

全国计算机等级考试四级复习纲要五[4] PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_\\_E5\\_85\\_A8\\_E5\\_9B\\_BD\\_E8\\_AE\\_A1\\_E7\\_c98\\_138663.htm](https://www.100test.com/kao_ti2020/138/2021_2022__E5_85_A8_E5_9B_BD_E8_AE_A1_E7_c98_138663.htm)

黑盒测试是根据规格说明所规定的功能来设计测试用例，它不考虑程序中的内部结构和处理过程。常用的黑盒测试技术有等价类划分、边值分析、错误猜测等。

1.等价类划分 前面已经讲过，不能穷举所有可能的输入数据来进行测试，所以只能选取少量有代表性的输入数据，来揭露尽可能多的程序错误。这里首先要介绍一个有效的输入数据和无效的输入数据。有效的输入数据是指符合规格说明要求的合理的输入数据，它主要用来检验程序是否实现了规格说明中的功能。无效的输入数据是指不符合规格说明要求的不合理或非合法的输入数据，它主要用来检验程序是否做了规格说明以外的事。如果把所有可能的输入数据（有效的和无效的）划分成若干个等价类，那么可以合理地做出假定:如果等价类中的一个输入数据能检测出一个错误，那么等价类中的其他输入数据也能检测出同一个错误.反之，如果一个输入数据不能检测出某个错误，那么等价类中其他输入数据也不能发现这一错误（除非这个等价类的某个子集还属于另一等价类）。等价类划分方法首先把输入数据划分成若干个有效等价类和若干个无效等价类，然后设计测试用例覆盖这些等价类。来源：www.examda.com

2.边值分析 大量的实践说明，程序中在处理边界情况时出错的概率比较大，因此设计一些测试用例，使程序运行在边界情况附近，这样揭露程序中错误的可能性就更大。所谓边界条件是指相对于输入与输出等价类直接在其边界上，或稍高于其边

界，或稍低于其边界的这些状态条件。使用等价类划分方法设计测试用例时，原则上讲，等价类中的任一输入数据都可作为该等价类的代表用作测试用例。而边值分析则是专门挑选那些位于边界附近的值作为测试用例。由于边值分析方法所设计的测试用例，更有可能发现程序中的错误，因此经常把边值分析方法与其他设计测试用例方法结合起来使用。

3. 错误猜测 错误猜测是一种凭直觉和经验推测某些可能存在的错误，从而针对这些可能存在的错误设计测试用例的方法。这种方法没有机械的执行步骤，主要依靠直觉和经验。

四、软件维护 软件维护阶段覆盖了从软件交付使用到软件被淘汰为止的整个时期，它是在软件交付使用后，为了改正软件中隐藏的错误，或者为了使软件适应新的环境，或者为了扩充和完善软件的功能或性能而修改软件的过程。一个软件的开发时间可能需要一两年，但它的使用时间可能要几年或几十年，而整个使用期都可能需要进行软件维护，所以软件维护的代价是很大的，而且维护的代价还在逐年上升，据1994年Software Engineering Encyclopedia记载，在整个软件生存周期所花费的代价中，20世纪80年代末用于软件维护的代价约为75%到90年代初为90%。因此，如何提高软件维护的效率、降低维护的代价成为十分重要的问题。

（一）软件维护的分类 根据引用软件维护的原因，软件维护通常可分成改正性维护、适应性维护、完善性维护、预防性维护。

1.改正性维护 由于程序正确性证明尚未得到圆满的解决，软件测试又不可能找出程序中的所有错误，因此，在交付使用的软件中都可能隐藏着某些尚未被发现的错误，而这些错误在某种使用环境下会暴露出来。改正性维护就是在使用过程中发现了隐藏

的错误后，为了诊断和改正这些隐藏错误而修改软件的活动。

2.适应性维护 由于计算机的发展非常迅速，新的机型、新的操作系统、新的软件系统不断地涌现，为了适应计算机的飞速发展，可能要更正正在运行的软件的运行环境，如新的机型、数据库管理系统等。适应性维护就是为了适应变化了的环境而修改软件的活动。

3.完善性维护 用户在使用软件的过程中，随着业务的发展，常常希望扩充原有软件的功能，或者希望改进原有的功能或性能，以满足用户的新要求，完善性维护就是为了扩充或完善原有软件的功能或性能而修改软件的活动。

4.预防性维护 软件维护活动主要是上述三类维护，另有一类维护称为预防性维护，它是为了提高软件的可维护性和可靠性，为未来的进一步改进打下基础而修改软件的活动。据有关资料统计，在整个软件维护活动中，改正性维护约占20%，适应性维护约占25%，完善性维护约占50%以上，其他维护约占4%。

（二）与软件维护有关的问题 软件维护人员通常不是该软件的开发人员，这给软件维护带来很大的困难，特别是有些软件在开发时没有遵循软件开发的准则，没有开发方法的支持，维护这样的软件就更困难。下面列举一些与软件维护有关的问题。

（1）要维护一个软件，首先要理解它。而理解别人的程序通常是非常困难的，尤其是对软件配置（指各种文档）不齐的软件，理解起来更为困难。

（2）需要维护的软件往往缺少合格的文档，或者文档资料不齐，甚至没有文档。在软件维护中，合格的文档十分重要，它有助于理解被维护的软件。合格的文档不仅要完整正确地反映开发过程各阶段的工作结果，而且应该容易理解并应程序源代码一致。而错误的文档会把对程序的理解引入歧

途。（3）在软件维护时，不要指望得到原来开发该软件的人员的帮助。开发人员开发完一个软件后，往往去从事另一软件的开发，甚至已调离开发单位。即使原先的开发人员还在，也可能因为相隔时间太久而遗忘了实现的细节。（4）多数软件在设计时没有考虑今后的修改，给软件的修改带来困难，而且在修改软件时容易带来新的差错。对那些缺乏模块独立性和非结构化的程序来说，更是如此。（5）软件维护通常不是一件吸引人的工作。从事维护工作常使维护人员感到缺乏成就感。这也严重影响维护工作。从而导致维护质量的不高。可以看出，上述的有些问题都与被维护的质量密切相关，所以在开发软件时，要认真写好各类文档，并且应注意提高软件的可维护性，这样可在很大程序上缓解软件维护的困难。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)