7.8.2 调试途径

无论采用什么方法，调试的目标都是寻找软件错误的原因并改正错误。通常需要把系统地分析、直觉和运气组合起来，才能实现上述目标。一般说来，有下列3种调试途径可以采用。

1. 蛮干法

蛮干法可能是寻找软件错误原因的最低效的方法。仅当所有其他方法都失败了的情况下，才应该使用这种方法。按照“让计算机自己寻找错误”的策略，这种方法印出内存的内容，激活对运行过程的跟踪，并在程序中到处都写上 WRITE(输出)语句，希望在这样生成的信息海洋的某个地方发现错误原因的线索。虽然所生成的大量信息也可能最终导致调试成功，但是，在更多情况下这样做只会浪费时间和精力。在使用任何一种调试方法之前，必须首先进行周密的思考，必须有明确的目的，应该尽量减少无关信息的数量。

2. 回溯法

回溯是一种相当常用的调试方法，当调试小程序时这种方法是有效的。具体做法是，从发现症状的地方开始，人工沿程序的控制流往回追踪分析源程序代码，直到找出错误原因为止。但是，随着程序规模的扩大，应该回溯的路径数目也变得越来越大，以至彻底回溯变成完全不可能了。

3. 原因排除法

对分查找法、归纳法和演绎法都属于原因排除法。

对分查找法的基本思路是，如果已经知道每个变量在程序内若干个关键点的正确值，则可以用赋值语句或输入语句在程序中点附近“注入”这些变量的正确值，然后运行程序并检查所得到的输出。如果输出结果是正确的，则错误原因在程序的前半部分；反之，错误原因在程序的后半部分。对错误原因所在的那部分再重复使用这个方法，直到把出错范围缩小到容易诊断的程度为止。

归纳法是从个别现象推断出一般性结论的思维方法。使用这种方法调试程序时，首先把和错误有关的数据组织起来进行分析，以便发现可能的错误原因。然后导出对错误原因的一个或多个假设，并利用已有的数据来证明或排除这些假设。当然，如果已有的数据尚不足以证明或排除这些假设，则需设计并执行一些新的测试用例，以获得更多的数据。

演绎法从一般原理或前提出发，经过排除和精化的过程推导出结论。采用这种方法调试程序时，首先设想出所有可能的出错原因，然后试图用测试来排除每一个假设的原因。如果测试表明某个假设的原因可能是真的原因，则对数据进行细化以准确定位错误。

上述3种调试途径都可以使用调试工具辅助完成，但是工具并不能代替对全部设计文档和源程序的仔细分析与评估。

如果用遍了各种调试方法和调试工具却仍然找不出错误原因，则应该向同行求助。