

**TRƯỜNG ĐẠI HỌC THỦY LỢI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



# **GIÁO TRÌNH**

## **THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

Hà Nội, 2.2025

# MỤC LỤC

CHƯƠNG 1. Làm quen.....	3
Bài 1) Tạo ứng dụng đầu tiên .....	3
1.1) Android Studio và Hello World .....	3
1.2) Giao diện người dùng tương tác đầu tiên .....	30
1.3) Trình chỉnh sửa bố cục .....	30
1.4) Văn bản và các chế độ cuộn .....	30
1.5) Tài nguyên có sẵn.....	30
Bài 2) Activities .....	30
2.1) Activity và Intent .....	30
2.2) Vòng đời của Activity và trạng thái .....	30
2.3) Intent ngầm định.....	30
Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ .....	30
3.1) Trình gỡ lỗi .....	30
3.2) Kiểm thử đơn vị.....	30
3.3) Thư viện hỗ trợ.....	30
CHƯƠNG 2. Trải nghiệm người dùng .....	31
Bài 1) Tương tác người dùng .....	31
1.1) Hình ảnh có thể chọn .....	31
1.2) Các điều khiển nhập liệu .....	31
1.3) Menu và bộ chọn .....	31
1.4) Điều hướng người dùng .....	31
1.5) RecyclerView .....	31
Bài 2) Trải nghiệm người dùng thú vị.....	31
2.1) Hình vẽ, định kiểu và chủ đề .....	31
2.2) Thẻ và màu sắc .....	31

2.3)	Bố cục thích ứng.....	31
Bài 3)	Kiểm thử giao diện người dùng.....	31
3.1)	Espresso cho việc kiểm tra UI .....	31
CHƯƠNG 3. Làm việc trong nền .....		31
Bài 1)	Các tác vụ nền.....	31
1.1)	AsyncTask .....	31
1.2)	AsyncTask và AsyncTaskLoader .....	31
1.3)	Broadcast receivers .....	31
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	31
2.1)	Thông báo .....	31
2.2)	Trình quản lý cảnh báo .....	31
2.3)	JobScheduler.....	31
CHƯƠNG 4. Lưu dữ liệu người dùng .....		32
Bài 1)	Tùy chọn và cài đặt.....	32
1.1)	Shared preferences.....	32
1.2)	Cài đặt ứng dụng.....	32
Bài 2)	Lưu trữ dữ liệu với Room .....	32
2.1)	Room, LiveData và ViewModel.....	32
2.2)	Room, LiveData và ViewModel.....	32
3.1)	Trình gỡ lỗi .....	

## CHƯƠNG 1. LÀM QUEN

### Bài 1) Tạo ứng dụng đầu tiên

#### 1.1) Android Studio và Hello World

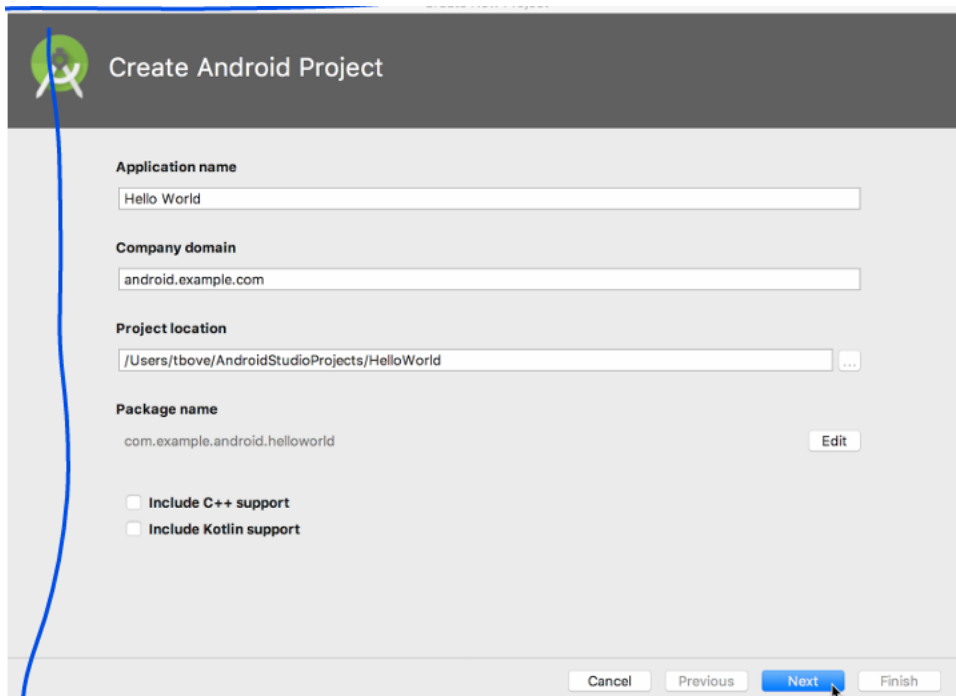
### Giới thiệu

Trong bài thực hành này, bạn sẽ tìm hiểu cách cài đặt Android Studio, môi trường phát triển Android. Bạn cũng sẽ tạo và chạy ứng dụng Android đầu tiên của mình, Hello World, trên một trình giả lập và trên một thiết bị vật lý.

## Những gì Bạn nên biết

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng lập trình hướng đối tượng sử dụng một IDE (môi trường phát triển tích hợp) như Android Studio.
- Chứng minh rằng bạn có ít nhất 1-3 năm kinh nghiệm trong lập trình hướng đối tượng, với một phần trong số đó tập trung vào ngôn ngữ lập trình Java. (Các bài thực hành này sẽ không giải thích về lập trình hướng đối tượng hoặc ngôn ngữ Java.



## Những gì Bạn sẽ cần:

- Một máy tính chạy Windows hoặc Linux, hoặc một Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một phương pháp thay thế để tải các cài đặt mới nhất của Android Studio và Java lên máy tính của bạn.

## Những gì bạn sẽ học

- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.
- Cách thêm thông điệp ghi lại vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

## Những gì bạn sẽ làm

- Cài đặt môi trường phát triển **Android Studio**.
- Tạo một trình giả lập (thiết bị ảo) để chạy ứng dụng của bạn trên máy tính.
- Tạo và chạy ứng dụng **Hello World** trên các thiết bị ảo và vật lý.
- Khám phá cấu trúc dự án.
- Tạo và xem các thông điệp ghi lại từ ứng dụng của bạn.
- Khám phá tệp **AndroidManifest.xml**

## Tổng quan về ứng dụng

Sau khi cài đặt thành công Android Studio, bạn sẽ tạo một dự án mới cho ứng dụng Hello World từ một mẫu. Ứng dụng đơn giản này hiển thị chuỗi "Hello World" trên màn hình của thiết bị Android ảo hoặc thiết bị vật lý.

Đây là ứng dụng hoàn thành sẽ trông như thế nào:

## Nhiệm vụ 1: Cài đặt Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp (IDE) hoàn chỉnh bao gồm trình chỉnh sửa mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ để phát triển, gỡ lỗi, thử nghiệm và hiệu suất giúp phát triển ứng dụng nhanh hơn và dễ dàng hơn. Bạn có thể kiểm tra ứng dụng của mình bằng nhiều trình mô phỏng được định cấu hình sẵn hoặc trên thiết bị di động của riêng mình, tạo ứng dụng chính thức và phát hành trên cửa hàng Google Play.

Lưu ý: Android Studio liên tục được cải thiện. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, hãy xem Android Studio .

Android Studio có sẵn cho máy tính chạy Windows hoặc Linux và Mac chạy macOS. OpenJDK (Java Development Kit) mới nhất được đi kèm với Android Studio. Để thiết lập và chạy Android Studio, trước tiên hãy kiểm tra các yêu cầu hệ thống để đảm bảo hệ thống của bạn đáp ứng các yêu cầu đó. Việc cài đặt tương tự cho tất cả các nền tảng. Bất kỳ sự khác biệt nào được lưu ý bên dưới

1. Điều hướng đến trang web dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio
2. Chấp nhận cấu hình mặc định cho tất cả các bước và đảm bảo rằng tất cả các thành phần được chọn để cài đặt
3. Sau khi hoàn tất cài đặt, Trình hướng dẫn cài đặt sẽ tải xuống và cài đặt một số thành phần bổ sung bao gồm SDK Android. Hãy kiên nhẫn, quá trình này có thể mất một chút thời gian tùy thuộc vào tốc độ Internet của bạn và một số bước có vẻ dư thừa
4. Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động và bạn đã sẵn sàng tạo dự án đầu tiên của mình.

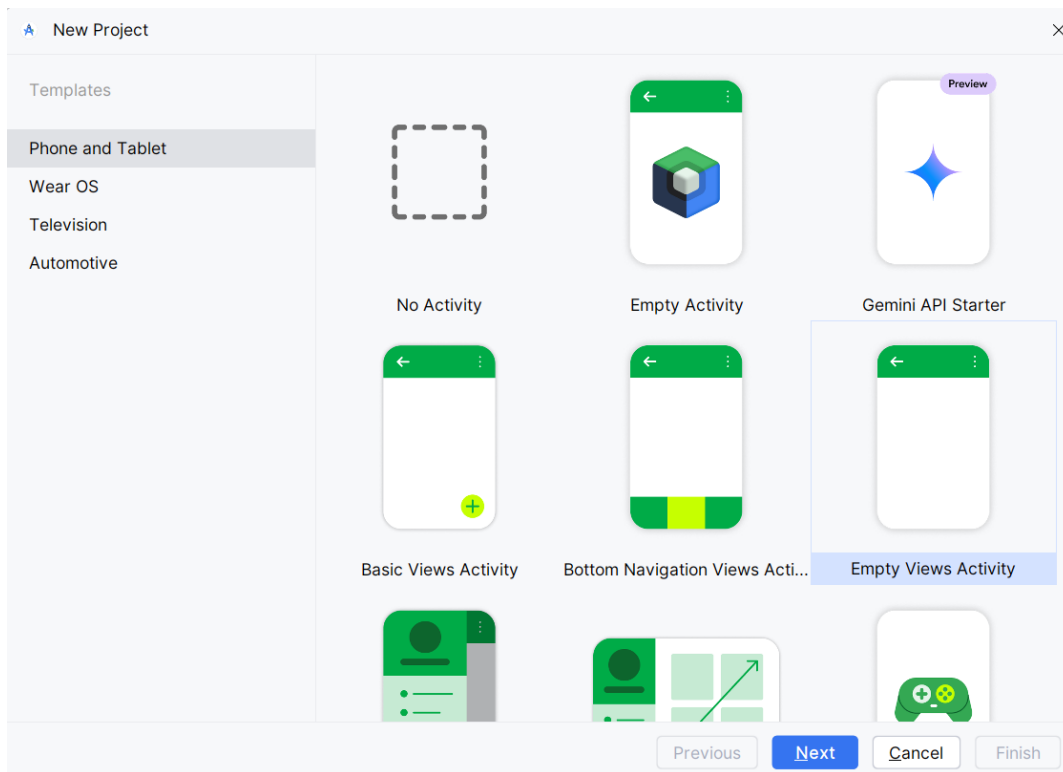
Khắc phục sự cố: Nếu bạn gặp sự cố khi cài đặt, hãy xem ghi chú phát hành Android Studio hoặc nhờ người hướng dẫn trợ giúp.

