

南京大學

计算机科学与技术系

软件工程实验报告

实验名称: 分析与测试实验

学 号: 191220008

姓 名: 陈南瞳

指导教师: 张天

实验地点: 基础实验楼乙区 211

实验时间: 2021.12.2 ~ 2021.12.5

一、 实验名称

分析与测试实验

二、 实验目的

本次实验主要是为了掌握基本的程序分析方法和测试方法。从静态分析、白盒测试、黑盒测试三个方面出发，将实验设置为以下三个部分：

- 1、代码静态分析
- 2、单元测试
- 3、基于 GUI 的系统级测试

三、 实验要求

- 1、完成实验所需工具的安装使用，可得基础分 30 分。
- 2、完成静态分析，并报告当前代码存在问题，可得 10 分。如果对所报告问题进行改进可适当可额外加 20 分。
- 3、针对移动应用的某些模块编写单元测试用例（所测试方法不能重复），至少 2 个，可得 5 分，普通模式下运行单元测试模块的测试用例，覆盖度模式下（RunWithCoverage)运行单元测试模块的测试用例，可运行自己写的单元测试用例，也可以运行原 app 自带的单元测试用例，不少于 10 个，可得 15 分。
- 4、针对 app 的某些功能场景编写 GUI 测试测试脚本（所测试功能不能重复），至少 5 个，可得 45 分。

四、 实验环境

1、软件：

JDK: 1.8.0_231

Android SDK: 23





Gradle: 2.10

Android Studio: 4.2.1

Appium: 1.18.3

Appium-Python-Client: 1.2.0

2、硬件（安卓模拟器版本）:

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel 2 API 30		1080 × 1920: 420dpi	30	Android API 30 (G...	x86_64	9.4 GB	  

3、项目名称:

SimpleExplorer-2.3.1

AmazeFileManager-3.6.3

五、 实验内容

- 1、安装、熟悉并使用实验所需工具
- 2、静态分析
- 3、单元测试
- 4、GUI 测试

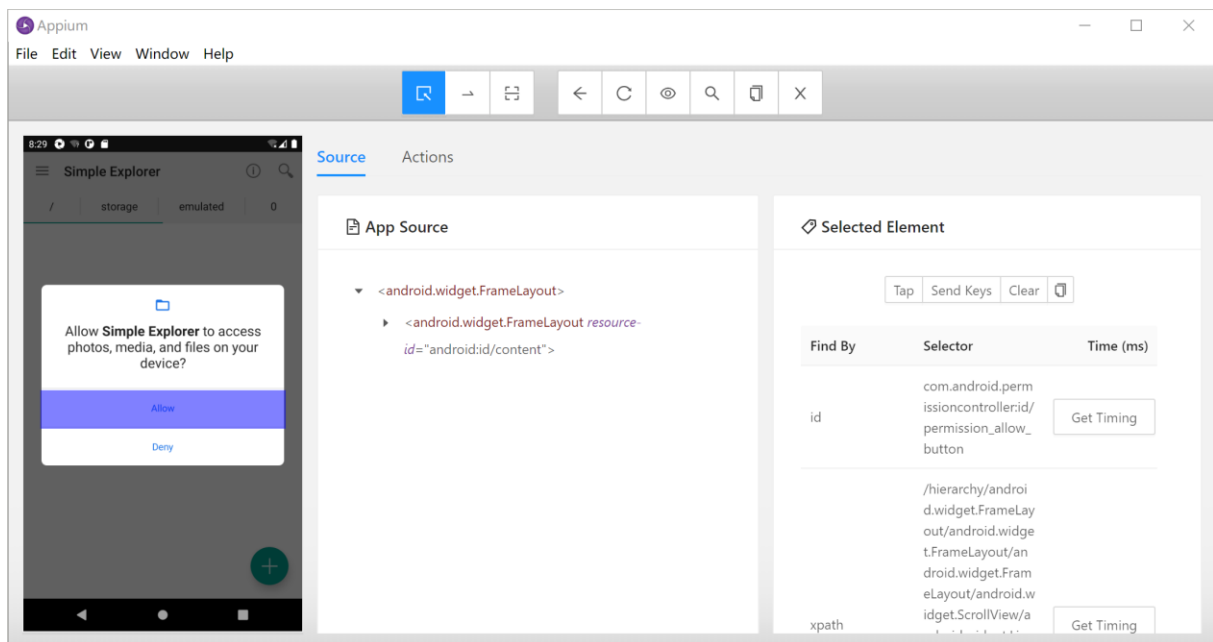
六、 实验步骤

- 1、静态分析
- 2、单元测试
- 3、GUI 测试

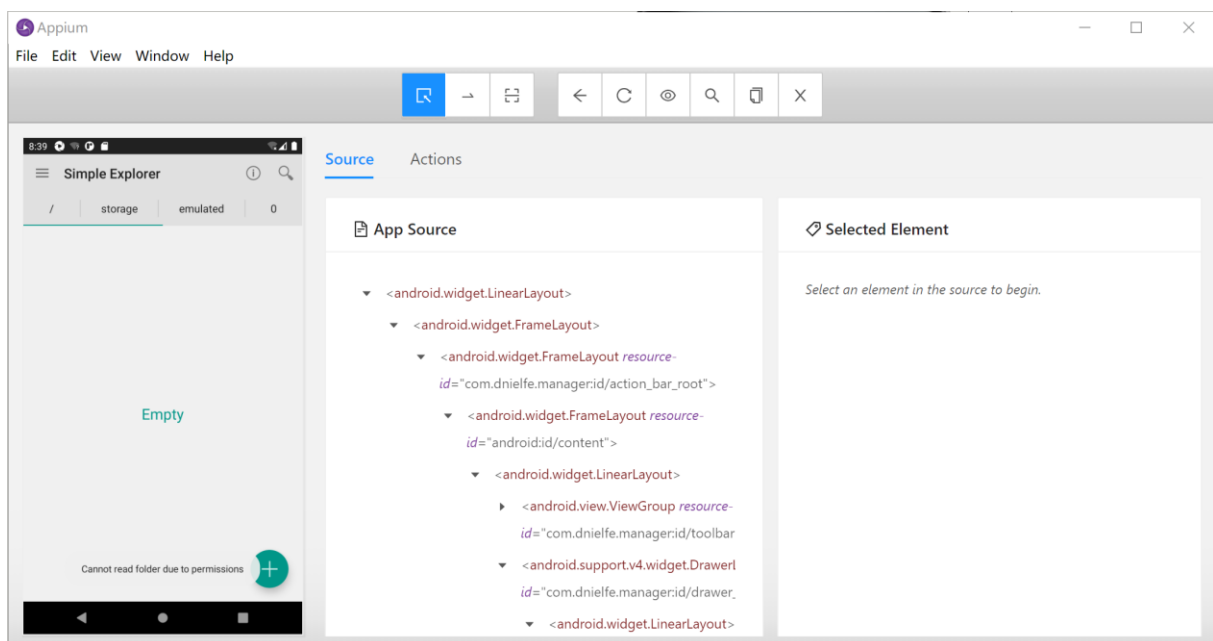
七、 实验结果

1、appium 应用使用截图

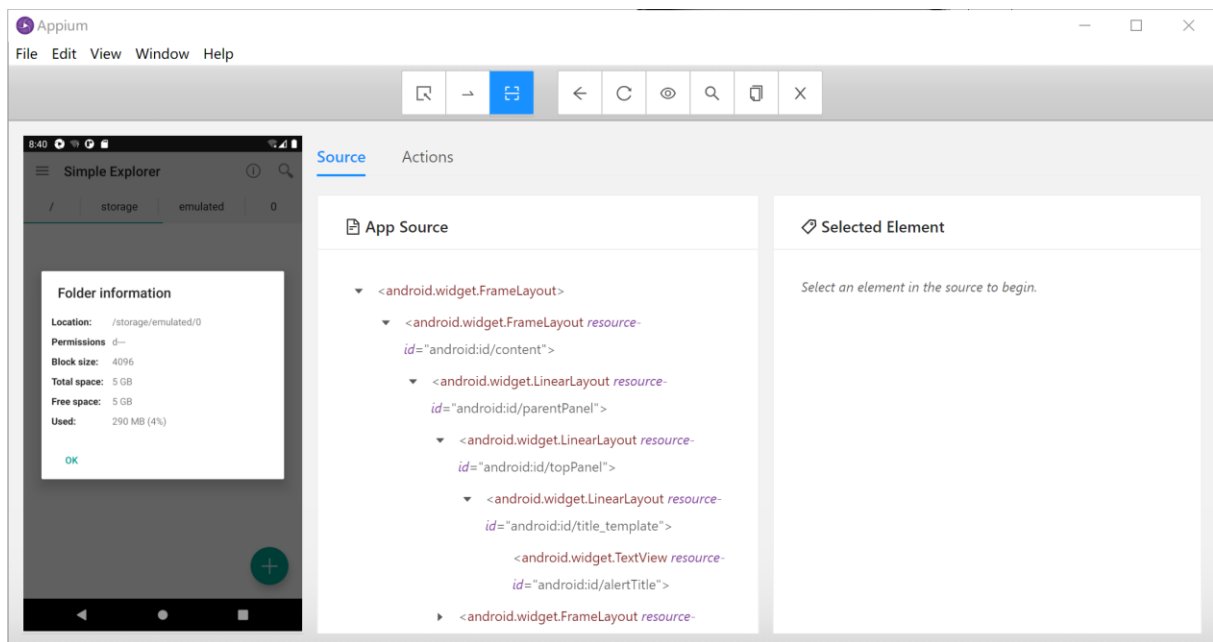
（1）点击控件



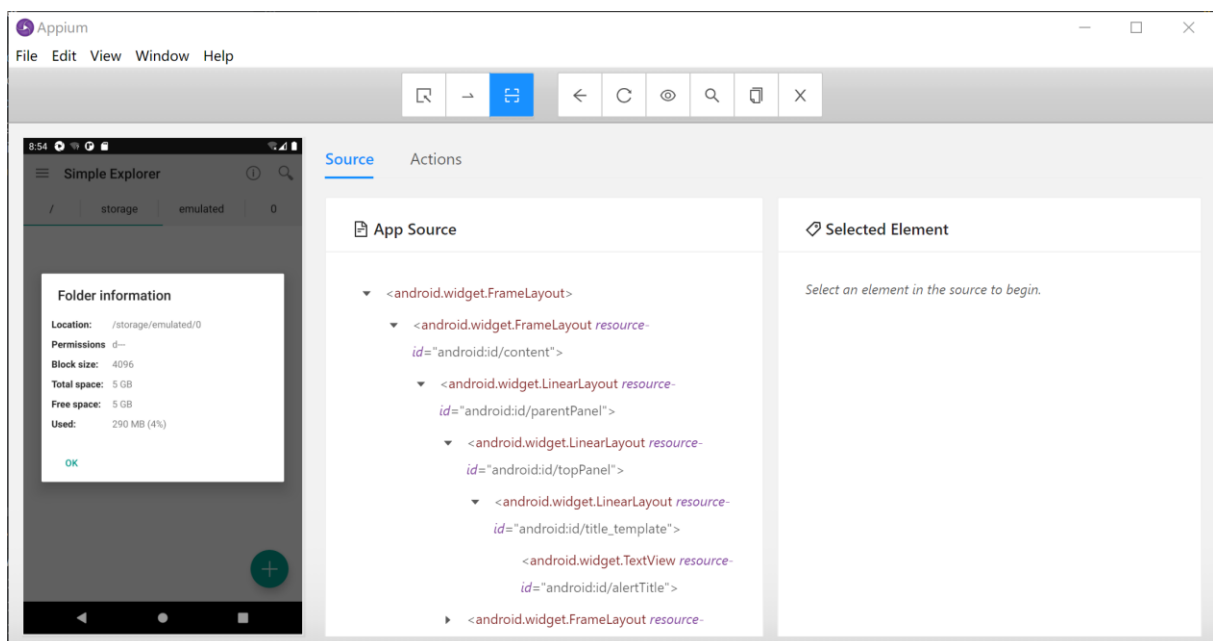
(2) 点击 Back



(3) 点击 Tap By Coordinates 后点击文件夹属性控件



(4) 点击 Refresh Source & Screenshot



2、静态分析

(1) 修改前:

类别	Error 数目	Warning 数目
Accessibility	0	1
Compliance	1	0
Correctness	0	33

Internationalization	0	73
Performance	0	160
Security	0	1
Usability	0	42

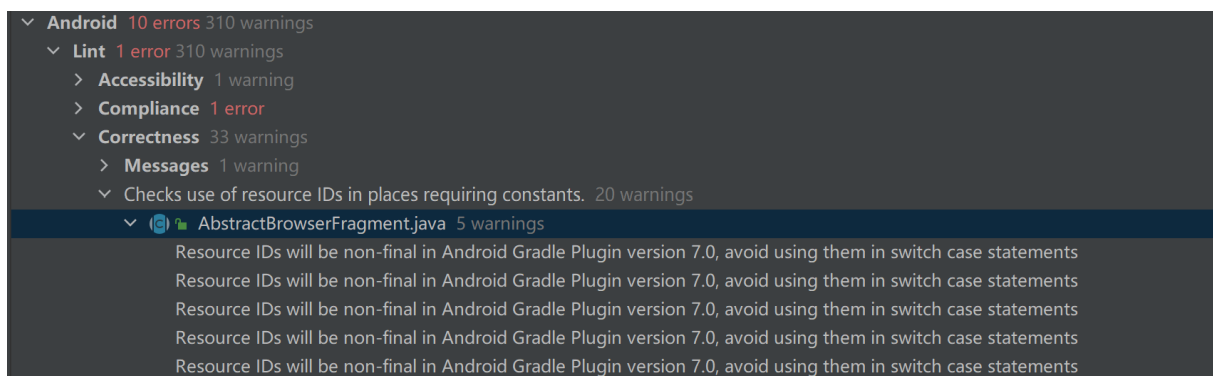
(2) 修改后:

类别	Error 数目	Warning 数目
Accessibility	0	1
Compliance	1	0
Correctness	0	28
Internationalization	0	49
Performance	0	160
Security	0	1
Usability	0	41

(3) 改进代码:

① Correctness:

在多个 java 文件中均出现了类似的 warning，google 官方给出的解决办法是把 switch 换成 if else，根据 google 官方的说法，这样的写法变更在 UI 中几乎没有性能损失。



于是找到 java 文件中的 switch 语句。

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.createfile:
            final DialogFragment dialog1 = new CreateFileDialog();
            dialog1.show(fm, AbstractBrowserActivity.TAG_DIALOG);
            return true;
        case R.id.createfolder:
            final DialogFragment dialog2 = new CreateFolderDialog();
            dialog2.show(fm, AbstractBrowserActivity.TAG_DIALOG);
            return true;
        default:
            return false;
    }
}

```

将每个分支均修改为 if else 即可。

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == R.id.createfile) {
        final DialogFragment dialog1 = new CreateFileDialog();
        dialog1.show(fm, AbstractBrowserActivity.TAG_DIALOG);
        return true;
    }
    else if (item.getItemId() == R.id.createfolder) {
        final DialogFragment dialog2 = new CreateFolderDialog();
        dialog2.show(fm, AbstractBrowserActivity.TAG_DIALOG);
        return true;
    }
    else {
        return false;
    }
}

```

② Internationalization:

在多个布局文件中均出现了类似的 warning，是因为 start/end 和 left/right 使用不匹配导致的。

```

v Android 10 errors 278 warnings
v Lint 1 error 278 warnings
  > Accessibility 1 warning
  > Compliance 1 error
  > Correctness 28 warnings
v Internationalization 73 warnings
  v Bidirectional Text 67 warnings
    > Padding and margin symmetry 12 warnings
    v Using left/right instead of start/end attributes 54 warnings
      v dialog_directory_info.xml 30 warnings
        Use "start" instead of "left" to ensure correct behavior in right-to-left locales
        Use "start" instead of "left" to ensure correct behavior in right-to-left locales
        Consider adding 'android:paddingEnd="10dp"' to better support right-to-left layouts
        Use "start" instead of "left" to ensure correct behavior in right-to-left locales
        Use "start" instead of "left" to ensure correct behavior in right-to-left locales
        Use "start" instead of "left" to ensure correct behavior in right-to-left locales
        Use "start" instead of "left" to ensure correct behavior in right-to-left locales
        Consider adding 'android:paddingEnd="10dp"' to better support right-to-left layouts

```

在布局文件代码中找到使用 left 的地方。

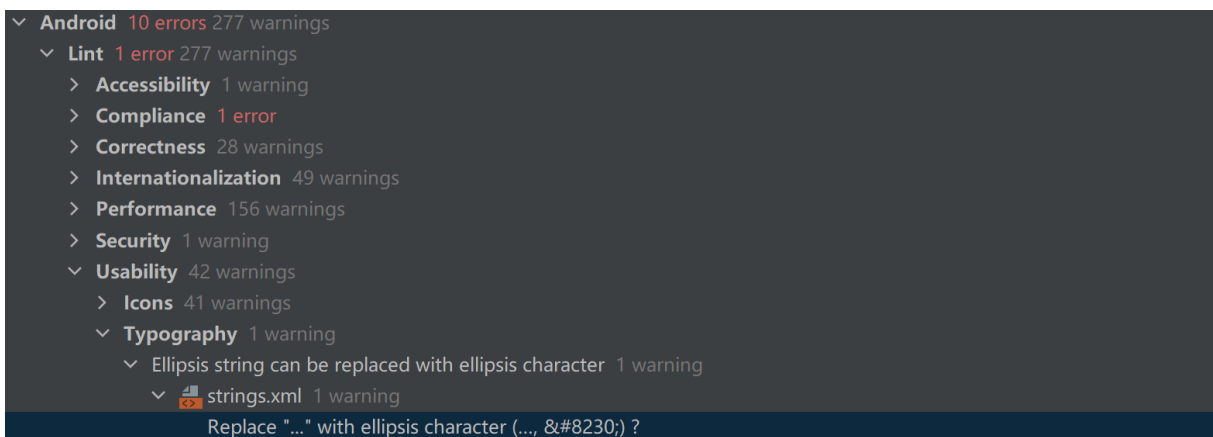
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left|center_vertical"
    android:ellipsize="end"
    android:gravity="left|center_vertical"
    android:paddingRight="10dp"
    android:singleLine="true"
    android:text="Location:"
    android:textIsSelectable="false"
    android:textAppearance="?android:attr/textAppearance"
    android:textStyle="bold" />
```

将每个 left 修改为 start 即可。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="start|center_vertical"
    android:ellipsize="end"
    android:gravity="start|center_vertical"
    android:paddingRight="10dp"
    android:singleLine="true"
    android:text="Location:"
    android:textIsSelectable="false"
    android:textAppearance="?android:attr/textAppearance"
    android:textStyle="bold" />
```

③ Usability

在一个 string.xml 中出现了省略号使用错误的报错，原因是大多数人的习惯用三个点来代替省略号。此处提示是要使用省略号：“…”。



在 string.xml 找到使用省略号的语句。

```
<string name="extractto">Çıkart...</string>
```


将三个点替换为省略号即可。

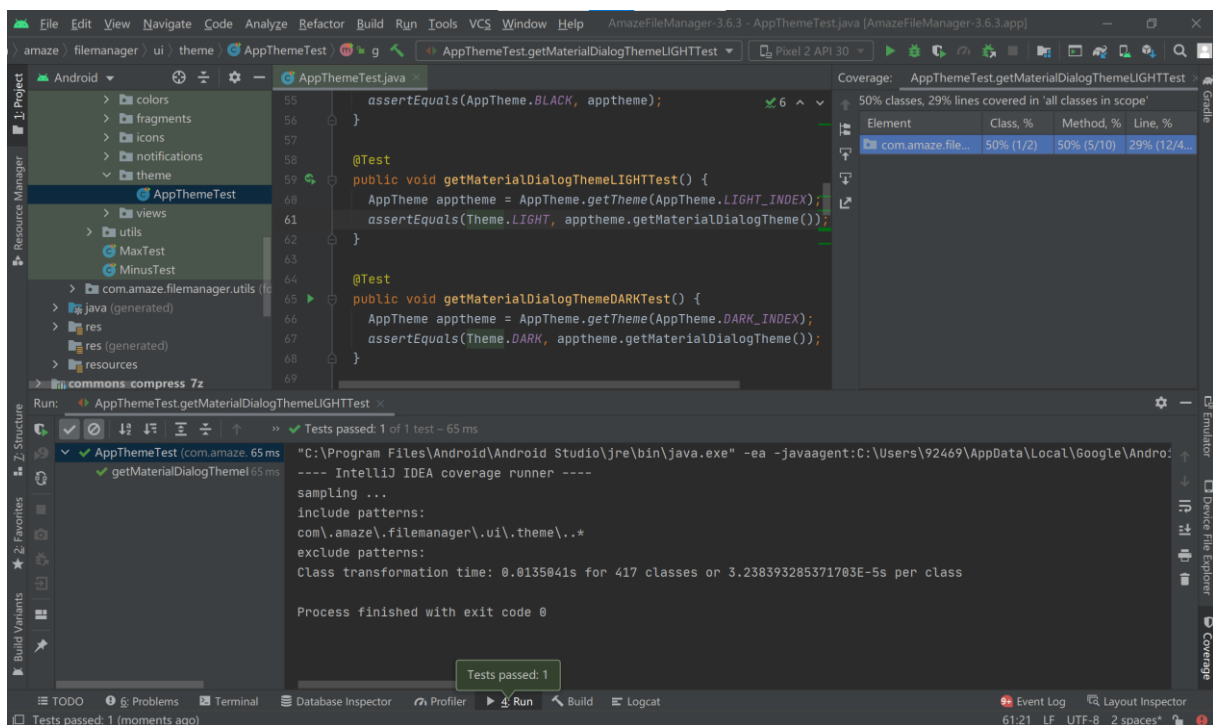
```
<string name="extractto">Çıkart...</string>
```

3、单元测试

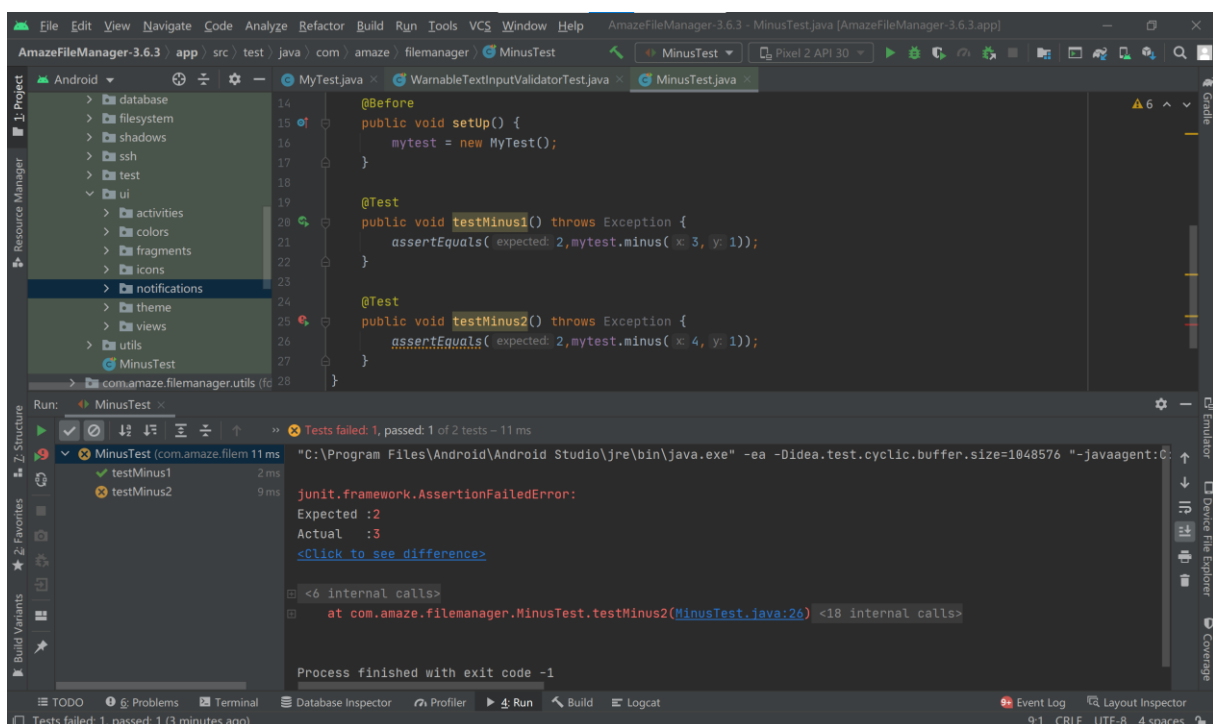
（因原来的 app 不方便，因此此处更换为 3 号 app）

所测试方法	所测试方法 功能描述	测 试 用 例 数 目	PASS 测 试 用 例 数 目	FAIL 测 试 用 例 数 目	测试覆盖 度(Class)	测试覆盖度 (Method)	测 试 覆 盖 度 (Line)
WarnableTextInputValidator. OnTextValidate() WarnableTextInputValidator. ReturnState() WarnableTextInputValidator. performClick()	文本输入	3	3	0	100%	66%	75%
AppTheme.getTheme() AppTheme.getSimpleTheme() AppTheme.getMaterialDialogTheme()	获取 APP 主 题	1	1	0	100%	57%	34%
NotificationConstants. SetMetadata()	设置元数据	6	6	0	100%	100%	91%
Icons.getTypeOfFile()	获取文件类 型	3	3	0	100%	66%	52%
CloudSheetFragment.isCloudProviderAvailable()	判断云提供 商是否可用	1	1	0	100%	16%	6%
ColorUtils.colorizeIcons()	图标颜色填 充	1	1	0	100%	100%	94%
TextEditorActivity. FindViewById()	通过 ID 获取 视图	6	6	0	50%	40%	27%
BookSorter.compare()	比较字典序	1	1	0	100%	100%	80%
ComputerParcelable.toString()	字符串转换	1	1	0	100%	50%	66%
MyTest.minus()	减法	2	1	1	100%	50%	40%
MyTest.max()	最大值	2	1	1	100%	50%	60%

Pass:



Fail:



4、GUI 测试

（因实验三中并未修改功能场景，因此暂时忽略“在原始的 Apk 上可以运行，但是在更新后的 Apk 上无法运行”的要求，但基本用到了实验三中修改的控件）

测试场景	测试该场景的测试用例数目	测试该场景的测试用例代码行数
文件搜索	3	58
查看文件属性	3	49
设置	2	32
新建文件（夹）	5	103
文件压缩	4	90
文件待粘贴	3	41

注：六个测试场景分别对应 testcase1.py ~ testcase6.py，在运行每个测试脚本前，须保证当前虚拟机处于原始状态（防止文件存储已被测试脚本修改，可在每次测试脚本运行前重新安装虚拟机。

八、 结果分析与结论

在本次实验中，对 APP 进行了代码静态分析、单元测试和基于 GUI 的系统级测试，较前几次实验而言更加有意思，初步了解如何从底层逻辑对 APP 进行软件测试，明白了一个完善的 APP 应该有一套完整的测试方法，以保证 APP 的稳定性和鲁棒性。

实验评分：_____

指导教师签字：_____

年 月 日