

面将分别作简要的说明。

表 9.2 测试用例的设计方法

黑 盒 方 法	白 盒 方 法	
等价分类法	逻辑覆盖法	路径测试法
边界值分析法	语句覆盖	结点覆盖
错误猜测法	判定覆盖	边覆盖
因果图法*	条件覆盖	路径覆盖
	条件组合覆盖	

注： 有*者初学者可以不学

1. 黑盒测试方法

前已指出，黑盒测试以程序的功能作为测试依据。不言而喻，这里的功能将随被测程序的范围而异。所以无论选用以下的何种方法，第一步都要认真阅读被测代码（整个程序或某一模块）的功能说明。

(1) 等价分类法 (equivalence partitioning)

这种方法把被测程序的输入域划分为若干等价类，把漫无边际的随机测试变成有针对性的等价类测试。它的出发点是，每一个测试用例都代表了一类与它等价的其它例子。如果用这个例子未能发现程序的错误，则与它等价的其它例子一般也不会发现程序的错误。这样，测试人员就有可能使用少量“有代表性”的测试用例，来代替大量相类似的测试，从而大大减少总的测试次数。

设计等价类的测试用例一般分为两步进行，即①划分等价类并给出定义；②选择测试用例。选择的原则是：有效等价类的测试用例尽量公用，以期进一步减少测试次数；无效等价类必须每类一例，以防漏掉本来可能发现的错误。下面举一个简单的例子。

例 9.1 某城市的电话号码由 3 部分组成。这 3 个部分的名称与内容分别是
地区码： 空白或 3 位数字；
前 缀： 非‘0’或‘1’开头的 3 位数字；
后 缀： 4 位数字。

假定被测程序能接受一切符合上述规定的电话号码，拒绝所有不符合规定的号码，就可用等价分类法来设计它的测试用例。

第一步： 划分等价类。表 9.3 列出了划分的结果，包括 4 个有效等价类，11 个无效等价类。在每一等价类之后均加有编号，以便识别。

表 9.3 电话号码程序的等价类划分

输 入 条 件	有 效 等 价 类	无 效 等 价 类
地 区 码	空白①，3 位数字②	有非数字字符⑤，少于 3 位数字⑥，多于 3 位数字 7，
前 缀	从 200 到 999 之间的 3 位数字③，	有非数字字符⑧，起始位为‘0’⑨，起始位为‘1’⑩， 少于 3 位数字⑪，多于 3 位数字 ⑫，
后 缀	4 位数字④，	有非数字字符⑬，少于 4 位数字 ⑭，多于 4 位数字 ⑮

第二步： 确定测试用例。表中有 4 个有效等价类，可以公用以下两个测试用例：

测试数据	测试范围	期望结果
() 276-2345	等价类①、③、④	有效
(635) 805-9321	等价类②、③、④	有效

对 11 个无效等价类，应选择 11 个测试用例。例如前 3 个无效等价类可能使用下列的 3 个测试用例：

测试数据	测试范围	期望结果
(20A) 123-4567	无效等价类⑤	无效
(33) 234-5678	无效等价类⑥	无效
(7777) 345-6789	无效等价类⑦	无效

后 8 个无效等价类的测试用例留给读者做练习。这样，本例的 15 个等价类将至少需要 13 个测试用例。
〔例 9.1 完〕

练习题

1. 为例 9.1 的无效等价类⑧～⑮各设计一个测试用例。
2. 为什么无效等价类必须每类一例？

划分等价类时，必须注意给出确切的定义。例如上例中的无效等价类⑤，初看起来似应定义为“含有既非空白又非数字的字符”，才能与等价类①、②同时对应。但如果这样定义，岂不是承认由空格和数字混合组成的地区码也有效了？这就与无效等价类⑥的定义直接矛盾。又如⑨、⑩两个等价类，也不应合并为“起始位为‘0’或‘1’”，因为如果合并，测试用例就可减为 1 个（以‘0’起始或者以‘1’起始），对程序的测试也就不够完整了。

(2) 边界值分析法 (boundary value analysis)

在等价分类法中，代表一个类的测试数据可以在这个类的允许值范围内任意选择。但如果把测试值选在等价类的边界上，往往有更好的效果，这就是边界值分析法的主要思想。

举例说，税法规定公民的个人收入调节税从超过 400 元开始征收。如果用一个程序来计算税款，则“收入 ≤ 400 ”可作为判别条件，满足条件者免税，否则要对超出 400 元的部分征税。选择测试用例时，固然可以用 300、500 两个测试数据分别代表免税和征税两个等价类，但如果程序中将判别条件误写为“收入 < 400 ”，这两个数据都不能发现错误。假如将边界值 400 用作测试数据，这个错误就无法藏身了。在大多数情况下，边界值及其邻近的数据都属于敏感区，容易暴露程序的错误。

边界值分析也适用于考察程序的输出值边界。例如，如果某工资程序规定，当职工的扣款金额超过当月工资的一半时，其超出部分应留待次月扣除。如果按边界值分析法选择此时的测试用例，就应有意识地让扣款达到和超过工资额的半数，分别观察被测程序计算的当月实发工资有何变化。

(3) 错误猜测法 (error guessing)

所谓猜错，就是猜测被测程序中哪些地方容易出错，并据此设计测试用例。如果说等价法和边界值法均有线索可寻，则猜错法将更多地依赖于测试人员的直觉与经验。所以在通常情况