МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**Дисциплина: Платформо - независимое программирование**

**Тема: «Библиотеки awt и swing для построения графического  
интерфейса пользователя. Обработка событий»**

Работу выполнил(а):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мазуренко А. А.

Направление подготовки: 02.03.02 Фундаментальная информатика

и информационные технологии

Направленность (профиль): Математическое и программное

обеспечение компьютерных технологий

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т. А. Приходько

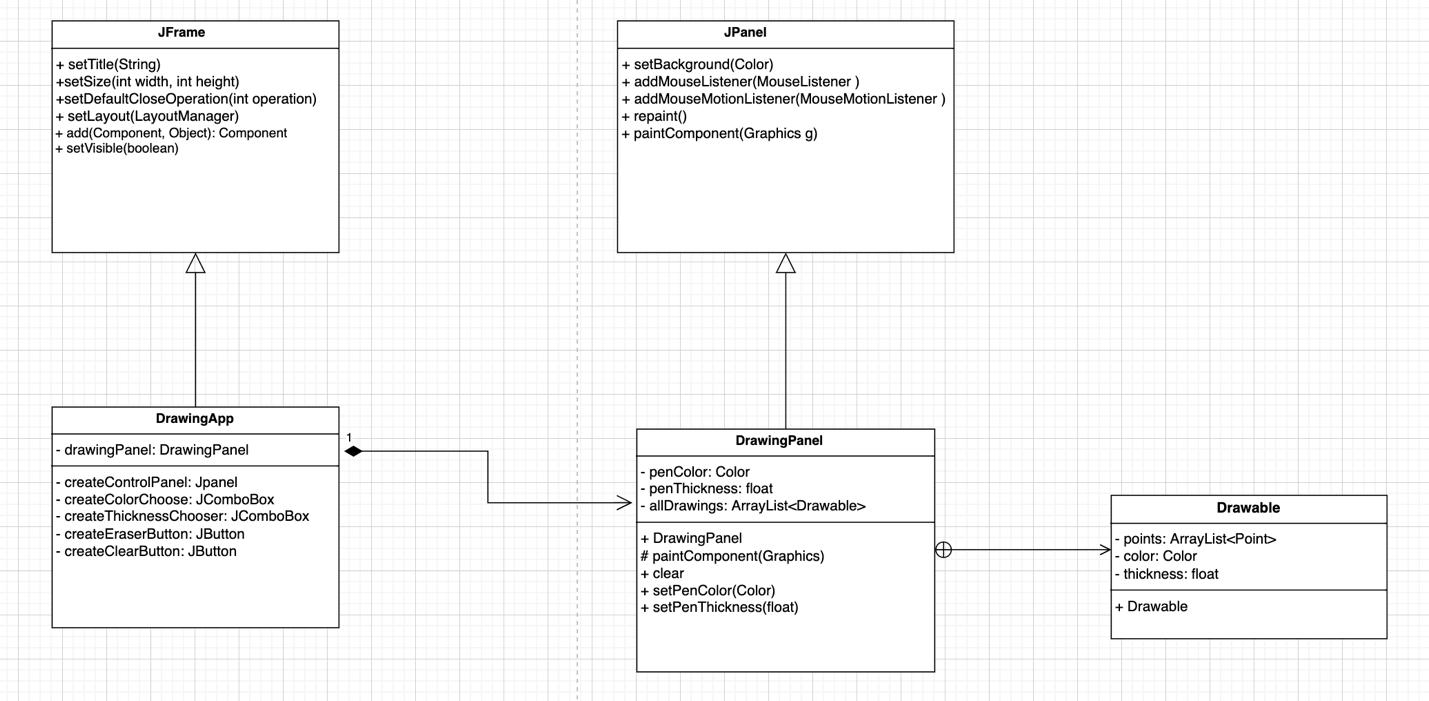
Краснодар

2025

**Цель работы:** Освоить методы разработки с графическим интерфейсом Java, ознакомиться с различными компоновщиками. Научиться обрабатывать события пользовательского интерфейса.

**Ход работы:**

1. **Диаграмма классов**

****

1. **Листинг программы**

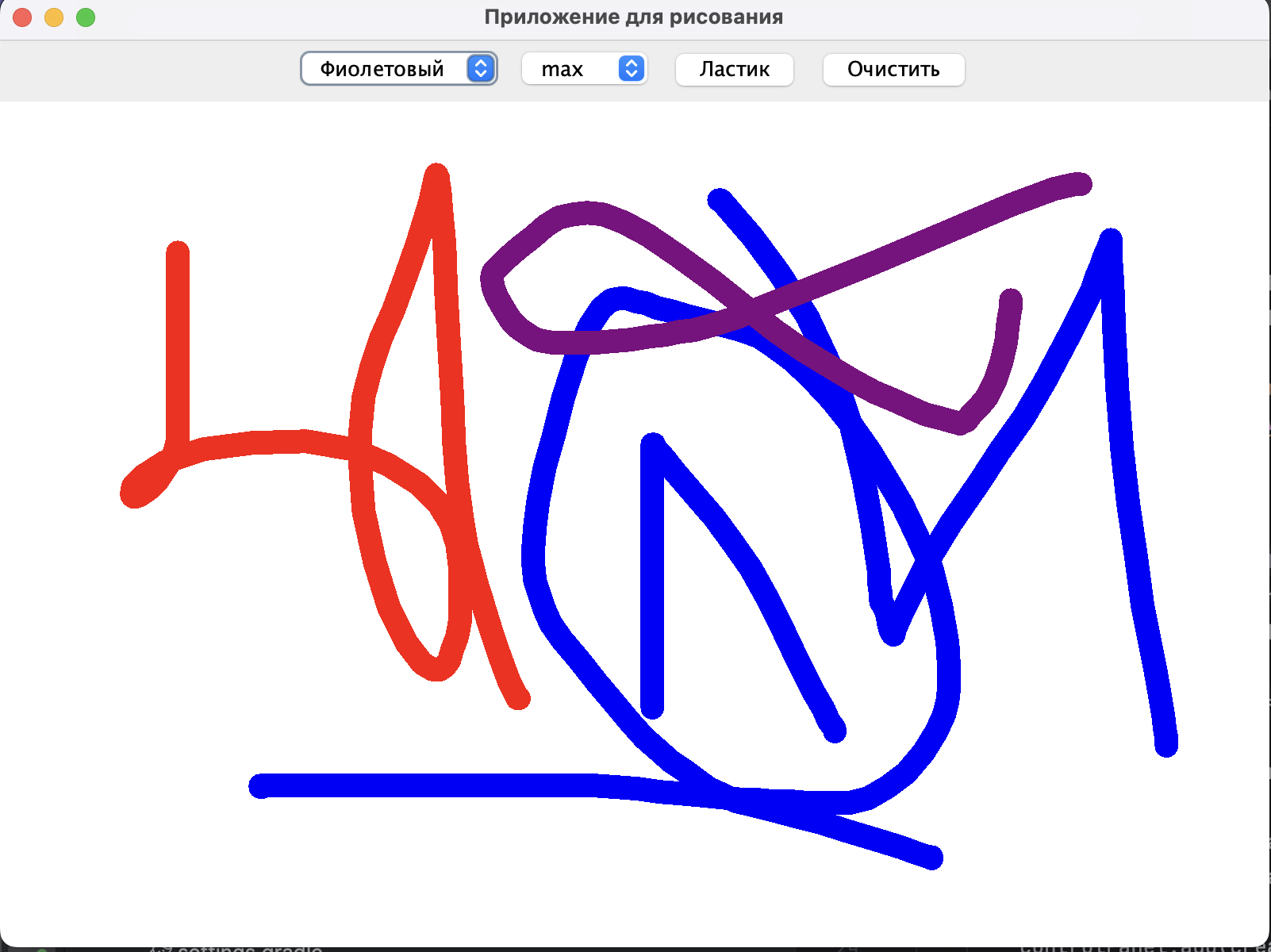
Файл DrawingApp.java :

package code;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class DrawingApp extends JFrame {  
 private DrawingPanel drawingPanel;  
  
 public DrawingApp() {  
 setTitle("Приложение для рисования");  
 setSize(800, 600);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new BorderLayout());  
  
 JPanel controlPanel = createControlPanel();  
  
 drawingPanel = new DrawingPanel();  
  
 controlPanel.add(createColorChooser());  
 controlPanel.add(createThicknessChooser());  
  
 controlPanel.add(createEraserButton());  
  
 controlPanel.add(createClearButton());  
  
 add(controlPanel, BorderLayout.*NORTH*);  
 add(drawingPanel, BorderLayout.*CENTER*);  
 }  
  
 private JPanel createControlPanel() {  
 JPanel controlPanel = new JPanel();  
 controlPanel.setLayout(new FlowLayout());  
 return controlPanel;  
 }  
  
 private JComboBox<String> createColorChooser() {  
 Color[] colorOptions = {  
 Color.*BLACK*,  
 Color.*RED*,  
 Color.*GREEN*,  
 Color.*BLUE*,  
 Color.*YELLOW*,  
 new Color(128, 0, 128),  
 Color.*ORANGE* };  
  
 String[] colorNames = {"Черный", "Красный", "Зеленый", "Синий", "Желтый", "Фиолетовый", "Оранжевый"};  
  
 JComboBox<String> colorComboBox = new JComboBox<>(colorNames);  
  
 colorComboBox.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 Color color = colorOptions[colorComboBox.getSelectedIndex()];  
 drawingPanel.setPenColor(color);  
 }  
 });  
  
 return colorComboBox;  
 }  
  
 private JComboBox<String> createThicknessChooser() {  
 String[] thicknessOptions = {"min", "mean", "max"};  
 float[] thicknessValues = {2.0f, 5.0f, 15.0f};  
  
 JComboBox<String> thicknessComboBox = new JComboBox<>(thicknessOptions);  
 thicknessComboBox.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 float penThickness = thicknessValues[thicknessComboBox.getSelectedIndex()];  
 drawingPanel.setPenThickness(penThickness);  
 }  
 });  
 return thicknessComboBox;  
 }  
  
 private JButton createEraserButton() {  
 JButton eraserButton = new JButton("Ластик");  
 eraserButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 drawingPanel.setPenColor(Color.*WHITE*);  
 }  
 });  
 return eraserButton;  
 }  
  
 private JButton createClearButton() {  
 JButton clearButton = new JButton("Очистить");  
 clearButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 drawingPanel.clear();  
 clearButton.setFocusable(false);  
 }  
 });  
 return clearButton;  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(() -> new DrawingApp().setVisible(true));  
 }  
}

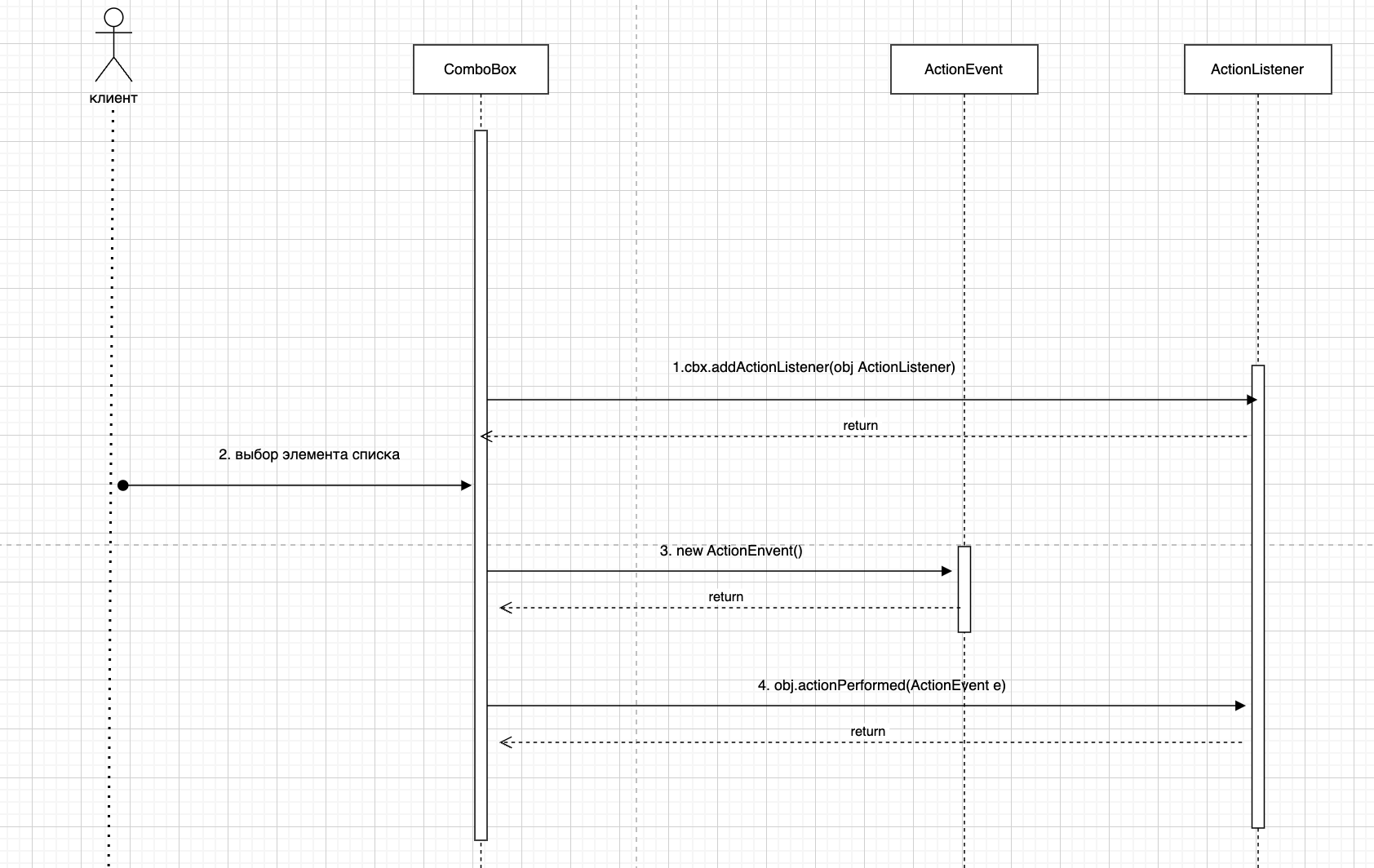
Файл DrawingPanel.java :

