**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

### 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

* Giới thiệu
  + Shared preferences cho phép bạn lưu một lượng nhỏ giữ liệu dưới dạng khoá/giá trị trong một tệp trên thiết bị. Để xử lý tệp tuỳ chọn cũng như đọc, ghi và quản lý dữ liệu, hãy sử dụng lớp Sharedpreferences. Khung Android tự quản lý tệp Shared preferences. Tất cả các thành phần trong ứng dụng của bạn đều truy cập được vào tệp này, nhưng các ứng dụng khác thì không thể truy cập được.
  + Dữ liệu bạn lưu vào shared preferences khác với dữ liệu trạng thái hoạt động đã lưu, cái mà bạn đã học ở chương trước.
    - Dữ liệu ở trạng thái phiên bản hoạt động đã lưu được giữ lại trên các phiên bản hoạt động trong cùng một phiên người dung.
    - Shared preference vẫn tồn tại qua các phiên người dùng. Shared preferences vẫn tông tại nếu ứng dụng của bạn dừng và khởi động lại hoặc nếu thiết bị khởi động lại.
  + Chỉ sử dụng shared preferences khi bạn cần lưu một lượng nhỏ dữ liệu dưới dạng cặp khóa/giá trị đơn giản. Để quản lý lượng dữ liệu ứng dụng cố định lớn hơn, hãy sử dụng phương thức lưu trữ như thư viện phòng hoặc cơ sở dữ liệu SQL.

Những điều bạn nên biết:

Bạn nên quen với:

* + - Tạo, xây dựng và chạy ứng dụng trong androi studio.
    - Thiết kế bố cục với các nút và văn bản.
    - Sử dụng kiểu và chủ đề.
    - Lưu và khôi phục trạng thái phiên bản hoạt động.

Bạn sẽ học được gì

Bạn sẽ học cách nào để:

* + - Xác định shared preferences là gì.
    - Tạo tệp shared preferences cho ứng dụng của bạn.
    - Lưu dữ liệu vào shared preferences, và đọc lại nó.
    - Xoá dữ liệu trong shared preferences.

Bạn sẽ làm gì

* + - Cập nhật ứng dụng để ứng dụng có thể lưu, truy xuất và đặt lại các tùy chọn chia sẻ.

Tổng quan về ứng dụng

Ứng dụng HelloSharedPrefs là một biến thể khác của ứng dụng HelloToast bạn đã tạo trong bài 1. Nó bao gồm các nút để tăng số lượng, thay đổi màu nền, và đặt lại cả số lượng và màu sắc về mặc định. Ứng dụng cũng sử chủ đề và kiểu dáng để xác định các nút.

A screenshot of a number

Description automatically generated

**Task 1: Khám phá HelloSharedPrefs**

Dự án ứng dụng khởi đầu hoàn chỉnh cho bài thực hành này có sẵn tại HelloSharedPrefs-Starter. Trong này nhiệm vụ của bạn, bạn tải dự án vào Android Studio và khám phá một số tính năng chính của ứng dụng.

* 1. **mở và chạy dự án HelloSharedPrefs**

1. Tải ứng dụng HelloSharedPrefs-Starter và giải nén tệp.

2. Mở dự án trong androi studio, và xây dựng và chạy ứng dụng. Hãy thử những điều này:

* + - * + Bấm vào nút Đếm để tăng số trong chế độ xem văn bản chính.
        + Bấm vào bất kỳ nút màu nào để thay đổi màu nền của chế độ xem văn bản chính.
        + Xoay thiết bị và lưu ý rằng cả màu nền và số lượng đều được giữ nguyên.
        + Bấm vào nút Đặt lại để đặt màu và đếm ngược về mặc định.

3. Buộc thoát khỏi ứng dụng bằng một trong các phương pháp sau:

* Trong androi studio, chọn Run > Stop ‘app’ hoặc bấm biểu tượng dừng lại  trong thanh công cụ.
* Trên thiết bị nhấn nút Gần đây (nút hình vuông ở góc dưới bên phải). Vuốt thẻ ứng dụng HelloSharedPrefs để thoát ứng dụng hoặc nhấp vào dấu X ở góc bên phải thẻ. Nếu bạn thoát ứng dụng theo cách này, hãy đợi vài giây trước khi khởi động lại để hệ thống có thể dọn dẹp.

4. Chạy lại ứng dụng. Ứng dụng khởi động lại với giao diện mặc định—số đếm là 0 và màu nền là màu xám.

**1.2 Khám phá mã hoạt động**

1. Mở Mainactivity

2. Xem xét mã và lưu ý những điều sau:

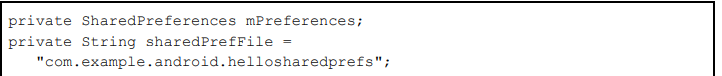
* Biến đếm (mCount) được định nghĩa là một số nguyên. Phương thức onClick của countUp () tăng giá trị này lên và cập nhật TextView chính.
* Biến màu (mClolor) cũng là số nguyên và được định nghĩa ban đầu là màu xám trong tệp tài nguyên colors.xml với tên là default\_background.
* Phương thức changeBackground () trong sự kiện onClick lấy màu nền của nút được nhấn, sau đó đặt màu đó làm nền cho phần văn bản chính.
* Both the mCount and mColor integers are saved to the instance state bundle in onSaveInstanceState (), and restored in onCreate (). The bundle keys for count and color are defined by private variables (COUNT\_KEY) and (COLOR\_KEY).

**Task 2: Lưu và khôi phục dữ liệu vào tệp SharedPreferences.**

Trong nhiệm vụ này, bạn lưu trạng thái của ứng dụng vào một tệp Shared Preferences và đọc lại dữ liệu đó khi ứng dụng được khởi động lại. Vì dữ liệu trạng thái mà bạn lưu vào Shared Preferences (số lượng hiện tại và màu sắc) cũng chính là dữ liệu mà bạn bảo toàn trong trạng thái instance, nên bạn không cần phải thực hiện việc này hai lần. Bạn có thể thay thế hoàn toàn trạng thái instance bằng trạng thái từ Shared Preferences.

* 1. **Khởi tạo các tùy chọn**

1. Thêm các biến thành viên vào lớp MainActivity để giữ tên của chia sẻ tệp tùy chọn và tham chiếu đến đối tượng SharedPreferences.



Bạn có thể đặt tên cho tệp tùy chọn chia sẻ của mình bất cứ điều gì bạn muốn, nhưng thông thường nó có cùng tên với tên gói ứng dụng của bạn.

1. Trong phương thức onCreate (), khởi tạo shared preferences. Chèn mã này trước câu lệnh if:



Phương thức getSharedPreferences () (từ Ngữ cảnh hoạt động) mở tệp tại tên tệp đã cho (sharedPrefFile) với chế độ MODE\_PRIVATE.

Lưu ý: Các phiên bản Android cũ hơn có các chế độ khác cho phép bạn tạo một ứng dụng có thể đọc được trên toàn thế giới hoặc tập tin tùy chọn chia sẻ có thể ghi trên thế giới. Các chế độ này không được dùng nữa trong API 17 và hiện tại không được khuyến khích mạnh mẽ vì lý do bảo mật. Nếu bạn cần chia sẻ dữ liệu với các ứng dụng khác, hãy cân nhắc sử dụng URI nội dung do FileProvider cung cấp.

Mã giải pháp cho MainActivity, một phần:

A screenshot of a computer program

Description automatically generated

* 1. **lưu tuỳ chọn trong onPause ()**

Việc lưu tùy chọn cũng giống như lưu trạng thái phiên bản -- cả hai thao tác đều đặt dữ liệu sang một gói đối tượng dưới dạng cặp khóa/giá trị. Tuy nhiên, đối với các tùy chọn chia sẻ, bạn lưu dữ liệu đó vào gọi lại vòng đời onPause () và bạn cần một đối tượng soạn thảo được chia sẻ (SharedPreferences.Editor) để ghi vào đối tượng tùy chọn chia sẻ.

1. Thêm phương thức vòng đời onPause () vào MainActivity.

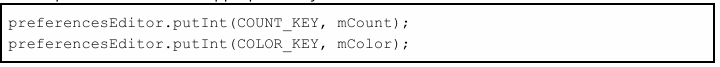
A white rectangular frame with black border

Description automatically generated

2. Trong onPause (), hãy lấy trình soạn thảo cho đối tượng SharedPreferences:

Một trình chỉnh sửa SharedPreferences là cần thiết để ghi vào đối tượng SharedPreferences. Thêm dòng này vào `onPause()` sau lệnh gọi `super.onPause()`.

1. Sử dụng phương thức `putInt()` để đưa cả hai giá trị số nguyên `mCount` và `mColor` vào SharedPreferences với các khóa phù hợp.



Lớp `SharedPreferences.Editor` bao gồm nhiều phương thức "put" cho các kiểu dữ liệu khác nhau, bao gồm `putInt()` và `putString()`.

1. Gọi `apply()` để lưu các tùy chọn preferences.



Phương thức `apply()` lưu các tùy chọn (preferences) một cách bất đồng bộ, ngoài luồng giao diện người dùng (UI thread). Trình chỉnh sửa SharedPreferences cũng có phương thức `commit()` để lưu các tùy chọn một cách đồng bộ. Tuy nhiên, phương thức `commit()` không được khuyến khích sử dụng vì nó có thể chặn các thao tác khác.

1. Xóa toàn bộ phương thức `onSaveInstanceState()`. Vì trạng thái instance của activity chứa cùng dữ liệu như SharedPreferences, bạn có thể thay thế hoàn toàn trạng thái instance.

Mã giải pháp cho phương thức `onPause()` trong `MainActivity`:

A screen shot of a computer code

Description automatically generated

* 1. **Khôi phục các tùy chọn trong onCreate()**

Tương tự như trạng thái phiên bản, ứng dụng của bạn đọc bất kỳ tùy chọn được chia sẻ đã lưu nào trong phương thức onCreate(). Một lần nữa, vì các tùy chọn được chia sẻ chứa dữ liệu giống như trạng thái phiên bản, chúng ta cũng có thể thay thế trạng thái bằng các tùy chọn ở đây. Mỗi khi onCreate() được gọi -- khi ứng dụng khởi động, khi thay đổi cấu hình -- các tùy chọn được chia sẻ được sử dụng để khôi phục trạng thái của chế độ xem.

Xác định vị trí phần của phương thức onCreate() kiểm tra xem đối số savedInstanceState có phải là null hay không và khôi phục trạng thái phiên bản:

A computer code with black text

Description automatically generated

Xóa toàn bộ khối.

Trong phương thức onCreate(), tại cùng vị trí có mã trạng thái phiên bản, hãy lấy đếm từ các tùy chọn bằng phím COUNT\_KEY và gán nó cho biến mCount.



Khi bạn đọc dữ liệu từ các tùy chọn, bạn không cần phải có tùy chọn chia sẻ biên tập viên. Sử dụng bất kỳ phương thức "get" nào trên đối tượng tùy chọn được chia sẻ (chẳng hạn như getInt() hoặc getString() để lấy dữ liệu ưu tiên.

Lưu ý rằng phương thức getInt() nhận hai đối số: một cho khóa và một cho giá trị mặc định nếu không tìm thấy khóa. Trong trường hợp này, giá trị mặc định là 0, là giá trị giống như giá trị ban đầu của mCount.

Cập nhật giá trị của TextView chính bằng số lượng mới.



Lấy màu từ tùy chọn bằng phím COLOR\_KEY và gán nó cho biến mColor.



Như trước, đối số thứ hai của getInt() là giá trị mặc định được sử dụng trong trường hợp khóa không tồn tại trong các tùy chọn được chia sẻ. Trong trường hợp này bạn chỉ có thể sử dụng lại giá trị của mColor, vừa được khởi tạo thành nền mặc định trong phương thức.

Cập nhật màu nền của chế độ xem văn bản chính.



Chạy ứng dụng. Bấm vào nút Đếm và thay đổi màu nền để cập nhật trạng thái cá thể và các ưu tiên.

Xoay thiết bị hoặc trình mô phỏng để xác minh rằng số lượng và màu sắc được lưu khi thay đổi cấu hình.

Buộc thoát khỏi ứng dụng bằng một trong các phương pháp sau:

Trong Android Studio, chọn Run > Stop 'app.'

Trên thiết bị nhấn nút Gần đây (nút hình vuông ở góc dưới bên phải). Vuốt thẻ ứng dụng HelloSharedPrefs để thoát ứng dụng hoặc nhấp vào dấu X ở góc bên phải thẻ.

Chạy lại ứng dụng. Ứng dụng khởi động lại và tải các tùy chọn, duy trì trạng thái.

Mã giải pháp cho phương thức MainActivity onCreate():

A computer screen shot of a program

Description automatically generated

**2.4 Khôi phục các tùy chọn trong onCreate()**

Nút đặt lại trong ứng dụng khởi động sẽ đặt lại cả số lượng và màu sắc cho hoạt động về giá trị mặc định. Vì tùy chọn giữ trạng thái của hoạt động nên điều quan trọng là phải xóa tùy chọn cùng một lúc.

1. Trong phương thức reset() onClick, sau khi đặt lại màu và số lượng, hãy tải trình chỉnh sửa cho đối tượng SharedPreferences:



1. Xóa tất cả các shared preferences:



1. Áp dụng các thay đổi:



Mã giải pháp cho phương thức reset():

A screenshot of a computer code

Description automatically generated

A white background with black text

Description automatically generated

**Solution code**

Dự án Android Studio: HelloSharedPrefs

**Thử thách viết mã**

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

**Thử thách:** Sửa đổi ứng dụng HelloSharedPrefs để thay vì tự động lưu trạng thái vào tệp tùy chọn, hãy thêm hoạt động thứ hai để thay đổi, đặt lại và lưu các tùy chọn đó. Thêm nút vào ứng dụng có tên Cài đặt để khởi chạy hoạt động đó. Bao gồm các nút chuyển đổi và nút xoay để sửa đổi tùy chọn cũng như các nút Lưu và Đặt lại để lưu và xóa tùy chọn.

**Bản tóm tắt**

* Lớp SharedPreferences cho phép ứng dụng lưu trữ một lượng nhỏ dữ liệu nguyên thủy dưới dạng cặp khóa-giá trị.
* Shared Preferences vẫn tồn tại trong các phiên người dùng khác nhau của cùng một ứng dụng.
* Để ghi vào các tùy chọn chia sẻ, hãy lấy đối tượng SharedPreferences.Editor.
* Sử dụng các phương thức "đặt" khác nhau trong đối tượng SharedPreferences.Editor, chẳng hạn như putInt() hoặc putString(), để đưa dữ liệu vào các tùy chọn chia sẻ bằng một khóa và một giá trị.
* Sử dụng các phương thức "get" khác nhau trong đối tượng SharedPreferences, chẳng hạn như getInt() hoặc getString() , để lấy dữ liệu ra khỏi các tùy chọn được chia sẻ bằng một khóa.
* Sử dụng phương thức clear() trong đối tượng SharedPreferences.Editor để xóa tất cả dữ liệu được lưu trữ trong Preferences.
* Sử dụng phương thức apply() trong đối tượng SharedPreferences.Editor để lưu các thay đổi vào tệp tùy chọn.

**Khái niệm liên quan**

Tài liệu khái niệm liên quan có trong 9.0: Lưu trữ dữ liệu và 9.1: Tùy chọn chia sẻ .

**Learn more**

Tài liệu dành cho nhà phát triển Android:

* Data and file storage overview
* Save key-value data
* SharedPreferences
* SharedPreferences.Editor

Tràn ngăn xếp:

* How to use SharedPreferences in Android to store, fetch and edit values
* onSavedInstanceState vs. SharedPreferences

**Bài tập về nhà**

**Xây dựng và chạy một ứng dụng**

Mở ứng dụng ScoreKeeper mà bạn đã tạo trong bài học Nguyên tắc cơ bản về Android 5.1: Bản vẽ, kiểu và chủ đề.

1. Thay thế trạng thái phiên bản đã lưu bằng các tùy chọn chung cho từng điểm số.
2. Để kiểm tra ứng dụng, hãy xoay thiết bị để đảm bảo rằng các thay đổi về cấu hình sẽ đọc các tùy chọn đã lưu và cập nhật giao diện người dùng.
3. Dừng ứng dụng và khởi động lại để đảm bảo rằng các tùy chọn được lưu.
4. Thêm nút Đặt lại để đặt lại giá trị điểm về 0 và xóa Shared Preferences.

**Answer these questions**

**Câu hỏi 1**

Bạn lưu trạng thái ứng dụng vào tùy chọn chia sẻ theo phương pháp vòng đời nào?

**Câu hỏi 2**

Bạn khôi phục trạng thái ứng dụng bằng phương pháp vòng đời nào?

**Câu hỏi 3**

Bạn có thể nghĩ ra trường hợp nào hợp lý khi có cả sở thích chung và trạng thái phiên bản không?

**Gửi ứng dụng của bạn để chấm điểm**

Hướng dẫn cho học sinh chấm điểm

Kiểm tra xem ứng dụng có các tính năng sau không:

* Ứng dụng giữ lại điểm khi xoay thiết bị.
* Ứng dụng sẽ giữ lại điểm hiện tại sau khi dừng và khởi động lại ứng dụng.
* Ứng dụng lưu điểm hiện tại vào tùy chọn được chia sẻ trong phương thức onPause().
* Ứng dụng khôi phục các tùy chọn được chia sẻ trong phương thức onCreate().
* Ứng dụng hiển thị nút Đặt lại để đặt lại điểm về 0.

Đảm bảo rằng việc triển khai phương thức xử lý khi nhấp chuột cho nút Đặt lại sẽ thực hiện những điều sau đây:

* Đặt lại cả hai biến số điểm về 0.
* Cập nhật cả chế độ xem văn bản.
* Xóa tùy chọn chia sẻ.

### Cài đặt ứng dụng

**Giới thiệu**

Ứng dụng thường bao gồm các cài đặt cho phép người dùng sửa đổi các tính năng và hành vi của ứng dụng. Ví dụ: một số ứng dụng cho phép người dùng đặt vị trí nhà riêng, đơn vị đo lường mặc định và các cài đặt khác áp dụng cho toàn bộ ứng dụng. Người dùng không truy cập cài đặt thường xuyên vì khi người dùng thay đổi cài đặt, chẳng hạn như vị trí nhà riêng, họ hiếm khi cần quay lại và thay đổi lại cài đặt đó.

Người dùng muốn điều hướng đến cài đặt ứng dụng bằng cách nhấn vào Cài đặt trong điều hướng bên cạnh, chẳng hạn như điều hướng ngăn kéo như minh họa ở phía bên trái của hình bên dưới hoặc trong menu tùy chọn trên thanh ứng dụng, hiển thị ở phía bên phải của hình bên dưới.

A screenshot of a phone

Description automatically generated

Trong hình trên:

1. Cài đặt trong điều hướng bên (ngăn điều hướng)

2. Cài đặt trong menu tùy chọn của thanh ứng dụng

Trong thực tế này, bạn thêm hoạt động cài đặt vào một ứng dụng. Người dùng sẽ có thể điều hướng đến ứng dụng cài đặt bằng cách nhấn vào Cài đặt, cài đặt này sẽ nằm trong menu tùy chọn trên thanh ứng dụng.

**Những điều bạn nên biết**

Bạn sẽ có thể:

* Tạo dự án Android Studio từ một mẫu và tạo bố cục chính.
* Chạy ứng dụng trên trình mô phỏng hoặc thiết bị được kết nối.
* Tạo và chỉnh sửa các thành phần giao diện người dùng bằng trình chỉnh sửa bố cục và mã XML.
* Trích xuất tài nguyên chuỗi và chỉnh sửa giá trị chuỗi.
* Truy cập các phần tử giao diện người dùng từ mã của bạn bằng cách sử dụng findViewById() .
* Xử lý thao tác bấm nút.
* Hiển thị thông báo Bánh mì nướng.
* Thêm Hoạt động vào ứng dụng.
* Tạo menu tùy chọn trong thanh ứng dụng.
* Thêm và chỉnh sửa các mục menu trong menu tùy chọn.
* Sử dụng phong cách và chủ đề trong dự án.
* Sử dụng SharedPreferences

**Bạn sẽ học được gì**

Bạn sẽ học cách:

* Thêm một đoạn để quản lý cài đặt.
* Tạo tệp tài nguyên XML của cài đặt với các thuộc tính của chúng.
* Tạo điều hướng đến cài đặt Hoạt động.
* Đặt giá trị mặc định của cài đặt.
* Đọc các giá trị cài đặt do người dùng thay đổi.
* Tùy chỉnh mẫu Hoạt động cài đặt.

**What you'll do**

* Tạo một ứng dụng bao gồm Cài đặt trong menu tùy chọn.
* Thêm nút chuyển đổi tùy chọn Cài đặt.
* Thêm mã để đặt giá trị mặc định cho cài đặt và truy cập giá trị cài đặt sau khi đã có thay đổi.
* Sử dụng và tùy chỉnh mẫu Hoạt động cài đặt Android Studio.

**Tổng quan về ứng dụng**

Android Studio cung cấp lối tắt để thiết lập menu tùy chọn với Cài đặt. Nếu bạn bắt đầu dự án Android Studio cho điện thoại hoặc máy tính bảng bằng mẫu Hoạt động cơ bản thì ứng dụng mới sẽ bao gồm Cài đặt như hiển thị bên dưới:

A screen shot of a phone

Description automatically generated

Mẫu cũng bao gồm một nút tác vụ nổi ở góc dưới bên phải màn hình với biểu tượng phong bì. Bạn có thể bỏ qua nút này vì bạn sẽ không sử dụng nút này.

Bạn sẽ bắt đầu bằng cách tạo một ứng dụng có tên AppWithSettings bằng mẫu Hoạt động cơ bản và bạn sẽ thêm cài đặt Hoạt động cung cấp một cài đặt công tắc bật tắt mà người dùng có thể bật hoặc tắt:

A screenshot of a phone

Description automatically generated

Bạn sẽ thêm mã để đọc cài đặt và thực hiện hành động dựa trên giá trị của nó. Để đơn giản, hành động sẽ là hiển thị thông báo Toast với giá trị cài đặt.

Trong nhiệm vụ thứ hai, bạn sẽ thêm mẫu Hoạt động cài đặt tiêu chuẩn do Android Studio cung cấp vào ứng dụng DroidCafeOptionsUp mà bạn đã tạo trong bài học trước.

Mẫu Hoạt động cài đặt được điền sẵn các cài đặt mà bạn có thể tùy chỉnh cho một ứng dụng và cung cấp bố cục khác cho điện thoại và máy tính bảng:

* Điện thoại : Màn hình Cài đặt chính có liên kết tiêu đề cho từng nhóm cài đặt, chẳng hạn như Chung cho cài đặt chung, như hiển thị bên dưới.

A screenshot of a phone

Description automatically generated

* Máy tính bảng : Bố cục màn hình chính/chi tiết có liên kết tiêu đề cho từng nhóm ở phía bên trái (chính) và nhóm cài đặt ở phía bên phải (chi tiết), như minh họa trong hình bên dưới.

A screenshot of a computer

Description automatically generated

Để tùy chỉnh mẫu, bạn sẽ thay đổi tiêu đề, đặt tiêu đề, mô tả cài đặt và giá trị cho cài đặt.

Ứng dụng DroidCafeOptionsUp đã được tạo trong bài học trước từ mẫu Hoạt động cơ bản, cung cấp menu tùy chọn trong thanh ứng dụng để đặt tùy chọn Cài đặt. Bạn sẽ tùy chỉnh mẫu Hoạt động cài đặt được cung cấp bằng cách thay đổi tiêu đề, mô tả, giá trị và giá trị mặc định của một cài đặt. Bạn sẽ thêm mã để đọc giá trị của cài đặt sau khi người dùng thay đổi và hiển thị giá trị đó.

**Task 1: Thêm cài đặt chuyển đổi vào ứng dụng**

Trong nhiệm vụ này, bạn thực hiện như sau:

* Tạo một dự án mới dựa trên mẫu Hoạt động cơ bản, cung cấp menu tùy chọn.
* Thêm công tắc bật tắt ( SwitchPreference ) với các thuộc tính trong tệp XML tùy chọn.
* Thêm hoạt động cho cài đặt và một đoạn cho cài đặt cụ thể. Để duy trì khả năng tương thích với AppCompatActivity , bạn sử dụng PreferenceFragmentCompat thay vì PreferenceFragment . Bạn cũng thêm thư viện android.support.v7.preference.
* Kết nối mục Cài đặt trong menu tùy chọn với hoạt động cài đặt.
  1. **Tạo dự án và thêm thư mục xml và tệp tài nguyên**

1. Trong Android Studio, tạo một dự án mới với các thông số sau:

|  |  |
| --- | --- |
| **Thuộc tính** | **Giá trị** |
| Tên ứng dụng | android.example.com (hoặc tên miền của riêng bạn) |
| Tên công ty | android.example.com (hoặc tên miền của riêng bạn) |
| Vị trí dự án | Đường dẫn đến thư mục dự án của bạn |
| SDK tối thiểu cho điện thoại và máy tính bảng | API15:Android4.0.3 IceCreamSandwich |
| Bản mẫu | Hoạt động cơ bản |
| Tên hoạt động | Hoạt động chính |
| Tên bố cục | hoạt động chính |
| Tiêu đề | Hoạt động chính |

2. Chạy ứng dụng và nhấn vào biểu tượng tràn trên thanh ứng dụng để xem menu tùy chọn, như minh họa trong hình bên dưới. Mục duy nhất trong menu tùy chọn là Cài đặt.

Screens screenshot of a phone

Description automatically generated

3. Bạn cần tạo một thư mục tài nguyên mới để chứa tệp XML chứa các cài đặt. Chọn thư mục res trong khung Project > Android và chọn File > New > Android Resource Directory . Hộp thoại Thư mục tài nguyên mới xuất hiện.

4. Trong menu thả xuống Loại tài nguyên, chọn xml . Tên thư mục tự động thay đổi thành xml . Bấm vào đồng ý .

5. Thư mục xml xuất hiện trong khung Project > Android bên trong thư mục res. Chọn xml và chọn Tệp > Mới > Tệp tài nguyên XML (hoặc nhấp chuột phải vào xml và chọn Mới > Tệp tài nguyên XML).

6. Nhập tên của tệp XML, tùy chọn, vào trường Tên tệp và nhấp vào OK. Tệp preferences.xml xuất hiện bên trong thư mục xml và trình chỉnh sửa bố cục xuất hiện, như minh họa trong hình bên dưới.

A screenshot of a computer

Description automatically generated

Trong hình trên:

1. Tệp preferences.xml bên trong thư mục xml.

2. Trình chỉnh sửa bố cục hiển thị nội dung preferences.xml.

* 1. **Thêm tùy chọn và thuộc tính XML cho cài đặt**
     + - 1. Kéo SwitchPreference từ ngăn Bảng màu ở bên trái lên trên cùng của bố cục, như minh họa trong hình bên dưới.

A screenshot of a computer

Description automatically generated

* + - * 1. Thay đổi các giá trị trong khung Thuộc tính ở bên phải của trình chỉnh sửa bố cục như sau và như minh họa trong hình bên dưới:

giá trị mặc định: true

phím: example\_switch

tiêu đề: Tùy chọn cài đặt

tóm tắt: Bật hoặc tắt tùy chọn này

A screenshot of a computer

Description automatically generated

* + - * 1. Nhấp vào tab Văn bản ở cuối trình chỉnh sửa bố cục để xem mã XML:

A close-up of a computer code

Description automatically generated A close-up of a white box

Description automatically generated

* + - * 1. Trích xuất tài nguyên chuỗi cho các giá trị thuộc tính android:title và android:tóm thành @string/switch\_title và @string/switch\_summary .

Các thuộc tính XML cho một tùy chọn là:

* android:defaultValue : Giá trị mặc định của cài đặt khi ứng dụng khởi động lần đầu tiên.
* android:title : Tiêu đề của cài đặt. Đối với SwitchPreference , tiêu đề xuất hiện ở bên trái của nút chuyển đổi.
* android:key : Phím dùng để lưu trữ giá trị cài đặt. Mỗi cài đặt có một cặp khóa-giá trị tương ứng mà hệ thống sử dụng để lưu cài đặt trong tệp SharedPreferences mặc định cho cài đặt ứng dụng của bạn.
* android:summary : Tóm tắt văn bản xuất hiện bên dưới cài đặt.
  1. **Sử dụng SwitchPreferenceCompat**

Để sử dụng phiên bản PreferenceFragmentCompat của PreferenceFragment , bạn cũng phải sử dụng phiên bản android.support.v7 của SwitchPreference ( SwitchPreferenceCompat ).

1. Trong ngăn Dự án > Android, hãy mở tệp build.gradle (Mô-đun: ứng dụng) trong thư mục Tập lệnh Gradle và thêm phần sau vào phần phụ thuộc:



Câu lệnh hiển thị ở trên bổ sung thư viện android.support.v7.preference để sử dụng phiên bản PreferenceFragmentCompat của PreferenceFragmet

2. Trong tệp preferences.xml trong thư mục xml, hãy thay đổi SwitchPreference trong mã thành android.support.v7.preference.SwitchPreferenceCompat

A close-up of a code

Description automatically generated

Dòng SwitchPreferenceCompat ở trên có thể hiển thị biểu tượng bóng đèn màu vàng kèm theo cảnh báo, nhưng hiện tại bạn có thể bỏ qua nó.

3. Mở tệp styles.xml trong thư mục giá trị và thêm mục preferencesTheme sau vào khai báo AppTheme:



Để sử dụng PreferenceFragmentCompat, bạn cũng phải khai báo preferencesTheme với kiểu PreferenceThemeOverlay cho chủ đề ứng dụng.

* 1. **Thêm Hoạt động cho cài đặt**

Để tạo Hoạt động cài đặt cung cấp giao diện người dùng cho cài đặt, hãy thêm Hoạt động trống vào ứng dụng. Thực hiện theo các bước sau:

1. Chọn ứng dụng ở đầu ngăn Dự án > Android và chọn Mới > Hoạt động > Hoạt động trống .

2. Đặt tên cho Hoạt động Cài đặt Hoạt động. Bỏ chọn tùy chọn Tạo tệp bố cục (bạn không cần tùy chọn này) và bỏ chọn tùy chọn Hoạt động của trình khởi chạy.

3. Chọn tùy chọn Tương thích ngược (AppCompat). Tên gói phải được đặt thành com.example.android. tên dự án .

4. Nhấp vào Hoàn tất.

**1.5 Thêm Đoạn cho cài đặt cụ thể**

Phân đoạn giống như một phần mô-đun của Hoạt động —nó có vòng đời riêng và nhận các sự kiện đầu vào riêng và bạn có thể thêm hoặc xóa Phân đoạn trong khi Hoạt động đang chạy. Bạn sử dụng lớp con Fragment chuyên dụng để hiển thị danh sách cài đặt. Cách tốt nhất là sử dụng Hoạt động thông thường lưu trữ PreferenceFragment hiển thị cài đặt ứng dụng. PreferenceFragmet cung cấp kiến ​​trúc linh hoạt hơn cho ứng dụng của bạn so với việc sử dụng Hoạt động cho tùy chọn.

Bạn sẽ sử dụng PreferenceFragmentCompat thay vì PreferenceFragmet để duy trì khả năng tương thích với AppCompatActivity .

Trong bước này, bạn sẽ thêm một Đoạn trống cho một nhóm cài đặt tương tự (không có bố cục, phương thức gốc hoặc lệnh gọi lại giao diện) vào ứng dụng và mở rộng PreferenceFragmetCompat .

Thực hiện theo các bước sau:

1. Chọn lại ứng dụng và chọn Mới > Đoạn > Đoạn (Trống) .

2. Đặt tên cho đoạn SettingFragment . Bỏ chọn Tạo bố cục XML? tùy chọn (bạn không cần).

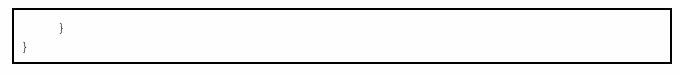
3. Bỏ chọn các tùy chọn để bao gồm các phương thức xuất xưởng của mảnh và lệnh gọi lại giao diện.

4. Bộ nguồn mục tiêu phải được đặt thành main .

5. Nhấp vào Hoàn tất. Kết quả là định nghĩa lớp sau trong SettingFragment :

A screenshot of a computer program

Description automatically generated



6. Chỉnh sửa định nghĩa lớp của SettingFragment để mở rộng PreferenceFragmetCompat :

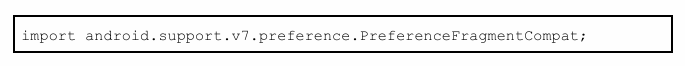


Khi bạn thay đổi định nghĩa lớp sao cho phù hợp với định nghĩa hiển thị ở trên, bóng đèn màu đỏ sẽ xuất hiện ở lề trái. Nhấp vào bóng đèn màu đỏ và chọn Phương pháp triển khai, sau đó chọn onCreatePferencs. Android Studio tạo sơ khai onCreatePreferencs() sau:

A close-up of a sign

Description automatically generated

Để mở rộng Fragment , Android Studio bổ sung câu lệnh nhập sau:



7. Xóa toàn bộ phương thức onCreateView() trong đoạn.

Lý do về cơ bản bạn thay thế onCreateView() bằng onCreatePreferencs() là vì bạn sẽ thêm phần Cài đặt này vào phần Cài đặt hiện có để hiển thị các tùy chọn thay vì hiển thị một màn hình Phân đoạn riêng biệt. Việc thêm nó vào Hoạt động hiện có giúp bạn dễ dàng thêm hoặc xóa Phân đoạn trong khi Hoạt động đang chạy. Đoạn tùy chọn được bắt nguồn từ PreferenceScreen bằng cách sử dụng rootKey .

Bạn cũng có thể xóa hàm tạo trống khỏi SettingFragment một cách an toàn, vì Fragment không tự hiển thị:

A close-up of a sign

Description automatically generated

8. Bạn cần liên kết với Fragment này tài nguyên cài đặt preferences.xml mà bạn đã tạo ở bước trước. Thêm vào lệnh gọi onCreatePreferencs() mới được tạo, gọi tới setPreferencesFromResource() chuyển id của tệp XML ( R.xml.preferences) và rootKey để xác định gốc tùy chọn trong PreferenceScreen:



Phương thức onCreatePreferences() bây giờ sẽ trông như thế này:

A white rectangular sign with black text

Description automatically generated

**1.6 Hiển thị đoạn trong Cài đặt Hoạt động**

Để hiển thị Đoạn trong Cài đặtHoạt động, hãy làm theo các bước sau:

Mở Cài đặtHoạt động.

Thêm đoạn mã sau vào cuối phương thức onCreate() để Fragment được hiển thị làm nội dung chính:

A black and white rectangular object with text

Description automatically generated

Đoạn mã trên sử dụng mẫu điển hình để thêm một đoạn vào một hoạt động sao cho đoạn đó xuất hiện dưới dạng nội dung chính của hoạt động:

* Sử dụng getFragmentManager() nếu lớp mở rộng Hoạt động và Đoạn mở rộng PreferenceFragment.
* Sử dụng getSupportFragmentManager() nếu lớp mở rộng AppCompatActivity và tới đoạn mở rộng PreferenceFramentCompat.

Toàn bộ phương thức onCreate() trong Cài đặtActivity bây giờ sẽ trông giống như sau:

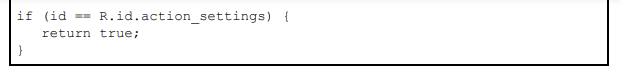
A screen shot of a computer code

Description automatically generated

**1.7 Kết nối mục menu Cài đặt với Cài đặt Hoạt động**

Sử dụng Ý định để khởi chạy Cài đặtHoạt động từ MainActivity khi người dùng chọn Cài đặt từ menu tùy chọn.

Mở MainActivity và tìm khối if trong phương thức onOptionsItemSelected() để xử lý thao tác nhấn vào Cài đặt trong menu tùy chọn:



Thêm Ý định vào khối if để khởi chạy Cài đặtHoạt động:

A screen shot of a computer program

Description automatically generated

Để thêm nút điều hướng Lên trên thanh ứng dụng vào Cài đặtHoạt động, bạn cần chỉnh sửa nút điều hướng đó khai báo trong tệp AndroidManifest.xml để xác định cha mẹ Cài đặt hoạt động là hoạt động chính. Mở AndroidManifest.xml và tìm phần khai báo Cài đặt hoạt động:



Thay đổi khai báo như sau:

A screen shot of a computer program

Description automatically generated

4. Chạy ứng dụng. Nhấn vào biểu tượng tràn cho menu tùy chọn, như minh họa ở phía bên trái của hình bên dưới. Nhấn vào Cài đặt để xem hoạt động cài đặt, như minh họa ở giữa hình dưới. Nhấn vào nút Lên trên thanh ứng dụng của hoạt động cài đặt, hiển thị ở bên phải của hình bên dưới, để quay lại hoạt động chính

A screenshot of a phone

Description automatically generated

**1.8 Lưu các giá trị mặc định trong tùy chọn chia sẻ**

Mặc dù giá trị mặc định cho cài đặt công tắc bật tắt đã được đặt trong thuộc tính android:defaultValue (ở Bước 1.2 của tác vụ này), ứng dụng phải lưu giá trị mặc định trong tệp SharedPreferences cho từng cài đặt khi người dùng mở ứng dụng lần đầu tiên. Thực hiện theo các bước sau để thiết lập giá trị mặc định cho công tắc bật tắt:

1. Mở MainActivity

2. Thêm phần sau vào cuối phương thức onCreate() sau FloatActionButton:

A black and white rectangular sign with black text

Description automatically generated

Đoạn mã trên đảm bảo rằng các cài đặt được khởi tạo đúng với giá trị mặc định của chúng. Phương thức PreferenceManager.setDefaultValues() nhận ba đối số:

* **Context của ứng dụng**, chẳng hạn như this.
* **ID tài nguyên** (preferences) cho tệp tài nguyên XML chứa một hoặc nhiều cài đặt.
* **Một giá trị boolean** chỉ ra liệu giá trị mặc định có được thiết lập lại nhiều lần hay không. Khi là false, hệ thống chỉ thiết lập giá trị mặc định nếu phương thức này chưa bao giờ được gọi. Miễn là bạn thiết lập tham số thứ ba là false, bạn có thể gọi phương thức này một cách an toàn mỗi khi MainActivity khởi động mà không ghi đè lên giá trị cài đặt đã lưu của người dùng. Tuy nhiên, nếu bạn thiết lập tham số này là true, phương thức sẽ ghi đè mọi giá trị trước đó bằng các giá trị mặc định.

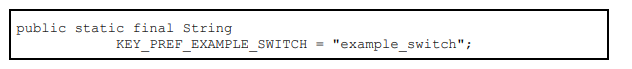
**1.9 Đọc giá trị cài đặt đã thay đổi từ Shared Preferences**

Khi ứng dụng khởi động, phương thức onCreate() của MainActivity có thể đọc các giá trị cài đặt đã thay đổi và sử dụng các giá trị đó thay vì giá trị mặc định.

Mỗi cài đặt được xác định bằng một cặp khóa-giá trị. Hệ thống Android sử dụng cặp khóa-giá trị này khi lưu hoặc truy xuất cài đặt từ tệp SharedPreferences của ứng dụng. Khi người dùng thay đổi một cài đặt, hệ thống sẽ cập nhật giá trị tương ứng trong tệp SharedPreferences.Để sử dụng giá trị của cài đặt, ứng dụng có thể sử dụng khóa để lấy cài đặt từ tệp SharedPreferences bằng phương thức PreferenceManager.getDefaultSharedPreferences().

Thực hiện các bước sau để thêm đoạn mã đó:

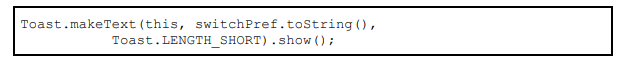
1. Mở SettingsActivity và tạo một biến String tĩnh để lưu trữ khóa cho giá trị:



1. Mở MainActivity và thêm đoạn sau vào cuối phương thức onCreate():

A computer code with black text

Description automatically generated



1. Chạy ứng dụng và nhấn vào **Settings** để mở **SettingsActivity**.
2. Nhấn vào cài đặt để thay đổi công tắc từ **bật** sang **tắt**, như trong hình bên trái bên dưới.
3. Nhấn nút **Up** trong **SettingsActivity** để quay lại **MainActivity**. Thông báo **Toast** sẽ xuất hiện trong **MainActivity** với giá trị của cài đặt, như trong hình bên phải bên dưới.
4. Lặp lại các bước này để thấy thông báo **Toast** thay đổi khi bạn thay đổi cài đặt.

Screens screenshot of a phone

Description automatically generated

Đoạn mã trên sử dụng các thành phần sau:

* android.support.v7.preference.PreferenceManager.getDefaultSharedPreferences(this) để lấy cài đặt dưới dạng đối tượng SharedPreferences (sharedPref).
* getBoolean() để lấy giá trị Boolean của cài đặt sử dụng khóa KEY\_PREF\_EXAMPLE\_SWITCH (được định nghĩa trong SettingsActivity) và gán nó vào switchPref. Nếu không có giá trị cho khóa này, phương thức getBoolean() sẽ đặt switchPref thành false. Đối với các loại giá trị khác như chuỗi (String), số nguyên (int), hoặc số thực (float), bạn có thể sử dụng các phương thức getString(), getInt(), hoặc getFloat() tương ứng.
* Toast.makeText() và show() để hiển thị giá trị của cài đặt switchPref.

Mỗi khi **MainActivity** khởi động hoặc khởi động lại, phương thức onCreate() sẽ đọc giá trị cài đặt để sử dụng trong ứng dụng. Phương thức Toast.makeText() sẽ được thay thế bằng một phương thức khởi tạo cài đặt.

Bây giờ, bạn đã có một **Settings Activity** hoạt động trong ứng dụng của mình.

**TASK 1 mã giải pháp**

Dự án Android Studio: AppWithSettings

TASK 2: : Sử dụng mẫu Hoạt động cài đặt

Nếu bạn cần tạo nhiều màn hình con trong phần cài đặt và muốn tận dụng màn hình có kích thước lớn trên máy tính bảng, đồng thời vẫn duy trì khả năng tương thích với các phiên bản Android cũ hơn dành cho máy tính bảng, Android Studio cung cấp một lối tắt: mẫu **Settings Activity**.

Trong nhiệm vụ trước, bạn đã học cách sử dụng một **Activity cài đặt trống** và một **Fragment trống** để thêm một cài đặt vào ứng dụng. **Nhiệm vụ 2** sẽ hướng dẫn bạn cách sử dụng mẫu **Settings Activity** có sẵn trong Android Studio để:

* Chia nhiều cài đặt thành các nhóm.
* Tùy chỉnh các cài đặt và giá trị của chúng.
* Hiển thị màn hình Cài đặt chính với một liên kết tiêu đề cho từng nhóm cài đặt, chẳng hạn như **General** cho cài đặt chung, như minh họa trong hình dưới đây.

Screens screenshot of a phone

Description automatically generated

* Hiển thị bố cục màn hình **chính/chi tiết** với một liên kết tiêu đề cho từng nhóm cài đặt ở bên trái (phần **chính**), và nhóm cài đặt tương ứng ở bên phải (phần **chi tiết**), như minh họa trong hình dưới đây.

A screenshot of a computer

Description automatically generated

Trong bài thực hành trước, bạn đã tạo một ứng dụng có tên **DroidCafeOptionsUp** bằng cách sử dụng mẫu **Basic Activity**, mẫu này cung cấp một **menu tùy chọn** trong thanh ứng dụng, như minh họa dưới đây.

A screenshot of a phone

Description automatically generated

Chú thích cho hình minh họa ở trên:

1. Thanh ứng dụng (App bar)
2. Biểu tượng hành động của menu tùy chọn (Options menu action icons)
3. Nút tràn (Overflow button)
4. Menu tràn tùy chọn (Options overflow menu)

**2.1 Khám phá mẫu Settings Activity**

Để thêm mẫu **Settings Activity** vào một dự án ứng dụng trong Android Studio, hãy làm theo các bước sau:

1. Sao chép thư mục dự án **DroidCafeOptionsUp** và đổi tên thành **DroidCafeWithSettings**. Chạy ứng dụng để đảm bảo nó hoạt động đúng.
2. Trong **Project** > **Android**, chọn **app**, sau đó chọn **New** > **Activity** > **Settings Activity**.
3. Trong hộp thoại xuất hiện, chấp nhận tên Activity mặc định (**SettingsActivity** là tên được đề xuất) và tiêu đề (**Settings**).
4. Nhấp vào **dấu ba chấm** ở cuối menu **Hierarchical Parent**, sau đó trong hộp thoại **Select Activity**, chọn tab **Project** (xem hình minh họa bên dưới).
5. Mở rộng thư mục **DroidCafeWithSettings** > **app** > **src** > **main** > **java** > **com.example.android.droidcafeinput**, chọn **MainActivity** làm Activity cha, như minh họa trong hình. Nhấn **OK**.

A screenshot of a program

Description automatically generated

Bạn chọn **MainActivity** làm **Activity cha** để nút **Up** trên thanh ứng dụng trong **SettingsActivity** đưa người dùng quay lại **MainActivity**. Việc chọn **Activity cha** sẽ tự động cập nhật tệp **AndroidManifest.xml** để hỗ trợ điều hướng bằng nút **Up**.

1. Nhấp vào **Finish**.
2. Trong **Project** > **Android**, mở rộng thư mục **app** > **res** > **xml** để xem các tệp XML được tạo bởi mẫu **Settings Activity**.

A screenshot of a computer

Description automatically generated

Bạn có thể mở và tùy chỉnh các tệp XML để thêm hoặc chỉnh sửa các cài đặt theo ý muốn:

* **pref\_data\_sync.xml**: Bố cục **PreferenceScreen** cho cài đặt "Data & sync".
* **pref\_general.xml**: Bố cục **PreferenceScreen** cho cài đặt "General".
* **pref\_headers.xml**: Bố cục tiêu đề cho màn hình chính của **Settings**.
* **pref\_notification.xml**: Bố cục **PreferenceScreen** cho cài đặt "Notifications".

Các tệp XML trên sử dụng nhiều **lớp con của Preference** thay vì **View**, trong đó một số lớp con cung cấp **container** cho bố cục chứa nhiều cài đặt khác nhau. Ví dụ, **PreferenceScreen** đại diện cho một **Preference** cấp cao nhất, đóng vai trò là **gốc của một hệ thống Preference**. Các tệp XML trên sử dụng **PreferenceScreen** ở đầu mỗi màn hình cài đặt. Các lớp con khác của **Preference** cung cấp giao diện người dùng phù hợp để người dùng có thể thay đổi cài đặt theo ý muốn. Ví dụ:

* **CheckBoxPreference**: Một hộp kiểm cho phép bật hoặc tắt một cài đặt.
* **ListPreference**: Một hộp thoại chứa danh sách các nút radio để chọn một tùy chọn.
* **SwitchPreference**: Một tùy chọn hai trạng thái có thể chuyển đổi (chẳng hạn như bật/tắt hoặc true/false).
* **EditTextPreference**: Một hộp thoại chứa **EditText** để nhập văn bản.
* **RingtonePreference**: Một hộp thoại hiển thị danh sách nhạc chuông có sẵn trên thiết bị.

A screenshot of a computer program

Description automatically generated

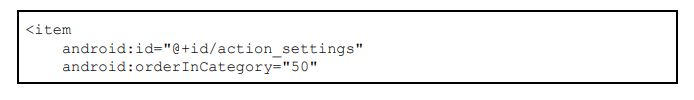
Mẫu **Settings Activity** cũng tạo ra các tệp sau:

* **SettingsActivity** trong thư mục java/com.example.android.projectname, bạn có thể sử dụng trực tiếp. Đây là **Activity** hiển thị cài đặt. **SettingsActivity** kế thừa từ **AppCompatPreferenceActivity** để duy trì khả năng tương thích với các phiên bản Android cũ hơn.
* **AppCompatPreferenceActivity** trong thư mục java/com.example.android.projectname, cũng có thể được sử dụng nguyên vẹn. Đây là một **Activity trợ giúp**, hỗ trợ **SettingsActivity** duy trì tính tương thích ngược với các phiên bản Android trước đó.

### ****2.2 Thêm mục menu Cài đặt và kết nối với Activity****

Như bạn đã học trong một bài thực hành khác, bạn có thể chỉnh sửa tệp **menu\_main.xml** để thêm hoặc xóa các mục menu trong **Options Menu**.

1. Trong **Project > Android**, mở rộng thư mục **res**, sau đó mở tệp **menu\_main.xml.**  Nhấp vào tab **Text** để hiển thị mã XML.
2. Thêm một mục menu mới có tên **Settings** với resource id là **action\_settings**:



### Room, LiveData và ViewModel

### Room, LiveData và ViewModel