

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

Московский Государственный Технический Университет имени Н.Э. Баумана  
факультет “Информатика и системы управления”

Лабораторная работа №2

Стасенко Н.В.

Группа ИУ 3 - 61

Вариант 14

Проверил:

Иванов А.М.

## Задание

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

## 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

## Требования к качеству

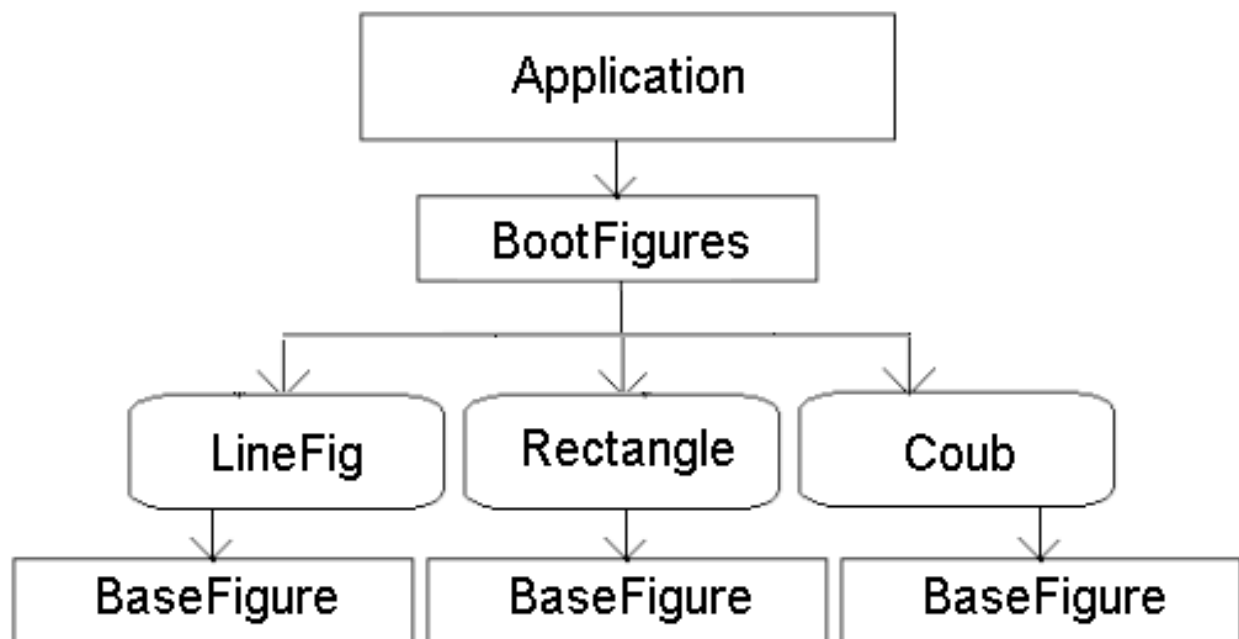
Модифицируемость

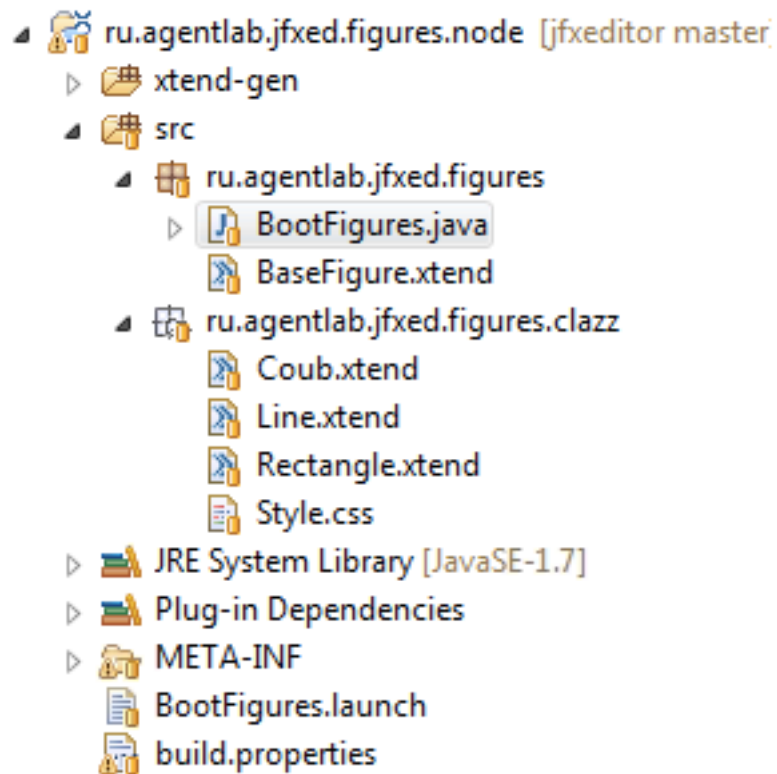
- а. Настраиваемость по цвету, размеру, шрифтам
- б. Оформление исходного кода Java, Xtend в соответствии Java Code Convention

Гибкость

- а. Блочная верстка
- б. Начальные размеры, не привязаны к определенным значениям

## Проектирование





## Структура программы

### Листинг программы

#### BootFigures.java:

```
package ru.agentlab.jfxed.figures;

import ru.agentlab.jfxed.figures.clazz.LineFig;
import ru.agentlab.jfxed.figures.clazz.Rectangle;
import ru.agentlab.jfxed.figures.clazz.Coub;
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();
        final Coub coub = new Coub();
        final LineFig line = new LineFig();
        root.getChildren().add(rect.getRoot());
```

```

        root.getChildren().add(line.getRoot());
        root.getChildren().add(coub.getRoot());
        Scene scene = new Scene(root, 250, 70);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}

```

### BaseFigure.xtend:

```

package ru.agentlab.jfxed.figures

import javafx.scene.layout.Pane

class BaseFigure {
    protected Pane root
    def public Pane getRoot() {
        return root
    }
}

```

### Coub.xtend:

```

package ru.agentlab.jfxed.figures.clazz

import javafx.scene.layout.VBox
import javafx.scene.paint.Color;
import javafx.scene.shape.Polygon
import javafx.scene.shape.Polyline
import javafx.scene.Group;
import ru.agentlab.jfxed.figures.BaseFigure

class Coub extends BaseFigure {

    new () {
        root = new VBox

        var pligon = new Polygon()
        pligon.getPoints().addAll(#[10.0, 0.0, 80.0, 00.0, 80.0, 50.0, 70.0, 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, 70.0, 10.0, 80.0, 0.0, 70.0, 10.0, 70.0, 60.0])
        var group = new Group(pligon,line)
        root.children += group

        root.translateX = 160
        root.translateY = 5
    }
}

```

### Line.xtend:

```

package ru.agentlab.jfxed.figures.clazz

import javafx.scene.layout.VBox
import ru.agentlab.jfxed.figures.BaseFigure
import javafx.scene.shape.Polyline

```

```

class LineFig extends BaseFigure {

    Polyline line
    var double [] points
    new () {
        root = new VBox

        line = new Polyline()
        points = #[7,3,10,0,20,0,20,10,17,13,17,3,7,3,7,13,17,13,17,3,20,0]
        line.getPoints().addAll(points)

        root.translateX = 100
        root.translateY = 10
        root.children += line
    }
}

```

### Rectangle.xtend:

```

package ru.agentlab.jfxed.figures.clazz

import javafx.scene.layout.VBox
import javafx.scene.text.Text
import ru.agentlab.jfxed.figures.BaseFigure

```

```

class Rectangle extends BaseFigure {

    new () {
        root = new VBox

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

        root.styleClass += "ClassFigure-Rect";

        root.translateX = 5
        root.translateY = 5

        root.children += new Text => [
            text = «Node" ]
    }
}

```

### Style.css:

```

.ClassFigure-Rect{
    -fx-background-color: CYAN;
    -fx-border-color: black;
    -fx-stroke-type: inside;
    -fx-arc-height: 5;
    -fx-arc-width: 5;
    -fx-padding: 20 40;
}

.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 14pt "Arial";  
-fx-stroke-type: outside;  
-fx-stroke-width: 0;  
}
```

