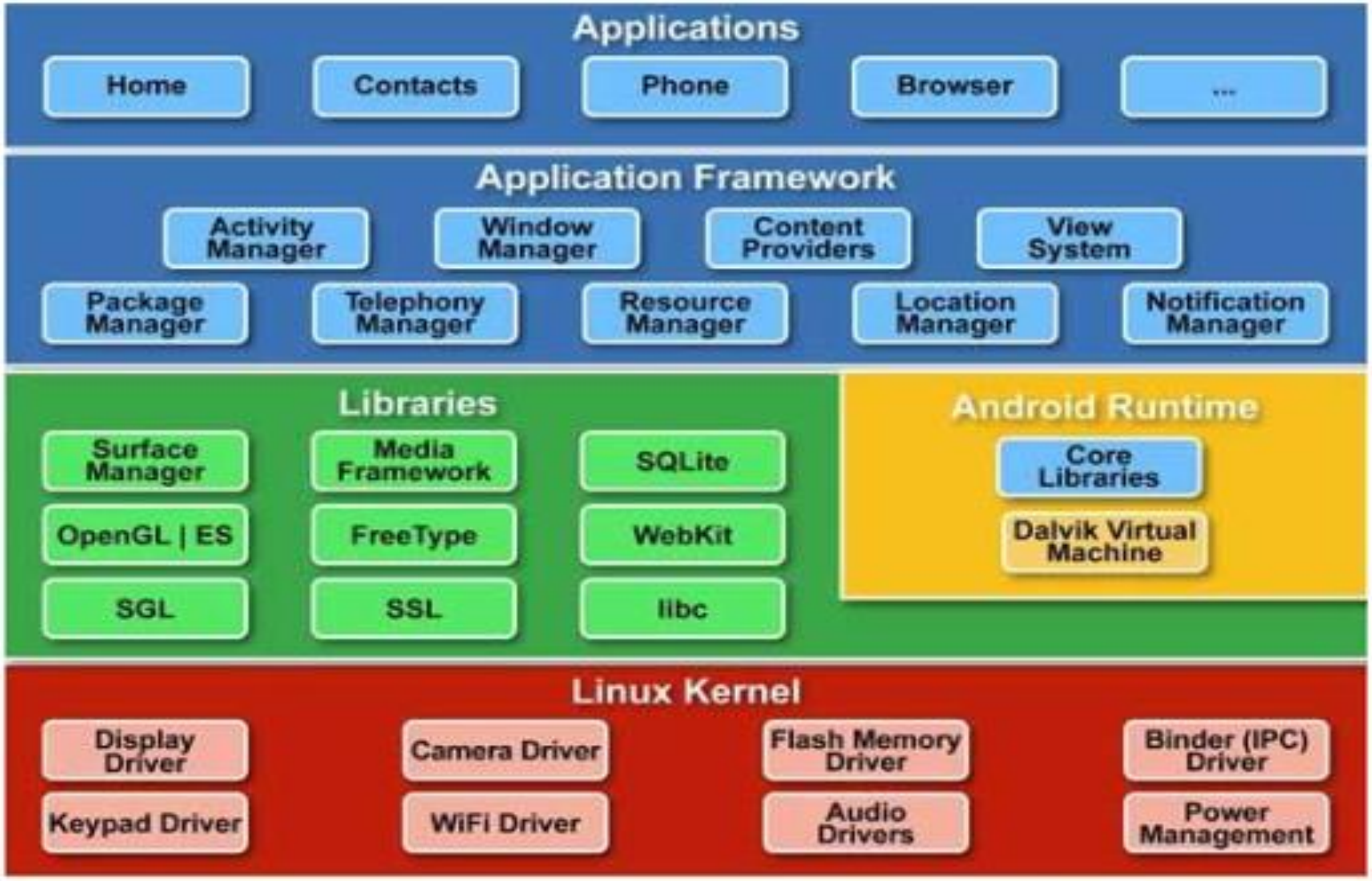
Bài 2: KIẾN TRÚC ANDROID VÀ VÒNG ĐỜI CỦA ACTIVITY

