**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 trữ 1 lượng nhỏ primitive data theo kiểu cặp key/value trong tệp trên thiết bị. Để lấy 1 tệp preference và để đọc, viết và quản lý preference data, sử dụng lớp SharedPreferences. Android framework quản lý tệp sharedPreferences của nó. Tệp này cho phép truy cập bởi mọi thành phần trong ứng dụng của bạn, nhưng mà nó không cho ứng dụng khác truy cập vào.

Dữ liệu của bạn lưu ở Shared preferences khác với dữ liệu trong saved activity state, chúng tôi đã đề cập đến saved activity state ở chương trước.

- Dữ liệu trong saved activity instance state được dữ lại trong cùng phiên làm việc của người dùng.

- Shared preferences tồn tại trong suốt phiên làm việc của người dùng. Shared preferences được dữ lại ngay cả khi nếu ứng dụng của bạn dừng và khởi động lại hoặc nếu thiết bị khởi động lại

Sử dụng Shared preferences chỉ khi bạn cần lưu 1 lượng nhỏ dữ liệu như là cặp key/value đơn giản. Để quản lý lượng lớn dữ liệu lưu trữ lâu dài trong ứng dụng, sử dụng phương pháp lưu trữ khác như là Room library hoặc SQL Database.

**CÓ THỂ BẠN ĐÃ BIẾT**

Bạn nên thành thạo về:

- Tạo, xây và chạy ứng dụng trên android studio

- Thiết kế giao diện với buttons và text views

- Lưu và khôi phục activity instance state.

Bạn sẽ học về?

Bạn sẽ được học cách:

Xác định Shared preferences là gì

Tạo tệp Shared preferences cho ứng dụng

Lưu dữ liệu vào Shared preferences và đọc lại dữ liệu đó

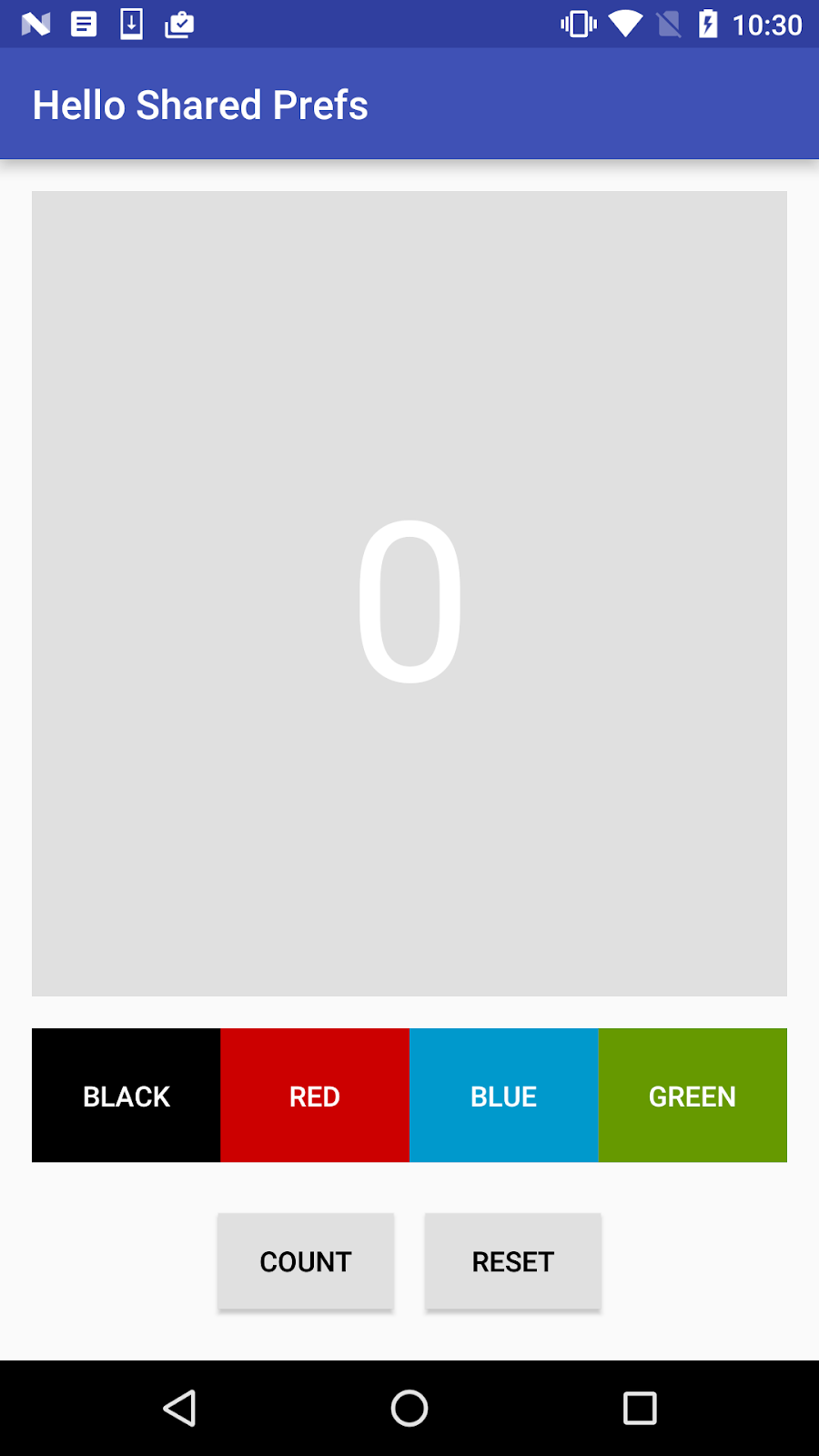
Xóa dữ liệu trong Shared preferences.

Bạn sẽ làm gì?

Nâng cấp ứng dụng để nó có thể lưu, truy xuất hoặc đặt lại Shared preferences.

Tổng quan về ứng dụng

Ứng dụng HelloSharedPrefs là 1 biến thể khác của ứng dụng HelloToast bạn đã tạo ở bài 1. nó có những buttons để tăng số count, thay đổi màu nền và khởi tạo lại count và màu về mặt định. Ứng dụng cũng sử dụng themes và style để định nghĩa buttons.



Bắt đầu với ứng dụng khở tạo và thêm shared preferences vào main activity code. Bạn cũng thêm cả button “Reset” để thiết lập count và màu nền về mặc định và xóa hết tệp preferences.

Nhiệm vụ 1: khám phá HelloSharedPrefs

Project ứng dụng starter cho bài thực hành này có sẵn tại HelloSharedPrefs-Srarter. Trong nhiệm vụ này, bạn sẽ tải project into Android Studio và khám phá và đặc điểm của key trong ứng dụng.

* 1. Mở và chạy project HelloSharedPrefs-Srarter

1. Tải ứng dụng HelloSharedPrefs-Srarter và giải nén tệp.
2. Mở project trên Android Studio sau đó xây và chạy thử ứng dụng. Thử những điều sau:

- Bấm vào nút “count” thì số trong text view chính sẽ tăng lên.

- Bấm bất kì nút màu để thay đổi màu nền và text view chính cũng sẽ thay đổi

- Xoay thiết bị và ghi chú rằng màu nền và count vẫn được giữ nguyên.

- Bấm button “Reset” để cài lại màu và count về mặc định.

1. Buộc dừng ứng dụng sử dựng 1 trong những phương pháp sau:

- Trong android, chọn Run > Stop ‘app’ hoặc ấn vào biểu tượng Stop  ở thanh công cụ.

- Trên thiết bị, ấn vào nút quay lại (nút hình vuông ở góc dưới bên phải, vuốt thẻ ứng dụng HelloSharedPrefs để thoát hoặc là ấn vào X trên góc phải của thẻ. Nếu bạn thoát ứng dụng theo cách này, hãy đợi vài giây để hệ thống dọn dẹp trước khi khởi động lại

1. Chạy lại ứng dụng:

Khi bạn khởi động lại ứng dụng, giao diện mặc điịnh sẽ hiển thị - count là 0 và màu nền là màu xám.

* 1. Khám phá Activity code

1. Mở MainActivity
2. Xem xét mã nguồn và lưu ý những điểm sau:

- Biến đếm (mCount) đã được định nghĩa ở kiểu integer. Phương thức onClick countUp() được gọi khi nhấn vào nút Count, giúp tăng giá trị này và cập nhật TextView chính.

- Màu nền (mColor) cũng là kiểu integer, ban đầu được đặt là màu xám từ tệp colors.xml với tên default\_background.

- Phương thức changeBackground() được gọi khi nhấn vào một trong các màu sau đó sẽ đặt text view chính theo màu đó.

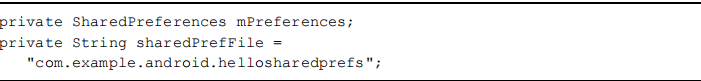
- Phương thức onSaveinstanceState() và khôi phục trong phương thức onCreate(). Các khóa(key) bundle cho count và màu được định nghĩa bởi các biến private COUNT\_KEY và COLOR\_KEY.

Nhiệm vụ 2: Lưu và khôi phục lại dữ liệu của tệp shared preferences

Trong nhiệm vụ này, bạn sẽ lưu lại trạng thái của ứng dụng ở tệp shared preferences và đọc dữ liệu lại mỗi khi ứng dụng khởi động lại. Bởi vì trạng thái dữ liệu mà bạn lưu ở shared preferences ( nơi lưu count và color gần đây) trùng với dữ liệu mà bạn lưu trữ trong intance state nên không cần phải lưu 2 lần mà thay thế hoàn toàn intance state bằng shared preferences

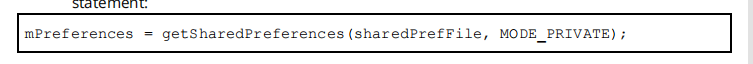
* 1. Bắt đầu shared preferences:

1. Thêm biến vào lớp MainActivity để lưu tên của tệp shared preferences và tham chiếu đến đối tượng shared preferences.



Bạn có thể đặt tên tệp shared preferences của bạn tùy thích nhưng theo quy ước thì phải trùng tên với tên package của ứng dụng.

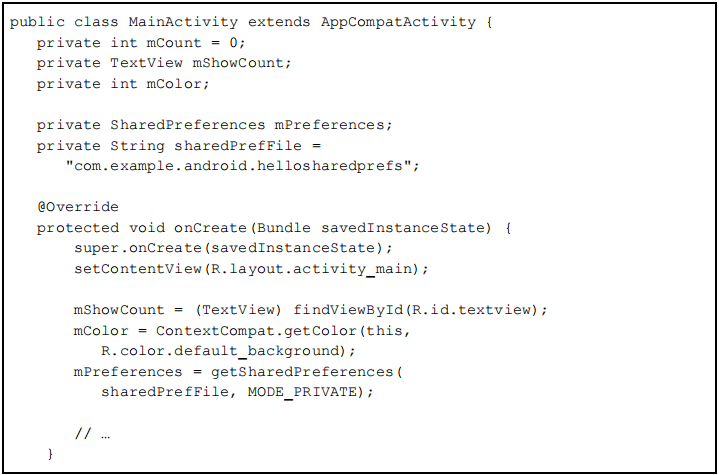
1. Trong phương thức onCreate(), khởi tạo đối tượng shared preferences bằng cách thêm dòng mã sau trước câu lệnh if:

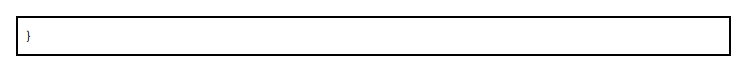


Phương thức getSharedPreferences ( từ Context của activity) mở tệp tại tên tệp được chỉ định (sharePreFile) với chế độ MODE\_PRIVATE.

LƯU Ý: các phiên bản android cũ hơn đã có các chế độ khác mà cho phép bạn tạo tệp preferences có thể đọc.ghi bởi các ứng dụng khác. Các chế độ này đã bị loại bỏ từ API 17 và hiện nay không được khuyến nghị vì lý do bảo mật. Nếu bạn cần chia sẻ dữ liệu với các ứng dụng khác, hãy cân nhắc sử dụng content URI do FileProvider cung cấp.

Đoạn mã giải quyết vấn đề cho MainActivit, chỉ có 1 phần được hiển thị thôi:





* 1. Lưu preferences trong onPause()

Lưu preferences khá giống với lưu instance state - cả 2 thao tác đều đặt dữ liệu dưới dạng cặp key/value vào 1 đối tượng Bundle. Tuy nhiên, bạn lưu dữ liệu đó trong callback vòng đời onPause() và cần 1 đối tượng editor (SharedPreferences.Editor) để ghi vào đối tượng Shared Preferences.