## **Nhiệm vụ 2: Tạo ứng dụng Hello Word**

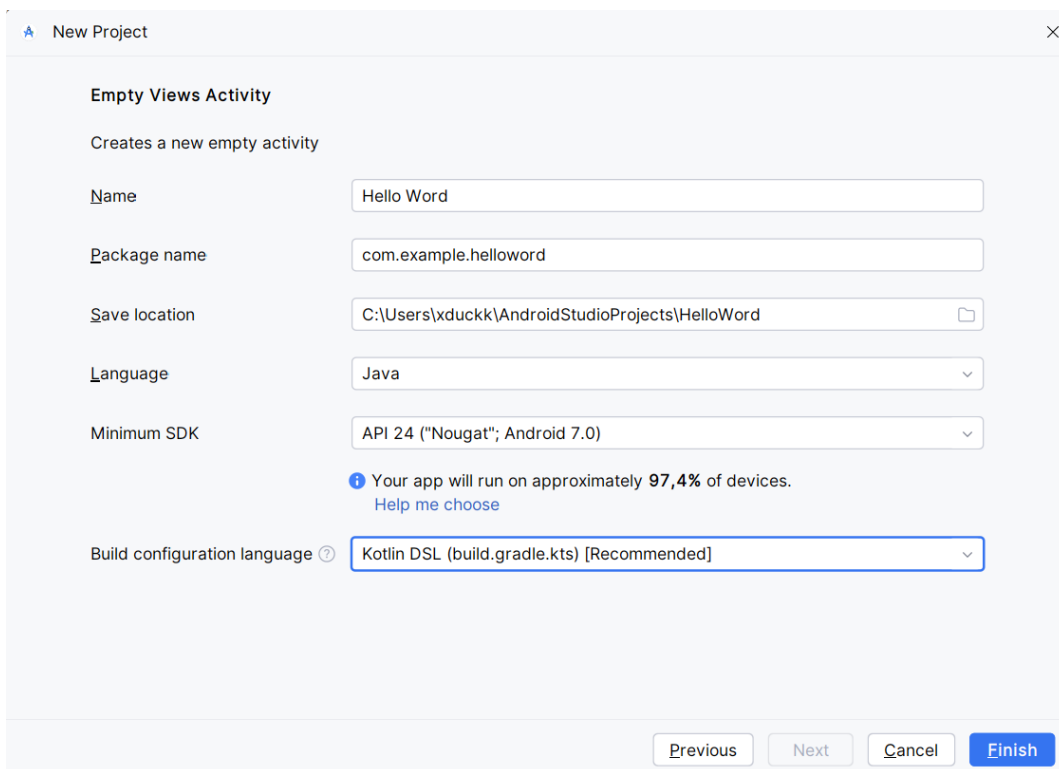
Trong tác vụ này, bạn sẽ tạo một ứng dụng hiển thị “Hello Word” để xác minh rằng Android Studio đã được cài đặt chính xác và tìm hiểu những kiến thức cơ bản về phát triển bằng Android Studio.

### **2.1 Tạo dự án ứng dụng**

1. Mở Android Studio nếu chưa mở.
2. Trong cửa sổ chính New Project, nhấp vào Empty Views Activity.



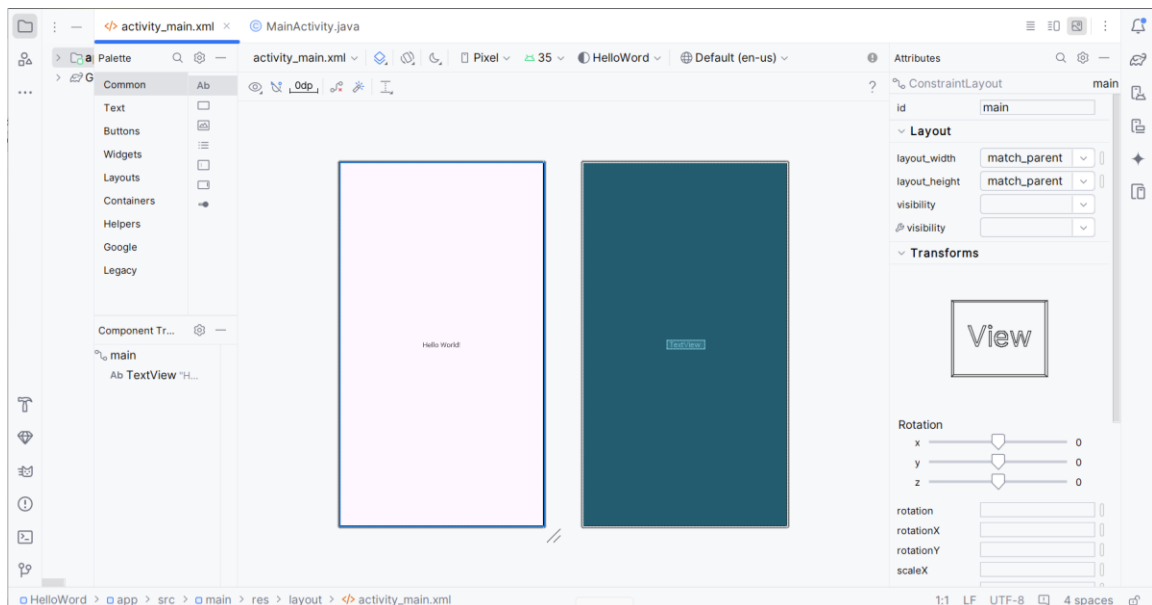
3. Trong cửa sổ Create Android Project, nhập Hello World cho tên ứng dụng.



4. Xác minh rằng Project location mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác hoặc thay đổi vị trí đó thành thư mục ưa thích của bạn.
5. Chấp nhận com.example.helloworld mặc định cho Package name.
6. Ngôn ngữ sử dụng là java.
7. Chọn API 24 (“Nougat”, Android 7.0) là Minimum SDK, cài đặt này làm cho ứng dụng Hello World của bạn tương thích với 97.4% thiết bị Android đang hoạt động trên Google Play Store.
8. Chọn Kotlin DSL làm ngôn ngữ cấu hình được xây dựng.
9. Chọn Finish để tạo.

Trình chỉnh sửa Android Studio sẽ xuất hiện. Làm theo các bước sau:

1. Nhấp vào tab activity\_main.xml để xem trình chỉnh sửa bố cục
2. Nhấp vào tab Design, nếu chưa chọn, biểu tượng đồ họa của bố cục hiển thị như hình dưới



