

Упражнение N5

Цел на упражнението: Запознаване с възможностите на библиотеката JavaFX, като създаване на форма и графика.

ПРИМЕР1:

1. Създаване на проект

File->New Project->JavaFX Application->LoginForm

2. Задаване на библиотеки на приложението

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.*;
import javafx.scene.*;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.*;
import javafx.geometry.*;

import javafx.scene.Scene;
```

```

import javafx.scene.control.Button;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Priority;

import javafx.scene.paint.Color;

import javafx.stage.Stage;

import javafx.scene.text.Font;

import javafx.scene.text.Text;

import javafx.scene.text.FontWeight;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.control.PasswordField;

import javafx.scene.control.ContentDisplay;

```

3. Задаване на заглавие на приложението

```

public class JavaFXExample extends Application {

    public static void main(String[] args) {

        launch(args);

    }

    @Override

    public void start(Stage primaryStage) {

        primaryStage.setTitle("JavaFX Потребителска форма");

        .....

        primaryStage.show();

    }

```

4. Добавете разположението на елементите **GridPane**, защото ви позволява да се създаде гъвкава мрежа от редове и колони, като компонентите се добавят отляво надясно и отгоре надолу. Позиционирайте кода по-долу преди реда **primaryStage.show();**

```

GridPane grid = new GridPane();
grid.setAlignment(Pos.CENTER);

```

```
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(25, 25, 25, 25));
```

```
Scene scene = new Scene(grid, 300, 275);
primaryStage.setScene(scene);
```

5. Добавяне на текст, етикети и текстови полета

Добавете следния текст след задаването на свойствата на запълване на решетката

```
Text scenetitle = new Text("Потребителски Вход");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
grid.add(scenetitle, 0, 0, 2, 1);
```

```
Label userName = new Label("Потребител:");
grid.add(userName, 0, 1);
```

```
TextField userTextField = new TextField();
grid.add(userTextField, 1, 1);
```

```
Label pw = new Label("Парола:");
grid.add(pw, 0, 2);
```

```
PasswordField pwBox = new PasswordField();
grid.add(pwBox, 1, 2);
```

6. Добавяне на бутон и текст

Контрол за бутона за приемане на данни.

Текстова контрола за изобразяване на съобщение.

Кода по-долу го добавете преди кода за scene.

```
Button btn = new Button("Вход");
HBox hbBtn = new HBox(10);
hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
hbBtn.getChildren().add(btn);
grid.add(hbBtn, 1, 4);
```

7. Добавете след това преди кода за scene за Текстовата контрола за изобразяване на съобщението

```
final Text actiontarget = new Text();
grid.add(actiontarget, 1, 6);
```

8. Добавяне на код за поддържане на събитието. Задава се бутона да изобрази текстово съобщение.

Кода по-долу го добавете преди кода за scene.

```
btn.setOnAction(new EventHandler<ActionEvent>() {  
  
    @Override  
    public void handle(ActionEvent e) {  
        actiontarget.setFill(Color.FIREBRICK);  
        actiontarget.setText("Бутон за вход е натиснат");  
    }  
});
```

9. Стартирайте приложението

ПРИМЕР2:

Създаване на анимация с библиотеката JavaFX. Примера представлява анимирано въртене на 3D кубчета

1. Задайте библиотеките към приложението в началото преди да запишете класа

```
import javafx.animation.Animation;  
import javafx.animation.KeyFrame;  
import javafx.animation.KeyValue;  
import javafx.animation.Timeline;  
import javafx.application.Application;  
import javafx.scene.*;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.RectangleBuilder;  
import javafx.scene.transform.Rotate;  
import javafx.scene.transform.Translate;  
import javafx.stage.Stage;  
import javafx.util.Duration;
```

2. Създайте класа Cube3DExample

```
public class Cube3DExample extends Application {  
  
    private Timeline animation;
```

.....

```
@Override public void start(Stage primaryStage) throws Exception {  
    init(primaryStage);  
    primaryStage.show();  
    play();  
}  
  
public static void main(String[] args) { launch(args); }  
}
```

2. Въведете метода init()

```
private void init(Stage primaryStage) {  
    Group root = new Group();  
    root.setDepthTest(DepthTest.ENABLE);  
    primaryStage.setResizable(false);  
    primaryStage.setScene(new Scene(root, 400, 150, true));  
    primaryStage.getScene().setCamera(new PerspectiveCamera());  
    root.getTransforms().addAll(  
        new Translate(400 / 2, 150 / 2),  
        new Rotate(180, Rotate.X_AXIS)  
    );  
    root.getChildren().add(create3dContent());  
}
```

3. След това създайте метода create3dContent()

```
public Node create3dContent() {  
    Cube c = new Cube(50, Color.RED, 1);  
    c.rx.setAngle(45);  
}
```

```

c.ry.setAngle(45);

Cube c2 = new Cube(50,Color.GREEN,1);

c2.setTranslateX(100);

c2.rx.setAngle(45);

c2.ry.setAngle(45);

