

Лабораторна робота №3

Тема: Створення багатовіконного додатку

Мета: Навчитися створювати додатки, що складаються з декількох активностей, і діалогових вікон, а також познайомитися з елементами тач-інтерфейсу.

Хід виконання роботи

Дана лабораторна робота присвячена різним способам розробки багатоекранних додатків. Пропонується розробити три додатки, кожен з яких присвячено окремому аспекту цієї тематики. По завершенню роботи над додатками можна розробити власні програми з використанням вивчених компонентів і технологій.

3.1 Створення багатоекранного додатку зі списком

1. Створіть проект MultiScreen. У java-файлі замініть Activity на ListActivity. Клас ListActivity розроблений таким чином, що на екрані є тільки прокручуваний список і йому не потрібна додаткова розмітка. Тому файл activity_main.xml можна видалити. Також слід видалити наступний рядок з класу MultiScreen:

```
setContentView(R.layout.activity_main);
```

так як layout-файл ми тільки що видалили.

2. Тепер нам потрібен посередник, який зв'яже список і назви елементів цього списку. Для цього в Android є інтерфейс Adapter. Нам буде потрібний спадкоємець цього класу ArrayAdapter:

```
new ArrayAdapter<Context>(context, textViewResourceId, objects)
```

В якості аргументів ArrayAdapter потребують поточний контекст, ідентифікатор ресурсу з розміткою для кожного рядка, масив рядків.

Ми можемо вказати в якості поточного контексту ListActivity (можна використовувати ключове слово this), готовий

системний ідентифікатор ресурсу

(android.R.layout.simple_list_item_1) і створений масив рядків (String[] islands = { "Канари", "Курили", "Мальдіви", "Філіпіни" });. А виглядати це буде так:

```
private ArrayAdapter<String> adapter; adapter = new  
ArrayAdapter<String>(this,  
android.R.layout.simple_list_item_1, islands);
```

Зверніть увагу на рядок android.R.layout.simple_list_item_1. У ньому вже міститься необхідна розмітка для елементів списку. Якщо вас не влаштовує системна розмітка, то можете створити власну розмітку в xml-файлі і підключити її.

Залишилося тільки підключити адаптер:

```
setListAdapter(adapter);
```

3. Тепер нам потрібно підключити обробку натискання. Для цього необхідно знати, на який пункт списку здійснюється натискання. Існує спеціальний інтерфейс OnItemClickListener з методом onItemClick().

Підключаємо обробник: listView.setOnItemClickListener()

```
setOnItemClickListener(itemListener); Набираємо все в тому ж onCreate
```

```
setOnItemClickListener(itemListener) = new OnItemClickListener() {  
}
```

Якщо підкреслити перше слово, то імпортуємо потрібний клас. Далі підведемо курсор до другого слова `new OnItemClickListener`. Нам буде запропоновано додати метод (Add unimplemented method). Погоджуємося і отримуємо заготовку:

```
OnItemClickListener itemListener = new OnItemClickListener() {  
  
    @Override  
    public void onItemClick (AdapterView <?> arg0, View arg1, int arg2, long arg3){  
        // TODO Auto-generated method stub  
    }  
};
```

Міняємо імена змінних `arg` на більш звичні і зрозумілі

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

4. Залишилося описати, що буде відбуватися при натисканні на елемент.

За нашим задумом кожен елемент буде відкривати нове вікно з відповідним вмістом. Для початку слід створити ще 4 класи: `Canary`, `Curyly`, `Maldivy`, `Philipiny`, і 4 `xml`-

оболонки до них. Можна просто створити і копіювати по одному файлу в обох директоріях, змінюючи тільки назву і вміст.

Наприклад, створимо файли `Canary.java` і `canary.xml` типу `Activity`. Зверніть увагу, що як тільки ми створили ще один `java`-файл, потрібно записати його в файл маніфесту, інакше програма не буде бачити цей новий клас. Файл `AndtoidManifest.xml` знаходиться відразу під папкою `res`. Додайте код з ім'ям класу в тегах `<activity>` `</ activity>`, відразу під головною активністю.

```
<activity android: name = "com.mypackage.multiscreen.Canary" android: label = "@  
string/can">  
</activity>
```

Рядок `can` потрібно створювати в файлі `strings.xml`, також як і інші рядки.