3. Nhấp vào tab MainActivity.java để xem trình sửa code như hình dưới

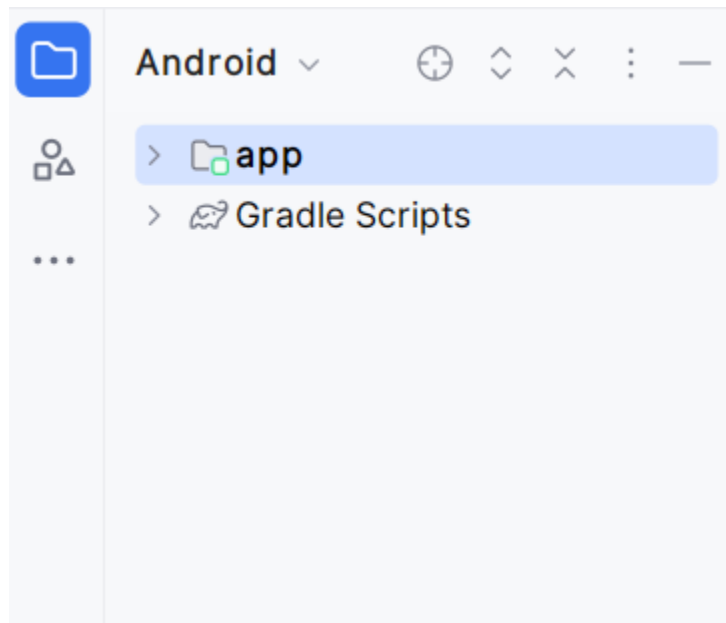




## 2.2. Khám phá khung Project > Android

Trong thực tế này, bạn sẽ khám phá cách tổ chức dự án trong Android Studio

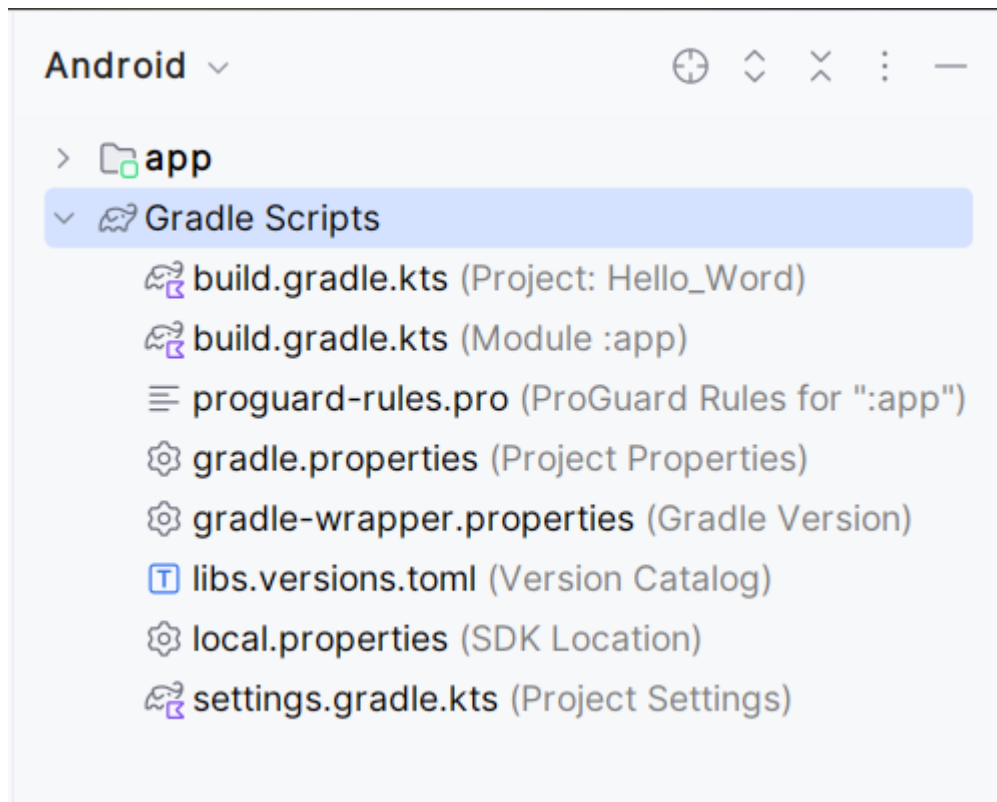
1. Nếu chưa chọn, hãy nhấp vào tab Project trong cột tab dọc ở phía bên trái của cửa sổ Android Studio. Khung Project xuất hiện
2. Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, hãy chọn Android từ menu bật lên ở đầu khung Project, như hình bên dưới



## 2.3. Khám phá thư mục Gradle Scripts

Hệ thống bản dựng Gradle trong Android Studio giúp bạn dễ dàng đưa các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng dưới dạng phần phụ thuộc

Khi bạn tạo dự án ứng dụng lần đầu tiên, khung Project > Android sẽ xuất hiện với thư mục Gradle được mở rộng như hình minh họa bên dưới



Làm theo các bước sau để khám phá hệ thống Gradle:

1. Nếu thư mục Gradle Scripts không được mở rộng, hãy nhấp vào hình tam giác để mở rộng. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng
2. Tìm tệp build.gradle.kts(Project:HelloWorld)

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô-đun tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một tệp bản dựng Gradle cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần thực hiện bất kỳ thay đổi nào đối với tệp này, nhưng nó vẫn hữu ích để hiểu nội dung của nó

Theo mặc định, tệp bản dựng cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ Gradle và các thành phần phụ thuộc chung cho tất cả các mô-đun trong dự án. Khi phần phụ thuộc của bạn không phải là thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong bất kỳ kho lưu trữ trực tuyến nào được chỉ định trong khối kho lưu trữ của tệp này

```
1  plugins {  
2      alias(libs.plugins.android.application)  
3  }
```

### 3. Tìm tệp build.gradle.kts(Module:app)

Ngoài tệp build.gradle.kts cấp độ dự án, mỗi mô-đun cũng có một build.gradle.kts riêng, cho phép bạn cấu hình các cài đặt build cho từng mô-đun cụ thể (trong ứng dụng HelloWorld, chỉ có một mô-đun duy nhất). Việc cấu hình các cài đặt build này giúp bạn tùy chỉnh các tùy chọn đóng gói, chẳng hạn như bổ sung kiểu build và biến thể sản phẩm. Ngoài ra bạn cũng có thể ghi đè các thiết lập trong tệp AndroidManifest.xml hoặc tệp build.gradle ở cấp dự án. Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phần phụ thuộc trong phần phần phụ thuộc. Bạn có thể khai báo phần phụ thuộc thư viện bằng cách sử dụng một trong một số cấu hình phần phụ thuộc khác nhau. Mỗi cấu hình phần phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ: câu lệnh thực hiện fileTree (dir: 'libs', include: ['\*.jar']) thêm phần phụ thuộc của tất cả các tệp ".jar" bên trong thư mục libs

Sau đây là tệp build.gradle.kts(Module:app):

```

1  plugins {
2      alias(libs.plugins.android.application)
3  }
4
5  android {
6      namespace = "com.example.helloworld"
7      compileSdk = 35
8
9      defaultConfig {
10         applicationId = "com.example.helloworld"
11         minSdk = 24
12         targetSdk = 35
13         versionCode = 1
14         versionName = "1.0"
15
16         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes {
20         release {
21             isMinifyEnabled = false
22             proguardFiles(
23                 getDefaultProguardFile("name: "proguard-android-optimize.txt"),
24                 "proguard-rules.pro"
25             )
26         }
27     }
28     compileOptions {
29         sourceCompatibility = JavaVersion.VERSION_11
30         targetCompatibility = JavaVersion.VERSION_11
31     }
32 }
33
34 dependencies {
35
36     implementation(libs.appcompat)
37     implementation(libs.material)
38     implementation(libs.activity)
39     implementation(libs.constraintlayout)
40     testImplementation(libs.junit)
41     androidTestImplementation(libs.ext.junit)
42     androidTestImplementation(libs.espresso.core)
43 }

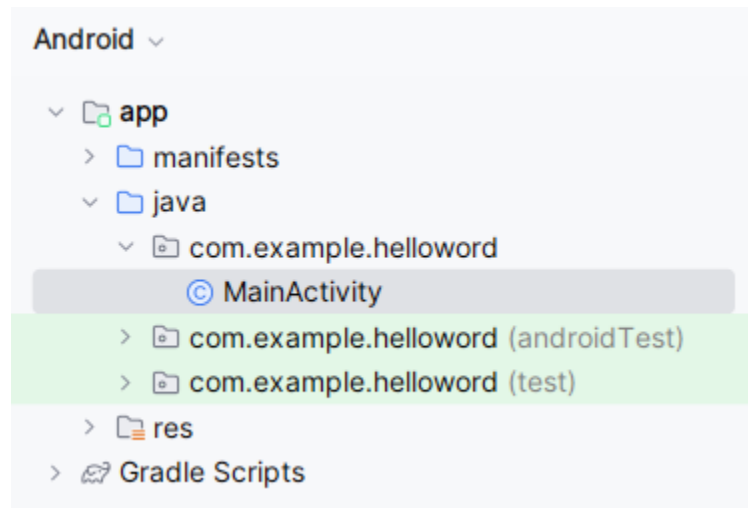
```

#### 4. Nhấp vào hình tam giác để đóng Gradle Scripts

### 2.4. Khám phá ứng dụng và thư mục res

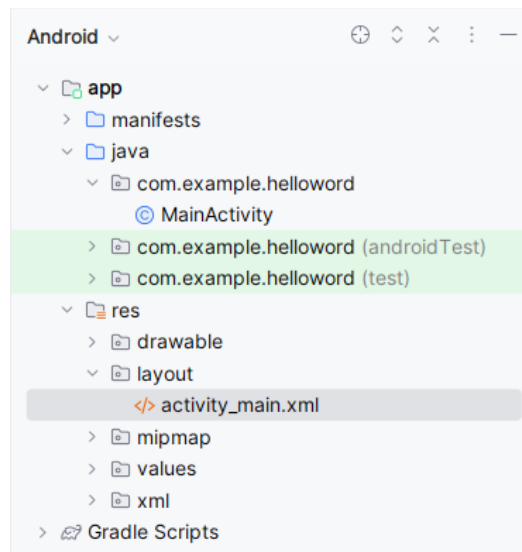
Tất cả mã nguồn và tài nguyên cho ứng dụng đều nằm trong thư mục app và res

1. Mở thư mục app, thư mục java, và thư mục com.example.android.helloworld để xem tệp java MainActivity. Nhấp đúp vào tệp để mở nó trong trình chỉnh sửa mã nguồn



Thư mục java bao gồm các tệp lớp java trong ba thư mục con, như hình trên. Thư mục com.example.android.helloworld (hoặc tên miền bạn chỉ định) chứa tất cả các tệp cho gói ứng dụng. Hai thư mục còn lại được sử dụng để kiểm tra và được đề cập trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một gói duy nhất, trong đó chứa MainActivity.java. Tên của Activity đầu tiên (màn hình đầu tiên mà người dùng nhìn thấy), đồng thời khởi tạo các tài nguyên dùng chung cho toàn bộ ứng dụng, theo thông lệ thường được đặt là MainActivity (phần mở rộng tệp được ẩn trong phần Project > Android)

2. Mở rộng thư mục res và thư mục layout đồng thời nhấp đúp vào tệp activity\_main.xml để mở nó trong trình chỉnh sửa bố cục



Thư mục res chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh. Hoạt động thường được liên kết với bố cục của các chế độ xem giao diện người dùng được xác định dưới dạng tệp XML. Tệp này thường được đặt tên theo Hoạt động của nó.

## 2.5 Khám phá thư mục kê khai

Thư mục tệp kê khai chứa các tệp cung cấp thông tin cần thiết về ứng dụng của bạn cho hệ thống Android, hệ thống phải có các tệp này trước khi có thể chạy bất kỳ mã nào của ứng dụng.

