源码学习篇

规则一，你读的越多，你就越容易读懂，因为高手写程序的思维都是趋同的，正所谓万剑归宗；当然你要找到这个“同”，是需要功力的。设计模式是“同”之一，一般碰到同类型问题，大家都倾向于用同样的“模式”处理，所以你了解了这种模式，下次你看某个源程序时，其中有类似问题，你肯定就会想到作者很可能是用这种模式处理的。这样你就会更容易看清作者的思路，理清程序的脉络。

    规则二，由上之下，逐步求精。不要一开始就想把所有的细节搞清楚，否则你就会陷入“只见树木，不见森林”的困境。先要理清程序的脉络，知道那个包是干什么的，那个类是干什么的，他们之间有什么样的联系。然后在一个一个问题深究。其思想就是，大而化下，再大而化小...你要细到什么程度，取决于你的要求及期望。一般我看到包，类一层就不会看了，除非我对某个算法感兴趣，我也会仔细在研究之。其实这也是面向对象的设计思想，由上至下，而不是由下至上。无论你看到哪一层，你都可以说“我了解这个框架的实现”，只是看到的粒度不同而已。

    规则三，调试。我认为调试程序员最重要的功底，而不是最重要的之一。断点下在哪里最有可能定位问题所在，但又不浪费时间，记住断点并不是越多越好。什么时候应该用条件断点。碰到一个新的程序，你肯定要在入口Main里面下个断点，这个Main就会分几个枝出来，然后对你感兴趣的枝再设断点(Main中也许就不需要设了)，依次类推。当然，如何用更好的方法调试某个程序，是需要具体问题具体分析的。这需要经验的积累。曾经有2两个3年经验的兄弟问我同一个问题"这个IF为什么不进去？"， 我说只有一种可能就是“IF的条件不满足。”，在IF那设个断点，一个一个条件看。