

Lab 1: Basic Android Programming Report

學號：109511286

姓名：蔡佩蓉

1. 請說明 android 程式如何處理 widget 物件?

Android 程式可以使用 Widget 元件來在應用程式的使用者介面中顯示資訊或提供互動元素。

XML 檔為版面配置檔案，每個 XML 檔都只能含有一個根元素，為 View 或 ViewGroup 物件，之後就可以在根元素之下新增子項元素。執行程式時，Android 會呼叫 JAVA 檔中的 onCreate() 設定應用程式的初始狀態，包括利用 setContentView() 來載入和顯示 XML 布局檔。

```
// 初始化活動
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); // 載入和顯示XML布局檔
```

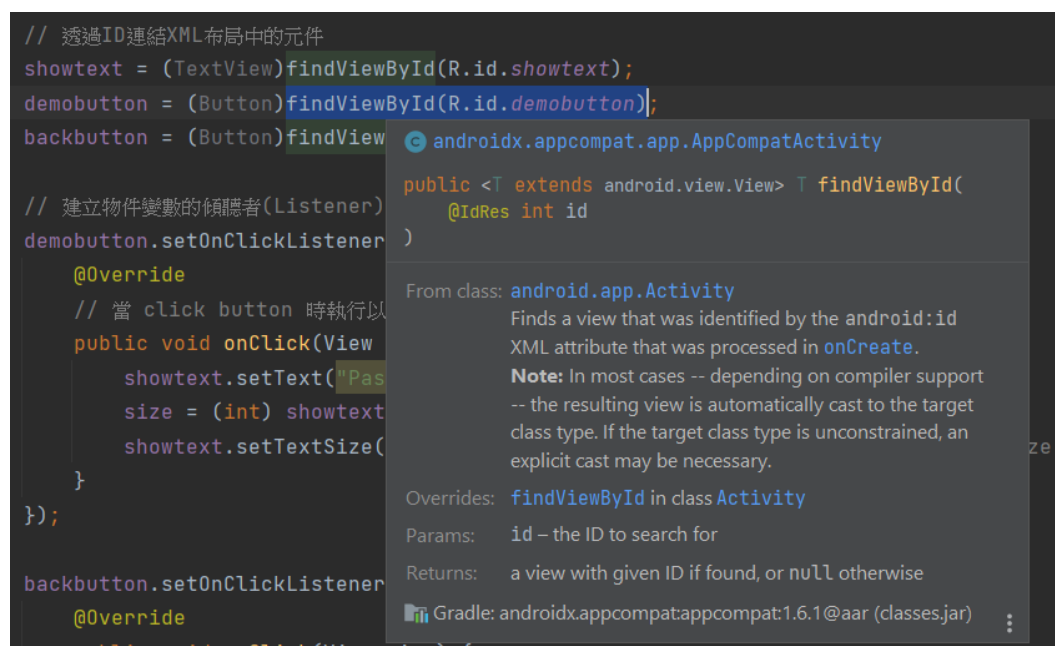
在 XML 檔中寫進欲加入的 Widget 元件，並設定 id, text 等元件特性；事件處理則寫在 Java 檔中，並透過 findViewById()，將 Java 中的變數與 XML 布局中的元件進行連結。之後在 JAVA 檔中設定 Listener 去監聽指定的元件，再利用 onClick(), onKeyDown() 等函式來設定若特定事件發生時，會發生什麼事。

```
// 透過ID連結XML布局中的元件
showtext = (TextView) findViewById(R.id.showtext);
demobutton = (Button) findViewById(R.id.demobutton);
// 建立物件變數的傾聽者(Listener) - 點擊監聽器
demobutton.setOnClickListener(new View.OnClickListener() {
    @Override
    // 當 click button 時執行以下程式
    public void onClick(View view) {
        showtext.setText("Pass!"); //將 text 設定成 Pass!
    }
});
```

2. 請說明 R.id.button1 是什麼?

R.id.button1 是 Button 的 resource ID，每個 resource ID 都是唯一值。在 Android 開發中，所有在 XML 布局檔中定義的視圖元件（如 Button、TextView 等）都需要一個唯一的 ID。這個 ID 用來在 Java 程式碼中找到並與相應的視圖元件進行連結。而這裡的 R.id.button1 可能是在 XML 布局檔中設定給某個 Button 的 ID。在 Java 程式碼中，透過 findViewById(R.id.button1) 就可以找到這個按鈕元件，並對其進行操作，例如設定點擊監聽器。

e.g.



