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

Đây là những gì bạn thấy sau khi hoàn thành ứng dụng:

A screen shot of a phone

AI-generated content may be incorrect.

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

Android Studio cung cấp một môi trường phát triển hoàn chỉnh(IDE), bao gồm trình chỉnh sửa mã nâng cao và một bộ ứng dụng mẫu. Ngoài ra nó còn chứa các công cụ cho phát triện, gỡ lỗi, thử nghiệm và hiệu suất đó giúp phát triển ứng dụng nhanh và dễ dàng hơn. Bạn có thể thử nghiệm ứng dụng của bạn với một loạt các trình giả lập được cấu hình sẵn trên thiết bị di động của bạn, xây dựng ứng dụng sản xuất và đưa nó lên của hàng Google Play.

Android Studio có sẵn cho máy tính sử dụng hệ điểu hành Windows hoặc Linux, và Macs sử dụng hệ điều hành macOS. Phiên bản OpenJDK mới nhất được đi kèm với Android Studio.

Để thiết lập và chạy Android Studio , đầu tiên kiểm tra system requirements để đảm bảo hệ thống của bạn đáp ứng được nó. Việc cài đặt tương tự cho tất cả các nền tảng khác. Các sự khác biệt được lưu ý ở bên dưới.

* Truy cập vào trang web của nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.
* Chấp nhận cấu hình mặc định tất cả các bước và đảm bảo rằng tất cả các thành phần được chọn để cài đặt.
* Sau khi hoàn tất cài đặt, the Setup Wizard sẽ tải xuống và cài đặt một số thư viện bổ sung bao gồm Android SDK. Hãy kiên nhẫn điều này có thể tốn một chút thời gian tùy thuộc vào tốc độ internet của bạn và một số bước này có thể không cần đến.
* Khi cài đặt thành công, Android Studio sẽ khởi động và bạn hãy sẵn sàng để tạo dự án đầu tiên.

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

Ở nhiệm vụ này bạn sẽ tạo một ứng dụng hiển thị trên màn hình “Hello World” để kiểm tra Android Studio được cài đặt chính xác, và để học một số kiến thức cơ bản của phát triển với Androi Studio.

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu chưa mở.

2. Ở cửa sổ chính của **Welcome to Android Studio**, chọn **Start a new Android Studio  project** .

3. Trong cửa sổ **Create Android Project**, nhập Hello World cho Application name

A screenshot of a computer

AI-generated content may be incorrect.

4. Xác minh rằng trong Project location là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác, hoặc thay đổi thành thư mục bạn muốn.

5. Chấp nhận android.example.com cho Company Domain hoặc tạo tên miền công ty duy nhất.

6. Bỏ các tùy chọn **Include C++ support and Include Kotlin support** , và chọn **Next**.

7. Trên màn hình **e Target Android Devices**, nên chọn **Phone and Tablet**. Đảm bảo API 15: Android 4.0.3 IceCreamSandwich được cài đặt là SDK nhỏ nhất.Nếu khôg, hãy sử dụng menu popup để thiết lập

A screenshot of a computer

AI-generated content may be incorrect.

Đây là các cài đặt được sử dụng trong ví dụ của khóa học này. Khi viết hướng dẫn này, các cài đặt tạo ra ứng dụng Hello world tương thích với 97% thiết bị Android đang hoạt động trên cửa hàng Google Play.

8. Bỏ chọn **Include Instant App support** và tất cả các lựa chọn khác. Sau khi chọn Next, nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK bạn đã chọn, Android Studio sẽ tự động cài đặt chúng.

9. Cửa số **Add an Activity** xuất hiện. Một Activity là một thứ duy nhất tập trung mà người dùng có thể làm. Nó là một thành phần quan trọng của các ứng dụng Android. Một Activity thường có bố cục được liên kết với nó để xác định thành phần giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp mẫu Activity để giúp bạn bắt đầu. Chọn **Empty Activity** cho dự án Hello World, và chọn Next.

A screenshot of a computer

AI-generated content may be incorrect.

10. Màn hình **Configure Activity** xuất hiện ( Khác nhau tùy thuộc mẫu bạn đã chọn). Với Activity trống mặc định cung cấp cho bạn một mẫu có tên MainActivity. Bạn có thể thay đổi nếu bạn muốn, nhưng ở khóa học sử dụng MainActivity.

A screenshot of a computer

AI-generated content may be incorrect.

11. Đảm bảo rằng lựa chọn **Generate Layout file** đã được chọn. Giao diện có tên mặc định là activity\_Main. Bạn có thể thay đổi nếu bạn muốn, nhưng khóa học sử dụng activity\_main.

12. Hãy đảm bảo rằng lựa chọn **Backwards Compatibility** (App Compat) được chọn. Việc này đảm bảo ứng dụng của bạn sẽ tương thích với phiên bản Android trước.

13. Chọn **Finish.**

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

TIP: Xem Configure your build trong trang nhà phát triển để biết thêm thông tin chi tiết.

Bạn cũng có thể xem thông điệp “tip of the day” với phím tắt và các mẹo hữu ích khác . Chọn Close để đóng thông điệp.

### Trình chỉnh sửa bố cục

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

* Chọn tab activity\_main.xml để xem trình chỉnh sửa bố cục
* Chọn tab giao diện chỉnh sửa Design, nếu chưa được chọn, để hiển thị biểu tượng đồ họa của bố cục như hình dưới đây.

A screenshot of a computer

AI-generated content may be incorrect.

3. Chọn tab MainActivity để xem trình chỉnh sửa mã như hình dưới đây.

A screenshot of a computer

AI-generated content may be incorrect.

2.2 Khám phá the project > Android pane

Trong thực tế bạn sẽ khám phá cách tổ chức dự án trong Android studio.

1. Nếu chưa được chọn , chọn **Projiect** trong cột tab dọc ở bên trái cửa sổ Android Studio. Ngăn dự án xuất hiện.

2. Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, chọn **Android** trong menu popup ở ngăn phía trên của dự án, như hình ảnh dưới đây.

A screenshot of a computer

AI-generated content may be incorrect.

2.3 Khám phá thư mục Gradle Scripts.

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

Khi ban tạo một dự án ứng dụng, ngăn **Project > Android** sẽ xuất hiện với thư mục Gradle Scripts được mở rộng như hình bên dưới.

A screenshot of a computer

AI-generated content may be incorrect.

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

1.Nếu thư mục Gradle Scripts không 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.

2. Tìm tệp build.gradle(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 xây dựng Gradle cấp cao. 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 dựng cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ Gradle và các phụ thuộc chung cho tất cả các mô-đun trong dự án. Khi phụ thuộc của bạn là thứ gì đó khác với 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. Theo mặc định, các dự án Android Studio mới khai báo JCenter và Google (bao gồm kho lưu trữ Google Maven ) là các vị trí kho lưu trữ:

A white rectangular object with black lines

AI-generated content may be incorrect.

3. Tìm tệp build.gradle(Module:app).

Ngoài tệp build.gradle cấp dự án, mỗi mô-đun đều có tệp build.gradle riêng , cho phép bạn định cấu hình cài đặt bản dựng cho từng mô-đun cụ thể ( ứng dụng HelloWorld chỉ có một mô-đun). Định cấu hình các cài đặt bản dựng này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như các loại bản dựng bổ sung và hương vị sản phẩm. Bạn cũng có thể ghi đè cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp cao nhất. Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phụ thuộc trong phần phụ thuộc. Bạn có thể khai báo phụ thuộc thư viện bằng một trong một số cấu hình phụ thuộc 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 phụ thuộc của tất cả các tệp “.jar” bên trong thư mục libs. Sau đây là tệp build.gradle(Module:app) cho ứng dụng HelloWorld:

A screenshot of a computer program

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

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

2.4 Khám phá ứng dụng và thư mục res.

Tất cả mã 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 sẽ mở tệp đó trong trình chỉnh sửa mã.

A screenshot of a computer

AI-generated content may be incorrect.

Thư mục java bao gồm các tệp lớp 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 một gói ứng dụng. Hai thư mục còn lại được sử dụng để thử nghiệm và được mô tả trong bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa MainActivity.java. Tên của Hoạt động đầu tiên (màn hình) mà người dùng nhìn thấy, cũng khởi tạo các tài nguyên trên toàn ứng dụng, thường được gọi là MainActivity (phần mở rộng tệp bị bỏ qua trong ngăn Dự án > Android).

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

A screenshot of a computer

AI-generated content may be incorrect.

Thư mục res chứa các resources, chẳng hạn như layouts, chuỗi ký tự và hình ảnh. Một Activity thường được liên kết với layouts của các chế độ xem UI được định nghĩa là tệp XML. Tệp này thường được đặt tên theo Activity 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, 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 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 Activity, phải được khai báo trong tệp XML này. Trong các bài học khác của khóa học, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để biết phần giới thiệu, hãy xem tổng quan về App Manifest .

**Nhiệm vụ 3: Sử dụng 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 (còn được gọi 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 Trình giả lập Android có các yêu cầu bổ sung ngoài các yêu cầu hệ thống cơ bản đối với Android Studio. 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 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.

3.1 Tạo 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 > Device Manager hoặc nhấp vào biểu tượng Device Manager trên thanh công cụ. Màn hình Device Manager 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.

A screenshot of a computer

AI-generated content may be incorrect.

2. Nhấp vào + add a new device. Cửa sổ Virtual Device Configuration 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 ).

A screenshot of a computer

AI-generated content may be incorrect.

3.2 Chạy ứng dụng trong thiết bị android ảo(AVD)

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

Trong Android Studio, chọn Run > Run app hoặc nhấp vào biểu tượng Run trên thanh công cụ.

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.

A white background with black text

AI-generated content may be incorrect.

**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 nên luôn kiểm tra ứng dụng của mình trên cả thiết bị ảo và 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 của bạn 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 phù hợp cho thiết bị của mình. Đối với trình điều khiển USB chạy 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 Gỡ lỗi USB trên thiết bị Android của mình. Tính năng này được bật trong cài đặt Tùy chọn nhà phát triển của thiết bị. Trên Android 4.2 trở lên, màn hình Tùy chọn nhà phát triển bị ẩn theo mặc định. Để hiển thị tùy chọn nhà phát triển và bật Gỡ lỗi USB:

1. Trên thiết bị của bạn, hãy mở Cài đặt, tìm kiếm Giới thiệu về điện thoại, nhấp vào Giới thiệu về điện thoại và chạm vào Số bản dựng bảy lần.

2. Quay lại màn hình trước đó (Cài đặt/Hệ thống). Tùy chọn nhà phát triển xuất hiện trong danh sách. Chạm vào Tùy chọn nhà phát triển.

3. Chọn Gỡ lỗi USB.

4.2 Chạy ứng dụng của bạn trên thiết bị

Bây giờ 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 bằng cáp USB.

2. Nhấp vào nút Chạy trên thanh công cụ. Cửa sổ Chọn mục tiêu triển khai mở ra với danh sách các trình giả lập và thiết bị được kết nối khả dụng.

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 ra thiết bị của bạn, hãy thử các bước sau:

1. Rút phích 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 "không được phép", hãy làm theo các bước sau:

1. Rút phích cắm thiết bị.

2. Trên thiết bị, hãy 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, hãy cấp ủy quyền.

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

Trong nhiệm vụ này, bạn sẽ thay đổi một số thông tin về cấu hình ứng dụng trong tệp build.gradle(Module:app) để tìm hiểu cách thực hiện 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

Thực hiện theo các bước sau:

1. Mở rộng thư mục Gradle Scripts nếu thư mục này chưa mở và nhấp đúp vào tệp build.gradle(Module:app). Nội dung của tệp sẽ xuất hiện trong trình soạn thảo mã.

2. Trong khối defaultConfig, hãy thay đổi giá trị của minSdkVersion thành 17 như hiển thị bên dưới (ban đầu giá trị này được đặt thành 15).

A screen shot of a computer

AI-generated content may be incorrect.

5.2 Đồng bộ cấu hình Gradle mới

Khi bạn thực hiện thay đổi đối với các tệp cấu hình bản dựng trong một dự án, Android Studio yêu cầu bạn đồng bộ 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 sẽ không tạo ra lỗi bản dựng.

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

Khi quá trình đồng bộ hóa 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 xem tài liệu Tổng quan về hệ thống bản dựng và Cấu hình bản dựng Gradle.

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

Ở nhiệm vụ này bạn sẽ thêm log statements vào ứng dụng của bạn, nó hiển thị thông báo trong ngăn Logcat. Thông báo log là một công cụ gỡ lỗi mạnh mẽ mà bạn có thể sử dụng để kiểm tra giá trị, đường dẫn thực thi và báo cáo ngoại lệ.

6.1 Xem ngăn Logcat.

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

A screenshot of a computer program

AI-generated content may be incorrect.

**Coding challenge**

Bây giờ bạn đã thiết lập 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 tên của một người nào đó có ngày sinh nhật gần đây.

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

4. Một cách sử dụng phổ biến của lớp Log là ghi lại 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 phương pháp bạn có thể sử dụng bao gồm ngoại lệ với thông báo Nhật ký. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại ngoại lệ.

**Tóm tắt**

● Để cài đặt Android Studio, hãy vào Android Studio và làm theo hướng dẫn để tải xuống và cài đặt.

● Khi tạo ứng dụng mới, hãy đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu.

● Để xem 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 hoặc thay đổi phiên bản thư viện.

● Tất cả mã và tài nguyên cho ứng dụng đều nằm trong thư mục app và res. Thư mục java bao gồm các hoạt động, bài 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 thành phần tính năng 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 một thiết bị ảo (còn được gọi là trình giả lập) để chạy ứng dụng của bạn.

● Thêm log statement vào ứng dụng của bạn, hiển thị các 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 và chạm vào Build number bảy lần. Quay lại màn hình trước đó (Settings) và chạm vào Developer options. Chọn USB Debugging .  .

### Giao diện người dùng tương tác đầu tiên

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 views— mọi thành phần của màn hình là một View. Lớp View biểu thị khối xây dựng cơ bản cho tất cả các thành phần UI và là lớp cơ sở cho các lớp cung cấp các thành phần UI tương tác như nút, hộp kiểm và trường nhập văn bản. Các lớp con View thường được sử dụng được mô tả trong 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 và các thành phần có thể nhấp khác (như RadioButton, CheckBox và Spinner) để cung cấp hành vi tương tác.

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

● ImageView để hiển thị hình ảnh.

● ConstraintLayout và LinearLayout để chứa các thành phần View khác và định vị chúng.

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

Ví dụ: MainActivity trong ứng dụng Hello World hiển thị một bố cục được định nghĩa trong tệp bố cục activity\_main.xml, bao gồm một TextView có văn bản "Hello World".

Trong các ứng dụng phức tạp hơn, một Activity có thể triển khai các hành động để phản hồi các lần chạm của người dùng, vẽ nội dung đồ họa hoặc yêu cầu dữ liệu từ cơ sở dữ liệu hoặc internet. Bạn sẽ tìm hiểu thêm về lớp Activity trong một bài học khác.

Trong bài thực hành này, bạn sẽ học cách tạo ứng dụng tương tác đầu tiên của mình—một ứng dụng cho phép người dùng tương tác. Bạn tạo một ứng dụng bằng mẫu Empty Activity. Bạn cũng sẽ 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 có thể hoàn thành các bài thực hành khác trong khóa học này.

Những điều 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 ứng dụng có hành vi tương tác

● Cách sử dụng Layout Editor để thiết kế layout.

● Cách chỉnh sửa bố cục trong XML.

● Nhiều thuật ngữ mới.

Những gì bạn sẽ làm

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

● Thao tác từng phần tử trong ConstraintLayout để giới hạn chúng vào lề và các phần tử khác.

● Thay đổi các thuộc tính phần tử UI.

● Chỉnh sửa bố cục của ứng dụng trong XML.

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

● Triển khai các phương thức xử lý nhấp để hiển thị thông báo trên màn hình khi người dùng chạm vào từng Button

Tổng quan về ứng dụng

Ứng dụng HelloToast bao gồm hai phần tử Button và một TextView. Khi người dùng chạm vào Button đầu tiên, nó sẽ hiển thị một thông báo ngắn (một Toast) trên màn hình. Chạm vào Button thứ hai sẽ tăng bộ đếm "nhấp chuột" được hiển thị trong TextView, bắt đầu từ số không.

Sau đây là giao diện của ứng dụng đã hoàn thành:

A yellow rectangular sign with a black border

AI-generated content may be incorrect.

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

Trong bài thực hành này, bạn thiết kế và triển khai 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. Khởi động Android Studio và tạo một dự án mới với các tham số sau:

|  |  |
| --- | --- |
| Thuộc tính | Giá trị |
| Application Name | Hello Toast |
| Company Name | com.example.android |
| 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 Apps hoặc nhấp vào biểu tượng Run trong thanh công cụ để xây dựng và thực thi 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 chỉnh sửa bố cục để nhanh chóng xây dựng bố cục của các thành phần giao diện người dùng (UI) của ứng dụng. Nó cho phép bạn kéo các thành phần vào chế độ xem thiết kế trực quan và bản thiết kế, định vị chúng trong bố cục, thêm ràng buộc và đặt thuộc tính. Ràng buộc xác định vị trí của thành phần UI trong bố cục. Ràng buộc thể hiện kết nối hoặc căn chỉnh với chế độ xem khác, bố cục cha hoặc hướng dẫn vô hình.

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

A screenshot of a computer

AI-generated content may be incorrect.

1. Trong thư mục app > res > layout trong ngăn Project > Android, hãy nhấp đúp vào tệp activity\_main.xml để mở tệp đó nếu tệp đó chưa được mở.

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

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

4. Ngăn Component tree hiển thị hệ thống phân cấp chế độ xem của các thành phần UI. Các thành phần View được sắp xếp thành hệ thống phân cấp cây gồm các thành phần cha và con, trong đó thành phần con kế thừa các thuộc tính của thành phần cha. Trong hình trên, TextView là thành phần con của ConstraintLayout. Bạn sẽ tìm hiểu về các thành phần này sau trong bài học này.

5. Ngăn design và blueprint của trình chỉnh sửa bố cục hiển thị các thành phần UI trong bố cục. Trong hình trên, bố cục chỉ hiển thị một thành phần: TextView hiển thị "Hello World".

6. Tab Attributes hiển thị ngăn Attributes để thiết lập các thuộc tính cho thành phần UI.

Nhiệm vụ 2: Thêm các thành phần View vào trình chỉnh sửa bố cục

Trong nhiệm vụ này, bạn tạo bố cục UI cho ứng dụng HelloToast trong trình chỉnh sửa bố cục bằng các tính năng ConstraintLayout. Bạn có thể tạo các ràng buộc theo cách thủ công, như được hiển thị sau, hoặc tự động bằng công cụ Autoconnect.

2.1 Kiểm tra các ràng buộc của phần tử

Thực hiện theo các bước sau: 1. Mở activity\_main.xml từ ngăn Project > Android nếu nó chưa mở. Nếu tab Design chưa được chọn, hãy nhấp vào tab đó.

Nếu không có bản thiết kế, hãy nhấp vào nút Select Design Surface trên thanh công cụ và chọn Design + Blueprint.

2. Công cụ Autoconnect cũng nằm trên thanh công cụ. Công cụ này được bật theo mặc định. Đối với bước này, hãy đảm bảo rằng công cụ không bị tắt.

3. Nhấp vào nút zoom in a close-up look. để phóng to các ngăn thiết kế và bản thiết kế cho

4. Chọn TextView trong ngăn Component Tree. TextView "Hello World" được tô sáng trong các ngăn thiết kế và bản thiết kế và các ràng buộc cho phần tử sẽ hiển thị.

5. Tham khảo hình ảnh động bên dưới để biết bước này. Nhấp vào tay cầm tròn ở bên phải của TextView để xóa ràng buộc theo chiều ngang liên kết chế độ xem với bên phải của bố cục. TextView nhảy sang bên trái vì nó không còn bị ràng buộc ở bên phải nữa. Để thêm lại ràng buộc theo chiều ngang, hãy nhấp vào cùng một tay cầm và kéo một đường sang phía bên phải của bố cục.

A blue and white background with text

AI-generated content may be incorrect.

Trong bản thiết kế hoặc ngăn thiết kế, các handle sau xuất hiện trên phần tử TextView:

● Constraint handle: Để tạo một constraint như được hiển thị trong hình động ở trên, hãy nhấp vào một constraint handle, được hiển thị dưới dạng một vòng tròn ở bên cạnh một phần tử. Sau đó, kéo handle đến một constraint handle khác hoặc đến ranh giới cha. Một đường ngoằn ngoèo biểu thị cho constraints.

A blue square with a white circle and a spiral

AI-generated content may be incorrect.

● Resizing handle: Để thay đổi kích thước phần tử, hãy kéo các handle thay đổi kích thước hình vuông. Handle sẽ thay đổi thành một góc nghiêng khi bạn kéo nó.

A blue square with a white square

AI-generated content may be incorrect.

2.2 Thêm Nút vào bố cục

Khi được bật, công cụ Tự động kết nối sẽ tự động tạo hai hoặc nhiều ràng buộc cho một phần tử UI vào bố cục cha. Sau khi bạn kéo phần tử vào bố cục, công cụ này sẽ tạo các ràng buộc dựa trên vị trí của phần tử.

Thực hiện theo các bước sau để thêm nút:

1. Bắt đầu với một bảng trắng. Phần tử TextView không cần thiết, vì vậy khi phần tử này vẫn được chọn, hãy nhấn phím Delete hoặc chọn Edit > Delete. Bây giờ bạn có một bố cục hoàn toàn trống.

2. Kéo Nút từ ngăn Bảng màu đến bất kỳ vị trí nào trong bố cục. Nếu bạn thả Nút vào vùng giữa trên cùng của bố cục, các ràng buộc có thể tự động xuất hiện. Nếu không, bạn có thể kéo các ràng buộc lên trên cùng, bên trái và bên phải của bố cục như minh họa trong hình động bên dưới.

A screenshot of a computer

AI-generated content may be incorrect.

2.3 Thêm Nút thứ hai vào bố cục

1. Kéo Nút khác từ ngăn Bảng màu vào giữa bố cục như minh họa trong hình động bên dưới. Autoconnect có thể cung cấp các ràng buộc theo chiều ngang cho bạn (nếu không, bạn có thể tự kéo chúng).

2. Kéo một ràng buộc theo chiều dọc xuống dưới cùng của bố cục (tham khảo hình bên dưới).

A purple and white button

AI-generated content may be incorrect.

Bạn có thể xóa ràng buộc khỏi một phần tử bằng cách chọn phần tử đó và di con trỏ qua nó để hiển thị nút Xóa ràng buộc.

Nhấp vào nút này để xóa tất cả các ràng buộc trên phần tử đã chọn. Để xóa một ràng buộc duy nhất, hãy nhấp vào tay cầm cụ thể đặt ràng buộc đó.

Để xóa tất cả các ràng buộc trong toàn bộ bố cục, hãy nhấp vào công cụ Clear All Constraints trên thanh công cụ. Công cụ này hữu ích nếu bạn muốn làm lại tất cả các ràng buộc trong bố cục của mình.

Nhiệm vụ 3: Thay đổi thuộc tính phần tử UI

Ngăn Thuộc tính cung cấp quyền truy cập vào tất cả các thuộc tính XML mà bạn có thể gán cho một phần tử UI. Bạn có thể tìm thấy các thuộc tính (được gọi là thuộc tính) chung cho tất cả các chế độ xem trong View class documentation .

Trong nhiệm vụ này, bạn nhập các giá trị mới và thay đổi các giá trị cho các thuộc tính Button quan trọng, áp dụng cho hầu hết các loại View.

3.1 Thay đổi kích thước nút

Trình chỉnh sửa bố cục cung cấp các nút điều khiển thay đổi kích thước ở cả bốn góc của Chế độ xem để bạn có thể thay đổi kích thước Chế độ xem một cách nhanh chóng. Bạn có thể kéo các nút điều khiển ở mỗi góc của Chế độ xem để thay đổi kích thước, nhưng làm như vậy sẽ mã hóa cứng các kích thước chiều rộng và chiều cao. Tránh mã hóa cứng các kích thước cho hầu hết các thành phần Chế độ xem, vì các kích thước được mã hóa cứng không thể thích ứng với các nội dung và kích thước màn hình khác nhau.

Thay vào đó, hãy sử dụng ngăn Thuộc tính ở bên phải của trình chỉnh sửa bố cục để chọn chế độ định cỡ không sử dụng các kích thước được mã hóa cứng. Ngăn Thuộc tính bao gồm một bảng điều khiển định cỡ hình vuông được gọi là trình kiểm tra chế độ xem ở trên cùng. Các ký hiệu bên trong hình vuông biểu thị các thiết lập chiều cao và chiều rộng như sau:

A screenshot of a computer

AI-generated content may be incorrect.

Trong hình trên:

1. Height control. Kiểm soát này chỉ định thuộc tính layout\_height và xuất hiện trong hai phân đoạn ở phía trên và phía dưới của hình vuông. Các góc cho biết rằng kiểm soát này được đặt thành wrap\_content, nghĩa là View sẽ mở rộng theo chiều dọc khi cần để vừa với nội dung của nó. "8" cho biết lề chuẩn được đặt thành 8dp.

2. Width control. Kiểm soát này chỉ định layout\_width và xuất hiện trong hai phân đoạn ở phía bên trái và bên phải của hình vuông. Các góc cho biết rằng kiểm soát này được đặt thành wrap\_content có nghĩa là View sẽ mở rộng theo chiều ngang khi cần để vừa với nội dung của nó, lên đến lề 8dp.

3. Nút đóng ngắn Attributes. Nhấp để đóng ngăn.

Thực hiện theo các bước sau:

1. Chọn Nút trên cùng trong ngăn Cây thành phần.

2. Nhấp vào tab Thuộc tính ở phía bên phải của cửa sổ trình chỉnh sửa bố cục.

A screenshot of a computer

AI-generated content may be incorrect.

3. Nhấp vào nút điều khiển chiều rộng hai lần—nhấp chuột đầu tiên sẽ thay đổi thành Cố định với các đường thẳng và nhấp chuột thứ hai sẽ thay đổi thành Phù hợp với các ràng buộc có cuộn lò xo, như thể hiện trong hình động bên dưới.

A screenshot of a screen

AI-generated content may be incorrect.

Kết quả của việc thay đổi điều khiển chiều rộng, thuộc tính layout\_width trong ngăn Thuộc tính hiển thị giá trị match\_constraint và phần tử Button kéo dài theo chiều ngang để lấp đầy khoảng trống giữa bên trái và bên phải của bố cục.

4. Chọn Button thứ hai và thực hiện các thay đổi tương tự đối với layout\_width như trong bước trước, như thể hiện trong hình bên dưới.

A screenshot of a computer

AI-generated content may be incorrect.

Như đã trình bày trong các bước trước, các thuộc tính layout\_width và layout\_height trong ngăn Thuộc tính thay đổi khi bạn thay đổi các điều khiển chiều cao và chiều rộng trong trình kiểm tra. Các thuộc tính này có thể lấy một trong ba giá trị cho bố cục, đó là ConstraintLayout:

● Thiết lập match\_constraint mở rộng phần tử View để lấp đầy phần tử cha của nó theo chiều rộng hoặc chiều cao—lên đến một lề, nếu có. Phần tử cha trong trường hợp này là ConstraintLayout. Bạn sẽ tìm hiểu thêm về ConstraintLayout trong tác vụ tiếp theo.

● Thiết lập wrap\_content thu nhỏ kích thước của phần tử View để nó chỉ đủ lớn để bao quanh nội dung của nó. Nếu không có nội dung, phần tử View sẽ trở nên vô hình. ● Để chỉ định kích thước cố định điều chỉnh theo kích thước màn hình của thiết bị, hãy sử dụng số lượng pixel không phụ thuộc vào density-independent pixels (đơn vị dp). Ví dụ: 16dp nghĩa là 16 pixel không phụ thuộc vào pixels.

3.2 Thay đổi các thuộc tính của Button

Để xác định duy nhất từng View trong một bố cục Activity, mỗi View hoặc lớp con View (chẳng hạn như Button) cần một ID duy nhất. Và để có thể sử dụng, các phần tử Button cần có văn bản. Các phần tử View cũng có thể có nền có thể là màu hoặc hình ảnh.

Ngăn Thuộc tính cung cấp quyền truy cập vào tất cả các thuộc tính mà bạn có thể gán cho một phần tử View. Bạn có thể nhập giá trị cho từng thuộc tính, chẳng hạn như các thuộc tính android:id, background, textColor và text .

Hình ảnh động sau đây minh họa cách thực hiện các bước này:

1. Sau khi chọn Button đầu tiên, hãy chỉnh sửa trường ID ở đầu ngăn Thuộc tính thành button\_toast cho thuộc tính android:id, được sử dụng để xác định phần tử trong bố cục.

2. Đặt thuộc tính background thành @color/colorPrimary. (Khi bạn nhập @c, các lựa chọn sẽ xuất hiện để dễ dàng lựa chọn.)

3. Đặt thuộc tính textColor thành @android:color/white.

4. Chỉnh sửa thuộc tính text thành Toast.

A screenshot of a computer

AI-generated content may be incorrect.

5. Thực hiện các thay đổi thuộc tính tương tự cho Button thứ hai, sử dụng button\_count làm ID, Count cho thuộc tính text và cùng màu cho nền và text như các bước trước.

colorPrimary là màu chính của theme, một trong những màu cơ sở theme được xác định trước trong tệp tài nguyên colors.xml. Nó được sử dụng cho thanh ứng dụng. Sử dụng màu cơ sở cho các thành phần UI khác sẽ tạo ra một UI thống nhất. Bạn sẽ tìm hiểu thêm về chủ đề ứng dụng và Material Design trong bài học khác.

4.1 Thêm TextView và ràng buộc

1. Như được hiển thị trong hình động bên dưới, kéo TextView từ ngăn Palette đến phần trên của bố cục và kéo ràng buộc từ trên cùng của TextView đến tay cầm ở dưới cùng của Toast Button. Thao tác này ràng buộc TextView nằm bên dưới Button.

A screenshot of a computer

AI-generated content may be incorrect.

2. Như được hiển thị trong hình động bên dưới, hãy kéo một ràng buộc từ dưới cùng của TextView đến tay cầm ở trên cùng của Count Button và từ các cạnh của TextView đến các cạnh của layout. Điều này ràng buộc TextView ở giữa layout giữa hai phần tử Button.

A screenshot of a computer

AI-generated content may be incorrect.

4.2 Thiết lập các thuộc tính của TextView

Với TextView được chọn, hãy mở ngăn Thuộc tính, nếu ngăn này chưa mở. Thiết lập các thuộc tính cho TextView như được hiển thị trong hình động bên dưới. Các thuộc tính bạn chưa gặp phải sẽ được giải thích sau hình:

1. Đặt ID thành show\_count.

2. Đặt văn bản thành 0.

3. Đặt textSize thành 160sp.

4. Đặt textStyle thành B (in đậm) và textAlignment thành ALIGNCENTER (căn giữa đoạn văn).

5. Thay đổi các điều khiển kích thước chế độ xem theo chiều ngang và chiều dọc ( layout\_width và layout\_height ) thành match\_constraint.

6. Đặt textColor thành @color/colorPrimary.

7. Cuộn xuống ngăn và nhấp vào Xem tất cả thuộc tính, cuộn xuống trang thuộc tính thứ hai đến nền, sau đó nhập #FFF00 để có màu vàng.

8. Cuộn xuống đến trọng lực, mở rộng trọng lực và chọn center\_ver (cho tâm theo chiều dọc).

A screenshot of a computer program

AI-generated content may be incorrect.

● textSize: Kích thước văn bản của TextView. Đối với bài học này, kích thước được đặt thành 160sp. Sp là viết tắt của pixel không phụ thuộc tỷ lệ và giống như dp, là đơn vị tỷ lệ với mật độ màn hình và sở thích về kích thước phông chữ của người dùng. Sử dụng đơn vị dp khi bạn chỉ định kích thước phông chữ để kích thước được điều chỉnh cho cả mật độ màn hình và sở thích của người dùng.

● textStyle và textAlignment: Kiểu văn bản, được đặt thành B (in đậm) trong bài học này và căn chỉnh văn bản, được đặt thành ALIGNCENTER (căn giữa đoạn văn).

●gravity: Thuộc tính gravity chỉ định cách một View được căn chỉnh trong View hoặc ViewGroup cha của nó. Trong bước này, bạn căn giữa TextView để căn giữa theo chiều dọc trong ConstraintLayout cha.

Bạn có thể nhận thấy rằng thuộc tính background nằm trên trang đầu tiên của ngăn Attributes đối với Button, nhưng lại nằm trên trang thứ hai của ngăn Attributes đối với TextView. Ngăn Attributes thay đổi đối với từng loại View: Các thuộc tính phổ biến nhất cho loại View sẽ xuất hiện trên trang đầu tiên và các thuộc tính còn lại được liệt kê trên trang thứ hai.

Để quay lại trang đầu tiên của ngăn Attributes, hãy nhấp vào biểu tượng trên thanh công cụ ở đầu ngăn.

Nhiệm vụ 5: Chỉnh sửa bố cục trong XML

Bố cục ứng dụng Hello Toast gần hoàn thành! Tuy nhiên, một dấu chấm than xuất hiện bên cạnh mỗi phần tử UI trong Cây thành phần. Di con trỏ qua các dấu chấm than này để xem các thông báo cảnh báo, như hiển thị bên dưới. Cảnh báo tương tự xuất hiện cho cả ba phần tử: chuỗi được mã hóa cứng phải sử dụng tài nguyên.

A screenshot of a computer

AI-generated content may be incorrect.

Cách dễ nhất để khắc phục sự cố bố cục là chỉnh sửa bố cục trong XML. Mặc dù trình chỉnh sửa bố cục là một công cụ mạnh mẽ, nhưng một số thay đổi dễ thực hiện trực tiếp trong mã nguồn XML.

5.1 Mở mã XML cho bố cục. Đối với tác vụ này, hãy mở tệp activity\_main.xml nếu tệp này chưa được mở và nhấp vào tab Văn bản ở cuối trình chỉnh sửa bố cục.

Trình soạn thảo XML xuất hiện, thay thế các ngăn thiết kế và bản thiết kế. Như bạn có thể thấy trong hình bên dưới, hiển thị một phần mã XML cho bố cục, các cảnh báo được tô sáng—các chuỗi được mã hóa cứng "Toast" và "Count". (Chuỗi "0" được mã hóa cứng cũng được tô sáng nhưng không hiển thị trong hình.) Di con trỏ qua chuỗi được mã hóa cứng "Toast" để xem thông báo cảnh báo.

A screen shot of a computer program

AI-generated content may be incorrect.

5.2 Trích xuất tài nguyên chuỗi

Thay vì mã hóa cứng các chuỗi, cách tốt nhất là sử dụng tài nguyên chuỗi, biểu diễn các chuỗi. Việc có các chuỗi trong một tệp riêng giúp quản lý chúng dễ dàng hơn, đặc biệt là nếu bạn sử dụng các chuỗi này nhiều lần. Ngoài ra, tài nguyên chuỗi là bắt buộc để dịch và bản địa hóa ứng dụng của bạn, vì bạn cần tạo tệp tài nguyên chuỗi cho từng ngôn ngữ.

1. Nhấp một lần vào từ "Toast" (cảnh báo được tô sáng đầu tiên).

2. Nhấn Alt-Enter trong Windows hoặc Option-Enter trong macOS và chọn Trích xuất tài nguyên chuỗi từ menu bật lên.

3. Nhập button\_label\_toast cho Tên tài nguyên.

4. Nhấp vào OK. Một tài nguyên chuỗi được tạo trong tệp values/res/string.xml và chuỗi trong mã của bạn được thay thế bằng tham chiếu đến tài nguyên: @string/button\_label\_toast

5. Trích xuất các chuỗi còn lại: button\_label\_count cho "Count" và count\_initial\_value cho "0" .

6. Trong ngăn Project > Android, hãy mở rộng các giá trị trong res, sau đó nhấp đúp vào strings.xml để xem tài nguyên chuỗi của bạn trong tệp strings.xml:

A screen shot of a computer

AI-generated content may be incorrect.

7. Bạn cần một chuỗi khác để sử dụng trong tác vụ tiếp theo hiển thị thông báo. Thêm vào tệp strings.xml một tài nguyên chuỗi khác có tên toast\_message cho cụm từ "Hello Toast!":

Nhiệm vụ 6: Thêm trình xử lý onClick cho các nút

Trong nhiệm vụ này, bạn thêm một phương thức Java cho mỗi Button trong MainActivity để thực thi khi người dùng chạm vào Button.

6.1 Thêm thuộc tính onClick và trình xử lý vào mỗi Button

Trình xử lý nhấp là phương thức được gọi khi người dùng nhấp hoặc chạm vào phần tử UI có thể nhấp. Trong Android Studio, bạn có thể chỉ định tên của phương thức trong trường onClick trong ngăn Thuộc tính của tab Thiết kế. Bạn cũng có thể chỉ định tên của phương thức xử lý trong trình soạn thảo XML bằng cách thêm thuộc tính android:onClick vào Button. Bạn sẽ sử dụng phương thức sau vì bạn chưa tạo các phương thức xử lý và trình soạn thảo XML cung cấp một cách tự động để tạo các phương thức đó.

1. Với trình soạn thảo XML mở (tab Văn bản), tìm Nút có android:id được đặt thành button\_toast

A screenshot of a computer program

AI-generated content may be incorrect.

2. Thêm thuộc tính android:onClick vào cuối phần tử button\_toast sau thuộc tính cuối cùng và trước chỉ báo kết thúc />:



3. Nhấp vào biểu tượng bóng đèn màu đỏ xuất hiện bên cạnh thuộc tính. Chọn Create click handler, chọn MainActivity và nhấp vào OK. Nếu biểu tượng bóng đèn màu đỏ không xuất hiện, hãy nhấp vào tên phương thức ("showToast"). Nhấn Alt-Enter (Option-Enter trên máy Mac), chọn Create 'showToast(view)' trong MainActivity và nhấp vào OK. Hành động này tạo một stub phương thức giữ chỗ cho phương thức showToast() trong MainActivity, như được hiển thị ở cuối các bước này.

4. Lặp lại hai bước cuối cùng với button\_count Button: Thêm thuộc tính android:onClick vào cuối và thêm trình xử lý nhấp:



Mã XML cho các phần tử UI trong ConstraintLayout bây giờ trông như thế này:

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

5. Nếu MainActivity.java chưa được mở, hãy mở rộng java trong chế độ xem Project > Android, mở rộng com.example.android.hellotoast, rồi nhấp đúp vào MainActivity. Trình chỉnh sửa mã xuất hiện với mã trong MainActivity

A screenshot of a computer program

AI-generated content may be incorrect.

6.2 Chỉnh sửa trình xử lý Nút Toast

Bây giờ bạn sẽ chỉnh sửa phương thức showToast()—trình xử lý nhấp chuột Nút Toast trong MainActivity —để nó hiển thị một thông báo. Toast cung cấp một cách để hiển thị một thông báo đơn giản trong một cửa sổ bật lên nhỏ. Nó chỉ lấp đầy lượng không gian cần thiết cho thông báo. Hoạt động hiện tại vẫn hiển thị và tương tác. Toast có thể hữu ích để kiểm tra tính tương tác trong ứng dụng của bạn—thêm một thông báo Toast để hiển thị kết quả của việc chạm vào Nút hoặc thực hiện một hành động.

Thực hiện theo các bước sau để chỉnh sửa trình xử lý nhấp chuột Nút Toast:

1. Xác định vị trí phương thức showToast() mới được tạo.

2. Để tạo một phiên bản của Toast, hãy gọi phương thức makeText() trên lớp Toast.

Câu lệnh này chưa hoàn tất cho đến khi bạn hoàn tất tất cả các bước.

3. Cung cấp ngữ cảnh của ứng dụng Activity. Vì Toast hiển thị ở trên cùng của Activity UI, hệ thống cần thông tin về Activity hiện tại. Khi bạn đã ở trong ngữ cảnh của Activity có ngữ cảnh bạn cần, hãy sử dụng điều này như một phím tắt.

4. Cung cấp thông báo để hiển thị, chẳng hạn như tài nguyên chuỗi (taste\_message bạn đã tạo ở bước trước). Tài nguyên chuỗi toast\_message được xác định bởi R.string.

5. Cung cấp thời lượng hiển thị. Ví dụ: Toast.LENGTH\_SHORT hiển thị toast trong thời gian tương đối ngắn.

6. Hiển thị Toast bằng cách gọi show() . Sau đây là toàn bộ phương thức showToast():

A screen shot of a computer

AI-generated content may be incorrect.

Chạy ứng dụng và xác minh rằng thông báo Toast xuất hiện khi nhấn vào nút Toast.

A yellow rectangular sign with a black number 0

AI-generated content may be incorrect.

6.3 Chỉnh sửa trình xử lý Nút đếm

Bây giờ bạn sẽ chỉnh sửa phương thức countUp()—trình xử lý nhấp vào Nút đếm trong MainActivity —để nó hiển thị số đếm hiện tại sau khi Count được chạm vào. Mỗi lần chạm sẽ tăng số đếm lên một.

Mã cho trình xử lý phải:

● Theo dõi số đếm khi nó thay đổi.

● Gửi số đếm đã cập nhật đến TextView để hiển thị. Thực hiện theo các bước sau để chỉnh sửa trình xử lý nhấp vào Nút đếm: 1. Xác định vị trí phương thức countUp() mới được tạo.

1. Xác định vị trí phương thức countUp() mới được tạo.

2. Để theo dõi số đếm, bạn cần một biến thành viên riêng. Mỗi lần nhấn nút Đếm sẽ làm tăng giá trị của biến này. Nhập nội dung sau, nội dung này sẽ được tô sáng màu đỏ và hiển thị biểu tượng bóng đèn màu đỏ:

A black screen with white text

AI-generated content may be incorrect.

3. Nhấp vào biểu tượng bóng đèn màu đỏ và chọn Tạo trường 'mCount' từ menu bật lên. Thao tác này tạo một biến thành viên riêng tư ở đầu MainActivity và Android Studio giả định rằng bạn muốn biến đó là số nguyên (int) và gán giá trị bằng 0:



4. Cùng với biến trên, bạn cũng cần một biến thành viên riêng tư để tham chiếu đến TextView show\_count, mà bạn sẽ thêm vào trình xử lý nhấp chuột. Gọi biến này là mShowCount

A screen shot of a computer program

AI-generated content may be incorrect.

5. Bây giờ bạn đã có mShowCount, bạn có thể lấy tham chiếu đến TextView bằng ID bạn đặt trong tệp bố cục. Để lấy tham chiếu này chỉ một lần, hãy chỉ định nó trong phương thức onCreate(). Như bạn đã học trong bài học khác, phương thức onCreate() được sử dụng để làm phồng bố cục, nghĩa là đặt chế độ xem nội dung của màn hình thành bố cục XML. Bạn cũng có thể sử dụng nó để lấy tham chiếu đến các thành phần UI khác trong bố cục, chẳng hạn như TextView. Xác định vị trí phương thức onCreate() trong MainActivity:

A screenshot of a computer screen

AI-generated content may be incorrect.

6. Thêm câu lệnh findViewById vào cuối phương thức:

A screen shot of a computer code

AI-generated content may be incorrect.

7. Bây giờ bạn đã gán TextView cho mShowCount, bạn có thể sử dụng biến này để đặt văn bản trong TextView thành giá trị của biến mCount. Thêm nội dung sau vào phương thức countUp():

A black background with white text

AI-generated content may be incorrect.

Toàn bộ phương thức countUp() hiện trông như thế này:

A computer screen shot of a code

AI-generated content may be incorrect.

8. Chạy ứng dụng để xác minh số đếm tăng lên khi bạn chạm vào nút Đếm.

A screenshot of a computer

AI-generated content may be incorrect.

### 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