

ĐỀ CƯƠNG BÀI GIẢNG

BÀI 6: THỰC HÀNH ỨNG DỤNG VỚI MENU VÀ INTENT

Nội dung bài học trước khi lên lớp (trang 1 đến trang 13):

- Thực hành sử dụng menu : Bài 6.1
- Thực hành cách sử dụng intent cơ bản : Bài 6.3

Nội dung bài học thực hiện lên lớp (trang 14 đến hết):

- Thực hành sử dụng menu ngữ cảnh (Context menu) : Bài 6.2
- Thực hành sử dụng truyền dữ liệu giữa 2 Intent với Bundle: Bài 6.4

Nội dung bài học sau khi lên lớp: Làm bài tập 6.5

NỘI DUNG BÀI HỌC

1. Tổng quan về menu trong Android	1
1.1. Option menu	2
Thao tác với MainActivity.java	8
1.2. Context Menu	10
2. Thực hành với ListView.....	13
2.1. ListView đơn giản	Error! Bookmark not defined.
2.2. Tùy biến ListView	13
3. Spinner.....	Error! Bookmark not defined.
4. TimePicker – DatePicker	Error! Bookmark not defined.

1. Tổng quan về menu trong Android

Menu là một thành phần giao diện người dùng phổ biến trong nhiều loại ứng dụng. Để cung cấp một trải nghiệm người dùng quen thuộc và nhất quán, bạn nên sử dụng các API Menu để trình bày hành động người dùng và các tùy chọn khác trong hoạt động của mình.

Bắt đầu với Android 3.0 (API mức 11), các thiết bị dựa trên nền tảng Android không còn phải cung cấp một nút Menu chuyên dụng nữa. Với sự thay đổi này, các ứng

dụng Android cần tránh khỏi sự phụ thuộc vào bảng điều khiển menu 6 mục truyền thống này mà thay vào đó cung cấp một thanh hành động để trình bày các hành động người dùng thông dụng.

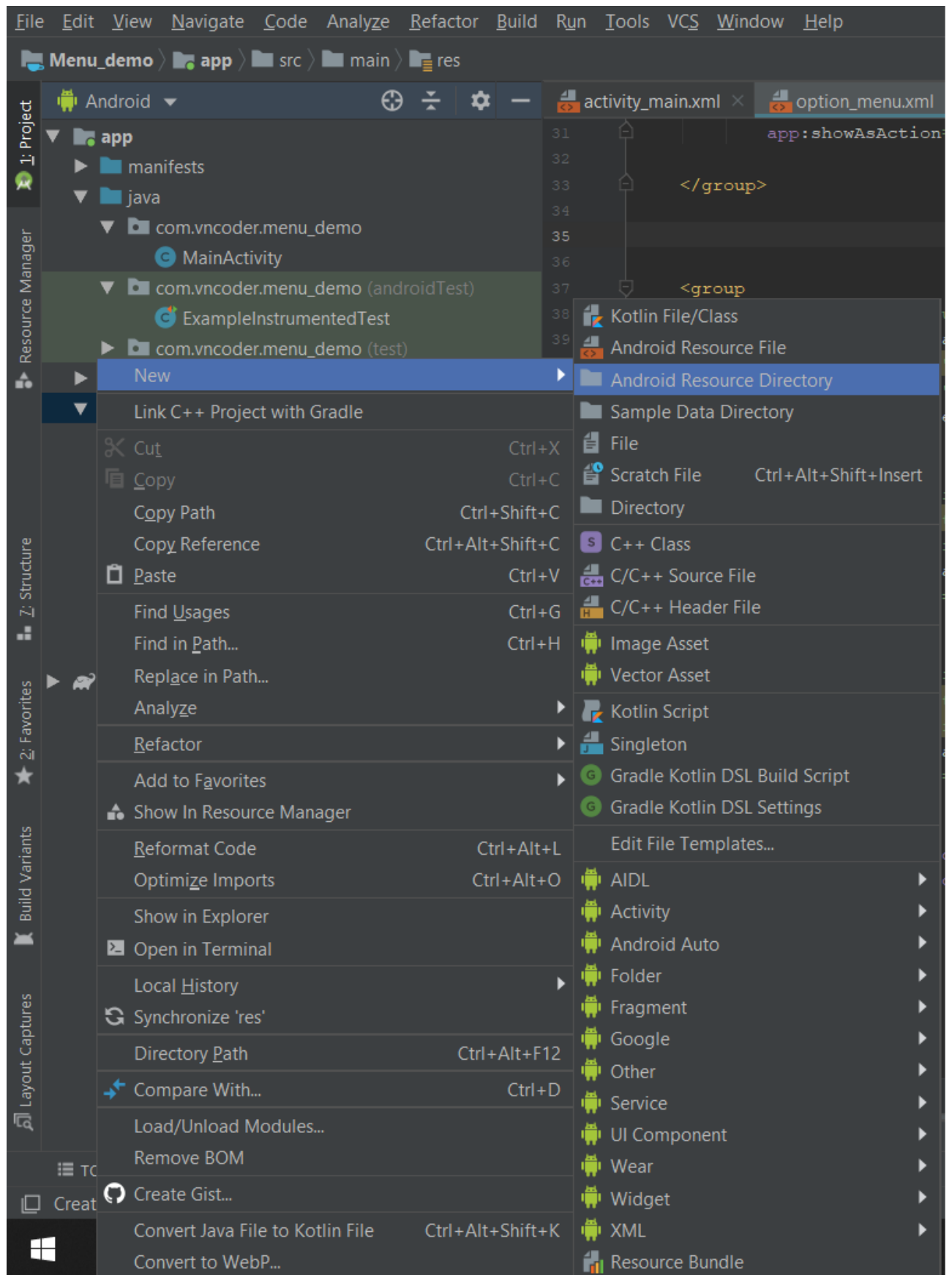
Menu trong Android có 2 loại menu chính là : `OptionsMenu` và `ContextMenu`.

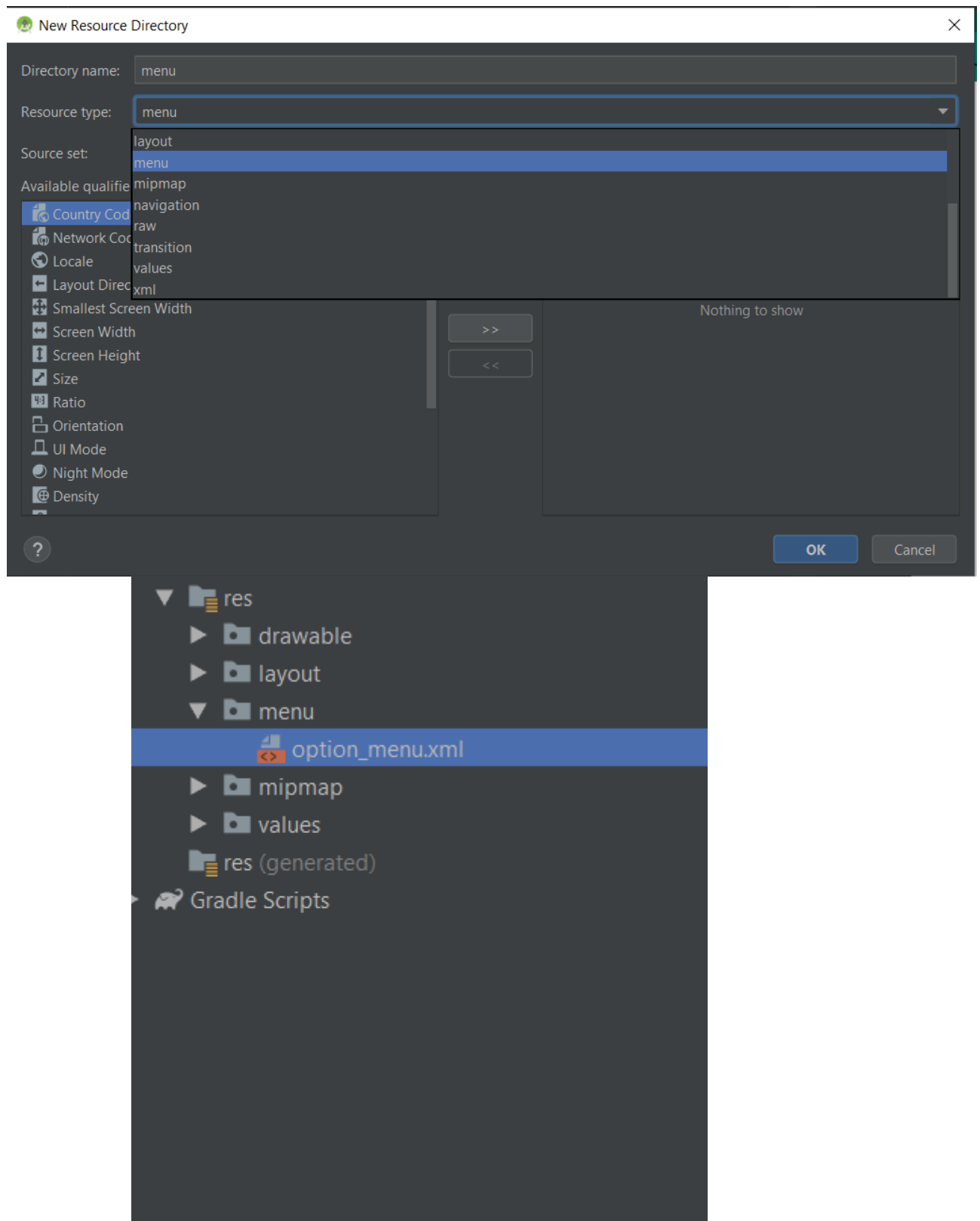
1.1. Option menu

Option Menu là loại menu nằm trên `ActionBar` hoặc `Toolbar` ở phía bên phải. Menu hiển thị dưới dạng text, text+icon hoặc icon “ba chấm”, menu cũng hỗ trợ tùy biến giao diện.

Để tạo menu đầu tiên bạn vào click chuột phải vào mục res - New - Android

Resource Directory để tạo một thư mục chứa menu và lưu ý là bạn phải đặt chính xác tên thư mục là menu.





Sau khi hoàn thành ta có thư mục menu chứa file option_menu.xml
Từ sau phiên bản android 3.0 và API > 11 thì android chuyển sang hiển thị
OptionsMenu trên thanh ActionBar thay vì hiển thị dưới dạng 6 nút cổ điển như
trước đây dù vậy việc thực thi tạo một menu vẫn không thay đổi.

Để tạo các ra menu ta dùng các thẻ <item>

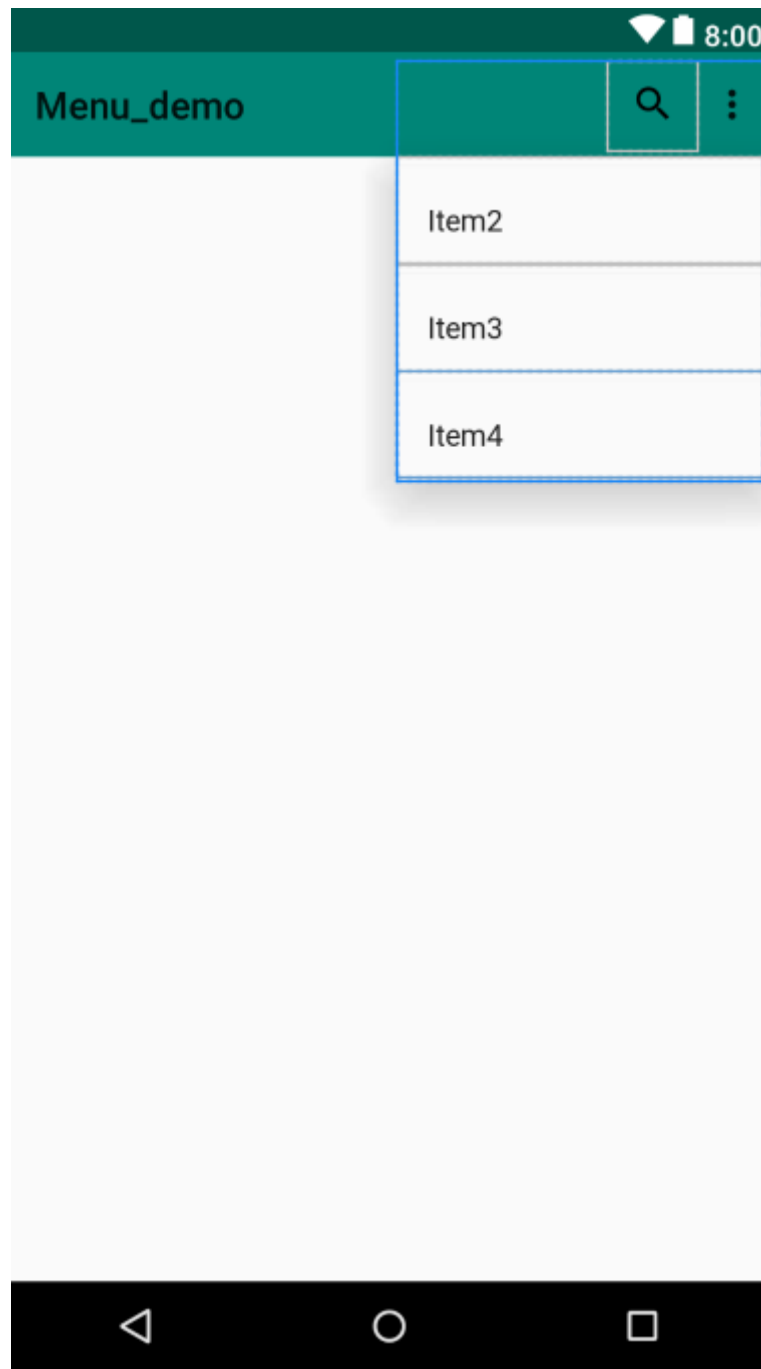
Trong thẻ item có những thuộc tính ta cần lưu ý :

1. id : dùng để tham chiếu tới item
2. title : tiêu đề của item
3. icon : hiển thị icon

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item android:id="@+id/item1"
        android:title="Item1"
        android:icon="@drawable/ic_search_black_24dp"
        android:orderInCategory="100"
        app:showAsAction="ifRoom|collapseActionView"
        app:actionViewClass="android.widget.SearchView"
        />
    <item android:id="@+id/item2"
        android:title="Item2"
        android:icon="@drawable/ic_people_black_24dp"
        android:orderInCategory="100"
        app:showAsAction="withText" />

    <item android:id="@+id/item3"
        android:title="Item3"
        android:icon="@drawable/ic_settings_black_24dp"
        android:orderInCategory="100"
        app:showAsAction="withText" />
    <item android:id="@+id/item4"
        android:title="Item4"
        android:icon="@drawable/ic_photo_camera_black_24dp"
        android:orderInCategory="100"
        app:showAsAction="withText" />
</menu>
```



Tạo menu con trong menu.

```
<group
    android:id="@+id/group1"
    android:checkableBehavior="none">

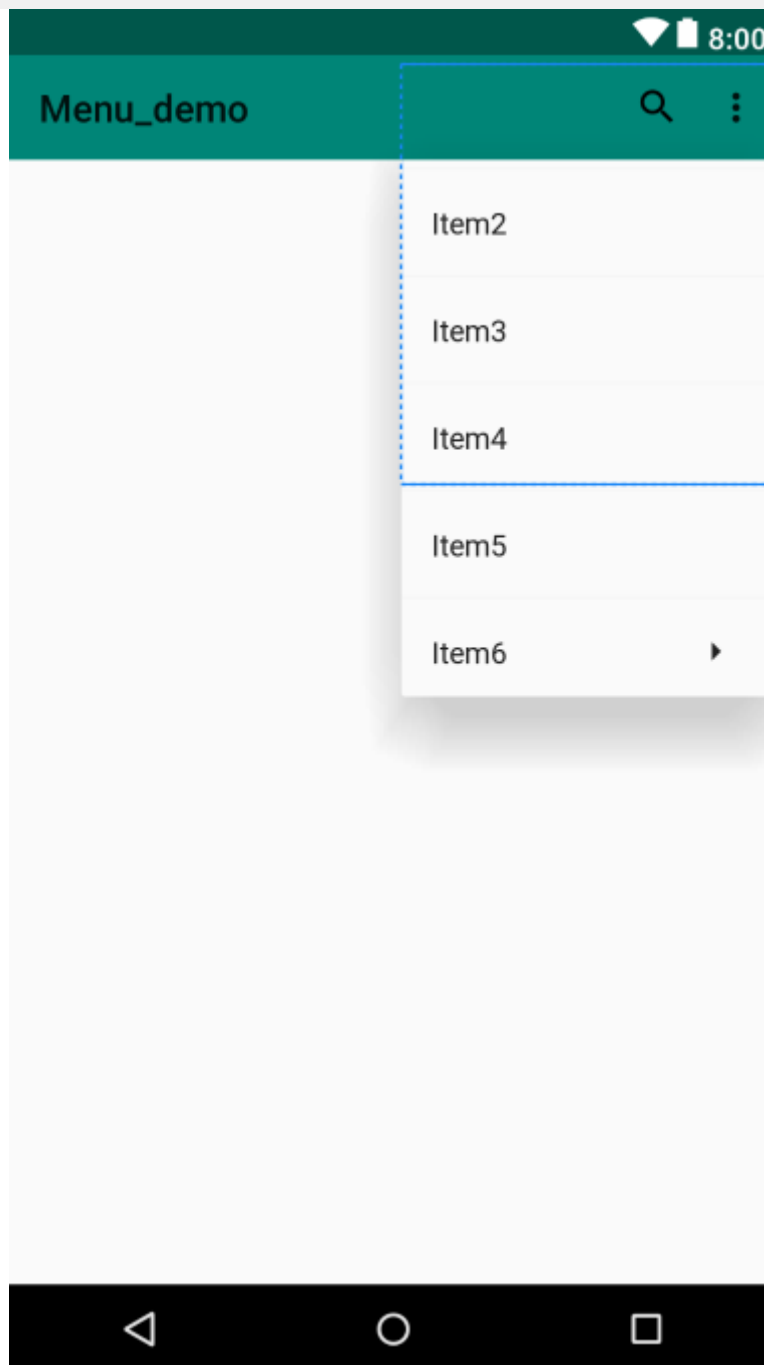
    <item android:id="@+id/item5"
        android:title="Item5"
        android:icon="@drawable/ic_security_black_24dp"
        android:orderInCategory="100"
        app:showAsAction="withText" />

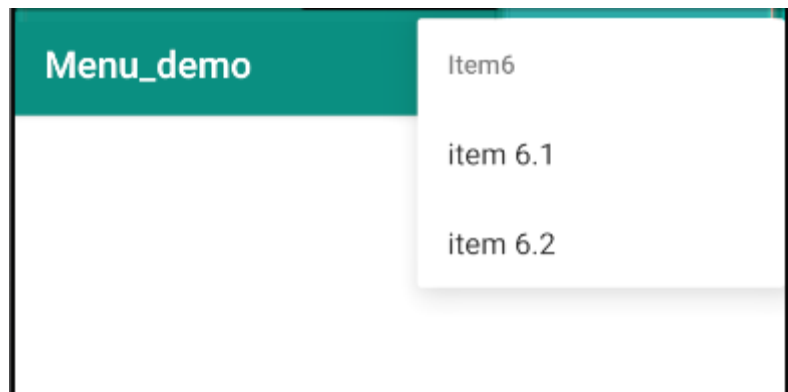
    <item android:id="@+id/item6"
```

```
android:title="Item6"
android:icon="@drawable/abc_vector_test"
android:orderInCategory="100"
app:showAsAction="collapseActionView" >

<menu>
  <item android:id="@+id/item6.1" android:title="item 6.1"/>
  <item android:id="@+id/item6.2" android:title="item 6.2"/>
</menu>

</item>
</group>
```





Thao tác với MainActivity.java

Trong **MainActivity.java** ta tiến hành Override hàm `onOptionsItemSelected` để tham chiếu tới menu

Gọi hàm `getMenuInflater()` để tạo một `MenuInflater` (`MenuInflater` không thể khởi tạo với "`new MenuInflater()`" mà dùng hàm `getMenuInflater()`).

@Override

```
public boolean onOptionsItemSelected(Menu menu) {  
    getMenuInflater().inflate(R.menu.option_menu, menu);  
    MenuItem menuItem = menu.findItem(R.id.item1);  
    SearchView searchView = (SearchView)  
MenuItemCompat.getActionView(menuItem);  
    return super.onOptionsItemSelected(menu);  
}
```

Tiếp theo ta tiến hành cho `MenuInflater().inflate(R.menu.mymenu, menu)` . Trong hàm `inflate()` tham số thứ nhất là menu cần lấy ra để tạo, tham số thứ 2 là `OptionsMenu`. Vậy là chúng ta đã tạo xong một `OptionsMenu`, còn những dòng code có comment là dùng để lấy ra `ActionView` của một Item, ở đây mình lấy ra cái `SearchView` của item đăng ký.

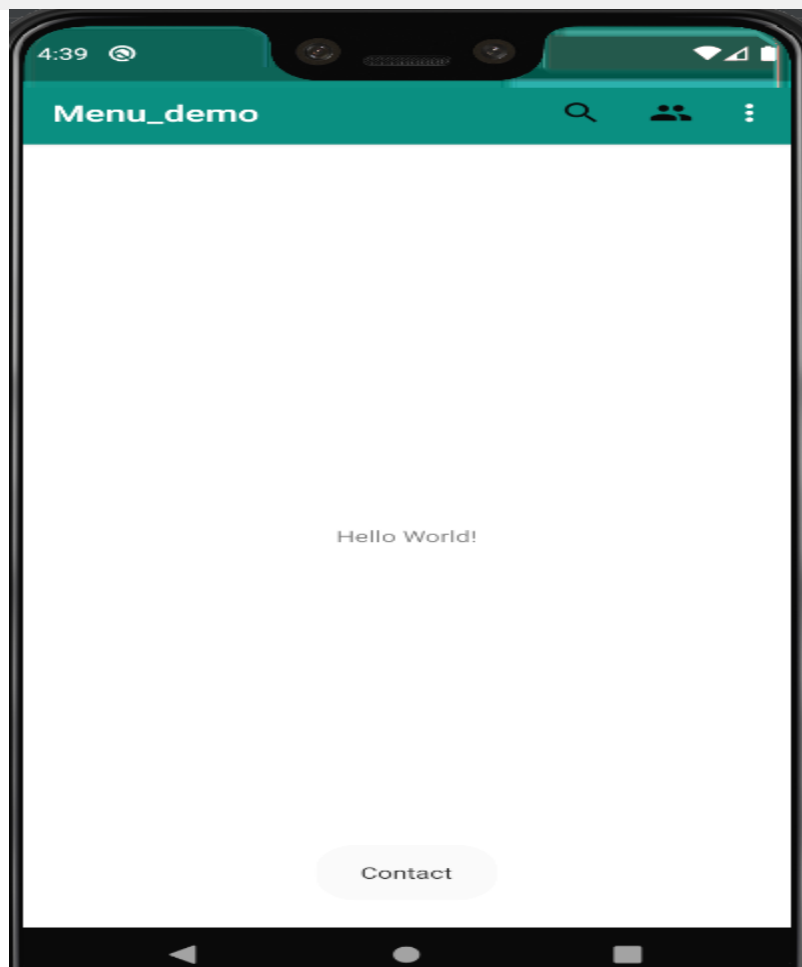
Để gán sự kiện cho các item trong menu ta tiến hành Override hàm `onOptionsItemSelected`.

@Override

```
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
    int id = item.getItemId();  
    switch (id){  
        case R.id.item2:  
  
Toast.makeText(getBaseContext(), "item2", Toast.LENGTH_LONG).show();  
        break;  
        case R.id.item3:  
  
Toast.makeText(getBaseContext(), "item3", Toast.LENGTH_LONG).show();  
        break;  
        case R.id.item4:
```



```
Toast.makeText(getBaseContext(), "item4", Toast.LENGTH_LONG).show();
    break;
    case R.id.item5:
        Toast.makeText(getBaseContext(), "item5", Toast.LENGTH_LONG).show();
        break;
    case R.id.item6:
        Toast.makeText(getBaseContext(), "item6", Toast.LENGTH_LONG).show();
        break;
    case R.id.item6_1:
        Toast.makeText(getBaseContext(), "item6.1", Toast.LENGTH_LONG).show();
        case R.id.item6_2:
            Toast.makeText(getBaseContext(), "item6.2", Toast.LENGTH_LONG).show();
            default:
        }
    return super.onOptionsItemSelected(item);
}
```

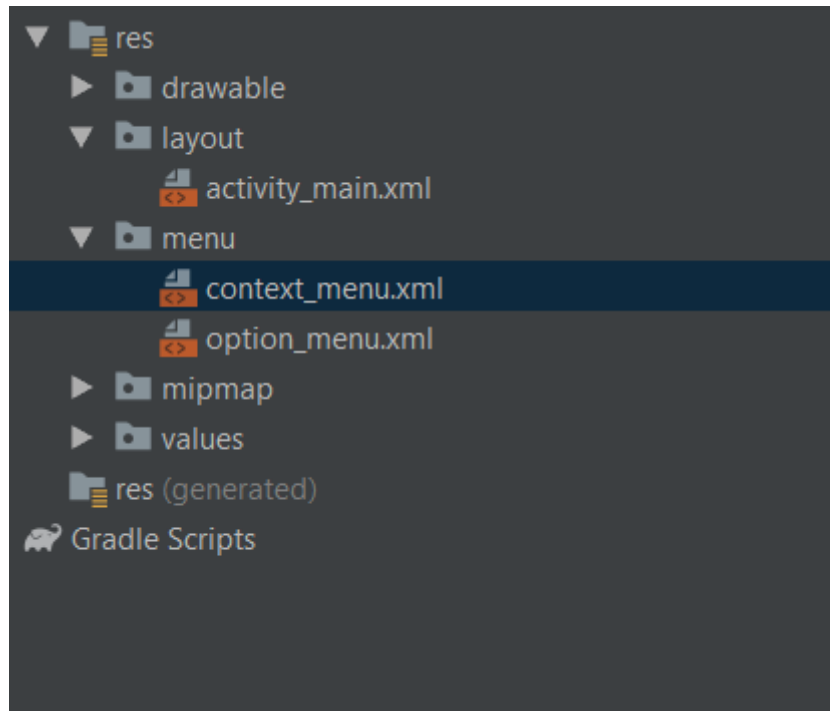


1.2. Context Menu

Context menu xuất hiện khi người dùng tác động lên một view nào đó. Chẳng hạn khi người dùng long click vào **Listview**, một vài thao tác được hiển thị để người dùng lựa chọn (context menu).



Giống như **OptionsMenu** ta cũng cần phải tạo một file menu đặt tên là **context_menu** như ở trên.

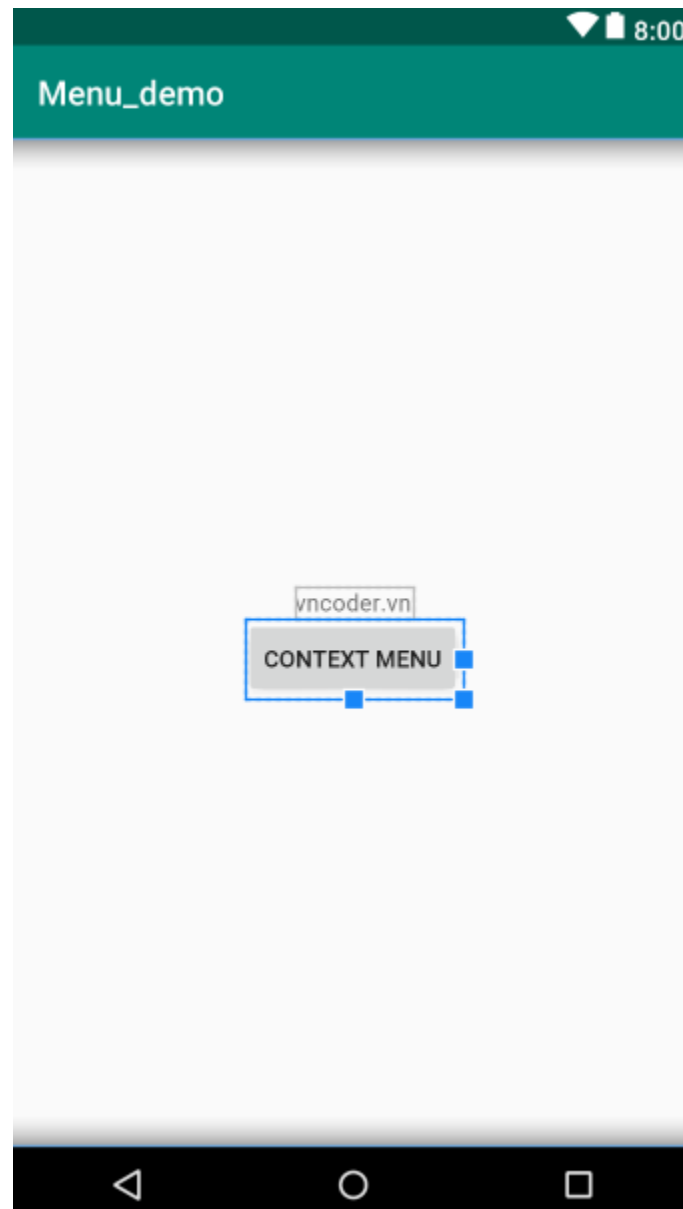
**File XML :**

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item android:id="@+id/context1"
    android:title="context menu 1"
    android:icon="@drawable/ic_settings_black_24dp"/>

<item android:id="@+id/context2"
    android:title="context menu 2"
    android:icon="@drawable/ic_settings_black_24dp"/>

</menu>
```

Trong giao diện của **MainActivity** ta thêm 1 nút như hình :



Ở trong Hàm **MainActivity.java** chúng ta thêm 2 hàm để định nghĩa **ContextMenu** và **view** của nó.

@Override

```
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    if (v == btn){
        getMenuInflater().inflate(R.menu.context_menu,menu);
    }
}
```

@Override

```
public boolean onContextItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()){
        case R.id.context1:
```

```
Toast.makeText(getApplicationContext(),"context1",Toast.LENGTH_LONG).show();  
    }  
    return super.onContextItemSelected(item);  
}
```

Không quên trong hàm **onCreate** thêm **registerForContextMenu(btn)**; để nhận view.

ContextMenu còn được ứng dụng trong nhiều trường hợp khác như hiển thị thông tin của một phần tử **listview**, thêm lựa chọn cho các button

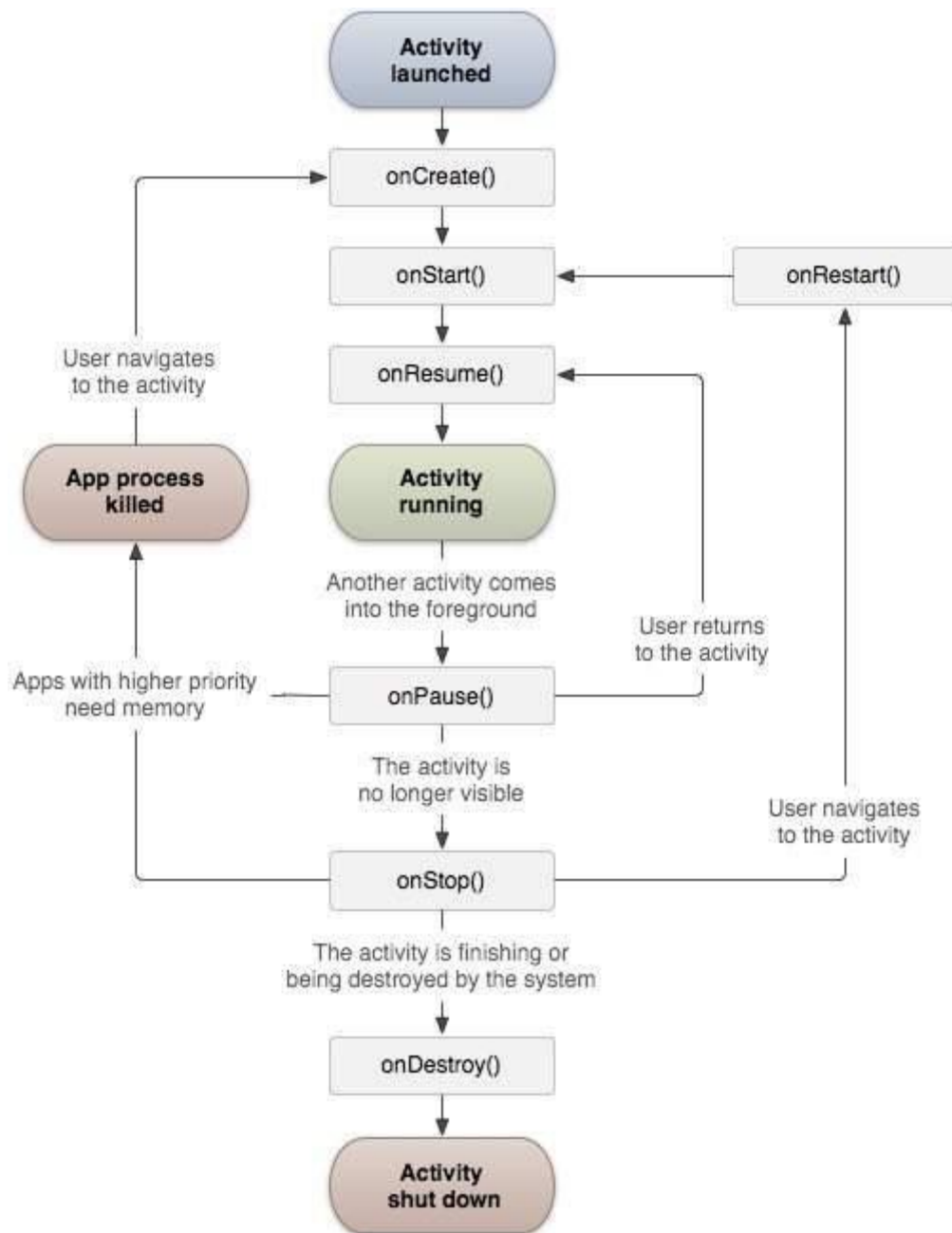
2. Thực hành truyền dữ liệu giữa 2 Intent

2.1. Vòng đời Intent

Một **Activity** trong Android là một màn hình giao diện người dùng của ứng dụng Android. Một ứng dụng Android có thể chứa **một hoặc nhiều Activity**, nghĩa là một hoặc nhiều màn hình giao diện. Ứng dụng Android bắt đầu bằng cách hiển thị **Activity chính** (Main Activity) và có thể mở thêm **nhiều màn hình Activity khác** tùy theo chức năng của ứng dụng.

a) Vòng đời của Activity (Activity Life Cycle)

Mọi Activity đều trải qua một **vòng đời (life cycle)** trong ứng dụng Android. Vòng đời của Activity trong Android được minh họa như hình dưới đây:



Hoạt động của những sự kiện gọi lại (**call backs**) trong **Activity life cycle** được mô tả như hình dưới đây:

TT	Callback & Mô tả
	<p>onCreate () Đây là callback đầu tiên và được gọi khi Activity được tạo lần đầu tiên.</p>

2	onStart() callback này được gọi khi Activity bắt đầu hiển thị cho người dùng.
3	onResume () callback được gọi khi người dùng bắt đầu tương tác với ứng dụng.
4	onPause () Activity bị tạm dừng không nhận được tương tác của người dùng.
5	onStop() callback này được gọi khi Activity không còn hiển thị (Ví dụ khi NSD tắt màn hình điện thoại hoặc chuyển sang Activity khác của ứng dụng)
6	onDestroy () callback này được gọi trước khi Activity bị hủy bỏ bởi hệ thống.
7	onRestart () callback này được gọi khi Activity khởi động lại sau khi bị dừng.

b) B. Ví dụ thực hành chạy Activity trong ứng dụng Android

Ví dụ này sẽ hướng dẫn bạn các bước đơn giản để **hiển thị vòng đời Activity** trong ứng dụng Android. Các bước thực hiện như sau:

B ước	Mô tả
1	Bạn sử dụng Android Studio để tạo một ứng dụng Android mới và đặt tên là <i>HelloWorld</i> theo gói <i>com.example</i>
2	Sửa đổi tệp Activity chính <i>MainActivity.java</i> như được giải thích bên dưới. Giữ phần còn lại của các tập tin không đổi



3

Chạy ứng dụng để xem kết quả.

Sau đây là nội dung của tệp Activity chính được sửa đổi **src / com.example.helloworld / MainActivity.java** . Tập tin này bao gồm các callback vòng đời cơ bản của Activity trong Android. Phương thức **Log.d()** được sử dụng để tạo thông báo log ra màn hình kết quả (console):

```
package com.example.helloworld;

import android.os.Bundle;
import android.app.Activity;
import android.util.Log;

public class MainActivity extends Activity {

    String msg = "Android : ";

    /** Called when the activity is first created. */
    @Override
    public void onCreate (Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        Log.d(msg, "The onCreate() event");
    }

    /** Called when the activity is about to become visible. */
    @Override
    protected void onStart () {

        super.onStart();

        Log.d(msg, "The onStart() event");
    }
}
```



```
}

/** Called when the activity has become visible. */

@Override

protected void onResume() {

    super.onResume();

    Log.d(msg, "The onResume() event");

}

/** Called when another activity is taking focus. */

@Override

protected void onPause() {

    super.onPause();

    Log.d(msg, "The onPause() event");

}

/** Called when the activity is no longer visible. */

@Override

protected void onStop() {

    super.onStop();

    Log.d(msg, "The onStop() event");

}

/** Called just before the activity is destroyed. */

@Override

public void onDestroy() {

    super.onDestroy();

}
```

```
Log.d(msg, "The onDestroy() event");

}

}
```

Một lớp **Activity** tải tất cả các thành phần UI bằng cách sử dụng tệp XML có sẵn trong thư mục *res / layout* của dự án. Câu lệnh sau tải các thành phần UI từ tệp *res/layout/activity_main.xml* :

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

Một ứng dụng có thể có **một hoặc nhiều Activity**. Mọi Activity bạn xác định cho ứng dụng của mình phải được khai báo trong tệp *AndroidManifest.xml* và **Activity chính** cho ứng dụng của bạn phải được khai báo trong manifest với `<intent-filter>` bao gồm MAIN action và LAUNCHER category như sau:

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.tutorialspoint7.myapplication">

    <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>
```

```
</application>
```

```
</manifest>
```

Nếu MAIN action hoặc LAUNCHER category không được khai báo cho một trong các Activity của bạn, thì biểu tượng ứng dụng của bạn sẽ không xuất hiện trong danh sách ứng dụng của Màn hình chính (không tìm thấy ứng dụng trên màn hình thiết bị cho dù bạn đã cài đặt nó vì thiếu MAIN action và LAUNCHER category không được khai báo cho Activity nào trong ứng dụng của bạn).

Khi bạn bắt đầu chạy ứng dụng mới này bằng AndroidStudio, trang MainActivity của ứng dụng sẽ được hiển thị lên giao diện máy ảo (hoặc máy thật nếu bạn kết nối thiết bị Android thật với máy tính để chạy ứng dụng). Khi ứng dụng chạy và hiển thị màn hình **MainActivity** thì tab **LogCat** trong AndroidStudio sẽ hiển thị log như sau:

```
08-23 10:32:07.682 4480-4480/com.example.helloworld D/Android :: The onCreate() event
```

```
08-23 10:32:07.683 4480-4480/com.example.helloworld D/Android :: The onStart() event
```

```
08-23 10:32:07.685 4480-4480/com.example.helloworld D/Android :: The onResume() event
```



Khi chúng ta bấm vào nút khóa màn hình (tắt màn hình) trên máy ảo Android và nó sẽ tạo ra các thông báo sự kiện sau trong cửa sổ **LogCat** trong AndroidStudio:

```
08-23 10:32:53.230 4480-4480/com.example.helloworld D/Android :: The onPause() event
```

```
08-23 10:32:53.294 4480-4480/com.example.helloworld D/Android :: The onStop() event
```

Khi chúng ta bấm vào nút mở khoá màn hình (bật màn hình), nó sẽ tạo ra các thông báo sự kiện sau trong cửa sổ **LogCat** trong AndroidStudio:

```
08-23 10:34:41.390 4480-4480/com.example.helloworld D/Android :: The onStart() event
```

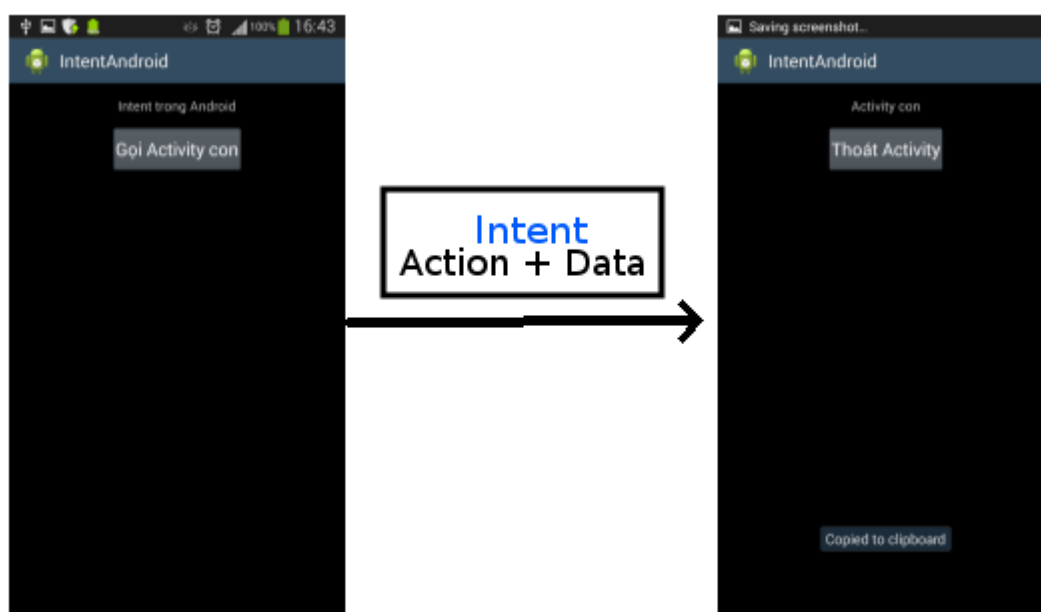
```
08-23 10:34:41.392 4480-4480/com.example.helloworld D/Android :: The onResume() event
```

Tiếp theo, chúng ta hãy thử bấm vào nút Quay lại (Back) trên máy ảo Android và nó sẽ tạo ra các thông báo sự kiện sau trong cửa sổ **LogCat** trong AndroidStudio: các callback sau được gọi lại khi ứng dụng Android được thoát ra và Activity bị hủy đi.

```
08-23 10:37:24.806 4480-4480/com.example.helloworld D/Android :: The onPause() event
08-23 10:37:25.668 4480-4480/com.example.helloworld D/Android :: The onStop() event
08-23 10:37:25.669 4480-4480/com.example.helloworld D/Android :: The onDestroy() event
```

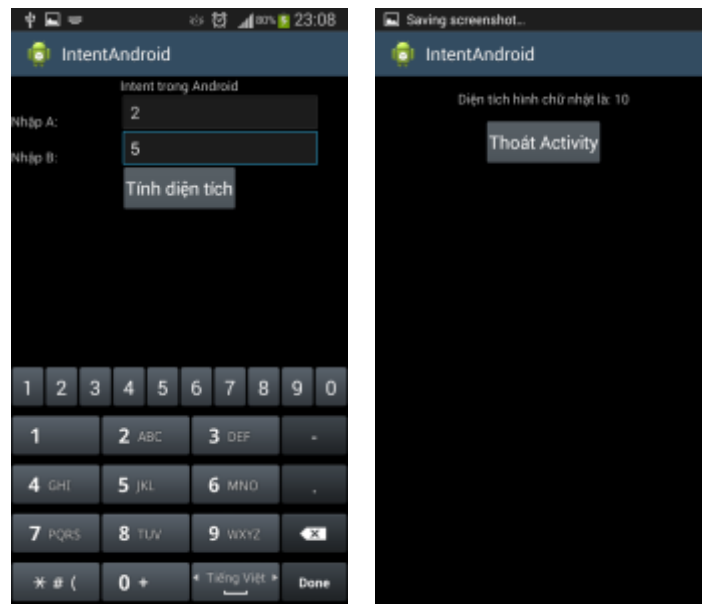
Như vậy, qua ví dụ trên, chúng ta đã chạy được một **Activity đơn giản trong ứng dụng Android** và quan sát được **cách hoạt động vòng đời của Activity** đó.

2.2. Truyền dữ liệu giữa 2 Intent



Truyền dữ liệu giữa hai Activity ta sẽ sử dụng thêm một đối tượng có tên Bundle. Đối tượng bundle dùng để đóng gói các dữ liệu lẻ thành một thùng chứa, sau đó Intent chỉ việc đem thùng chứa này qua cho Activity khác.

Thực tế bạn hoàn toàn có thể truyền dữ liệu giữa hai Activity mà không cần dùng bundle, nhưng các lập trình viên khuyên bạn nên sử dụng nó vì nó thể hiện tính rõ ràng, nhất quán. Khi bạn đã biết truyền dữ liệu sử dụng Bundle thì bạn sẽ tự biết truyền dữ liệu mà không sử dụng bundle.



Chúng ta sẽ đi xây dựng một Apps nhỏ để tính diện tích hình chữ nhật như hình ở trên, ở Activity1 sẽ nhập giá trị, khi ấn nút xem kết quả nó sẽ gọi Activity2 đồng thời truyền dữ liệu sang cho Activity2 để nó xử lý và hiện kết quả.

Trước hết là giao diện cho Activity đầu tiên với file activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="${packageName}.${activityClass}" >
```

<EditText

```
    android:id="@+id/etB"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/etA"
    android:layout_below="@+id/etA"
    android:ems="10"
    android:inputType="numberSigned" >
```

```
    <requestFocus />
```

</EditText>

<TextView

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
```

```
android:text="Intent trong Android" />
```

<Button

```
android:id="@+id/button"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@+id/etB"  
android:layout_centerHorizontal="true"  
android:text="Tính diện tích" />
```

<EditText

```
android:id="@+id/etA"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/button"  
android:layout_below="@+id/textView1"  
android:ems="10"  
android:inputType="numberSigned" />
```

<TextView

```
android:id="@+id/TextView02"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_above="@+id/button"  
android:layout_alignParentLeft="true"  
android:text="Nhập B:" />
```

<TextView

```
android:id="@+id/TextView01"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBottom="@+id/etA"  
android:layout_alignParentLeft="true"  
android:text="Nhập A:" />
```

</RelativeLayout>

Tại file MainActivity.java ta sẽ xử lý như sau:

```
package clbtinhoc.ictu.intentandroid;
```

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

```
import android.content.Intent;
```

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

```
import android.view.Menu;
```



```
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```
        final EditText etA = (EditText) findViewById(R.id.etA);  
        final EditText etB = (EditText) findViewById(R.id.etB);
```

```
        final Button button = (Button) findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

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

```
                // nạp Class ChildActivity vào intent.
```

```
                // Đối thứ nhất là context của Activity hiện tại, đối thứ hai là tên Class chứa  
                // Activity sẽ được gọi
```

```
                Intent intent = new Intent(MainActivity.this, ChildActivity.class);
```

```
                int a = Integer.valueOf(etA.getText().toString());
```

```
                int b = Integer.valueOf(etB.getText().toString());
```

```
                // Truyền dữ liệu giữa hai Activity ta sử dụng đối tượng Bundle để chứa các  
                // dữ liệu.
```

```
                // Các bạn hiểu Intent là người đưa thư còn Bundle là thùng chứa các lá thư
```

```
                // đưa các dữ liệu vào "thùng chứa" Bundle
```

```
                Bundle bundle = new Bundle();
```

```
                bundle.putInt("CanhDai", a); // CanhDai là tên giao dịch
```

```
                bundle.putInt("CanhRong", b);
```

```
                // Đưa thùng chứa Bundle cho người đưa thư Intent, trong đó GoiTin là tên  
                // của thùng chứa
```

```
                intent.putExtra("GoiTin", bundle);
```

```
                // gọi activity.
```

```
                startActivity(intent);
```

```
            }
```

```
        });
```

```
    }  
}
```

Học kết hợp

```
}  
}
```

Ở đoạn code trên bạn thấy đoạn code `bundle.putInt("CanhDai", a)`; thì "CanhDai" ở đây chính là action mà mình đã nói ở bài trước, còn `a` chính là data. Tương tự vậy lệnh `intent.putExtra("GoiTin", bundle)`; thì "GoiTin" là action và `bundle` là data. action ở đây chỉ là một chuỗi ký tự định danh cho các gói dữ liệu giúp cho việc xử lý các dữ liệu không bị nhầm lẫn.

Khi ấn vào Button thì `ChildActivity.java` sẽ được gọi

```
package clbtinhoc.ictu.intentandroid;
```

```
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;
```

```
public class ChildActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // Để ý dòng code này các bạn thấy giao diện đặt theo file child_activity.xml  
        setContentView(R.layout.child_activity);
```

```
        final TextView textView = (TextView) findViewById(R.id.textView);
```

```
        // Lấy intent của Activity này  
        Intent intent = getIntent();  
        // lấy thùng chứa Bundle với tên giao dịch là "GoiTin"  
        Bundle bundle = intent.getBundleExtra("GoiTin");
```

```
        // lấy giá trị kiểu nguyên với tên giao dịch là CanhDai  
        int a = bundle.getInt("CanhDai");  
        // lấy giá trị kiểu nguyên với tên giao dịch là CanhRong  
        int b = bundle.getInt("CanhRong");
```

```
        textView.setText("Diện tích hình chữ nhật là: " + a*b);
```



```
final Button button = (Button) findViewById(R.id.buttonExit);  
button.setOnClickListener(new View.OnClickListener() {
```

```
    @Override  
    public void onClick(View v) {  
        // Thoát khỏi activity hiện tại  
        finish();  
    }  
});  
  
}  
}
```

Còn đây là giao diện cho file ChildActivity.java

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context="$${packageName}.$${activityClass}" >
```

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="14dp" />
```

```
<Button  
    android:id="@+id/buttonExit"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/textView1"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="46dp"  
    android:text="Thoát Activity" />
```

```
</RelativeLayout>
```