



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет Информационных технологий**

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

## **ЛАБОРАТОРНАЯ РАБОТА № 8**

**Дисциплина:** Технологии кроссплатформенного программирования

**Тема:** Модель обработки событий в Java.

**Выполнила:** студентка группы 211-725

Алюбаева Карина Ислямбековна

**Дата** 01.12.2023

**Москва**

**2023**

# Оглавление

Введение .....	2
Цель работы: .....	2
Задание: .....	2
Теоретическое обоснование .....	3
Ход работы.....	4
Выполнение Задания 1.....	4
Выполнение Задания 2.....	5
Листинг.....	5
Результат работы программы: .....	15

# Введение

## Цель работы:

Научиться работать с обработчиками событий в Java на примере интерфейса своего варианта.

## Задание:

1. Найти описание следующих классов-слушателей и выписать основное: `ChangeListener`, `MouseWheelListener`.
2. 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.)

# Теоретическое обоснование

Обработка событий в языке Java организована с использованием специальных классов-слушателей (listeners). Т.е., если мы хотим обработать нажатие на кнопку, то мы регистрируем для этой кнопки объект-слушатель, в одном из методов которого прописываем действия на произошедшее событие. Причем сделать это можно как минимум двумя способами:

- ☺ 1 – с помощью анонимного вложенного класса;
- ☺ 2 – с помощью реализации нужного интерфейса.

В данной лабораторной работе я буду использовать следующие слушатели:

Класс события	Интерфейс слушателя	Методы слушателя	Описание метода
ActionEvent	ActionListener	actionPerformed()	Нажатие на кнопку, нажатие клавиши Enter при вводе текста (для компонентов типа JButton, JList, JTextField)
MouseEvent	MouseListener	mouseEntered()	Курсор мыши переместился на объект

# Ход работы

## Выполнение Задания 1.

☺ `ChangeListener`:

**Описание:** `ChangeListener` - это интерфейс в библиотеке Swing, предназначенный для обработки событий изменения состояния компонентов. Он используется в основном с компонентами, значения которых могут изменяться в течение времени, такими как ползунки (`JSlider`), спиннеры (`JSpinner`) и другие компоненты.

**Основное:** Метод `stateChanged(ChangeEvent e)`: Вызывается при изменении состояния компонента, который реализует данный слушатель. Объект `ChangeEvent` предоставляет информацию об изменении, например, источник события.

☺ `MouseWheelListener`:

**Описание:** `MouseWheelListener` - это интерфейс в библиотеке AWT и Swing, который обрабатывает события вращения колесика мыши. Он позволяет реагировать на прокрутку колесика мыши вверх или вниз.

**Основное:** Метод `mouseWheelMoved(MouseWheelEvent e)`: Вызывается при вращении колеса мыши. Объект `MouseWheelEvent` предоставляет информацию о событии, такую как компонент, над которым произошло событие, и количество единиц прокрутки. Положительное значение указывает прокрутку вперед, отрицательное - назад.

## Выполнение Задания 2

### Листинг

У меня вариант 1 (первая в списке)

Изменения в коде, вызванные добавлением слушателей, я выделила маркером, чтобы лучше было заметно. Остальное осталось таким, каким было раньше.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class MyFrame {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Slider Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 320);

        // Создаем JSlider
        JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 100, 50);

        // Создаем JPanel для размещения JSlider
        JPanel sliderPanel = new JPanel();
        sliderPanel.setLayout(new BorderLayout());
        sliderPanel.add(slider, BorderLayout.SOUTH);

        //////////////////////////////////////
        /////
    }
}
```

```

// Создаем JPanel для текстовых полей с использованием массива
JPanel textFieldsPanel = new JPanel();
textFieldsPanel.setLayout(new GridLayout(2, 3)); // 2 строки, 3 столбца

JTextField[] textFields = new JTextField[6];
for (int i = 0; i < textFields.length; i++) {
    textFields[i] = new JTextField(10); // 10 символов ширина
    textFieldsPanel.add(textFields[i]);
}

////////////////////////////////////
/////

// Создаем JPanel для чек-боксов и их заголовков с использованием мас-
сива
JPanel checkBoxesPanel = new JPanel();
checkBoxesPanel.setLayout(new BoxLayout(checkBoxesPanel, BoxLay-
out.Y_AXIS));
checkBoxesPanel.setPreferredSize(new Dimension(100, 0));

JCheckBox[] checkBoxes = new JCheckBox[9];
for (int i = 0; i < checkBoxes.length; i++) {
    checkBoxes[i] = new JCheckBox("Выбор " + (i + 1));
    checkBoxesPanel.add(checkBoxes[i]);
}