Giảng viên: Lê Quốc Anh

Nội dung

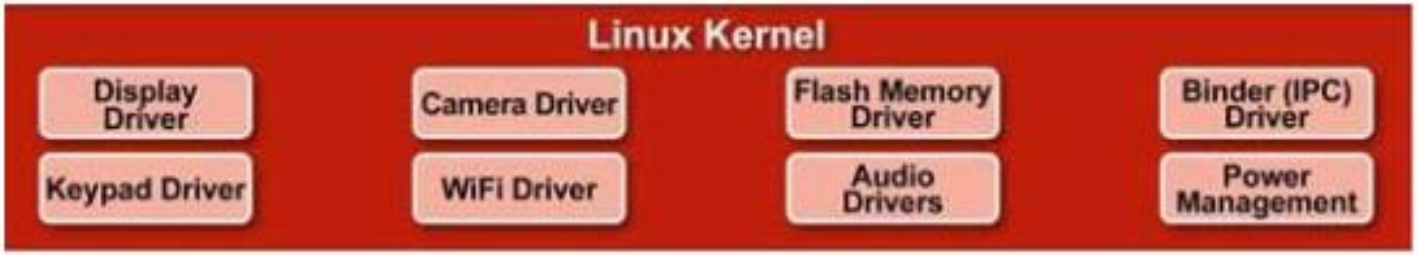
1. Kiến trúc Android
2. Các thành phần ứng dụng Android
3. Vòng đời của Activity

1. Kiến trúc Android



1. Kiến trúc Android

- **Linux kernel: (Nhân Linux)**
  - Là lớp dưới cùng và là trung tâm của kiến trúc Android.
  - Quản lý tất cả các trình điều khiển (drivers) như trình điều khiển hiển thị, trình điều khiển máy ảnh, trình điều khiển Bluetooth,… những trình điều khiển này chủ yếu được yêu cầu cho thiết bị Android trong thời gian thực hiện
  - Cung cấp lớp trừu tượng giữa phần cứng thiết bị và phần còn lại của ngăn xếp
    - Chịu trách nhiệm quản lý bộ nhớ, quản lý năng lượng, quản lý thiết bị truy cập tài nguyên,…

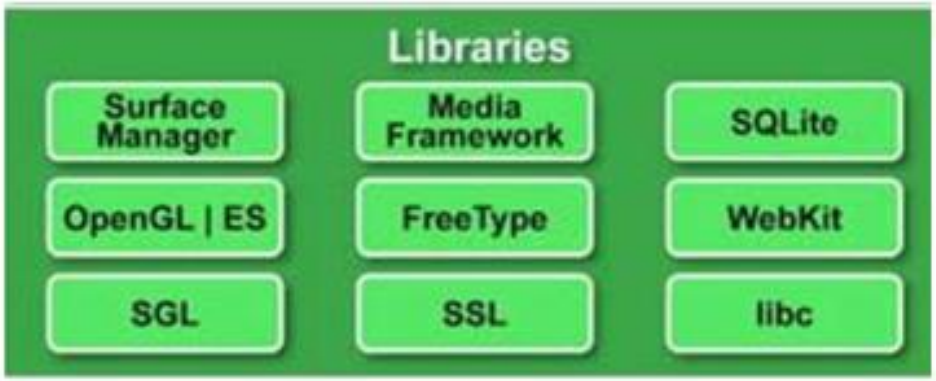


1. Kiến trúc Android

- **Android Runtime (ART)**
  - Là phần quan trọng của Android, được thiết kế để chạy các ứng dụng trong môi trường hạn chế về xử lý và bộ nhớ, pin. Bao gồm 2 thành phần:
    - Core libraries (Thư viện lõi)
    - Davik virtual Machine (Máy ảo Dalvik)
  - Là công cụ hỗ trợ các ứng dụng cùng các thư viện và tạo cơ sở nền tảng ứng dụng (Application Framework)
  - Máy ảo Davik (DVM)
    - Là máy ảo dựa gần giống như máy ảo Java (JVM)
    - Được thiết kế và tối ưu hóa đặc biệt cho Android để đảm bảo rằng thiết bị có thể chạy nhiều phiên bản một cách hiệu quả
    - Dựa vào nhân Linux để phân luồng và quản lý bộ nhớ cấp thấp
  - Core libraries: Cho phép thực hiện các ứng dụng Android bằng các ngôn như Java/ Kotlin/ C++

