



河北师范大学软件学院
Software College of Hebei Normal University



Android 基础开发

第七章 第一讲 Android的文件操作



Java与移动智能设备开发



教学目标

- 掌握Android中文件的存储



目录

1

文件存储的简介

2

SharedPreferences的使用

3

File的存储



文件存储简介

- 一般的应用程序都涉及到数据的输入和输出，应用程序的参数设置、程序运行状态数据都需要保存到外部的存储器上，这样系统关机后数据才不会丢失。





文件存储简介

- Android常见的数据存储方式：
 - 系统配置暂存的轻量级存储，SharedPreferences。
 - 系统中的中小型存储机制，File存储。
 - 系统中的相对大型存储机制，SQLite。





目录

- 1 文件存储的简介
- 2 **SharedPreferences的使用**
- 3 File的存储



SharedPreferences的使用

- SharedPreferences是一种轻量级数据存储机制，适用情况：
 - 要存储的数据格式简单。
 - 都是普通的字符串，标准类型的值。
 - 一般用来存储应用程序的配置信息等。

The screenshot shows an Android application window with the title "SharedPreferences". The status bar at the top indicates the time is 12:38. The app's interface consists of a white background with a blue header. Below the header, there are two input fields: "用户名" (Username) and "密码" (Password). The "用户名" field has a pink underline, and the "密码" field has a grey underline. Below these fields is a grey button labeled "登录" (Login). At the bottom, there are two checkboxes: "记住密码" (Remember password) and "自动登录" (Auto login).



SharedPreferences的使用

- SharedPreferences存储数据采用key/value的形式，常用的方法如下：

```
contains(String key) : boolean - SharedPreferences
edit() : Editor - SharedPreferences
equals(Object o) : boolean - Object
getAll() : Map<String,?> - SharedPreferences
getBoolean(String key, boolean defValue) : boolean - SharedPreferences
getClass() : Class<? extends Object> - Object
getFloat(String key, float defValue) : float - SharedPreferences
getInt(String key, int defValue) : int - SharedPreferences
getLong(String key, long defValue) : long - SharedPreferences
getString(String key, String defValue) : String - SharedPreferences
hashCode() : int - Object
notify() : void - Object
notifyAll() : void - Object
registerOnSharedPreferenceChangeListener(OnSharedPreferenceChangeListener listener) : void - SharedPreferences
toString() : String - Object
unregisterOnSharedPreferenceChangeListener(OnSharedPreferenceChangeListener listener) : void - SharedPreferences
wait() : void - Object
wait(long millis) : void - Object
wait(long millis, int nanos) : void - Object
```




SharedPreferences的使用

- SharedPreferences本身是一个接口，无法创建实例，一般通过的getSharedPreferences()方法获得实例，其中形式有三种：
 - Context.MODE_PRIVATE：本程序内访问。
 - Context.MODE_MULTI_PROCESS：多进程形式。
 - Context.MODE_WORLD_READABLE：全局可读。
 - Context.MODE_WORLD_WRITEABLE：全局可写。



SharedPreferences的使用

- SharedPreferences接口本身没有提供写入数据的能力，而是通过一个内部接口Editor类的对象来实现的。
 - 通过SharedPreferences的edit()方法获得本对象的Editor对象。
 - Editor通过如下方法在SharedPreferences中写输入数据。

```
• clear() : Editor - Editor
• commit() : boolean - Editor
• equals(Object o) : boolean - Object
• getClass() : Class<? extends Object> - Object
• hashCode() : int - Object
• notify() : void - Object
• notifyAll() : void - Object
• putBoolean(String key, boolean value) : Editor - Editor
• putFloat(String key, float value) : Editor - Editor
• putInt(String key, int value) : Editor - Editor
• putLong(String key, long value) : Editor - Editor
• putString(String key, String value) : Editor - Editor
• remove(String key) : Editor - Editor
• toString() : String - Object
• wait() : void - Object
• wait(long millis) : void - Object
• wait(long millis, int nanos) : void - Object
```



SharedPreferences的使用

- SharedPreferences的本程序内写入方式：

```
preferences = getSharedPreferences(sharename,  
                                MODE_PRIVATE);  
Editor ed = preferences.edit();  
ed.put***(key,value);  
ed.commit();
```

- SharedPreferences的本程序内读取方式：

```
preferences = getSharedPreferences(sharename,  
                                MODE_PRIVATE);  
value = preferences.get***(key,default);
```



