

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО
Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

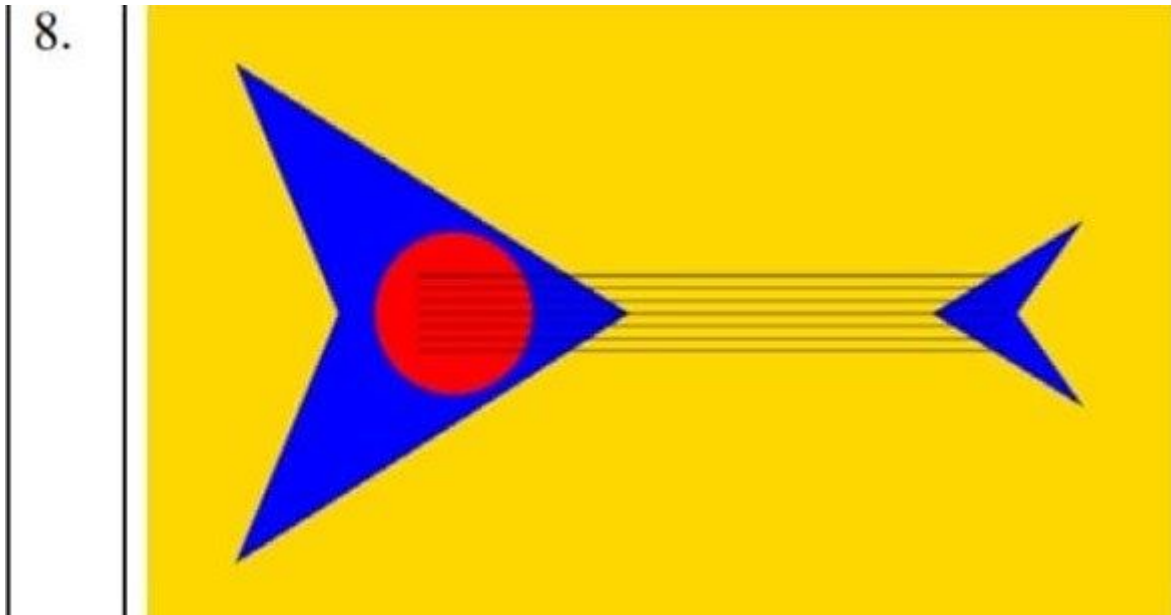
ЗВІТ
з лабораторної роботи № 2
Варіант 8

Виконав:
студент 3-го курсу,
групи КП-82,
спеціальності 121 – Інженерія
програмного забезпечення
Клапатюк Олександр Петрович

Тема: Побудова та анімація зображень за допомогою Java2D

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

(Картинка з Лабораторної 1)Варіант 8:



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.

Варіанти завдань

№	Типи анімації	Тип лінії рамки
1	1, 5	JOIN BEVEL
2	1, 10	JOIN MITER
3	1, 7	JOIN ROUND
4	3, 10	JOIN BEVEL
5	1, 9	JOIN MITER
6	7, 9	JOIN ROUND
7	3, 5	JOIN BEVEL
8	9, 10	JOIN MITER
9	3, 7	JOIN ROUND
10	7, 10	JOIN BEVEL
11	3, 9	JOIN MITER
12	1, 8	JOIN ROUND

Типи анімації:

1. Рух по колу проти годинникової стрілки
2. Рух по колу за годинниковою стрілкою
3. Рух по квадрату проти годинникової стрілки
4. Рух по квадрату за годинниковою стрілкою
5. Обертання навколо центру малюнка за годинниковою стрілкою
6. Обертання навколо центру малюнка проти годинникової стрілки
7. Обертання навколо кута малюнка за годинниковою стрілкою
8. Обертання навколо кута малюнка проти годинникової стрілки
9. Зміна прозорості
10. Масштабування

