

LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

CHƯƠNG 5:

STORAGE VÀ SHARED PREFERENCES

- ❖ Giảng viên: Ths. Nguyễn Trung Hiếu
- ❖ Email:hieunt.tg@ptithcm.edu.vn
- ❖ Mobie: 0983051825

- ◉ Kết thúc bài học này bạn có khả năng
 - ◉ Lưu trữ thông tin
 - ◉ Sử dụng Shared Preferences



Phần I: Lưu trữ thông tin

-  Lưu trữ thông tin trên thiết bị
-  Lưu trữ thông tin trên thẻ nhớ

Phần II: Shared Preferences

-  Shared Preferences

1

Lưu trữ thông tin

LƯU TRỮ DỮ LIỆU

- Android cung cấp một số lựa chọn để lưu trữ dữ liệu
- Lựa chọn giải pháp nào tùy thuộc vào nhu cầu bảo mật của ứng dụng và kích thước của file cần lưu trữ
- Các tùy chọn lưu trữ bao gồm:
 - Shared Preferences
 - Internal Storage
 - External Storage
 - SQLite Database
 - Network Connection

INTERNAL STORAGE

- Bạn có thể lưu file trực tiếp vào bộ nhớ trong của thiết bị
- Mặc định, các file được lưu ở bộ nhớ trong là riêng tư và các ứng dụng khác không có quyền truy cập đến
- Khi người dùng xóa ứng dụng, các file này sẽ tự động bị xóa theo



- Tạo và ghi private file tới internal storage
 - Gọi openFileOutput() với tên file và tham số chế độ hoạt động. Trả về một FileOutputStream
 - Ghi tới file sử dụng write()
 - Đóng stream sử dụng close()

```
FileOutputStream fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);
fos.write(string.getBytes());
fos.close();
```

INTERNAL STORAGE

- MODE_PRIVATE sẽ tạo file (hoặc thay thế file nếu trùng tên) và thiết lập file này là riêng tư đối với các ứng dụng khác
- Các mode khác có thể là:
MODE_APPEND, MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE



INTERNAL STORAGE

- Đọc file từ Internal Storage

- Gọi openFileInput() và truyền tên file muốn đọc. Trả về FileInputStream
- Đọc sử dụng read()
- Sau đó đóng stream sử dụng close()



Lưu trữ Cache files

- Nếu bạn muốn lưu cache file, sử dụng getCacheDir() để mở File đại diện cho thư mục nội tại là nơi mà ứng dụng lưu thông tin cache tạm thời
- Khi thiết bị ở trạng thái thiếu dung lượng, Android có thể xóa một số file cache để giải phóng dung lượng.
- Tuy nhiên, không nên dựa vào Android sẽ dọn dẹp cho bạn
- Cỡ của file cache tối đa 1 MB
- Khi bạn xóa bỏ ứng dụng, file cache sẽ bị xóa theo

Tạo cache file

```
public void createCache()
{
    try {
        File pathCacheDir = getCacheDir();
        String strCacheFileName = "myCacheFile.cache";
        String strFileContents = editdata.getText()+"";
        File newCacheFile = new
                            File(pathCacheDir, strCacheFileName);
        newCacheFile.createNewFile();
        FileOutputStream foCache =
            new FileOutputStream(
                newCacheFile.getAbsolutePath());
        foCache.write(strFileContents.getBytes());
        foCache.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Đọc cache file

```
public void readCache() {  
    try {  
        File pathCacheDir = getCacheDir();  
        String strCacheFileName = "myCacheFile.cache";  
        File newCacheFile = new  
            File(pathCacheDir, strCacheFileName);  
        Scanner sc=new Scanner(newCacheFile);  
        String data="";  
        while(sc.hasNext())  
        {  
            data+=sc.nextLine()+"\n";  
        }  
        editdata.setText(data);  
        sc.close();  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
}
```

Lấy thông tin tất cả cache files

