**课程实践作业四**

**6.2.3当事情变得困难时**

肯尼迪总统的父亲Joseph P.Kennedy据说曾对他的孩子说：“当事情变得困难时，坚强的人才能继续走下去”。但是他从来没有调试过软件。这一部分包含了一些实际的提示关于当调试变得困难的时候该做些什么。

寻找一些通常的猜测，例如：你

以错误的顺序将参数传递给了函数；

将名字拼写错了，比如：当你应该输入一个大写字母的时候你输入了一

个小写字母。

未能初始化一个变量；

测试两个浮点数的值相等而不是近似相等（记住浮点数的运算和你在学

校里学到的运算时不一样的）；

当你意思是对象相等（比如ID（L1）==ID（L2））时，测试值是否相

等（比如比较两列，通过写出表达式L1==L2）；

忘记一些内置函数的副作用；

忘记了括号能将一个引用类型函数的对象变成一个函数的调用；

创建了一个意外的别名，或者

犯了一些其他的你经常犯的错误。

停止责问自己为什么这个程序没能执行你想要的结果，要问自己为什么程序会变成这个样子，这应该是一个更容易回答的问题，并且可能是解决怎样修复这个程序的第一步。

记住错误可能不在你所认为的地方。若果在你所认为的地方，你可能很久之前就找到了。一种典型的方法来决定去哪寻找错误就是问一下自己错误不可能发生在哪。正如夏洛克福尔摩斯所说：“排除一切的不可能，剩下的即便再不可思议，那也一定是真相。”

试着向别人解释这个问题。我们都在开发盲点。经常发生这样的情况，当你仅仅尝试向别人解释问题，将会是你看清你错过的东西。解释时一件很好的事情就是为什么错误不在某些位置上。

不要相信你读到的一切。尤其不要相信参考文件。代码不可能像注释所建议的那样运行。

停止调试并且开始编写文档。这将帮你从不同的角度看待问题。

离开然后第二天再试一次。这可能意味着错误讲比你坚持调试更迟解决，但是你可能话更少的时间用来寻找错误。也就是说，你可以延迟完成调试来换取效率。（同学们，这是一个很好的理由来早点开始处理程序问题，而不是一味地拖延。）

**6.2.4当你发现错误是**

当你认为你在代码中发现了错误的时候，开始编写和测试你的修正的诱惑几乎是不可抗拒的。然而通常情况下，更好的办法是慢一点。记住，我们最终的目的不是单单解决这一个错误，而是更加快速有效地编写没有错误的程序。

问问自己是否这个错误解释了所有观测到的症状还是仅仅只是冰山一角。如果是后者，最好要考虑到照顾这个错误何其引起的其它变化。举例来说，假如你发现这个错误是一个意外突变的而结果，你可以在本地避免这个问题（比如复制一个列表），或者你可以考虑使用一个数组来代替列表（因为数组是不可改变的），可能在代码的其他地方消除相似的错误。

在作出改变之前，试想一下你所想做的修改将会带来什么样的结果。它会引起其他的结果吗？它会引入多余的麻烦吗？它能提供整理代码其它部分的机会吗？

要确保你能回到你所处的地方。没有什么事情比意识到自己做了一大堆的调试却离自己的目标越来越远，并且没有办法回到原点这件事更让人感到沮丧了。磁盘的空间通常非常丰富，使用它存储你之前的版本。

最后，如果有很多不能解释的错误，你可能要思考一下是否一次性地寻找和修改错误是一个好办法。也许一个更好的方法是思考一下是否有更好的方法去组织程序，或者是否有更简单并且正确的实现更容易的算法。

**小结：**翻译完这两段话，让我收获了很多，因为自己以前编程时候遇到了错误就会很急躁，大半天也找不到错误。但是随着自己越来越成熟，现在遇到不会的会耐心的上网查资料，或者问同学，比如之前一直不会修改大纲级别。今天上网查了一下才知道。编程也是如此，今天在Eclipse中编写程序运行时出现了错误，然后按照网上的方法clean了一下解决了。所以在以后编程时，一定要冷静的思考。按照上面两段话所说的那样，来要求自己。