lab2.java (основний файл програми)

```
package lab2;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import java.awt.geom.Line2D;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

import javafx.scene.shape.Line;
import javafx.scene.shape.Polygon;

public class lab2 extends JPanel implements ActionListener {
    double BigPoligon[][] = {
        { 62/2, 44/2 }, { 136/2, 223/2 }, { 62/2, 401/2 }, { 342/2, 223/2 }
    };

    double LittlePoligon[][] = {
        { 564/2, 223/2 }, { 670/2, 157/2 }, { 625/2, 223/2 }, { 670/2, 288/2 }
    };
}
```

```

double arrowPoligon[][] = {
{235+400, 174+300}, {330+400,55+300}, {424+400,174+300},
{377+400,174+300},{377+400,295+300},{283+400,294+300},{284+400,174+300}
};
public void crearRendering(Graphics2D g2d)
{
    // Встановлюємо параметри рендерингу
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);

    g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
    RenderingHints.VALUE_RENDER_QUALITY);
}

public void createField(Graphics2D g2d)
{
    g2d.setColor(new Color(255,255,255));
    g2d.fillRect(0, 0, 500, 350);
    g2d.setColor(new Color(200,200,200));
    g2d.fillRect(500, 0, 480, 350);
    g2d.setColor(new Color(130,130,130));
    g2d.fillRect(0, 350, 500, 350);
    g2d.setColor(new Color(255,130,130));
    g2d.fillRect(500, 350, 480, 350);
}

public void createGuitarStrings( Graphics2D g2d)
{
    g2d.setColor(new Color(0,0,0));
    g2d.draw(new Line2D.Double(192/2, 196/2, 616/2, 196/2));
    g2d.draw(new Line2D.Double(192/2, 205/2, 616/2, 205/2));
    g2d.draw(new Line2D.Double(192/2, 214/2, 616/2, 214/2));
    g2d.draw(new Line2D.Double(192/2, 223/2, 616/2, 223/2));
    g2d.draw(new Line2D.Double(192/2, 232/2, 616/2, 232/2));
    g2d.draw(new Line2D.Double(192/2, 241/2, 616/2, 241/2));
    g2d.draw(new Line2D.Double(192/2, 250/2, 616/2, 250/2));
    g2d.draw(new Line2D.Double(192/2, 197/2, 616/2, 197/2));
    g2d.setColor(new Color(235,235,235));
    g2d.draw(new Line2D.Double(192/2, 206/2, 616/2, 206/2));
    g2d.draw(new Line2D.Double(192/2, 215/2, 616/2, 215/2));
    g2d.draw(new Line2D.Double(192/2, 224/2, 616/2, 224/2));
    g2d.draw(new Line2D.Double(192/2, 233/2, 616/2, 233/2));
    g2d.draw(new Line2D.Double(192/2, 242/2, 616/2, 242/2));
    g2d.draw(new Line2D.Double(192/2, 251/2, 616/2, 251/2));
}

public GeneralPath createPoligon(double arr[][] )
{
    GeneralPath poligon = new GeneralPath();
    poligon.moveTo(arr[0][0], arr[0][1]);
    for (int k = 1; k < arr.length; k++)
        poligon.lineTo(arr[k][0], arr[k][1]);
    poligon.closePath();
    return poligon;
}

Timer timer;
int diamentroOfAnimanitionCircle = 40;

// Для анімації масштабування
private double scale = 1;
private double delta = 0.01;

// Для анімації руху
private double dx = 1;
private double dy = 1;

//границі анімованої рамки
int minX =507;
int minY =5+5;
int maxX =918+50;
int maxY =289+56;

//coordinates of moveble circle
int yAnimation= (int)( (Math.random() * ((maxY - minY) + 1)) + minY);

```

```

int xAnimation= (int)( (Math.random() * ((maxX - minX) + 1)) + minX);

private static int maxWidth;
private static int maxHeight;
public lab2() {
    // Таймер генеруватиме подію що 10 мс
    timer = new Timer(20, this);
    timer.start();
}
public void paint(Graphics g) {
    super.paint(g);
    Graphics2D g2d = (Graphics2D)g;
    createField(g2d);
    crearRendering(g2d);
    GeneralPath guitarDeck = createPoligon(BigPoligon);
    GeneralPath headstock = createPoligon(LittlePoligon);
    GeneralPath arrow = createPoligon(arrowPoligon);
    g2d.setColor(new Color(0,0,255));
    g2d.setColor(new Color(0,0,255));
    g2d.setStroke(new BasicStroke(0.01f));
    g2d.fill(guitarDeck);
    g2d.fill(headstock);
    g2d.setColor(new Color(255,0,0));
    g2d.fillOval((80), (83), diamentrOfAnimanitionCircle, diamentrOfAnimanitionCircle);
    GradientPaint blueToBlack = new GradientPaint(330+400,55+300, Color.BLUE,
    283+400,234+300, Color.white);
    g2d.setPaint(blueToBlack);
    g2d.fill(arrow);
    createGuitarStrings(g2d);
    g2d.setColor(new Color(255,255,0));
    BasicStroke bs1 = new BasicStroke(10, BasicStroke.CAP_ROUND,
    BasicStroke.JOIN_MITER);
    g2d.setStroke(bs1);
    g2d.drawRect(500, 5, 475, 345);
    g2d.setStroke(new BasicStroke(2f));
    g2d.setColor(Color.green);
    g2d.draw(new Line2D.Double(507, 6, 507, 6));
    g2d.setStroke(new BasicStroke(2f));
    g2d.draw(new Line2D.Double(973, 6, 973, 6));
    g2d.setStroke(new BasicStroke(2f));
    g2d.draw(new Line2D.Double(973, 329, 973, 329));
    g2d.draw(new Line2D.Double(507, 329, 507, 329));
    g2d.drawRect(175, 400, 100, 100);

    // Перетворення для анімації зміни прозорості
    g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER,
    (float)scale));
    g2d.fillOval(xAnimation, yAnimation, (int)(diamentrOfAnimanitionCircle*scale),
    (int)(diamentrOfAnimanitionCircle*scale));
}
public static void main(String[] args) {
    JFrame frame = new JFrame("П р и к л а д а н і м а ц і ї");
    frame.add(new lab2());
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1000, 700);
    frame.setResizable(false);
    frame.setLocationRelativeTo(null);
    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;
}

// Цей метод буде викликано щоразу, як спрацює таймер
public void actionPerformed(ActionEvent e) {

    if ( scale < 0.01) {
        delta = -delta;
    } else if (scale > 0.99) {
        delta = -delta;
    }

    if(xAnimation+(diamentrOfAnimanitionCircle*scale)>maxX)
    {

        xAnimation = (int)(maxX - diamentrOfAnimanitionCircle*scale -1);
        System.out.println("Hello world! xAnimation  maxX  " + xAnimation );
        dx=-dx;
    }

    else if(xAnimation<minX)

