|  |  |
| --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования «Полоцкий государственный университет» | |
|  | Факультет информационных технологий  Кафедра технологий программирования |
| Лабораторная работа №2 по курсу «Программирование мобильных систем»  «Использование элементов пользовательского интерфейса. Использование ListView» | |
| Выполнил | Студент гр. 21-ИТ-1  Чиникайло А.П. |
| Проверил | Преподаватель  Васильева Д.М. |
| Полоцк, 2024г. | |

**Цель работы:** ознакомиться с различными типами элементов интерфейса. Научиться использовать элементы пользовательского интерфейса в разрабатываемых приложениях. Ознакомиться с использованием адаптеров в Android. Научиться использовать элемент ListView в разрабатываемых приложениях.

**Ход работы**

**Задание:**

1. Выполнить размещение элементов пользовательского интерфейса на одной из activity приложения, используя LinearLayout.

2. Создать activity с использованием RelativeLayout.

3. Добавить нужные строки в файл res/values/strings.xml

4. Отредактируйте файл AndroidManifest.xml, чтобы изменить тему, используемую приложением.

5. Создать activity с использованием TabWidget

6. Создать activity с использованием WebView

7. Создать новый Android проект ListViewSample

Листинг макета с использованием LinearLayout представлен в листинге

Листинг 1 – activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center"

tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Это главная Activity"

android:textSize="24sp"

android:textStyle="bold"

android:layout\_marginBottom="20dp"/>

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Перейти к другой Activity" />

</LinearLayout>

Листинг макета с использованием RelativeLayout представлен в листинге 2.

Листинг 2 – activity\_second.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 tools:context=".SecondActivity">  
  
 <TextView  
 android:id="@+id/label"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/label\_text" />  
  
 <EditText  
 android:id="@+id/entry"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/label"  
 android:background="@android:drawable/editbox\_background"  
 android:hint="@string/entry\_hint"/>  
  
 <Button  
 android:id="@+id/ok"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_below="@id/entry"  
 android:layout\_marginStart="10dip"  
 android:text="@android:string/ok" />  
  
 <Button  
 android:id="@+id/cancel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignTop="@id/ok"  
 android:layout\_toStartOf="@id/ok"  
 android:text="@android:string/cancel" />  
  
</RelativeLayout>

Листинг класса TabWidgetActivity представлен в листинге 3.

Листинг 3 – TabWidgetActivity.kt

package com.example.pms\_lab\_2

import android.os.Bundle

import android.view.LayoutInflater

import android.widget.TabHost

import androidx.activity.ComponentActivity

class TabWidgetActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_tab\_widget)

setupTabs()

}

private fun setupTabs() {

val tabHost = findViewById<TabHost>(android.R.id.tabhost)

tabHost.setup()

val inflater = LayoutInflater.from(this)

val tab1View = inflater.inflate(R.layout.tab1\_content, null)

val tab2View = inflater.inflate(R.layout.tab2\_content, null)

val tab1 = tabHost.newTabSpec("Tab 1")

tab1.setIndicator("Tab 1")

tab1.setContent { tab1View }

tabHost.addTab(tab1)

val tab2 = tabHost.newTabSpec("Tab 2")

tab2.setIndicator("Tab 2")

tab2.setContent { tab2View }

tabHost.addTab(tab2)

tabHost.currentTab = 0

}

}

Листинг класса WebViewActivity представлен в листинге 4.

Листинг 4 – WebViewActivity.kt

package com.example.pms\_lab\_2

import android.os.Bundle

import android.webkit.WebView

import androidx.activity.ComponentActivity

class WebViewActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

val webView = WebView(this)

setContentView(webView)

val url = "https://www.google.com"

webView.loadUrl(url)

}

}

Листинг класса ListActivity представлен в листинге 5.

