

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 АКТИВІТУ. РОБОТА З КІЛЬКОМА
АКТИВІТУ»

Виконав:

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

Перевірив:

к. т. н, старший викладач
Хайдуров Владислав
Володимирович

Київ – 2020

ВСТУП	3
Завдання до лабораторної роботи	4
Короткі теоретичні відомості	5
Програмна реалізація задачі	6
Завдання 1	6
Завдання 2	10
Завдання 3	13
Висновки	18

ВСТУП

Метою даної роботи є ознайомлення із основними принципами створення мобільних застосунків з використанням кількох вікон (activity) та набуття вміння створювати багатовіконні застосунки з використанням Activity та Fragment.

Завдання до лабораторної роботи

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.

Короткі теоретичні відомості

У системі Android є велика кількість можливостей для створення багатовіконних та складних додатків. Основними з них є використання `Activity`, `Fragment` та `ViewPager` (або його аналоги такі як `ViewFlipper` та ін).

`Activity` є найбільш підходящим методом створення окремої сторінки як з зі сторони користувацького інтерфейсу (анімації переходу між сторінками), так і зі сторони програмної реалізації.

`Fragment` використовується для створення частин додатку, які можна повторно використовувати незалежно від контексту. Аналогом є компоненти с фреймворків для Front-end розробки таких як `Angular` та `React`.

`ViewPager` та допоміжні для нього класи (додаткові менеджери анімацій та переходів) використовуються там, де є певна кількість схожих сторінок, які можна використовувати в одному вікні, тобто в одній `Activity`. `ViewPager` використовує `Fragment` для власної роботи, тому є скоріш різновидом використання цього компоненту.

Програмна реалізація задачі

Завдання 1

У першому завданні було використано Activity для передачі даних від основної активності MainActivity до DisplayCountryActivity.

Частина програмного коду реалізації передачі країни як параметр до активності

```
private void openCountryActivity(Country c) {
    Intent in = new Intent();
    in.setClass(getContext(), DisplayCountryActivity.class);
    in.putExtra(Country.MODEL_INTENT_KEY, c);
    getContext().startActivity(in);
}
```

Кнопки переходу до країн генеруються у відповідності до списку країн. Відповідно, для кожної країни не потрібно окремо створювати активність. Ми створюємо список кнопок через адаптер масиву CountryAdapter та виводимо список кнопок

Частина коду виводу списку кнопок

```
private void appendButtons(final Country[] countries) {
    ListView container = findViewById(R.id.countries_container);
    final ArrayList<Country> countriesList = new
    ArrayList<Country>(Arrays.asList(countries));
    CountryAdapter countryAdapter = new CountryAdapter(this, countriesList);
    container.setAdapter(countryAdapter);
}
```

CountryAdapter

```
public class CountryAdapter extends ArrayAdapter<Country> {

    public CountryAdapter(Activity ctx, ArrayList<Country> countries) {
        super(ctx, 0, countries);
    }

    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
```

```

ViewGroup parent) {
    View listItemView = convertView;
    if(listItemView == null) {
        listItemView =
LayoutInflater.from(getContext()).inflate(R.layout.country_button, parent,
false);
    }

    final Country currentCountry = getItem(position);
    Button btn = listItemView.findViewById(R.id.country_button);
    if (btn != null && currentCountry != null) {
        btn.setText(currentCountry.name);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openCountryActivity(currentCountry);
            }
        });
    } else {
        return null;
    }

    return listItemView;
}

private void openCountryActivity(Country c) {
    Intent in = new Intent();
    in.setClass(getContext(), DisplayCountryActivity.class);
    in.putExtra(Country.MODEL_INTENT_KEY, c);
    getContext().startActivity(in);
}
}

```

Частина коду DisplayCountryActivity

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_display_country);
    this.initViews();
    try {
        Country country = (Country)
getIntent().getExtras().get(Country.MODEL_INTENT_KEY);
        this.onReceiveCountry(country);
    } catch (Exception exp) {
        System.out.println(exp.getMessage());
        finish();
    }
}

public void onFinishReading(View v) {
    this.finish();
}

```

```

    }

    private void initView() {
        this.nameTextView = findViewById(R.id.country_name);
        this.imageView = findViewById(R.id.country_image);
        this.populationView = findViewById(R.id.country_population);
        this.descriptionView = findViewById(R.id.country_description);
    }

    private void onReceiveCountry(Country c) {
        this.nameTextView.setText(c.name);
        this.setImageFromUrl(this.imageView, c.imagePath);
        this.populationView.setText(getString(R.string.population_text,
c.population));
        this.descriptionView.setText(c.description);
    }
}

```

Country

```

package com.example.task2;

import java.io.Serializable;

public class Country implements Serializable {
    public String name;
    public String imagePath;
    public int population;
    public String description;

    public static String MODEL_INTENT_KEY = "Country";

    public Country(String name,
                    String imagePath,
                    int population,
                    String description) {
        this.name = name;
        this.imagePath = imagePath;
        this.population = population;
        this.description = description;
    }
}

```

Розмітку опустимо, оскільки там немає нічого особливого та складного, що необхідно для опису виконаної роботи.

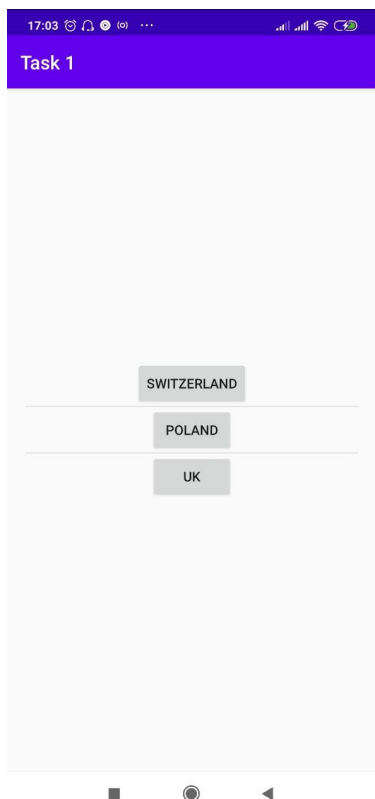


рис. 1.1, Головний екран додатку

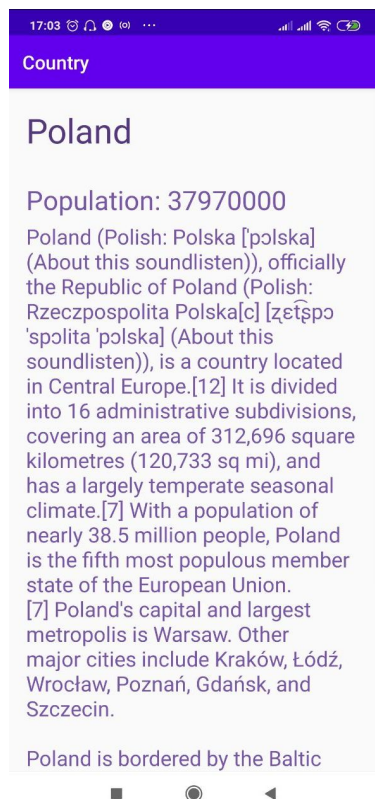


рис 1.2, сторінка країни



рис 1.3, кнопка, що повертає на головну сторінку

Завдання 2

У завданні 2 було необхідно використовувати фрагменти для відображення країни. Довелося внести деякі зміни до програмного коду, щоб змінити роботу з активностями на фрагменти.

Оновлений CountryAdapter

```
public class CountryAdapter extends ArrayAdapter<Country> {

    private FragmentTransaction ft;

    public CountryAdapter(Activity ctx, ArrayList<Country> countries) {
        super(ctx, 0, countries);
    }

    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        View listItemView = convertView;
        if(listItemView == null) {
            listItemView =
                LayoutInflater.from(getContext()).inflate(R.layout.country_button, parent,
                    false);
        }

        final Country currentCountry = getItem(position);
        Button btn = listItemView.findViewById(R.id.country_button);
        if (btn != null && currentCountry != null) {
            btn.setText(currentCountry.name);
            btn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    // will work only if parent is MainActivity
                    openCountryFragment(currentCountry,
                        ((MainActivity)getContext()).countryContainerResId);
                }
            });
        } else {
            return null;
        }

        return listItemView;
    }

    private void openCountryFragment(Country c, @IdRes int parentContainerResId)
    {
        FragmentTransaction ft = getFragmentTransaction();
        CountryDetails cdf = CountryDetails.newInstance(c);
        ft.replace(parentContainerResId, cdf);
        ft.addToBackStack(null);
        ft.commit();
    }
}
```

