*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

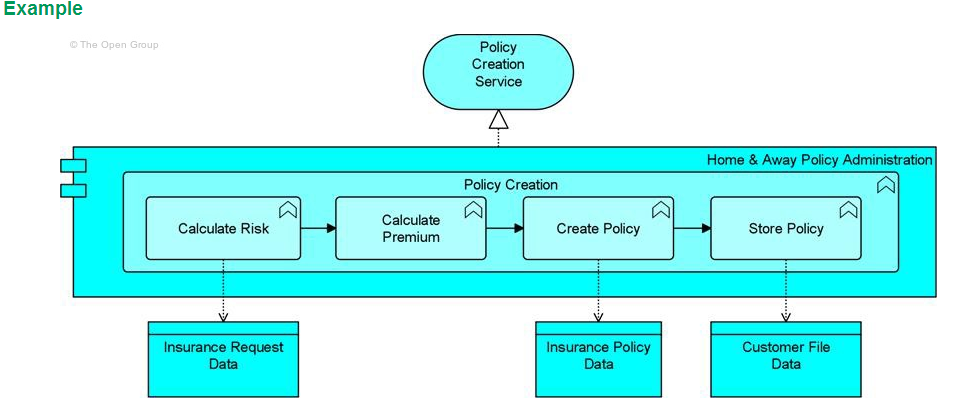
3. Сделать редакторе Protege модель классов и свойств заданной диаграммы в соответствии со своим вариантом

* 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 blueOvalClass = m.createClass(***NS*** + "blueOval");

blueOvalClass.addSuperClass(concept);

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

lBlueSmallRectClass.addSuperClass(concept);

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

lBlueBigRectClass.addSuperClass(concept);

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

blueBigRectClass.addSuperClass(concept);

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

blueSmallRectClass.addSuperClass(concept);

//create arrows

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

oneWayDotsArrowClass.addSuperClass(relation);

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

oneWayWhiteArrowClass.addSuperClass(relation);

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

oneWayBlackArrowClass.addSuperClass(relation);

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

backgroundConnectClass.addSuperClass(relation);

//create properties

//for onewaywhitearrow

ObjectProperty propOneWayWhiteArrow = m.createObjectProperty(***NS*** + "OneWayWhiteArrow");

propOneWayWhiteArrow.addDomain(blueBigRectClass);

propOneWayWhiteArrow.addRange(blueOvalClass);

//for onewayblackarrow

ObjectProperty propOneWayBlackArrow = m.createObjectProperty(***NS*** + "OneWayBlackArrow");

propOneWayBlackArrow.addDomain(lBlueSmallRectClass);

propOneWayBlackArrow.addRange(lBlueSmallRectClass);

//for onewaydotsarrow

ObjectProperty propOneWayDotsArrow = m.createObjectProperty(***NS*** + "OneWayDotsArrow");

propOneWayDotsArrow.addDomain(lBlueSmallRectClass);

propOneWayDotsArrow.addRange(blueSmallRectClass);

//for backgroundConnect

ObjectProperty propBackgroundConnect = m.createObjectProperty(***NS*** + "backgroundConnect");

propBackgroundConnect.addDomain(lBlueSmallRectClass);

propBackgroundConnect.addRange(lBlueBigRectClass);

ObjectProperty propBackgroundConnect1 = m.createObjectProperty(***NS*** + "backgroundConnect1");

propBackgroundConnect1.addDomain(lBlueBigRectClass);

propBackgroundConnect1.addRange(blueBigRectClass);

//create individuals

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

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

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

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

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

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

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

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

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

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

//\*\*\*\*

//create ind links

//\*\*\*\*

//create all onewaywhitearrow links

//create 1 links

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

link11.addProperty(propOneWayWhiteArrow, polCreServ);

//create all backgroundconnection links

//create 2 links

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

link21.addProperty(propBackgroundConnect1, homeAPolAdmin);

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

link22.addProperty(propBackgroundConnect, policCreat);

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

link23.addProperty(propBackgroundConnect, policCreat);

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

link24.addProperty(propBackgroundConnect, policCreat);

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

link25.addProperty(propBackgroundConnect, policCreat);

//create all 1wayblackarrow links

//create 3 links

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

link31.addProperty(propOneWayBlackArrow, calcPrem);

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

link32.addProperty(propOneWayBlackArrow, creatPol);

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

link33.addProperty(propOneWayBlackArrow, storePol);

//create all 1waywdotsarrow links

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

link41.addProperty(propOneWayDotsArrow, insurReqD);

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

link42.addProperty(propOneWayDotsArrow, insPolData);

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

link43.addProperty(propOneWayDotsArrow, 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#OneWayWhiteArrow, 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#oneWayWhiteArrow]

[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#blueBigRectClass]

[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#blueOval]

[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]

[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]

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#blueSmallRectClass]

[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#blueSmallRectClass]

[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#blueSmallRectClass]

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

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

Результаты моделирования в Protégé:

