

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

ЗВІТ
з лабораторної роботи № 4
«СТВОРЕННЯ ОБ'ЄКТУ ACTIVITY. РОБОТА З КІЛЬКОМА ACTIVITY»

Виконав:

студент 3-го курсу, групи КП-83,
спеціальності 121 – Інженерія
програмного забезпечення

Палій Дмитро Володимирович

Перевірив:

к. т. н, старший викладач

Хайдуров Владислав Володимирович

ЗМІСТ

ВСТУП

1. Постановка задачі
2. Короткі теоретичні відомості методу розв'язання задачі
3. Програмна реалізація задачі
4. Результати роботи програми

ВИСНОВКИ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

ВСТУП

В даній роботі буде розглянуто основні принципи створення мобільних застосунків з використанням кількох вікон (activity), створено багатовіконні застосунки з використанням Activity та Fragment.

1. Постановка задачі

1. Ознайомитись із усіма теоретичними відомостями до лабораторної роботи.
2. Переглянути програмні реалізації багатовіконних додатків, що наведені у матеріалах до лабораторної роботи.
3. Створити програмний застосунок, що складається з чотирьох вікон – об'єкти типу Activity. Перша Activity. Повинна містити список (або кнопки) з трьох довільно обраних країн світу. За допомогою списку (або кнопок) повинні виконуватись переходи до інших трьох Activity, які містять коротку інформацію про обрані країни разом із фотографією цих країн.
4. Створити копію мобільного застосунку з попереднього завдання та використати замість Activity об'єкти Fragment. Перехід між Fragment здійснювати за допомогою кнопок або списку.
5. Створити копію мобільного застосунку з попереднього завдання та реалізувати можливість перелистування вікон на базі ViewPager.
6. Усі результати (програмний код, код розмітки, алгоритмічна частина коду та скріншоти роботи) розроблення мобільних застосунків занести до звіту. За бажанням можна оформити додатки у звіті з лабораторної роботи.
7. Захист звіту передбачає демонстрацію роботи розроблених у лабораторній роботі програмних застосунків.

2. Короткі теоретичні відомості методу розв'язання задачі

Робота з Activity

1. Створення багатовіконного проекту «My Activities»

В Android Studio створимо новий проект, новий додаток – «My Activities». Натиснемо next, empty activity, скоротимо назву Layout Name до main і натиснемо finish. Система підготує файли проекту.

2. Редагування зовнішнього вигляду головної активності

У файлі main.xml видалимо текстовий рядок TextView «Hello World», він нам більше не знадобиться. У файлі Strings.xml додамо кілька текстових значень, рядкова змінна TextBt1 зі значенням – «Перша activity», так само додамо ще два рядкових елемента TextBt2 і TextBt3, зі значеннями «Друга activity» і «Третя

activity». Очистимо значення елемента TextView android:text і дамо йому android:text="@string/TextBt1", на даній формі у нас буде написано «Перша activity». З компонентів Layouts додамо LinearLayout, на яку помістимо кнопки: для виклику другої, третьої активності. Розташуємо їх вертикально, простіше це зробити через текстову версію main.xml. Розташуємо їх одну під одним, додамо ще одну кнопку, вона нам потрібна буде для закриття форми. Так само в string.xml допишемо текстове значення btnClose зі значенням вихід «Закрити». Дамо кнопкам через "@string/" значення надписів на них. Збережемо наш проект.

3. Створення нових Layout-файлів

В папці res/layout додамо новий xml-layout файл activity_main2.xml і activity_main3.xml, для опису розмітки наших майбутніх activity. Що б не створювати знову зовнішній вигляд, скопіюємо з файлу main.xml і вставимо в наші xml-файли розмітку (в нових Activity повинні бути LinearLayout і TextView). У main3.xml переприсвоїмо значення TextView TextBt3, що б у нас був напис «Третя activity», вставимо розмітку і в main2.xml, так само дамо TextView значення TextBt2, відповідно це буде «Друга activity». Збережемо проект.

4. Створення класів для Activity

У папці Java нашого проекту створимо додатково два класи: Main2Activity і Main3Activity. Додамо в Activity2 код з MainActivity, що б не набирати його знову, змінимо ім'я класу R.layout.activity_main на R.layout.activity_main2, скопіюємо код і вставимо в Main3Activity, виправимо ім'я активності в layout activity_main3. Збережемо проект.

