МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Кафедра прикладной информатики**

Лабораторная работа №3

ПО ДИСЦИПЛИНЕ «Программирование мобильных устройств»

НА ТЕМУ:

«**Использование механизма событий Android**»

**Выполнил:**

студент группы ПИН-Б-З-22-1

Пензев Константин Сергеевич

**Проверил:**

Старший преподаватель института перспективной инженерии

Мартыновская Александра Сергеевна

Ставрополь, 2024 г.

**Лабораторная работа №3. Использование механизма событий Android**

**Цель работы:** Научиться обрабатывать события, такие как клик по кнопке, и использовать механизмы подписки на события в Android.

**1. Введение**

В данной лабораторной работе мы изучили механизм обработки событий в Android, таких как клик по кнопке. События позволяют приложению реагировать на действия пользователя, например, нажатие на кнопку, ввод текста или выбор элемента из списка. Мы рассмотрели два основных способа обработки событий: через XML-разметку и через код Java (Code-Behind).

**2. Описание работы**

**2.1. Создание нового проекта**

1. **Создание проекта:**
   * Запустили Android Studio и создали новый проект, выбрав шаблон **Empty Activity**.
   * Указали имя проекта — **EventHandlingApp** и выбрали язык программирования — **Java**.
   * Установили минимальную версию SDK — **API 24: Android 7.0 (Nougat)**.
2. **Структура проекта:**
   * После создания проекта изучили структуру файлов:
     + **app/java/com.example.eventhandlingapp** — папка с Java-кодом.
     + **app/res/layout** — папка с XML-файлами, отвечающими за интерфейс приложения.
     + **app/res/values** — папка для хранения строковых ресурсов и констант.

**2.2. Добавление кнопки в интерфейс**

1. **Редактирование макета:**
   * В файле **activity\_main.xml** добавили кнопку с текстом "Нажми меня".
   * Пример кода:

xml

<Button

android:id="@+id/myButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Нажми меня"

android:layout\_centerInParent="true" />

1. **Настройка свойств кнопки:**
   * Установили уникальный идентификатор кнопки с помощью атрибута **android:id**.
   * Задали текст кнопки с помощью атрибута **android:text**. 

**2.3. Обработка события клика через XML**

1. **Добавление обработчика события:**
   * В XML-разметке добавили атрибут **android:onClick**для кнопки, указав имя метода, который будет вызван при нажатии.
   * Пример:

xml

<Button

android:id="@+id/myButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Нажми меня"

android:onClick="onButtonClick"

android:layout\_centerInParent="true" />

1. **Создание метода обработки события:**
   * В классе **MainActivity** создали метод **onButtonClick**, который будет вызываться при нажатии на кнопку.
   * Пример кода:

java

public void onButtonClick(View view) {

Toast.makeText(this, "Кнопка нажата!", Toast.LENGTH\_SHORT).show();

}

1. **Тестирование:**
   * Запустили приложение на эмуляторе и убедились, что при нажатии на кнопку появляется всплывающее сообщение (Toast).

**2.4. Обработка события клика через Code-Behind**

1. **Добавление кнопки в код:**
   * В классе **MainActivity** получили ссылку на кнопку с помощью метода **findViewById**.
   * Пример кода:

java

Button myButton = findViewById(R.id.myButton);

1. **Создание слушателя события:**
   * Создали объект **View.OnClickListener** и переопределили метод **onClick**.
   * Пример кода:

java

myButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Toast.makeText(MainActivity.this, "Кнопка нажата через Code-Behind!", Toast.LENGTH\_SHORT).show();

}

});

1. **Тестирование:**
   * Запустили приложение и убедились, что при нажатии на кнопку появляется всплывающее сообщение.

**2.5. Динамическое изменение текста кнопки**

1. **Изменение текста кнопки:**
   * В методе обработки события изменили текст кнопки с помощью метода **setText**.
   * Пример кода:

java

public void onButtonClick(View view) {

Button button = (Button) view;

button.setText("Нажато!");

Toast.makeText(this, "Кнопка нажата!", Toast.LENGTH\_SHORT).show();

}

1. **Тестирование:**
   * Запустили приложение и убедились, что текст кнопки изменяется после нажатия.

**2.6. Использование аргумента View**

1. **Передача аргумента View:**
   * В методе обработки события использовали аргумент **View** для получения ссылки на кнопку.
   * Пример кода:

java

public void onButtonClick(View view) {

Button button = (Button) view;

button.setText("Нажато!");

Toast.makeText(this, "Кнопка нажата!", Toast.LENGTH\_SHORT).show();

}

1. **Тестирование:**
   * Запустили приложение и убедились, что текст кнопки изменяется после нажатия.



**3. Результаты**

* Создан новый проект с использованием шаблона **Empty Activity**.
* Добавлена кнопка в интерфейс приложения.
* Реализована обработка события клика через XML и Code-Behind.
* Текст кнопки изменяется динамически после нажатия.
* Приложение успешно запущено на эмуляторе, все функции работают корректно.

**4. Выводы**

* Механизм событий позволяет приложению реагировать на действия пользователя.
* Обработку событий можно реализовать как через XML-разметку, так и через код Java.
* Освоены базовые навыки работы с событиями в Android.