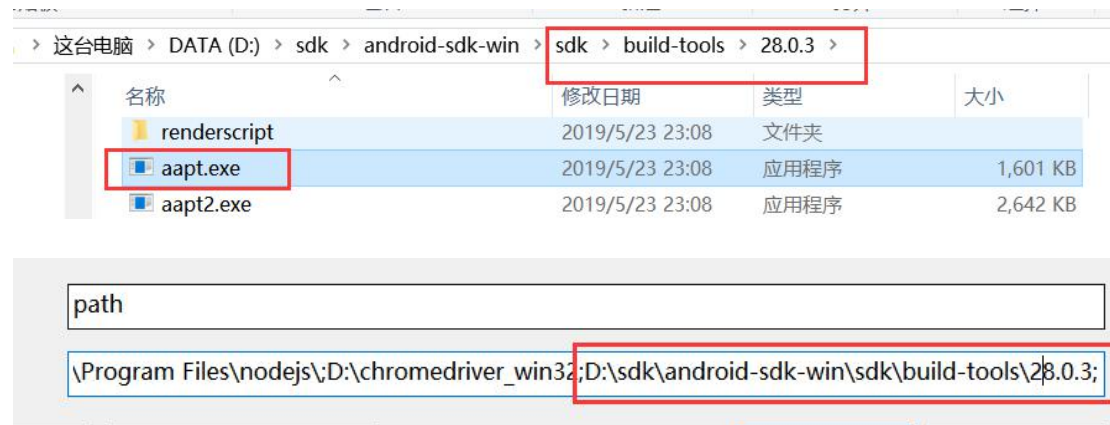


关于代码的说明：

- 1: 是 Windows 下面的代码，没来得及打包成 Linux 下面的；
- 2: 对于代码当中的 aapt，我是将 aapt 设置成为环境变量，于是在代码当中直接执行指令的，aapt 为 sdk 的 tools 里面的 aapt.exe（sdk 版本详见下面），助教可以添加自己的 aapt 到环境变量里面去（方法参考：<https://www.cnblogs.com/tzxy/p/11009027.html>），我不确定自己提供的 aapt 助教您是否是可以使用的：



- 3: 在测试的时候，我仅仅在自己的“雷电模拟器”上面进行过测试，所以不是很清楚其余安卓设备上测试会怎么样；

Jar 运行：

1: 运行环境：

```
C:\Users\asus>node -v
v6.17.1

C:\Users\asus>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)

C:\Users\asus>appium -v
1.4.16

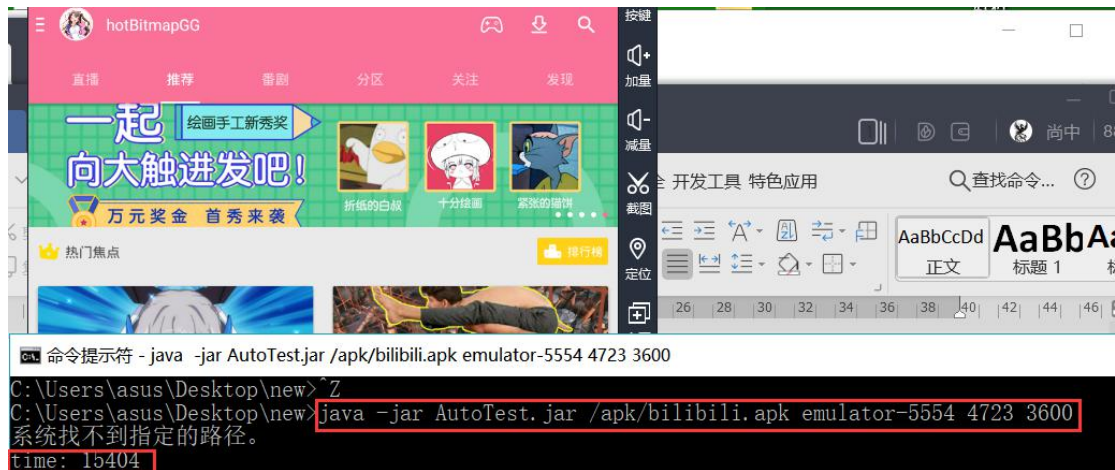
C:\Users\asus>adb
Android Debug Bridge version 1.0.40
Version 28.0.2-5303910
Installed as D:\sdk\android-sdk-win\sdk\platform-tools\adb.exe
```

2: 运行示意：

- (1)：运行出错：有的时候虽然看起来 appium 等等都在运行，但是也有可能是因为卡住了，无法进行下去。当“模拟器”/“手机”在 app 的一个页面停留超过 15s，就有可能是这种情况。这不是代码的锅啊~

