**Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования**

Московский Государственный Технический Университет имени Н.Э. Баумана

**факультет “Информатика и системы управления”**

**Лабораторная работа №5**

Выполнила:

Анацкая Л.С.

Группа ИУ 3 - 61

Вариант 2

Проверил:

Иванов А.М.

Москва

**2014**

Листинг кода:

**package** ru.agentlab.jfxed.figures.clazz

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

**import** javafx.scene.shape.QuadCurveTo

**import** javafx.scene.paint.Color

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

**import** javafx.scene.shape.VLineTo

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

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

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

**import** javafx.scene.layout.GridPane;

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

**class** Rectangle **extends** XNode **implements** IFigure {

**var** GridPane gp

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

**var** String name

**new**(String name) {

**super**(name)

**this**.name = name

}

**new**() {

**super**("Class");

name = " Class";

gp = **new** GridPane

node = gp

**val** s = Rectangle.getResource("ClassFigure.css").toExternalForm()

gp.stylesheets += s

**var** path = **new** Path()

path.getElements().addAll(

**new** MoveTo(0, 80),

**new** QuadCurveTo(40, 20, 80, 80),

**new** QuadCurveTo(120, 120, 160, 80),

**new** VLineTo(0),

**new** HLineTo(0),

**new** VLineTo(80)

)

path.setStroke(Color.*BLACK*);

path.setFill(Color.*CYAN*)

path.setStrokeWidth(2);

gp.children += path

gp.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.figures;

import ru.agentlab.jfxed.figures.clazz.\*;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class BootFigures extends Application {

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) throws Exception {

primaryStage.setTitle("Representation!");

AnchorPane root = new AnchorPane();

final Rectangle rect = new Rectangle();

root.getChildren().add(rect.getRoot());

root.getChildren().add(new VBox());

Scene scene = new Scene(root, 200, 150);

primaryStage.setScene(scene);

primaryStage.show();

}

}

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.clazz.Rectangle

public class ClazzDiagram implements IDiagram {

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

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»#BusinessActor>

}

'''

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 Rectangle() => [

layoutX = 280

layoutY = 280

name = x.localName

]

jfxDiagram => [

nodes += target

]

}

qe.close()

}

}

