

Android APP手機程式設計實務



講師：賴貴平



本節課程內容

- 本節課程內容將包含以下教學內容:

Android 意圖(Intent)與程式生命週期

- 意圖(Intent)
- 明確意圖與資料傳遞
- 隱含意圖與資料傳遞
- 活動的生命周期



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）：
 - 是一個啟動其他**Android**活動、服務和廣播接收器的系統訊息
 - 一種抽象方式來描述希望執行的操作，可以告訴**Android** 作業系統想作什麼？執行什麼動作？
 - 例如：
 - 啟動其他活動
 - 告訴指定服務可以啟動
 - 停止與送出廣播



ANDROID 意圖(Intent)與程式生命週期

- 意圖 (**intents**) :
 - Android應用程式送出意圖的訊息
 - 經過Android作業系統來判斷接收者是誰
 - 使用意圖篩選 (**intent filters**) 找出有能力處理的活動或內建應用程式，然後才將訊息送給接收者
- 活動 (**Activity**) :
 - 活動是一個佔滿行動裝置螢幕的視窗，在同一Android應用程式可以擁有多個活動
 - 在活動中是一至多個介面元件建立的使用介面



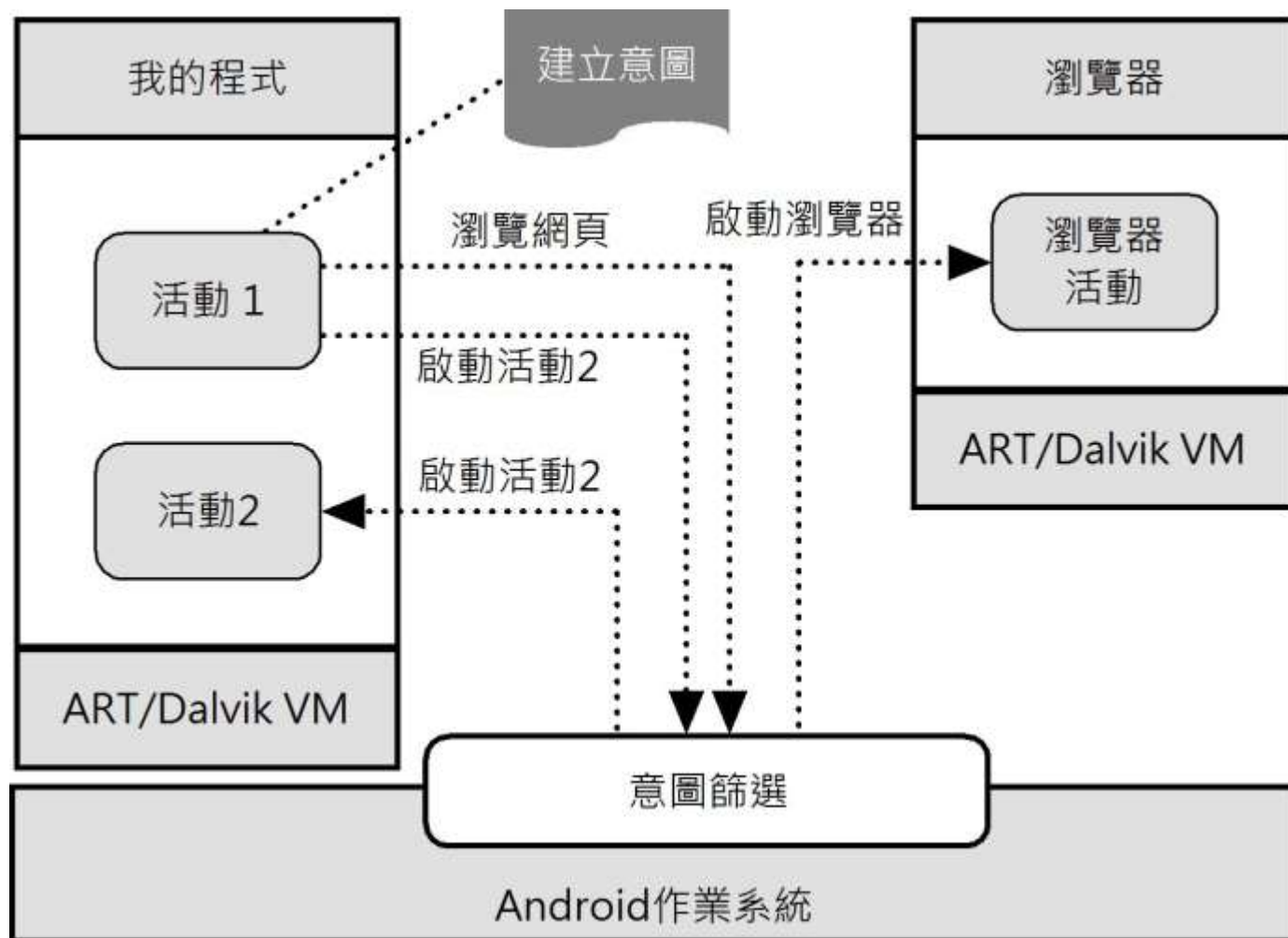
ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - Android作業系統的意圖可以分為兩種
 - 明確意圖 (explicit intent)
 - 指明目標活動接收者名稱，即明確指明接收對象，通常是使用在連接同一應用程式內部的多個活動
 - 隱含意圖 (implicit intent)
 - 意圖只有指出執行的動作型態和資料，並沒有目標接收者的確實活動名稱，Android作業系統任何可以完成此工作的應用程式都可以是接收者，例如瀏覽網頁



ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :





ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - Android應用程式的活動可以分成兩大類型
 - 獨立活動
 - 一種沒有資料交換的活動，單純只是從一個螢幕轉換至下一個螢幕，利用意圖啟動活動間的轉換
 - 相依活動
 - 一種類似Web網頁之間資料傳遞的活動，在活動之間有資料交換，我們需要將資料傳遞至下一個活動，和取得回傳資料，利用意圖啟動活動間的轉換，並進行參數資料傳遞



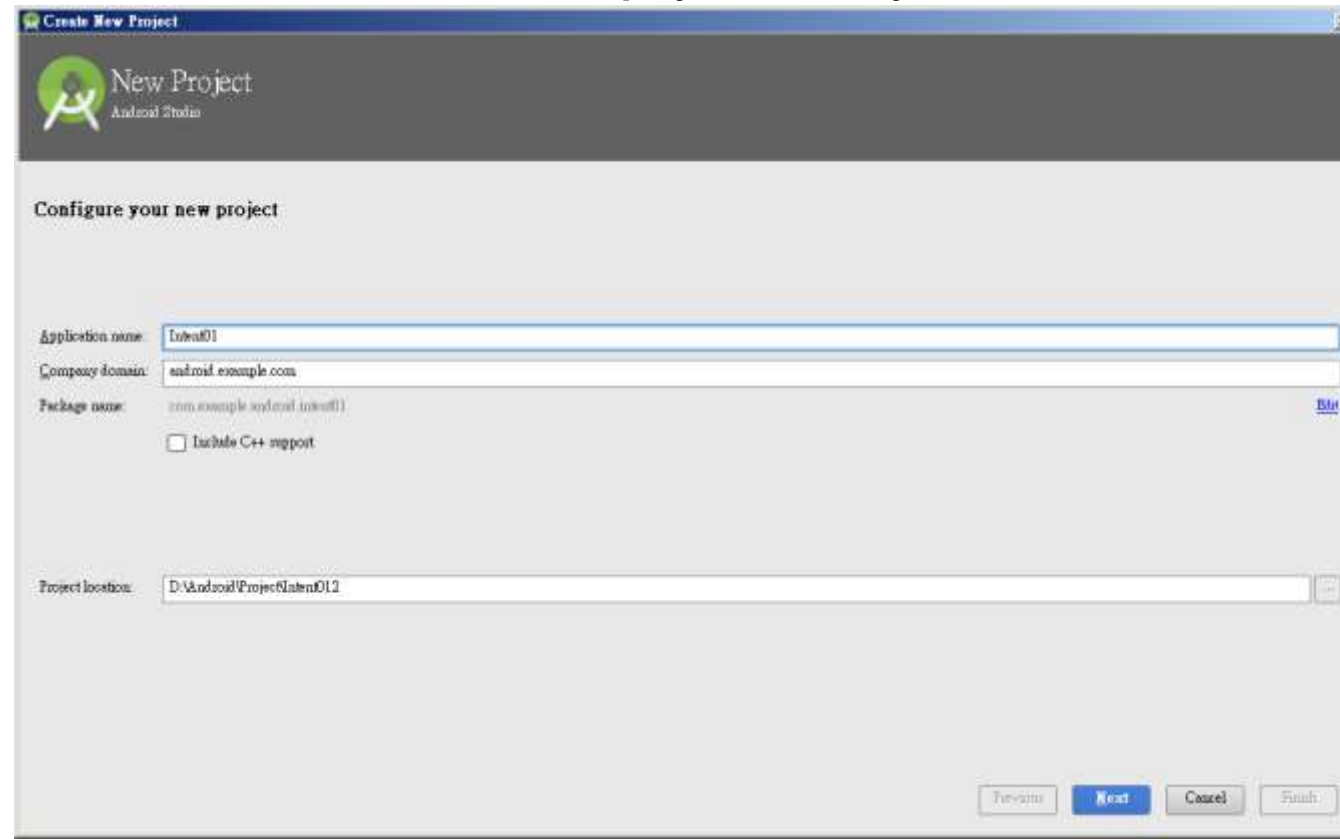
ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - 在專案新增活動
 - Android應用程式使用意圖來啟動其他活動之前，需要先在專案新增活動
 - 在Android Studio專案新增活動類別，建立一個新活動
 - 在AndroidManifest.xml設定檔註冊新增的活動
 - <activity>標籤下新增另一個<activity>標籤
 - 語法
 - <activity android:name=".Activity名稱"></activity>



ANDROID 意圖(Intent)與程式生命週期

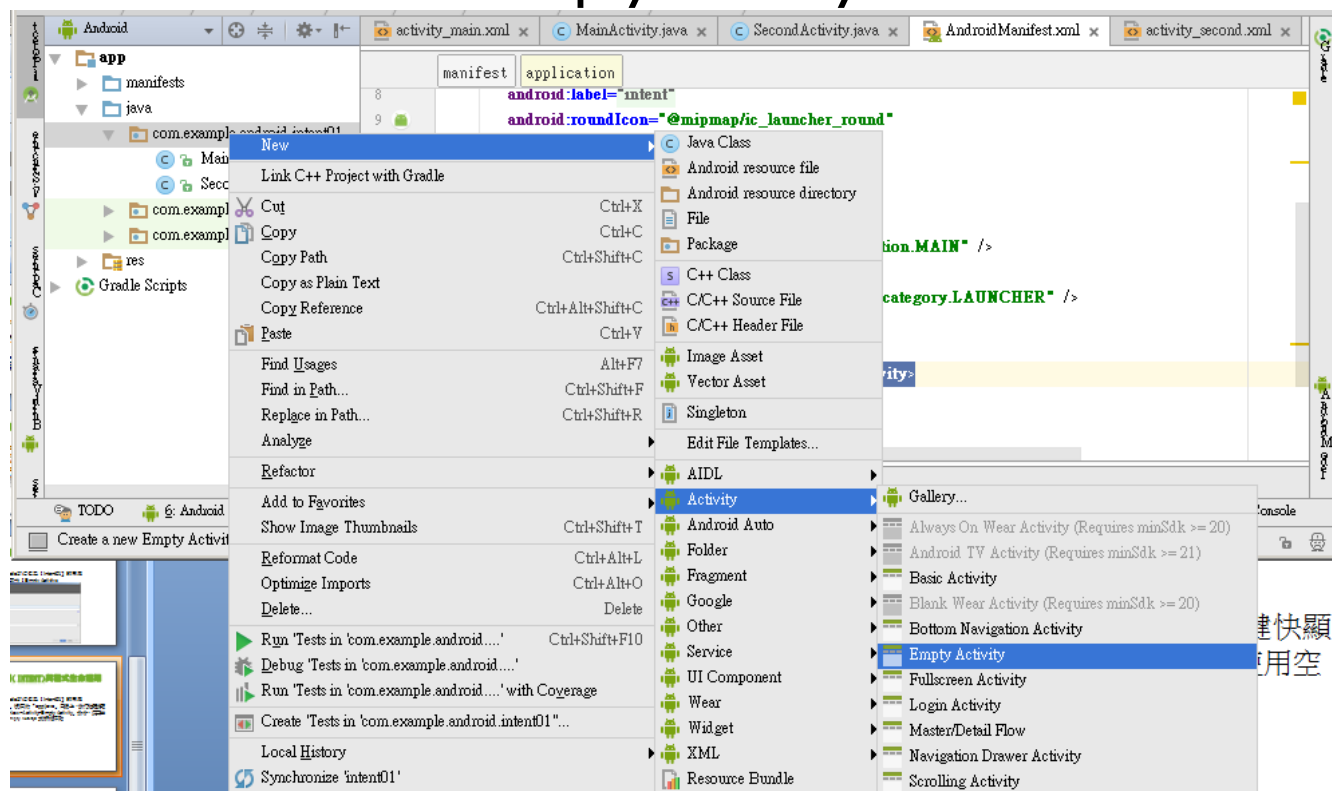
- 意圖 (intents) :
 - 在Android Studio新增名為【Intent01】的專案
 - 使用的活動範本【Empty Activity】





ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - 新增【Intent01】後在「Project」視窗的「app\java」目錄上，執行右鍵快顯功能表的「New>Activity>Empty Activity」命令，用空活動範本【Empty Activity】來新增活動 SecondActivity

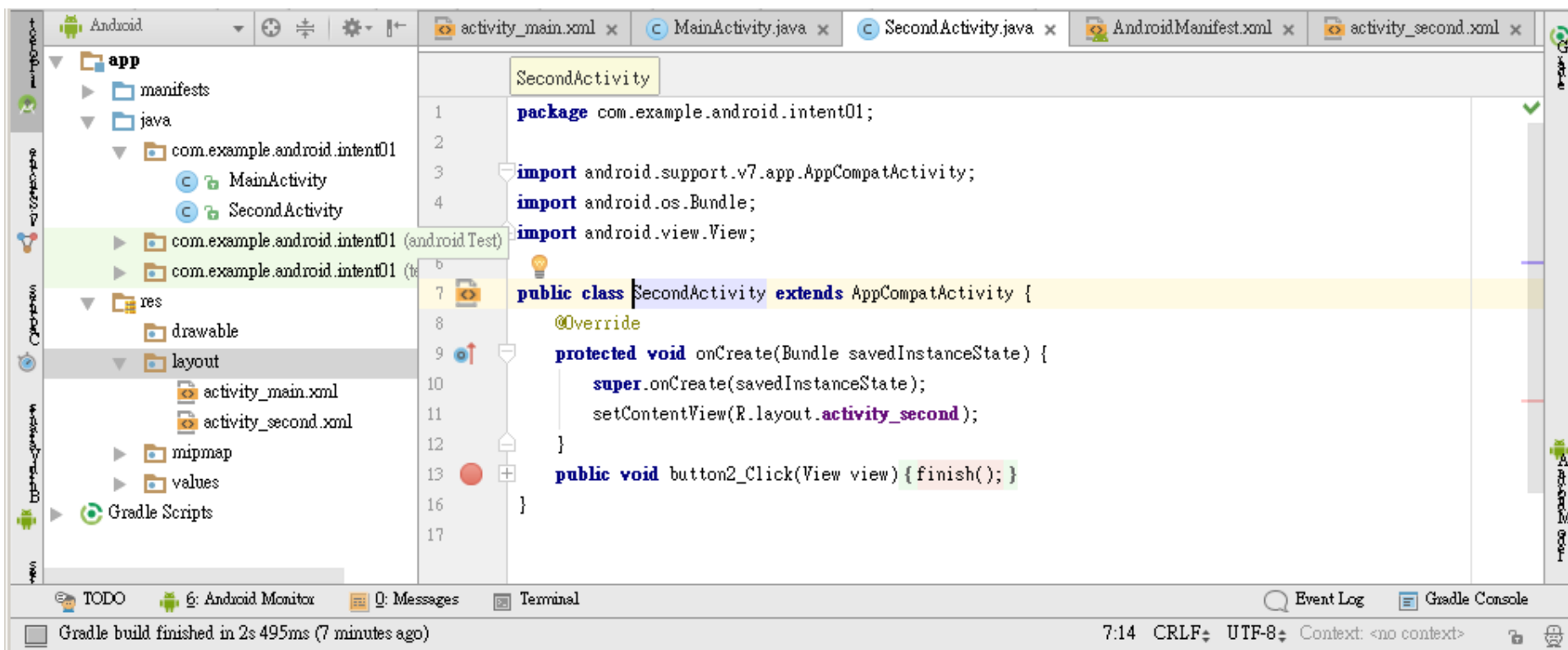




ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :

- 在「Project」專案視窗的「app\java」目錄下，可以看到Java程式檔【SecondActivity.java】，在「res\layout」目錄下的佈局檔【activity_second.xml】





ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - 上述步驟會新增名為【SecondActivity】的第二個活動類別和佈局檔
 - 使用活動範本建立活動，預設會自動註冊新增的活動
 - 請開啟位在「app\manifests」目錄下的AndroidManifest.xml檔，可以看到已經註冊第二個活動





ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - AndroidManifest.xml設定檔，提供Android作業系統關於應用程式的資訊，一個功能清單
 - 不同於Windows作業系統，Android作業系統需要透過AndroidManifest.xml檔案先認識這個應用程式，才能正確執行應用程式
 - 應用程式完整名稱（包含Java套件），一個唯一的識別名稱，讓Android作業系統和Google Play找到此應用程式
 - 應用程式包含的活動、內容提供者、廣播接收器和服務元件
 - 宣告應用程式執行時需要的權限



ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - AndroidManifest.xml設定檔是使用<application>標籤宣告Android應用程式擁有的元件
 - 子標籤<activity>宣告應用程式擁有的活動
 - 例如 :
 - <activity android:name=".MainActivity">
 - <intent-filter>
 - <action android:name="android.intent.action.MAIN" />
 -
 - </intent-filter>
 - </activity>
 - <activity android:name=".SecondActivity"></activity>



ANDROID 意圖(Intent)與程式生命週期

- 意圖 (intents) :
 - 當Android應用程式擁有多個活動時，使用意圖來啟動其他活動
 - 呼叫startActivity()方法來啟動程式中的其他活動，參數Intent意圖物件指明開啟的是哪一個活動
 - startActivity(intent);
 - 使用建構子來建立Intent物件
 - Intent intent = new Intent(this, SecondActivity.class);
 - 建構子的第1個參數是活動自己(this)，第2個參數是欲開啟的活動類別SecondActivity.class，在「.»前是活動名稱，之後class表示是活動類別



ANDROID 意圖(Intent)與程式生命週期

- 意圖 (**intents**) :
 - 在活動呼叫**finish()**方法，可以關閉目前開啟的活動
 - **finish()**;
 - 如果目前活動是由來源活動所開啟，關閉活動就會回到來源活動
 - **Intent01** 範例應用程式擁有兩個活動，按下第**1**個活動的按鈕，可以開啟第**2**個活動，關閉第**2**個活動可以回到第**1**個活動



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **AndroidManifest.xml.xml**
 - `<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=".SecondActivity"></activity>`
 - `</application>`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_main.xml`
 - `<TextView`
 - `android:id="@+id/textView"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text="主活動" />`
 - `<Button`
 - `android:text="啟動第二個活動"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_marginTop="51dp"`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_main.xml`
 - `android:id="@+id/button"`
 - `android:layout_below="@+id/textView"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:onClick="button_Click" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_second.xml`
 - `<TextView`
 - `android:text="第二個活動"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_alignParentTop="true"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:id="@+id/textView2" />`
 - `<Button`
 - `android:text="關閉第二個活動"`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_second.xml`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_below="@+id/textView2"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:layout_marginTop="42dp"`
 - `android:id="@+id/button2"`
 - `android:onClick="button2_Click" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - package com.example.android.intent01;
 - import android.content.Intent;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - import android.view.View;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_main);
 - }



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - `public void button_Click(View view) {`
 - `Intent intent = new Intent(this,SecondActivity.class);`
 - `startActivity(intent);`
 - `}`
 - `}`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `SecondActivity.java`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `import android.view.View;`
 - `public class SecondActivity extends AppCompatActivity {`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_second);`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **SecondActivity.java**
 - `public void button2_Click(View view) {`
 - `finish();`
 - `}`
 - `}`



ANDROID 意圖(INTENT)與程式生命週期

- 成果





ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）間傳遞資料：
 - **Intent**物件除了可以啟動活動，還可以攜帶資料，將這些資料一併傳遞給目標活動
 - 例如：在主活動輸入攝氏溫度，然後將輸入值作為傳遞資料傳至目標活動來計算轉換華氏溫度的結果
 - **Intent**物件攜帶的資料是**Bundle**物件
 - **Bundle**為一種目錄物件來儲存字串型態鍵值對應的各種資料型態資料
 - 建立**Intent**物件
 - `Intent intent = new Intent(this, FActivity.class);`



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）間傳遞資料：
 - 建立Intent物件後，接著建立Bundle物件
 - Bundle物件以索引值儲存資料，將需要傳遞的資料都存放進去
 - Bundle bundle = new Bundle();
 - bundle.putString("TEMPC", txtC.getText().toString());
 - 上述程式碼使用putString()方法新增字串資料，第1個參數是字串的索引"TEMPC"，第2個參數是值
 - 常用的方法有putInt()放入整數、putDouble()放入浮點數和putByte()放入位元組資料等



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）間傳遞資料：
 - 使用Intent物件的putExtras()方法附加Bundle物件
 - 附加Bundle後的Intent物件不只有啟動活動的資訊，還攜帶有資料，然後呼叫startActivity()方法啟動活動
 - intent.putExtras(bundle);
 - startActivity(intent);
 - 在目標活動呼叫活動物件的getIntent()方法取得Intent物件
 - Intent intent01 = this.getIntent();



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）間傳遞資料：
 - Intent物件的getExtras()方法取得攜帶的Bundle物件
 - Bundle bundle = intent01.getExtras();
 - if條件判斷是否有攜帶資料，如果有，就使用getString()方法取出資料，參數是之前指定的字串鍵值"TEMPC"
 - if (bundle != null) {
 - c = Integer.parseInt(bundle.getString("TEMPC"));
 - ...
 - }



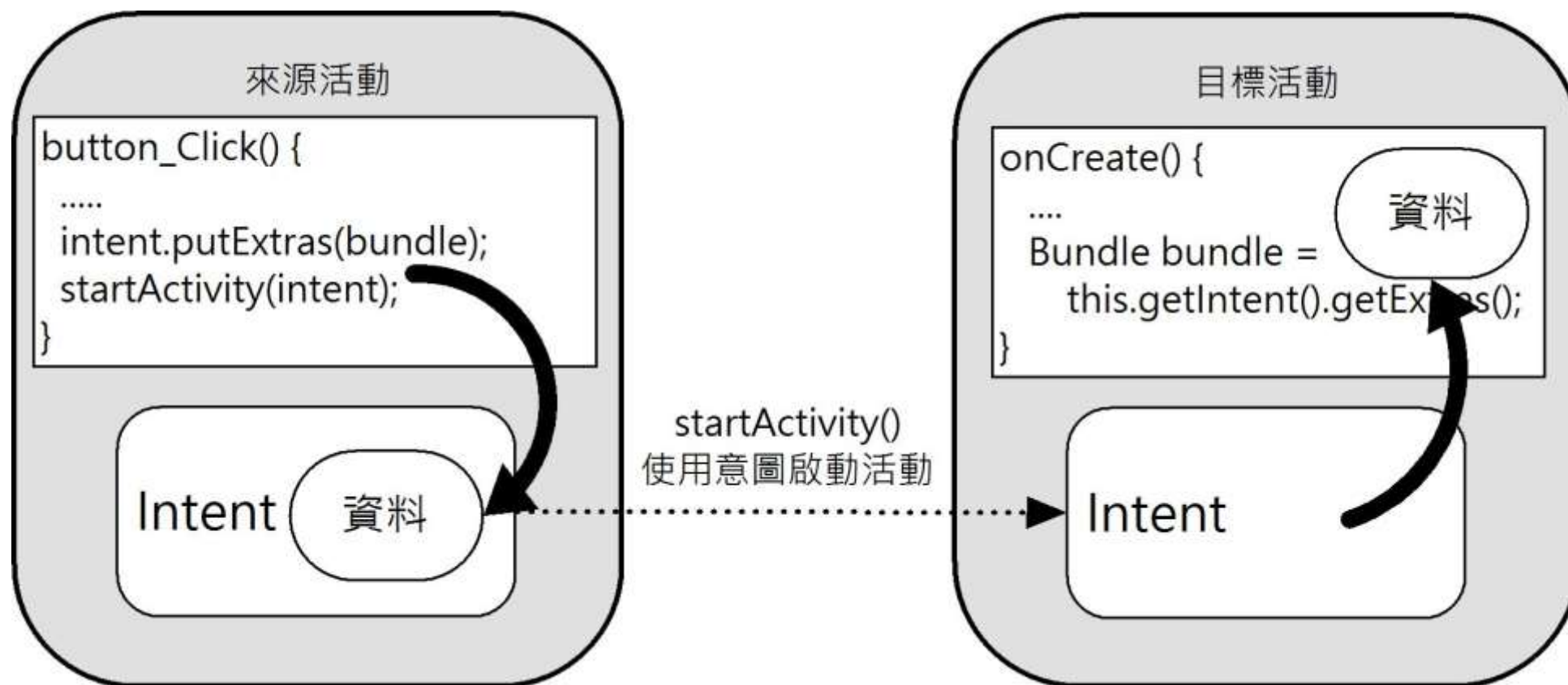
ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）間傳遞資料：
 - 可以使用`getInt()`取出整數、`getDouble()`取出浮點數和`getBytes()`取出位元組資料
 - **Intent 02**範例程式建立擁有2個活動的溫度轉換程式，在第一個活動輸入攝氏溫度後，將資料傳遞給第二個活動來計算轉換結果的華氏溫度



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）間傳遞資料:





ANDROID 意圖(Intent)與程式生命週期

- 開啟 `AndroidManifest.xml`
 - `package="com.example.android.intent02">`
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:supportsRtl="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=".FActivity"></activity>`
 - `</application>`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_main.xml`
 - `<TextView`
 - `android:text="攝氏:"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_alignParentTop="true"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:id="@+id/textView" />`
 - `<EditText`
 - `android:layout_width="wrap_content"`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **activity_main.xml**
 - `android:layout_height="wrap_content"`
 - `android:inputType="number"`
 - `android:ems="10"`
 - `android:id="@+id/txtC"`
 - `android:text="100"`
 - `android:layout_below="@+id/textView"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:layout_marginTop="20dp" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_main.xml`
 - `<Button`
 - `android:text="轉換"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:id="@+id/button"`
 - `android:onClick="button_Click"`
 - `android:layout_alignTop="@+id/txtC"`
 - `android:layout_alignParentRight="true"`
 - `android:layout_alignParentEnd="true"`
 - `android:layout_marginRight="22dp"`
 - `android:layout_marginEnd="22dp" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 activity_f.xml
 - <TextView
 - android:text="TextView"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:layout_alignParentTop="true"
 - android:layout_alignParentLeft="true"
 - android:layout_alignParentStart="true"
 - android:id="@+id/lblOutput" />
 - <Button
 - android:text="關閉activity_f"



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_f.xml`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_below="@+id/textView"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:layout_marginTop="42dp"`
 - `android:id="@+id/button2"`
 - `android:onClick="button2_Click" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_main);
 - }
 - // Button元件的事件處理
 - public void button_Click(View view) {
 - // 取得EditText元件
 - EditText txtC = (EditText) findViewById(R.id.txtC);



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - // 建立Intent物件
 - Intent intent = new Intent(this, FActivity.class);
 - // 建立傳遞資料的Bundle物件
 - Bundle bundle = new Bundle();
 - bundle.putString("TEMPC",txtC.getText().toString());
 - intent.putExtras(bundle); // 加上資料
 - startActivity(intent); // 啟動活動
 - }
 - }



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **FActivity.java**
 - `public class FActivity extends AppCompatActivity {`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_f);`
 - `convertTempature();`
 - `}`
 - `// 轉換溫度`
 - `private void convertTempature() {`
 - `int c;`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **FActivity.java**
 - `double f = 0.0;`
 - `// 取得傳遞的資料`
 - `Bundle bundle = this getIntent().getExtras();`
 - `if (bundle != null) {`
 - `c = Integer.parseInt(bundle.getString("TEMP_C"));`
 - `// 攝氏轉華氏的公式`
 - `f = (9.0 * c) / 5.0 + 32.0;`
 - `// 顯示華氏溫度`
 - `TextView o = (TextView) findViewById(R.id.lblOutput);`
 - `o.setText("華氏溫度:" + Double.toString(f));`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **FActivity.java**
 - }
 - }
 - public void button2_Click(View view) {
 - finish();
 - }



ANDROID 意圖(INTENT)與程式生命週期

- 成果





ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：
 - 使用**startActivity()**方法啟動其他活動，只能將資料傳遞至目標活動，並不能取得活動的回傳資料
 - 取得活動的回傳資料需要改用**startActivityForResult()**方法啟動活動
- **Intent 03**範例程式建立一個簡易計算機，整個使用介面有**2**個活動，在第一個活動輸入**2**個運算元後，將輸入資料傳遞給第二個活動，讓使用者選擇運算子，在計算後，回傳給第一個活動來顯示



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：
 - 建立Intent物件，建構子的第1個參數是活動自己，第2個參數是OpActivity.class
 - 建立Bundle物件傳遞2個字串資料
 - Intent intent = new Intent(this, OpActivity.class);
 - Bundle bundle = new Bundle();
 - bundle.putString("OPERAND01", txtOpd1.getText().toString());
 - bundle.putString("OPERAND02", txtOpd2.getText().toString());
 - intent.putExtras(bundle);



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：
 - **startActivityForResult()**方法
 - 使用參數的意圖和請求碼來啟動新活動，可以取得啟動新活動的回傳值
 - **startActivityForResult(intent, SET_RESULT);**
 - **startActivityForResult** 的第1個參數是Intent物件，第2個參數是請求碼的整數值
 - **SET_RESULT**值是用來在活動中識別是哪一個活動的回傳資料（因為同一活動可能啟動多個有回傳值的活動）



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：
 - 回傳值的活動呼叫`getIntent()`方法取得Intent物件後，即可取出傳遞的資料
 - 以下取得傳遞的2個字串，並且轉換成整數
 - `Bundle bundle = this.getIntent().getExtras();`
 - `opd1 = Integer.parseInt(`
`bundle.getString("OPERAND01"));`
 - `opd2 = Integer.parseInt(`
`bundle.getString("OPERAND02"));`



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：
 - 要回傳值的活動結束前，也是使用Intent物件建立回傳資料
 - 以下建立Bundle物件、並附加至Intent物件
 - `Intent rIntent = new Intent();`
 - `Bundle rbundle = new Bundle();`
 - `rbundle.putDouble("RESULT", result);`
 - `rIntent.putExtras(rbundle);`
 - 利用設定的結果碼（**result code**）回傳資料
 - `setResult(RESULT_OK, rIntent);`



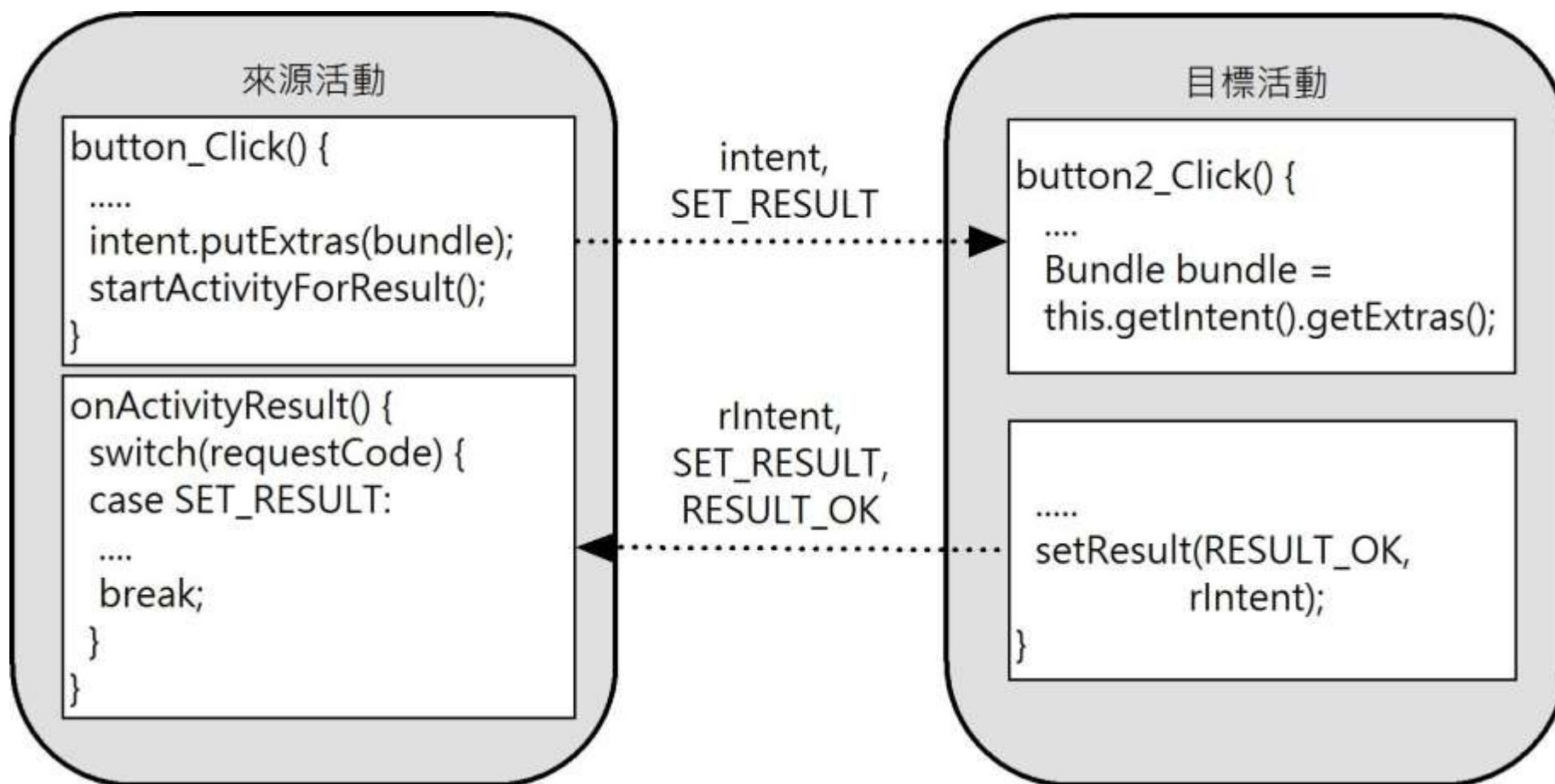
ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：
 - 覆寫來源活動onActivityResult()方法以取得回傳資料，
 - onActivityResult 3個參數是請求碼、結果碼和Intent物件
 - @Override
 - protected void onActivityResult(int requestCode,
 - int resultCode, Intent data) {
 - super.onActivityResult(requestCode,resultCode,data);
 - switch(requestCode) {
 - case SET_RESULT:
 -
 - }
 - break;