5. Внесення активностей до маніфесту

У файлі AndroidManifest.xml опишемо наші активності, додамо ще два записи, скопіюємо опис MainActivity і виправимо android:name=".Main2Activity", змінимо категорію LAUNCHER на DEFAULT. Повторимо процедуру для Main3Activity змінивши ім'я активності, збережемо файл.

Робота з Fragment

Створення фрагмента

Для створення фрагмента необхідно створити підклас класу Fragment (або його існуючого підкласу). Клас Fragment має код, багато в чому схожий з кодом Activity. Він містить методи зворотного виклику, аналогічні методам операції, такі як onCreate(), onStart(), onPause() і onStop(). На практиці, якщо розробник бажає перевести існуючу програму Android так, щоб в ньому використовувалися фрагменти, досить просто перемістити код з методів зворотного виклику операції у відповідні методи зворотного виклику фрагмента.

Створення макета

У наведеному коді конструкція R.layout.example_fragment є посиланням на ресурс макета з іменем example_fragment.xml, що зберігається в ресурсах додатка.

Додавання фрагмента в операцію

Як правило, фрагмент додає частина призначеного для користувача інтерфейсу в операцію, і цей інтерфейс вбудовується в загальну ієрархію представлень операції. Розробник може додати фрагмент в макет операції двома способами оголосивши фрагмент в файлі макета операції. У цьому випадку можна вказати властивості макета для фрагмента, ніби він є поданням. Наприклад, файл макета операції з двома фрагментами може виглядати так:

```
<? Xml version = "1.0" encoding = "utf-8"?>
<LinearLayout                                xmlns:android
="http://schemas.android.com/apk/res/android"
    android: orientation = "horizontal"
    android: layout_width = "match_parent"
    android: layout_height = "match_parent">
    <Fragment android:name = "com.example.news.ArticleListFragment"
    android:id = "@ + id / list"
    android:layout_weight = "1"
    android:layout_width = "0dp"
    android:layout_height = "match_parent" />
    <Fragment android: name = "com.example.news.ArticleReaderFragment"
    android:id = "@ + id / viewer"
    android:layout_weight = "2"
```

```
android:layout_width = "0dp"  
android:layout_height = "match_parent" />  
</LinearLayout>
```

Атрибут `android:name` в елементі `<fragment>` визначає клас `Fragment`, екземпляр якого створюється в макеті. Коли система створює цей макет операції, вона створює екземпляр кожного фрагмента, визначеного в макеті, і для кожного викликає метод `onCreateView()`, щоб отримати макет кожного фрагмента.

Система вставляє об'єкт `View`, повернений фрагментом, безпосередньо замість елемента `<fragment>`.

Робота з ViewPager

В Android є компонент `ViewPager`, який дозволяє прокручувати екран вліво-вправо. Але він має ряд недоліків. `ViewPager` не дозволяє з легкістю добитися ефекту прив'язки, коли екрани рухаються разом з пальцем. Також потрібен механізм автоматичного доведення екранів в ту чи іншу сторону. Google запропонувала інший варіант – використовувати компонент `ViewPager`, який входить до складу `Support Package`. У нових проектах необхідна бібліотека додається автоматично, тому ми відразу можемо спробувати написати демонстраційний приклад.

Так як компонент відноситься до пакету сумісності, то використовувані фрагменти повинні належати до класу `android.support.v4.app.Fragment`, навіть якщо розробник збирається писати додаток для нових пристроїв, які підтримують стандартні фрагменти. `ViewPager` відноситься до категорії `ViewGroup` і схожий по роботі з компонентами на основі `AdapterView` (`ListView` і `Gallery`). На панелі інструментів `Android Studio` компонент можна знайти в розділі `Containers`. При використанні `ViewPager` в розмітці використовується повне ім'я класу.

3. Програмна реалізація задачі

Java-код `Main Activity` першого завдання:

```
package com.example.myapplication;
```

Звіт до лабораторної роботи No 4. © Палій Д. В.

