МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №7

Специальность ПО-5(о)

Выполнил

Д. С. Бриштен,

студент группы ПО-5

Проверил

А. А. Крощенко

ст. преп. каф. ИИТ

Брест 2021

Цель работы: освоить возможности языка программирования Java в построении графических приложений.

Вариант 4

Задание 1. Изобразить разносторонний треугольник, вращающийся в плоскости апплета вокруг своего центра тяжести.

Выполнение:

Код программы

import java.awt.Color;

import java.awt.Dimension;

import java.awt.EventQueue;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Point;

import java.awt.geom.AffineTransform;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.UIManager;

import javax.swing.UnsupportedLookAndFeelException;

public class Figure extends JFrame {

    private int[] xPoints = new int[]{0, 50, 110};

    private int[] yPoints = new int[]{90, 0, 100};

    double angle = 0f;

    public Figure() {

        super("Triangle rotation");

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setContentPane(new DrawTrianglePanel());

        pack();

        setLocationRelativeTo(null);

        setVisible(true);

    }

    private class DrawTrianglePanel extends JPanel {

        private Point clickPoint;

        private long t = System.nanoTime();

        public DrawTrianglePanel() {

            clickPoint = new Point(100, 100);

        }

        @Override

        public Dimension getPreferredSize() {

            return new Dimension(250, 250);

        }

        protected Dimension getTriangleSize() {

            int maxX = 0;

            int maxY = 0;

            for (int index = 0; index < xPoints.length; index++) {

                maxX = Math.max(maxX, xPoints[index]);

            }

            for (int index = 0; index < yPoints.length; index++) {

                maxY = Math.max(maxY, yPoints[index]);

            }

            return new Dimension(maxX, maxY);

        }

        /\*\*

         \* Drawing the triangle.

         \*

         \* @param g the g

         \* @see javax.swing.JComponent#paintComponent(java.awt.Graphics)

         \*/

        @Override

        protected void paintComponent(Graphics g) {

            super.paintComponent(g);

            Graphics2D g2d = (Graphics2D) g.create();

            AffineTransform at = new AffineTransform();

            Dimension size = getTriangleSize();

            int x = clickPoint.x - (size.width / 2);

            int y = clickPoint.y - (size.height / 2);

            at.translate(x, y);

            at.rotate(Math.toRadians(angle), clickPoint.x - x, clickPoint.y - y);

            g2d.setTransform(at);

            g2d.drawPolygon(xPoints, yPoints, 3);

            g2d.setColor(Color.RED);

            g2d.drawLine(size.width / 2 - 5, 0, size.width / 2, size.height / 2);

            g2d.dispose();

            long tm = System.nanoTime() - t;

            double Angle = tm / 300000000.0;

            double sn = Math.cos(Angle);

            double cs = Math.sin(Angle);

            int x1 = (int) (75 \* cs + 130);

            int y1 = (int) (75 \* sn + 120);

            angle = -Math.toDegrees(Math.atan2(x1 - clickPoint.x, y1 - clickPoint.x)) + 180;

            repaint();

        }

    }

        public static void main(String[] args) {

            EventQueue.invokeLater(new Runnable() {

            @Override

            public void run() {

                try {

                    UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

                } catch (ClassNotFoundException | InstantiationException | IllegalAccessException | UnsupportedLookAndFeelException ex) {

                    ex.printStackTrace();

                }

                new Figure();

            }

            });

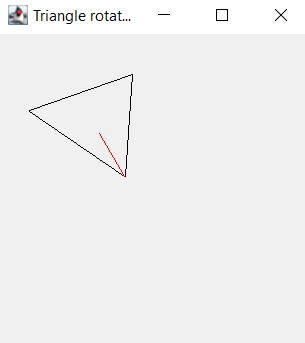
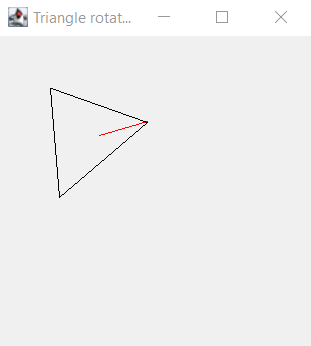
        }

}

Спецификация вывода:

Панель с разносторонним треугольником.

Пример:

Задание 2. Ковер Серпинского.

Выполнение:

Код программы

import java.awt.Color;

import java.awt.Graphics;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingUtilities;

import javax.swing.WindowConstants;

public class Carpet {

    public static Color BACKGROUNDCOLOR = new Color(0, 0, 150);

    public static Color FOREGROUNDCOLOR = new Color(255, 180, 0);

    public static int DEEP = 5;

    public Carpet(int deep) {

        JFrame frame = new JFrame();

        frame.setTitle("...: Recursive Squares with deep " + deep + " :...");

        frame.setDefaultCloseOperation(WindowConstants.DISPOSE\_ON\_CLOSE);

        frame.setSize(400, 400);

        frame.setLocationRelativeTo(null);

        frame.add(new PrintArea(deep));

        frame.setVisible(true);

    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable() {

            public void run() {

                for (int i = DEEP; i >= 1; --i) {

                    new Carpet(i);

                }

            }

        });

    }

}

class PrintArea extends JLabel {

    private static final long serialVersionUID = 1L;

    int deep;

    public PrintArea(int deep) {

        super();

        setBackground(Carpet.BACKGROUNDCOLOR);

        setOpaque(true);

        this.deep = deep;

