



МІНІСТЕРСТВО ОСВІТИ 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”

Виконала
студентка III курсу
групи КП-83

Снітко Маріанна Дмитрівна
(*прізвище, ім'я, по батькові*)
Варіант № 19

Зарахована
“ ____ ” “ ____ ” 2021 р.
викладачем
Шкурат О. С.
(*прізвище, ім'я, по батькові*)

Варіант завдання № 19

Тип лінії рамки: JOIN_BEVEL

Типи анімації: 2. Рух по колу за годинниковою стрілкою

10. Масштабування

Лістинг коду програми

Lab2.java

```
package lab2;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

public class Lab2 extends JPanel implements ActionListener {
    // Scaling is done relatively to the center of coordinates,
    // so it is better to draw a figure also symmetrically to the center
    double points[][] = {
        { -100, -15 }, { -25, -25 }, { 0, -90 }, { 25, -25 },
        { 100, -15 }, { 50, 25 }, { 60, 100 }, { 0, 50 },
        { -60, 100 }, { -50, 25 }, { -100, -15 }
    };

    double width = 350;
    double height = 210;

    Timer timer;
    // Rotate animation
    private double angle = 0;

    // Scaling animation
    private double scale = 1;
    private double delta = 0.01;

    // Moving animation
    private double dx = 1;
    private double tx = height/2;
    private double dy = 1;
    private double ty = 0;
    private static int maxWidth;
    private static int maxHeight;

    public Lab2() {
        // Timer generates event every 10 ms
        timer = new Timer(10, this);
        timer.start();
    }

    public void paint(Graphics g){
        super.paint(g);

        Graphics2D g2d = (Graphics2D)g;
```

```

Stroke defaultStroke = g2d.getStroke();
g2d.setBackground(Color.ORANGE);
g2d.clearRect(0, 0, maxWidth+1, maxHeight+1);

g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
    RenderingHints.VALUE_RENDER_QUALITY);

this.drawBorder(g2d);

GeneralPath bab = new GeneralPath();
double points2[][] = {
    {width/7*2-width/2, height/4.2*0.6-height/2},
    {width/7*2.8-width/2, height/4.2*1.5-height/2},
    {width/7*5-width/2, height/4.2*1.5-height/2},
    {width/7*5-width/2, height/4.2*2.3-height/2},
    {width/7*2-width/2, height/4.2*2.3-height/2} };
bab.moveTo(points2[0][0], points2[0][1]);
for (int k = 1; k < points2.length; k++)
    bab.lineTo(points2[k][0], points2[k][1]);
bab.closePath();

g2d.translate(maxWidth/2, maxHeight/2);
g2d.translate(tx,ty);
g2d.scale(scale, scale);

g2d.setStroke(defaultStroke);

GradientPaint gp = new GradientPaint(
    0, 0, new Color(0, 144, 255),
    50, 50, new Color(0.0F, 1.0F, 1.0F),
    true);
g2d.setPaint(gp);
g2d.fill(bab);

g2d.setColor(new Color(117,77,0));
g2d.drawLine((int)(width/7*2.6-width/2), (int)(height/4.2*2.3-height/2),
(int)(width/7*4-width/2), (int)(height/4.2*3.2-height/2));
g2d.drawLine((int)(width/7*4.4-width/2), (int)(height/4.2*2.3-height/2),
(int)(width/7*3-width/2), (int)(height/4.2*3.2-height/2));

g2d.setColor(new Color(155,102,0));
g2d.drawLine((int)(width/7*5-width/2), (int)(height/4.2*1.5-height/2),
(int)(width/7*5.65-width/2), (int)(height/4.2*0.8-height/2));

g2d.setColor(new Color(0.5411765F, 0.16862746F, 0.8862745F));

g2d.fillOval((int)(width/7*2.6-width/2)-(int)(width/7*0.44), (int)(height/4.2*3.4-height/2)-(int)(width/7*0.44), (int)(width/7*0.44)*2, (int)(width/7*0.44)*2);

g2d.fillOval((int)(width/7*4.4-width/2)-(int)(width/7*0.44), (int)(height/4.2*3.4-height/2)-(int)(width/7*0.44), (int)(width/7*0.44)*2, (int)(width/7*0.44)*2);

}

private void drawBorder(Graphics2D g2d) {
    g2d.setColor(Color.BLACK);
    BasicStroke basicStroke = new BasicStroke(5, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_BEVEL);
    g2d.setStroke(basicStroke);
    g2d.drawRect(5, 5, maxWidth - 10, maxHeight - 10);
}

public static void main(String[] args) {
    JFrame frame = new JFrame("Lab 2 Variant 19");
    frame.add(new Lab2());
}

```

```

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 500);
        frame.setResizable(true);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);

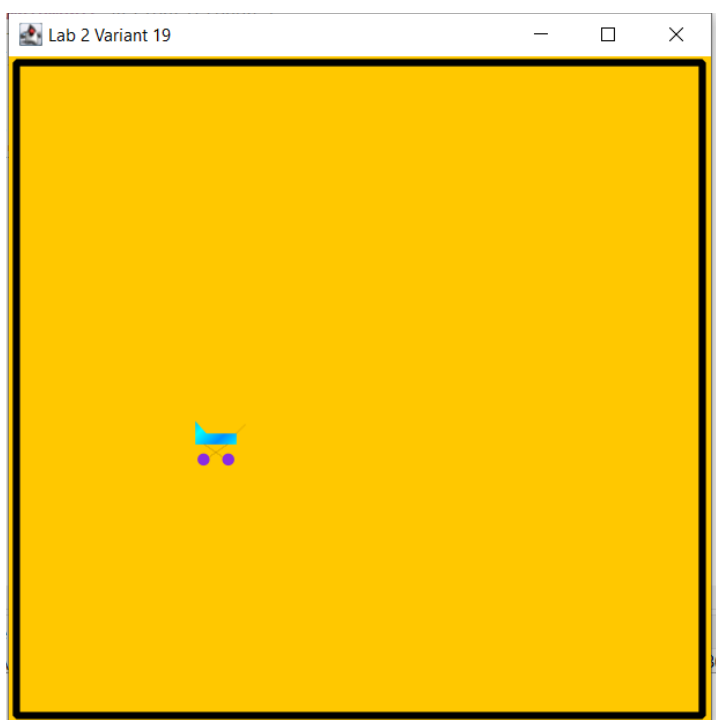
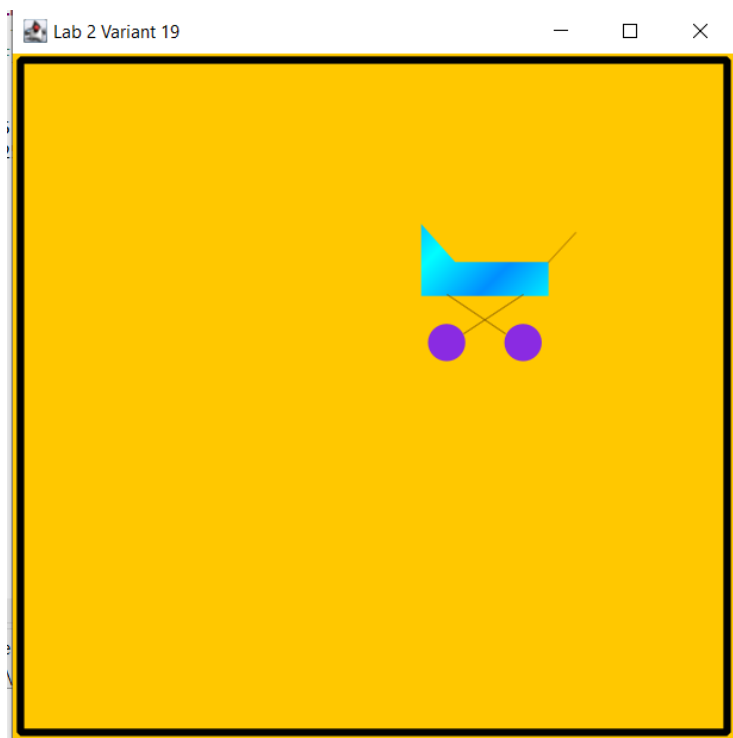
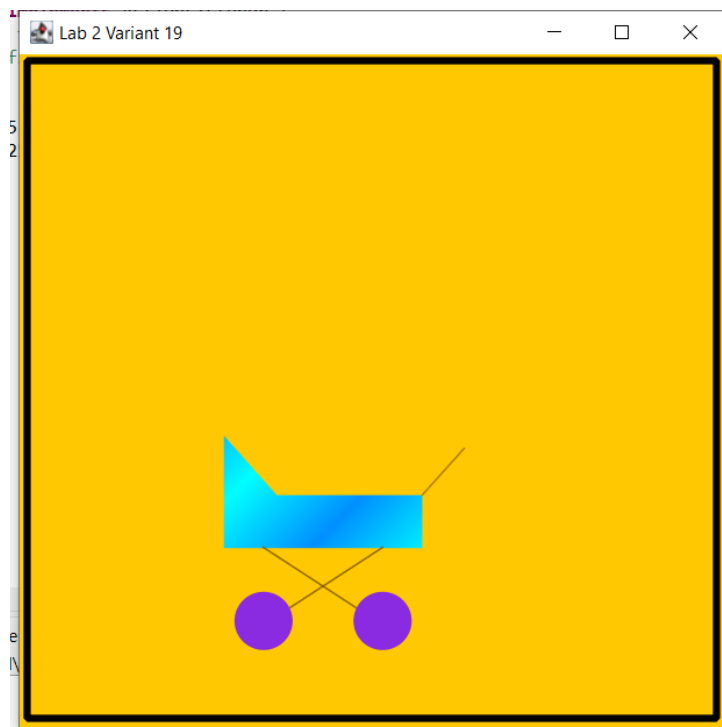
        Dimension size = frame.getSize();
        Insets insets = frame.getInsets();
        maxWidth = size.width - insets.left - insets.right - 1;
        maxHeight = size.height - insets.top - insets.bottom - 1;
    }
    // This method executes every time the timer sets off
    public void actionPerformed(ActionEvent e) {
        if ( scale < 0.01 ) {
            delta = -delta;
        } else if (scale > 0.99) {
            delta = -delta;
        }

        scale += delta;
        angle += 0.01;
        ty = height/2*Math.sin(angle);
        tx = height/2*Math.cos(angle);

        repaint();
    }
}

```

Результати роботи програми



Висновки

Виконавши дану лабораторну роботу, я ознайомилася з можливостями побудови зображень та їх анімації у Java2D

Система була написана на мові програмування **Java**.