```

private FragmentManager getFragmentManager() {
    return ((Activity) getContext()).getFragmentManager();
}

private FragmentTransaction getFragmentTransaction() {
    return this.getFragmentManager().beginTransaction();
}
}

```

Код з DisplayCountryActivity було частково перенесено у CountryDetailsFragment. Нижче показано необхідні зміни

CountryDetailsFragment

```

public static CountryDetails newInstance(Country country) {
    CountryDetails fragment = new CountryDetails();
    Bundle args = new Bundle();
    args.putSerializable(ARG_PARAM, country);
    fragment.setArguments(args);
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        country = (Country) getArguments().getSerializable(ARG_PARAM);
    }
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_country_details,
container, false);
    this.initViews(view);
    this.setUpEventListeners(view);
    this.onReceiveCountry(this.country);
    return view;
}

private void setUpEventListeners(View root) {
    Button finishReadingButton =
root.findViewById(R.id.finish_reading_button);
    if (finishReadingButton != null) {
        finishReadingButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
getFragmentManager().beginTransaction().remove(CountryDetails.this).commit();
            }
        });
    }
}
}

```

Зовнішній вигляд було трохи змінено для демонстрації роботи фрагментів

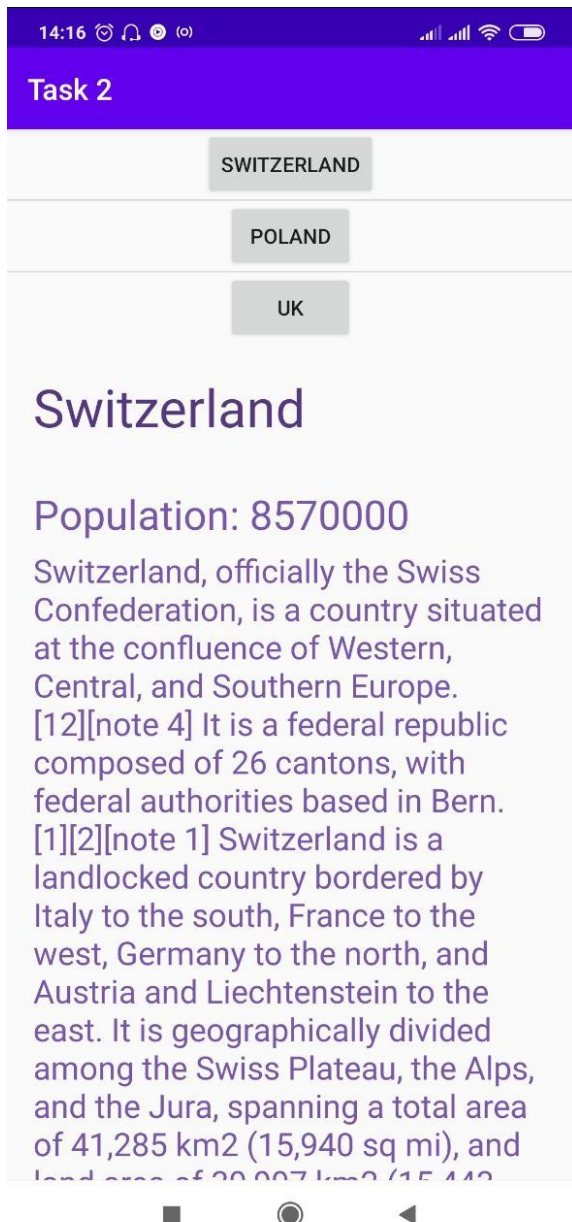


Рис 2-1, Зовнішній вигляд додатку побудованому на Fragment

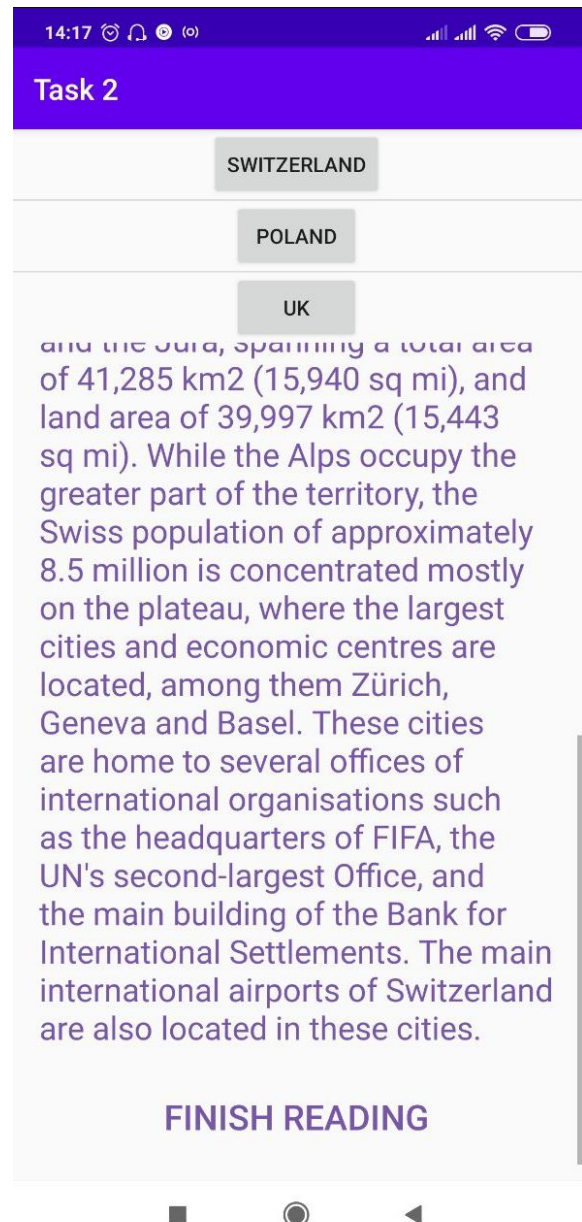


Рис 2-2, Можливість прибрати фрагмент

Завдання 3

Завдання 3 було зроблено з використанням **ViewPager**. У цьому завданні деяких змін потерпіла основна активність. Вона на початку не відображає кнопки країн, а є контейнером для **ViewPager**

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.viewpager.widget.ViewPager
    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:id="@+id/viewpager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</androidx.viewpager.widget.ViewPager>
```

