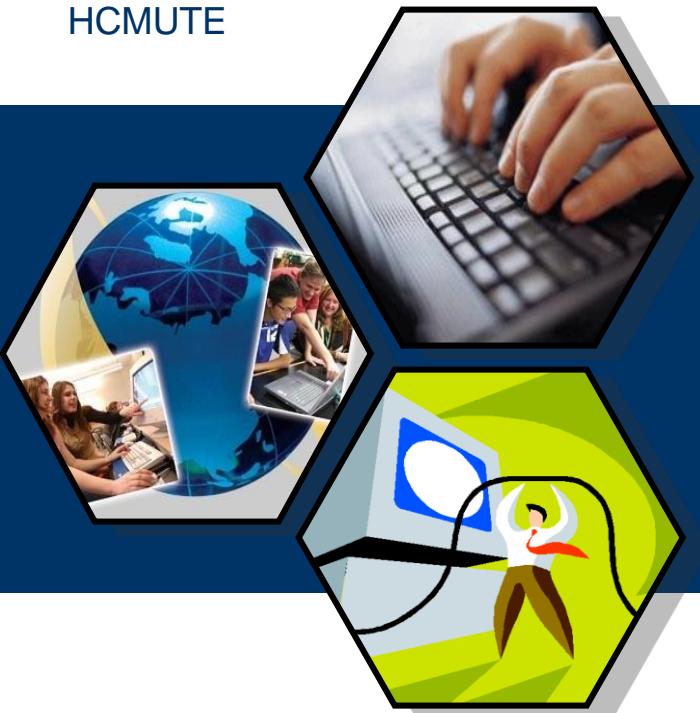




HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN

UTEx



## GIỚI THIỆU ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



ThS. Nguyễn Hữu Trung

# ANDROID LÀ GÌ?

4

- Android là một hệ điều hành mã nguồn mở và là một hệ điều hành dựa trên Linux. Ban đầu Android được phát triển bởi Công ty Android với sự hỗ trợ tài chính từ Google, sau đó được Google mua lại vào năm 2005.
- Phiên bản beta của Android Software Development Kit (SDK) được công bố bởi Google vào năm 2007, trong khi phiên bản thương mại đầu tiên Android 1.0 được công bố 9/2008.
- Vào 27/6/2012, tạo hội nghị Google I/O, Google công bố phiên bản Android tiếp theo là 4.1 Jelly Bean. Jelly Bean là một bản cập nhật với mục đích đầu tiên là cải thiện giao diện người dùng (User Interface), cả về tính năng lẫn hiệu suất.
- Mã nguồn cho Android là miễn phí. Google công bố hầu hết các code dưới Apache License version 2.0, và phần còn lại, các thay đổi Linux Kernel dưới GNU General Public License version 2.

- **Android** là mã nguồn mở
- Bất kỳ ai cũng tùy biến lại **Android Platform**
- Người sử dụng có thể lựa chọn nhiều ứng dụng trên **Android**.
- **Android** cung cấp nhiều tính năng thú vị như: thời tiết, mở màn hình, RSS (Really Simple Syndication) vv
- **Android** cũng hỗ trợ cho các dịch vụ nhắn tin (SMS và MMS), trình duyệt web, lưu trữ dữ liệu (SQLite), kết nối (GSM, CDMA, BlueTooth, Wi-Fi vv), media....

- Android chiếm hơn 87,7% (2017), 69.74% (2022) thị phần điện thoại thông minh trên toàn thế giới với khoảng hơn 2 tỷ thiết bị đã được kích hoạt và hơn 1,3 triệu lượt kích hoạt mỗi ngày.



Music



News



Multimedia



Sports



Lifestyle



Food &amp; Drink



Travel



Weather



Books



Business



Reference



Navigation



Social Media



Utilities



Finance

- Hiện tại, dãy Codename của Android từ A tới T, như Astro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwitch, Jelly Bean, KitKat và Lollipop,..., Pie 2018, Android 10 Q (2019), Red velvet cake(2020), Snow Cone(2021), Tiramisu (2022).



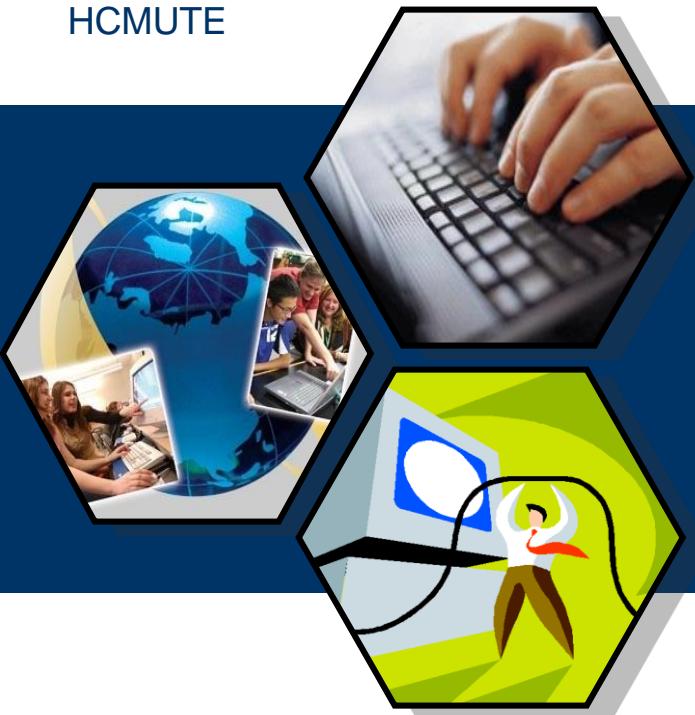


HCMUTE

KHOA CÔNG NGHỆ THÔNG TIN

UTEx

## CÀI ĐẶT JDK và ANDROID



Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

ThS. Nguyễn Hữu Trung



9

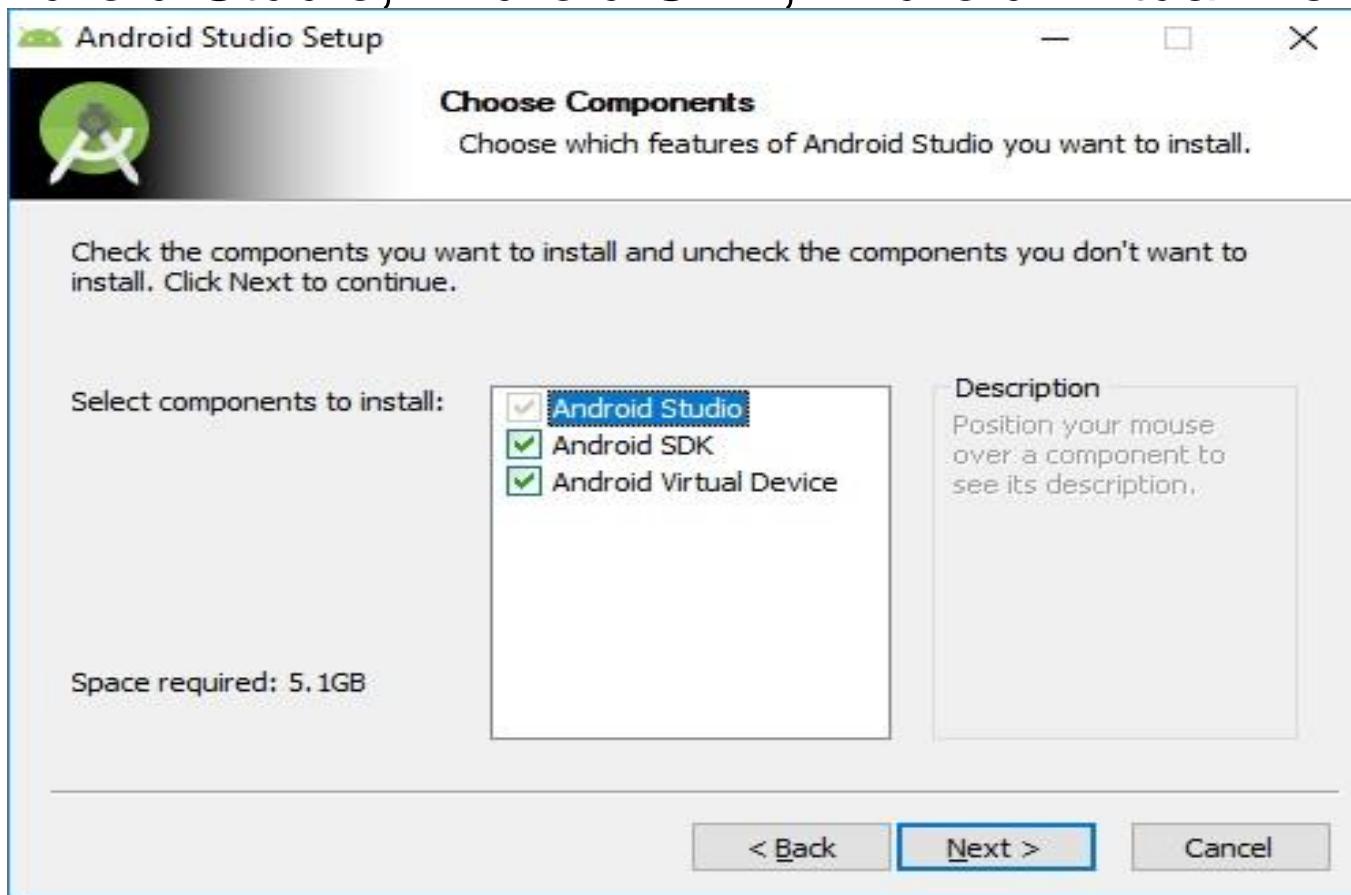
- ❑ Tải JDK 8 trở lên và cài đặt vào máy.
- ❑ Cấu hình môi trường JDK trên máy tính.

- Bạn có thể phát triển ứng dụng Android ở môi trường Windows, Mac OS và cả Linux. Ngoài ra có rất nhiều công cụ để xây dựng các ứng dụng Android phải kể đến như Android Studio, IntelliJ IDEA, Eclipse, Xamarin...
  - Thiết lập JDK (Java Development Kit)
  - Thiết lập Android IDEs

- **Bước 1:** Bạn nhấp đúp chuột vào tập tin Android Studio vừa tải về, sau đó nhấn Next để xác nhận cài đặt Android Studio.



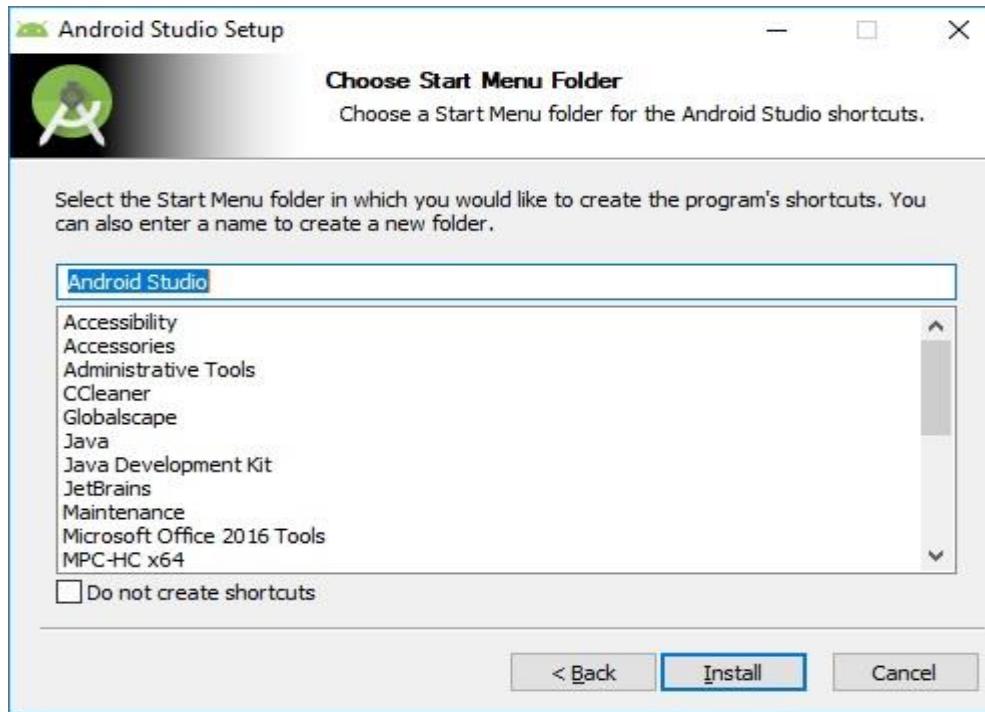
- Bạn hãy tích chọn tất cả các tùy chọn hiện có bao gồm Android Studio, Android SDK, Android Virtual Device.



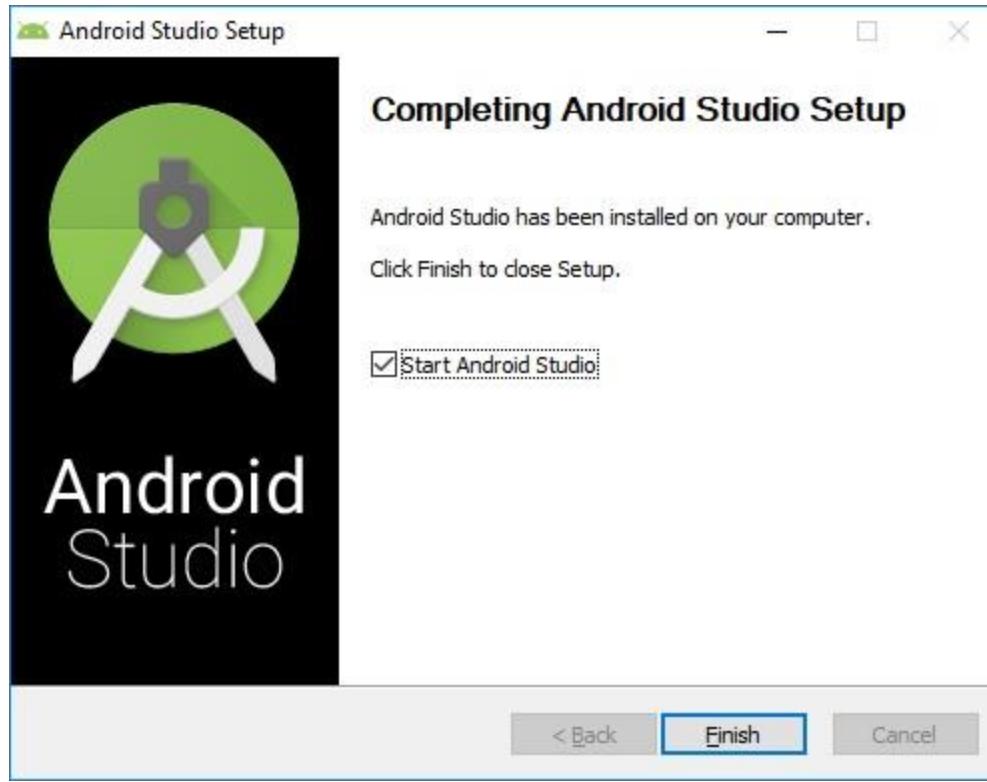
- Đọc xong phần giấy phép, điều khoản xong thì nhấn chọn **I Agree** để đồng ý cài đặt Android Studio.



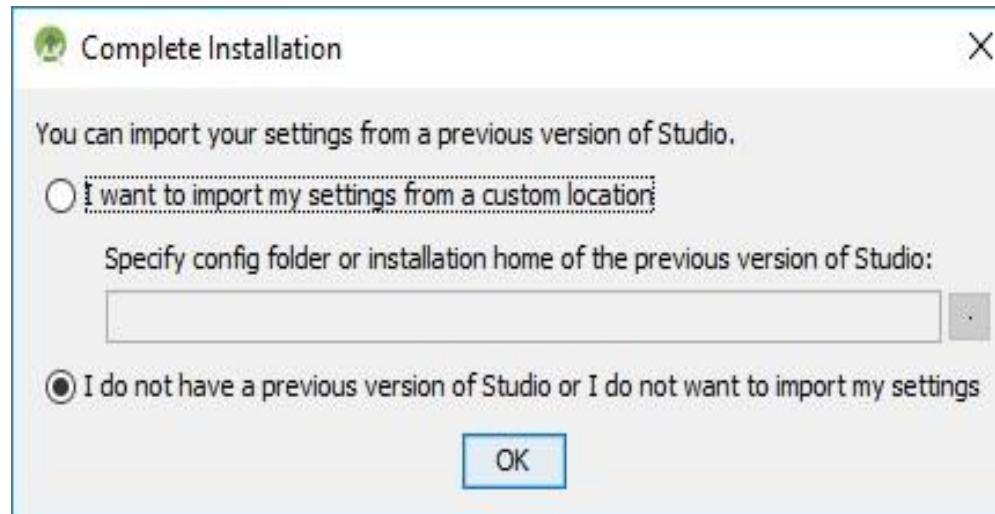
- Màn hình tiếp theo là lựa chọn nơi lưu trữ, bạn có thể để mặc định và nhấn **Next** để qua bước kế tiếp.
- Bây giờ hãy nhấn chọn **Install** để bắt đầu tiến hành cài đặt Android Studio.



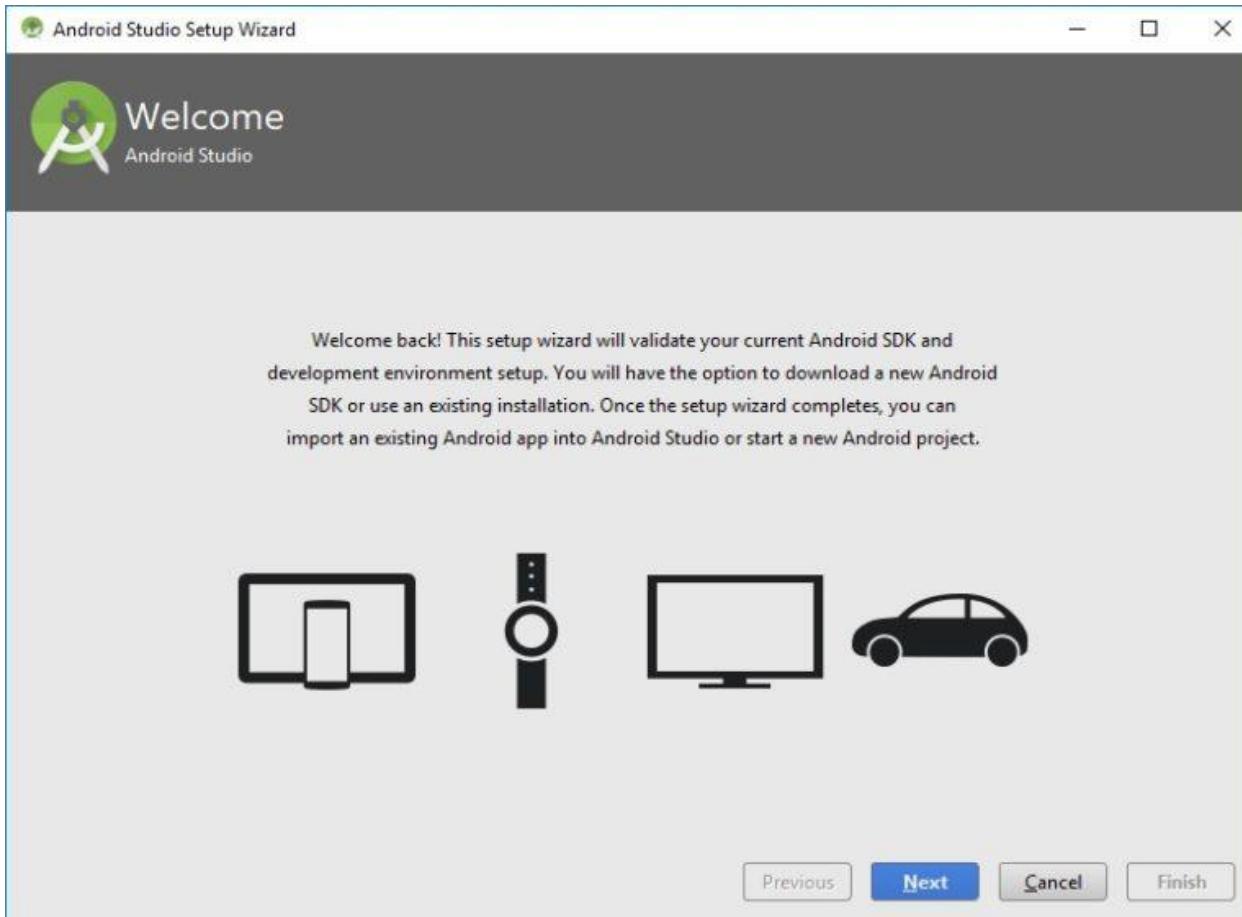
- Hãy chờ đợi quá trình cài đặt diễn ra tùy cấu hình từng máy, sau có thông báo cài đặt thành công bạn nhấn **Finish**.



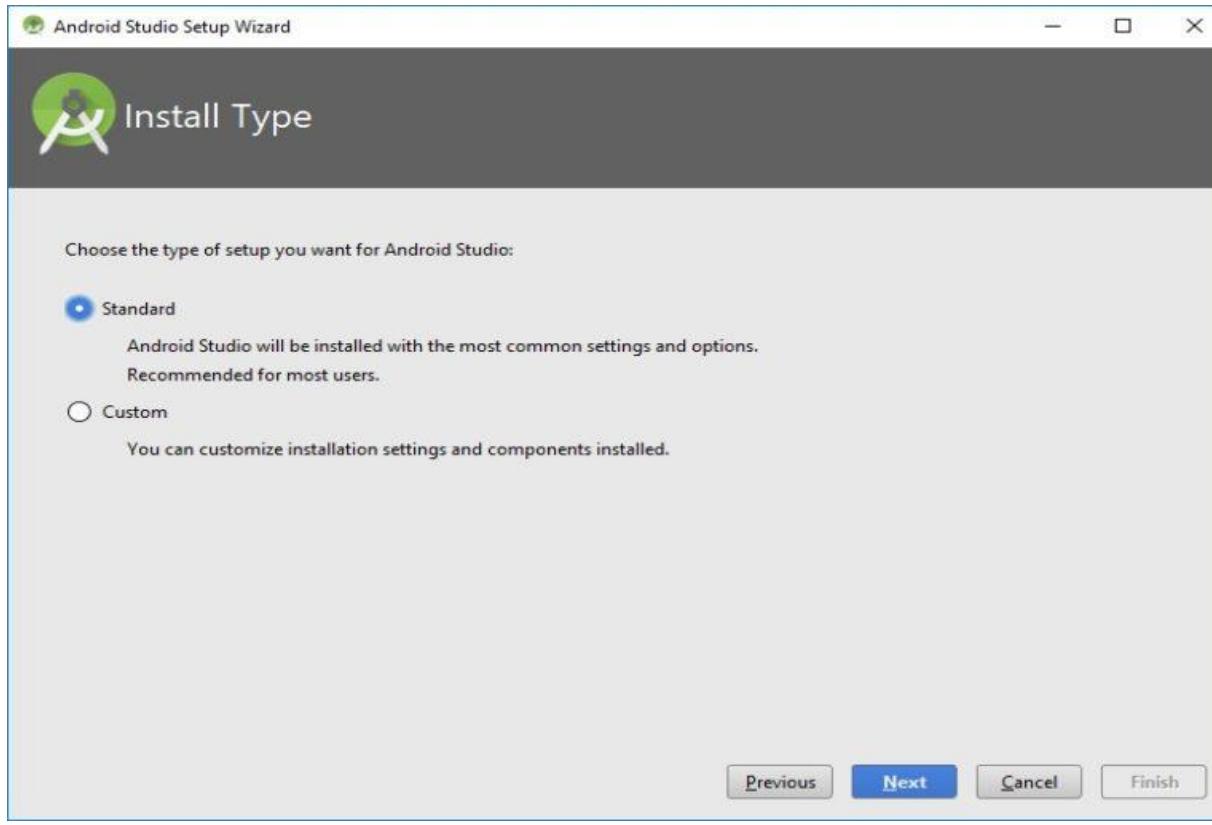
- **Bước 2:** Bây giờ bạn hãy nhấn chọn các bước như sau đây.



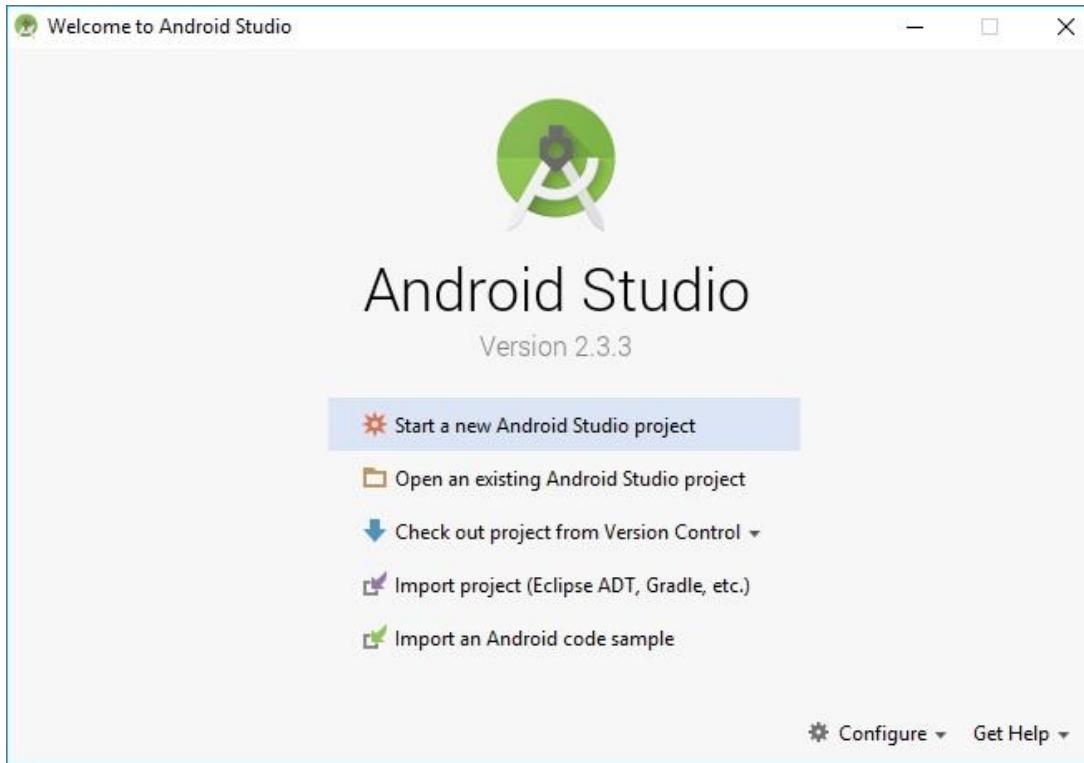
- Nhấn **Next** để qua bước kế tiếp.



- Nhấn **Next** để qua bước kế tiếp. Rồi nhấn **Finish** để qua bước kế tiếp



- **Bước 3:** Tiếp tục chờ đợi để quá trình cài đặt diễn ra, sau khi cài đặt xong nhấn **Finish** để hoàn thành quá trình.



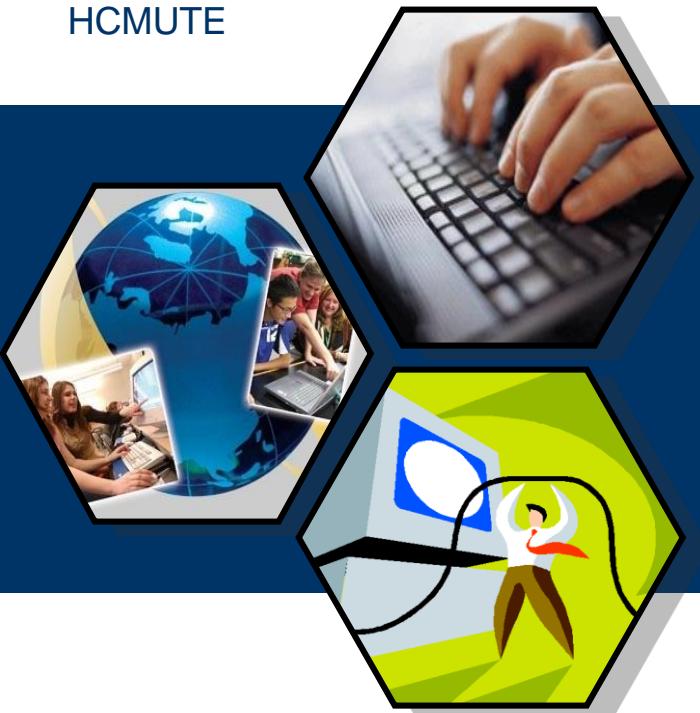


HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN



## TẠO PROJECT ANDROID



**Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM**

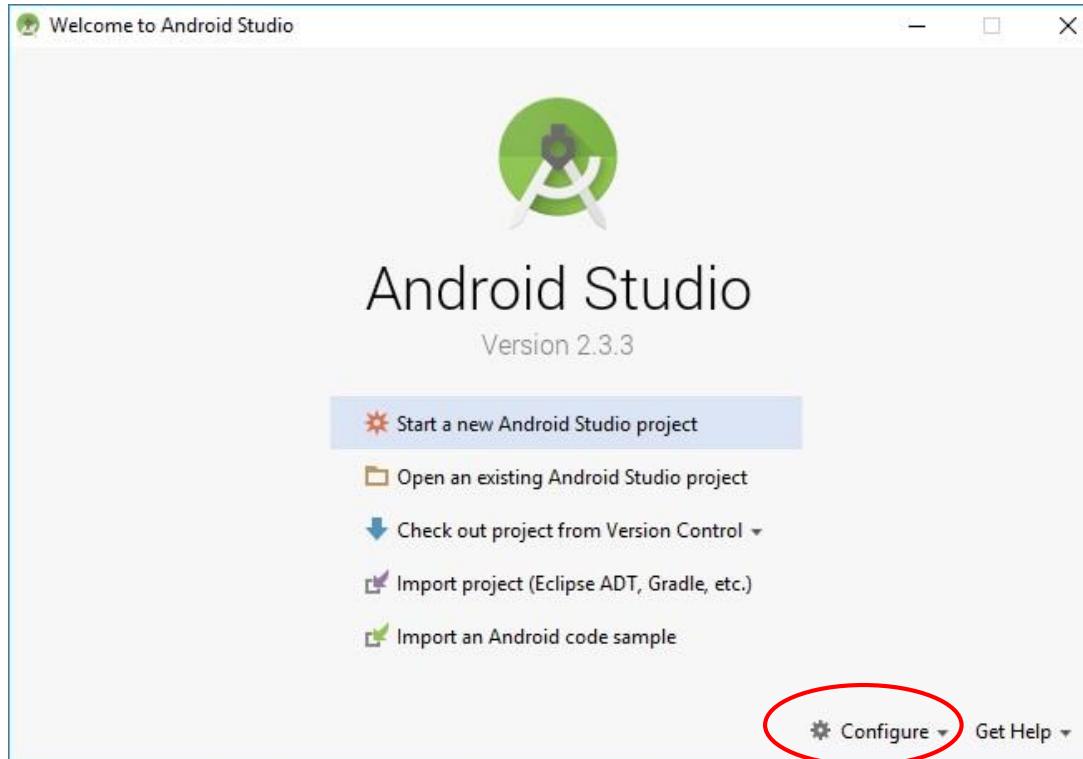
ThS. Nguyễn Hữu Trung



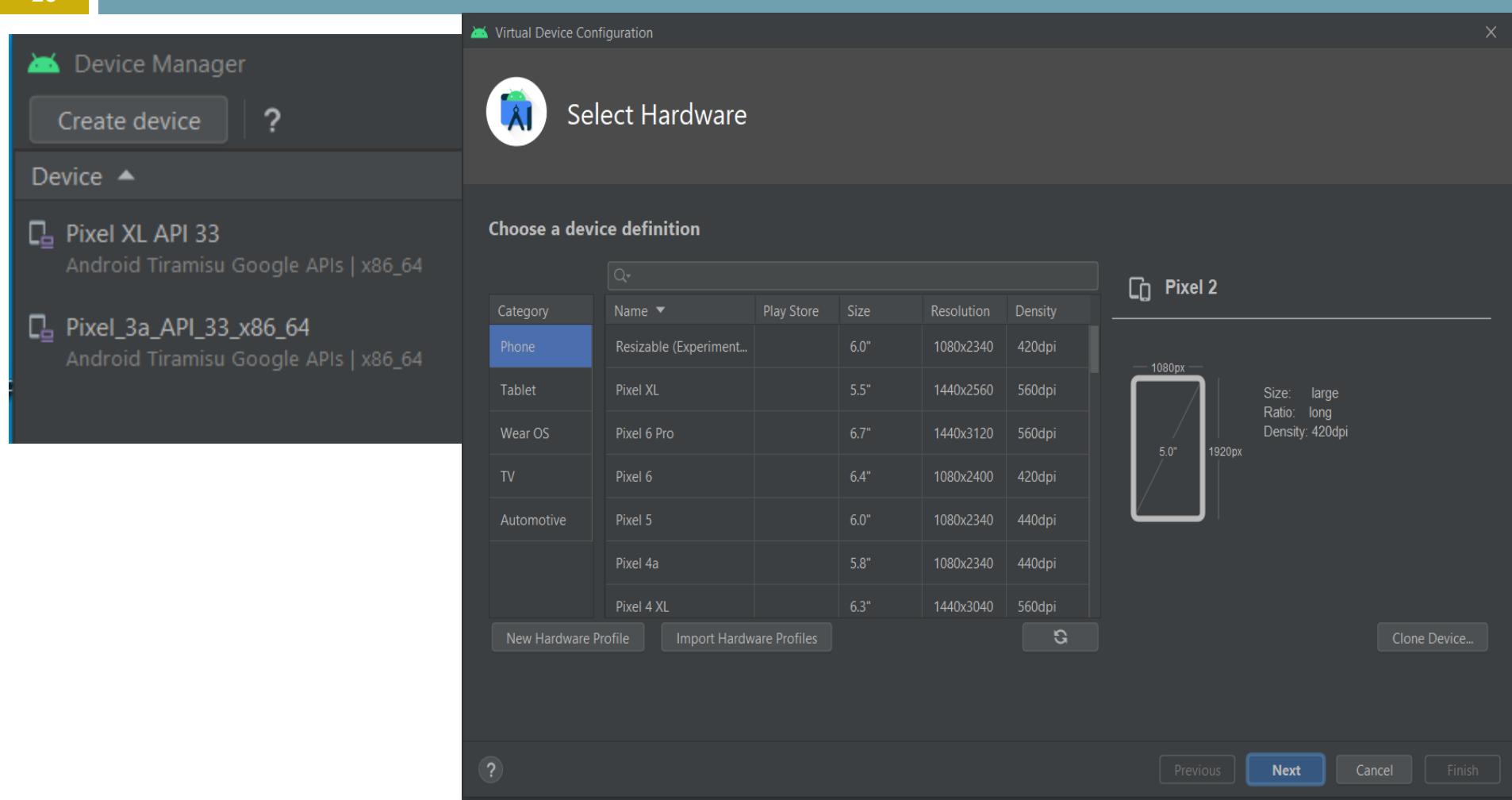
- Khởi động Android Studio lần đầu tiên, bạn chọn Configure để cấu hình máy ảo mô phỏng.



