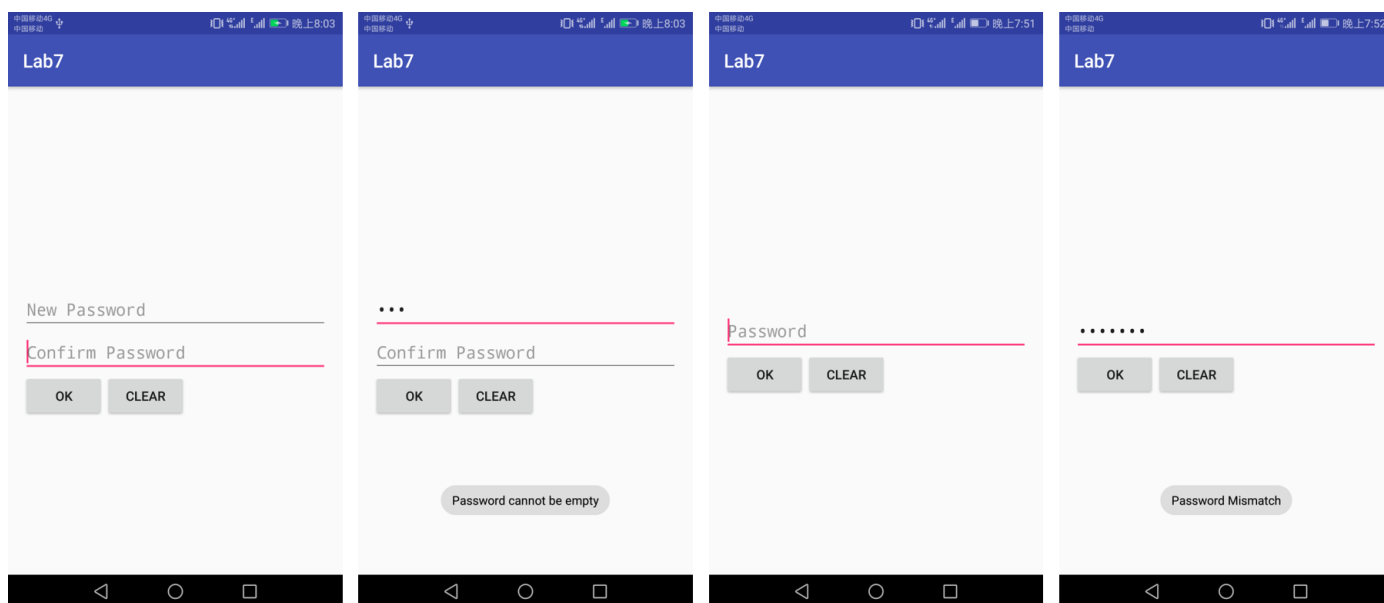


实验七 数据存储（一）

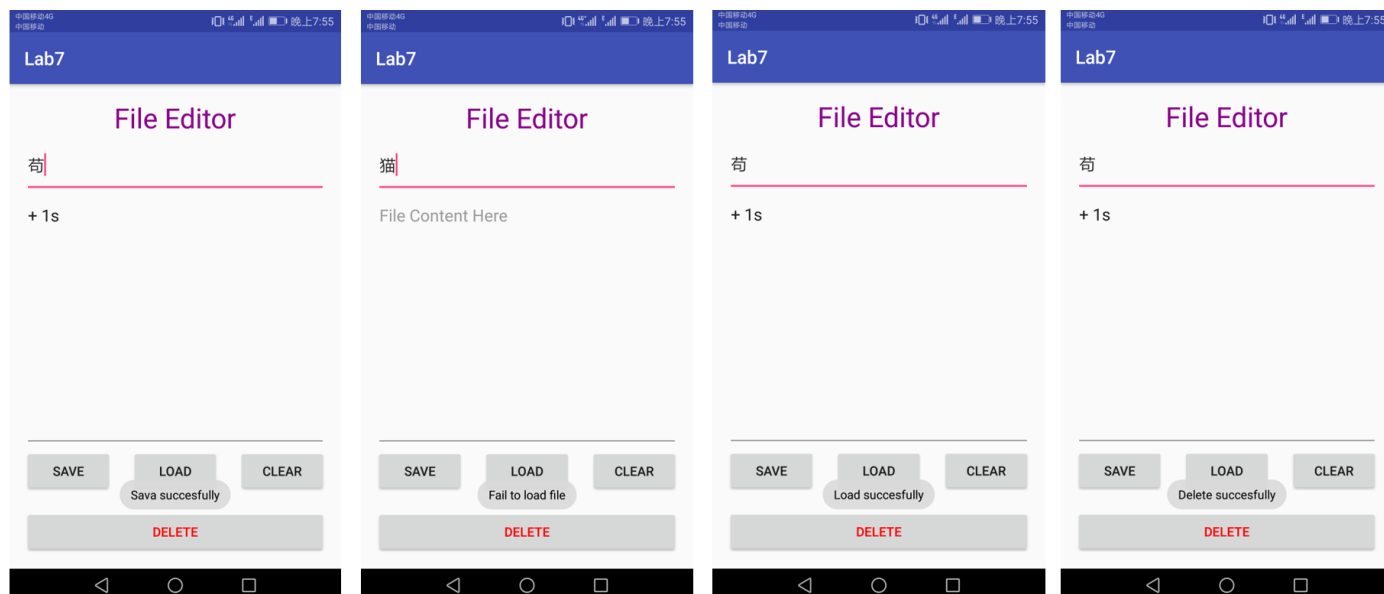
1、实验目的

- a、学习 SharedPreferences 的基本使用；
- b、学习 Android 中常见的文件操作方法；
- c、复习 Android 界面编程。

2、实验内容



从左至右，依次为：初始密码界面、密码为空提示、密码匹配后重新进入界面、密码错误提示。



从左至右，依次为：保存成功提示、写入失败提示、写入成功提示、删除成功提示。

- 1、如图所示，本次实验需要实现两个 activity；

2、首先，需要实现一个密码输入 activity：

a、如果应用首次启动，则界面呈现出两个输入框，分别为新密码输入和确认密码输入框；

b、输入框下方有两个按钮：

- OK 按钮，点击之后：

- 若 new password 为空，则弹出密码为空的提示；

- 若 new password 与 confirm password 不匹配，则弹出不匹配的提示；

- 若密码不为空且互相匹配，则保存密码，进入文件编辑界面。

- CLEAR 按钮，点击之后清除所有输入框的内容。

