**UNIVERSITATEA TEHNICĂ A MOLDOVEI**

**FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ**

**CATEDRA CALCULATOARE**

**Raport**

**LUCRARE DE LABORATOR NR.6**

**la Tehnici avansate de programare**

**A realizat: Nicolenco Eugeniu gr. C-162**

**A verificat: Lector univ. Rotaru Lilia**

**Chișinău 2018**

**1. Tema lucrării:**

Cercetarea JavaFx

**2.**  **Scopul lucrării:**

* Însuşirea modalităţilor de lucru cu grafica in JavaFx;

**3. Etapele de realizare:**

* 1. Realizarea tipurilor grafice
  2. Realizarea animatiilor;
  3. Crearea functiilor proprii;
  4. Crearea interfeţii programului;
  5. Prezentarea lucrării.

*Varianta 8*

1. Să se reprezinte un dreptunghi, ce se roteste în planul ecramului în jurul unuia dintre virfuri.

***Program Listingul***

import javafx.application.Application;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.paint.Color;

import javafx.scene.shape.Rectangle;

import javafx.stage.Stage;

import javafx.animation.RotateTransition;

import javafx.util.Duration;

import javafx.scene.Node;

import javafx.scene.transform.Translate;

import javafx.scene.control.\*;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

public class javafxapplication1 extends Application {

private void movePivot(Node node, double x, double y){

node.getTransforms().add(new Translate(-x,-y));

node.setTranslateX(x); node.setTranslateY(y);

}

int angleDirection=-360;

private int Direction(int x){

return -x;

}

int[] pivotSetX={-40,40,40,-40};

int[] pivotSetY={-45,-45,45,45};

int pivotIndex=0;

int zoom=0;

boolean removed=false;

@Override

public void start(Stage stage) {

///Creating Objects

Button reverseDirectionBtn = new Button();

Button changeCornerBtn = new Button();

Button changeColorBtn = new Button();

Button plus = new Button();

Button minus = new Button();

Button start = new Button();

Button stop = new Button();

Button resetPivot = new Button();

Rectangle rectangle1 = new Rectangle(150, 150, 80, 90);

Rectangle rectangle2 = new Rectangle(150, 150, 80, 90);

RotateTransition rotateTransition = new RotateTransition();

Label textarea = new Label();

Label secondtextarea = new Label();

//-----End of Creating Objects

//----Settings

rectangle1.setFill(Color.BLUE);

rectangle1.setStroke(Color.BLACK);

rectangle2.setFill(Color.BURLYWOOD);

rectangle2.setStroke(Color.BLACK);

movePivot(rectangle2, pivotSetX[0], pivotSetY[0]);

rotateTransition.setDuration(Duration.millis(1000));

rotateTransition.setNode(rectangle2);

rotateTransition.setByAngle(angleDirection);

rotateTransition.setCycleCount(-1);

rotateTransition.setAutoReverse(false);

rotateTransition.play();

reverseDirectionBtn.setText("Change Direction");

changeCornerBtn.setText("Change Corner");

changeCornerBtn.relocate(120, 0);

changeColorBtn.setText("Change Color");

changeColorBtn.relocate(0,25);

minus.setText("Zoom in");

minus.relocate(100, 25);

plus.setText("Zoom out");

plus.relocate(170,25);

start.setText("Start");

start.relocate(250,0);

stop.setText("Stop");

stop.relocate(250,25);

resetPivot.setText("Remove zoom");

resetPivot.relocate(400, 0);

textarea.setLayoutX(400);

textarea.setLayoutY(30);

secondtextarea.setLayoutX(400);

secondtextarea.setLayoutY(60);

secondtextarea.setText("Animatia in process");

//-----Button Actions -----//

reverseDirectionBtn.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

angleDirection=Direction(angleDirection);

rotateTransition.stop();

rotateTransition.setByAngle(angleDirection);

rotateTransition.play();

}

});

changeCornerBtn.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

rotateTransition.stop();

if(pivotIndex>2){

movePivot(rectangle2,Direction(pivotSetX[pivotIndex]),Direction(pivotSetY[pivotIndex]));

pivotIndex=0;

movePivot(rectangle2,pivotSetX[pivotIndex],pivotSetY[pivotIndex]);

}else{

movePivot(rectangle2,Direction(pivotSetX[pivotIndex]),Direction(pivotSetY[pivotIndex]));

pivotIndex++;

movePivot(rectangle2,pivotSetX[pivotIndex],pivotSetY[pivotIndex]);

}

rotateTransition.play();

}

});

changeColorBtn.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

rotateTransition.stop();

rectangle2.setFill(Color.color(Math.random(),Math.random(),Math.random()));

rotateTransition.play();

}

});

plus.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

rotateTransition.stop();

rectangle2.getTransforms().add(new Translate(1,1));

zoom--;

rotateTransition.play();

removed=true;

Integer x=zoom;

textarea.setText(x.toString());

}

});

minus.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

rotateTransition.stop();

rectangle2.getTransforms().add(new Translate(-1,-1));

zoom++;

rotateTransition.play();

removed=true;

Integer x=zoom;

textarea.setText(x.toString());

}

});

resetPivot.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

if(removed){

rectangle2.getTransforms().add(new Translate(zoom,zoom));

}

removed=false;

zoom=0;

Integer x=zoom;

textarea.setText(x.toString());

}

});

start.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

rotateTransition.play();

secondtextarea.setText("Animatie in process");

}

});

stop.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

rotateTransition.stop();

secondtextarea.setText("Animatie oprita");

}

});

Group root = new Group(rectangle1, rectangle2);

root.getChildren().add(reverseDirectionBtn);

root.getChildren().add(changeCornerBtn);

root.getChildren().add(plus);

root.getChildren().add(minus);

root.getChildren().add(changeColorBtn);

root.getChildren().add(start);

root.getChildren().add(stop);

root.getChildren().add(resetPivot);

root.getChildren().add(textarea);

root.getChildren().add(secondtextarea);

Scene scene = new Scene(root, 600, 300);

stage.setTitle("Lab 6 ");

stage.setScene(scene);

stage.show();

}

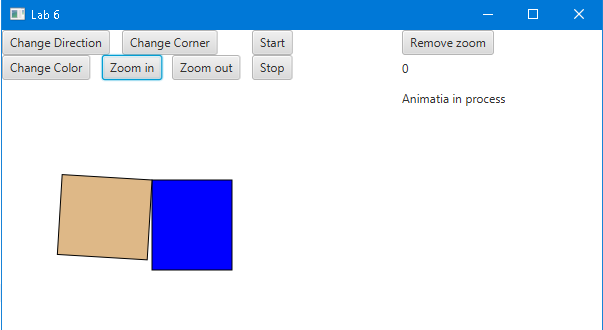
public static void main(String args[]){

launch(args);

}

}

***Rezultatul executarii:***



**Concluzie:**

In urma efectuarii acestei lucrari am insusit modalitatile de creare si realizare a graficii in JavaFx. In afara cerintelor curiculare am adaugat citeva obiecte pentru manipularea cu obiectul general(Dreptunghiul) acestea sunt: Rotirea in diferite directii;

Schimbarea culorii, Marirea/Miscarea distantei, Schimbarea coltului asupra caruia se roteste; pausa/start a animatiei.