

Bài 05:

TÀI NGUYÊN ỨNG DỤNG TRONG ANDROID

GVGD: ThS. Đặng Thế Hân

Biên soạn: ThS. Giang Hào Côn

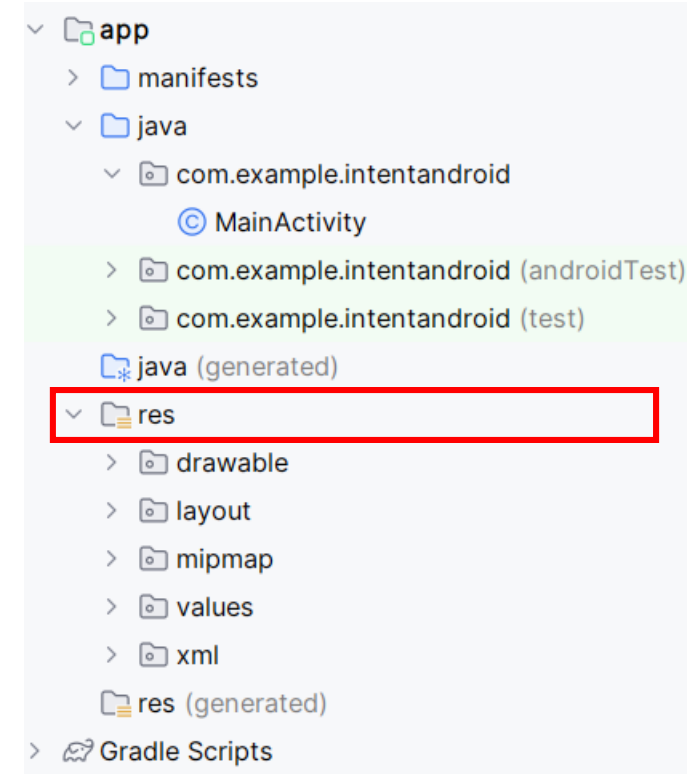
Mục tiêu

Cung cấp cho sinh viên kiến thức cơ bản về các loại tài nguyên trong ứng dụng Android và cách sử dụng Intent trong ngôn ngữ lập trình ứng dụng trên Android.

- 1) Các tài nguyên (resource) trong Android
 - a) Drawable
 - b) Values
 - c) Dimens
 - d) String
- 2) Intent trong Android
 - Explicit Intent (Intent tường minh)
 - Implicit Intent (Intent không tường minh)

2.1/ Resource trong ứng dụng Android

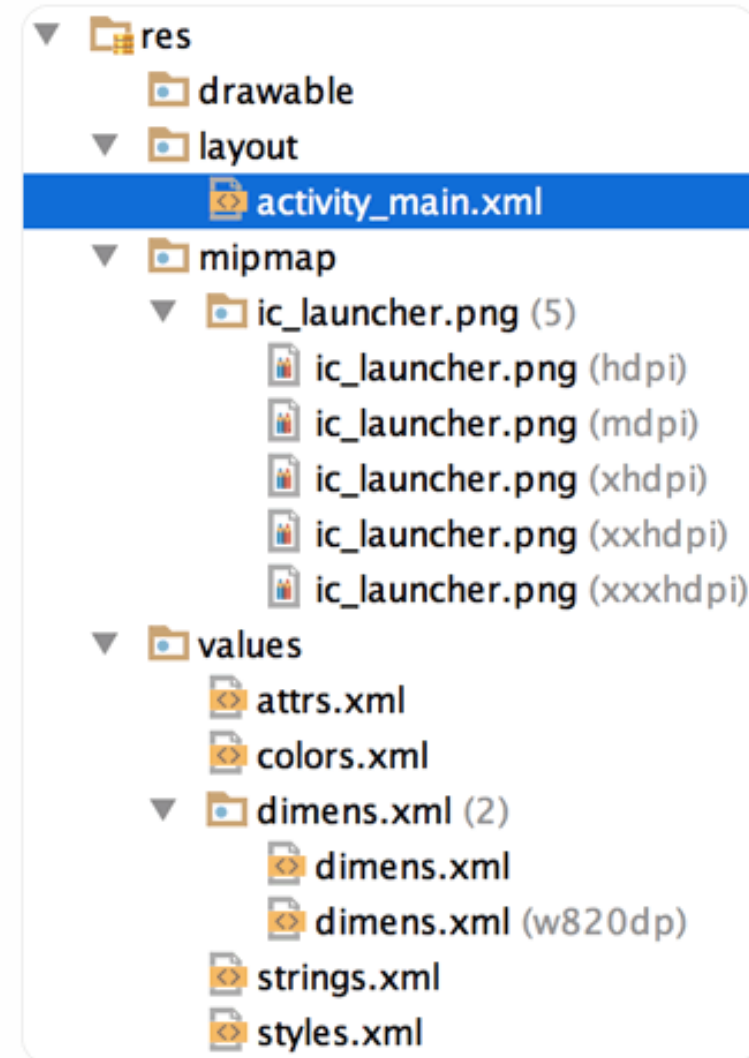
- **Resource** (nguồn) được hiểu là **hình ảnh**, **âm thanh**, **văn bản** được đặt trong các thư mục của project và sử dụng trong ứng dụng như **icon**, **tên ứng dụng**, ...
- Khi tạo project Android sẽ có 1 thư mục **res** (res là viết tắt của resource) được tạo ra. Trong thư mục này chứa nhiều thư mục khác nữa, dùng để chứa các loại resource khác nhau như **drawable**, **layout**, **xml**, **values**, **mipmap**, ...



2.1/ Resource trong ứng dụng Android

Cấu trúc thư mục res trong Android

- drawable ?
- layout ?
- mipmap ?
- values ?
- menu ?
- raw ?
- xml ?



2.1/ Resource trong ứng dụng Android

Sử dụng resource trong Android

Hầu hết trong tập tin layout, đều có thể truy cập resource thông qua:

```
@resource_type/value
```

- **resource_type**: thường có các giá trị là **drawable, color, dimen, string, style, bool, integer**
- **values** có các giá trị:
 - **id**: của file (thường sử dụng trong trường hợp drawable).
 - **name**: định nghĩa trong tập tin đó, thường được dùng trong các file nằm trong thư mục values.