1. Mở rộng thư mục tệp kê khai.

2. Mở tệp AndroidManifest.xml.

Tệp AndroidManifest.xml mô tả tất cả các thành phần của ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như mỗi Hoạt động phải được khai báo trong tệp XML này. Trong các bài học khóa học khác, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để biết thông tin giới thiệu, hãy xem Tổng quan về tệp kê khai ứng dụng .

## Nhiệm vụ 3: Sử dụng thiết bị ảo (trình giả lập)

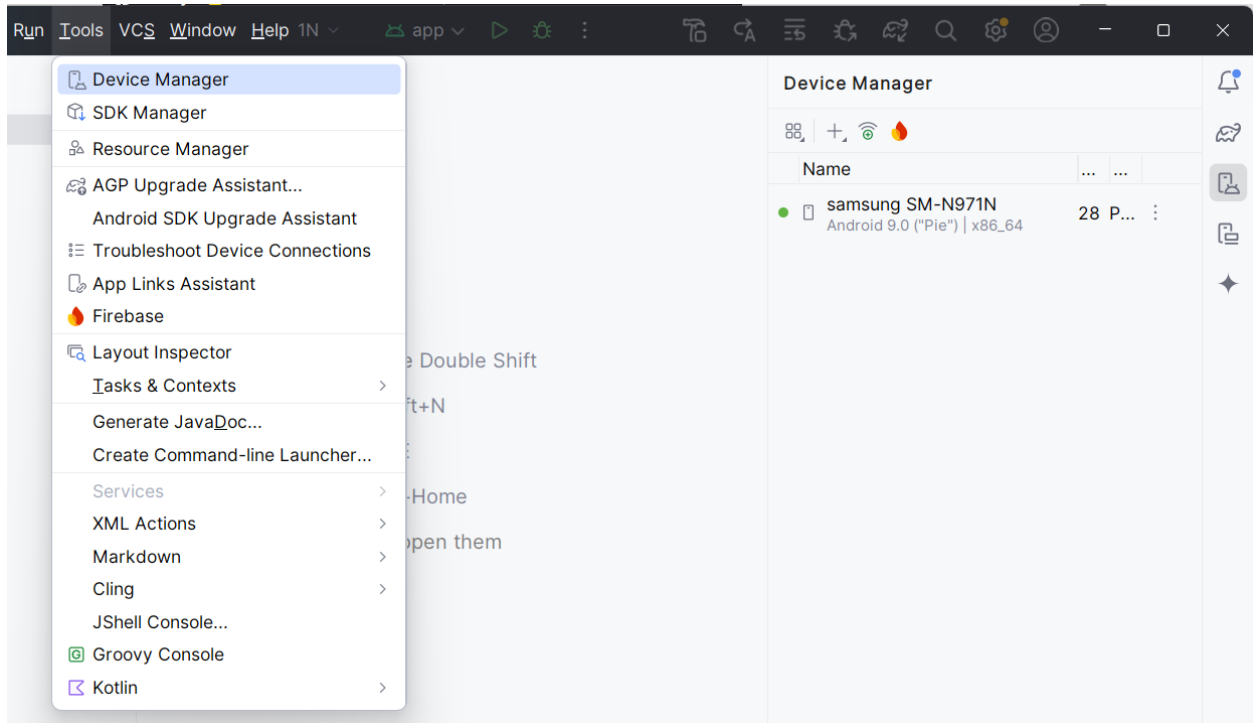
Trong tác vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo một thiết bị ảo (còn được gọi là trình mô phỏng) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Xin lưu ý rằng Trình mô phỏng Android có các yêu cầu bổ sung ngoài các yêu cầu hệ thống cơ bản đối với Android Studio.

Khi sử dụng Trình quản lý AVD, bạn xác định các đặc điểm phần cứng của thiết bị, cấp độ API, bộ nhớ, giao diện và các thuộc tính khác và lưu thiết bị đó dưới dạng thiết bị ảo. Với thiết bị ảo, bạn có thể kiểm tra ứng dụng trên các cấu hình thiết bị khác nhau (chẳng hạn như máy tính bảng và điện thoại) với các cấp độ API khác nhau mà không cần phải sử dụng thiết bị thực.

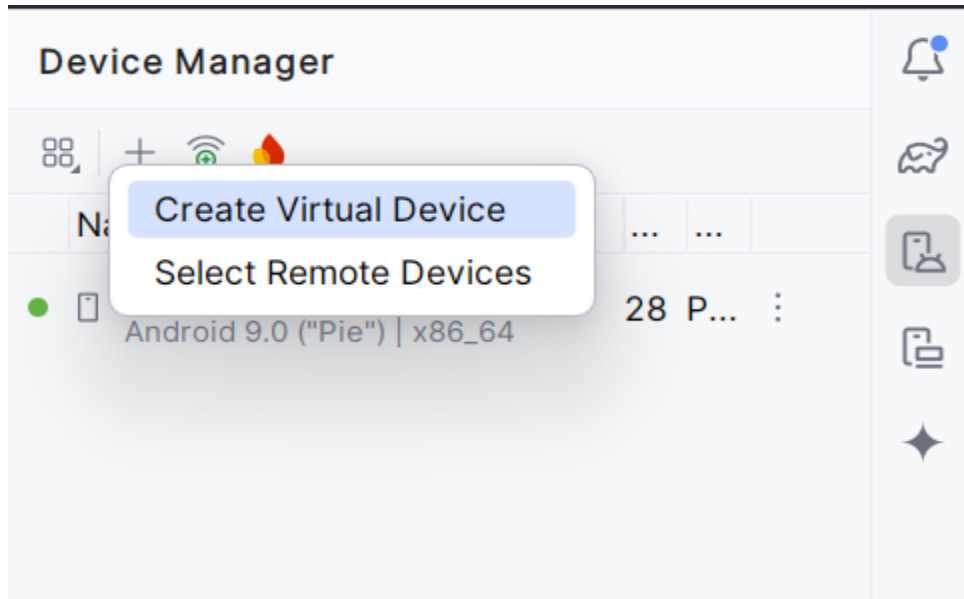
### 3.1 Tạo thiết bị ảo Android (AVD)

Để chạy trình mô phỏng trên máy tính, bạn phải tạo một cấu hình mô tả thiết bị ảo.

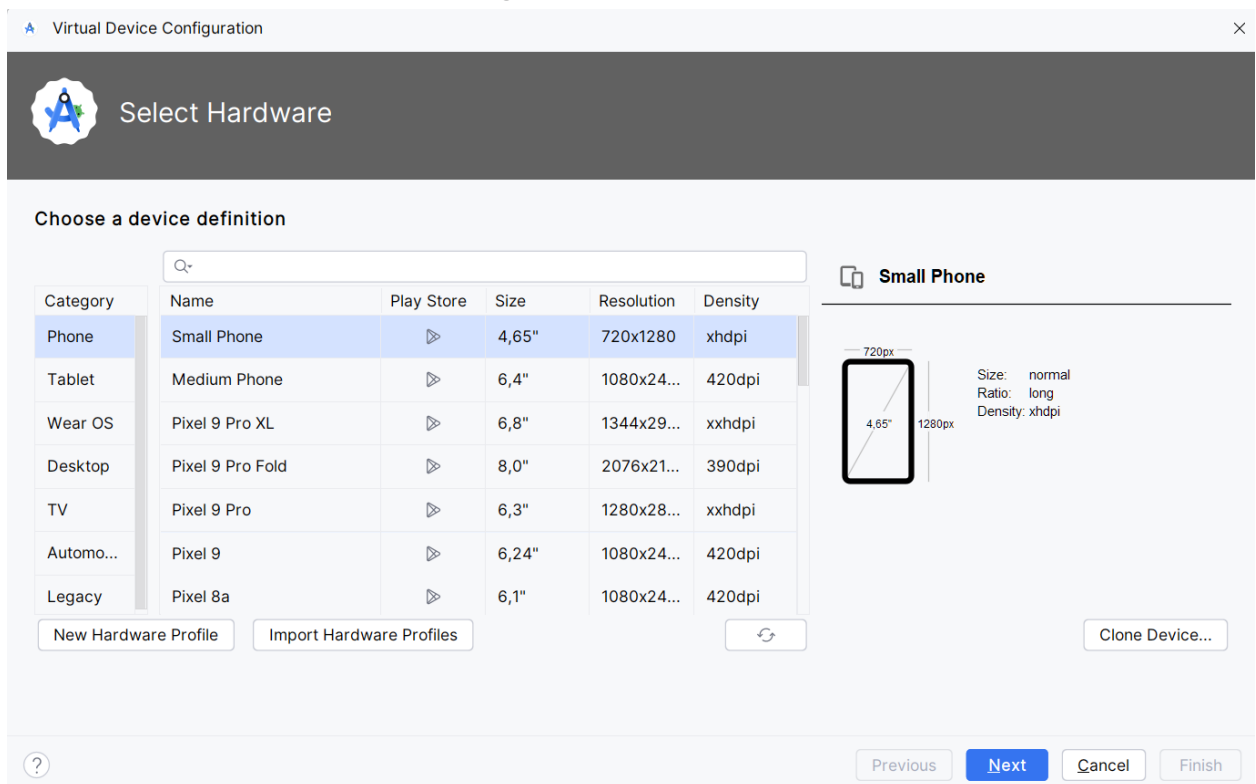
1. Trong Android Studio, hãy chọn Tools > AVD Manager hoặc nhấp vào biểu tượng Trình quản lý AVD trên thanh công cụ. Màn hình Thiết bị ảo của bạn xuất hiện. Nếu bạn đã tạo thiết bị ảo, màn hình sẽ hiển thị chúng (như trong hình bên dưới); nếu không, bạn sẽ thấy một danh sách trống.



2. Nếu chưa kết nối thiết bị, nhấp vào Add a new device > Create Virtual Device . Cửa sổ Chọn phần cứng xuất hiện hiển thị danh sách các thiết bị phần cứng được định cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo ( Kích thước ), độ phân giải màn hình tính bằng pixel ( Độ phân giải ) và mật độ điểm ảnh ( Mật độ ).

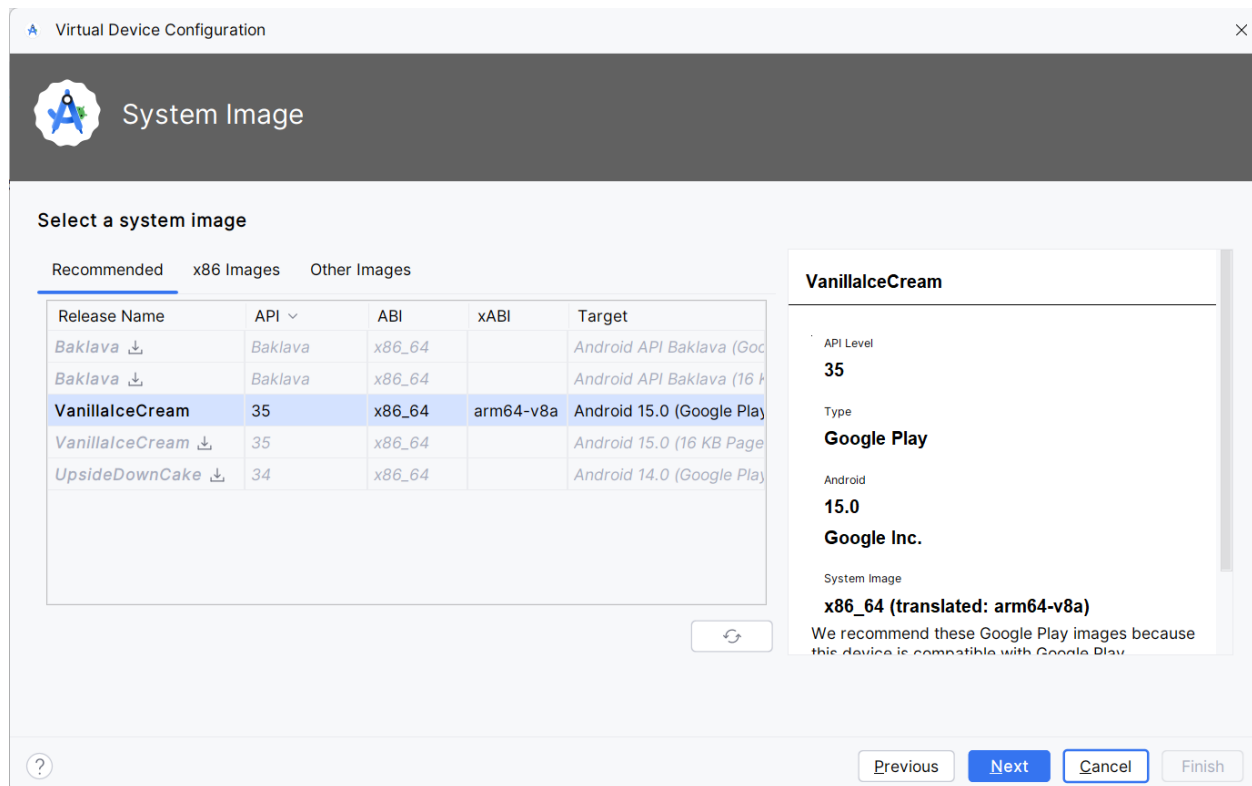


3. Chọn một thiết bị như Small Phone hoặc Medium Phone và nhấp vào Tiếp theo . Màn hình Hình ảnh hệ thống xuất hiện.



4. Nhấp vào tab Recommended nếu nó chưa được chọn và chọn phiên bản hệ thống Android để chạy trên thiết bị ảo (chẳng hạn như Oreo ).

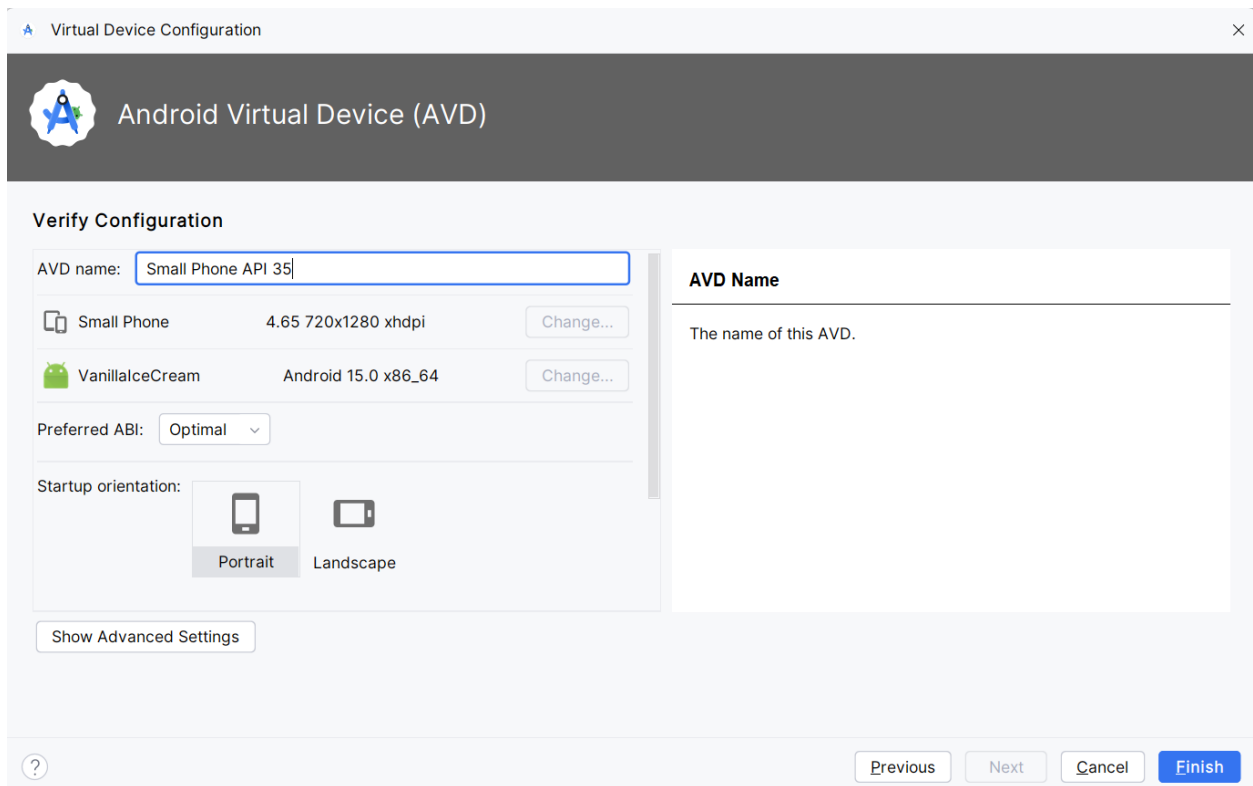




Có nhiều phiên bản hơn được hiển thị trong tab Recommended. Nhìn vào tab Hình ảnh x86\_64 và Hình ảnh khác để xem chúng.

Nếu liên kết Tải xuống hiển thị bên cạnh hình ảnh hệ thống bạn muốn sử dụng, thì liên kết đó chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào Kết thúc khi hoàn tất.

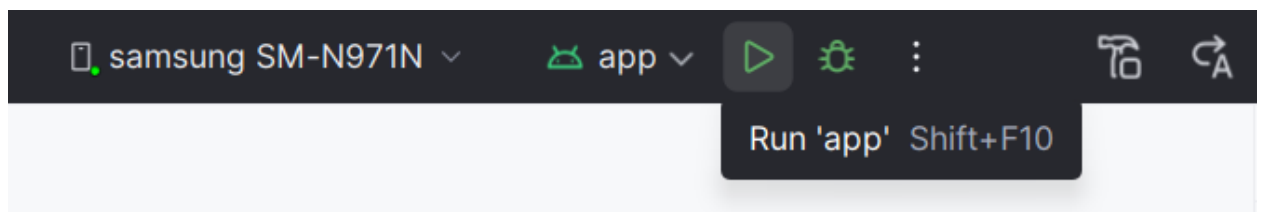
- Sau khi chọn hình ảnh hệ thống, hãy nhấp vào Tiếp theo . Cửa sổ Thiết bị ảo Android (AVD) sẽ xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào Finish .



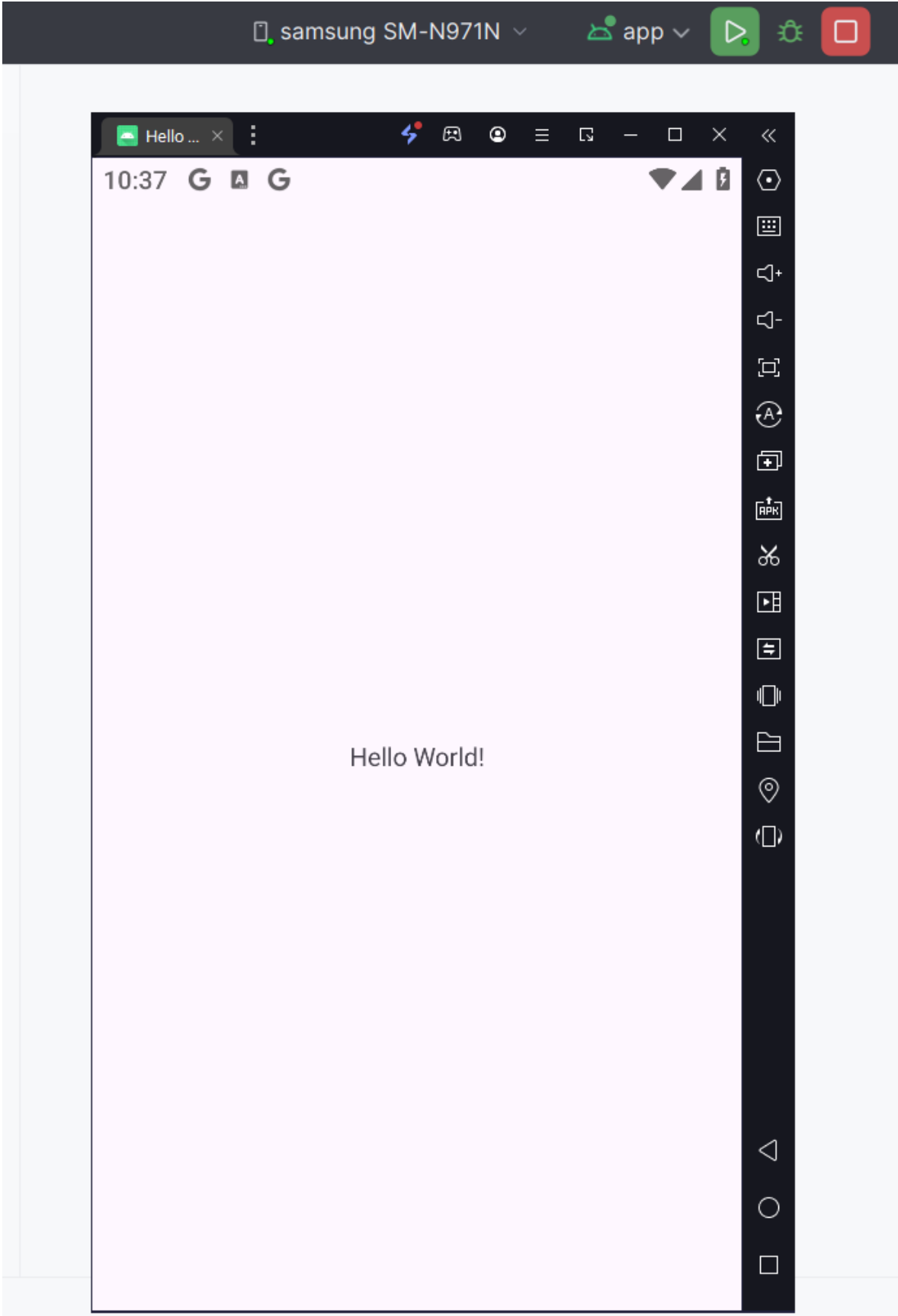
### 3.2 Chạy ứng dụng trên thiết bị ảo

Trong tác vụ này, cuối cùng bạn sẽ chạy ứng dụng Hello World của mình.

1. Trong Android Studio, nhấp vào biểu tượng Run hoặc tổ hợp phím Shift + F10 trên thanh công cụ.



Trình giả lập khởi động giống như một thiết bị vật lý. Tùy thuộc vào tốc độ máy tính của bạn, quá trình này có thể mất một lúc. Ứng dụng của bạn sẽ được xây dựng và sau khi trình mô phỏng đã sẵn sàng, Android Studio sẽ tải ứng dụng lên trình mô phỏng và chạy ứng dụng đó. Bạn sẽ thấy ứng dụng Hello World như trong hình sau.



Mẹo: Khi thử nghiệm trên một thiết bị ảo, bạn nên khởi động nó một lần, ngay từ đầu phiên của bạn. Bạn không nên đóng ứng dụng cho đến khi hoàn tất việc kiểm tra ứng dụng của mình để ứng dụng của bạn không phải trải qua quá trình khởi động thiết bị một lần nữa. Để đóng thiết bị ảo, hãy nhấp vào nút X ở đầu trình giả lập, chọn Thoát từ menu hoặc nhấn Control-Q trong Windows hoặc Command-Q trong macOS.

## **Nhiệm vụ 4: Sử dụng thiết bị vật lý**

Trong nhiệm vụ cuối cùng này, bạn sẽ chạy ứng dụng của mình trên thiết bị di động vật lý như điện thoại hoặc máy tính bảng. Bạn phải luôn kiểm tra ứng dụng của mình trên cả thiết bị ảo và thiết bị vật lý.

Những gì bạn cần:

- Thiết bị Android như điện thoại hoặc máy tính bảng.
- Cáp dữ liệu để kết nối thiết bị Android với máy tính qua cổng USB.
- Nếu bạn đang sử dụng hệ thống Linux hoặc Windows, bạn có thể cần thực hiện các bước bổ sung để chạy trên thiết bị phần cứng. Kiểm tra tài liệu Sử dụng thiết bị phần cứng. Bạn cũng có thể cần cài đặt trình điều khiển USB thích hợp cho thiết bị của mình. Đối với trình điều khiển USB dựa trên Windows, hãy xem Trình điều khiển USB OEM .

### **4.1 Bật gỡ lỗi USB**

Để cho phép Android Studio giao tiếp với thiết bị của bạn, bạn phải bật tính năng Gỡ lỗi USB trên thiết bị Android của mình. Tính năng này được bật trong cài đặt Tùy chọn nhà phát triển trên thiết bị của bạn.

Trên Android 4.2 trở lên, màn hình Tùy chọn nhà phát triển bị ẩn theo mặc định. Để hiển thị các tùy chọn dành cho nhà phát triển và bật Gỡ lỗi USB:

1. Trên thiết bị của bạn, mở Settings, tìm kiếm About phone, nhấp vào About phone và nhấn vào Build number bảy lần.
2. Quay lại màn hình trước đó ( Settings / System ). Tùy chọn nhà phát triển xuất hiện trong danh sách. Nhấn vào Developer options.
3. Chọn USB Debugging.

### **4.2 Chạy ứng dụng của bạn trên thiết bị**

Giờ đây, bạn có thể kết nối thiết bị của mình và chạy ứng dụng từ Android Studio.

1. Kết nối thiết bị của bạn với máy phát triển của bạn bằng cáp USB.
2. Nhấp vào nút Run trên thanh công cụ. Cửa sổ Chọn mục tiêu triển khai sẽ mở ra với danh sách các trình giả lập có sẵn và thiết bị được kết nối.
3. Chọn thiết bị của bạn và nhấp vào OK . Android Studio cài đặt và chạy ứng dụng trên thiết bị của bạn.

### **Khắc phục sự cố**

Nếu Android Studio không nhận dạng thiết bị của bạn, hãy thử các cách sau:

1. Rút dây cắm và cắm lại thiết bị của bạn.
2. Khởi động lại Android Studio.

Nếu máy tính của bạn vẫn không tìm thấy thiết bị hoặc tuyên bố thiết bị là "không được phép", hãy làm theo các bước sau:

1. Rút phích cắm của thiết bị.
2. Trên thiết bị, mở Tùy chọn nhà phát triển trong ứng dụng Cài đặt.
3. Nhấn vào Thu hồi ủy quyền gỡ lỗi USB.
4. Kết nối lại thiết bị với máy tính của bạn.
5. Khi được nhắc, cấp ủy quyền.

Bạn có thể cần cài đặt trình điều khiển USB thích hợp cho thiết bị của mình. Xem tài liệu Sử dụng thiết bị phần cứng .

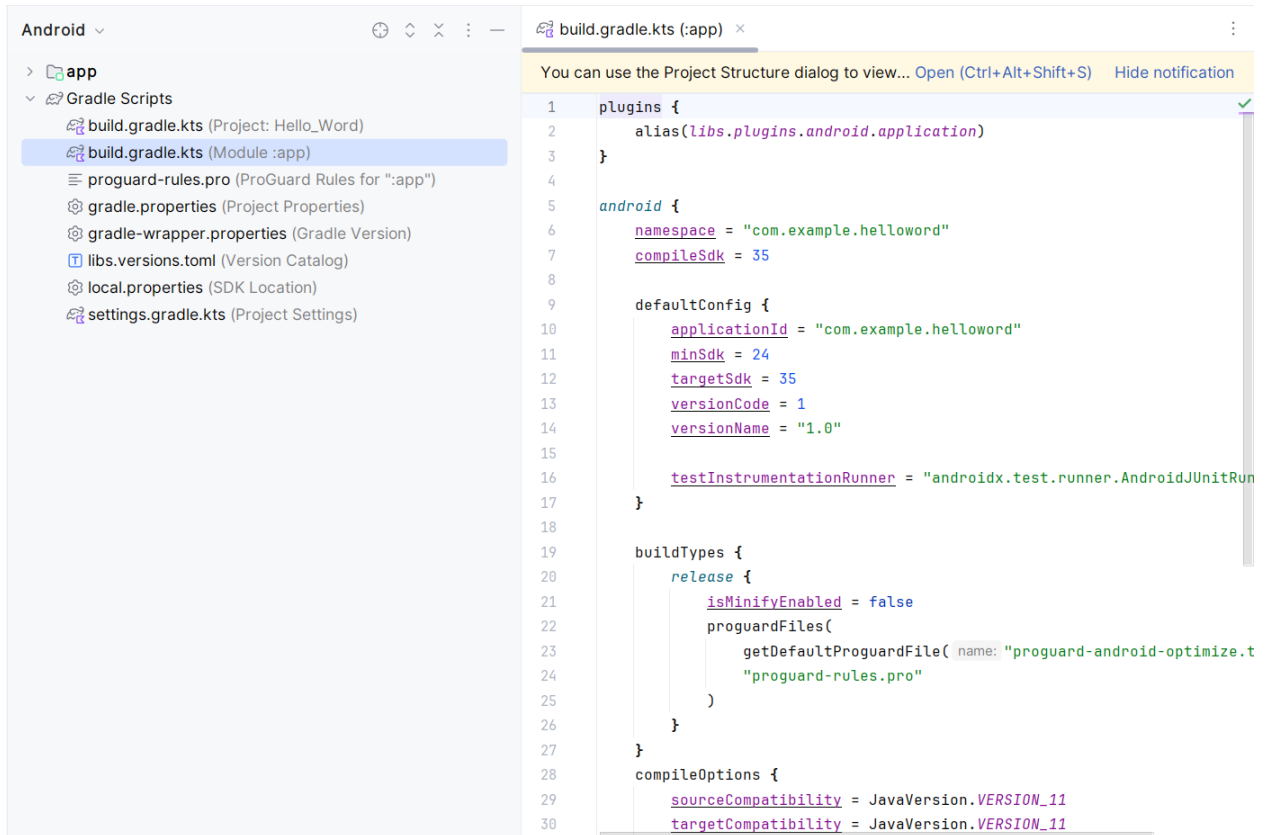
## **Nhiệm vụ 5: Thay đổi cấu hình Gradle của ứng dụng**

Trong tác vụ này, bạn sẽ thay đổi điều gì đó về cấu hình ứng dụng trong tệp `build.gradle(Module:app)` để tìm hiểu cách thực hiện các thay đổi và đồng bộ hóa chúng với dự án Android Studio của bạn.

### **5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng**

Làm theo các bước sau:

1. Mở rộng thư mục Tập lệnh Gradle nếu thư mục chưa mở và nhấp đúp vào tệp `build.gradle(Module:app)`. Nội dung của tệp xuất hiện trong trình soạn thảo mã.
2. Trong khối `defaultConfig`, giá trị của `minSdkVersion` là 24



Trình chỉnh sửa mã hiển thị thanh thông báo ở trên cùng với liên kết Đồng bộ hóa ngay.

## 5.2 Đồng bộ hóa cấu hình Gradle mới

Khi bạn thực hiện các thay đổi đối với tệp cấu hình bản dựng trong dự án, Android Studio yêu cầu bạn đồng bộ hóa các tệp dự án để có thể nhập các thay đổi về cấu hình bản dựng và chạy một số kiểm tra để đảm bảo cấu hình sẽ không tạo ra lỗi bản dựng.

Để đồng bộ hóa các tệp dự án, hãy nhấp vào Đồng bộ hóa ngay trong thanh thông báo xuất hiện khi thực hiện thay đổi (như trong hình trước) hoặc nhấp vào biểu tượng

Đồng bộ hóa dự án với tệp Gradle trên thanh công cụ.

Khi quá trình đồng bộ hóa Gradle kết thúc, thông báo Bản dựng Gradle đã hoàn tất sẽ xuất hiện ở góc dưới cùng bên trái của cửa sổ Android Studio.

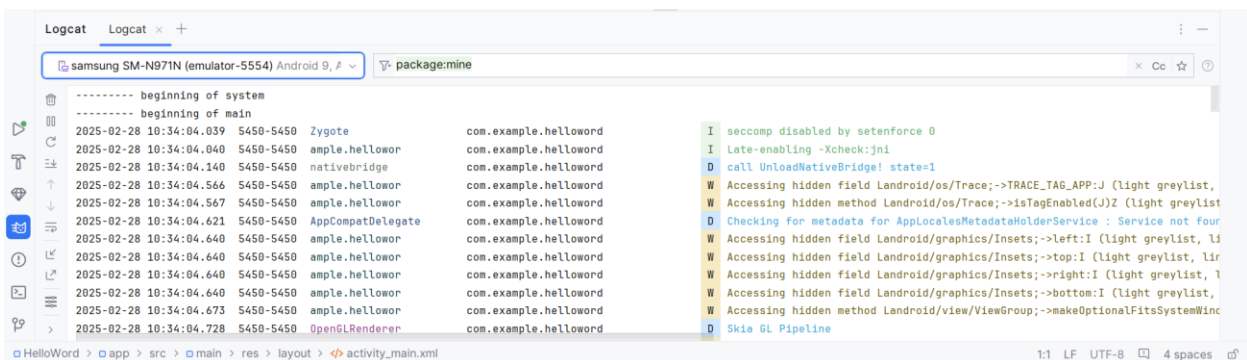
Để tìm hiểu sâu hơn về Gradle, hãy xem tài liệu Tổng quan về hệ thống xây dựng và Định cấu hình bản dựng Gradle.

### Nhiệm vụ 6: Thêm câu lệnh nhật ký vào ứng dụng của bạn

Trong nhiệm vụ này, bạn sẽ thêm câu lệnh Nhật ký vào ứng dụng của mình, cụm từ này hiển thị thông báo trong ngăn Logcat. Thông báo nhật ký là một công cụ gỡ lỗi mạnh mẽ mà bạn có thể sử dụng để kiểm tra các giá trị, đường dẫn thực thi và báo cáo ngoại lệ.

## 6.1 View ngăn Logcat

Để xem ngẩn Logcat, hãy nhấp vào thẻ Logcat ở thanh dọc góc trái Android Studio như trong hình bên dưới.



Trong hình trên:

1. Tab Logcat để mở và đóng ngăn Logcat, hiển thị thông tin về ứng dụng của bạn khi ứng dụng đang chạy. Nếu bạn thêm câu lệnh Nhật ký vào ứng dụng của mình, thông báo Nhật ký sẽ xuất hiện ở đây.

## 6.2 Thêm câu lệnh nhật ký vào ứng dụng của bạn

Câu lệnh nhật ký trong mã ứng dụng của bạn sẽ hiển thị thông báo trong ngăn Logcat. Chẳng hạn:

```
Log.d("MainActivity", "Hello World");
```

Các phần của tin nhắn là:

- Log: Lớp Nhật ký để gửi tin nhắn nhật ký đến ngăn Logcat.
- d: Cài đặt mức Nhật ký gỡ lỗi để lọc thông báo nhật ký hiển thị trong ngăn Logcat. Các cấp độ nhật ký khác là e cho Lỗi, w cho Cảnh báo và i cho Thông tin.

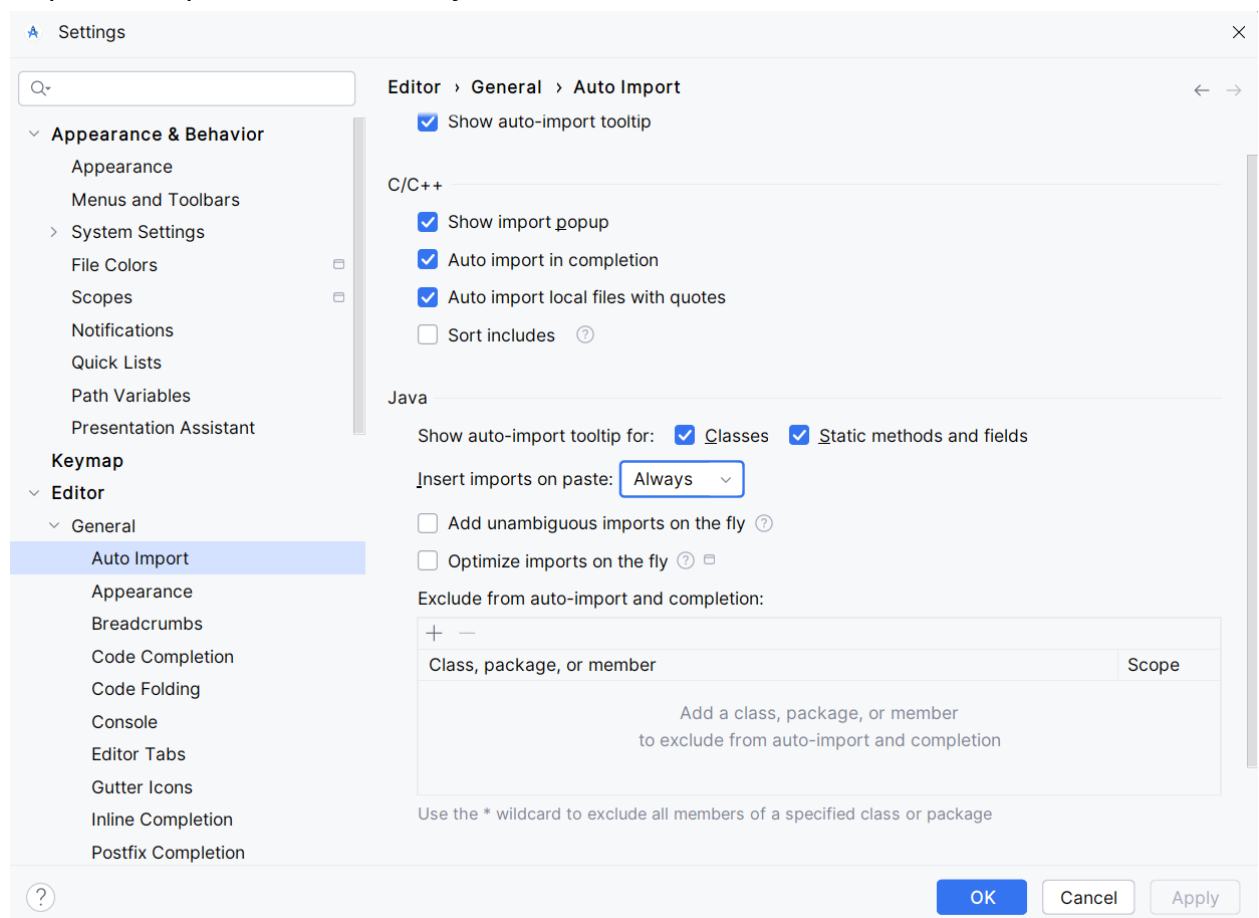
- "MainActivity": Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong ngăn Logcat. Đây thường là tên của Hoạt động mà thông điệp bắt đầu. Tuy nhiên, bạn có thể làm cho điều này bất cứ thứ gì hữu ích cho bạn để gỡ lỗi. Theo quy ước, thẻ nhật ký được định nghĩa là hằng số cho Hoạt động

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

- "Hello world": Đối số thứ hai là thông điệp thực tế.

Làm theo các bước sau:

1. Mở ứng dụng Hello World của bạn trong Android studio và mở MainActivity .
2. Để tự động thêm các mục nhập rõ ràng vào dự án của bạn (chẳng hạn như android.util.Log cần thiết để sử dụng Nhật ký ), hãy chọn File > Settings trong Windows hoặc Android Studio > Preferences trong macOS.
3. Chọn Editor > General >Auto Import . Chọn tất cả các hộp kiểm và đặt Insert imports on paste thành Always .

