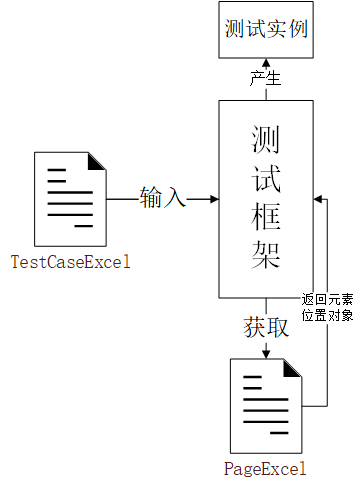
# 框架流程说明

基于关键字驱动的selenium自动化测试框架是一种功能性的自动化测试框架，也称为基于表格驱动的测试或基于动作词的测试。基于关键字驱动的自动化测试框架的目的是将测试编码从测试用例和测试逻辑分离出来，测试人员只需要将含有测试步骤，逻辑和数据的测试用例excel输入到测试框架之中，并准备好相应的元素位置对象的excel，即可实现测试脚本的自动编写，实现测试的自动化。图1展示了这个框架与excel之间的交互流程。

 图1 框架与EXCEL的交互流程

基于关键字驱动的测试框架的基本工作是首先将测试用例分解成一步步的测试步骤。然后把测试步骤划分为三个主要的部分，其一是测试对象，即元素位置对象，其二是测试动作以及最后一个是测试数据。图2是一个简单的框架工作流程图。

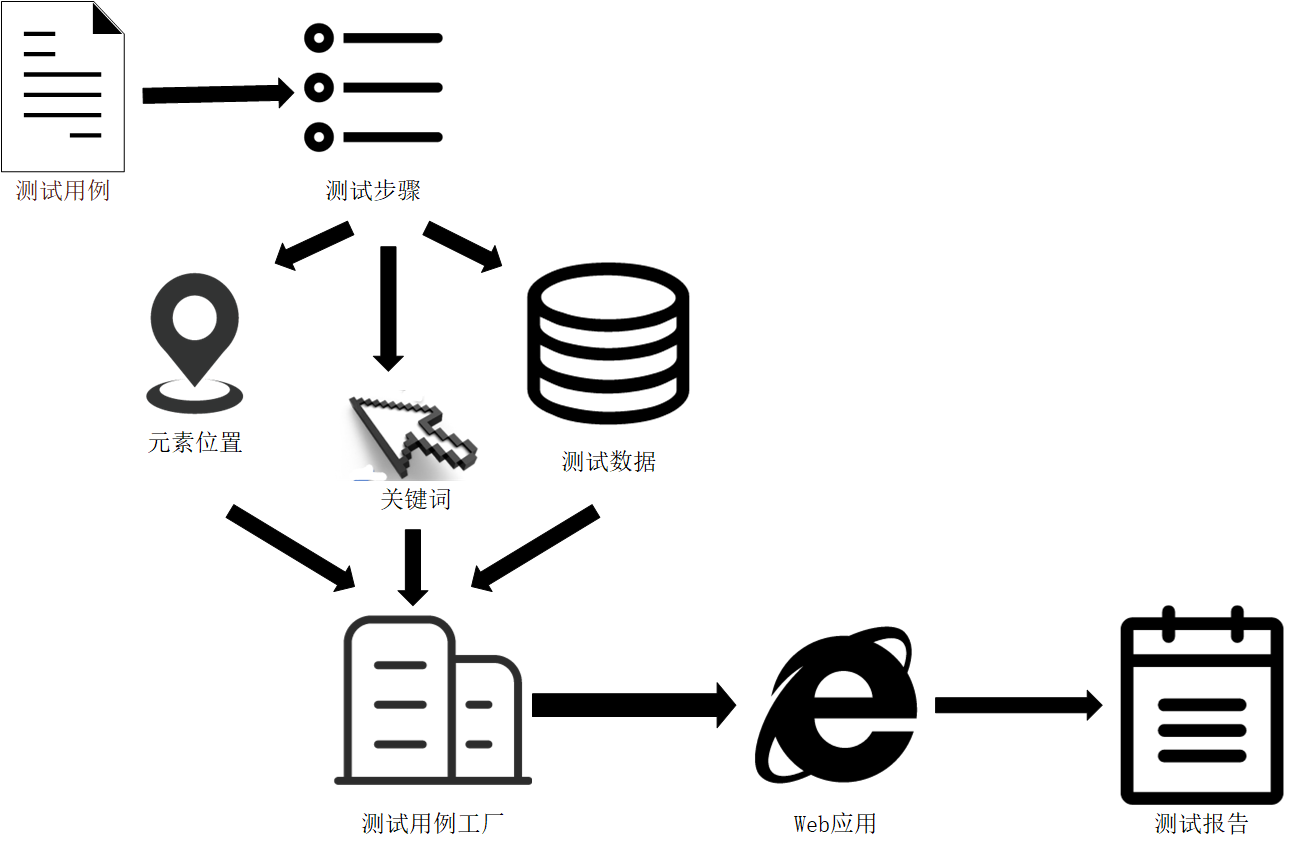


图2 简单框架工作流程图

基于关键字驱动的测试框架运行的入口类为MainBlock包下的MainRun类的main方法运行后是一个Swing界面，然后在Swing界面选择相应的测试用例的路径，并填写相应的运行次数，然后点击上面的执行测试按钮，执行测试。测试时生成一个TestCreator的一个实例对象，并调用其中的runtest方法，实现自动化测试。该框架下的主要代码类逻辑流程如图3所示。

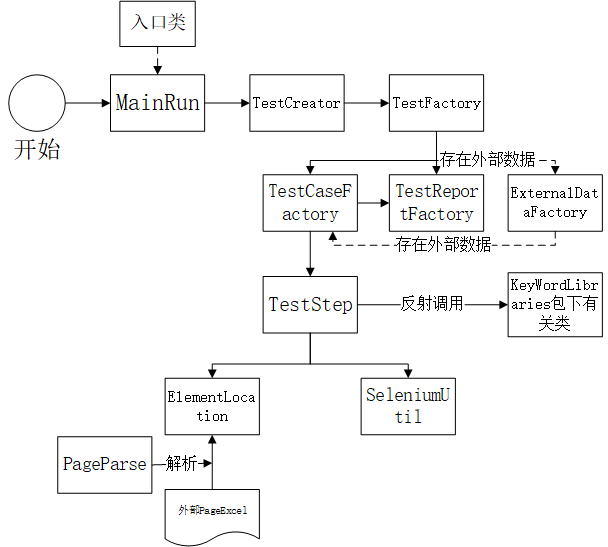


图3 框架代码主要类逻辑流程图

# 二． 框架结构说明

本章节简略叙述基于关键字驱动的selenium自动化测试框架的主要结构。包括外部结构和代码结构。

## （一） 框架外层结构

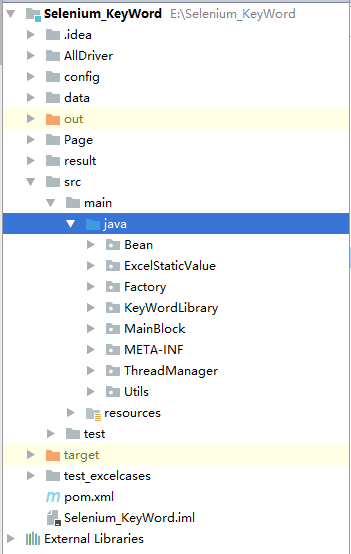
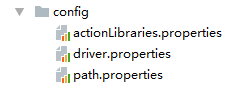
 图4 项目外层结构示例图

图 4是本项目一个外层结构示例图，其中有AllDriver, config, data, Page, result, src, test\_excelcases, 以及maven项目必有的pom.xml文件。以下简单叙述各个文件夹下的作用。

1. AllDriver: 放置所有必要浏览器启动driver所用。现在已经放置了目前常用的浏览器启动driver。
2. Config: 一个属性配置文件夹，里面放的都是一些配置文件，目前有三个properties文件。actionLibraries文件用于读取excel关键字与相应类的映射，driver文件用于映射driver路径，path里面是很多其他文件路径的配置，比如测试报告路径的配置等等。
3. data：现在已经废弃，没有什么特别作用，可忽略。
4. Page: 重要文件夹，里面放置的是有关页面元素位置的excel。
5. result: 里面有三个子文件夹，分别放置log,测试报告以及失败时的截图。
6. src: 里面放置的是框架代码。
7. test\_excelcases: 这个里面放置的是一些测试用例，但是测试用例不需要一定放在这个文件夹下，可放在任意目录下，此文件夹可有可无。