实例



Name	Size	Date	Time	Permissions	Info
> config		2016-03-31	04:18	dr-x-----	
> d		2016-03-31	04:18	lrwxrwxrwx	-> /sys/ker...
> data		2016-03-30	03:37	drwxrwx--x	
> adb		2016-03-29	02:17	drwx-----	
> anr		2016-03-31	03:53	drwxrwxr-x	
> app		2016-03-31	04:19	drwxrwx--x	
> app-asec		2016-03-29	02:17	drwx-----	
> app-lib		2016-03-29	02:17	drwxrwx--x	
> app-private		2016-03-29	02:17	drwxrwx--x	
> backup		2016-03-31	04:19	drwx-----	
> bootchart		2016-03-29	02:17	drwxr-xr-x	
bugreports		2016-03-29	02:17	lrwxrwxrwx	-> /data/d...
> dalvik-cache		2016-03-29	02:17	drwxrwx--x	
> data		2016-03-30	03:53	drwxrwx--x	
> cn.hebtu.edu.sharedpreferences		2016-03-30	03:59	drwxr-x--x	
> cache		2016-03-30	03:53	drwxrwx--x	
> code_cache		2016-03-30	03:53	drwxrwx--x	
> shared_prefs		2016-03-31	04:19	drwxrwx--x	
userinfo.xml	190	2016-03-31	04:19	-rw-rw----	
> com.amaze.filemanager		2016-03-29	02:17	drwxr-x--x	
> com.android.backupconfirm		2016-03-29	02:17	drwxr-x--x	
> com.android.bluetooth		2016-03-31	04:19	drwxr-x--x	
> com.android.bluetoothmidiservice		2016-03-29	02:17	drwxr-x--x	



SharedPreferences的使用

- SharedPreferences的本质是文件读写。
- 位置是 /data/data/<packagename>/shared_prefs/shared_name

```
<?xml version='1.0' encoding='utf-8'
standalone='yes' ?>
<map>
    <int name="sex" value="2131034115" />
    <string name="name">name</string>
    <string name="age">10</string>
</map>
```



SharedPreferences的使用

- 读写其它应用程序的SharedPreferences的步骤：

1. 前提创建的SharedPreferences全区可读写的。

2. 创建对应的Context。

```
context = creatPackageContext(包名, 权限);
```

3. 通过context.getSharedPreferences获得对象。

4. 如需读出数据。

```
value = preferences.get***(key,default);
```

如需写入数据。

```
Editor ed = preferences.edit();
```

```
ed.put***(key,value);
```



目录

- 1 文件存储的简介
- 2 SharedPreferences的使用
- 3 **File**的存储



File的存储

- 同JavaSE一样，Android对于文件的操作通过流来实现。
 - FileInputStream
 - FileOutputStream
- 一般常见的Android的文件处理分为两类：
 - 本应用程序内部的数据文件。
 - SD卡上的文件。



应用程序内部的数据文件的读写

- Context提供了如下两个方法来获得文件的IO流：
 - `FileInputStream openFileInput(name)`
 - `FileOutputStream openFileOutput(name, mode)`
 - `MODE_PRIVATE`：该文件只能被当前程序读写。
 - `MODE_APPEND`：以追加的方式打开该文件，允许应用程序向 该文件中追加内容。
 - ~~`MODE_WORLD_READABLE`~~：该文件可以被其他程序读取。
 - ~~`MODE_WORLD_WRITEABLE`~~：该文件可以被其他程序读写。



应用程序内部的数据文件的读写

- Context提供了如下方法来访问数据文件夹
 - `getDir(name, mode)` : 在应用程序的数据文件夹下获取 **name** 对应的子目录。
 - `File getFilesDir()` : 获取应用程序数据文件夹的绝对路径。
 - `String[] fileList()` : 返回应用程序数据文件夹的全部文件。
 - `deleteFile(String)` : 删除应用程序数据文件夹某个文件。



SD卡上的文件读写

- SD卡（Secure Digital Memory Card）是一种基于半导体快闪记忆器的新一代记忆设备，它被广泛地于便携式装置上使用，例如数码相机、个人数码助理(PDA)和多媒体播放器等。
- SD卡和手机存储空间的优缺点：
 - SD卡存储速度相对较慢，但是空间相对较大。
 - SD卡在使用过程中需要进行检查和判断。



SD卡上的文件读写

- 读写SD卡上的文件的步骤：

1. 判断是否手机插入SD卡，并且程序有读写SD卡的权限。

```
Environment.getExternalStorageState().equals(android.os.Environment.MEDIA_MOUNTED);
```

2. 获得SD卡的外部目录getExternalStorageState()。
3. 使用常用的流操作进行SD卡的文件读写。
 - FileInputStream
 - FileOutputStream
 - FileReader
 - FileWriter



SD卡上的文件读写

- 读写SD卡上的文件的注意事项：
 - 手机插入SD卡，在AVD中通过mkcard命令来创建。
 - 获得读写SD卡的权限需要在AndroidManifest.xml中添加读写权限。

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



内容回顾

- 1 文件存储的简介
- 2 SharedPreferences的使用
- 3 File的存储



Thank you!

