**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**

### Giao diện người dùng tương tác đầu tiên

### Trình chỉnh sửa bố cục

### Văn bản và các chế độ cuộn

### Tài nguyên có sẵn

## Activities

### Activity và Intent

### Vòng đời của Activity và trạng thái

### Intent ngầm định

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

### Trình gỡ lỗi

### Kiểm thử đơn vị

### Thư viện hỗ trợ

# TRẢI NGHIỆM NGƯỜI DÙNG

## Tương tác người dùng

### Hình ảnh có thể chọn

Giới thiệu

Giao diện người dùng (UI) xuất hiện trên màn hình của một thiết bị Android bao gồm một hệ thống phân cấp các đối tượng được gọi là View.

Lớp View đại diện cho khối xây dựng cơ bản của tất cả các thành phần UI. View là lớp cơ sở cho các lớp cung cấp các thành phần UI tương tác, chẳng hạn như phần tử Button. Một Button là một phần tử UI mà người dùng có thể nhấn hoặc bấm vào để thực hiện một hành động.

Bạn có thể biến bất cứ View nào, chẳng hạn như ImageView, thành một thành phần UI có thể nhấn hoặc bấm vào. Bạn phải lưu trữ hình ảnh cho ImageView trong thư mục drawables của dự án của bạn.

Trong bài thực hành này, bạn sẽ học cách sử dụng các hình ảnh làm phần tử mà người dùng có thể nhấn hoặc bấm vào.

Những gì Bạn nên biết

Bạn nên có khả năng:

* Tạo một dự án Android Studio từ một mẫu và tạo bố cục chính.
* Chạy ứng dụng trên mô hình giả lập hoặc thiết bị được kết nối.
* Tạo và chỉnh sửa phần tử UI bằng trình chỉnh sửa bố cục và mã XML.
* Truy cập các phần tử UI từ mã của bạn sử dụng findViewById().
* Xử lý thao tác bấm vào Button.
* Hiển thị thông báo Toast.
* Thêm những hình ảnh vào thư mục drawable của dự án.

Những gì Bạn sẽ cần:

* Cách sử dụng hình ảnh như một phần tử tương tác để thực hiện hành động.
* Cách thiết lập các thuộc tính cho phần tử ImageView trong trình chỉnh sửa bố cục.
* Cách thêm phương thức onClick() để hiển thị thông báo Toast.

Những gì bạn sẽ làm

* Tạo một dự án Android Studio mới cho một ứng dụng giả lập đặt món tráng miệng sử dụng hình ảnh làm phần tử tương tác.
* Thiết lập xử lý onClick() cho các hình ảnh để hiển thị ra các thông báo Toast khác nhau.
* Thay đổi nút hành động nổi do mẫu cung cấp để hiển thị biểu tượng khác nhau và khởi chạy một Activity khác.

**Tổng quan ứng dụng**

Trong bài thực hành này, bạn tạo và xây dựng một ứng dụng mới bắt đầu bằng mẫu Basic Activity, mô phỏng một ứng dụng đặt món tráng miệng. Người dùng có thể nhấn vào hình ảnh để thực hiện một hành động—trong trường hợp này là hiển thị một thông báo Toast—như minh họa trong hình bên dưới. Người dùng cũng có thể nhấn vào nút giỏ hàng để tiến tới Activity tiếp theo.



**Nhiệm vụ 1: Thêm hình ảnh vào bố cục**

Bạn có thể làm một chế độ xem có thể nhấn được, như một nút, bằng cách thêm thuộc tính android:onClick trong bố cục XML. Ví dụ, bạn có thể làm cho ảnh hoạt động giống như một nút bằng cách thêm android:onClick vào ImageView.

Trong nhiệm vụ này bạn tạo một nguyên mẫu của ứng dụng đặt món tráng miệng từ một quán cà phê. Sau khi bắt đầu một dự án mới dựa trên mẫu Basic Activity, bạn sửa đổi TextView “Hello world” thành nội dung phù hợp và thêm hình ảnh mà người dùng có thể nhấn vào.

**1.1 Bắt đầu dự án mới**

1. Bắt đầu một dự án Android Studio mới với tên ứng dụng là Droid Cafe.

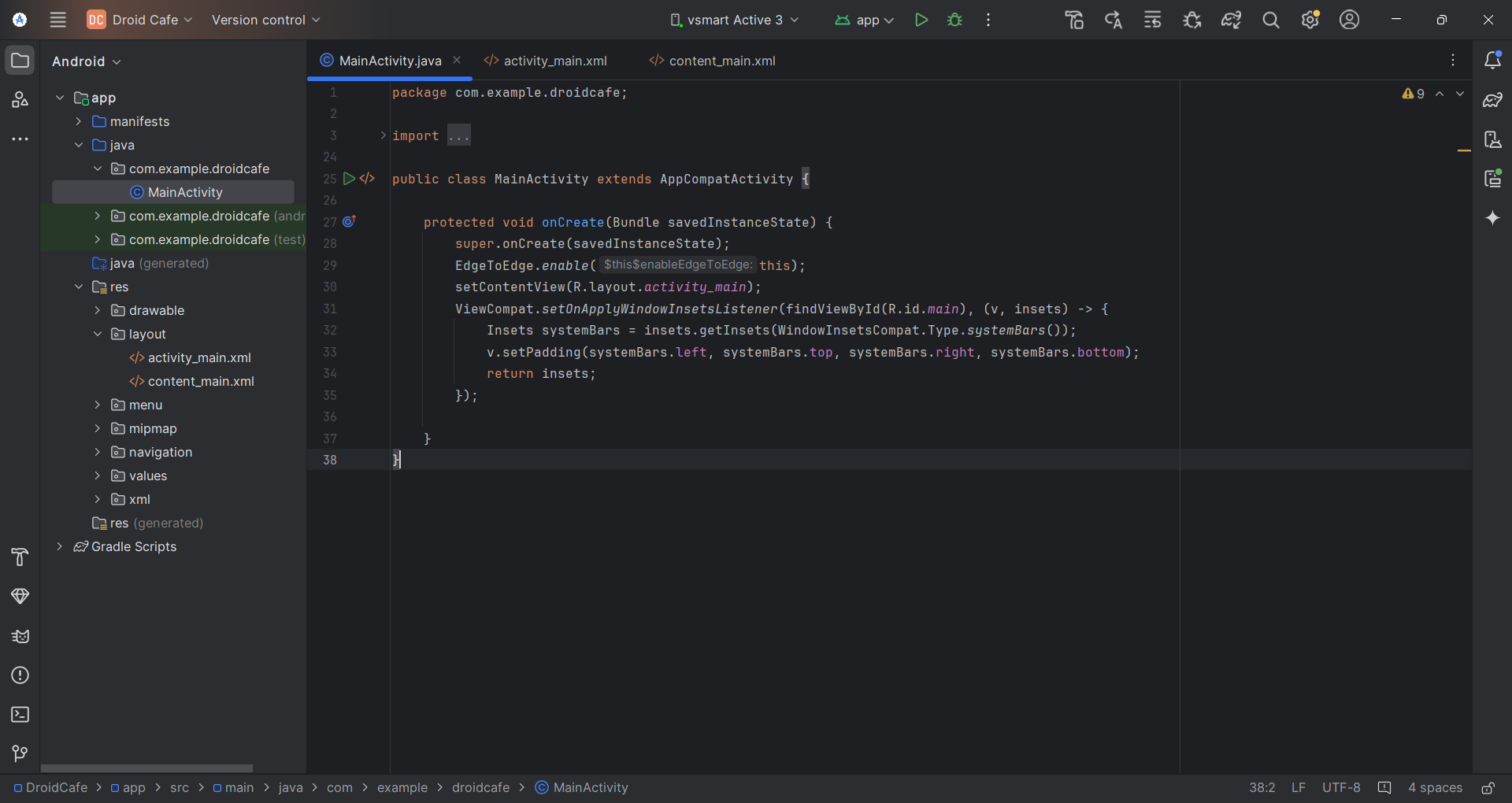
2. Chọn mẫu Basic Activity, và chấp nhận tên Activity mặc định (MainActivity). Đảm bảo các tùy chọn **Generate Layout file** và **Backwards Compatibility (AppCompat)** được chọn.

A screenshot of a computer

AI-generated content may be incorrect.

3. Bấm **Finish**.

Dự án mở ra với 2 bố cục trong trong thư mục **res > layout**: activity\_main.xml dành cho thanh ứng dụng và nút hành động nổi (Bạn không thay đổi trong nhiệm vụ này), và content\_main.xml dành cho mọi thứ khác trong giao diện.



4. Mở **content\_main.xml** và bấm vào tab **Design** (nếu nó không được chọn) để hiển thị trình chỉnh sửa bố cục.

A screenshot of a computer

AI-generated content may be incorrect.

5. Chọn TextView “Hello World” trong bố cục và mở bảng Attributes.

A screenshot of a computer

AI-generated content may be incorrect.

6. Thay đổi thuộc tính textintro như sau:

|  |  |
| --- | --- |
| **Trường thuộc tính** | **Nhập giá trị sau:** |
| ID | textintro |
| text | Đổi Hello World sang Droid Desserts |
| textStyle | B (bold) |
| textSize | 24sp |

Việc này sẽ thêm thuộc tính android:id vào TextView với id được thiết lập thành textintro, thay đổi văn bản, làm văn bản đậm, và thiết lập kích thước văn bản lớn hơn là 24sp.

A screenshot of a computer

AI-generated content may be incorrect.

7. Xóa ràng buộc kéo từ phía dưới của TextView textintro xuống đáy của bố cục, để TextView nằm ở phía trên của bố cục, và chọn **8** (8dp) cho khoảng cách lề trên như hình dưới

A screenshot of a computer

AI-generated content may be incorrect.

8. Trong bài học trước, bạn đã học cách trích xuất một chuỗi tài nguyên từ một chuỗi văn bản. Bấm vào tab **Text** để chuyển sang mã XML, và trích xuất chuỗi “Droid Desserts ” trong TextView và nhập intro\_text làm tên string resources.

A screenshot of a computer

AI-generated content may be incorrect.

**1.2 Thêm các ảnh**

Ba hình ảnh (donut\_circle.png, froyo\_circle.png và icecream\_circle.png) được cung cấp cho ví dụ này, mà bạn có thể tải xuống. Ngoài ra, bạn có thể thay bằng hình ảnh của riêng mình dưới dạng tệp PNG, nhưng chúng phải có kích thước khoảng 113 x 113 pixels để sử dụng trong ví dụ này.

Bước này cũng giới thiệu một kỹ thuật mới trong trình chỉnh sửa bố cục: sử dụng nút **Fix** trong các thông báo cảnh báo để trích xuất string resources.

1. Để sao chép các hình ảnh vào dự án của bạn, đầu tiên hãy đóng dự án.

2. Sao chép tệp hình ảnh vào thư mục **drawable** của dự án. Tìm thư mục **drawable** trong một dự án bằng cách sử dụng đường dẫn này: project\_name **> app > src > res > drawable**.

3. Mở lại dự án của bạn.

4. Mở tệp tin **content\_main.xml** và bấm vào tab **Design** (nếu nó chưa được chọn).

5. Kéo một ImageView vào bố cục, chọn hình ảnh **donut\_circle** cho nó, và giới hạn nó vào TextView trên cùng và phía bên trái của bố cục với khoảng cách **24** (24dp) cho cả hai ràng buộc, như được hiển thị btrong hình động bên dưới.

6. Trong bảng Attributes, nhập các giá trị cho các thuộc tính:

|  |  |
| --- | --- |
| **Trường thuộc tính** | **Nhập giá trị sau:** |
| ID | donut |
| contentDescription | Donuts are glazed and sprinkled with candy. (Bạn có thể sao chép và dán văn bản vào trường này.) |

7. Kéo ImageView thứ hai vào bố cục, chọn hình ảnh **icecream\_circle** cho nó, và ràng buộc nó bên dưới ImageView thứ nhất và bên trái của bố cục với khoảng cách lề **24** (24dp) cho cả hai ràng buộc.

8. Trong bảng **Attributes**, nhập các giá trị cho các thuộc tính:

|  |  |
| --- | --- |
| **Trường thuộc tính** | **Nhập giá trị sau:** |
| ID | Ice\_cream |
| contentDescription | Ice cream sandwiches have chocolate wafes and vanilla filling. (Bạn có thể sao chép và dán văn bản vào trường này.) |

9. Kéo một ImageView thứ ba vào bố cục, chọn hình ảnh **froyo\_circle** cho nó, và ràng buộc nó bên dưới của ImageView thứ hai và bên trái của bố cục với khoảng cách lề **24** (24dp) cho cả hai ràng buộc.

10. Trong bảng **Attributes**, nhập các giá trị cho các thuộc tính:

|  |  |
| --- | --- |
| **Trường thuộc tính** | **Nhập giá trị sau:** |
| ID | froyo |
| contentDescription | FroYo is premium self-serve frozen yogurt. (Bạn có thể sao chép và dán văn bản vào trường này.) |

11. Bấm vào biểu tượng cảnh báo ở góc trên bên trái của trình chỉnh sửa bố cục để mở bảng cảnh báo, nó hiển thị các cảnh báo về hardcoded text

12. Mở rộng từng cảnh báo **Hardcoded text**, cuộn xuống cuối thông báo của cảnh báo và bấm vào nút **Fix** như bên dưới:

Việc sửa từng cảnh báo hardcoded text trích xuất string resources cho văn bản. Hộp thoại **Extract Resource** xuất hiện, và bạn có thể nhập tên cho string resources. Nhập tên string resources theo:

|  |  |
| --- | --- |
| **Chuỗi** | **Nhập tên theo:** |
| Donuts are glazed and sprinkled with candy. | donuts |
| Ice cream sandwiches have chocolate wafers and  vanilla filling. | ice\_cream\_sandwiches |
| FroYo is premium self‑serve frozen yogurt. | froyo |

Bố cục bây giờ sẽ trông giống như hình bên dưới:

**1.3 Thêm mô tả văn bản**

Trong bước này bạn thêm một văn bản mô tả (TextView) cho mỗi món tráng miệng. Vì bạn đã trích string resources cho trường contentDescription cho mỗi phần tử ImageView, bạn có thể sử dụng cùng các string resources này cho mỗi TextView mô tả.

1. Kéo một phần tử TextView vào bố cục.

2. Ràng buộc bên trái của phần tử này vào bên phải của ImageView donut và ràng buộc bên trên của nó vào cạnh trên của ImageView donut, cả hai với khoảng cách **24** (24dp).

3. Ràng buộc cạnh phải của phần rử này vào cạnh phải của bố cục và cũng sử dụng khoảng cách **24** (24dp). Nhập **donut\_description** cho trường ID trong bảng **Attributes**. TextView mới sẽ xuất hiện bên cạnh hình ảnh donut như hình bên dưới.

4. Trong bảng **Attributes** thay đổi chiều rộng trong inspector pane thành **Match** **Constraints**:

5. Trong bảng Attributes, bắt đầu nhập string resources cho trường text bằng cách đặt ký hiệu @ trước: @d. Bấm vào tên string resources (@string/donuts) khi nó xuất hiện gợi ý:

6. Lặp lại các bước trên để thêm một TextView thứ hai, được ràng buộc vào cạnh phải và bên trên của ImageView ice\_cream, và cạnh phải của bố cục. Nhập vào bảng **Attributes**:

|  |  |
| --- | --- |
| **Trường Attribute** | **Nhập theo sau:** |
| ID | ice\_cream\_descriptiom |
| Left, right, and top margins | 24 |
| layout\_width | match\_constraint |
| text | @string/ice\_cream\_sandwiches |

7. Lặp lại các bước trên để thêm một TextView thứ ba, được ràng buộc vào cạnh phải và cạnh trên của ImageView froyo, và cạnh phải của nó vào cạnh phải của bố cục. Nhập vào bảng **Attributes**:

|  |  |
| --- | --- |
| **Trường Attribute** | **Nhập theo sau:** |
| ID | froyo\_descriptiom |
| Left, right, and top margins | 24 |
| layout\_width | match\_constraint |
| text | @string/froyo |

Bố cục bây giờ sẽ trông giống như hình bên dưới:

**Mã giải pháp nhiệm vụ 1**

Bố cục XML cho tệp content.xml được hiển thị bên dưới

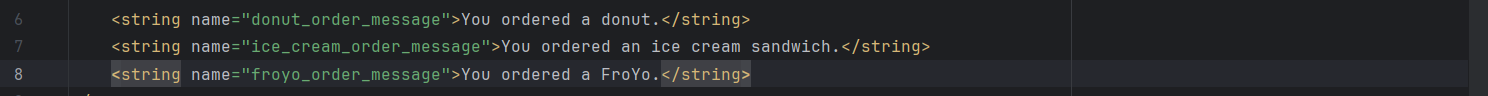
**Nhiệm vụ 2: Thêm phương thức onClick cho hình ảnh**

Để làm cho một View có thể bấm được để người dùng có thể chạm (hoặc bấm) vào nó, hãy thêm thuộc tính android:onClick trong bố cục XML và chỉ định phương thức xử lý sự kiện click. Ví dụ, banh có thể làm một ImageView hoạt động như một Button bằng cách thêm android:onClick vào ImageView. Trong nhiệm vụ này, bạn sẽ làm cho các hình ảnh trong bố cục của bạn có thể nhấn được

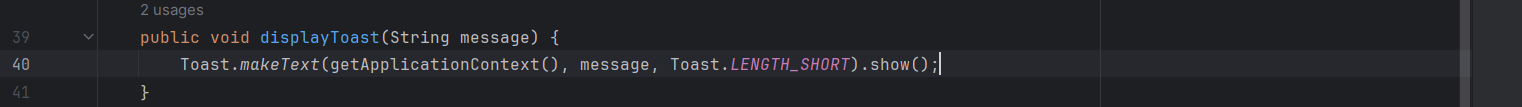
**2.1 Tạo phương thức Toast**

Trong nhiệm vụ này, bạn thêm từng phương thức cho thuộc tính android:onClick để gọi khi mỗi hình ảnh được nhấn vào. Trong nhiệm vụ này, các phương thức đơn giản hiển thị một thông báo Toast, nó cho biết hình ảnh đã được chạm vào. (Trong một chương khác, bạn sẽ sửa đổi các phương thức này để khởi động một Activity khác.)

1. Để sử dụng string resources trong code Java, bạn nên thêm chúng vào tệp strings.xml. Mở rộng **res > values** trong bảng điều khiển **Project > Android**, và mở **string.xml**. Thêm các string resources sau cho các chuỗi để hiển thị trong thông báo Toast:



2. Mở **MainActivity**, và thêm phương thức displayToast() sau vào cuối **MainActivity** (trước dấu ngoặc đóng):



Mặc dù bạn có thể thêm phương thức này ở bất cứ vị trí nào trong **MainActivity**, nhưng thực tiễn tốt nhất là đặt các phương thức bên dưới các phương thức đã được cung cấp trong **MainActivity** bởi mẫu.

**2.2 Tạo trình xử lý sự kiện click**

Mỗi hình ảnh có thể nhấp cần một trình xử lý sự kiện nhấp chuột—một phương thức cho thuộc tính android:onClick để gọi. Trình xử lý sự kiện nhấp chuột, nếu được gọi từ thuộc tính android:onClick, phải là public, trả về void, và định nghĩa một View như là tham số duy nhất của nó. Thực hiện các bước sau để thêm các trình xử lý sự kiện nhấp chuột:

1. Thêm phương thức showDonutOrder() sau vào **MainActivity**. Trong nhiệm vụ này, sử dụng phương thức displayToast() đã tạo trước đó để hiển thị một thông báo Toast.

A black screen with text

AI-generated content may be incorrect.

Ba dòng đầu tiên là một nhận xét theo định dạng Javadoc, nó làm code dễ hiểu hơn và cũng giúp tạo tài liệu cho code của bạn. Đây là một thực tiễn tốt nhất để thêm nhận xét như vậy cho mỗi phương thức mới mà bạn tạo. Để biết thêm thông tin về cách viết nhận xét, hãy xem How to Write Doc Comments for the Javadoc Tool.

2. Thêm nhiều phương thức vào cuối MainActivity cho mỗi món tráng miệng:

A screen shot of a computer

AI-generated content may be incorrect.

3. (Tùy chọn) **Chọn Code > Reformat** Code để định dạng lại code mà bạn đã thêm vào MainActivity để phù hợp với các tiêu chuẩn và làm cho nó dễ đọc hơn.

**2.3 Thêm thuộc tính onClick**

Trong bước này bạn thêm android:onClick vào từng phần tử ImageView trong bố cục. Thuộc tính android:onClick gọi trình xử lý sự kiện nhấn cho từng phần tử.

1. Mở tệp **content\_main.xml**, và nhấp vào tab **Text** trong trình chỉnh sửa bố cục để hiển thị mã XML.

2. Thêm thuộc tính android:onClick vào ImageView donut. Khi bạn nhập, các gợi ý sẽ xuất hiện cho các trình xử lý nhấn. Chọn trình xử lý nhấn showDonutOrder. Code bây giờ trông như sau:

Dòng cuối cùng (android:onClick="showDonutOrder") gán trình xử lý nhấp chuột (showDonutOrder) cho ImageView.

3. (Tùy chọn) Chọn **Code > Reformat Code** để định dạng lại mã XML mà bạn đã thêm vào content\_main.xml để phù hợp với các tiêu chuẩn và làm cho nó dễ đọc hơn. Android Studio sẽ tự động di chuyển thuộc tính android:onClick lên vài dòng để kết hợp chúng với các thuộc tính khác bắt đầu bằng android: như lời nói đầu.

4. Thực hiện theo cùng một quy trình để thêm thuộc tính android:onClick vào các phần tử ImageView ice\_cream và froyo. Chọn các trình xử lý nhấn showDonutOrder và show FroyoOrder. Bạn có thể tùy chỉnh chọn **Code > Reformat Code** để định dạng lại code XML. Code bây giờ trông như sau:

### 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