1. Thêm phương thức vòng đời onPause vào MainActivity



1. Trong onPause(), lấy 1 đối tượng editor cho đối tượng SharedPreferences:



Một editor của preferences được chấp thuận để viết vào đối tượng shared preferences. Thêm dòng này vào onPause() sau khi gọi đến super.onPause().

1. Sử dụng phương thức putInt() để đẩy kiểu integer của mCount và mColor đến shared preferences với các khóa tương ứng:



Lớp SharedPreferences.Editors bao gồm đa phương thức “đẩy” cho những kiểu dữ liệu khác nhau bao gồm puInt() và puString().

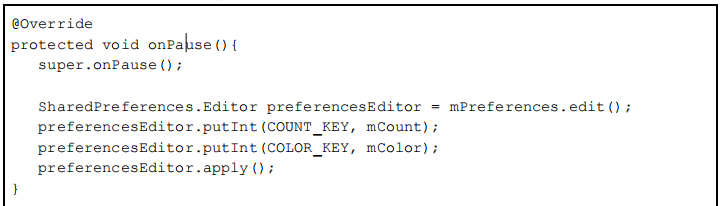
1. Gọi apply() để lưu preferences:



Phương thức apply() lưu shared preferences 1 cách bất đồng bộ, ngoài luồng giao diện người dùng(UI thread). Đối tượng Shared preferences, editor cũng có phương thức commit() để lưu preferences 1 cách đồng bộ. Phương thức commit() không được khuyến khích vì nó có thể chặn các thao tác khác.

1. Xóa toàn bộ phương thức onSaveInstanceState(). Bởi vì ví activity instance state bao gồm dữ liệu giống như shared preferences, bạn có thể thay thế hoàn toàn instance state bằng shared preferences.

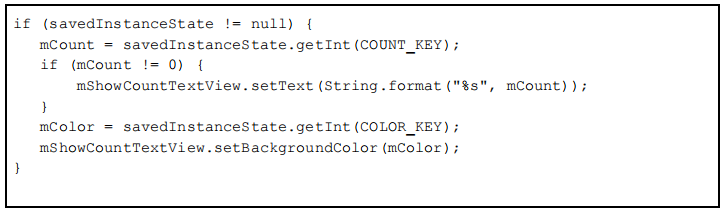
Code giải pháp cho phướng thức onPause(): trong MainActivity



* 1. Phục hồi preferences trong onCreate()

Cũng giống như instance state, phần mềm của bạn đọc tất cả shared preferences trong phương thức onCreate(). 1 lần nữa, bởi vì shared preferences bao gồm dữ liệu giống như intance state, chúng ta có thể thay thế intance state bằng shared preferences ở đây. Mỗi lần onCreate() được gọi - khi ứng dụng khổi động hoặc khi có thay đổi cấu hình - shared preferences sẽ được sử dụng để khôi phục trạng thái của giao diện.

1. Xác định phần trong phương thức onCreate() mà thử nếu đối số savedInstanceState là null và phục hồi instance state:



1. Xóa toàn bộ khối đó
2. Trong phương thức onCreate(), tại vị trí tương tự như phần code instance state, lấy biến đếm từ trong preferences với COUNT\_KEY và gán cho nó biến mCount.



Khi bạn đọc dữ liệu từ preferences bạn không cần lấy shared preferences editor. Sử dụng bất cứ phương thức get nào trong đối tượng shared preferences (như là getInt() hoặc getString() để truy xuất dữ liệu preference.

Lưu ý rằng phương thức getInt() lấy 2 đối số: 1 cho key và 1 cho giá trị mặc định nếu key không tìm thấy. Trong trường hợp này giá trị mặc định là 0, giống với khở tạo của mCount.

1. Cập nhật giá trị của TextView chính với count mới:



1. Lấy màu từ preferences bằng key COLOR\_KEY và gán cho nó biến mColor



### Cài đặt ứng dụng

## Lưu trữ dữ liệu với Room

### Room, LiveData và ViewModel

### Room, LiveData và ViewModel