ANDROID 意圖(Intent)與程式生命週期

- 意圖（**intents**）取得活動的回傳資料：





ANDROID 意圖(Intent)與程式生命週期

- 開啟 **AndroidManifest.xml**
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:supportsRtl="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=".OpActivity"></activity>`
 - `</application>`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_main.xml`
 - `<Button`
 - `android:text="選擇運算子"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_below="@+id/txtOpd2"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:layout_marginTop="31dp"`
 - `android:id="@+id/button"`
 - `android:onClick="button_Click" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_op.xml`
 - `<RadioGroup`
 - `android:layout_width="match_parent"`
 - `android:layout_height="wrap_content"`
 - `android:layout_alignParentTop="true"`
 - `android:layout_alignParentLeft="true"`
 - `android:layout_alignParentStart="true"`
 - `android:orientation="horizontal"`
 - `android:id="@+id/rg">`
 - `<RadioButton`
 - `android:text="+"`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_op.xml`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:id="@+id/rdbAdd"`
 - `android:layout_weight="1" />`
 - `<RadioButton`
 - `android:text="-"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:id="@+id/rdbSubtract"`
 - `android:layout_weight="1" />`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 `activity_op.xml`

- `<RadioButton`
- `android:text="*"`
- `android:layout_width="wrap_content"`
- `android:layout_height="wrap_content"`
- `android:id="@+id/rdbMultiply"`
- `android:layout_weight="1" />`
- `<RadioButton`
- `android:text="/"`
- `android:layout_width="wrap_content"`
- `android:layout_height="wrap_content"`
- `android:id="@+id/rdbDivide"`
- `android:layout_weight="1" />`
- `</RadioGroup>`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_main);
 - // 取得TextView元件
 - output = (TextView) findViewById(R.id.lblOutput);
 - }
 - // Button元件的事件處理
 - public void button_Click(View view) {
 - EditText txtOpd1, txtOpd2;
 - // 取得EditText元件



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **MainActivity.java**
 - `txtOpd1 = (EditText) findViewById(R.id.txtOpd1);`
 - `txtOpd2 = (EditText) findViewById(R.id.txtOpd2);`
 - `// 建立Intent物件`
 - `Intent intent = new Intent(this, OpActivity.class);`
 - `// 建立傳遞資料的Bundle物件`
 - `Bundle bundle = new Bundle();`
 - `bundle.putString("OPERAND01",txtOpd1.getText().toString());`
 - `bundle.putString("OPERAND02",txtOpd2.getText().toString());`
 - `intent.putExtras(bundle); // 加上資料`
 - `// 啟動活動且有回傳資料`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java

- startActivityForResult(intent, SET_RESULT);
- }
- @Override
- protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
- super.onActivityResult(requestCode, resultCode, data);
- switch(requestCode) {
- case SET_RESULT:
- if (resultCode == RESULT_OK) {
- Bundle bundle = data.getExtras();
- output.setText("計算結果:" + bundle.getDouble("RESULT"));
- }
- break;
- }
- }



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **OpActivity.java**

- `public class OpActivity extends AppCompatActivity {`
- `@Override`
- `protected void onCreate(Bundle savedInstanceState) {`
- `super.onCreate(savedInstanceState);`
- `setContentView(R.layout.activity_op);`
- `}`
- `// Button元件的事件處理方法`
- `public void button2_Click(View view) {`
- `int opd1, opd2;`
- `double result = 0.0;`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **OpActivity.java**

- `RadioButton rdbAdd, rdbSubtract, rdbMultiply, rdbDivide;`
- `CheckBox chkDivide;`
- `// 取得傳遞的資料`
- `Bundle bundle = this getIntent().getExtras();`
- `if (bundle == null) return;`
- `// 取得運算元`
- `opd1 = Integer.parseInt(bundle.getString("OPERAND01"));`
- `opd2 = Integer.parseInt(bundle.getString("OPERAND02"));`
- `// 取得選取的運算子`
- `rdbAdd = (RadioButton) findViewById(R.id.rdbAdd);`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **OpActivity.java**

- `if (rdbAdd.isChecked()) {`
- `result = opd1 + opd2; // 加`
- `}`
- `rdbSubtract = (RadioButton) findViewById(R.id.rdbSubtract);`
- `if (rdbSubtract.isChecked()) {`
- `result = opd1 - opd2; // 減`
- `}`
- `rdbMultiply = (RadioButton) findViewById(R.id.rdbMultiply);`
- `if (rdbMultiply.isChecked()) {`
- `result = opd1 * opd2; // 乘`
- `}`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **OpActivity.java**

- `rdbDivide = (RadioButton) findViewById(R.id.rdbDivide);`
- `chkDivide = (CheckBox) findViewById(R.id.chkDivide);`
- `if (rdbDivide.isChecked()) {`
- `if (chkDivide.isChecked())`
- `result = opd1 / opd2; // 整數除法`
- `else`
- `result = opd1 / (double) opd2;`
- `}`
- `Intent rIntent = new Intent();`
- `// 建立傳回值`
- `Bundle rbundle = new Bundle();`



ANDROID 意圖(Intent)與程式生命週期

- 開啟 **OpActivity.java**
 - `rbundle.putDouble("RESULT", result);`
 - `rIntent.putExtras(rbundle);` // 加上資料
 - `setResult(RESULT_OK, rIntent);` // 設定傳回
 - `finish();` // 結束活動
 - `}`



ANDROID 意圖(INTENT)與程式生命週期

- 成果





ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖（implicit intents）：
 - 隱含意圖（implicit intent）只有指出執行的動作型態和資料，並沒有目標接收者的確實名稱
 - Android作業系統安裝的程式之中，有任何1個可以完成此工作的程式都可以是接收者
- 假設2個隱含意圖的動作類型
 - 打電話
 - 瀏覽網頁



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (implicit intents) :
 - 當送出Intent物件啟動程式後，Android作業系統的操作過程
 - 打電話動作類型
 - 系統從AndroidManifest.xml設定檔的描述中，找到【撥號】程式可以打電話，所以啟動此程式
 - 同時將電話號碼傳入程式，然後可以看到【撥號】程式的撥號畫面，撥出的就是傳入的電話號碼
 - 瀏覽網頁動作類型
 - 系統從AndroidManifest.xml設定檔的描述中，找到【瀏覽器】程式可以瀏覽網頁，所以啟動此程式
 - 同時將URL網址傳入程式，然後可以看到【瀏覽器】程式顯示URL網址的首頁



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖（**implicit intents**）：
 - 指定動作類型和資料來建立隱含意圖的Intent物件
 - 使用預先定義的動作類型，**ACTION_VIEW**
 - 為Intent-filter元素action子元素的android.intent.action.VIEW屬性值
 - 將ACTION_VIEW使用在隱含意圖
 - Intent意圖物件可以在建構子的第1個參數指定動作類型
 - `Intent i = new Intent(Intent.ACTION_VIEW);`



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (**implicit intents**) :
 - 另一種方式是在建立Intent物件後，再呼叫setAction()方法指定動作類型
 - Intent i = new Intent();
 - i.setAction(Intent.ACTION_VIEW);
 - 指定目標的資料 (Data)
 - Intent建構子的第2個參數是URI (Universal Resource Identifier) 字串
 - Intent i = new Intent(Intent.ACTION_VIEW,
 - Uri.parse("http://www.google.com.tw"));
 - URL網址字串呼叫Uri.parse()方法建立成URI



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (implicit intents) :
 - 隱含意圖常用動作類型

動作類型	定義
ACTION_VIEW	顯示資料給使用者檢視
ACTION_EDIT	顯示資料給使用者編輯
ACTION_DIAL	顯示撥號
ACTION_CALL	打電話
ACTION_PICK	選取URI目錄下的資料
ACTION_SENDTO	寄送電子郵件
ACTION_WEB_SEARCH	Web搜尋
ACTION_MAIN	啟動如同是程式進入點的主程式



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (**implicit intents**) :
 - 或使用Intent物件的setData()方法來指定
 - Intent i = new Intent(Intent.ACTION_VIEW);
 - i.setData(Uri.parse("http://www.google.com.tw"));
 - 萬用資源識別URI
 - 用來定位Android系統的資源，幫助Intent意圖物件的動作取得或找到操作的資料



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (**implicit intents**) :
 - Android 常用的 URI
 - URL網址：URI可以直接使用URL網址
 - <http://www.google.com.tw/>
 - 地圖位置：GPS定位的座標值 (GeoPoint格式)
 - geo:25.04692437135412,121.5161783959678
 - 電話號碼：指定撥打的電話號碼
 - tel:+1234567
 - 寄送郵件：寄送郵件至指定的電子郵件地址
 - mailto:hueyan@ms2.hinet.net



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (implicit intents) :
 - Intent04 範例程式
 - 活動啟動內建瀏覽器：建立ACTION_VIEW動作，URI為URL網址的
 - Intent i = new Intent(Intent.ACTION_VIEW,
 - Uri.parse("http://www.google.com.tw"));
 - startActivity(i);
 - 活動啟動內建Google地圖：建立ACTION_VIEW動作，URI為GPS座標值
 - Intent i = new Intent(Intent.ACTION_VIEW,Uri.parse(
 - "geo:25.04692437135412,121.5161783959678"));
 - startActivity(i);



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖 (implicit intents) :
 - Intent04 範例程式
 - 活動啟動內建撥號程式：建立ACTION_DIAL動作，
URI為電話號碼
 - Intent i = new Intent(Intent.ACTION_DIAL);
 - i.setData(Uri.parse("tel:+1234567"));
 - startActivity(i);



ANDROID 意圖(Intent)與程式生命週期

• 成果





ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖的資料傳遞:
 - 隱含意圖啟動的程式一樣可以傳遞資料到開啟的活動
 - Intent05範例程式
 - 在Intent物件附加資料，啟動內建瀏覽器執行Web搜尋
 - 使用ACTION_WEB_SEARCH動作
 - Intent i = new Intent(Intent.ACTION_WEB_SEARCH);
 - i.putExtra(SearchManager.QUERY, "Android");
 - startActivity(i);



ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖的資料傳遞:
 - 活動選取聯絡人後，建立Intent物件顯示選取聯絡人的詳細資料
 - 使用ACTION_PICK動作建立Intent物件
 - 使用Intent物件的setType()方法指定回傳資料的MIME型態
 - CONTENT_TYPE是一種CONTENT_URI，提供聯絡人清單來執行選取動作
 - 請在【通訊錄】程式新增一些聯絡人資料
 - Intent i = new Intent(Intent.ACTION_PICK);
 - i.setType(ContactsContract.Contacts.CONTENT_TYPE);
 - startActivityForResult(i, GET_CONTACT);



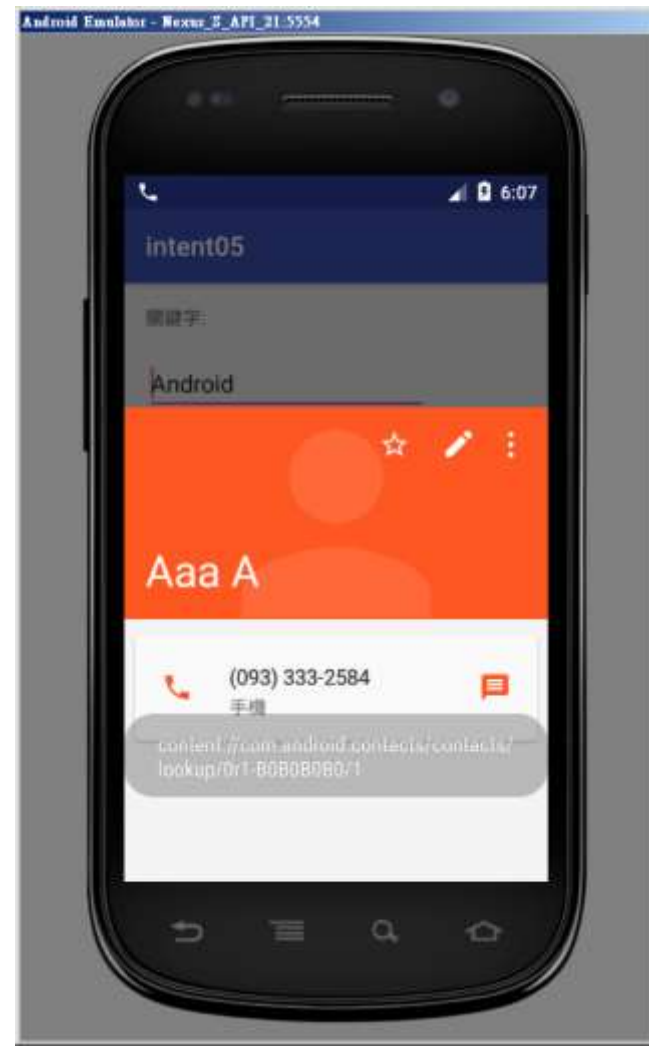
ANDROID 意圖(Intent)與程式生命週期

- 隱含意圖的資料傳遞:
 - 啟動有回傳資料的活動，需要覆寫onActivityResult()方法來取回資料
 - @Override
 - protected void onActivityResult(int requestCode,
 - int resultCode, Intent data) {
 - super.onActivityResult(requestCode,resultCode, data);
 - if (requestCode == GET_CONTACT) {
 - if (resultCode == RESULT_OK) {
 -
 - }



ANDROID 意圖(Intent)與程式生命週期

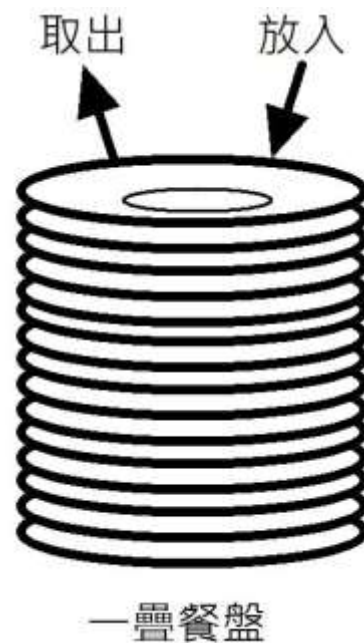
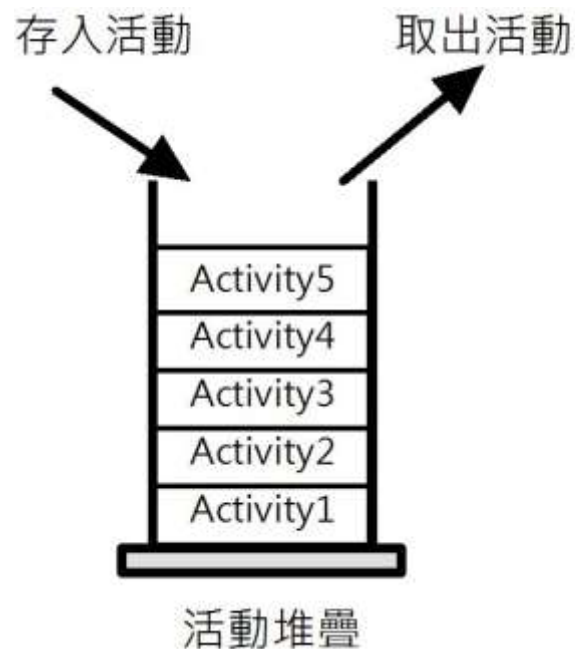
• 成果