Таким чином логіка створення кнопок була перенесена в **CountryButtonsFragment**

CountryButtonsFragment

```
public class CountryButtonsFragment extends Fragment {

    private static final String COUNTRIES = "countries";

    private ListView countriesContainer;

    private Country[] countries;

    public CountryButtonsFragment() {
        // Required empty public constructor
    }

    public static CountryButtonsFragment newInstance(Country[] countries) {
        CountryButtonsFragment fragment = new CountryButtonsFragment();
        Bundle args = new Bundle();
        args.putSerializable(COUNTRIES, countries);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            countries = (Country[]) getArguments().getSerializable(COUNTRIES);
        }
    }
}
```

```

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View parent = inflater.inflate(R.layout.fragment_country_buttons,
        container, false);
        this.countriesContainer = parent.findViewById(R.id.countries_container);
        appendButtons(countriesContainer, countries);
        return parent;
    }

    private void appendButtons(ListView container, final Country[] countries) {
        final ArrayList<Country> countriesList = new
        ArrayList<Country>(Arrays.asList(countries));
        CountryAdapter countryAdapter = new CountryAdapter(getActivity(),
        countriesList);
        container.setAdapter(countryAdapter);
    }
}

```

Було створено необхідний **CountryPagerAdapter** для роботи зі сторінками. Для кожної країни щоразу створюється фрагмент з необхідними даними.