2.1/ Resource trong ứng dụng Android

Ví dụ 01: tạo file **dimens.xml**

Dùng để khai báo các
chỉ số **khoảng cách lề**
(margin), **khoảng cách**
đệm (padding), **kích**
thước font chữ, ...

```
<?xml version="1.0" encoding="utf-8"?>|
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <!--define value dimen for tvContent-->
    <dimen name="text_view_content_width">40dp</dimen>
    <dimen name="text_view_content_height">40dp</dimen>
    <dimen name="text_view_content_margin_left">3dp</dimen>
    <dimen name="text_view_content_margin_right">3dp</dimen>
    <dimen name="text_view_content_margin_top">8dp</dimen>
    <dimen name="text_view_content_margin_bottom">8dp</dimen>
    <dimen name="text_view_content_padding_left">10dp</dimen>
    <dimen name="text_view_content_padding_right">10dp</dimen>
    <dimen name="text_view_content_padding_top">6dp</dimen>
    <dimen name="text_view_content_padding_bottom">6dp</dimen>
    <!--define text size for tvContent-->
    <dimen name="text_content_text_size">16sp</dimen>
</resources>
```

2.1/ Resource trong ứng dụng Android

Ví dụ 02: tạo file **strings.xml**

Dùng để khai báo các tiêu đề ứng dụng, nội dung văn bản hiển thị trong ứng dụng ...

```
<resources>
    <string name="app_name">IntentAndroid</string>
    <string name="text_content">gianghaocon.com</string>
</resources>
```


2.1/ Resource trong ứng dụng Android

Ví dụ 03: tạo file **colors.xml**

Dùng để khai báo sẵn các tên màu sử dụng trong ứng dụng ...

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="red">#ff0000</color>
</resources>
```

2.1/ Resource trong ứng dụng Android

Ví dụ 04: tạo file **styles.xml**

Dùng để khai báo sẵn các **style** sử dụng trong ứng dụng ...

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
    <style name="TextViewContentStyle">
        <item name="android:textStyle">bold</item>
    </style>
</resources>
```

2.1/ Resource trong ứng dụng Android

Ví dụ 05: tạo file **bools.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <bool name="Checked">true</bool>
    <bool name="NotChecked">false</bool>
</resources>
```

Sử dụng cho những View nào có thuộc tính chấp nhận là true hay false, trong trường hợp này sử dụng CheckBox

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="@bool/Checked"/>
```

2.1/ Resource trong ứng dụng Android

Ví dụ 05: tạo file **integers.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <integer name="minimum">0</integer>
    <integer name="maximum">100</integer>
</resources>
```

Và được sử dụng cho những thuộc tính của View chấp nhận giá trị kiểu int, trường hợp này sử dụng SeekBar.

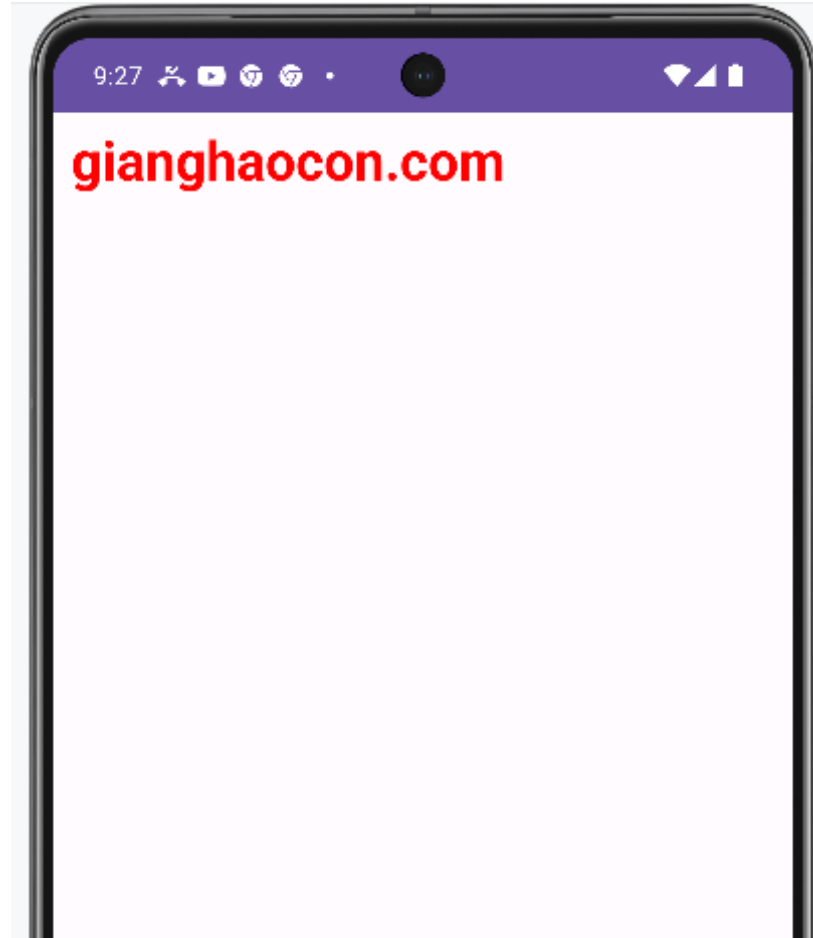
```
<SeekBar
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="@integer/maximum"
    android:progress="@integer/minimum" />
```

2.1/ Resource trong ứng dụng Android

Sử dụng các tài nguyên đã định nghĩa ở trên

<TextView

```
style="@style/TextViewContentStyle"  
android:layout_width="@dimen/text_view_content_width"  
android:layout_height="@dimen/text_view_content_height"  
android:layout_marginBottom="@dimen/text_view_content_margin_bottom"  
android:layout_marginLeft="@dimen/text_view_content_margin_left"  
android:layout_marginRight="@dimen/text_view_content_margin_right"  
android:layout_marginTop="@dimen/text_view_content_margin_top"  
android:paddingBottom="@dimen/text_view_content_padding_bottom"  
android:paddingLeft="@dimen/text_view_content_padding_left"  
android:paddingRight="@dimen/text_view_content_padding_right"  
android:paddingTop="@dimen/text_view_content_padding_top"  
android:text="@string/text_content"  
android:textColor="@color/red"  
android:textSize="@dimen/text_content_text_size"  
tools:ignore="MissingConstraints" />
```



2.1/ Resource trong ứng dụng Android

Sử dụng resource đã định nghĩa ở trên trong code Java

Trong tập tin **MainAActivity.java**, để truy cập vào resource phải thông qua: **R.resource_type.value**, với **resource_type** và **value** đã giải thích ở slide trên và sử dụng phương thức:

```
getResources().getResourceType(R.resource_type.value);
```

Với **getResourceType**
là các phương thức:

- getColor()
- getDrawable()
- getInteger()
- getBoolean()
- getDimension()
- getString()
- ...

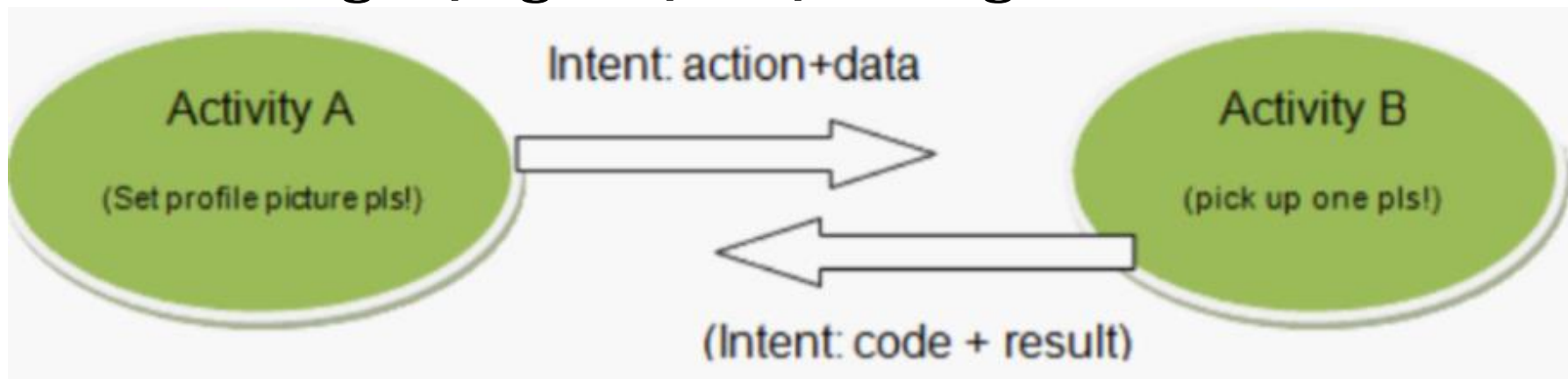
2.1/ Resource trong ứng dụng Android

Ví dụ lấy 1 số resource đã sử dụng trong bài viết

```
// Get color resource
int color = getResources().getColor(R.color.red);
// Get integer resource
int max = getResources().getInteger(R.integer.max);
// Get boolean resource
boolean checked = getResources().getBoolean(R.bool.Checked);
// Get dimen resource
int margin = (int)getResources().getDimension(R.dimen.activity_horizontal_margin);
// Get drawable resource
Drawable drawable = getResources().getDrawable(R.drawable.icon_what_app);
// Get String resource
String str = getResources().getString(R.string.app_name);
```


2.2/ Intent là gì ?

- **Intent** là một cấu trúc thông điệp mô tả hành động sẽ được thực thi. Nó đại diện cho một hành động đi kèm với dữ liệu theo một ngữ cảnh xác định.
- **Intent** là chuẩn giao tiếp giữa các thành phần trong ứng dụng Android, được sử dụng để chạy Activity, Service hoặc một thành phần khác của ứng dụng hoặc hệ thống.



2.2.1/ Sử dụng Intent

- Về cơ bản, **Intent** là đối tượng của lớp: `android.content.Intent`.
- Thành phần chính của **Intent** bao gồm:
 - **Action**: xác định hành động sẽ được thực thi, các hành động này có thể là: `ACTION_VIEW`, `ACTION_EDIT`, `ACTION_MAIN`...
 - **Data**: là các dữ liệu đi kèm với Intent, được sử dụng để hành động (Action) thao tác trên nó.

2.2.2/ Cách gọi Intent

- Tạo đối tượng **Intent** với tham số **action** và **data**.

```
Intent myActivity = new Intent (action, data);  
startActivity (myActivity);
```

Built-in
or
user-created
activity

Primary data (as an URI)
tel://
http://
sendto://

2.2.2/ Các action và data thường dùng

ACTION_DIAL *tel:123*

Display the phone dialer with the given number filled in.

ACTION_VIEW *http://www.google.com*

Show Google page in a browser view. Note how the VIEW action does what is considered the most reasonable thing for a particular URI.

ACTION_EDIT *content://contacts/people/2*

Edit information about the person whose identifier is "2".

ACTION_VIEW *content://contacts/people/2*

Used to start an activity to display 2-nd person.

ACTION_VIEW *content://contacts/people/*

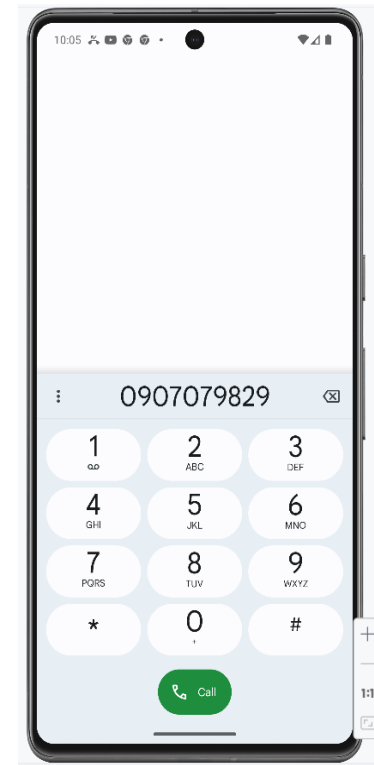
Display a list of people, which the user can browse through. Selecting a particular person to view would result in a new intent

2.2.2/ Các action và data thường dùng

Ví dụ sử dụng action và data

```
Intent it = new Intent(Intent.ACTION_DIAL, Uri.parse(uriString: "tel: 0907079829"));
startActivity(it);
```

Khi thực hiện app sẽ xuất hiện màn hình quay số như sau:

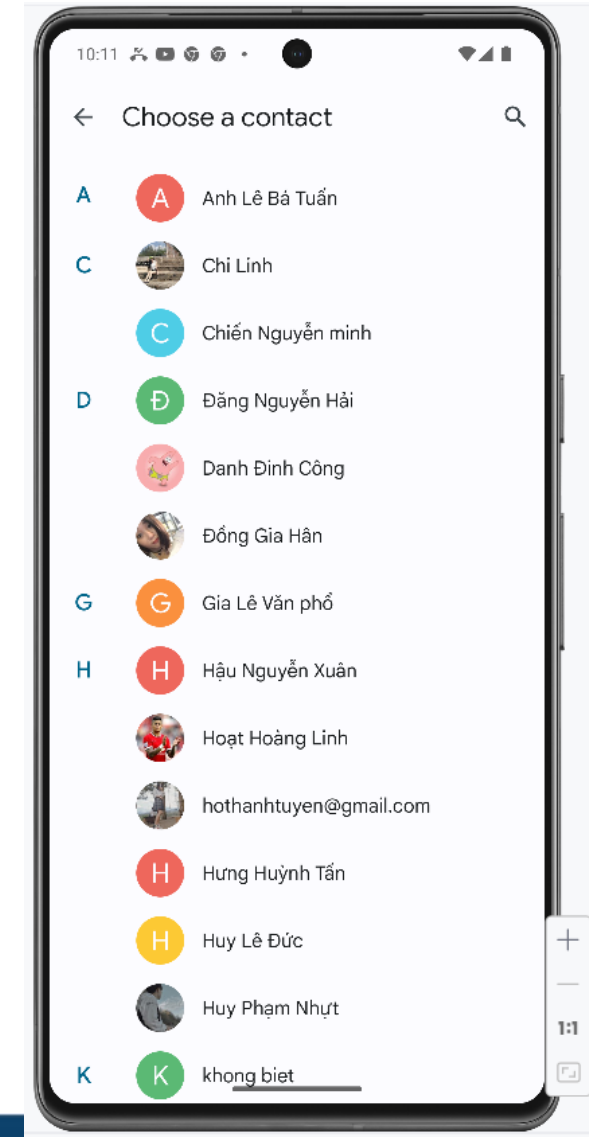


2.2.2/ Các action và data thường dùng

Ví dụ sử dụng action và data

```
Intent it = new Intent(Intent.ACTION_PICK);  
it.setType(ContactsContract.Contacts.CONTENT_TYPE);  
startActivity(it);
```

Khi thực hiện app sẽ mở màn hình danh bạ cho người dùng chọn liên hệ trong danh bạ này:

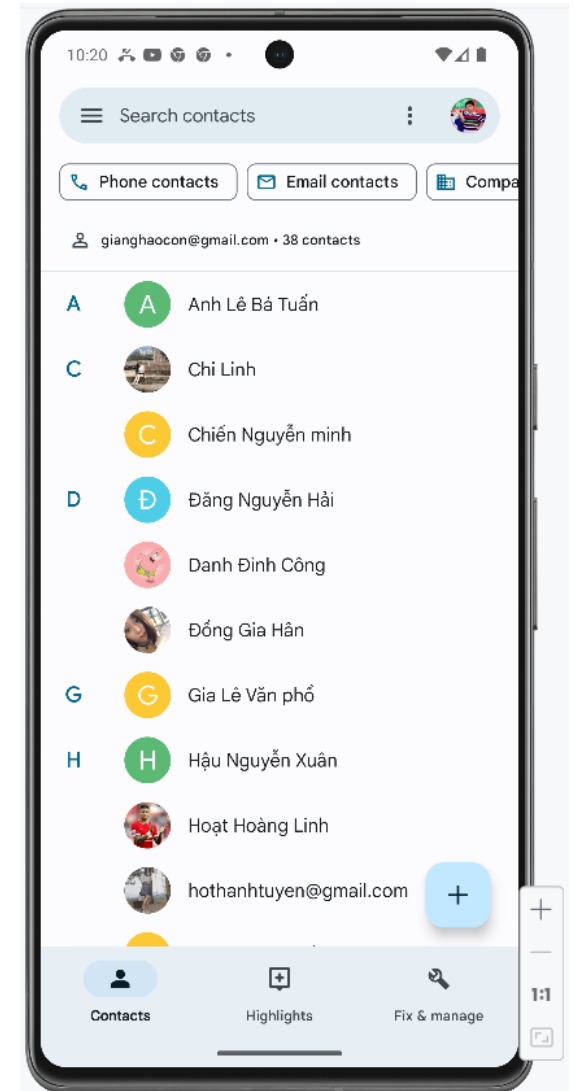


2.2.2/ Các action và data thường dùng

Ví dụ sử dụng action và data

```
String myData = "content://contacts/people/";  
Intent it = new Intent(Intent.ACTION_VIEW, Uri.parse(myData));  
startActivity(it);
```

Khi thực hiện app sẽ mở màn hình danh bạ liên lạc:

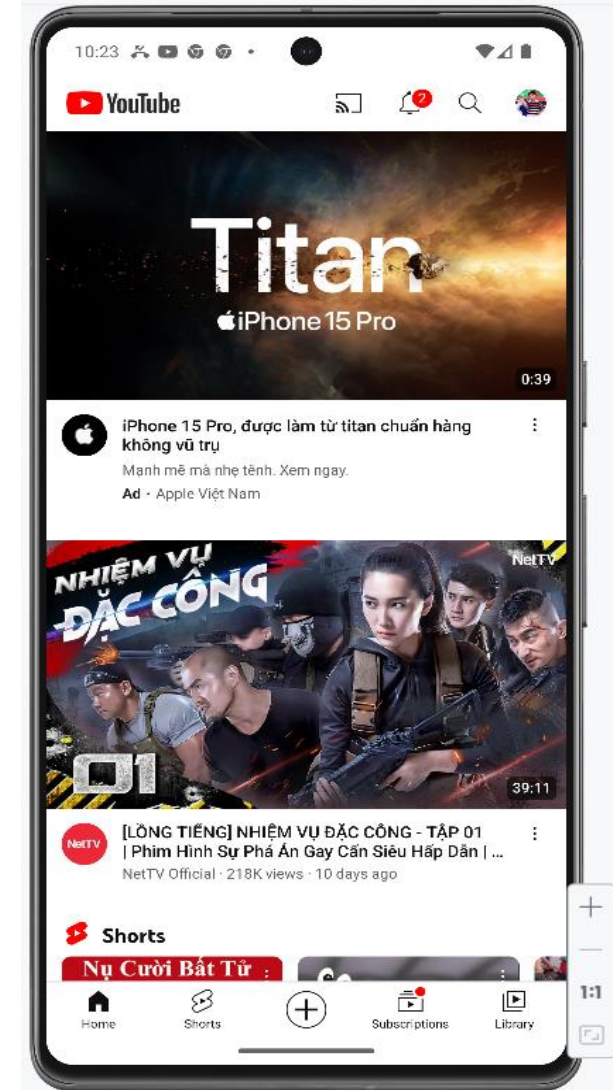


2.2.2/ Các action và data thường dùng

Ví dụ sử dụng action và data

```
String myData = "http://www.youtube.com";  
Intent it = new Intent(Intent.ACTION_VIEW, Uri.parse(myData));  
startActivity(it);
```

Khi thực hiện app sẽ mở màn hình cho xem nội dung web:

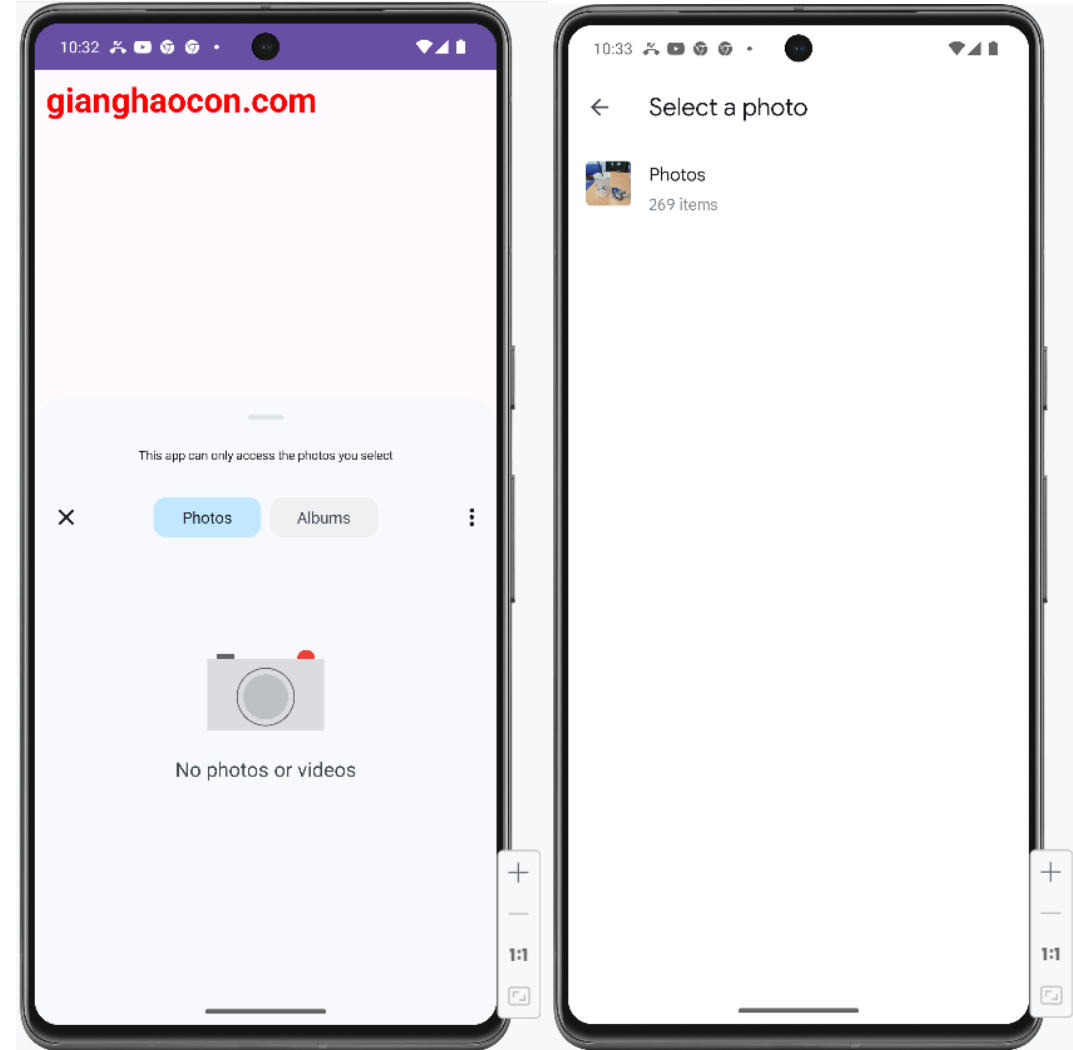


2.2.2/ Các action và data thường dùng

Ví dụ sử dụng action và data