Перейдемо в файл `canary.xml` і створимо на екрані `TextView`. Екрани будуть відрізнятися один від одного картинками. Візьміть 4 будь-які картинки з вашого комп'ютера (можете також знайти їх в інтернеті) і перетягніть з провідника `Windows` в папку `res\drawable`. Тепер ви можете помістити елемент `ImageView` на екран і, вибравши потрібну картинку з ресурсів проекту, підключити її.

Решта екрани створюються аналогічно.

5. Тепер перейдемо до головного класу і опишемо обробку події `onItemClick ()`. Створимо перемикач, який залежить від номеру елемента.

```
switch (position) {  
    case 0:  
  
        break; case 1:  
  
        break; case 2:  
  
        break; case 3:
```

```

        break;
    }

```

Для запуску нового екрану необхідно створити екземпляр класу Intent і вказати клас, на який будемо переходити (у нас їх 4, тому для кожного випадку вибираємо свою). Після цього викликається метод startActivity (), який і запускає новий екран.

```
Intent intent = new Intent (MultiScreen.MainActivity.this, Canary.class); startActivity (intent);
```

6. Тепер залишилося додати спливаюче вікно Toast, до яке буде показувати, який елемент ми вибрали. Цей віджет можна імпортувати так само, як і попередні. Нам буде потрібно метод makeText (), у якого є три параметра: контекст додатку, текстове повідомлення і тривалість часу показу повідомлення.

```
Toast.makeText (getApplicationContext(), "Ви вибрали" +
parent.getItemAtPosition(position).toString(),Toast.LENGTH_SHORT).show();
```

Повні листинги файлів проекту, в яких були зроблені зміни, див. Нижче.

Лістинг 3.1 - Файл MultiScreenMainActivity.java

```

import android.os.Bundle;
import android.view.View;
import android.app.ListActivity; import
android.content.Intent; import
android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Toast;

public class MainActivity extends ListActivity{

    String[] islands = { "Канари", "Курили", "Мальдіви", "Філіппіни" }; @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, islands);
        setListAdapter(adapter);

        OnItemClickListener itemListener = new OnItemClickListener() { @Override
            public void onItemClick(AdapterView<?> parent, View v, int position,
long id) {

                switch (position) {
                    case 0:
                        Intent intent = new Intent(MainActivity.this,
CanaryActivity.class);
                        startActivity(intent);
                        break;
                    case 1:
                        Intent intent1 = new Intent(MainActivity.this,
Curyly.class);
                        startActivity(intent1);
                        break;
                }
            }
        }
    }
}

```

```

        case 2:
            Intent intent2 = new Intent(MainActivity.this,
                Maldivy.class);
            startActivity(intent2);
            break;

        case 3:
            Intent intent3 = new Intent(MainActivity.this,
                Philipiny.class);
            startActivity(intent3);
            break;
    }
    Toast.makeText(getApplicationContext(), "Ви вибрали" +
        parent.getItemAtPosition(position).toString(),
        Toast.LENGTH_SHORT).show();
    }
};
getListView().setOnClickListener(itemListener);
}
}

```

Лістинг 3.2. Файл Canary.java

```

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

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

```

Лістинг 3.3. Файл canary.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:padding="10dip" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="186dp"
        android:src="@drawable/canary" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"

```

```

android:layout_centerHorizontal="true"
android:layout_marginTop="80dp" android:text="@string/enjoy"
android:textAppearance="?android:attr/textAppearanceLarge" />

```

</RelativeLayout>

Лістинг 3.4. Файл strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Куди б поїхати в відпустку?</string>
  <string name="action_settings">Settings</string>
  <string name="enjoy">Enjoy yourself!</string>
  <string name="can">Канари</string>
  <string name="phil">Філіпіни</string>
  <string name="cur">Курили</string>
  <string name="mal">Мальдіви</string>
</resources>

