



4.黑盒测试

- 黑盒测试
- 等价类划分

黑盒测试

测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性

只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明

又叫做功能测试或数据驱动测试。

黑盒测试

检查范围

是否有不正确或遗漏了的功能?

在接口上，输入能否正确地接受？能否输出正确的结果？

是否有数据结构错误或外部信息访问错误？

性能上是否能够满足要求？

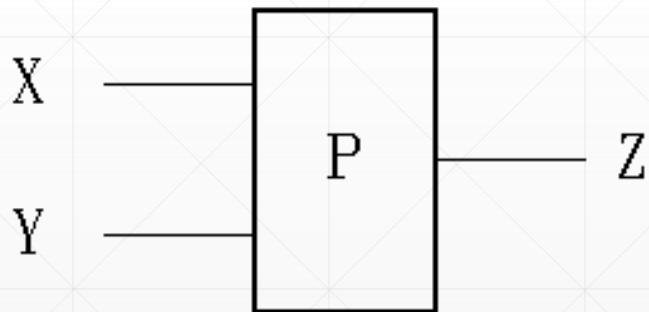
是否有初始化或终止性错误？

黑盒测试

完全测试的困难性：如果考虑所有可能的输入条件和输出条件中，黑盒测试同样可能是天文数字。

例

程序P有输入量X和Y及输出量Z。在字长为32位的计算机上运行。若X、Y取整数



可能采用的测试数据组： $2^{32} \times 2^{32} = 2^{64}$

设：

每一条路径测试需要1毫秒

一年工作 365×24 小时

需：

5亿年。

等价类划分

基本思想

把所有可能的输入数据，即程序的输入域划分成若干部分，然后从每一部分中选取少数有代表性的数据做为测试用例。



测试步骤

划分等价类
选取测试用例

等价类划分

等价类：某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是等效的。

有效等价类：对于程序的规格说明来说，是合理的，有意义的输入数据构成的集合。

无效等价类：对于程序的规格说明来说，是不合理的，无意义的输入数据构成的集合。



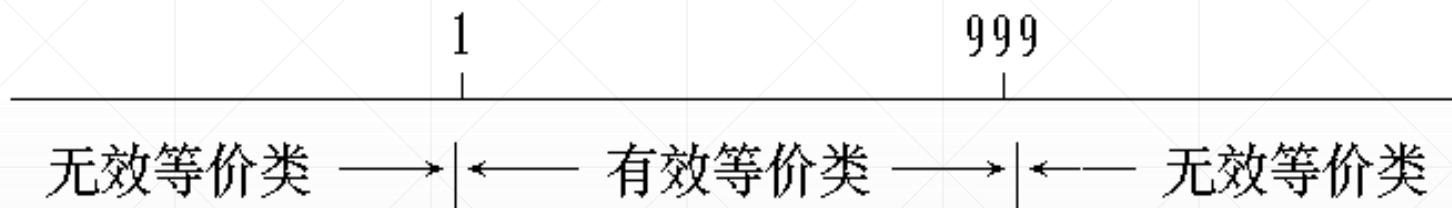
同时考虑

等价类划分

原则1：如果输入条件规定了取值范围，或值的个数，则可以确立一个有效等价类和两个无效等价类。

例

•在程序的规格说明中，对输入条件有一句话：“..... 项数可以从1到999”



输入条件	有效等价类	无效等价类
项数可以从1到999	$1 \leq \text{项数} \leq 999$ (1)	项数 < 1 (2) ; 项数 > 999 (3)

等价类划分

原则2：如果输入条件规定了输入值的集合，或者规定了“必须如何”的条件，这时可确立一个有效等价类和一个无效等价类。

例

在Pascal语言中对变量标识符规定为“以字母打头的.....串”。

输入条件	有效等价类	无效等价类
以字母打头的.....串	所有以字母打头的.....串(1)	所有不是以字母打头的.....串(2)

等价类划分

原则3：如果输入条件是一个布尔量，则可以确定一个有效等价类和一个无效等价类。

例

在教师管理系统中，教师的在岗状态为布尔量，是或否

输入条件	有效等价类	无效等价类
教师的在岗状态	是(1)	否(2)

等价类划分

原则4：如果规定了输入数据的一组值，而且要对每个输入值分别进行处理。可为每一个输入值确立一个有效等价类，所有不允许的输入值集合为一个无效类。

例

在教师上岗方案中规定对教授、副教授、讲师和助教分别计算分数，做相应的处理。

输入条件	有效等价类	无效等价类
不同职称的分别计数	教授 (1)；副教授 (2)；讲师 (3)；助教 (4)	不在{教授、副教授、讲师、助教}中的其他值集合(5)

等价类划分

原则5：如果规定了输入数据必须遵守的规则，则可以确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。

例

Pascal语言规定 “一个语句必须以分号 ‘;’ 结束”。

输入条件	有效等价类	无效等价类
一个语句必须以分号 ‘;’ 结束	以分号 ‘;’ 结束的语句 (1)	以 ‘:’ 结束 (2) ; 以 ‘,’ 结束 (3); 以 ‘ ’ 结束 (4) : 以LF结束 (5)等

等价类划分

等价类划分步骤

(一)

- 确定等价类

(二)

- 建立等价类表，列出所有划分出的等价类

(三)

- 为每一个等价类规定一个唯一编号；

(四)

- 设计一个新的测试用例，尽可能多地覆盖尚未被覆盖的有效等价类，重复这一步，直到所有的有效等价类都被覆盖为止；

(五)

- 设计一个新的测试用例，仅覆盖一个尚未被覆盖的无效等价类，重复这一步，直到所有的无效等价类都被覆盖为止

等价类划分

例

在某一PASCAL语言版本中规定：“标识符是由字母开头，后跟字母或数字的任意组合构成。有效字符数为不超过8个。”

并且规定：“标识符必须先说明，再使用。” “在同一说明语句中，标识符至少必须有一个。”

等价类划分

例

- 1、确定等价类

2、建立等价类表

3、为等价类唯一编号
-

输入条件	有效等价类	无效等价类
标识符个数	≥1个 (1)	0个 (2)
标识符字符数	8个 (3)	0个 (4) ; >8个 (5)
标识符组成	字母、数字的任意组合 (6) ;	非字母数字字符 (7) ; 保留字 (8)
第一个字符	字母 (9)	非字母 (10)
标识符使用	先说明后使用 (11)	未说明已使用 (12)

唯一编号

等价类划分

例

4、设计用例，覆盖尽量多的有效等价类
重复，直至覆盖所有有效等价类



```
① VAR x , T1234567 : REAL ;  
    BEGIN x := 3.414 ;  
          T1234567 := 2.732 ;  
          .....  
          (1), (3), (6), (9), (11)
```


等价类划分

例

5、设计用例，仅覆盖一个无效等价类

重复，直至覆盖所有无效等价类



```
② VAR : REAL ; (2)
③ VAR x , : REAL; (4)
④ VAR T12345678 : REAL ; (5)
⑤ VAR T$ : CHAR ; (7)
⑥VAR GOTO : INTEGER ; (8)
⑦ VAR 2T : REAL ; (10)
⑧ VAR PAR : REAL ; (12)
  BEGIN .....
    PAP := SIN (3.14 * 0.8) / 6 ;
```