## （二） 框架代码结构

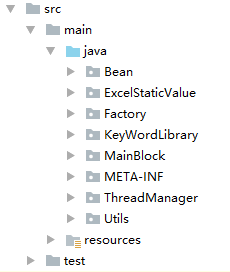
 图5 框架代码结构示例图

图5示例了框架代码的基本结构，其中有如下几个包。

1. Bean: 实体类对象包，里面含有必要的几个bean对象的实体类，其中重要的实体类TestStep类在此包下。
2. ExcelStaticValue： 存储excel静态值的包，里面类里的属性均为static final变量，excel里的列索引，sheet名等均有这些变量决定。
3. Factory： 里面有四个工厂类，分别是TestFactory, TestcaseFactory, TestReportFactory以及ExternalDataFactory四个工厂类，分别用来创建和组装测试所需的相应的功能和数据。
4. KeyWordLibrary： 里面存储一个接口和有关关键字类，所有的关键字类实现了这个名为ActionSuper的接口。
5. MainBlock: 运行的入口类所在包。
6. ThreadManager： 线程管理包，里面含有线程池等有关类，目前并未使用，可以忽略。
7. Utils： 工具包。里面含有了大量的工具类，其中有重要的ExcelCommonUtil, BrowserOperateUtil, PageParse以及已经封装好的SeleniumUtil类。 ExcelCommonUtil类主要用来读写EXCEL等。BrowserOperateUtil类主要用来启动浏览器，对浏览器进行有关操作，如最大化等。PageParse主要用来解析元素位置信息。SeleniumUtil类主要封装了大量的selenium的操作（api），必要时可直接调用里面的方法实现相关操作。其他的工具类实现相应功能，但并不是很关键，使用量有限。

# 三． 框架使用说明

本章节通过一些简单的步骤来说明其使用。

第一步，运行MainBlock包下的MainRun类的main方法，运行完后是一个swing界面，如图6所示。

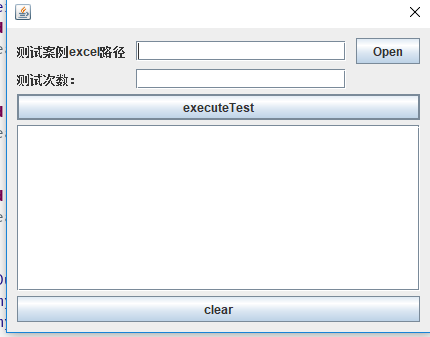


图6 swing界面

第二步，输入测试案例excel路径或者点击open按钮打开相关文件。

第三步，输入测试次数。

第四步，点击executeTest按钮执行测试。

接下来框架就会执行这个测试案例，在此之前，你必须要将所要用到的元素位置excel放入到该框架的Page文件夹下，有关测试案例excel和元素位置excel的说明和制作详见基于关键字驱动的自动化测试--Excel说明文档。框架执行完时，这个swing界面中的textarea区域将会显示有关的log信息。可以通过clear按钮清空这些显示信息。

# 四． 框架维护指南和注意事项

## （一） 项目搭建

1. 本项目框架是基于maven项目，使用的是idea。其JDK必须使用jdk 1.8+，并配置好相应的环境变量。

2. idea中可能需要设置一下项目的jdk。否则可能项目中有些代码会报类似于@since 1.6+等类型的错误。

## （二） 使用维护

1. driver和浏览器启动维护：selenium的driver是与浏览器的版本相对应，即浏览器的不同版本对应着不同版本的driver。现在本框架内含的driver是分别对应着当前最近版本的浏览器，如果你在使用该框架启动浏览器时发生了相应异常，出现了相关提示，则必须通过网上搜索下载浏览器相应版本的driver，这里提供一个谷歌浏览器driver的有关链接：<https://blog.csdn.net/huilan_same/article/details/51896672>。同时这里注意，并不是所有浏览器版本都能找到相应的driver版本，浏览器并不是版本越高越好！！！

同时，如果改变了driver的版本和放置路径，一定要修改config文件夹下的driver.pro文件，把里面的路径修改成你放置driver的路径！

有可能浏览器driver版本已经无误，但是仍然无法启动浏览器，则需要检查浏览器的安装位置，如果不是默认位置，则需要在测试案例excel—预处理sheet里启动浏览器那一行的数据（值）那一列的单元格里写上你的安装路径，写法为：浏览器英文名字+“|”+浏览器安装的绝对路径，比如firefox| D:\software\firefox51\firefox-sdk\bin\firefox.exe。或者直接重新安装浏览器至默认位置。

2. 文件夹路径维护：本框架存储元素位置的excel文件默认放置在框架的Page文件夹下，如果需要变更文件夹的位置，则需要修改框架config文件夹下的path.pro配置文件里Page文件夹的路径。同样的，测试报告，截图的路径如果要修改，也是通过修改path.pro配置文件里的相应路径。

3.代码维护：关键字维护和页面元素解析维护详见基于关键字驱动的自动化测试--Excel说明文档，其他代码维护可通过之前提到的代码逻辑流程图和代码结构说明，以及代码中有关注释进行维护。