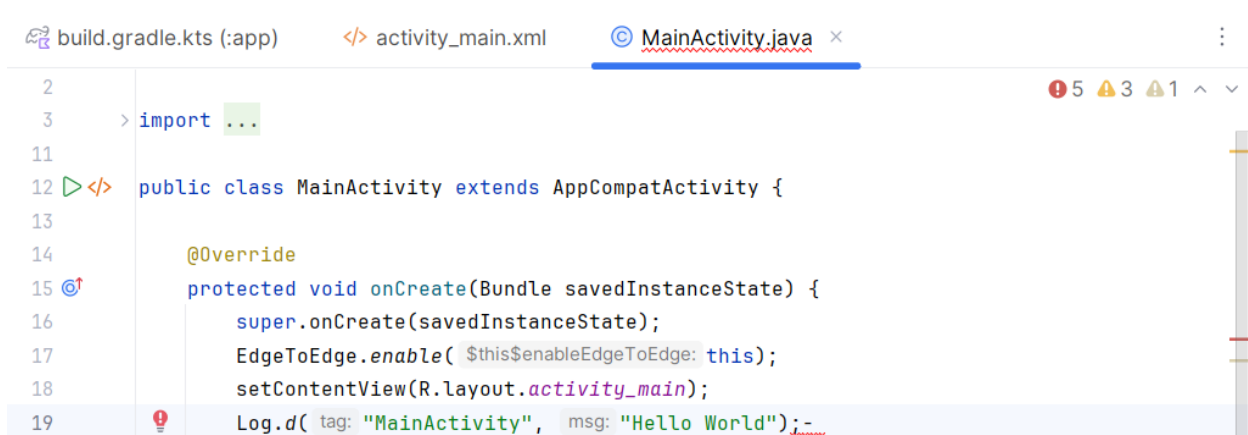




4. Nhấp vào Apply và sau đó nhấp vào OK .
5. Trong phương thức onCreate() của MainActivity , thêm câu lệnh sau:

```
Log.d("MainActivity", "Hello World");
```

Phương thức onCreate() bây giờ sẽ giống như mã sau:



```
build.gradle.kts (:app)  </> activity_main.xml  MainActivity.java x
2
3  > import ...
11
12  </> public class MainActivity extends AppCompatActivity {
13
14      @Override
15      protected void onCreate(Bundle savedInstanceState) {
16          super.onCreate(savedInstanceState);
17          EdgeToEdge.enable( $this$enableEdgeToEdge: this);
18          setContentView(R.layout.activity_main);
19      Log.d( tag: "MainActivity", msg: "Hello World");-
```

6. Nếu ngăn Logcat chưa mở, hãy nhấp vào tab Logcat ở cuối Android Studio để mở.
7. Kiểm tra xem tên mục tiêu và tên gói của ứng dụng có chính xác không.
8. Thay đổi mức nhật ký trong ngăn Logcat thành Gỡ lỗi (hoặc để nguyên chi tiết vì có rất ít thông báo nhật ký).
9. Chạy ứng dụng của bạn.

## Thử thách mã hóa

Lưu ý: Tất cả các thử thách mã hóa đều là tùy chọn và không phải là điều kiện tiên quyết cho các bài học sau này.

Thách thức: Bây giờ bạn đã thiết lập và làm quen với quy trình phát triển cơ bản, hãy làm như sau:

1. Tạo một dự án mới trong Android Studio.
2. Thay đổi lời chào "Hello World" thành "Happy Birthday thành" và tên của một người có sinh nhật gần đây.

3. (Tùy chọn) Chụp ảnh màn hình ứng dụng đã hoàn thành của bạn và gửi email cho người mà bạn quên ngày sinh.

4. Một cách sử dụng phổ biến của lớp Log là ghi nhật ký các ngoại lệ Java khi chúng xảy ra trong chương trình của bạn. Có một số phương thức hữu ích, chẳng hạn như Log.e() , mà bạn có thể sử dụng cho mục đích này. Khám phá các phương thức bạn có thể sử dụng để bao gồm ngoại lệ với thông báo Nhật ký. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại một ngoại lệ.

### **Tóm tắt**

- Để cài đặt Android Studio, hãy truy cập Android Studio và làm theo hướng dẫn để tải xuống và cài đặt nó.
- Khi tạo ứng dụng mới, hãy đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu.
- Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy nhấp vào tab Dự án trong cột tab dọc, sau đó chọn Android trong menu bật lên ở trên cùng.
- Chỉnh sửa tệp build.gradle(Module:app) khi bạn cần thêm thư viện mới vào dự án của mình hoặc thay đổi phiên bản thư viện.
- Tất cả mã và tài nguyên cho ứng dụng đều nằm trong các thư mục ứng dụng và res. Thư mục java bao gồm các hoạt động, kiểm tra và các thành phần khác trong mã nguồn Java. Thư mục res chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh.
- Chỉnh sửa tệp AndroidManifest.xml để thêm các tính năng, thành phần và quyền vào ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML này.
- Sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo thiết bị ảo (còn được gọi là trình giả lập) để chạy ứng dụng của bạn.
- Thêm câu lệnh Nhật ký vào ứng dụng của bạn, hiển thị thông báo trong ngăn Logcat như một công cụ cơ bản để gỡ lỗi.
- Để chạy ứng dụng của bạn trên thiết bị Android vật lý bằng Android Studio, hãy bật Gỡ lỗi USB trên thiết bị. Mở Settings > About phone về điện thoại và nhấn vào Build number bảy lần. Quay lại màn hình trước đó ( Settings) và nhấn vào Developer options. Chọn USB Debugging.

## Các khái niệm liên quan

Tài liệu về khái niệm liên quan có trong 1.0: Introduction to Android 1.1 Your first Android app .

## Tìm hiểu thêm

- Tài liệu về Androi Studio
- Android Studio download page
- Android Studio release notes
- Meet Android Studio
- Logcat command-line tool
- Android Virtual Device (AVD) manager
- App Manifest Overview
- Configure your build
- Log class
- Create and Manage Virtual Devices

Khác:

- How do I install Java?
- Installing the JDK Software and Setting JAVA\_HOME
- Gradle site
- Apache Groovy syntax
- Gradle Wikipedia page

## Bài tập về nhà

Xây dựng và chạy một ứng dụng

- Tạo một dự án Android mới từ Mẫu trống.
- Thêm câu lệnh ghi nhật ký cho các cấp độ nhật ký khác nhau trong onCreate() trong hoạt động chính.
- Tạo trình giả lập cho thiết bị, nhắm mục tiêu bất kỳ phiên bản Android nào bạn thích và chạy ứng dụng.
- Sử dụng bộ lọc trong Logcat để tìm các câu lệnh nhật ký của bạn và điều chỉnh các cấp độ để chỉ hiển thị các câu lệnh gỡ lỗi hoặc ghi nhật ký lỗi

Trả lời các câu hỏi sau:

Câu hỏi 1:

Tên của tệp bố cục cho hoạt động chính là gì?

- MainActivity.java
- AndroidManifest.xml
- activity\_main.xml
- build.gradle

Câu hỏi 2:

Tên của tài nguyên chuỗi chỉ định tên của ứng dụng là gì?

- app\_name
- xmlns:app
- android:name
- applicationId

Câu hỏi 3

Bạn sử dụng công cụ nào để tạo trình giả lập mới?

- Android Device Monitor
- AVD Manager
- SDK Manager
- Theme Editor

Câu hỏi 4:

Giả sử rằng ứng dụng của bạn bao gồm câu lệnh ghi nhật ký sau:

```
Log.i("MainActivity", "MainActivity layout is complete");
```

Bạn thấy câu lệnh "Bố cục MainActivity đã hoàn tất" trong ngăn Logcat nếu trình đơn Cập nhật ký được đặt thành tùy chọn nào sau đây? (Gợi ý: nhiều câu trả lời là được.)

- Verbose

- Debug
- Info
- Warn
- Error
- Assert

Gửi ứng dụng của bạn để chấm điểm

Kiểm tra để đảm bảo ứng dụng có những điều sau:

- Hoạt động hiển thị "Hello World" trên màn hình.
- Ghi lại các câu lệnh trong onCreate() trong hoạt động chính.
- Mức nhật ký trong ngăn Logcat chỉ hiển thị các câu lệnh gỡ lỗi hoặc ghi nhật ký lỗi.

**1.2) Giao diện người dùng tương tác đầu tiên**

**1.3) Trình chỉnh sửa bố cục**

**1.4) Văn bản và các chế độ cuộn**

**1.5) Tài nguyên có sẵn**

## **Bài 2) Activities**

**2.1) Activity và Intent**

**2.2) Vòng đời của Activity và trạng thái**

**2.3) Intent ngầm định**

## **Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ**

**3.1) Trình gỡ lỗi**

**3.2) Kiểm thử đơn vị**

**3.3) Thư viện hỗ trợ**

## **CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG**

### **Bài 1) Tương tác người dùng**

- 1.1) Hình ảnh có thể chọn**
- 1.2) Các điều khiển nhập liệu**
- 1.3) Menu và bộ chọn**
- 1.4) Điều hướng người dùng**
- 1.5) RecyclerView**

### **Bài 2) Trải nghiệm người dùng thú vị**

- 2.1) Hình vẽ, định kiểu và chủ đề**
- 2.2) Thẻ và màu sắc**
- 2.3) Bố cục thích ứng**

### **Bài 3) Kiểm thử giao diện người dùng**

- 3.1) Espresso cho việc kiểm tra UI**

## **CHƯƠNG 3. LÀM VIỆC TRONG NỀN**

### **Bài 1) Các tác vụ nền**

- 1.1) AsyncTask**
- 1.2) AsyncTask và AsyncTaskLoader**
- 1.3) Broadcast receivers**

### **Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền**

- 2.1) Thông báo**
- 2.2) Trình quản lý cảnh báo**
- 2.3) JobScheduler**

## **CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG**

### **Bài 1) Tùy chọn và cài đặt**

#### **1.1) Shared preferences**

#### **1.2) Cài đặt ứng dụng**

### **Bài 2) Lưu trữ dữ liệu với Room**

#### **2.1) Room, LiveData và ViewModel**

#### **2.2) Room, LiveData và ViewModel**