```

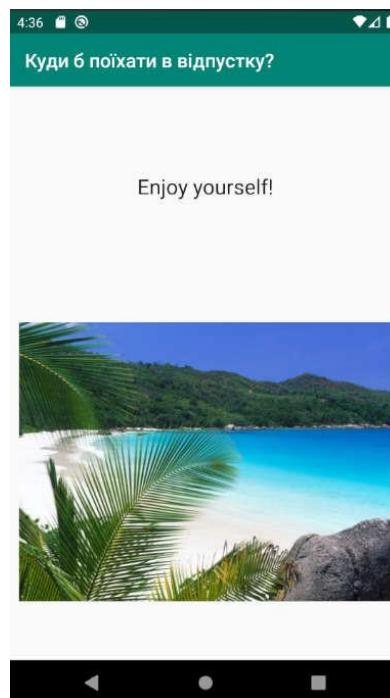
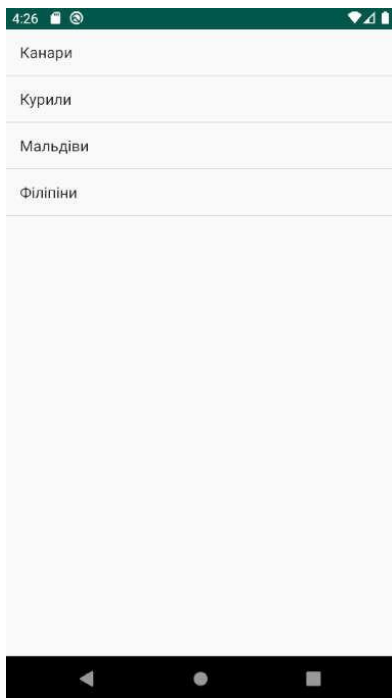


Рис. 3.1 - Додаток "Куди б поїхати у відпустку?", Запущений на пристрої

3.2 Створення діалогового вікна

1. Створіть новий проект Dialog
2. Створіть кнопку на вашій активності та назвіть її "Вибрати фон" (для цього потрібно в файлі strings створити рядок відповідного змісту). Дайте активності і кнопці id.

Лістинг 3.5 - Код файлу activity_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/relativelayout"
  android:layout_width="fill_parent" android:layout_height="fill_parent" >

```

```

<Button
    android:id="@+id/background_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="174dp"
    android:text="@string/bg"/>
</RelativeLayout>

```

3. Наш додаток змінює фон на обраний. Значить, нам потрібно створити кольори і їх назви в файлі strings.xml. Як ви пам'ятаєте, цей файл знаходиться в папці values, яка в свою чергу знаходиться в папці res. Також створимо рядок messages, яка нам знадобиться для діалогового вікна.

Лістинг 3.6 - файл strings.xml

```

<resources>
    <string name = "app_name"> Dialog </string>
    <string name = "action_settings"> Settings </string>
    <string name = "bg"> Вибрати фон </string>
    <string name = "message"> Хочете поміняти фон? </string>
    <string name = "red"> Червоний </string>
    <string name = "yellow"> Жовтий </string>
    <string name = "green"> Зелений </string>
</resources>

```

Лістинг 3.7 - файл colors.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
    <color name = "redColor"> #FF0000 </color>
    <color name = "yellowColor"> #FFFF00 </color>
    <color name = "greenColor"> #00FF00 </color>>
</resources>

```

4. Перейдемо в файл MainActivity.java. Створіть наступні змінні:

```

private Button bgButton;
public RelativeLayout relativeLayout; Context context;

```

Якщо компілятор підкреслює тип і повідомляє про помилку, наприклад, підкреслює Context, наведіть курсор на підкреслене слово: має з'явитися контекстне меню, яке пропонує варіанти, як можна виправити помилку. Виберете Import 'Context', щоб імпортувати бібліотеку.

5. Тепер потрібно описати, що буде відбуватися при натисканні на нашу кнопку.

Для початку зв'яжемо об'єкти з activity_main.xml і змінні в MainActivity.java через id (в onCreate):

```

bgButton = (Button) findViewById (R.id.background_button); relativeLayout = (RelativeLayout)
findViewById (R.id.relativelayout);

```

Context - це об'єкт, який надає доступ до базових функцій програми.

