**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий Кафедра Информатики и информационных технологий***

# направление подготовки

**09.03.02 «Информационные системы и технологии»**

ЛАБОРАТОРНАЯ РАБОТА № 9

**Дисциплина:** Технология кроссплатформенного программирования

**Тема:** Создание меню, графика в языке Java

# Выполнил(а): студент(ка) группы

Хайруллина Алина Ринатовна

(Фамилия И.О.)

**Дата, подпись**

(Дата) (Подпись)

**Проверил:**

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись**

(Дата) (Подпись)

# Замечания:

**Москва 2023**

# ЛАБОРАТОРНАЯ РАБОТА № 9

Задачи:

1. Обязательная задача для всех:

Переделать пример с рисованием фигур так, чтобы фигуры рисовались выбором соответствующего пункта меню.

1. Индивидуальное задание.

Нарисовать свою фамилию линиями разной толщины, цвета и стиля (и вообще – проявить фантазию).

Листинг

package com.example;

import java.awt.BasicStroke;

import java.awt.Color;

import java.awt.GradientPaint;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.RenderingHints;

import java.awt.geom.Ellipse2D;

import javax.swing.JPanel;

public class JMyPanel extends JPanel {

  public static enum Figure {

    LINE,

    OVAL,

    RECT,

    ROUNDRECT,

    CLEAR,

  }

  private Figure vibor = Figure.CLEAR;

  public JMyPanel() {}

  public void ris(String *s*) {

    vibor = Figure.valueOf(*s*);

    repaint();

  }

  public void paintComponent(Graphics *gr*) {

    super.paintComponent(*gr*);

    Graphics2D g = (Graphics2D) *gr*;

    BasicStroke pen;

    g.setRenderingHint(

      RenderingHints.KEY\_ANTIALIASING,

      RenderingHints.VALUE\_ANTIALIAS\_ON

    );

    switch (vibor) {

      case LINE:

        pen =

          new BasicStroke(20, BasicStroke.CAP\_ROUND, BasicStroke.JOIN\_ROUND);

        g.setStroke(pen);

        g.setColor(Color.blue);

        g.drawLine(20, 20, 100, 100);

        break;

      case OVAL:

        float[] dash = { 10, 30 };

        pen =

          new BasicStroke(

            10,

            BasicStroke.CAP\_SQUARE,

            BasicStroke.JOIN\_ROUND,

            10,

            dash,

            0

          );

        g.setStroke(pen);

        g.setColor(Color.red);

        g.setPaint(

          new GradientPaint(30, 30, Color.red, 50, 50, Color.green, true)

        );

        g.fill(new Ellipse2D.Double(20, 20, 100, 100));

        break;

      case RECT:

        float[] dash2 = { 20, 20 };

        pen =

          new BasicStroke(

            5,

            BasicStroke.CAP\_SQUARE,

            BasicStroke.JOIN\_BEVEL,

            1,

            dash2,

            0

          );

        g.setStroke(pen);

        g.setColor(Color.magenta);

        g.drawRect(20, 20, 100, 100);

        break;

      case ROUNDRECT:

        float[] dash3 = { 20, 20, 2, 20, 2, 20 };

        pen =

          new BasicStroke(

            10,

            BasicStroke.CAP\_ROUND,

            BasicStroke.JOIN\_BEVEL,

            1,

            dash3,

            0

          );

        g.setStroke(pen);

        g.setColor(Color.yellow);

        g.drawRoundRect(20, 20, 100, 100, 60, 60);

        break;

      case CLEAR:

        g.clearRect(0, 0, getSize().width, getSize().height);

        break;

    }

  }

}

package com.example;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

public class Task1 extends JFrame implements ActionListener {

  public static void main(String[] *args*) {

    new Task1("Окно с графикой");

  }

  private JMyPanel myPanel = new JMyPanel();

  public Task1(String *s*) {

    super(*s*);

    JMenuBar menu = new JMenuBar();

    JMenu select = new JMenu("Choice");

    JMenuItem clear = new JMenuItem("CLEAR");

    clear.addActionListener(this);

