

Lab 1: Basic Android Programming Report

學號：109511207 姓名：蔡宗儒

1. 請說明 android 程式如何處理 widget 物件？

當我們要創造 widget 物件時，要去 xml 檔裡選擇我們要創造的物件，像是 Button 物件，再將他放置到我們想要設計的位置上面，並記得給他 id。以這次有用到的 Button 為例，每次按下 Button 時都會 trigger event，會有相對應的 listener 可以知道 Button 是否被觸發。如此一來當按下按鈕後，listener 就會知道 Button 被觸發，我們就可以先寫好該事件對應的程式碼，並讓 listener 呼叫這段程式碼並被執行，從而產生相對應的回應，這就是 event handling，一種能在特定事件發生時自動執行預定義程式碼的機制，讓使用者能夠得到對應的互動。

2. 請說明 R.id.button1 是什麼？

button1 就是 xml 檔裡某個 Button 物件的 id，當我們要在程式碼中撰寫關於 button1 這個物件的相關程式碼時，要先宣告 data type 相對應的變數，像是這邊要宣告一個 data type 為 Button 的變數，再透過 (Button)findViewById(R.id.button1); 藉由物件的 id 來取得 xml 檔裡的 Button 物件，就可以透過該變數來撰寫相關程式碼。R 則表示 R.java，該類別由 ADT 自動產生。而同理當我們要在程式碼中撰寫關於某個 TextView 物件的相關程式碼時，要宣告 data type 為 TextView 的變數，再透過 (TextView)findViewById(R.id.該 TextView 在 xml 檔裡的 id); 來取得 xml 檔裡的 TextView 物件。

3. 請舉例一個有結尾標籤的元件 (可截圖)

LinearLayout

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"></LinearLayout>
```

4. 請貼上自己的程式碼並附上註解

Q1 & Q2

```
// Package folder path, used for classification
package com.example.app109511207;

// Import the required libraries.
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;

// MainActivity inherits AppCompatActivity
public class MainActivity extends AppCompatActivity {
```

```
// Declare the object variables that will be used
TextView showtext;
Button demobutton;
Button backbutton;

// Override the onCreate() function of the parent class AppCompatActivity
@Override
protected void onCreate(Bundle savedInstanceState) {

    // Access members of the parent class
    super.onCreate(savedInstanceState);

    // Transform the layout through the setContentView() function
    setContentView(R.layout.activity_main);

    // Obtain the object through the id set in xml file
    showtext = (TextView)findViewById(R.id.showtext);
    demobutton = (Button)findViewById(R.id.demobutton);
    backbutton = (Button)findViewById(R.id.backbutton);

    // Listen for whether the demobutton triggers an event
    demobutton.setOnClickListener(new View.OnClickListener() {

        // Override the onClick() function of the parent class AppCompatActivity
        @Override
        public void onClick(View v) {

            // Update showtext and set it to "Pass!"
            showtext.setText("Pass!");

            // Get text size
            int size = (int)showtext.getTextSize();

            // Text size + 5
            size += 5;

            // Update showtext size
            showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size);
        }
    });

    // Listen for whether the backbutton triggers an event
```

```

backbutton.setOnClickListener(new View.OnClickListener() {

    // Override the onClick() function of the parent class AppCompatActivity
    @Override
    public void onClick(View v) {

        // Update showtext and set it to "Android Lab1 demo"
        showtext.setText("Android Lab1 demo");

        // Get text size
        int size = (int)showtext.getTextSize();

        // Text size - 5
        size -= 5;

        // Update showtext size
        showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size);
    }
});
}
}

```

Q3

```

// Package folder path, used for classification
package com.example.app109511207_lab1_3;

// Import the required libraries
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

// MainActivity inherits AppCompatActivity
public class MainActivity extends AppCompatActivity {

    // Declare the object variables that will be used
    TextView showtext;
    Button setbutton;
    Button resetbutton;
    EditText name;

```

```

// Override the onCreate() function of the parent class AppCompatActivity
@Override
protected void onCreate(Bundle savedInstanceState) {

    // Access members of the parent class
    super.onCreate(savedInstanceState);

    // Transform the layout through the setContentView() function
    setContentView(R.layout.activity_main);

    // Obtain the object through the id set in xml file
    showtext = (TextView)findViewById(R.id.showtext);
    setbutton = (Button)findViewById(R.id.setbutton);
    resetbutton = (Button)findViewById(R.id.resetbutton);
    name = (EditText)findViewById(R.id.name);

    // Listen for whether the setbutton triggers an event
    setbutton.setOnClickListener(new View.OnClickListener() {

        // Override the onClick() function of the parent class AppCompatActivity
        @Override
        public void onClick(View v) {

            // Update showtext and set it to "Welcome to Android, " + EditText
            showtext.setText("Welcome to Android, " + name.getText().toString());
        }
    });

    // Listen for whether the resetbutton triggers an event
    resetbutton.setOnClickListener(new View.OnClickListener() {

        // Override the onClick() function of the parent class AppCompatActivity
        @Override
        public void onClick(View v) {

            // Update showtext and set it to "Hello World!"
            showtext.setText("Hello World!");
        }
    });
}
}

```

5. 心得

這次 lab 是我第一次寫 java，Android 提供了許多現成的函式庫讓使用者使用，而我也了解到了使用 XML(eXtensible Markup Language)能讓使用者在寫一個 app 或設計 UI 時變得簡單很多。Android 提供了兩種方式來建立 UI，一種是直接程式碼中設計，另一種則是透過 XML 為基礎的版面設計描述檔來完成。XML 與 HTML 一樣是屬於標籤語言的一種。但 XML 有一個很大的特點就是自定標籤。

HTML 具有預設標籤，例如使用插入圖片，<A>添加超連結。相較之下，XML 需要使用者自定義各種標籤。一個 app 可以自定義自己的標籤，如果使用者想跟 app 溝通，就要瞭解 app 的自定標籤如何使用。舉例來說，假設一個通訊 app 定義了<NAME>表示連絡人姓名、<TEL>表示電話、<ADDR>來表示地址，當使用者希望 app 讀進一筆連絡人資料時，就要按照 app 定義的標籤，寫出相應的 XML 檔案來傳遞資料。這強調了 XML 的主要目的是描述資訊的內容及其代表的意義，而 HTML 則主要用於顯示資料。

而在 Android 中常使用 XML 來設計 UI。Android 定義了一些標籤，像是<Button>表示按鈕元件，<ImageView>表示圖片元件，<TextView>表示文字方塊元件。這些標籤具有不同的屬性，例如<TextView>標籤的 android:text 屬性用於設定文字方塊中顯示的文字內容，因此不必直接修改 Java 程式碼，便能夠寫出簡單 Android app。

另外我們使用的 Android Studio 也非常好用，像是有手機模擬器的功能，這樣如果跑的程式有問題的話就能夠馬上發現，而程式碼出現問題也可以透過 Alt+Enter 來快速找出問題所在，像是我一開始沒有 import android.util.TypedValue，所以 compiler 一直看不懂 TypedValue.COMPLEX_UNIT_PX 是什麼，但我按下 Alt+Enter 後，它就幫我自己 import 好了，可以說相當方便也很好 debug。

我自己認為這次 lab 沒有很難，雖然之前沒打過 Java，但因為 Android 都提供了現成的函式了，所以大概了解一下需要用到的函式就可以輕鬆完成這次 lab 了，設計 UI 的部分我在之前大二時修蔡媽開的 AOOP 時也有做過類似需要設計 UI 的專題了，也大概了解這些 event handling 的物件怎麼使用。

6. reference

[1][使用 XML 描述使用者介面](#)

[2][Java 什麼是覆寫\(Override\)](#)