遇到这种情况，可以重新运行，下次很有可能就不会出错了；

- (2)：运行示意图：



PS: 不需要关注命令行上面打印的内容, 那些只是我为了 debug 容易而打印的;

(3): 有的时候, app 当中会含有跳出外面的链接, 我已经很努力重启 app 了, 但是很有可能还有一些特殊的时候没有顾及到。所以如果跳出了 app, 测起来手机上面其余应用的时候, 希望你可以见谅。

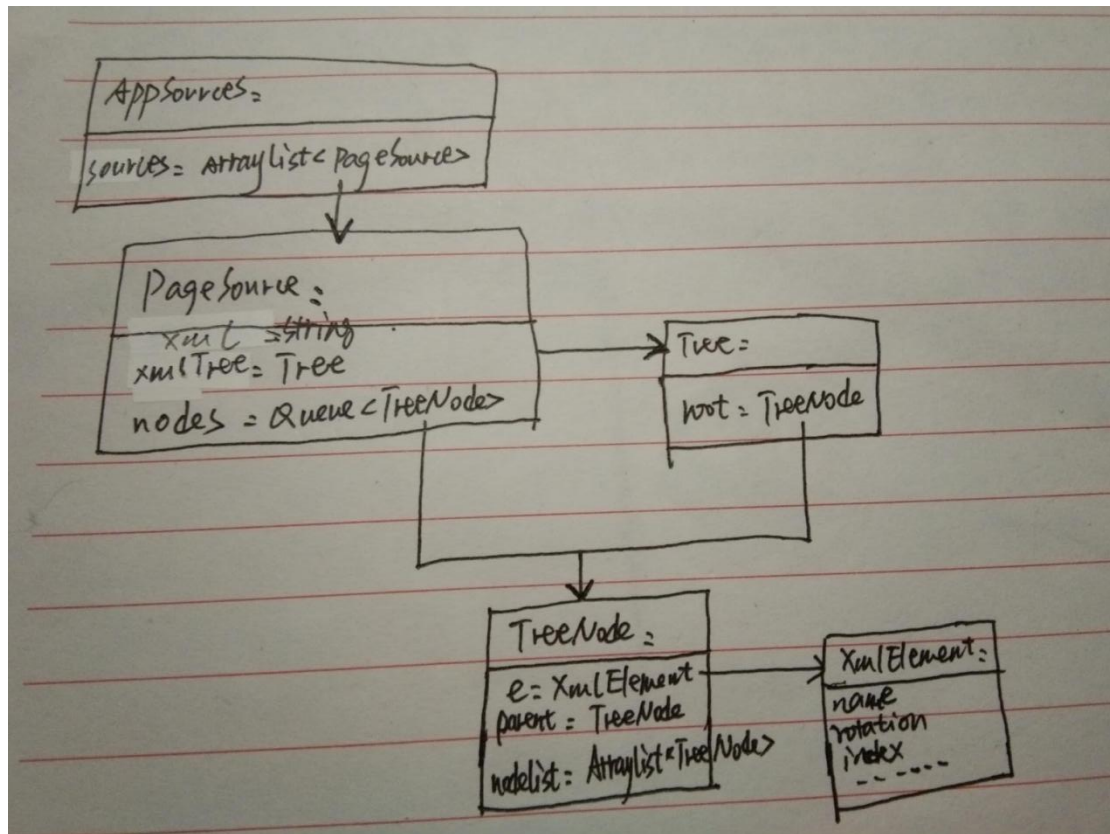
(4): 有的时候, 会出现重复测试的情况, 那可能是因为多个元素可以产生相同的效果, 而我每个元素都会测试一遍。也有可能是当前的 pagesource 和之前已经测试过的 pagesource 是不同的, 所以代码默认是重新再次测试这个 pagesource。

(5): 10 个 apk 我都测试了 ≥ 1 次, 所以如果有的 apk 的测试是无法进行的/刚开始就结束的, 那么可能不是我代码的问题。如果助教您愿意, 可以联系我, 我会看看是什么地方出问题。

代码思路:

1: 结构如下:

(1): 数据结构:



(2)：代码总体思路：

Main 中开始初始化，运行，service 当中含有需要的功能代码。

Main2 和 APP 两个 java 文件是测试的时候用的，没有什么重要的作用。

(3)：代码详细思路：

整体进程新建一个 AppSource 存储点击到达过的 app 页面的 source 情况，以便于下次再次到达相同页面的时候，跳过之前测试过的 element，直接测试其余的 element。代码如下：

```

driver.manage().timeouts().implicitlyWait(8, TimeUnit.SECONDS); // 设
/*
 * 余下的测试逻辑请按照题目要求进行编写
 */
PageSourceService pss = new PageSourceServiceImpl();
AppSources app = new AppSources();
pss.testTreeNodeInQueue(appPath, app, driver, startTime, runtime);
}
  
```

Main 中仅有一个 service 的实例，他是为了调用 `testTreeNodeInQueue` 方法进入测试的。

测试的整体思路是采用轮询的方式，当代码运行到指定时间的时候，会结束运行，否则会一直测试下去。代码如下：

```
// TODO Auto-generated method stub
while((new Date().getTime() - startTime) <= runtime) {
    long time = new Date().getTime() - startTime;
    System.out.println("time: " + time);
    try {
        Thread.sleep(500); // 等待0.5s
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    //未超时，则继续测试
    String sourceXml = driver.getPageSource();

    //如果登录/注册的ank，则需要特殊处理
}
```

测试的时候，会检测那个 element 是否值得测试，如果不值得，就进行下一个按钮的测试。代码如下：

```
while(!testOrNot(e)) {
    countPassNoAct ++;
    System.out.println("pass no action!");
}
```

如果值得，就获取 element（有 by.id 和 by.xpath 共 5 种方法）的 WebElement 去进行测试（测试函数为 testEl）。代码如下：

```
//元素属性---常见的元素则会被测试
String resource_id = e.getResource_id();
String _className = e.getName();
String _class = e.get_class();
String index = e.getIndex();
String text = e.getText();
String content_desc = e.getContent_desc();
//定位内容
String path_id = resource_id;//id
String path_xpath1 = "/" + _className + "[@text='";
String path_xpath2 = "/" + _className + "[@content";
String path_xpath3 = "/" + _className + "[@class='";
String path_xpath4 = "/" + _className + "[@index='";
System.out.println("p224" + path_xpath1);
System.out.println("p256" + path_xpath2);
System.out.println("p256" + path_xpath3);
System.out.println("p256" + path_xpath4);
//三个定位方法，哪个是唯一的，就使用那种方法，不冲突

if(list1.size() == 1) {
    //测试WebElement元素
    System.out.println("test id!");
    testEl(name, app, driver, startTime, runtime, list1.get(0), true);
}
else if(list2.size() == 1) {
    //测试WebElement元素
    System.out.println("test xpath1!");
    testEl(name, app, driver, startTime, runtime, list2.get(0), true);
}
else if(list3.size() == 1) {
    //测试WebElement元素
    System.out.println("test xpath2!");
    testEl(name, app, driver, startTime, runtime, list3.get(0), true);
}
```


测试单个 element 的函数逻辑：testEl 函数；

测试整体 app 会有特殊情况的考虑，对于 jianshi 等等 apk：

```
/*
 * 哔哩哔哩：需要登录；
 * 简诗：需要注册+登录；
 */
name = name.toLowerCase();
if((name.contains("bilibili"))
    && (sourceXml.contains("请输入您的哔哩哔哩账号"))) {
    driver.findElement(By.id("com.hotbitmapgg.ohmybilibili:"))
    driver.findElement(By.id("com.hotbitmapgg.ohmybilibili:"))
    driver.findElement(By.id("com.hotbitmapgg.ohmybilibili:"))
    driver.findElement(By.id("com.hotbitmapgg.ohmybilibili:"))
    driver.findElement(By.id("com.hotbitmapgg.ohmybilibili:"))
}
else if((name.contains("jianshi"))
    && (sourceXml.contains("回归文字的本质，回归美好"))
    && (sourceXml.contains("邮箱"))) {
    //先注册简诗
    driver.findElement(By.id("com.wingjay.android.jianshi:"))
    driver.findElement(By.id("com.wingjay.android.jianshi:"))
    driver.findElement(By.id("com.wingjay.android.jianshi:"))
    driver.findElement(By.id("com.wingjay.android.jianshi:"))
    driver.findElement(By.id("com.wingjay.android.jianshi:"))
}
```

特殊考虑的情况当中，也会包含一些保护措施，例如：意外关闭就重启，无法点击就回退上一页等等：

```
else if((name.contains("gudong"))
    && (sourceXml.contains("导入导出"))) { //页面没有可以处理的元素
    driver.navigate().back(); //回退上一页
}
else if(sourceXml.contains("已停止运行")) { //已停止运行则重启
    driver.findElement(By.xpath("//android.widget.Button[@text='确定']"));
    driver.resetApp();
    try {
        Thread.sleep(6000); // 等待6s，待应用完全启动
    } catch (InterruptedException e) {}
}
```

还有一些保护措施，以避免页面出错，element 总是不检测也跳不出当前页面：

```
if(countPassNoAct >= 50) { //超过50个没有可以操作的元素，为了避免页面出错，跳不出来，就返回上一级，下次再测
    driver.navigate().back();
    break;
}
```

PS：如果有的函数需要理解大概功能，可以看 PageSourceService 类，里面会有解释：

谢谢助教！比心~