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 ИУ3-61

Учанев И.Л.

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.figures.device

import de.fxdiagram.core.XNode

import javafx.scene.Group

import javafx.scene.layout.GridPane

import javafx.scene.layout.HBox

import javafx.scene.paint.Color

import javafx.scene.shape.Polygon

import javafx.scene.shape.Polyline

import javafx.scene.text.Text

import ru.agentlab.jfxed.IFigure

class Coub extends XNode implements IFigure {

var GridPane root

val nameTextBox = new Text

var String name

new(String name) {

super(name)

this.name = name

}

new (){

super("Class");

name = "Class";

root = new GridPane

node = root

var pligon = new Polygon()

pligon.getPoints().addAll(#[

10.0, 0.0,

105.0, 00.0,

105.0, 50.0,

95.0, 60.0,

0.0, 60.0,

0.0, 10.0

])

pligon.setFill(Color.CYAN)

pligon.setStroke(Color.BLACK)

pligon.setStrokeWidth(0.8)

var line = new Polyline(#[

0.0, 10.0,

95.0, 10.0,

105.0, 0.0,

95.0, 10.0,

95.0, 60.0

])

var line1 = new Polyline(#[

80.0, 24.0,

84.0, 20.0,

87.0, 20.0,

91.0, 24.0,

80.0, 24.0

])

var line2 = new Polyline(#[

80.0, 18.0,

80.0, 16.0,

81,15,

90,15,

91.0, 16.0,

91.0, 18.0,

90,19,

81,19,

80.0, 18.0

])

var group = new Group(pligon ,line,line1, line2)

val s = Coub.getResource("ClassFigure.css").toExternalForm()

root.stylesheets += s

root.children += group

root.children += new HBox => [

styleClass += "ClassFigure1"

root.children += nameTextBox => [

text = name

styleClass += "ClassFigure-Text"

]

]

}

override setName(String name) {

nameTextBox.text = name

}

def getName() {

nameTextBox.text

}

override getRoot() {

this

}

}

**package ru.agentlab.jfxed.diagramms.clazz**

**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.device.Coub**

**public class ClazzDiagram implements IDiagram {**

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

**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»#BusinessObject>**

**}**

**'''**

**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 Coub() => [**

**layoutX = 280**

**layoutY = 280**

**name = x.localName**

**]**

**jfxDiagram => [**

**nodes += target**

**]**

**}**

**qe.close()**

**}**

**}**

<?xml version="1.0" encoding="UTF-8"?>

<?eclipse version="3.0"?>

<plugin>

<extension

point="ru.agentlab.jfxed.diagram">

<diagram name="Диаграмма Блоки и стрелки" class="ru.agentlab.jfxed.diagramms.clazz.ClazzDiagram" ontoUri="http://www.agentlab.ru/jfxed/onto/igorek"/>

</extension>

</plugin>

Manifest-Version: 1.0

**Bundle-ManifestVersion**: 2

**Bundle-Name**: Blockandarrow

**Bundle-SymbolicName**: ru.agentlab.jfxed.diagramms.clazz98798;*singleton*:=true

**Bundle-Version**: 1.0.0.qualifier

**Bundle-Vendor**: AgentLab

**Bundle-RequiredExecutionEnvironment**: JavaSE-1.8

**Bundle-ActivationPolicy**: lazy

**Require-Bundle**: org.eclipse.core.runtime,

org.eclipse.emf.ecore;*visibility*:=reexport,

org.eclipse.xtext.xbase.lib;*visibility*:=reexport,

com.google.guava,

org.eclipse.fx.core,

org.eclipse.fx.javafx,

org.eclipse.fx.osgi.util,

de.fxdiagram.lib,

de.fxdiagram.core,

de.fxdiagram.annotations,

ru.agentlab.jfxed,

ru.agentlab.jfxed.figures.device;*bundle-version*="1.0.0",

org.apache.jena.core;*bundle-version*="2.11.0",

org.apache.jena.iri;*bundle-version*="0.9.5"

**Import-Package**: com.hp.hpl.jena.ontology