```
Intent it = new Intent();  
it.setType("image/pictures/*");  
it.setAction(Intent.ACTION_GET_CONTENT);  
startActivity(it);
```

Khi thực hiện app sẽ mở màn hình gallery hiển thị các hình ảnh của phone:



2.2.3/ Các loại Intent

■ Có 2 loại Intent:

- Explicit Intent (Intent tường minh) [?](#)
- Implicit Intent (Intent không tường minh) [?](#)



2.2.3/ Các loại Intent – **Implicit Intent**

- **Intent** không chỉ định rõ một component cụ thể. Nhưng nó sẽ chứa các thông tin cần thiết (Action/Data) để hệ thống có thể xác định component thực thi Intent đó.
- **Intent** này thường được dùng để chạy các Component của ứng dụng khác.
- **Ví dụ: mở Activity xem nội dung trang web**

```
String myData = "https://www.w3schools.com/";  
Intent it = new Intent(Intent.ACTION_VIEW, Uri.parse(myData));  
startActivity(it);
```

2.2.3/ Các loại Intent – **Explicit Intent**

- **Intent** xác định rõ một component để xử lý (chẳng hạn tên Activity). Có thể được gán một giá trị cụ thể sử dụng hàm `setComponent()` hoặc `setClass()`.
- **Intent** này thường không chứa bất kỳ thông tin nào khác (như category, type) và thường được dùng để kết nối với các Activity trong cùng một ứng dụng.
- **Ví dụ: khởi chạy 1 Activity khác**

```
Intent it = new Intent(this, SecondActivity.class);  
startActivity(it);
```

2.2.4/ Dữ liệu phụ (extra) của Intent

- Dữ liệu phụ (extra) là các thông tin kèm theo được dùng để cung cấp thông tin bổ sung cho **component như dữ liệu các biến,...**
- Có 2 cách để gắn dữ liệu phụ cho Intent:
 - Dùng hàm **putExtra()** của Intent theo dạng :
`intent.putExtra(<key_name>, <value>)`
 - Dùng đối tượng **Bundle** để gói nhiều dữ liệu.

2.2.4/ Dữ liệu phụ (extra) của Intent

- Ví dụ sử dụng putExtra()

```
Intent it = new Intent(this, SecondActivity.class);  
it.putExtra("HoTen", "Giang Vinh Kiệt");  
it.putExtra("tuoi", 22);
```

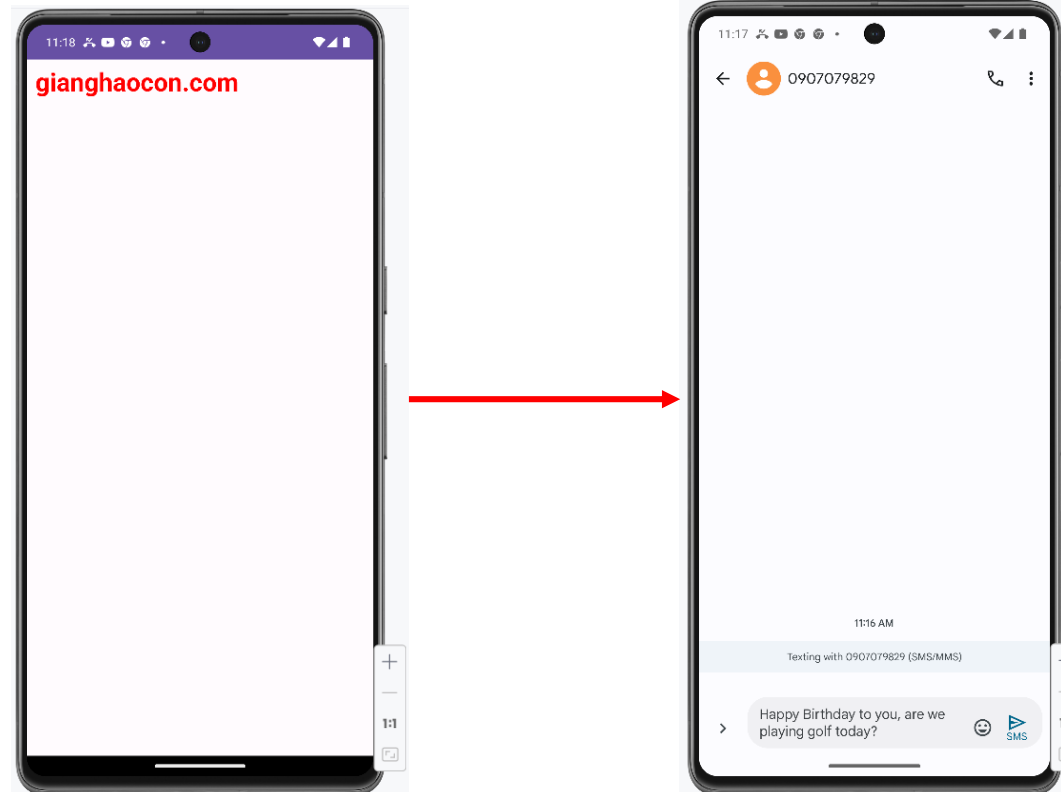
- Ví dụ sử dụng gói Bundle để gắn vào Intent