////////////////////////////////////
/////

// Создаем JTextArea для большой текстовой области

```





```

JComboBox<String> comboBox1 = new JComboBox<>(options1);
// Добавляем компонент с выпадающим списком на панель
comboBoxPanel2.add(comboBox1);
// Создаем компонент с однострочным полем выбора (спиннер)
SpinnerModel spinnerModel = new SpinnerListModel(options1);
JSpinner spinner = new JSpinner(spinnerModel);
spinnerPanel.add(spinner);

```

```

// Создаем панель для РАДИОКНОПКА
JPanel radioPanel = new JPanel();
JRadioButton radioButton = new JRadioButton("Радио1");
radioPanel.add(radioButton);
// Создаем панель для радиокнопки
JPanel radioPanel2 = new JPanel();
JRadioButton radioButton2 = new JRadioButton("Радио2");
radioPanel2.add(radioButton2);

```

//ПАНЕЛЬ ДЛЯ ПАНЕЛЕЙ С ВЫПАДАЮЩИМ СПИСКОМ, СПИННЕ-  
РОМ И РАДИОКНОПКОЙ 1

```

JPanel centerPanel1 = new JPanel();
centerPanel1.setLayout(new GridLayout(1, 3));
centerPanel1.add(comboBoxPanel);
centerPanel1.add(spinnerPanel);
centerPanel1.add(radioPanel);

```

//ЦЕНТРАЛЬНАЯ ПАНЕЛЬ ДЛЯ ПАНЕЛИ С ПАНЕЛЯМИ И НИЖНЕЙ  
РАДИОКНОПКОЙ 2

```

centerPanel.add(centerPanel1);

```

```
centerPanel.add(radioPanel2);
```

```
//ДОБАВЛЕНИЕ СЛУШАТЕЛЕЙ////////////////////////////////////
```

```
for (JCheckBox checkBox : checkBoxes) {  
    checkBox.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            JOptionPane.showMessageDialog(frame, "Вы выбрали чек-бокс");  
        }  
    });  
}
```

```
// Добавляем слушателя для текстовых полей
```

```
for (JTextField textField : textFields) {  
    textField.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            String text = textField.getText();  
            textArea.append(text + "\n");  
        }  
    });  
}
```

```
// Добавляем слушателя для радио-кнопок
```

```
radioButton.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseEntered(MouseEvent e) {  
        JOptionPane.showMessageDialog(frame, "Нажми на меня");  
    }  
});
```

```
}  
});
```

```
radioButton2.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseEntered(MouseEvent e) {  
        JOptionPane.showMessageDialog(frame, "Нажми на меня");  
    }  
});
```

```
/////////////////////////////////////  
/////  
////////ЗАПОЛНЕНИЕ  
ОКНА////////////////////////////////////
```

```
// Добавляем панель с выпадающим списком в окно  
frame.add(centerPanel, BorderLayout.CENTER);
```

```
// Добавляем JPanel с текстовыми полями в верхнюю часть окна  
frame.add(textFieldsPanel, BorderLayout.NORTH);
```

```
// Добавляем JPanel с чек-боксами в левую часть окна  
frame.add(checkBoxesPanel, BorderLayout.WEST);
```

```
// Добавляем JPanel с JSlider в нижнюю часть окна  
frame.add(sliderPanel, BorderLayout.SOUTH);
```

```
//Добавляем JPanel с текстовой областью в правую часть экрана  
frame.add(textAreaScrollPane, BorderLayout.EAST);
```

```

        frame.setVisible(true);
    }
}

```

```

import javax.swing.*;

//import org.w3c.dom.events.MouseEvent;

import java.awt.*;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class MyFrame {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Slider Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 320);

        // Создаем JSlider
        JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 100, 50);

        // Создаем JPanel для размещения JSlider
        JPanel sliderPanel = new JPanel();
        sliderPanel.setLayout(new BorderLayout());
        sliderPanel.add(slider, BorderLayout.SOUTH);

        ///////////////////////////////////////////////////
        ///////////////////////////////////////////////////
        // Создаем JPanel для текстовых полей с использованием массива
        JPanel textFieldsPanel = new JPanel();
        textFieldsPanel.setLayout(new GridLayout(2, 3)); // 2 строки, 3 столбца

        JTextField[] textFields = new JTextField[6];
        for (int i = 0; i < textFields.length; i++) {
            textFields[i] = new JTextField(10); // 10 символов ширина
            textFieldsPanel.add(textFields[i]);
        }

        ///////////////////////////////////////////////////
        ///////////////////////////////////////////////////
        // Создаем JPanel для чек-боксов и их заголовков с использованием массива
        JPanel checkBoxesPanel = new JPanel();
    }
}

```

```

        checkBoxesPanel.setLayout(new BorderLayout(checkBoxesPanel, BorderLayout.Y_AXIS));
        checkBoxesPanel.setPreferredSize(new Dimension(100, 0));

