomponentClass);**

**//create individuals**

**Individual polCreServ = m.createIndividual(NS + "PolicyCreationService", applicationserviceClass);**

**Individual homeAPolAdmin = m.createIndividual(NS + "HomeAndAwayPolicyAdministration", applicationcomponentClass);**

**Individual policCreat = m.createIndividual(NS + "PolicyCreation",applicationfunctionClass);**

**Individual calcRisk = m.createIndividual(NS + "CalculateRisk", applicationfunctionClass);**

**Individual calcPrem = m.createIndividual(NS + "CalculatePremium", applicationfunctionClass);**

**Individual creatPol = m.createIndividual(NS + "CreatePolicy", applicationfunctionClass);**

**Individual storePol= m.createIndividual(NS + "StorePolicy", applicationfunctionClass);**

**Individual insurReqD = m.createIndividual(NS + "InsuranceRequestData",dataobjectClass);**

**Individual insPolData = m.createIndividual(NS + "InsurancePolicyData", dataobjectClass);**

**Individual custFileData = m.createIndividual(NS + "CustomerFileData",dataobjectClass);**

**//\*\*\*\***

**//create ind links**

**//\*\*\*\***

**//create all onewaywhitearrow links**

**//create 1 links**

**Individual link11 = m.createIndividual(NS + "HomeAndAwayPolicyAdministration", realizationClass);**

**link11.addProperty(proprealizationealization, polCreServ);**

**//create all backgroundconnection links**

**//create 2 links**

**Individual link21 = m.createIndividual(NS + "PolicyCreation", compositionClass);**

**link21.addProperty(propcomposition, homeAPolAdmin);**

**Individual link22 = m.createIndividual(NS + "CalculateRisk", compositionClass);**

**link22.addProperty(propcomposition, policCreat);**

**Individual link23 = m.createIndividual(NS + "CalculatePremium", compositionClass);**

**link23.addProperty(propcomposition, policCreat);**

**Individual link24 = m.createIndividual(NS + "CreatePolicy", compositionClass);**

**link24.addProperty(propcomposition, policCreat);**

**Individual link25 = m.createIndividual(NS + "StorePolicy", compositionClass);**

**link25.addProperty(propcomposition, policCreat);**

**//create all 1wayblackarrow links**

**//create 3 links**

**Individual link31 = m.createIndividual(NS + "CalculateRisk", triggeringClass);**

**link31.addProperty(proptriggering, calcPrem);**

**Individual link32 = m.createIndividual(NS + "CalculatePremium", triggeringClass);**

**link32.addProperty(proptriggering, creatPol);**

**Individual link33 = m.createIndividual(NS + "CreatePolicy", triggeringClass);**

**link33.addProperty(proptriggering, storePol);**

**//create all 1waywdotsarrow links**

**Individual link41 = m.createIndividual(NS + "CalculateRisk", accessClass);**

**link41.addProperty(propaccess, insurReqD);**

**Individual link42 = m.createIndividual(NS + "CreatePolicy", accessClass);**

**link42.addProperty(propaccess, insPolData);**

**Individual link43 = m.createIndividual(NS + "StorePolicy", accessClass);**

**link43.addProperty(propaccess, custFileData);**

**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 = polCreServ.listProperties(); it.hasNext();)**

**{**

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

**}**

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

**//вывод всего, что связано с Inturance Policy Data для всей модели**

**OntClass c = insPolData.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("Nastya.owl"), "RDF/XML");**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**}**

**}**Результаты запросов:

log4j:WARN No appenders could be found for logger (org.apache.jena.riot.stream.JenaIOEnvironment).

log4j:WARN Please initialize the log4j system properly.

log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

Request1

[http://www.agentlab.ru/jfxed/onto/strategicrationale#HomeAndAwayPolicyAdministration, http://www.agentlab.ru/jfxed/onto/strategicrationale#realizationealization, http://www.agentlab.ru/jfxed/onto/strategicrationale#PolicyCreationService]

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

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

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

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

Request2

[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#applicationserviceClass]

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

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

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

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

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

Результаты моделирования в Protégé:

