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

Chương 4.Lưu dữ liệu người dùng

Bài 1) Tùy chọn và cài đặt

1.1) Shared preferences

Giới thiệu

Shared preferences cho phép bạn lưu trữ lượng nhỏ dữ liệu cũ dưới dạng key/value trong một tập tin trên thiết bị. Để xử lí một tập tin tùy chọn , cũng như đọc , viết , và quản lý dữ liệu tùy chọn , sử dụng lớp Shared preferences. Android framework tự quản lí các tệp tin tùy chọn được chia sẻ. Tất cả các thành phần trong ứng dụng của bạn có thể truy cập vào tập này , nhưng nó không thể truy cập vào các ứng dụng khác.

Dữ liệu bạn lưu vào shared preferences khác với dữ liệu trong trạng thái hoạt động đã lưu mà bạn tìm hiểu trong chương trước.

- Dữ liệu trong saved instance state của một Activity được giữ lại trong suốt phiên làm việc của người dùng, ngay cả khi Activity bị hủy và khởi tạo lại.
- Shared Preferences tồn tại qua nhiều phiên làm việc khác nhau. Chúng vẫn được lưu giữ ngay cả khi ứng dụng bị đóng, khởi động lại hoặc thiết bị được khởi động lại.

Chỉ sử dụng Shared Preferences khi bạn cần lưu một lượng nhỏ dữ liệu dưới dạng cặp key/value đơn giản. Đối với dữ liệu lớn hơn và cần lưu trữ lâu dài, hãy sử dụng các phương pháp khác như thư viện Room hoặc cơ sở dữ liệu SQL.

Những điều bạn nên biết trước

Bạn nên làm quen với :

- Tạo , xây dựng và chạy ứng dụng trong Android Studio.
- Thiết kế giao diện với button và text view
- Lưu và khôi phục trạng thái của activity

Những gì bạn sẽ học

Bạn sẽ học cách:

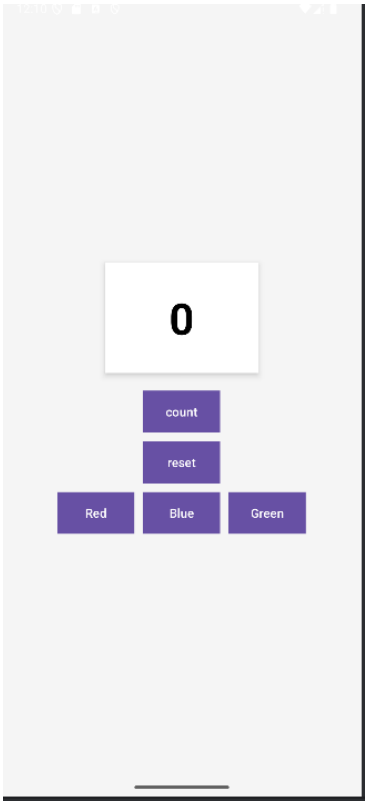
- Xác định Shared Preferences là gì.
- Tạo tệp Shared Preferences cho ứng dụng của bạn.
- Lưu dữ liệu vào Shared Preferences và đọc lại các giá trị đã lưu.
- Xóa dữ liệu trong Shared Preferences

Những điều bạn sẽ phải làm

- Cập nhật ứng dụng để có thể lưu, truy xuất và đặt lại Shared Preferences.

Tổng quan về ứng dụng

Ứng dụng HelloSharedPrefs là một biến thể khác của ứng dụng HelloToast mà bạn đã tạo trong bài học 1.Ứng dụng này bao gồm các nút để tăng số đếm, thay đổi màu nền và đặt lại cả số đếm lẫn màu sắc về mặc định.Ngoài ra, ứng dụng còn sử dụng themes và styles để định dạng các nút.



Bạn bắt đầu với ứng dụng mẫu và thêm Shared Preferences vào mã của MainActivity.Bạn cũng thêm một nút Reset, giúp đặt lại cả số đếm và màu nền về mặc định, đồng thời xóa tệp Shared Preferences.

