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

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Кафедра прикладной информатики**

Лабораторная работа №2

ПО ДИСЦИПЛИНЕ «Программирование мобильных устройств»

НА ТЕМУ:

«**Разработка дизайна страницы с помощью языка XML. Менеджеры размещения**»

**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 №2. Разработка дизайна страницы с помощью языка XML. Менеджеры размещения**

**Цель работы:** Изучение XML-разметки для создания интерфейса приложения и освоение работы с менеджерами размещения.

**1. Введение**

В данной лабораторной работе мы изучили основы создания пользовательского интерфейса с использованием языка XML. XML-разметка позволяет декларативно описывать структуру интерфейса приложения, что делает процесс разработки более удобным и гибким. Также мы познакомились с различными менеджерами размещения, такими как **LinearLayout**, **RelativeLayout** и **GridLayout**, которые помогают организовать элементы на экране.

**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 — **MyApplication** 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.myapplication** — 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. Изучение XML-разметки**

1. **Основы XML:**
   * XML (Extensible Markup Language) — это язык разметки, который используется для описания структуры интерфейса приложения.
   * Каждый элемент интерфейса описывается с помощью тегов, например:

xml

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!" />

1. **Основные атрибуты элементов:**
   * **android:layout\_width**и **android:layout\_height**— определяют ширину и высоту элемента.
   * **android:text**— задаёт текст, отображаемый в элементе.
   * **android:id**— уникальный идентификатор элемента, который используется для доступа к нему из кода.

**2.3. Работа с менеджерами размещения**

1. **LinearLayout:**
   * **LinearLayout** — это менеджер размещения, который располагает элементы либо вертикально, либо горизонтально.
   * Пример использования:

xml

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 1" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 2" />

</LinearLayout>

1. **RelativeLayout:**
   * **RelativeLayout** — это менеджер размещения, который позволяет располагать элементы относительно друг друга или относительно родительского контейнера.
   * Пример использования:

xml

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 1"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 2"

android:layout\_below="@id/button1"

android:layout\_centerHorizontal="true" />

</RelativeLayout>

1. **GridLayout:**
   * **GridLayout** — это менеджер размещения, который организует элементы в виде сетки (строки и столбцы).
   * Пример использования:

xml

<GridLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:rowCount="2"

android:columnCount="2">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 1"

android:layout\_row="0"

android:layout\_column="0" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 2"

android:layout\_row="0"

android:layout\_column="1" />

</GridLayout>

**2.4. Создание интерфейса приложения**

1. **Создание макета страницы:**
   * В файле **activity\_main.xml** создали макет страницы с использованием **LinearLayout**.
   * Добавили несколько элементов управления, таких как **TextView**, **Button** и **EditText**.
2. **Настройка свойств элементов:**
   * Для каждого элемента настроили свойства, такие как ширина, высота, текст, отступы и выравнивание.
   * Пример:

xml

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Введите ваше имя:"

android:layout\_marginTop="16dp" />

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Имя"

android:layout\_marginTop="8dp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Отправить"

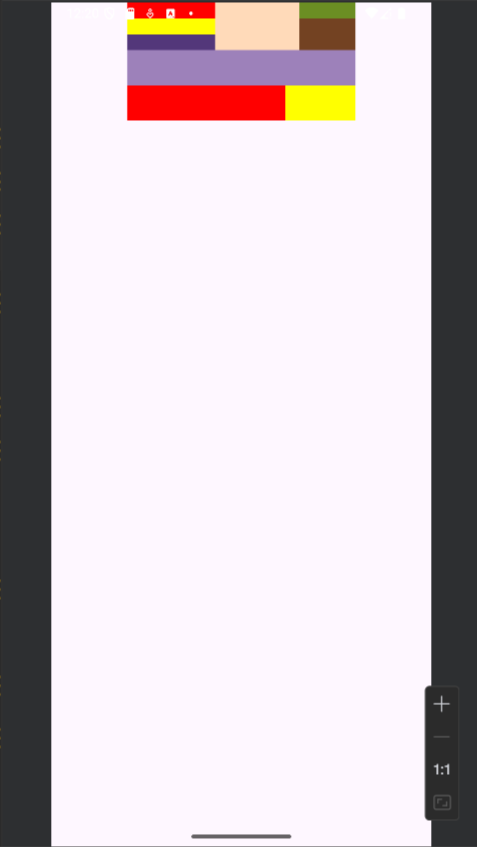
android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="16dp" />

1. **Использование вложенных менеджеров размещения:**
   * Для создания более сложных интерфейсов использовали вложенные менеджеры размещения. Например, внутри **LinearLayout** добавили **RelativeLayout** для организации элементов более гибко.



**2.5. Тестирование интерфейса**

1. **Запуск приложения:**
   * Запустили приложение на эмуляторе и убедились, что интерфейс отображается корректно.
   * Проверили, что все элементы управления работают как ожидалось (например, кнопки реагируют на нажатия). 

**3. Результаты**

* Создан новый проект с использованием шаблона **Empty Activity**.
* Изучены основы XML-разметки для создания интерфейса приложения.
* Освоены основные менеджеры размещения: **LinearLayout**, **RelativeLayout** и **GridLayout**.
* Создан макет страницы с использованием **LinearLayout** и добавлены элементы управления.
* Приложение успешно запущено на эмуляторе, интерфейс отображается корректно.

**4. Выводы**

* XML-разметка предоставляет удобный способ описания интерфейса приложения.
* Менеджеры размещения позволяют гибко организовывать элементы на экране.
* Освоены базовые навыки создания пользовательского интерфейса в Android Studio.