**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 4](#_Toc190855147)

[Bài 1) Tạo ứng dụng đầu tiên 4](#_Toc190855148)

[1.1) Android Studio và Hello World 4](#_Toc190855149)

[1.2) Giao diện người dùng tương tác đầu tiên 5](#_Toc190855150)

[1.3) Trình chỉnh sửa bố cục 5](#_Toc190855151)

[1.4) Văn bản và các chế độ cuộn 5](#_Toc190855152)

[1.5) Tài nguyên có sẵn 5](#_Toc190855153)

[Bài 2) Activities 5](#_Toc190855154)

[2.1) Activity và Intent 5](#_Toc190855155)

[2.2) Vòng đời của Activity và trạng thái 5](#_Toc190855156)

[2.3) Intent ngầm định 5](#_Toc190855157)

[Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ 5](#_Toc190855158)

[3.1) Trình gỡ lỗi 5](#_Toc190855159)

[3.2) Kiểm thử đơn vị 5](#_Toc190855160)

[3.3) Thư viện hỗ trợ 5](#_Toc190855161)

[CHƯƠNG 2. Trải nghiệm người dùng 6](#_Toc190855162)

[Bài 1) Tương tác người dùng 6](#_Toc190855163)

[1.1) Hình ảnh có thể chọn 6](#_Toc190855164)

[1.2) Các điều khiển nhập liệu 6](#_Toc190855165)

[1.3) Menu và bộ chọn 6](#_Toc190855166)

[1.4) Điều hướng người dùng 6](#_Toc190855167)

[1.5) RecycleView 6](#_Toc190855168)

[Bài 2) Trải nghiệm người dùng thú vị 6](#_Toc190855169)

[2.1) Hình vẽ, định kiểu và chủ đề 6](#_Toc190855170)

[2.2) Thẻ và màu sắc 6](#_Toc190855171)

[2.3) Bố cục thích ứng 6](#_Toc190855172)

[Bài 3) Kiểm thử giao diện người dùng 6](#_Toc190855173)

[3.1) Espresso cho việc kiểm tra UI 6](#_Toc190855174)

[CHƯƠNG 3. Làm việc trong nền 6](#_Toc190855175)

[Bài 1) Các tác vụ nền 6](#_Toc190855176)

[1.1) AsyncTask 6](#_Toc190855177)

[1.2) AsyncTask và AsyncTaskLoader 6](#_Toc190855178)

[1.3) Broadcast receivers 6](#_Toc190855179)

[Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền 6](#_Toc190855180)

[2.1) Thông báo 6](#_Toc190855181)

[2.2) Trình quản lý cảnh báo 6](#_Toc190855182)

[2.3) JobScheduler 6](#_Toc190855183)

[CHƯƠNG 4. Lưu dữ liệu người dùng 7](#_Toc190855184)

[Bài 1) Tùy chọn và cài đặt 7](#_Toc190855185)

[1.1) Shared preferences 7](#_Toc190855186)

[1.2) Cài đặt ứng dụng 7](#_Toc190855187)

[Bài 2) Lưu trữ dữ liệu với Room 7](#_Toc190855188)

[2.1) Room, LiveData và ViewModel 7](#_Toc190855189)

[2.2) Room, LiveData và ViewModel 7](#_Toc190855190)

3.1) Trinfh gowx loi ……………………………………………………………………...

# LÀM QUEN

## Tạo ứng dụng đầu tiên

### 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**

### Giao diện người dùng tương tác đầu tiên

Sau khi bạn đã cài đặt thành công Android Studio, bạn sẽ tạo ứng dụng Hello World từ một mẫu, một dự án mới. Đó là một ứng dụng đơn giản hiển thị dòng chữ “Hello World” trên màn hình của thiết bị Android ảo hoặc thiết bị vật lý.