Nhiệm vụ 1: Khám phá ứng dụng HelloSharedPrefs

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

1.1Mở và chạy dự án HelloSharedPrefs

1 Tải xuống và giải nén ứng dụng HelloSharedPrefs-Starter.
2 Mở dự án trong Android Studio, sau đó build và chạy ứng dụng. Thử nghiệm các chức năng sau:
   - Nhấn nút Count để tăng số hiển thị trên TextView chính.
   - Nhấn các nút màu sắc để thay đổi màu nền của TextView chính.
   - Xoay màn hình thiết bị và kiểm tra xem màu nền và số đếm có được giữ lại hay không.
   - Nhấn nút Reset để đặt lại số đếm và màu sắc về mặc định.
3 Thoát ứng dụng hoàn toàn bằng một trong các cách sau:
   - Trong Android Studio, chọn Run > Stop 'app' hoặc nhấn biểu tượng Stop trên thanh công cụ.
   - Trên thiết bị, nhấn nút Recent Apps (nút vuông ở góc dưới bên phải), sau đó:Vuốt thẻ ứng dụng HelloSharedPrefs để đóng.Hoặc nhấn vào dấu X ở góc phải của thẻ ứng dụng.Nếu đóng ứng dụng theo cách này, hãy đợi vài giây trước khi mở lại để hệ thống xử lý hoàn tất.
4 Chạy lại ứng dụng. Ứng dụng sẽ khởi động lại với giao diện mặc định—số đếm là 0 và màu nền là xám.

1.2 Khám phá mã hoạt động

1. Mở MainActivity.
2. Kiểm tra mã nguồn và lưu ý sau:

- Biến đếm (mCount) được định nghĩa là một số nguyên. Phương thức countUp( ) trong sự kiện onClick sẽ tăng giá trị này và cập nhật TextView chính.
- Biến màu (mColor) cũng là một số nguyên, ban đầu được đặt là màu xám trong tệp colors.xml với tên default_background.
- Phương thức changeBackground( ) trong sự kiện onClick sẽ lấy màu nền của nút được nhấn, sau đó đặt màu đó cho TextView chính.
- Cả hai số nguyên mCount và mColor được lưu vào Bundle trạng thái phiên làm việc trong onSaveInstanceState ( ) và được khôi phục trong onCreate( ).Các khóa của Bundle để lưu giá trị của Count và Color được định nghĩa bằng các biến private (COUNT_KEY và COLOR_KEY).

Nhiệm vụ 2: Lưu và khôi phục dữ liệu từ tệp Shared Preferences

Trong nhiệm vụ này, bạn sẽ 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 (gồm số đếm hiện tại và màu sắc) cũng chính là dữ liệu mà bạn bảo toàn trong instance state, nên bạn không cần lưu trữ hai lần.Bạn có thể thay thế hoàn toàn instance state bằng Shared Preferences.

2.1 Khởi tạo các tùy chọn

1. Thêm các biến thành viên vào lớp MainActivity để lưu tên tệp Shared Preferences và một đối tượng SharedPreferences.

```
private SharedPreferences mPreferences;
1 usage
private String sharedPrefFile = "com.example.android.hellosharedprefs";
```

Bạn có thể đặt tên tệp Shared Preferences theo ý muốn, nhưng theo thông lệ, tên tệp thường trùng với tên package của ứng dụng.

2. Trong phương thức onCreate(), khởi tạo Shared Preferences.Chèn đoạn mã này trước câu lệnh if.

```
mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);
```

Phương thức getSharedPreferences() (từ Context của Activity) sẽ mở với tên được chỉ định (sharedPrefFile) ở chế độ MODE_PRIVATE.

Lưu ý:Các phiên bản Android cũ hơn có các chế độ khác cho phép tạo tệp Shared Preferences có thể đọc hoặc ghi bởi toàn bộ hệ thống (world-readable hoặc world-writable). Tuy nhiên, các chế độ này đã bị loại bỏ từ API 17 và hiện nay không được khuyến khích do lý do bảo mật.Nếu bạn cần chia sẻ dữ liệu với ứng dụng khác, hãy cân nhắc sử dụng content URI được cung cấp bởi FileProvider.

Solution code for MainActivity

```
package com.example.hellosharedprefs;

import ...
public class MainActivity extends AppCompatActivity {
    7 usages
    private int mCount = 0;
    7 usages
    private TextView mShowCount;
    7 usages
    private int mColor;
    4 usages
    private SharedPreferences mPreferences;
    1 usage
    private String sharedPrefFile = "com.example.android.hellosharedprefs";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mShowCount = findViewById(R.id.textViewCount);
        mColor = Color.WHITE;
        mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);
        mCount = mPreferences.getInt( key: "count", defValue: 0);
        mColor = mPreferences.getInt( key: "color", Color.WHITE);
        mShowCount.setText(String.valueOf(mCount));
        mShowCount.setBackgroundColor(mColor);

        Button buttonIncrease = findViewById(R.id.buttonIncrease);
        Button buttonReset = findViewById(R.id.buttonReset);
        Button buttonRed = findViewById(R.id.buttonRed);
        Button buttonBlue = findViewById(R.id.buttonBlue);
        Button buttonGreen = findViewById(R.id.buttonGreen);
```

```
Button buttonGreen = findViewById(R.id.buttonGreen);

buttonIncrease.setOnClickListener(view -> {
    mCount++;
    mShowCount.setText(String.valueOf(mCount));
    savePreferences();
});

buttonReset.setOnClickListener(view -> {
    mCount = 0;
    mColor = Color.WHITE;
    mShowCount.setText(String.valueOf(mCount));
    mShowCount.setBackgroundColor(mColor);
    savePreferences();
});

buttonRed.setOnClickListener(view -> changeBackgroundColor(Color.RED));
buttonBlue.setOnClickListener(view -> changeBackgroundColor(Color.BLUE));
buttonGreen.setOnClickListener(view -> changeBackgroundColor(Co
}

3 usages
private void changeBackgroundColor(int color) {
    mColor = color;
    mShowCount.setBackgroundColor(color);
    savePreferences();
}

3 usages
private void savePreferences() {
```

2.2 Lưu các tùy chọn trong onPause( )

Lưu Preferences khá giống với việc lưu instance state—cả hai đều lưu dữ liệu vào một Bundle dưới dạng cặp key/value.Tuy nhiên, với Shared Preferences, bạn sẽ lưu dữ liệu trong phương thức onPause() của vòng đời Activity.Để ghi dữ liệu vào Shared Preferences, bạn cần sử dụng một đối tượng SharedPreferences.Editor.

1. Thêm phương thức onPause( ) vào lớp MainActivity.

```
protected void onPause() {
    super.onPause();
    savePreferences();
}
```

2. Trong phương thức onPause(), lấy một đối tượng Editor cho SharedPreferences.

Để ghi dữ liệu vào Shared Preferences, cần có một đối tượng Editor.

Thêm dòng này vào onPause(), ngay sau khi gọi super.onPause( ).

3. Sử dụng phương thức putInt() để lưu cả hai giá trị mCount và mColor vào Shared Preferences với các khóa tương ứng.

   Lớp SharedPreferences.Editor cung cấp nhiều phương thức "put" cho các kiểu dữ liệu khác nhau, bao gồm putInt() và putString().
4. Gọi phương thức apply() để lưu Shared Preferences.
   Phương thức apply() lưu Shared Preferences một cách bất đồng bộ (asynchronously), tách biệt khỏi luồng giao diện người dùng (UI thread).Ngoài ra, SharedPreferences.Editor cũng có phương thức commit() để lưu dữ liệu một cách đồng bộ (synchronously). Tuy nhiên, commit() không được khuyến khích vì có thể gây chặn các hoạt động khác.

   1. Xóa toàn bộ phương thức onSaveInstanceState( ).Vì instance state chứa cùng dữ liệu với Shared Preferences, nên bạn có thể thay thế hoàn toàn instance state bằng Shared Preferences.

   Solution code for MainActivity onPause() method:

```
protected void onPause() {
    super.onPause();
    savePreferences();
}

4 usages
private void savePreferences() {
    SharedPreferences.Editor editor = mPreferences.edit();

    editor.putInt("count", mCount);
    editor.putInt("color", mColor);
    editor.apply();
}
```

2.3 Khôi phục tùy chọn với onCreate( )

Giống như với instance state, ứng dụng của bạn sẽ đọc dữ liệu đã lưu trong Shared Preferences trong phương thức onCreate().Vì Shared Preferences chứa cùng dữ liệu với instance state, nên ta có thể thay thế hoàn toàn instance state bằng Shared Preferences.Mỗi khi onCreate() được gọi—dù là khi ứng dụng khởi động hay khi có sự thay đổi cấu hình—Shared Preferences sẽ được dùng để khôi phục trạng thái giao diện.

1. Xác định vị trí trong phương thức onCreate( ) kiểm tra xem tham số savedInstanceState có null không và khôi phục instance state.

```
if (savedInstanceState != null) {
    mCount = savedInstanceState.getInt(COUNT_KEY);
    if (mCount != 0) {
        mShowCountTextView.setText(String.format("%s", mCount));
    }
    mColor = savedInstanceState.getInt(COLOR_KEY);
    mShowCountTextView.setBackgroundColor(mColor);
}
```

2. Xóa toàn bộ khối mã đó
3. Trong phương thức onCreate(), tại vị trí trước đó của mã khôi phục instance state, lấy giá trị mCount từ Shared Preferences bằng khóa COUNT_KEY và gán nó cho biến mCount.

```
mCount = mPreferences.getInt(COUNT_KEY, defValue: 0);
```

Khi đọc dữ liệu từ Shared Preferences, bạn không cần lấy một đối tượng của SharedPreferences .Thay vào đó, hãy sử dụng các phương thức "get" trên đối tượng SharedPreferences (chẳng hạn như getInt() hoặc getString()) để truy xuất dữ liệu đã lưu.
Lưu ý rằng phương thức getInt() nhận hai đối số:Khóa (key) để lấy giá trị từ Shared Preferences.Giá trị mặc định nếu khóa không được tìm thấy. Trong trường hợp này, giá trị mặc định là 0, giống với giá trị khởi tạo của mCount.

4. Cập nhật giá trị của chế độ xem văn bản chính với số lượng mới.

```
mShowCountTextView.setText(String.valueOf(mCount));
```

5. Lấy màu từ tùy chọn bằng phím COLOR_KEY và gán nó cho biến mColor.

```
mColor = mPreferences.getInt(COLOR_KEY, Color.WHITE);
```

Giống như trước, đối số thứ hai của getInt() là giá trị mặc định sẽ được sử dụng nếu khóa không tồn tại trong Shared Preferences.Trong trường hợp này, bạn có thể tái sử dụng giá trị của mColor, vì nó vừa được khởi tạo với màu nền mặc định ở phần trên của phương thức.

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

```
mShowCountTextView.setBackgroundColor(mColor);
```

7. Chạy ứng dụng. Nhấp vào nút đếm và thay đổi màu nền để cập nhật trạng thái phiên bản và tùy chọn.
8. Xoay thiết bị hoặc trình giả lập để xác minh số lượng và màu sắc được lưu qua các thay đổi cấu hình.
9. Buộc thoát ứng dụng bằng một trong các phương pháp sau:
    - Trong Android Studio, chọn Chạy > Dừng 'ứng dụng'.
    - Trên thiết bị, nhấp vào nút Gần đây (nút hình vuông ở góc dưới bên phải). Vuốt thẻ để thoát ứng dụng HelloSharedPrefs hoặc nhấp vào X ở góc bên phải.
10. Chạy lại ứng dụng. Ứng dụng sẽ khởi động lại và tải các tùy chọn, duy trì trạng thái.

Solution code for MainActivity onCreate( ) method:

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mShowCountTextView = findViewById(R.id.textViewCount);
    Button buttonIncrease = findViewById(R.id.buttonIncrease);
    Button buttonReset = findViewById(R.id.buttonReset);
    Button buttonRed = findViewById(R.id.buttonRed);
    Button buttonBlue = findViewById(R.id.buttonBlue);
    Button buttonGreen = findViewById(R.id.buttonGreen);

    mPreferences = getSharedPreferences(mSharedPrefFile, MODE_PRIVATE);
    mCount = mPreferences.getInt(COUNT_KEY,  defValue: 0);

    mColor = mPreferences.getInt(COLOR_KEY, Color.WHITE);

    mShowCountTextView.setText(String.valueOf(mCount));

    mShowCountTextView.setBackgroundColor(mColor);
```

2.4 Đặt lại tùy chọn trong trình xử lý nhấp chuột reset( )

Nút reset trong ứng dụng khởi động sẽ reset cả số đếm 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 lúc.

1. Trong phương thức xử lý nhấp chuột reset(), sau khi màu sắc và số lượng được đặt lại, hãy lấy trình soạn thảo cho đối tượng SharedPreferences:
2. Xóa tất cả các tùy chọn chia sẻ
3. Áp dụng các thay đổi:

Solution code for the reset() method:

```
1 usage
private void reset() {
    mCount = 0;
    mColor = ContextCompat.getColor( context: this, R.color.white);
    mShowCountTextView.setText(String.valueOf(mCount));
    mShowCountTextView.setBackgroundColor(mColor);

    // Lưu vào SharedPreferences
    savePreferences();
}
```