*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

***«Московский государственный технический университет имени Н.Э. Баумана»   
(МГТУ им. Н.Э. Баумана)***

***Кафедра ИУ3, 3 курс, 6 семестр.***

**Отчёт**

**по лабораторной работе №2**

# “[Создание фигуры диаграммы на JavaFX](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=54001678)

# [с использованием Xtend2](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=54001678)”

**по курсу**

**“Разработка программного обеспечения”**

Выполнил:

Гольдштейн М.

Группа:

ИУ3-61

Проверил:

Иванов А.М.

**Москва, 2014**

# 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.

Вариант 7:

C:\Users\Mike\Desktop\Безымян.png

# 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 EclipseIDE для доступа к репозитариюGitHubпопротоколу 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>
  + тамвыбрать "Latest Release"

Для создания фигуры по заданию была использован метод с добавлением структуры GridPane иHBox, внутри которого была реализована группа с добавлением в нее полилинии и текста. Описание методов создания формы находится в файле ClassFigure.xtend. Описание стилей для компонентов формы находится в файле ClassFigure.css.

Листинг:

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

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

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

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

**import** ru.agentlab.jfxed.figures.BaseFigure

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

**import** javafx.scene.shape.Polygon

**import** javafx.scene.shape.Polyline

**import** javafx.scene.Group;

**class** ClassFigure **extends** BaseFigure {

String name = "ClassName";

**new** (){

root = **new** VBox

initHandlers()

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

root.stylesheets += s

root.styleClass += "ClassFigure";

//root.children += group

root.children += **new** HBox => [

styleClass += "ClassFigurePol"

children += **new** Polygon => [

points += #[

800.0, 25.0,

800.0, 20.0,

805.0, 15.0,

810.0, 20.0,

810.0, 25.0,

805.0, 20.0

]

setFill(Color.*YELLOW*)

setStroke(Color.*BLACK*)

]

]

root.children += **new** VBox => [

styleClass += "ClassFigure-Methods-List"

children += **new** Text => [

text = "Function"

]

]

}

ClassFigure{

-fx-background-color: yellow;

-fx-background-radius: 5;

-fx-border-color: black;

-fx-border-width: 1;

-fx-border-radius: 5;

/\*-fx-alignment: CENTER;\*/

-fx-padding: 0;

}

.ClassFigure-Rect{

-fx-fill: CYAN;

-fx-stroke: #FFFFFF;

-fx-stroke-type: inside;

-fx-border-radius: 5;

-fx-background-radius: 5;

-fx-arc-height: 5;

-fx-arc-width: 5;

-fx-padding: 5;

}

/\* .ClassFigure-Name-Section{

-fx-alignment: center;

-fx-padding: 5;

-fx-border-color: black;

-fx-border-width: 0 0 1 0;

-fx-border-radius: 5 5 0 0;

}\*/

.ClassFigure-Line{

}

.ClassFigure-Text{

-fx-font: bold 30pt *"Arial"*;

-fx-stroke-type: outside;

-fx-stroke-width: 0;

}

.ClassFigure-Methods-List{

-fx-padding: 20;

-fx-spacing: 15;

}

.ClassFigurePol {

-fx-background-color: yellow;

-fx-alignment: CENTER;

-fx-padding: 0 -80 0 0;

} **package** ru.agentlab.jfxed.figures;

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

**import** ru.agentlab.jfxed.figures.clazz2.Path;

**import** javafx.application.Application;

**import** javafx.event.EventHandler;

**import** javafx.scene.Node;

**import** javafx.scene.Scene;

**import** javafx.scene.input.MouseEvent;

**import** javafx.scene.layout.AnchorPane;

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

**import** javafx.stage.Stage;

**public** **class** BootFigures **extends** Application {

ClassFigure sourceToMove;

**double** sceneStartX;

**double** sceneStartY;

**double** figureStartTranslateX;

**double** figureStartTranslateY;

**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** ClassFigure source = **new** ClassFigure();

**final** Path source2 = **new** Path();

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

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

root.getChildren().add(source2.getRoot());

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

Scene scene = **new** Scene(root, 800, 600);

//scene.getStylesheets().addAll(ClassFigure.class.getResource("ClassFigure.css").toExternalForm());

primaryStage.setScene(scene);

primaryStage.show();

/\*source.getRoot().setOnMousePressed(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event) {

System.out.println("setOnMousePressed");

sceneStartX = event.getSceneX();

sceneStartY = event.getSceneY();

figureStartTranslateX = ((Node) event.getSource()).getTranslateX();

figureStartTranslateY = ((Node) event.getSource()).getTranslateY();

}

});

source.getRoot().setOnMouseDragged(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event) {

System.out.println("setOnMouseDragged");

System.out.println(event.getSceneX() + " " + event.getSceneY());

source.getRoot().setTranslateX(figureStartTranslateX + event.getSceneX() - sceneStartX);

source.getRoot().setTranslateY(figureStartTranslateY + event.getSceneY() - sceneStartY);

}

});

}

}

Результат компиляции:

