**项目背景**

为了提高测试执行效率以及基础测试数据的录入，此框架实现了MIS系统基础数据及主流程的自动化测试执行。

**项目介绍**

项目基于开源selenium框架实现，框架原理、操作方法可参照官方文档。

1. **主要包含如下几个模块**

1.action

主要是对一个独立功能操作元素动作进行封装，如：登录操作login函数

2.configure

配置文件，如：

日志功能loger.conf配置文件（模板）

页面对象元素定位pageObjectRepository.ini配置文件

3.FilePath

封装各文件路径，主要用于定义一些全局变量，用于存储一些文件的路径，便于调用，更改一处全局生效，如：

页面对象仓库路径：pageObjectRepositoryPath

 Excel数据表路径：excelDataPath

4.Log

用于记录测试过程中系统日志，便于定位问题

5.PageObject

页面对象模式，是本框架的核心。

以一个页面为框架的操作单元，一个页面写在一个类中，获取此页面上的相关元素写成类中的方法。

6.Script

主要实现的是测试用例，需要登录系统，操作用例验证的页面、获取表格测试数据、断言、写入测试结果。

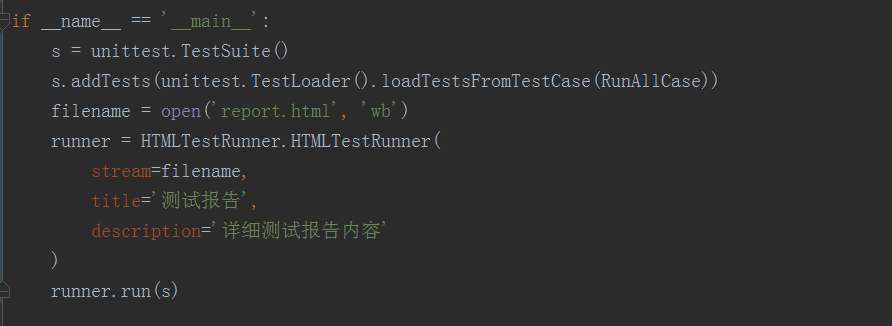
7.TestData

测试数据

包含测试期望结果文件FileLibrary、测试实际结果文件ListData、测试异常截屏文件ScreenCaptures、测试数据Excel表

8.Report

报告定制以HTMLTestRunner为模板，报告文件以HTML格式展示，如目录下的report.html，浏览器打开即可呈现。



9.Toolkit

整个测试执行过程中附加的操作方法，如对比文件、定位元素、时间格式、获取日志等



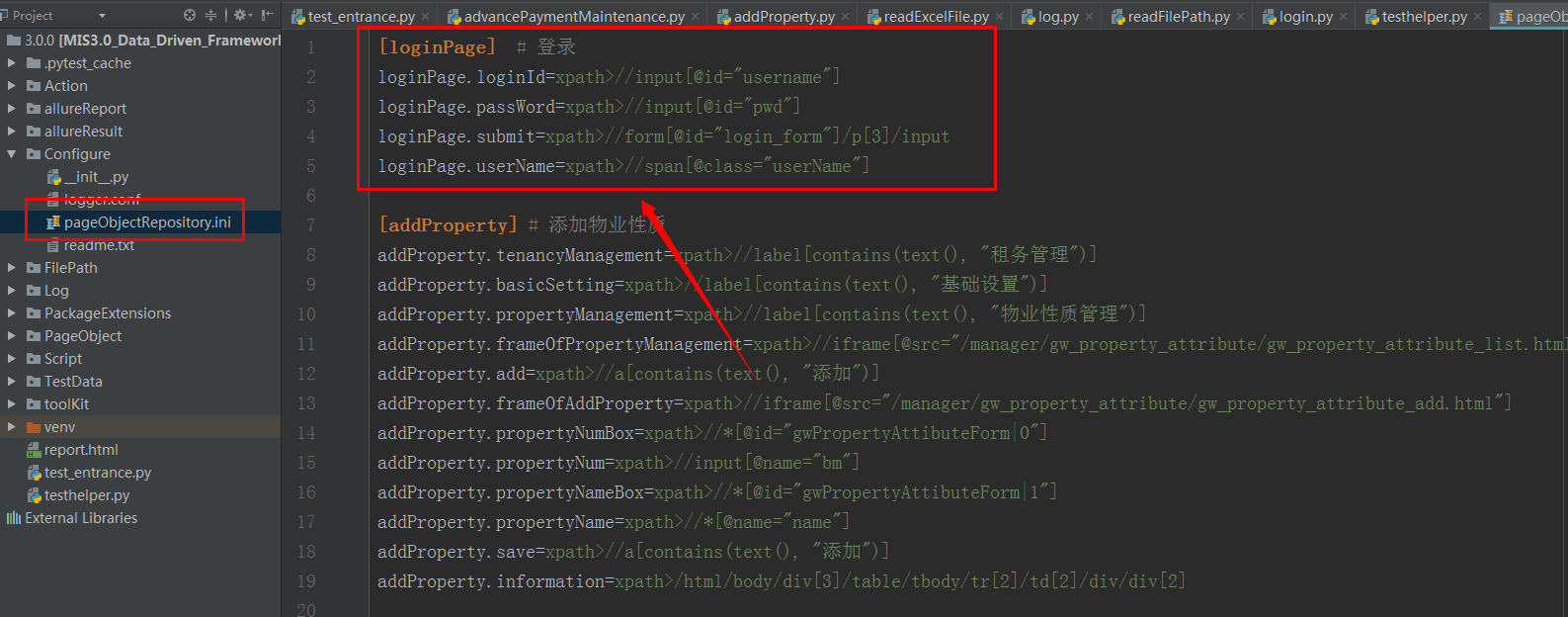
10.Test\_entrance

测试执行入口

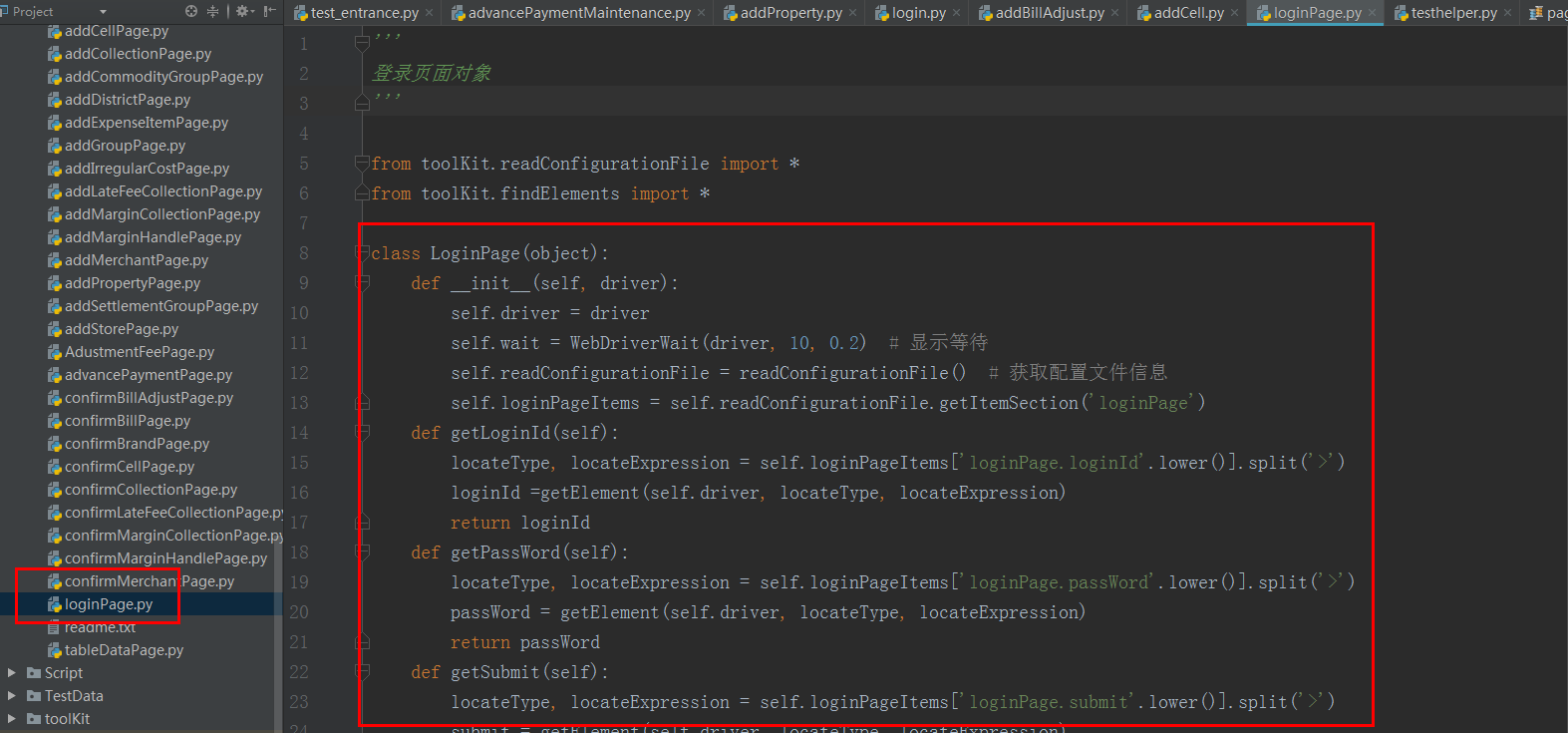
导入相关执行模块，在main（）方法中运行。

1. **编写一个测试用例实例**

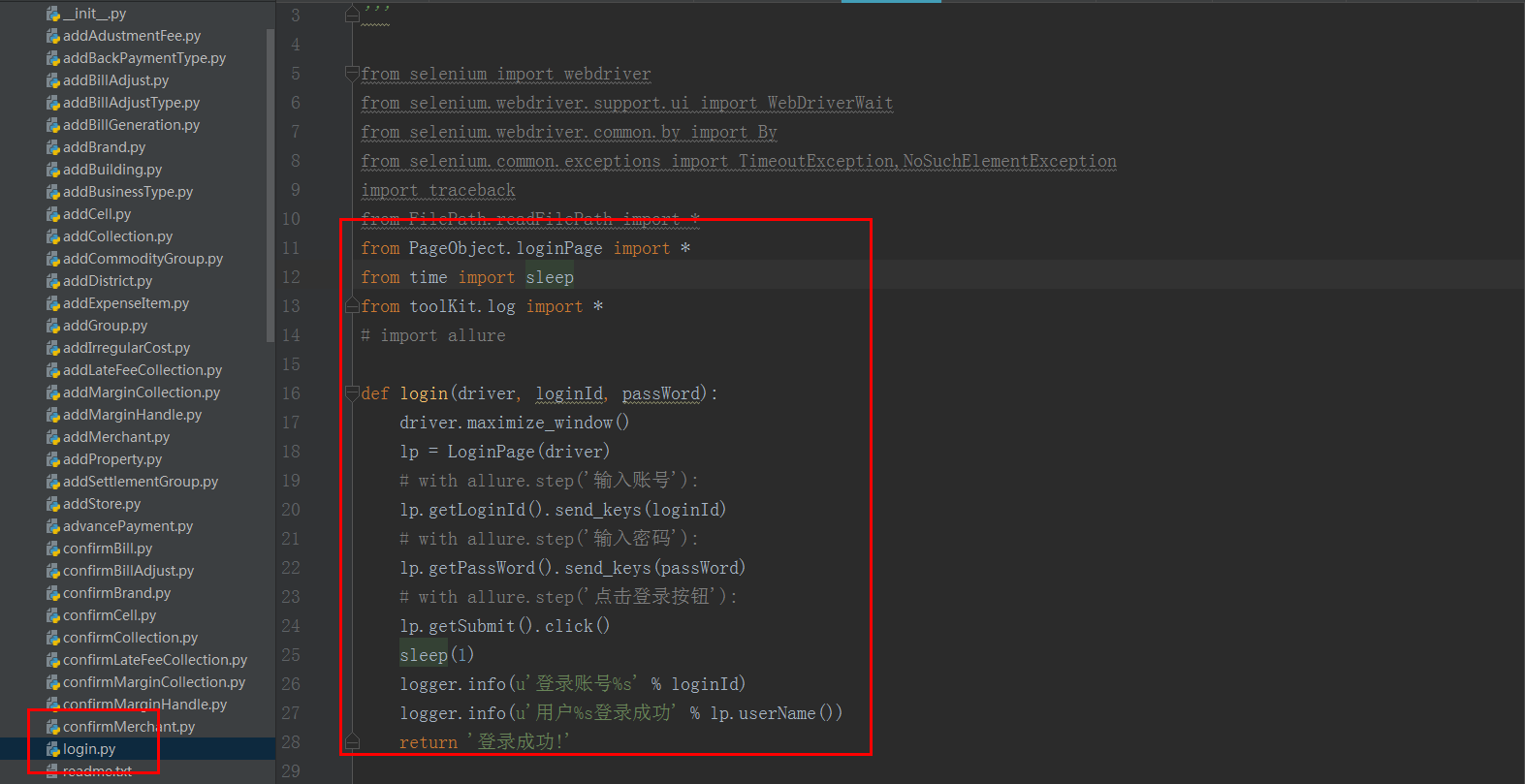
1.Configure->pageObjectRepository.ini中填写元素定位表达式



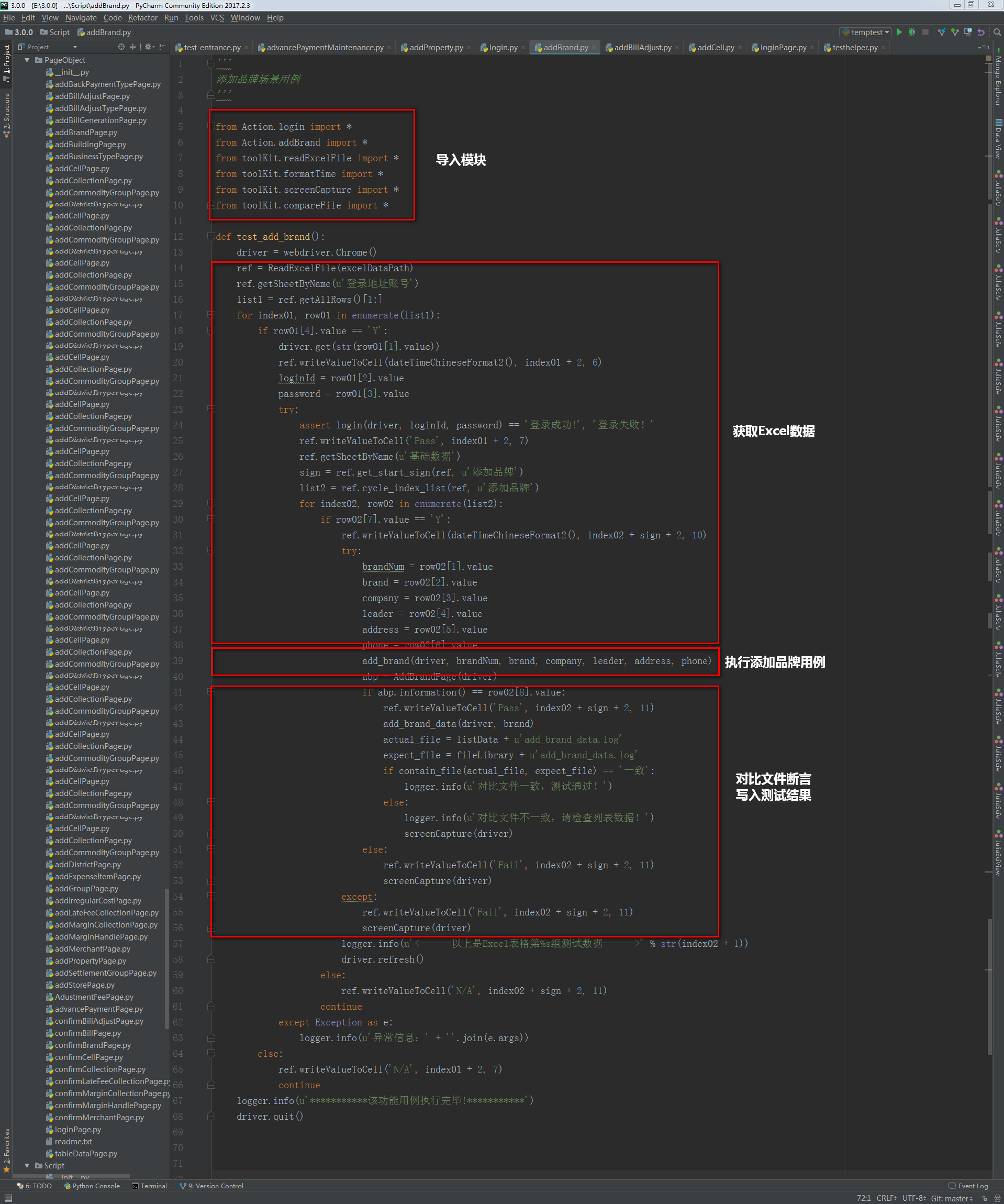
2.PageObject->编写页面对象文件，因需要读取步骤1中配置文件数据及查找元素，需要从toolkit导入相关模块，如截图。



3.Action->编写执行某一功能的函数，即步骤2页面对象中所有页面元素组成的完成一个特定功能的函数，如截图：login.py



4.Script->编写用例，以添加品牌（addBrand）为例：一个完整的用例需要首先登录（从action中导入login），准备测试数据（从Excel导入），执行测试（从action中导入addBrand），对比文件做断言，然后向Excel表写入测试结果。



1. **编写测试用例流程图**

