Факультет ИУ "Информатика и системы управления"

Кафедра ИУ-3 "Информационные системы и телекоммуникации"

Отчет к лабораторной работе №2

по курсу "**Разработка программного обеспечения**"

направления 2304000062

# "[[Создание фигуры диаграммы на JavaFX с использованием Xtend2](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=55771143)](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=54001678)[”](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=55771143)

Продолжительность 4 часа.

Выполнил:

студент группы ИУ3-61

Алборов З.В.

Проверил:

Иванов А.М.

# Цель работы

Цель работы — формирование у студентов ряда навыков:

* Использование передовых технологий разработки GUI толстых клиентов
* Использование DSL языков на примере Xtend2
* Основы работы в системе контроля версий Git

# Задание

Создать Eclipse-проект с векторной фигуркой  в соответствии со своим вариантом задания и требованиями, добавить его в репозитарий системы контроля версий.

# Настройка инструментов

Предварительные требования:

* Eclipse с установленным e(fx)clipse с предыдущей лабораторной работы

Последовательность настройки:

1. Создать аккаунт на Github и выслать логин на [amivanoff@gmail.com](mailto:amivanoff@gmail.com) для предоставления прав на запись

2. Настроить подключение Eclipse к Github

* Использование руководство [Настройка Eclipse IDE для доступа к репозитарию GitHub по протоколу HTTP](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=41058472)
  + Указать другой адрес репозитария, скопировав его со страницы <https://github.com/agentlab/jfxeditor>

3. Установить в Eclipse поддержку языка Xtend2

* См. <https://www.eclipse.org/xtend/download.html>
  + там выбрать "Latest Release"

Исходный код программы:

package ru.agentlab.jfxed.figures;

import ru.agentlab.jfxed.figures.product.Rectangle;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

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("Hello World!");

AnchorPane root = new AnchorPane();

final Rectangle rect = new Rectangle();

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

Scene scene = new Scene(root, 200, 100);

primaryStage.setScene(scene);

primaryStage.show();

}

}

package ru.agentlab.jfxed.figures.product

import de.fxdiagram.core.XNode

import javafx.scene.layout.VBox

import javafx.scene.layout.HBox

import javafx.scene.text.Text

import javafx.scene.shape.Polyline

import ru.agentlab.jfxed.IFigure

import javafx.scene.layout.GridPane

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

node = new VBox => [

stylesheets += Rectangle.getResource("Style.css").toExternalForm()

styleClass += "ClassFigure";

children += new HBox => [

children += new Polyline(#[0,10,70,10,70,0])

]

children += new HBox => [

styleClass += "ClassFigure-Name-Section"

children += nameTextBox => [

styleClass += "ClassFigure-Text"

text = "Product"

]

]

]

}

override setName(String name) {

nameTextBox.text = name

}

def getName() {

nameTextBox.text

}

override getRoot() {

return this

}

}

.ClassFigure{

-fx-background-color: LIGHTGREEN;

-fx-background-radius: 5.0;

-fx-border-color: black;

-fx-border-width: 1.0;

-fx-border-radius: 5.0;

-fx-padding: 0.0;

}

.ClassFigure-Name-Section{

-fx-padding: 15.0;

-fx-border-color: black;

-fx-border-width: 0.0 0.0 1.0 0.0;

-fx-border-radius: 5.0 5.0 0.0 0.0;

}

.iconCENTRED {

-fx-alignment: top-left;

-fx-padding: 5.0;

}

.ClassFigure-Text{

-fx-alignment: center;

-fx-font: bold 20.0pt "Arial";

-fx-stroke-width: 0.0;

}