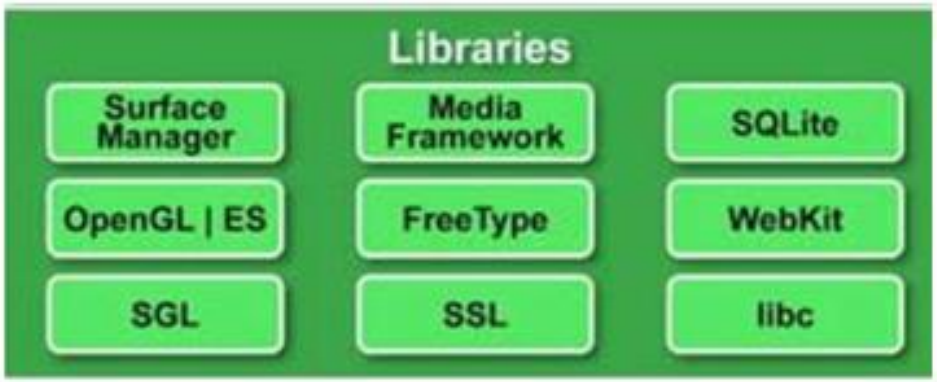
1. Kiến trúc Android

- **Libraries**
  - Gồm các thư viện lõi C/C++ và thư viện dựa trên Java như SSL, Libc, Graphics, SQLite, Webkit, Media, Surface Manager, Open GL… cung cấp hỗ trợ cho phát triển Android
  - Một số thư viện lõi (Core Android libraries)
    - Media Library
      - Chơi và ghi lại âm thanh và video
    - Surface Manager library
      - Quản lý hiển thị



1. Kiến trúc Android

- Một số thư viện lõi (Core Android libraries)
  - SGL và OpenGL Graphics libraries
    - Đồ họa 2D/3D
  - SQLite
    - Cơ sở dữ liệu
  - Web-kit
    - Hỗ trợ trình duyệt Web



1. Kiến trúc Android

- **Application Framework**
  - Nó là một tập hợp các API được viết bằng Java, cung cấp cho các nhà phát triển có quyền truy cập vào thành phần Hệ điều hành Android để tạo các ứng dụng.
  - Cung cấp sự trừu tượng chung cho việc truy cập phần cứng và quản lý giao diện người dung cũng như tài nguyên ứng dụng.
  - Bao gồm các dịch vụ có thể được sử dụng để phát triển ứng dụng:
    - Dịch vụ điện thoại, dịch vụ định vị, trình quản lý thông báo, dịch vụ NFC …

1. Kiến trúc Android

- **Applications**
  - Là lớp trên cùng của kiến trúc Android
  - Gồm các ứng dụng gốc và của bên thứ ba như danh bạ, email, nhạc, thư viện, đồng hồ, trò chơi, …
    - Gồm các ứng dụng do lập trình viên tạo ra
  - Chạy trong Android Runtime và dung các lớp, dịch vụ của Application Framework

2. Các thành phần ứng dụng Android

- Cấu tạo cơ bản một ứng dụng Android gồm các thành phần như:
  - Activity
  - Intent
  - Manifest
  - Context

Activity trong Android

- Các ứng dụng được tạo ra bằng sự hòa hợp với một hay nhiều thành phần được gọi là **Activity**.
- Một Activity có thể là một module, thành phần chức năng độc lập của ứng dụng mà mỗi Activity thường tương ứng với một giao diện người dùng (UI) và các chức năng đáp lại sự tương tác với người dùng.
- Một ứng dụng có thể có một hay nhiều Activity
- Các Activity được xây dựng để có thể sử dụng lại và tương tác, chia sẻ giữa các ứng dụng.
- Một Activity được xây dựng bằng cách kế thừa lớp **Activity**, một Activity **không thể gọi trực tiếp** các phương thức hay truy cập vào dữ liệu của một Activity khác
- Phải dùng tới thành phần gọi là **Intent** từ Activity này kích hoạt Activity khác

Intent trong Android

- Các **Intent** là cơ chế để một Activity có thể khởi chạy Activity khác trong ứng dụng.
- Trong **Intent** có thông tin về nhiệm vụ, các tùy chọn, dữ liệu để Activity thi hành được.
  - *Explicit intent*: là dạng tường minh, nghĩa là chúng yêu cầu chạy một Activity cụ thể chỉ ra bởi tên lớp của Activity đó.
  - *Implicit intent*: là dạng không tường mình, bằng cách chỉ cung cấp hoạt động mong muốn, hoặc cung cấp dữ liệu mà căn cứ vào dữ liệu đó tương ứng với hành động, hệ thống Android ở thời điểm chạy sẽ chọn Activity để kích hoạt.