```
public void loadAllCache()
{
    File pathCacheDir = getCacheDir();
    File []listCache= pathCacheDir.listFiles();
    for(File f :listCache)
    {
        //process f here
        f.delete();
    }
}
```

INTERNAL STORAGE

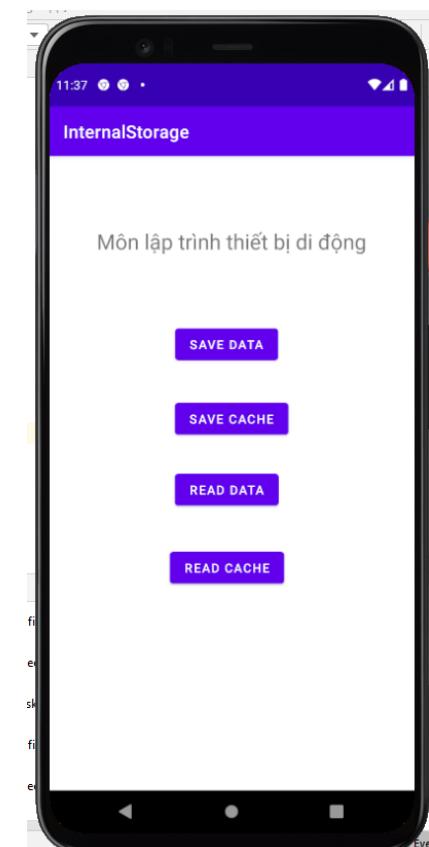
DEMO

- Đọc dữ liệu từ file (Internal)

```
private void saveData() {  
    FileOutputStream fileOutputStream = null;  
    try {  
        fileOutputStream = openFileOutput(filename, Context.MODE_PRIVATE);  
        fileOutputStream.write(content.getBytes());  
        fileOutputStream.close();  
        Toast.makeText(MainActivity.this, "Save data successfully", Toast.LENGTH_SHORT);  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

```
private void ReadData() {  
    FileInputStream fileInputStream = null;  
    try {  
        fileInputStream = openFileInput(filename);  
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(fileInputStream));  
        StringBuffer buffer = new StringBuffer();  
        String line = null;  
        while ((line=bufferedReader.readLine()) !=null) {  
            buffer.append(line).append("\n");  
        }  
        txtShowData.setText(buffer.toString());  
        Log.d(TAG, "ReadData: " + buffer.toString());  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

INTERNAL STORAGE



```

private void ReadDataCache() {
    FileInputStream fileInputStream = null;
    File file = null;
    try {
        file = new File(getCacheDir(),filename);
        fileInputStream = new FileInputStream(file) ;
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(fileInputStream));
        StringBuffer buffer = new StringBuffer();
        String line = null;
        while ((line=bufferedReader.readLine()) !=null) {
            buffer.append(line).append("\n");
        }
        txtShowData.setText(buffer.toString());
        Log.d(TAG, "ReadDataCache: " + buffer.toString());
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

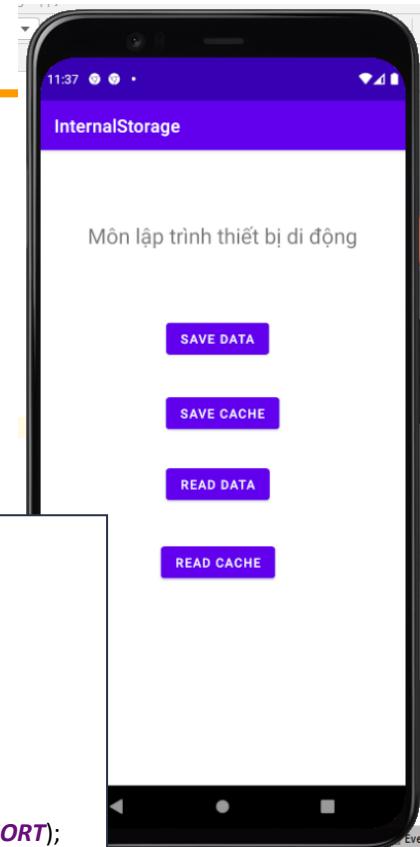
```

```

private void saveDataCache() {
    FileOutputStream fileOutputStream = null;
    File file = null;
    try {
        file = new File(getCacheDir(),filename);
        Log.d(TAG, "Thư mục chứa Cache: " + getCacheDir().getAbsolutePath());
        fileOutputStream = new FileOutputStream(file);
        fileOutputStream.write(content.getBytes());
        fileOutputStream.close();
        Toast.makeText(MainActivity.this,"Save data to cache successfully",Toast.LENGTH_SHORT);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

