

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** **1**

з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”

|  |  |  |  |
| --- | --- | --- | --- |
| Виконав  студент III курсу  групи КП-82  Мельничук Олексій Геннадійович  (*прізвище, ім’я, по батькові*)  варіант № 12 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2021

**Варіант завдання**

**Завдання**: Створити малюнок за варіантом користуючись графічними примітивами бібліотеки JavaFX.

**Варіант: 12**



# 

**Код програми**

|  |
| --- |
| **Main.java** |
| package application;    import javafx.application.Application;  import javafx.stage.Stage;  import javafx.scene.Scene;  import javafx.scene.Group;  import javafx.scene.paint.Color;  import javafx.scene.shape.\*;  public class Main extends Application{      double WIDTH = 640;  double HEIGHT = 480;    double DELTA = 5;      public static void main (String args[]) {  launch(args); // main method  }    public double convertDegToRad(double deg)  {  return deg \* Math.PI / 180;  }    public void paintSunRays(Group root, double radius, double xc, double yc, double number, Color col)  {  //rays are icosceles triangles  Polygon ray;  double k1 = 0.375; //allows for spacing between rays  double k2 = 1 - k1;  for(int i = 0; i < number; i++)  {  ray = new Polygon();  double angle1 = 360 / number \* (i - k1) + 90;  double angle2 = 360 / number \* (i + 1 - k2) + 90;  double angle3 = 360 / number \* i + 90;  ray.getPoints().addAll(new Double[]{  radius \* Math.cos(convertDegToRad(angle1)) + xc, radius \* Math.sin(convertDegToRad(angle1)) + yc, //right point  radius \* Math.cos(convertDegToRad(angle2)) + xc, radius \* Math.sin(convertDegToRad(angle2)) + yc, //left point  (radius \* 2) \* Math.cos(convertDegToRad(angle3)) + xc, (radius \* 2) \* Math.sin(convertDegToRad(angle3)) + yc //top point  });    ray.setFill(col);  root.getChildren().add(ray);  }    }      public void paintNavalSunRays(Group root, double radius, double xc, double yc, double number, Color col)  {  //rays are right triangles  Polygon ray;  double k1 = 0.375;  double k2 = 1 - k1;  double radius2 = radius - DELTA / 2;  for(int i = 0; i < number; i++)  {  ray = new Polygon();  double angle2 = 360 / number \* (i + 1 - k2) + 90 - DELTA / 2;  double angle3 = 360 / number \* i + 90;  ray.getPoints().addAll(new Double[]{  radius2 \* Math.cos(convertDegToRad(angle3)) + xc, radius2 \* Math.sin(convertDegToRad(angle3)) + yc,  radius2 \* Math.cos(convertDegToRad(angle2)) + xc, radius2 \* Math.sin(convertDegToRad(angle2)) + yc,  (radius2 \* 2 - DELTA \* 2) \* Math.cos(convertDegToRad(angle3)) + xc, (radius2 \* 2 - DELTA \* 2) \* Math.sin(convertDegToRad(angle3)) + yc  });    ray.setFill(col);  root.getChildren().add(ray);  }    }    public void paintRandomStars(Group root, double radius, double xc, double yc, Color col, int number)  {  Polygon star;  double x = 0;  double y = 0;  double starR = 0;    double randR = 0;  double randP = 0;    double lower = radius \* 2; //zone for coordinates  double upper = radius \* 3.5;    for(int i = 0; i < number; i++)  {  randR = Math.random() \* (upper - lower) + lower;  randP = convertDegToRad(Math.random() \* 360);    x = randR \* Math.cos(randP) + xc;  y = randR \* Math.sin(randP) + yc;    starR = Math.random() \* DELTA \* 2 + DELTA \* 0.125;  star = getFourPointStar(starR, x, y);  star.setFill(col);  root.getChildren().add(star);  }  }    public Polygon getFourPointStar(double radius, double xc, double yc)  {  double k = radius / 9;  Polygon star = new Polygon();  star.getPoints().addAll(new Double[]{  xc, yc - radius, //north x,y  xc + k, yc - k, //north east  xc + radius, yc, //east  xc + k, yc + k, //south east  xc, yc + radius, //south  xc - k, yc + k, //south west  xc - radius, yc, //west  xc - k, yc - k //south west  });  return star;  }    public void paintEye(Group root, double radius, double xc, double yc, Color sun, Color bg)  {  double edgeLeftX = xc - radius + DELTA \* 2;  double edgeLeftY = yc;  double edgeRightX = xc + radius - DELTA \* 2;  double edgeRightY = yc;    double controlPoint1X = xc;  double controlPoint1Y = yc - radius \* 5 / 6;  double controlPoint2X = xc;  double controlPoint2Y = yc + radius / 6 \* 5;      QuadCurve curve1 = new QuadCurve();    curve1.setStartX(edgeLeftX);  curve1.setStartY(edgeLeftY);  curve1.setEndX(edgeRightX);  curve1.setEndY(edgeRightY);  curve1.setControlX(controlPoint1X);  curve1.setControlY(controlPoint1Y);    curve1.setFill(bg);  root.getChildren().add(curve1);      QuadCurve curve2 = new QuadCurve();    curve2.setStartX(edgeLeftX);  curve2.setStartY(edgeLeftY);  curve2.setEndX(edgeRightX);  curve2.setEndY(edgeRightY);  curve2.setControlX(controlPoint2X);  curve2.setControlY(controlPoint2Y);    curve2.setFill(bg);  root.getChildren().add(curve2);      Circle iris = new Circle(xc, yc, radius / 2.75);  iris.setFill(sun);  root.getChildren().add(iris);    Polygon pupil = new Polygon();  pupil.getPoints().addAll(new Double[]{  xc, yc - radius / 10,  xc + radius / 20, yc,  xc, yc + radius / 10,  xc - radius / 20, yc  });  pupil.setFill(bg);  root.getChildren().add(pupil);  }    @Override  public void start(Stage stage) // start - is the main entry point for all JavaFX applications  {  double radius = HEIGHT / 8;  double xc = WIDTH / 2;  double yc = HEIGHT / 2;    Color bgCol = Color.rgb(242,230,208);  Color sunCol = Color.rgb(43,44,64);    Group root = new Group(); // making the root of all scene's objects  Scene scene = new Scene (root, WIDTH, HEIGHT); // making the scene for root object with size of 500\*400 pixels;    scene.setFill(bgCol);    //stars as backdrop  paintRandomStars(root, radius, xc, yc, sunCol, 32);    //drawing the sun: rays  paintSunRays(root, radius, xc, yc, 24, sunCol);    //main circle  Circle c1 = new Circle(xc, yc, radius);  c1.setFill(sunCol);  root.getChildren().add(c1);    //details for rays  paintNavalSunRays(root, radius, xc, yc, 24, bgCol);  //two circles for creating a circuit  Circle c2 = new Circle(xc, yc, radius - DELTA / 2);  c2.setFill(bgCol);  root.getChildren().add(c2);    Circle c3 = new Circle(xc, yc, radius - DELTA);  c3.setFill(sunCol);  root.getChildren().add(c3);    //two curves: top and bottom for eye sclera  paintEye(root, radius, xc, yc, sunCol, bgCol);    //detail for eye's iris  paintSunRays(root, radius / 6, xc, yc, 32, bgCol);    stage.setScene(scene);  stage.show();    }  } |