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnUkraine = (Button)findViewById(R.id.btn_ukraine);
        Button btnPoland = (Button)findViewById(R.id.btn_poland);
        Button btnTurkey = (Button)findViewById(R.id.btn_turkey);

        btnUkraine.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in = new Intent();
                in.setClass(MainActivity.this, UkraineActivity.class);
                startActivity(in);
            }
        });

        btnPoland.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in = new Intent();
                in.setClass(MainActivity.this, PolandActivity.class);
                startActivity(in);
            }
        });

        btnTurkey.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in = new Intent();
                in.setClass(MainActivity.this, TurkeyActivity.class);
                startActivity(in);
            }
        });
    }
}
```

Java-код Activity з описом країни:

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
```



```
public class PolandActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_poland);  
    }  
}
```

Layout Main Activity першого завдання:

```
<?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">  
  
    <LinearLayout  
        android:layout_width="0dp"  
        android:layout_height="0dp"  
        android:orientation="vertical"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent">  
  
        <Button  
            android:id="@+id/btn_poland"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="@string/btn_poland" />  
  
        <Button  
            android:id="@+id/btn_turkey"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="@string/btn_turkey"  
            app:layout_constraintEnd_toEndOf="parent"  
            app:layout_constraintStart_toStartOf="parent" />  
  
        <Button  
            android:id="@+id/btn_ukraine"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="@string/btn_ukraine" />  
    </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Layout Activity з описом країни:

```
<?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">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/flag_poland" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:gravity="center"
        android:text="@string/desc_poland"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

AndroidManifest першого завдання

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity android:name=".UkraineActivity">
            <intent-filter>
```

```
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.DEFAULT"
/>
    </intent-filter>
</activity>
<activity android:name=".TurkeyActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.DEFAULT"
/>
    </intent-filter>
</activity>
<activity android:name=".PolandActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.DEFAULT"
/>
    </intent-filter>
</activity>
</application>

</manifest>
```

Java-код Main Activity другого завдання:

```
package com.example.myapplication;

import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnUkraine = (Button)findViewById(R.id.btn_ukraine);
        Button btnPoland = (Button)findViewById(R.id.btn_poland);
        Button btnTurkey = (Button)findViewById(R.id.btn_turkey);

        final FragmentManager fragmentManager = getFragmentManager();
        final Fragment polandFragment = new PolandFragment();
        final Fragment ukraineFragment = new UkraineFragment();
        final Fragment turkeyFragment = new TurkeyFragment();
```

```
        btnUkraine.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.country_details,
ukraineFragment);
                fragmentTransaction.addToBackStack (null);
                fragmentTransaction.commit ();
            }
        });

        btnPoland.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.country_details,
polandFragment);
                fragmentTransaction.addToBackStack (null);
                fragmentTransaction.commit ();
            }
        });

        btnTurkey.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.country_details,
turkeyFragment);
                fragmentTransaction.addToBackStack (null);
                fragmentTransaction.commit ();
            }
        });
    }
}
```

Java-код фрагменту з описом країни:

```
package com.example.myapplication;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class PolandFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_poland, container,
false);
    }
}
```

Layout Main Activity другого завдання:

```
<?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">

    <Button
        android:id="@+id/btn_poland"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_poland"
        app:layout_constraintBottom_toTopOf="@+id/btn_turkey"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btn_turkey"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_turkey"
        app:layout_constraintBottom_toTopOf="@+id/btn_ukraine"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_poland" />

    <Button
        android:id="@+id/btn_ukraine"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_ukraine"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_turkey" />

    <FrameLayout
        android:id="@+id/country_details"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_ukraine">

    </FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Layout фрагменту з описом країни такий самий як і в попередньому завданні.

Java-код Main Activity третього завдання:

```
package com.example.myapplication;

import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;
import androidx.viewpager.widget.ViewPager;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        MyAdapter adapter = new MyAdapter(getSupportFragmentManager());
        ViewPager viewPager = findViewById(R.id.viewpager);
        viewPager.setAdapter(adapter);
        viewPager.setCurrentItem(1);
    }

    public static class MyAdapter extends FragmentPagerAdapter {
        MyAdapter(@NonNull FragmentManager fm) {
            super(fm);
        }
        @Override
        public int getCount() {
            return 3;
        }
        @NonNull
        @Override
        public Fragment getItem(int position) {
            switch (position) {
                case 0:
                    return new PolandFragment();
                case 1:
                    return new UkraineFragment();
                case 2:
                    return new TurkeyFragment();
                default:
                    return new UkraineFragment();
            }
        }
    }
}
```

