*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:

Шорох Д.К.

Группа:

ИУ3-62

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

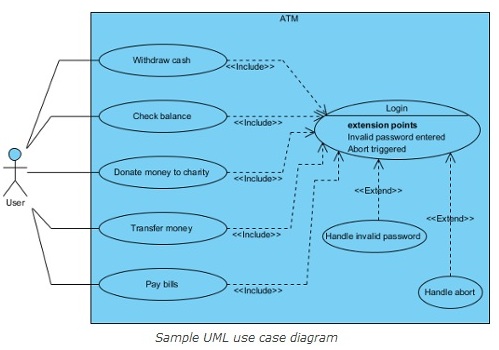
3. Сделать редакторе Protege модель классов и свойств заданной диаграммы в соответствии со своим вариантом

* 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** ru.agentlab.jfxed.diagramms.UMLUseCase;

**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.query.Query;

**import** com.hp.hpl.jena.query.QueryExecution;

**import** com.hp.hpl.jena.query.QueryExecutionFactory;

**import** com.hp.hpl.jena.query.QueryFactory;

**import** com.hp.hpl.jena.query.ResultSetFormatter;

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

**import** com.hp.hpl.jena.rdf.model.Resource;

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

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

**public** **class** UMLUseCase {

**static** String *SOURCE* = "http://www.eswc2006.org/technologies/ontology";

**static** String *NS* = *SOURCE* + "#";

// test from sourcetree

**public** **static** **void** main(String[] args) {

OntModel m = ModelFactory.*createOntologyModel*();// пустая база знаний

//для Protege

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

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

OntClass actorClass = m.createClass(*NS* + "Actor");// actor

actorClass.addSuperClass(concept);

OntClass usecaseClass = m.createClass(*NS* + "UseCase");// блок Use Case

usecaseClass.addSuperClass(concept);

OntClass associationClass = m.createClass(*NS* + "Association");// связь Association

associationClass.addSuperClass(relation);

OntClass inclusionClass = m.createClass(*NS* + "Include");// связь Include

inclusionClass.addSuperClass(relation);

OntClass extensionClass = m.createClass(*NS* + "Extend");// связь Extend

extensionClass.addSuperClass(relation);

OntClass systemClass = m.createClass(*NS* + "ATM");// блок System

systemClass.addSuperClass(concept);

// связь actor - use case

ObjectProperty assocPropTo = m.createObjectProperty(*NS* + "AssociationTo");

assocPropTo.addRange(usecaseClass);

assocPropTo.addDomain(associationClass);

ObjectProperty assocPropFrom = m.createObjectProperty(*NS* + "AssociationFrom");

assocPropFrom.addDomain(associationClass);

assocPropFrom.addRange(actorClass);

// связи include

ObjectProperty incPropTo = m.createObjectProperty(*NS* + "InclusionTo");

incPropTo.addDomain(inclusionClass);

incPropTo.addRange(usecaseClass);

ObjectProperty incPropFrom = m.createObjectProperty(*NS* + "InclusionFrom");

incPropFrom.addRange(usecaseClass);

incPropFrom.addDomain(inclusionClass);

// связи extend

ObjectProperty extPropTo = m.createObjectProperty(*NS* + "ExtensionTo");

extPropTo.addDomain(extensionClass);

extPropTo.addRange(usecaseClass);

ObjectProperty extPropFrom = m.createObjectProperty(*NS* + "ExtensionFrom");

extPropFrom.addRange(usecaseClass);

extPropFrom.addDomain(extensionClass);

// связь use case - system

ObjectProperty systemPropTo = m.createObjectProperty(*NS* + "SystemTo");

assocPropTo.addRange(usecaseClass);

assocPropTo.addDomain(associationClass);

ObjectProperty systemPropFrom = m.createObjectProperty(*NS* + "SystemFrom");

assocPropFrom.addDomain(associationClass);

assocPropFrom.addRange(usecaseClass);

// наполнение базы

Individual user = m.createIndividual(*NS* + "User", actorClass);

Individual withdrawCash = m.createIndividual(*NS* + "WithdrawCash", usecaseClass);

Individual checkBalance = m.createIndividual(*NS* + "Checkbalance", usecaseClass);

Individual donate = m.createIndividual(*NS* + "DonateMoneyToCharity", usecaseClass);

Individual transferMoney = m.createIndividual(*NS* + "TransferMoney", usecaseClass);

Individual payBills = m.createIndividual(*NS* + "PayBills", usecaseClass);

Individual login = m.createIndividual(*NS* + "ExtensionPoits.InvalidPasswordEntered.AbortTriggered", usecaseClass);

