

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий

Кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 7

Дисциплина: Технологии кроссплатформенного программирования

Тема: Создание программ с графическим интерфейсом пользователя на языке Java. Классы пакета Swing

Выполнила: студентка группы 211-725

Алюбаева Карина Ислямбековна

Дата 28.10.2023

Москва

2023

Оглавление

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

Введение

Цель работы:

Научиться создавать интерфейсы

Задание:

1. Выписать все классы и методы, использованные в примере, и их описание из справки, и еще несколько полезных методов для каждого класса.
2. Создать интерфейс по изображению (использовать разные виды компонентов, если одинаковых объектов больше 3, то использовать массив

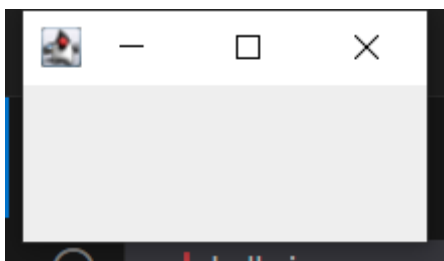
Теоретическое обоснование

Для создания графического интерфейса пользователя в языке Java есть несколько графических пакетов библиотек (AWT, Swing, SWT и др.). Пакет Swing является одним из наиболее простых в применении и содержит классы для реализации большинства современных элементов GUI (Graphical User Interface – графический интерфейс пользователя).

Базовым объектом при создании пользовательского интерфейса является окно. В классификации языка Java им является класс JFrame. Для создания окна достаточно следующего кода:

```
import
javax.swing.*
;
public class hello {
    public static void main(String s[]) {
        JFrame frame = new JFrame("FrameDemo");// создаем окно с заголовком
FrameDemo
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);//делаем базовым
действием при
        //закрытие окна выход из приложения, иначе окно закроется, а программа в
памяти останется
        frame.setSize(175,100);//задаем размер окна
        frame.setVisible(true);//делаем его видимым
    }
}
```

Результатом работы будет вот такое окно:



Дальнейшая работа с окном предусматривает использование менеджеров компоновки или компоновщиков. Это специальные классы, которые позволяют упаковывать содержимое окна и задавать нужное поведение всех элементов окна в зависимости от изменения размеров окна.

Ход работы

Выполнение Задания 1.

Пример, который вы предоставили, использует множество классов и методов из библиотеки Swing в Java. Вот список классов и методов, которые показались мне очень полезными:

1. JFrame:
 - a. JFrame(String title): Конструктор для создания окна с указанным заголовком.
 - b. setDefaultCloseOperation(int operation): Устанавливает операцию, которая выполняется при закрытии окна.
 - c. setSize(int width, int height): Устанавливает размер окна.
 - d. setVisible(boolean visible): Делает окно видимым.
 - e. pack(): Упаковывает окно, чтобы привести его к оптимальному размеру.
 - f. setMinimumSize(Dimension minimumSize): Устанавливает минимальный размер окна.
2. JPanel:
 - a. JPanel(): Конструктор для создания панели.
 - b. setLayout(LayoutManager manager): Устанавливает компоновщик для панели.
 - c. add(Component comp): Добавляет компонент на панель.
3. JButton(String text): Конструктор для создания кнопки с указанным текстом.
4. JLabel(String text): Конструктор для создания метки с указанным текстом.
5. JCheckBox(String text): Конструктор для создания флажка с указанным текстом.
6. ButtonGroup(): Конструктор для создания группы радиокнопок.

7. `JRadioButton(String text)`: Конструктор для создания радиокнопки с указанным текстом.
8. `TextArea()`: Конструктор для создания многострочного текстового поля.
9. `TextField()`: Конструктор для создания однострочного текстового поля.
10. `Box(int axis)`: Конструктор для создания объекта `Box` с указанным ориентацией (горизонтальной или вертикальной).

Выполнение Задания 2

У меня вариант 1 (первая в списке)

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
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,
BoxLayout.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(radiusPanel2);
```

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

////

////////ЗАПОЛНЕНИЕ

[illegible]

```
// 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 java.awt.*;

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
JTextArea textArea = new JTextArea(5, 20); // 5 строк, 20 символов в
стро-ке
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 СПИСОКОМ, 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);

////////////////////////////////////
////////////////////////////////////
////////ЗАПОЛНЕНИЕ ОК-
НА////////////////////////////////////
////////////////////////////////////

    // 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:

Slider Example

<input type="checkbox"/> Выбор 1	1	0	<input type="radio"/> Радио1
<input type="checkbox"/> Выбор 2			
<input type="checkbox"/> Выбор 3			
<input type="checkbox"/> Выбор 4			
<input type="checkbox"/> Выбор 5			
<input type="checkbox"/> Выбор 6			<input type="radio"/> Радио2
<input type="checkbox"/> Выбор 7			
<input type="checkbox"/> Выбор 8			
<input type="checkbox"/> Выбор 9			

Slider bar at the bottom.