Cube c3 = new Cube(50,Color.ORANGE,1);

c3.setTranslateX(-100);

c3.rx.setAngle(45);

c3.ry.setAngle(45);


animation = new Timeline();

animation.getKeyFrames().addAll(

    new KeyFrame(Duration.ZERO,

        new KeyValue(c.ry.angleProperty(), 0d),

        new KeyValue(c2.rx.angleProperty(), 0d),

        new KeyValue(c3.rz.angleProperty(), 0d)

    ),

    new KeyFrame(Duration.seconds(1),

        new KeyValue(c.ry.angleProperty(), 360d),

        new KeyValue(c2.rx.angleProperty(), 360d),

        new KeyValue(c3.rz.angleProperty(), 360d)

    ));

animation.setCycleCount(Animation.INDEFINITE);


return new Group(c,c2,c3);

}

```

4. След което добавете метода в класа за стартиране на анимацията

```

public void play() {

    animation.play();

}

```

5. Метода стоп за спиране на анимацията

```

@Override public void stop() {

    animation.pause();

}

```

6. Създайте вътрешния клас Cube, който наследява класа javafx.scene.Group. Този клас представлява групов възел съдържа ObservableList на под групи, които се визуализират по ред, когато този възел се визуализира. Групата ще поеме колективните граници на своите под групи и не може да бъде пряко изменена. Всяка трансформация, ефект или състояние, приложени към група, ще се прилагат за всички под групи от тази група.

Класа се добавя в основния клас след метода stop()

```

public class Cube extends Group {

    final Rotate rx = new Rotate(0, Rotate.X_AXIS);

    final Rotate ry = new Rotate(0, Rotate.Y_AXIS);

    final Rotate rz = new Rotate(0, Rotate.Z_AXIS);

    public Cube(double size, Color color, double shade) {

        getTransforms().addAll(rz, ry, rx);

        getChildren().addAll(

            RectangleBuilder.create() // задно лице

                .width(size).height(size)

                .fill(color.deriveColor(0.0, 1.0, (1 - 0.5*shade), 1.0))

                .translateX(-0.5*size)

                .translateY(-0.5*size)

                .translateZ(0.5*size)

                .build(),

            RectangleBuilder.create() // долно лице

```

```
.width(size).height(size)

.fill(color.deriveColor(0.0, 1.0, (1 - 0.4*shade), 1.0))

.translateX(-0.5*size)

.translateY(0)

.rotationAxis(Rotate.X_AXIS)

.rotate(90)

.build(),
```

RectangleBuilder.create() //дясно лице

```
.width(size).height(size)

.fill(color.deriveColor(0.0, 1.0, (1 - 0.3*shade), 1.0))

.translateX(-1*size)

.translateY(-0.5*size)

.rotationAxis(Rotate.Y_AXIS)

.rotate(90)

.build(),
```

RectangleBuilder.create() // ляво лице

```
.width(size).height(size)

.fill(color.deriveColor(0.0, 1.0, (1 - 0.2*shade), 1.0))

.translateX(0)

.translateY(-0.5*size)

.rotationAxis(Rotate.Y_AXIS)

.rotate(90)

.build(),
```

RectangleBuilder.create() //горно лице

```
.width(size).height(size)

.fill(color.deriveColor(0.0, 1.0, (1 - 0.1*shade), 1.0))

.translateX(-0.5*size)
```



```

        .translateY(-1*size)

        .rotationAxis(Rotate.X_AXIS)

        .rotate(90)

        .build(),
RectangleBuilder.create() //горно лице

        .width(size).height(size)

        .fill(color)

        .translateX(-0.5*size)

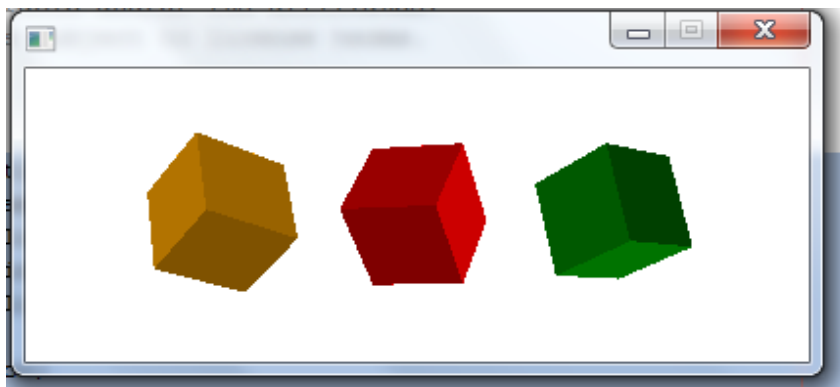
        .translateY(-0.5*size)

        .translateZ(-0.5*size)

        .build()

    );
}
}

```



Задачи:

1. Сменете цвета на кубчетата
2. Добавете четвърто кубче към анимацията дадена в Пример 2