- Khởi động Android Studio lần đầu tiên, bạn chọn Configure để cấu hình máy ảo mô phỏng (Virtual Device manager).



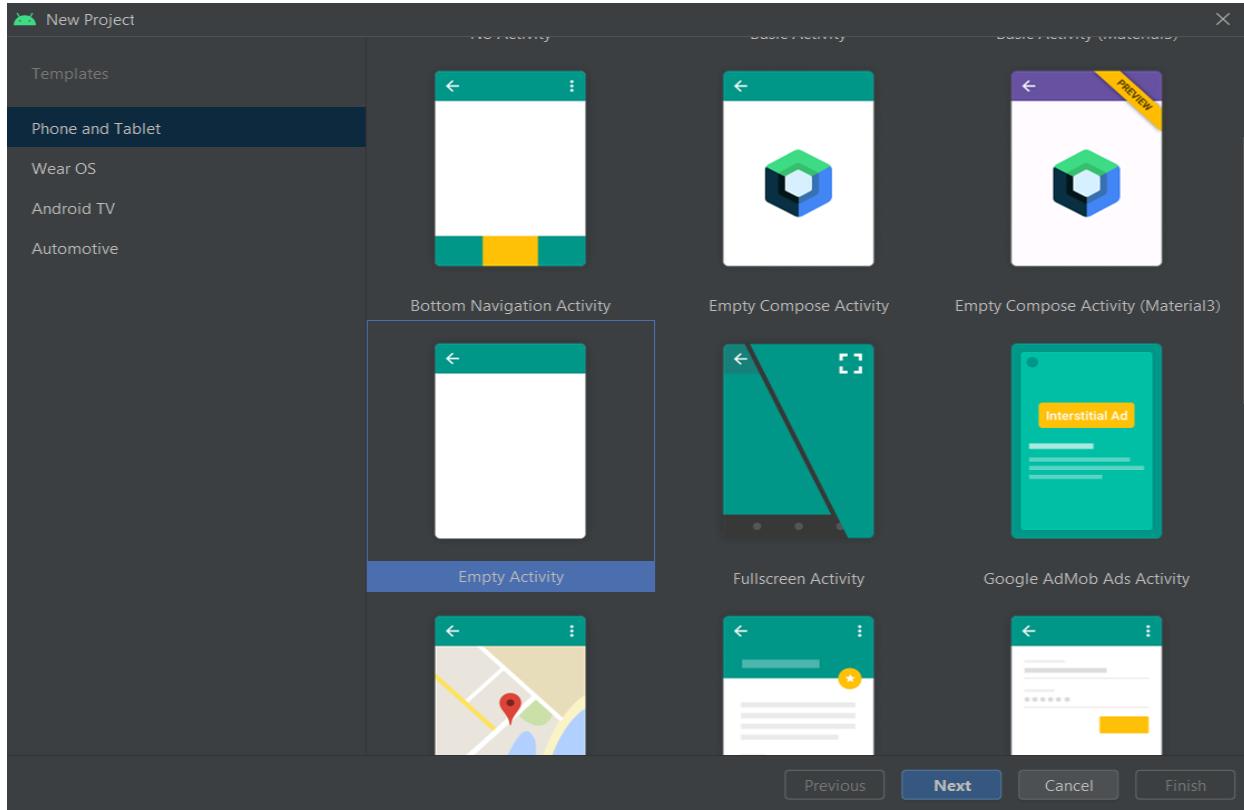
23

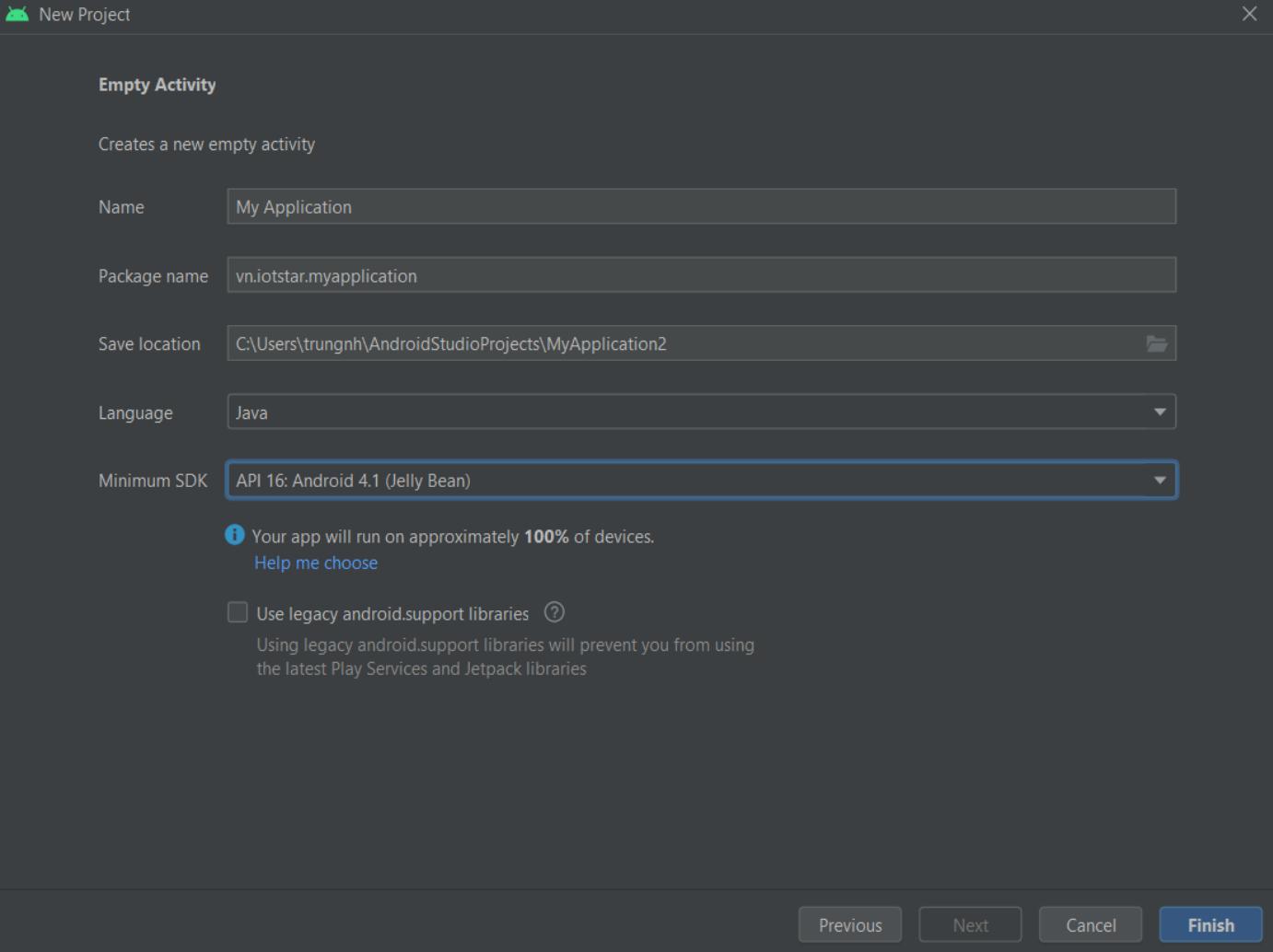


The screenshot shows the "Virtual Device Configuration" window. On the left, the "Device Manager" sidebar lists two devices: "Pixel XL API 33" and "Pixel\_3a\_API\_33\_x86\_64". The main window is titled "Select Hardware" and displays a table of device definitions under the heading "Choose a device definition". A search bar is at the top of the table. The table has columns for Category, Name, Play Store, Size, Resolution, and Density. The "Phone" category is selected. A row for "Pixel 2" is highlighted, showing its dimensions: 1080px width, 5.0" diagonal, and 1920px height. The table also lists other devices like Pixel XL, Pixel 6 Pro, Pixel 6, Pixel 5, Pixel 4a, and Pixel 4 XL. At the bottom of the table are buttons for "New Hardware Profile", "Import Hardware Profiles", and a refresh icon. Below the table are buttons for "?", "Previous", "Next", "Cancel", and "Finish".

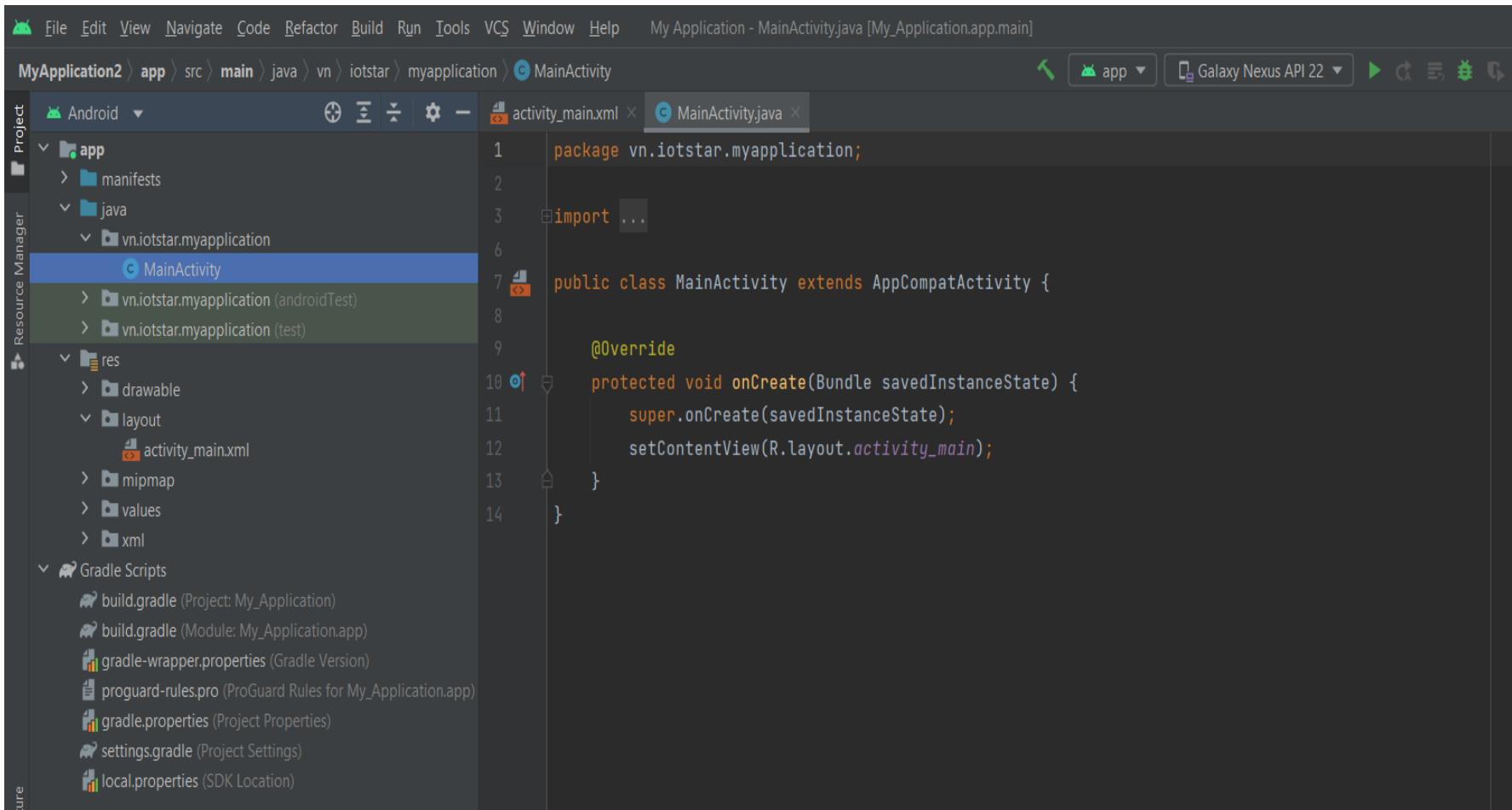
Category	Name	Play Store	Size	Resolution	Density
Phone	Resizable (Experiment...)		6.0"	1080x2340	420dpi
Tablet	Pixel XL		5.5"	1440x2560	560dpi
Wear OS	Pixel 6 Pro		6.7"	1440x3120	560dpi
TV	Pixel 6		6.4"	1080x2400	420dpi
Automotive	Pixel 5		6.0"	1080x2340	440dpi
	Pixel 4a		5.8"	1080x2340	440dpi
	Pixel 4 XL		6.3"	1440x3040	560dpi

- Khởi động Android Studio chọn New project rồi chọn Empty Activity





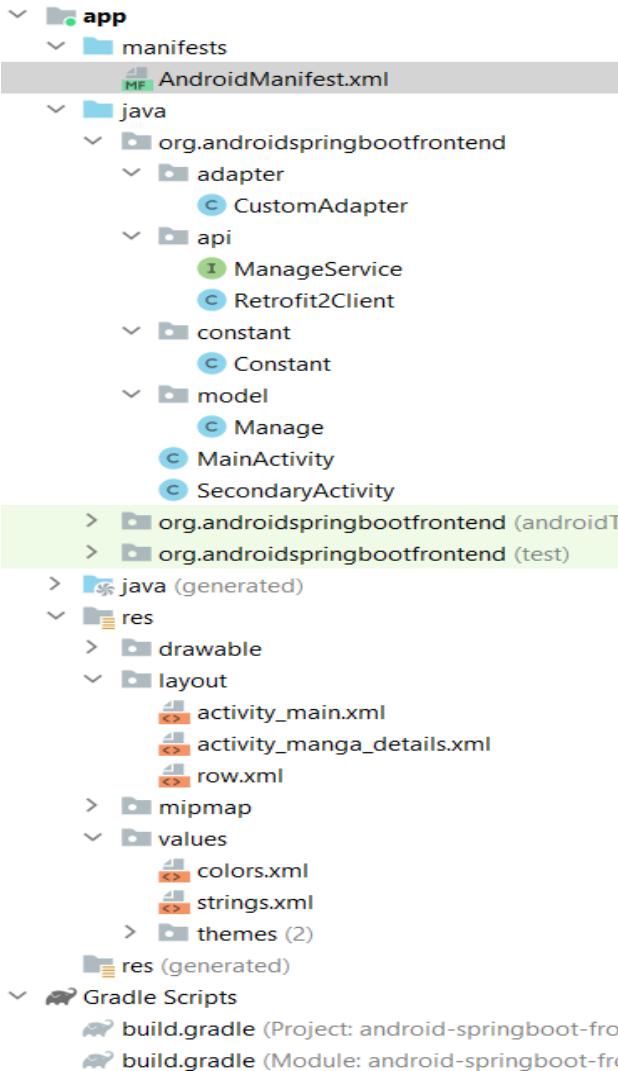
□ Chọn  
các  
thông tin  
rồi bấm  
Finish



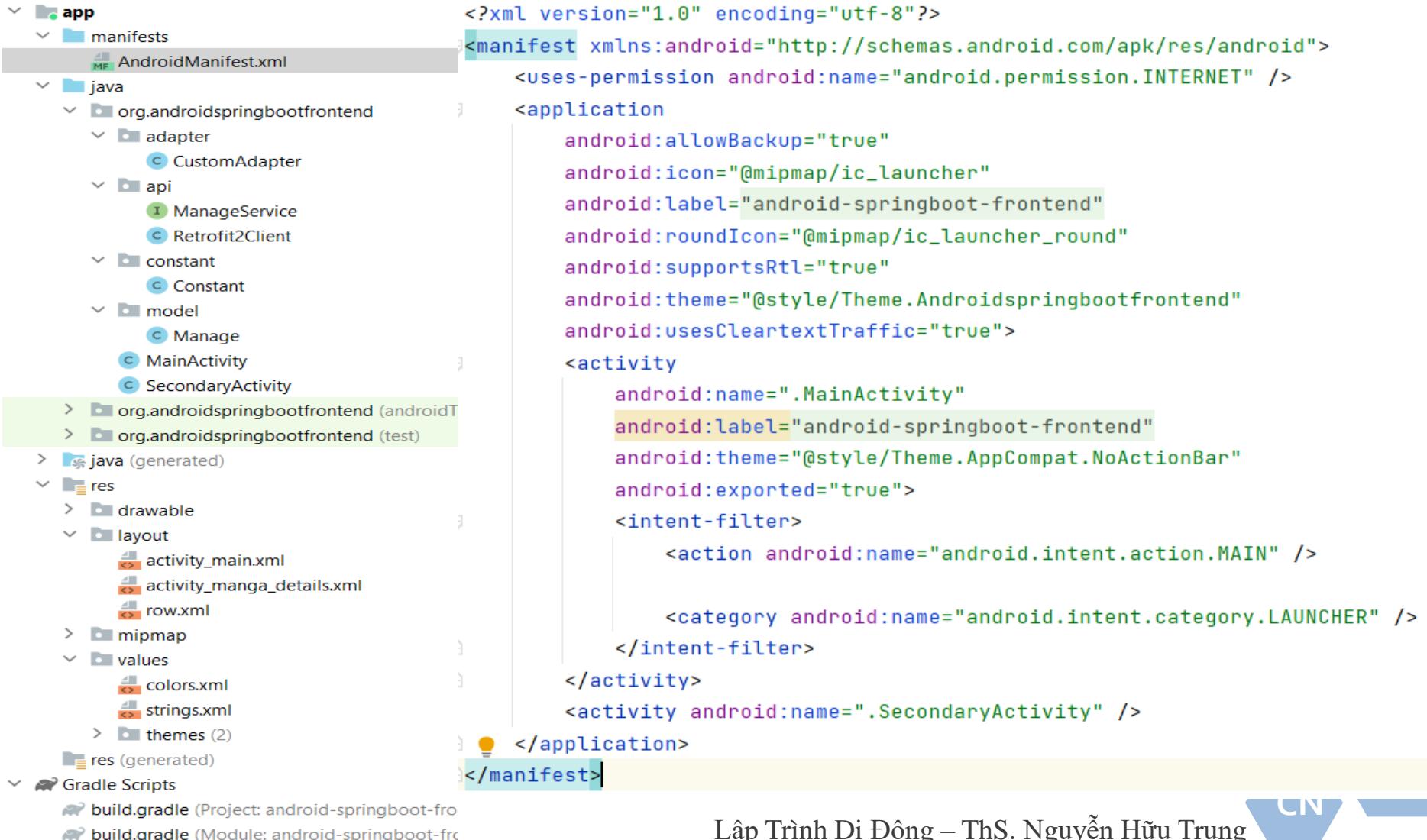
The screenshot shows the Android Studio interface with the following details:

- File Bar:** File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
- Title Bar:** My Application - MainActivity.java [My\_Application.app.main]
- Project Tree (Project Manager):**
  - app
    - manifests
    - java
      - vn.iotstar.myapplication
      - MainActivity
    - vn.iotstar.myapplication (androidTest)
    - vn.iotstar.myapplication (test)
  - res
    - drawable
    - layout
      - activity\_main.xml
    - mipmap
    - values
    - xml
  - Gradle Scripts
    - build.gradle (Project: My\_Application)
    - build.gradle (Module: My\_Application.app)
    - gradle-wrapper.properties (Gradle Version)
    - proguard-rules.pro (ProGuard Rules for My\_Application.app)
    - gradle.properties (Project Properties)
    - settings.gradle (Project Settings)
    - local.properties (SDK Location)

- Code Editor:** activity\_main.xml and MainActivity.java
- Toolbar:** Includes icons for Undo, Redo, Run, Stop, and others.



- **Thư mục manifests:** chứa thông tin cấu hình của ứng dụng. **AndroidManifest.xml:** tập tin XML chứa tất cả các thông tin cấu hình dùng để build ứng dụng và các thành phần của ứng dụng (activity, service, broadcast receivers, content providers v.v.). Mỗi ứng dụng đều có một tập tin AndroidManifest.xml. Trong ứng dụng, Activity nào muốn sử dụng đều bắt buộc phải có khai báo trong AndroidManifest.xml



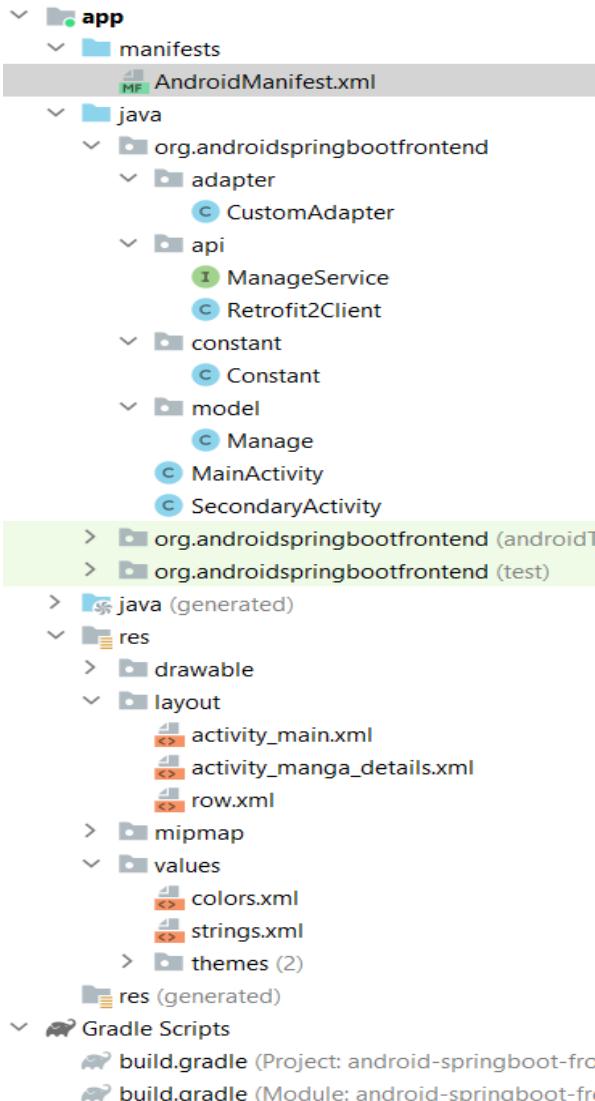
The screenshot shows the Android Studio project structure on the left and the content of the `AndroidManifest.xml` file on the right.

**Project Structure:**

- app**:
  - manifests**: `AndroidManifest.xml`
  - java**:
    - org.androidspringbootfrontend**:
      - adapter**: `CustomAdapter`
      - api**:
        - `ManageService`
        - `Retrofit2Client`
      - constant**: `Constant`
      - model**:
        - `Manage`
        - `MainActivity`
        - `SecondaryActivity`
    - org.androidspringbootfrontend** (androidTest)
    - org.androidspringbootfrontend** (test)
  - java (generated)**
  - res**:
    - drawable**
    - layout**:
      - `activity_main.xml`
      - `activity_manga_details.xml`
      - `row.xml`
    - mipmap**
    - values**:
      - `colors.xml`
      - `strings.xml`
    - themes (2)**
  - res (generated)**
- Gradle Scripts**:
  - `build.gradle` (Project: android-springboot-fro)
  - `build.gradle` (Module: android-springboot-frc)

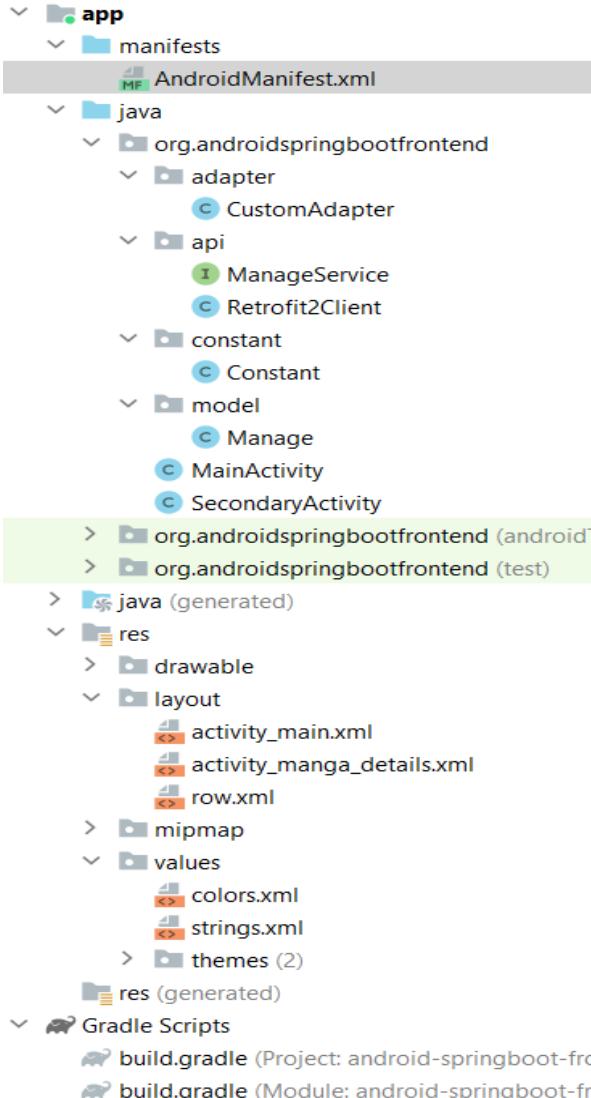
# Cấu trúc project

29



- **Thư mục java:** chứa tất cả các file mã nguồn .java của ứng dụng. Tương ứng với mỗi **Activity** thì file mã nguồn sẽ chứa các xử lý trên Activity đó. Activity nào được khởi chạy đầu tiên khi ứng dụng hoạt động sẽ được khai báo đầu tiên trong tập tin **AndroidManifest.xml**.
- **Thư mục res:** chứa các tài nguyên của ứng dụng, bao gồm các tập tin hình ảnh, các thiết kế giao diện, thực đơn,... của ứng dụng.

# Cấu trúc project



- **Thư mục res:** chứa các tài nguyên của ứng dụng, bao gồm các tập tin hình ảnh, các thiết kế giao diện, thực đơn,... của ứng dụng.

Thư mục	Mô tả
<code>drawable/</code>	Thư mục chứa các hình ảnh của ứng dụng
<code>layout/</code>	Thư mục chứa các tập tin XML dùng để thiết kế giao diện của ứng dụng. Hiện giờ đang có tập tin <i>activity_main.xml</i> chính là phần thiết kế giao diện của ứng dụng <b>Hello world</b> .
<code>mipmap/</code>	Thư mục thường chứa các hình ảnh dạng icon của ứng dụng.
<code>values/</code>	Thư mục quản lý các giá trị, đó có thể là các giá trị chuỗi ( <i>strings.xml</i> ), màu sắc ( <i>colors.xml</i> ),...

# Cấu trúc project

31

app

- manifests
- AndroidManifest.xml**
- java
  - org.androidspringbootfrontend
    - adapter
    - CustomAdapter
  - api
    - ManageService
    - Retrofit2Client
  - constant
  - Constant
  - model
    - Manage
    - MainActivity
    - SecondaryActivity
- org.androidspringbootfrontend (androidT)
- org.androidspringbootfrontend (test)
- java (generated)

res

- drawable
- layout
  - activity\_main.xml
  - activity\_manga\_details.xml
  - row.xml
- mipmap
- values
  - colors.xml
  - strings.xml
- themes (2)

res (generated)

Gradle Scripts

- build.gradle (Project: android-springboot-fro)
- build.gradle (Module: android-springboot-frc)

## Thiết lập và truy xuất Android Resources

```
res/
  drawable/
    icon.png
  layout/
    activity_main.xml
    info.xml
  values/
    strings.xml
```

Thư mục **res/** chứa tất cả các thư mục con. Trong ví dụ trên chúng ta có 1 **image resource**, 2 **layout resource**, và 1 **string resource**.

- anim/**: Chứa các file xml định nghĩa các hiệu ứng. Chúng được truy xuất từ class **R.anim**.
- color/**: Là các file xml định nghĩa về màu sắc. Chúng được truy xuất từ class **R.color**.
- drawable/**: Chứa các file hình ảnh **.png**, **.jpg**, **.gif** hoặc các file xml đã được biên dịch thành bitmaps. Chúng được truy xuất từ class **R.drawable**.
- layout/**: Chứa các file xml định nghĩa giao diện người dùng. Chúng được truy xuất từ class **R.layout**.

app

- manifests
- AndroidManifest.xml
- java

  - org.androidspringbootfrontend

    - adapter
    - CustomAdapter

  - api

    - ManageService
    - Retrofit2Client

  - constant

    - Constant

  - model

    - Manage
    - MainActivity
    - SecondaryActivity

org.androidspringbootfrontend (androidT)

org.androidspringbootfrontend (test)

java (generated)

res

- drawable
- layout

  - activity\_main.xml
  - activity\_manga\_details.xml
  - row.xml

- mipmap
- values

  - colors.xml
  - strings.xml

- themes (2)

res (generated)

Gradle Scripts

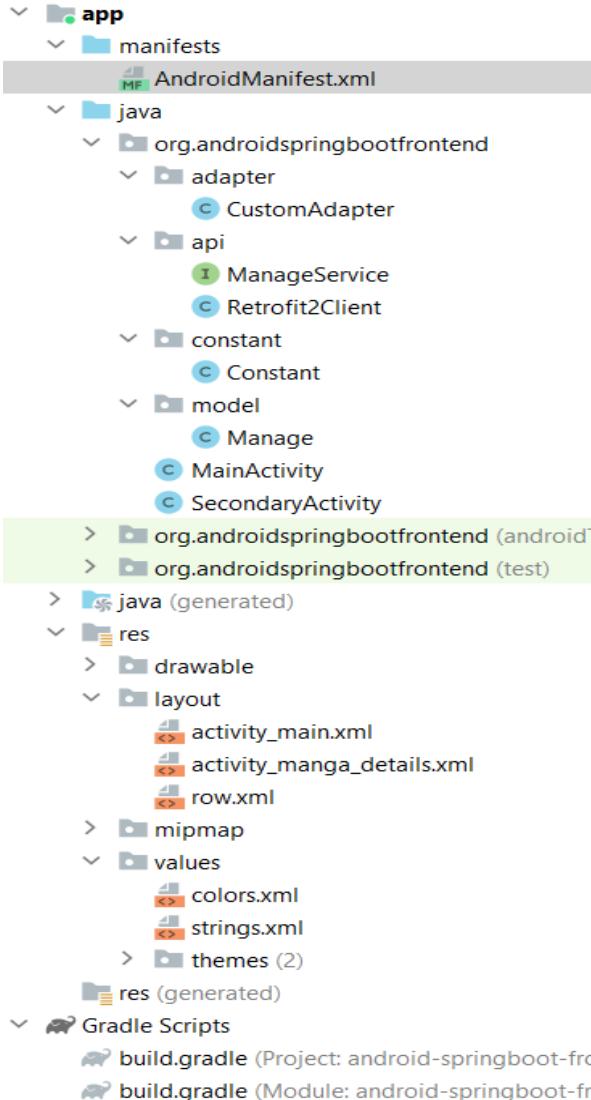
- build.gradle (Project: android-springboot-fro)
- build.gradle (Module: android-springboot-frc)

## Thiết lập và truy xuất Android Resources

```
res/
    drawable/
        icon.png
    layout/
        activity_main.xml
        info.xml
    values/
        strings.xml
```

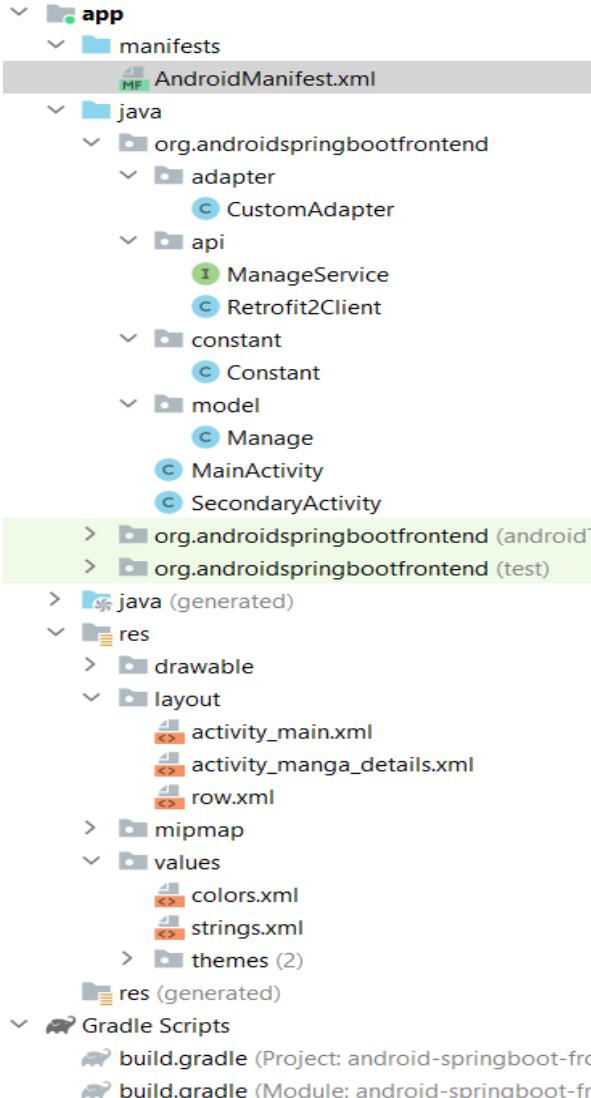
Thư mục **res/** chứa tất cả các thư mục con. Trong ví dụ trên chúng ta có 1 **image resource**, 2 **layout resource**, và 1 **string resource**.

- menu/**: Chứa các file xml định nghĩa menu của ứng dụng, có thể là Options Menu, Context Menu, Sub Menu. Chúng được truy xuất từ class **R.menu**.
- raw/**: Chứa các file bất kỳ, tùy ý. Cần phải gọi **Resources.openRawResource()**. Với **resource ID**, nó là **R.raw.filename** để mở 1 file **raw**.

A screenshot of the Android Studio project structure. The 'app' module is expanded, showing its contents. The 'res' folder is also expanded, showing 'drawable', 'layout', 'values', and 'themes'. The 'values' folder contains 'colors.xml' and 'strings.xml'. The 'layout' folder contains 'activity\_main.xml', 'activity\_manga\_details.xml', and 'row.xml'. The 'res' folder (generated) contains 'mipmap'. The 'java' folder contains several packages: 'org.androidspringbootfrontend' (containing 'adapter', 'api', 'constant', 'model', and 'ManageService'), 'org.androidspringbootfrontend (androidTest)', 'org.androidspringbootfrontend (test)', and 'generated'. The 'Gradle Scripts' folder contains 'build.gradle' (Project) and 'build.gradle' (Module).

## Thiết lập và truy xuất Android Resources

- **values/**: Chứa các file xml để lưu các giá trị đơn giản như string, integers, colors,...
  - + **arrays.xml**: Dùng để định nghĩa các mảng dữ liệu, và được truy xuất từ class **R.array**.
  - + **integers.xml**: Dùng để định nghĩa các số integer và được truy xuất từ class **R.integer**.
  - + **bools.xml**: Dùng để định nghĩa kiểu dữ liệu bool và được truy xuất từ class **R.bool**.
  - + **colors.xml**: Dùng để định nghĩa các giá trị color và được truy xuất từ class **R.color**.



## Thiết lập và truy xuất Android Resources

- **values/**: Chứa các file xml để lưu các giá trị đơn giản như string, integers, colors,...
  - + **dimens.xml**: Dùng để định nghĩa các giá trị về kích thước, chiều. Được truy xuất từ class **R.dimen**.
  - + **strings.xml**: Dùng để định nghĩa các string và được gọi từ class **R.string**.
- **styles.xml**: Dùng để định nghĩa các styles và được gọi từ class **R.style**.
- **xml/**: Chứa các file xml bất kỳ và có thể được đọc bằng cách gọi hàm `Resources.getXML()`. Bạn có thể lưu các file XML cấu hình ở đây và nó sẽ được dùng khi ứng dụng chạy.

## Thiết lập và truy xuất Android Resources

**Ví dụ:** Truy xuất `res/drawable/myImage.png`

```
ImageView imageView = (ImageView)  
findViewById(R.id.myimageview);  
  
imageView.setImageResource(R.drawable.myimage);
```

**Ví dụ:** Truy xuất `res/values/strings.xml`

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <string name="hello">Hello, World!</string>  
</resources>
```

Để sử dụng string: '`hello`'

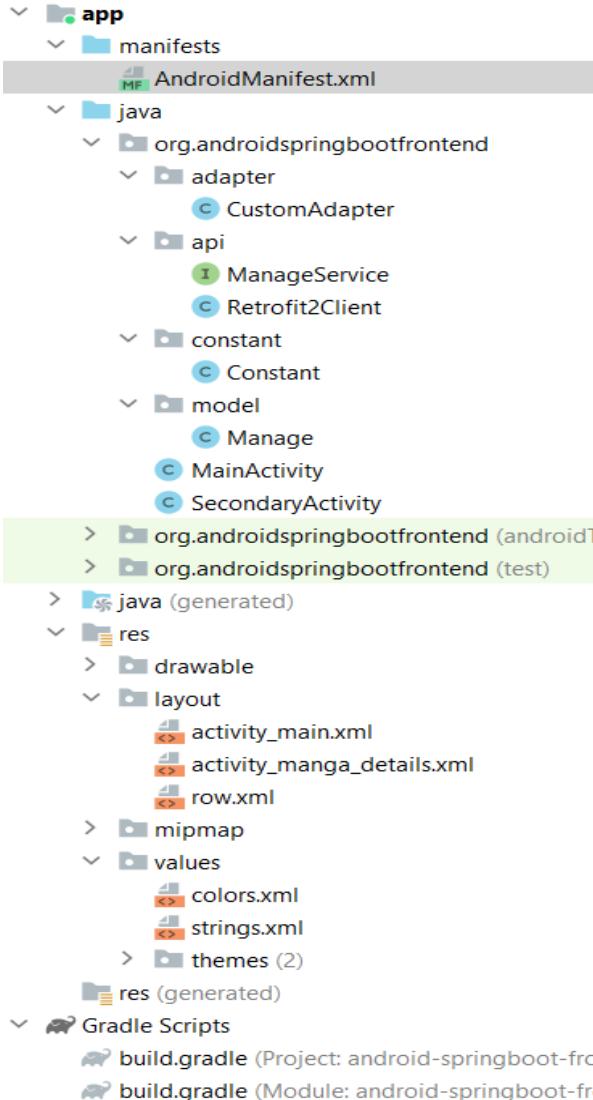
```
TextView msgTextView = (TextView) findViewById(R.id.msg);  
msgTextView.setText(R.string.hello);
```

**Truy xuất resources trong XML**

**Ví dụ:** Truy xuất `res/values/strings.xml`

```
<?xml version="1.0" encoding="utf-8"?>  
  
<resources>  
  
    <color name="opaque_red">#f00</color>  
  
    <string name="hello">Hello!</string>  
  
</resources>
```

```
        android:textColor="@color  
        /opaque_red"  
        android:text="@string/hel  
        lo" />
```

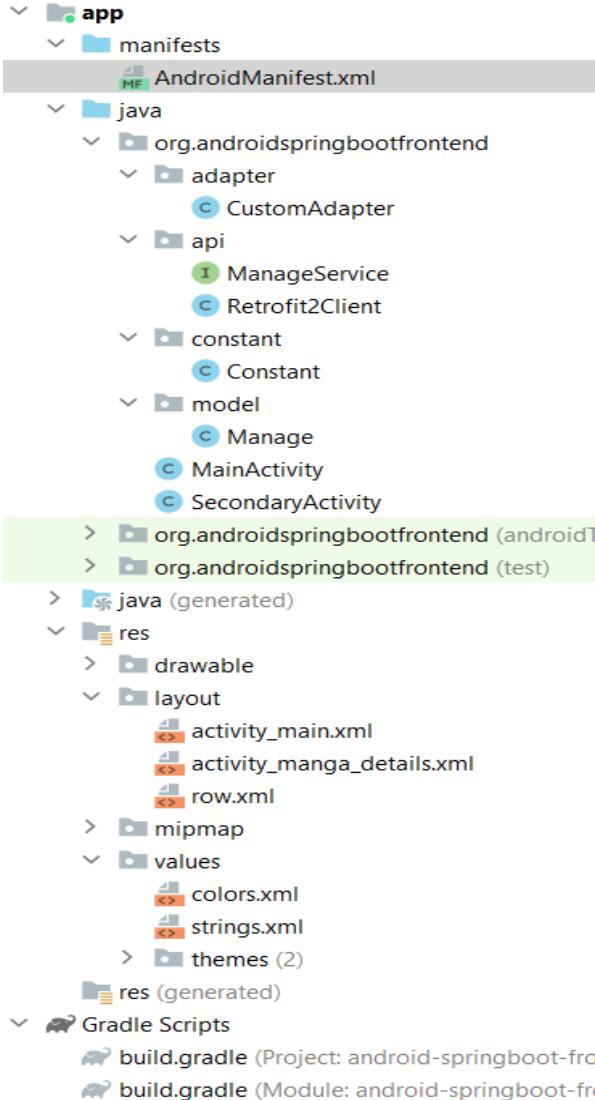


**Android** R.java là một tập tin được tạo tự động bởi aapt (Android Asset Packaging Tool) . Nó chứa tất cả các ID trong thư mục res /. Nó chứa nhiều class tĩnh : **menu, id, layout, attr, drawable, string .v.v**

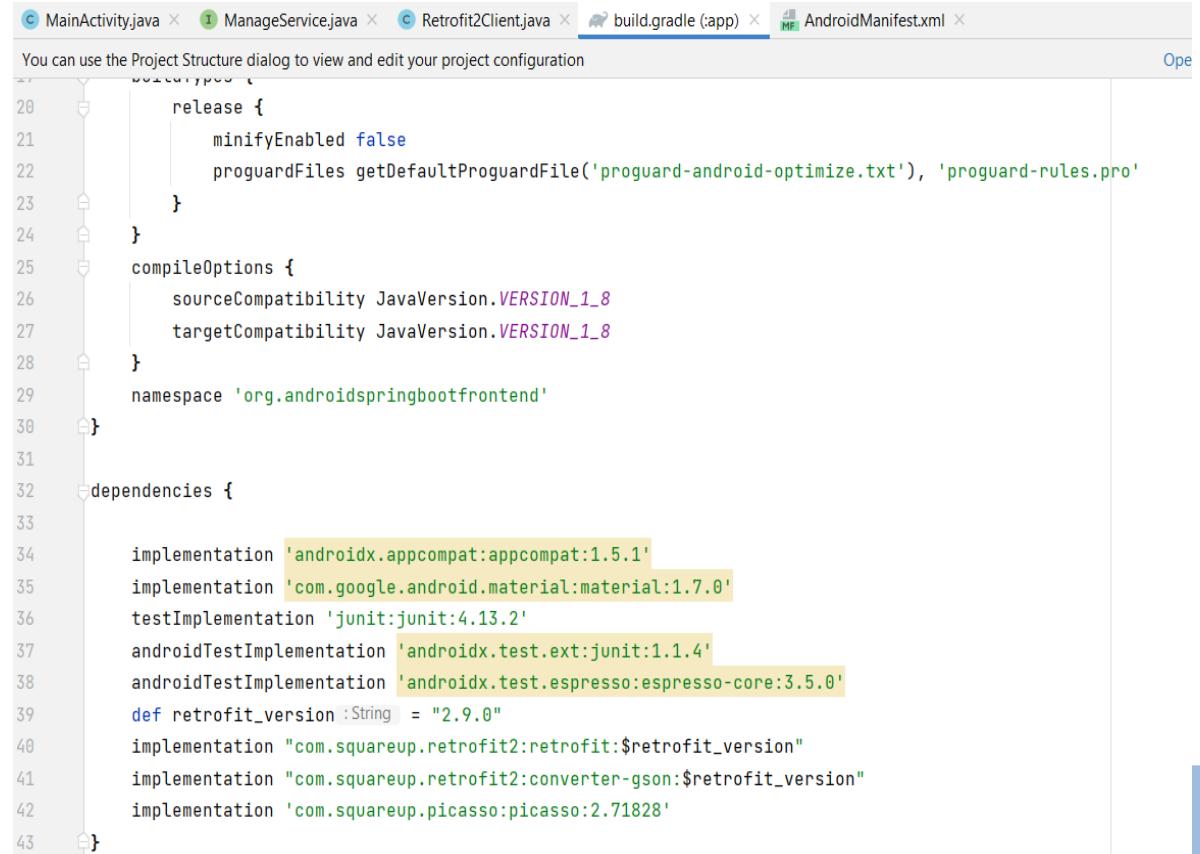
Truy xuất: **R.id, R.layout, R.menu,...**

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

```
Toolbar toolbar = findViewById(R.id.toolbar);
```

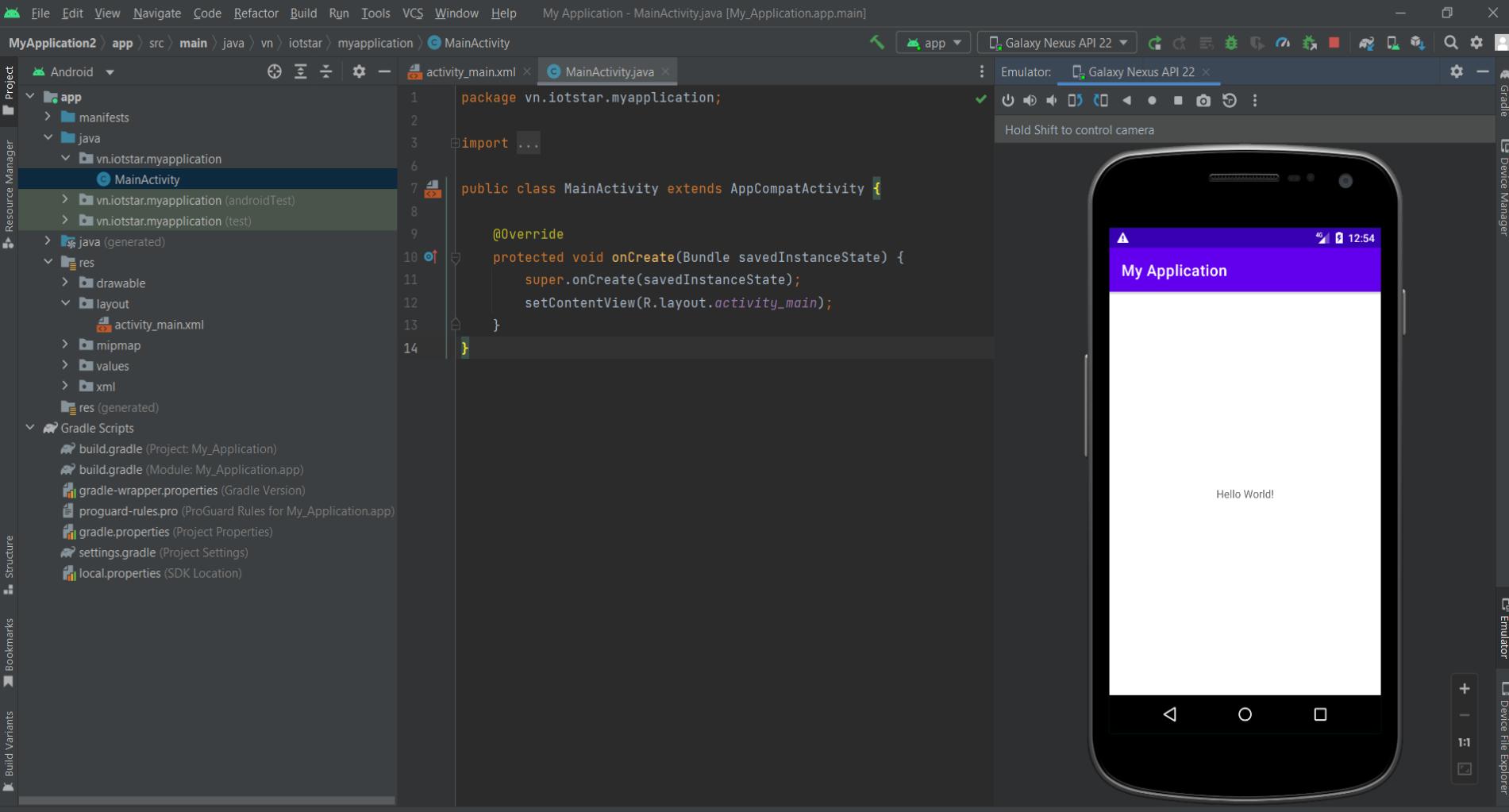


- **Build.gradle:** thông tin cấu hình project, các dependency của project.



```
build.gradle (app)
You can use the Project Structure dialog to view and edit your project configuration
release {
    minifyEnabled false
    proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
}
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
namespace 'org.androidspringbootfrontend'

dependencies {
    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'
    def retrofit_version :String = "2.9.0"
    implementation "com.squareup.retrofit2:retrofit:$retrofit_version"
    implementation "com.squareup.retrofit2:converter-gson:$retrofit_version"
    implementation 'com.squareup.picasso:picasso:2.71828'
}
```



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "MyApplication2". The `MainActivity.java` file is selected in the Project tree.
- Code Editor:** The `MainActivity.java` file contains the following Java code:

```
package vn.iotstar.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```
- Emulator:** An Android emulator for a "Galaxy Nexus API 22" device is running, displaying the application's UI.
- UI Preview:** The emulator screen shows a purple header bar with the text "My Application" and a white content area with the text "Hello World!".
- Toolbars and Panels:** The top bar includes standard Android Studio menu items like File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help. The bottom right corner has panels for Device Manager and Device File Explorer.

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

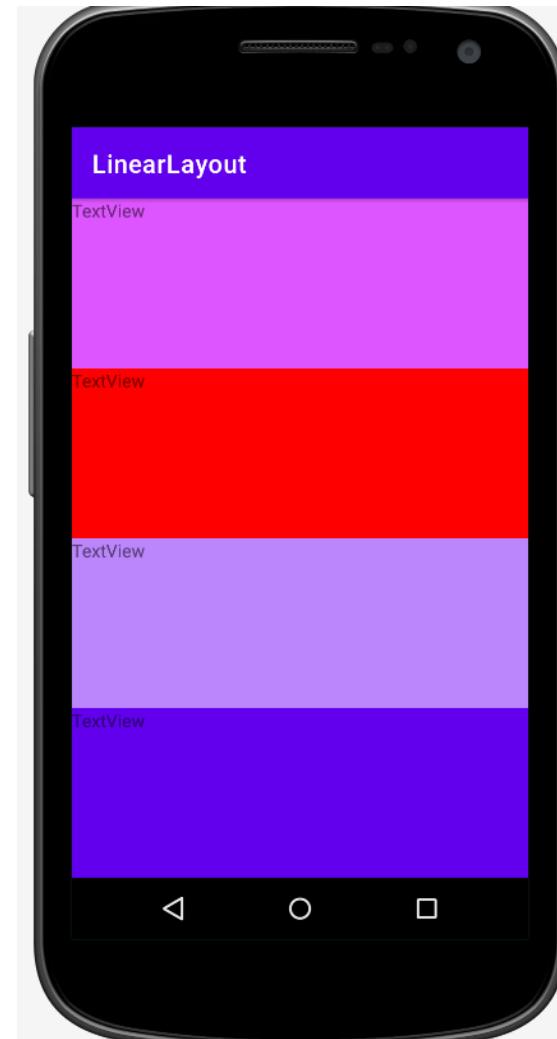
```
requestWindowFeature(Window.FEATURE_NO_TITLE); //will hide  
the title not the title bar
```

```
this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_F  
ULLSCREEN,
```

```
    WindowManager.LayoutParams.FLAG_FULLSCREEN); //int  
flag, int mask
```

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

```
}
```



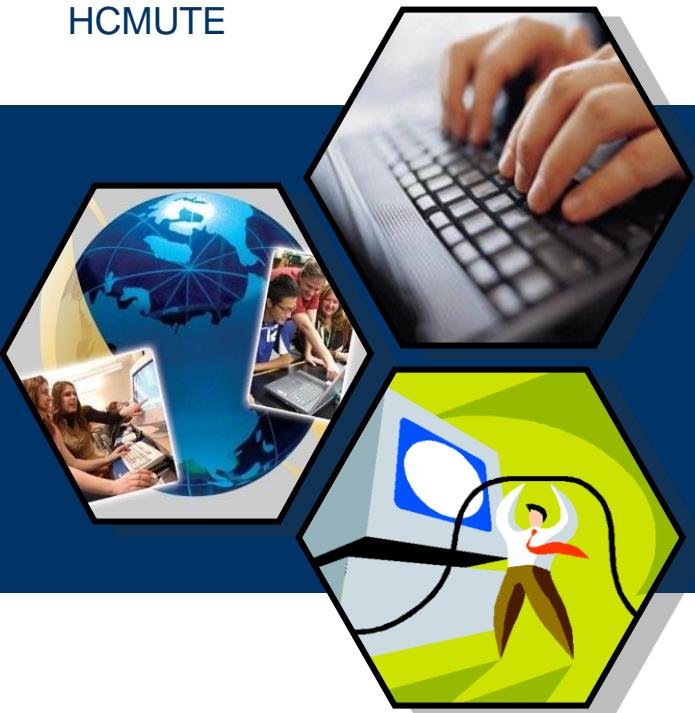


HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN

UTEx

## QUY TRÌNH THIẾT KẾ ỨNG DỤNG

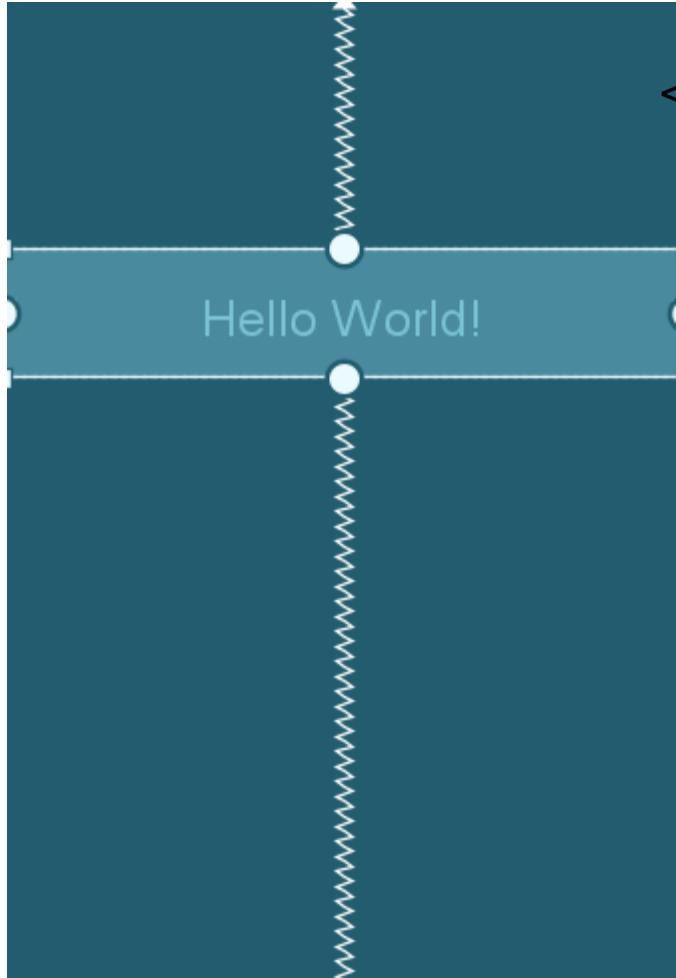


Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

ThS. Nguyễn Hữu Trung



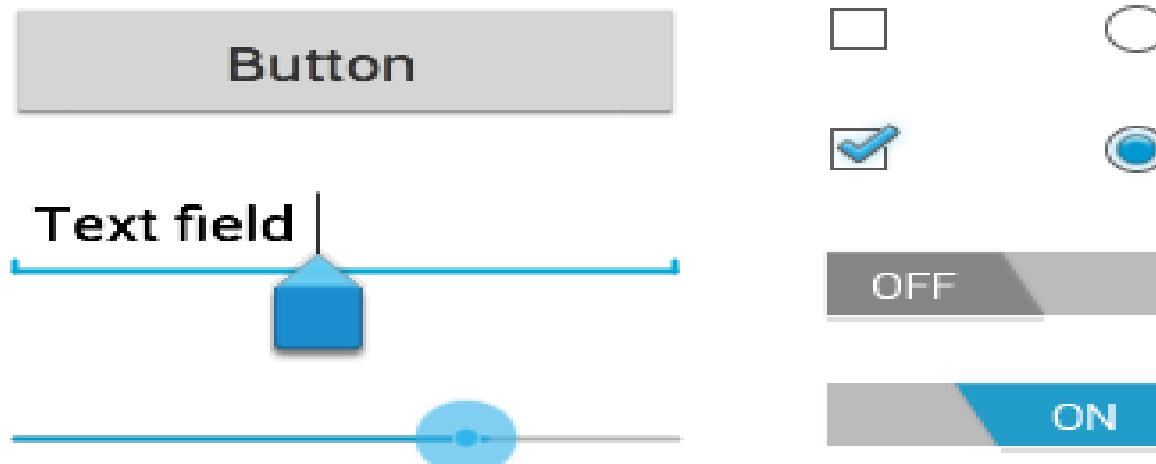
- **Bước 1 Kéo thả: thiết kế giao diện (Layout, Control)**
- **Bước 2 Ánh xạ: các view trên giao diện**
- **Bước 3 Viết code: điều khiển các view và bài toán.**



```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:padding="16dp"  
    android:text="Hello World!"  
    android:textAlignment="center"  
    android:textColor="#FF9800"  
    android:textSize="34sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.44"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.234" />
```

```
public class MainActivity extends AppCompatActivity {  
  
    TextView txtNoiDung1; //khai báo toàn cục  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //ánh xạ dựa vào Id  
        txtNoiDung1 = (TextView) findViewById(R.id.textView1);  
  
        //viết code điều khiển textView1  
        txtNoiDung1.setText("Chào bạn");  
    }  
}
```

- Input Control là các thành phần có tính tương tác trong giao diện UI của ứng dụng. Android cung cấp nhiều control đa dạng để bạn có thể sử dụng trong UI như button, text field, seek bar, checkbox, zoom button, toggle button, ...



- **TextView** được sử dụng để hiển thị text tới người dùng.

```
<TextView android:id="@+id/text_id"  
         android:layout_width="wrap_content"  
         android:layout_height="wrap_content"  
         android:text="Hello World" />
```

```
TextView myText = (TextView) findViewById(R.id.text_id);
```

- EditText Là một lớp con được định nghĩa trước của TextView mà bao gồm các khả năng chỉnh sửa đa dạng.

```
<EditText  
    android:id="@+id/textview"  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:inputType="text"  
    android:maxLines="1"  
    android:lines="1"/>
```

Thuộc tính **android:inputType** dùng để thiết lập kiểu nhập liệu, hiện thị

- **AutoCompleteTextView** trong Android hỗ trợ cho người dùng những gợi ý liên quan khi bạn nhập vào trường EditText.

```
<AutoCompleteTextView  
    android:id="@+id/autoCompleteTextView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentLeft="true"  
    android:layout_below="@+id/textView1"  
    android:layout_marginLeft="36dp"  
    android:layout_marginTop="17dp"  
    android:ems="10"  
    android:text="">  
    <requestFocus />  
</AutoCompleteTextView>
```

```
java.lang.Object  
↳ android.view.View  
    ↳ android.widget.TextView  
        ↳ android.widget.EditText  
            ↳ android.widget.AutoCompleteTextView
```

&lt;Button

```
    android:id="@+id	btnThucHien"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="180dp"
    android:text="Thực hiện"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView1"
    app:layout_constraintVertical_bias="0.353" />
```

//viết code điều khiển btnClick

```
btnClick.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //viết code điều khiển textView1
        txtNoiDung1.setText("Chào bạn");
    }
});
```

**ĐĂNG NHẬP**

&lt;EditText

```
    android:id="@+id/editTextPassword"
    android:layout_width="345dp"
    android:layout_height="51dp"
    android:layout_marginTop="12dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextName" />
```

&lt;EditText

```
    android:id="@+id/editTextName"
    android:layout_width="345dp"
    android:layout_height="51dp"
    android:layout_marginTop="32dp"
    android:ems="10"
    android:hint="UserName"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"/>
```

UserName

Password

ĐĂNG NHẬP

```
<Button  
    android:id="@+id/btnDangNhap"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="Đăng nhập"  
    android:textColor="#fff"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.86"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf=  
    "@+id/editTextPassword" />
```

```
btnDangnhap =(Button) findViewById(R.id.btnDangNhaph);  
editTextname = (EditText) findViewById(R.id.editTextName);  
editTextpass = (EditText) findViewById(R.id.editTextPassword);
```

```
btnDangnhap.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        //viết code  
        if(editTextname.getText().toString().equals("admin")  
            && editTextpass.getText().toString().equals("admin")){  
            txtNoiDung1.setText("Đăng nhập thành công!");  
        }else{  
            txtNoiDung1.setText("Đăng nhập thất bại" +  
editTextpass.getText().toString());  
        }  
    }  
});
```

My Application

8

SINH SỐ NGẪU NHIÊN

```
<TextView  
    android:id="@+id/textViewSo"  
    android:layout_width="match_parent"  
    android:layout_height="40dp"  
    android:textAlignment="gravity"  
    android:gravity="center"  
  
    android:text="" />
```

```
<Button  
    android:id="@+id/buttonRnd"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="92dp"  
    android:text="Sinh số ngẫu nhiên"
```

## My Application

8

SINH SỐ NGẪU NHIÊN

```
//ánh xạ
txtSoN = (TextView)
findViewById(R.id.textViewSo);
btnRnd = (Button)
findViewById(R.id.buttonRnd);

//viết code sinh ngẫu nhiên
btnRnd.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //tạo số ngẫu nhiên
        Random random = new Random();
        int number = random.nextInt(10);
        txtSoN.setText(number + ""); //number +
        ép kiểu
    }
});
```