Khi mà ứng dụng chạy thành công thì sẽ hiển thị như sau:

**Nhiệm vụ 1: Cài đặt Android Studio**

Android Studio cung cấp một môi trường tích hợp (IDE) hoàn chỉnh bao gồm chỉnh sửa mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra nó còn chứa các công cụ dành cho phát triển, sửa lỗi, kiểm thử và hiệu suất giúp việc phát triển các ứng dụng trở nên nhanh chóng và dễ dàng hơn. Bạn có thể thử nghiệm ứng dụng của mình với một loạt các trình giả lập đã được cấu hình sẵn hoặc trên thiết bị di động của riêng bạn, xây dụng ứng dụng sẵn sàng sử dụng và xuất bản trên cửa hàng Google Play.

**Chú ý:** Android Studio đang liên tục được cải tiến. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, xem Android Studio.

Android Studio có sẵn cho các máy tính chạy Windows hoặc Linux và cho máy Mac chạy macOS. OpenJDK mới nhất ( Java Development Kit) được đi kèm với Android Studio.

Để khởi động và chạy với Android Studio, đầu tiên kiểm tra các yêu cầu hệ thống để đảm bảo rằng hệ thống đáp ứng chúng. Việc cài đặt tương tự cho tát cả các nền tẳng. Bất kì sự khác biệt nào cũng được ghi chú dưới đây:

1. Điều hướng đến Android developers site và làm theo các hướng dẫn để tải về 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 thành việc cài đặt, Setup Wizard sẽ được tải xuống và cài đặt một số thành phần bổ sung bao gồm Android SDK. Hãy kiên nhẫn, điều này có thể mất một thời gian tùy thuộc vào tốc độ mạng của bạn và một số bước có thể bị lặp lại.
4. Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động và bạn đã sẵn sàng để tạ dự án đầu tiên của mình.

**Nhiệm vụ 2: Tạo ứng dụng Hello World**

Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển hiện thị “Hello World” để xác minh rằng Android Studio đã được cài đặt đúng cách và để tìm hiểu những điều cơ bản của việc phát triển với Android Studio.

**Bước 1: Tạo dự án ứng dụng**

1. Mở Android Studio nếu nó chưa được mở.
2. Trong cửa sổ chính **Welcom to Android Studio**, nhấp chuột vào **Start a new Android Studio project**.
3. Trong cửa sổ **Create Android Studio**, gõ **Hello World** vào dòng **Application name**.
4. Xác nhận rằng **Project location** mặc định là nơi mà bạn muốn lưu trữ ứng dụng Hello World của bạn và các dự án Android Studio khác hoặc thay đổi thành thư mục được chọn bởi bạn.
5. Chấp nhận **android.example.com** mặc định cho tên miền **Company Domanic** hoặc tạo một tên miền công ty riêng biệt.

Nếu bạn không có kế hoạch về phát hành ứng dụng của bạn, bạn có thể chấp nhận mặc định. Hãy lưu ý rằng việc thay đổi gói tên của ứng dụng sau này sẽ tốn thêm công sức.

1. Bỏ chọn các tùy chọn **Include C++ support** và **Include Kotlin support**, sau đó nhấp vào **Next**.
2. Trên màn hình **Taget Android Devices**, hãy đảm bảo rằng **Phone and Tablet** được chọn. Đảm bảo rằng **API 15: Android 4.0.3 IceCreamSandwich** được đặt là Minimum SDK; nếu không hãy sử dụng hộp thoại menu để đặt lại.

Đây là các cài đặt được sử dụng trong các ví dụ trong các bài học của khóa học này. Tính đến thời điểm này, các cài đặt giúp ứng dụng Hello World của bạn tương thích với 97% các thiết bị Android đang hoạt động trên cửa hàng Google Play.

1. Bỏ chọn tùy chọn **Include Instant App support** và tất cả các tùy chọn khác. Sau đó, nhấp vào Next. Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK mục tiêu đã chọn, Andoid Studio sẽ tự động cài đặt chúng.
2. Cửa sổ **Add an Activity** xuất hiện. Activity là đơn vị duy nhất, tâp trung những thứ mà người dùng có thể thực hiện. Nó là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Activity thường có một giao diện (layout) liên quan đến nó, định nghĩa cách mà các phần tử giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp các mẫu Activity để giúp bạn bắt đầu. Đối với dự án Hello World, chọn **Empty Activity** như hình dưới đây và nhấp vào **Next.**

A screenshot of a computer

AI-generated content may be incorrect.