```
Bundle extra = new Bundle();  
extra.putString("noisinh", "TP.HCM");  
extra.putInt("namsinh", 2000);  
it.putExtras(extra);
```

2.2.4/ Dữ liệu phụ (extra) của Intent

■ Ví dụ sử dụng Intent – Extra để gửi tin nhắn SMS

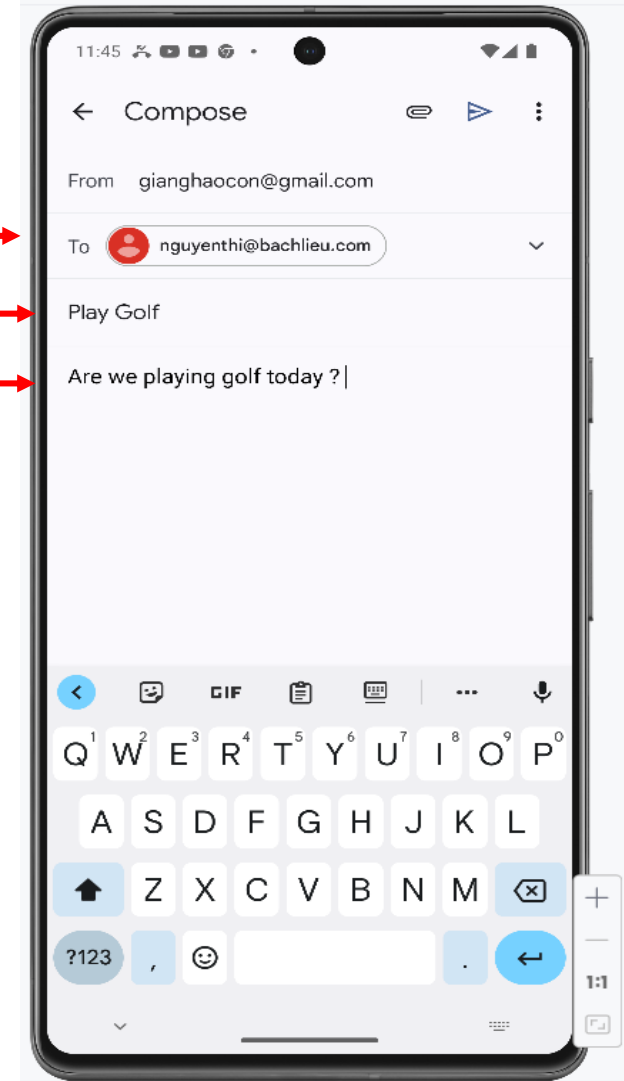
```
Intent it = new Intent(Intent.ACTION_SENDTO, Uri.parse("sms:0907079829"));  
it.putExtra("sms_body", "Happy Birthday to you, are we playing golf today?");  
startActivity(it);
```



2.2.4/ Dữ liệu phụ (extra) của Intent

■ Ví dụ sử dụng Intent – Extra để gửi email

```
Intent it = new Intent(Intent.ACTION_SENDTO);  
it.setData(Uri.parse("mailto:"));  
it.putExtra(Intent.EXTRA_EMAIL, new String[]{"nguyenthi@bachlieu.com"});  
it.putExtra(Intent.EXTRA_SUBJECT, "Play Golf");  
it.putExtra(Intent.EXTRA_TEXT, "Are we playing golf today ?");  
startActivity(it);
```



2.2.5/ Truy xuất dữ liệu extra

- Truy xuất trực tiếp từ đối tượng intent, dùng các phương thức có sẵn của Intent theo dạng:

`Intent.get<Kiểu dữ liệu>Extra(<key_name>, <default_value>);`

- Ví dụ:

```
String name = it.getStringExtra("student_name");
```

```
int age = it.getIntExtra("student_age", 0)
```


2.2.5/ Truy xuất dữ liệu extra

Truy xuất thông qua đối tượng bundle:

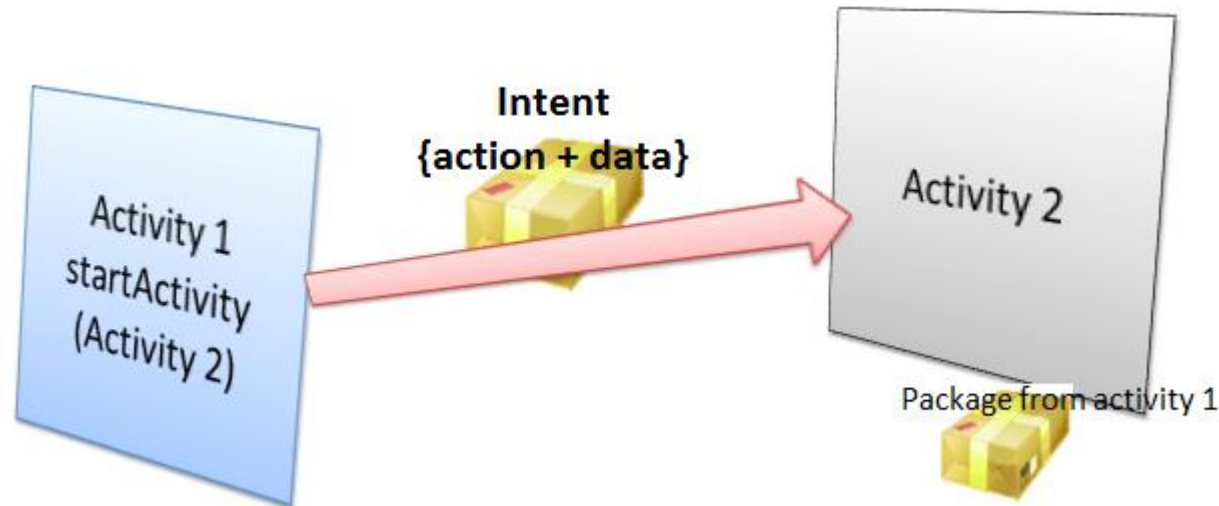
- Dùng hàm `getExtras()` để truy xuất đối tượng Bundle trong Intent.
- Dùng hàm `bundle.get<Kiểu dữ liệu>(<key_name>, <default_value>)` để truy xuất dữ liệu.

Ví dụ:

```
Bundle bundle = it.getExtras();  
String country = bundle.getString("student_country");  
int year = bundle.getInt("student_yearbirth");
```

2.3/ Truyền dữ liệu giữa các Activity

- Một thao tác phổ biến trong lập trình Android là **truyền dữ liệu cho Activity mới được mở ra**.
- Activity mới **xử lý dữ liệu** từ Activity và **trả kết quả cho Activity gửi**.
- Activity ban đầu nhận kết quả và **hiển thị cho người dùng**.



2.3/ Các bước thực hiện

Bước 1: Mở Activity mới với đối tượng Intent thông qua phương thức

`startActivityForResult(Intent, requestCode)`

Bước 2: Nhận và xử lý Intent, sau đó xác nhận thông tin phản hồi thông qua phương thức `setResult()`:

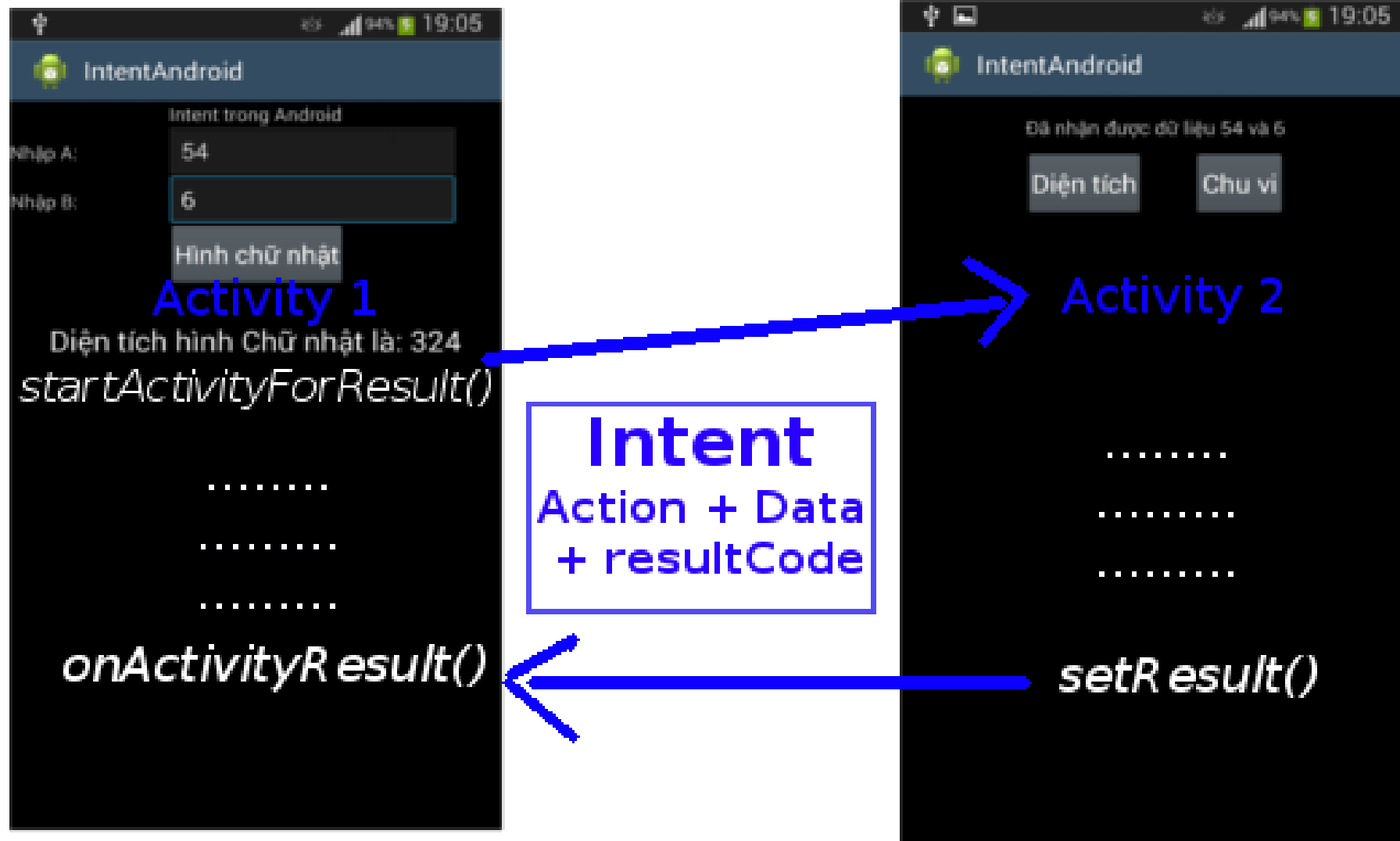
`Intent intent1 = getIntent();`

`setResult(resultCode, Intent2)`

Bước 3: Activity ban đầu ghi đè phương thức sự kiện `onActivityResult()` để nhận kết quả

`onActivityResult(requestCode, resultCode, Intent)`

2.3/ Các bước thực hiện – mô hình



2.4/ Ví dụ truyền dữ liệu giữa các Activity

- ❑ Activity A truyền “**student_name**”, “**student_dtb**”,... cho Activity B (dùng phương thức putExtra).
- ❑ Activity B nhận Intent và dữ liệu của Activity A. Xử lý theo yêu cầu ứng dụng (**Hiển thị ra màn hình logcat, hoặc trên giao diện**)
- ❑ Activity B trả kết quả cho Activity A. Tùy theo **DTB** để trả kết quả là: “Xuat Sac”, “Binh Thuong”
- ❑ Activity A nhận và xử lý kết quả theo yêu cầu (Hiển thị ra màn hình logcat hoặc giao diện).

Bước 01: Mở Activity với dữ liệu

Để truyền dữ liệu cho Activity mới, gán dữ liệu cần truyền vào đối tượng Intent.

- `Intent it = new Intent(this, StudentActivity.class);`
- `it.putExtra("student_name", "Nguyen Tat Thanh");`
- `it.putExtra("student_dtb", 8.5);`
- // Yêu cầu start Activity chỉ định trong Intent.
- `this.startActivityForResult(it, 1234);`

Bước 02: nhận dữ liệu từ Activity gửi

Trong hàm `onCreate` của Activity mới, ta có thể lấy dữ liệu được truyền sang theo các hàm sau:

- `Intent it = getIntent();`
- `String name = itn.getStringExtra("student_name");`
- `float dtb = it.getIntExtra("student_dtb", 0);`

Trong Activity mới, sau khi nhận và xử lý dữ liệu xong, có thể xử lý và gửi trả lại kết quả đã xử lý cho Activity ban đầu như sau:

- `String xloai = (dtb > 8.0) ? "Xuat Sac" : "Binh Thuong";`
- `Intent itRet = new Intent();`
- `itRet.putExtra("xeploai", xloai);`
- `setResult(RESULT_OK, itRet);`

Bước 03: Lấy kết quả trả về

Để lấy kết quả trả về, Activity gửi có thể lấy thông qua hàm sự kiện `onActivityResult()`

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if (requestCode == 1234 && resultCode == RESULT_OK) {
        String xloai = data.getStringExtra("xeploai");
        Log.d(TAG, "Xep loai: " + xloai);
    }
}
```


Ví dụ: Xem Danh Bạ

Tạo ứng dụng cho phép người dùng chọn một contact từ danh bạ và hiển thị chi tiết của contact đó.

- Khởi chạy Activity của ứng dụng hệ thống (Contacts – danh bạ) để chọn một liên lạc. Và chờ nhận kết quả (không đồng bộ).
- Sau khi người dùng chọn một liên lạc từ danh bạ. Ứng dụng danh bạ sẽ trả thông tin cho Activity gọi.
- Activity gọi xử lý dữ liệu nhận được và hiển thị cho người dùng.

Ví dụ: Xem Danh Bạ

- Khởi chạy ứng dụng danh bạ để chọn

```
Intent it = new Intent(Intent.ACTION_PICK);  
it.setType(ContactsContract.Contacts.CONTENT_TYPE);  
startActivityForResult(it, 1234);
```

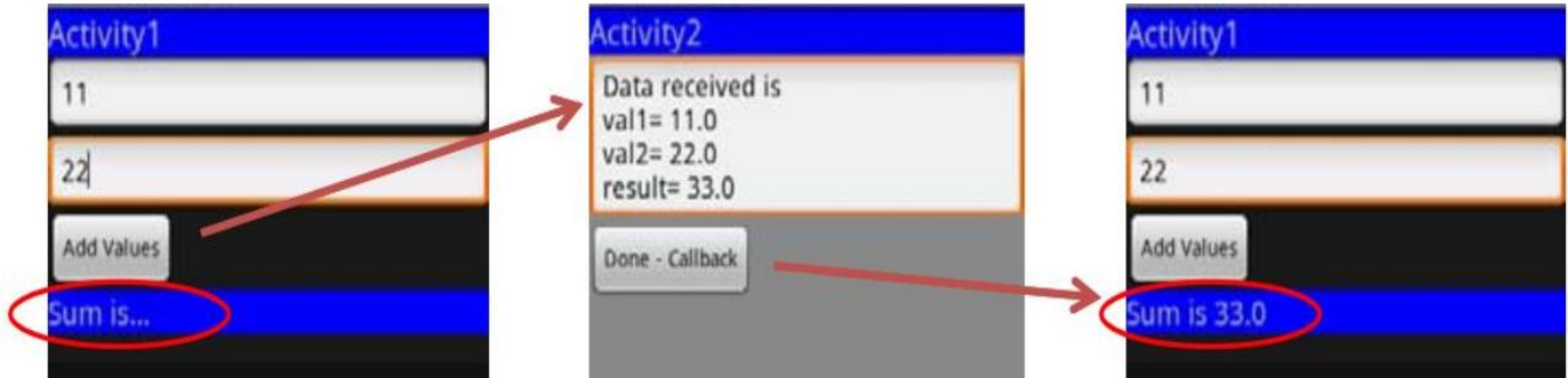
Ví dụ: Xem Danh Bạ

- Activity gọi chờ nhận kết quả và xử lý:

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
                                @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if( requestCode == 1234 && resultCode == RESULT_OK) {
        // View details of selected contact
        Uri uri = data.getData();
        Intent it = new Intent(Intent.ACTION_VIEW);
        it.setData(uri);
        startActivity(it);
    }
}
```

Thực hành: Trao đổi dữ liệu AActivity

Tạo ứng dụng như sau: Activity 1 truyền 2 số thực từ UI cho Activity 2 (theo dạng Bundle). Activity 2 xử lý và trả kết quả tổng cho Activity 1 như sau.



Activity 1: Tạo Intent với Bundle

```
btnAdd.setOnClickListener(new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // get values from the UI  
        Double v1 = Double.parseDouble(txtVal1.getText().toString());  
        Double v2 = Double.parseDouble(txtVal2.getText().toString());  
  
        // create intent to call Activity2  
        Intent myIntentA1A2 = new Intent (Activity1.this, Activity2.class);  
        // create a container to ship data  
        Bundle myData = new Bundle();  
  
        // add <key,value> data items to the container  
        myData.putDouble("val1", v1);  
        myData.putDouble("val2", v2);  
  
        // attach the container to the intent  
        myIntentA1A2.putExtras(myData);  
  
        // call Activity2, tell your local listener to wait for response  
        startActivityForResult(myIntentA1A2, 101);  
    }  
});
```

Activity 2: Nhận Intent và xử lý dữ liệu

```
dataReceived = (EditText) findViewById(R.id.etDataReceived);  
// pick call made to Activity2 via Intent  
Intent myLocalIntent = getIntent();  
  
// look into the bundle sent to Activity2 for data items  
Bundle myBundle = myLocalIntent.getExtras();  
Double v1 = myBundle.getDouble("val1");  
Double v2 = myBundle.getDouble("val2");  
// operate on the input data  
vResult = v1 + v2;  
// for illustration purposes. show data received & result  
dataReceived.setText("Data received is \n"  
    + "val1= " + v1 + "\nval2= " + v2  
    + "\n\nresult= " + vResult);
```

Activity 2: Trả về kết quả cho Activity 1

```
btnDone = (Button) findViewById(R.id.btnDone);  
btnDone.setOnClickListener(new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // add to the bundle the computed result  
        myBundle.putDouble("vresult", vResult);  
  
        // attach updated bundle to invoking intent  
        myLocalIntent.putExtras(myBundle);  
  
        // return sending an OK signal to calling activity  
        setResult(Activity.RESULT_OK, myLocalIntent);  
  
        // terminate Activity2  
        finish();  
    }  
});
```


Activity 1: Nhận kết quả và hiển thị tổng

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    try {
        if ((requestCode == 101 ) && (resultCode == Activity.RESULT_OK)){

            Bundle myResults = data.getExtras();

            Double vresult = myResults.getDouble("vresult");

            lblResult.setText("Sum is " + vresult);

        }
    }
    catch (Exception e) {
        lblResult.setText("Problems - " + requestCode + " " + resultCode);
    }
}
//onActivityResult
```

2.5/ Intent Filter

- Bất cứ thành phần nào (Activity, BroadcastReceiver, Service) khi muốn sử dụng trong ứng dụng đều phải được đăng kí trong file AndroidManifest.xml.
- Intent-Filter là bản đặc tả có cấu trúc của các giá trị của Intent dùng để xác định component phù hợp để thực hiện các hành động được đặc tả trong Intent.
- Android dùng Intent-Filter để xác định các Activity, Service, hay Broadcast Receiver nào phù hợp với Intent thông các thuộc tính Action, Category, Data scheme tương ứng với Intent đó.

2.5/ Intent Filter

```
<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=".LayoutActivity">
    <intent-filter>
        <action android:name="android.intent.action.SENDTO" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="sms" />
    </intent-filter>
</activity>
```

2.5/ Intent Filter

```
AndroidManifest.xml ×
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.o7planning.explicitintentexample" >

    <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=".GreetingActivity" >
        </activity>

    </application>
</manifest>
```


2.6/ Luật xác định Intent-Filter

- Hệ thống sẽ tiến hành xem xét từ trên xuống
- Khi một Intent được gọi, Android sẽ tìm kiếm những component có action-name phù hợp với Intent. Nếu có sẽ mở component đó lên để thực thi.
 - Nếu có nhiều hơn một component có action-name phù hợp thì Android sẽ yêu cầu người dùng chọn component phù hợp.
 - Ngược lại, nếu không có component nào phù hợp Android sẽ tiến hành xem xét kiểu dữ liệu của Intent, sau đó là Category.

Câu hỏi thảo luận

- 1) Resource là gì ? Trình bày các loại Resource của ứng dụng Android ?
- 2) Intent là gì ? Có mấy loại Intent ? Trình bày các loại Intent
- 3) Trình bày các bước truyền dữ liệu từ Activity A cho Activity B và ngược lại.