Московский Государственный Технический Университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ-3

**Отчёт по лабораторной работе №5**

по курсу

**«Разработка ПО»**

Выполнил Нурматов А.Т.

Проверил: Иванов А.М.

Москва, 2014

## Задание

Создайте расширение для Eclipse используя фигуру полученную в процессе лабораторной работы №2.

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**

## Цель задания

Цель задания: познакомит студентов с технологией расширения Eclipse своими плагинами.

## Реализация

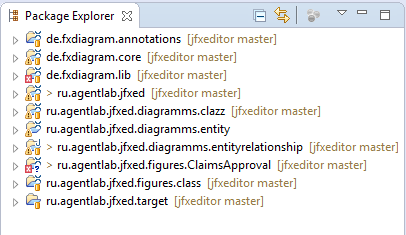


Рис.1 – Структура папок

Файл ClassFigure.xtend (для рисования фигуры)

**package** ru.agentlab.jfxed.figures.app

**import** javafx.scene.text.Text

**import** ru.agentlab.jfxed.IFigure

**import** javafx.scene.shape.ArcTo

**import** javafx.scene.shape.Path

**import** javafx.scene.shape.MoveTo

**import** de.fxdiagram.core.XNode

**import** javafx.scene.shape.LineTo

**import** javafx.scene.Group

**import** javafx.scene.layout.HBox

**import** javafx.scene.layout.VBox

**import** javafx.scene.shape.HLineTo

**import** javafx.scene.shape.CubicCurveTo

**class** ClassFigure **extends** XNode **implements** IFigure {

**private** Path path

**val** nameTextBox = **new** Text

**var** String name

**new**(String name) {

**super**(name)

**this**.name = name

}

**new**() {

**super**(" Fast[Claims \n Approval]")

name = " Fast[Claims \n Approval]"

path = **new** Path => [

styleClass += "ClassFigure"

elements += **new** MoveTo(-35, -3)

elements += **new** HLineTo(-15)

elements += **new** CubicCurveTo(-15, -65, 82, -3, 150, -32)

elements += **new** ArcTo(3, 10, 0, 150, 29, **false**, **true**)

elements += **new** MoveTo(158, -3)

elements += **new** LineTo(170, -3)

elements += **new** MoveTo(-15, -8)

elements += **new** CubicCurveTo(-7, 77, -45, -13, 150, 29)

]

node = **new** VBox => [

children += **new** Group => [

children += path

children += nameTextBox => [

text = name

styleClass += "ClassFigure-Text"

]

]

]

}

**def** setName(String name) {

nameTextBox.text = name

}

**def** getName() {

nameTextBox.text

}

**override** getRoot() {

**return** **this**

}

}

Файл ClazzDiagram.xtend

package ru.agentlab.jfxed.diagramms.cloud

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.clouds.ClassFigure

public class ClazzDiagram implements IDiagram {

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

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»#ConceptClass>

}

'''

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 ClassFigure() => [

layoutX = 180

layoutY = 180

name = x.localName

]

jfxDiagram => [

nodes += target

]

}

qe.close()

}

}

## Тестирование

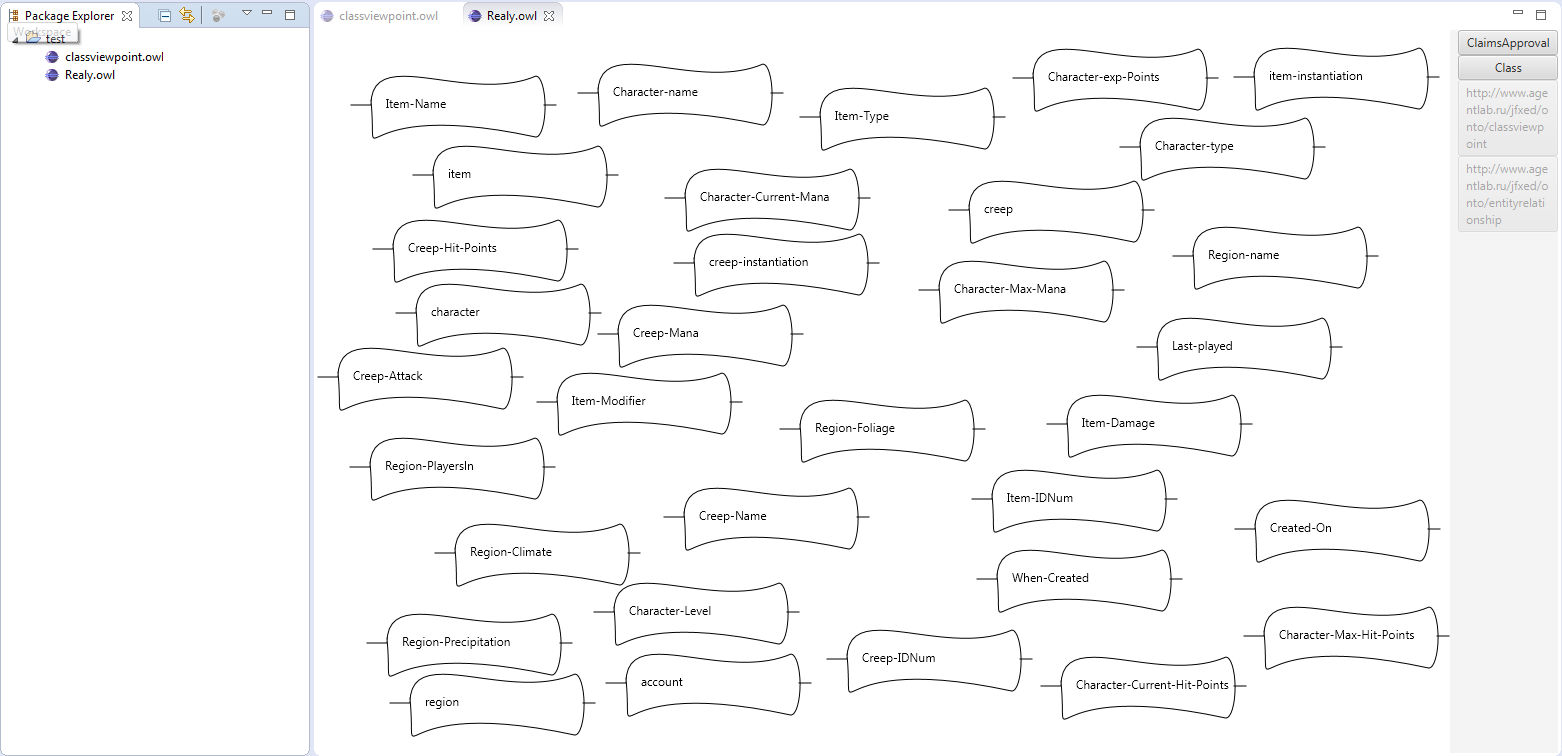
**

Рис.2 – Рисование индивидуального примера с помощью реализованной фигуры, соответствующего диаграмме