



**BÀI 7:**

**LẬP TRÌNH ANDROID CƠ BẢN**

**STORAGE VÀ SHARED PREFERENCES**

- ⊙ 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





# **BÀI 7: STORAGE VÀ SHARED PREFERENCES**

## **PHẦN I: LƯU TRỮ THÔNG TIN**

- 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

- 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();
```

- `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`





- 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();  
    }  
}
```



# DEMO

- Đọc dữ liệu từ file (Internal)



## Lưu trữ thông tin trên thẻ nhớ

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



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

Threads Heap Allocation Tracker Network Statistics File		
Name	Size	
init.usb.rc	3915	1
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();
    }
}
```



# DEMO

- Đọc dữ liệu từ file (External)





# **BÀI 7: STORAGE VÀ SHARED PREFERENCES**

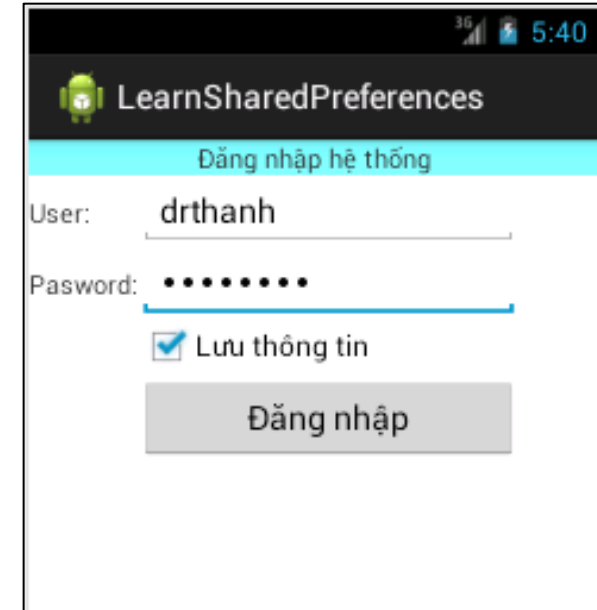
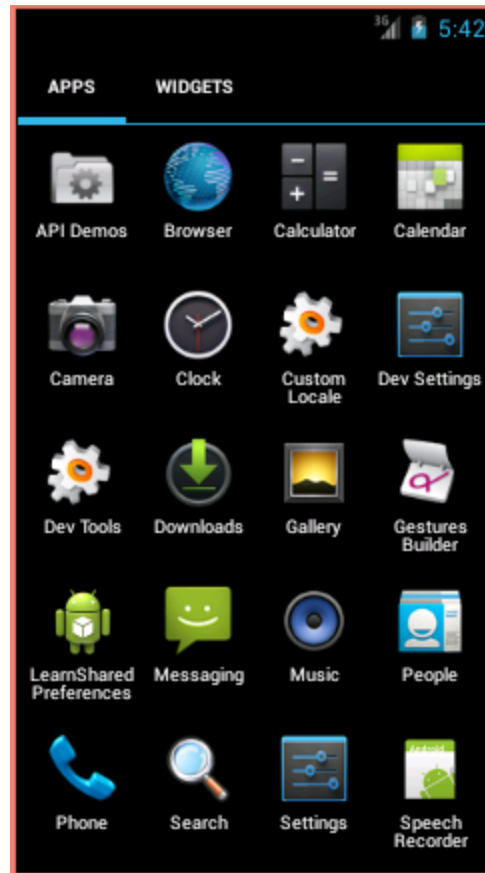
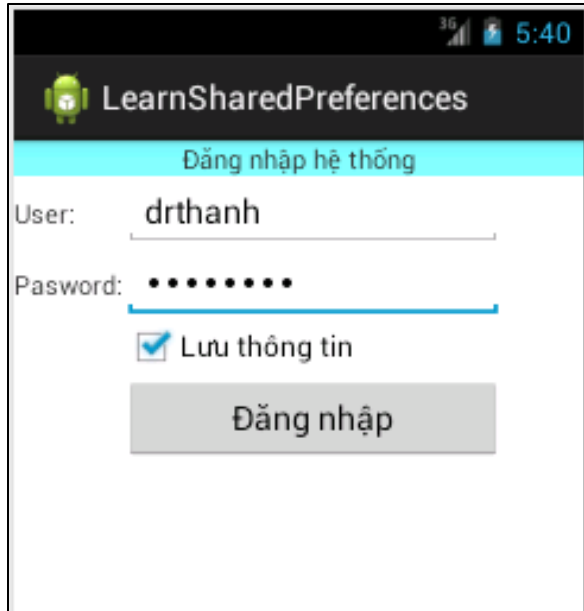
## **PHẦN II: SHARED PREFERENCES**

- 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)



SharedPreferences

- 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`,...



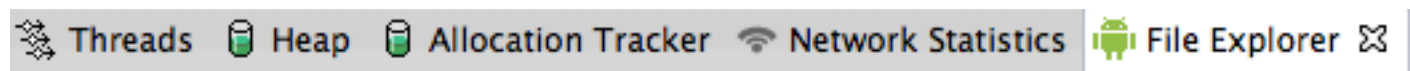
Check và click vào nút Login

Mở lại ứng dụng → thông tin được khôi phục lại

```
private void savingPreferences() {
    //tạo đối tượng getSharedPreferences
    SharedPreferences sharepreference = getSharedPreferences(filename, MODE_PRIVATE);
    //tạo đối tượng Editor để lưu thay đổi
    SharedPreferences.Editor editor = sharepreference.edit();
    String user = txtUsername.getText().toString();
    String password = txtPassword.getText().toString();
    boolean chk = chkStatus.isChecked();
    if (!chk)
    {
        //Xóa mọi lưu trữ trước đó
        editor.clear();
    }
    else {
        //lưu vào editor
        editor.putString("username", user);
        editor.putString("password", password);
        editor.putBoolean("savestatus", chk);
    }
    //chấp nhận lưu xuống file
    editor.commit();
}
```

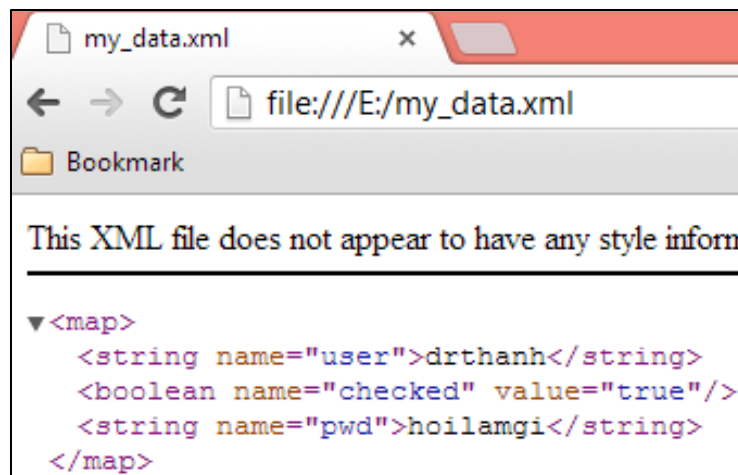


```
private void restoringPreferences() {
    SharedPreferences pref = getSharedPreferences(filename, MODE_PRIVATE);
    boolean chk = pref.getBoolean("savestatus", false);
    if (chk)
    {
        //lấy user, pwd, nếu không thấy giá trị mặc định là rỗng
        String user = pref.getString("username",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

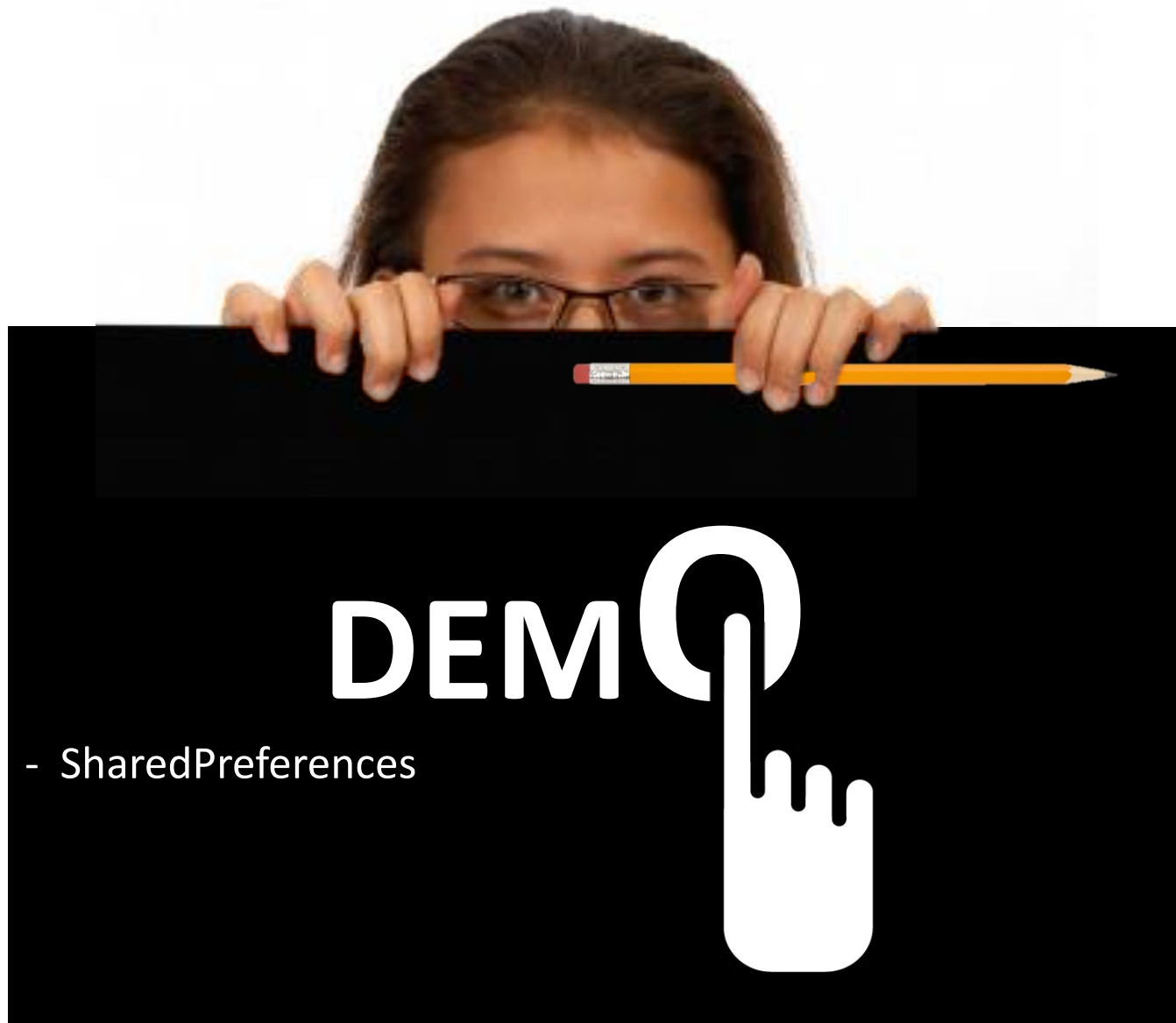


Name	Size	Date	Time	Perm
▼ folder fpoly.android.com		2014-04-03	03:13	drwxr
▶ folder cache		2014-04-01	17:07	drwxr
folder lib		2014-04-03	03:13	lrwxr
▼ folder shared_prefs		2014-04-02	17:36	drwxr
file mydata.xml	203	2014-04-02	17:36	-rw-r

Vị trí lưu  
SharePreferences  
(DDMS/File  
Explorer/data/data/  
package  
name/shared\_prefs



Định dạng  
XML



## 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





**Cảm ơn**