```

```

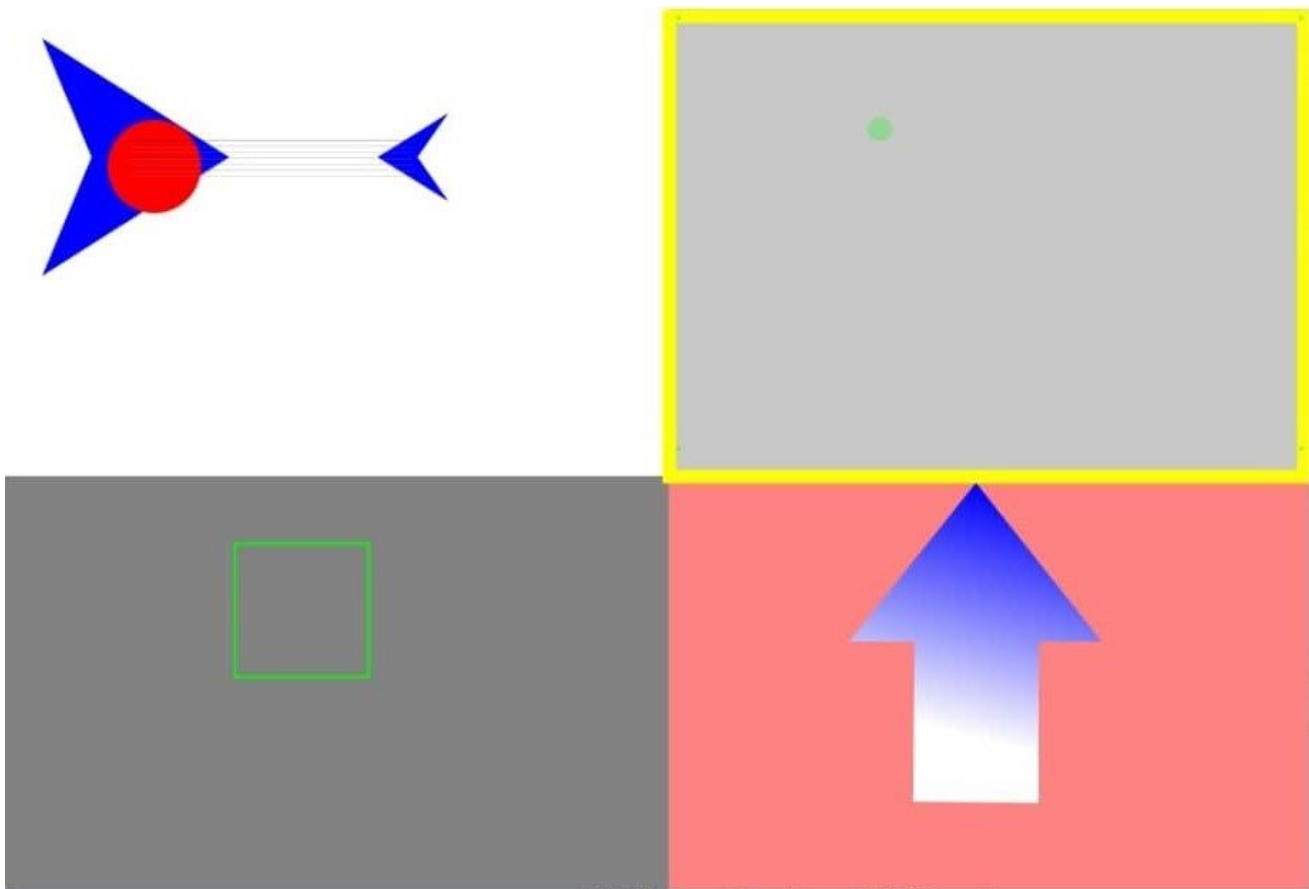
    {
        xAnimation = (int)(minX+1);
        System.out.println("Hello world! xAnimation  minX  " + xAnimation );
        dx=-dx;
    }
    if(yAnimation+(diamentrOfAnimanitionCircle*scale)>maxY)
    {
        yAnimation = (int)(maxY - diamentrOfAnimanitionCircle*scale -1);
        System.out.println("Hello world! yAnimation  maxY  " + yAnimation );

