**9. Работа с библиотекой SWING**

Swing — это библиотека для создания графических пользовательских интерфейсов (GUI) в Java, которая является частью стандартной библиотеки Java (в пакете javax.swing). Swing позволяет разрабатывать интерфейсы с элементами управления, такими как окна, кнопки, текстовые поля, списки и другие визуальные компоненты.

Swing работает поверх Abstract Window Toolkit (AWT), но предоставляет более гибкий и мощный набор инструментов для разработки GUI, чем AWT. Это дает разработчикам возможность создавать кросс-платформенные графические приложения с более сложными и настраиваемыми интерфейсами.

**Основные компоненты Swing**

* JFrame: Основное окно приложения.
* JPanel: Контейнер для других компонентов.
* JButton: Кнопка для взаимодействия с пользователем.
* JLabel: Метка, отображающая текст или изображения.
* JTextField: Поле для ввода текста.
* JTextArea: Многострочное текстовое поле.
* JComboBox: Выпадающий список.
* JCheckBox: Флажок для выбора/отмены выбора.
* JRadioButton: Радио-кнопка для выбора одного элемента из группы.
* JList: Список элементов.
* JTable: Таблица для отображения данных.

**1. Основное окно: JFrame**

Для создания основного окна приложения в Swing используется класс JFrame. Он представляет собой контейнер, в котором могут размещаться другие компоненты.

Пример создания простого окна:

import javax.swing.\*;

public class SwingExample {

public static void main(String[] args) {

// Создаем основное окно

JFrame frame = new JFrame("Пример Swing");

// Задаем поведение окна при закрытии

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

// Устанавливаем размер окна

frame.setSize(300, 200);

// Делаем окно видимым

frame.setVisible(true);

}

}

* JFrame — это окно приложения.
* setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE) — задает поведение при закрытии окна (выход из приложения).
* setSize(300, 200) — устанавливает размер окна.
* setVisible(true) — делает окно видимым на экране.

**2. Добавление компонентов: JPanel**

Для размещения элементов управления внутри окна используется панель JPanel, которая может содержать другие компоненты, такие как кнопки, текстовые поля и метки.

Пример с добавлением панели и кнопки:

import javax.swing.\*;

public class SwingExample {

public static void main(String[] args) {

// Создаем основное окно

JFrame frame = new JFrame("Пример Swing");

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

frame.setSize(300, 200);

// Создаем панель для размещения компонентов

JPanel panel = new JPanel();

// Создаем кнопку

JButton button = new JButton("Нажми меня");

// Добавляем кнопку на панель

panel.add(button);

// Добавляем панель в окно

frame.add(panel);

// Делаем окно видимым

frame.setVisible(true);

}

}

* JPanel — это контейнер для других компонентов.
* JButton — это кнопка, которую можно добавить в панель.
* frame.add(panel) — добавляем панель в окно.

**3. Обработчики событий**

Для взаимодействия с пользователем в Swing используют обработчики событий. Когда пользователь нажимает кнопку или взаимодействует с другими элементами, нужно выполнить определенное действие.

Пример: обработка нажатия на кнопку:

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class SwingExample {

public static void main(String[] args) {

// Создаем основное окно

JFrame frame = new JFrame("Пример Swing");

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

frame.setSize(300, 200);

// Создаем панель

JPanel panel = new JPanel();

// Создаем кнопку

JButton button = new JButton("Нажми меня");

// Добавляем обработчик события для кнопки

button.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Кнопка была нажата!");

}

});

// Добавляем кнопку в панель

panel.add(button);

// Добавляем панель в окно

frame.add(panel);

// Делаем окно видимым

frame.setVisible(true);

}

}

* addActionListener — метод для добавления обработчика событий на кнопку.
* JOptionPane.showMessageDialog() — показывает простое диалоговое окно с сообщением.

**4. Основные компоненты Swing**

**Кнопка (JButton)**

JButton button = new JButton("Нажми меня");

**Метка (JLabel)**

JLabel label = new JLabel("Это метка");

**Текстовое поле (JTextField)**

JTextField textField = new JTextField(20); // 20 - количество символов

**Многострочное текстовое поле (JTextArea)**

JTextArea textArea = new JTextArea(5, 20); // 5 строк, 20 столбцов

**Выпадающий список (JComboBox)**

String[] options = {"Опция 1", "Опция 2", "Опция 3"};

JComboBox<String> comboBox = new JComboBox<>(options);

**Список (JList)**

String[] items = {"Элемент 1", "Элемент 2", "Элемент 3"};

JList<String> list = new JList<>(items);

**Флажок (JCheckBox)**

JCheckBox checkBox = new JCheckBox("Согласен");

**Радиокнопки (JRadioButton)**

JRadioButton radioButton1 = new JRadioButton("Опция 1");

JRadioButton radioButton2 = new JRadioButton("Опция 2");

// Группа радиокнопок, чтобы можно было выбрать только одну

ButtonGroup group = new ButtonGroup();

group.add(radioButton1);

group.add(radioButton2);

**5. Использование Layout Managers**

Swing поддерживает различные менеджеры компоновки (layout managers), которые управляют расположением компонентов в контейнерах. Некоторые из них:

* FlowLayout — компоненты размещаются по порядку в строке (по умолчанию).
* BorderLayout — компоненты размещаются в пяти областях (север, юг, восток, запад и центр).
* GridLayout — компоненты размещаются в сетке с одинаковыми размерами.
* BoxLayout — компоненты размещаются по одной оси (по горизонтали или вертикали).

Пример использования BorderLayout:

import javax.swing.\*;

import java.awt.\*;

public class SwingExample {

public static void main(String[] args) {

// Создаем основное окно

JFrame frame = new JFrame("Пример Swing с BorderLayout");

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

frame.setSize(400, 300);

// Создаем панель с BorderLayout

JPanel panel = new JPanel(new BorderLayout());

// Создаем компоненты

JButton button1 = new JButton("Север");

JButton button2 = new JButton("Юг");

JButton button3 = new JButton("Центр");

// Размещаем компоненты

panel.add(button1, BorderLayout.NORTH);

panel.add(button2, BorderLayout.SOUTH);

panel.add(button3, BorderLayout.CENTER);

// Добавляем панель в окно

frame.add(panel);

// Делаем окно видимым

frame.setVisible(true);

}

}

**6. Диалоговые окна в Swing**

Swing предоставляет класс JOptionPane для создания стандартных диалоговых окон: для сообщений, ввода данных и подтверждений.

Пример диалогового окна с подтверждением:

import javax.swing.\*;

public class SwingExample {

public static void main(String[] args) {

int option = JOptionPane.showConfirmDialog(null, "Вы уверены?", "Подтверждение", JOptionPane.YES\_NO\_OPTION);

if (option == JOptionPane.YES\_OPTION) {

JOptionPane.showMessageDialog(null, "Вы выбрали: Да");

} else {

JOptionPane.showMessageDialog(null, "Вы выбрали: Нет");

}

}

}

Заключение

Swing — это мощная и гибкая библиотека для создания графических пользовательских интерфейсов в Java. Она предоставляет множество компонентов и инструментов для разработки приложений с графическим интерфейсом. Хотя Swing несколько устарел по сравнению с более современным JavaFX, он все еще широко используется благодаря своей кросс-платформенности и простоте в использовании.