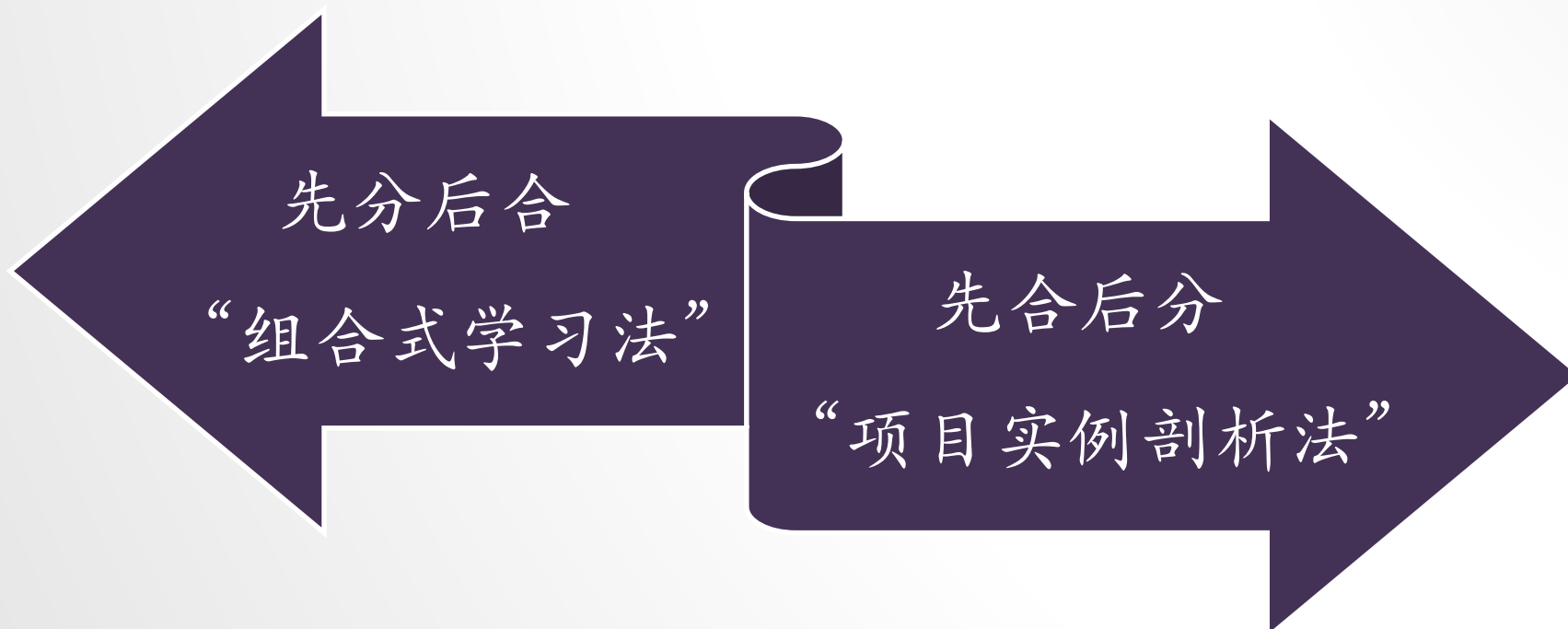


《面向对象软件开发实践（专业技能训练）》

实战：开发Android文件管理器

北京理工大学计算机学院
金旭亮

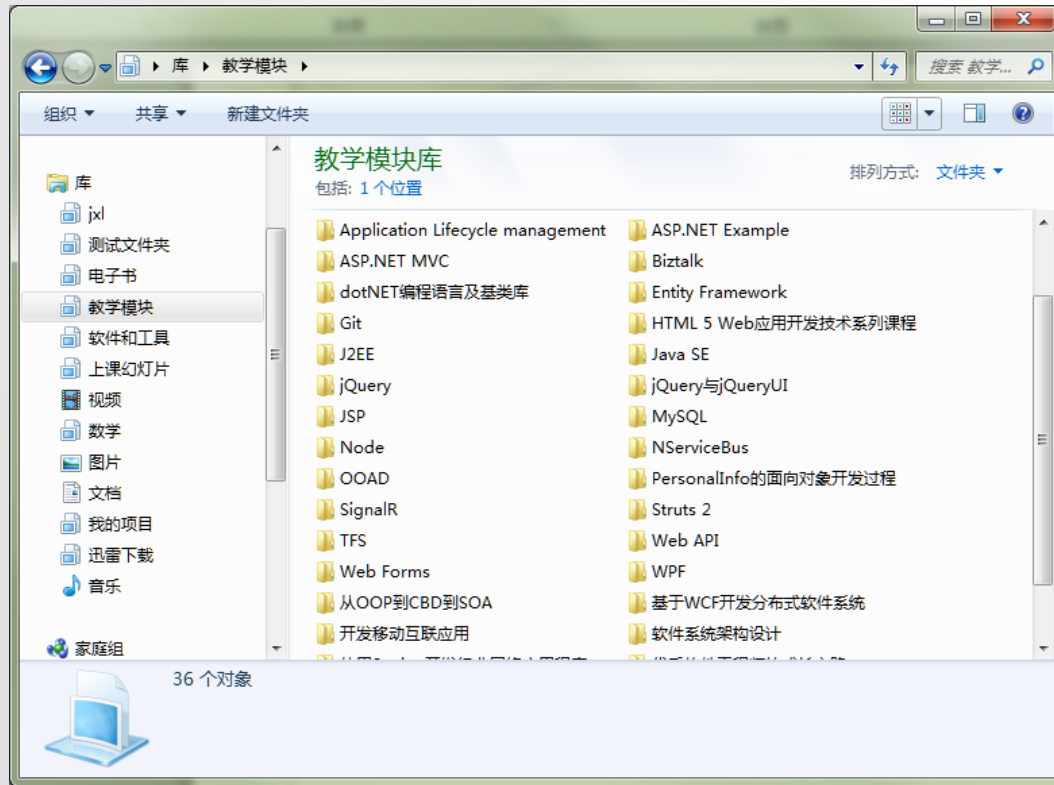
软件技术的学习方式



一、Android文件管理器系统分析与功能展示

需求分析-1：手机用户需求

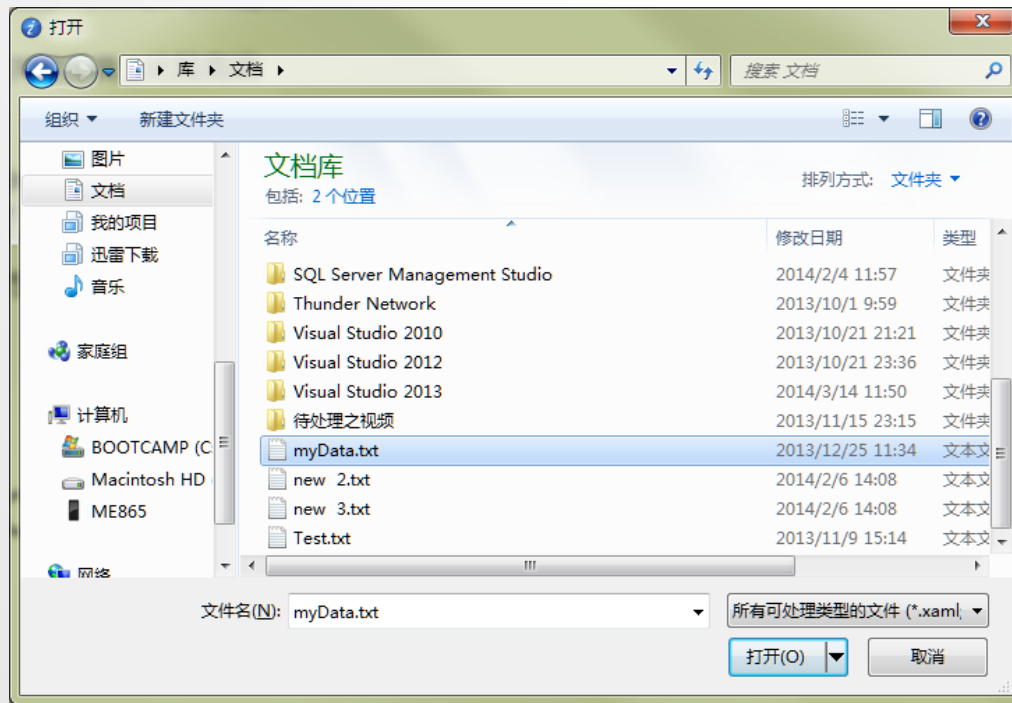
不知什么考虑，Android从1.0版本开始，本身没有提供一个类似Windows文件资源管理器那样的工具，让用户很好地查看和操作文件：



许多第三方文件管理器要不包容强制推送广告等垃圾信息，要不是就集成了过多的其实用户并不需要的功能，同样也不令人满意。

需求分析-2: Android应用的开发需求

- 在实际开发中，经常需要集成文件系统管理功能，比如让用户指定一个目录存储他的照片，或者从新插入的SD卡中读取内容。遗憾的是，Android同样没有提供象Windows那样“选择文件/文件夹”的标准对话框：



自己动手，
丰衣足食！

解决问题：写个自己的文件管理器

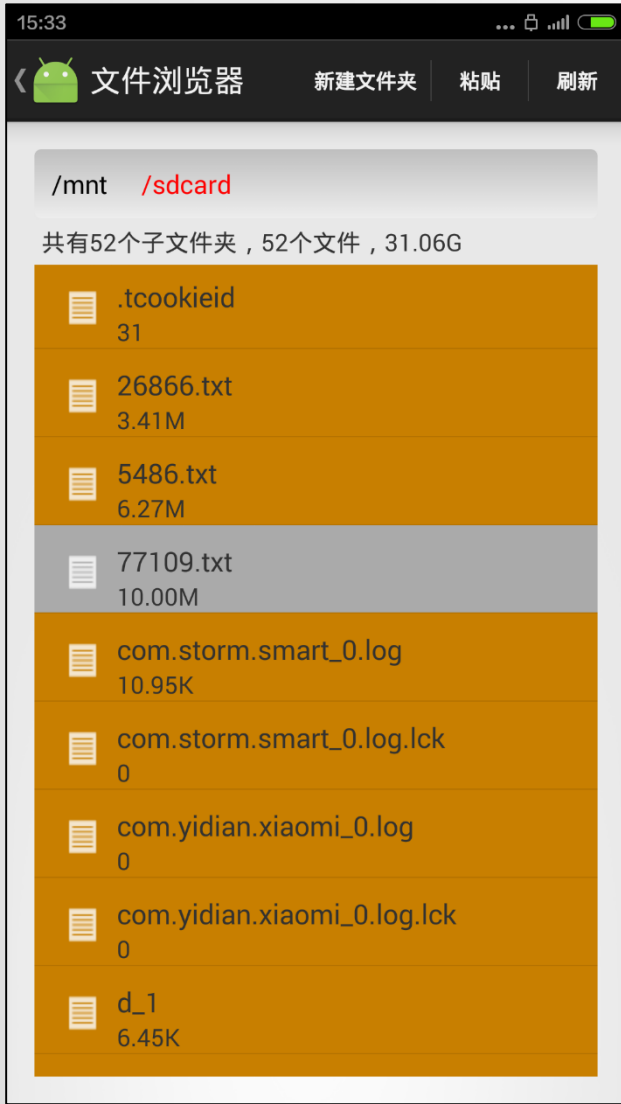
- 鉴于以上想法，确定要开发的文件管理器的功能集合



标准的文件管理功能：删除、改名、复制.....

用户选择一个文件之后，界面关闭，选择结果被返回给调用者

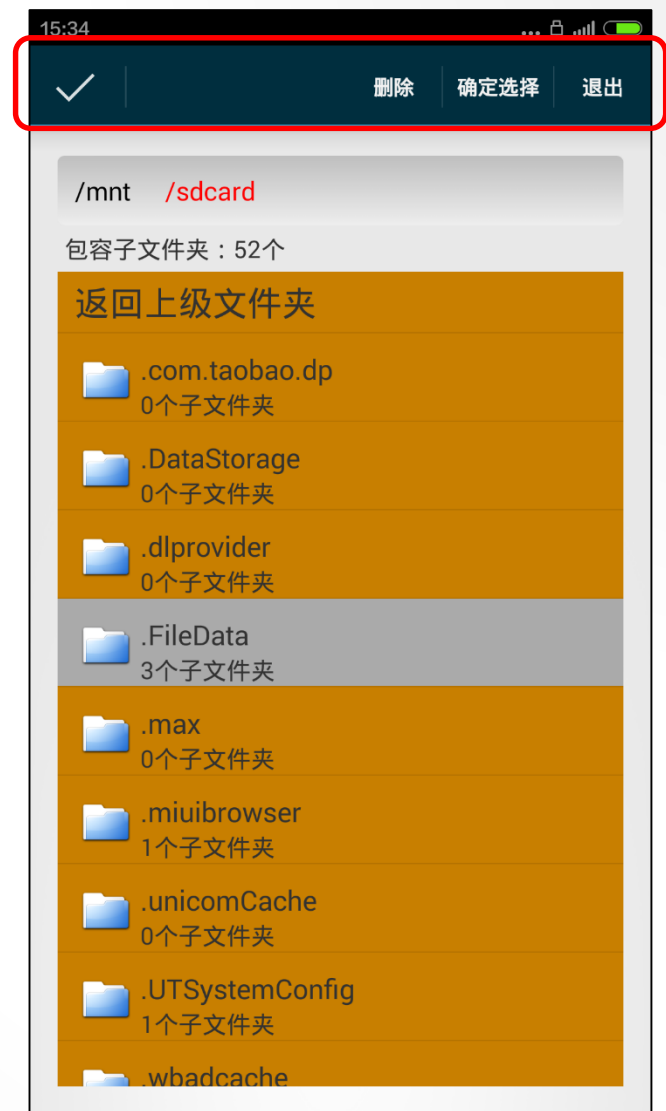
用户选择一个文件夹之后，界面关闭，选择结果被返回给调用者



常规模式



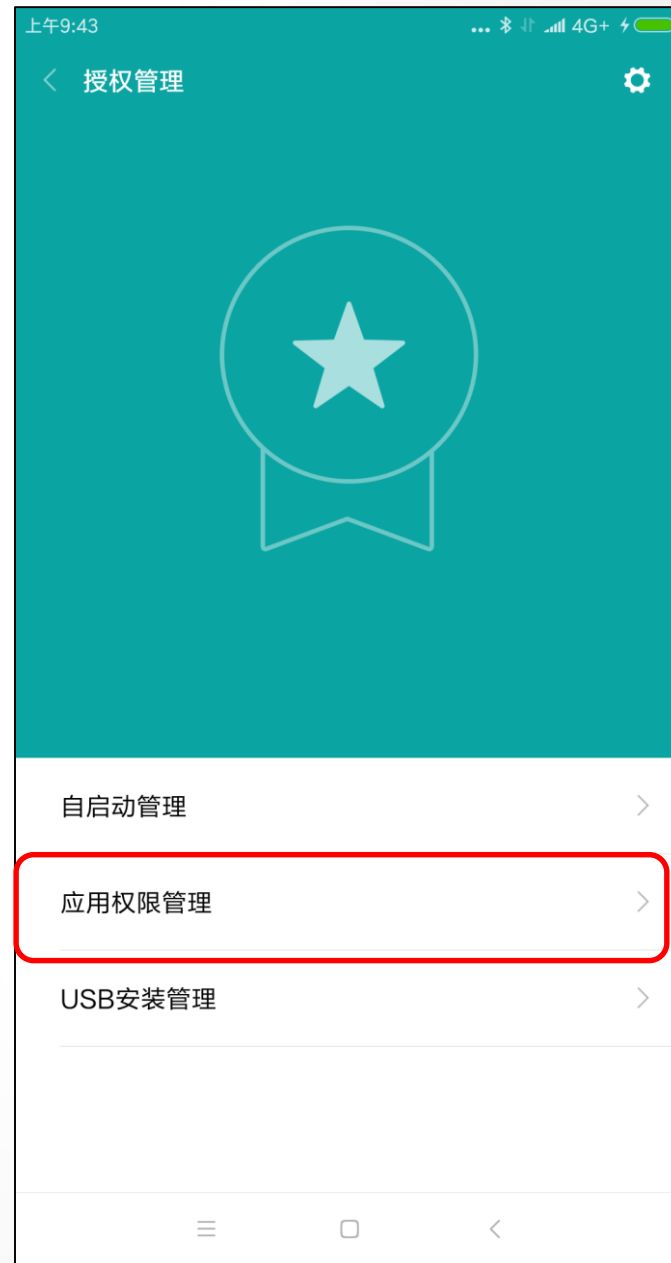
选择文件模式

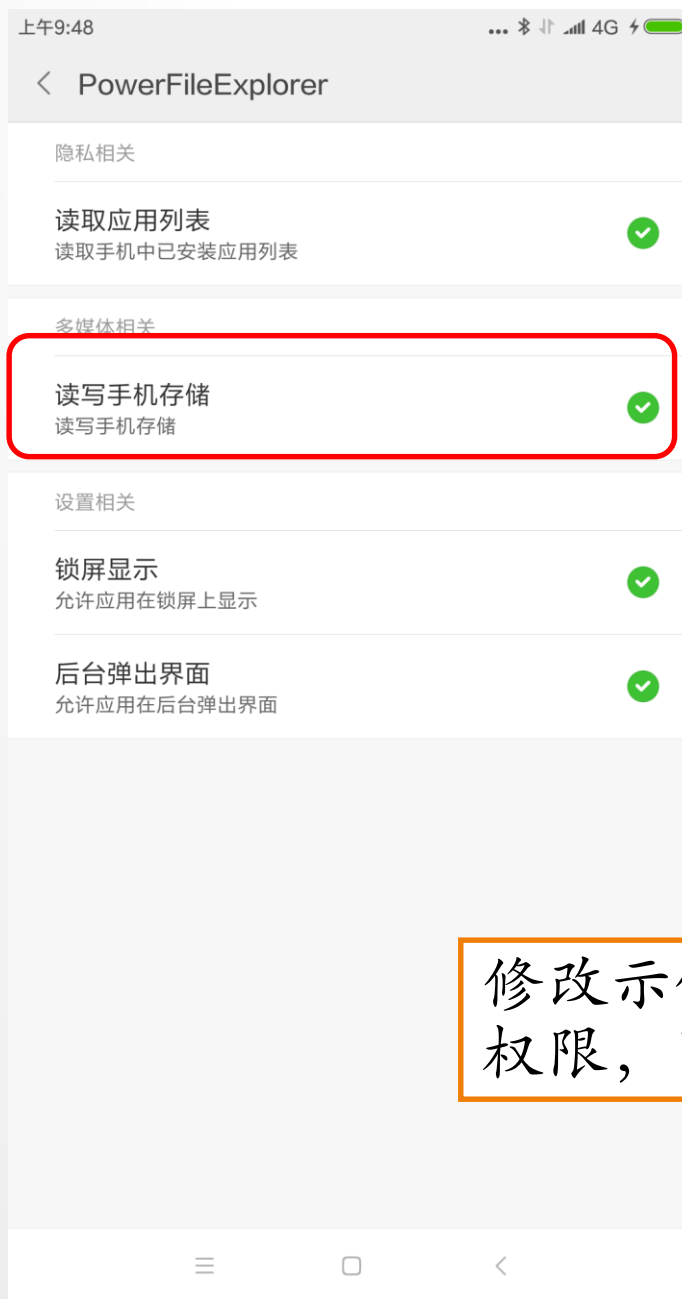
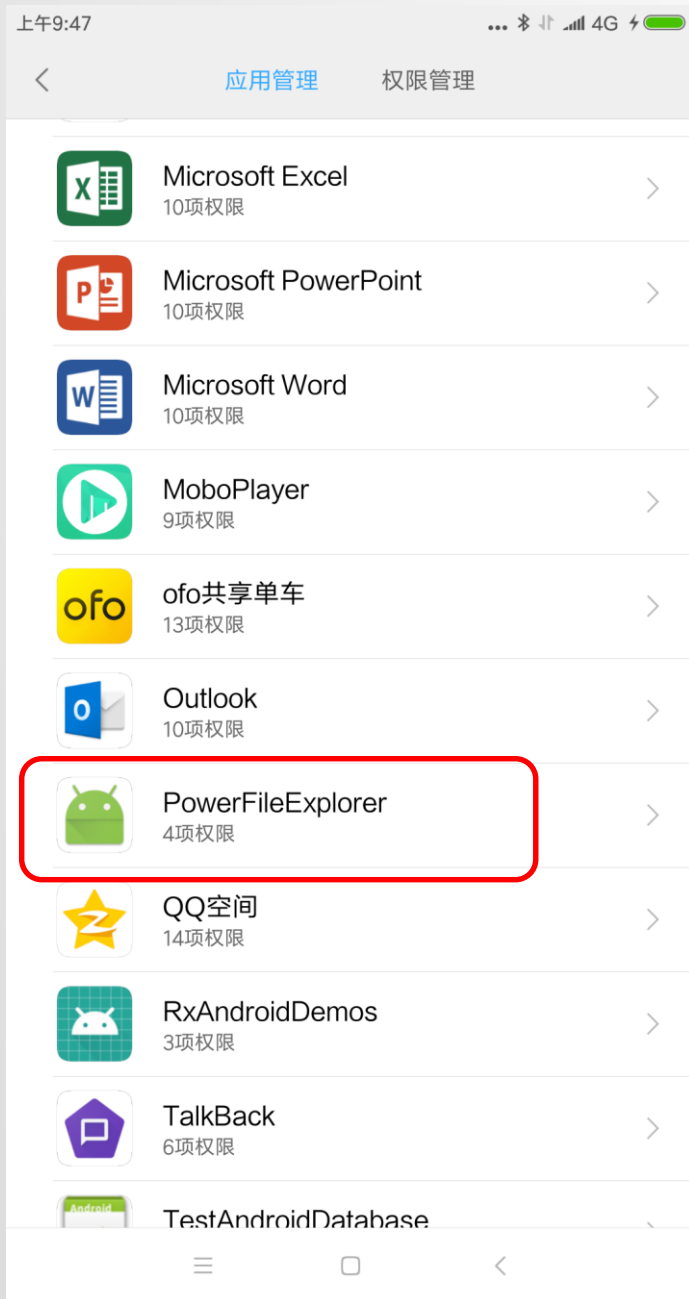


选择文件夹模式

高版本测试注意

对Android 6.0以上版本，访问SD卡需要在运行时动态申请权限。本示例未加入此功能，所以，如果想体验文件的改名复制等功能，请手动给示例添加权限，如左图所示。









选中此项权限

修改示例，添加运行时动态申请权限，留给大家作为练习。

数据结构设计

封装软件运行模式

- 本软件有三种功能模式，为此，定义以下枚举：

 E	cn.edu.bit.cs.myfileexplorer.model.ExplorerMode
	ExplorerMode NORMAL
	ExplorerMode CHOOSE_FILE_SINGLE
	ExplorerMode CHOOSE_DIRECTORY_SINGLE

封装提示信息

```
cn.edu.bit.cs.myfileexplorer.model.FileAndDirectorySummary
```

- String fullPath
- int dirCount
- int fileCount
- long totalFileLength

- String getFullPath()
- void setFullPath(String fullPath)
- int getDirCount()
- void setDirCount(int dirCount)
- int getFileCount()
- void setFileCount(int fileCount)
- long getTotalFileLength()
- void setTotalFileLength(long totalFileLength)
- String toString()

15:39

文件浏览器 新建文件夹 粘贴 刷新

/mnt /sdcard

共有52个子文件夹，52个文件，31.06G

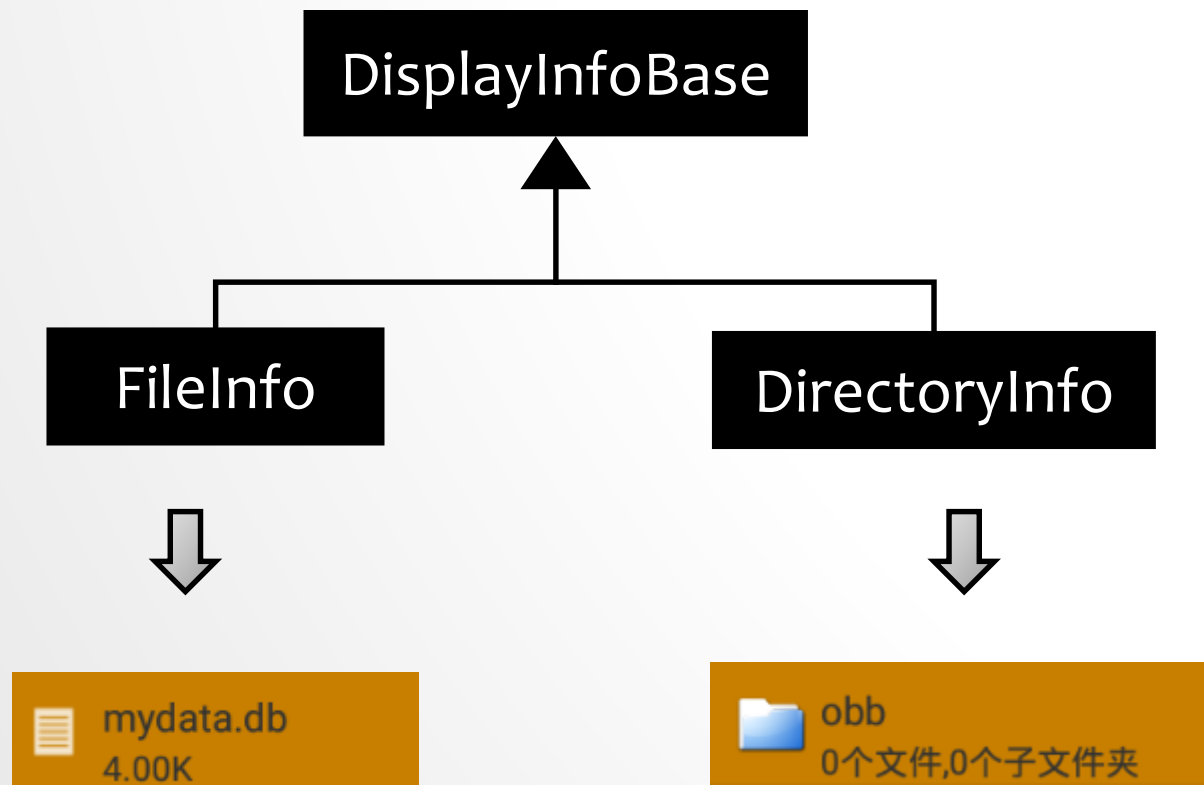
返回上级文件夹

- .com.taobao.dp
1个文件,0个子文件夹
- .DataStorage
1个文件,0个子文件夹
- .dlprovider
3个文件,0个子文件夹
- .FileData
0个文件,3个子文件夹
- .max
1个文件,0个子文件夹
- .miuibrowser
0个文件,1个子文件夹
- .unicomCache
1个文件,0个子文件夹
- .UTSystemConfig
0个文件,1个子文件夹
- wbadcache



封装文件和文件夹信息

文件与文件夹有许多信息是相同的，因此，给它们定义一个基类：





系统功能类的设计

为了提升代码的可维护性，
应该尽量地把功能代码从UI界面中剥离出来。

使用第三方组件，简化主Activity代码

使用Butter Knife简化Android开发：
<http://jakewharton.github.io/butterknife/>

```
@InjectView(R.id.btnTest)
Button testButton;

@InjectView(R.id.main_rdoChooseSingleFile)
RadioButton rdoSingleFile;

@InjectView(R.id.main_rdoChooseFolder)
RadioButton rdoSingleDirectory;
```

不再需要findViewById().....



```
@OnClick(R.id.btnTest)
void test() {...}
```

不再需要setOnClickListener().....

分离UI与功能代码

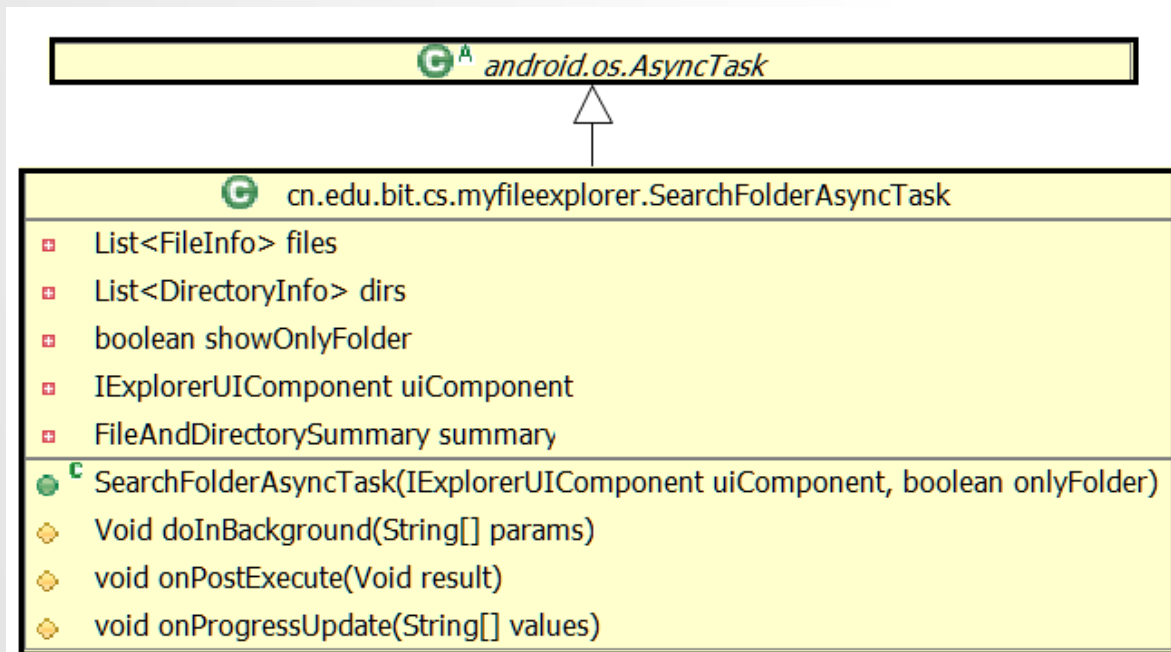
- 将功能代码从Fragment中抽取为独立的类，采用“Inject”的方式关联。

```
cn.edu.bit.cs.myfileexplorer.FileExplorerFunctionWrapper
  FileExplorerFragment fragment
  FileInfo fileToCopyOrCut
  ProgressDialog progressDialog
  boolean isCut
  FileExplorerFunctionWrapper(FileExplorerFragment fragment)
  void copyFile(FileInfo file)
  void cutFile(FileInfo fileInfo)
  void pasteFile(String destPath)
  void createDirectory(String curDirectory)
  void renameFileOrDir(DisplayInfoBase fileOrDir, String curDirectory)
  void deleteFileOrDir(DisplayInfoBase dirOrFile)
```

FileExplorerFragment

```
@Override
public void onActivityCreated(Bundle savedInstanceState) {
    setHasOptionsMenu(true);
    functionWrapper = new FileExplorerFunctionWrapper(this);
    super.onActivityCreated(savedInstanceState);
}
```

搜索功能必须采用异步调用方式实现



调用

FileManager

完成选择文件夹，查看并打开文件，修改文件（夹）名，删除文件（夹）的功能

分离SearchFolderAsyncTask与UI

为了把SearchFolderAsyncTask类与UI界面（FileExplorerFragment）分离，特定义以下接口，让FileExplorerFragment实现它。

```
I cn.edu.bit.cs.myfileexplorer.SearchFolderAsyncTask$IExplorerUIComponent  
void showFileAndDirectoryInfos(List<DirectoryInfo> dirs, List<FileInfo> files)  
void showProgress(String processInfo)  
void setFileAndDirectorySummaryFields(FileAndDirectorySummary summary)
```

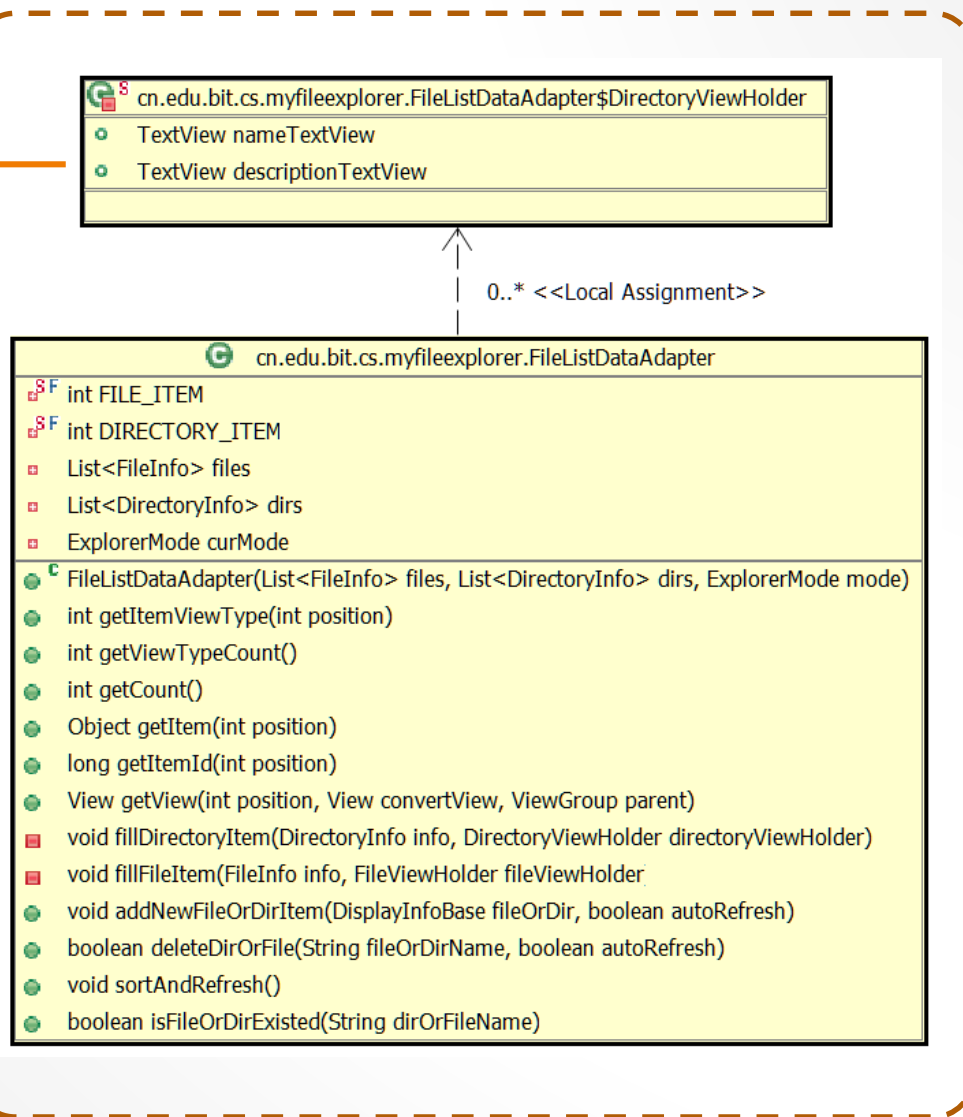


```
public class FileExplorerFragment  
    extends Fragment  
    implements  
        SearchFolderAsyncTask.IExplorerUIComponent,  
        HorizontalScrollList.ItemClickResponser {
```

支持两种数据行类型的数据适配器

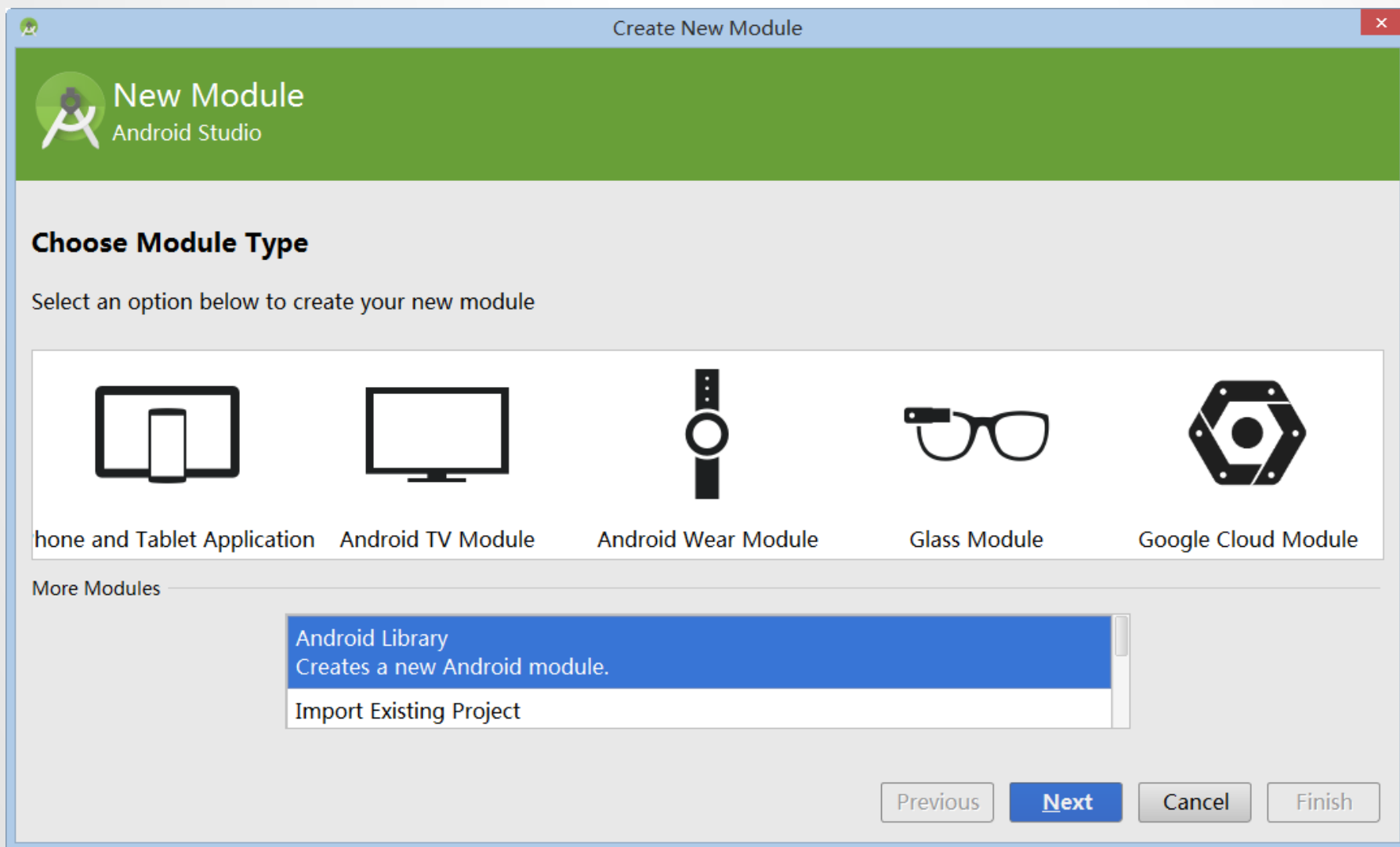
为了提升性能，内部使用了
ViewHolder设计模式，注意有两种
类型的ViewHolder

自定义一个DataAdapter，内部封装
文件和文件夹两个数据对象集
合，并且提供CRUD的功能。

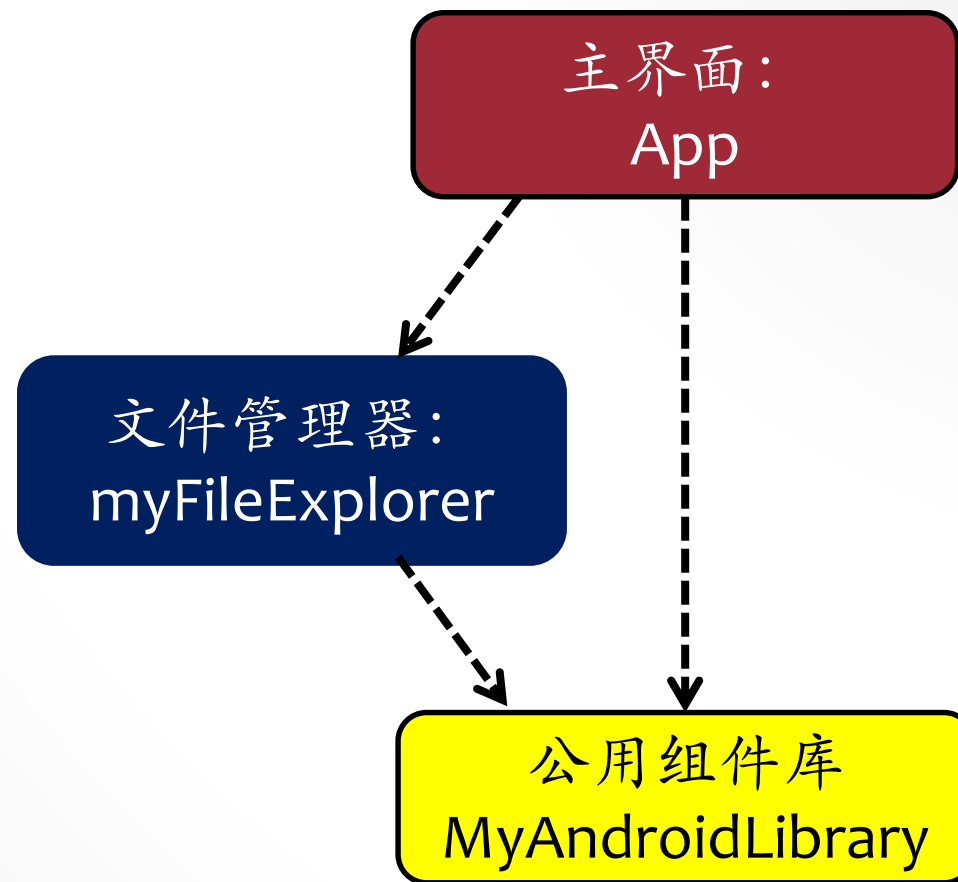
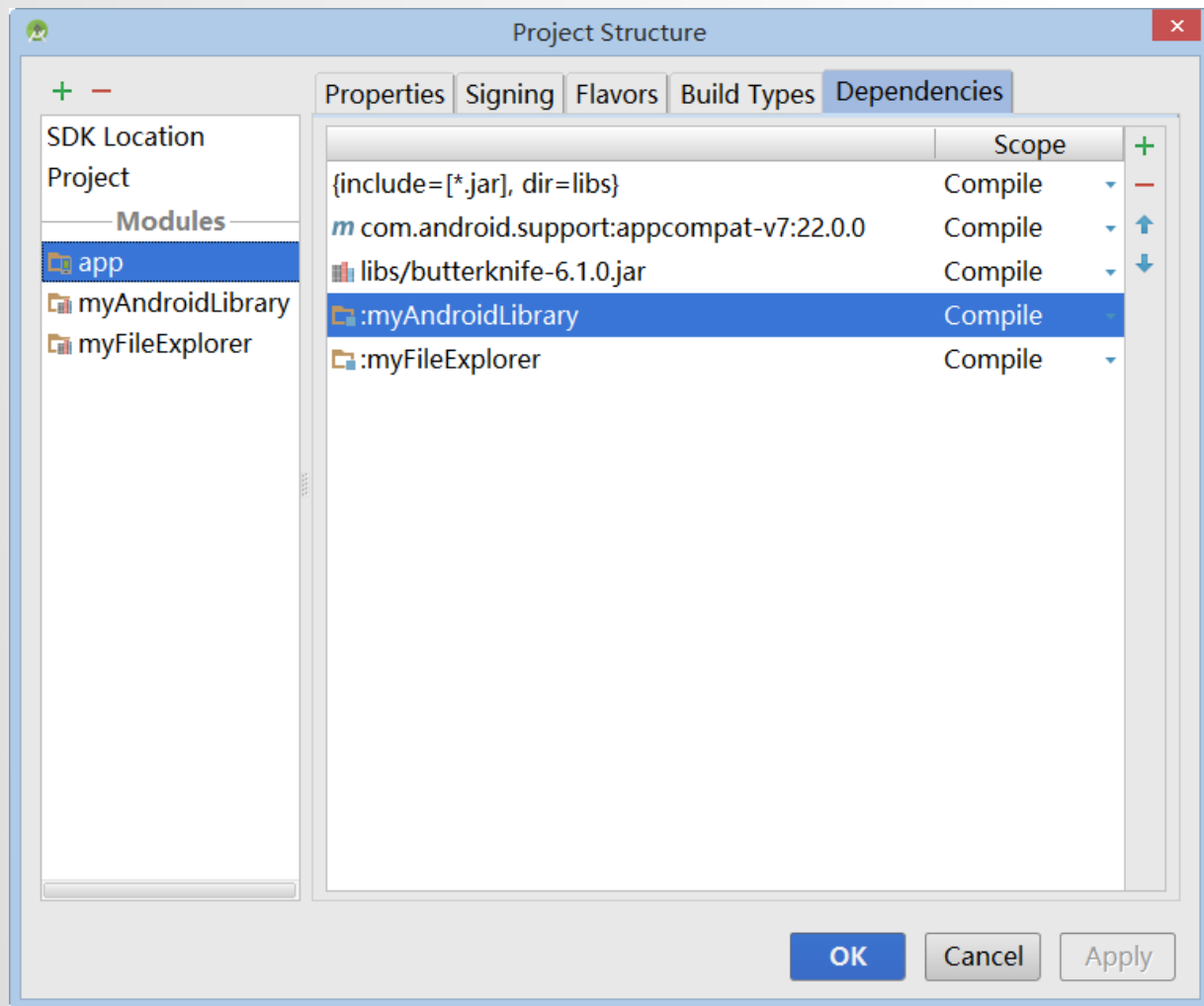


组件化的开发方式

使用Android Studio的Module实施组件化开发



文件管理器程序的三个模块





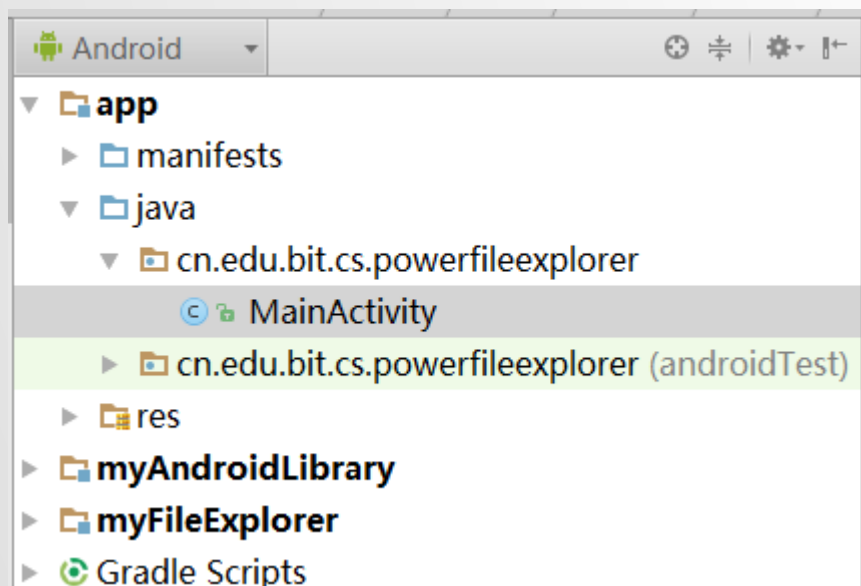
整个界面使用一个
Fragment实现之

这是一个**自定义控件**，派生自
HorizontalScrollView，实现水平滚动的列表

这是一个LinearLayout控件，
被加入到ListView的Header中

ListView控件，它的数据行的根控件是
自定义控件SelectableRelativeLayout，
实现了**Checkable**接口，以支持高亮显
示选中的文件

- 通过引用MyFileExplorer和myAndroidLibrary模块，主模块app的代码变得非常简洁，只需定义一个Activity即可：



在其他项目中重用模块

- 创建完项目之后，File→Import Module.....