# **Результати роботи програми**

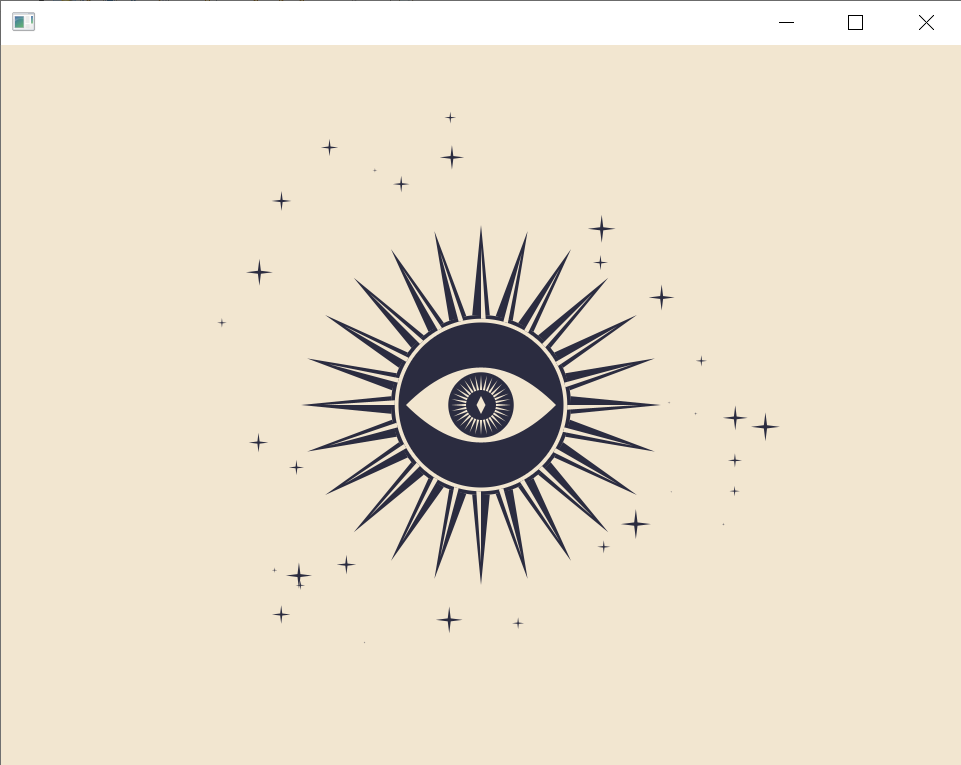


Рис.1. Скріншот вікна завдання