

**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
Bài 1)	Tạo ứng dụng đầu tiên.....	4
1.1)	Android Studio và Hello World	4
1.2)	Giao diện người dùng tương tác đầu tiên.....	23
1.3)	Trình chỉnh sửa bố cục	39
1.4)	Văn bản và các chế độ cuộn.....	39
1.5)	Tài nguyên có sẵn	39
Bài 2)	Activities	39
2.1)	Activity và Intent	39
2.2)	Vòng đời của Activity và trạng thái	39
2.3)	Intent ngầm định	39
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ.....	39
3.1)	Trình gỡ lỗi.....	39
3.2)	Kiểm thử đơn vị	39
3.3)	Thư viện hỗ trợ.....	39
CHƯƠNG 2.	Trải nghiệm người dùng	40
Bài 1)	Tương tác người dùng.....	40
1.1)	Hình ảnh có thể chọn	40
1.2)	Các điều khiển nhập liệu.....	40
1.3)	Menu và bộ chọn.....	40
1.4)	Điều hướng người dùng	40
1.5)	RecyclerView	40
Bài 2)	Trải nghiệm người dùng thú vị	40
2.1)	Hình vẽ, định kiểu và chủ đề.....	40
2.2)	Thẻ và màu sắc.....	40
2.3)	Bố cục thích ứng	40
Bài 3)	Kiểm thử giao diện người dùng	40

3.1)	Espresso cho việc kiểm tra UI.....	40
CHƯƠNG 3. Làm việc trong nền		40
Bài 1)	Các tác vụ nền	40
1.1)	AsyncTask	40
1.2)	AsyncTask và AsyncTaskLoader	40
1.3)	Broadcast receivers	40
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền	40
2.1)	Thông báo.....	40
2.2)	Trình quản lý cảnh báo.....	40
2.3)	JobScheduler	40
CHƯƠNG 4. Lưu dữ liệu người dùng		40
Bài 1)	Tùy chọn và cài đặt	40
1.1)	Shared preferences	40
1.2)	Cài đặt ứng dụng	41
Bài 2)	Lưu trữ dữ liệu với Room	41
2.1)	Room, LiveData và ViewModel	41
2.2)	Room, LiveData và ViewModel	41

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

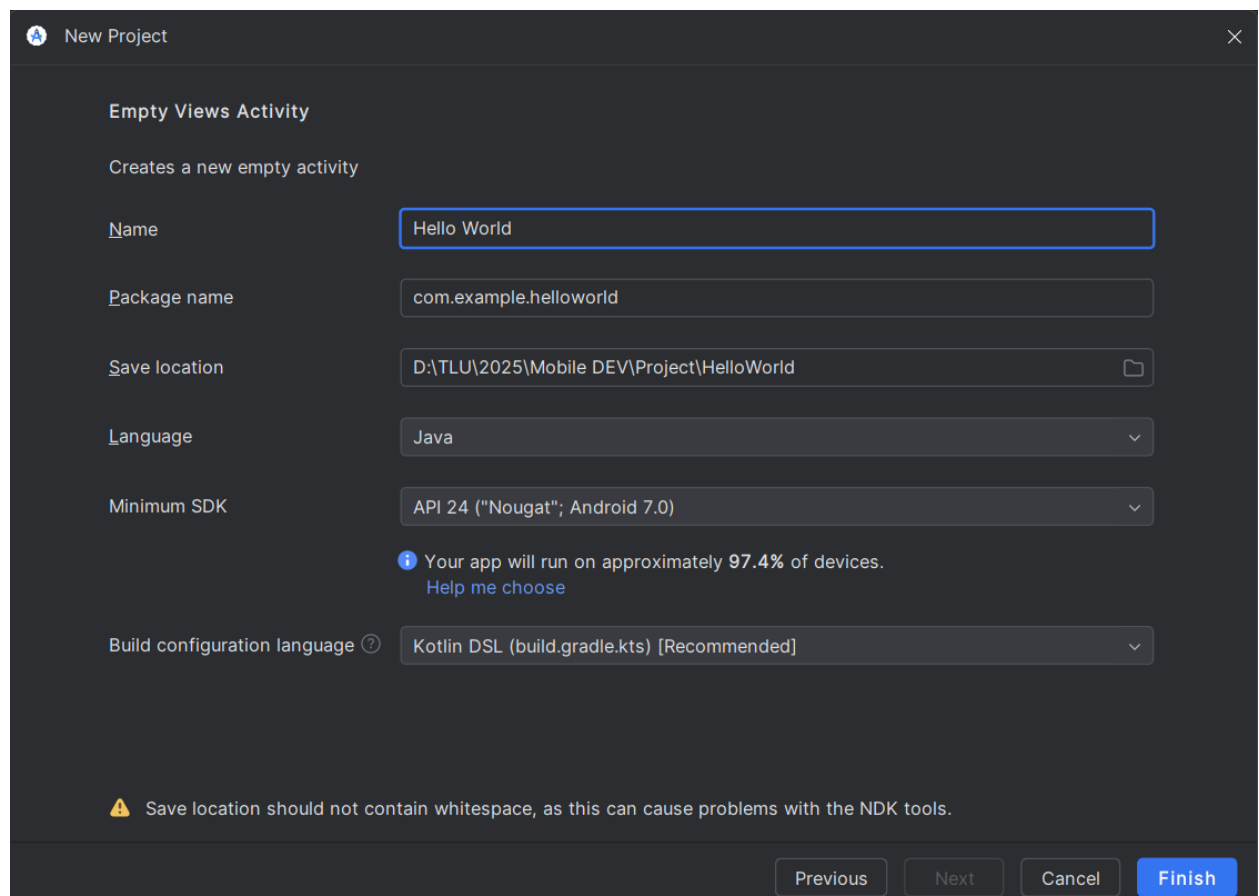
Giới thiệu

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

Những gì Bạn nên biết

Bạn nên có khả năng:

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



Những gì Bạn sẽ cần:

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

Những gì bạn sẽ học

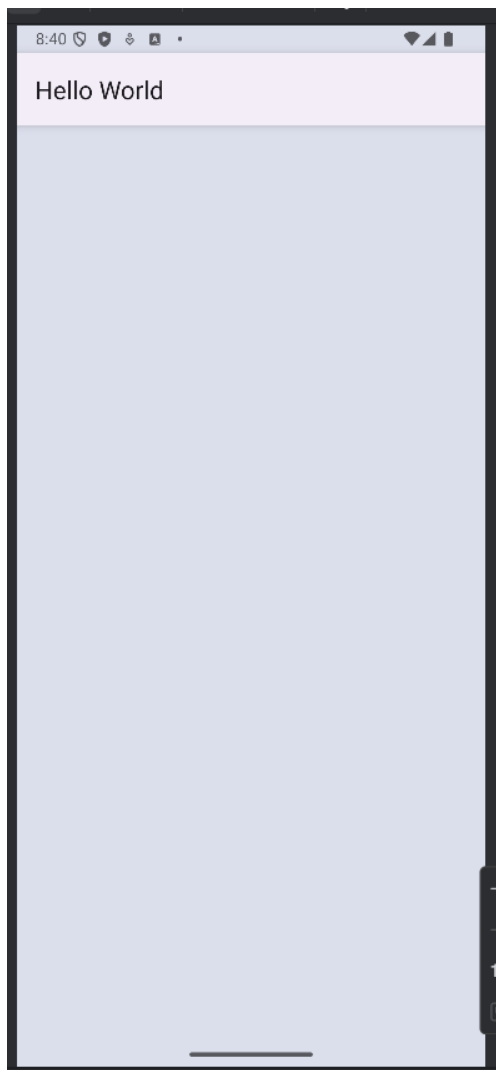
- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.
- Cách thêm thông điệp ghi lại vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

Những gì bạn sẽ làm

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

Tổng quan về ứng dụng

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



Nhiệm vụ 1: Cài đặt Android Studio

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

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

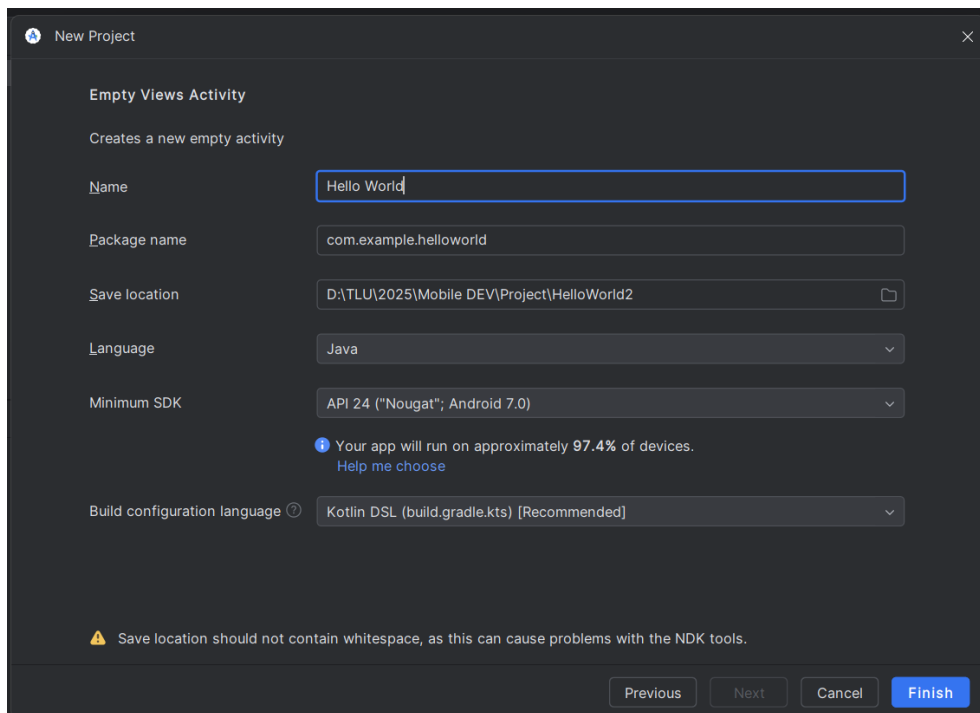
1. Truy cập trang web dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.
2. Chấp nhận cấu hình mặc định cho tất cả các bước và đảm bảo rằng tất cả các thành phần được chọn để cài đặt.
3. Sau khi hoàn tất cài đặt, Trình hướng dẫn cài đặt sẽ tải xuống và cài đặt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

Nhiệm vụ 2: Tạo ứng dụng Hello World

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

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu chưa mở.
2. Trong cửa sổ chính Welcome to Android Studio, nhấp vào Start a new Android Studio project .
3. Trong cửa sổ Create Android Project, nhập Hello World cho tên ứng dụng.



4. Xác minh rằng vị trí Dự án mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác hoặc thay đổi vị trí đó thành thư mục ưa thích của bạn.

5. Chấp nhận android.example.com mặc định cho Tên miền công ty hoặc tạo một miền công ty duy nhất. Nếu không có kế hoạch phát hành ứng dụng của mình, bạn có thể chấp nhận mặc định. Lưu ý rằng việc thay đổi tên gói của ứng dụng sau này là một công việc bổ sung.

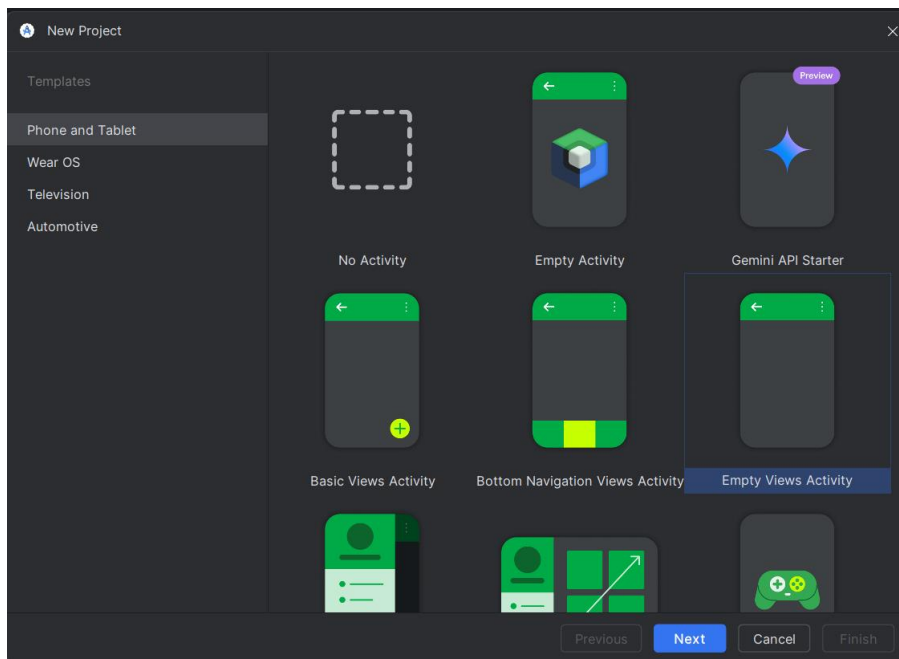
6. Bỏ chọn các tùy chọn Bao gồm hỗ trợ C++ và Bao gồm hỗ trợ Kotlin và nhấp vào Tiếp theo.

7. Trên màn hình Thiết bị Android mục tiêu, Điện thoại và Máy tính bảng sẽ được chọn. Đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu; Nếu không, hãy sử dụng menu bật lên để thiết lập.

Đây là các cài đặt được sử dụng bởi các ví dụ trong các bài học cho khóa học này. Khi viết bài này, các cài đặt này làm cho ứng dụng Hello World của bạn tương thích với 97% thiết bị Android đang hoạt động trên Cửa hàng Google Play.

8. Bỏ chọn Bao gồm hỗ trợ ứng dụng tức thì và tất cả các tùy chọn khác. Sau đó nhấp vào Tiếp theo . Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK mục tiêu bạn đã chọn, Android Studio sẽ tự động cài đặt các thành phần đó.

9. Cửa sổ Thêm hoạt động xuất hiện. Hoạt động là một việc tập trung duy nhất mà người dùng có thể làm. Nó là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Hoạt động thường có bố cục được liên kết với nó xác định cách các thành phần giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp các mẫu Hoạt động để giúp bạn bắt đầu. Đối với dự án Hello World, hãy chọn Hoạt động trông như hình dưới đây và nhấp vào Tiếp theo.



10. Mô hình Định cấu hình hoạt động màn hình xuất hiện (khác nhau tùy thuộc vào mẫu bạn đã chọn ở bước trước). Theo mặc định, Hoạt động trông do mẫu cung cấp có tên là MainActivity. Bạn có thể thay đổi điều này nếu muốn, nhưng bài học này sử dụng MainActivity.

11. Đảm bảo rằng tùy chọn Tạo tệp bố cục được chọn. Tên bố cục theo mặc định là activity_main. Bạn có thể thay đổi điều này nếu muốn, nhưng bài học này sử dụng activity_main.

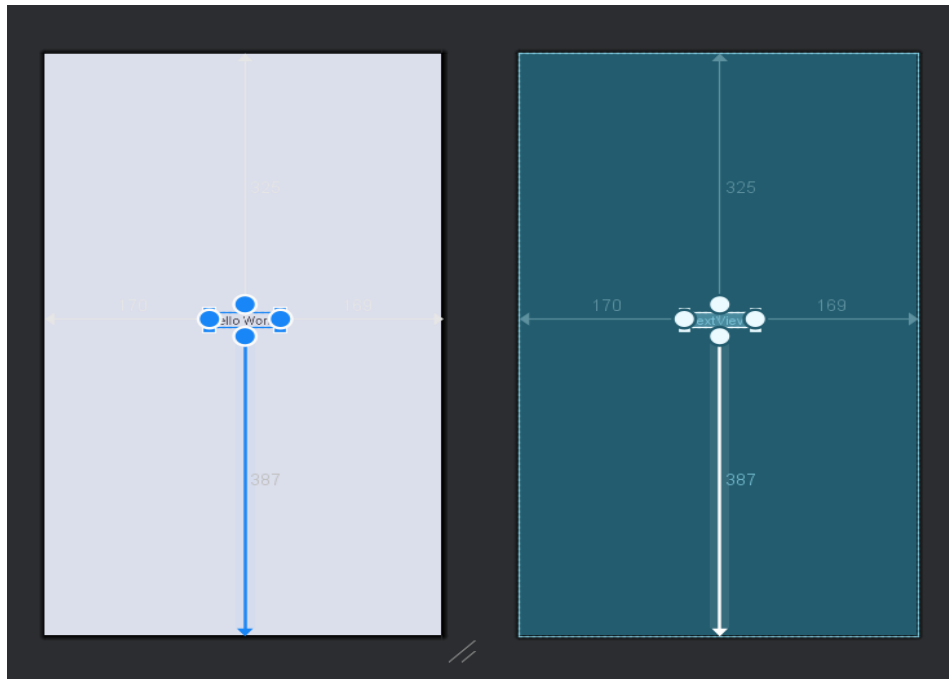
12. Đảm bảo rằng tùy chọn Tương thích ngược (Tương thích ứng dụng) được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước đó.

13. Nhấp vào Kết thúc

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

Mẹo: Xem trang Định cấu hình nhà phát triển bản dựng của bạn để biết thông tin chi tiết. Bạn cũng có thể thấy thông báo "Mẹo trong ngày" với các phím tắt và các mẹo hữu ích khác. Nhấp vào Đóng để đóng thư.

Trình chỉnh sửa Android Studio sẽ xuất hiện. Làm theo các bước sau: 1. Nhấp vào tab `activity_main.xml` để xem trình chỉnh sửa bố cục. 2. Nhấp vào tab Thiết kế trình chỉnh sửa bố cục, nếu chưa được chọn, để hiển thị biểu tượng đồ họa của bố cục như hình dưới đây.



3. Nhấp vào tab `MainActivity.java` để xem trình chỉnh sửa mã như hình bên dưới.

```
package com.example.helloworld;

import ...

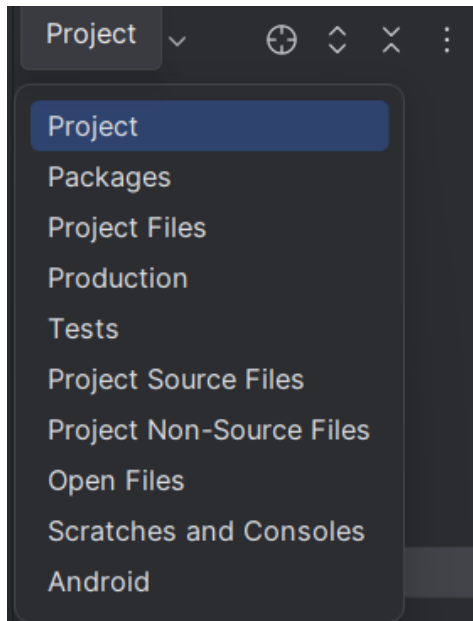
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}
```

2.2 Khám phá ngăn Project > Android

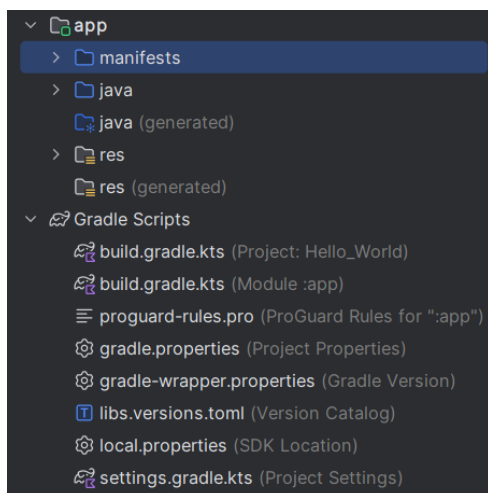
Trong thực tế này, bạn sẽ khám phá cách tổ chức dự án trong Android Studio.

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



2.3 Khám phá thư mục Tập lệnh Gradle

Hệ thống bản dựng Gradle trong Android Studio giúp bạn dễ dàng đưa các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng dưới dạng phần phụ thuộc. Khi bạn tạo dự án ứng dụng lần đầu tiên, ngăn Project > Android sẽ xuất hiện với thư mục Gradle Scripts được mở rộng như hình bên dưới.



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

1. Nếu thư mục Tập lệnh Gradle không được mở rộng, hãy nhấp vào hình tam giác để mở rộng. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng.
2. Tìm tệp build.gradle(Project: HelloWorld). Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô-đun tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một tệp bản dựng Gradle cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần thực hiện bất kỳ thay đổi nào đối với tệp này, nhưng vẫn hữu ích khi hiểu nội dung của nó. Theo mặc định, tệp bản dựng cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ Gradle và các phần phụ thuộc chung cho tất cả các mô-đun trong dự án. Khi phần phụ thuộc của bạn không phải là thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong bất kỳ kho lưu trữ trực tuyến nào được chỉ định trong khối kho lưu trữ của tệp này. Theo mặc định, các dự án Android Studio mới khai báo JCenter và Google (bao gồm kho lưu trữ Google Maven) là 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  
}
```

3. 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 định cấu hình cài đặt bản dựng cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Việc định cấu hình các tùy chọn 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 các tùy chọn 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 phần phụ thuộc trong phần phụ thuộc. Bạn có thể khai báo phần phụ thuộc thư viện bằng cách sử dụng một trong một số cấu hình phần phụ thuộc khác nhau. Mỗi cấu hình phần phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ: câu lệnh thực hiện fileTree(dir: 'libs', include: ['*.jar']) thêm phần phụ thuộc của tất cả các tệp ".jar" bên trong thư mục libs. Sau đây là tệp build.gradle(Module:app) cho ứng dụng HelloWorld:

```
plugins {  
    alias(libs.plugins.android.application)  
}  
  
android {
```

```

namespace = "com.example.helloworld"
compileSdk = 35

defaultConfig {
    applicationId = "com.example.helloworld"
    minSdk = 24
    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)
}

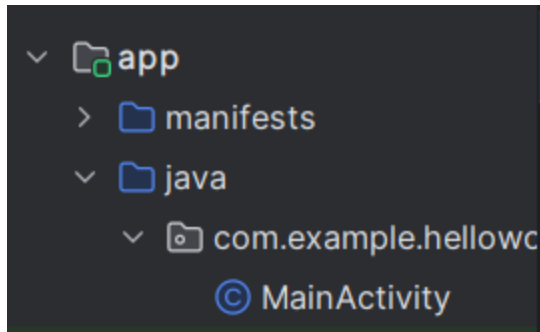
```

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 nằm trong thư mục ứng dụng và res.

1. Mở rộng thư mục ứng dụng, 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 soạn thảo mã.



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 gói ứng dụng. Hai thư mục còn lại được sử dụng để kiểm tra và được mô tả trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa MainActivity.java . Tên của Hoạt động (màn hình) đầu tiên mà người dùng nhìn thấy, cũng là khởi tạo 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 bố cục, đồng thời nhấp đúp vào tệp activity_main.xml để mở nó trong trình chỉnh sửa bố cục.



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

2.5 Khám phá thư mục kê khai

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

1. Mở rộng thư mục tệp kê khai.
2. Mở tệp AndroidManifest.xml.

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

Nhiệm vụ 3: Sử dụng thiết bị ảo (trình mô phỏng)

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

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

3.1 Tạo thiết bị ảo Android (AVD)

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

1. Trong Android Studio, chọn Công cụ > Trình quản lý AVD > Android hoặc nhấp vào biểu tượng Trình quản lý AVD trên thanh công cụ. Màn hình Thiết bị ảo của bạn xuất hiện. Nếu bạn đã tạo thiết bị ảo, màn hình sẽ hiển thị chúng (như trong hình bên dưới); nếu không, bạn sẽ thấy một danh sách trống.
2. Nhấp vào +Tạo thiết bị ảo . Cửa sổ Chọn phần cứng xuất hiện hiển thị danh sách các thiết bị phần cứng được định cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo (Kích thước), độ phân giải màn hình tính bằng pixel (Độ phân giải) và mật độ điểm ảnh (Mật độ).
3. Chọn một thiết bị như Nexus 5x hoặc Pixel XL và nhấp vào Tiếp theo . Màn hình Hình ảnh hệ thống xuất hiện.
4. Nhấp vào tab Đề xuất nếu nó chưa được chọn và chọn phiên bản hệ thống Android để chạy trên thiết bị ảo (chẳng hạn như Oreo).
1. Có nhiều phiên bản hơn được hiển thị trong tab Đề xuất. Nhìn vào tab Hình ảnh x86 và Hình ảnh khác để xem chúng. Nếu liên kết Tải xuống hiển thị bên cạnh hình ảnh hệ thống bạn muốn sử dụng, thì liên kết đó chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào Kết thúc khi hoàn tất.

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

3.2 Chạy ứng dụng trên thiết bị ảo

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

1. In Android Studio, choose Run > Run app or click the Run icon in the toolbar.
2. The Select Deployment Target window, under Available Virtual Devices , select the virtual device, which you just created, and click OK .

The emulator starts and boots just like a physical device. Depending on the speed of your computer, this may take a while. Your app builds, and once the emulator is ready, Android Studio will upload the app to the emulator and run it. You should see the Hello World app as shown in the following figure.

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

Nhiệm vụ 4: (Tùy chọn) Sử dụng thiết bị vật lý

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

Những gì bạn cần:

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

4.1 Bật tính năng gỡ lỗi USB

Để cho phép Android Studio giao tiếp với thiết bị của bạn, bạn phải bật tính năng gỡ lỗi USB trên thiết bị Android của mình. Tính năng này được bật trong cài đặt Tùy chọn nhà phát triển trên thiết bị của bạn. Trên Android 4.2 trở lên, màn hình Tùy chọn nhà phát triển bị ẩn theo mặc định. Để hiển thị các tùy chọn dành cho nhà phát triển và bật Gỡ lỗi USB:

1. Trên thiết bị của bạn, mở Cài đặt , tìm kiếm Giới thiệu về điện thoại , nhấp vào Giới thiệu về điện thoại và nhấn 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. Nhấn vào Tùy chọn nhà phát triển .
3. Chọn gỡ lỗi USB.

4.2 Chạy ứng dụng trên thiết bị

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

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

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

Khắc phục sự cố

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

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

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

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

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

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

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

5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng

Làm theo các bước sau:

1. Mở rộng thư mục Tập lệnh Gradle nếu thư mục chưa mở và nhấp đúp vào tệp `build.gradle(Module:app)`. Nội dung của tệp xuất hiện trong trình soạn thảo mã.
2. Trong khối `defaultConfig`, thay đổi giá trị của `minSdkVersion` thành 17 như hình bên dưới (ban đầu nó được đặt thành 15).

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

5.2 Đồng bộ hóa cấu hình Gradle mới

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

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

Khi quá trình đồng bộ hóa Gradle kết thúc, thông báo Bản dựng Gradle đã hoàn tất sẽ xuất hiện ở góc dưới cùng bên trái của cửa sổ Android Studio. Để tìm hiểu sâu hơn về Gradle, hãy xem tài liệu Tổng quan về hệ thống xây dựng và Định cấu hình bản dựng Gradle.

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

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

6.1 Xem ngăn Logcat

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

Trong hình trên:

1. Tab Logcat để mở và đóng ngăn Logcat, hiển thị thông tin về ứng dụng của bạn khi ứng dụng đang chạy. Nếu bạn thêm câu lệnh Nhật ký vào ứng dụng của mình, thông báo Nhật ký sẽ xuất hiện ở đây.
2. Menu Mức nhật ký được đặt thành Chi tiết (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 câu lệnh nhật ký vào ứng dụng

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

Các phần của tin nhắn là:

- Nhật ký: Lớp Nhật ký để gửi tin nhắn nhật ký đến ngăn Logcat.
- d: Cài đặt mức Nhật ký gỡ lỗi để lọc thông báo nhật ký hiển thị trong ngăn Logcat. Các cấp độ nhật ký khác là e cho Lỗi, w cho Cảnh báo và i cho Thông tin.
- "MainActivity": Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong ngăn Logcat. Đây thường là tên của Hoạt động mà thông điệp bắt đầu. Tuy nhiên, bạn có thể làm cho điều này bất cứ thứ gì hữu ích cho bạn để gỡ lỗi. Theo quy ước, thẻ nhật ký được định nghĩa là hằng số cho Activity
- "Xin chào thế giới": Đối số thứ hai là thông điệp thực tế. Làm theo các bước sau:
 1. Mở ứng dụng Hello World của bạn trong Android studio và mở MainActivity.
 2. Để tự động thêm các mục nhập rõ ràng vào dự án của bạn (chẳng hạn như android.util.Log cần thiết để sử dụng Nhật ký), hãy chọn Cài đặt > tệp trong Windows hoặc Tùy chọn Android Studio > trong macOS.
 3. Chọn Trình chỉnh sửa > Chung > Tự động nhập. Chọn tất cả các hộp kiểm và đặt Chèn nhập khi dán thành Tất cả.
 4. Nhấp vào Áp dụng và sau đó nhấp vào OK.
 5. Trong phương thức onCreate() của MainActivity, thêm câu lệnh sau:

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

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

Thông báo sau sẽ xuất hiện trong ngăn Logcat:

Thử thách mã hóa

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

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

1. Tạo một dự án mới trong Android Studio.
2. Thay đổi lời chào "Hello World" thành "Happy Birthday thành" và tên của một người có sinh nhật gần đây.
3. (Tùy chọn) Chụp ảnh màn hình ứng dụng đã hoàn thành của bạn và gửi email cho người mà bạn quên ngày sinh.
4. Một cách sử dụng phổ biến của lớp Log là ghi nhật ký các ngoại lệ Java khi chúng xảy ra trong chương trình của bạn. Có một số phương thức hữu ích, chẳng hạn như Log.e() , mà bạn có thể sử dụng cho mục đích này. Khám phá các phương thức bạn có thể sử dụng để bao gồm một ngoại lệ với thông báo Nhật ký. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại một ngoại lệ.

Tóm tắt

- Để cài đặt Android Studio, hãy truy cập Android Studio và làm theo hướng dẫn để tải xuống và cài đặt nó.
- Khi tạo ứng dụng mới, hãy đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu.
- Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy nhấp vào tab Dự án trong cột tab dọc, sau đó chọn Android trong menu bật lên ở trên cùng.
- Chỉnh sửa tệp build.gradle(Module:app) khi bạn cần thêm thư viện mới vào dự án của mình hoặc thay đổi phiên bản thư viện.

- Tất cả mã và tài nguyên cho ứng dụng đều nằm trong các thư mục ứng dụng và res. Thư mục java bao gồm các hoạt động, kiểm tra và các thành phần khác trong mã nguồn Java. Thư mục res chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh.

- Chỉnh sửa tệp AndroidManifest.xml để thêm các tính năng, thành phần và quyền vào ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML này.

- Sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo thiết bị ảo (còn được gọi là trình giả lập) để chạy ứng dụng của bạn.

- Thêm câu lệnh Nhật ký vào ứng dụng của bạn, hiển thị thông báo trong ngăn Logcat như một công cụ cơ bản để gỡ lỗi.

- Để chạy ứng dụng của bạn trên thiết bị Android vật lý bằng Android Studio, hãy bật Gỡ lỗi USB trên thiết bị. Mở 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 về khái niệm liên quan có trong 1.0: Giới thiệu về Android và 1.1 Ứng dụng Android đầu tiên của bạn .

Tìm hiểu thêm tài liệu về Android Studio:

- Trang tải xuống Android Studio
- Ghi chú phát hành Android Studio
- Gặp gỡ Android Studio
- Công cụ dòng lệnh Logcat
- Trình quản lý thiết bị ảo Android (AVD)
- Tổng quan về tệp kê khai ứng dụng
- Định cấu hình bản dựng của bạn
- Lớp nhật ký
- Tạo và quản lý thiết bị ảo

Khác:

- Làm cách nào để cài đặt Java?
- Cài đặt phần mềm JDK và cài đặt JAVA_HOME
- Trang web Gradle
- Cú pháp Apache Groovy
- Trang Wikipedia Gradle

Bài tập về nhà

Xây dựng và chạy ứng dụng

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

Trả lời các câu hỏi sau:

Câu hỏi 1: Tên của tệp bố cục cho hoạt động chính là gì?

- MainActivity.java
- AndroidManifest.xml
- activity_main.xml
- build.gradle

Câu hỏi 2: Tên của tài nguyên chuỗi chỉ định tên của ứng dụng là gì?

- app_name
- xmlns:app
- android:name
- applicationId

Câu hỏi 3 Bạn sử dụng công cụ nào để tạo trình giả lập mới?

- Giám sát thiết bị Android
- Trình quản lý AVD
- Trình quản lý SDK
- Trình chỉnh sửa chủ đề

Câu hỏi 4 Giả sử rằng ứng dụng của bạn bao gồm câu lệnh ghi nhật ký sau:

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

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

- Chi tiết
- Gỡ lỗi
- Thông tin
- Cảnh báo
- Lỗi
- Xác nhận Gửi ứng dụng của bạn để chấm điểm Kiểm tra để đảm bảo ứng dụng có những điều sau:
 - Hoạt động hiển thị "Hello World" trên màn hình.
 - Ghi lại các câu lệnh trong onCreate() trong hoạt động chính.
 - Mức nhật ký trong ngăn Logcat chỉ hiển thị các câu lệnh gỡ lỗi hoặc ghi nhật ký lỗi.

1.2) Giao diện người dùng tương tác đầu tiên

*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 phần tử của màn hình là một Chế độ xem . Lớp View đại diện cho khối xây dựng cơ bản cho tất cả các thành phần giao diện người dùng và lớp cơ sở cho các lớp cung cấp các thành phần giao diện người dùng 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.
- Nút và các yếu tố có thể nhấp khác (chẳng hạn 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 phần tử View khác và định vị chúng.

Mã Java hiển thị và điều khiển giao diện người dùng được chứa trong một lớp mở rộng từ Activity. Một Activity thường được liên kết với một bố cục giao diện người dùng được định nghĩa trong một tệp XML (eXtended Markup Language). Tệp XML này thường được đặt tên theo Activity của nó và xác định bố cục của các phần tử View trên màn hình.

Ví dụ, mã MainActivity trong ứng dụng "Hello World" hiển thị một bố cục được định nghĩa trong tệp bố cục activity_main.xml, trong đó có một TextView chứa văn bản "Hello World".

Trong các ứng dụng phức tạp hơn, một Activity có thể thực hiện các hành động như phản hồi thao tác 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 bài học khác.

Trong bài thực hành này, bạn sẽ học cách tạo ứng dụng tương tác đầu tiên—một ứng dụng cho phép người dùng tương tác. Bạn sẽ tạo một ứng dụng bằng cách sử dụng mẫu Empty Activity. Ngoài ra, bạn cũng sẽ học cách sử dụng trình chỉnh sửa bố cục để thiết kế bố cục và chỉnh sửa bố cục trong XML. Bạn cần phát triển những kỹ năng này để có thể hoàn thành các bài thực hành khác trong khóa học.

Những gì bạn nên biết Bạn nên làm quen:

- 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 với 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 bảng thuật ngữ từ vựng và khái niệm để biết các định nghĩa thân thiện.

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 hạn chế chúng ở lề và các phần tử khác.
- Thay đổi thuộc tính phần tử giao diện người dùng.
- Chỉnh sửa bố cục của ứng dụng trong XML.
- Trích xuất các chuỗi được mã hóa cứng vào tài nguyên chuỗi.
- Triển khai các phương pháp xử lý nhấp chuột để hiển thị thông báo trên màn hình khi người dùng nhấn vào từng nút

Tổng quan về ứng dụng Ứng dụng HelloToast bao gồm hai phần tử Button và một TextView. Khi người dùng nhấn vào Nút đầu tiên, nó sẽ hiển thị một thông báo ngắn (Toast) trên màn hình. Nhấn vào Nút thứ hai sẽ tăng bộ đếm "nhấp chuột" được hiển thị trong TextView, bắt đầu từ không. Đây là những gì ứng dụng đã hoàn thành trông như thế nào:



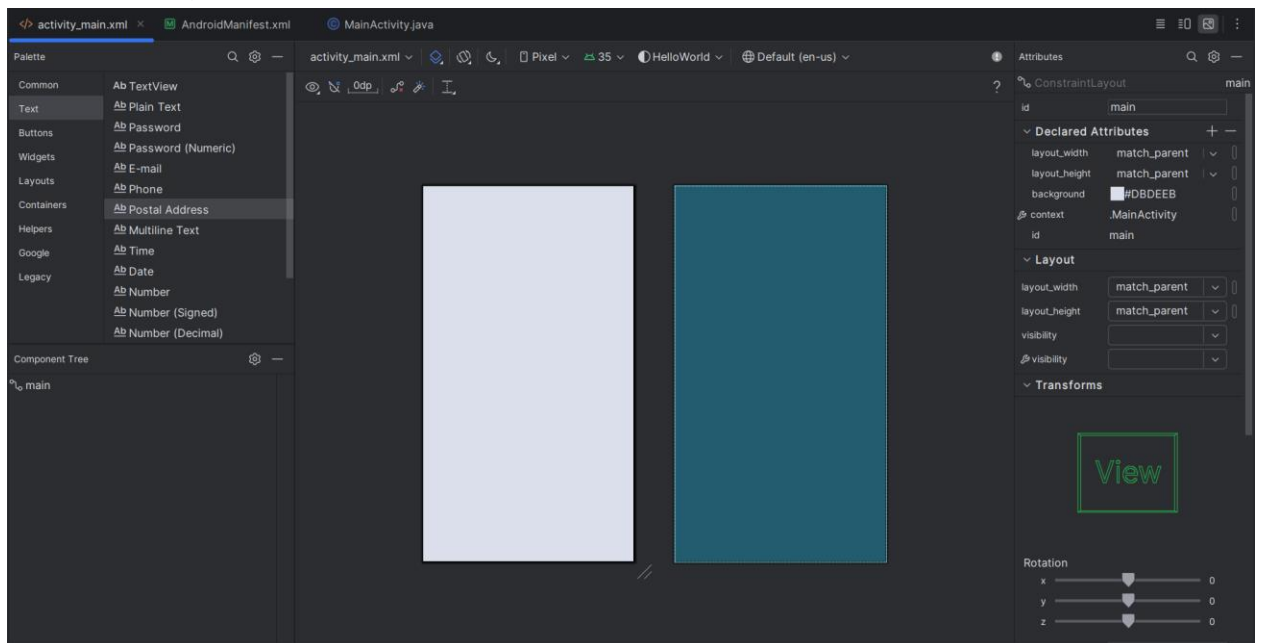
Nhiệm vụ 1: Tạo và khám phá một dự án mới Trong thực tế này, bạn thiết kế và triển khai một dự án cho ứng dụng HelloToast. Một liên kết đến mã giải pháp được cung cấp ở cuối.

1.1 Tạo dự án Android Studio

14. Khởi động Android Studio và tạo một dự án mới với các thông số sau:
15. Select Run > Run app or click the Run icon in the toolbar to build and execute the app on the emulator or your device.

1.2 Khám phá trình chỉnh sửa bố cục

Android Studio cung cấp trình chỉnh sửa bố cục để nhanh chóng tạo bố cục các phần tử giao diện người dùng (UI) của ứng dụng. Nó cho phép bạn kéo các phần tử 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. Các ràng buộc xác định vị trí của một phần tử giao diện người dùng trong bố cục. Một ràng buộc đại diện cho một kết nối hoặc căn chỉnh với một chế độ xem khác, bố cục cha hoặc một hướng dẫn vô hình. Khám phá trình chỉnh sửa bố cục và tham khảo hình dưới đây khi bạn làm theo các bước được đánh số:



1. Trong ứng dụng > res > thư mục bố cục trong ngăn Project > Android, nhấp đúp vào tệp activity_main.xml để mở, nếu nó chưa mở.
2. Nhấp vào tab Thiết kế nếu nó chưa được chọn. Bạn sử dụng tab Thiết kế để thao tác với các phần tử và bố cục, và tab Văn bản để chỉnh sửa mã XML cho bố cục.
3. Ngăn Bảng màu hiển thị các thành phần giao diện người dùng mà bạn có thể sử dụng trong bố cục của ứng dụng.
4. Ngăn cây thành phần hiển thị hệ thống phân cấp chế độ xem của các phần tử giao diện người dùng. Các phần tử View được tổ chức thành một hệ thống phân cấp cây gồm cha mẹ và con, trong đó con kế thừa các thuộc tính của cha mẹ của

nó. Trong hình trên, TextView là con của ConstraintLayout . Các em sẽ học về các yếu tố này ở phần sau của bài học này.

5. Các ngăn thiết kế và bản thiết kế của trình chỉnh sửa bố cục hiển thị các yếu tố giao diện người dùng trong bố cục. Trong hình trên, bố cục chỉ hiển thị một phần tử: TextView hiển thị "Hello World".

6. Tab Thuộc tính hiển thị ngăn Thuộc tính để thiết lập thuộc tính cho phần tử giao diện người dùng.

Nhiệm vụ 2: Thêm các thành phần View trong trình chỉnh sửa bố cục Trong tác vụ này, bạn tạo bố cục giao diện người dùng cho ứng dụng HelloToast trong trình chỉnh sửa bố cục bằng cách sử dụng các tính năng ConstraintLayout. Bạn có thể tạo các ràng buộc theo cách thủ công, như được hiển thị sau hoặc tự động bằng cách sử dụng công cụ Tự động kết nối.

2.1 Kiểm tra các ràng buộc của phần tử

Làm theo các bước sau:

1. Mở activity_main.xml từ ngăn Project > Android nếu nó chưa mở. Nếu tab Thiết kế 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 Tự động kết nối cũng nằm trong thanh công cụ. Nó được bật theo mặc định. Đối với bước này, hãy đảm bảo rằng công cụ không bị tắt.

3. Nhấp vào thu phóng trong một cái nhìn cận cảnh. để phóng to các ngăn Thiết kế và Bản thiết kế cho

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

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

Trong các ngăn thiết kế hoặc thiết kế, các tay cầm sau xuất hiện trên phần tử TextView:

- Tay cầm ràng buộc: Để tạo một ràng buộc như trong hình động ở trên, hãy nhấp vào một tay cầm ràng buộc, được hiển thị dưới dạng vòng tròn ở bên cạnh của một phần tử. Sau đó, kéo tay cầm đến một tay cầm ràng buộc khác hoặc đến ranh giới mẹ. Một đường ngoằn ngoèo đại diện cho ràng buộc.

- Thay đổi kích thước tay cầm : Để thay đổi kích thước phần tử, hãy kéo các tay cầm thay đổi kích thước hình vuông. Tay cầm thay đổi thành một góc cạnh trong khi bạn đang kéo nó.

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 giao diện người dùng với bố cục gốc. Sau khi bạn kéo phần tử vào bố cục, nó sẽ tạo ra các ràng buộc dựa trên vị trí của phần tử.

Làm theo các bước sau để thêm Nút:

1. Bắt đầu với một bảng sạch. Phần tử TextView là không cần thiết, vì vậy trong khi nó vẫn được chọn, hãy nhấn phím Delete hoặc chọn Chỉnh sửa > Xóa . Bây giờ bạn có một bố cục hoàn toàn trống.
2. Kéo một nút từ ngăn Bảng màu đến bất kỳ vị trí nào trong bố cục. Nếu bạn thả Nút ở khu vực giữa trên cùng của bố cục, các ràng buộc có thể tự động xuất hiện. Nếu không, bạn có thể kéo các ràng buộc vào phía trên, bên trái và bên phải của bố cục như trong hình động bên dưới.

2.3 Thêm một nút thứ hai vào bố cục

1. Kéo một nút khác từ ngăn Bảng màu vào giữa bố cục như trong hình động bên dưới. Tự động kết nối có thể cung cấp các ràng buộc ngang cho bạn (nếu không, bạn có thể tự kéo chúng).
2. Kéo một ràng buộc dọc xuống cuối bố cục (tham khảo hình bên dưới).

Bạn có thể xóa các ràng buộc khỏi một phần tử bằng cách chọn phần tử và di con trỏ lên phần tử đó để hiển thị nút Xóa ràng buộc. Nhấp vào nút này để loại bỏ 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. Để xóa tất cả các ràng buộc trong toàn bộ bố cục, hãy nhấp vào công cụ Xóa tất cả các ràng buộc trên thanh công cụ. Công cụ này rất 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 giao diện người dùng Ngăn

Thuộc tính cung cấp quyền truy cập vào tất cả các thuộc tính XML mà bạn có thể gán cho phần tử giao diện người dùng. Bạn có thể tìm thấy các thuộc tính (được gọi là thuộc tính) chung cho tất cả các chế độ xem trong tài liệu lớp View . Trong tác vụ này, bạn nhập các giá trị mới và thay đổi giá trị cho các thuộc tính Nút quan trọng, có thể áp dụng cho hầu hết các loại Chế độ xem.

Thay đổi kích thước Nút Trình chỉnh sửa bố cục cung cấp các tay cầm thay đổi kích thước trên 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 tay cầm trên mỗi góc của Chế độ xem để thay đổi kích thước, nhưng làm như vậy sẽ mã hóa cứng kích thước chiều rộng và chiều cao. Tránh mã hóa cứng kích thước cho hầu hết các thành phần Chế độ xem vì 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 phía bên phải của trình soạn thảo bố cục để chọn chế độ định cỡ không sử dụng kích thước được mã hóa cứng. Ngăn Thuộc tính bao gồm một bảng điều chỉnh kích thước hình vuông được gọi là trình kiểm tra chế độ xem ở trên cùng. Các ký hiệu bên trong hình vuông đại diện cho cài đặt chiều cao và chiều rộng như sau:

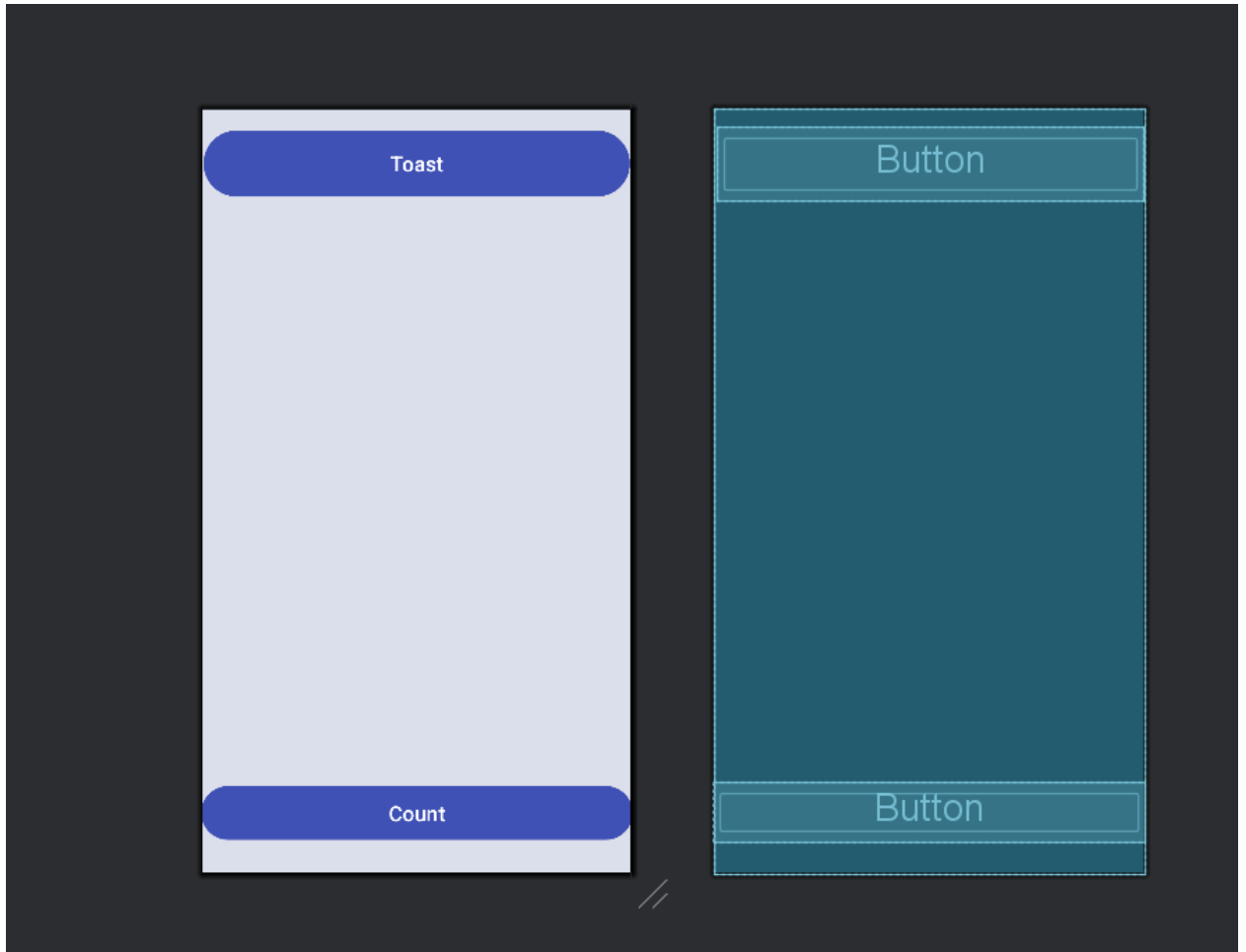
Trong hình trên:

3. Kiểm soát độ cao. Điều khiển này chỉ định thuộc tính `layout_height` và xuất hiện trong hai phân đoạn ở cạnh trên và dưới của hình vuông. Các góc chỉ ra rằng điều khiển này được đặt thành `wrap_content` , có nghĩa là Chế độ xem sẽ mở rộng theo chiều dọc khi cần thiết để phù hợp với nội dung của nó. Dấu "8" cho biết ký quỹ tiêu chuẩn được đặt thành 8dp.
3. Kiểm soát chiều rộng. Điều khiển này chỉ định `layout_width` và xuất hiện trong hai phân đoạn ở bên trái và bên phải của hình vuông. Các góc cho biết rằng điều khiển này được đặt thành `wrap_content` có nghĩa là Chế độ xem sẽ mở rộng theo chiều ngang khi cần thiết để phù hợp với nội dung của nó, lên đến biên độ 8dp.
4. Nút đóng ngăn thuộc tính. Bấm để đóng ngăn.

Làm theo các bước sau:

1. Chọn nút trên cùng trong ngăn Cây thành phần.
2. Nhấp vào tab Attributes ở phía bên phải của cửa sổ trình chỉnh sửa bố cục.

3. Nhấp vào điều khiển chiều rộng hai lần — lần nhấp đầu tiên thay đổi nó thành Cố định với các đường thẳng và lần nhấp thứ hai thay đổi nó thành Kết hợp các ràng buộc với cuộn lò xo, như được hiển thị trong hình ảnh động bên dưới.



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 Nút thứ hai và thực hiện các thay đổi tương tự đối với `layout_width` như ở bước trước, như thể hiện trong hình bên dưới.

Như được hiển thị trong các bước trước, các thuộc tính `layout_width` và `layout_height` trong khung 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`: ● Cài đặt `match_constraint`

mở rộng phần tử View để lấp đầy phần tử cha của nó theo chiều rộng hoặc chiều cao—lên đến lề, nếu được đặt. Cha trong trường hợp này là ConstraintLayout . Bạn tìm hiểu thêm về ConstraintLayout trong nhiệm vụ tiếp theo.

- Cài đặt `wrap_content` thu nhỏ kích thước của phần tử View để nó vừa đủ lớn để bao bọc 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 cho kích thước màn hình của thiết bị, hãy sử dụng một số pixel cố định không phụ thuộc vào mật độ (đơn vị dp). Ví dụ: 16dp có nghĩa là 16 pixel không phụ thuộc vào mật độ.

Mẹo: Nếu bạn thay đổi thuộc tính `layout_width` bằng menu bật lên của nó, thuộc tính `layout_width` được đặt thành không vì không có kích thước được đặt. Cài đặt này giống như `match_constraint` - chế độ xem có thể mở rộng càng nhiều càng tốt để đáp ứng các ràng buộc và cài đặt lề.

3.2 Thay đổi thuộc tính Nút

Để xác định từng Chế độ xem duy nhất trong bố cục Hoạt động, mỗi lớp con Chế độ xem hoặc Chế độ xem (chẳng hạn như Nút) cần có một ID duy nhất. Và để có bất kỳ công dụng nào, các phần tử Button cần văn bản. Các thành phần chế độ xem cũng có thể có hình nền có thể là màu sắc hoặc hình ảnh. Ngăn Thuộc tính cung cấp quyền truy cập vào tất cả các thuộc tính mà bạn có thể gán cho phần tử Xem. Bạn có thể nhập các giá trị cho từng thuộc tính, chẳng hạn như `android:id` , `background` , `textColor` và thuộc tính `text`.

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

1. Sau khi chọn Nút đầu tiên, hãy chỉnh sửa trường ID ở đầu ngăn Thuộc tính để `button_toast` thuộc tính `android:id`, thuộc tính này 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 văn bản thành `Toast` .
5. Thực hiện các thay đổi thuộc tính tương tự cho Nút thứ hai , sử dụng `button_count` làm ID, Đếm cho thuộc tính văn bản và cùng màu cho nền và văn bản như các bước trước. `colorPrimary` là màu chính của chủ đề, một trong những màu cơ sở chủ đề được xác định trước được xác định

trong tệp tài nguyên colors.xml. Nó được sử dụng cho thanh ứng dụng. Sử dụng màu cơ bản cho các yếu tố giao diện người dùng khác sẽ tạo ra một giao diện người dùng thống nhất. Bạn sẽ tìm hiểu thêm về chủ đề ứng dụng và Material Design trong một bài học khác.

Nhiệm vụ 4: Thêm TextEdit và đặt các thuộc tính của nó

Một trong những lợi ích của ConstraintLayout là khả năng căn chỉnh hoặc hạn chế các phần tử liên quan đến các phần tử khác. Trong nhiệm vụ này, bạn sẽ thêm một TextView ở giữa bố cục và hạn chế nó theo chiều ngang vào lề và theo chiều dọc với hai phần tử Button. Sau đó, bạn sẽ thay đổi các thuộc tính cho TextView trong ngăn Attributes (Thuộc tính).

4.1 Thêm TextView và các ràng buộc

1. Như thể hiện trong hình động bên dưới, hãy kéo TextView từ ngăn Palette đến phần trên của bố cục và kéo một ràng buộc từ đầu TextView đến tay cầm ở cuối Nút Toast . Điều này hạn chế TextView nằm bên dưới Button .
2. Như thể hiện trong hình động bên dưới, kéo một ràng buộc từ dưới cùng của TextView đến tay cầm ở trên cùng của Nút đếm và từ các cạnh của TextView đến các cạnh của bố cục. Điều này hạn chế TextView ở giữa bố cục giữa hai phần tử Button.

4.2 Đặt thuộc tính TextView

Với TextView đã chọn, hãy mở ngăn Thuộc tính, nếu nó chưa mở. Đặt thuộc tính cho TextView như trong hình động bên dưới. Các thuộc tính bạn chưa gặp 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 ngang và 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 các thuộc tính , cuộn xuống trang thứ hai của thuộc tính sang nền , sau đó nhập #FFF00 cho màu vàng.

8. Cuộn xuống trọng lực , mở rộng trọng lực và chọn center_ver (đối với trung tâm-dọc).

- textSize: Kích thước văn bản của TextView. Đối với bài học này, kích thước được đặt thành 160sp . sp là viết tắt của pixel-independent tỷ lệ và giống như dp , là một đơn vị chia tỷ lệ theo mật độ màn hình và tùy chọn 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.

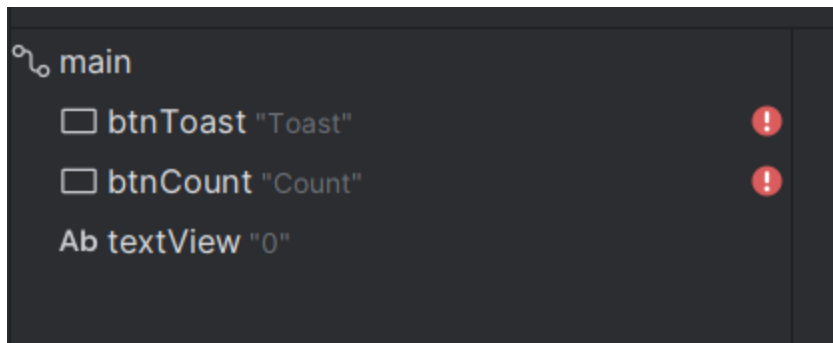
- textStyle và textAlignment: Kiểu văn bản, được đặt thành B (in đậm) trong bài học này và căn chỉnh văn bản, được đặt thành ALIGNCENTER (căn giữa đoạn văn).

- trọng lực: Thuộc tính trọng lực chỉ định cách một Chế độ xem được căn chỉnh trong Chế độ xem hoặc ViewGroup gốc của nó. Trong bước này, bạn căn giữa TextView được căn giữa theo chiều dọc trong ConstraintLayout gốc.

Bạn có thể nhận thấy rằng thuộc tính nền nằm trên trang đầu tiên của ngăn Thuộc tính cho Nút, nhưng trên trang thứ hai của ngăn Thuộc tính cho TextView . Ngăn Thuộc tính thay đổi cho từng loại Dạng xem: Các thuộc tính phổ biến nhất cho loại Chế độ xem xuất hiện trên trang đầu tiên và phần còn lại được liệt kê trên trang thứ hai. Để quay lại trang đầu tiên của ngăn Thuộc tính, hãy bấm 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 giao diện người dùng trong Cây thành phần. Di con trỏ của bạn qua các dấu chấm than này để xem thông báo cảnh báo, như được hiển thị bên dưới. Cảnh báo tương tự xuất hiện cho cả ba phần tử: chuỗi được mã hóa cứng nên sử dụng tài nguyên.



Cách dễ nhất để khắc phục sự cố bố cục là chỉnh sửa bố cục trong XML. Mặc dù trình soạn thảo bố cục là một công cụ mạnh mẽ, nhưng một số thay đổi dễ dàng 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 nó 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, cho thấy một phần của mã XML cho bố cục, các cảnh báo được đánh dấu — các chuỗi được mã hóa cứng "Toast" và "Count". (Mã hóa cứng "0" cũng được đánh dấu nhưng không được hiển thị trong hình.) Di con trỏ của bạn qua chuỗi được mã hóa cứng "Toast" để xem thông báo cảnh báo.

```
<Button
    android:id="@+id/btnToast"
    android:layout_width="409dp"
    android:layout_height="71dp"
    android:layout_marginTop="16dp"
    android:backgroundTint="#3F51B5"
    android:text="Toast"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btnCount"
    android:layout_width="414dp"
    android:layout_height="59dp"
    android:layout_marginStart="154dp"
    android:layout_marginEnd="167dp"
    android:layout_marginBottom="28dp"
    android:backgroundTint="#3F51B5"
    android:text="Count"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.481"
app:layout_constraintStart_toStartOf="parent"
/>

<TextView
    android:id="@+id/textView"
    android:layout_width="368dp"
    android:layout_height="530dp"
    android:layout_marginStart="50dp"
    android:layout_marginEnd="51dp"
    android:background="#FFEB3B"
    android:gravity="center_horizontal|center_vertical"
    android:text="0"
    android:textSize="96sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.444"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

5.2 Trích xuất tài nguyên chuỗi

Thay vì mã hóa các chuỗi cứng, cách tốt nhất là sử dụng tài nguyên chuỗi, đại diện cho các chuỗi. Việc có các chuỗi trong một tệp riêng biệt giúp quản lý chúng dễ dàng hơn, đặc biệt 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 đánh dấu 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 . 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 "Đếm" và count_initial_value cho "0" .
6. Trong ngăn Project > Android, 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:

```

<resources>
    <string name="app_name">Hello Toast</string>
    <string name="button_label_toast">Hello Toast</string>
    <string name="button_label_count">Hello Toast</string>

```

```
<string name="count_initial_value">0</string>
</resources>
```

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!":

```
<resources>
    <string name="app_name">Hello Toast</string>
    <string name="button_label_toast">Hello Toast</string>
    <string name="button_label_count">Hello Toast</string>
    <string name="count_initial_value">0</string>
    <string name="toast_message">Hello Toast!</string>
</resources>
```

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

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

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

Trình xử lý nhấp chuột là một phương thức được gọi khi người dùng nhấp hoặc nhấn vào phần tử giao diện người dùng có thể nhấp vào. Trong Android Studio, bạn có thể chỉ định tên của phương thức trong trường onClick trong ngăn Attributes của tab Design. 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 nút. Bạn sẽ sử dụng phương thức thứ hai 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 chỉnh sửa XML đang mở (tab Văn bản), hãy tìm Nút với android:id được đặt thành button_toast
2. Thêm thuộc tính android:onClick vào cuối phần tử button_toast sau thuộc tính cuối cùng và trước chỉ báo kết thúc />:

```
android:onClick="showToast" />
```

3. Nhấp vào biểu tượng bóng đèn màu đỏ xuất hiện bên cạnh thuộc tính. Chọn Tạo trình xử lý nhấp chuột, 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 Tạo 'showToast(view)' trong MainActivity và nhấp vào

OK . Hành động này tạo sơ khai 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 nút button_count : Thêm thuộc tính android:onClick vào cuối và thêm trình xử lý nhấp chuột:

```
android:onClick="countUp"
```

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

1.3) Trình chỉnh sửa bố cục

1.4) Văn bản và các chế độ cuộn

1.5) Tài nguyên có sẵn

Bài 2) Activities

2.1) Activity và Intent

2.2) Vòng đời của Activity và trạng thái

2.3) Intent ngầm định

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

3.1) Trình gỡ lỗi

3.2) Kiểm thử đơn vị

3.3) Thư viện hỗ trợ

CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG

Bài 1) Tương tác người dùng

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

Bài 2) Trải nghiệm người dùng thú vị

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

Bài 3) Kiểm thử giao diện người dùng

- 3.1) Espresso cho việc kiểm tra UI**

CHƯƠNG 3. LÀM VIỆC TRONG NỀN

Bài 1) Các tác vụ nền

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

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

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

CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG

Bài 1) Tùy chọn và cài đặt

- 1.1) Shared preferences**

1.2) Cài đặt ứng dụng

Bài 2) Lưu trữ dữ liệu với Room

2.1) Room, LiveData và ViewModel

2.2) Room, LiveData và ViewModel