INTERNAL STORAGE



EXTERNAL STORAGE

Lưu trữ thông tin trên thẻ nhớ

- Lưu trữ thông tin trên thẻ nhớ SDCard
- Ưu điểm là dung lượng lưu trữ lớn hơn



EXTERNAL STORAGE

```
<uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"  
/>
```

Name	Size
init.usb.rc	3915
↳ mnt	2
↳ asec	2
↳ obb	2
↳ sdcard	2
↳ Alarms	2
↳ DCIM	2
myfile.txt	73

Đọc từ thẻ nhớ

```
public void readData()
{
    String sdcard=Environment
                .getExternalStorageDirectory()
                .getAbsolutePath() + "/myfile.txt";
    try {
        Scanner scan=new Scanner(new File(sdcard));
        String data="";
        while(scan.hasNext())
        {
            data+=scan.nextLine() + "\n";
        }
        scan.close();
        editdata.setText(data+ "");
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

Ghi vào thẻ nhớ

```
public void writeData()
{
    String sdcard=Environment
                .getExternalStorageDirectory()
                .getAbsolutePath() + "/myfile.txt";
    try {
        OutputStreamWriter writer=
            new OutputStreamWriter(
            new FileOutputStream(sdcard));
        writer.write(editdata.getText()+"");
        writer.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

EXTERNAL STORAGE

The image features a young girl with dark hair and glasses, peeking over a black rectangular board. She is holding a yellow pencil horizontally across the top of the board. The board has the word "DEMO" written on it in large white letters. A white hand cursor icon is positioned over the letter "O". Below the board, there is a line of Vietnamese text: "- Đọc dữ liệu từ file (External)".

DEMO

- Đọc dữ liệu từ file (External)

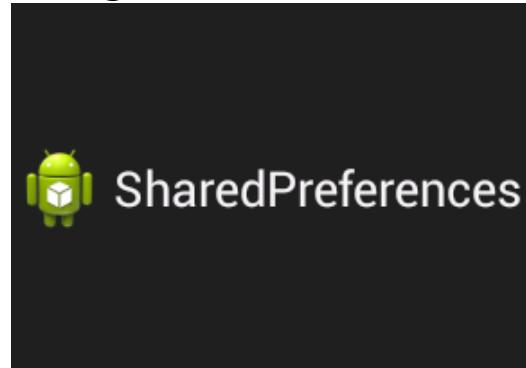
EXTERNAL STORAGE

2

Shared Preferences

SHARED PREFERENCES

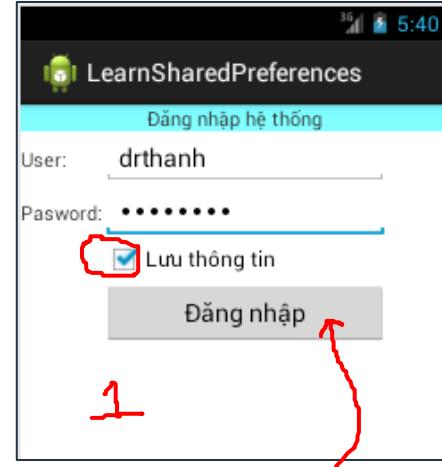
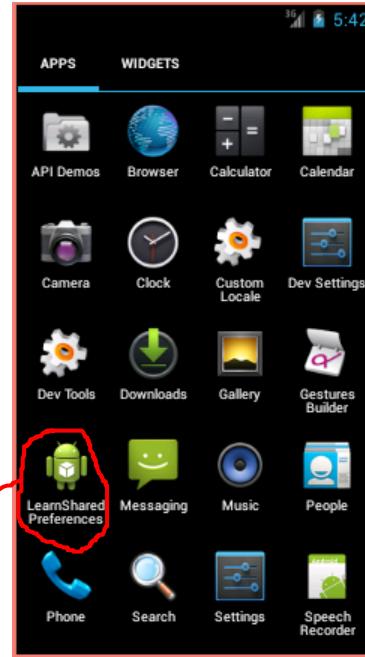
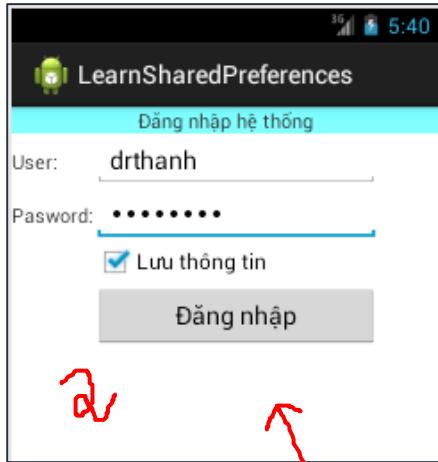
- Lớp SharedPreferences cung cấp một framework chung để lưu trữ dữ liệu dưới dạng key-value của dữ liệu kiểu nguyên thủy
- Có thể sử dụng để lưu trữ dữ liệu kiểu boolean, float, int, long, string
- Dữ liệu này được lưu trữ trong các phiên làm việc của người dùng (ngay cả khi ứng dụng bị kill)