Додаємо в код

```
context = MainActivity.this;
```

6. Тепер потрібно додати подію onClick і навесити на кнопку OnClickListener. Додаємо в заголовок класу MainActivity implements OnClickListener. Щоб зв'язати кнопку і Listener в onCreate пишемо

```
bgButton.setOnClickListener (this);
```

Створимо тепер подію onClick

```
@Override
```

```
public void onClick (View v) {  
}
```

В цьому об'єкті створюємо власне діалог і називаємо його:

```
AlertDialog.Builder builder = new AlertDialog.Builder (this); builder.setTitle (R.string.message);
```

```
AlertDialog alert = builder.create (); alert.show ();
```

Ми будемо створювати діалогове вікно, яке надає користувачеві вибір зі списку. Для цього буде потрібно ще одна змінна, яка сформує список з наявних рядків.

```
final CharSequence[] items = {  
    getText (R.string.red),          getText (R.string.yellow),          getText  
(R.string.green)  
};
```

7. Сформуємо власне наш список і поставимо ще один Listener, який буде змінювати колір фону на обраний:

```
builder.setItems (items, new DialogInterface.OnClickListener () {  
    public void onClick (DialogInterface dialog, int item) { switch (item) {  
        case 0: {relativeLayout.setBackgroundResource (R.color.redColor); break;}  
        case 1: {relativeLayout.setBackgroundResource (R.color.yellowColor); break;}  
        case 2: {relativeLayout.setBackgroundResource (R.color.greenColor); break;}  
    }  
}
```

8. Залишилося додати в кожен case спливаючі вікна Toast, і додаток повністю готовий!
Toast.makeText(context, R.string.green, Toast.LENGTH_LONG).show();

Лістинг 3.8. Код файлу MainActivity.java

```
package com.example.i.dialog;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle; import
```

```
android.app.Activity; import
```

```
android.app.AlertDialog; import
```

```
android.content.Context;
```

```
import android.content.DialogInterface;
```

```
import android.view.View;
```

```

import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.Toast;

```

```

public class Dialog extends Activity implements OnClickListener {

```

```

    private Button bgButton;
    public RelativeLayout relativeLayout; Context context;

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dialog);
        bgButton = (Button) findViewById(R.id.background_button);
        bgButton.setOnClickListener(this); context =
        Dialog.this;
        relativeLayout = (RelativeLayout)findViewById(R.id.relativelayout);
    }

```

```

    @Override

```

```

    public void onClick(View v) {

```

```

        final CharSequence[] items={getText(R.string.red), getText(R.string.yellow),
        getText(R.string.green)

```

```

        };

```

```

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle(R.string.message);

```

```

            builder.setItems(items, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int item) {

```

```

                    switch (item) {

```

```

                        case 0: {

```

```

                            relativeLayout.setBackgroundResource(R.color.redColor);

```

```

                                Toast.makeText(context, R.string.red,

```

```

                                Toast.LENGTH_LONG).show();

```

```

                                    break;} case

```

```

                                1:

```

```

                            { relativeLayout.setBackgroundResource(R.color.yellowColor);

```

```

                                Toast.makeText(context, R.string.yellow,

```

```

                                Toast.LENGTH_LONG).show();

```

```

                                    break;} case

```

```

                                2:

```

```

                            { relativeLayout.setBackgroundResource(R.color.greenColor);

```

```

                                Toast.makeText(context, R.string.green,

```

```

                                Toast.LENGTH_LONG).show();

```

```

                                    break;}

```

```

                                }

```

```

                            }

```

```

                        });

```

```

                    AlertDialog alert = builder.create(); alert.show();

```

```

                }

```

```

    }

```


Скріншоти працюючого додатку



Рис. 3.2 - Додаток "Dialog", запущений на емуляторі

Завдання

Подумайте над власним додатком, що поєднує різні можливості проектування багатовіконних додатків, розглянутих вище. Створіть прототип цього додатку і налаштуйте його користувацький інтерфейс.