

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 2**

з дисципліни “ МАОКГ”

|  |  |  |
| --- | --- | --- |
| Виконав  студент III курсу  групи КП-82  Залізний Захар Юрійович  (*прізвище, ім’я, по батькові*)  Варіант № 7 |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2021 р.  викладачем  Шкурат О. С.  (*прізвище, ім’я, по батькові*) |

Київ-2021

**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). 11

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.

**Результат виконання програми :**



**Код програми:**

|  |
| --- |
| **Lab2.java** |
| import javax.swing.\*;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class Lab2 extends JPanel implements ActionListener {  private static int *maxWidth*;  private static int *maxHeight*;  Timer timer;  static final int *rectWidth* = 300;  static final int *rectHeight* = 300;  private int dx = 0;  private int tx = 0;  private int dy = 1;  private int ty = 1;  private double scale = 1;  private double angle = 0;  private boolean v = false;  private double speed = 5;  private double a = 300;  private double pos = a/2;  public Lab2() {  timer = new Timer(10, this);  timer.start();  }  public void paint(Graphics g) {  super.paint(g);  Graphics2D g2d = (Graphics2D) g;  RenderingHints rh = new RenderingHints(RenderingHints.*KEY\_ANTIALIASING*, RenderingHints.*VALUE\_ANTIALIAS\_ON*);  rh.put(RenderingHints.*KEY\_RENDERING*, RenderingHints.*VALUE\_RENDER\_QUALITY*);  g2d.setRenderingHints(rh);  g2d.setBackground(Color.*LIGHT\_GRAY*);  g2d.clearRect(0, 0, *maxWidth*, *maxHeight*);  drawBorder(g2d);  // Set center  g2d.translate(-tx + 600 , -ty + 600 );  g2d.rotate(Math.*toRadians*(angle\*100),125,100);  g2d.setComposite(AlphaComposite.*getInstance*(AlphaComposite.*SRC\_OVER*, (float) scale));  // Polygon1  GradientPaint gp = new GradientPaint(10, 100, Color.*GREEN*, 20, 2, Color.*GRAY*, true);  g2d.setPaint(gp);  int[] xRect = new int[]{100, 200, 130, 50};  int[] yRect = new int[]{40, 70, 100, 90};  Polygon rect = new Polygon(xRect, yRect, 4);  g2d.drawPolygon(rect);  g2d.fillPolygon(rect);  //Polygon2  GradientPaint gp1 = new GradientPaint(10, 100, Color.*GREEN*, 20, 2, Color.*GRAY*, true);  g2d.setPaint(gp1);  int[] xRect1 = new int[]{50, 130, 150, 70};  int[] yRect1 = new int[]{90, 100, 140, 145};  Polygon rect1 = new Polygon(xRect1, yRect1, 4);  g2d.drawPolygon(rect1);  g2d.fillPolygon(rect1);  // Triangle with lines  g2d.drawLine(140, 102, 175, 89);  g2d.drawLine(175, 89, 155, 130);  g2d.drawLine(155, 130, 140, 102);  // Eye1  GradientPaint gp3 = new GradientPaint(10, 100, Color.*BLACK*, 20, 2, Color.*BLACK*, true);  g2d.setPaint(gp3);  int[] xRect3 = new int[]{82, 87, 82, 87};  int[] yRect3 = new int[]{78, 78, 83, 83};  Polygon rect3 = new Polygon(xRect3, yRect3, 4);  g2d.drawPolygon(rect3);  g2d.fillPolygon(rect3);  // Eye2  GradientPaint gp4 = new GradientPaint(10, 100, Color.*BLACK*, 20, 2, Color.*BLACK*, true);  g2d.setPaint(gp4);  int[] xRect4 = new int[]{77, 82, 77, 82};  int[] yRect4 = new int[]{108, 108, 113, 113};  Polygon rect4 = new Polygon(xRect4, yRect4, 4);  g2d.drawPolygon(rect4);  g2d.fillPolygon(rect4);  //Lines  g2d.setColor(Color.*BLACK*);  g2d.drawLine(30, 33, 62, 78);  g2d.drawLine(25, 130, 59, 117);  }  private void drawBorder(Graphics2D g2d) {  BasicStroke basicStroke = new BasicStroke(5, BasicStroke.*CAP\_ROUND*, BasicStroke.*JOIN\_BEVEL*);  g2d.setStroke(basicStroke);  GradientPaint gp = new GradientPaint(5, 20, Color.*BLUE*, 20, 2, Color.*YELLOW*, true);  g2d.setPaint(gp);  g2d.drawRect(10, 10, *maxWidth* - 20, *maxHeight* - 20);  }  public static void main(String[] args) {  JFrame jFrame = new JFrame("lab2");  jFrame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  jFrame.setSize(1200, 1000);  jFrame.setLocationRelativeTo(null);  jFrame.setResizable(false);  jFrame.add(new Lab2());  jFrame.setVisible(true);  Dimension size = jFrame.getSize();  Insets insets = jFrame.getInsets();  *maxWidth* = size.width - insets.left - insets.right - 1;  *maxHeight* = size.height - insets.top - insets.bottom - 1;  }  @Override  public void actionPerformed(ActionEvent actionEvent) {  if (tx >= *rectWidth* && ty <= 0) {  dx = -1;  dy = 0;  }  if (ty >= *rectHeight* && tx >= *rectWidth*) {  dx = 0;  dy = -1;  }  if (tx <= 0 && ty >= *rectHeight*) {  dx = 1;  dy = 0;  }  if (ty <= 0 && tx <= 0) {  dx = 0;  dy = 1;  }  ty+=dy;  tx+=dx;  repaint();  angle += 0.02;  }  } |