CountryPagerAdapter

```

public class CountriesPagerAdapter extends FragmentPagerAdapter {

    private Country[] countries;
    CountriesPagerAdapter(@NonNull FragmentManager fm, Country[] countries) {
        super(fm);
        this.countries = countries;
    }
    @Override
    public int getCount() {
        return this.countries.length + 1;
    }

    @NonNull
    @Override
    public Fragment getItem(int position) {
        if (position <= 0 || position > (this.countries.length + 1)) {
            return CountryButtonsFragment.newInstance(this.countries);
        } else {
            return CountryDetailsFragment.newInstance(this.countries[position -
1]);
        }
    }
}

```

```
}
```

Значні зміни отримала головна активність, яка, як було описано вище, стала контейнером для **ViewPager**

MainActivity

```
public class MainActivity extends AppCompatActivity {

    private ViewPager viewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        CountriesPagerAdapter countriesPagerAdapter = new
CountriesPagerAdapter(getSupportFragmentManager(), getCountries());
        viewPager = findViewById(R.id.viewpager);
        viewPager.setAdapter(countriesPagerAdapter);
        viewPager.setCurrentItem(0);
    }

    public void onGoToCountry(Country c) {
        if (c == null) {
            viewPager.setCurrentItem(0);
            return;
        }

        int index = -1;
        Country[] countries = getCountries();
        for (int i = 0; i < getCountries().length; i++) {
            if (countries[i].name == c.name) {
                viewPager.setCurrentItem(i + 1);
                index = i;
                break;
            }
        }
        if (index < 0) {
            viewPager.setCurrentItem(0);
        }
    }

    private Country[] getCountries() {
        // ..
    }

}
```

Також для завантаження та відображення прапорів країн було додано **LoadImageByUrlTask** з кешуванням для запобігання повторних запитів

кожного разу під час відкриття фрагменту з країною та пришивдшення відображення.

LoadImageByUrlTask

```
public class LoadImageByUrlTask extends AsyncTask<String, Void, Bitmap> {
    ImageView imageView;

    private static Map<String, Bitmap> cache = new HashMap<String, Bitmap>();

    public LoadImageByUrlTask(ImageView iv) {
        this.imageView = iv;
    }

    @Override
    protected Bitmap doInBackground(String ...urls) {
        String URL = urls[0];
        if (cache.containsKey(URL)) {
            return cache.get(URL);
        }
        try {
            InputStream in = new URL(URL).openStream();
            Bitmap bitmap = BitmapFactory.decodeStream(in);
            cache.put(URL, bitmap);
            return bitmap;
        } catch (Exception e) {
            System.out.println("Loading image error " + e.getMessage());
            return null;
        }
    }

    protected void onPostExecute(Bitmap result) {
        if (result == null) {
            return;
        }
        imageView.setImageBitmap(result);
    }
}
```

Використання LoadImageByUrlTask

```
private void setImageFromUrl(ImageView imageView, String url) {
    new LoadImageByUrlTask(imageView).execute(url);
}
```


Результати роботи додатку



Рис 3.1, відображення сторінки про Велику Британію

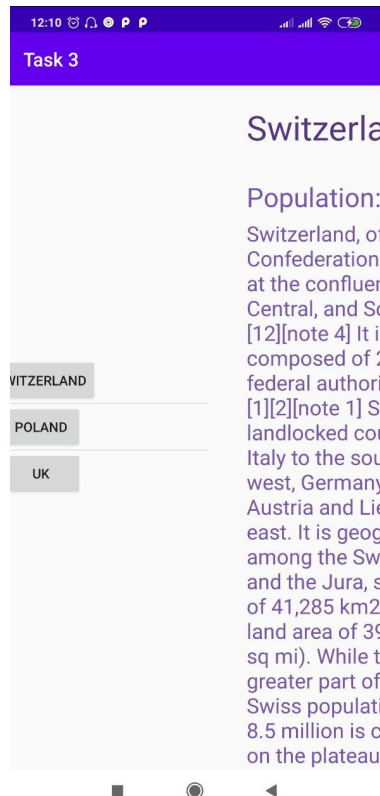


Рис 3.2, процес переходу між сторінками



Рис 3.3, відображення сторінки про Польщу

Висновки

Я ознайомився із основними принципами створення мобільних застосунків з використанням кількох вікон (activity), набув вміння створювати багатовіконні застосунки з використанням Activity та Fragment та навчився завантажувати сторонні файли, а саме зображення прапорів країн, з зовнішніх ресурсів використовуючи AsyncTask.