**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 № 8

**Дисциплина:** Технология кроссплатформенного программирования

**Тема:** Модель обработки событий в 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 № 8

**Задачи: I. Обязательная задача для всех:** Найти описание следующих классов-слушателей и выписать основное: ChangeListener, MouseWheelListener.

1. ChangeListener - это интерфейс в Java, используемый для обработки изменений состояния компонентов, которые имеют изменяемое значение, например, ползунки (слайдеры), переключатели и др.

* Основной метод: void stateChanged(ChangeEvent e): Этот метод вызывается, когда состояние компонента, реализующего интерфейс ChangeListener, изменяется.

1. MouseWheelListener - это интерфейс в Java, который позволяет отслеживать события вращения колеса мыши.

* Основной метод: void mouseWheelMoved(MouseWheelEvent e): Этот метод вызывается при вращении колеса мыши. Параметр e содержит информацию о событии вращения колеса.

Оба эти интерфейса являются частью механизма обработки событий в Java и позволяют реагировать на изменения и действия пользователя, такие как перемещение ползунка (слайдера) или вращение колеса мыши.

**Индивидуальное задание.** К созданному интерфейсу из предыдущей лаб. раб. добавить функциональность – как минимум 3 различных слушателя (например, при нажатии кнопки выводить текст в текстовое поле, при наведении курсора мыши выводить соответствующее сообщение, при изменении ползунка типа JSlider выводить его значение, при нажатии клавиши Enter после набора текста перемещать его куда-нибудь и т.д.). У студентов с одним вариантом интерфейса не должны быть одинаковые слушатели!

Листинг

//radioButton 1 radioButton 2 и slider

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import javax.swing.JCheckBox;

import javax.swing.JColorChooser;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.JSlider;

import javax.swing.JSpinner;

import javax.swing.JTextArea;

import javax.swing.JTextField;

class Program {

public static void addTextFields(JFrame jfr) { GridLayout gridLayout = new GridLayout(2, 3); JPanel textFieldsPanel = new JPanel(); textFieldsPanel.setLayout(gridLayout);

// textFieldsPanel.setSize(20, 10);

JTextField textFieldArray[] = new JTextField[6]; for (int i = 0; i < 6; i++) {

textFieldArray[i] = new JTextField("Текстовое поле " + (i + 1), 0);

}

for (JTextField i : textFieldArray) { textFieldsPanel.add(i);

}

jfr.add(textFieldsPanel, BorderLayout.NORTH);

}

public static void addLeftPanel(JFrame jfr) {

// centerPanel.

JPanel centerPanel = new JPanel();

GridLayout gridLayout2 = new GridLayout(10, 1); centerPanel.setLayout(gridLayout2);

JPanel radioButtonPanel = new JPanel(); radioButtonPanel.setLayout(new BorderLayout());

GridLayout gridLayout = new GridLayout(9, 1); JPanel leftPanel = new JPanel(); leftPanel.setLayout(new BorderLayout(20, 0));

JPanel checkBoxPanel = new JPanel(); checkBoxPanel.setLayout(gridLayout);

JCheckBox checkBoxArray[] = new JCheckBox[9]; for (int i = 0; i < 9; i++) {

checkBoxArray[i] = new JCheckBox("Выбор " + (i + 1));

}

for (JCheckBox i : checkBoxArray) { checkBoxPanel.add(i);

}

JPanel multyPanel = new JPanel(); multyPanel.setLayout(new FlowLayout());

JComboBox comboBox = new JComboBox();

multyPanel.add(comboBox);

JSpinner spinner = new JSpinner(); multyPanel.add(spinner);

JRadioButton radioButton1 = new JRadioButton("Радиокнопка 1"); multyPanel.add(radioButton1);

JRadioButton radioButton2 = new JRadioButton("Радиокнопка 2"); radioButtonPanel.add(radioButton2, BorderLayout.CENTER);

centerPanel.add(multyPanel); centerPanel.add(radioButtonPanel);

leftPanel.add(checkBoxPanel, BorderLayout.WEST); leftPanel.add(centerPanel, BorderLayout.EAST);

jfr.add(leftPanel, BorderLayout.WEST);

radioButton1.addActionListener(e -> {

// System.out.print("123"); JOptionPane.showMessageDialog(jfr, radioButton1.getText());

});

radioButton2.addActionListener(e -> { JOptionPane.showMessageDialog(jfr, radioButton2.getText());

});

}

static JTextArea textArea = new JTextArea(20, 25);

public static void addRightTextPanel(JFrame jfr) { JPanel rightTextPanel = new JPanel(); rightTextPanel.setLayout(new BorderLayout());

textArea.setText("Текстовая область"); textArea.setLineWrap(true); textArea.setWrapStyleWord(true); rightTextPanel.add(textArea, BorderLayout.EAST); jfr.add(rightTextPanel, BorderLayout.EAST);

}

public static void addSlider(JFrame jfr) { JPanel sliderPanel = new JPanel(); sliderPanel.setLayout(new BorderLayout()); JSlider slider = new JSlider();

slider.setOrientation(JSlider.HORIZONTAL); sliderPanel.add(slider, BorderLayout.SOUTH); jfr.add(sliderPanel, BorderLayout.SOUTH);

slider.addChangeListener(e -> {

Color color = JColorChooser.showDialog(null, "Выберите цвет", Color.BLACK);

// set bg color slider.setBackground(color);

});

}

public static void main(String[] args) { JFrame frame = new JFrame("Lab8");

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

// fix size frame.setSize(630, 400); frame.setResizable(false);

addTextFields(frame); addRightTextPanel(frame); addLeftPanel(frame); addSlider(frame);

frame.setVisible(true);

}

}