    JMenuItem[] figs = new JMenuItem[4];

    for (int i = 0; i < 4; i++) {

      figs[i] = new JMenuItem(JMyPanel.Figure.values()[i].toString());

      select.add(figs[i]);

      figs[i].addActionListener(this);

    }

    setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

    add(myPanel, BorderLayout.CENTER);

    Dimension size = getSize();

    size.setSize(size.width, size.height + 200);

    select.add(clear);

    menu.add(select);

    setJMenuBar(menu);

    setMinimumSize(size);

    pack();

    setVisible(true);

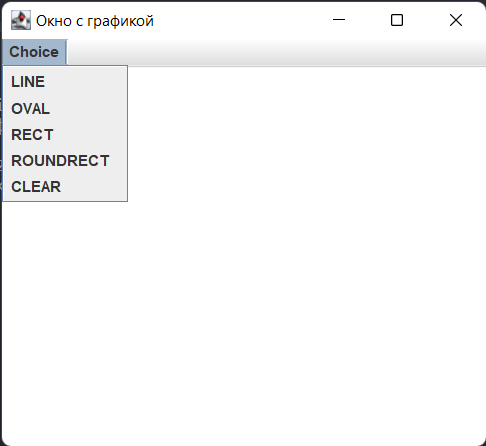
  }

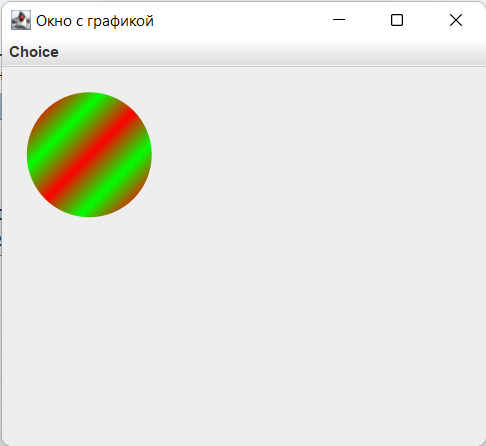
  public void actionPerformed(ActionEvent *e*) {

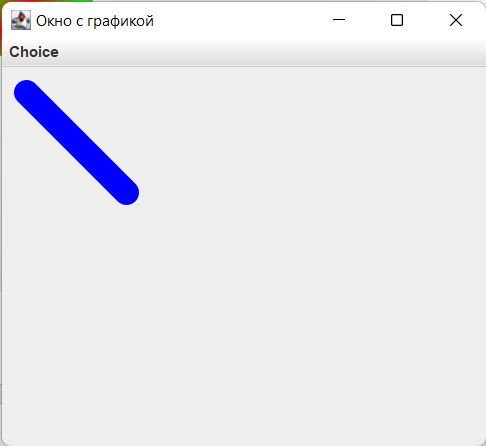
    myPanel.ris(*e*.getActionCommand());

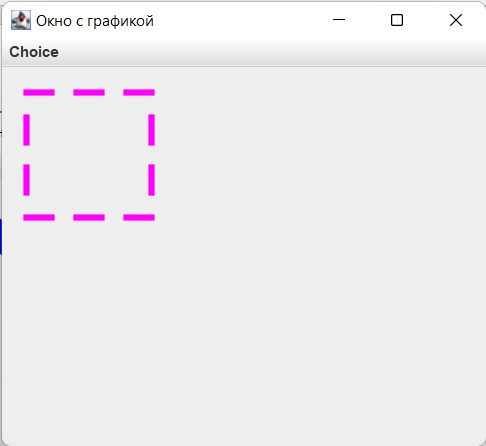
  }

}









import java.awt.BasicStroke;

import java.awt.Color;

import java.awt.GradientPaint;

import java.awt.Graphics;

import java.awt.Graphics2D;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.SwingUtilities;

public class DrawWord extends JPanel {

  public static void main(String[] *args*) {

    SwingUtilities.invokeLater(() -> {

      JFrame frame = new JFrame("DrawWord");

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

      DrawWord creativeWordDrawing = new DrawWord();

      frame.add(creativeWordDrawing);

      frame.setSize(700, 200);

      frame.setVisible(true);

    });