    }

    public void paint(Graphics g) {

        super.paint(g);

        g.setColor(Carpet.FOREGROUNDCOLOR);

        int n = printSquares(g, 0, 0, getWidth(), getHeight(), this.deep);

        System.out.println("Deep = " + deep + ", squares painted: " + n);

    }

    private int printSquares(Graphics g, int xi, int yi, int width, int height, int currentDeep) {

        int newWidth = width / 3;

        int newHeight = height / 3;

        int x = (width / 3) + xi;

        int y = (height / 3) + yi;

        g.fillRect(x, y, newWidth, newHeight);

        int sX = 0;

        int sY = 0;

        if (currentDeep > 1) {

            int sum = 0;

            for (int i = 0; i < 3; i++) {

                for (int j = 0; j < 3; j++) {

                    sX = i \* (width / 3) + xi;

                    sY = j \* (height / 3) + yi;

                    sum += printSquares(g, sX, sY, newWidth, newHeight, currentDeep - 1);

                }

            }

            return 1 + sum;

        } else

            return 1;

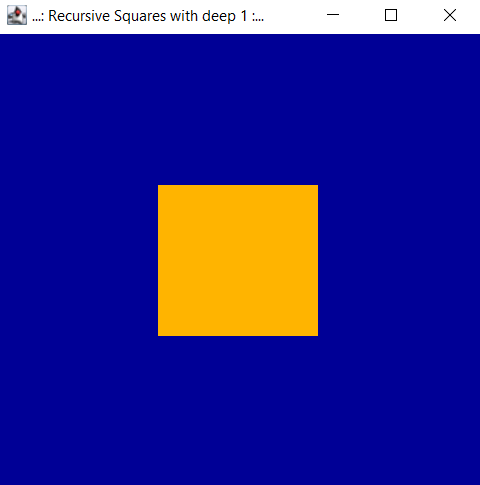
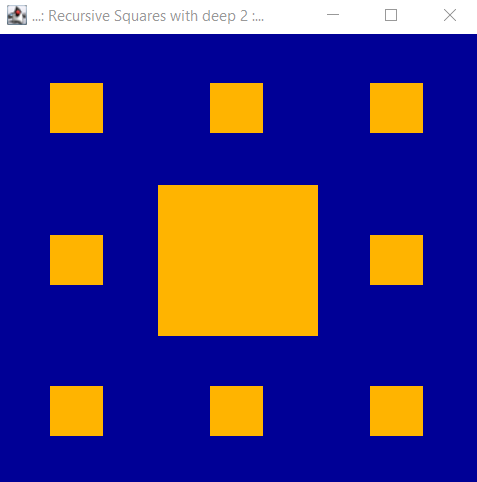
    }

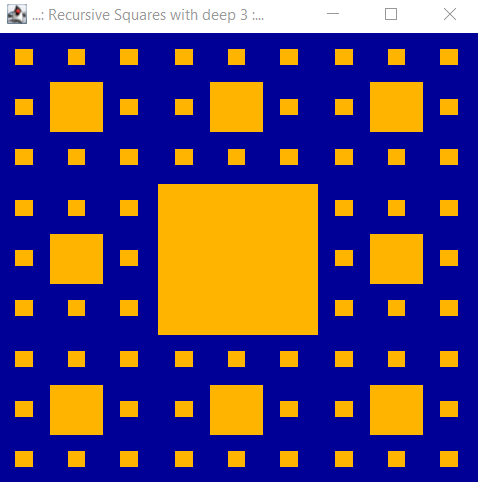
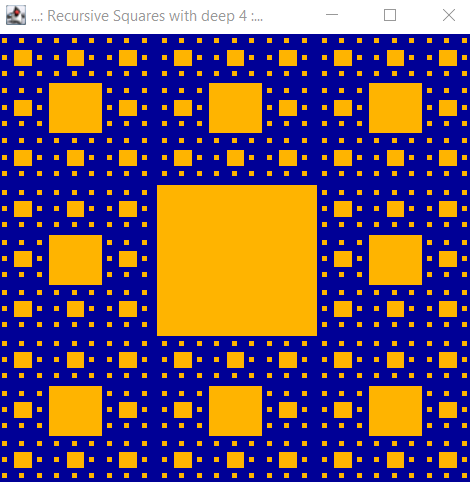
}

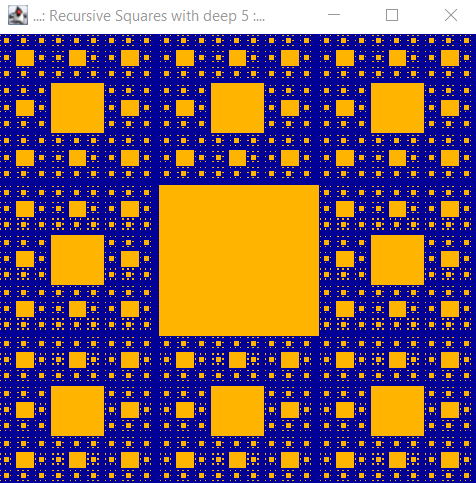
Спецификация вывода:

5 панелей с растущей глубиной.

Пример:



Вывод: Ознакомился с графическими примитивами при решении практических задач в языке программирования Java.