**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.

A screenshot of a computer

AI-generated content may be incorrect.

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.

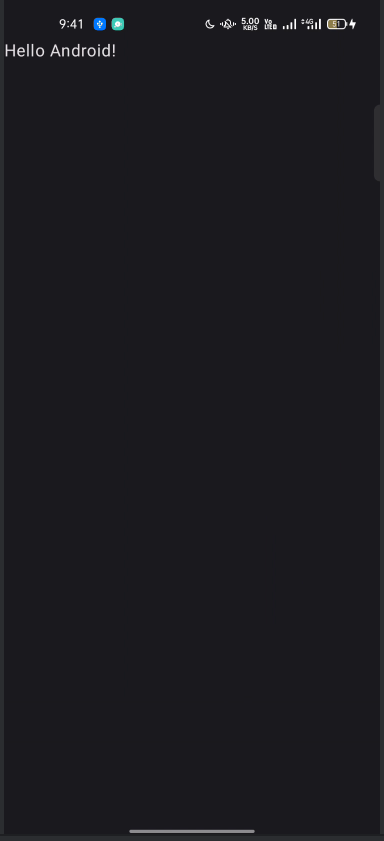
Những gì bạn sẽ làm

* 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 bạn tải thành công Android Studio, bạn sẽ tạo, từ một bản mẫu, một dự án mới cho ứng dụng Hello World. Ứng dụng cơ bản này hiển thị chuỗi “Hello World” trên màn hình của một máy ảo Android hoặc một thiết bị vật lý.

Những gì ứng dụng kết thúc sẽ trông như sau:



**Nhiệm vụ 1: Tải Android Studio**

Android Studio cung cấp một môi trường phát triển tích hợp hoàn chỉnh (IDE) bao gồm một trình chỉnh sửa mã nâng cao và một tập hợp các mẫu ứng dụng. Ngoài ra, nó bao gồm những công cụ để phát triển, gỡ lỗi, kiểm thử và hiệu suất giúp nó nhanh hơn và dễ hơn để phát triển các ứng dụng. Bạn có thể kiểm tra những ứng dụng của bạn cùng một loạt các trình giả lập được cấu hình sẵn hoặc trên thiết bị di động riêng của bạn, xây dựng các ứng dụng sản xuất và công khai trên của hàng Google Play.

Android Studio có sẵn cho các máy chạy Windows hoặc Linux, and cho Macs chạy macOS. OpenJDK mới nhất (Bộ phát triển Java) được bó với Android Studio.

Để tổ chức và chạy với Android Studio, đầu tiên kiểm tra các yêu cầu hệ thống để đảm bảo rằng hệ thống của bạn đáp ứng chúng. Việc cài đặt tương tự cho tất cả các nền tảng. Bất cứ sự khác biệt được ghi nhận dưới đây.

Điều hướng tới trang web của nhà phát triển Android và theo dõi những hướng dẫn để tải xuống và cài đặt Android Studio

Chấp nhận cấu hình mặc định cho tất cả các bước, và đảm bảo rằng tất cả các thành phần đều được chọn để cài đặt.

Sau khi kết thúc cài đặt, Setup Wizard sẽ tải xuống và cài đặt một vài thành phần bổ sung bao gồm Android SDK. Hãy kiên nhẫn, điều này có thể mất một vài thời gian phụ thuộc vào tốc độ mạng của bạn, và một số bước có vẻ dư thừa

Khi mà hoàn thành tải xuống, Android Studio sẽ bắt đầu và bạn đã sẵn sàng để tạo một dự án cá nhân đầu tiên.

**Nhiệm vụ 2: tạo phần mềm Hello World**

Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị “Hello World” để xác minh rằng Android Studio là cài đặt thành công, và để tìm hiểu những điều cơ bản về phát triển với Android Studio.

2.1 tạo một dự án ứng dụng

* + - * 1. Mở Android Studio nếu nó chưa được mở
        2. Trong cửa sổ chính của Android Studio, ấn vào bắt đầu một dự án Android Studio mới
        3. Trong cửa sổ tạo dự dán Android, nhập Hello World cho tên ứng dụng

A screenshot of a computer

AI-generated content may be incorrect.

* + - * 1. Xác minh rằng vị trí dự án mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World của bạn và các dự án Android Studio khác, hoặc thay đổi nó thành thư mục ưa thích của bạn.
        2. Chấp nhận mặc định android.example.com cho tên miền công ty, hoặc tạo một tên miền công ty độc nhất.

Nếu bạn không có kế hoạch công khai ứng dụng của bạn, bạn có thể chấp nhận mặc định. Hãy lưu ý rằng việc thay đổi tên gói của ứng dụng của bạn sau này là công việc thêm phụ

* + - * 1. Bỏ chọn tùy chọn Include C++ support và Include Kotlin support rồi nhấp vào Tiếp theo
        2. Trên màn hình các thiết bị Target Android, điện thoại và máy tính bảng nên được chọn. Đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được thiết lập làm SDK tối thiểu, nếu không hãy sử dụng menu bật lên để thiết lập nó. Đây là các cài đặt được sử dụng bởi các ví dụ trong các bài học của khóa học này. Như của bài viết này, các cài đặt này làm ứng dụng Hello World của bạn tương thích với 97% các thiết bị Android hoạt động trên của hàng Google Play.
        3. Bỏ chọn tùy chọn Include Instant App support và tất cả các tùy chọn khác. Sau đó nhấn tiếp. Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK mục tiêu mà bạn đã chọn, Android Studio sẽ tải xuống chúng tự động.
        4. Cửa sổ thêm sự kiện sẽ hiện ra. Một sự kiện là duy nhất, tập trung vào thứ mà người dung có thể làm. Nó là một thành phần chủ yếu của bất kì ứng dụng Android nào. Một sự kiện thông thường có một bố cục liên quan đến cái mà định nghĩa các yếu tố UI xuất hiện trên một màn hình. Android Studio cung cấp mẫu các sự kiện giúp bạn bắt đầu. với dự án Hello World, chọn Empty Activity như hình dưới đây và nhấn Next.

A screenshot of a computer

AI-generated content may be incorrect.

* + - * 1. Màn hình Configure Activity xuất hiện (khác nhau phụ thuộc vào mẫu mà bạn chọn ở bước trước). Theo mặc định, Empty Activity được cung cấp bởi mẫu được đặt tên là MainActivity. Bạn có thể thay đổi nó nếu bạn muốn, nhưng bài học này sử dụng MainActivity.

A screenshot of a computer

AI-generated content may be incorrect.

* + - * 1. Đảm bảo rằng tùy chọn Generate Layout File đã được chọn. Tên bố cục theo mặc định là activity\_main. Bạn có thể thay đổi nó nếu bạn muốn, nhưng bài học này sử dụng acvitity\_main.
        2. Đảm bảo rằng tùy chọn Backwards Compatibility (AppCompat) đã được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước đó.
        3. Nhấn Finish.

Android Studio tạo ra một thư mục cho dự án của bạn, và xây dựng dự án với Gradle (cái này có thể mất một vài khoảnh khắc).

Mẹo: Xem trang cấu hình trang Nhà phát triển bản dựng của bạn để biết thông tin chi tiết.

Bạn có thể cũng nhìn thấy một thông báo “Tip of the day” cùng với các phím tắt và các mẹo tiện ích khác. Nhấn close để tắt thông báo.

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

1. Nhấn vào tab activity\_main.xml để xem trình chỉnh sửa bố cục.
2. Nhấn vào tab chỉnh sửa bố cục Design, nếu không được chọn, để hiển thị một bản đồ họa của bố cục hiển thị như bên dưới.

A screenshot of a computer

AI-generated content may be incorrect.

1. Nhấn vào tab MainActivity.java để hiển thị trình chỉnh sửa mã như hình bên dưới.

A computer screen shot of a black screen

AI-generated content may be incorrect.

**2.2 Khám phá dự án > ngăn Android**

Trong bài này, bạn sẽ khám phá cách dự án được tổ chức trong Android Studio như thế nào.

Nếu không được chọn, nhấn vào tab Project trong cột tab dọc bên trái của cửa sổ Android Studio. Ngăn Project xuất hiện.

Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, chọn Android từ menu ở phía đầu ngăn Project, như hình bên dưới.

A computer screen shot of a program

AI-generated content may be incorrect.

**Chú ý:** chương này và các chương khác đề cập đến ngăn Project, khi được đặt thành Android, như sau Project > Android.

**2.3 Khám phá thư mục Gradle Scripts**

Hệ thống xây dựng Gradle trong Android Studio giúp nó trở nên dễ dàng để đưa các tệp nhị phân bên ngoài hoặc các thư viện khác vào mô-đun vào bản dựng của bạn dưới dạng phụ thuộc.

Khi bạn tạo một dự án ứng dụng đầu tiên, Project > Android sẽ xuất hiện cùng thư mục Gradle Scripts được mở rộng như hình bên dưới.

A screenshot of a computer program

AI-generated content may be incorrect.

Thực hiện các bước sau để khám phá hệ thống Gradle:

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

Đây là nơi bạn sẽ tìm thấy những tùy chọn cấu hình cái mà phổ biến với tất cả các mô-đun tạo nên dự án của bạn. Mọi dự án Android Studio đều chứa một Gradle cấp cao nhất xây dựng tập tin. Hầu hết thời gian, bạn sẽ không cần làm bất kì thay đổi nào tới tệp tin này. Nhưng nó vẫn tiện ích để hiểu nội dung của nó.

Theo mặc định, tệp xây dựng cấp cao sử dụng khối buildscripts để định nghĩa kho lưu trữ và phụ thuộc Gradle phổ biến cho tất cả các mô-đun trong dự án. Khi sự phụ thuộc của bạn là cái gì đó khác hơn một thư viện cục bộ hoặc cây tệp, Gradle tìm các tệp trong bất kì kho lưu trữ trực tuyến nào được chỉ định trong khối kho lưu trữ của tệp này. Theo mặc định, những dự án Android Studio mới khai báo JCenter và Google (cái mà bao gồm kho lưu trữ Google Maven) làm vị trí kho lưu trữ:

// Top-level build file where you can add configuration options common to all sub-projects/modules.

plugins {

alias(libs.plugins.android.application) apply false

}

1. Tìm tệp build.gradle(Module:app).

Ngoài tệp build.gradle cấp độ dự án, mỗi mô-đun có một tệp build.gradle của riêng mình, cái mà cho phép bạn cài đặt xây dựng cấu hình cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có duy nhất một mô-đun). Định cấu hình các cài đặt xây dựng này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, như bổ sung các kiểu xây dựng và flavors sản xuất. Bạn có thể cũng ghi đè những cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp độ cao.

Tệp này thì thường là tệp để chỉnh sửa khi thay đổi cấu hình cấp độ ứng dụng, như khai báo các phụ thuộc trong phần phụ thuộc. Bạn có thể khai báo một phụ thuộc thư viện sử dụng một trong một số cấu hình phục thuộc khác nhau. Mỗi cấu hình phụ thuộc cung cấp các hướng dẫn Gradle khác nhau về cách sử dụng thư viện như thế nào? Ví dụ, việc thực hiện câu lệnh của cây tệp (dir: 'libs', include: ['\*.jar']) them sự phụ thuộc của tất cả các tệp.

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

plugins {

alias(libs.plugins.android.application)

}

android {

namespace = "com.example.myapplication"

compileSdk = 35

defaultConfig {

applicationId = "com.example.myapplication"

minSdk = 23

targetSdk = 35

versionCode = 1

versionName = "1.0"

testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

isMinifyEnabled = false

proguardFiles(

getDefaultProguardFile("proguard-android-optimize.txt"),

"proguard-rules.pro"

)

}

}

compileOptions {

sourceCompatibility = JavaVersion.VERSION\_11

targetCompatibility = JavaVersion.VERSION\_11

}

}

dependencies {

implementation(libs.appcompat)

implementation(libs.material)

implementation(libs.activity)

implementation(libs.constraintlayout)

testImplementation(libs.junit)

androidTestImplementation(libs.ext.junit)

androidTestImplementation(libs.espresso.core)

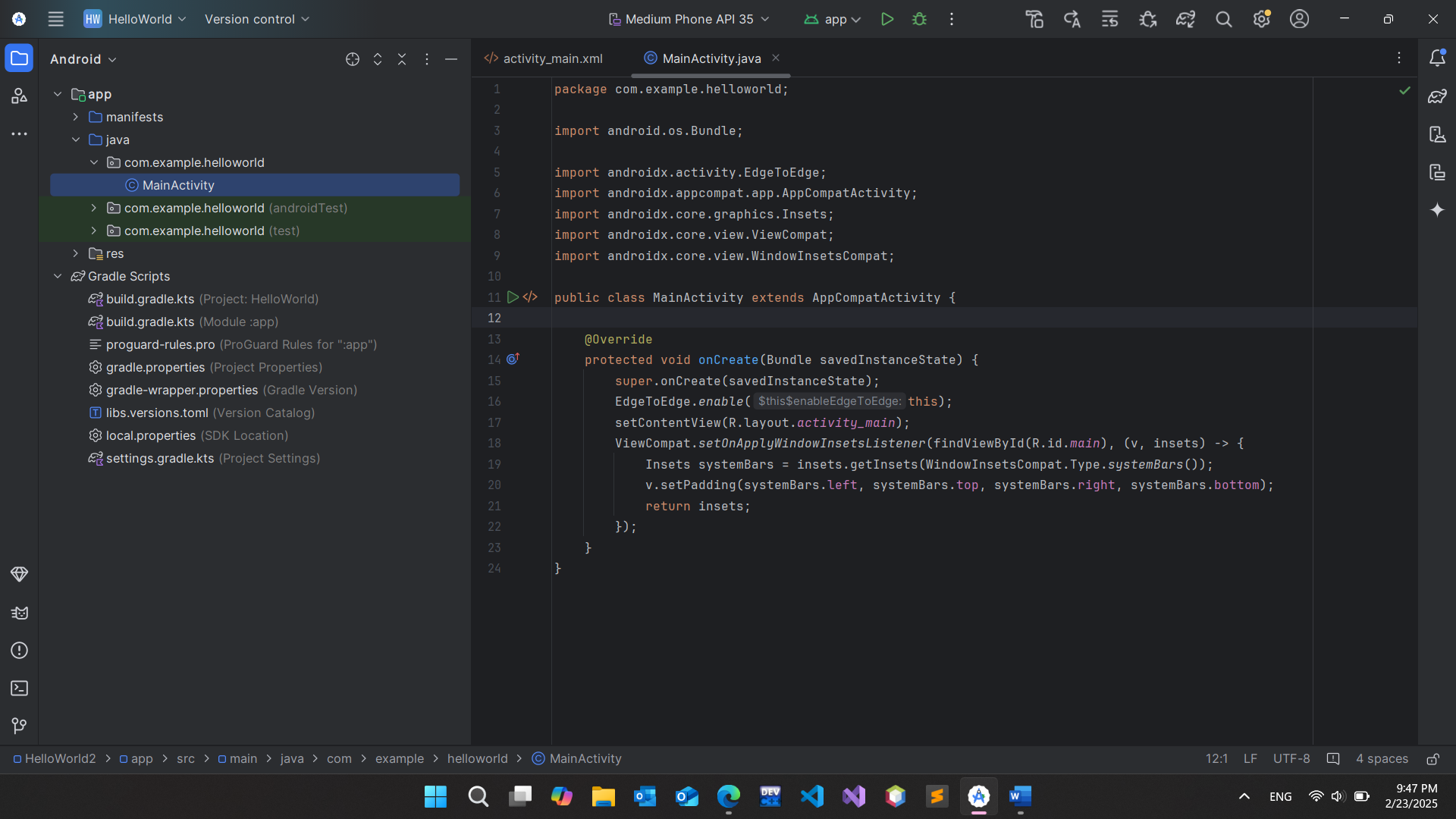
}

1. Nhấn hình tam giác để đóng Gradle Scripts.

**2.4 Khám phá các thư mục ứng dụng và res**

Tất cả các mã và tài nguyễn cho ứng dụng đều nằm trong các thư mục ứng dụng và res.

1. Mở rộng thư mục **app**, thư mục **java**, và thư mục com.example.android.helloworld để thấy tệp java MainActivity. Nhấn đúp chuột sẽ mở tệp trong trình chỉnh sửa mã.



Thư mục java bao gồm các tệp class java trong ba thư mục con, như thể hiện trong hình trên. Thư mục com.example.hello.helloworld (hoặc tên miền bạn đã chỉ định) chứa tất cả các tệp cho gói ứng dụng. Hai thư mục khác được sử dụng cho kiểm tra và miêu tả trong bài học khác. Đối với ứng dụng HelloWorld, chỉ có duy nhất một gói và nó bao gồm MainActivity.java. Tên của sự kiện đầu tiên (màn hình) mà người dùng nhìn thấy, cái mà cũng khởi tạo tài nguyên rộng khắp ứng dụng, thường được gọi là MainActivity (phần mở rộng tệp bị bỏ qua trong khung Project > Android).

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

A screenshot of a computer

AI-generated content may be incorrect.

Thư mục res giữ tài nguyên, như bố cục, chuỗi, và ảnh. Một sự kiện thường được liên kết với bố cục các chế độ xem UI được xác định như một tệp XML. Tệp này thường được đặt tên theo sự kiện của nó.

**2.5 Khám phá thư mục manifests**

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

1. Mở rộng thư mục manifests
2. Mở tệp AndroidManifest.xml

Tệp AndroidManifest.xml miêu tả tất cả các thành phần của ứng dụng của bạn. Tất cả các thành phần cho một ứng dụng, như mỗi sự kiện, phải được khai báo trong tệp xml này. Trong những bài học khác bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để giới thiệu, xem tổng quát về bảng kê khai ứng dụng.

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

Trong nhiệm vụ này, bạn sẽ sử dụng trình quản lý thiết bị ảo Android (AVD) để tạo một thiết bị ảo (cũng như một trình giả lập) cái mà mô phỏng cấu hình cho một loại thiết bị Android cụ thể, và sử dụng thiết bị ảo đó để chạy ứng dụng. Chú ý là trình giả lập Android có các yêu cầu bổ sung ngoài những yêu cầu hệ thống cơ bản cho Android Studio.

Sử dụng trình quản lý AVD, bạn xác định các đặc tính phần cứng của một thiết bị, mức độ API, lưu trữ, bề ngoài và các thuộc tính khác của nó và nó lưu trữ như một thiết bị ảo. Với những thiết bị ảo, bạn có thể kiểm tra các ứng dụng trên những cấu hình thiết bị khác nhau (như các máy tính bảng và điện thoại) với các mức độ API khác nhau, không phải sử dụng các thiết bị vật lý.

* 1. **Tạo một thiết bị Android ảo (AVD)**

Để chạy một trình giả lập trên máy tính của bạn, bạn phải tạo một cấu hình để miêu tả thiết bị ảo đó.

1. Trong Android Studio, chọn Tools > Android > AVD Manager, hoặc nhấn vào biểu tượng AVD Manager trên thanh công cụ. Màn hình Your Virtual Devices xuất hiện. Nếu bạn đã tạo những thiết bị ảo, màn hình hiển thị chúng (như hiển thị trong hình ảnh dưới), nếu không bạn thấy một danh sách trống.
2. Nhấn vào +Create Virtual Device. Cửa số Select Hardware xuất hiện hiển thị một danh sách những thiết bị phần cứng được cấu hình trước. Với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo, độ phân giải màn hình theo pixel, và mật độ pixel.
3. Chọn một thiết bị như Nexus 5x hoặc Pixel XL, và nhấn Next. Màn hình System Image xuất hiện.
4. Nhấn vào tab Recommended nếu nó không được chọn, và chọn phiên bản nào của hệ thống Android để chạy trên thiết bị ảo (như Oreo).

Có nhiều phiên bản có sẵn hơn so với hiển thị trong tab Recommended. Nhìn vào x86 Images và những tab Other Images để xem chúng.

Nếu một đường dẫn Download được hiển thị bên cạnh hình ảnh hệ thống bạn muốn sử dụng, nó vẫn chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào Finish khi nó hoàn thành.

1. Sau khi chọn một ảnh hệ thống, nhấn Next. Cửa sổ Android Virtual Device xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấn Finish.

**3.2 Chạy ứng dụng trên thiết bị ảo**

Trong nhiệm vụ này, bạn sẽ kết thúc chạy ứng dụng Hello World của bạn.

1. Trong Android Studio, chọn Run > Run app hoặc nhấn biểu tượng Run trên thanh công cụ.
2. Cửa sổ Select Deployment Target, dưới Available Virtual Devices, chọn thiết bị ảo, cái mà bạn đã tạo và nhấn OK

Trình giả lập bắt đầu và khởi động giống như một thiết bị vật lý. Tùy thuộc vào tốc độ của máy tính của bạn, điều này có thể mất một lúc. Ứng dụng của bạn được xây dựng và một khi trình giả lập đã sẵn sàng, Android Studio sẽ tải ứng dụng lên trình giả lập và chạy nó.

Bạn sẽ thấy ứng dụng Hello World như được hiển thị trong hình sau.

Mẹo: Khi thử nghiệm trên một thiết bị ảo, đó là một thực tế tốt để khởi động nó một lần, vào đầu phiên của bạn. Bạn không nên đóng nó cho đến khi bạn hoàn thành 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, nhấp vào nút X ở đầu trình giả lập, chọn thoát khỏi menu hoặc nhấn Control-Q trong Windows hoặc Command-Q trong macOS.

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

Trong nhiệm vụ cuối cùng này, bạn sẽ hạy ứng dụng của bạn trên một thiết bị vật lý như một điện thoại hoặc máy tính bảng. Bạn nên luôn luôn kiểm tra các ứng dụng của bạn trên cả những thiết bị vật lý và những thiết bị ảo.

Những thứ bạn cần:

* Một thiết bị Android như điện thoại hoặc máy tính bảng.
* Một cáp dữ liệu để kết nối thiết bị Android của bạn với máy tính của bạn qua cổng USB.
* Nếu bạn đang sử dụng một hệ thống Linux hoặc Windows, bạn có thể cần thực hiện các bước bổ sung để chạy trên một thiết bị phần cứng. Kiểm tra tài liệu Using Hardware Devices. Bạn cũng có thể cần cài đặt trình điều khiển USB thích hợp cho thiết bị của bạn. Cho trình điều khiển USD dựa trên Windows, hãy xem trình điều khiển USB OEM.

4.1 Bật gỡ lỗi USB

Để cho Android Studio giao tiếp với thiết bị của bạn, bạn phải bật gỡ lỗi USB trên thiết bị Android của bạn. Điều đó được bật trong cài đặt tùy chọn nhà phát triển của thiết bị của bạn.

Trên Android 4.2 và cao hơn, màn hình developer options được ẩn theo mặc định. Để hiển thị các tùy chọn của nhà phát triển và cho phép gỡ lỗi USB.

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

4.2 Chạy ứng dụng của bạn trên một thiết bị

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

1. Kết nối thiết bị của bạn tới máy phát triển của bạn bằng một cáp USB.
2. Nhấn nút Run trên thanh công cụ. Cửa sổ Select Deployment Target mở cùng với danh sách của những trình giả lập có sẵn và những thiết bị được kết nối.
3. Chọn thiết bị của bạn, và nhấn OK

Cài đặt Android Studio và chạy những ứng dụng trên thiết bị của bạn.

Troubleshooting

Nếu Android Studio của bạn không nhận ra thiết bị của bạn, thử lại như sau:

1. Rút ra 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 nó thông báo “unauthorized”, làm theo các bước sau:

1. Tháo thiết bị ra
2. Trên thiết bị, mở ứng dụng Developer Options trong cài đặt
3. Nhấn thu hồi các 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 bạn. Xem tài liệu sử dụng thiết bị phần cứng.

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

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

* 1. Thay đổi phiên bản SDK thấp nhất cho ứng dụng

Làm theo các bước sau:

1. Mở thư mục Gradle Scripts nếu nó không được mở, và nhấn đúp chuột vào tệp build.gradle(Module:app).
2. Cùng với khối defaultConfig, thay đổi giá trị của minSdkVersion đến 17 như hiển thị bên dưới (nó ban đầu được đặt là 15)

Trình chỉnh sửa mã hiển thị một thanh thông báo ở đầu với liên kết Sync Now.

* 1. Đồng bộ hóa cấu hình Gradle mới

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

Để đồng bộ hóa các tệp dự án, 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 SYNC Files với biểu tượng File Files trong thanh công cụ.

Khi việc đồng bộ hóa Gradle kết thúc, bản dựng thông điệp Gradle hoàn thành xuất hiện ở góc dưới bên trái của cửa sổ Android Studio.

Để có cái nhìn sâu sắc hơn về Gradle, hãy xem tổng quan về hệ thống xây dựng và định cấu hình tài liệu xây dựng Gradle.

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

Trong nhiệm vụ này, bạn sẽ thêm log statementsvào ứng dụng của mình, cụm từ này hiển thị thông báo trong ngăn Logcat.

Log mesageslà 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 các trường hợp ngoại lệ.

6.1 View the Logcat pane

Để xem ngăn Logcat, hãy nhấp vào thẻ Logcat ở cuối cửa sổ Android Studio như trong hình bên dưới.

A screenshot of a computer

AI-generated content may be incorrect.

Trong hình trên:

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

2. Menu cấp độ nhật ký được đặt thành verbose (mặc định), hiển thị tất cả các thông báo nhật ký. Các cài đặt khác bao gồm gỡ lỗi, lỗi, thông tin và cảnh báo.

6.2 . Thêm log statements vào ứng dụng của bạn

Log statements trong mã ứng dụng của bạn sẽ hiển thị thông báo trong ngăn Logcat. Chẳng hạn: Log.d("MainActivity", "Hello World");

Các phần của thông điệp là:

Log: Lớp Nhật ký để gửi tin nhắn nhật ký đến ngăn Logcat.

d: Cài đặt mức Debug log để lọc thông báo nhật ký hiển thị trong ngăn Logcat. Nhật ký khác các cấp độ 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 từ đó tin nhắn

Nguồn gốc. 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 Activity:

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 bắt buộc để sử dụng Nhật ký ), chọn Cài đặt > tệp trong Windows hoặc Android Studio > Tùy chọn trong macOS.

3. Chọn Trình chỉnh sửa > Chung >Tự động nhập . Chọn tất cả các hộp kiểm và đặt Chèn nhập vào dán vào Tất cả .

4. Nhấp vào Áp dụng 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() sẽ nhìn như đoạn mã bên dưới:

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Log.d("MainActivity", "Hello World");

}

6.Nếu ngăn Logcat chưa mở, hãy nhấp vào thẻ Logcat ở cuối Android Studio

để mở nó.

7. Kiểm tra xem tên mục tiêu và tên gói của ứng dụng có chính xác không.

8. Thay đổi cấp nhật ký trong ngăn Logcat thành Debug (hoặc để nguyên Verbose vì có ít thông báo nhật ký).

9. Chạy ứng dụng của bạn.

Sau đó thông báo sẽ xuất hiện trong Logcat pane:

11 24 14:06:59.001 4696 4696/? D/MainActivity: Hello World

Thử thách coding

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 thực hiện sau:

3. Tạo một dự án mới trong Android Studio.

2. Thay đổi lời chào "Hello World" thành "Happy Birthday to" và tên của ai đó có sinh nhật gần đây.

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

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 một ngoại lệ với log message. Sau đó, viết mã trong để kích hoạt và ghi nhật ký 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 một ứng dụng mới, hãy đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt là SDK tối thiểu.
* Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy nhấp vào tab Dự án ở đường dọc
* tab rồi 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. Tiếng 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. Tất cả các thành phần cho một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML.
* 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 Cài đặt > Giới thiệu về điện thoại và nhấn vào Số xây dựng bảy lần. Quay lại màn hình trước đó ( Cài đặt ) và nhấn vào Tùy chọn nhà phát triển . Chọn Gỡ lỗi USB.

**Các khái niệm liên quan**

Tài liệu khái niệm liên quan là trong 1.0: Giới thiệu về Android và 1.1 Ứng dụng Android đầu tiên của bạn

**Learn more**

Tài liệu Android 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 các câu lệnh ghi nhật ký cho các mức nhật ký khác nhau trong oncreate () trong hoạt động chính.

● Tạo trình giả lập cho một thiết bị, nhắm mục tiêu bất kỳ phiên bản nào của Android bạn thích và chạy ứng dụng.

Answer these questions

Question 1

What is the name of the layout file for the main activity?

● MainActivity.java

● AndroidManifest.xml

● activity\_main.xml

● build.gradle

Question 2

What is the name of the string resource that specifies the application's name?

● app\_name

● xmlns:app

● android:name

● applicationId

Question 3

Which tool do you use to create a new emulator?

● Android Device Monitor

● AVD Manager

● SDK Manager

● Theme Editor

Question 4

Assume that your app includes this logging statement:



You see the statement "MainActivity layout is complete" in the Logcat pane if the Log level menu is

set to which of the following? (Hint: multiple answers are OK.)

● Verbose

● Debug

● Info

● Warn

● Error

● Assert

**Gửi ứng dụng của bạn để chấm điểm**

Kiểm tra để đảm bảo ứng dụng có như sau:

● Một hoạt động hiển thị "Hello World" trên màn hình.

● Các câu lệnh đăng nhập trong oncreate () trong hoạt động chính.

● Cấp độ nhật ký trong ngăn logcat chỉ hiển thị các câu lệnh Gỡ lỗi hoặc lỗi.

**Bài 1.2 Phần A: UI tương tác đầu tiên của bạn**

**Giới thiệu**

Giao diện người dùng (UI) xuất hiện trên màn hình của thiết bị Android bao gồm một hệ thống phân cấp các đối tượng được gọi là chế độ xem - mọi yếu tố của màn hình là một chế độ xem. Lớp xem đại diện cho khối xây dựng cơ bản cho tất cả các thành phần UI và lớp cơ sở cho các lớp cung cấp các thành phần UI tương tác như nút, hộp kiểm và các trường nhập văn bản. Các phân lớp xem thường được sử dụng được mô tả qua một số bài học bao gồm:

TextView để hiển thị văn bản.

● EditText để cho phép người dùng nhập và chỉnh sửa văn bản.

● Button and other clickable elements (such as RadioButton, CheckBox, and Spinner) để cung cấp hành vi tương tác.

● ScrollView and RecyclerView để hiển thị các mục có thể cuộn.

● ImageView để hiển thị hình ảnh.

● ConstraintLayout and LinearLayout để chứa các yếu tố View khác và định vị chúng.

Mã Java hiển thị và điều khiển UI được chứa trong một lớp mở rộng hoạt động. Một hoạt động thường được liên kết với bố cục các chế độ xem UI được xác định là tệp XML (Ngôn ngữ đánh dấu mở rộng). Tệp XML này thường được đặt tên theo hoạt động của nó và xác định bố cục của các phần tử xem trên màn hình.

Ví dụ: mã chính trong ứng dụng Hello World hiển thị một bố cục được xác định trong tệp bố cục Activity\_main.xml, bao gồm một văn bản với văn bản "Hello World".

Trong các ứng dụng phức tạp hơn, một hoạt động có thể thực hiện các hành động để phản hồi vòi của người dùng, vẽ nội dung đồ họa hoặc yêu cầu dữ liệu từ cơ sở dữ liệu hoặc Internet. Bạn tìm hiểu thêm về lớp hoạt động trong một bài học khác.

Trong thực tế này, bạn tìm hiểu cách tạo ứng dụng tương tác đầu tiên của bạn, một ứng dụng cho phép tương tác người dùng. Bạn tạo một ứng dụng bằng mẫu hoạt động trống. Bạn cũng học cách sử dụng trình chỉnh sửa bố cục để thiết kế bố cục và cách chỉnh sửa bố cục trong XML. Bạn cần phát triển các kỹ năng này để bạn có thể hoàn thành các thực tiễn khác trong khóa học này.

**Những gì bạn nên biết**

Bạn nên quen thuộc với:

● Cách cài đặt và mở Android Studio.

● Cách tạo ứng dụng Helloworld.

● Cách chạy ứng dụng Helloworld.

**Những gì bạn sẽ học**

● Cách tạo một ứng dụng có hành vi tương tác.

● Cách sử dụng trình chỉnh sửa bố cục để thiết kế bố cục.

● Cách chỉnh sửa bố cục trong XML.

● Rất nhiều thuật ngữ mới. Kiểm tra các từ vựng và khái niệm chú giải cho các định nghĩa thân thiện.

**Những gì bạn sẽ làm**

● Tạo một ứng dụng và thêm hai phần tử nút và TextView vào bố cục.

● Thao tác từng phần tử trong ràng buộc để ràng buộc chúng vào lề và các phần tử khác.

● Thay đổi thuộc tính phần tử UI.

● Chỉnh sửa bố cục của ứng dụng trong XML.

● Trích xuất các chuỗi được mã hóa cứng vào tài nguyên chuỗi.

● Thực hiện các phương thức Nhấp chuột để hiển thị thông báo trên màn hình khi người dùng nhấn từng nút.

**Tổng quan về ứng dụng**

Ứng dụng Hellotoast bao gồm hai phần tử nút và một TextView. Khi người dùng nhấn nút đầu tiên, nó sẽ hiển thị một thông báo ngắn (bánh mì nướng) trên màn hình. Nhấn vào nút thứ hai tăng bộ đếm "nhấp chuột" được hiển thị trong TextView, bắt đầu từ 0.

Đây là những gì ứng dụng đã hoàn thành trông như thế nào:

A yellow and blue sign with a number 0

AI-generated content may be incorrect.

**Nhiệm vụ 1: Tạo và khám phá một dự án mới**

Trong thực tế này, bạn thiết kế và thực hiện một dự án cho ứng dụng Hellotoast. Một liên kết đến mã giải pháp được cung cấp ở cuối

1.1 Tạo dự án Android Studio

14. Bắt đầu Android Studio và tạo một dự án mới với các tham số sau:

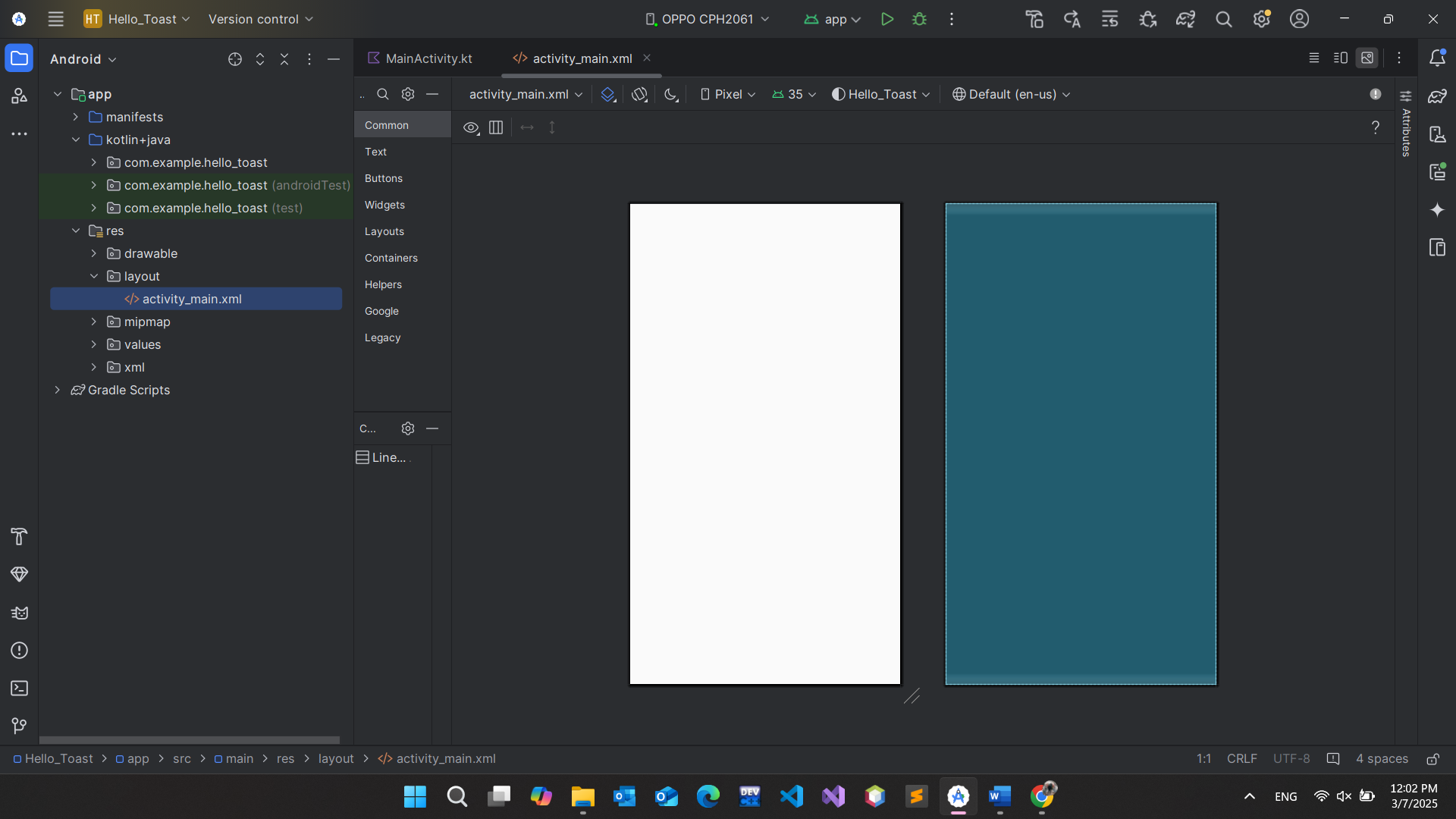
|  |  |
| --- | --- |
| Attribute | Value |
| Application Name | **Hello Toast** |
| Company Name | **com.example.android** (or your own domain) |
| Phone and Tablet Minimum SDK | **API15: Android 4.0.3 IceCreamSandwich** |
| Template | **Empty Activity** |
| Generate Layout file box | Selected |
| Backwards Compatibility box | Selected |

15. Chọn Run> Run app hoặc nhấp vào Run icon  trên thanh công cụ để xây dựng và thực hiện ứng dụng trên trình giả lập hoặc thiết bị của bạn.

**1.2 Khám phá trình chỉnh sửa bố cục**

Android Studio cung cấp trình soạn thảo bố cục để nhanh chóng xây dựng bố cục các yếu tố giao diện người dùng (UI) của ứng dụng. Nó cho phép bạn kéo các phần tử vào một thiết kế trực quan và chế độ xem bản thiết kế, định vị chúng trong bố cục, thêm các ràng buộc và đặt các thuộc tính. Các ràng buộc xác định vị trí của phần tử UI trong bố cục. Một ràng buộc đại diện cho một kết nối hoặc căn chỉnh với chế độ xem khác, bố cục cha hoặc hướng dẫn vô hình.

Khám phá trình soạn thảo bố cục và tham khảo hình bên dưới khi bạn làm theo các bước được đánh số:



1. Trong thư mục app>res> thư mục layout trong Project> Android, nhấp đúp vào tệp Activity\_main.xml để mở nó, nếu nó chưa mở.

2. Nhấp vào tab Design nếu nó chưa được chọn. Bạn sử dụng tab Design để thao tác các phần tử và bố cục và tab Text để chỉnh sửa mã XML cho bố cục.

3. Palettes hiển thị các thành phần UI mà bạn có thể sử dụng trong bố cục của ứng dụng.

4. Component tree hiển thị phân cấp chế độ xem các phần tử UI. Xem các yếu tố được tổ chức thành một hệ thống phân cấp cây của cha mẹ và con cái, trong đó một đứa trẻ thừa hưởng các thuộc tính của cha mẹ của nó. Trong hình trên, TextView là con của ConstraintLayout. Bạn sẽ tìm hiểu về những yếu tố này sau này trong bài học này.

5. Các tấm thiết kế và kế hoạch chi tiết của trình soạn thảo bố cục hiển thị các phần tử UI trong bố cục. Trong hình trên, bố cục chỉ hiển thị một yếu tố: một quan điểm văn bản hiển thị "Hello World".

6. Tab Attributes hiển thị ngăn Attributes để cài đặt Thuộc tính cho phần tử UI.

Mẹo: Xem Xây dựng UI với trình chỉnh sửa bố cục để biết chi tiết về việc sử dụng trình soạn thảo bố cục và gặp gỡ Android Studio để biết tài liệu Android Studio đầy đủ.

2.1 Kiểm tra các ràng buộc phần tử

Thực hiện theo các bước sau:

1. Open Activity\_main.xml từ ngăn dự án> Android nếu nó chưa mở. Nếu

Tab Thiết kế chưa được chọn, nhấp vào nó. Nếu không có kế hoạch chi tiết, hãy nhấp vào nút Chọn Surface Surface trong thanh công cụ và chọn Design + Blueprint.

2. Công cụ AutoConnect cũng được đặt trên thanh công cụ. Nó được bật theo mặc định. Đối với bước này, đảm bảo rằng công cụ không bị vô hiệu hóa.

3. Nhấp vào nút Zoom in để phóng to thiết kế và bảng kế hoạch chi tiết để nhìn cận cảnh.

4. Chọn TextView trong ngăn cây thành phần. TextView "Hello World" được đánh dấu trong các tấm thiết kế và kế hoạch chi tiết và các ràng buộc cho phần tử có thể nhìn thấy được.

5. Tham khảo hình hoạt hình dưới đây cho bước này. Nhấp vào tay cầm tròn ở phía bên phải của TextView để xóa ràng buộc ngang liên kết khung nhìn vào phía bên phải của bố cục. TextView nhảy sang phía bên trái vì nó không còn bị giới hạn ở phía bên phải. Để thêm lại ràng buộc ngang, nhấp vào cùng một tay cầm và kéo một đường vào bên phải của bố cục.

### Giao diện người dùng tương tác đầu tiên

### Trình chỉnh sửa bố cục

### 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