Листинг 5 – ListActivity.kt

package com.example.listviewsample

import android.os.Bundle

import android.widget.ArrayAdapter

import android.widget.ListView

import android.widget.Toast

import androidx.activity.ComponentActivity

import com.example.pms\_lab\_2.R

class ListActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

val stationsArray = resources.getStringArray(R.array.stations)

val adapter = ArrayAdapter(this, android.R.layout.simple\_list\_item\_1, stationsArray)

val listView = ListView(this)

listView.adapter = adapter

setContentView(listView)

listView.setOnItemClickListener { parent, view, position, id ->

val selectedItem = parent.getItemAtPosition(position).toString()

Toast.makeText(applicationContext, selectedItem, Toast.LENGTH\_LONG).show()

}

}

}

**Результат выполнения работы:**

Размещение элементов пользовательского интерфейса на одной из activity приложения, используя RelativeLayout, представлено на рисунке 1.

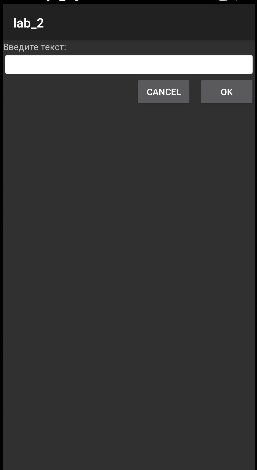


Рисунок 2 – Размещение элементов, используя RelativeLayout  
  
Activity с использованием TabWidget представлена на рисунке 2.

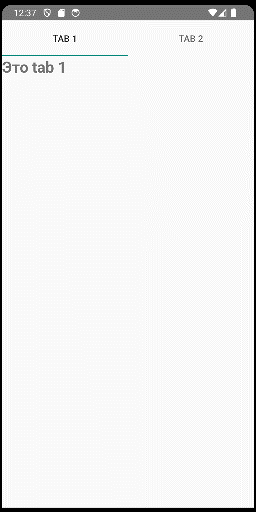


Рисунок 2 – Activity с TabWidget

ListActivity представлена на рисунке 3.

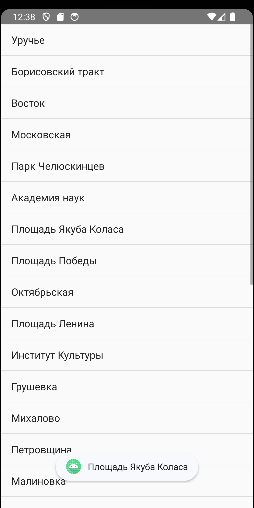


Рисунок 3 – ListActivity

**Вывод:** в результате выполнения лабораторной работы мы ознакомились с различными типами элементов интерфейса. Научились использовать элементы пользовательского интерфейса в разрабатываемых приложениях. Ознакомились с использованием адаптеров в Android. Научились использовать элемент ListView в разрабатываемых приложениях.

**Контрольные вопросы:**

1. Представление: это базовый элемент пользовательского интерфейса в Android, представляющий отображаемый объект, такой как кнопка, текстовое поле или изображение.
2. Основные представления включают TextView, EditText, Button, ImageView, CheckBox, RadioButton, и др. Они используются для отображения информации и взаимодействия с пользователем.
3. Основные виды разметки: LinearLayout, RelativeLayout, FrameLayout, TableLayout, GridLayout, ConstraintLayout. Разметка определяет структуру и расположение представлений на экране.
4. Разметка экрана Android приложения реализуется в XML-файлах, которые находятся в каталоге res/layout вашего проекта.
5. Да, возможно использование нескольких видов разметки на одной активности. Вы можете вкладывать одну разметку в другую, создавая сложные макеты интерфейса.
6. Адаптер это компонент, который связывает данные с представлениями в пользовательском интерфейсе. Он предоставляет данные представлениям и управляет их отображением.
7. Основные виды адаптеров: ArrayAdapter (для работы с массивом или списком данных), CursorAdapter (для работы с данными из базы данных), BaseAdapter (абстрактный класс, который можно расширить для создания собственного адаптера), RecyclerView.Adapter (для работы с RecyclerView, более гибкий и производительный).
8. Для работы с результатом запроса СУБД часто используется CursorAdapter.
9. Представление ListView - это список вертикально расположенных представлений, который позволяет прокручивать содержимое, когда список становится длинным.
10. Внешний ресурс, содержащий массив данных, обычно хранится в ресурсах приложения в виде массива строк в XML-файле.