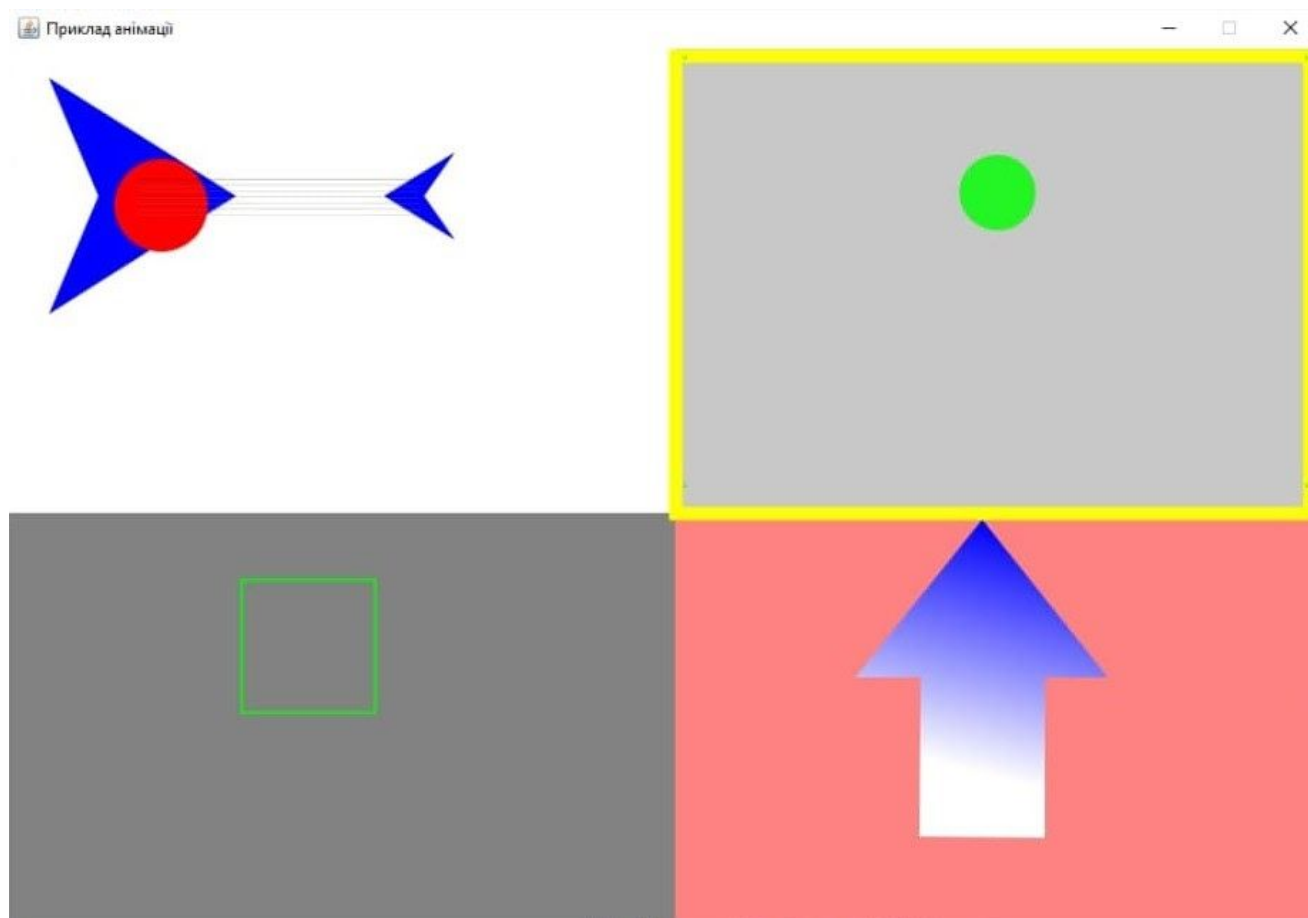
        dy=-dy;
    }
    else if(yAnimation<minY)
    {
        yAnimation = (int)(minY+1);
        System.out.println("Hello world! yAnimation  minY  " + yAnimation );
        dy=-dy;
    }

    scale += delta;
    xAnimation += dx;
    yAnimation += dy;
    repaint();
}
}

```

Приклад роботи програми :





Висновок: Ознайомився з можливостями побудови зображень та їх анімації у Java2D.