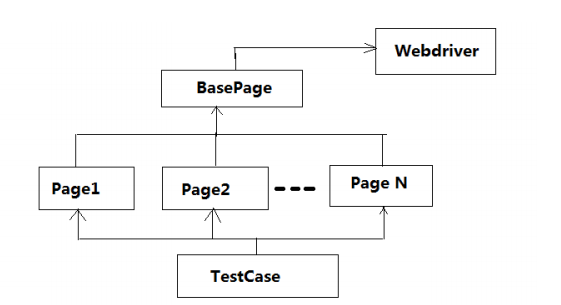
po模式是应用类技巧，因此本笔记只提供思想指导，辅助学习，具体的方法前往视频录播中和示例代码观看，之后再自行总结出自己的理解。

# PO设计原理

* 将页面封装成对象
* 页面对象内封装业务方法
* 元素定位方法可存放到其他配置文件



PO设计是一种思想，任何UI自动化测试都可以套用这种思想，不仅限于selenium的webUI模式

# PO设计实践规范

页面对象化

* 每个页面封装成对象

例如：登录页面可以设计成LoginPage类

* 封装具体业务方法

例如：登录页面的登陆方法有login(username,password)

* 不保存具体的元素定位

例如：定位用户名和密码框的表达式不写在代码里，放在外部配置文件当中

页面元素属性化

* 只涉及到要操作的元素名称
* 具体的定位方式方法不写再代码里

元素定位可配置化

* 配置以键值对形式存在
* 区分出不同的页面
* 保存元素配置的文件形式没有限制，可以是文本文件、EXCEL、数据库等

# 页面类的设计

页面类的层级抽象

* 构造页面对象

创建页面基类，封装基本操作方法，由其他业务类继承

* 封装相关业务方法

业务页面对象中封装常用的业务操作

* 抽取元素定位

抽取具体的元素定位方法存放到配置文件

* 测试案例构成

结合页面对象编写具体的业务流

# 附录

* Yaml格式配置文件特点
  + YAML 的数据组织主要依靠的是空白，缩进，分行等结构，可读性好。
  + YAML 实现简单，解析成本很低，和脚本语言的交互性好。
  + YAML 使用实现语言的数据类型。
  + YAML 表达能力强，扩展性好。
  + 安装yaml模块方法---pip install pyyaml