Broadcast Intent

- Đây là một Intent đặc biệt, Broadcast Intent sẽ gửi thông tin ra bên ngoài, gửi đến tất cả các ứng dụng (ứng dụng có đăng ký nhận bằng Broadcast Receiver).
- **Broadcast Receiver**
  - Đây là cơ chế để ứng dụng phản ứng lại với các Broadcast Intent, nó cần được đăng ký bởi ứng dụng và cấu hình với **Intent Filter** để biết được loại Intent nào muốn nhận.

```
<activity android:name=".ExampleActivity"
android:icon="@drawable/app_icon">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
    </intent-filter>
</activity>
```

Android Services

- Android Service là process (tiến trình) chạy ngầm (background) và không có giao diện người dùng. Nó có thể khởi chạy và quản lý bởi các Activity, Broadcast Receiver, hay bởi Service khác.
- Android Service dùng trong tình huống một ứng dụng cần tiếp tục thi hành các tác vụ nhưng không cần tới sự xuất hiện của giao diện người dùng.
- Cho dù Service thiếu đi giao diện người dùng, nó vẫn có thể thông báo cho người dùng các tự kiện bằng sử dụng Notification hoặc Toast.

Content Provider

- **Content Provider** là cơ chế cho phép chia sẻ dữ liệu giữa các ứng dụng với nhau.
- Bất kỳ ứng dụng nào đều có thể cung cấp cho ứng dụng khác khả năng truy cập dữ liệu của mình thông qua Content Provider với các chức năng thêm, bớt, truy vấn dữ liệu.
- Việc truy cập này được cung cấp thông qua URI định nghĩa và cung cấp bởi Content Provicder
- Dữ liệu chia sẻ ở dạng một file hoặc CSDL SQLite. Các ứng dụng gốc Android cung cấp sẵn nhiều Content Provider cho phép ứng dụng truy cập dữ liệu như trình danh bạ, quản lý file media.

Manifest

- Manifest của ứng dụng định nghĩa trong file định dạng XML, nó mô tả để hệ thống Android hiểu khái quát về ứng dụng như các Activity, các Service, Broadcast Receiver, Content Provider, các quyền truy cập. Những thông tin này cần thiết để hệ thống Android chạy được ứng dụng như mong muốn.

