

## Hướng dẫn lập trình cơ bản với Android - Bài 4

Reflink: <http://diendan.vietandroid.com/cac-bai-hoc-co-kem-ma-nguon/831-huong-dan-lap-trinh-co-ban-voi-android-bai-4-a.html>

### List tutorial

Bài 0 - Cài đặt và sử dụng Android với Eclipse  
Bài 1 - Cơ bản Android  
Bài 2 - Xây dựng giao diện đơn giản  
Bài 3 - ViewGroup và Custom Adapter  
Bài 4 - Intent và Broadcast Receiver  
Bài 5 - Service  
Bài 6 - SQLite  
Bài 7 - Content Provider

Sorry mọi người vì quá lâu mới ra được bài tiếp theo. Một phần vì công việc bận bịu, một phần vì lười + ngại ^\_^ mà đã trễ hẹn với anh em. Mình đoán những newbie đọc bài của mình từ 1, 2, 3 giờ đã sắp thành pro hết rồi, vì vậy đành dành bài này cho các newbie mới. Trong bài này mình sẽ đi sâu nói rõ về Intent, phần cơ bản và đóng vai trò rất quan trọng trong lập trình ứng dụng Android.

### Khái niệm về Intent:

Theo định nghĩa của Google, Intent là một miêu tả về một hoạt động cần được thực hiện. Còn nói một cách đơn giản và dễ hiểu hơn, Intent là một cơ cấu cho phép truyền thông điệp giữa các thành phần của 1 ứng dụng và giữa các ứng dụng với nhau.

### Các thuộc tính của Intent:

- **action**: là hành động được thực hiện, vd : ACTION\_VIEW, ACTION\_MAIN
- **data**: là dữ liệu sẽ được xử lý trong action, thường được diễn tả là một Uri (Uniform Resource Identifier, tham khảo [http://en.wikipedia.org/wiki/Uniform...rce\\_Identifier](http://en.wikipedia.org/wiki/Uniform_Resource_Identifier) để hiểu rõ thêm chi tiết).

VD:

**ACTION\_VIEW content://contacts/people/1** - Hiển thị thông tin về người với mã danh 1

**ACTION\_DIAL content://contacts/people/1** - Hiển thị màn hình gọi đến người với mã danh 1

**ACTION\_DIAL tel:123** - Hiển thị màn hình gọi với số gọi là 123

Ngoài ra còn có 1 số thuộc tính mà ta có thể bổ sung vào Intent:

- **category**: bổ sung thêm thông tin cho action của Intent. VD: CATEGORY\_LAUNCHER thông báo sẽ thêm vào Launcher như là một ứng dụng top-level
- **type**: chỉ rõ kiểu của data
- **component**: chỉ rõ thành phần sẽ nhận và xử lý intent. Khi thuộc tính này được xác định thì các thuộc tính khác sẽ trở thành thuộc tính phụ.
- **extras**: mang theo đối tượng Bundle chứa các giá trị bổ sung.

VD:

**ACTION\_MAIN** và **CATEGORY\_HOME**: trở về màn hình Home của Android (khi bấm nút Home của di động)

### Phân loại Intent:

Intent được chia làm 2 loại:

- **Explicit Intents:** intent đã được xác định thuộc tính component, nghĩa là đã chỉ rõ thành phần sẽ nhận và xử lý intent. Thông thường intent dạng này sẽ không bổ sung thêm các thuộc tính khác như action, data. Explicit Intent thường được sử dụng để khởi chạy các activity trong cùng 1 ứng dụng.

- **Implicit Intents:** Intent không chỉ rõ component xử lý, thay vào đó nó bổ sung thông tin trong các thuộc tính. Khi intent được gửi đi, hệ thống sẽ dựa vào những thông tin này để quyết định component nào thích hợp nhất để xử lý nó.

VD:

**ACTION\_DIAL tel:123** thông thường sẽ được hệ thống giao cho activity Phone Dialer mặc định của Android xử lý.

### Một số action thường sử dụng trong Intent:

**ACTION\_ANSWER** - mở Activity để xử lý cuộc gọi tới, thường là Phone Dialer của Android

**ACTION\_CALL** - mở 1 Phone Dialer (mặc định là PD của Android) và ngay lập tức thực hiện cuộc gọi dựa vào thông tin trong data URI

**ACTION\_DELETE** - mở Activity cho phép xóa dữ liệu mà địa chỉ của nó chứa trong data URI

**ACTION\_DIAL** - mở 1 Phone Dialer (mặc định là PD của Android) và điền thông tin lấy từ địa chỉ chứa trong data URI

**ACTION\_EDIT** - mở 1 Activity cho phép chỉnh sửa dữ liệu mà địa chỉ lấy từ data URI

**ACTION\_SEND** - mở 1 Activity cho phép gửi dữ liệu lấy từ data URI, kiểu của dữ liệu xác định trong thuộc tính type

**ACTION\_SENDTO** - mở 1 Activity cho phép gửi thông điệp tới địa chỉ lấy từ data URI

**ACTION\_VIEW** - action thông dụng nhất, khởi chạy activity thích hợp để hiển thị dữ liệu trong data URI

**ACTION\_MAIN** - sử dụng để khởi chạy 1 Activity

OK, lý thuyết như thế là đã tạm ổn. Giờ chúng ta sẽ chuyển qua phần thực hành để hiểu rõ cách sử dụng Intent. Như đã nêu ở trên, Intent chia làm 2 loại: explicit intent và implicit intent. Mỗi loại Intent sẽ có cách cài đặt và sử dụng khác nhau.

### Using Explicit Intents

**Yêu cầu:** Xây dựng chương trình gồm 2 Activity. Activity1 là Activity chạy ban đầu lúc khởi động ứng dụng, cho phép nhập vào 1 giá trị, cho phép khởi chạy Activity2 và gửi giá trị này tới Activity2. Activity2 sẽ nhận và hiển thị giá trị, rồi lại gửi giá trị này tới 1 BroadcastReceiver. Cơ chế gửi và khởi chạy Activity sử dụng thông qua Intent.

**B1:** Khởi tạo project: File -> New -> Android Project

Project name: Explicit Intent Example

Build Target: Chọn Android 1.5

Application name: Explicit Intent Example

Package name: at.exam

Create Activity: Activity1

=> Kích nút Finish.

**B2:** Tạo giao diện cho Activity1 -> res\layout\main.xml chuyển tên thành activity1\_layout.xml

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
```

```

        android:layout_height="fill_parent"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Activity 1 - Send value"
            android:typeface="normal"
            android:textSize="14px"
            android:textStyle="bold"
            android:textColor="#cccccc"
            android:background="#333333"

        />
        <EditText
            android:id="@+id/value_edit"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize="20px"
            android:gravity="center"
            android:lines="1"
            android:numeric="integer"
        />
        <RelativeLayout
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <Button
                android:id="@+id/send_button"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Send to Activity 2"
                android:layout_alignParentBottom="true"

            />
        </RelativeLayout>
    </LinearLayout>

```

Layout cho Activity1 bao gồm 1 LinearLayout chứa 1 TextView, 1 EditText để nhập giá trị (đã giới hạn kiểu nhập là number), và 1 RelativeLayout có 1 Button để khởi chạy Activity2. Mình sử dụng RelativeLayout để có thể xếp Button này xuống phía cuối của giao diện.

**B3:** Tạo giao diện cho Activity2 -> Chuột phải vào folder res\layout -> New -> Android XML File -> Gõ tên là activity2\_layout.xml  
Mã:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Activity 2 - Receive value"
            android:typeface="normal"
            android:textSize="14px"
            android:textStyle="bold"
            android:textColor="#cccccc"

```

```

        android:background="#333333"
    />
    <EditText
        android:id="@+id/value_receive"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:gravity="center"
        android:lines="1"
        android:numeric="integer"
        android:enabled="false"
    />
    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <Button
            android:id="@+id/call_button"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Call Broadcast Receiver"
            android:layout_alignParentBottom="true"

        />
    </RelativeLayout>
</LinearLayout>