1. Màn hình **Configure Activity** xuất hiện (khác nhau tùy thuộc vào mẫu bạn chọn ở bước trước). Theo mặc định, Activity trống do mẫu cung cấp có tên là **MainActivity**. Bạn có thể thay đổi tên này nếu muốn, nhưng bài học này sử dụng **MainActivity**.
2. Đảm bảo rằng tùy chọn **Generate Layout** file được chọn. Tên bố cục theo mặc định là activity\_main. Bạn có thể thay đổi nếu muốn, nhưng bài học này sử dụng activity\_main.
3. Đảm bảo rằng tùy chọn **Backwards Compatibility** (App Compat) được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước đó.
4. Nhấp vào **Finish**.

A screenshot of a computer

AI-generated content may be incorrect.

Android Studio tạo một thư mục cho các dự án của bạn và xây dụng dự án với Gradle (điều này có thể mất vài phút).

**Mẹo**: Xem trang **Configure your build developer** để biết thông tin chi tiết.

Bạn cũng có thể thấy thông báo “Tip of the day” với các phím tắt và các mẹo hữu ích khác. Nhấp vào **Close** để tắt thông báo.

Trình chỉnh sửa Android Studio sẽ xuất hiện. Thực hiện theo những bước sau:

1. Nhấp vào tab activity\_main.xml để xme trình chỉnh sửa bố cục.
2. Nhấp vào tab Design của trình chỉnh sửa bố cục nếu chưa chọn, để hiển thị bản trình bày đồ họa của bố cục, được hiển thị như bên dưới.
3. Nhấp vào MainActivity.java để xem trình chỉnh sửa mã, được hiển thị như bên dưới:

**Bước 2: Khám phá dự án > bảng điều khiển Android**

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

Nếu chưa chọn một dự án nào, nhấp vào tab **Project** trong cột tab dọc ở phía bên trái của cửa sổ Android Studio. Cửa sổ Project sẽ xuất hiện.

Để xem dự án trong phân cấp dự án Android chuẩn, hãy chọn **Android** từ hộp thoại menu bật lên phía trên của bảng Project, như hiển thị bên dưới.

**Chú ý**: Trong chương này và các chương khác đề cập đến khung Project, khi cài đặt là **Android** như **Project > bảng Android**.

#### Khám phá thu mục Gradle Scripts

Hệ thống xây dựng Gradle trong Android Studio giúp dễ dàng bao gồm các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào quá trình xây dựng của bạn dưới dạng phụ thuộc.

Khi bạn lần đầu tiên tạo một dự án ứng dụng, **Project > bảng Android** sẽ xuất hiện với thư mục **Gradle Scripts** được mở rộng như hình dưới đây:

Thực hiện theo từng bước để khám phá hệ thống Gradle:

1. Nếu thư mục **Gradle Scripts** chưa được mở rộng, hãy nhấp vào hình tam giác để mở rộng nó.

Thư mục này chứa tất cả các tệp cần thiết cho xây dựng hệ thống.

1. Tìm tệp **build.gradle (Dự án: 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 duy nhất một tệp xây dựng Gradle cấp cao. Hầu hết thời gian, bạn sẽ không cần phải thực hiện bất kỳ thay đổi nào đối với tệp này nhưng vẫn rất hữu ích để hiểu nội dung của nó.

Theo mặc định, tệp xây dựng cấp cao sử dụng khổi buildscript để định nghĩa các kho lưu trữ và phụ thuộc Gradle chung cho tất cả các mô-đun trong dự án. Khi các yếu tố phụ thuộc của bạn là một thứ gì đó khác ngoài thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong các kh lưu trữ trực tuyến được chỉ định trong khối lưu trữ của tệp. Theo mặc định, các dự án Android Studio mới khai bào Jcenter và Google (bao gồm cả Google Maven responsitory) là các vị trí kho lưu trữ:

1. Tìm tệp build.gradle(Module:app).

Ngoài tệp build.gradle cấp dự án, mỗi mô-đun có một tệp build.gradle riêng, cho phép bạn cấu hình các cài đặt xây dựng cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Việc cấu hình các cài đặt xây dựng này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như bổ sung các kiểu xây dựng và các phiên bản của sản phẩm. Bạn cũng có thể ghi đè các cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp cao.

Tệp này thường là tệp cần chỉnh sửa khi thay đổi các cấu hình cấp ứng dụng, chẳng hạn như khai báo các yếu tố trong phần denpendencies. Bạn có thể khai báo một thư viện phụ thuộc bằng nhiều cách cấu hình khác nhau. Mỗi cấu hình phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ: câu lệnh implementation fileTree(dir: ‘libs’, include: ['\*.jar']) thêm một phụ thuộc cho tất cả các tệp “.jar” bên trong thư mục libs.

Dưới đây là tệp **build.gradle(Module:app)** cho ứng dụng HelloWorld:

1. Nhấp vào hình tam giác để đóng Gradle Scripts.

#### Khám phá thư mục app và thư mục res

Tất cả mã và tài nguyên của ứng dụng đều nằm trong thư mục app và res.

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

Thư mục **java** chứa các tệp lớp Java được tổ chức trong ba thư mục con như hình minh họa trong hình trên. Thư mục **com.example.android.helloworld** (hoặc tên miền đã chỉ định) chứa tất cả các tệp của một gói ứng dụng. Hai thư mục còn lại được sử dụng cho mục đích kiểm thử và sẽ được mô tả trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một goid duy nhất và nó chứa tệp MainActivity.java. Đây là Activity (màn hình) đầu tiên mà người dùng nhìn thấy, đồng thời khởi tạo các tài nguyên chung cho ứng dụng, thường được đặt tên là **MainActivity** (phần mở rộng tệp bị ẩn trong bảng **Project > Android**).

1. Mở rộng thư mục res và thư mục layout, và nhấp đúp chuột vào tệp activity\_main.xml để mở tệp đó trong trình chỉnh sửa layout.

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

#### Khám phá thư mục mainfests

Thư mục manifests chứ các file cung cấp thông tin cần thiết về ứng dụng của bạn cho hệ thống Android, mà hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng.

1. Mở rộng thư mục **manifests**.
2. Mở tệp **AndroidManifests.xml**.

Tệp AndroidManifests.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 Activity phải được khai báo trong tệp XML này. Trogn các bài học khác của khóa học, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để đọc phần giới thiệu, hãy xem App Manifest Overview.

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

Trong nhiệm vụ này, bạn sẽ sử dụng Android Virtual Device (AVD) manager để tạo một thiết bị ảo (cũng có thể hiểu là trình giả lập) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng Android Emulator có các yêu cầu bổ sung ngoài các yêu cầu hệ thống cơ bản đối với Android Studio.

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

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

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

1. Trong Android Studio, chọn **Tools > Android > AVD Manager** hoặc nhấp vào biểu tượng AVD Manager trên thanh công cụ. Màn hình Your Virtual Devices 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 danh sách trống.
2. Nhấp vào **+Create Virtual Device**. Cửa sổ **Select Hardware** sẽ xuất hiện, hiển thị danh sách các thiết bị phần cứng được cấu hình trước. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước màn hình chéo (**Size**), độ phân giải màn hình tính bằng pixel (**Resolution**) và mật độ pixel (**Density**).
3. Chọn một thiết bị như **Nexus 5x** hoặc **Pixel XL** và nhấp vào **Next**. Màn hình **System Image** xuất hiện.
4. Nhấp vào tab **Recommended** nếu chưa chọn và chọn phiên bản hệ thống Android nào để chạy trên thiết bị ảo (như **Oreo**).

Có nhiều phiên bản khả dụng hơn so với những phiên bản được hiển thị trong tab **Recommended**. Hãy xem các tab **x86 Image** và **Other Images** để xem chúng.

Nếu liên kết **Download** hiển thị bên cạnh ảnh hệ thống mà bạn muốn sử dụng, nếu ảnh đó vẫn chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào **Finish** khi hoàn tất.

1. Sau khi chọn ảnh hệ thống, hãy nhấp vào **Next**. Cửa sổ Android Virtual Device (AVD) manager xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào **Finish**.

#### Chạy ứng dụng trên thiết bị ảo

Trong nhiệm vụ này, bạn sẽ hoàn tất việc chạy ứng dụng Hello World của bạn.

1. Trong Android Studio, chọn **Run > Run app** hoặc nhấp vào biểu tượng **Run** trên thanh công cụ.
2. Cửa sổ **Select Deployment Target**, bên dưới **Available Virtual Devices**, chọn thiết bị ảo bạn vừa tạo và nhấp **OK**.

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

**Mẹo**: Khi thử nghiệm trên thiết bị ảo, bạn nên khởi động thiết bị một lần, ngay khi bắt đầu phiên làm việc. Bạn không nên đóng thiết bị cho đến khi hoàn tất thử nghiệm ứng dụng, để ứng dụng 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 **Quit** từ menu hoặc nhấn **Control-Q** trong Windows hoặc Command-Q trong macOS.

**Nhiệm vụ 4: (Không bắt buộc) Sử dụng thiết bị vật lý**

Trong nhiệm vụ cuối này, bạn sẽ chạy ứng dụng của bạn trên một thiết bị di động như điện thoại hoặc máy tính bảng. Bạn nên luôn luôn thử nghiệm ứng dụng của bạn trên cả thiết bị ảo và thiết bị vật lý.

Những điều bạn cần có:

* Một thiết bị Android như điện thoại hoặc máy tính bảng.
* Một dây cáp truyền dữ liệu kết nối thiết bị Android của bạn với máy tính của bạn thông qua cổng USB.
* Nếu bạn đang sử dụng Hệ thống Linux hoặc Windows, bạn có thể cần thực hiện thêm các bước để chạy trên phần cúng thiết bị. Kiểm tra tài liệu **Using Hardware Devices**. Bạn cũng có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Đối với trình điều khiển USB dựa trên Windows, hãy xem **OEM USB Drivers**.

**Bước 1: Bật chế độ USB debugging**

Để Android Studio kết nối với thiết bị của bạn, bạn phải bật chế độ USB Debugging triên thiết bị Android của bạn. Tùy chọn này được kích hoạt trong phần **Developer options** trên thiết bị của bạn.

Trên Android 4.2 trở lên, màn hình **Developer options** bị ẩn theo mặc định. Để hiển thị tùy chọn nhà phát triển và bật USB Debugging, hãy làm theo các bước sau:

1. Trên thiết bị của bạn, mở **Cài đặt** (**Settings**), tìm **Giới thiệu về điện thoại** (**About phone**), nhấn vào đó và chạm vào **Số bản dựng** (**Build number**) bảy lần liên tiếp.
2. Quay lại màn hình trước đó (**Cài đặt/ Hệ thống – Settings/ System**), bạn sẽ thấy mục **Developer options** xuất hiện trong danh sách. Nhấn vào nó.
3. Tìm và bật **USB Debugging**.

**Bước 2: Chạy ứng dụng của bạn trên một thiết bị**

Bây giờ ban có thể kết nối thiết bị của bạn và chạy ứng dụng từ Android Studio.

1. Kết nối thiết bị của bạn với máy tính phát triển bằng cáp USB.
2. Nhấn vào nút **Run** trên thanh công cụ. Cửa sổ **Select Deployment Target** sẽ mở ra, hiển thị danh sách các trình giả lập và thiết bị đã kết nối.
3. Chọn thiết bị của bạn và nhấn **OK**.

Android Studio sẽ cài đặt và chạy ứng dụng trên thiết bị.

**Khắc phục sự cố:**

Nếu Android Studio không nhận diện thiết bị của bạn, hãy thử:

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

Nếu máy tính vẫn không nhận diện thiết bị hoặc báo “unauthorized”, hãy làm theo các bước sau:

1. Rút thiết bị ra.
2. Trên thiết bị, mở **Developer Options** trong ứng dụng **Cài đặt (Settings)**.
3. Nhấn vào Revoke **USB Debugging** authorizations.
4. Kết nối lại thiết bị với máy tính.
5. Khi có thông báo yêu cầu cấp quyền, hãy chấp nhận.

Bạn có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Xem thêm tại Using Hardware Devices documentation.

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

Trong nhiệm vụ này, bạn sẽ thay đổi một số cấu hình của ứng dụng trong tệp build.gradle (Module:app) để tìm hiểu cách thực hiện thay đổi và đồng bộ chúng với dự án Android Studio của bạn.

**Bước 1: Thay đổi phiên bản tối thiểu SDK cho ứng dụng**

Thực hiện theo các bước sau:

1. Mở rộng thư mục **Gradle** **Scripts** nếu nó chưa được mở, sau đó nhấp đúp vào tệp **build.gradle (Module:app).**  
   Nội dung của tệp sẽ hiển thị trong trình chỉnh sửa mã.
2. Trong khối defaultConfig, thay đổi giá trị của minSdkVersion thành 17, như minh họa bên dưới (giá trị ban đầu được đặt là 15).

Trình chỉnh sửa mã hiển thị một thanh thông báo ở phía trên cùng với liên kết **Sync Now**.

**Bước 2: Đồng bộ cấu hình Gradle mới**

Khi bạn thực hiện thay đổi trong các tệp cấu hình build của dự án, Android Studio yêu cầu bạn đồng bộ (sync) các tệp của dự án để có thể nhập các thay đổi cấu hình build và chạy một số kiểm tra nhằm đảm bảo rằng cấu hình sẽ không gây ra lỗi khi biên dịch.

Để đồng bộ tệp dự án, hãy nhấp vào **Sync** **Now** trên thanh thông báo xuất hiện sau khi thực hiện thay đổi (như trong hình trước đó), hoặc nhấp vào biểu tượng **Sync Project with Gradle Files** trên thanh công cụ.

Khi quá trình đồng bộ Gradle hoàn tất, thông báo Gradle build finished sẽ xuất hiện ở góc dưới bên trái của cửa sổ Android Studio.

Để tìm hiểu sâu hơn về Gradle, hãy tham khảo tài liệu Build System Overview và Configuring Gradle Builds.

**Nhiệm vụ 6: Thêm câu lệnh log vào ứng dụng của bạn**

Trong nhiệm vụ này, bạn sẽ thêm các câu lệnh Log vào ứng dụng của mình, giúp hiển thị thông báo trong bảng **Logcat**.

Các thông báo Log là một công cụ gỡ lỗi mạnh mẽ, cho phép bạn kiểm tra giá trị, theo dõi luồng thực thi và báo cáo các ngoại lệ.

**Bước 1: Xem bảng Logcat**

Để mở bảng **Logcat**, nhấp vào tab **Logcat** ở phía dưới cửa sổ Android Studio, như hình minh họa dưới đây:

Trong hình trên:

1. Tab **Logcat** dùng để mở và đóng bảng **Logcat**, nơi hiển thị thông tin về ứng dụng khi nó đang chạy. Nếu bạn thêm các câu lệnh Log vào ứng dụng, các thông báo Log sẽ xuất hiện tại đây.
2. Menu Log level được đặt ở chế độ **Verbose** (mặc định), hiển thị tất cả các thông báo Log. Các tùy chọn khác bao gồm **Debug**, **Error**, **Info** và **Warn**.