  }

  @Override

  protected void paintComponent(Graphics *g*) {

    super.paintComponent(*g*);

    int startX = 20;

    int startY = 100;

    drawK(*g*, startX, startY);

    drawH(*g*, startX + 60, startY);

    drawA(*g*, startX + 120, startY);

    drawY(*g*, startX + 180, startY);

    drawR(*g*, startX + 240, startY);

    drawU(*g*, startX + 300, startY);

    drawL(*g*, startX + 360, startY);

    drawL(*g*, startX + 420, startY);

    drawI(*g*, startX + 480, startY);

    drawN(*g*, startX + 540, startY);

    drawA(*g*, startX + 600, startY);

  }

  private void drawK(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.BLUE,

*x* + 30,

*y* + 100,

      Color.CYAN

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x* - 10, *y*, *x*, *y* + 110);

    g2d.drawLine(*x*, *y* + 50, *x* + 30, *y*);

    g2d.drawLine(*x*, *y* + 50, *x* + 30, *y* + 100);

  }

  private void drawH(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.RED,

*x* + 20,

*y* + 100,

      Color.YELLOW

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x*, *y* - 10, *x*, *y* + 100);

    g2d.drawLine(*x*, *y* + 50, *x* + 20, *y* + 50);

    g2d.drawLine(*x* + 20, *y*, *x* + 20, *y* + 110);

  }

  private void drawA(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.GREEN,

*x* + 20,

*y* + 100,

      Color.ORANGE

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x*, *y* + 100, *x* + 10, *y*);

    g2d.drawLine(*x*, *y*, *x* + 30, *y* + 100);

    g2d.drawLine(*x* + 5, *y* + 50, *x* + 15, *y* + 50);

  }

  private void drawY(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.ORANGE,

*x* + 20,

*y* + 100,

      Color.RED

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x* - 15, *y*, *x* + 10, *y* + 50);

    g2d.drawLine(*x* + 10, *y* + 50, *x* + 40, *y*);

    g2d.drawLine(*x* + 10, *y* + 50, *x* + 10, *y* + 100);

  }

  private void drawR(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.MAGENTA,

*x* + 20,

*y* + 100,

      Color.PINK

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x*, *y*, *x*, *y* + 100);

    g2d.drawLine(*x*, *y*, *x* + 20, *y*);

    g2d.drawLine(*x*, *y* + 50, *x* + 20, *y* + 50);

    g2d.drawLine(*x* + 20, *y*, *x* + 20, *y* + 50);

    g2d.drawLine(*x*, *y* + 50, *x* + 20, *y* + 100);

  }

  private void drawU(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.CYAN,

*x* + 20,

*y* + 100,

      Color.BLUE

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x*, *y* + 10, *x*, *y* + 100);

    g2d.drawLine(*x*, *y* + 100, *x* + 20, *y* + 100);

    g2d.drawLine(*x* + 20, *y* + 25, *x* + 20, *y* + 100);

  }

  private void drawL(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.PINK,

*x* + 20,

*y* + 100,

      Color.RED

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x* - 30, *y*, *x*, *y* + 100);

    g2d.drawLine(*x*, *y* + 100, *x* + 20, *y* + 100);

  }

  private void drawI(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.YELLOW,

*x* + 20,

*y* + 100,

      Color.GREEN

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x* + 10, *y*, *x* + 10, *y* + 100);

    g2d.drawLine(*x* - 30, *y*, *x* + 20, *y*);

    g2d.drawLine(*x*, *y* + 100, *x* + 50, *y* + 100);

  }

  private void drawN(Graphics *g*, int *x*, int *y*) {

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

    GradientPaint gradient = new GradientPaint(

*x*,

*y*,

      Color.LIGHT\_GRAY,

*x* + 20,

*y* + 100,

      Color.DARK\_GRAY

    );

    g2d.setPaint(gradient);

    g2d.setStroke(new BasicStroke(5));

    g2d.drawLine(*x*, *y*, *x*, *y* + 130);

    g2d.drawLine(*x*, *y*, *x* + 20, *y* + 100);

    g2d.drawLine(*x* + 20, *y* + 100, *x* + 20, *y* - 30);

  }

}

