

TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



GIÁO TRÌNH

THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG

Hà Nội, 2.2025

MỤC LỤC

CHƯƠNG 1.	Làm quen.....	3
Bài 1)	Tạo ứng dụng đầu tiên	3
1.1)	Android Studio và Hello World	3
1.2)	Giao diện người dùng tương tác đầu tiên	5
1.3)	Trình chỉnh sửa bố cục	9
1.4)	Văn bản và các chế độ cuộn	9
1.5)	Tài nguyên có sẵn.....	9
Bài 2)	Activities	9
2.1)	Activity và Intent	9
2.2)	Vòng đời của Activity và trạng thái	9
2.3)	Intent ngầm định.....	9
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ	9
3.1)	Trình gỡ lỗi	9
3.2)	Kiểm thử đơn vị.....	9
3.3)	Thư viện hỗ trợ.....	9
CHƯƠNG 2.	Trải nghiệm người dùng	10
Bài 1)	Tương tác người dùng	10
1.1)	Hình ảnh có thể chọn	10
1.2)	Các điều khiển nhập liệu	10
1.3)	Menu và bộ chọn	10
1.4)	Điều hướng người dùng	10
1.5)	RecyclerView	10
Bài 2)	Trải nghiệm người dùng thú vị.....	10
2.1)	Hình vẽ, định kiểu và chủ đề	10
2.2)	Thẻ và màu sắc	10

2.3)	Bố cục thích ứng.....	10
Bài 3)	Kiểm thử giao diện người dùng.....	10
3.1)	Espresso cho việc kiểm tra UI	10
CHƯƠNG 3. Làm việc trong nền		10
Bài 1)	Các tác vụ nền.....	10
1.1)	AsyncTask	10
1.2)	AsyncTask và AsyncTaskLoader	10
1.3)	Broadcast receivers	10
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	10
2.1)	Thông báo	10
2.2)	Trình quản lý cảnh báo	10
2.3)	JobScheduler.....	10
CHƯƠNG 4. Lưu dữ liệu người dùng		11
Bài 1)	Tùy chọn và cài đặt.....	11
1.1)	Shared preferences.....	11
1.2)	Cài đặt ứng dụng.....	11
Bài 2)	Lưu trữ dữ liệu với Room	11
2.1)	Room, LiveData và ViewModel.....	11
2.2)	Room, LiveData và ViewModel.....	11
3.1)	Trình gỡ lỗi	

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

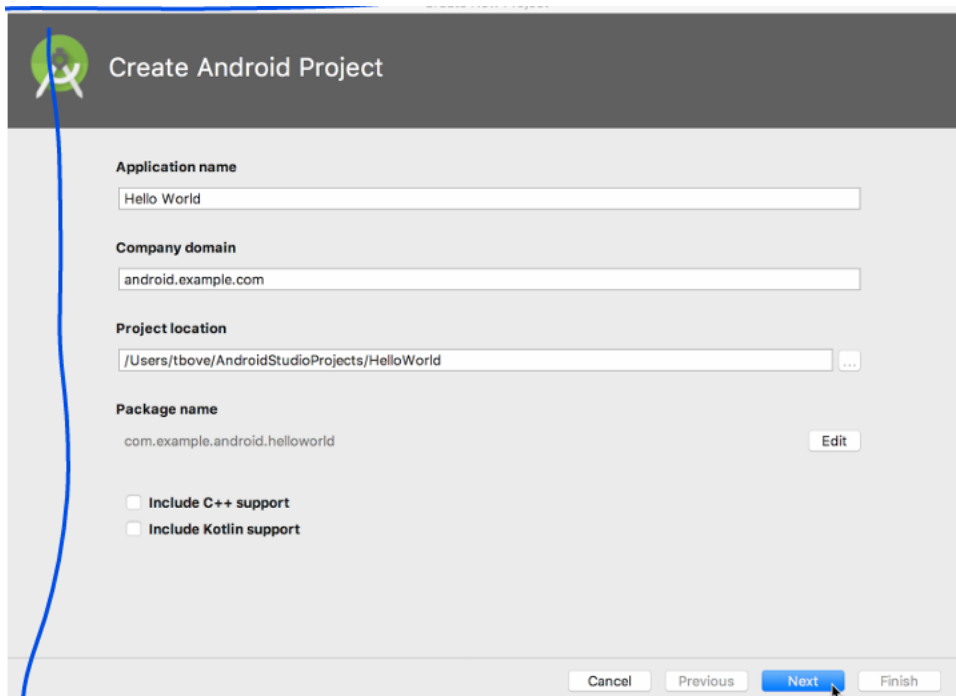
Giới thiệu

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

Những gì Bạn nên biết

Bạn nên có khả năng:

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



Những gì Bạn sẽ cần:

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

Những gì bạn sẽ học

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

Những gì bạn sẽ làm

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

1.2) Giao diện người dùng tương tác đầu tiên

Giao diện

Giao diện người dùng (UI) xuất hiện trên màn hình của thiết bị Android bao gồm một hệ thống phân cấp các đối tượng gọi là views – mọi phần tử của màn hình đều là 1 View . Lớp View đại diện cho khối xây dựng cơ bản của tất cả các thành phần UI và là lớp cơ sở cho các lớp cung cấp các thành phần UI như nút bấm, checkboxes, và trường nhập văn bản. Các lớp của View con phổ biến sẽ được mô tả ở bài tập sau:

- TextView để hiển thị văn bản
- EditText cho phép người dùng nhập và chỉnh sửa văn bản
- Button và các phần tử có thể nhấp khác (chẳng hạn như RadioButton, Checkbox , và Spinner) cung cấp hành vi tương tác
- ScrollView và RecyclerView để hiển thị các mục có thể cuộn.
- ImageView để hiển thị ảnh.
- ConstraintLayout and LinearLayout để chứa các phần tử View và định vị chúng.

Mã java điều khiển và hiển thị giao diện người dùng được chứa trong 1 lớp kế thừa Activity. 1 Activity thường được liên kết với 1 bố cục của UI views đc định nghĩa trong tập tin xml. Tập tin XML thường được đặt tên theo Activity tương ứng và định nghĩa bố cục của phần tử View trên màn hình.

Ví dụ, trong ứng dụng HelloWorld mã trong MainActivity hiển thị bố cục được định nghĩa trong tập tin activity_main.xml, trong đó có 1 TextView với nội dung “Hello World”.

Trong các ứng dụng phức tạp hơn, 1 Activity phải thực thi nhiều hành động để 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 Activity trong 1 bài học khác.

Trong bài thực hành này, bạn học được cách tạo ứng dụng tương tác đầu tiên- 1 ứng dụng có thể cho phép người dùng tương tác. Bạn tạo 1 ứng dụng sử dụng mẫu Empty Activity. Bạn cũng học cách sử dụng trình chỉnh sửa bố cục, và làm thế nào để chỉnh sửa bố cục trong XML. Bạn cần phát triển các kỹ năng này để bạn có thể hoàn thành các bài thực hành khác trong khóa học này.

Những kiến thức bạn cần biết trước

- Cách cài đặt và mở Android Studio.
- Cách tạo 1 ứng dụng HelloWorld.
- Cách chạy ứng dụng HelloWorld.

Những kiến thức bạn sẽ tìm hiểu

- Làm thế nào để tạo 1 ứng dụng với giao diện tương tác.
- Làm thế nào sử dụng trình chỉnh sửa bố cục để thiết kế 1 bố cục.
- Làm thế nào để chỉnh sửa bố cục trong XML.
- Nhiều thuật ngữ mới. Kiểm tra trong [Vocabulary and concepts glossary](#) để hiểu rõ hơn về các định nghĩa

Những cái bạn sẽ làm

- Điều chỉnh từng phần tử trong **ConstraintLayout** để ràng buộc chúng với lề và các phần tử khác.
- Thay đổi thuộc tính của các phần tử giao diện người dùng (UI).
- Chỉnh sửa bố cục của ứng dụng bằng **XML**.
- Trích xuất các chuỗi mã cứng (**hardcoded strings**) vào tài nguyên chuỗi (**string resources**).
- Triển khai phương thức xử lý sự kiện nhấp (**click-handler methods**) để hiển thị thông báo trên màn hình khi người dùng nhấn vào **Button**.

Tổng quan về ứng dụng

Ứng dụng **HelloToast** bao gồm hai phần tử **Button** và một **TextView**.

- Khi người dùng nhấn vào **Button** đầu tiên, một thông báo ngắn (**Toast**) sẽ hiển thị trên màn hình.
- Khi nhấn vào **Button** thứ hai, bộ đếm số lần nhấn (**click counter**) sẽ tăng lên. Giá trị của bộ đếm được hiển thị trong **TextView**, bắt đầu từ 0.

Đây là giao diện của ứng dụng sau khi hoàn thành:

Nhiệm vụ 1: Tạo và khám phá 1 dự án

Trong bài thực hành này, bạn sẽ thiết kế và triển khai một dự án cho ứng dụng **HelloToast**.

Liên kết đến mã nguồn giải pháp sẽ được cung cấp ở cuối bài.

1.1 Tạo dự án Android Studio

a) Mở Android Studio và tạo một dự án mới với các thông số sau:

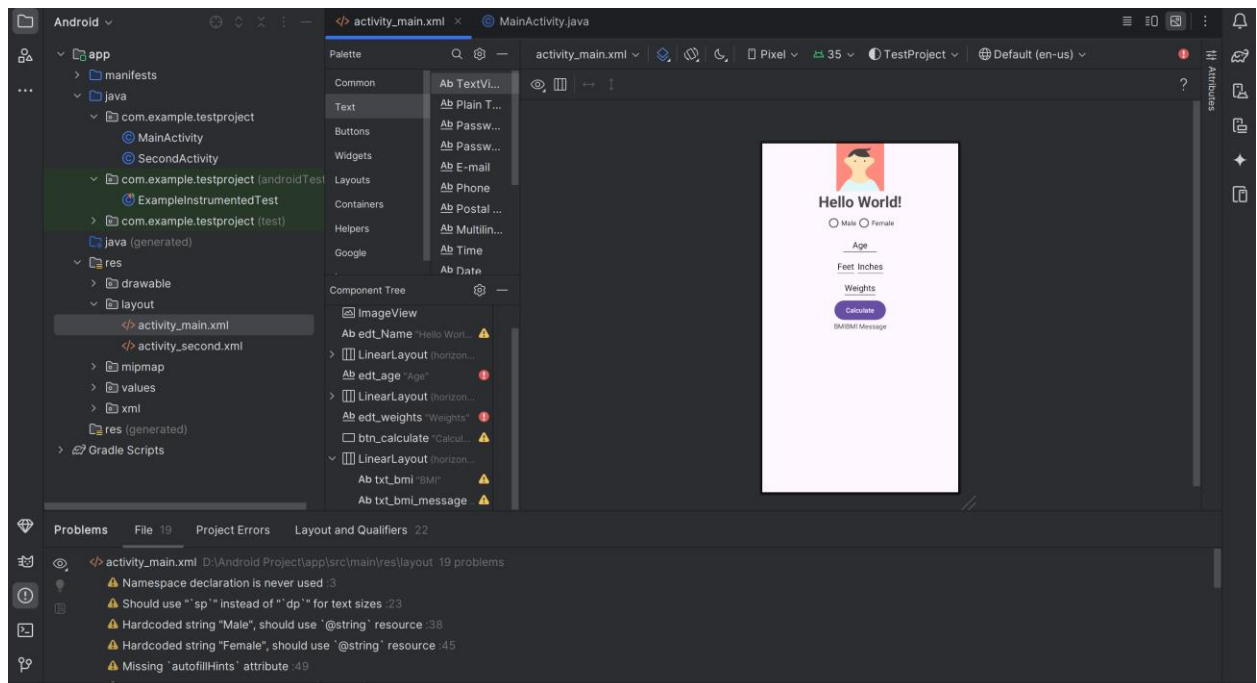
Attribute	Value
Application Name	Hello Toast
Company Name	com.example.android (or your own domain)
Phone and Tablet Minimum SDK	API15: Android 4.0.3 IceCreamSandwich
Template	Empty Activity
Generate Layout file box	Selected
Backwards Compatibility box	Selected

15. Chọn **Run > Run app** hoặc nhấn vào **biểu tượng Run** trên thanh công cụ để biên dịch và chạy ứng dụng trên trình giả lập hoặc thiết bị của bạn.

1.2 Khám phá trình chỉnh sửa bố cục (Layout Editor)

Android Studio cung cấp **layout editor** để giúp bạn nhanh chóng xây dựng bố cục giao diện người dùng (**UI**). Công cụ này cho phép bạn:

- Kéo thả các phần tử vào **chế độ thiết kế trực quan** và **chế độ bản vẽ**.
- Sắp xếp vị trí các phần tử trong bố cục.
- Thêm **constraints** (ràng buộc) để xác định vị trí của phần tử trong giao diện.



1. Trong **app > res > thư mục layout** trong **Project > Android** panel, điều hướng đến **app > res > layout**, sau đó **double-click** vào tệp **activity_main.xml** để mở (nếu chưa mở).
2. Nhấn vào tab **Design** (nếu chưa được chọn).
 - Tab **Design** dùng để thao tác với các phần tử và bố cục.
 - Tab **Text** dùng để chỉnh sửa mã XML của bố cục.
3. **Palettes pane** hiển thị các phần tử UI mà bạn có thể sử dụng trong bố cục ứng dụng.
4. **Component tree pane** hiển thị **cây phân cấp của các phần tử UI**.
 - Các **View** được tổ chức theo cấu trúc cha - con, trong đó **con** kế thừa thuộc tính từ **cha**.

- Ví dụ trong hình minh họa, **TextView** là **con** của **ConstraintLayout**.
5. **Design & Blueprint panes** hiển thị giao diện bố cục với các phần tử UI.
- Trong hình minh họa, bố cục chỉ có một phần tử: **TextView** hiển thị dòng chữ "Hello World".
6. **Attributes tab** hiển thị các thuộc tính của phần tử UI và cho phép chỉnh sửa trực tiếp.

Task 2: Thêm các phần tử View trong Layout Editor

Trong nhiệm vụ này

Trình chỉnh sửa bố cục

1.3) Văn bản và các chế độ cuộn

1.4) 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