package code;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.util.ArrayList;  
  
public class DrawingPanel extends JPanel {  
 private Color penColor = Color.*BLACK*;  
 private float penThickness = 2.0f;  
 private ArrayList<Drawable> allDrawings = new ArrayList<>();  
  
 private static class Drawable {  
 ArrayList<Point> points;  
 Color color;  
 float thickness;  
  
 Drawable(ArrayList<Point> points, Color color, float thickness) {  
 this.points = points;  
 this.color = color;  
 this.thickness = thickness;  
 }  
 }  
  
 public DrawingPanel() {  
 setBackground(Color.*WHITE*);  
 addMouseListener(new MouseAdapter() {  
 @Override  
 public void mousePressed(MouseEvent e) {  
 ArrayList<Point> points = new ArrayList<>();  
 points.add(e.getPoint());  
 allDrawings.add(new Drawable(points, penColor, penThickness));  
 }  
 });  
  
 addMouseMotionListener(new MouseAdapter() {  
 @Override  
 public void mouseDragged(MouseEvent e) {  
 Drawable currentDrawing = allDrawings.get(allDrawings.size() - 1);  
 currentDrawing.points.add(e.getPoint());  
 repaint();  
 }  
 });  
 }  
  
  
 @Override  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
  
  
 Graphics2D g2d = (Graphics2D) g;  
  
  
 for (Drawable drawing : allDrawings) {  
 g2d.setColor(drawing.color);  
 g2d.setStroke(new BasicStroke(drawing.thickness, BasicStroke.*CAP\_ROUND*, BasicStroke.*JOIN\_ROUND*));  
  
 for (int i = 1; i < drawing.points.size(); i++) {  
 Point p1 = drawing.points.get(i - 1);  
 Point p2 = drawing.points.get(i);  
 g2d.drawLine(p1.x, p1.y, p2.x, p2.y);  
 }  
 }  
 }  
  
 public void clear() {  
 allDrawings.clear();  
 repaint();  
 }  
  
 public void setPenColor(Color color) {  
 this.penColor = color;  
 }  
  
 public void setPenThickness(float thickness) {  
 this.penThickness = thickness;  
 }  
}

1. **Результат выполнения программы**



1. **Диаграмма последовательности для обработки события выбора элемента в комбинированном списке**



**Вывод:**

В ходе лабораторной работы мы исследовали методы разработки графического интерфейса на Java, ознакомились с различными компоновщиками и изучили принципы обработки событий пользовательского интерфейса. Мы рассмотрели возможности библиотек AWT и Swing, создали интерфейс с использованием различных менеджеров компоновки и реализовали обработку событий, таких как нажатие кнопок, ввод данных в текстовые поля и взаимодействие с элементами управления.