可以帶入題目的 `R.id.button1` 為以上 Java 程式碼中的 `R.id.demobutton`。從上圖可見 `findViewById()` 裡的 parameter 為 `int id`，一個用來查找的 ID。

單純用 Android SDK 自帶的 `android` 命令來生成 Android 項目時，編譯後能找到一個 `R.java` 檔，裡面紀錄了各元件的 `id`，為一個 `public static final int`。

(Source: [从源码分析来理解 R.id.mybutton r.id.btn assure-CSDN 博客](#))

3. 請舉例一個有結尾標籤的元件（可截圖）

```
<LinearLayout
    android:layout_width="409dp"
    android:layout_height="387dp"
    android:orientation="horizontal"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="343dp">

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="TextView" />
</LinearLayout>
```

在 XML 中，每個標籤一定要有結尾標籤，例如使用了<LinearLayout>這個標籤，後頭就需要有一個</LinearLayout>標籤，若標籤為沒有結尾標籤的單獨標籤，則在「>」符號前必須要有「/」符號。

若不需要在兩個標籤中間放其他元素（子元素），該標籤被稱為單獨標籤。

e.g.

```
<LinearLayout
    android:layout_width="409dp"
    android:layout_height="387dp"
    android:orientation="horizontal"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="343dp" />
```

4. 請貼上自己的程式碼並附上註解

Code Q1 & Q2

```
package com.example.app109511286;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.TypedValue;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    // 宣告物件變數
    TextView showtext;
    Button demobutton;
    Button backbutton;
    int size;

    // 初始化活動
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // 載入和顯示XML布局檔

        // 透過id連結XML布局中的元件
        showtext = (TextView) findViewById(R.id.showtext);
        demobutton = (Button) findViewById(R.id.demobutton);
        backbutton = (Button) findViewById(R.id.backbutton);

        // 建立物件變數的傾聽者 (Listener) - 點擊監聽器
        demobutton.setOnClickListener(new View.OnClickListener() {
            @Override
            // 當 click button 時執行以下程式
            public void onClick(View view) {
                showtext.setText("Pass!"); //將 text 設定成 Pass!
                size = (int) showtext.getTextSize() + 10; // 原本的 text size 再 + 10
                showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size); // 設定新的text size
            }
        });

        backbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showtext.setText("Android Lab1 Demo"); //將 text 設定成 Android Lab1 Demo
                size = (int) showtext.getTextSize() - 10; // 原本的 text size 再 - 10
                showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size);
            }
        });
    }
}
```

Code Q3

```
package com.example.app109511286;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.TypedValue;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    // 宣告物件變數
    TextView showtext;
    Button demobutton;
    Button backbutton;
    int size;

    // 初始化活動
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // 載入和顯示XML布局檔

        // 透過ID連結XML布局中的元件
        showtext = (TextView) findViewById(R.id.showtext);
        demobutton = (Button) findViewById(R.id.demobutton);
        backbutton = (Button) findViewById(R.id.backbutton);

        // 建立物件變數的傾聽者 (Listener) - 點擊監聽器
        demobutton.setOnClickListener(new View.OnClickListener() {
            @Override
            // 當 click button 時執行以下程式
            public void onClick(View view) {
                showtext.setText("Pass!"); //將 text 設定成 Pass!
                size = (int) showtext.getTextSize() + 10; // 原本的 text size 再 + 10
                showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size); // 設定新的text size
            }
        });

        backbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showtext.setText("Android Lab1 Demo"); //將 text 設定成 Android Lab1 Demo
                size = (int) showtext.getTextSize() - 10; // 原本的 text size 再 - 10
                showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size);
            }
        });
    }
}
```

5. 心得

這次實驗了解到 **Android Project** 的基本撰寫方式與運作方法，包括掌握了 **Android** 項目的建立，在 **XML** 檔中定義 **Widget** 元件，使用 **Java** 程式碼與這些元件進行互動，以及設定事件監聽器來處理使用者交互。此外我也學習到如何透過 **XML** 來做版面的設計。

這次實驗除了前面設定環境有點花時間解決 **API** 問題之外，整體做下來還是蠻順利的！