Bước 2: Thêm câu lệnh log vào ứng dụng của bạn

Các câu lệnh Log trong mã ứng dụng sẽ hiển thị thông báo trong bảng Logcat. Ví dụ:

Các thành phần của thông báo log:

* Log: Lớp Log dùng để gửi thông báo log đến bảng Logcat
* d: Mức log **Debug** để lọc và hiển thị thông báo trong bảng Logcat. Các ức log khác bao gồm: **e** cho **Error**, **w** cho **Warn** và **i** cho **Info**.
* “MainActivity”: Đối số đầu tiên là một tag (nhãn), có thể được sử dụng để lọc thông báo trong bảng Logcat. Thông thường, tag này là tên của Activity nơi thông báo được ghi lại. Tuy nhiên, bạn có thể đặt bất kỳ giá trị nào hữu ích cho quá trình gỡ lỗi.

Theo quy ước, các tag log thường được định nghĩa dưới dạng hằng số trong Activity.

* “Hello world”: Đối số thứ hai là thông điệp thực tế.

Thực hiện theo các bước sau:

1. Mở ứng dụng Hello World trong Android Studio và mở tệp MainActivity.
2. Để tự động thêm các thư viện import cần thiết vào dự án (chẳng hạn như android.util.Log để sử dụng Log), thực hiện: với Windows: Vào **File > Settings** và với macOS: Vào **Android Studio > Preferences**.
3. Chọn **Editor > General > Auto Import**. Tích chọn tất cả các ô và đặt **Insert imports on paste** thành **All**.
4. Nhấn **Apply**, sau đó nhấn **OK**.
5. Trong phương thức onCreate() của MainActivity, thêm câu lệnh sau:

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

1. Nếu bảng Logcat chưa mở, hãy nhấp vào tab Logcat ở phía dưới Android Studio để mở.
2. Kiểm tra xem tên mục tiêu (target) và tên gói (package name) của ứng dụng có chính xác không.
3. Thay đổi mức Log trong bảng Logcat thành Debug (hoặc giữ nguyên Verbose nếu có ít thông báo log).
4. Chạy ứng dụng của bạn.

Thông báo sau sẽ xuất hiện trong bảng Logcat:

**Thử thách lập trình**

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

**Thử** **thách**: Bây giờ bạn đã thiết lập xong và quen thuộc với quy trình phát triển cơ bản, hãy thực hiện các bước sau:

1. Tạo một dự án mới trong Android Studio.
2. Thay đổi lời chào “Hello World” thành Happy Birthday to “ và thêm tên của một người vừa có sinh nhật gần đây.
3. (Tùy chọn) Chụp ảnh màn hình ứng dụng đã hoàn thành và gửi email cho ai đó mà bạn quên chúc mừng sinh nhật.
4. Một cách sử dụng phổ biến của lớp Log là ghi lại các ngoại lệ (Java exceptions) khi chúng xảy ra trong chương trình của bạn. Bạn có thể sử dụng một số phương thức hữu ích như Log.e() để thực hiện điều này. Khám phá các phương thức mà bạn có thể sử dụng để đính kèm một ngoại lệ (exception) vào thông báo Log. 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, truy cập trang Android Studio và làm theo hướng dẫn để tải xuống và cài đặt.
* Khi tạo ứng dụng mới, đảm bảo rằng **API 15: Android 4.0.3 IceCreamSandwich** được đặt làm Minimum SDK.
* Để xem cấu trúc phân cấp của ứng dụng trong Project pane, nhấp vào tab **Project** 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 hoặc thay đổi phiên bản thư viện.
* Tất cả mã nguồn và tài nguyên của ứng dụng được đặt trong thư mục app và res. Thư mục java chứa các Activity, bài kiểm tra và các thành phần khác được viết bằng mã nguồn Java. Thư mục res chứa tài nguyên như bố cục (layouts), chuỗi ký tự (strings), và hình ảnh (images).
* Chỉnh sửa tệp AndroidManifest.xml để thêm các thành phần tính năng và quyền (permissions) cho ứng dụng Android. Tất cả các thành phần của ứng dụng, chẳng hạn như nhiều Activity, phải được khai báo trong tệp XML này.
* Sử dụng Android Virtual Device (AVD) Manager để tạo một thiết bị ảo (emulator) chạy ứng dụng của bạn.
* Thêm các câu lệnh Log vào ứng dụng để hiển thị thông báo trong bảng Logcat như một công cụ gỡ lỗi cơ bản.
* Để chạy ứng dụng trên thiết bị Android thực tế bằng Android Studio, hãy bật USB Debugging trên thiết bị: Mở **Cài đặt > Giới thiệu điện thoại (Settings > About phone)**, sau đó nhấn vào **Số bản dựng (Build number)** bảy lần. Quay lại màn hình trước (**Cài đặt**), nhấn vào **Tùy chọn nhà phát triển (Developer options),** sau đó bật **Gỡ lỗi USB (USB Debugging)**.

**Các khái niệm liên quan**

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

**Tìm hiểu thêm**

Tài liệu về Android Studio:

Khác:

**Bài tập về nhà**

**Xây dựng và chạy một ứng dụng:**

* Tạo một dự án Android mới từ mẫu Empty Template.
* Thêm các câu lệnh log với các cấp độ khác nhau trong phương thức onCreate() của MainActivity.
* Tạo một thiết bị giả lập (emulator) với phiên bản Android tùy chọn và chạy ứng dụng.
* Sử dụng bộ lọc trong **Logcat** để tìm các câu lệnh log của bạn và điều chỉnh mức hiển thị chỉ bao gồm debug hoặc error.

**Trả lời các câu hỏi**

Câu 1: Tên của tệp tin layout cho main activity là gì ?

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

Câu 2: Tên của tài nguyên chuỗi (string resource) xác định tên của ứng dụng là gì ?

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

Câu 3: Bạn sử dụng công cụ nào để tạo một trình giả lập (emulator) mới ?

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

Câu 4: Giả sử ứng dụng của bạn có câu lệnh ghi log sau:

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

Bạn sẽ thấy dòng log “MainActivity layout is complete” trong Logcat nếu mức Log được đặt thành tùy chọn nào sau đây? (Gợi ý: Có thể có nhiều đáp án đúng.)

* Verbose
* Debug
* Info
* Warn
* Error
* Assert

**Nộp ứng dụng của bạn để chấm điểm**

Hãy kiểm tra để đảm bảo ứng dụng có các yêu cầu sau:

* Một Activity hiển thị “Hello World” trên màn hình.
* Câu lệnh log trong phương thức onCreate() của MainActivity.
* Mức Log trong **Logcat** chỉ hiển thị lệnh debug hoặc error.

### Văn bản và các chế độ cuộn

### Tài nguyên có sẵn

## Activities

### Activity và Intent

### Vòng đời của Activity và trạng thái

### Intent ngầm định

## Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ

### Trình gỡ lỗi

### Kiểm thử đơn vị

### Thư viện hỗ trợ

# TRẢI NGHIỆM NGƯỜI DÙNG

## Tương tác người dùng

### Hình ảnh có thể chọn

### Các điều khiển nhập liệu

### Menu và bộ chọn

### Điều hướng người dùng

### RecycleView

## Trải nghiệm người dùng thú vị

### Hình vẽ, định kiểu và chủ đề

### Thẻ và màu sắc

### Bố cục thích ứng

## Kiểm thử giao diện người dùng

### Espresso cho việc kiểm tra UI

# LÀM VIỆC TRONG NỀN

## Các tác vụ nền

### AsyncTask

### AsyncTask và AsyncTaskLoader

### Broadcast receivers

## Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền

### Thông báo

### Trình quản lý cảnh báo

### JobScheduler

# LƯU DỮ LIỆU NGƯỜI DÙNG

## Tùy chọn và cài đặt

### Shared preferences

### Cài đặt ứng dụng

## Lưu trữ dữ liệu với Room

### Room, LiveData và ViewModel

### Room, LiveData và ViewModel