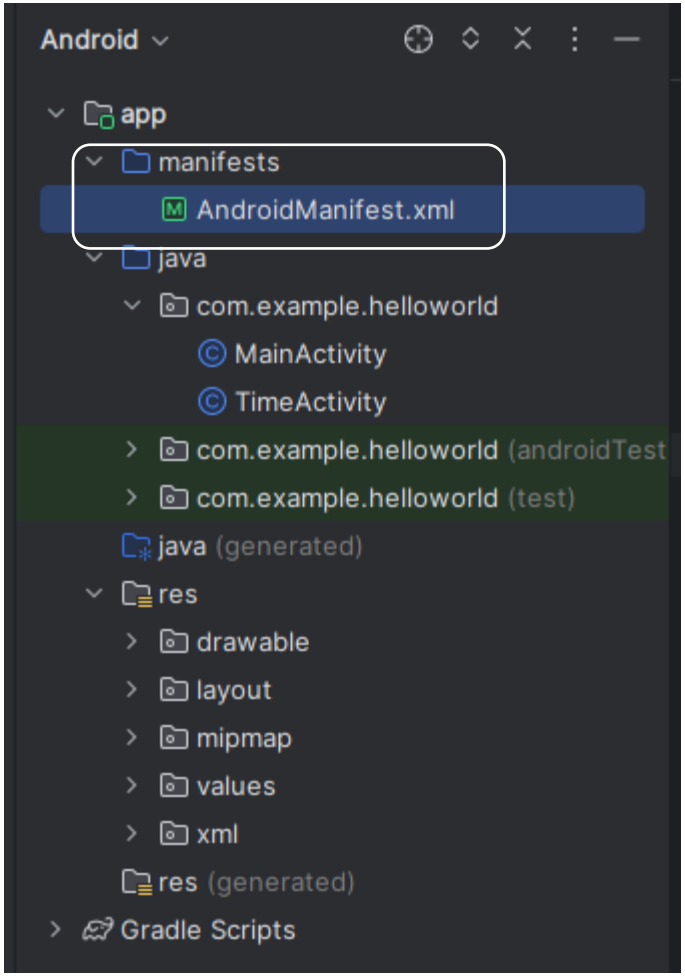
Resource

- Ứng dụng Android được đóng gói ngoài các loại file byte code, nó còn chứa tập hợp các file tài nguyên (Resource) như chuỗi ký tự, hình ảnh, font chữ, màu sắc ... những thành phần xuất hiện trong giao diện người dùng và được trình bày với file XML. Mặc định những file này lưu trữ bên trong thư mục **/res**

Context ứng dụng

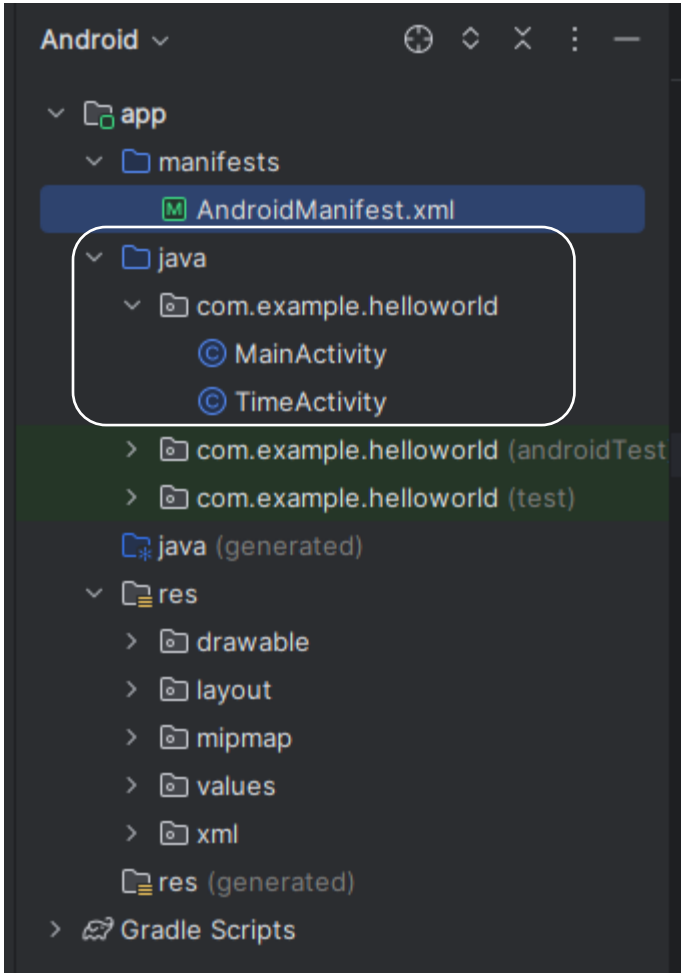
- Khi ứng dụng biên dịch, một lớp có tên là **R** được tự động tạo ra, nó chứa các tham khảo trỏ đến tài nguyên của ứng dụng.
- Các file manifest và tài nguyên sẽ được kết hợp lại với nhau được hiểu là Context ứng dụng.
- Context trong ứng dụng được biểu diễn bằng lớp Context, được sử dụng trong ứng dụng để thông qua nó truy cập tới các loại tài nguyên khi ứng dụng đang chạy.
- Ngoài ra, có nhiều phương thức dựa vào context để lấy các thông tin về môi trường ứng dụng hoạt động khi chạy.

Cấu trúc project



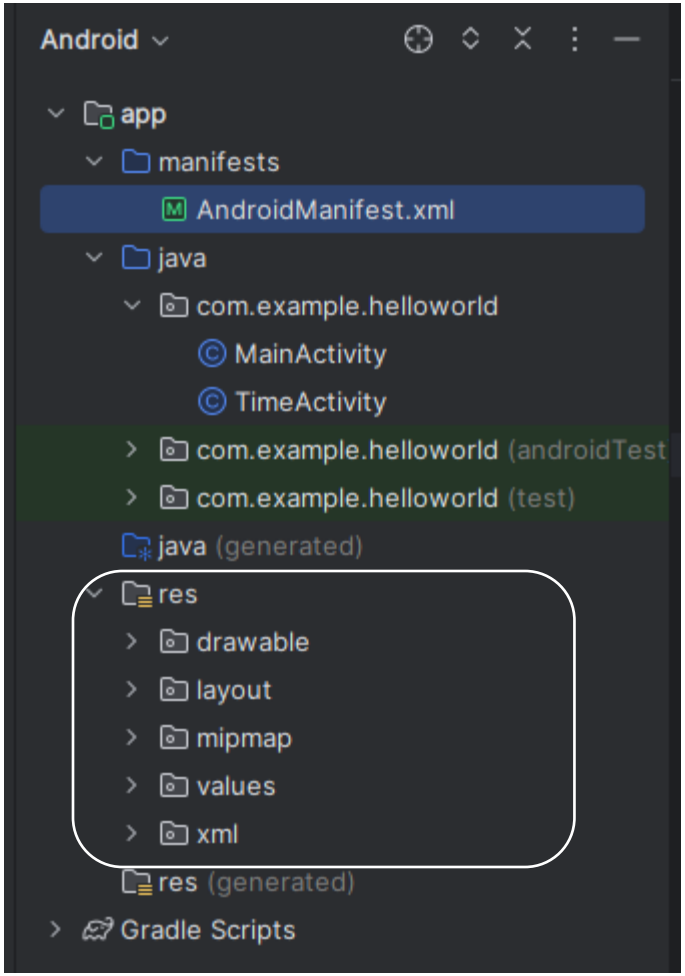
- **manifests :** tham chiếu đến file **AndroidManifest.xml**, file này để bạn mô tả cấu trúc chính của App, nhằm giúp Android OS biết được các thiết lập cơ bản của App để khởi chạy được nó, như Activiy nào sẽ được chạy, ứng dụng xin những quyền gì trong thiết bị ...

Cấu trúc project



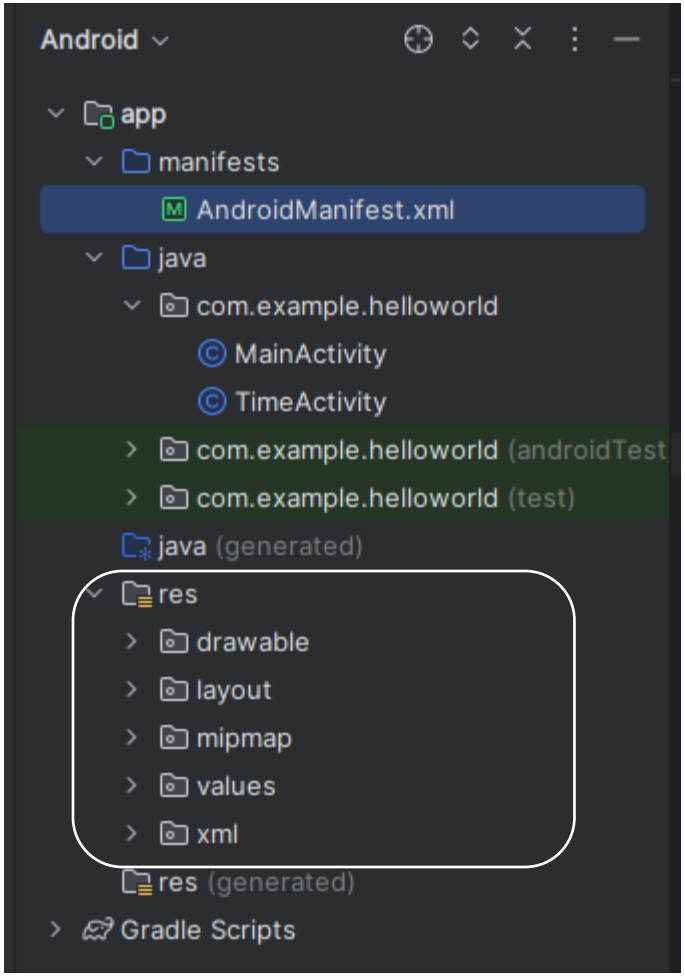
- **java :** thư mục lưu trữ các file code java của ứng dụng, lớp FirstActivity được định nghĩa với file **FirstActivity.java** lưu trong cấu trúc thự mục này

Cấu trúc project



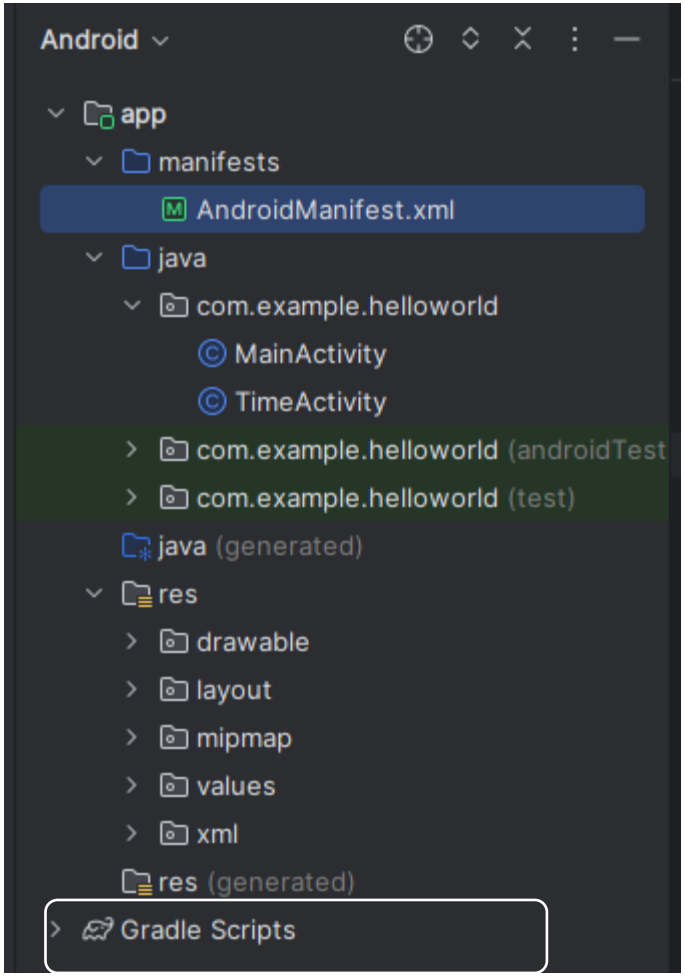
- **res** : lưu trữ các file tài nguyên mà ứng dụng sẽ sử dụng đến, nó tổ chức thành các thư mục con như:
  - **drawable** : ở đây cơ bản lưu các đối tượng đồ họa như các ảnh dạng png
  - **layout** : lưu trữ các file xml biểu diễn về thành phần, bố cục của các thành phần hiện thị được trên màn hình

Cấu trúc project



- **res** : lưu trữ các file tài nguyên mà ứng dụng sẽ sử dụng đến, nó tổ chức thành các thư mục con như:
  - **mipmap :** cũng để lưu các đối tượng hình ảnh, ví dụ icon ứng dụng ic_launcher đặt ở đây
  - **values:** chứa các file như colors.xml, dimens.xml, strings.xml, styles.xml, đây là các file xml định nghĩa các giá trị có thể sử dụng trong ứng dụng như màu sắc, kích thước, các chuỗi, các theme ...

Cấu trúc project



- **Gradle Scripts** chứa nhiều nhánh con như build.gradle , local.properties ... là nơi bạn thiết lập các thông số để Gradle build ứng dụng.

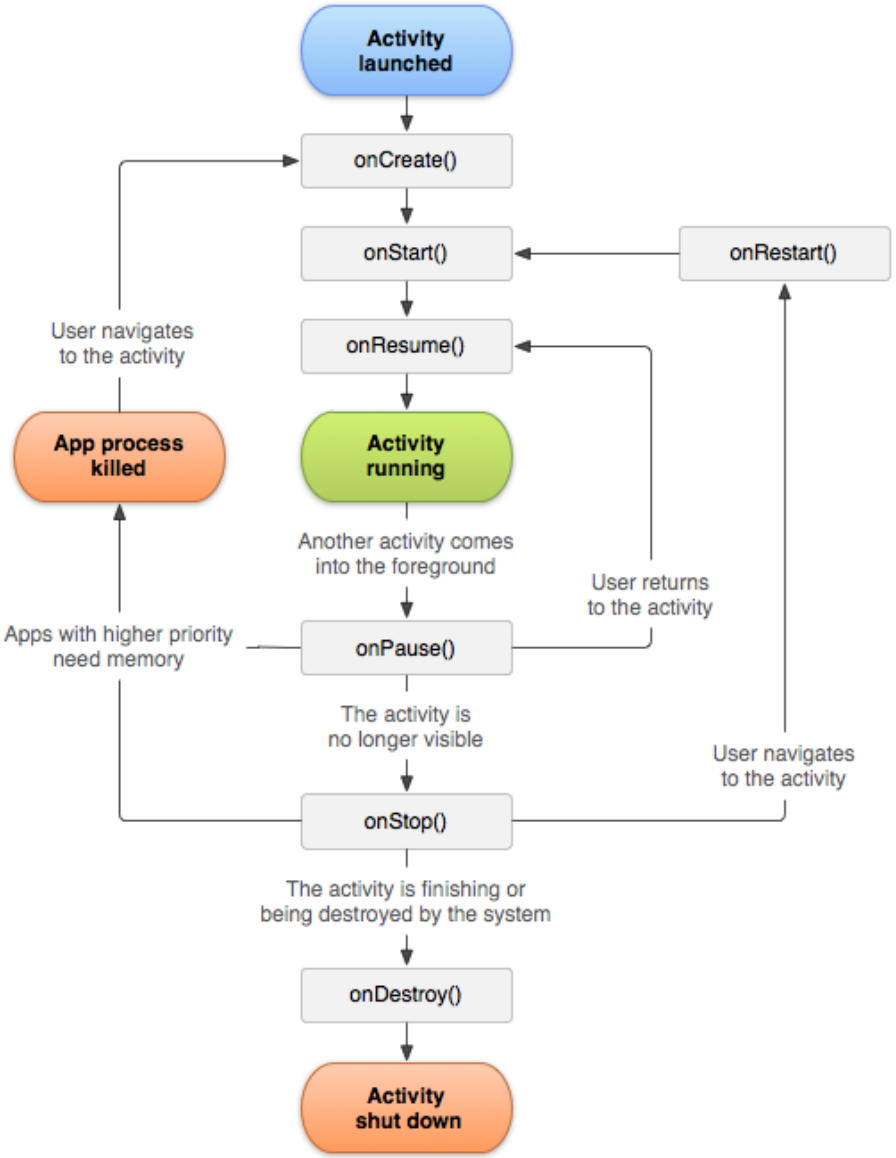
3. Vòng đời của Activity

- Lớp **Activity** là thành phần quan trọng nhất của ứng dụng Android, cách mà chúng hoạt động tạo thành nền tảng cơ bản của mô hình lập trình ứng dụng. Android khởi chạy một ứng dụng thông thường bằng kích hoạt một Activity tương ứng với vòng đời cụ thể của nó trong quá trình hoạt động.
- Thường một Activity cung cấp một của sổ, ở đó ứng dụng sẽ dựng các thành phần UI (User Interface - giao diện người dùng)
- Mặc định cửa sổ này là đầy màn hình thiết bị, nhưng có một vài trường hợp riêng sẽ nhỏ hơn và nổi phía trên cửa sổ khác.

3. Vòng đời của Activity

- Hầu hết các ứng dụng đều sử dụng nhiều màn hình khác nhau, có nghĩa nó sẽ phải có nhiều Activity khác nhau. Khi một Activity chỉ định là Activity chính, nó sẽ là màn hình đầu tiên khi khởi chạy ứng dụng. Một Activity này lại có thể gọi và kích hoạt một Activity khác.

3. Vòng đời của Activity



Một số phương thức của Activity

- onCreate()
  - Khi hoạt động mới được tạo
- onStart()
  - Được gọi ngay trước khi activity hiển thị trên màn hình
- onResume()
  - Được gọi ngay khi activity bắt đầu có thể tương tác với người dùng
- onPause()
  - Được gọi khi hệ thống sắp kích hoạt một activity khác
- onStop()
  - Được gọi khi activity bị ẩn đi

- onRestart()
  - Được gọi khi người dung kích hoạt lại Activity bị ẩn
- onDestroy()
  - gọi khi Activity bị hủy hoàn toàn (ví dụ gọi finish(), hoặc người dùng kill Activity)

THANK YOU for YOUR ATTENTION