ANDROID 意圖(Intent)與程式生命週期

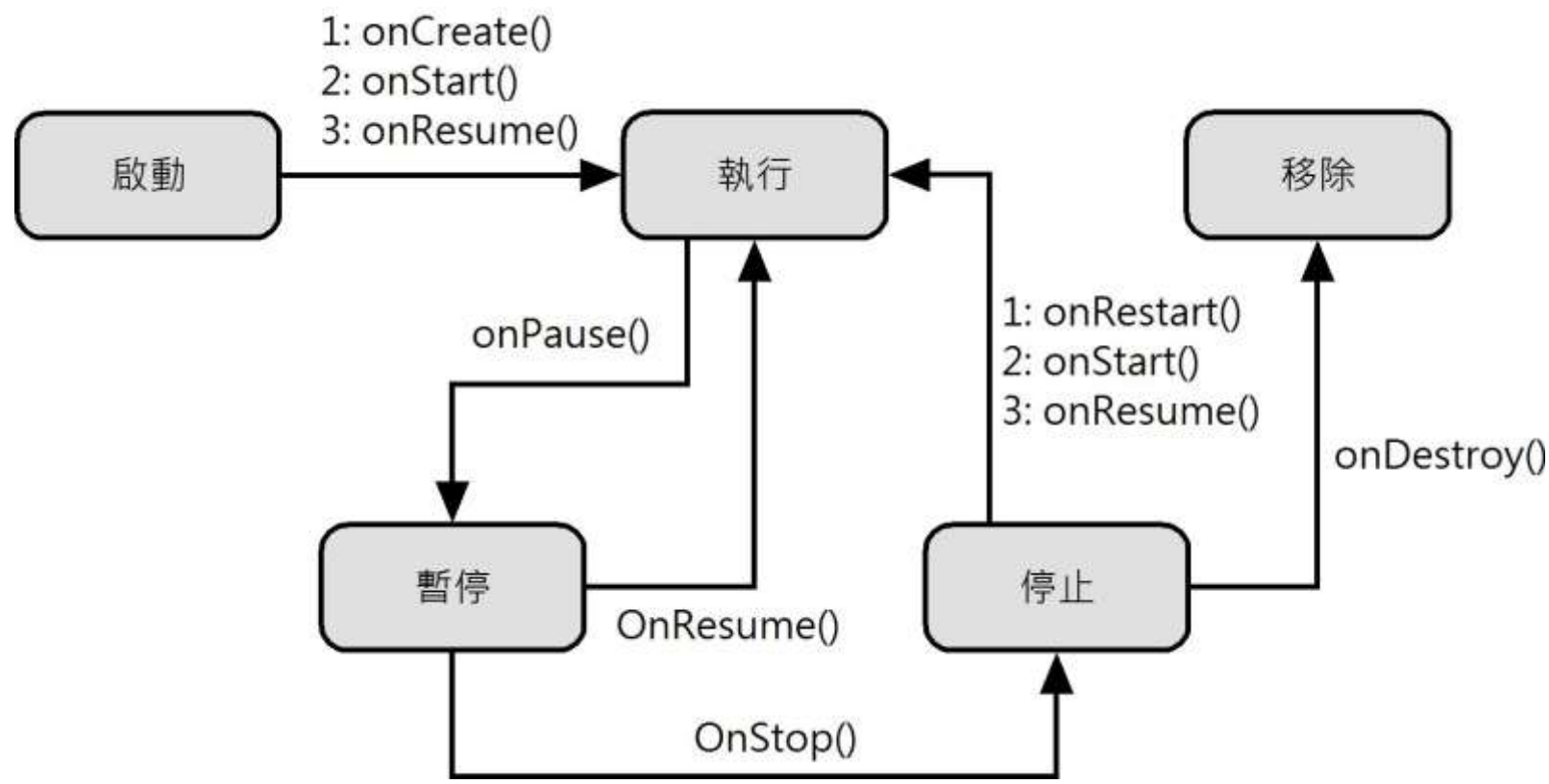
- 活動的生命周期:
 - Android作業系統使用活動堆疊（ activity stack ）來管理啟動中的眾多活動
 - 如同廚房工人清洗餐盤，將洗好的餐盤疊在一起，每洗好一個餐盤就放在這疊餐盤的頂端





ANDROID 意圖(INTENT)與程式生命週期

- 活動的生命周期:
 - 活動的生命周期重點在活動的狀態管理
 - 利用程式碼來讓Android APP在不同狀態之間轉換





ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:
 - 當Android應用程式不在記憶體中時，執行程式就是進入啟動狀態
 - 依序呼叫onCreate()、onStart()和onResume()方法進入執行狀態，這三個方法為「回呼方法」(callback methods)
 - 利用回呼方法撰寫程式碼來回應活動狀態的改變
 - **Activity**在執行狀態，表示它目前在螢幕上顯示且與使用者進行互動，對比Windows作業系統，就是視窗取得焦點 (focus)



ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:
 - **Android**作業系統在任何時間都只會有一個位在執行狀態的**Activity**
 - 該 **Activity** 擁有最高權限來使用記憶體與資源來提昇執行效率，以便能夠更快速的回應使用者操作
 - 暫停狀態是指活動沒有取得焦點，也沒有與使用者互動，但是仍然顯示在螢幕上
 - 例如顯示對話方塊時並不會佔用整個螢幕，背景的活動仍然看的見
 - 此時呼叫**onPause()**方法，從執行狀態轉換至暫停狀態



ANDROID 意圖(INTENT)與程式生命週期

- 活動的生命周期:
 - 停止狀態的活動仍然保留在記憶體中，只是使用者看不到它
 - **Android**作業系統之所以將活動保留在記憶體，因為使用者有可能不久就會再返回此活動
 - 從停止狀態回到執行狀態，比起從啟動狀態轉換至執行狀態耗費較少的資源



ANDROID 意圖(Intent)與程式生命週期

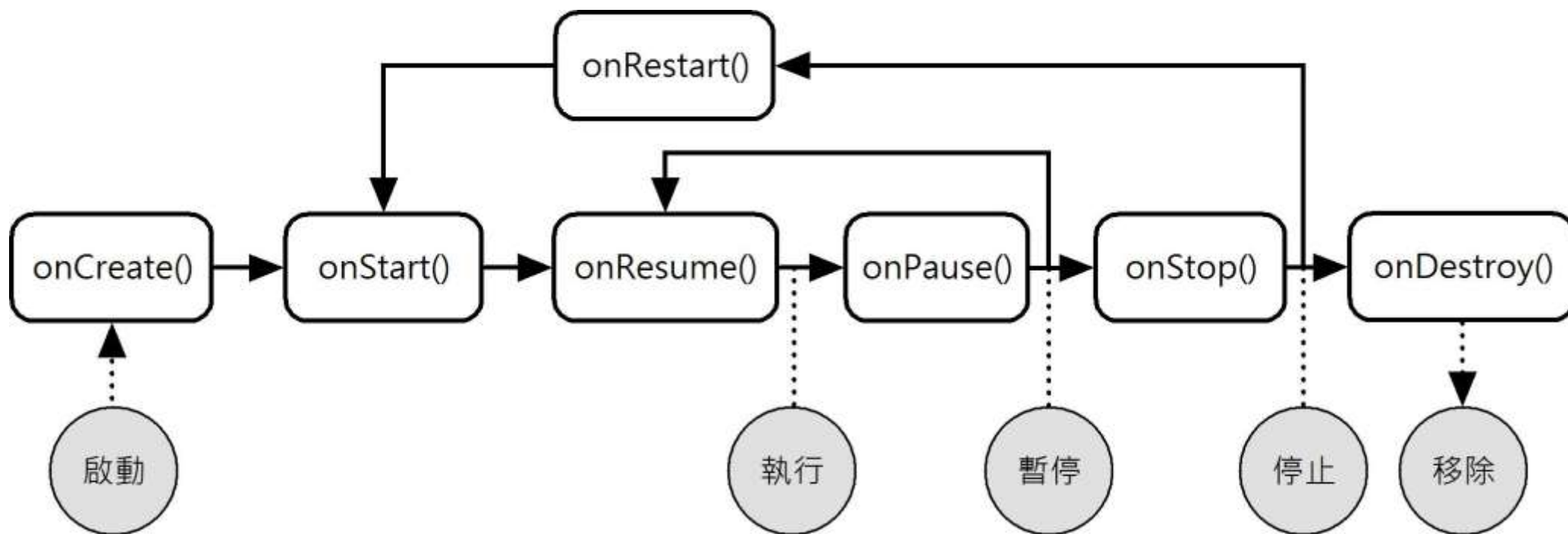
- 活動的生命周期:
 - 活動在移除狀態，表示已經釋放活動佔用的資源，活動已經刪除且不存在記憶體之中
 - 活動管理員會依據記憶體的使用情況，決定活動是否需要刪除
 - 刪除活動可空出更多記憶體空間讓執行狀態的活動能夠正常的運作
- 對**Android**行動裝置使用者來說，活動狀態只有三種：
 - 可見（執行與暫停）
 - 取得焦點（執行）
 - 不可見（停止與移除）



ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:

- 活動生命周期共有7個方法可在活動的狀態轉換時呼叫，
- 利用這些方法回應狀態改變，開發、撰寫Java程式碼





ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:

方法	說明
onCreate()	在活動建立時呼叫，可以在此方法執行活動的初始化，即建立活動的使用介面元件，我們可以將此方法視為活動的進入點
onStart()	在使用者被看見時呼叫
onResume()	在與使用者互動時呼叫
onPause()	在暫停目前活動時呼叫，例如：顯示對話方塊，通常我們會在此方法儲存尚未儲存的資料，和任何使用者變更的資料
onStop()	在使用者看不見時呼叫，可能情況有三種：啟動新活動、之前活動返回螢幕或活動將被刪除
onRestart()	在活動重新返回螢幕時呼叫，例如：收到簡訊，停止目前的活動，等到閱讀完簡訊後，呼叫此方法返回之前的活動
onDestroy()	在刪除活動前呼叫



ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:
 - Intent06範例程式
 - 在Android應用程式建立活動的7個方法來測試活動的生命周期
 - 使用Log.d()方法顯示呼叫過程的訊息文字
 - Android Studio下方「Android Monitor」視窗的【LogCat】標籤檢視輸出訊息



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - public class MainActivity extends AppCompatActivity {
 - private static final String TAG = "intent06";
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_main);
 - Log.d(TAG, "MainActivity:onCreate");
 - }
 - @Override
 - protected void onStart() {
 - super.onStart();
 - Log.d(TAG, "MainActivity:onStart");
 - }



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - @Override
 - protected void onResume() {
 - super.onResume();
 - Log.d(TAG, "MainActivity:onResume");
 - }
 - @Override
 - protected void onStop() {
 - super.onStop();
 - Log.d(TAG, "MainActivity:onStop");
 - }
 - @Override
 - protected void onPause() {



ANDROID 意圖(Intent)與程式生命週期

- 開啟 MainActivity.java
 - `Log.d(TAG, "MainActivity.onPause");`
 - `}`
 - `@Override`
 - `protected void onRestart() {`
 - `super.onRestart();`
 - `Log.d(TAG, "MainActivity.onRestart");`
 - `}`
 - `@Override`
 - `protected void onDestroy() {`
 - `super.onDestroy();`
 - `Log.d(TAG, "MainActivity.onDestroy");`
 - `}`



ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:
 - 操作一
 - 請在Android模擬器按右方工具列的第4個按鈕向左旋轉螢幕，因為旋轉螢幕會結束活動後重新啟動
 - 在【LogCat】標籤可以看到首先呼叫onPause()、onStop()和onDestroy()方法結束活動
 - 再呼叫onCreate()、onStart()和onResume()方法啟動活動



ANDROID 意圖(Intent)與程式生命週期

• 成果

The screenshot displays the Android Studio 2.3.3 interface. The top toolbar shows various icons for file operations, navigation, and running the application. The main editor area shows the `AndroidManifest.xml` file for the `Intent06` project. The file content is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.intent06" />
```

The Android Monitor at the bottom shows the log for the `com.example.android.intent06` application. The log entries are as follows:

```
12-10 18:37:25.980 8038-8038/com.example.android.intent06 W/art: Before Android 4.1, method int android.support.v7.widget.ListViewCompat.lookForSelectable
12-10 18:37:47.197 8038-8038/com.example.android.intent06 D/intent06: MainActivity.onPause
12-10 18:37:47.203 8038-8038/com.example.android.intent06 D/intent06: MainActivity.onStop
12-10 18:37:47.203 8038-8038/com.example.android.intent06 D/intent06: MainActivity.onDestroy
12-10 18:37:47.261 8038-8038/com.example.android.intent06 D/intent06: MainActivity.onCreate
12-10 18:37:47.264 8038-8038/com.example.android.intent06 D/intent06: MainActivity.onStart
12-10 18:37:47.264 8038-8038/com.example.android.intent06 D/intent06: MainActivity.onResume
12-10 18:37:47.319 8038-8055/com.example.android.intent06 D/EGL_emulation: eglMakeCurrent: 0x7eff295cf200: ver 2 0
```

The bottom status bar shows the current build and installation status: "Performing full build and install: // Instant Run detected that a resource referenced from the merged AndroidManifest has changed... (3 minutes ago) 38:1 CRLF UTF-8 Context: <no context>".



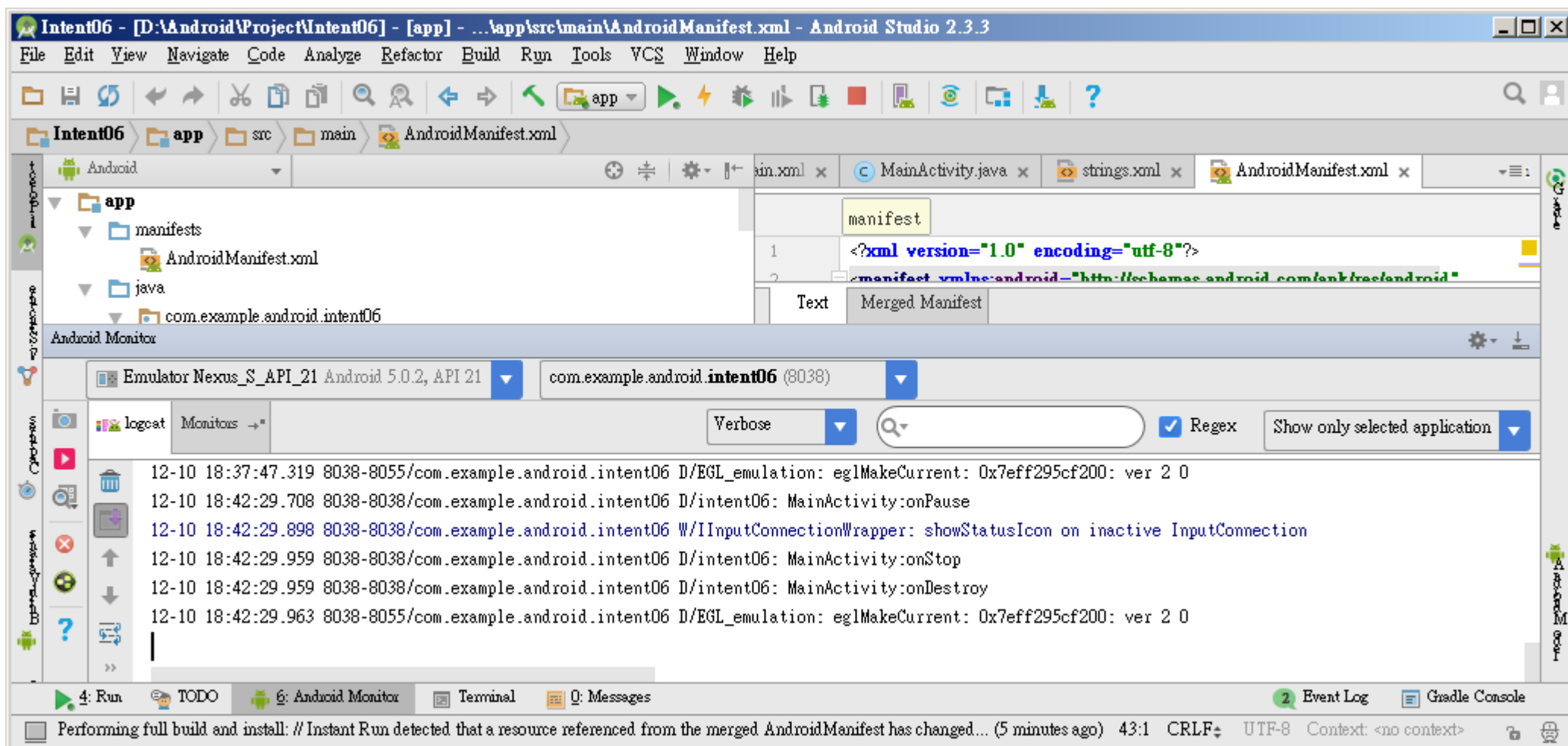
ANDROID 意圖(Intent)與程式生命週期

- 活動的生命周期:
 - 操作二
 - 在Android模擬器按【返回】鍵離開程式
 - 在【LogCat】標籤看到呼叫OnPause()、OnStop()和OnDestroy()方法結束活動



ANDROID 意圖(Intent)與程式生命週期

• 成果





 **THE END**