Факультет ИУ "Информатика и системы управления"

Кафедра ИУ-3 "Информационные системы и телекоммуникации"

Отчет к лабораторной работе №5

по курсу "**Разработка программного обеспечения**"

направления 2304000062

# " [Расширение Eclipse своими плагинами](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=55771143)"

Продолжительность 4 часа.

Выполнил:

студент группы ИУ3-62

Даварашвили И.Б.

Проверил:

Иванов А.М.

# Цели лабораторной работы

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

# Задание

1. Чтобы познакомиться с технологией выполните указания обучающего материала

* [Extending the Eclipse IDE - Plug-in development](http://www.vogella.com/tutorials/EclipsePlugIn/article.html)

2. Откройте Eclipse и обновите локальную копию репозитария

* В перспективе Git, надо сделать Pull

3. Переключитесь на стандартную целевую платформу Eclipse

* Настройки Eclipse > Plug-in Development > Target Platform, установить **Running Platform**

4. Импортируйте в Workspace проекты:

* **ru.agentlab.jfxed**
* **ru.agentlab.jfxed.figures**
* **ru.agentlab.jfxed.figures.class**

4. Запустите редактор

* Конфигурация запуска **Eclipse Jfxed**

Исходный код программы:

**package** ru.agentlab.jfxed.diagramms.InfDia

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

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

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

**import** com.hp.hpl.jena.query.QuerySolution

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

**import** de.fxdiagram.core.XDiagram

**import** ru.agentlab.jfxed.IDiagram

**import** ru.agentlab.jfxed.figures.ass.Assessment

**public** **class** ClazzDiagram **implements** IDiagram {

**static** String *SOURCE* = "http://www.agentlab.ru/jfxed/onto/influencediagram"

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

**override** createJfx(OntModel jenaModel, XDiagram jfxDiagram) {

**val** queryString ='''

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?uri

where {

?uri rdf:type <«*SOURCE*»#generalVariable>

}

'''

**val** query = QueryFactory.*create*(queryString)

// Execute the query and obtain results

**val** qe = QueryExecutionFactory.*create*(query, jenaModel)

**val** results = qe.execSelect()

**for** ( ; results.hasNext() ; )

{

**val** QuerySolution soln = results.nextSolution()

**val** Resource x = soln.getResource("uri") // Get a result variable by name.

**val** target = **new** Assessment() => [

layoutX = 280

layoutY = 280

name = x.localName

]

jfxDiagram => [

nodes += target

]

}

qe.close()

}

}

package ru.agentlab.jfxed.diagramms.InfDia;

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.QuerySolution;

import com.hp.hpl.jena.query.ResultSet;

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

import de.fxdiagram.core.XDiagram;

import de.fxdiagram.core.XNode;

import javafx.collections.ObservableList;

import org.eclipse.xtend2.lib.StringConcatenation;

import org.eclipse.xtext.xbase.lib.ObjectExtensions;

import org.eclipse.xtext.xbase.lib.Procedures.Procedure1;

import ru.agentlab.jfxed.IDiagram;

import ru.agentlab.jfxed.figures.ass.Assessment;

@SuppressWarnings("all")

public class ClazzDiagram implements IDiagram {

private static String SOURCE = "http://www.agentlab.ru/jfxed/onto/influencediagram";

private static String NS = (ClazzDiagram.SOURCE + "#");

public void createJfx(final OntModel jenaModel, final XDiagram jfxDiagram) {

StringConcatenation \_builder = new StringConcatenation();

\_builder.append("PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ");

\_builder.newLine();

\_builder.append("select ?uri ");

\_builder.newLine();

\_builder.append("where { ");

\_builder.newLine();

\_builder.append("\t");

\_builder.append("?uri rdf:type <");

\_builder.append(ClazzDiagram.SOURCE, "\t");

\_builder.append("#generalVariable> ");

\_builder.newLineIfNotEmpty();

\_builder.append("} ");

\_builder.newLine();

final String queryString = \_builder.toString();

final Query query = QueryFactory.create(queryString);

final QueryExecution qe = QueryExecutionFactory.create(query, jenaModel);

final ResultSet results = qe.execSelect();

for (; results.hasNext();) {

{

final QuerySolution soln = results.nextSolution();

final Resource x = soln.getResource("uri");

Assessment \_assessment = new Assessment();

final Procedure1<Assessment> \_function = new Procedure1<Assessment>() {

public void apply(final Assessment it) {

it.setLayoutX(280);

it.setLayoutY(280);

String \_localName = x.getLocalName();

it.setName(\_localName);

}

};

final Assessment target = ObjectExtensions.<Assessment>operator\_doubleArrow(\_assessment, \_function);

final Procedure1<XDiagram> \_function\_1 = new Procedure1<XDiagram>() {

public void apply(final XDiagram it) {

ObservableList<XNode> \_nodes = it.getNodes();

\_nodes.add(target);

}

};

ObjectExtensions.<XDiagram>operator\_doubleArrow(jfxDiagram, \_function\_1);

}

}

qe.close();

}

}