```

Layout của Activity2 tương tự như Activity1, nhưng Button bây giờ là để gọi BroadCast Receiver. Ngoài ra mình dùng EditText để hiển thị value nhận được (do nó có cái đường bao ngoài đẹp hơn TextView ^^) nên không cho phép nhập giá trị vào EditText này

Mã:

```

android:enabled="false"

```

#### **B4:**Sửa lại nội dung của Activity1.java như sau:

Mã:

```

package at.exam;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class Activity1 extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity1_layout);

        final EditText editValue = (EditText)
        findViewById(R.id.value_edit);
    }
}

```

```

        final Button sendButton = (Button)
findViewById(R.id.send_button);

        sendButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                String valueString =
editValue.getText().toString();
                long value;
                if (valueString != null) {
                    value =
Long.parseLong(valueString);
                }
                else {
                    value = 0;
                }

                //Tạo 1 đối tượng Bundle để gửi đi cùng
Intent

                Bundle sendBundle = new Bundle();
                sendBundle.putLong("value", value);

                //Tạo Intent để khởi chạy Activity2 và
gắn sendBundble vào Intent

                Intent i = new Intent(Activity1.this,
Activity2.class);

                i.putExtras(sendBundle);
                startActivity(i);

                //Giải phóng Activity1 khỏi Activity
Stack vì ta sẽ ko quay lại nó nữa
                finish();
            }
        });
    }
}

```

**B5:** Tạo mới 1 Class Activity2.java trong package at.exam -> chỉnh sửa nội dung:  
Mã:

```

package at.exam;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class Activity2 extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity2_layout);
    }
}

```

```

        final EditText receiveValueEdit = (EditText)
findViewById(R.id.value_receive);
        final Button callReceiverButton = (Button)
findViewById(R.id.call_button);

        //Lấy về Bundle được gửi kèm Intent rồi lấy ra giá trị
        Bundle receiveBundle = this getIntent().getExtras();
        final long receiveValue = receiveBundle.getLong("value");

        receiveValueEdit.setText(String.valueOf(receiveValue));

        callReceiverButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                //Khởi tạo 1 Intent để gửi tới Broadcast
                //Gắn giá trị vào Intent, lần này ko cần
                Intent i = new Intent(Activity2.this,
                Receiver.class);
                i.putExtra("new value", receiveValue -
                10);
                sendBroadcast(i);
            }
        });
    }
}

```

**B6:** Tạo Broadcast Receiver để nhận Intent mà Activity2 gửi tới -> Tạo 1 file Receiver.java trong at.exam -> Nội dung:  
Mã:

```

package at.exam;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class Receiver extends BroadcastReceiver{

    @Override
    public void onReceive(Context context, Intent intent) {
        long value = intent.getLongExtra("new value", -10) + 10;
        Toast toast = Toast.makeText(context, "Broadcast
Receiver catch an Intent"
                                + " \n" + "The value is stored in the
Intent is "
                                + String.valueOf(value),
        Toast.LENGTH_LONG);
        toast.show();
    }
}

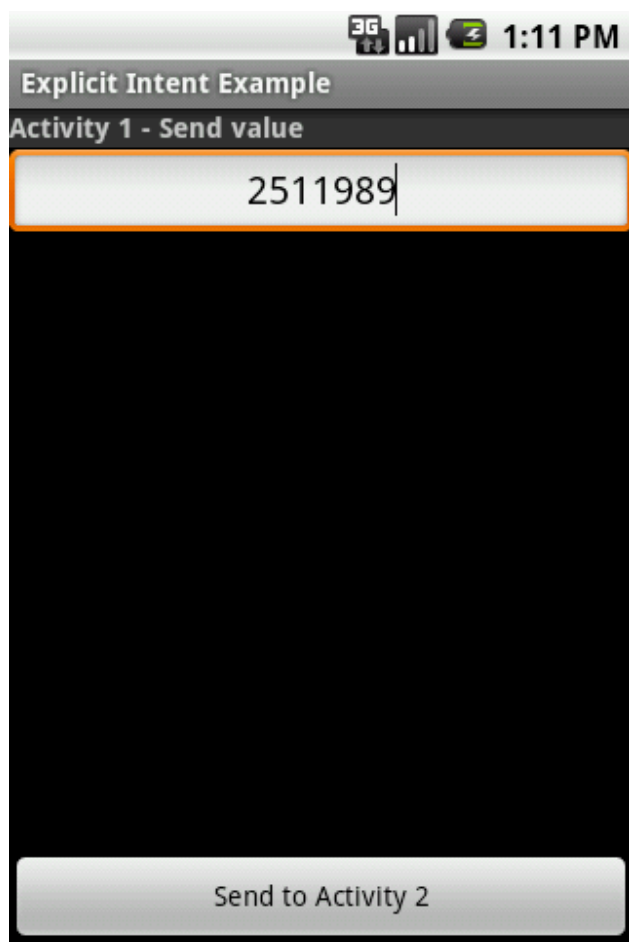
```

Code không hề khó hiểu, và mình cũng đã add comment. Chỉ cần lưu ý ở đây là Toast là lớp để hiển thị một thông báo đơn giản trong 1 khoảng thời gian cố định, và ko thể thay đổi thời gian này T\_T (why???) chỉ có thể chọn giữa LENGTH\_SHORT với LENGTH\_LONG

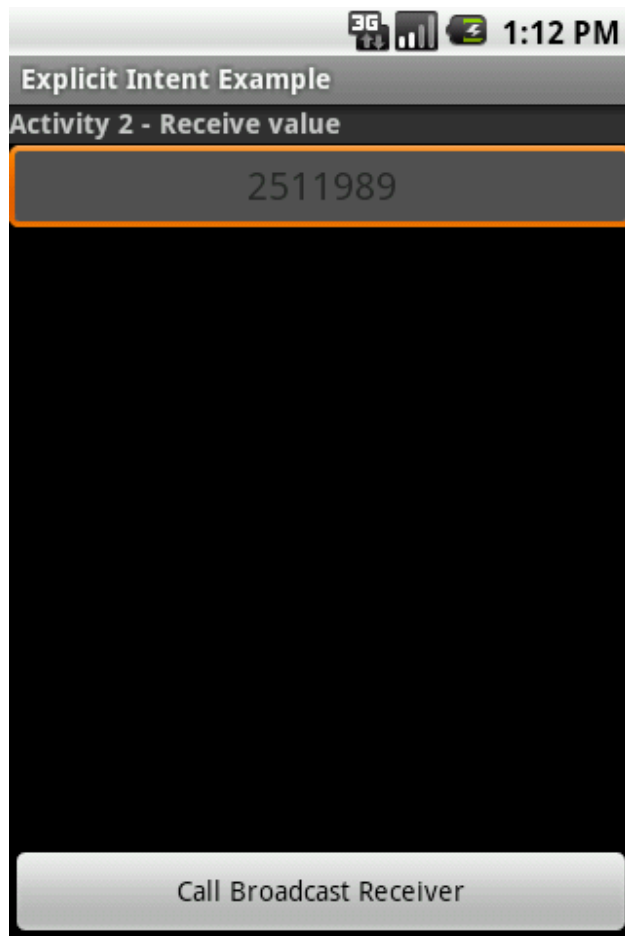
**B7:** Bổ sung thêm thông tin về component mới vào AndroidManifest.xml:

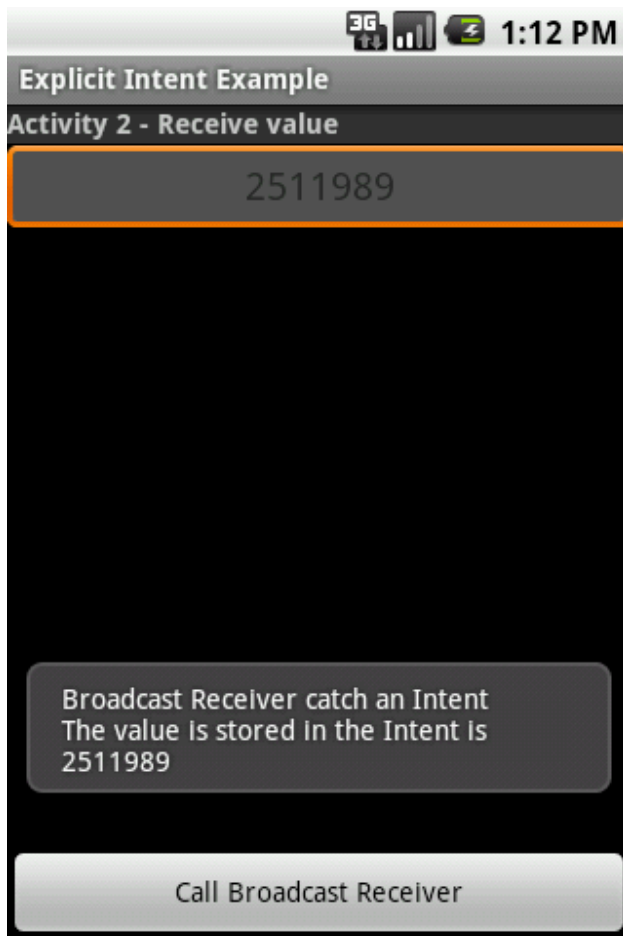
Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="at.exam"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".Activity1"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Activity2"></activity>
        <receiver android:name=".Receiver"></receiver>
    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```









Đã xong sử dụng Explicit, giờ đến lượt Implicit Intent. Trước khi đi vào ví dụ, hãy dạo qua 1 chút kiến thức về Intent Filter và vai trò của nó.

## Intent Filter là gì

Activity, Service và Broadcast Receiver sử dụng Intent Filter để thông báo cho hệ thống biết các dạng Implicit Intent mà nó có thể xử lý. Nói cách khác, Intent Filter là bộ lọc Intent, chỉ cho những Intent được phép đi qua nó.

Intent Filter mô tả khả năng của component định nghĩa nó. Khi hệ thống bắt được 1 Implicit Intent (chỉ chứa 1 số thông tin chung chung về action, data và category...), nó sẽ sử dụng những thông tin trong Intent này, kiểm tra đối chiếu với Intent Filter của các component các ứng dụng, sau đó quyết định khởi chạy ứng dụng nào thích hợp nhất để xử lý Intent bắt được. Nếu có 2 hay nhiều hơn ứng dụng thích hợp, người dùng sẽ được lựa chọn ứng dụng mình muốn.

VD:

Mã:

```
<activity android:name=".ExampleActivity"
android:label="@string/activity_name">
    <intent-filter>
        <action android:name="android.intent.action.SENDTO" />
        <category
android:name="android.intent.category.DEFAULT" />
```

```
        <data android:scheme="sms" />
    </intent-filter>
</activity>
```

Trên là 1 Activity với bộ lọc Intent cho phép bắt và xử lý các Intent gửi SMS. Hãy lưu ý từ khóa Mã:

android:scheme

Từ khóa này cho biết protocol (luật) để xử lý dữ liệu trong URI. Nói 1 cách đơn giản thì nó là kiểu của dữ liệu. 1 số kiểu khác như http, https, ftp, content...

## Using Implicit Intent:

**Yêu cầu:** Xây dựng chương trình nhập số và gọi. Lưu ý chương trình của mình ở đây chỉ xây dựng đến mức khi nhấn nút Call của di động thì sẽ chạy ứng dụng và hiển thị giao diện cho phép nhập số. Phần gọi dành cho ai yêu thích tìm hiểu thêm ^\_^ Phần này không hề khó nhưng ở đây mình chỉ muốn minh họa Implicit Intent nên sẽ không đưa vào.

**B1:** Khởi tạo project: File -> New -> Android Project

Project name: Implicit Intent Example

Build Target: Chọn Android 1.5

Application name: Implicit Intent Example

Package name: at.exam

Create Activity: Example

=> Kích nút Finish.

**B2:** Đây là bước quan trọng nhất và cũng là bước có ý nghĩa duy nhất trong cả project này, các bước còn lại chỉ là bước râu ria mà mình thêm vào cho cái project nó ra hồn 1 chút. Bước này sẽ thêm 1 bộ lọc Intent Filter vào cho activity Example của chúng ta để bắt sự kiện nhấn nút Call của di động

-> Vào AndroidManifest.xml chỉnh sửa như sau:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="at.exam"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".Example"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action
                    android:name="android.intent.action.CALL_BUTTON" />
                <category
                    android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
```

```
<uses-sdk android:minSdkVersion="3" />

</manifest>
```

Thực chất chỉ là bổ sung thêm dòng chữ để mình đánh dấu thôi ^\_^

**B3:** Xây dựng giao diện trong main.xml, bước này ko quan trọng, chỉ là râu ria cho activity có cái giao diện:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    >
    <TextView
        android:paddingTop="10px"
        android:id="@+id/number_display"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="30px"
        android:gravity="center"
        android:lines="2"
        android:background="#ffffff"
        android:textColor="#000000"
        />
    <TableLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        >
        <TableRow
            android:gravity="center"
            android:paddingTop="30px"
            >
            <Button
                android:id="@+id/button1"
                android:layout_width="80px"
                android:layout_height="80px"
                android:gravity="center"
                android:text="1"
                android:textSize="25px"

            />
            <Button
                android:id="@+id/button2"
                android:layout_width="80px"
                android:layout_height="80px"
                android:gravity="center"
                android:text="2"
                android:textSize="25px"

            />
            <Button
                android:id="@+id/button3"
                android:layout_width="80px"
```

```

        android:layout_height="80px"
        android:gravity="center"
        android:text="3"
        android:textSize="25px"
    />
</TableRow>
<TableRow
    android:gravity="center"
>
    <Button
        android:id="@+id/button4"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="4"
        android:textSize="25px"

    />
    <Button
        android:id="@+id/button5"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="5"
        android:textSize="25px"

    />
    <Button
        android:id="@+id/button6"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="6"
        android:textSize="25px"

    />
</TableRow>
<TableRow
    android:gravity="center"
>
    <Button
        android:id="@+id/button7"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="7"
        android:textSize="25px"

    />
    <Button
        android:id="@+id/button8"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="8"
        android:textSize="25px"

    />
    <Button

```

```

        android:id="@+id/button9"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="9"
        android:textSize="25px"
    />
</TableRow>
<TableRow
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
>
    <Button
        android:id="@+id/button_star"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="*"
        android:textSize="25px"
    />
    <Button
        android:id="@+id/button0"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="0"
        android:textSize="25px"
    />
    <Button
        android:id="@+id/button_clear"
        android:layout_width="80px"
        android:layout_height="80px"
        android:gravity="center"
        android:text="Clear"
        android:textSize="25px"
    />
</TableRow>
</TableLayout>
</LinearLayout>

```

LinearLayout chứa 1 TextView để hiển thị số nhĩn, 1 TableLayout có các Button tương ứng với các số và 1 Button để clear cho TextView.

**B4:** Code code code... So tired... Tutorial is really take time. Chĩnh Example.java:

Mã:

```

package at.exam;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

```

```

public class Example extends Activity {
    Button button1, button2, button3;
    Button button4, button5, button6;
    Button button7, button8, button9;
    Button button0, buttonStar, buttonClear;

    TextView numberView;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        numberView = (TextView) findViewById(R.id.number_display);

        button1 = (Button) findViewById(R.id.button1);
        button2 = (Button) findViewById(R.id.button2);
        button3 = (Button) findViewById(R.id.button3);

        button4 = (Button) findViewById(R.id.button4);
        button5 = (Button) findViewById(R.id.button5);
        button6 = (Button) findViewById(R.id.button6);

        button7 = (Button) findViewById(R.id.button7);
        button8 = (Button) findViewById(R.id.button8);
        button9 = (Button) findViewById(R.id.button9);

        button0 = (Button) findViewById(R.id.button0);
        buttonStar = (Button) findViewById(R.id.button_star);
        buttonClear = (Button) findViewById(R.id.button_clear);

        button1.setOnClickListener(this.appendString("1"));
        button2.setOnClickListener(this.appendString("2"));
        button3.setOnClickListener(this.appendString("3"));
        button4.setOnClickListener(this.appendString("4"));
        button5.setOnClickListener(this.appendString("5"));
        button6.setOnClickListener(this.appendString("6"));
        button7.setOnClickListener(this.appendString("7"));
        button8.setOnClickListener(this.appendString("8"));
        button9.setOnClickListener(this.appendString("9"));
        button0.setOnClickListener(this.appendString("0"));
        buttonStar.setOnClickListener(this.appendString("*"));

        buttonClear = (Button) findViewById(R.id.button_clear);
        buttonClear.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                numberView.setText("");
            }
        });
    }

    public OnClickListener appendString(final String number) {
        return new OnClickListener() {
            public void onClick(View arg0) {

```

```

                                numberView.append(number);
                                }
                                };
        }

        public boolean onCreateOptionsMenu(Menu menu) {
            super.onCreateOptionsMenu(menu);
            menu.add(0, Menu.FIRST, 0, "Exit"
).setIcon(android.R.drawable.ic_delete);
            return true;
        }

        public boolean onOptionsItemSelected(MenuItem item) {
            switch (item.getItemId()) {
                case Menu.FIRST: {
                    finish();
                    break;
                }
            }
            return false;
        }
    }
}

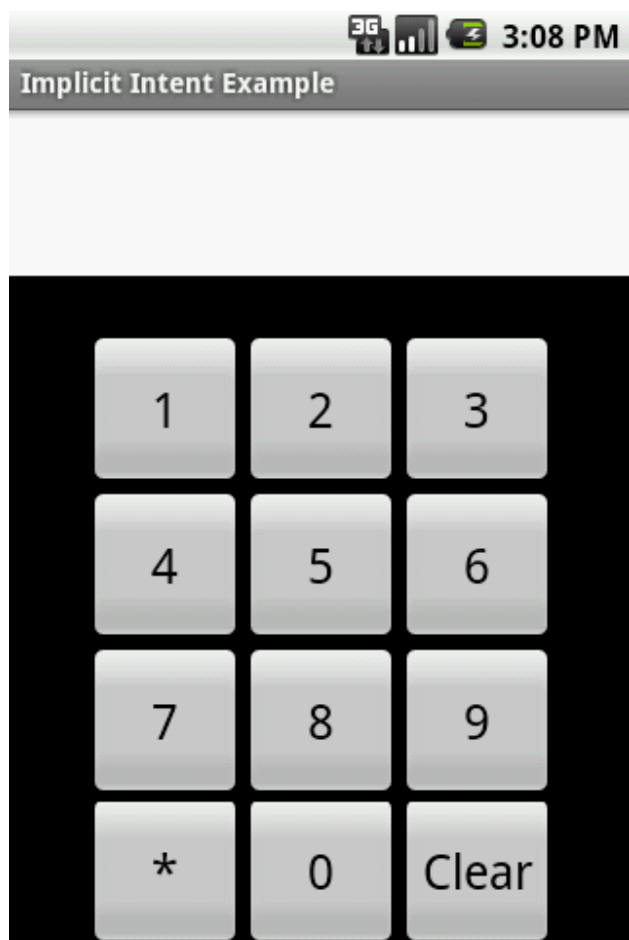
```

