МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

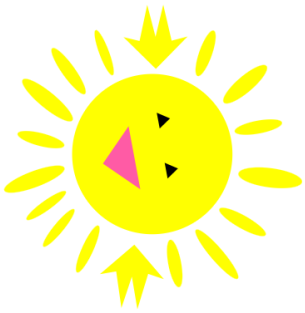
Лабораторна робота № 3
з дисципліни “ МАОКГ”

Виконав
студент III курсу
групи КП-81
Янковський Дмитро Олексійович
(*прізвище, ім’я, по батькові*)
Варіант № 21

Зарахована
“____” “_____” 2021 р.
викладачем
Шкурат О. С.
(*прізвище, ім’я, по батькові*)

Київ-2020

Результат :



Лістинг коду програми (class Sun):

```
import
javafx.animation.*
;
                    import javafx.application.Application;
                    import javafx.scene.Group;
                    import javafx.scene.Scene;
                    import javafx.scene.paint.Color;
                    import javafx.scene.shape.*;
                    import javafx.stage.Stage;
                    import javafx.util.Duration;
                    import javafx.scene.paint.Color;
                    import javafx.scene.paint.Paint;


                    public class Sun extends Application {


                        private static double X (double originalX){
                            return originalX + 300;
                        }
                        private static double Y (double originalY){
                            return originalY + 200;
                        }


                        public static void main(String[] args) {
                            launch(args);
                        }


                        public void start(Stage primaryStage) {
                            Group root = new Group();
                            Scene scene = new Scene(root, 1200, 600);


                            //head
                            Ellipse head = new Ellipse(X(45), Y(0), 100, 100);
                            head.setFill(Color.rgb(255, 255, 1));
                            root.getChildren().add(head);
```

```
Polygon mouth = new Polygon();
mouth.getPoints().addAll(320.0, 220.0,
        400.0, 220.0,
        360.0, 260.0);
mouth.setFill(Color.rgb(255,91,165));
mouth.setRotate(-20);
root.getChildren().add(mouth);


Polygon eye1 = new Polygon();
eye1.getPoints().addAll(290.0, 200.0,
        310.0, 200.0,
        300.0, 185.0);
eye1.setFill(Color.rgb(0,0,0));
eye1.setRotate(-20);
root.getChildren().add(eye1);


Polygon eye2 = new Polygon();
eye2.getPoints().addAll(350.0, 180.0,
        370.0, 180.0,
        360.0, 165.0);
eye2.setFill(Color.rgb(0,0,0));
eye2.setRotate(-20);
root.getChildren().add(eye2);




//lower beams
Ellipse beam1 = new Ellipse(X(-40), Y(120), 8, 35);
beam1.setRotate(45);
beam1.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam1);


Ellipse beam2 = new Ellipse(X(-70), Y(80), 6, 30);
```

```
beam2.setRotate(60);
beam2.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam2);


Ellipse beam3 = new Ellipse(X(0), Y(140), 10, 40);
beam3.setRotate(30);
beam3.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam3);


Ellipse beam4 = new Ellipse(X(50), Y(140), 8, 30);
beam4.setRotate(5);
beam4.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam4);


Ellipse beam5 = new Ellipse(X(100), Y(140), 10, 40);
beam5.setRotate(-30);
beam5.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam5);


Ellipse beam6 = new Ellipse(X(130), Y(120), 6, 30);
beam6.setRotate(-45);
beam6.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam6);


Ellipse beam7 = new Ellipse(X(155), Y(100), 6, 30);
beam7.setRotate(-60);
beam7.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam7);


Ellipse beam8 = new Ellipse(X(175), Y(60), 7, 32);
beam8.setRotate(-75);
beam8.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam8);
```

```
//upper beams
Ellipse beam9 = new Ellipse(X(175), Y(-50), 8, 30);
beam9.setRotate(-110);
beam9.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam9);


Ellipse beam10 = new Ellipse(X(155), Y(-100), 8, 40);
beam10.setRotate(-140);
beam10.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam10);


Ellipse beam11 = new Ellipse(X(110), Y(-125), 8, 33);
beam11.setRotate(-160);
beam11.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam11);


Ellipse beam12 = new Ellipse(X(50), Y(-150), 12, 40);
beam12.setRotate(-182);
beam12.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam12);


Ellipse beam13 = new Ellipse(X(-10), Y(-125), 8, 30);
beam13.setRotate(-200);
beam13.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam13);


Ellipse beam14 = new Ellipse(X(-60), Y(-100), 10, 35);
beam14.setRotate(-230);
beam14.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam14);
```

```
Ellipse beam15 = new Ellipse(X(-90), Y(-45), 8, 30);
beam15.setRotate(-260);
beam15.setFill(Color.rgb(255, 255, 1));
root.getChildren().add(beam15);


//arms


Polygon arm_left = new Polygon();
arm_left.getPoints().addAll(240.0, 220.0,
        200.0, 180.0,
        205.0, 200.0,
        160.0, 210.0,
        200.0, 220.0,
        160.0, 230.0,
        205.0, 240.0,
        200.0, 260.0);
arm_left.setFill(Color.rgb(255, 255, 1));
arm_left.setRotate(-10);
root.getChildren().add(arm_left);


Polygon arm_right = new Polygon();
arm_right.getPoints().addAll(460.0, 200.0,
        500.0, 160.0,
        495.0, 180.0,
        540.0, 190.0,
        500.0, 200.0,
        540.0, 210.0,
        495.0, 220.0,
        500.0, 240.0);
arm_right.setFill(Color.rgb(255, 255, 1));
arm_right.setRotate(-5);
root.getChildren().add(arm_right);
```

```
         //Animation
        int cycleCount = 2;
        int time = 2000;


        ScaleTransition scaleTransition = new
ScaleTransition(Duration.millis(time), root);
        scaleTransition.setToX(2);
        scaleTransition.setToY(2);
        scaleTransition.setAutoReverse(true);


        RotateTransition rotateTransition = new
RotateTransition(Duration.millis(time), root);
        rotateTransition.setByAngle(360f);
        rotateTransition.setCycleCount(cycleCount);
        rotateTransition.setAutoReverse(true);


        TranslateTransition translateTransition = new
TranslateTransition(Duration.millis(time), root);
        translateTransition.setFromX(150);
        translateTransition.setToX(50);
        translateTransition.setCycleCount(cycleCount+1);
        translateTransition.setAutoReverse(true);


        TranslateTransition translateTransition2 = new
TranslateTransition(Duration.millis(time), root);
        translateTransition2.setFromX(50);
        translateTransition2.setToX(150);
        translateTransition2.setCycleCount(cycleCount+1);
        translateTransition2.setAutoReverse(true);
```

```
        ScaleTransition scaleTransition2 = new
ScaleTransition(Duration.millis(time), root);
        scaleTransition2.setToX(0.1);
        scaleTransition2.setToY(0.1);
        scaleTransition2.setCycleCount(cycleCount);
        scaleTransition2.setAutoReverse(true);


        ParallelTransition parallelTransition = new
ParallelTransition();
        parallelTransition.getChildren().addAll(
                rotateTransition,
                scaleTransition,
                scaleTransition2,
                translateTransition
        );
        parallelTransition.setCycleCount(Timeline.INDEFINITE);
        parallelTransition.play();
        //End of animation


        primaryStage.setResizable(false);
        primaryStage.setTitle("Lab 3");
        primaryStage.setScene(scene);
        primaryStage.show();
    }


}
```