        JCheckBox[] checkBoxes = new JCheckBox[9];
        for (int i = 0; i < checkBoxes.length; i++) {
            checkBoxes[i] = new JCheckBox("Выбор " + (i + 1));
            checkBoxesPanel.add(checkBoxes[i]);
        }

////////////////////////////////////
////////////////////////////////////
        // Создаем JTextArea для большой текстовой области
        JTextArea textArea = new JTextArea(5, 20); // 5 строк, 20 символов в строке
        textArea.setWrapStyleWord(true);
        textArea.setLineWrap(true);
        JScrollPane textAreaScrollPane = new JScrollPane(textArea);

////////////////////////////////////
////////////////////////////////////
        JPanel centerPanel = new JPanel();
        centerPanel.setLayout(new GridLayout(2, 1)); // 2 строки, 1 столбец

        // Создаем JPanel для компонента с ВЫПАДАЮЩИЙ СПИСОК
        JPanel comboBoxPanel = new JPanel();
        // Создаем массив с вариантами
        String[] options = {"1", "11", "111"};
        // Создаем компонент
        JComboBox<String> comboBox = new JComboBox<>(options);
        // Добавляем компонент на панель
        comboBoxPanel.add(comboBox);

        //СПИННЕР
        JPanel spinnerPanel = new JPanel();
        // Создаем JPanel для компонента с выпадающим списком
        JPanel comboBoxPanel2 = new JPanel();
        // Создаем массив с вариантами для выпадающего списка
        String[] options1 = {"0", "1", "2", "3", "4", "5"};
        // Создаем компонент с выпадающим списком
        JComboBox<String> comboBox1 = new JComboBox<>(options1);
        // Добавляем компонент с выпадающим списком на панель
        comboBoxPanel2.add(comboBox1);
        // Создаем компонент с однострочным полем выбора (спиннер)
        SpinnerModel spinnerModel = new SpinnerListModel(options1);
        JSpinner spinner = new JSpinner(spinnerModel);

```

```

spinnerPanel.add(spinner);

// Создаем панель для РАДИОКНОПКА
JPanel radioPanel = new JPanel();
JRadioButton radioButton = new JRadioButton("Радио1");
radioPanel.add(radioButton);
// Создаем панель для радиокнопки
JPanel radioPanel2 = new JPanel();
JRadioButton radioButton2 = new JRadioButton("Радио2");
radioPanel2.add(radioButton2);

//ПАНЕЛЬ ДЛЯ ПАНЕЛЕЙ С ВЫПАДАЮЩИМ СПИСКОМ, СПИННЕ-РОМ И РАДИОКНОПКОЙ 1
JPanel centerPanel1 = new JPanel();
centerPanel1.setLayout(new GridLayout(1, 3));
centerPanel1.add(comboBoxPanel);
centerPanel1.add(spinnerPanel);
centerPanel1.add(radioPanel);

//ЦЕНТРАЛЬНАЯ ПАНЕЛЬ ДЛЯ ПАНЕЛИ С ПАНЕЛЯМИ И НИЖНЕЙ РАДИОКНОПКОЙ 2
centerPanel.add(centerPanel1);
centerPanel.add(radioPanel2);

////////////////////////////////////
////////////////////////////////////
////////ДОБАВЛЕНИЕ СЛУШАТЕ-
ЛЕЙ////////////////////////////////////
////////////////////////////////////

for (JCheckBox checkBox : checkBoxes) {
    checkBox.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(frame, "Вы выбрали чек-бокс");
        }
    });
}

// Добавляем слушателя для текстовых полей
for (JTextField textField : textFields) {
    textField.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String text = textField.getText();
            textArea.append(text + "\n");
        }
    });
}
}

```

```

        // Добавляем слушателя для радио-кнопок
        radioButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                JOptionPane.showMessageDialog(frame, "Нажми на меня");
            }
        });

        radioButton2.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                JOptionPane.showMessageDialog(frame, "Нажми на меня");
            }
        });

////////////////////////////////////
////////////////////////////////////
////////ЗАПОЛНЕНИЕ
ОКНА////////////////////////////////////
////////////////////////////////////

        // Добавляем панель с выпадающим списком в окно
        frame.add(centerPanel, BorderLayout.CENTER);

        // Добавляем JPanel с текстовыми полями в верхнюю часть окна
        frame.add(textFieldsPanel, BorderLayout.NORTH);

        // Добавляем JPanel с чек-боксами в левую часть окна
        frame.add(checkBoxesPanel, BorderLayout.WEST);

        // Добавляем JPanel с JSlider в нижнюю часть окна
        frame.add/sliderPanel, BorderLayout.SOUTH);

        //Добавляем JPanel с текстовой областью в правую часть экрана
        frame.add(textAreaScrollPane, BorderLayout.EAST);

        frame.setVisible(true);
    }
}

```

## Результат работы программы:





