

基于 Android 测试框架的安卓单元测试

谭学芹

(广州软件学院, 广东 广州 510000)

摘要: 安卓是一种基于Linux内核的操作系统,从2008年安卓1.0发布到现在已经历经十几年,安卓系统已经成为手机系统的领军者。随着安卓操作系统的面世,基于它开发出来的移动APP层出不穷,其质量也参差不齐,为了保证上线的app尽量减少隐藏的缺陷,在此之前需要软件测试或开发人员对它进行足够的测试。通常对安卓应用的测试包括单元测试、功能测试、自动化测试等,如何对安卓应用进行单元测试或自动化测试?在实际行业中,真正对安卓应用测试非常精通的软件测试人员十分少。为了系统的阐述如何对安卓应用进行系统的测试,本论文将对安卓测试框架进行介绍,阐述测试安卓应用与测试普通的Java应用的区别,并通过简要的用例代码介绍如何利用安卓开发环境中的测试包对安卓组件进行单元测试。

关键词: 安卓app;单元测试;安卓测试框架;安卓组件

在一个安卓项目中,程序开发出来后,首先就要进行单元测试。单元测试是软件测试中最基本测试,且是最小粒度的测试,它是保证开发出来的软件质量的最基本手段。安卓应用的单元测试相对纯Java应用的单元测试又有很明显的不同。基础的Java应用程序可以直接通过JUnit单元测试框架直接开发对应的单元测试用例,但是安卓应用程序的测试则不能。安卓应用由四大组件构成,各个组件又需要与安卓操作系统进行交互,要测试与系统进行交互的各个组件通过JUnit已经无法办到,需要借助对应的安卓测试框架。

1 安卓测试与普通Java测试的区别

针对Java应用通过单元测试框架JUnit可以进行测试用例开发,开发测试用例主要经过三部曲:实例化被测对象、调用被测方法、断言结果^[1]。通常我们的被测对象依赖其他的协作对象,采用隔离测试的思想进行测试,主要策略包括两种:针对依赖的对象是大部件如数据库、服务器等采用粗粒度的Stub技术进行隔离测试;如果依赖的对象粒度较小,采用MockObjects技术进行测试^[2]。

安卓应用由不同的组件构成,对应组件的创建、启动等过程依赖安卓操作系统回调对应生命周期函数来进行。如果要测试安卓应用的各个组件功能逻辑,通过JUnit测试框架无法办到,如果只是测试安卓应用中与系统打交道的纯java代码暂且可以,但是要测试与系统进行交互的组件JUnit已经无能为力。那么如果要测试安卓应用程序该怎么办呢?

2 安卓测试基础

对安卓应用进行测试之前需要具备一些基础条件。

(1) 对被测系统组件深入了解。要对安卓应用进行高质量的单元测试的前提是了解被测应用组件。充分理解各类组件如Activity、Content Provider、Service及Broad Cast的工作原理及实现过程。只有这样才能针对它们的功能写出好的单元测试用例,这就是测试领域的知己知彼。

(2) 系统测试框架分析。了解安卓开发环境是否自带测试框架,了解测试框架对应的API的用法。

(3) 具备丰富的测试基础知识。掌握测试基础知识是一名软件测试人员应该具备的最根本的业务素质,应该熟练掌握通过边界值、等价类等方法设计测试用例,需要掌握单元测试的隔离测试思想等。

3 安卓组件

安卓四大组件包括Activity、Content Provider、Service及Broad Cast。要想对各类组件进行单元测试需要先深入了解它们。

3.1 Activity

Activity为活动页面,或者说是安卓应用的用户界面,不同的Activity之间通过信使Intent进行通信,在Activity上面会布局各种小组件如EditText、Text View、Button等,通过在Activity的逻辑文件中给各个组件设置各类监听事件来响应处理用户的操作。

3.2 Content Provider

为了实现进程之间的数据交换,Android提供了Content Provider,它是不同应用程序之间交换数据的标准API,当一个应用程序需要将自己的数据暴露给其他应用程序使用时,该应用程序通过提供Content Provider来实现,其他应用程序通过Content Resolver来操作Content Provider暴露的数据。这些数据可以存储在数据库、文件系统等。

3.3 Service

后台服务程序,可以认为是没有用户界面的Activity,Service一直在后台运行。它可以用来开发监控类程序或者一些无需交互的程序,如后台的定位服务。

3.4 Broadcast

广播也没有用户界面,但可以通过通知栏显示消息通知。系统或者应用通过NotificationManager通知其他组件,也可以启动一个Activity或Service来响应它们收到的消息。

以上就是安卓四大组件,它们是安卓应用程序测试的焦点,需要花足够的精力去测试它们,这些组件的测试离不开Android测试框架。

4 Android测试框架

安卓测试框架是安卓开发环境的一个组成部分,它能够用来测试安卓应用的各个方面,包括单元测试、UI测试等。下面为Android测试框架见图1:

Android测试框架最重要的角色就是Instrumentation,它充当了测试包和被测包之间的桥梁,可以将它们加载在同一个进程中。正因为它们处于同一个进程当中,所以测试代码完全可以调用Instrumentation提供的方法,可以修改和验证

控件的相关数据^[3]。此外Android测试框架主要集成在安卓开发环境的android.test和android.test.mock包中,通过里面的相关测试API可以完成安卓组件的单元测试、UI测试等。Android测试框架中的测试类关系图如下图2:

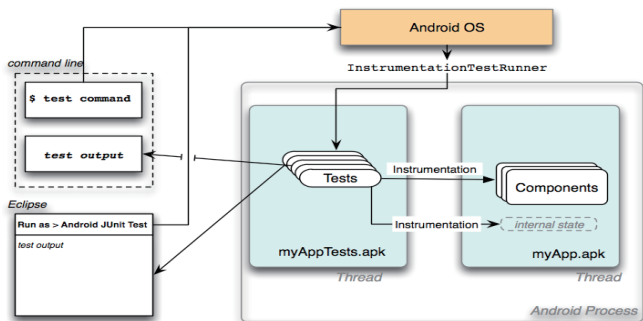


图1 Android测试框架

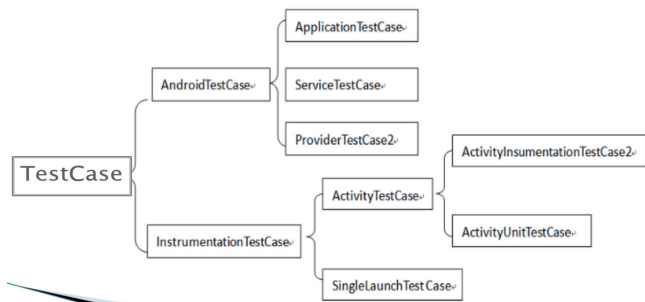


图2 测试类关系图

安卓测试用例类是在JUnit3的测试用例类TestCase基础上派生出来的,包括Android Test Case和Instrumentation Test Case及其子类。Android Test Case这个类主要用作安卓测试用例的基类,用它及其子类开发测试用例可以注入上下文Context,主要可以测试Application、Service、Content Provider。Instrumentation Test Case类是一系列可以使用Instrumentation类的直接或间接基类,通过它可以获取Instrumentation对象,从而在测试过程中控制组件的生命周期。

5 Activity测试

对安卓组件Activity进行测试,通常用到图2中的测试用例类Activity Instrumentation Test Case2和Activity Unit Test Case。前者是安卓测试时用到最多的类,它提供了单个Activity的功能测试。通过它可以测试活动页面的布局和交互,如页面组件是否存在、组件间的对齐方式、按钮是否能够完成功能等。通过ActivityUnitTestCase可以测试活动页面的行为,如活动页面能否正常启动、能否正常关闭。测试用例类中提供了相关方法用来判断测试过程中页面启动、关闭过程。

6 Content Provider测试

Content Provider内容提供者允许获取真实的用户数据,如果要测试它需要构造一个隔离的测试环境。在安卓测试框架中测试用例类Provider Test Case2类提供了一个隔离的测试环境,这个类允许使用Android Mock对象类,如Isolated Context和Mock Content Resolver。Isolated Context对象允许访问文件和数据库操作,但是与Android系统隔离。Mock Content Resolver对象作为测试的resolver使用,通过它可以

在测试环境中访问操作ContentProvider暴露出的数据操作接口。部分测试代码示例如下:

```
public class MyContentProviderTest extends Provider
TestCase2<MyContentProvider> {
    ...
    public void testDelete() {
        Uri uri = Uri.withAppendedPath(MyContentProvider.
CONTENT_URI, "dummy");
        final int actual = contentResolver.delete(uri, "_id = ?",
new String[] { "1" });
        final int expected = 1;
        assertEquals(expected, actual);
    }
    ...
}
```

7 Service测试

Service是Android四大组件中与Activity最相似的组件,主要区别是Service一直运行在后台,没有用户界面来与用户进行交互。服务组件也拥有自己的生命周期,但是它的生命周期回调函数与Activity有明显的不同。Service不能自己运行,需要通过一个活动页面Activity或其他上下文对象来调用,在安卓应用中启动服务包括两种方式:通过上下文的start Service()方法,这种方式启动的服务与调用者没有关联,调用者结束生命周期,启动的服务仍然在后台运行;第二种启动服务的方式通过上下文的bind Service()方法,这种方式启动的服务调用者与服务进行了绑定,一旦调用者退出,启动的服务就结束运行^[4]。

要测试Service,需要用安卓测试框架中的测试用例类ServiceTestCase,通过测试用例提供的启动服务的方法,可以在测试过程,测试服务能够正常启动。部分测试代码示例如下:

```
public class LocalServiceTest extends
ServiceTestCase<LocalService> {
    ...
    public void testStartable() {
        Intent startIntent = new Intent();
        startIntent.setClass(getContext(), LocalService.
class);
        assertNotNull(startIntent);
        startService(startIntent);
        assertNotNull(getService());
    }
    ...
}
```

8 结语

至此,我们发现要对安卓应用进行单元测试相对普通Java应用的单元测试存在很大的不同,需要借助基于安卓开发环境的安卓测试包,依托于安卓测试框架。作为一名开发安卓

(下转第100页)

II 经理世界

M131万能磨床磨锥柄及 24外圆。

2.3 可换刀头体加工工艺

X52立式铣床铣四方块(两件)。划钻孔及铣口线。X62卧式铣床用1mm刀片开镶刀片口。钳工钻绞个顶丝孔,倒角。热处理进行高频淬火之后发蓝处理。

2.4 定心轴加工工艺

C616车各部, 10大外圆留加长工艺用料,各部留磨量。M131万能磨床夹加长料处磨各部及拨稍磨定心尖。C616车去加长料。热处理进行高频淬火之后发蓝处理。

2.5 心轴备帽加工工艺

C620车外圆各部及内孔各部即M24×1.5丝和 8H7孔,并保证两孔同轴度在0.02mm以内。X52铣板口扁。热处理进行高频淬火之后发蓝处理。

2.6 心轴弹簧和钢球为标准件。

3 刀具的装配

优先的3可换刀头体上安装切割刀片,刀片可以根据所加工零部件的材质不同进行更换,运用M5标准螺栓在可换刀头体的侧面将刀片固定在刀头体上^[2]。将固定完整的刀头体装夹在横梁两侧的调整槽内,在底部用M6标准螺钉固定在刀具横梁体标准范围内,此横梁体可加工的范围在 40mm-300mm之间,横梁体上具有刻度标识,可以方便工人在使用的过程中根据零部件所需要加工的尺寸任意进行调整。并且此创新刀具拥有两个刀头体,一次可以加工出外圆与内孔,相对于传统的工艺,大大的提高了生产效率。

优先的2莫氏稍柄下端的螺纹穿过刀具横梁体的中间孔,将6心轴弹簧装入2莫氏稍柄底端中心孔内,然后装入7钢球,钢球与弹簧起到保护4定心轴灵活上下窜动及转动的作用。在加工中4定心芯轴起到了关键作用,它与主体装配后同心位置度必须保证在0.01以内,所以在制做定心芯轴的时候精度要保证超高的精度^[3]。最后用5心轴备帽将芯轴、横梁、莫氏稍柄紧密固定在一起。

4 刀具的运用

自制刀具应用过程如图4:

根据被加工工件材质安装相对应刀片,调整好刀片高度(保证刀片高度可以达到切削工件所需厚度)。按工件内孔及

外圆尺寸调整刀头尺寸,并运用螺丝紧固。将莫氏3号稍柄装入机床主轴孔中。床面铺垫10mm以上厚度橡胶板或其他垫板。(注意垫板不宜过硬,以免在刀片切透工件时对刀片磨损太大。裁剪与被加工工件相对应的正四方毛坯。(尺寸略大于被加工件大外圆尺寸3mm-5mm)

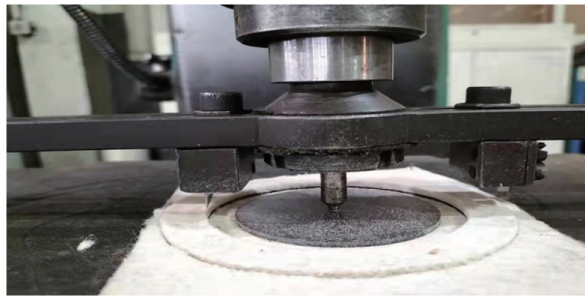


图4 定心芯轴

将正四方毛坯放在橡胶垫板上(较厚毛毡可在毛坯四角用小铁钉加以固定,以免刀具转动时带动工件转动)。

毛坯中心放置略小于被加工件内孔尺寸的圆压板。(圆压板要有中心孔,用于切割刀具加工工件时定心)。运用刀具定心轴顶住圆压板中心孔,开动机床,转速800r/min,进刀量转换成手动。对工件进行切割,割透后抬起刀具即可,工件加工完成。运用自制刀具与夹具加工完成的零件,与之前的对比效果明显。大大的提高了工作效率。

5 结语

刀具加工工件可以达到的效果,原来加工工件需要划线、用剪刀手动裁剪,加工的工件不但效率低下,而且工件边缘极不规整,不管是从外观还是尺寸精度都难以达到图纸要求。现在机床上运用双圆双刃刀具按被加工件图纸要求调整好两个刀体的尺寸后,将零部件外圆与内孔一次加工完成,工件的外观及尺寸精度都得以保证,提高了工作效率。

参考文献

- [1] 罗金龙. 梯形螺纹高速钢数控刀具制作 [J]. 金属加工(冷加工), 2018(3):54-56.
- [2] 巴新华, 聂福全. 摩擦焊工艺在刀具制作中的应用 [J]. 金属加工(热加工), 2008(22):43-44,49.
- [3] 张桐强. 数控机床 PLC 系统的设计、调试工作 [J]. 商品与质量, 2020(15):65.

(上接第98页)

单元测试的开发或测试人员需要充分掌握安卓组件的相关知识、需要掌握安卓测试框架以及安卓的相关测试类。安卓各类组件的测试用例需要继承测试包里对应的测试用例基类或者子类。为了提升安卓app的质量,安卓测试人员或开发人员需要不断提升个人技能,才能推动行业的不断发展和繁荣。

参考文献

- [1] 彭鑫, 谭彰, 黄文君, 等. 基于 Android 的工业控制监控软件设计

[J]. 新型工业化, 2012(5):32-38.

- [2] 宫向一. 基于 OSGi 的 Android 应用模块动态加载框架设计与实现 [D]. 中国海洋大学, 2015.
- [3] 李金明. 基于 ANDROID 平台的多参数心脏远程监测系统 [D]. 天津工业大学, 2017.
- [4] 刘萍萍. 基于 Android 的旅游记忆软件设计与实现 [D]. 山东科技大学, 2015.