5**聊一聊自动化测试框架封装思路**

自动化测试框架的核心思想是将常用的脚本代码或者测试逻辑进行抽象和总结，然后将这些进行面向对象设计，将需要复用的代码封装到可公用的类方法中。通过调用公用类方法，测试类中的脚本复杂度会被大大降低，让更多脚本能力不强的测试人员来实施自动化测试。（该句摘自吴晓华老师的selenium实战宝典）

**数据驱动框架:**

笔者跟大家讲述的就是数据驱动测试框架，比如通过正确的手机号、密码登录或者错误的密码登录，来验证功能是否正确。

另外还有关键字驱动框架、数据和关键字混合框架，行为驱动框架等，笔者了解不深，如果大家感兴趣，还请自行百度。

**搭建UI自动化测试框架的步骤:**

1. 将手工操作的功能用简单的自动化代码实现

2. 将重复高的代码或者业务逻辑进行封装，在单独的类中进行封装方法的编写

3. 根据测试业务，选择数据驱动框架或者关键字框架，上述讲述的是数据驱动。

4. 将框架中公用的类进行单独封装，如测试数据处理，数据库操作，发送邮件操作，日志记录等功能进行类方法的封装实现

5. 采用Page Object基于控件的设计模式思想和unittest框架编写测试脚本，使用框架进行自动化测试

**搭建接口自动化测试框架的步骤:**

1.   将普通的接口用简单的自动化代码实现

2.   将测试数据和测试代码分离，即实体类和测试类分开

3.   将公用的方法抽象封装到基类方法中，继承基类即可

4.   将框架中公用的类进行单独封装，如读取配置文件，数据库操作，发送邮件操作，日志记录等功能进行类方法的封装实现

5. 采用unittest框架编写测试脚本，使用框架进行自动化测试

在实践操作过程中，根据出现的问题，在不断的优化框架代码或者公用类库的代码。思想是一样的，重复的代码就不断的抽象出来，通过面向对象的思想将其封装成类对象或者基类中的类方法等。

笔者重在讲的是UI和接口自动化框架的搭建及分层思路，也许有很多错误或者描述不清的地方，欢迎大家指正。