Java-код фрагменту з описом країни:

```
package com.example.myapplication;
```

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
public class PolandFragment extends Fragment {
    public PolandFragment() {
    }
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container, @Nullable
Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_poland, container,
false);
    }
}
```

Layout Main Activity третього завдання:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.viewpager.widget.ViewPager
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/viewpager"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Layout фрагменту з описом країни такий самий як і в попередньому завданні.

3. Результати роботи програми

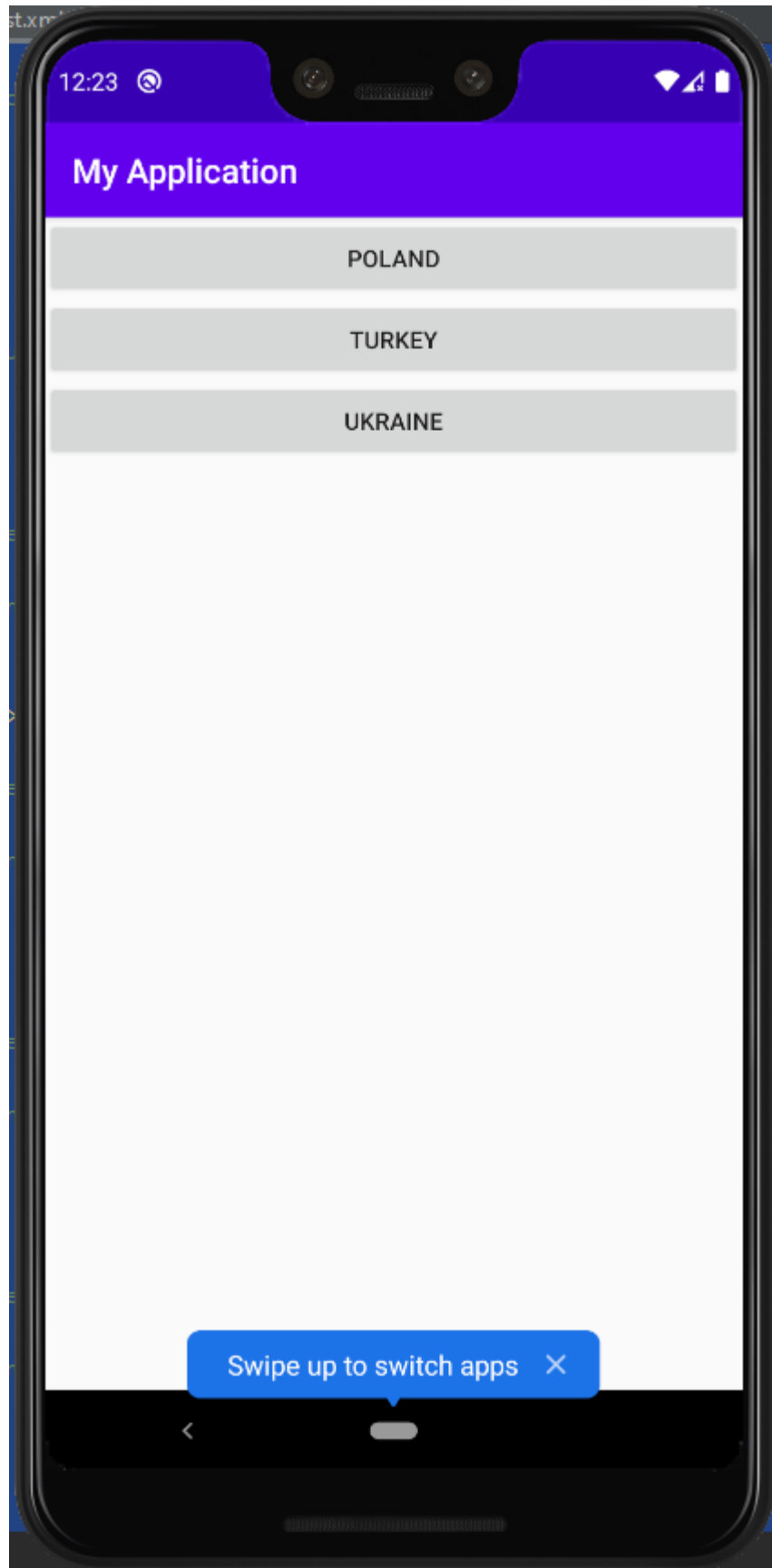


Рис. 1. Main Activity першого завдання



Рис. 2. Activity з описом країни

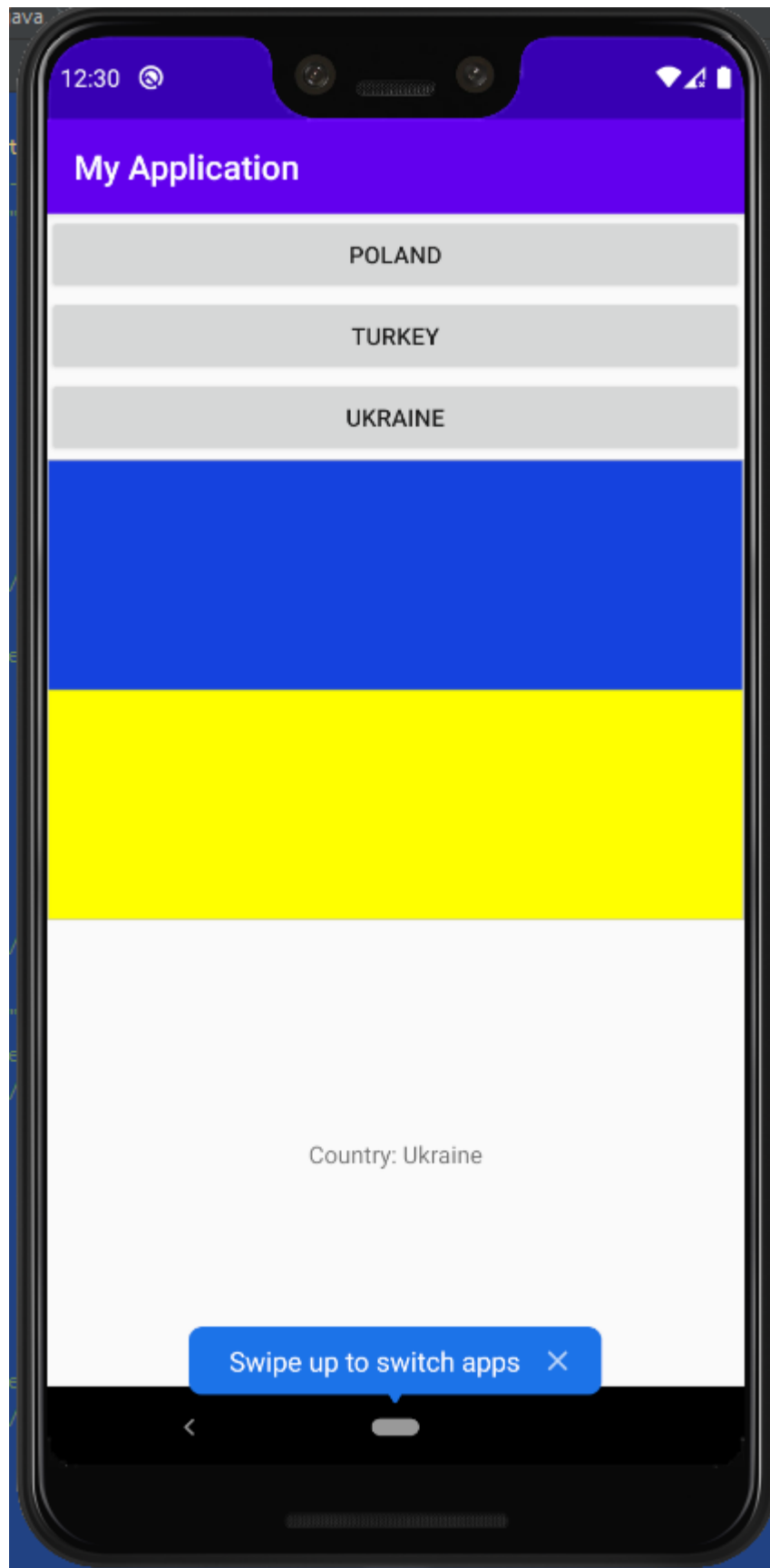


Рис. 3. Друге завдання



Рис. 4. Третє завдання

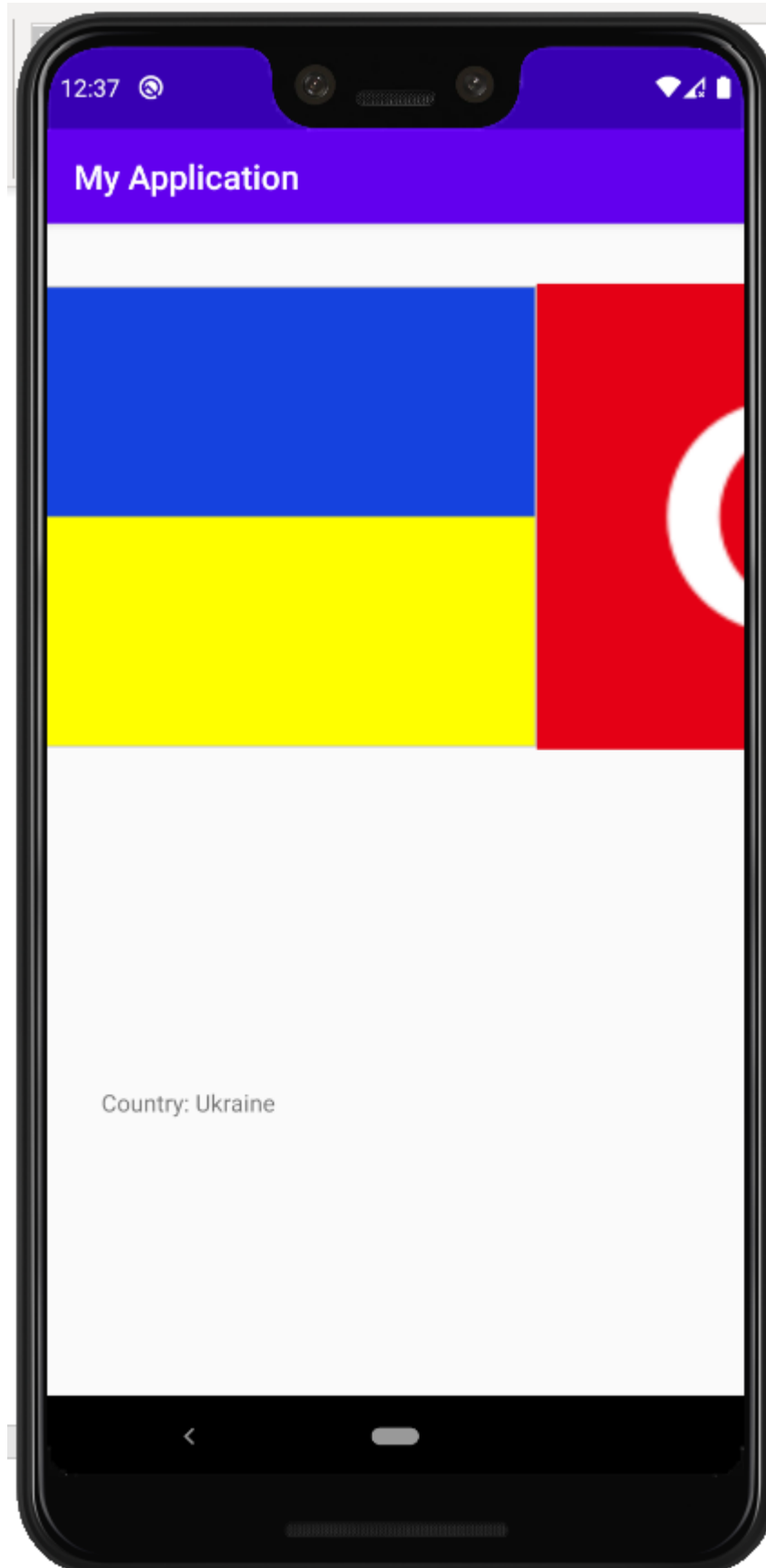


Рис. 5. Третє завдання (в процесі перелистування)

ВИСНОВОК

В ході розв’язання поставленої задачі було розглянуто основні принципи створення мобільних застосунків з використанням кількох вікон (activity), створено багатовіконні застосунки з використанням Activity та Fragment.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. <https://metanit.com/java/android/2.1.php>
2. <https://metanit.com/java/android/8.1.php>
3. <https://metanit.com/java/android/9.1.php>