Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 6

по курсу «Языки и методы программирования»

«Программа с графическим пользовательским интерфейсом»

Студент группы ИУ9-22Б Лавров Р. Д.

Преподаватель Посевин Д. П.

Москва 2025

1 Задание

1. «Домик» с двускатной крышей, у которого n этажей, а на каждом этаже по m окон. Печное отопление – по желанию пользователя.

2. Сектор круга радиуса r с длиной дуги l, который по выбору пользователя может быть закрашен.

2 Результаты

Исходный код 1– 7.

Листинг 1 — PictureForm для 1ой задачи

```java
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;

public class PictureForm {
    private JSpinner floorsSpinner;
    private JSpinner windowsSpinner;
    private JCheckBox trubaBox;
    private JPanel mainPanel;
    private CanvasPanel canvasPanel1;
    public PictureForm() {
        $$$setupUI$$$();
        floorsSpinner.setValue(1);
        windowsSpinner.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent e) {
                int windows = (int) windowsSpinner.getValue();
                canvasPanel1.setWindows(windows);
            }
        });
        floorsSpinner.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent e) {
                int floors = (int) floorsSpinner.getValue();
                canvasPanel1.setFloors(floors);
            }
        });

        trubaBox.addItemListener(l -> {
            boolean truba = trubaBox.isSelected();
            canvasPanel1.setTruba(truba);
        });
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame("PictureForm");
        frame.setContentPane(new PictureForm().mainPanel);
        frame.setSize(500, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
    private void createUIComponents() {
        canvasPanel1 = new CanvasPanel();
    }
```

Листинг 2 — PictureForm для 1ой задачи (продолжение)

```
    private void $$$setupUI$$$() {
        createUIComponents();
        mainPanel = new JPanel();
        mainPanel.setLayout(new com.intellij.uiDesigner.core.GridLayoutManager(4, 1, new Insets(0, 0, 0, 0), -1, -1));
        floorsSpinner = new JSpinner();
        mainPanel.add(floorsSpinner, new com.intellij.uiDesigner.core.GridConstraints(0, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_HORIZONTAL, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_WANT_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
        windowsSpinner = new JSpinner();
        mainPanel.add(windowsSpinner, new com.intellij.uiDesigner.core.GridConstraints(1, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_HORIZONTAL, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_WANT_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
        trubaBox = new JCheckBox();
        trubaBox.setText("Печное отопление");
        mainPanel.add(trubaBox, new com.intellij.uiDesigner.core.GridConstraints(2, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_NONE, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_SHRINK | com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
        mainPanel.add(canvasPanel1, new com.intellij.uiDesigner.core.GridConstraints(3, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_CENTER, com.intellij.uiDesigner.core.GridConstraints.FILL_NONE, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_SHRINK | com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_SHRINK | com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_GROW, null, new Dimension(600, 600), null, 0, false));
    }
    public JComponent $$$getRootComponent$$$() {
        return mainPanel;
    }
}
```

Листинг 3 — CanvasPanel для 1ой задачи

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

public class CanvasPanel extends JPanel {
    private int floors = 1;
    private int windows = 0;
    private boolean truba = false;

    private int WINDOW_WIDTH = 20;
    private int WINDOW_HEIGHT = 20;
    private int WINDOW_SPACING = 40;
    private int FLOOR_HEIGHT = 40;
    private int ROOF_HEIGHT = 100;
    private int BUILDING_MARGIN = 50;

    private Polygon drawWindow(int x, int y) {
        Polygon window = new Polygon();
        window.addPoint(x, y);
        window.addPoint(x + WINDOW_WIDTH, y);
        window.addPoint(x + WINDOW_WIDTH, y - WINDOW_HEIGHT);
        window.addPoint(x, y - WINDOW_HEIGHT);
        return window;
    }

    public void setFloors(int floors) {
        this.floors = Math.max(1, floors);
        repaint();
    }

    public void setTruba(boolean truba) {
        this.truba = truba;
        repaint();
    }

    public void setWindows(int windows) {
        this.windows = Math.max(1, windows);
        repaint();
    }
```

Листинг 4 — CanvasPanel для 1ой задачи (продолжение)

```
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        int height = getHeight();
        int width = getWidth();
        int centerX = width / 2;

        int buildingWidth = windows * WINDOW_SPACING + BUILDING_MARGIN;
        int buildingHeight = floors * FLOOR_HEIGHT;
        int buildingTop = height / 5;

        Polygon roof = new Polygon();
        roof.addPoint(centerX, buildingTop);
        roof.addPoint(centerX - buildingWidth / 2, buildingTop + ROOF_HEIGHT);
        roof.addPoint(centerX + buildingWidth / 2, buildingTop + ROOF_HEIGHT);
        g.setColor(Color.RED);
        g.fillPolygon(roof);
        if (truba){
            Polygon truba = new Polygon();
            truba.addPoint(width / 2 + 10 + (windows * 10), buildingTop + 10);
            truba.addPoint(width / 2 + 20 + (windows * 10), buildingTop + 10);
            truba.addPoint(width / 2 + 20 + (windows * 10), buildingTop + 70);
            truba.addPoint(width / 2 + 10 + (windows * 10), buildingTop + 70);
            g.setColor(Color.BLACK);
            g.fillPolygon(truba);
        }

        Polygon body = new Polygon();
        body.addPoint(centerX - buildingWidth / 2, buildingTop + ROOF_HEIGHT);
        body.addPoint(centerX + buildingWidth / 2, buildingTop + ROOF_HEIGHT);
        body.addPoint(centerX + buildingWidth / 2, buildingTop + ROOF_HEIGHT + buildingHeight);
        body.addPoint(centerX - buildingWidth / 2, buildingTop + ROOF_HEIGHT + buildingHeight);
        g.setColor(Color.GREEN);
        g.fillPolygon(body);

        int windowRowY = buildingTop + ROOF_HEIGHT + WINDOW_HEIGHT + 10;
        for (int floor = 0; floor < floors; floor++) {
            int totalWindowsWidth = windows * WINDOW_WIDTH + (windows - 1) * (WINDOW_SPACING - WINDOW_WIDTH);
            int windowRowStartX = centerX - totalWindowsWidth / 2;

            for (int win = 0; win < windows; win++) {
                int windowX = windowRowStartX + win * WINDOW_SPACING;
                Polygon winPolygon = drawWindow(windowX, windowRowY);
                g.setColor(Color.CYAN);
                g.fillPolygon(winPolygon);
            }
            windowRowY += FLOOR_HEIGHT - 5;
        }
    }
}
```

Листинг 5 — PictureForm для 2ой задачи

```java
import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import java.awt.*;

public class PictureForm {
    private JPanel mainPanel;
    private CanvasPanel canvasPanel1;
    private JSlider slider1;
    private JSlider slider2;
    private JLabel label2;
    private JLabel label1;
    private JCheckBox fillCheckBox;

    public PictureForm() {
        $$$setupUI$$$();
        slider1.setValue(100);
        slider2.setValue(20);
        label1.setText("100");
        label2.setText("20");

        slider1.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent e) {
                int radius = (int) slider1.getValue();
                canvasPanel1.setRadius(radius);
                label1.setText(String.format("%d", radius));
            }
        });

        slider2.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent e) {
                int len = (int) slider2.getValue();
                canvasPanel1.setArcLength(len);
                label2.setText(String.format("%d", len));
            }
        });

        fillCheckBox.addItemListener(l -> {
            boolean fill = fillCheckBox.isSelected();
            canvasPanel1.setFilled( fill );
        });
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame("Сектор круга");
        frame.add(new PictureForm().mainPanel);
        frame.setSize(400, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
    private void createUIComponents() {
        canvasPanel1 = new CanvasPanel();
    }
```

Листинг 6 — PictureForm для 2ой задачи (продолжение)

```
    private void $$$setupUI$$$() {
        createUIComponents();
        mainPanel = new JPanel();
        mainPanel.setLayout(new com.intellij.uiDesigner.core.GridLayoutManager(4, 2, new Insets(0, 0, 0, 0), -1, -1));
        mainPanel.add(canvasPanel1, new com.intellij.uiDesigner.core.GridConstraints(3, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_CENTER, com.intellij.uiDesigner.core.GridConstraints.FILL_NONE, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_SHRINK | com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_SHRINK | com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_GROW, null, null, null, 0, false));
        slider1 = new JSlider();
        mainPanel.add(slider1, new com.intellij.uiDesigner.core.GridConstraints(0, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_HORIZONTAL, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_WANT_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
        slider2 = new JSlider();
        slider2.setMaximum(700);
        mainPanel.add(slider2, new com.intellij.uiDesigner.core.GridConstraints(1, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_HORIZONTAL, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_WANT_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
        label1 = new JLabel();
        label1.setText("Label");
        mainPanel.add(label1, new com.intellij.uiDesigner.core.GridConstraints(0, 1, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_NONE, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, new Dimension(20, 25), null, null, 0, false));
        label2 = new JLabel();
        label2.setText("Label");
        mainPanel.add(label2, new com.intellij.uiDesigner.core.GridConstraints(1, 1, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_NONE, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, new Dimension(20, 25), null, null, 0, false));
        fillCheckBox = new JCheckBox();
        fillCheckBox.setText("fill");
        mainPanel.add(fillCheckBox, new com.intellij.uiDesigner.core.GridConstraints(2, 0, 1, 1, com.intellij.uiDesigner.core.GridConstraints.ANCHOR_WEST, com.intellij.uiDesigner.core.GridConstraints.FILL_NONE, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_SHRINK | com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_CAN_GROW, com.intellij.uiDesigner.core.GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
    }

    /**
     * @noinspection ALL
     */
    public JComponent $$$getRootComponent$$$() {
        return mainPanel;
    }
}
```

Листинг 7 — CanvasPanel для 2ой задачи

```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.Arc2D;

public class CanvasPanel extends JPanel {
    private int radius = 100;
    private double angle = 20;
    private boolean filled = false;
    private Color sectorColor = Color.BLUE;

    public CanvasPanel() {
        setPreferredSize(new Dimension(300, 300));
    }

    public void setRadius(int radius) {
        this.radius = Math.max(1,radius);
        repaint();
    }

    public void setArcLength(int arcLength) {
        this.angle = arcLength * 360 / (2 * Math.PI * this.radius);
        repaint();
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
        repaint();
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.
    VALUE_ANTIALIAS_ON);

        int centerX = getWidth() / 2;
        int centerY = getHeight() / 2;

        int diameter = radius * 2;
        int maxDiameter = Math.min(getWidth(), getHeight()) - 40;
        if (diameter > maxDiameter) {
            diameter = maxDiameter;
        }

        Arc2D sector = new Arc2D.Double(
                centerX - diameter/2,
                centerY - diameter/2,
                diameter, diameter,
                0, angle,
                filled ? Arc2D.PIE : Arc2D.OPEN
        );

        g2d.setColor(sectorColor);
        if (filled) {
            g2d.fill(sector);
        }

        g2d.setColor(Color.BLACK);
        g2d.draw(sector);
    }
}
```

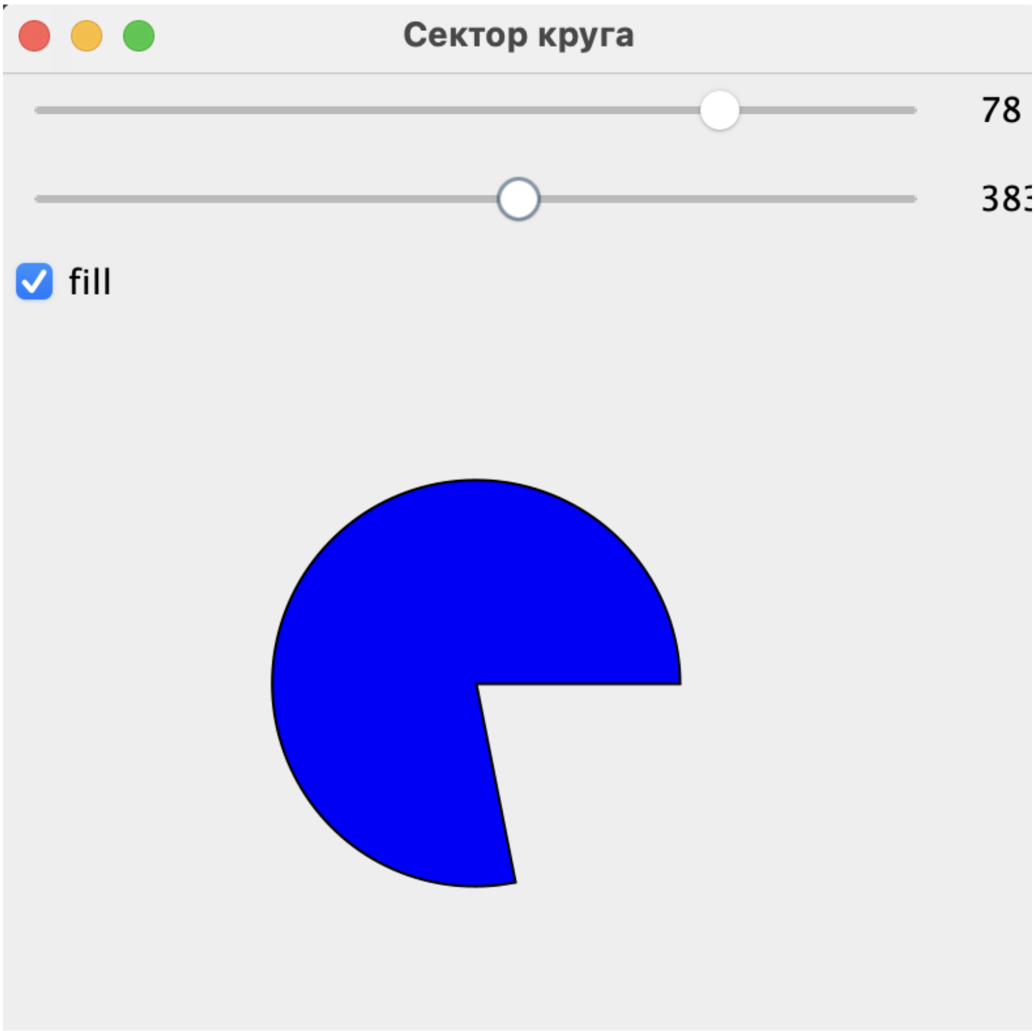
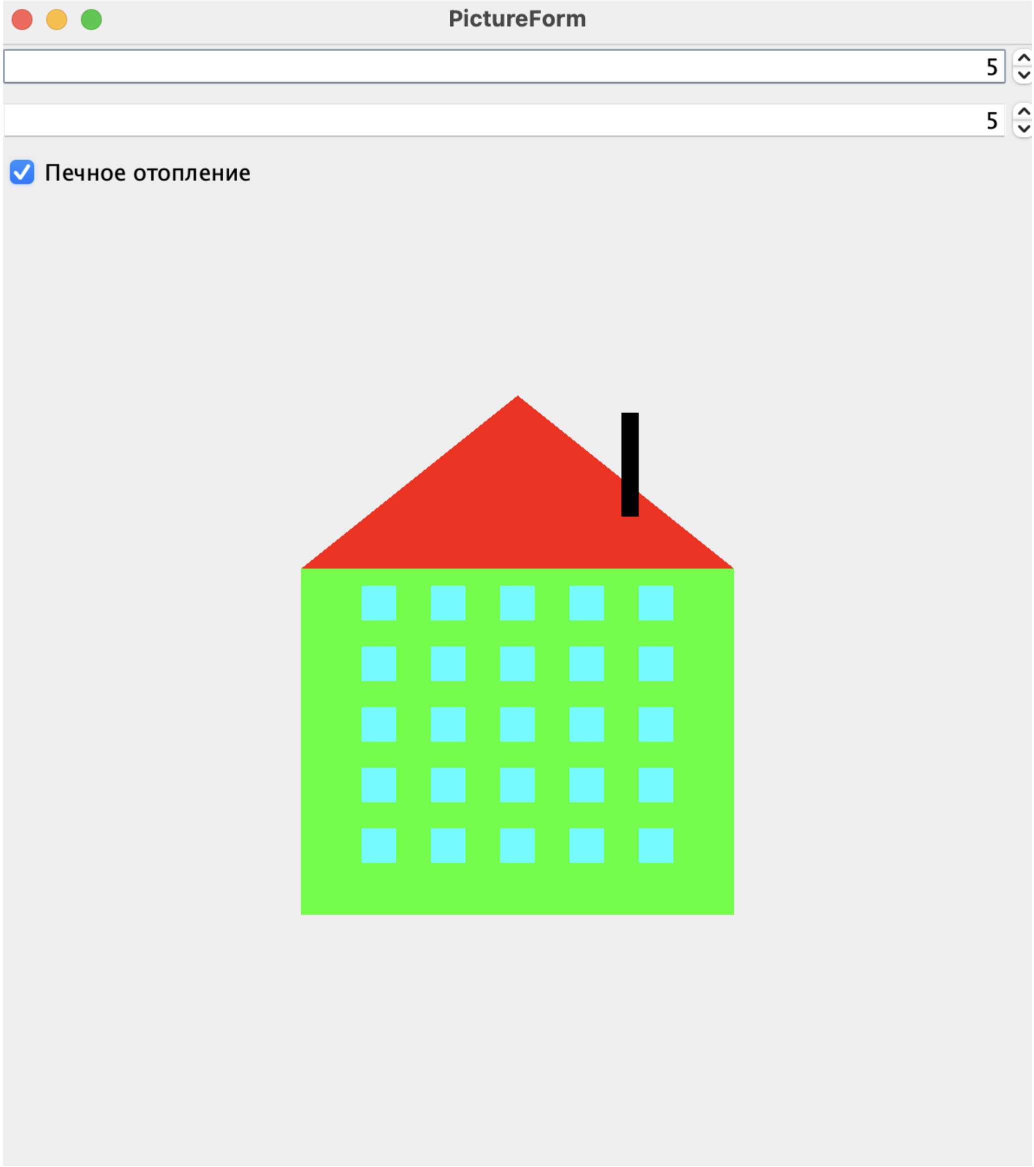


Рис. 1 — Результат работы 1ой программы



5

5

☑ Печное отопление

Рис. 2 — Результат работы 2ой программы

3 Вывод

Я научился писать программы с графическим пользовательским интерфейсом