**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №5**

**з курсу**

**«Архітектура мобільних операційних систем і пристроїв»**

*Студентки 4 курсу*

*групи ПП-42*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Матвіїв А.Ю.

*Викладач:*

Краснощок В.М.

**Київ – 2024**

***Тема:*** Робота із віджетами та керуючими елементами. а

***Мета:*** Ознайомитись з використанням віджетів TabWidget, WebView, ListView. Навчитися використовувати в інтерфейсі користувача різні керуючі елементи: кнопки із зображеннями, радіокнопки, чекбокси та ін.

***Хід роботи:***

В результаті виконання роботи було створено 4 проекти: TabWidgetSample, WebViewSample, ControlsSample, ListViewSample.

1. TabWidgetSample

У вкладці проекту, де знаходиться MainActivity створимо нові 3 активності і дамо назву ClassesActivity, StudentActivity та TeacherActivity. Використаємо даний код:  
**MainActivity.kt**

package com.example.laba5  
  
import android.app.TabActivity  
import android.content.Intent  
import android.content.res.Resources  
import android.os.Bundle  
import android.widget.TabHost  
  
@Suppress("deprecation")  
class MainActivity : TabActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main)  
  
 val res: Resources = resources  
 val tabHost: TabHost = tabHost  
  
   
 var intent = Intent().setClass(this, StudentsActivity::class.java)  
 var spec = tabHost.newTabSpec("students")  
 .setIndicator(res.getString(R.string.tab1\_indicator))  
 .setContent(intent)  
 tabHost.addTab(spec)  
  
   
 intent = Intent().setClass(this, TeachersActivity::class.java)  
 spec = tabHost.newTabSpec("teachers")  
 .setIndicator(res.getString(R.string.tab2\_indicator))  
 .setContent(intent)  
 tabHost.addTab(spec)  
  
   
 intent = Intent().setClass(this, ClassesActivity::class.java)  
 spec = tabHost.newTabSpec("classes")  
 .setIndicator(res.getString(R.string.tab3\_indicator))  
 .setContent(intent)  
 tabHost.addTab(spec)  
  
 tabHost.currentTab = 0   
 }  
}

**ClassesActivity.kt**

package com.example.laba5  
  
import android.os.Bundle  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
  
class ClassesActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val contentText = getString(R.string.*tab3\_content*)  
 val textView = TextView(this)  
 textView.*text* = contentText  
 setContentView(textView)  
 }  
}

**StudentActivity.kt**

package com.example.laba5  
  
import android.os.Bundle  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
  
class StudentsActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val contentText = getString(R.string.*tab1\_content*)  
 val textView = TextView(this)  
 textView.*text* = contentText  
 setContentView(textView)  
 }  
}

**TeacherActivity.kt**

package com.example.laba5  
  
import android.os.Bundle  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
  
class TeachersActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val contentText = getString(R.string.tab2\_content)  
 val textView = TextView(this)  
 textView.text = contentText  
 setContentView(textView)  
 }  
}

Для кожної вкладки створимо окремі файли активності з розширенням .xml:

**activity\_main.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@android:id/tabhost"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="5dp">  
  
 <TabWidget  
 android:id="@android:id/tabs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <FrameLayout  
 android:id="@android:id/tabcontent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="5dp" />  
 </LinearLayout>  
</TabHost>

**activity\_classes.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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=".ClassesActivity">  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_students.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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=".StudentsActivity">  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_teachers.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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=".TeachersActivity">  
  
</androidx.constraintlayout.widget.ConstraintLayout>

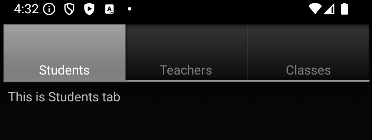
В **AndroidManifest.xml** добавим код:

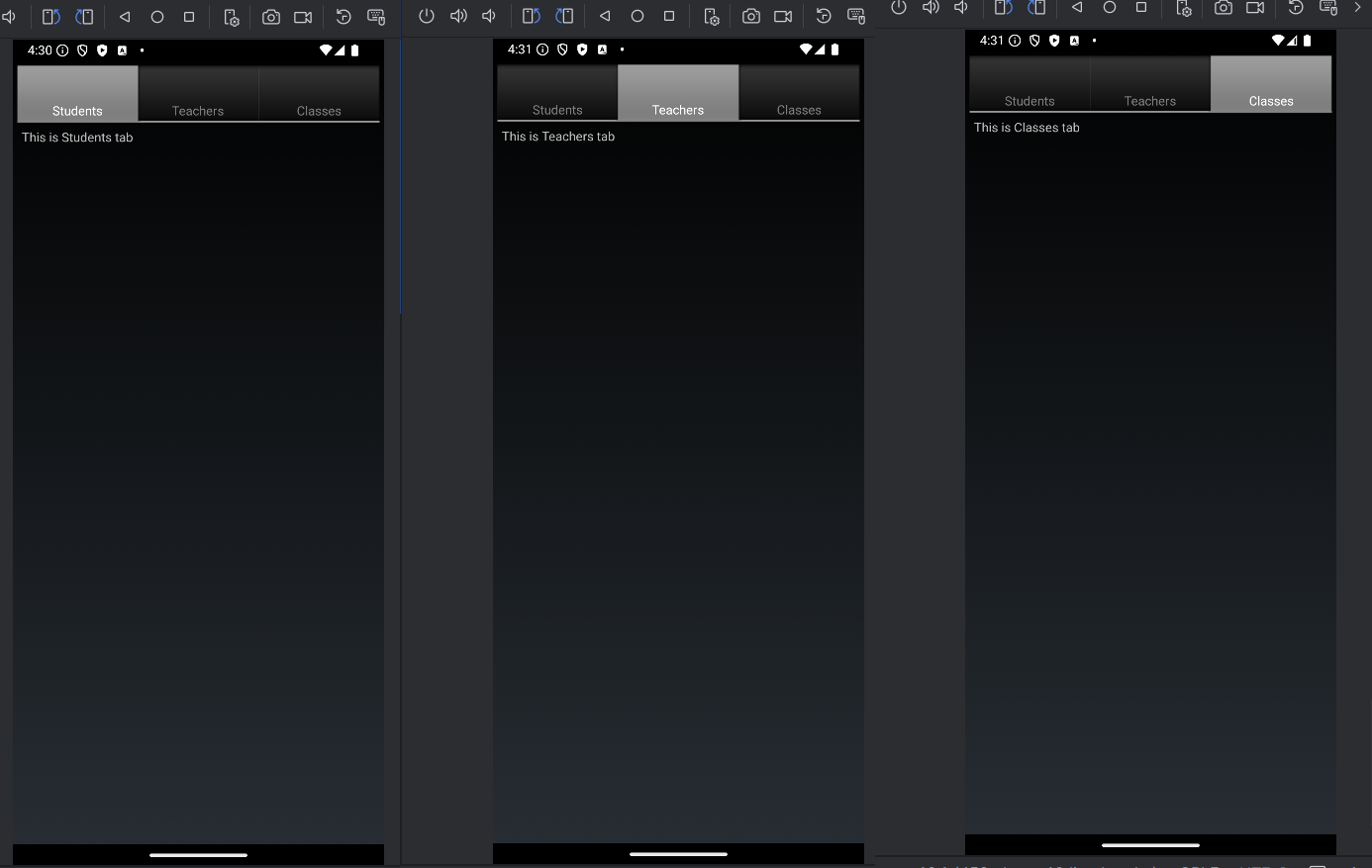
<activity android:name=".StudentsActivity" android:exported="true" />  
  
<activity android:name=".TeachersActivity" android:exported="true" />  
  
<activity android:name=".ClassesActivity" android:exported="true" />

Також потрібно забезпечити працювання додатку на різних мовах. По замовчуванню стоїть англійський інтерфейс. Додамо ще українську. Для цього в папці values створимо нову папку values-uk, і добавимо файл string-uk.xml:

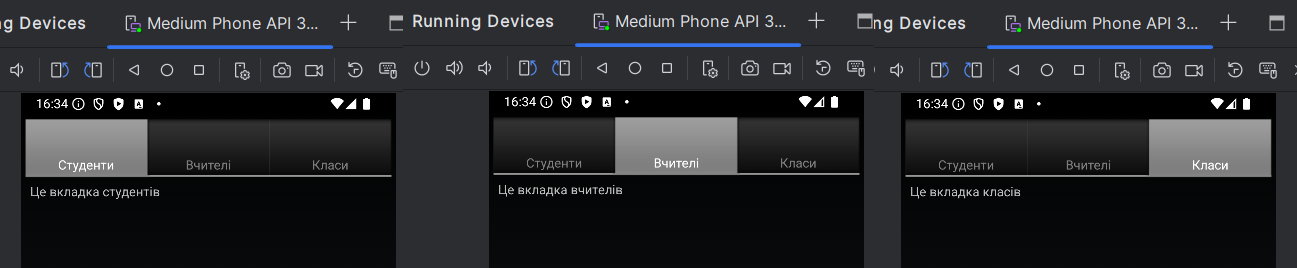
<?xml version="1.0" encoding= "utf-8"?>  
<resources>  
 <string name="app\_name">TabWidgetSample</string>  
 <string name="tab1\_indicator">Студенти</string>  
 <string name="tab2\_indicator">Вчителі</string>  
 <string name="tab3\_indicator">Класи</string>  
 <string name="tab1\_content">Це вкладка студентів</string>  
 <string name="tab2\_content">Це вкладка вчителів</string>  
 <string name="tab3\_content">Це вкладка класів</string>  
  
</resources>

Результат роботи:





Український інтерфейс:



1. WebViewSample

**MainActivity.kt:**

package com.example.webviewsample  
  
import android.content.Intent  
import android.os.Bundle  
import androidx.appcompat.app.AppCompatActivity  
import android.widget.Button  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
   
 val openWebViewButton = findViewById<Button>(R.id.*openWebViewButton*)  
 openWebViewButton.setOnClickListener **{** val intent = Intent(this, WebViewSampleActivity::class.*java*)  
 startActivity(intent)  
 **}** }  
}

**WebViewSampleActivity.kt:**

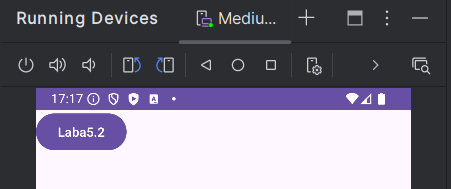
package com.example.webviewsample  
  
import android.os.Bundle  
import android.webkit.WebView  
import android.webkit.WebViewClient  
import androidx.appcompat.app.AppCompatActivity  
  
class WebViewSampleActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_web\_view\_sample*)  
  
 val webView = findViewById<WebView>(R.id.*webview*)  
 webView.*webViewClient* = WebViewClient()  
 webView.*settings*.*javaScriptEnabled* = true  
 webView.loadUrl("https://www.google.com.ua")  
 }  
}

**activity\_main.xml:**

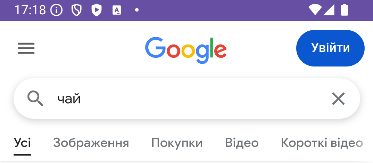
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <Button  
 android:id="@+id/openWebViewButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Laba5.2" />  
</LinearLayout>

**activity\_web\_view\_sample.xml:**

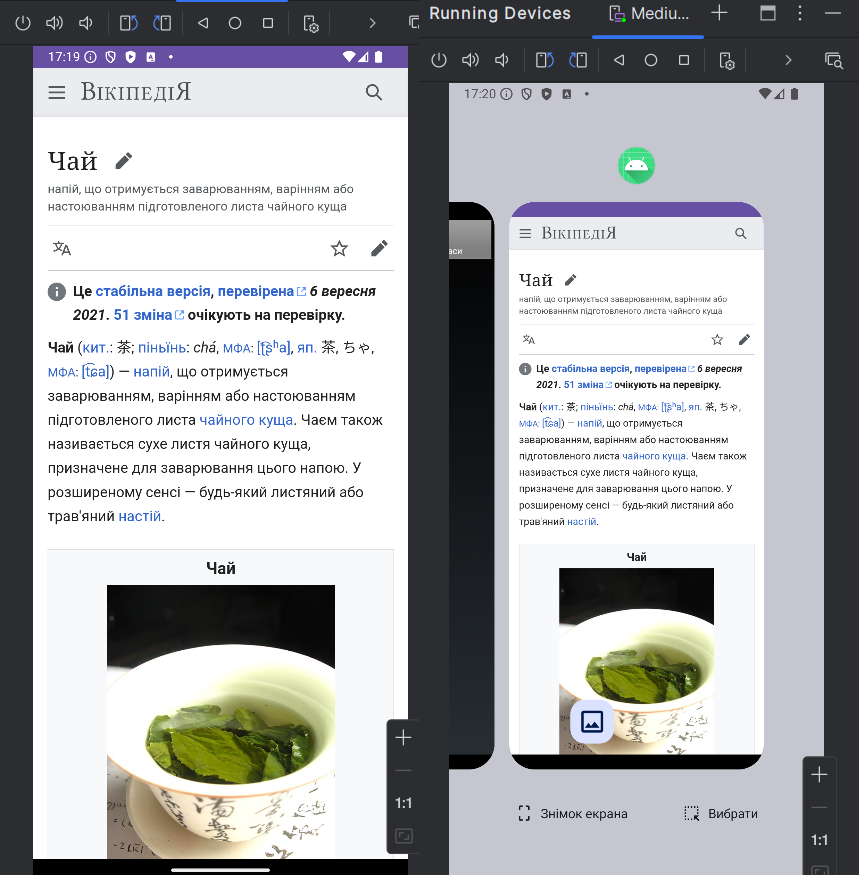
<?xml version="1.0" encoding="utf-8"?>  
<WebView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/webview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />

Результат роботи:  
Є кнопка при натисканні на неї програма переходить в гугл пошук і вся робота залишається в застосунку:  


Введемо запит на пошук:



Перейдемо на будь-який сайт:



Бачимо, що веб-сторінка все ще находиться в додатку.

1. ListViewSample

**MainActivity.kt:**

package com.example.listviewsample  
  
import android.app.ListActivity  
import android.os.Bundle  
import android.widget.AdapterView.OnItemClickListener  
import android.widget.ArrayAdapter  
import android.widget.TextView  
import android.widget.Toast  
  
class MainActivity : ListActivity() {  
 public override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val r = resources  
 val stationsArray = r.getStringArray(R.array.stations)  
 val aa = ArrayAdapter(this, R.layout.list\_item, stationsArray)  
 listAdapter = aa  
 val lv = listView  
 val duration = Toast.LENGTH\_LONG  
 val context = applicationContext  
 val toast = Toast.makeText(context, intent.action, duration)  
 toast.show()  
 lv.onItemClickListener =  
 OnItemClickListener **{** parent, v, position, id **->** val text = (v as TextView).text  
 val duration = Toast.LENGTH\_LONG  
 val context = applicationContext  
 Toast.makeText(context, text, duration).show()  
 **}** }  
}

**activity\_main.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

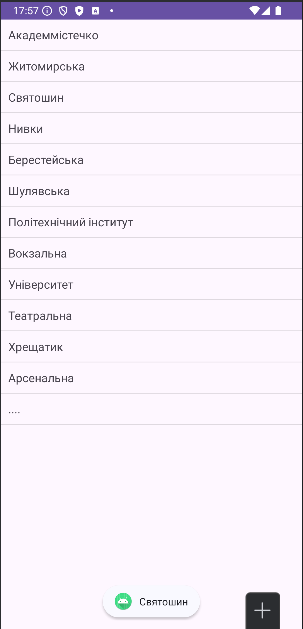
**list\_item.xml:**

<?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>

**arrays.xml:**

<?xml version="1.0" encoding= "utf-8"?>  
<resources>  
 <string-array name="stations">  
 <item >Академмістечко</item >  
 <item >Житомирська</item >  
 <item>Святошин</item>  
 <item>Нивки </item>  
 <item >Берестейська</item >  
 <item >Шулявська</item >  
 <item>Політехнічний інститут </item>  
 <item >Вокзальна</item >  
 <item> Університет</item>  
 <item > Театральна</item >  
 <item>Хрещатик </item>  
 <item> Арсенальна </item>  
 <item>....</item>  
 </string-array>  
</resources>

Результат роботи:



1. ControlsSample

**MainActivity.kt:**

package com.example.controlssample  
  
import android.content.Intent  
import android.content.SharedPreferences  
import android.os.Bundle  
import android.view.KeyEvent  
import android.view.View  
import android.widget.CheckBox  
import android.widget.EditText  
import android.widget.RadioButton  
import android.widget.RadioGroup  
import android.widget.Toast  
import android.widget.ToggleButton  
import androidx.appcompat.app.AppCompatActivity  
  
class MainActivity : AppCompatActivity() {  
 var mSettings: SharedPreferences? = null  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val userName = findViewById<EditText>(R.id.*user\_name*)  
 userName.setOnKeyListener **{** v, keyCode, event **->** if ((event.*action* == KeyEvent.*ACTION\_DOWN*) && (keyCode == KeyEvent.*KEYCODE\_ENTER*)) {  
 Toast.makeText(*applicationContext*, userName.*text*, Toast.*LENGTH\_SHORT*).show()  
 return@setOnKeyListener true  
 }  
 false  
 **}** mSettings = getSharedPreferences(getString(R.string.*preference\_file\_key*), *MODE\_PRIVATE*)  
  
 val checkBox = findViewById<CheckBox>(R.id.*checkbox*)  
 val toggleButton = findViewById<ToggleButton>(R.id.*toggle\_button*)  
 val radioGroup = findViewById<RadioGroup>(R.id.*radio\_group*)  
 val editText = findViewById<EditText>(R.id.*user\_name*)  
  
 checkBox.*isChecked* = mSettings?.getBoolean(getString(R.string.*SAVED\_CHECK\_BOX\_KEY*), false) ?: false  
 toggleButton.*isChecked* = mSettings?.getBoolean(getString(R.string.*SAVED\_TOGGLE\_BUTTON\_KEY*), false) ?: false  
 radioGroup.check(mSettings?.getInt(getString(R.string.*SAVED\_RADIO\_GROUP\_KEY*), -1) ?: -1)  
 editText.setText(mSettings?.getString(getString(R.string.*SAVED\_EDIT\_TEXT\_KEY*), ""))  
 }  
  
 override fun onPause() {  
 super.onPause()  
 val editor = mSettings?.edit()  
 val checkBox = findViewById<CheckBox>(R.id.*checkbox*)  
 val toggleButton = findViewById<ToggleButton>(R.id.*toggle\_button*)  
 val radioGroup = findViewById<RadioGroup>(R.id.*radio\_group*)  
 val editText = findViewById<EditText>(R.id.*user\_name*)  
  
 editor?.putBoolean(getString(R.string.*SAVED\_CHECK\_BOX\_KEY*), checkBox.*isChecked*)  
 editor?.putBoolean(getString(R.string.*SAVED\_TOGGLE\_BUTTON\_KEY*), toggleButton.*isChecked*)  
 editor?.putInt(getString(R.string.*SAVED\_RADIO\_GROUP\_KEY*), radioGroup.*checkedRadioButtonId*)  
 editor?.putString(getString(R.string.*SAVED\_EDIT\_TEXT\_KEY*), editText.*text*.toString())  
 editor?.apply()  
 }  
  
 fun onButtonClicked(v: View?) {  
 Toast.makeText(this, "Кнопка натиснута", Toast.*LENGTH\_SHORT*).show()  
 }  
  
 fun onClearButtonClicked(v: View?) {  
 val userName = findViewById<EditText>(R.id.*user\_name*)  
 userName.setText("")  
 }  
  
 fun onCheckboxClicked(v: View) {  
 val checkBox = v as CheckBox  
 val message = if (checkBox.*isChecked*) "Відмічено" else "Не відмічено"  
 Toast.makeText(this, message, Toast.*LENGTH\_SHORT*).show()  
 }  
  
 fun onToggleClicked(v: View) {  
 val toggleButton = v as ToggleButton  
 val message = if (toggleButton.*isChecked*) "Увімкнено" else "Вимкнено"  
 Toast.makeText(this, message, Toast.*LENGTH\_SHORT*).show()  
 }  
  
 fun onRadioButtonClicked(v: View) {  
 val radioButton = v as RadioButton  
 Toast.makeText(this, "Вибраний звір: ${radioButton.*text*}", Toast.*LENGTH\_SHORT*).show()  
 }  
}

**activity\_main.xml:**

<?xml version="1.0" encoding= "utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <!-- Button -->  
 <androidx.appcompat.widget.AppCompatButton  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/smile\_button"  
 android:onClick="onButtonClicked"  
 android:padding="10dp" />  
  
 <!-- Checkbox -->  
 <CheckBox  
 android:id="@+id/checkbox"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onCheckboxClicked"  
 android:text="Вибери мене" />  
  
 <!-- Toggle Button -->  
 <ToggleButton  
 android:id="@+id/toggle\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:checked="false"  
 android:clickable="true"  
 android:onClick="onToggleClicked"  
 android:textOff="Дзвінок вимкнено"  
 android:textOn="Дзвінок увімкнено" />  
  
 <!-- Radio Group -->  
 <RadioGroup  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:id="@+id/radio\_group">  
 <RadioButton  
 android:id="@+id/radio\_dog"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="Собачка" />  
 <RadioButton  
 android:id="@+id/radio\_cat"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="Котик" />  
 <RadioButton  
 android:id="@+id/radio\_rabbit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="Кролик" />  
 </RadioGroup>  
  
 <!-- EditText -->  
 <EditText  
 android:id="@+id/user\_name"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Введіть ім’я" />  
  
 <!-- Clear Button -->  
 <Button  
 android:id="@+id/clear\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onClearButtonClicked"  
 android:text="Очистити"  
 android:padding="10dp" />  
  
</LinearLayout>

**string.xml:**

<resources>  
 <string name="app\_name">ControlsSample</string>  
 <string name="SAVED\_EDIT\_TEXT\_KEY">SavedEditTextKey</string>  
 <string name="SAVED\_RADIO\_GROUP\_KEY">SavedRadioGroupKey</string>  
 <string name="SAVED\_TOGGLE\_BUTTON\_KEY">SavedToggleButtonKey</string>  
 <string name="SAVED\_CHECK\_BOX\_KEY">SavedCheckBoxKey</string>  
 <string name="preference\_file\_key">com.example.myapp.PREFERENCE\_FILE\_KEY</string>  
</resources>

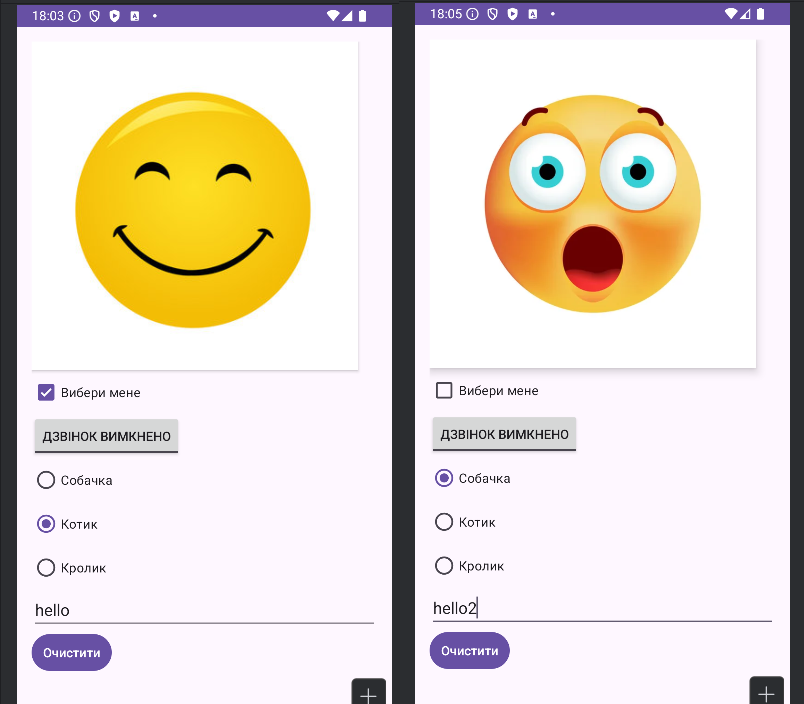
**smile\_button.xml:**

<?xml version="1.0" encoding= "utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:drawable="@drawable/smile\_pressed"  
 android:state\_pressed= "true"/>  
 <item android:drawable="@drawable/smile\_focused"  
 android:state\_focused= "true"/>  
 <item android:drawable="@drawable/smile\_normal"/>  
</selector>

Добавимо 2 фотографії для кнопки smile\_button і назвемо їх smile\_normal.jpg і smile\_pressed.jpg .

Результат роботи.

Зліва приклад без натискання на зображення, справа – натиснуто. Також бачимо, що можна змінювати інші параметри:



Після виключення програми всі дані зберігаються.

***Висновок:*** На даній лабораторній роботі я навчилася працювати з різними віджетами Android, а також з елементами керування інтерфейсом користувача, такими як кнопки із зображеннями, радіокнопки та чекбокси. Я зрозуміла принципи використання вкладок у додатках через TabWidget, відображення веб-контенту за допомогою WebView, створення та налаштування списків за допомогою ListView, а також додавання інтерактивних елементів для покращення взаємодії користувача з додатком з ControlsSample .

***Контрольні запитання:***

1. ***У чому відмінність між методом setOnClickListener () і методом setOnitemQickListener () елемента ListView?***

setOnItemClickListener() реагує на натискання на весь елемент списку.

setOnClickListener(): реагує на натискання на конкретний елемент всередині елемента списку (наприклад, кнопку).

1. ***Який механізм виступає в ролі «сполучної ланки» між джерелом даних і елементом ListView?***

Це є адаптер. Він бере дані з будь-якого місця, створює елементи списку для кожного продукту, поєднує дані і елементи. Тобто, пише назву в відповідному місці списку.

1. ***Який елемент-контейнер найкраще підходить для вирівнювання дочірніх елементів View щодо батьківського елементу?***

Для вирівнювання елементів на екрані найчастіше використовують RelativeLayout. Він дозволяє визначати розташування елементів відносно один одного (ліворуч, праворуч, зверху, знизу) або відносно батьківського елемента.

1. ***Особливості використання RelativeLayout.***

RelativeLayout дозволяє розміщувати елементи відносно один одного і батьківського контейнера. При великій кількості елементів може стати важко розібратися в ієрархії і обчисленнях позицій. Але він ідеально підходить для складних макетів, де потрібно точно контролювати розташування елементів.

1. ***Обробка подій життєвого циклу додатку.***

Життєвий цикл додатку - це послідовність станів, через які проходить додаток від запуску до закриття. Для кожного стану є свої методи, які ти можеш перевизначити, щоб виконати певні дії. Наприклад, коли додаток запускається вперше, викликається метод onCreate(). А коли додаток переходить у фоновий режим, викликається метод onPause().

Це є важливим тому що користувач може зберігати дані, коли додаток переходить у фоновий режим. Є можливість відстежування стану додатку, виконувати рідні дії від нього, наприклад: звільнювати ресурси, коли додаток закривається.