Individual handleInvalidPassword = m.createIndividual(*NS* + "HandleInvalidPassword", usecaseClass);

Individual handleAbort = m.createIndividual(*NS* + "HandleAbort", usecaseClass);

Individual atm = m.createIndividual(*NS* + "ATM", systemClass);

Individual systemIndividual = m.createIndividual(*NS* + "agg1", associationClass);

systemIndividual.addProperty(systemPropFrom, withdrawCash);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, checkBalance);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, donate);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, transferMoney);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, payBills);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, login);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, handleInvalidPassword);

systemIndividual.addProperty(systemPropTo, atm);

systemIndividual.addProperty(systemPropFrom, handleAbort);

systemIndividual.addProperty(systemPropTo, atm);

Individual associationIndividual = m.createIndividual(*NS* + "agg2", associationClass);

associationIndividual.addProperty(assocPropFrom, user);

associationIndividual.addProperty(assocPropTo, withdrawCash);

associationIndividual.addProperty(assocPropFrom, user);

associationIndividual.addProperty(assocPropTo, checkBalance);

associationIndividual.addProperty(assocPropFrom, user);

associationIndividual.addProperty(assocPropTo, donate);

associationIndividual.addProperty(assocPropFrom, user);

associationIndividual.addProperty(assocPropTo, transferMoney);

associationIndividual.addProperty(assocPropFrom, user);

associationIndividual.addProperty(assocPropTo, payBills);

Individual inclusionIndividual = m.createIndividual(*NS* + "agg3", inclusionClass);

inclusionIndividual.addProperty(incPropFrom, withdrawCash);

inclusionIndividual.addProperty(incPropTo, login);

inclusionIndividual.addProperty(incPropFrom, checkBalance);

inclusionIndividual.addProperty(incPropTo, login);

inclusionIndividual.addProperty(incPropFrom, donate);

inclusionIndividual.addProperty(incPropTo, login);

inclusionIndividual.addProperty(incPropFrom, transferMoney);

inclusionIndividual.addProperty(incPropTo, login);

inclusionIndividual.addProperty(incPropFrom, payBills);

inclusionIndividual.addProperty(incPropTo, login);

Individual extensionIndividual = m.createIndividual(*NS* + "agg4", extensionClass);

extensionIndividual.addProperty(extPropFrom, handleInvalidPassword);

extensionIndividual.addProperty(extPropTo, login);

extensionIndividual.addProperty(extPropFrom, handleAbort);

extensionIndividual.addProperty(extPropTo, login);

m.write(System.out);// в консоль

System.out.println("Привет");

//Запросы

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

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

{

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

}

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

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

{

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

}

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

//вывод всего и всех где есть user

OntClass c = user.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("ex1.owl"));//xml

} **catch** (IOException e){

e.printStackTrace();

}

}

}

Результаты запросов:

[http://www.eswc2006.org/technologies/ontology#Include, http://www.w3.org/2000/01/rdf-schema#subClassOf, http://www.w3.org/2000/01/rdf-schema#Resource]

[http://www.eswc2006.org/technologies/ontology#Include, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/01/rdf-schema#Class]

[http://www.eswc2006.org/technologies/ontology#Include, http://www.w3.org/2000/01/rdf-schema#subClassOf, http://www.eswc2006.org/technologies/ontology#relation]

[http://www.eswc2006.org/technologies/ontology#Include, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2002/07/owl#Class]

[http://www.eswc2006.org/technologies/ontology#Include, http://www.w3.org/2000/01/rdf-schema#subClassOf, http://www.eswc2006.org/technologies/ontology#Include]

[http://www.eswc2006.org/technologies/ontology#Include, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/01/rdf-schema#Resource]

[http://www.eswc2006.org/technologies/ontology#Checkbalance, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.eswc2006.org/technologies/ontology#UseCase]

[http://www.eswc2006.org/technologies/ontology#Checkbalance, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.eswc2006.org/technologies/ontology#concept]

[http://www.eswc2006.org/technologies/ontology#Checkbalance, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/01/rdf-schema#Resource]

[http://www.eswc2006.org/technologies/ontology#User, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.eswc2006.org/technologies/ontology#Actor]

[http://www.eswc2006.org/technologies/ontology#AssociationFrom, http://www.w3.org/2000/01/rdf-schema#range, http://www.eswc2006.org/technologies/ontology#Actor]

[http://www.eswc2006.org/technologies/ontology#Actor, http://www.w3.org/2000/01/rdf-schema#subClassOf, <http://www.eswc2006.org/technologies/ontology#Actor>]

Результаты моделирования в Protégé:

