

# 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 № 2

з дисципліни "Математичні та алгоритмічні основи комп'ютерної графіки"

Виконав		Зарахована
студент III курсу		" 20 p.
групи КП-82		викладачем
Мельничук Олексій Геннадійович (прізвище, ім'я, по батькові)	Шкурат Оксаною Сергіївною (прізвище, ім'я, по батькові)	

варіант № 12

**Тема**: Побудова та анімація зображень за допомогою Java2D

**Мета**: Ознайомитися з можливостями побудови зображень та їх анімації у Java2D

### Завдання

За допомогою Java 2D намалювати картинку з лабораторної роботи №1 (за варіантом).

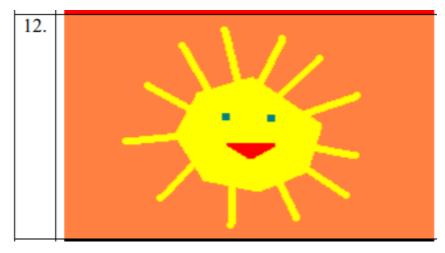
Додатково виконати:

- 1. Хоча б 1 стандартний примітив, та хоча б 1 фігуру, побудовану по точкам (ламаною).
- 2. Хоча б 1 фігуру залити градієнтною фарбою за вибором (в цьому випадку колір може не співпадати з варіантом із лабораторної роботи № 1).
- 3. На достатній відстані від побудованого малюнку намалювати прямокутну рамку, всередині якої відбуватиметься анімація. Тип лінії рамки задано за варіантом.
- 4. Виконати анімацію малюнку, за варіантом. При цьому рамка повинна залишатися статичною. Взаємодія з рамкою не обов'язкова, якщо не передбачено варіантом.

# Варіант: 12

**Типи анімації:** 1 (Рух по колу проти годинникової стрілки), 8 (Обертання навколо кута малюнка проти годинникової стрілки)

Тип лінії рамки: JOIN\_ROUND



### Код програми

## Main.java

```
package application;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;
import java.lang.Math;
import java.awt.geom.*;
public class Star extends JPanel implements ActionListener {
      Timer timer;
      private double angle = 0;
      private double xAxis;
      private double yAxis;
      private static int padding = 20;
      private double tx = 6;
      private double ty = 6;
      private static double radius;
      private double bigAngle = 0;
      private static int windowWidth = 1080;
      private static int windowHeight = 640;
      private static int maxWidth;
      private static int maxHeight;
      private static Color bgColor = new Color(255,128,64);
      private static Color eyeColor = new Color(124,188,68);
      private static Color sunColor = Color.yellow;
      private static Color mouthColor = Color.red;
      private double starPoints[][] = {
                    \{4, -90\}, \{117, -9\}, \{96, 62\}, \{14, 94\},
                    \{-75, 76\}, \{-119, 5\}, \{-84, -61\}
      };
      private double mouthPoints[][] = {
                    \{-35, 21\}, \{40, 21\}, \{2, 41\}
      };
      private static int linePoints[][] = {
                    { 250, 135 }, {203, 53 },
                    { 300, 118 }, {275, 26 },
                    { 331, 128 }, {376, 42 },
                    { 367, 143 }, {440, 74 },
```

```
{ 400, 178 }, {505, 141 },
                    { 424, 229 }, {502, 243 },
                    { 406, 261 }, {485, 312 },
                    { 370, 288 }, {399, 350 },
                    { 290, 289 }, {288, 362 },
                    { 226, 254 }, {166, 316 },
                    { 201, 209 }, {107, 218 },
                    { 222, 168 }, {143, 121 }
      };
      public static void editLinepoints()
             for(int i = 0; i < linePoints.length; i++)</pre>
                    linePoints[i][0] -= 319;
                    linePoints[i][1] -= 204;
             }
      public Star() {
             //renewing timer
             timer = new Timer(10, this);
             timer.start();
      public void setRendering(Graphics2D g2d)
             g2d.setRenderingHint(RenderingHints.KEY ANTIALIASING,
RenderingHints. VALUE ANTIALIAS ON);
             g2d.setRenderingHint(RenderingHints.KEY RENDERING,
RenderingHints.VALUE RENDER QUALITY);
      }
      public void paint(Graphics g) {
             radius = maxHeight / 3 - 2 * padding;
             Graphics2D g2d = (Graphics2D)g;
             setRendering(g2d);
             g2d.setBackground(bgColor);
             g2d.clearRect(0, 0, windowWidth, windowHeight);
             g2d.setColor(sunColor);
             DrawStroke(g2d, 14);
             //move coordinate center
             g2d.translate(maxWidth/2, maxHeight/2);
             //move on orbit
             g2d.translate(tx, ty);
             //sun stroke
             BasicStroke bs2 = new BasicStroke(5, BasicStroke.CAP BUTT,
BasicStroke.JOIN_MITER);
             g2d.setStroke(bs2);
             GeneralPath sunFigure = getPolygon(starPoints);
             //calculate axis of polygon rotation
```

```
xAxis = sunFigure.getBounds2D().getMinX();
             yAxis = sunFigure.getBounds2D().getMinY();
             double gradX = sunFigure.getBounds2D().getMaxX();
             double gradY = sunFigure.getBounds2D().getMaxY();
             g2d.rotate(angle, xAxis, yAxis);
             //draw sun rays
             g2d.setStroke(new BasicStroke(10, BasicStroke.CAP_ROUND,
BasicStroke.CAP_ROUND));
             for(int i = 0; i < linePoints.length; i+=2)</pre>
                    g2d.drawLine(linePoints[i][0], linePoints[i][1], linePoints[i+1][0],
linePoints[i+1][1]);
             GradientPaint gp = new GradientPaint((int)xAxis, (int)yAxis, sunColor,
(int)gradX, (int)gradY, bgColor, true);
             //draw sun polygon
             g2d.setPaint(gp);
             g2d.fill(sunFigure);
             //draw eyes
             g2d.setColor(eyeColor);
             g2d.fillRect(-40, -29, 10, 10);
             g2d.fillRect(30, -26, 10, 10);
             //draw mouth
             g2d.setColor(mouthColor);
             GeneralPath mouthFigure = getPolygon(mouthPoints);
             g2d.fill(mouthFigure);
      }
      public void actionPerformed(ActionEvent e) {
             bigAngle -= 0.03;
             tx = Math.cos(bigAngle) * radius;
             ty = Math.sin(bigAngle) * radius;
             angle -= 0.01;
             repaint();
      public void DrawStroke(Graphics2D g2d, int strokeWidth)
             BasicStroke bs1 = new BasicStroke((float)strokeWidth, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_ROUND);
             g2d.setStroke(bs1);
             g2d.drawRect(padding, padding, maxWidth - 2 * padding, maxHeight - 2 *
padding);
      public GeneralPath getPolygon(double points[][])
             GeneralPath polygon = new GeneralPath();
```

```
polygon.moveTo(points[0][0], points[0][1]);
      for (int k = 1; k < points.length; k++)
             polygon.lineTo(points[k][0], points[k][1]);
      polygon.closePath();
      return polygon;
public static void main(String[] args) {
      editLinepoints();
      JFrame frame = new JFrame("Приклад анімації");
      frame.add(new Star());
      frame.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
      frame.setSize(windowWidth, windowHeight);
      frame.setResizable(false);
      frame.setLocationRelativeTo(null);
      frame.setVisible(true);
      Dimension size = frame.getSize();
      Insets insets = frame.getInsets();
      maxWidth = windowWidth - insets.left - insets.right - 1;
      maxHeight = windowHeight - insets.top - insets.bottom - 1;
```

# Результати роботи програми



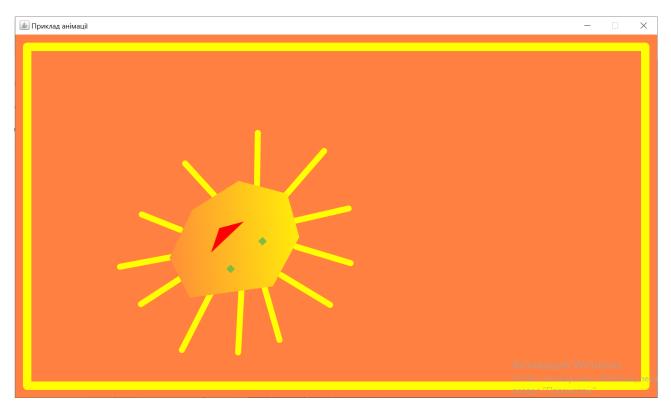


Рис.1-2. Скріншоти вікна завдання

# Висновки

Виконавши дану лабораторну роботу я ознайомився з Java2D, його методами та можливостями побудови та анімації зображень. З допомогою Java2D було намальовано декілька полігонів по точкам, рамка та градієнт, а також зображення було анімовано для руху проти часової стрілки по колу та навколо осі.