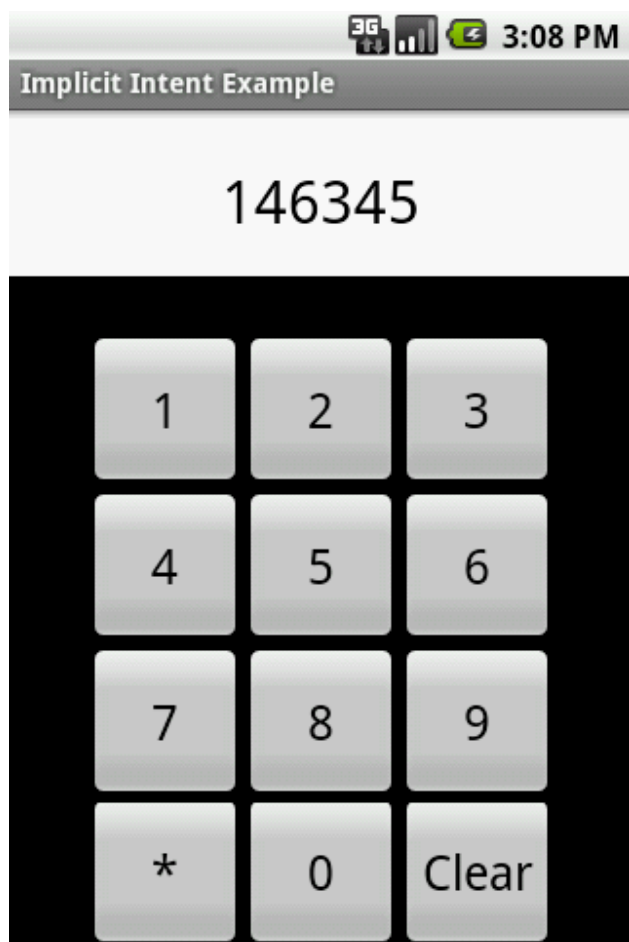
Code quá đơn giản, mình còn ko thèm comment nữa. Lưu ý có 1 Option Menu để đóng Activity và cũng là đóng luôn ứng dụng.

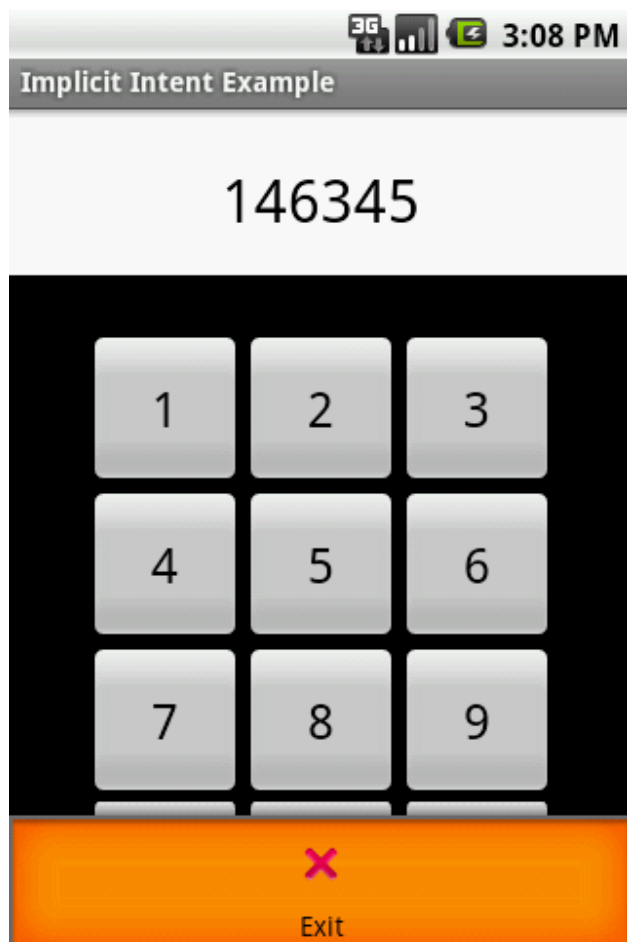
**B5:** Time to test... Khởi chạy project, rồi sử dụng Option Menu của mình (bấm nút Menu của Emulator hoặc di động Android) để thoát khỏi chương trình. Ok, sau khi chọn Exit ta có thể chắc chắn là ứng dụng đã được đóng hoàn toàn, activity ko còn tồn tại trong stack của Emulator/di động nữa. Giờ nhấn nút Call của Emulator/di động, Tadaaaaaaaaaa











#### **Kết thúc bài 4**