c、完成创建密码后，退出应用再进入应用，则只呈现一个密码输入框；

- 点击 OK 按钮后，如果输入的密码与保存的密码不匹配，则弹出 Toast 提示；

- 点击 CLEAR 按钮后，清除密码输入框的内容。

d、出于学习的目的，我们使用 **SharedPreferences** 来保存密码，但是在实际应用中我们会用更加安全的机制来保存这些隐私信息，更多可以参考[链接一](#)和[链接二](#)。

3、然后，实现一个文件编辑 activity：

a、界面底部有两行四个按钮，第一行三个按钮高度一致，顶对齐，按钮水平均匀分布。按钮上方除了 ActionBar 和 StatusBar 之外的空间由标题和两个 EditText 占据，文件内容编辑的 EditText 需要占据除去其他控件的全部屏幕空间，且内部文字竖直方向置顶，左对齐；

b、在文件名输入框内输入文件名，在文件内容编辑区域输入任意内容，点击 SAVE 按钮后能够保存到指定文件，成功保存后弹出 Toast 提示；

c、点击 CLEAR 按钮，能够清空文件内容编辑区域内的内容；

d、点击 LOAD 按钮，能够按照文件名从内存中读取文件内容，并将文件内容写入到编辑框中。如果成功导入，则弹出成功的 Toast 提示，如果导入失败（例如：文件不存在），则弹出读取失败的 Toast 提示。

e、点击 DELETE 按钮，能够按照文件名从内容中删除文件，删除文件后再载入文件，弹出导入失败的 Toast 提示。

4、特殊要求：进入文件编辑的 Activity 之后，如果点击返回按钮，则直接返回 Home 界面，不再返回密码输入界面。

3、基础知识

3.1 SharedPreferences 的使用

3.1.1 SharedPreferences 的读取

在 Android 中，我们用来获取 SharedPreferences 里面存储数据的接口是 [getSharedPreferences\(String, int\)](#) 函数。两个参数的意义如下：

String: Desired preferences file. If a preferences file by this name does not exist, it will be created when you retrieve an editor.

int: Operating mode. Use 0 or MODE_PRIVATE for the default operation.

我们对 SharedPreferences 的读取操作是通过 [getSharedPreferences\(String, int\)](#) 函数返回的 SharedPreferences 对象的方法来完成的。查阅[文档](#)可以看到，SharedPreferences 支持如下几种方法读取之

前存储的数据：

```
abstract Map<String, ?> getAll()
abstract boolean getBoolean(String key, boolean defValue)
abstract float getFloat(String key, float defValue)
abstract int getInt(String key, int defValue)
abstract long getLong(String key, long defValue)
abstract String getString(String key, String defValue)
abstract Set<String> getStringSet(String key, Set<String> defValues)
```

所有的方法都需要传入一个 defValue 函数，在给定的 key 不存在的时候，SharedPreferences 会直接返回这个默认值。

3.1.1 SharedPreferences 的写入

所有对 SharedPreferences 的写入操作，都必须通过 SharedPreferences.edit()函数返回的 Editor 对象来完成。举例如下：

```
Context context = getActivity();
SharedPreferences sharedPref = context.getSharedPreferences("MY_PREFERENCE",
Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putInt("KEY_SCORE", newHighScore);
editor.commit();
```

3.2 Android 中的文件操作

Android 中的存储空间分为两种：Internal Storage 和 External Storage。

3.2.1 Internal Storage

默认情况下，保存在 Internal Storage 的文件只有应用程序可见，其他应用包括用户自身是无法访问这些文件的。

向 Internal Storage 写入文件的示例代码如下：

```
try (FileOutputStream fileOutputStream = openFileOutput(FILE_NAME, MODE_PRIVATE)) {
    String str = "Hello, World!";
    fileOutputStream.write(str.getBytes());
    Log.i("TAG", "Successfully saved file.");
}
catch (IOException ex) {
    Log.e("TAG", "Fail to save file.");
}
```

如果对应的文件不存在，openFileOutput(String, int)函数就会直接新建一个文件，这里需要注意文件名里不能含有路径分隔符（也就是'/'），这个函数会返回一个 FileOutputStream 对象，可以调用里面的 write 方法将文本写文件中；

与之对应的，文件的读取可以使用 openFileInput(String)函数来读取文件，这个函数返回一个 FileInputStream

对象，调用里面的 read 方法可以读取文件中的内容。示例代码如下：

```
try (FileInputStream fileInputStream = openFileInput(FILE_NAME)) {
    byte[] contents = new byte[fileInputStream.available()];
    fileInputStream.read(contents);
}
catch (IOException ex) {
    Log.e("TAG", "Fail to read file.");
}
```

3.2.2 External Storage（不需要实现）

Android 支持使用 Java 文件 API 来读写文件，但是关键在于要有一个合适的路径，如果你要存储一些公开的，体积较大的文件（例如：媒体文件），External Storage 就是一个比较合适的地方。

无论是否支持外置 SD 卡，所有的 Android 设备都会将存储空间分为 Internal Storage 和 External Storage 这两个部分。

要往 External Storage 里面写入文件，需要在 AndroidManifest.xml 文件中声明权限：

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

之后调用 `getExternalFileDir(String type)` 或者 `Environment.getExternalStorageDirectory()` 来获取 SD 卡路径。两者的区别在于：前者指向的目录会在应用卸载的时候被删除，而后者不会。

上面这两个函数均会返回一个 File 对象，代表一个目录路径，使用这个 File 对象，再结合文件名，就可以创建 `FileInputStream` 和 `FileOutputStream` 来进行文件的读取与写入。

举例如下：

```
void createExternalStoragePrivateFile() {
    // Create a path where we will place our private file on external
    // storage.
    File file = new File(getExternalFilesDir(null), "DemoFile.jpg");
    try {
        // Very simple code to copy a picture from the application's
        // resource into the external file. Note that this code does
        // no error checking, and assumes the picture is small (does not
        // try to copy it in chunks). Note that if external storage is
        // not currently mounted this will silently fail.
        InputStream is = getResources().openRawResource(R.drawable.balloons);
        OutputStream os = new FileOutputStream(file);
        byte[] data = new byte[is.available()];
        is.read(data);
        os.write(data);
        is.close();
    }
```

```

        os.close();
    }
    catch (IOException e) {
        // Unable to create file, likely because external storage is
        // not currently mounted.
        Log.w("ExternalStorage", "Error writing " + file, e);
    }
}

```

4、参考问题

4.1 如何使 EditText 占据指定大小的空间？

使用 LinearLayout 和 layout_weight 属性，更多请查阅[文档](#)。

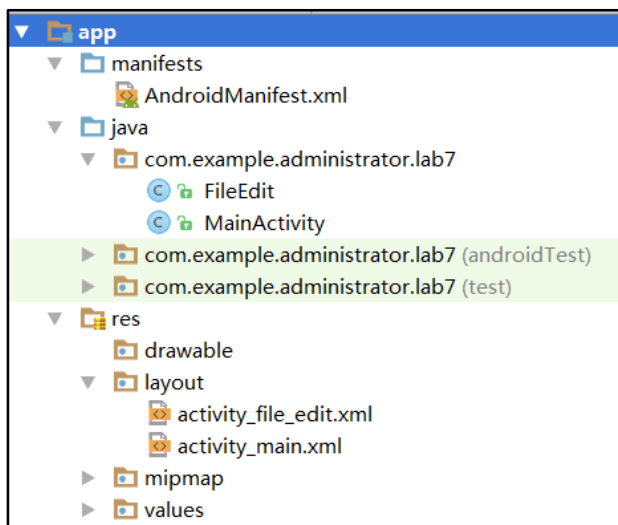
4.2 当 Activity 不可见时，如何将其从 activity stack 中除去（按返回键直接返回 Home）？

在 AndroidManifest.xml 中设置 noHistory 属性。

4.3 如何根据需要隐藏/显示特定的控件？

请查阅[文档](#)。

4.4 参考工程目录结构



5、检查内容

1、布局显示与示例一致。

2、应用逻辑与上文描述一致：

- 异常情况弹出 Toast 提示；
- 创建密码后重新启动应用，直接显示单个输入框；
- 文件编辑界面，能够正常增、删、改文件。

3、在实验报告中简要描述 Internal Storage 和 External Storage 的区别以及他们分别适用的场景。

4、不需要实现 External Storage。

6、提交说明

- 1、Deadline：下一次实验课前晚上 23:59；
- 2、提交作业地址：<ftp://edin.sysu.edu.cn>
- 3、文件命名及格式要求：学号_姓名_labX.zip（姓名中文拼音即可）
- 4、目录结构

```
15331111_huashen_lab1 --  
|  
-- lab1实验报告.pdf  
|  
-- lab1_code（包含项目代码文件）
```

提交之前先 clean！