My Application

Nhập số thứ nhất

Nhập số thứ hai

SINH SỐ NGẪU NHIÊN

My Application

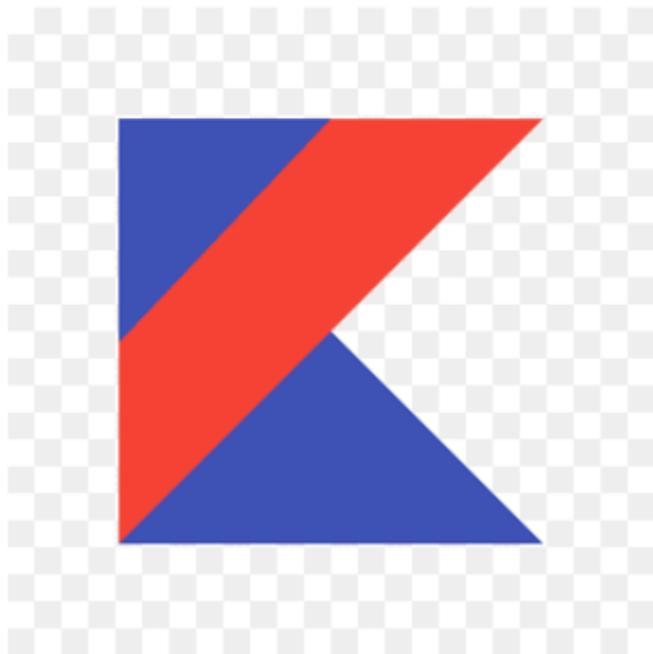
5

3

7

SINH SỐ NGẪU NHIÊN

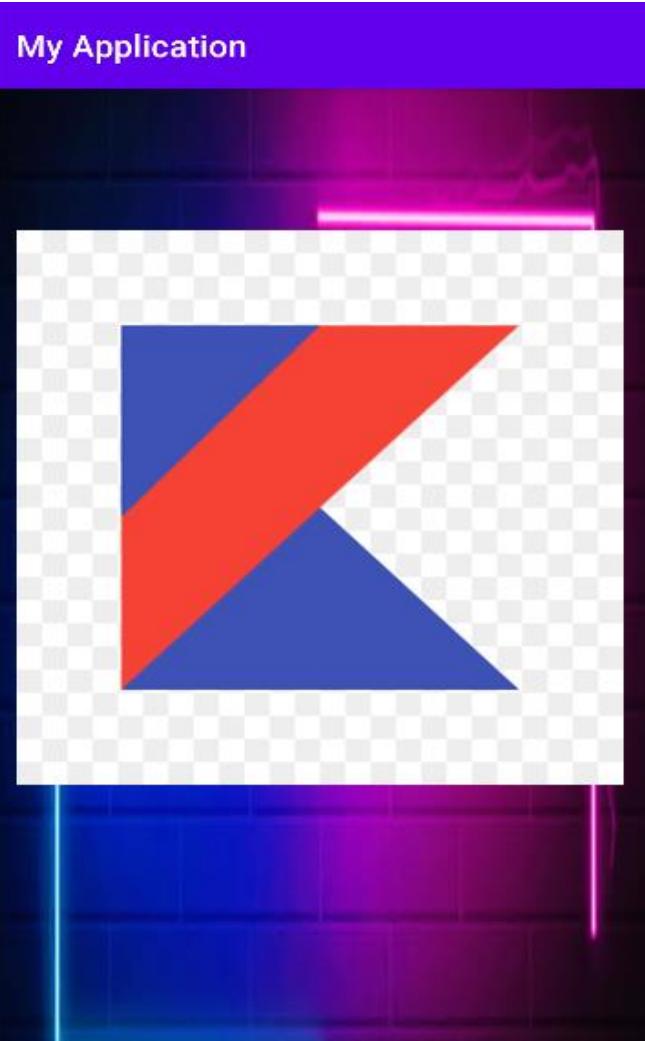
My Application



```
<ImageView  
    android:scaleType="center"  
    android:id="@+id/imageView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="92dp"  
    android:src="@drawable/kotlin"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
ImageView img1= (ImageView)  
findViewById(R.id.imageView1);  
  
img1.setImageResource(R.drawable.kotlin);
```

My Application

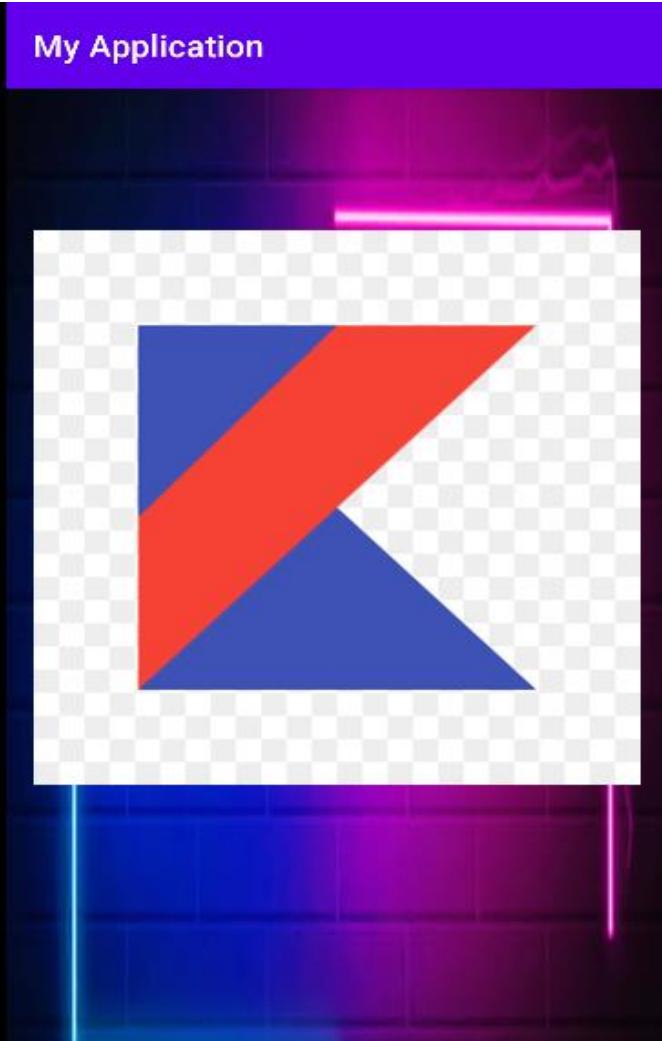


```
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/  
    android"  
        xmlns:app="http://schemas.android.com/apk/res-  
        auto"  
        xmlns:tools="http://schemas.android.com/tools"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:background="@drawable/bg1"  
        android:id="@+id/constraintLayout1"  
    >
```

```
ConstraintLayout bg = (ConstraintLayout)  
findViewById(R.id.constraintLayout1);
```

```
bg.setBackgroundColor(Color.BLUE);
```

```
bg.setBackgroundResource(R.drawable.bg2);
```



## Random background

```
ArrayList<Integer> arrayList = new ArrayList<>();  
arrayList.add(R.drawable.bg1);  
arrayList.add(R.drawable.bg2);  
arrayList.add(R.drawable.bg3);  
arrayList.add(R.drawable.bg4);
```

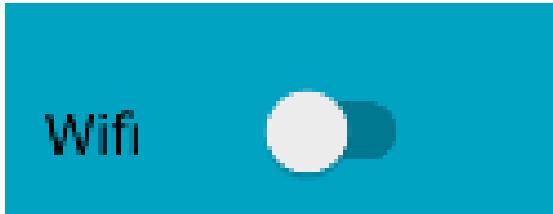
```
Random random = new Random();  
int vitri = random.nextInt(arrayList.size());
```

```
bg.setBackgroundResource(arrayList.get(vitri));
```



```
<ImageButton  
    android:id="@+id/imageButton1"  
    android:layout_width="80dp"  
    android:layout_height="80dp"  
    android:background="@drawable/on"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.45"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.732" />
```

```
ImageButton img2 = (ImageButton)  
findViewById(R.id.imageButton1);  
img2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        mg1.setImageResource(R.drawable.dart);  
        img1.getLayoutParams().width=550;  
        img1.getLayoutParams().height=550;  
    }  
});
```



```
<Switch  
    android:id="@+id/switch1"  
    android:layout_width="97dp"  
    android:layout_height="38dp"  
    android:text="Wifi"/>
```

```
//switch  
Switch sw = (Switch) findViewById(R.id.switch1);  
sw.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        if(isChecked){ //isChecked = true  
            Toast.makeText(ControlActivity.this,"Wifi đang bật",Toast.LENGTH_LONG).show()  
        }else{  
            Toast.makeText(ControlActivity.this,"Wifi đang tắt",Toast.LENGTH_LONG).show();  
        }  
    }  
});
```



```
<CheckBox  
    android:id="@+id/checkBox"  
    android:layout_width="151dp"  
    android:layout_height="66dp"  
    android:text="Background 1"
```

```
CheckBox ck1 = (CheckBox) findViewById(R.id.checkBox2);  
ck1.setOnCheckedChangeListener(new  
CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)  
{  
    if(isChecked){  
        bg.setBackgroundResource(R.drawable.bg3);  
    }else{  
        bg.setBackgroundResource(R.drawable.bg4);  
    }  
}  
});
```

# CheckBox

61



```
//RadioGroup
RadioGroup radioGroup = (RadioGroup)
findViewById(R.id.radioGroup1);
radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup group,
    int checkedId) {
        //checkID trả về ID radio
        switch (checkedId){
            case R.id.radioButton:
                bg.setBackgroundResource(R.drawable.bg3);
                break;
            case R.id.radioButton2:
                bg.setBackgroundResource(R.drawable.bg4);
                break;
        }
    }
});
```

```
<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" >
    <RadioButton
        android:id="@+id radioButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Background 1" />
    <RadioButton
        android:id="@+id radioButton2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Background 2" />
</RadioGroup>
```



```
<ProgressBar  
    android:id="@+id/progressBar2"  
    android:progress="40"  
    android:max="100"  
    android:secondaryProgress="60"  
    style="@android:style/Widget.ProgressBar.Horizontal"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"/>
```

```
//progressbar  
ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar2);  
img2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        int current = progressBar.getProgress();  
        progressBar.setProgress(current + 10);  
    }  
});
```

# ProgressBar

63



```
<ProgressBar
    android:id="@+id/progressBar2"
    android:progress="40"
    android:max="100"
    android:secondaryProgress="60"
    style="@android:style/Widget.ProgressBar.Horizontal"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
```

//progrebar

```
ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar2);
img2.setOnClickListener(new View.OnClickListener() {
```

@Override

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

```
    int current = progressBar.getProgress();
```

//chạy đều đều >100 quay về 0

```
    if (current>= progressBar.getMax()) {
```

```
        current=0;
```

```
}
```

```
    progressBar.setProgress(current + 10); //thiết lập progress
```

```
}
```

```
};
```

```
//progressbar
ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar2);
img2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //tự đếm progress
        CountDownTimer countDownTimer = new CountDownTimer(10000,1000) {
            @Override
            public void onTick(long millisUntilFinished) {
                int current = progressBar.getProgress();
                //chạy đều đều >100 quay về 0
                if (current>= progressBar.getMax()){
                    current=0;
                }
                progressBar.setProgress(current + 10); //thiết lập progress
            }
            @Override
            public void onFinish() {
                Toast.makeText(ControlActivity.this,"Hết giờ",Toast.LENGTH_LONG).show();
            }
        };
        countDownTimer.start();
    }
});
```



```
<SeekBar  
    android:id="@+id/seekBar"  
    style="@android:style/Widget.DeviceDefault.Light.SeekBar"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:max="100"  
    android:progress="40"  
    android:secondaryProgress="70"  
    android:thumb="@android:mipmap/sym_def_app_icon"
```

//SeekBar

```
SeekBar seekBar = (SeekBar) findViewById(R.id.seekBar);  
seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
    @Override  
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
        //progress: giá trị của seekbar  
        Log.d("AAA","Giá trị:" + progress);  
    }  
    @Override  
    public void onStartTrackingTouch(SeekBar seekBar) {  
        Log.d("AAA","Start");  
    }  
    @Override  
    public void onStopTrackingTouch(SeekBar seekBar) {  
        Log.d("AAA","Stop");  
    }});
```

- Bước 1: Vẽ button lên ViewGroup
- Bước 2: Tạo Drawble Resource File

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
    android:shape="rectangle">  
    <solid android:color="#ff00" />  
    <size android:height="550dp"  
        android:width="200dp"/>  
    <corners android:radius="30dp" />  
    <gradient android:centerColor="@color/purple_200"/>  
</shape>
```

- Bước 3: Gọi Drawble

```
    android:background="@drawable/button_custom"
```

- Bước 1: Vẽ button lên ViewGroup
- Bước 2: Tạo Drawble Resource File

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
    android:shape="rectangle">  
    <solid android:color="#ff00" />  
    <size android:height="550dp"  
        android:width="200dp"/>  
    <corners android:radius="30dp" />  
    <gradient android:centerColor="@color/purple_200"/>  
</shape>
```

- Bước 3: Gọi Drawble

```
    android:background="@drawable/button_custom"
```

- Bước 1: Tạo thư mục menu trong Resources
- Bước 2: Tạo Menu Resource File

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="Setting"/>
    <item android:title="Share"/>
    <item android:title="Logout"/>
</menu>
```

- Bước 3: Gọi menu

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_setting,menu);
    return super.onCreateOptionsMenu(menu);
}
```

## □ Bước 1: Bắt sự kiện menu

```
<?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:title="Setting"
        android:id="@+id/menuSetting"
        android:icon="@android:drawable/ic_menu_set_as"
        app:showAsAction="always"/>
    <item android:title="Share"
        android:orderInCategory="1"
        android:id="@+id/menuShare"/>
    <item android:title="Logout"
        android:orderInCategory="2"
        android:id="@+id/menuLogout"/>
    <item android:title="Menu con"
        android:orderInCategory="3">
        <menu>
            <item android:title="Menu con 1"/>
            <item android:title="Menu con 2"/>
        </menu>
    </item>
</menu>
```

## □ Bước1: Bắt sự kiện menu

@Override

```
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.menuSetting:  
            //lệnh  
            break;  
        case R.id.menuShare:  
            break;  
        case R.id.menuLogout:  
            break;  
    }  
  
    return super.onOptionsItemSelected(item);  
}
```

## □ Bước1: Gọi Popup menu

//popup menu

```
private void ShowPopupMenu(){  
    PopupMenu popupMenu = new PopupMenu(this,btnButton);  
  
    popupMenu.getMenuInflater().inflate(R.menu.menu_setting,popupMenu.getMen  
    u());  
    popupMenu.show();  
}
```

## □ Bước 2: Bắt sự kiện Popup menu

```
//popup menu
private void ShowPopupMenu(){
    PopupMenu popupMenu = new PopupMenu(this,btnButton);
    popupMenu.getMenuInflater().inflate(R.menu.menu_setting,popupMenu.getMenu());
    //bắt sự kiện
    popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item) {
            switch (item.getItemId()){
                case R.id.menuSetting:
                    //lệnh
                    Toast.makeText(ControlActivity2.this,"Bạn đang chọn Setting",Toast.LENGTH_LONG).show();
                    break;
                case R.id.menuShare:
                    btnButton.setText("Chia sẻ");
                    break;
                case R.id.menuLogout:
                    break;
            }
            return false;
        }
    });
    popupMenu.show();
}
```

## □ Bước 1: Gọi Context menu

```
@Override  
public void onCreateContextMenu(ContextMenu menu, View v,  
ContextMenu.ContextMenuItemInfo menuInfo) {  
    getMenuInflater().inflate(R.menu.menu_setting,menu);  
    menu.setHeaderTitle("Context Menu");  
    menu.setHeaderIcon(R.mipmap.ic_launcher);  
    super.onCreateContextMenu(menu, v, menuInfo);  
}
```

//đăng ký view cho context menu trong onCreate  
registerForContextMenu(btnButton);

## □ Bước 2: bắt sự kiện Context menu

//bắt sự kiện Context Menu

@Override

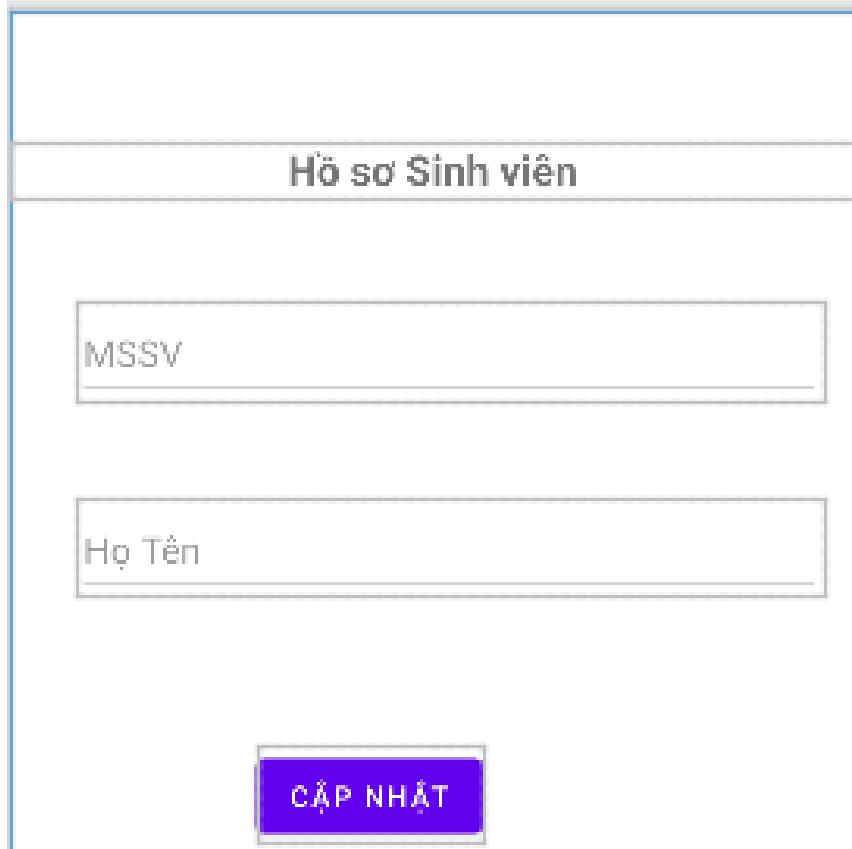
```
public boolean onContextItemSelected(@NonNull MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.menuSetting:  
            //lệnh  
            Toast.makeText(ControlActivity2.this, "Bạn đang chọn  
Setting", Toast.LENGTH_LONG).show();  
            break;  
        case R.id.menuShare:  
            btnButton.setText("Chia sẻ");  
            break;  
        case R.id.menuLogout:  
            break;  
    }  
    return super.onContextItemSelected(item);  
}
```

# Alert Dialog

75

```
private void XacNhanXoa( final int vitri){  
    AlertDialog.Builder alert = new AlertDialog.Builder(this);  
    alert.setTitle("Thông báo");  
    alert.setMessage("Bạn có muốn đăng xuất không");  
    alert.setPositiveButton("Có", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialog, int which) {  
            //lệnh nút có  
        }  
    });  
    alert.setNegativeButton("Không", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialog, int which) {  
            //lệnh nút không  
        }  
    });  
    alert.show();  
}
```

## □ Bước 1: Tạo layout DiaLog



```
//tạo hàm Dialog
private void DiaLog1(){
    Dialog dialog = new Dialog(this);

    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
    dialog.setContentView(R.layout.dialog);

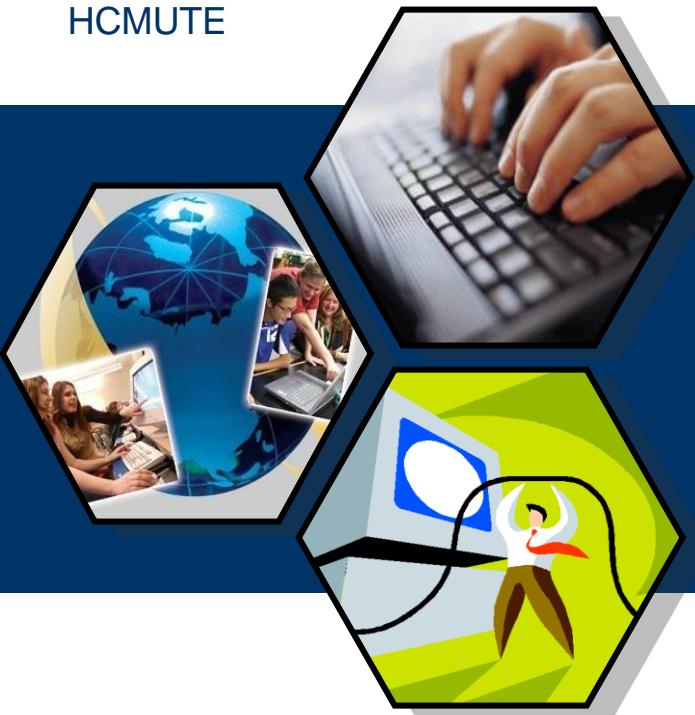
    dialog.setCanceledOnTouchOutside(false);
    //Ánh xạ
    EditText editText1 = (EditText)
    dialog.findViewById(R.id.editNumber1);
    //viết code sự kiện
    //bắt sự kiện Dialog
    dialog.show(); //hủy gọi dialog.dismiss();
}
```



HCMUTE

KHOA CÔNG NGHỆ THÔNG TIN

UTEx



## LAYOUT TRONG ANDROID

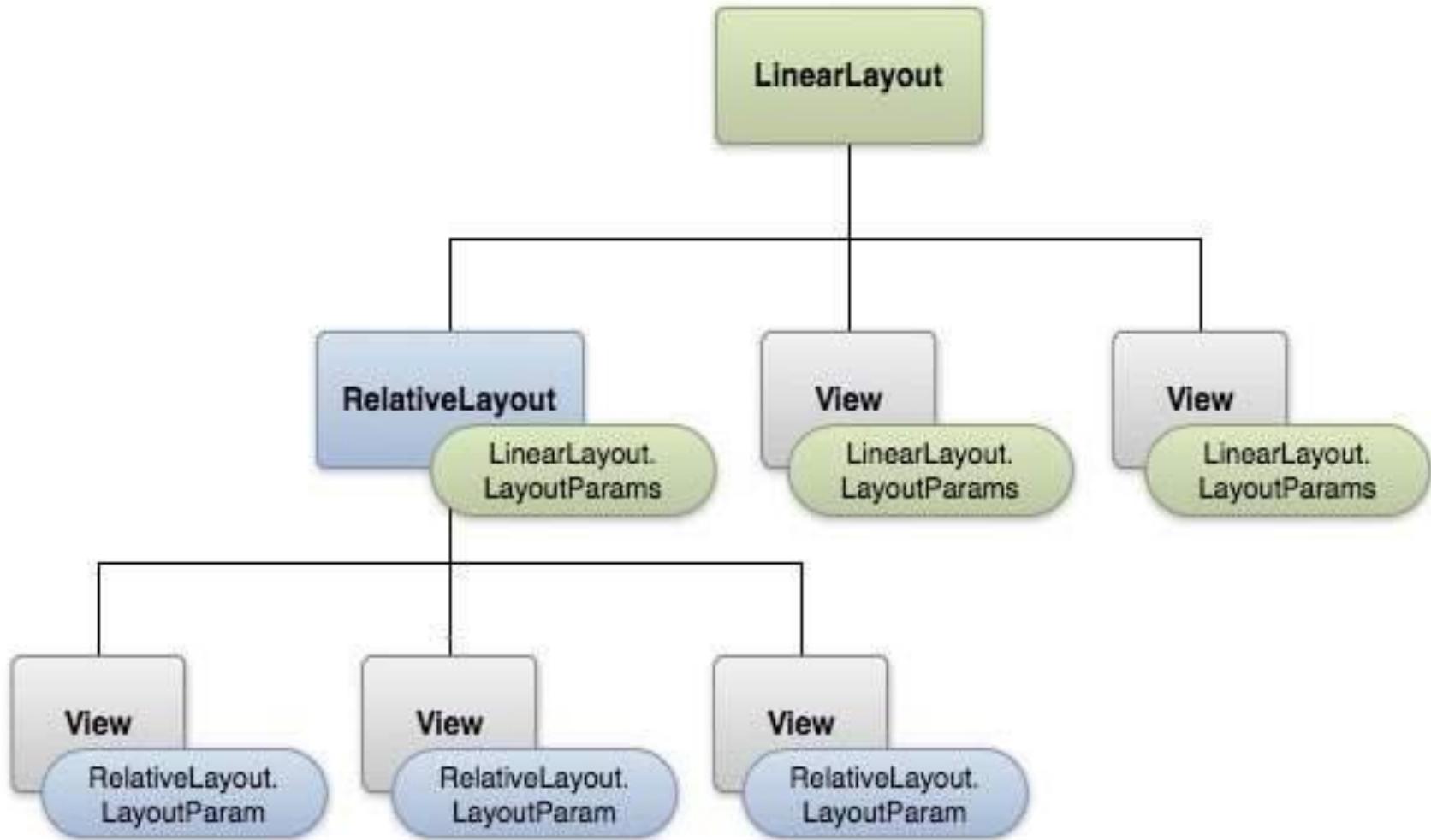
Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

ThS. Nguyễn Hữu Trung



- Kiến trúc nền tảng cho giao diện UI là một đối tượng View, view là lớp cơ sở cho Widget để tạo các thành phần UI có tính tương tác như button, các trường text, ...
- **ViewGroup** là một lớp cung cấp Container giữ các View khác hoặc các ViewGroup khác và định nghĩa các thuộc tính Layout của chúng.

- ❑ Layout đặc trưng định nghĩa cấu trúc nhìn thấy cho một giao diện UI trong Android và có thể được tạo bởi sử dụng các đối tượng **View**/**ViewGroup** hoặc bạn có thể khai báo Layout của bạn bởi sử dụng XML file đơn giản là **main\_Layout.xml**, được đặt trong thư mục res/layout của Project.



Layout	Miêu tả
Linear Layout	Linear Layout là một view group mà căn chỉnh các view con theo một hướng nào đó: <b>chiều dọc hay chiều ngang</b> . Thuộc tính: <b>android:orientation</b>
Relative Layout	Là Layout hiển thị các View con với các vị trí tương đối. Vị trí của mỗi View có thể được xác định so với các View khác hoặc với thành phần cha của chúng ( thông qua id). Bạn có thể sắp xếp View sang bên phải, bên dưới một View khác, giữa màn hình, v.v.. Để định nghĩa vị trí cho mỗi View bạn sử dụng nhiều thuộc tính có sẵn từ <b>RelativeLayout.LayoutParams</b> .
Table Layout	Table Layout là một view mà nhóm tất cả các view vào <b>trong các hàng và các cột</b> .

**Layout****Công dụng****Absolute  
Layout**

Là layout cho phép bạn chỉ định vị trí chính xác ( tọa độ x / y) của các đối tượng. Absolute layouts thường ít linh hoạt và khó duy trì hơn các loại layouts khác.

**Frame  
Layout**

FrameLayout là một trong những layout hữu ích được cung cấp bởi hệ thống Android, cho phép các đối tượng giao diện người dùng được xếp chồng chéo với nhau.

Layout	Công dụng
	<p>Constraint layout được giới thiệu lần đầu tiên tại sự kiện <b>Google I/O 2016</b>, Constraint Layout sẵn dùng với bản Android 2.3 (API level 9)</p> <p><b>match_constraint</b>: xác định width và height của một View</p>
Constraint layout	<p>VD : TextView dưới được set nằm giữa view parent</p> <pre>&lt;TextView android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:text="Text View 1"          app:layout_constraintBottom_toBottomOf="parent"         app:layout_constraintLeft_toLeftOf="parent"         app:layout_constraintRight_toRightOf="parent"         app:layout_constraintTop_toTopOf="parent" /&gt;</pre>

# Các thuộc tính của Layout

84

Thuộc tính	Ý nghĩa
<b>android:id</b> <b>android:id="@+id/mybutton"</b>	ID nhận diện duy nhất View, Button myButton = (Button) findViewById(R.id.mybutton);
<b>android:layout_width</b>	Độ rộng của Layout, dp (Density-independent pixel), sp (Scale-independent pixel), pt (các point là 1/72 inch), px (pixel), mm (mili met), và in (inch).
<b>android:layout_height</b>	Chiều cao của Layout, <b>wrap_content</b> theo nội dung, <b>fill_parent/match_parent</b> theo view cha.
<b>android:layout_marginTop</b>	Lề cạnh trên của Layout
<b>android:layout_marginBottom</b>	Lề cạnh dưới của Layout
<b>android:layout_marginLeft</b>	Lề cạnh trái của Layout
<b>android:layout_marginRight</b>	Lề cạnh phải Layout
<b>android:layout_gravity</b>	Xác định cách các view con được đặt tại đâu

Thuộc tính	Ý nghĩa
<b>android:layout_weight</b>	Xác định có bao nhiêu phần trong Layout được cấp phát tới View đó
<b>android:layout_x</b>	Xác định tọa độ x của Layout
<b>android:layout_y</b>	Xác định tọa độ y của Layout
<b>android:paddingLeft</b>	Left padding được điền cho Layout
<b>android:paddingRight</b>	Right padding được điền cho Layout
<b>android:paddingTop</b>	Top padding được điền cho Layout
<b>android:paddingBottom</b>	Bottom padding được điền cho Layout

# Ví dụ:

86

```

        android:layout_marginLeft="40dp"
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="40dp"
    android:layout_marginTop="30dp"
    android:background="#72acdf"
    android:paddingBottom="50dp"
    android:paddingLeft="50dp"
    android:paddingRight="50dp"
    android:paddingTop="50dp"
    tools:context="dev4u.com.helloworld.MainActivity">

    <Button
        android:id="@+id	btnMyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button Demo" />

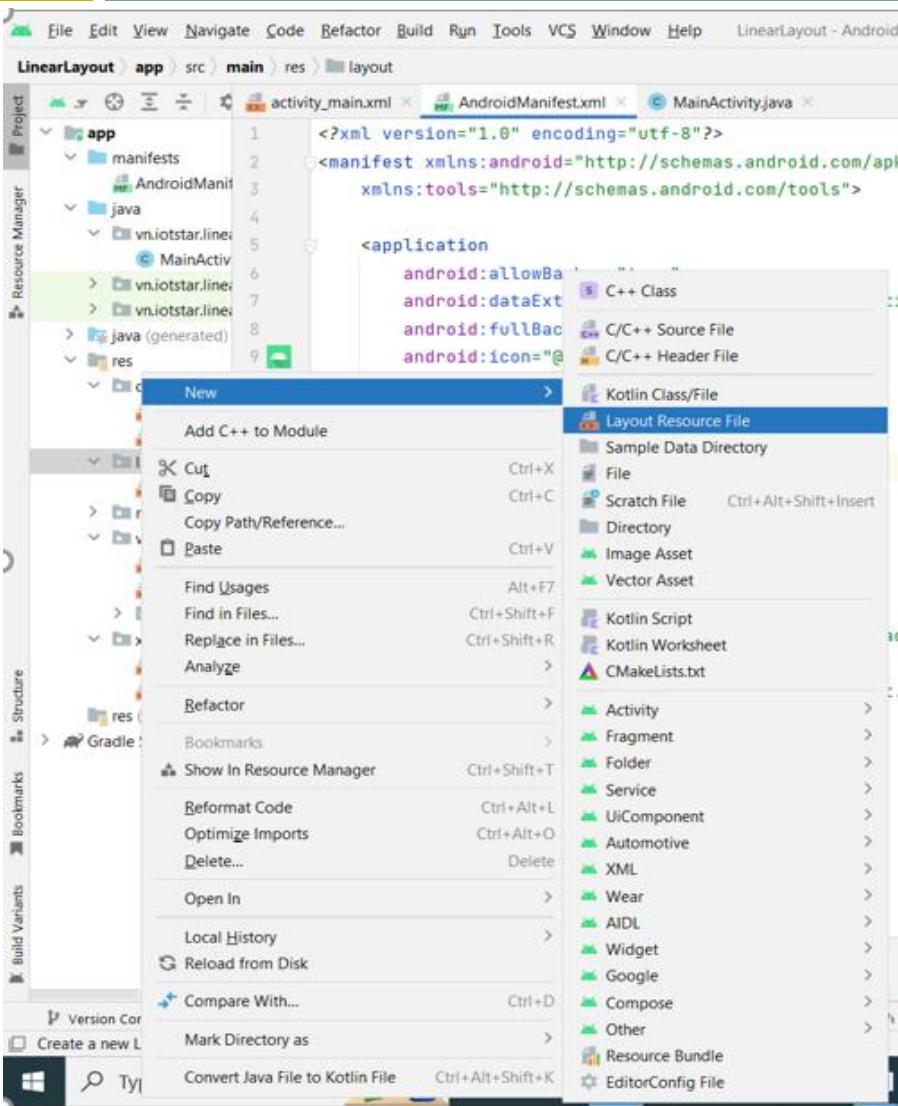
    <!--<TextView-->
    <!--android:layout_width="wrap_content"-->
    <!--android:layout_height="wrap_content"-->
```





```
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7      <uses-sdk
8          android:minSdkVersion="8"
9          android:targetSdkVersion="21" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:label="@string/app_name"
15         android:theme="@style/AppTheme" >
16         <activity
17             android:name=".MainActivity"
18             android:label="@string/app_name" >
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25     </application>
26
27 </manifest>
```

# Đổi Layout mặc định bằng một Layout khác bất kỳ



Đặt tên my\_new\_layout.xml

Bây giờ vào lại MainActivity.java. Sửa lại dòng lệnh setContentView thành:

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

89

```
✓ LayoutDemo
  ✓ src
    ✓ com.example.layoutdemo
      > MainActivity.java
  > gen [Generated Java Files]
  > Android 7.1.1
  > Android Private Libraries
  > Android Dependencies
  ✓ assets
  ✓ bin
  ✓ libs
  ✓ res
    ✓ drawable-hdpi
    ✓ drawable-ldpi
    ✓ drawable-mdpi
    ✓ drawable-xhdpi
    ✓ drawable-xxhdpi
  ✓ layout
    activity_main.xml
  ✓ menu
  ✓ values
  ✓ values-v11
  ✓ values-v14
  ✓ values-w820dp
  AndroidManifest.xml
  ic_launcher-web.png
  proguard-project.txt
  project.properties
```

```
5 import android.view.Window;
6 import android.view.MenuItem;
7
8 public class MainActivity extends Activity {
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14    }
15
16    @Override
17    public boolean onCreateOptionsMenu(Menu menu) {
18        // Inflate the menu; this adds items to the action bar if it is present.
19        getMenuInflater().inflate(R.menu.main, menu);
20        return true;
21    }
22
23    @Override
24    public boolean onOptionsItemSelected(MenuItem item) {
25        // Handle action bar item clicks here. The action bar will
26        // automatically handle clicks on the Home/Up button, so long
27        // as you specify a parent activity in AndroidManifest.xml.
28        int id = item.getItemId();
29        if (id == R.id.action_settings) {
30            return true;
31        }
32        return super.onOptionsItemSelected(item);
33    }
34}
```

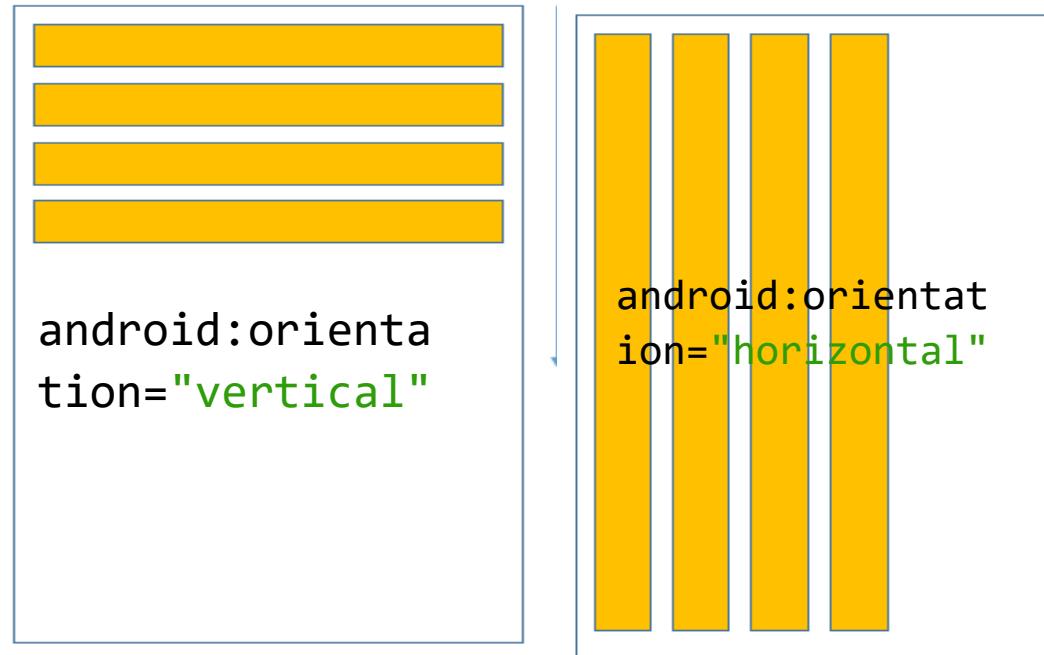
FrameLayout là một ViewGroup được sử dụng rất nhiều trong android. Bởi vì nó là ViewGroup đơn giản nhất,

FrameLayout được định nghĩa bắt đầu bởi thẻ <FrameLayout> và thẻ đóng </FrameLayout>. Ở giữa thẻ đóng và thẻ mở chính là các view con của nó.

Quy tắc layout các view con trong FrameLayout là các view sẽ nắn chồng lên nhau, view thêm vào sau sẽ nằm đè lên view nằm phía dưới

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textColor="#2c3e50"
        android:textSize="32sp" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher_background" />
</FrameLayout>
```

- LinearLayout là một view group việc căn chỉnh các view con theo một hướng nào đó: chiều dọc hay chiều ngang.
- Khi Layout được tạo, bạn có thể tải layout resource từ phần code, trong phương thức callback là **Activity.onCreate()**, như sau:



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

# Linear Layout

92

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

```
<TextView android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a TextView" />
```

```
<Button android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a Button" />
```

```
<!-- More GUI components go here -->
```

```
</LinearLayout>
```

```
<TextView  
    android:background="#ff0"  
    android:fontFamily="sans-serif"  
    android:textColor="#FF5722"  
    android:textSize="30sp"  
    android:id="@+id/textView5"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World"/>
```

```
<TextView  
    android:layout_width="0dp"  
    android:layout_height="match_parent"  
    android:layout_weight="1"  
    android:background="#af5656" />
```

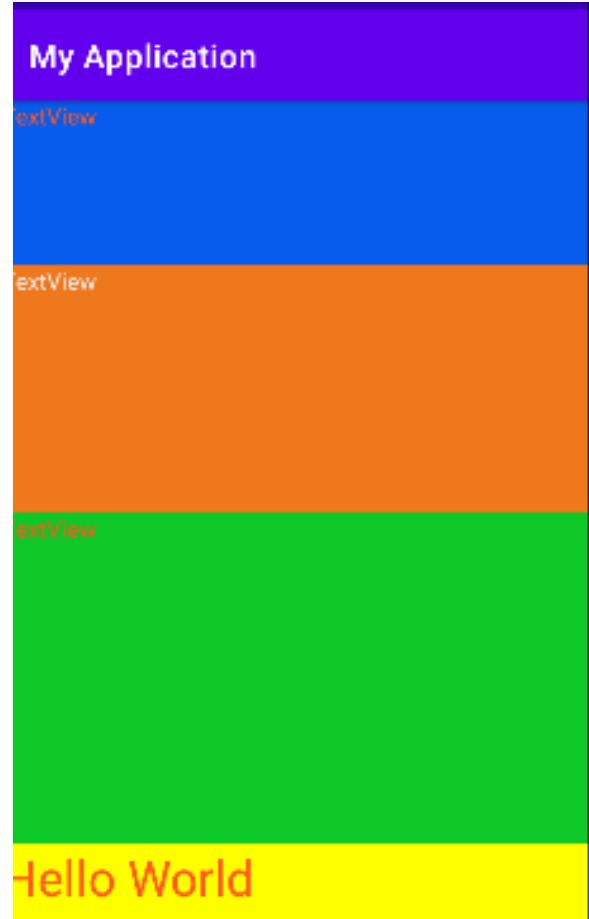
```
<TextView  
    android:layout_width="0dp"  
    android:layout_height="match_parent"  
    android:layout_weight="4"  
    android:background="#28c6a6" />
```

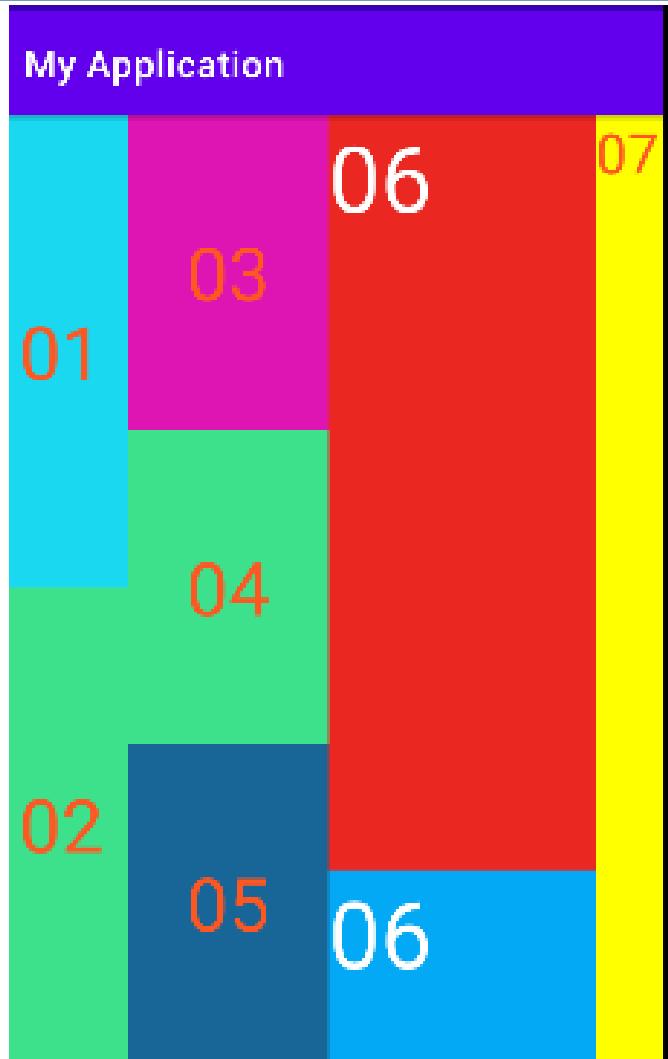
## Chia tỉ lệ layout

Ngoài cách sử dụng **LinearLayout** thông thường thì **LinearLayout** thường được sử dụng để phân chia tỉ lệ layout bằng cách sử dụng thuộc tính **layout\_weight** và **weightSum**.

- **weightSum**: Xác định trọng số của **LinearLayout** hiện tại.
- **layout\_weight**: Trọng số mà view con trong **LinearLayout** chiếm giữ.

```
<TextView  
    android:background="#ff0"  
    android:fontFamily="sans-serif"  
    android:textColor="#FF5722"  
    android:textSize="30sp"  
    android:id="@+id/textView5"  
    android:layout_width="match_parent"  
    android:layout_height="100dp"  
    android:text="Hello World"  
    android:layout_weight="1" //chia trọng số, tỉ lệ  
/>  
<LinearLayout  
    android:weightSum="10" /> <!-- 2 - 3 - 4 - 1 -->
```





# Relative Layout

96

- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối. Do đó khi làm RelativeLayout bạn phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id bạn mới đổi).
- RelativeLayout là ViewGroup cũng được sử dụng khá nhiều trong android, RelativeLayout được định nghĩa trong xml bởi cặp thẻ đóng mở <RelativeLayout> và thẻ đóng </RelativeLayout>.

# Relative Layout

Quy tắc layout của RelativeLayout khá giống như FrameLayout, nhưng có một số điểm đặc biệt là các view có thể xác định bằng vị trí **tương đối (relative)** với các view khác thông qua id.

Các vị trí tương đối này như sau:

- **android:layout\_above="id\_name"**: view hiện tại sẽ nằm phía trên view có thuộc tính id là id\_name.
- **android:layout\_below="id\_name"**: view hiện tại sẽ nằm phía dưới view có thuộc tính id là id\_name.
- **android:layout\_toLeftOf="id\_name"**: view hiện tại sẽ nằm bên trái dưới view có thuộc tính id là id\_name.
- **android:layout\_toRightOf="id\_name"**: view hiện tại sẽ nằm phía bên phải view có thuộc tính id là id\_name.

# Relative Layout

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent">
6.
7.     <TextView
8.         android:id="@+id/tv_username"
9.         android:layout_width="wrap_content"
10.        android:layout_height="wrap_content"
11.       android:text="username" />
12.
13.     <EditText
14.         android:id="@+id/edt_username"
15.         android:layout_width="match_parent"
16.         android:layout_height="wrap_content"
17.         android:layout_below="@+id/tv_username" />
18.
19.     <TextView
20.         android:id="@+id/tv_password"
21.         android:layout_width="wrap_content"
22.         android:layout_height="wrap_content"
23.         android:layout_below="@+id/edt_username"
24.         android:text="password" />
25.
26.     <EditText
27.         android:id="@+id/edt_password"
28.         android:layout_width="match_parent"
29.         android:layout_height="wrap_content"
30.         android:layout_below="@+id/tv_password"
31.         android:inputType="textPassword" />
32. </RelativeLayout>
```

Một số thuộc tính khác thường xuyên sử dụng với RelativeLayout:

- **android:layout\_alignParentBottom="boolean"**: Căn dưới phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout\_alignParentTop="boolean"**: Căn trên phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout\_alignParentRight="boolean"**: Căn phải phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout\_alignParentLeft="boolean"**: Căn trái phần tử hiện tại theo phần tử cha nếu set là true.

# Table Layout

99

- Table Layout là một view mà nhóm tất cả các view vào trong các hàng và các cột.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    android:stretchColumns="*"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TableRow>
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:src="@drawable/android"/>
        <TextView
            android:layout_span="3"
            android:layout_gravity="center"
            android:textSize="30sp"
            android:text="dong 1"/>
    </TableRow>
</TableLayout>
```

## Các thuộc tính quan trọng:

- android:stretchColumns
- android:shrinkColumns
- android:collapseColumns

Dùng **layout\_span** để trộn các cột

```
<TableRow>
    <TextView android:text="URL:" />
    <EditText
        android:id="@+id/entry"
        android:layout_span="3" />
</TableRow>
```

# Constraint Layout

100

- Constraint layout được giới thiệu lần đầu tiên tại sự kiện **Google I/O 2016**, Constraint Layout sẵn dùng với bản Android 2.3 (API level 9)
- **match\_constraint**: xác định width và height của một View
  - VD : TextView dưới được set nằm giữa view parent

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Text View 1"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

# Constraint Layout

101

- Thực hiện 02 điểm neo: ở trên bên trái và ở trên bên phải (hoặc ở dưới bên trái và ở dưới bên phải)

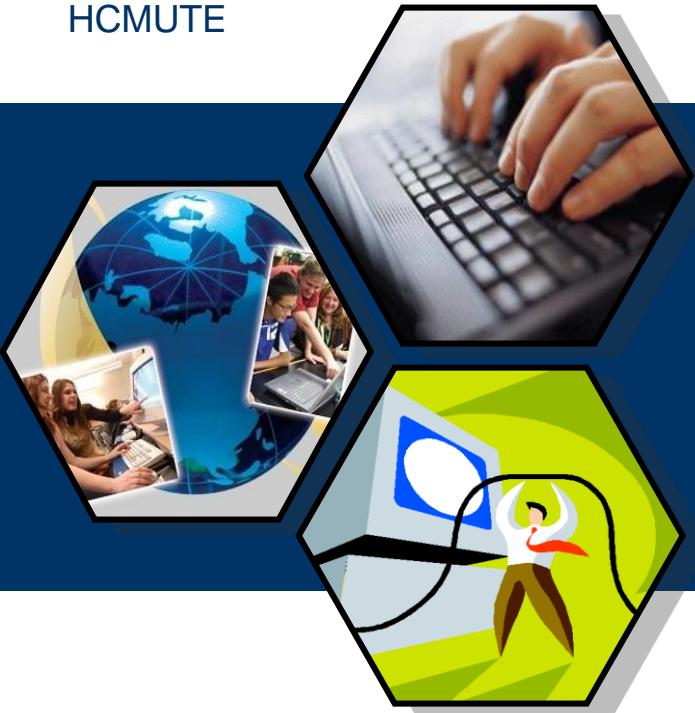
```
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
        xmlns:tools="http://schemas.android.com/tools"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        tools:context=".MainActivity">  
    <TextView  
        android:id="@+id/textView"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Hello World!"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.206" />
```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="88dp"  
    android:text="Button"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.233"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf  
    = "@+id/textView" />  
  
</androidx.constraintlayout.widget.ConstraintLayout  
>
```



# KHOA CÔNG NGHỆ THÔNG TIN

UTEx



## Điều khiển sự kiện (Event - Handling)

**Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM**

ThS. Nguyễn Hữu Trung



Có 3 khái niệm liên quan đến quản lý sự kiện:

- **Event Listeners:** Là một giao diện trong một View, View này chứa một phương thức duy nhất là callback. Phương thức này được gọi bởi Framework Android, khi View đã được đăng ký sự kiện và người dùng tác động lên View trong UI (giao diện người dùng).
- **Event Listeners Registration:** Event Registration là một quá trình mà một Event Handler đã được đăng ký với Event Listener để mà Handler này được gọi khi Event Listener kích hoạt sự kiện.
- **Event Handlers:** Khi một Event xảy ra, và chúng ta đã đăng ký một Event Listener cho sự kiện, thì Event Listener gọi Event Handler, là phương thức mà thực sự xử lý sự kiện đó

Event Handler	Event Listener Name
<b>onClick()</b>	OnClickListener() Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, image vv. Chúng ta sẽ sử dụng phương onClick() để xử lý sự kiện.
<b>onLongClick()</b>	OnLongClickListener() Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, image vv. trong một hoặc nhiều giây. Chúng ta sẽ sử dụng phương onLongClick() để xử lý sự kiện.
<b>onFocusChange()</b>	OnFocusChangeListener() Sự kiện phát sinh khi widget mất focus.

Event Handler	Event Listener Name
onKey()	OnFocusChangeListener() Sự kiện phát sinh khi người dùng focus trên widget và nhấn (presse) hoặc thả (release) một phím trên thiết bị.
onTouch()	OnTouchListener() Sự kiện phát sinh khi người dùng nhấn phím, nhả phím, hoặc bất kỳ cử chỉ chuyển động trên màn hình.
onMenuItemClick()	OnMenuItemClickListener() Sự kiện phát sinh khi người dùng chọn một mục trong menu.
onCreateContextMenu()	OnCreateContextMenuListener() Sự kiện phát sinh khi người dùng chọn một mục trong menu ngữ cảnh (Context Menu)

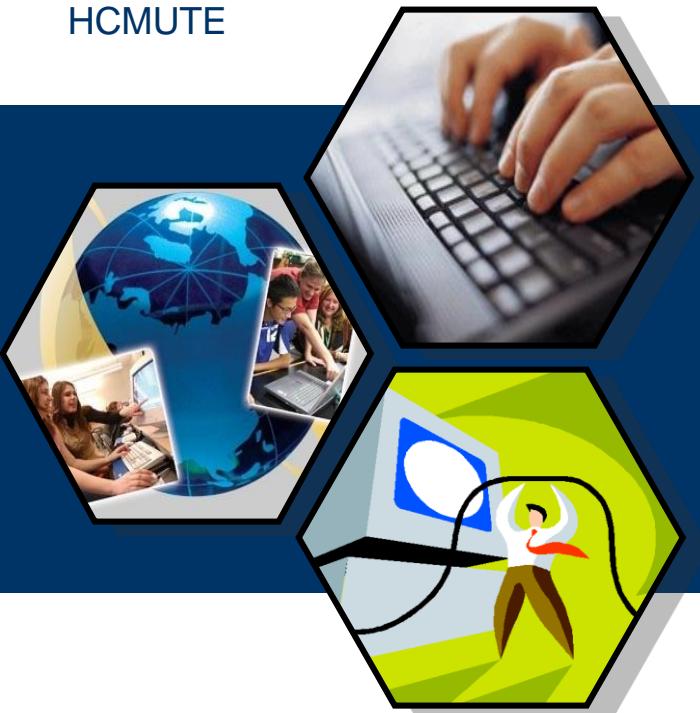
1. Xử lý sự kiện trong Layout (Handle event in Layout)
2. Xử lý sự kiện bằng lớp nặc danh (Inline anonymous listener)
3. Kế thừa Interface OnClickListener (Implements OnClickListener Interface)
4. Sử dụng biến (Event Listener using Variable)
5. Bắt sự kiện thông qua lớp lắng nghe (Listener Class)
6. PerformClick method



HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN

UTEx

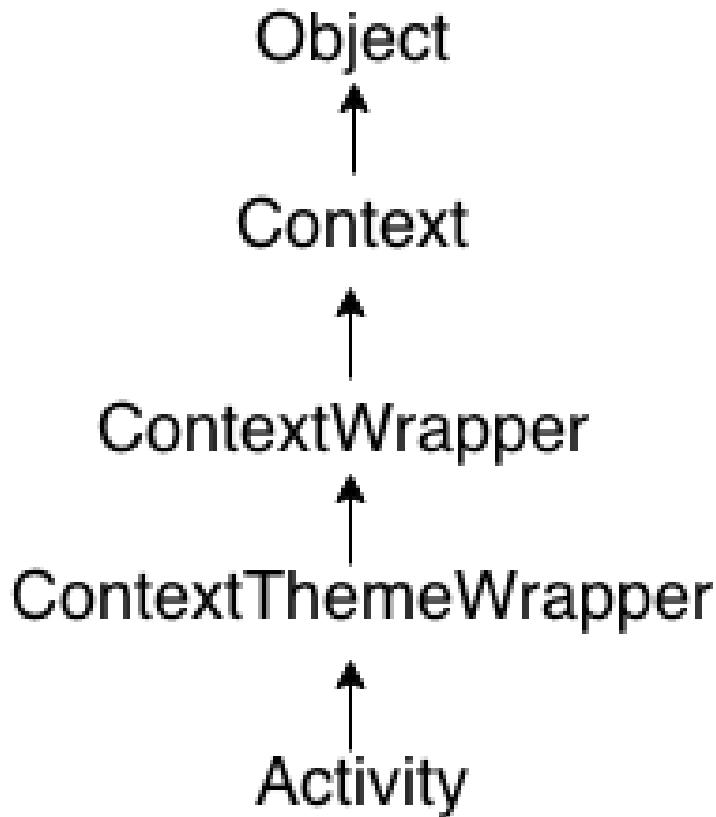


## ACTIVITY TRONG ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

ThS. Nguyễn Hữu Trung



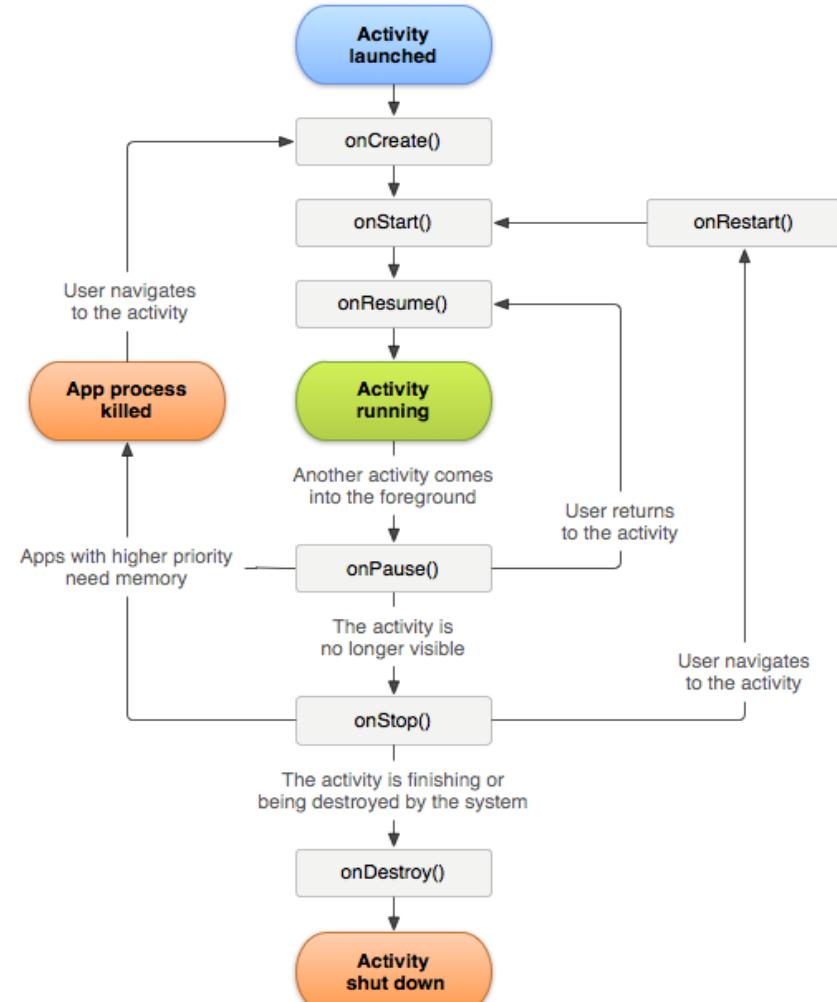


- Một **Activity** đại diện cho một màn hình với một giao diện người dùng. Giống như window hay Frame trong Java, Bạn có thể đặt các Widget vào một màn hình đơn giản

# Vòng đời của một Activity

108

- Vòng đời của một Activity trong Android gồm có 7 phương thức, mỗi phương thức thể hiện hành vi khác nhau của **Activity**.
- Chú ý: Phương thức `onCreate()` và `onDestroy()` được gọi một lần trong suốt vòng đời của Activity



Phương thức	Ý nghĩa
<b>onCreate()</b>	Nó là phương thức đầu tiên được gọi dùng để tạo một activity vào lần đầu tiên activity được gọi.
<b>onStart()</b>	Sẽ được gọi khi nó hiện hữu với người dùng.
<b>onResume()</b>	Sẽ được gọi khi người dùng tương tác với các ứng dụng.
<b>onPause()</b>	Tạm dừng một activity, không nhận dữ liệu do người dùng nhập vào và không thể thực thi lệnh nào. Phương thức này được gọi khi activity hiện tại đang được tạm dừng, và activity trước đó đang được tiếp tục.
<b>onStop()</b>	Được gọi khi một activity đã không được nhìn thấy trong thời gian dài.
<b>onRestart()</b>	Được gọi khi activity cần được dùng trở lại sau khi bị gọi onStop();
<b>onDestroy()</b>	Được gọi trước khi hệ thống hủy activity.

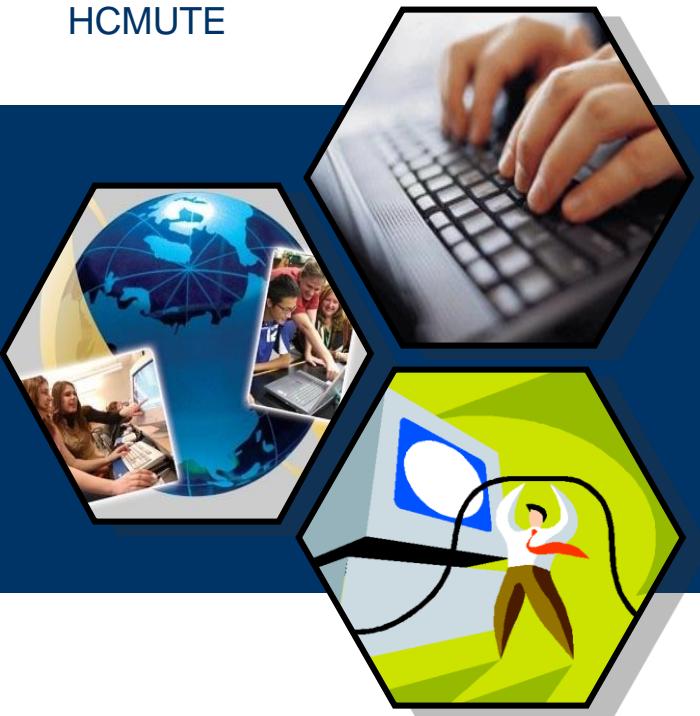
# Ví dụ

110



HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN



## Intents trong Android



- Intents là một thành phần quan trọng trong android. Nó cho phép các thành phần ứng dụng có thể yêu cầu các hàm từ các thành phần ứng dụng android khác. Ví dụ một activity có thể chạy một activity khác ở bên ngoài để chụp ảnh.
- Để chạy một activity, broadcast receivers thì sử dụng phương thức **startActivity(intent)**. Phương thức này được định nghĩa trong đối tượng **context**.
- Android intents thường được sử dụng chính:
  - Start dịch vụ
  - Gọi một activity
  - Hiển thị một trang web
  - Hiển thị danh sách liên hệ
  - Gởi một tin nhắn
  - Gọi điện thoại.
- Có 2 loại intent trong Android: implicit và explicit.

- Đối tượng dữ liệu của Intent là một cấu trúc dữ liệu được dùng bởi các component trong ứng dụng hoặc hệ thống nhận được và có những xử lý thích hợp.

Đối tượng dữ liệu của Intent có thể chứa các thuộc tính:

- **Action**

- Tên của các action mà Intent sẽ được thực hiện.
- Action có thể được định nghĩa sẵn Android cung cấp hoặc do người lập trình tự định nghĩa cho riêng ứng dụng (như dùng trong BroadcastReceiver).

- **Data**

- Dữ liệu mà thành phần được gọi (activity, service,...) sẽ xử lý.
- Được định dạng là đối tượng URI (tham khảo class Uri class).
- Data truyền vào sẽ được xử lý và hành động tùy theo “the MIME type” mà dữ liệu sử dụng.

### □ Category

- Là chuỗi ký tự chứa thông tin về nhóm phân loại các đối tượng để xử lý các intent.
- Có 2 category được thấy trong các file manifest:
  - **CATEGORY\_BROWSABLE**: Cho phép start một activity bằng Web browser để hiển thị dữ liệu định dạng là một liên kết ví dụ như một e-mail hay một bức hình trên mạng.
  - **CATEGORY\_LAUNCHER**: Khai báo để chỉ định activity sẽ được start khi bắt đầu mở ứng dụng.

### □ Extras

- Chứa tất cả các cặp key-value pairs chứa các thông tin dữ liệu bổ sung truyền qua.
- Thông số extras sẽ được gán và đọc bằng phương thức **putExtras()/getExtras()** methods tương ứng với thông số là đối tượng có cấu trúc Bundle.

### □ Flags

- Những giá trị của flag này sẽ hướng dẫn hệ thống Android cách để start một activity và kết thúc một activity.

# Intent Filter là gì?

115

- 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.
- Ví dụ chúng ta thiết lập 1 Activity với bộ lọc Intent cho phép bắt và xử lý các Intent gửi SMS.

```
<activity android:name=".MainActivity"
    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>
```

- Explicit Intents là những intent đã chỉ rõ thành phần sẽ nhận và xử lý. Thông thường những intent này sẽ không cần gán bổ sung thêm các thuộc tính khác như action, data. Explicit Intent thường được sử dụng để khởi chạy các activity, hoặc service trong cùng 1 ứng dụng

```
Intent intent = new Intent(getApplicationContext(),  
SecondActivity.class);  
  
startActivity(intent);  
  
finish();
```

# Implicit Intent

117

- Implicit Intent: là loại Intent có các Action được Android xây dựng sẵn, nó không chỉ rõ các Component xử lý (các class xử lý) mà nó sẽ cung cấp cho hệ điều hành một loạt các thông tin yêu cầu sau đó hệ điều hành sẽ đổi chiều xem trong hệ thống có bao nhiêu phần mềm khác có thể đáp ứng xử lý yêu cầu này rồi hiện ra một Dialog chứa tên của các phần mềm có thể xử lý để người dùng chọn
- Ví dụ, dưới đây sẽ yêu cầu hệ thống android để xem một trang web. Tất cả các trình duyệt web phải được đăng ký tương ứng vào dữ liệu intent thông qua bộ lọc intent.

```
Intent intent=new Intent(Intent.ACTION_VIEW);  
  
intent.setData(Uri.parse("https://google.com"));  
  
startActivity(intent);
```

Ví dụ: Gọi điện thoại trong android

```
Uri uri = Uri.parse("tel:" + "900");
```

```
Intent intent = new Intent(Intent.ACTION_CALL, uri);
```

// Để có thể call được thì trong AndroidManifest.xml phải khai báo

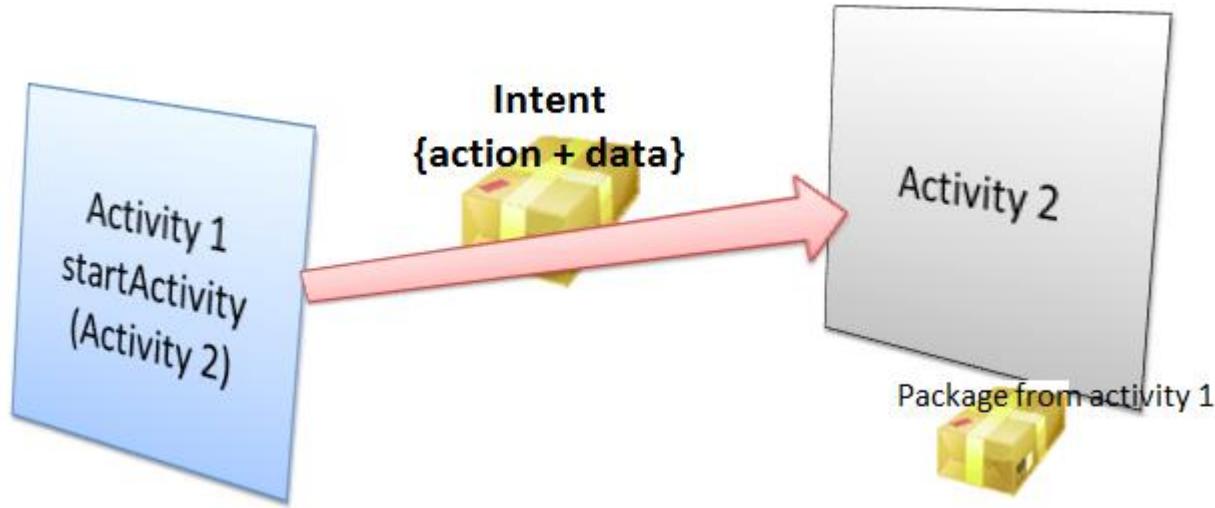
```
// <uses-permission android:name="android.permission.CALL_PHONE"
/>
```

```
startActivity(intent);
```

# Implicit Intent

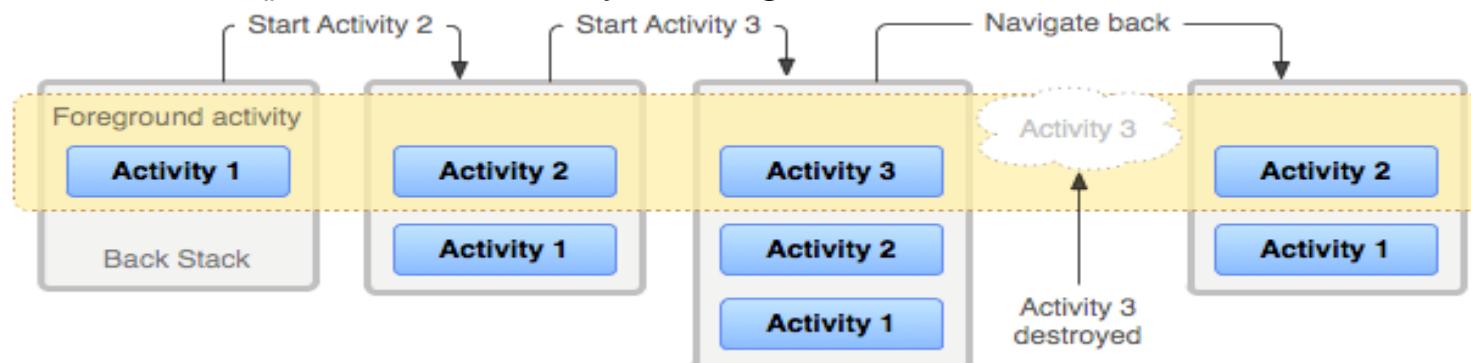
119

Định dạng	Action	Mô tả
tel:phone_number	ACTION_VIEW	Mở một Dial form
tel:phone_number	ACTION_CALL	Thực hiện gọi tới số
http://web_address	ACTION_VIEW	Mở trình duyệt web
https://web_address		theo địa chỉ
http://web_address	ACTION_WEB_SEARCH	Thực hiện tìm kiếm
https://web_address		
sms://	ACTION_SENDTO	Gửi tin nhắn
geo:latitude,longitude		
geo:latitude,longitude?z=zoom		
geo:0,0?q=my+street+address	ACTION_VIEW	Mở ứng dụng Bản đồ
geo:0,0?q=business+near+city		và trả tới vị trí được xác định



□ Không gửi kèm dữ liệu:

- Khi chuyển sang KetQuaActivity thì MainActivity được đẩy xuống dưới Back Stack và KetQuaActivity sẽ ở trên đầu của Back Stack. Vì vậy khi nhấn trở lại từ KetQuaActivity thì MainActivity được đẩy lên và onResume() của MainActivity được gọi.



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

Mọi Activity tạo ra phải được khai báo trong file **AndroidManifest.xml**, nếu không khai báo mà vẫn sử dụng Activity này thì hệ thống sẽ ném ra ngoại lệ, điều này cũng được áp dụng khi sử dụng Service trong Android.

```
<activity android:name=".KetQuaActivity"></activity>
```

### □ Gửi kèm dữ liệu

- Khi tạo Intent phải đặt các dữ liệu kèm theo qua đối tượng intent này, sử dụng các phương thức có tiền tố là put của đối tượng intent để gửi dữ liệu, mỗi dữ liệu đặt vào sẽ có 1 key đi kèm.
- Char, boolean, int, float, long, String hay kiểu mảng boolean[], char[], int[] , kiểu dữ liệu object, tuy nhiên những kiểu dữ liệu này phải implement interface Parcelable hoặc interface Serializable.

```
Intent intent = new Intent(MainActivity.this, KetQuaActivity.class);
intent.putExtra("int_key", 4);
intent.putExtra("char_key", 'r');
intent.putExtra("boolean_key", true);
intent.putExtra("long_key", 323L);
intent.putExtra("float_key", 3.2f);
intent.putExtra("string_key", "Chuyen Activity trong Android");
intent.putExtra("double", 32D);
intent.putExtra("int_array_key", new int[]{1, 2, 9});
intent.putExtra("boolean_array_key", new boolean[]{true, false, true, true});
intent.putExtra("char_array_Key", new char[] {'e', 'i', 't', 'g', 'u', 'i', 'd', 'e'});
intent.putExtra("rect_key", new Rect(0,0, 200, 200));
```

## □ Gửi dữ liệu đi sử dụng Bundle

- Thay vì sử dụng bằng cách đặt dữ liệu vào intent thì sử dụng 1 đối tượng gọi là **Bundle**, đặt các dữ liệu vào đối tượng này và sử dụng phương thức **putExtras()** để đặt hết vào intent.

```
Intent intent = new Intent(MainActivity.this, KetQuaActivity.class);
Bundle bundle = new Bundle();
bundle.putChar("char", 'e');
bundle.putInt("int", 3);
bundle.putString("string", "Nguyễn Hữu Trung");
bundle.putFloat("float", 5.2f);
bundle.putDouble("double", 843D);
bundle.putLong("long", 55343L);
bundle.putParcelable("parcelable", new Rect(0, 0, 300, 300));
intent.putExtras(bundle);
startActivity(intent);
```

## □ Nhận dữ liệu

- Sau khi startActivity(), ở Activity B (KetQuaActivity) có được dữ liệu mà Activity A (MainActivity) gửi đến như sau:
  - Lấy ra đối tượng Intent mà Activity A gửi qua bằng phương thức getIntent().
  - Từ đối tượng intent có được từ getIntent(), lấy những dữ liệu trong đối tượng này bằng phương thức getXExtra() (với X là kiểu dữ liệu muốn lấy).

```
// Get data from MainActivity
int intValue = intent.getIntExtra("int_key", 0);
char charValue = intent.getCharExtra("char_key", 'a');
float floatValue = intent.getFloatExtra("float_key", 0f);
boolean booleanValue = intent.getBooleanExtra("boolean_key", false);
String stringValue = intent.getStringExtra("string_key");
int[] intArrayValue = intent.getIntArrayExtra("int_array_key");
char[] charArrayValue = intent.getCharArrayExtra("char_array_key");
Rect rect = intent.getParcelableExtra("rect_key");
```

## □ Nhận dữ liệu

- Sau khi startActivity(), ở Activity B (KetQuaActivity) có được dữ liệu mà Activity A (MainActivity) gửi đến như sau:
  - Lấy ra đối tượng Intent mà Activity A gửi qua bằng phương thức getIntent().
  - Từ đối tượng intent có được từ getIntent(), lấy những dữ liệu trong đối tượng này bằng phương thức getXExtra() (với X là kiểu dữ liệu muốn lấy).

Nếu gửi dữ liệu bằng **Bundle** thì làm như sau:

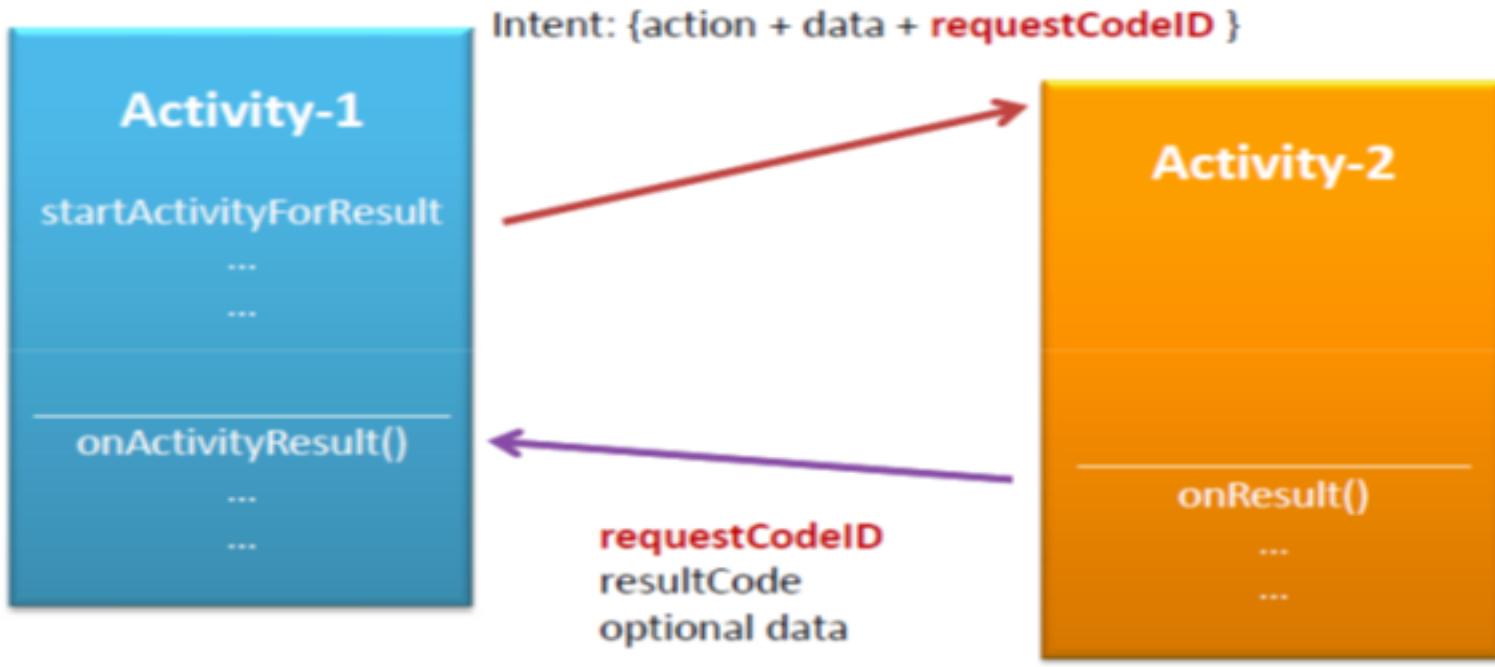
```
Intent intent = getIntent();
Bundle bundle = intent.getExtras();
char charValue = bundle.getChar("char");
int intValue = bundle.getInt("int");
String stringValue = bundle.getString("string");
float floatValue = bundle.getFloat("float");
double doubleValue = bundle.getDouble("double");
long longValue = bundle.getLong("long");
Rect rect = bundle.getParcelable("parcelable");
```

Truyền Đối Tượng qua lại giữa các Activity, các đối tượng này phải được Serialize

```
class Person implements Serializable
{
    private int id;
    private String name;
    public Person(int id, String name)
    {
        this.id=id;
        this.name=name;
    }
    public String toString()
    {
        return this.id+" - "+this.name;
    }
}
```

```
Bundle bundle=new Bundle();
Person p=new Person(1, "teo");
bundle.putSerializable("t1", p);
//...
Person t1=(Person) bundle.getSerializable("t1");
```

- Bằng cách sử dụng `startActivityForResult(Intent intent, int requestCode)` thay vì dùng `startActivity()` bạn có thể start một Activity và sau đó nhận kết quả trả về từ Activity đó thông qua phương thức `onActivityResult()`.



- Chúng ta sẽ dựa vào requestCodeID và resultCode để xử lý.

```
startActivityForResult ( Intent, requestCodeID )
```



- Việc tạo Intent trong trường hợp này cũng như trường hợp trước. Nó chỉ khác hàm gọi :

```
onActivityResult ( requestCodeID, resultCode, Intent )
```



```
public static final int REQUEST_CODE_INPUT=113;
```

```
//Mở Activity với REQUEST_CODE_INPUT  
  
Intent intent = new  
Intent(getApplicationContext(),ActivityInput.class);  
  
//gọi startActivityForResult  
  
startActivityForResult(intent, REQUEST_CODE_INPUT);
```

@Override

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

    //Kiểm tra có đúng requestCode =REQUEST_CODE_INPUT hay không
    //Vì ta có thể mở Activity với những RequestCode khác nhau
    if(requestCode==REQUEST_CODE_INPUT)

    {
        //Kiểm tra ResultCode trả về, cái này ở bên InputDataActivity truyền về
        //có nó rồi thì xử lý trả lên thông thường
        switch(resultCode)
        {
            case RESULT_CODE_SAVE1:
                //giá trị từ InputDataActivity
                int v1= data.getIntExtra("data", 0);}
```

# Ví dụ

131

```
11 public class FirstActivity extends AppCompatActivity {
12     public static final int REQ_CODE = 100;
13     TextView textViewReceivedText;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_first);
19         setTitle("First Activity");
20
21         textViewReceivedText = findViewById(R.id.textViewReceivedText);
22
23         findViewById(R.id.buttonGotoSecond).setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View view) {
26                 Intent intent = new Intent(getApplicationContext(), SecondActivity.class);
27                 startActivityForResult(intent, REQ_CODE);
28             }
29         });
30     }
31
32     @Override
33     protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
34         super.onActivityResult(requestCode, resultCode, data);
35         if(requestCode == REQ_CODE){
36             if(resultCode == RESULT_OK){
37                 if(data != null & data.getStringExtra(SecondActivity.KEY_NAME) != null){
38                     textViewReceivedText.setText(data.getStringExtra(SecondActivity.KEY_NAME));
39                 }
40             }
41         }
42     }
43 }
```

- Từ androidx.appcompat 1.3.1 trở lên thì `startActivityForResult` không được hỗ trợ nữa.

```
public class FirstActivity extends AppCompatActivity {
    public static final int REQ_CODE = 100;
    TextView textViewReceivedText;

    ActivityResultLauncher<Intent> startForResult = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            if(result != null && result.getResultCode() == RESULT_OK){
                if(result.getData() != null && result.getData().getStringExtra(SecondActivity.KEY_NAME) != null){
                    textViewReceivedText.setText(result.getData().getStringExtra(SecondActivity.KEY_NAME));
                }
            }
        }
    });

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
        setTitle("First Activity");

        textViewReceivedText = findViewById(R.id.textViewReceivedText);

        findViewById(R.id.buttonGotoSecond).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(packageContext: FirstActivity.this, SecondActivity.class);
                startForResult.launch(intent);
            }
        });
    }
}
```

# Xử lý lỗi Deprecated for startActivityForResult

133

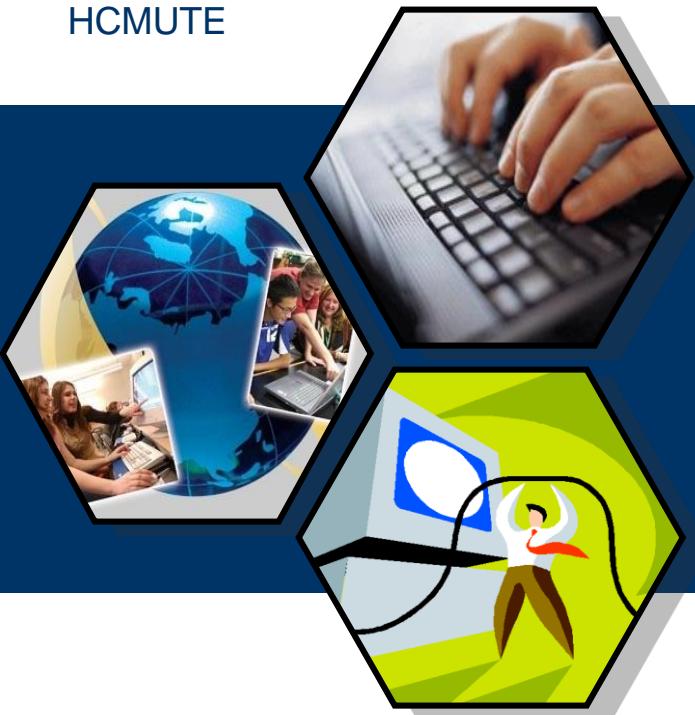
```
10 public class SecondActivity extends AppCompatActivity {
11     EditText editTextToSend;
12     public static final String KEY_NAME = "NAME";
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_second);
18         setTitle("Second Activity");
19
20         editTextToSend = findViewById(R.id.editTextToSend);
21
22         findViewById(R.id.buttonSend).setOnClickListener(new View.OnClickListener() {
23             @Override
24             public void onClick(View view) {
25                 String enteredText = editTextToSend.getText().toString();
26                 Intent intent = new Intent();
27                 intent.putExtra(KEY_NAME, enteredText);
28                 setResult(RESULT_OK, intent);
29                 finish();
30             }
31         });
32     }
33 }
34 }
```



HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN

UTEx



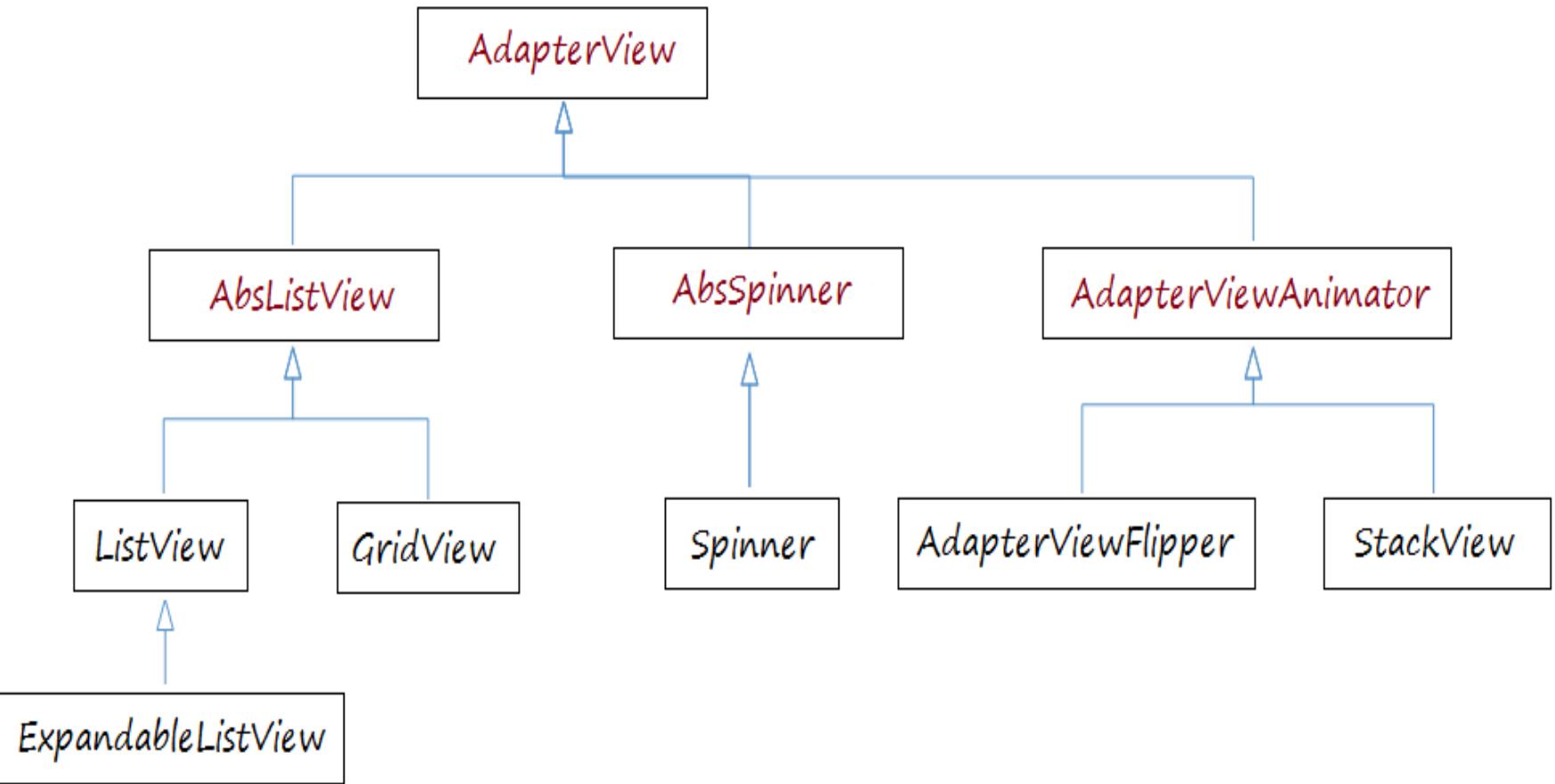
**LISTVIEW**

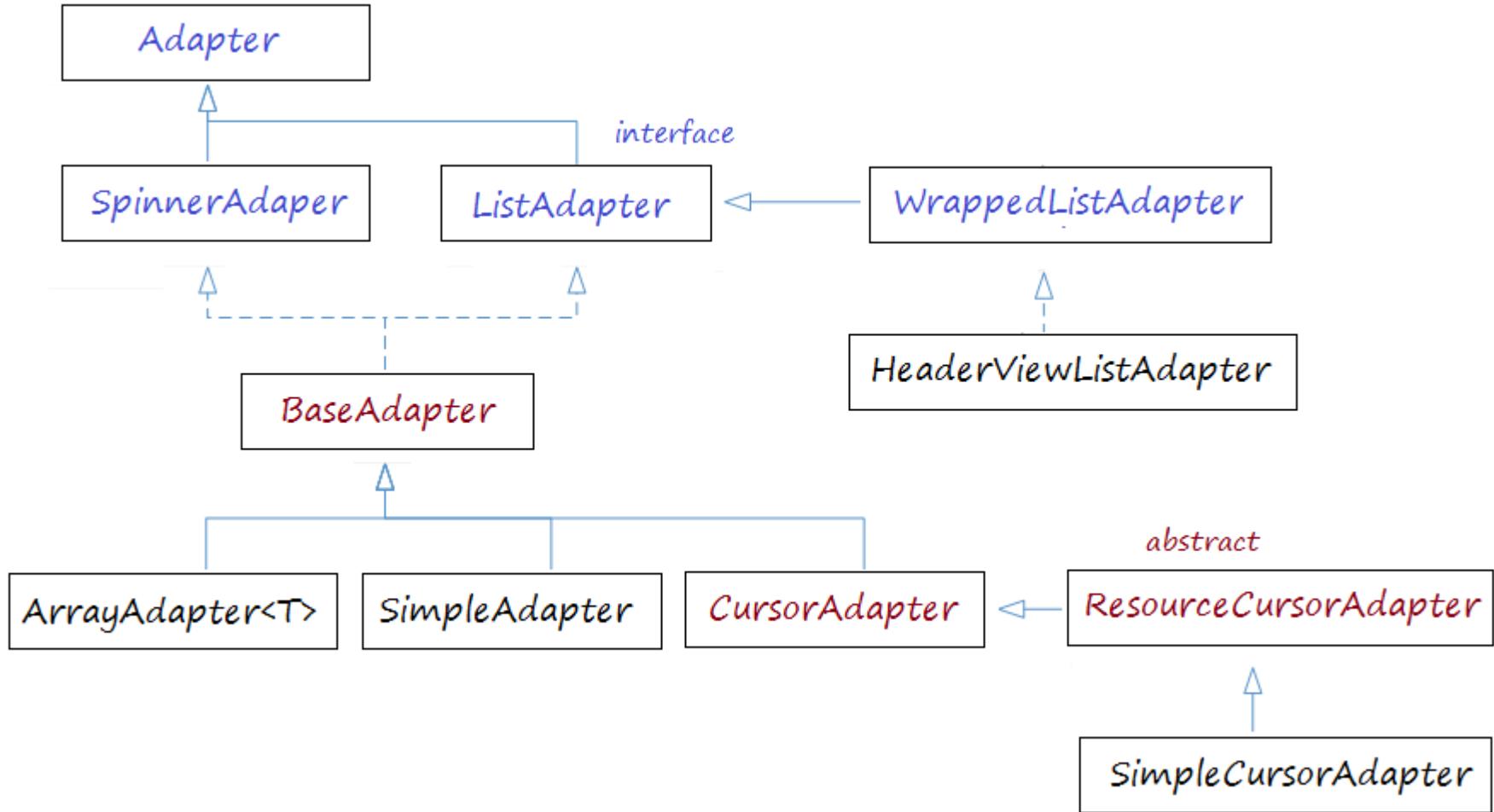
**Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM**

ThS. Nguyễn Hữu Trung



- Có nhiều View cần tới Android Adapter để quản lý dữ liệu hiển thị, các View này là con của class AdapterView

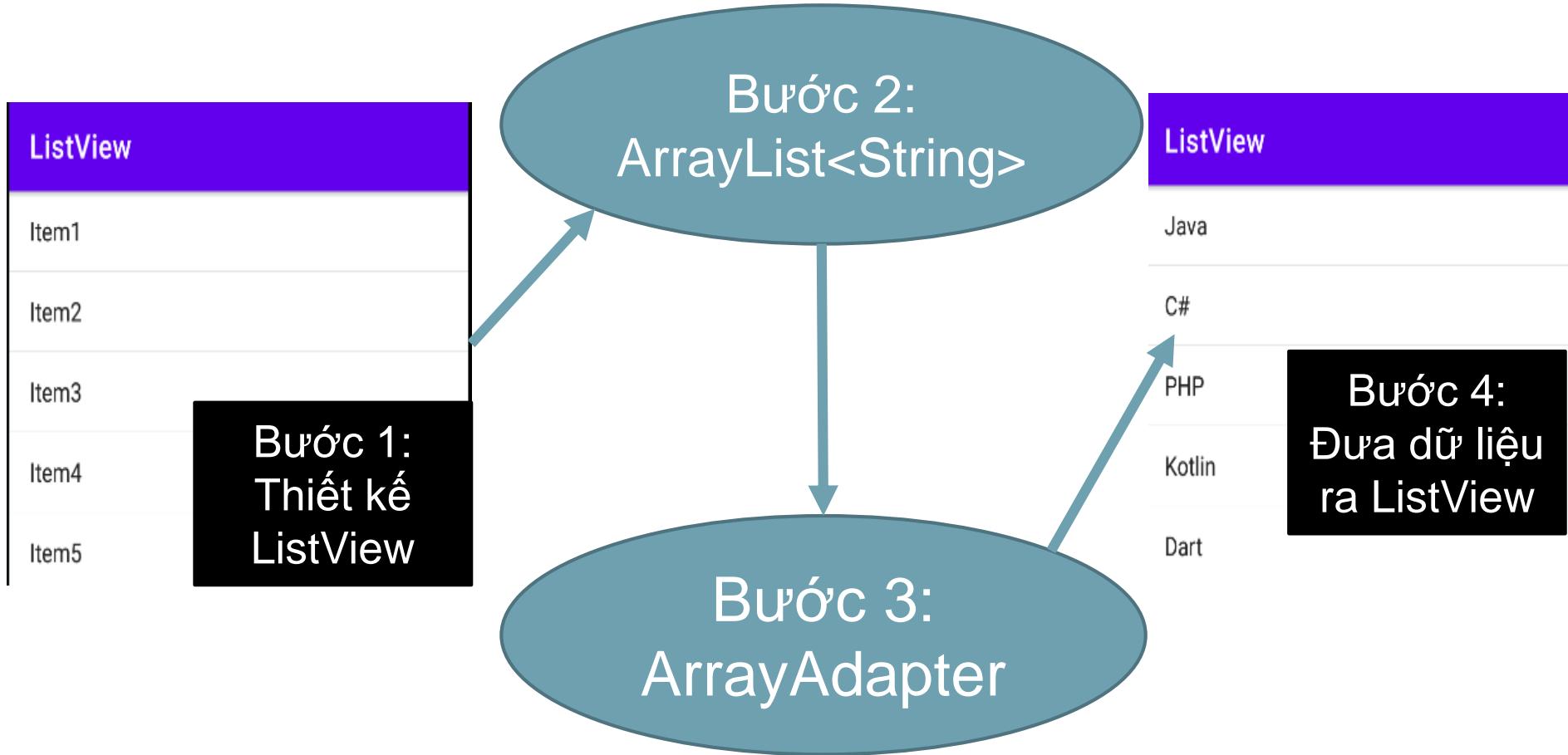




- ❑ **ListView** trong Android là một view group, hiển thị các item theo một danh sách có thể cuộn được theo chiều thẳng đứng.

```
<ListView  
    android:id="@+id/listview"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
</ListView>
```

Thuộc tính	Mô tả
<b>android:id</b>	ID là duy nhất để nhận diện Layout
<b>android:divider</b>	Nó có thể vẽ hoặc tô màu giữa các item trong danh sách.
<b>android:dividerHeight</b>	Nó xác định chiều cao của dải phân cách. Điều này có thể là px, dp, sp, in hoặc mm.
<b>android:entries</b>	Chỉ định tham chiếu đến một tài nguyên mảng sẽ điền vào ListView.
<b>android:footerDividersEnabled</b>	Khi được đặt thành false, ListView sẽ không vẽ dải phân cách trước mỗi chế độ xem chân trang. Giá trị mặc định là true.
<b>android:headerDividersEnabled</b>	Khi được đặt thành false, ListView sẽ không vẽ dải phân cách sau mỗi chế độ xem tiêu đề. Giá trị mặc định là true.



## ListView

Item1

Item2

Item3

Item4

Item5

Bước 1:  
Thiết kế  
ListView

Mở file .xml thiết kế ListView

<ListView

```
    android:id="@+id/listview1"  
    android:layout_width="409dp"  
    android:layout_height="354dp"  
    tools:layout_editor_absoluteX="1dp"  
    tools:layout_editor_absoluteY="1dp"  
    tools:ignore="MissingConstraints" />
```

Bước 2:  
ArrayList<String>

```
public class MainActivity extends  
AppCompatActivity {  
    //khai báo  
    ListView listView;  
    ArrayList<String> arrayList;
```

```
//ánh xạ  
listView = (ListView) findViewById(R.id.listview1);  
//Thêm dữ liệu vào List  
arrayList = new ArrayList<>();  
arrayList.add("Java");  
arrayList.add("C#");  
arrayList.add("PHP");  
arrayList.add("Kotlin");  
arrayList.add("Dart");
```

### Bước 3: ArrayAdapter

ArrayAdapter có 03 tham số:

- Context: màn hình hiển thị
- Dạng Layout muốn đổ vào
- Dữ liệu List đổ vào

```
//Tạo ArrayAdapter
ArrayAdapter adapter = new ArrayAdapter(
    MainActivity.this,
    android.R.layout.simple_list_item_1,
    arrayList
);
```

ListView

Java

C#

PHP

Kotlin

Dart

Bước 4:  
Đưa dữ liệu  
ra ListView

//truyền dữ liệu từ adapter ra listview

`listView.setAdapter(adapter);`

## ListView

Java

C#

PHP

Kotlin

Dart

2

//bắt sự kiện click nhanh trên từng dòng của Listview

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        //code yêu cầu
        //i: trả về vị trí click chuột trên ListView -> i ban đầu =0
        Toast.makeText(MainActivity.this,""+i,
        Toast.LENGTH_SHORT).show();
    }
});
```

Lấy nội dung: arrayList.get(i)

## ListView

Java

C#

PHP

Kotlin

Dart

2

```
//bắt sự kiện click giữ trên từng dòng của ListView
listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean
onItemLongClick(AdapterView<?> adapterView,
View view, int i, long l) {
        //code yêu cầu
        //i: trả về vị trí click chuột trên ListView -> i
        ban đầu =0
        Toast.makeText(MainActivity.this,"Bạn đang
nhấn giữ "+ i + "-" + arrayList.get(i) ,
Toast.LENGTH_SHORT).show();
        return false;
    }
});
```

ListView

Thêm môn học:

THÊM

```
EditText editText1;  
Button btnNhaph;
```

ASP.NET

Java

C#

PHP

Kotlin

Dart

```
editText1 = (EditText) findViewById(R.id.editText1);  
btnNhaph = (Button) findViewById(R.id.btnNhaph);
```

```
btnNhaph.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String name = editText1.getText().toString();  
        arrayList.add(name);  
        adapter.notifyDataSetChanged();  
    }  
});
```

## ListView

Thêm môn học:

```
//Khai báo  
Button btnCapnhat;  
int vitri = -1;
```

THÊM

CẬP NHẬT

Java

C#

PHP

```
//ánh xạ  
btnCapNhat = (Button) findViewById(R.id.btnCapNhat);
```

```
//bắt sự kiện trên từng dòng của Listview  
listView.setOnItemClickListener(new  
AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view,  
    int i, long l) {  
        //lấy nội dung đưa lên edittext  
        editText1.setText(arrayList.get(i));  
        vitri = i;  
    }  
});
```

## ListView

Thêm môn học:

THÊM

CẬP NHẬT

Java

C#

PHP

```
btnCapNhat.setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        arrayList.set(vitri, editText1.getText().toString());  
        adapter.notifyDataSetChanged();  
    }  
});
```

```
//Xóa item  
arrayList.remove(i);  
adapter.notifyDataSetChanged();
```

Bước 1:  
Tạo Class  
MonHoc

Bước 3:  
ArrayList<MonHoc>

Java  
Java 1

C#  
C# 1

Bước 5:  
Đưa dữ liệu  
ra ListView

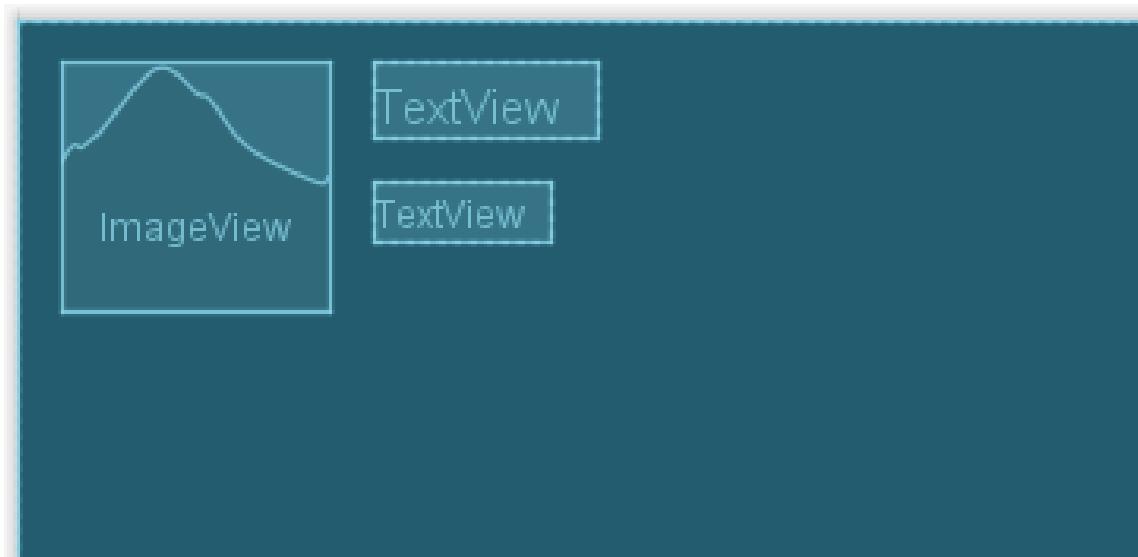
Bước 2:  
row\_monhoc.xml

Bước 4:  
MonHocAdapter

Bước 1:  
Tạo Class  
MonHoc

```
package vn.iotstar.listview;  
  
public class MonHoc {  
    private String name;  
    private String desc;  
    private int pic;  
    //kích phải chọn Generate.. Chọn  
    constructor, getter and setter
```

Bước 2: tạo layout  
row\_monhoc.xml



## Bước 4: MonHocAdapter

```
//trả về số dòng  
@Override  
public int getCount() {  
    return monHocList.size(); //lấy  
    kích thước monhoclist  
}
```

```
public class MonhocAdapter extends  
BaseAdapter {  
    //khai báo  
    private Context context;  
    private int layout;  
    private List<MonHoc> monHocList;  
  
    //tạo Constructors  
  
    public MonhocAdapter(Context context,  
    int layout, List<MonHoc> monHocList) {  
        this.context = context;  
        this.layout = layout;  
        this.monHocList = monHocList;  
    }
```

@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {  
    //lấy context  
    LayoutInflater inflater = (LayoutInflater)  
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
    //gọi view chưa layout  
    view = inflater.inflate(layout,null);  
    //ánh xạ view  
    TextView textName = (TextView) view.findViewById(R.id.textName);  
    TextView textDesc = (TextView) view.findViewById(R.id.textDesc);  
    ImageView imagePic = (ImageView) view.findViewById(R.id.imagePic);  
  
    //gán giá trị  
    MonHoc monHoc = monHocList.get(i);  
    textName.setText(monHoc.getName());  
    textDesc.setText(monHoc.getDesc());  
    imagePic.setImageResource(monHoc.getPic());  
    //trả về view  
    return view;  
}
```



Java

Java 1

C#

C# 1

PHP

PHP 1

Bước 5:  
Đưa dữ liệu  
ra ListView

```
public class MainActivity extends AppCompatActivity {  
    //khai báo  
    ListView listView;  
    ArrayList<MonHoc> arrayList;  
    MonhocAdapter adapter;  
  
    //ánh xạ  
    AnhXa();  
  
    //Tạo Adapter  
    adapter = new MonhocAdapter(MainActivity.this,  
        R.layout.row_monhoc,  
        arrayList  
    );  
    //truyền dữ liệu từ adapter ra listview  
    listView.setAdapter(adapter);
```



Java

Java 1

C#

C# 1

PHP



PHP 1

Bước 5:  
Đưa dữ liệu  
ra ListView

```
private void AnhXa() {
    listView = (ListView) findViewById(R.id.listview1);

    editText1 = (EditText) findViewById(R.id.editText1);
    btnNhaph = (Button) findViewById(R.id.btnNhaph);
    btnCapNhat = (Button) findViewById(R.id.btnCapNhat);

    //Thêm dữ liệu vào List
    arrayList = new ArrayList<>();
    arrayList.add(new MonHoc("Java","Java 1",R.drawable.java1));
    arrayList.add(new MonHoc("C#","C# 1",R.drawable.c));
    arrayList.add(new MonHoc("PHP","PHP 1",R.drawable.php));
    arrayList.add(new MonHoc("Kotlin","Kotlin
1",R.drawable.kotlin));
    arrayList.add(new MonHoc("Dart","Dart 1",R.drawable.dart));
}
```

```
Mở MonHocAdapter lên  
//tạo class viewholder  
private class ViewHolder{  
    TextView textView, textDesc;  
    ImageView imagePic;  
}
```

# ViewHolder trong ListView

157

```
@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    //khởi tạo viewholder
    ViewHolder viewHolder;
    //lấy context
    if (view==null){
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        //gọi view chưa layout
        view = inflater.inflate(layout,null);
        //ánh xạ view
        viewHolder = new ViewHolder();
        viewHolder.textName = (TextView) view.findViewById(R.id.textName);
        viewHolder.textDesc = (TextView) view.findViewById(R.id.textDesc);
        viewHolder.imagePic = (ImageView) view.findViewById(R.id.imagePic);
        view.setTag(viewHolder);
    }else{
        viewHolder= (ViewHolder) view.getTag();
    }
}
```

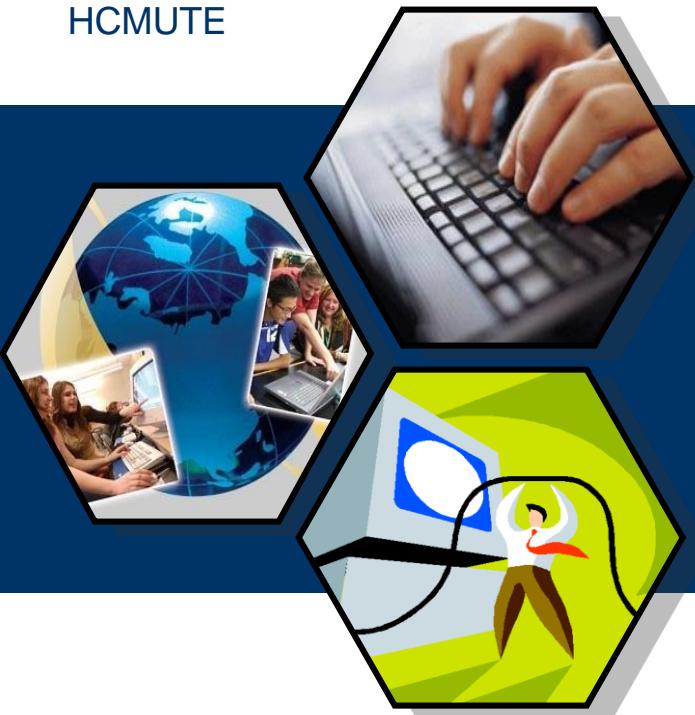
```
//gán giá trị  
MonHoc monHoc = monHocList.get(i);  
viewHolder.textName.setText(monHoc.getName());  
viewHolder.textDesc.setText(monHoc.getDesc());  
viewHolder.imagePic.setImageResource(monHoc.getPic());  
//trả về view  
return view;  
}
```



HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN

UTEx



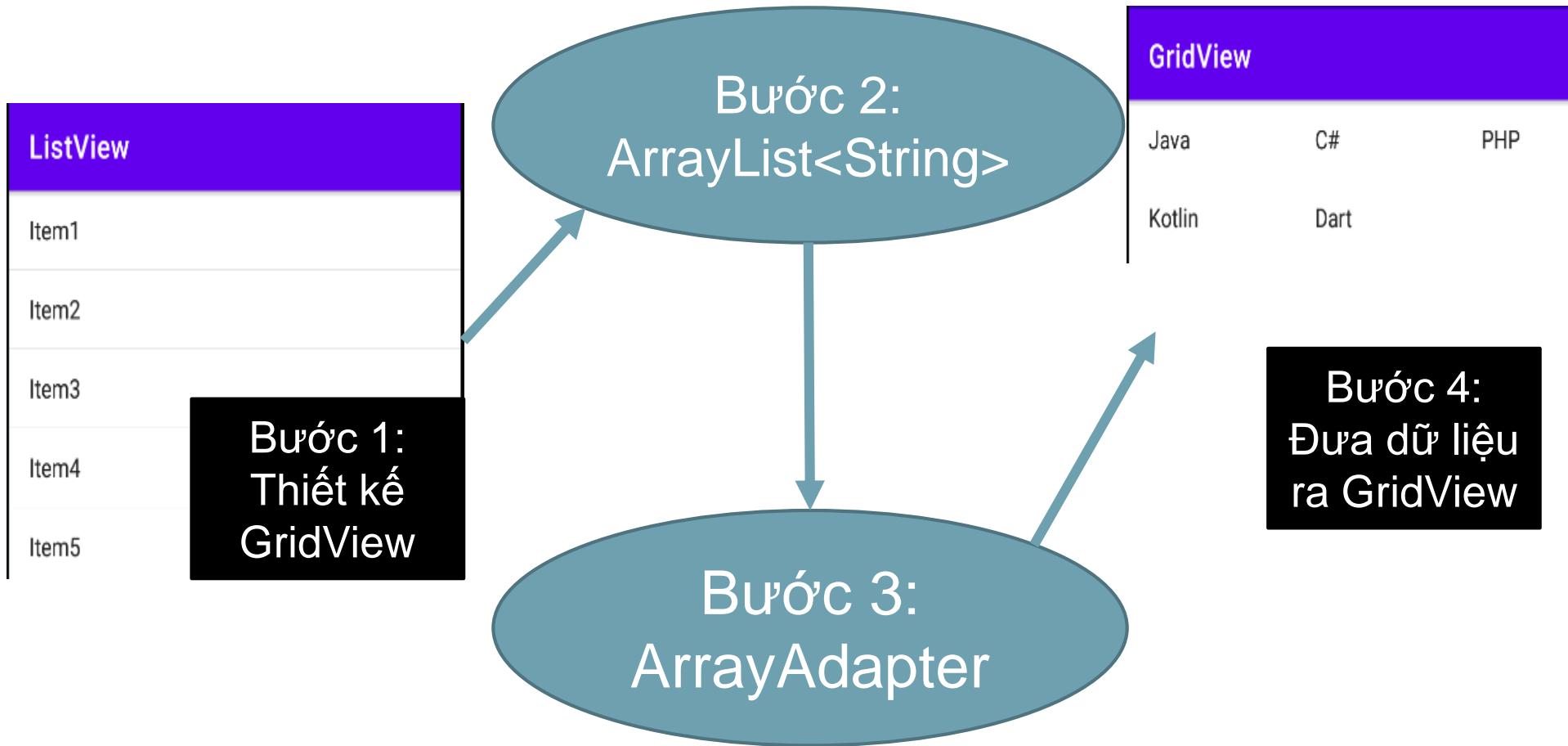
**GRIDVIEW**

**Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM**

ThS. Nguyễn Hữu Trung



- GridView trong Android hiển thị các item trong mảng lưới hai chiều có thể scroll và các item này không cần thiết phải được định nghĩa trước, nhưng chúng tự động chèn vào Layout bởi sử dụng một ListAdapter.
- Các thuộc tính của GridView:
  - **android:verticalSpacing** -> Định nghĩa khoảng cách mặc định theo chiều dọc giữa các hàng. Có thể là trong px, dp, sp, in, hoặc mm.
  - **android:horizontalSpacing** -> Định nghĩa khoảng cách mặc định theo chiều ngang giữa các cột. Có thể là trong px, dp, sp, in, hoặc mm.
  - **android:numColumns** -> Xác định có bao nhiêu cột để hiển thị



## GridView

Java

C#

PHP

Kotlin

Dart

Bước 1:  
Thiết kế  
ListView

## Mở file .xml thiết kế GridView

```
<GridView  
    android:numColumns="3"  
    android:id="@+id/gridview1"  
    android:layout_width="409dp"  
    android:layout_height="729dp"  
    tools:layout_editor_absoluteX="1dp"  
    tools:layout_editor_absoluteY="1dp"  
    tools:ignore="MissingConstraints" />
```

Bước 2:  
ArrayList<String>

```
public class MainActivity extends  
AppCompatActivity {  
    //khai báo  
    GridView gridView;  
    ArrayList<String> arrayList;
```

```
//ánh xạ  
gridView = (GridView) findViewById(R.id.gridview1);  
//Thêm dữ liệu vào List  
arrayList = new ArrayList<>();  
arrayList.add("Java");  
arrayList.add("C#");  
arrayList.add("PHP");  
arrayList.add("Kotlin");  
arrayList.add("Dart");
```

### Bước 3: ArrayAdapter

ArrayAdapter có 03 tham số:

- Context: màn hình hiển thị
- Dạng Layout muốn đổ vào
- Dữ liệu List đổ vào

```
//Tạo ArrayAdapter
ArrayAdapter adapter = new ArrayAdapter(
    MainActivity.this,
    android.R.layout.simple_list_item_1,
    arrayList
);
```

## GridView

Java

C#

PHP

Kotlin

Dart

//truyền dữ liệu từ adapter ra gridview  
`gridView.setAdapter(adapter);`

Bước 4:  
Đưa dữ liệu  
ra ListView

## ListView

Java

C#

PHP

Kotlin

Dart

2

//bắt sự kiện click nhanh trên từng dòng của Gridview

```
gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        //code yêu cầu
        //i: trả về vị trí click chuột trên GridView -> i ban đầu =0
        Toast.makeText(MainActivity.this, "" + i,
        Toast.LENGTH_SHORT).show();
    }
});
```

Lấy nội dung: arrayList.get(i)

## ListView

Java

C#

PHP

Kotlin

Dart

2

//bắt sự kiện click giữ trên từng dòng của Gridview  
gridView.setOnItemLongClickListener(new  
AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean  
onItemLongClick(AdapterView<?> adapterView,  
View view, int i, long l) {  
 //code yêu cầu  
 //i: trả về vị trí click chuột trên GridView -> i  
 ban đầu =0  
 Toast.makeText(MainActivity.this,"Bạn đang  
nhấn giữ "+ i + "-" + arrayList.get(i) ,  
Toast.LENGTH\_SHORT).show();  
 return false;  
}  
});

ListView

Thêm môn học:

THÊM

```
EditText editText1;  
Button btnNhaph;
```

ASP.NET

Java

C#

PHP

Kotlin

Dart

```
editText1 = (EditText) findViewById(R.id.editText1);  
btnNhaph = (Button) findViewById(R.id.btnNhaph);
```

```
btnNhaph.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String name = editText1.getText().toString();  
        arrayList.add(name);  
        adapter.notifyDataSetChanged();  
    }  
});
```

## ListView

Thêm môn học:

```
//Khai báo  
Button btnCapnhat;  
int vitri = -1;
```

THÊM

CẬP NHẬT

Java

C#

PHP

```
//ánh xạ  
btnCapNhat = (Button) findViewById(R.id.btnCapNhat);
```

```
//bắt sự kiện trên từng dòng của Gridview  
gridView.setOnItemClickListener(new  
AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view,  
    int i, long l) {  
        //lấy nội dung đưa lên edittext  
        editText1.setText(arrayList.get(i));  
        vitri = i;  
    }  
});
```

## ListView

Thêm môn học:

THÊM

CẬP NHẬT

Java

C#

PHP

```
btnCapNhat.setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        arrayList.set(vitri, editText1.getText().toString());  
        adapter.notifyDataSetChanged();  
    }  
});
```

```
//Xóa item  
arrayList.remove(i);  
adapter.notifyDataSetChanged();
```

Bước 1:  
Tạo Class  
MonHoc

Bước 3:  
ArrayList<MonHoc>

Bước 2:  
row\_monhoc.xml

Bước 4:  
MonHocAdapter

GridView



Java

Java 1



Kotlin

Kotlin 1



C#

C# 1



Dart



PHP

PHP 1



Dart 2

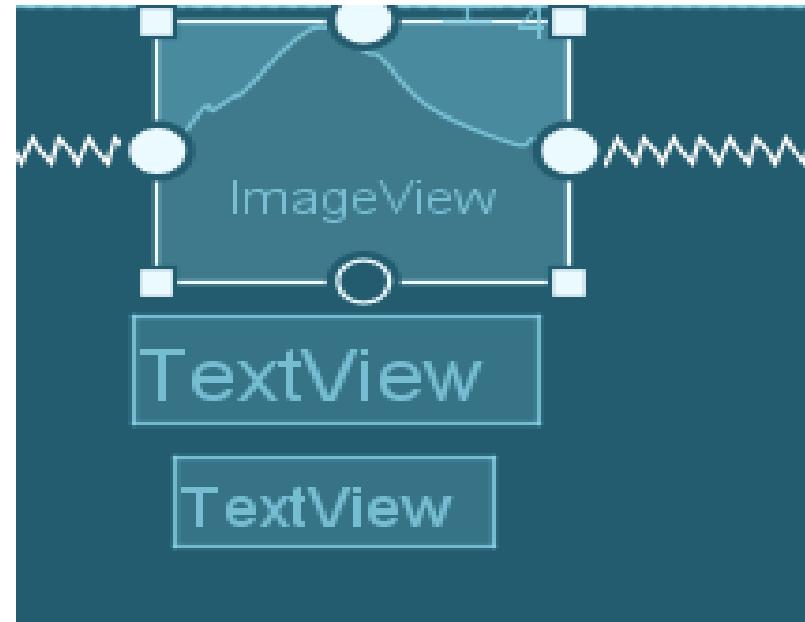
Bước 5:  
Đưa dữ liệu  
ra GridView

Bước 1:  
Tạo Class  
MonHoc

```
package vn.iotstar.gridview;

public class MonHoc {
    private String name;
    private String desc;
    private int pic;
    //kích phải chọn Generate.. Chọn
    constructor, getter and setter
```

Bước 2: tạo layout  
row\_monhoc.xml



## Bước 4: MonHocAdapter

```
//trả về số dòng  
@Override  
public int getCount() {  
    return monHocList.size(); //lấy  
kích thước monhoclist  
}
```

```
public class MonhocAdapter extends  
BaseAdapter {  
    //khai báo  
    private Context context;  
    private int layout;  
    private List<MonHoc> monHocList;  
  
    //tạo Constructors  
  
    public MonhocAdapter(Context context,  
int layout, List<MonHoc> monHocList) {  
        this.context = context;  
        this.layout = layout;  
        this.monHocList = monHocList;  
    }
```

monHocList == null ? 0 :monHocList.size()

@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {  
    //lấy context  
    LayoutInflater inflater = (LayoutInflater)  
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
    //gọi view chưa layout  
    view = inflater.inflate(layout,null);  
    //ánh xạ view  
    TextView textName = (TextView) view.findViewById(R.id.textName);  
    TextView textDesc = (TextView) view.findViewById(R.id.textDesc);  
    ImageView imagePic = (ImageView) view.findViewById(R.id.imagePic);  
  
    //gán giá trị  
    MonHoc monHoc = monHocList.get(i);  
    textName.setText(monHoc.getName());  
    textDesc.setText(monHoc.getDesc());  
    imagePic.setImageResource(monHoc.getPic());  
    //trả về view  
    return view;  
}
```

## GridView



Bước 5:  
Đưa dữ liệu  
ra ListView

```
public class MainActivity extends AppCompatActivity {  
    //khai báo  
    ListView listView;  
    ArrayList<MonHoc> arrayList;  
    MonhocAdapter adapter;  
  
    //ánh xạ  
    AnhXa();  
  
    //Tạo Adapter  
    adapter = new MonhocAdapter(MainActivity.this,  
        R.layout.row_monhoc,  
        arrayList  
    );  
    //truyền dữ liệu từ adapter ra listview  
    listView.setAdapter(adapter);
```

## GridView



Bước 5:  
Đưa dữ liệu  
ra ListView

```
private void AnhXa() {  
    gridView = (GridView) findViewById(R.id.gridview1);  
  
    editText1 = (EditText) findViewById(R.id.editText1);  
    btnNhaph = (Button) findViewById(R.id.btnNhaph);  
    btnCapNhat = (Button) findViewById(R.id.btnCapNhat);  
  
    //Thêm dữ liệu vào List  
    arrayList = new ArrayList<>();  
    arrayList.add(new MonHoc("Java", "Java 1", R.drawable.java1));  
    arrayList.add(new MonHoc("C#", "C# 1", R.drawable.c));  
    arrayList.add(new MonHoc("PHP", "PHP 1", R.drawable.php));  
    arrayList.add(new MonHoc("Kotlin", "Kotlin  
1", R.drawable.kotlin));  
    arrayList.add(new MonHoc("Dart", "Dart 1", R.drawable.dart));  
}
```

Mở MonHocAdapter lên  
//tạo class viewholder  
**private class ViewHolder{**  
    **TextView textView, textDesc;**  
    **ImageView imagePic;**  
**}**

# ViewHolder trong GridView

179

```
@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    //khởi tạo viewholder
    ViewHolder viewHolder;
    //lấy context
    if (view==null){
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        //gọi view chưa layout
        view = inflater.inflate(layout,null);
        //ánh xạ view
        viewHolder = new ViewHolder();
        viewHolder.textName = (TextView) view.findViewById(R.id.textName);
        viewHolder.textDesc = (TextView) view.findViewById(R.id.textDesc);
        viewHolder.imagePic = (ImageView) view.findViewById(R.id.imagePic);
        view.setTag(viewHolder);
    }else{
        viewHolder= (ViewHolder) view.getTag();
    }
}
```

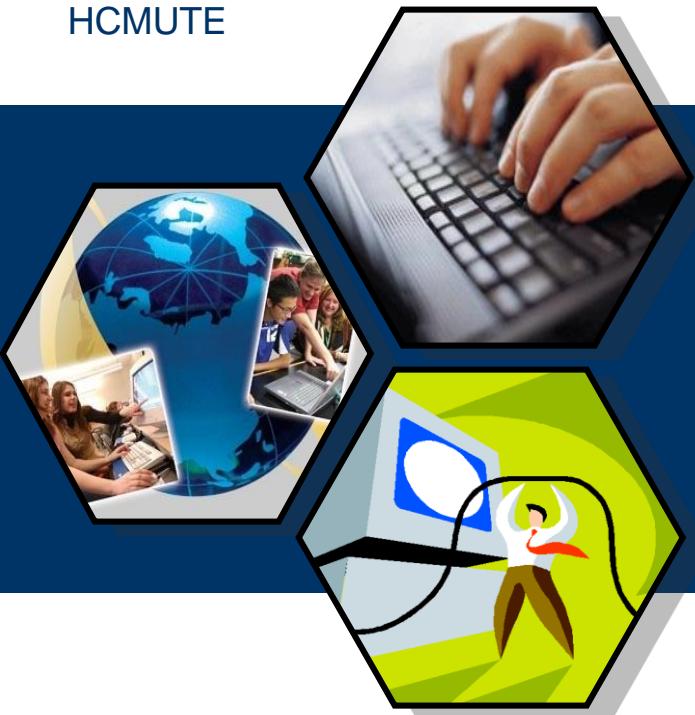
```
//gán giá trị  
MonHoc monHoc = monHocList.get(i);  
viewHolder.textName.setText(monHoc.getName());  
viewHolder.textDesc.setText(monHoc.getDesc());  
viewHolder.imagePic.setImageResource(monHoc.getPic());  
//trả về view  
return view;  
}
```



HCMUTE

# KHOA CÔNG NGHỆ THÔNG TIN

UTEx



## RECYCLEVIEW

**Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM**

ThS. Nguyễn Hữu Trung



- RecyclerView là 1 ViewGroup, được giới thiệu trong Android L (Android API 21), đây là 1 ViewGroup, có chức năng tương tự ListView nhưng nó mạnh mẽ, linh hoạt hơn rất nhiều, ListView chỉ hỗ trợ cuộn các item trong ListView theo chiều dọc (vertical) mà không hỗ trợ cuộn theo chiều ngang (horizontal), RecyclerView hỗ trợ được tất cả và hơn thế nữa.

- So với ListView thì RecyclerView có những điểm vượt trội hơn như sau:
  - **ListView** không cần sử dụng ViewHolder để cải thiện hiệu suất của ListView. Nhưng ngược lại, khi tạo 1 Adapter sử dụng với **RecyclerView**, bắt buộc phải sử dụng ViewHolder để cải thiện hiệu suất.
  - Mục đích sử dụng **ViewHolder**, để tái sử dụng View, nhằm tránh việc tạo View mới và **findViewById** quá nhiều.
  - ListView chỉ hỗ trợ danh sách dạng cuộn dọc, RecyclerView cung cấp **RecyclerView.LayoutManager** cho phép layout các item trong ListView theo các kiểu khác nhau (ngang, dọc, dạng lưới, dạng staggered grid).
  - Xử lý animation cho các item trong ListView không dễ dàng nhưng RecyclerView có hỗ trợ ItemAnimator giúp xử lý animation khi thêm vào hay xóa 1 item ra khỏi Recycler 1 cách dễ dàng. Mặc định RecyclerView, sử dụng DefaultItemAnimator.
  - ListView sử dụng divider không được linh hoạt nhưng với RecyclerView có hỗ trợ ItemDecoration, cho phép vẽ divider 1 cách tùy thích.
  - ListView hỗ trợ các phương thức `setOnItemClickListener()` và `setOnLongItemClickListener()` để chọn 1 item trong ListView. RecyclerView chỉ hỗ trợ 1 phương thức `onItemTouchListener()`.

- RecyclerView.Adapter

- Đây là thành phần xử lý dữ liệu collection (dữ liệu kiểu danh sách) và bind (gắn) những dữ liệu này, lên các item của RecyclerView.
- Khi tạo custom Adapter phải override lại 2 phương thức chính là:
  - onCreateViewHolder(): dùng để tạo View mới cho RecyclerView, nếu RecyclerView đã cached lại View thì phương thức này sẽ không được gọi.
  - onBindViewHolder(): dùng gắn dữ liệu vào View.

- LayoutManager

- Là thành phần có chức năng sắp xếp các item trong RecyclerView. Các item cuộn dọc hay ngang đều phụ thuộc vào đặt LayoutManager cho RecyclerView.
- Các lớp con của LayoutManager:
  - LinearLayoutManager: hỗ trợ cuộn các item theo chiều ngang hay chiều dọc.
  - GridLayoutManager: layout các item trong RecyclerView dưới dạng Grid giống như khi sử dụng GridView.
  - StaggeredGridLayoutManager: layout các item trong ListView dưới dạng lưới so le.
  - ItemAnimator: là thành phần hỗ trợ animation khi thêm vào hay xóa 1 item ra khỏi RecyclerView. Để tìm hiểu rõ phần này, cần tìm hiểu các lớp sau:
    - ItemAnimator: là lớp đại diện cho khung sườn của animation trong RecyclerView.
    - SimpleItemAnimator: là lớp wrapper lại ItemAnimator.
    - DefaultItemAnimator: lớp xử lý animation mặc định sử dụng trong RecyclerView.

1. Tạo model lớp để chứa dữ liệu.
2. Thêm RecyclerView vào **main\_activity.xml**.
3. Tạo giao diện cho 1 dòng.
4. Tạo custom Adapter và gán dữ liệu cho từng dòng trong Adapter.
5. Cài đặt RecyclerView trong **MainActivity.java**.

## □ Bước 1: tạo model để chứa dữ liệu

```
public class SongModel implements Serializable {  
    private String mCode;  
    private String mTitle;  
    private String mLyric;  
    private String mArtist;  
    //TẠO CONSTRUCTOR, GETTER VÀ SETTER BẰNG  
    CÁCH KÍCH PHẢI CHUỘT CHỌN GENERATE..  
}
```

## □ Bước 2: thêm RecyclerView vào main\_activity.xml

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/rv_songs"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```



□ **Bước 3: tạo giao diện cho 1 row để hiển thị lên RecyclerView**

File **row\_item\_song.xml** trong thư mục layout của project, được thiết kế để hiển thị dòng như hình dưới đây:

```
<TextView  
    android:id="@+id/tv_title"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ellipsize="end"  
    android:textColor="#2c3e50"  
    android:textSize="16dp"  
    android:textStyle="bold" />  
  
<TextView  
    android:id="@+id/tv_lyric"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ellipsize="end"  
    android:textColor="#34495e"  
    android:textSize="14dp" />  
  
<TextView  
    android:id="@+id/tv_artist"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textColor="#7f8c8d"  
    android:textSize="14dp" />
```

```
<TextView  
    android:id="@+id/tv_code"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textColor="#19b395"  
    android:textSize="18dp" />
```

□ Bước 4: tạo custom Adapter và gắn dữ liệu cho từng dòng trong Adapter

```
public class SongAdapter extends RecyclerView.Adapter<SongAdapter.SongViewHolder> {  
    private static final String TAG = "SongAdapter";  
    private List<SongModel> mSongs;  
    private Context mContext;  
    private LayoutInflator mLayoutInflater;  
    public SongAdapter(Context context, List<SongModel> datas) {  
        mContext = context;  
        mSongs = datas;  
        mLayoutInflater = LayoutInflator.from(context);  
    }  
    @Override  
    public SongViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
        // Inflate view from row_item_song.xml  
        View itemView = mLayoutInflater.inflate(R.layout.row_item_song, parent, attachToRoot: false);  
        return new SongViewHolder(itemView);  
    }  
    @Override  
    public void onBindViewHolder(SongViewHolder holder, int position) {  
        // Get song in mSong via position  
        SongModel song = mSongs.get(position);  
        //bind data to viewholder  
        holder.tvCode.setText(song.getmCode());  
        holder.tvTitle.setText(song.getmTitle());  
        holder.tvLyric.setText(song.getmLyric());  
        holder.tvArtist.setText(song.getmArtist());  
    }  
}
```

□ Bước 4: tạo custom Adapter và gắn dữ liệu cho từng dòng trong Adapter

```
@Override  
public int getItemCount() {  
    return mSongs.size();  
}  
  
class SongViewHolder extends RecyclerView.ViewHolder {  
    private TextView tvCode;  
    private TextView tvTitle;  
    private TextView tvLyric;  
    private TextView tvArtist;  
  
    public SongViewHolder(View itemView) {  
        super(itemView);  
        tvCode = (TextView) itemView.findViewById(R.id.tv_code);  
        tvTitle = (TextView) itemView.findViewById(R.id.tv_title);  
        tvLyric = (TextView) itemView.findViewById(R.id.tv_lyric);  
        tvArtist = (TextView) itemView.findViewById(R.id.tv_artist);  
    }  
}
```

Tạo **Custom Adapter** trong **RecyclerView** dễ dàng hơn so với **ListView**, chỉ cần tạo 1 lớp kế thừa **RecyclerView**. Còn **ViewHolder** chỉ việc xử lý trong **onCreateViewHolder** để tạo mới **ViewHolder**. Và **onBindViewHolder** để gán dữ liệu lấy từ collection vào **ViewHolder**.

## □ Bước 5: cài đặt RecyclerView trong MainActivity.java

Không giống như **ListView** chỉ cần đặt Adapter cho ListView là đủ. Đối với **RecyclerView** phải đặt 2 thành phần bắt buộc dưới đây:

- Đặt **adapter** cho **RecyclerView** sử dụng phương thức **setAdapter()**.
- Đặt **layout manager** để muôn layout các item như các kiểu dọc, ngang, lưới, lưới so le bằng cách sử dụng phương thức **setLayoutManager()**.

```
rvSongs.setAdapter(mSongAdapter);
LinearLayoutManager linearLayoutManager = new
LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);
rvSongs.setLayoutManager(linearLayoutManager);
```

Vì muốn RecyclerView cuộn theo chiều dọc nên sử dụng `LinearLayoutManager.VERTICAL`. Nếu muốn cuộn theo chiều ngang thì truyền vào `LinearLayoutManager.HORIZONTAL`:

```
LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL,
false);
```

Sử dụng những lớp sau để `setLayoutManager()` cho RecyclerView:

- `GridLayoutManager`: layout các item trong RecyclerView theo dạng lưới.
- `StaggeredLayoutManager`: layout các item trong RecyclerView theo dạng lưới so le nhau.

## □ Bước 5: cài đặt RecyclerView trong MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private RecyclerView rvSongs;
    private SongAdapter mSongAdapter;
    private List<SongModel> mSongs;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rvSongs = (RecyclerView) findViewById(R.id.rv_songs);

        // Create song data
        mSongs = new ArrayList<>();
        mSongs.add(new SongModel( mCode: "60696", mTitle: "NẾU EM CÒN TỒN TẠI", mLyric: "Khi anh bắt đầu 1 tình yêu Là lúc anh tự thay", mArtist: "Trịnh Đình Quang"));
        mSongs.add(new SongModel( mCode: "60701", mTitle: "NGỐC", mLyric: "Có rất nhiều những câu chuyện Em dấu riêng mình em biết", mArtist: "Khắc Việt"));
        mSongs.add(new SongModel( mCode: "60650", mTitle: "HÃY TIN ANH LẦN NỮA", mLyric: "Dẫu cho ta đã sai khi ở bên nhau Cô yêu thương", mArtist: "Thiên Dũng"));
        mSongs.add(new SongModel( mCode: "60610", mTitle: "CHUỖI NGÀY VẮNG EM", mLyric: "Từ khi em bước ra đi cõi lòng anh ngập tràn bao", mArtist: "Duy Cường"));
        mSongs.add(new SongModel( mCode: "60656", mTitle: "KHI NGƯỜI MÌNH YÊU KHÓC", mLyric: "Nước mắt em đang rơi trên những ngón tay Nước mắt em", mArtist: "Phạm Mạnh Quỳnh"));
        mSongs.add(new SongModel( mCode: "60685", mTitle: "MỎ", mLyric: "Anh mơ gặp em anh mơ được ôm anh mơ được gần", mArtist: "Trịnh Thăng Bình"));
        mSongs.add(new SongModel( mCode: "60752", mTitle: "TÌNH YÊU CHẮP VÁ", mLyric: "Muốn đi xa nơi yêu thương mình từng có Để không nghe", mArtist: "Mr. Siro"));
        mSongs.add(new SongModel( mCode: "60608", mTitle: "CHỞ NGÀY MÙA TAN", mLyric: "1 ngày mưa và em khuất xa nơi anh bóng dáng cứ", mArtist: "Trung Đức"));
        mSongs.add(new SongModel( mCode: "60603", mTitle: "CÂU HỎI EM CHUA TRÀ LỜI", mLyric: "Cần nơi em 1 lời giải thích thật lòng Đừng lặng im", mArtist: "Yuki Huy Nam"));
        mSongs.add(new SongModel( mCode: "60720", mTitle: "QUA ĐÌ LĂNG LỄ", mLyric: "Đôi khi đến với nhau yêu thương chẳng được lâu nhưng khi", mArtist: "Phan Mạnh Quỳnh"));
        mSongs.add(new SongModel( mCode: "60856", mTitle: "QUÊN ANH LÀ ĐIỀU EM KHÔNG THỂ - REMIX", mLyric: "Cần thêm bao lâu để em quên đi niềm đau Cần thêm", mArtist: "Thiện Ngôn"));
        mSongAdapter = new SongAdapter( context: this, mSongs);
        rvSongs.setAdapter(mSongAdapter);
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager( context: this, LinearLayoutManager.VERTICAL, reverseLayout: false);
        rvSongs.setLayoutManager(linearLayoutManager);
    }
}
```

- Chọn item trong RecyclerView, bằng cách setOnClickListener cho itemView, trong constructor của SongViewHolder như sau:

```
//thiết lập sự kiện
itemView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SongModel song = mSongs.get(getAdapterPosition());
        Toast.makeText(mContext, song.getmTitle(),
        Toast.LENGTH_SHORT).show();
    }
});
```

- Giới thiệu và cách tạo RecyclerView nâng cao, RecyclerView với nhiều dòng khác nhau hay còn được gọi là RecyclerView Multiple View Type, có thể nhận thấy tính năng News Feed của Facebook.
- Việc xây dựng RecylerView có nhiều dòng item khác nhau cũng giống như RecyclerView chỉ có 1 dòng item nhưng khác biệt ở custom adapter như sau:
  - Tạo n ViewHolder ứng với n dòng item.
  - Override lại phương thức getItemViewType(int position) để lấy về kiểu tương ứng với từng vị trí trong collection.
  - Phương thức onCreateViewHolder() phải dựa vào kiểu để tạo ViewHolder tương ứng.
  - Phương thức onBindViewHolder() phải dựa vào kiểu để gắn dữ liệu tương ứng vào ViewHolder.

- Hướng dẫn tạo Custom Adapter
- Tạo 3 file XML tương ứng với 3 dòng item như sau:
  - ▣ **row\_text.xml**: có 1 TextView để hiển thị văn bản.
  - ▣ **row\_image.xml**: hiển thị 1 hình ảnh ImageView.
  - ▣ **row\_user.xml**: có 2 TextView hiển thị tên và địa chỉ của user.

- Hướng dẫn tạo Custom Adapter
- Tạo 3 file XML tương ứng với 3 dòng item như sau:
  - ▣ **row\_text.xml**: có 1 TextView để hiển thị văn bản.
  - ▣ **row\_image.xml**: hiển thị 1 hình ảnh ImageView.
  - ▣ **row\_user.xml**: có 2 TextView hiển thị tên và địa chỉ của user.

## row\_text.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_marginTop="3dp"
        android:layout_height="wrap_content">
    <TextView
        android:background="#ecf0f1"
        android:padding="5dp"
        android:textColor="#2c3e50"
        android:id="@+id/tv_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</FrameLayout>
```

## row\_image.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp">
    <ImageView
        android:id="@+id/imv_image"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#95a5a6" />
</FrameLayout>
```

## row\_user.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="3dp"
    android:background="#bdc3c7"
    android:orientation="vertical">
    <TextView
        android:id="@+id/tv_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#34495e"
        android:textSize="18dp" />
    <TextView
        android:id="@+id/tv_address"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#16a085"
        android:textSize="16dp" />
</LinearLayout>
```

## Tạo UserModel.java

```
public class UserModel implements Serializable {  
    private String name;  
    private String address;  
    //tạo Contructor, getter và setter bằng cách nhấp chuột phải  
    //chọn Generate...  
}
```

## Tạo UserModel.java

```
public class UserModel implements Serializable {  
    private String name;  
    private String address;  
    //tạo Contructor, getter và setter bằng cách nhấp chuột phải  
    //chọn Generate...  
}
```

Tiếp theo, tiến hành tạo 1 CustomAdapter.

```
public class CustomAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {  
    private Context mContext;  
    private List<Object> mObjects;  
    public static final int TEXT = 0;  
    public static final int IMAGE = 1;  
    public static final int USER = 2;  
    public CustomAdapter(Context context, List<Object> objects) {  
        mContext = context;  
        mObjects = objects;  
    }
```

- Có 3 ViewHolder như sau:

```
public class TextViewHolder extends RecyclerView.ViewHolder {  
    private TextView tvText;  
  
    public TextViewHolder(View itemView) {  
        super(itemView);  
        tvText = (TextView) itemView.findViewById(R.id.tv_text);  
  
        itemView.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast.makeText(mContext, mObjects.get(getAdapterPosition()).toString(), Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

- Có 3 ViewHolder như sau:

```
public class ImageViewHolder extends RecyclerView.ViewHolder {  
    private ImageView imvImage;  
  
    public ImageViewHolder(View itemView) {  
        super(itemView);  
  
        imvImage = (ImageView) itemView.findViewById(R.id.imv_image);  
        itemView.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast.makeText(mContext, mObjects.get(getAdapterPosition()).toString(), Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

## □ Có 3 ViewHolder như sau:

```
public class UserViewHolder extends RecyclerView.ViewHolder {  
    private TextView tvName;  
    private TextView tvAddress;  
    public UserViewHolder(View itemView) {  
        super(itemView);  
        tvName = (TextView) itemView.findViewById(R.id.tv_name);  
        tvAddress = (TextView) itemView.findViewById(R.id.tv_address);  
  
        itemView.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                UserModel user = (UserModel) mObjects.get(getAdapterPosition());  
                Toast.makeText(mContext, text: user.getName() + ", " + user.getAddress(), Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

## ❑ Override lại phương thức getItemViewType():

```
@Override  
public int getItemViewType(int position) {  
    if (mObjects.get(position) instanceof String)  
        return TEXT;  
    else if (mObjects.get(position) instanceof Integer)  
        return IMAGE;  
    else if (mObjects.get(position) instanceof UserModel)  
        return USER;  
    return -1;  
}
```

## □ Override lại phương thức onCreateViewHolder():

```
@Override  
public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    LayoutInflater li = LayoutInflater.from(mContext);  
    switch (viewType) {  
        case TEXT:  
            View itemView0 = li.inflate(R.layout.row_text, parent, attachToRoot: false);  
            return new TextViewHolder(itemView0);  
        case IMAGE:  
            View itemView1 = li.inflate(R.layout.row_image, parent, attachToRoot: false);  
            return new ImageViewHolder(itemView1);  
        case USER:  
            View itemView2 = li.inflate(R.layout.row_user, parent, attachToRoot: false);  
            return new UserViewHolder(itemView2);  
        default:  
            break;  
    }  
    return null;  
}
```

## ❑ Override lại phương thức onBindViewHolder():

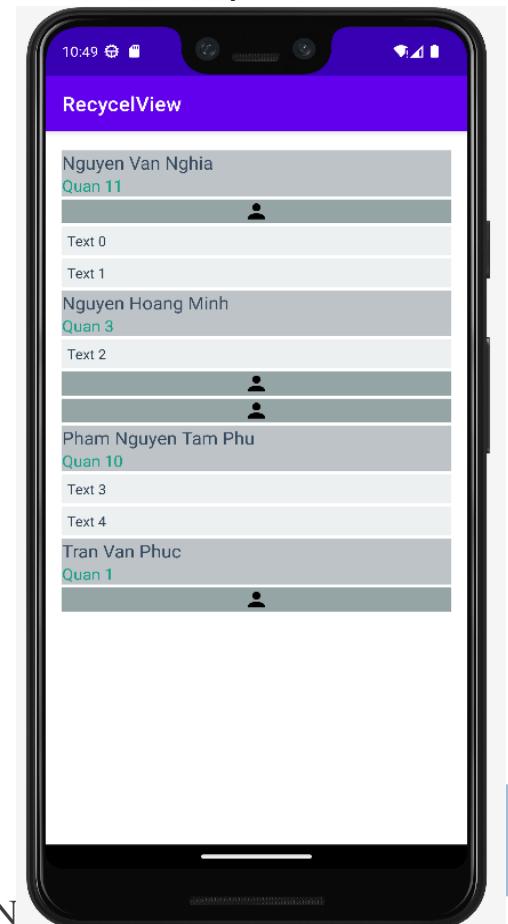
```
@Override  
public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {  
    switch (getItemViewType(position)) {  
        case TEXT:  
            TextViewHolder textViewHolder = (TextViewHolder) holder;  
            textViewHolder.tvText.setText(mObjects.get(position).toString());  
            break;  
        case IMAGE:  
            ImageViewHolder imageViewHolder = (ImageViewHolder) holder;  
            imageViewHolder.imvImage.setImageResource((int) mObjects.get(position));  
            break;  
        case USER:  
            UserModel user = (UserModel) mObjects.get(position);  
            UserViewHolder userViewHolder = (UserViewHolder) holder;  
            userViewHolder.tvName.setText(user.getName());  
            userViewHolder.tvAddress.setText(user.getAddress());  
            break;  
    }  
  
    @Override  
    public int getItemCount() {  
        return mObjects.size();  
    }  
}
```

- Sau khi đã tạo xong CustomAdapter thì thêm RecyclerView vào **activity\_user.xml** và cài đặt cho RecyclerView trong **UserActivity.java** như sau:

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/rv_multipe_view_type"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginStart="16dp"  
    android:layout_marginTop="16dp"  
    android:layout_marginEnd="16dp"  
    android:layout_marginBottom="16dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

- Sau khi đã tạo xong CustomAdapter thì thêm RecyclerView vào **activity\_user.xml** và cài đặt cho RecyclerView trong **UserActivity.java** như sau:
- Mỗi phần tử trong danh sách đều có kiểu Object từ đó có thể thêm bất cứ kiểu dữ liệu vào danh sách này (tất cả đối tượng trong Java đều kế thừa từ Object).

```
public class UserActivity extends AppCompatActivity {  
    private static final String TAG = "MainActivity";  
    private RecyclerView rvMultipleViewType;  
    private List<Object> mData;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_user);  
        rvMultipleViewType = (RecyclerView) findViewById(R.id.rv_multiple_view_type);  
        mData = new ArrayList<>();  
        mData.add(new UserModel( name: "Nguyen Van Nghia", address: "Quan 11"));  
        mData.add(R.drawable.avatar_1);  
        mData.add("Text 0");  
        mData.add("Text 1");  
        mData.add(new UserModel( name: "Nguyen Hoang Minh", address: "Quan 3"));  
        mData.add("Text 2");  
        mData.add(R.drawable.avatar_2);  
        mData.add(R.drawable.avatar_3);  
        mData.add(new UserModel( name: "Pham Nguyen Tam Phu", address: "Quan 10"));  
        mData.add("Text 3");  
        mData.add("Text 4");  
        mData.add(new UserModel( name: "Tran Van Phuc", address: "Quan 1"));  
        mData.add(R.drawable.avatar_4);  
        CustomAdapter adapter = new CustomAdapter( context: this, mData);  
        rvMultipleViewType.setAdapter(adapter);  
        rvMultipleViewType.setLayoutManager(new LinearLayoutManager( context: this));  
    }  
}
```



- ItemAnimator là thành phần hỗ trợ animation khi thêm vào hay xóa 1 item ra khỏi RecyclerView có các lớp chính sau:
  - ItemAnimator: là lớp đại diện, khung sườn của animation trong RecyclerView.
  - SimpleItemAnimator: lớp wrapper lại ItemAnimator.
  - DefaultItemAnimator: lớp xử lý animation mặc định, sử dụng trong RecyclerView.
- Viết lại lớp xử lý animation bằng cách kế thừa lại lớp SimpleItemAnimator.
- Để đặt ItemAnimator cho ReyclerView, sử dụng phương thức **setItemAnimator(ItemAnimator)**.
- Tuy nhiên, phải tạo custom adapter và gọi thông báo đúng trường hợp, ứng với các thao tác thêm và xóa item.

- Thông báo Adapter xử lý animation: Không giống như ListView khi có thay đổi về mặt dữ liệu thì phải gọi notifyDataSetChanged để hiển thị lại ListView, RecyclerView có 1 số điểm khác như sau:
  - Gọi notify tại vị trí thay đổi bằng các phương thức:

<b>notifyItemChanged(int)</b>	Notify item tại vị trí position nếu có thay đổi, phương thức này cũng được sử dụng nếu muốn thực hiện animation khi item changed.
<b>notifyItemInserted(int)</b>	Notify nếu insert 1 item tại vị trí position, phương thức này cũng được sử dụng nếu muốn thực hiện animation khi add 1 item vào RecyclerView.
<b>notifyItemRemoved(int)</b>	Notify khi remove 1 item tại vị trí position, phương thức này cũng được sử dụng nếu muốn thực hiện animation khi remove 1 item ra khỏi RecyclerView.
<b>notifyItemRangeChanged(int, int)</b>	Notify những item thay đổi trên 1 miền từ fromPosition, phương thức này cũng được sử dụng nếu muốn xử lý animation những item changed.
<b>notifyItemRangeInserted(int, int)</b>	Notify các item được insert vào từ fromPosition, phương thức này cũng được sử dụng nếu muốn xử lý animation những item được insert.
<b>notifyItemRangeRemoved(int, int)</b>	Notify các item remove ra khỏi RecyclerView từ fromPosition, phương thức này cũng được sử dụng nếu muốn xử lý animation những item được removed.

- ❑ Ngoài ra RecyclerView cũng có phương thức notifyDataSetChanged() giống như ListView, tuy nhiên phương thức này sẽ không xảy ra animation.

Ví dụ khi thêm 1 item vào Adapter:

```
public void addItem(String item) {  
    mDatas.add(item);  
    // Notify tại vị trí add item.  
    notifyItemInserted(mDatas.size() - 1);  
}
```

Hoặc:

```
public void addItem(int position, String item) {  
    mDatas.add(position, item);  
    notifyItemInserted(position);  
}
```

## Xóa 1 item:

```
public void removeItem(int position) {  
    mDatas.remove(position);  
    notifyItemRemoved(position);  
}
```

Hoặc:

```
public void removeItem(String item) {  
    int index = mDatas.indexOf(item);  
    if (index < 0)  
        return;  
    mDatas.remove(index);  
    notifyItemRemoved(index);  
}
```

Đổi 1 item:

```
public void replaceItem(int position, String item) {  
    mDatas.remove(position);  
    mDatas.add(position, item);  
    notifyItemChanged(position);  
}
```

Việc sử dụng ItemAnimator:

- Đặt ItemAnimator cho RecyclerView thông qua phương thức setItemAnimator()
- Viết custom adapter sử dụng những phương thức thông báo đã giới thiệu ở trên.

## Ứng dụng demo

Tổng quan về ứng dụng như sau:

- 1 button dùng để thực hiện thêm item vào adapter và có animation.
- Khi nhấp chuột vào item thì thực hiện thay đổi animation trên item đó.
- Khi nhấp giữ item thì thực hiện xóa animation trên item đó.

## Ứng dụng demo

Tổng quan về ứng dụng như sau:

- 1 button dùng để thực hiện thêm item vào adapter và có animation.
- Khi nhấp chuột vào item thì thực hiện thay đổi animation trên item đó.
- Khi nhấp giữ item thì thực hiện xóa animation trên item đó.

```
<Button  
    android:id="@+id/btn_add_item"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="112dp"  
    android:text="Button"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/rv_items"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="64dp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/btn_add_item" />
```

activity\_animation\_main.xml

## row\_animation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp"
        android:background="#ecf0f1">
    <TextView
        android:id="@+id/tv_item"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp"
        android:textColor="#2c3e50"
        android:textSize="16dp" />
</FrameLayout>
```

## CustomAnimationAdapter.java

```
public class CustomAnimationAdapter extends RecyclerView.Adapter<CustomAnimationAdapter.ViewHolder> {
    private List<String> mDatas;
    public CustomAnimationAdapter(List<String> data) {
        mDatas = data;
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflator li = LayoutInflator.from(parent.getContext());
        View itemView = li.inflate(R.layout.row_animation, parent, attachToRoot: false);
        return new ViewHolder(itemView);
    }
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        String item = mDatas.get(position);
        holder.tvItem.setText(item);
    }
    @Override
    public int getItemCount() {
        return mDatas.size();
    }
    public void addItem(String item) {
        mDatas.add(item);
        notifyDataSetChanged( position: mDatas.size() - 1 );
    }
    public void addItem(int position, String item) {
        mDatas.add(position, item);
        notifyDataSetChanged(position);
    }
}
```

## CustomAnimationAdapter.java

```
public void removeItem(int position) {  
    mDatas.remove(position);  
    notifyItemRemoved(position);  
}  
  
public void removeItem(String item) {  
    int index = mDatas.indexOf(item);  
    if (index < 0)  
        return;  
    mDatas.remove(index);  
    notifyItemRemoved(index);  
}  
  
public void replaceItem(int postion, String item) {  
    mDatas.remove(postion);  
    mDatas.add(postion, item);  
    notifyItemChanged(postion);  
}
```

## CustomAnimationAdapter.java

```
public class ViewHolder extends RecyclerView.ViewHolder {  
    private TextView tvItem;  
    public ViewHolder(final View itemView) {  
        super(itemView);  
        tvItem = (TextView) itemView.findViewById(R.id.tv_item);  
        itemView.setOnLongClickListener(new View.OnLongClickListener() {  
            @Override  
            public boolean onLongClick(View view) {  
                removeItem(getAdapterPosition());  
                Toast.makeText(itemView.getContext(), text: "Removed Animation", Toast.LENGTH_SHORT).show();  
                return false;  
            }  
        });  
        itemView.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                replaceItem(getAdapterPosition(), item: "item changed");  
                Toast.makeText(itemView.getContext(), text: "Changed Animation", Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

## MainAnimationActivity.java

```
public class MainAnimationActivity extends AppCompatActivity {
    private Button btnAddItem;
    private RecyclerView rvItems;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_animation);
        btnAddItem = (Button) findViewById(R.id.btn_add_item);
        rvItems = (RecyclerView) findViewById(R.id.rv_items);
        List<String> data = new ArrayList<>();
        for (int i = 0; i < 5; i++) {
            data.add("item " + i);
        }
        final CustomAnimationAdapter adapter = new CustomAnimationAdapter(data);
        rvItems.setAdapter(adapter);
        rvItems.setLayoutManager(new LinearLayoutManager(context: this));
        //set ItemAnimator for RecyclerView
        rvItems.setItemAnimator(new DefaultItemAnimator());
        btnAddItem.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                adapter.addItem("new item");
                rvItems.scrollToPosition(adapter.getItemCount() - 1);
            }
        });
    }
}
```

Vì sử dụng DefaultAnimator nên animation sẽ như sau:

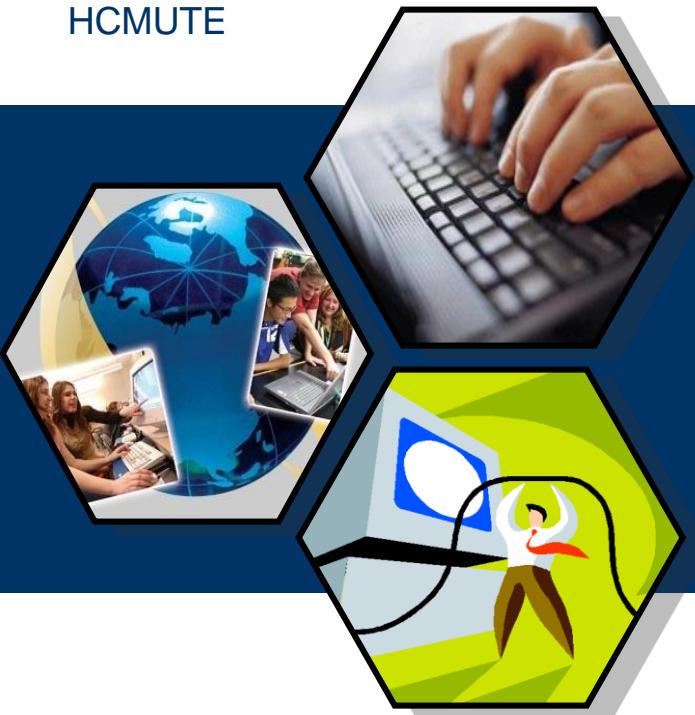
- Animation Added: thực hiện thay đổi giá trị alpha của itemView từ 0 đến 1.
- Animation Removed: thực hiện thay đổi giá trị alpha của itemView từ 1 về 0.
- Animation Changed: thực hiện animation từ 1 về 0, sau đó đặt animation từ 0 đến 1 để thay đổi item.

Đó là cách mà DefaultItemAnimator xử lý, nếu muốn animation theo mong muốn thì phải viết lại animation bằng cách kế thừa từ SimpleItemAnimator.



KHOA CÔNG NGHỆ THÔNG TIN

UTEx



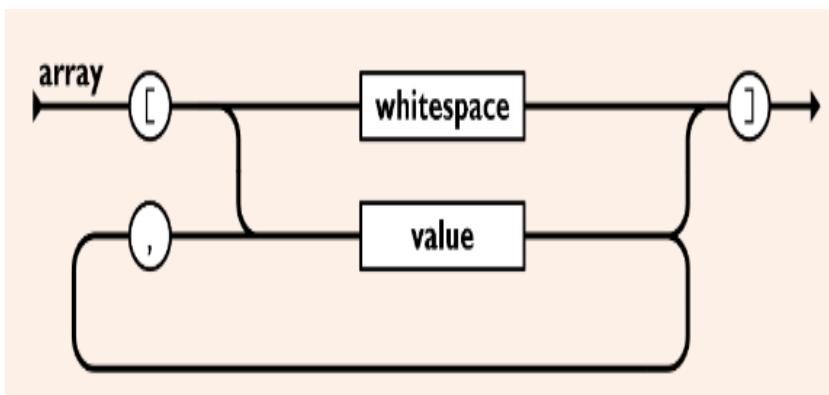
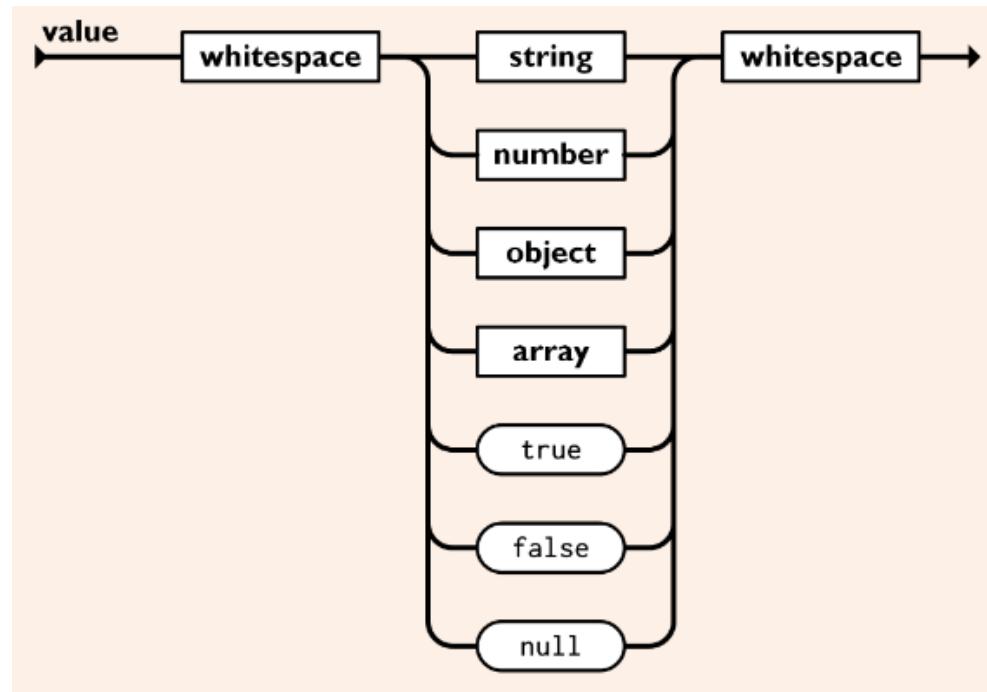
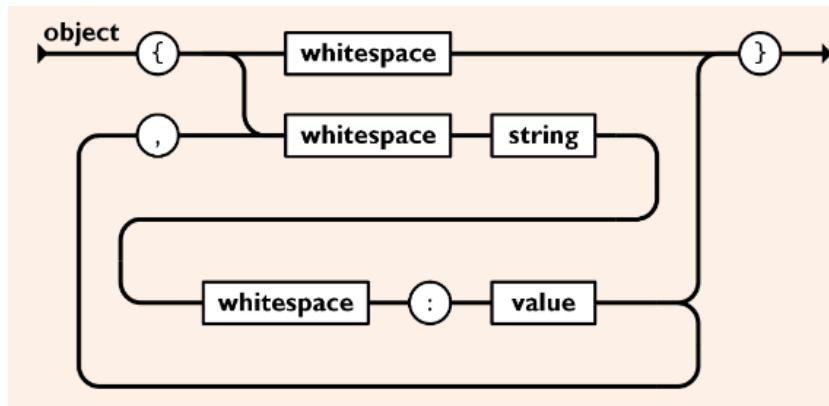
## Kết nối Android với JSON

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

ThS. Nguyễn Hữu Trung



- Truy cập trang json.org xem cấu trúc JSON (Javascript Object Notation)



- Tìm json formatter & Validator

**JSON Data/URL**

```
{"name" : "Java" ,  
 "desc" : "Java 1"}
```

**Formatted JSON Data**

```
{  
    "name" : "Java" ,  
    "desc" : "Java 1"  
}
```

Tải file data.json về bỏ vào web

← → ⌛ Not secure | 192.168.31.214:8090/data.json

 Android - Login Scr...  Kotlin Tutorial  Setting Up Swagger...

```
{  
    "name": "Java",  
    "desc": "Java 1"  
}
```

## Network Connection Details

## Network Connection Details:

Property	Value
Connection-specific DNS S...	
Description	Realtek RTL8188EU Wireless LAN 802.11n L
Physical Address	50-3E-AA-A3-2C-61
DHCP Enabled	Yes
IPv4 Address	192.168.31.214
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Friday, January 13, 2023 6:45:45 PM
Lease Expires	Saturday, January 14, 2023 6:45:39 AM
IPv4 Default Gateway	192.168.31.1
IPv4 DHCP Server	192.168.31.1
IPv4 DNS Server	192.168.31.1
IPv4 WINS Server	
NetBIOS over Tcpip Enabl...	Yes
Link-local IPv6 Address	fe80::5b2d:3aef:36cc:2856%41
IPv6 Default Gateway	
IPv6 DNS Server	

< >

Close

- Khởi tạo dự án Empty Android. Tạo Class ReadJSON

//Tạo Class đọc Json

```
private class ReadJSONObject extends AsyncTask<String,Void,String>{  
    @Override  
    protected String doInBackground(String... strings) {  
        StringBuilder content = new StringBuilder();  
        try {  
            URL url = new URL(strings[0]);  
            InputStreamReader inputStreamReader = new  
InputStreamReader(url.openConnection().getInputStream());  
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);  
            String line = "";  
            while ((line = bufferedReader.readLine()) != null){  
                content.append(line);  
            }  
            bufferedReader.close();  
        } catch (MalformedURLException e) {  
            e.printStackTrace();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        return content.toString();  
    }  
}
```

## @Override

```
protected void onPostExecute(String s) {  
    super.onPostExecute(s);  
    //phân tích JSON  
    try {  
        JSONObject object = new JSONObject(s);  
        String name = object.getString("name");  
        String desc = object.getString("desc");  
        String pic = object.getString("pic");  
  
        String kq = name + "\n" + desc + "\n" + pic;  
        Toast.makeText(MainActivity.this,kq,Toast.LENGTH_SHORT).show();  
  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
}
```

**JSON Data/URL**

```
{  
    "monhoc": [  
        {  
            "name": "Java", "desc": "Java 1", "pic": "pic1.png"},  
        {  
            "name": "C#", "desc": "C# 1", "pic": "pic2.png"},  
        {  
            "name": "Kotlin", "desc": "Kotlin 1", "pic": "pic3.png"},  
        {  
            "name": "PHP", "desc": "PHP 1", "pic": "pic4.png"},  
        {  
            "name": "Dart", "desc": "Dart 1", "pic": "pic5.png"}  
    ]  
}
```

## JSON Data/URL

```
{  
    "monhoc": [  
        {  
            "name": "Java", "desc": "Java 1", "pic": "pic1.png"},  
        {  
            "name": "C#", "desc": "C# 1", "pic": "pic2.png"},  
        {  
            "name": "Kotlin", "desc": "Kotlin 1", "pic": "pic3.png"},  
        {  
            "name": "PHP", "desc": "PHP 1", "pic": "pic4.png"},  
        {  
            "name": "Dart", "desc": "Dart 1", "pic": "pic5.png"}  
    ]  
}
```

```
new ReadJSONObject().execute("http://192.168.31.214:8090/data3.json");
```

# Ví dụ JSON Array

232

@Override

```
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    //phân tích JSON
    try {
        JSONObject object = new JSONObject(s);
        //xử lý mảng
        JSONArray array = object.getJSONArray("monhoc");
        //đọc các phần tử trong mảng
        for(int i=0;i<array.length();i++){
            JSONObject object1 = array.getJSONObject(i);
            String name = object1.getString("name");
            String desc = object1.getString("desc");
            String pic = object1.getString("pic");
            String kq = name + "\n" + desc + "\n" + pic;
            Toast.makeText(MainActivity.this,kq,Toast.LENGTH_SHORT).show();
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

@Override

```
protected void onPostExecute(String s) {  
    super.onPostExecute(s);  
    //phân tích JSON  
    try {  
        JSONObject object = new JSONObject(s);  
        JSONObject objectLang = object.getJSONObject("language");  
        JSONObject objectVN = objectLang.getJSONObject("vn");  
        String ten = objectVN.getString("ten");  
  
        Toast.makeText(MainActivity.this,ten,Toast.LENGTH_SHORT).show();  
    }  
} catch (JSONException e) {  
    e.printStackTrace();  
}
```

# Thông tin liên lạc

234

- Nguyễn Hữu Trung
- 0908617108
- [trungh@hcmute.edu.vn](mailto:trungh@hcmute.edu.vn)
- [utex.hcmute.edu.vn](http://utex.hcmute.edu.vn)