

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  студент III курсу  групи КП-83  Шаповалов Данило Дмитрович  (*прізвище, ім’я, по батькові*)  варіант № 21 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2021

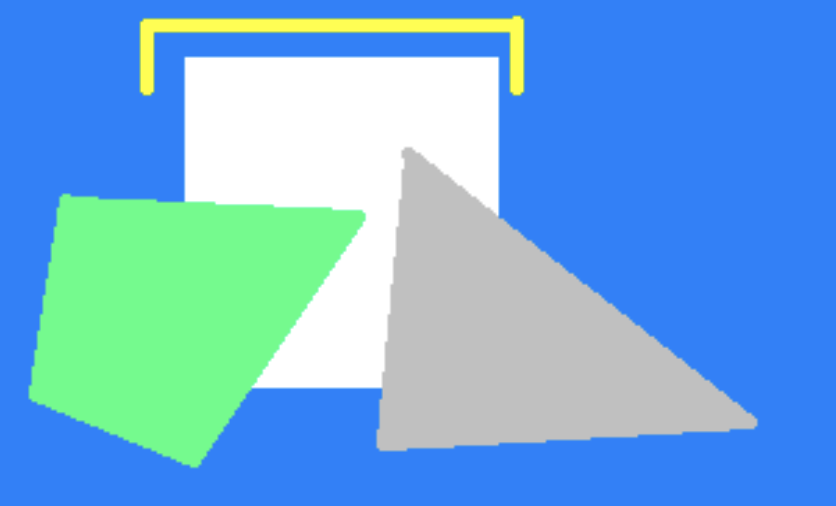
**Варіант завдання**

**Завдання**: За допомогою Java 2D намалювати картинку з лабораторної роботи No1 (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 No 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:**



**Лістинг коду програми**

package sample;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.shape.Polygon;

import javafx.scene.shape.Polyline;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.geom.Ellipse2D;

import javax.swing.\*;

public class Main extends JPanel implements ActionListener {

Timer timer;

private static int maxWidth = 1600;

private static int maxHeight = 900;

private double angle = 0;

private float alpha = 1.0f;

private float delta = 0.05f;

private final double center\_x = 1;

private final double center\_y = 1;

public Main() {

timer = new Timer(20, this);

timer.start();

}

public void paint(Graphics g) {

// Setting up the scene

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(new Color(51, 128, 246));

g2d.clearRect(0, 0, maxWidth, maxHeight);

// Border drawing

g2d.setColor(new Color(255, 254, 84));

BasicStroke bs1 = new BasicStroke(16, BasicStroke.CAP\_ROUND,

BasicStroke.JOIN\_ROUND);

g2d.setStroke(bs1);

g2d.drawRect(820, 20, 760, 830);

// static picture

BasicStroke bs2 = new BasicStroke(9, BasicStroke.CAP\_ROUND,

BasicStroke.JOIN\_ROUND);

g2d.setStroke(bs2);

g2d.drawPolyline(new int[]{61, 61, 217, 217}, new int[]{39, 17, 17, 39}, 4);

GradientPaint gp = new GradientPaint(5, 25, Color.WHITE, 20, 2, new Color(155, 221, 167), true);

g2d.setPaint(gp);

g2d.fillRect(69, 28, 139, 139);

g2d.setColor(new Color(117, 250, 142));

g2d.fillPolygon(new int[]{28, 150, 69, 14}, new int[]{94, 100, 200, 167}, 4);

g2d.setColor(new Color(192, 192, 192));

g2d.fillPolygon(new int[]{169, 317, 161}, new int[]{78, 183, 194}, 3);

// changing of angle

g2d.translate(3 \* maxWidth / 4, maxHeight / 2);

g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER, alpha));

g2d.rotate(angle, center\_x, center\_y);

// draw sub-picture

g2d.setColor(new Color(255, 254, 84));

g2d.drawPolyline(new int[]{61, 61, 217, 217}, new int[]{39, 17, 17, 39}, 4);

g2d.setPaint(gp);

g2d.fillRect(69, 28, 139, 139);

g2d.setColor(new Color(117, 250, 142));

g2d.fillPolygon(new int[]{28, 150, 69, 14}, new int[]{94, 100, 200, 167}, 4);

g2d.setColor(new Color(192, 192, 192));

g2d.fillPolygon(new int[]{169, 317, 161}, new int[]{78, 183, 194}, 3);

}

public void actionPerformed(ActionEvent e) {

if (alpha < 0.05f || alpha > 0.95f) {

delta = -delta;

}

angle -= 0.05;

alpha += delta;

repaint();

}

public static void main(String[] args) {

JFrame frame = new JFrame("Java2D");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(maxWidth, maxHeight);

frame.setLocationRelativeTo(null);

frame.setResizable(false);

frame.add(new Main());

frame.setVisible(true);

Dimension size = frame.getSize();

Insets insets = frame.getInsets();

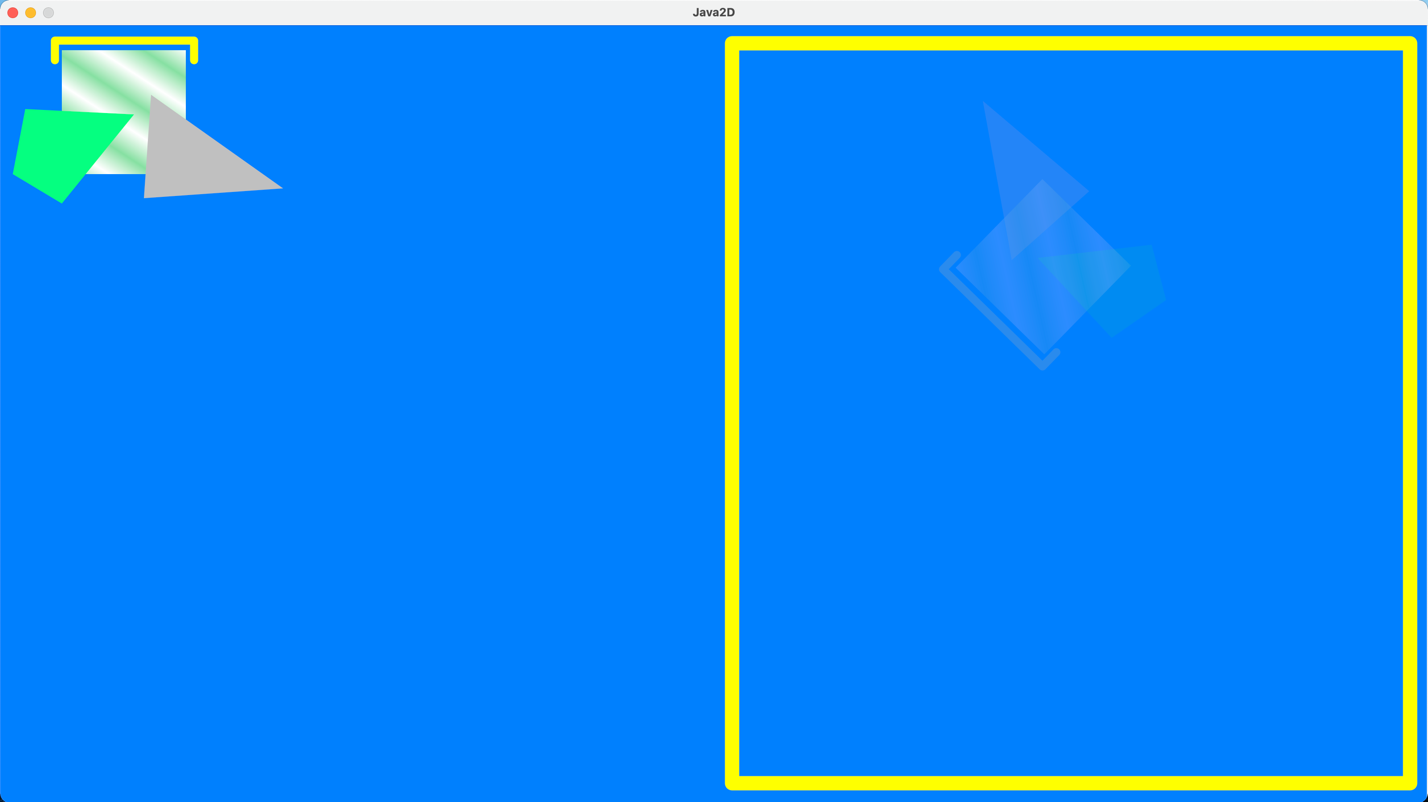
maxWidth = size.width - insets.left - insets.right - 1;

maxHeight = size.height - insets.top - insets.bottom - 1;

}

}

**Результат**

****