МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**ОТЧЕТ ПО**

**ЛАБОРАТОРНОЙ РАБОТЕ №2**

Дисциплина: **«**Программирование для мобильных устройств**»**

Тема: **«**Использование элементов пользовательского интерфейса. Использование ListView**»**

ВЫПОЛНИЛ студент группы 17-ИТ-4

Трубкин А.Д.

ПРОВЕРИЛ Лукьянов А.О.

Полоцк 2020 г.

**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.

**Теоретическая часть**

**Основные понятия и связи между ними**

• Представления (View) являются базовым классом для все визуальных элементов UI (элементов управления (Control) и виджетов (Widget)). Все эти элементы, в том числе и разметка (Layout), являются расширениями класса View.

• Группы Представлений (ViewGroup) – потомки класса View; могут содержать в себе несколько дочерних Представлений. Расширение класса ViewGroup используется для создания сложных Представлений, состоящих из взаимосвязанных компонентов. Класс ViewGroup также является базовым для различных Разметок (Layout).

• Активности (Activity) – отображаемые экраны или окна (с точки зрения построения UI), являются «андроидными эквивалентами» форм. Для отображения UI Активности используют Представления (как правило, Разметки (Layout)).

Для создания приложений с уникальным интерфейсом разработчику иногда приходится расширять и модифицировать стандартные Представления, комбинируя их со стандартными.

**Представления (View)**

Android предоставляет разработчику возможность использования богатого набора готовых Представлений:

• **TextView.** Стандартный элемент, предназначенный для вывода текста. Поддерживает многострочное отображение, форматирование и автоматический перенос.

• **EditText.** Редактируемое поле для ввода текста. Поддерживает многострочный ввод, перенос слов на новую строку и и текст подсказки.

• **ListView.** Группа представлений (ViewGroup), которая формирует вертикальный список элементов, отображая их в виде строк внутри списка. Простейший объект ListView использует TextView для вывода на экран значений toString(), принадлежащих элементом массива.

• **Spinner.** Составной элемент, отображающий TextView в сочетании с соответствующим ListView, которое позволяет выбрать элемент списка для отображения в текстовой строке. Сама строка состоит из объекта TextView и кнопки, при нажатии на которую всплывает диалог выбора. Внешне этот элемент напоминает тэг <SELECT> в HTML.

• **Button.** Стандартная кнопка, которую можно нажимать.

• **CheckBox.** Кнопка, имеющая два состояния. Представлена в виде отмеченного или неотмеченного флажка («галки»).

• **RadioButton.** «Радиокнопка», позволяет выбрать только один из нескольких вариантов.

• **ViewFlipper.** Группа представлений (ViewGroup), позволяющая определить набор элементов и горизонтальную строку, в которой может выводиться только одно Представление (View). При этом переходы между отображающимися элементами осуществляются с помощью анимации.

Android предлагает и более продвинутые реализации Представлений, такие, как элементы для выбора даты и времени, поля ввода с автоматическим дополнением, галереи, вкладки и даже карты (MapView).

Более полный список поддерживаемых системой Представлений можно увидеть по адресу http://developer.android.com/guide/tutorials/views/index.html

Кроме готовых Представлений, разработчик, при необходимости, может создавать собственные, расширяя класс View или его подклассы.

**Разметка (Layout)**

Разметка (Layout) является расширением класса ViewGroup и используется для размещения дочерних компонентов на экране устройства. Используя вложенные Разметки, можно создавать пользовательские интерфейсы любой сложности.

Наиболее часто используемые виды Разметки:

• **FrameLayout.** Самая простая разметка, прикрепляет каждое новое дочернее Представление к левому верхнему углу экрана, накладывая новый элемент на предыдущий, заслоняя его.

• **LinearLayout.** Помещает дочерние Представления в горизонтальный или вертикальный ряд. Вертикальная разметка представляет собой колонку, а горизонтальная – строку с элементами. Данная разметка позволяет задавать не только размеры, но и «относительный вес» дочерних элементов, благодаря чему можно гибко контролировать их размещение на экране.

• **RelativeLayout.** Наиболее гибкий среди стандартных видов разметки. Позволяет указывать позиции дочерних Представлений относительно границ свободного пространства и других Представлений.

• **TableLayout.** Позволяет размещать дочерние Представления внутри ячеек «сетки», состоящей из строк и столбцов. Размеры ячеек могут оставаться постоянными или автоматически растягиваться при необходимости.

• **Gallery.** Представляет элементы в виде прокручиваемого горизонтального списка (обычно графические элементы).

**Адаптеры в Android**

Адаптеры в Android являются связующими классами между данными приложения иПредставлениями. Адаптер отвечает за создание дочерних Представлений,отображающими каждый элемент внутри родительского виджета, а также обеспечиваетдоступ к исходным данным, используемым приложением. Представления, использующиепривязку к Адаптеру, должны быть потомками абстрактного класса AdapterView.

Android содержит набор стандартных Адаптеров, которые доставляют данные встандартные виджеты пользовательского интерфейса. Двумя наиболее полезными и частоиспользуемыми Адаптерами являются ArrayAdapter и SimpleCursorAdapter.

**ArrayAdapter** использует механизм обобщенных типов (generics) языка Java для привязкиродительского класса AdapterView к массиву объектов указанного типа. По умолчаниюArrayAdapter использует метод toString() для каждого элемента в массиве, чтобы создать изаполнить текстовыми данными виджеты TextView.

**SimpleCursorAdapter** привязывает указанное в разметке Представление к столбцамКурсора, ставшего результатом запроса к СУБД или Контент-Провайдеру. Для егоиспользования требуется описать разметку в формате XML, а затем привязать каждыйстолбец к Представлениям из этой разметки. Адаптер создаст Представления для каждойзаписи из Курсора и наполнит их данными из соответствующих столбцов.

**SimpleAdapter** позволяет привязать ListView к списку ArrayList, содержащему объектытипа Map (ассоциативные массивы, содержащие пары «ключ-значение»). Для каждоготакого объекта при отображении используется один элемент из ListView. Как и дляSimpleCursorAdapter, для отображения применяется XML-разметка, к элементам которойпривязываются члены каждого объекта типа Map.

**Выполнение лабараторной работы**

В данной лабораторной работе было выполнено и скомпилировано приложения ,в котором будут реализованы TabWidget,Webiew и ListView.

В начале реализуем расположение панелей в первом приложении:

Листинг1 – activity\_main

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

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

*android:layout\_width="fill\_parent"*

*android:layout\_height="fill\_parent" >*

*<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\_alignParentRight="true"*

*android:layout\_below="@id/entry"*

*android:layout\_marginLeft="10dip"*

*android:text="@android:string/ok" />*

*<Button*

*android:id="@+id/can"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:layout\_alignTop="@id/ok"*

*android:layout\_toLeftOf="@id/ok"*

*android:text="@android:string/cancel" />*

*<Button*

*android:id="@+id/list"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:layout\_below="@+id/ok"*

*android:text="list" />*

*</RelativeLayout>*

Листинг2 – activity\_tab

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

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

*xmlns:app="http://schemas.android.com/apk/res-auto"*

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

*android:layout\_width="match\_parent"*

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

*tools:context=".TabActivity">*

*<TabHost*

*android:id="@android:id/tabhost"*

*android:layout\_width="368dp"*

*android:layout\_height="495dp"*

*tools:layout\_editor\_absoluteY="8dp"*

*tools:layout\_editor\_absoluteX="8dp"*

*tools:ignore="MissingConstraints">*

*<LinearLayout*

*android:layout\_width="match\_parent"*

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

*android:orientation="vertical">*

*<TabWidget*

*android:id="@android:id/tabs"*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content">*

*</TabWidget>*

*<FrameLayout*

*android:id="@android:id/tabcontent"*

*android:layout\_width="match\_parent"*

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

*<LinearLayout*

*android:id="@+id/tab1"*

*android:layout\_width="match\_parent"*

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

*<Button*

*android:id="@+id/button1"*

*style="@style/Widget.AppCompat.Button.Colored"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:background="@android:drawable/dialog\_holo\_dark\_frame"*

*android:text="Start" />*

*</LinearLayout>*

*<LinearLayout*

*android:id="@+id/tab2"*

*android:layout\_width="match\_parent"*

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

*<WebView*

*android:id="@+id/web\_view"*

*android:layout\_width="368dp"*

*android:layout\_height="495dp"*

*tools:layout\_editor\_absoluteX="8dp"*

*tools:layout\_editor\_absoluteY="8dp" />*

*</LinearLayout>*

*</FrameLayout>*

*</LinearLayout>*

*</TabHost>*

*</RelativeLayout>*

Листинг3 – activity\_web

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

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

*xmlns:app="http://schemas.android.com/apk/res-auto"*

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

*android:layout\_width="match\_parent"*

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

*tools:context=".WebActivity">*

*<WebView*

*android:id="@+id/web\_view"*

*android:layout\_width="368dp"*

*android:layout\_height="495dp"*

*tools:layout\_editor\_absoluteX="8dp"*

*tools:layout\_editor\_absoluteY="8dp"*

*tools:ignore="MissingConstraints" />*

*</RelativeLayout>*

Листинг4 – activity\_list

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

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

*xmlns:app="http://schemas.android.com/apk/res-auto"*

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

*android:layout\_width="match\_parent"*

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

*tools:context=".ListActivity">*

*<ListView*

*android:id="@android:id/list"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*/>*

*</RelativeLayout>*

Листинг5–MainActivity

*package com.example.myapp;*

*import android.content.Intent;*

*import android.os.Bundle;*

*import android.view.View;*

*import android.widget.Button;*

*import android.widget.Toast;*

*import androidx.appcompat.app.AppCompatActivity;*

*public class MainActivity extends AppCompatActivity implements View.OnClickListener{*

*@Override*

*protected void onCreate(Bundle savedInstanceState) {*

*super.onCreate(savedInstanceState);*

*setContentView(R.layout.activity\_main);*

*Button btnC = (Button) findViewById(R.id.can);*

*btnC.setOnClickListener(this);*

*Button btnOk = (Button) findViewById(R.id.ok);*

*btnOk.setOnClickListener(this);*

*Button btnList = (Button) findViewById(R.id.list);*

*btnList.setOnClickListener(this);*

*}*

*public void onClick(View view)*

*{*

*switch (view.getId())*

*{*

*case R.id.can:*

*Intent intent = new Intent(this, TabActivity.class);*

*startActivity(intent);*

*Toast.makeText(this, "main", Toast.LENGTH\_LONG).show();*

*break;*

*case R.id.ok:*

*Intent intent1 = new Intent(this, WebActivity.class);*

*startActivity(intent1);*

*break;*

*case R.id.list:*

*Intent intent2 = new Intent(this, ListActivity.class);*

*startActivity(intent2);*

*break;*

*default:*

*break;*

*}*

*}*

*}*

Листинг 6 – TabActivity

*package com.example.myapp;*

*import android.os.Bundle;*

*import android.webkit.WebView;*

*import android.widget.TabHost;*

*import android.widget.Toast;*

*import androidx.appcompat.app.AppCompatActivity;*

*public class TabActivity extends AppCompatActivity {*

*@Override*

*protected void onCreate(Bundle savedInstanceState) {*

*Toast.makeText(this, "1", Toast.LENGTH\_LONG).show();*

*super.onCreate(savedInstanceState);*

*setContentView(R.layout.activity\_tab);*

*WebView webView = (WebView) findViewById(R.id.web\_view);*

*TabHost tabHost = (TabHost) findViewById(android.R.id.tabhost);*

*tabHost.setup();*

*TabHost.TabSpec tabSpec;*

*tabSpec = tabHost.newTabSpec("tag1");*

*tabSpec.setContent(R.id.tab1);*

*tabSpec.setIndicator("1 Tab");*

*tabHost.addTab(tabSpec);*

*tabSpec = tabHost.newTabSpec("tag2");*

*tabSpec.setContent(R.id.tab2);*

*tabSpec.setIndicator("2 Tab");*

*tabHost.addTab(tabSpec);*

*webView.loadUrl("https://google.com/");*

*tabHost.setCurrentTabByTag("tag1");*

*}*

*}*

Листинг7 – WebActivity

*package com.example.myapp;*

*import android.os.Bundle;*

*import android.widget.Toast;*

*import android.webkit.WebView;*

*import androidx.appcompat.app.AppCompatActivity;*

*public class WebActivity extends AppCompatActivity {*

*@Override*

*protected void onCreate(Bundle savedInstanceState) {*

*super.onCreate(savedInstanceState);*

*setContentView(R.layout.activity\_web);*

*Toast.makeText(this, "OK", Toast.LENGTH\_LONG).show();*

*WebView webView = (WebView) findViewById(R.id.web\_view);*

*webView.loadUrl("https://google.com/");*

*}*

*}*

Листинг 8 – ListActivity

*package com.example.myapp;*

*import android.content.Context;*

*import android.content.res.Resources;*

*import android.os.Bundle;*

*import android.view.View;*

*import android.widget.AdapterView;*

*import android.widget.ArrayAdapter;*

*import android.widget.ListView;*

*import android.widget.TextView;*

*import android.widget.Toast;*

*import com.example.myapp.R;*

*public class ListActivity extends android.app.ListActivity {*

*@Override*

*protected void onCreate(Bundle savedInstanceState) {*

*super.onCreate(savedInstanceState);*

*setContentView(R.layout.activity\_list);*

*Resources r = getResources();*

*String[] stationsArray = r.getStringArray(R.array.stations);*

*ArrayAdapter<String> aa = new ArrayAdapter<String>(this, R.layout.list\_item, stationsArray);*

*setListAdapter(aa);*

*ListView lv = getListView();*

*lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {*

*public void onItemClick(AdapterView<?> parent, View v, int position, long id) {*

*CharSequence text = ((TextView) v).getText();*

*int duration = Toast.LENGTH\_LONG;*

*Context context = getApplicationContext();*

*Toast.makeText(context, text, duration).show();*

*}*

*});*

*}*

*}*

Листинг9 – Установка текста для приложения

*<resources>*

*<string name="label\_text">Р’РІРµРґРёС‚Рµ С‚РµРєСЃС‚:</string>*

*<string name="entry\_hint">РџРѕР»Рµ РІРІРѕРґР°</string>*

*<string name="app\_name"> HelloAndroidWorld </string>*

*<string name="title\_activity\_tab">TabActivity</string>*

*<string name="tab\_text\_1">Tab 1</string>*

*<string name="tab\_text\_2">Tab 2</string>*

*</resources>*

Листинг10 – list\_item

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

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

*android:layout\_width="fill\_parent"*

*android:layout\_height="fill\_parent"*

*android:padding="10dp"*

*android:textSize="16sp" >*

*</TextView>*

Листинг 11 – arrays.xml

*<?xml version="1.0" encoding="utf-8"?>  
<resources>  
<string-array name="stations">  
<item>Уручье</item>  
<item>Борисовский тракт</item>  
<item>Восток</item>  
<item>Московская</item>  
<item>Парк Челюскинцев</item>  
<item>Академия наук</item>  
<item>Площадь Якуба Коласа</item>  
<item>Площадь Победы</item>  
<item>Октябрьская</item>  
<item>Площадь Ленина</item>  
<item>Институт Культуры</item>  
<item>Грушевка</item>  
<item>Михалово</item>  
<item>Петровщина</item>  
<item>Малиновка</item>  
<item>Каменная Горка</item>  
<item>Кунцевщина</item>  
<item>Спортивная</item>  
<item>Пушкинская</item>  
<item>Молодёжная</item>  
<item>Фрунзенская</item>  
<item>Немига</item>  
<item>Купаловская</item>  
<item>Первомайская</item>  
<item>Пролетарская</item>  
<item>Тракторный завод</item>  
<item>Партизанская</item>  
<item>Автозаводская</item>  
<item>Могилёвская</item>  
</string-array>  
</resources>*

**Резулътат проделанной работы:**

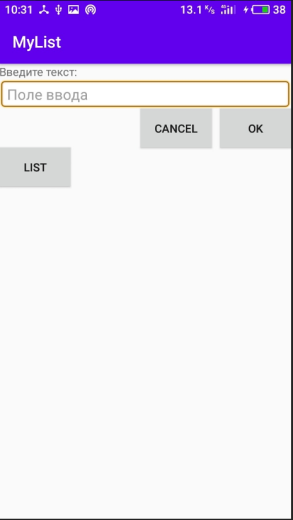
****

Рисунок 1 –Основное окно

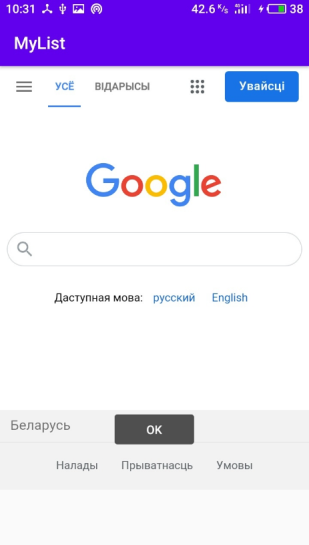
****

Рисунок 2 –WebActivity

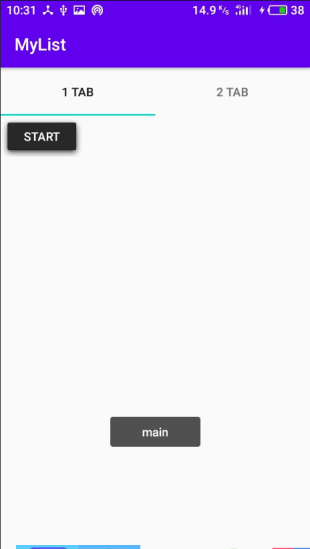
****

Рисунок 3 – TabActivity 1Tab

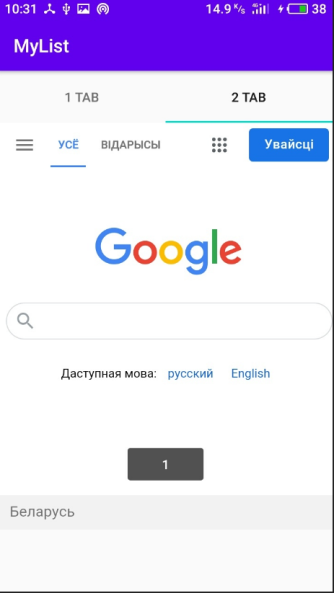
****

Рисунок 4 – TabActivity 2Tab

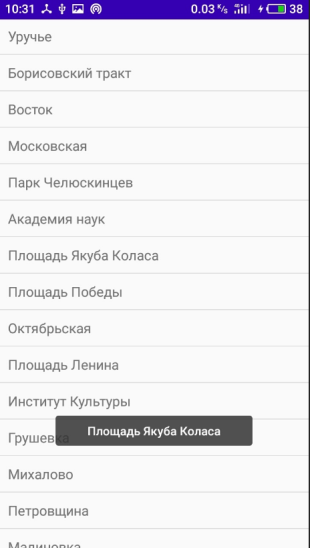
****

Рисунок 5 – ListActivity

**Вывод**: В ходе лабараторной работы было разработано приложение показывающее необходимые активности в зависимости от нажатия.