SHARED PREFERENCES

- Gọi getSharedPreferences()
- Ghi thông tin:
 - Gọi edit() để nhận SharedPreferences.Editor
 - Bổ sung giá trị sử dụng putBoolean, putString,...
 - Commit giá trị mới sử dụng commit()
- Đọc thông tin: sử dụng phương thức như putBoolean, putString,...

SHARED PREFERENCES



Check và click vào nút Login

Mở lại ứng dụng →
thông tin được khôi
phục lại

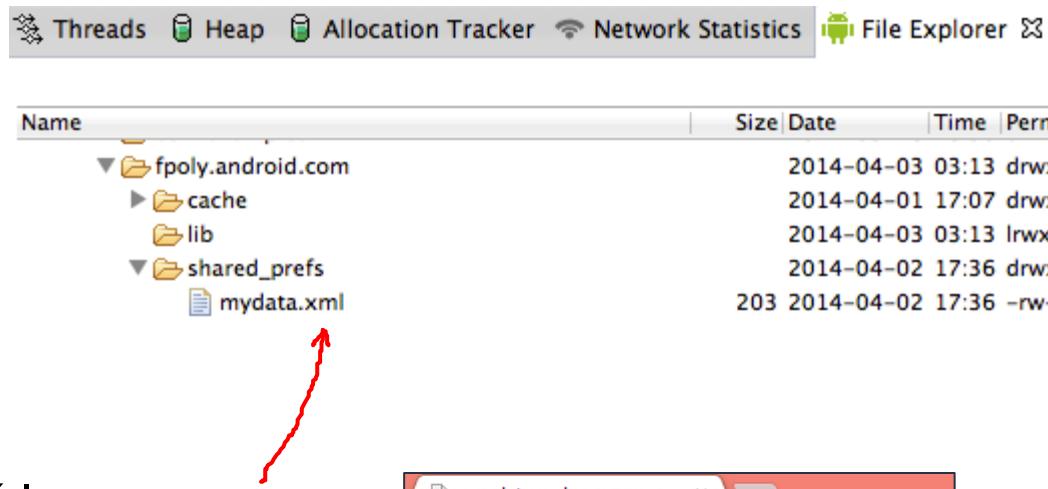
SHARED PREFERENCES

```
private void savingPreferences() {
    //tao doi tuong getSharedPreferences
    SharedPreferences sharepreference = getSharedPreferences(filename, MODE_PRIVATE);
    //tao doi tuong Editor de luu thay doi
    SharedPreferences.Editor editor = sharepreference.edit();
    String user = txtUsername.getText().toString();
    String password = txtPassword.getText().toString();
    boolean chk = chkStatus.isChecked();
    if (!chk)
    {
        //Xoa moi luu tru truoc do
        editor.clear();
    }
    else {
        //luu vao editor
        editor.putString("username", user);
        editor.putString("password", password);
        editor.putBoolean("savestatus", chk);
    }
    //chấp nhận lưu xuống file
    editor.commit();
}
```

SHARED PREFERENCES

```
private void restoringPreferences() {
    SharedPreferences pref = getSharedPreferences(filename, MODE_PRIVATE);
    boolean chk = pref.getBoolean("savestatus",false);
    if (chk)
    {
        //lấy user, pwd, nếu không thấy giá trị mặc định là rỗng
        String user = pref.getString("username","");
        String password = pref.getString("password","");
        txtUsername.setText(user);
        txtPassword.setText(password);
    }
    chkStatus.setChecked(chk);
}
```

SHARED PREFERENCES



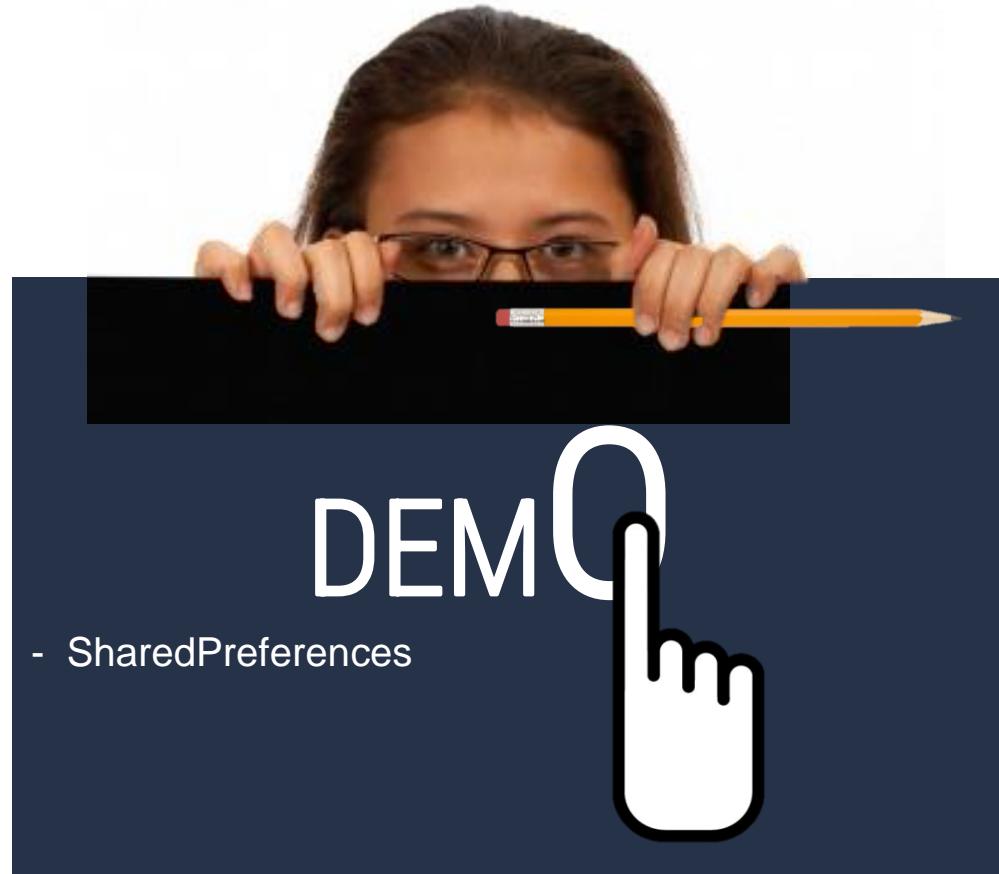
Vị trí lưu
SharedPreferences
(DDMS/File
Explorer/data/data/
package
name/shared_pref

A screenshot of a file viewer window titled 'my_data.xml'. The address bar shows 'file:///E:/my_data.xml'. The content pane displays the XML code:

```
<map>
    <string name="user">drthanh</string>
    <boolean name="checked" value="true"/>
    <string name="pwd">hoilamgi</string>
</map>
```

Định dạng
XML

SHARED PREFERENCES



TỔNG KẾT NỘI DUNG BÀI HỌC

📖 Phần I: Lưu trữ thông tin

📖 Lưu trữ thông tin trên thiết bị

📖 Lưu trữ thông tin trên thẻ nhớ

📖 Phần II: Shared Preferences

📖 Shared Preferences

