

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”
Варіант № 4

Виконав:

студент 3-го курсу, групи КП-83,
Дереворіз Назар

Перевірив:

викладач
Шкурат Оксана Сергіївна

Київ - 2021

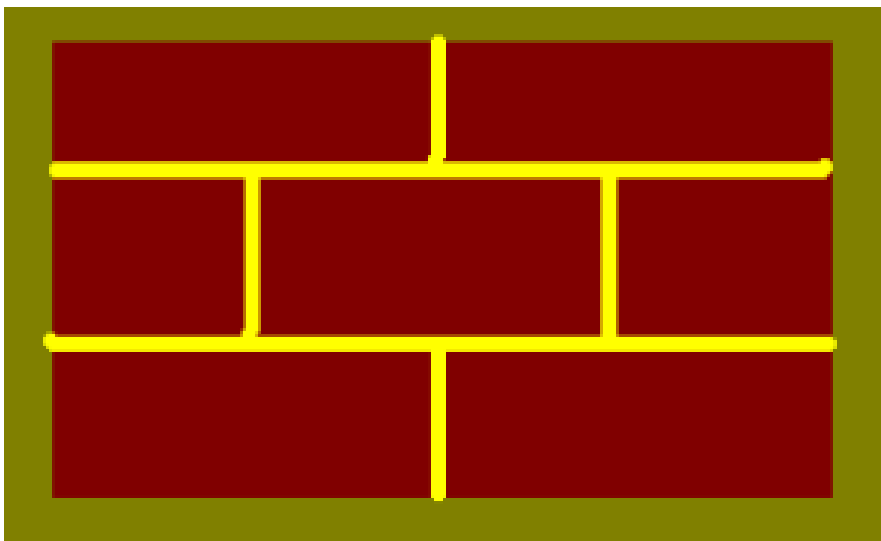
Завдання: За допомогою Java 2D намалювати картинку з лабораторної роботи №1 (за варіантом). Додатково виконати:

1. Хоча б 1 стандартний примітив, та хоча б 1 фігуру, побудовану по точкам (ламаную).
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

4	3, 10	JOIN_BEVEL
---	-------	------------

3. Рух по квадрату проти годинникової стрілки

10. Масштабування



Код програми:

Skeleton.java

```
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

```
import java.awt.geom.GeneralPath;
import javax.swing.*;

@SuppressWarnings("serial")
public class Skeleton extends JPanel implements
ActionListener {

    private static int maxWidth;
    private static int maxHeight;

    Timer timer;

    private double scale = 0.75;
    private double delta = 0.005;

    private int horizontal_gap = 50;
    private int vertical_gap = horizontal_gap * 2;
    private int rectWidth;
    private int rectHeight;

    private double dx = 0;
    private int tx = horizontal_gap + 17;
    private double dy = 1;
    private int ty = -130 + 17;
    private boolean clockwise = false;

    public Skeleton() {
        timer = new Timer(10, this);
        timer.start();
    }

    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;

        rectWidth = getWidth()/2 - horizontal_gap*2;
        rectHeight = maxHeight - vertical_gap*2;

        g2d.setBackground(new Color(128,128,0));
        g2d.clearRect(0,0,maxWidth,maxHeight);

        Color Maroon = new Color(128,0,0);

        GradientPaint gp = new
```

```
GradientPaint(-rectWidth - horizontal_gap,  
-rectHeight/2,
```

```
Maroon.brighter().brighter().brighter(),  
-horizontal_gap, rectHeight/2,  
Maroon.darker().darker().darker());  
g2d.setPaint(gp);
```

```
double points[][] = {  
    {-rectWidth - horizontal_gap,  
rectHeight/2}, {-rectWidth -  
horizontal_gap,-rectHeight/2},  
    {- horizontal_gap, -rectHeight/2},{-  
horizontal_gap,rectHeight/2}  
};
```

```
GeneralPath wall = new GeneralPath();  
wall.moveTo(points[0][0], points[0][1]);  
for (int k = 1; k < points.length; k++)  
    wall.lineTo(points[k][0], points[k][1]);  
wall.closePath();
```

```
g2d.translate((int)maxWidth/2, (int)maxHeight/2);  
g2d.fill(wall);
```

```
g2d.setColor(Color.yellow);  
g2d.setStroke(new BasicStroke(6f));  
// Horizontal 1  
g2d.drawLine(-rectWidth - horizontal_gap +  
3,-rectHeight/6,-horizontal_gap - 3,  
-rectHeight/6);  
// Horizontal 2  
g2d.drawLine(-rectWidth - horizontal_gap +  
3,rectHeight/6,-horizontal_gap - 3,rectHeight/6);  
// Vertical 1  
g2d.drawLine(-rectWidth/2 -  
horizontal_gap,-rectHeight/2 + 3,-rectWidth/2 -  
horizontal_gap,-rectHeight/6 - 3);  
// Vertical 2  
g2d.drawLine(-rectWidth/2 -  
horizontal_gap,rectHeight/6 + 3,-rectWidth/2 -  
horizontal_gap,rectHeight/2 - 3);  
// Vertical 3
```

```

        g2d.drawLine(-rectWidth/4 -
horizontal_gap,-rectHeight/6 + 3,-rectWidth/4 -
horizontal_gap,rectHeight/6 - 3);
        // Vertical 4
        g2d.drawLine(-3*rectWidth/4 -
horizontal_gap,-rectHeight/6 + 3,-3*rectWidth/4 -
horizontal_gap,rectHeight/6 - 3);

```

```

        g2d.setColor(Color.WHITE);
        BasicStroke strokeStyle = new BasicStroke(16,
BasicStroke.CAP_SQUARE,
        BasicStroke.JOIN_BEVEL);
        g2d.setStroke(strokeStyle);

```

```

g2d.drawRect(horizontal_gap,-rectHeight/2,rectWidth,
rectHeight);

```

```

        g2d.setColor(Color.BLACK);
        g2d.setStroke(new BasicStroke(2f));

```

```

g2d.drawRect(tx,ty,(int)(30*scale),(int)(30*scale));
    }

```

```

    public void actionPerformed(ActionEvent e) {

```

```

        if (clockwise == true) {
            if ( tx > horizontal_gap + rectWidth -
40) {
                dx = 0;
                dy = -1;
                clockwise = false;
            } else if ( tx < horizontal_gap + 10) {
                dx = 0;
                dy = 1;
                clockwise = false;
            }
        } else {
            if ( ty > rectHeight/2 - 40) {
                dy = 0;
                dx = 1;
                clockwise = true;
            } else if ( ty < -rectHeight/2 + 10) {

```

```
        dy = 0;
        dx = -1;
        clockwise = true;
    }
}
```

```
    if (scale < 0.25 || scale > 1) {
        delta = -delta;
    }
    scale += delta;
```

```
    tx += dx;
    ty += dy;
```

```
    repaint();
}
```

```
public static void main(String[] args) {
    JFrame frame = new JFrame("Привіт, Java
2D!");
```

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
;
```

```
    frame.setSize(1000, 500);
    frame.setLocationRelativeTo(null);
    frame.setResizable(false);
    frame.add(new Skeleton());
    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 роботи програми:

