

# 测试用例设计 - 黑盒测试

---

# 测试用例设计 - 黑盒测试

---

- 测试用例设计概述
- 黑盒测试的概念
- 黑盒测试的数学知识
- 黑盒测试用例设计技术

# 测试用例设计概述

---

- 测试用例的定义
- 测试用例的重要性
- 好测试用例的特点
- 设计测试用例的基本准则
- 设计测试用例的着眼点
- 测试用例的编写标准
- 测试用例设计的误区

# 测试用例设计概述

---

## □ 什么是软件测试用例

- (1) 测试用例是为特定的目的而设计的一组测试输入、 执行条件和预期的结果。
- (2) 测试用例是执行的最小实体。

# 测试用例设计概述

---

## □ 好测试用例的特点

### 1. 完整的

完整性是对测试用例最基本的要求，尤其是一些基本功能项上，如有遗漏，那是不可原谅的。

完整性还体现在临界测试、压力测试、性能测试等方面，这方面测试用例也要能够涉及到。

### 2. 准确

按测试用例的输入一步步测完后，要能够根据测试用例描述的输出得出正确的结论，不能出现模糊不清的语言。

### 3. 清晰、简洁

好的测试用例描述清晰，每一步都应有响应的作用，有很强的针对性，不应出现一些冗繁无用的操作步骤。测试用例不应太简单，也不能太过复杂，最大操作步骤最好控制在15步之内。

# 测试用例设计概述

---

## 4. 可维护性

由于软件开发过程中需求变更等原因的影响，常常需对测试用例进行修改、增加、删除等，以便测试用例符合相应测试要求。测试用例应具备这方面的功能。

## 5. 适当性

测试例应该适合特定的测试环境以及符合整个团队的测试水平，如纯英语环境下的测试用例最好使用英文编写。

## 6. 可复用性

要求不同测试者在同样测试环境下使用同样测试用例都能得出相同结论。

# 测试用例设计概述

---

## 7. 其他

如可追溯性、可移植性也是对编写测试用例的一个要求。另外，好的测试用例也是最有可能抓住错误的；不重复、多余的；是一组相似测试用例中最有效的；

# 测试用例设计概述

---

- 为使测试有效，必须利用策略，发现尽可能多的缺陷
- 为使测试有效率，必须用最少的测试去发现最多的缺陷
- 测试就像侦探..

为了更好地发现缺陷，测试人员必须理解设计人员和程序员的思路。

测试人员不能遗忘任何不保险的东西，应怀疑一切。

测试人员不能花费太多时间，必须有效率。



# 测试用例设计概述

---

## □ 测试用例一质量

- 具有合理的捕获缺陷的概率
- 执行了重要的区域
- 做了应引起注意的事情
- 不做多余的事情
- 既不太简单也不太复杂
- 不与其它测试用例冗余
- 使得缺陷显而易见
- 考虑缺陷的隔离和识别

# 测试用例设计概述

---

## □ 测试用例的编写标准

在**ANSI/IEEE829-1983**标准中列出了和测试设计相关的测试用例编写规范和模板。标准模板中主要元素如下：

- 标识符——惟一标识每一个测试用例
- 测试项——准确的描述所需要测试的项及其特征
- 测试环境要求——表征执行该测试用例需要的测试环境
- 输入标准——执行测试用例的输入需求（这些输入可能包括数据、文件或者操作）
- 输出标准——按照指定的环境和输入标准得到的期望输出结果
- 测试用例之间的关联——标识该测试用例与其它的测试（或其它测试用例）之间的依赖关系

# 测试用例设计概述

---

- 测试用例示例：对Windows记事本程序进行测试，选取其中的一个测试项——文件菜单栏的测试：测试对象——记事本程序文件菜单栏（测试用例标识10000，下同）

所包含的子测试用例描述如下：

- |-----文件/新建（1001）
- |-----文件/打开（1002）
- |-----文件/保存（1003）
- |-----文件/另存为（1004）
- |-----文件/页面设置（1005）
- |-----文件/打印（1006）
- |-----文件/退出（1007）

选取其中的一个子测试用例——文件/退出（1007）作为例子，测试用例如下表所示。

字段名称	描 述	
标识符	1007	
测试项	记事本程序，“文件”菜单栏中的“文件” “退出”命令的功能测试	
测试环境要求	Windows 2000 Professional 中文版	
输入标准	(1) 打开记事本程序，不输入任何字符，单击“文件” “退出”命令 (2) 打开记事本程序，输入一些字符，不保存文件，单击“文件” “退出”命令 (3) 打开记事本程序，输入一些字符，保存文件，单击“文件” “退出”命令 (4) 打开一个记事本文件 (*.txt)，不做任何修改，单击“文件” “退出”命令 (5) 打开一个记事本文件，修改后不保存，单击“文件” “退出”命令	
输出标准	(1) 记事本未做修改，单击“文件” “退出”命令，能正确地退出应用程序，无提示信息 (2) 记事本做修改未保存或者另存，单击“文件” “退出”命令，会提示“未定标题文件的文字已经改变，想保存文件吗？”单击“是”，Windows将打开“保存”/“另存为”对话框，单击“否”，文件将不被保存并退出记事本程序，单击“取消”将返回记事本窗口	
测试用例间的关联	14	

# 测试用例设计概述

---

## □ 下表是另一个测试用例示例(登录功能测试)

该测试用例并未涵盖所有的非法输入，如非法输入中可能会有“**user=\*,pw=\***”的组合，对回车的默认操作，空格输入，对输入上溢的处理的处理（可能会跳过身份验证）等等。

如果你有兴趣，至少可以再补充**5-10**条左右的输入组合（当然，如果步骤超过**15**步，用例的易操作性就降低，可以再创建一个测试用例）。

项目/软件	技术出口合同网络申领系统（企业端）	程序版本	1.0.25			
功能模块名	Login	编制人	xxx			
用例编号-	TC-TEP_Login_1	编制时间	2002.10.12			
相关的用例	无					
功能特性	用户身份验证					
测试目的	验证是否输入合法的信息，允许合法登陆，阻止非法登陆					
预置条件	无	特殊规程说明	如数据库访问权限			
参考信息	需求说明中关于“登陆”的说明					
测试数据	用户名=yiyh 密码=1					16

操作步骤	操作描述	数据	期望结果	实际结果	测试状态
1	输入用户名称，按“登陆”按钮。	用户名=yiyh，密码为空	显示警告信息“请输入用户名和密码！”		
2	输入密码，按“登陆”按钮。	用户名为空，密码=1	显示警告信息“请输入用户名和密码！”		
3	输入用户名和密码，按“登陆”按钮。	用户名=yiyh，密码=2	显示警告信息“请输入用户名和密码！”		
4	输入用户名和密码，按“登陆”按钮。	用户名=xxx，密码=1	显示警告信息“请输入用户名和密码！”		
5	输入用户名和密码，按“登陆”按钮。	用户名=xxx，密码=2	显示警告信息“请输入用户名和密码！”		
6	输入用户名和密码，按“登陆”按钮。	用户名=空，密码=空	显示警告信息“请输入用户名和密码！”		

7	输入用户名和密码，按“登陆”按钮。	用户名=yiyh，密码=1	进入系统页面。		
8	输入用户名和密码，按“登陆”按钮。	用户名=Admin，密码=admin	进入系统维护页面。		
9	输入用户名和密码，按“登陆”按钮。	用户名=yiyh',密码=1	显示警告信息“请输入用户名和密码！”		
10	输入用户名和密码，按“登陆”按钮。	用户名=yiyh，密码=1'	显示警告信息“请输入用户名和密码！”		
11	输入用户名和密码，按“重置”按钮。	用户名=yiyh，密码=1	清空输入信息		
测试人员		开发人员			



# 测试用例设计概述

---

## □ 测试用例—维护

### ■ 增加

针对新被测特征的测试用例

### ■ 删除

零碎的测试用例

过时的测试用例

不受控制的测试用例

冗余的测试用例

### ■ 修改

为适应被测对象的变更进行修改

# 测试用例设计概述

---

## □ 测试用例设计的误区

- 能发现到目前为止没有发现的缺陷的用例是好的用例  
作为测试实施依据的测试用例，应当作为一个集合来认识，必须要能完整覆盖测试需求，而不应该针对单个的测试用例去评判好坏。
- 测试用例应该详细记录所有的操作信息，使一个没有接触过系统的人员也能进行测试  
测试用例维护费用太高，测试资源难保证
- 测试用例设计是一劳永逸的事情  
测试用例是动态的，一旦测试环境、需求、设计、实现发生了变化，测试用例都需要相应发生变化

# 测试用例设计概述

---

## ■ 测试用例不应该包含实际的数据

测试用例是“一组输入、执行条件、预期结果”、毫无疑问地应该包括清晰的输入数据和预期输出，没有测试数据的用例最多只具有指导性的意义，不具有可执行性。当然，测试用例中包含输入数据会带来维护、与测试环境同步之类的问题。

## ■ 测试用例中不需要明显的验证手段

“预期结果”的含义应不只是程序的可见行为。例如，对于一个订货系统，输入订货数据，点击“确定”按钮后，系统提示“订货成功”，这样是不是一个完整的用例呢？是不是系统输出的“订货成功”就应该作为我们唯一的验证手段呢？显然不是。订货是否成功还需要查看相应的数据记录是否更新，因此，在这样的一个用例中，还应该包含对测试结果的显式的验证手段：在数据库中执行查询语句进行查询，看查询结果是否与预期的一致。

# 黑盒测试

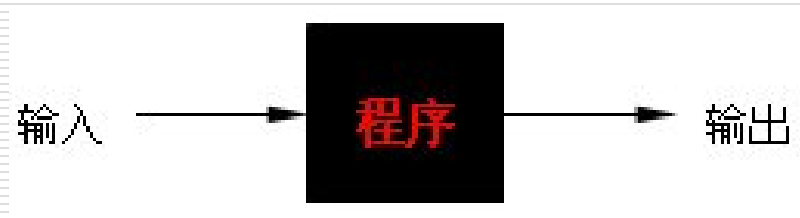
---

## □ 黑盒测试的概念

### ■ 什么是黑盒测试？

黑盒测试又称功能测试、数据驱动测试或基于规格说明书的测试，是一种从用户观点出发的测试。

黑盒测试示意图



测试人员把被测程序当作一个黑盒子。

# 黑盒测试

---

## □ 黑盒测试特点

- 测试人员不需了解程序内部的代码和实现
- 测试用例设计不依赖于系统内部的设计与实现
- 测试用例设计基于功能的定义和需求说明书
- 从用户角度出发进行测试，能较容易知道用户的需求和用户会遇到的问题
- 关注于测试数据的选择和测试结果的分析
- 对测试人员的编程技术要求不高
- 在自动化测试时较为方便
- 不易发现代码部分的缺陷

# 黑盒测试

---

## □ 采用黑盒测试的目的

主要是在已知软件产品所应具有的功能的基础上，进行：

- 1) 检查程序功能能否按需求规格说明书的规定正常使用，测试各个功能是否有遗漏，检测性能等特性要求是否满足。
- 2) 检测人机交互是否错误，检测数据结构或外部数据库访问是否错误，程序是否能适当地接收输入数据而产生正确的输出结果，并保持外部信息（如数据库或文件）的完整性。
- 3) 检测程序初始化和终止方面的错误。

# 黑盒测试

---

□ 黑盒测试主要测试的错误类型有：

- ①不正确或遗漏的功能；
- ②接口、界面错误；
- ③性能错误；
- ④数据结构或外部数据访问错误；
- ⑤初始化或终止条件错误等等。



# 黑盒测试

---

□ 黑盒测试要考虑以下问题：

- 1) 如何测试功能的有效性？
- 2) 何种类型的输入会产生好的测试用例？
- 3) 系统是否对特定的输入值尤其敏感？
- 4) 如何分隔数据类的边界？
- 5) 系统能够承受何种数据率和数据量？
- 6) 特定类型的数据组合会对系统产生何种影响？



# 黑盒测试

---

## □ 黑盒测试的应用

- 运用黑盒测试方法，可以导出满足以下标准的测试用例集：
  - 1) 所设计的测试用例能够减少达到合理测试所需的附加测试用例数；
  - 2) 所设计的测试用例能够告知某些类型错误的存在或不存在，而不是仅仅与特定测试相关的错误。
- 黑盒测试又可分为功能测试和非功能测试
  - 功能测试主要有：等价类划分、边界值分析、因果图法、判定表法、场景法、正交实验法，随机测试法、错误推测法等。
  - 非功能测试主要有：配置/安装测试、兼容性测试、互操作性测试、文档和帮助测试、性能测试、可靠性测试、易用性测试和界面测试等。

# 黑盒测试

---

## □ 穷举测试的不可能性

理论上，用黑盒测试发现程序中的错误，必须把所有可能的输入和输出作为测试数据考虑，来检查程序是否都能产生正确结果。

但这是不可能的，因为穷举测试数量太大，无法完成。如

# 黑盒测试

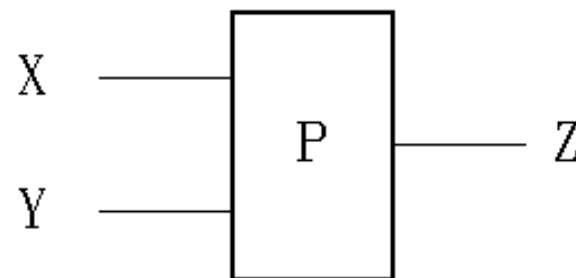
假设一个程序P有输入量X和Y及输出量Z。在字长为32位的计算机上运行。若X、Y取整数，按黑盒方法进行穷举测试：

可能采用的测试数据组：

$$2^{32} \times 2^{32} = 2^{64}$$

如果测试一组数据需要1毫秒，一年工作  
 $365 \times 24$ 小时，完成所有测试需5亿年。

因此，我们只能在大量可能的数据中，选取其中一部分作为测试用例。



# 黑盒测试的数学知识

---

## □ 集合论

### 1 集合与成员

某些指定的对象集在一起就成为一个集合（集）。其中每一个对象叫元素。

$$M_{30} = \{ 4月, 6月, 9月, 11月 \}$$

### 2 集合的表示

列举法：列出集合元素 如  $M_{30}$

描述法：给出集合元素判定规则

如  $Y = \{ 年 | 1900 < 年 < 2199 \}$

图文法

# 数学知识 - 集合论

---

## 3 空集的表示

$\Phi$

$\Phi = \{\text{年}: 2012 \leq \text{年} \leq 1812\}$

注:  $\Phi$ ,  $\{\Phi\}$  是不同的。

## 4 维恩图(Venn Diagram)

The Venn Diagram is made up of two or more overlapping circles. It is often used in mathematics to show relationships between sets

## 5 集合的关系

A 是 B 的子集 :  $A \subseteq B$

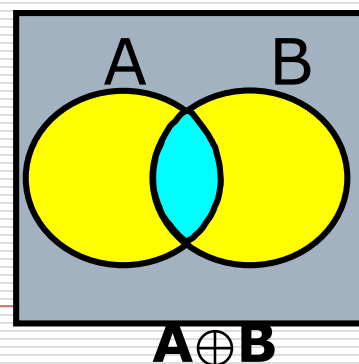
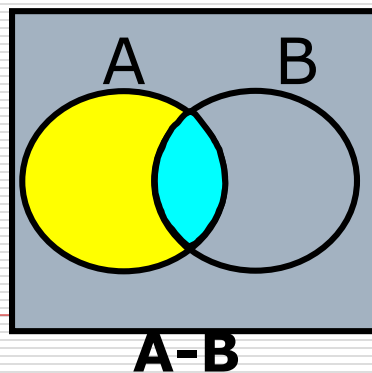
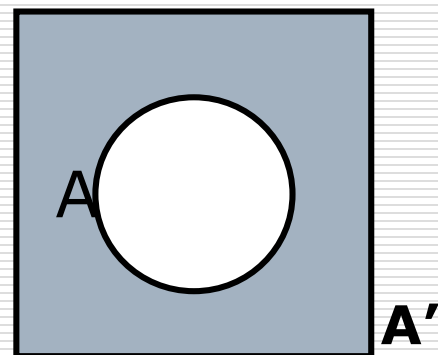
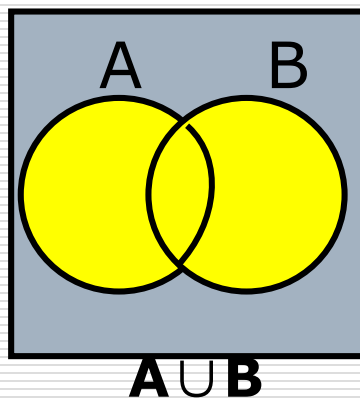
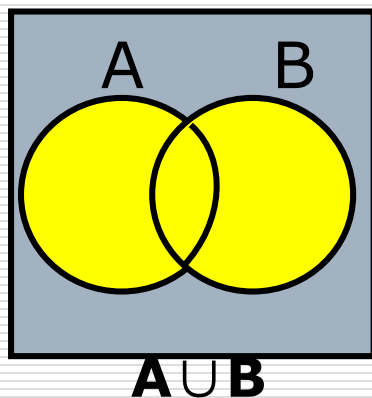
A 是 B 的真子集 :  $A \subset B$

A 和 B 是相等集合:  $A = B$

# 数学知识 - 集合论

## 6 集合操作

集合基本操作：并、交和补





# 数学知识 - 集合论

---

## 7 两个集合的笛卡儿集

两个集合A和B的笛卡儿积（又叫叉积），是集合 $A \times B$

$$A \times B = \{ \langle x, y \rangle : x \in A \wedge y \in B \}$$

例

$$A = \{1, 2\}, B = \{0, 1\}$$

$$A \times B = \{(1, 0), (1, 1), (2, 0), (2, 1)\}$$

维恩图不能显示笛卡儿积

# 数学知识 - 集合论

---

## 8 集合的划分

若 $A_1, A_2, \dots, A_n$ 是集合 $A$ 的子集，则

$A_1, A_2, \dots, A_n$ 是集合 $A$ 的一个划分，当且仅当

$A_1 \cup A_2 \cup \dots \cup A_n = A$  且

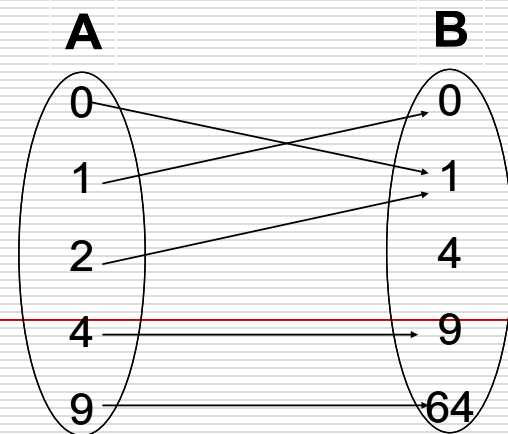
$A_i \cap A_j = \emptyset \ (i \neq j)$

测试（1）完备性

（2）无冗余性



# 数学知识 - 函数



## □ 函数

- 任何程序都可以看成将其输出与输入关联起来的函数，因此函数是开发测试的核心概念。
- 函数：一般地，设A、B是两个非空的数集，如果按某种对应法则f，对于集合A中的每一个（任意性）元素x，在集合B中都有（存在性）唯一（唯一性）的元素y和它对应，这样的对应叫做集合A到集合B的一个函数（三性缺一不可）

函数的本质：建立在两个非空数集上的特殊对应。这种“特殊对应”：  
1). 可以是“一对一” 2). 可以是“多对一” 3). 不能“一对多”  
4). A中不能有剩余元素 5). B中可以有剩余元素

程序实现的功能大多数是多对一的函数。这对测试很重要（多对一测试可选代表等价类1对1，功能相似也可分等价类）

# 数学知识 - 命题逻辑

□ 命题：有真假值，且无歧义

$p$	$q$	$p \wedge q$	$p \vee q$	$\sim p$	$p \oplus q$
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	T
F	F	F	F	T	F

# 黑盒测试

---

## □ 测试用例设计技术

- 等价类划分方法
- 边界值分析方法
- 错误推测方法
- 因果图方法
- 判定表驱动分析方法
- 场景设计方法
- 其他几种方法
- 功能性测试小结

# 等价类划分

---

- 概述
- 等价类的划分原则
- 等价类划分法的测试用例设计
- 常见等价类划分测试形式
- 使用等价类划分法测试的实例



# 等价类划分（概述）

---

□ 等价类划分法是一种重要的、常用的黑盒测试方法，它将不能穷举的测试过程进行合理分类，从而保证设计出来的测试用例具有完整性和代表性。

□ 等价类划分法

是把所有可能的输入数据，即程序的输入域划分为若干个等价类（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例。

□ 等价类

所谓等价类是指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，它们具有等价特性。因此，可以合理的假定：测试某等价类的代表值就是等效于对于这一类其它值的测试。

## 等价类划分法（续）

---

例：设计这样的测试用例，来实现一个对所有实数进行开平方运算（ $y = \text{sqrt}(x)$ ）的程序的测试。

➤思考方向：

由于开平方运算只对非负实数有效，这时需要将所有的实数（输入域 $x$ ）进行划分，可以分成：正实数、0 和 负实数。假设我们选定+1.4444代表正实数，-2.345代表负实数，则为该程序设计的测试用例的输入为+1.4444、0 和 -2.345。

## 3.4.1 等价类的划分原则

---

### □ 等价类特性

等价类是输入域的某个子集合，而所有等价类的并集就是整个输入域。因此，等价类对于测试有两个重要的意义：

- **完备性**——整个输入域提供一种形式的完备性
- **无冗余性**——若互不相交则可保证一种形式的无冗余性

### □ 如何划分？

划分等价类重要的是：集合的划分，划分为互不相交的一组子集，而子集的并是整个集合。

先从程序的规格说明书中找出各个输入条件，再为每个输入条件划分多个等价类，形成若干互不相交的子集。再在同一类中标识（选择）一个测试用例。（同一等价类中，往往处理相同，相同处理映射到“相同的执行路径”）

# 等价类划分

---

□ 采用等价类划分法设计测试用例通常分两步进行：

- 1) 确定等价类，列出等价类表。
- 2) 确定测试用例。

□ 划分等价类可分为两种情况：

(1) 有效等价类

是指对软件规格说明而言，是有意义的、合理的输入数据所组成的集合。

利用有效等价类，能够检验程序是否实现了规格说明中预先规定的功能和性能。



# 等价类的划分原则（续）

---

## （2）无效等价类

是指对软件规格说明而言，是无意义的、不合理的输入数据所构成的集合。

利用无效等价类，可以鉴别程序异常处理的情况，检查被测对象的功能和性能的实现是否有不符合规格说明要求的地方。

设计测试用例时,要同时考虑这两种等价类.因为,软件不仅要能接收合理的数据,也要能经受意外的考验.这样的测试才能确保软件具有更高的可靠性.

# 等价类的划分原则（续）

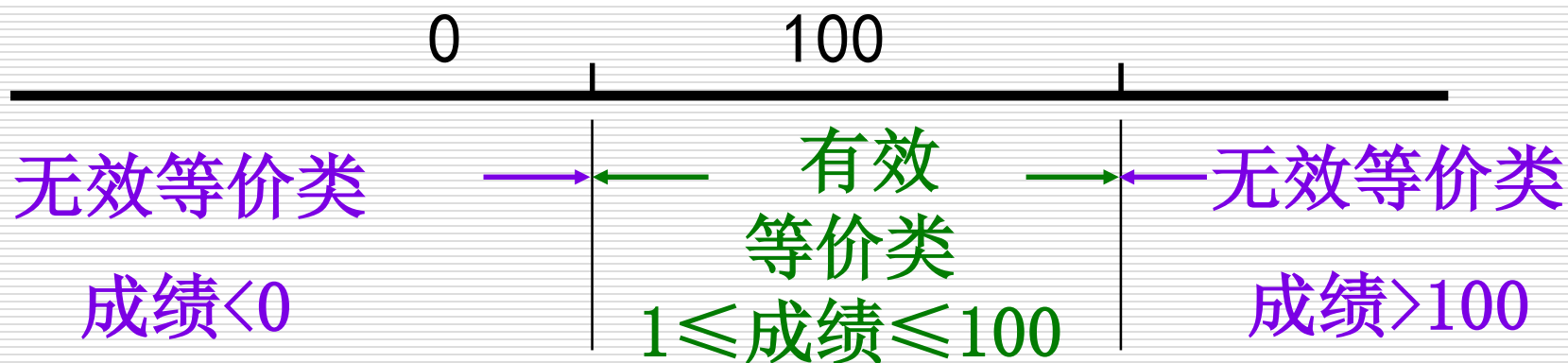
## □ 划分等价类的方法

下面给出六条确定等价类的原则

### ①按照区间划分

在输入条件规定了取值范围或值的个数的情况下,则可以确立一个有效等价类和两个无效等价类。

如：输入值是学生成绩，范围是0~100



# 等价类的划分原则（续）

---

## ②按照输入限制划分

在输入条件规定了输入值的集合或者规定了“必须如何”的条件下,可确立一个有效等价类和一个无效等价类。

如规定输入必须为数字,可确定一个有效等价类：数字和一个无效等价类：字母。

## ③按照输入布尔量划分

在输入条件是一个布尔量的情况下,可确定一个有效等价类和一个无效等价类。

# 等价类的划分原则（续）

---

## ④按照数值划分

在规定了输入数据的一组值（假定 $n$ 个），并且程序要对每一个输入值分别处理的情况下，可确立 $n$ 个有效等价类（每个值确定一个有效等价类）和一个无效等价类（所有不允许的输入值的集合）。

例：输入条件说明学历可为：专科、本科、硕士、博士四种之一，则分别取这四种这四个值作为四个有效等价类，另外把四种学历之外的任何学历作为无效等价类。

# 等价类的划分原则（续）

---

## ⑤按照限制条件或规则划分

在规定了输入数据必须遵守的规则的情况下,可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）。

如：程序中，成绩输入为0~100的整数，可确立一个有效等价类：89，然后再确定若干个无效等价类：89.1，-1，101，x等

# 等价类的划分原则（续）

---

## ⑥细分等价类

在确知已划分的等价类中各元素在程序处理中的方式不同的情况下,则应再将该等价类进一步的划分为更小的等价类。

如：程序中，成绩输入为0~100的整数，程序对不同的分数进行优、良、中、及格等的转换。

除按第一条原则划分，还可进一步划分为更小的等价类：  
85~100（优），75~80（良）等。

# 等价类划分法的测试用例设计

---

## □ 设计测试用例

- 在确立了等价类后, 可建立等价类表, 列出所有划分出的等价类

输入条件	有效等价类	无效等价类
...	...	...
...	...	...

- 在设计测试用例时, 应同时考虑有效等价类和无效等价类测试用例的设计。

# 等价类划分法的测试用例设计

---

- 然后从划分出的等价类中按以下三个原则设计测试用例：
  - ①为每一个等价类规定一个唯一的编号。
  - ②设计一个新的测试用例，使其尽可能多地覆盖尚未被覆盖地有效等价类。重复这一步，直到所有的有效等价类都被覆盖为止。
  - ③设计一个新的测试用例，使其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。



# 常见等价类划分测试形式

---

□ 针对缺陷相关性假设，可将等价类测试分为

- 弱等价类测试(单缺陷假设)
- 强等价类测试(多缺陷假设)。

强等价类测试基于多缺陷假设，因此需要等价类笛卡尔积的每个元素对应的测试用例。

# 常见等价类划分测试形式

---

- 针对是否对无效数据进行测试，可以将等价类测试分为：标准等价类测试和健壮等价类测试。
  - 标准等价类测试——不考虑无效数据值，测试用例使用每个等价类中的一个值。
  - 健壮等价类测试——主要的出发点是考虑了无效等价类。对有效输入，测试用例从每个有效等价类中取一个值；对无效输入，一个测试用例有一个无效值，其他值均取有效值。

# 等价类划分举例

---

## □ 举例

例1 给定一个两变量x1和x2的函数f，如果f实现为一程序，则输入变量x1和x2将拥有以下边界及边界内的区间：

$a \leq x1 \leq d$ ，区间为[a,b)，[b,c)，[c,d]

$e \leq x2 \leq g$ ，区间为[e,f)，[f,g]

给出其强、弱、健壮等价类测试。

# 等价类划分举例

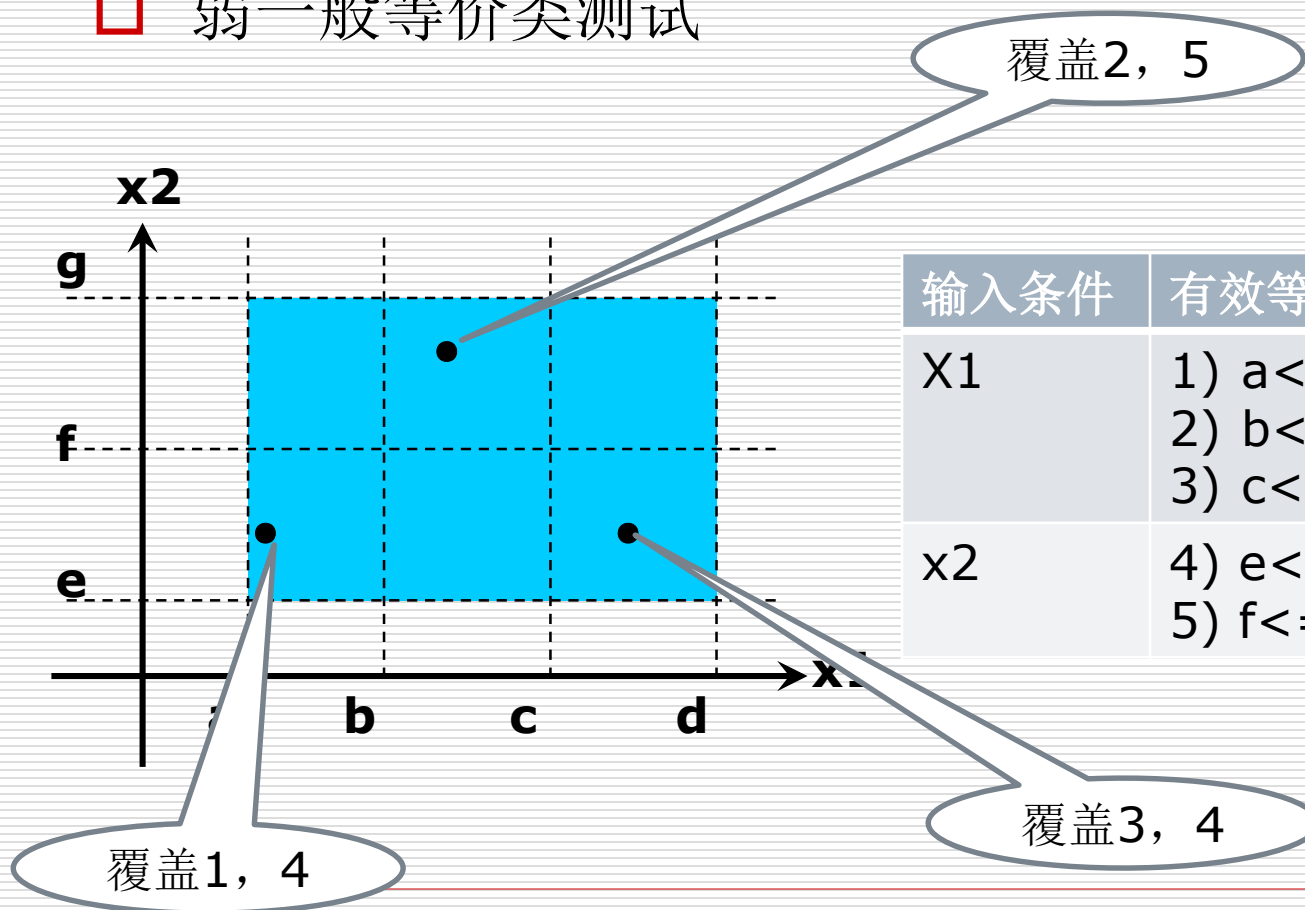
□ 确立等价类，建立等价类表，列出所有划分出的等价类

。

输入条件	有效等价类	无效等价类
X1	1) $a \leq x1 < b$ 2) $b \leq x1 \leq c$ 3) $c \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
x2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $x2 < e$ 9) $x2 > g$

# 等价类划分举例

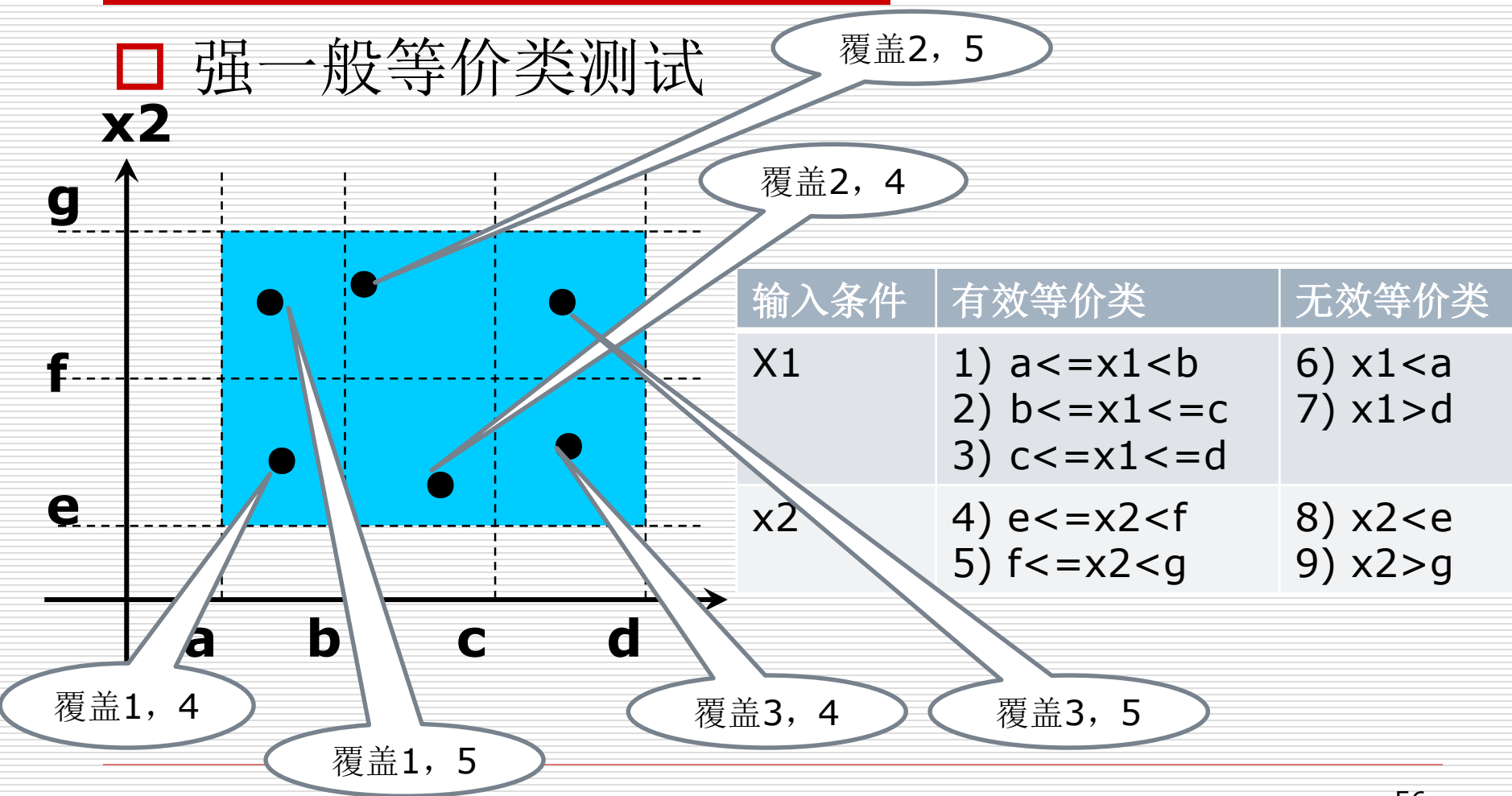
## 弱一般等价类测试



输入条件	有效等价类	无效等价类
x1	1) $a \leq x1 < b$ 2) $b \leq x1 \leq c$ 3) $c \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
x2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $x2 < e$ 9) $x2 > g$

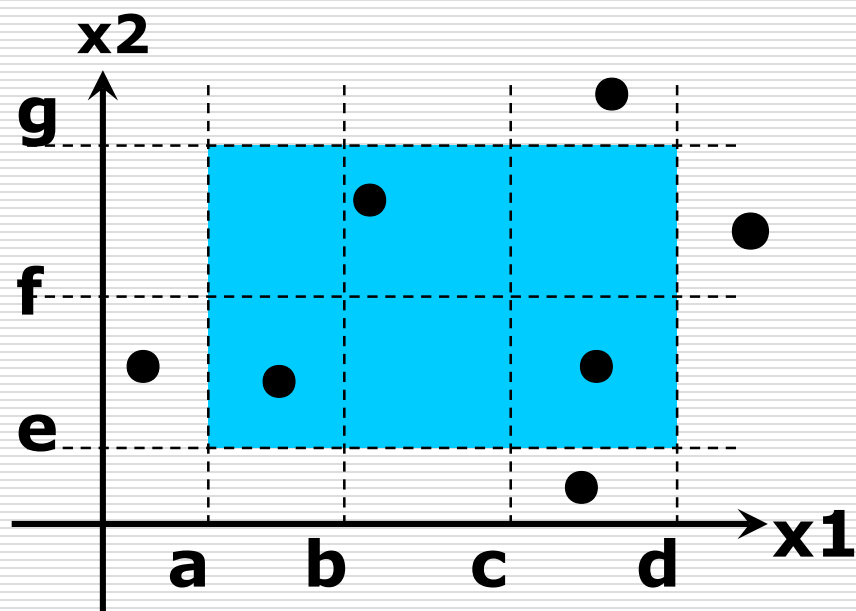
# 等价类划分举例

## □ 强一般等价类测试



# 等价类划分举例

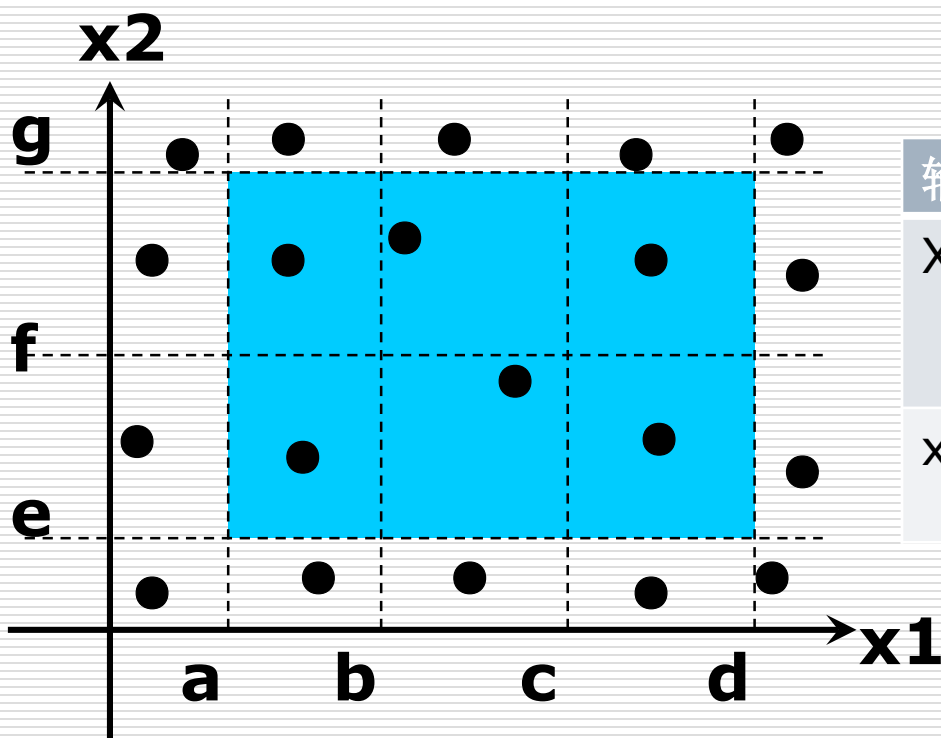
## □ 弱健壮等价类测试



输入条件	有效等价类	无效等价类
x1	1) $a \leq x1 < b$ 2) $b \leq x1 \leq c$ 3) $c \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
x2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $x2 < e$ 9) $x2 > g$

# 等价类划分举例

## □ 强健壮等价类测试



输入条件	有效等价类	无效等价类
$x_1$	1) $a \leq x_1 < b$ 2) $B \leq x_1 \leq c$ 3) $C \leq x_1 \leq d$	6) $x_1 < a$ 7) $x_1 > d$
$x_2$	4) $e \leq x_2 < f$ 5) $f \leq x_2 < g$	8) $x_2 < e$ 9) $x_2 > g$



# 等价类划分举例

---

- 例2：设有一个档案管理系统，要求用户输入以年月表示的日期。假设日期限定在1990年1月~2049年12月，并规定日期由6位数字字符组成，前4位表示年，后2位表示月。现用等价类划分法设计测试用例，来测试程序的“日期检查功能”。

# 等价类划分举例

## 1) 划分等价类并编号

下表等价类划分的结果

输入等价类	有效等价类	无效等价类
日期的类型及长度	①6位数字字符	②有非数字字符 ③少于6位数字字符 ④多于6位数字字符
年份范围	⑤在1990~2049之间	⑥小于1990 ⑦大于2049
月份范围	⑧在01~12之间	⑨等于00 ⑩大于12

# 等价类划分举例

## 2) 设计测试用例，以便覆盖所有有效等价类

在表中列出了3个有效等价类，编号分别为①、⑤、⑧，设计的测试用例如下：

测试数据	期望结果	覆盖的有效等价类
200211	输入有效	①、⑤、⑧

输入等价类	有效等价类	无效等价类
日期的类型及长度	①6位数字字符	②有非数字字符③…④…
年份范围	⑤在1990~2049之间	⑥小于1990 ⑦大于2049
月份范围	⑧在01~12之间	⑨等于00 ⑩大于12

# 等价类划分举例

3) 为每一个无效等价类设计一个测试用例，设计结果如下：

测试数据    期望结果    覆盖的无效等价类

95June	无效输入	②
20036	无效输入	③
2001006	无效输入	④
198912	无效输入	⑥
200401	无效输入	⑦
200100	无效输入	⑨
200113	无效输入	⑩

输入等价类	无效等价类
日期的类型及长度	②有非数字字符 ③少于6位数字字符 ④多于6位数字字符
年份范围	⑥小于1990 ⑦大于2049
月份范围	⑨等于00 ⑩大于12

# 等价类划分举例

---

□ 例3 某城市的电话号码由三部分组成，这三部分的名称和内容分别是：

地区码：空白或3位数字；

前缀：非“0”和非“1”开头的3位数字；

后缀：4位数字

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

# 等价类划分举例

第一步 划分等价类。包括4个有效等价类，11个无效等价类。

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, 少于三位数字⑥, 多于三位数字⑦
前 缀	200-999之间的三位数字③	有非数字字符⑧, 起始位为0⑨, 起始位为1⑩, 少于三位数字11, 多于三位数字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

# 等价类划分举例

## □ 第二步 设计测试用例

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

输入条件	有效等价类
地区码	空白①, 3位数字②
前 缀	200-999之间的 三位数字 ③
后 缀	4位数字④

对11个无效等价类，选择11个测试用例如下：

# 等价类划分举例

测试数据	测试范围	期望结果
(20A) 423-4567	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
(111) P12-9876	无效等价类⑧	无效
(111) 021-5432	无效等价类⑨	无效
(222) 121-1378	无效等价类⑩	无效
(333) 52-1378	无效等价类11	无效
(333) 6751-1378	无效等价类12	无效
(027) 345-6B01	无效等价类13	无效
(020) 345-601	无效等价类14	无效
(031) 478-23452	无效等价类15	无效



# 等价类划分举例

例4 保险公司计算保费费率的程序

某保险公司的人寿保险的保费计算方式为：

$$\text{投保额} \times \text{保险费率}$$

其中，保险费率依点数不同而有别，10点及10点以上保险费率为0.6%，10点以下保险费率为0.1%；而点数又是由投保人的年龄、性别、婚姻状况和抚养人数来决定，具体规则如下：

年龄			性别		婚姻		抚养人数
20~39	40~59	其它	M	F	已婚	未婚	1人扣0.5点 最多扣3点 (四舍五入取整)
6点	4点	2点	5点	3点	3点	5点	

对其保险费率的计算进行测试

# 计算保费费率的程序

---

1) 分析程序规格说明中给出和隐含的对输入条件的要求，列出等价类表（包括有效等价类和无效等价类）。

- 年龄：一位或两位非零整数，值的有效范围为1~99
- 性别：一位英文字符，只能取值 ‘M’或’ F’
- 婚姻：字符，只能取值 ‘已婚’或 ‘未婚’
- 抚养人数：空白或一位非零整数（1~9）
- 点数：一位或两位非零整数，值的范围为1~99

等价类表见下页

2) 根据1) 中的等价类表，设计能覆盖所有等价类的测试用例。

输入条件	有效等价类	编号	无效等价类	编号
年龄	20~39岁	1		
	40~59岁	2		
	1~19岁	3	小于1	12
	60~99岁		大于99	13
性别	单个英文字符	4	非英文字符	14
			非单个英文字符	15
	‘M’	5	除 ‘M’和 ‘F’之外的 其它单个字符	16
	‘F’	6		
婚姻	已婚	7	除’ 已婚’ 和’ 未婚’ 之外的其它字符	17
	未婚	8		
抚养人数	空白	9	除空白和数字之外的 的其它字符	18
	1~6人	10	小于1	19
	6~9人	11	大于9	20

测试用例 编号	输入数据				预期输出
	年龄	性别	婚姻	抚养人数	保险费率
1	27	F	未婚	空白	0.6%
2	50	M	已婚	2	0.6%
3	70	F	已婚	7	0.1%
4	0	M	未婚	空白	无法推算
5	100	F	已婚	3	无法推算
6	99	男	已婚	4	无法推算
7	1	Child	未婚	空白	无法推算
8	45	N	已婚	5	无法推算
9	38	F	离婚	1	无法推算
10	62	M	已婚	没有	无法推算
11	18	F	未婚	0	无法推算
12	40	M	未婚	10	无法推算

# 使用等价类划分法测试的实例

---

例5：输入三个整数a、b、c(假定在1~100之间)，分别作为三角形的三条边，现通过程序判断由三条边构成的三角形的类型为等边三角形、等腰三角形、一般三角形（特殊的还有直角三角形），以及构不成三角形。

用等价类划分方法为该程序进行测试用例设计。

（三角形问题的复杂之处在于输入与输出之间的关系比较复杂。）

# 使用等价类划分法测试的实例

---

□ 解法一：分析题目中给出和隐含的对输入条件的要求：

- (1) 整数      (2) 三个数      (3) 非零数
- (4) 正数      (5) 两边之和大于第三边
- (6) 等腰      (7) 等边

如果 **a**、**b**、**c** 满足条件 ( 1 ) ~ ( 4 )，则输出下列四种情况之一：

1. 如果不满足条件 (5)，则程序输出为 “ 非三角形 ”。
2. 如果三条边相等即满足条件 (7)，则程序输出为 “ 等边三角形 ”。
3. 如果只有两条边相等、即满足条件 (6)，则程序输出为 “ 等腰三角形 ”。
4. 如果三条边都不相等，则程序输出为 “ 一般三角形 ”。

列出等价类表并编号

		有效等价类	号码	无效等价类	号码
	输入条件	整数	1	a为非整数	12
				一边为非整数 b为非整数	13
				c为非整数	14
				a, b为非整数	15
				两边非整数 b, c为非整数	16
				a, c为非整数	17
				三边a,b,c均为非整数	18
		三个数	2	只给a	19
				只给一边 只给b	20
				只给c	21
				只给a, b	22
				只给两边 只给b, c	23
				只给a, c	24
				给出三个以上	25
		非零数	3	a为0	26
				一边为零 b为0	27
				c为0	28
				a, b为0	29
				二为边零 b, c为0	30
				a, c为0	31
				三边 <b>a,b,c</b> 均为 <b>0</b>	32

		有效等价类	号码	无效等价类	号码
输入条件	输入三个整数	正数	4	$a < 0$ 一边 $< 0$ $b < 0$ $c < 0$ $a < 0$ 且 $b < 0$ 二边 $< 0$ $a < 0$ 且 $c < 0$ $b < 0$ 且 $c < 0$ 三边均 $< 0$ ; $a < 0$ 且 $b < 0$ 且 $c < 0$	33 34 35 36 37 38 39
输出条件	构成一般三角形	$a + b > c$ $b + c > a$ $a + c > b$	5	$a + b < c$	40
			6	$a + b = c$	41
			7	$b + c < a$	42
				$b + c = a$	43
	构成等腰三角形	$a = b$ 且两边 $b = c$ 之和大 $a = c$ 于第三边	8	$a + c < b$	44
			9	$a + c = b$	45
			10		
	构成等边三角形	$a = b = c$	11		



# 使用等价类划分法测试的实例

---

覆盖有效等价类的测试用例：

a	b	c	覆盖等价类号码
3	4	5	(1) -- (7)
4	4	5	(1) -- (7) , (8)
4	5	5	(1) -- (7) , (9)
5	4	5	(1) -- (7) , (10)
4	4	4	(1) -- (7) , (11)

覆盖无效等价类的测试用例：

---

a	b	c	覆盖等价类号码	a	b	c	覆盖等价类号码
2.5	4	5	12	0	0	5	29
3	4.5	5	13	3	0	0	30
3	4	5.5	14	0	4	0	31
3.5	4.5	5	15	0	0	0	32
3	4.5	5.5	16	-3	4	5	33
3.5	4	5.5	17	3	-4	5	34
3.5	4.5	5.5	18	3	4	-5	35
3			19	-3	-4	5	36
	4		20	-3	4	-5	37
		5	21	3	-4	-5	38
3	4		22	-3	-4	-5	39
	4	5	23	3	1	5	40
3		5	24	3	2	5	41
3	4	5	25	3	1	1	42
0	4	5	26	3	2	1	43
3	0	5	27	1	4	2	44
3	4	0	28	3	4	1	45

非整数

边为0

非正数

非3个数

两边之和大于第三边

边为0

# 使用等价类划分法测试的实例

---

- 解法2：在多数情况下，是从输入域划分等价类的，但并非不能从被测程序的输出域反过来定义等价类，事实上，这对于三角形问题却是最简单的划分方法。

在三角形问题中，有四种可能的输出：等边三角形、等腰三角形、一般三角形和非三角形。利用这些信息能够确定下列输出（值域）等价类。

$R1 = \{ \langle a,b,c \rangle : \text{边为} a,b,c \text{的等边三角形} \}$

$R2 = \{ \langle a,b,c \rangle : \text{边为} a,b,c \text{的等腰三角形} \}$

$R3 = \{ \langle a,b,c \rangle : \text{边为} a,b,c \text{的一般三角形} \}$

$R4 = \{ \langle a,b,c \rangle : \text{边为} a,b,c \text{不能组成三角形} \}$

# 使用等价类划分法测试的实例

---

测试用例	a	b	c	预期输出
Test1	10	10	10	等边三角形
Test2	10	10	5	等腰三角形
Test3	3	4	5	一般三角形
Test4	4	1	2	非三角形

三角形问题的标准等价类测试用例

# 使用等价类划分法测试的实例

---

测试用例	a	b	c	预期输出
Test11	-1	5	5	a值超出输入值定义域
Test12	5	-1	5	b值超出输入值定义域
Test13	5	5	-1	c值超出输入值定义域
Test14	101	5	5	a值超出输入值定义域
Test15	5	101	5	b值超出输入值定义域
Test16	5	5	101	c值超出输入值定义域

三角形问题的弱健壮等价类测试用例

---

# NextDate 函数等价类测试用例

---

- 例6: NextDate 函数包含三个变量: month 、 day 和 year , 函数的输出为输入日期后一天的日期。 例如, 输入为 2006年3月7日, 则函数的输出为 2006年3月8日 。

要求输入变量 month 、 day 和 year 均为整数值, 并且满足下列条件:

1)  $1 \leq \text{month} \leq 12$

2)  $1 \leq \text{day} \leq 31$

3)  $1812 \leq \text{year} \leq 2012$

- 有效等价类为:

$M_1 = \{\text{月份}: 1 \leq \text{月份} \leq 12\}$

$D_1 = \{\text{日期}: 1 \leq \text{日期} \leq 31\}$

$Y_1 = \{\text{年}: 1812 \leq \text{年} \leq 2012\}$

# NextDate 函数等价类测试用例

---

- 若条件（1）~（3）中任何一个条件失效，则 NextDate 函数都会产生一个输出，指明相应的变量超出取值范围，比如“month 的值不在 1-12 范围当中”。显然还存在着大量的 year、month、day 的无效组合，NextDate 函数将这些组合作统一的输出：“无效日期”。其无效等价类为：

$M_2 = \{\text{月份: 月份} < 1\}$

$M_3 = \{\text{月份: 月份} > 12\}$

$D_2 = \{\text{日期: 日期} < 1\}$

$D_3 = \{\text{日期: 日期} > 31\}$

$Y_2 = \{\text{年: 年} < 1812\}$

$Y_3 = \{\text{年: 年} > 2012\}$

# NextDate 函数等价类测试用例

---

输入条件	有效等价类	无效等价类
月份	$M_1) 1 \leq \text{月份} \leq 12$	$M_2) \text{月份} < 1$ $M_3) \text{月份} > 12$
日期	$D_1) 1 \leq \text{日期} \leq 31$	$D_2) \text{日期} < 1$ $D_3) \text{日期} > 31$
年	$Y_1) 1812 \leq \text{年} \leq 2012$	$Y_2) \text{年} < 1812$ $Y_3) \text{年} > 2012$



# NextDate 函数等价类测试用例

## 弱一般等价类测试用例

月份	日期	年	预期输出
6	15	1912	1912年6月16日

## 强一般等价类测试用例同弱一般等价类测试用例

注：弱——有单缺陷假设  
健壮——考虑了无效值

输入条件	有效等价类	无效等价类
月份	$M_1) 1 \leq \text{月份} \leq 12$	$M_2) \text{月份} < 1$ $M_3) \text{月份} > 12$
日期	$D_1) 1 \leq \text{日期} \leq 31$	$D_2) \text{日期} < 1$ $D_3) \text{日期} > 31$
年	$Y_1) 1812 \leq \text{年} \leq 2012$	$Y_2) \text{年} < 1812$ $Y_3) \text{年} > 2012$

# NextDate 函数等价类测试用例

## □ 弱健壮等价类测试

用例ID	月份	日期	年	预期输出
WR <sub>1</sub>	6	15	1912	1912年6月16日
WR <sub>2</sub>	-1	15	1912	无效日期
WR <sub>3</sub>	13	15	1912	无效日期
WR <sub>4</sub>	6	-1	1912	无效日期
WR <sub>5</sub>	6	32	1912	无效日期
WR <sub>6</sub>	6	15	1811	无效日期
WR <sub>7</sub>	6	15	2013	无效日期

输入条件	无效等价类
月份	M <sub>2</sub> ) 月份<1 M <sub>3</sub> ) 月份>12
日期	D <sub>2</sub> ) 日期<1 D <sub>3</sub> ) 日期>31
年	Y <sub>2</sub> ) 年<1812 Y <sub>1</sub> ) 年>2012

# NextDate 函数等价类测试用例

---

## □ 强健壮等价类测试（27个中的一个“角”）

用例ID	月份	日期	年	预期输出
SR <sub>1</sub>	-1	15	1912	月份不在1~12中
SR <sub>2</sub>	6	-1	1912	日期不在1~31中
SR <sub>3</sub>	6	15	1811	年份不在1812~2012中
SR <sub>4</sub>	-1	-1	1912	两个无效一个有效
SR <sub>5</sub>	6	-1	1811	两个无效一个有效
SR <sub>6</sub>	-1	15	1811	两个无效一个有效
SR <sub>7</sub>	-1	-1	1811	三个无效

# 习题

---

问题：给出下面的有效和无效等价类

输入条件：“...统计全国各省、市、自治区的人口...”

输入条件：“标识符应以字母开头...”

输入条件：长度为**1-20**的字符串

输入条件：数据库中的值域, **CHAR(20), NOT NULL**

# 边界值分析法

---

## □ 什么是边界值分析法

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

## □ 为什么使用边界值分析法？

无数的测试实践表明，大量的故障往往发生在输入定义域或输出值域的边界上，而不是在其内部。因此，针对各种边界情况设计测试用例，通常会取得很好的测试效果。

# 边界值分析法

---

## □ 举例 —— 常见的边界值

- 对16-bit 的整数而言 32767 和 -32768 是边界
- 屏幕上光标在最左上、最右下位置
- 报表的第一行和最后一行
- 数组元素的第一个和最后一个
- 循环的第 0 次、第 1 次和倒数第 2 次、最后一次

# 边界值分析法

---

## □ 怎样用边界值分析法设计测试用例？

- （1）首先确定边界情况。通常输入或输出等价类的边界就是应该着重测试的边界情况。
- （2）选取正好等于、刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值。

# 边界值分析法

---

## □ 与等价划分的区别

- 边界值分析使用与等价类划分法相同的划分，只是边界值分析假定错误更多地存在于划分的边界上，因此在等价类的边界上以及两侧的情况设计测试用例。
- 边界值分析不是从某等价类中随便挑一个作为代表，而是使这个等价类的每个边界都要作为测试条件。
- 边界值分析不仅考虑输入条件，还要考虑输出空间产生的测试情况。



# 边界值分析法

---

## □ 例7：测试计算平方根的函数

输入/输出：实数；

规格说明：当输入一个0或比0大的数的时候，返回其正平方根；  
当输入一个小于0的数时，显示错误信息并返回0。

## ■ 等价类划分

可以考虑作出如下划分：

输入：(i)  $\geq 0$  和 (ii)  $< 0$

输出：(a)  $\geq 0$  和 (b) Error

测试用例有两个：

输入4，输出2。 对应于 (i) 和 (a) 。

输入-10，输出0和错误提示。对应于 (ii) 和 (b) 。

# 边界值分析法

---

## ■ 边界值分析：

划分(i)的边界为0和最大正实数；划分(ii)的边界为最小负实数和0。由此得到以下测试用例：

输入 {最小负实数}

输入 {绝对值很小的负数}

输入 0

输入 {绝对值很小的正数}

输入 {最大正实数}

还要考虑输出(省略)

# 边界值分析法

---

## □ 基于边界值分析方法选择测试用例的原则

- 1) 如果输入条件规定了值的范围,则应取刚达到这个范围的边界的值,以及刚刚超越这个范围边界的值作为测试输入数据。

例如, 如果程序的规格说明中规定: “重量在**10**公斤至**50**公斤范围内的邮件, 其邮费计算公式为.....”。作为测试用例, 我们应取**10**及**50**, 还应取**10.01**,**49.99**,**9.99**及**50.01**等。

# 边界值分析法

---

2) 如果输入条件规定了值的个数,则用最大个数,最小个数,比最小个数少一,比最大个数多一的数作为测试数据。

比如,一个输入文件应包括1~255个记录,则测试用例可取1和255,还应取0及256等。

# 边界值分析法

---

3) 将规则1) 和2) 应用于输出条件，即设计测试用例使输出值达到边界值及其左右的值。

例如，某程序的规格说明要求计算出“每月保险金扣除额为0至1165.25元”，其测试用例可取0.00及1165.24、还可取-0.01及1165.26等。

再如一程序属于情报检索系统，要求每次“最少显示1条、最多显示4条情报摘要”，这时我们应考虑测试用例包括1和4，还应包括0和5等。

# 边界值分析法

---

- 4) 如果程序的规格说明给出的输入域或输出域是有序集合（如有序表、顺序文件等）,则应选取集合的第一个元素和最后一个元素作为测试用例。
- 5) 如果程序中使用了一个内部数据结构,则应当选择这个内部数据结构的边界上的值作为测试用例。
- 6) 分析规格说明,找出其它可能的边界条件。

# 边界值分析

## □ 上述原则的一些说明

- 1) 在通常情况下，软件测试所包含的边界检验有几种类型： 数字、字符、位置、质量、大小、速度、方位、尺寸、空间等。相应地，以上类型的边界值应该在：最大/最小、首位/末位、上/下、最快/最慢、最高/最低、 最短/最长、 空/满等情况下。 如：

项	边界值	测试用例的设计思路
数值	最小值-1/最大值+1	假设某软件的数据输入域要求输入 <b>5</b> 位的数据值，可以使用 <b>10000</b> 作为最小值、 <b>99999</b> 作为最大值；然后使用刚好小于 <b>5</b> 位和大于 <b>5</b> 位的 数值来作为边界条件。
空间	小于空余空间一点/大于满空间一点	例如在用 <b>U</b> 盘存储数据时，使用比剩余磁盘空间大一点（几 <b>KB</b> ）的文件作为边界条件。

# 边界值分析

---

## 2)内部边界值条件

多数情况下，边界值条件可从软件的规格说明或常识中得到，最终用户也很容易发现问题。然而，某些边界值条件是不需要呈现给用户的，用户很难注意到，但同时确实属于检验范畴内的边界条件，称为内部边界值条件或子边界值条件。内部边界值条件主要有下面几种：

- 数值的边界值检验
- 字符的边界值检验
- 其它边界值检验

如：



# 数值的边界值检验

- 计算机是基于二进制进行工作的，因此，软件的任何数值运算都有一定的范围限制。

计算机数值范围

项	范围或值
位 (bit)	0 或 1
字节 (byte)	0 ~ 255
字 (word)	0~65535 (单字) 或 0~4294967295 (双字)
K (Kilo)	$2^{10}=1024$ (字节)
M (Meg)	$2^{20}=1048576$ (字节)
G (Giga)	$2^{30}=1073741824$ (字节)
T (Tera)	$2^{40}=1,099,511,627,776$ (字节)
P (Peta)	$2^{50}=1,125,899,906,842,624$ (字节)
E (Exa)	$2^{60}=1,152,921,504,606,846,976$ (字节)
Z (Zetta)	$2^{70}=1,180,591,620,717,411,303,424$ (字节)
Y (Yotta)	$2^{80} = 1208925819614629174706176$ (字节)

# 字符的边界值检验

- 在计算机软件中，字符也是很重要的表示元素，其中ASCII和Unicode是常见的编码方式。下表中列出了一些常用字符对应的ASCII码值。

字符	ASCII码值	字符	ASCII码值
空 (null)	0	A	65
空格 (space)	32	a	97
斜杠 (/)	47	Z	90
0	48	z	122
冒号 (:) )	58	单引号 ( ' )	96
@	64		

# 边界值分析

---

## 3) “单故障”假设

边界值分析法是基于可靠性理论中称为“单故障”的假设，即有两个或两个以上故障同时出现而导致软件失效的情况很少，也就是说，软件失效基本上是由单故障引起的。

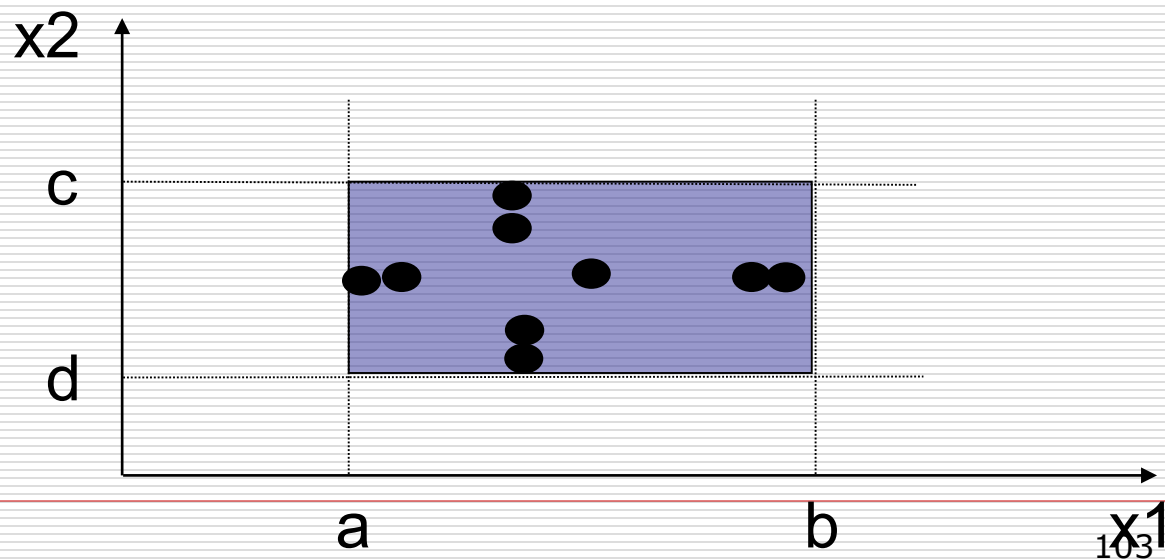
因此，在多变量的边界值分析法中获取测试用例的方法是：

- (1) 每次保留程序中一个变量，让其余的变量取正常值，被保留的变量依次取min、min+、nom、max-和max。
- (2) 对程序中的每个变量重复 (1) 。

# 边界值分析法 - 举例

□ 例8：有两个输入变量 $x_1(a \leq x_1 \leq b)$ 和 $x_2(c \leq x_2 \leq d)$ 的程序F的边界值分析测试用例如下：

{  $\langle x_{1nom}, x_{2min} \rangle$ ,  $\langle x_{1nom}, x_{2min+} \rangle$ ,  $\langle x_{1nom}, x_{2nom} \rangle$ ,  
 $\langle x_{1nom}, x_{2max} \rangle$ ,  $\langle x_{1nom}, x_{2max-} \rangle$ ,  $\langle x_{1min}, x_{2nom} \rangle$ ,  
 $\langle x_{1min+}, x_{2nom} \rangle$ ,  $\langle x_{1max}, x_{2nom} \rangle$ ,  $\langle x_{1max-}, x_{2nom} \rangle$  }



# 边界值分析法 - 举例

---

- 例9：有二元函数 $f(x,y)$ ，其中 $x \in [1,12]$ ， $y \in [1,31]$ 。  
则采用边界值分析法设计的测试用例是：

{  $\langle 1,15 \rangle$ ,  $\langle 2,15 \rangle$ ,  $\langle 11,15 \rangle$ ,  $\langle 12,15 \rangle$ ,  $\langle 6,15 \rangle$ ,  $\langle 6,1 \rangle$ ,  
 $\langle 6,2 \rangle$ ,  $\langle 6,30 \rangle$ ,  $\langle 6,31 \rangle$  }

- 推论：对于一个含有 $n$ 个变量的程序，采用边界值分析法测试程序会产生 $4n+1$ 个测试用例（一个变量取最小值，略高于最小值，正常值，略低于最大值，最大值外，其余变量取正常值。对每个变量都重复进行）。

# 边界值分析法 - 举例

---

- 再如：有函数 $f(x,y,x)$ ，其中 $x \in [1900, 2100]$ ， $y \in [1, 12]$ ， $z \in [1, 31]$ 的。请写出该函数采用边界值分析法设计的测试用例。

共13个，分别是：{  $\langle 2000, 6, 1 \rangle$ ,  $\langle 2000, 6, 2 \rangle$ ,  
 $\langle 2000, 6, 30 \rangle$ ,  $\langle 2000, 6, 31 \rangle$ ,  $\langle 2000, 1, 15 \rangle$ ,  
 $\langle 2000, 2, 15 \rangle$ ,  $\langle 2000, 11, 15 \rangle$ ,  $\langle 2000, 12, 15 \rangle$ ,  
 $\langle 1900, 6, 15 \rangle$ ,  $\langle 1901, 6, 15 \rangle$ ,  $\langle 2099, 6, 15 \rangle$ ,  
 $\langle 2100, 6, 15 \rangle$ ,  $\langle 2000, 6, 15 \rangle$  }

# 边界值分析法 - 举例

例10：现有一个学生  
标准化考试批阅试  
卷,产生成绩报告  
的程序。其规格说  
明如下:程序的输  
入文件由一些有  
**80**个字符的记录  
组成,如右图所示,  
所有记录分为**3**组:

(试题部分)				
标 题				
1				80
试题数		标准答案 (1~50 题)		2
1	3 4	9 10	59 60	79 80
试题数		标准答案 (51~100 题)		2
1	3 4	9 10	59 60	79 80
.....				
(学生答卷部分)				
学号 1	学生答案 (1~50 题)			3
1	9 10			59 60 79 80
学号 1	学生答案 (51~100 题)			3
1	9 10			59 60 79 80
.....				

# 边界值分析法 - 举例

## ① 标题:

这一组只有一个记录，  
其内容为输出成绩报  
告的名字。

(试题部分)

标 题				
1				80
试题数		标准答案 (1~50 题)		2
1	3 4	9 10	59 60	79 80
试题数		标准答案 (51~100 题)		2
1	3 4	9 10	59 60	79 80

.....

(学生答卷部分)

学号 1	学生答案 (1~50 题)			3
1	9 10	59 60	79 80	
学号 1	学生答案 (51~100 题)			3
1	9 10	59 60	79 80	

.....



# 边界值分析法 - 举例

- ②试卷各题标准答案记录：  
每个记录均在第80个字符处标以数字“2”。  
该组的第一个记录的第1至第3个字符为题目编号（取值为1—999）。第10至第59个字符给出第1至第50题的答案（每个合法字符表示一个答案）。  
该组的第2，第3.....个记录相应为第51至第100，第101至第150，...题的答案。

(试题部分)			
标 题			
1			80
试题数		标准答案 (1~50 题)	2
1	3 4	9 10	59 60 79 80
试题数		标准答案 (51~100 题)	2
1	3 4	9 10	59 60 79 80
.....			
(学生答卷部分)			
学号 1	学生答案 (1~50 题)		3
1	9 10	59 60	79 80
学号 1	学生答案 (51~100 题)		3
1	9 10	59 60	79 80
.....			

# 边界值分析法 - 举例

③每个学生的答卷描述：  
该组中每个记录的第80个字符均为数字“3”。  
每个学生的答卷在若干个记录中给出。如甲的首记录第1至第9字符给出学生姓名及学号，第10至第59字符列出的是甲所做的第1至第50题的答案。  
若试题数超过50，则第2，第3.....纪录分别给出他的第51至第100，第101至第150.....题的解答。  
然后是学生乙的答卷记录。

(试题部分)				
标 题				
1				80
试题数		标准答案 (1~50 题)		2
1	3 4	9 10	59 60	79 80
试题数		标准答案 (51~100 题)		2
1	3 4	9 10	59 60	79 80
.....				
(学生答卷部分)				
学号 1		学生答案 (1~50 题)		3
1		9 10	59 60	79 80
学号 1		学生答案 (51~100 题)		3
1		9 10	59 60	79 80
.....				

# 边界值分析法 - 举例

---

程序的输出有4个报告：

- a)按学号排列的成绩单，列出每个学生的成绩、名次。
- b)按学生成绩排序的成绩单。
- c)平均分数及标准偏差的报告。
- d)试题分析报告。按试题号排序，列出各题学生答对的百分比。

解答：分别考虑输入条件和输出条件，以及边界条件。给出下表所示的输入条件及相应的测试用例。

# 边界值分析法 - 举例

输入条件	测试用例			
输入文件	空输入文件			
标题	没有标题 标题只有一个字符 标题有 80 个字符	(试题部分)		
试题数	试题数为 1 试题数为 50 试题数为 51 试题数为 100 试题数为 0 试题数含有非数字字符	标 题		
		1		
		试题数		标准答案 (1~50 题)
		1	3 4 9 10	59 60
		试题数		标准答案 (51~100 题)
		1	3 4 9 10	59 60
标准答案记录	没有标准答案记录，有标题 标准答案记录多于一个 标准答案记录少一个	(学生答卷部分)		
		学号 1	学生答案 (1~50 题)	
		1	9 10	59 60
学生人数	0 个学生 1 个学生 200 个学生 201 个学生	学号 1	学生答案 (51~100 题)	
		1	9 10	59 60
		.....		
		.....		

# 边界值分析法 - 举例

学生答题	某学生只有一个回答记录，但有两个标准答案记录 该学生是文件中的第一个学生 该学生是文件中的最后一个学生（记录数出错的学生）
学生答题	某学生有两个回答记录，但只有一个标准答案记录 该学生是文件中的第一个学生（记录数出错的学生） 该学生是文件中的最后一个学生
学生成绩	所有学生的成绩都相等 每个学生的成绩都不相等 部分学生的成绩相同 （检查是否能按成绩正确排名次） 有个学生 0 分 有个学生 100 分

（试题部分）

标 题		
1		
试题数		标准答案（1~50 题）
1	3 4	9 10
试题数		标准答案（51~100 题）
1	3 4	9 10

（学生答卷部分）

学号 1	学生答案（1~50 题）
1	9 10
学号 1	学生答案（51~100 题）
1	9 10

# 边界值分析法 - 举例

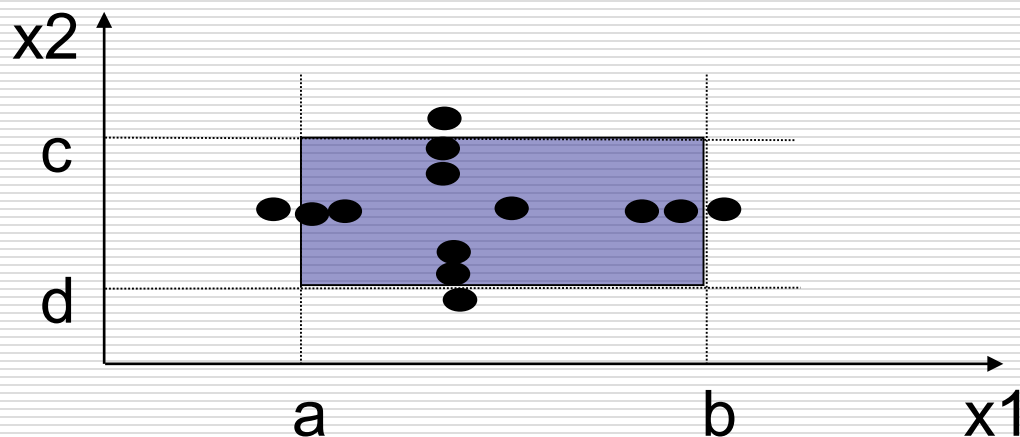
右表为输出条件及相应的测试用例表。

- a)按学号排列。
- b)按成绩排序。
- c)平均分数及标准偏差的报告。
- d)试题分析报告。按试题号排序，列出各题学生答对的百分比。

输出条件	测试用例
输出报告 a、b	有个学生的学号最小（检查按序号排序是否正确） 有个学生的学号最大（检查按序号排序是否正确） 适当的学生人数，使产生的报告刚好满一页（检查打印页数） 学生人数比刚才多出 1 人（检查打印换页）
输出报告 c	平均成绩 100 平均成绩 0 标准偏差为最大值（有一半的 0 分，其他 100 分） 标准偏差为 0（所有成绩相等）
输出报告 d	所有学生都答对了第一题 所有学生都答错了第一题 所有学生都答对了最后一题 所有学生都答错了最后一题 选择适当的试题数，是第四个报告刚好打满一页 试题数比刚才多 1，使报告打满一页后，刚好剩下一题未打

# 健壮性测试

- 健壮性测试是作为边界值分析的一个简单的扩充，它除了对变量的5个边界值分析取值外，还需要增加一个略大于最大值(max+)以及略小于最小值(min-)的取值，检查超过极限值时系统的情况。因此，对于有n个变量的函数采用健壮性测试需要 $6n+1$ 个测试用例。
- 前面例8中的程序F的健壮性测试如下图所示：



- 练习：请为例9中的函数 $f(x,y)$ 写出相应的健壮性测试用例。

# 边界值分析法 - 举例

---

## □ 例11 三角形问题的边界值分析测试用例

在三角形问题描述中，三角形每边边长的取范围值设值为[1, 100]。



测试用例	a	b	c	预期输出
Test 1	60	60	1	等腰三角形
Test2	60	60	2	等腰三角形
Test3	60	60	60	等边三角形
Test4	50	50	99	等腰三角形
Test5	50	50	100	非三角形
Test6	60	1	60	等腰三角形
Test7	60	2	60	等腰三角形
Test8	50	99	50	等腰三角形
Test9	50	100	50	非三角形
Test10	1	60	60	等腰三角形
Test11	2	60	60	等腰三角形
Test12	99	50	50	等腰三角形
Test13	100	50	50	非三角形

# 边界值分析法 - 举例

---

## □ 例12 NextDate函数的健壮性边界值分析测试用例

在NextDate函数中，隐含规定了变量mouth和变量day的取值范围为 $1 \leq \text{mouth} \leq 12$ 和 $1 \leq \text{day} \leq 31$ ，并设定变量year的取值范围为 $1912 \leq \text{year} \leq 2050$ 。

测试用例	mouth	day	year	预期输出
Test 1	6	15	1911	1911.6.16
Test2	6	15	1912	1912.6.16
Test3	6	15	1913	1913.6.16
Test4	6	15	1975	1975.6.16
Test5	6	15	2049	2049.6.16
Test6	6	15	2050	2050.6.16
Test7	6	15	2051	2051.6.16
Test8	6	-1	2001	day超出[1...31]
Test9	6	1	2001	2001.6.2
Test10	6	2	2001	2001.6.3
Test11	6	30	2001	2001.7.1
Test12	6	31	2001	输入日期超界
Test13	6	32	2001	day超出[1...31]
Test14	-1	15	2001	Mouth超出[1...12]
Test15	1	15	2001	2001.1.16
Test16	2	15	2001	2001.2.16
Test17	11	15	2001	2001.11.16
Test18	12	15	2001	2001.12.16
Test19	13	15	2001	Mouth超出[1...12]

# 边界值分析法 - 举例

---

## □ 例13 找零钱最佳组合

假设商店货品价格(**R**) 都不大于**100**元（且为整数），若顾客付款(**P**)在**100**元内，现有一个程序能在每位顾客付款后给出找零钱的最佳组合（找给顾客货币张数最少）。假定此商店的货币面值只包括：**50元(N50)**、**10元(N10)**、**5元(N5)**、**1元(N1)** 四种。

请结合等价类划分法和边界值分析法为上述程序设计出相应的测试用例。

# 找零钱最佳组合

## □ 分析输入的情形

无效等价类	有效等价类	无效等价类
$R \leq 0$	$0 < R \leq 100$	$R > 100$
$P < R$	$R \leq P \leq 100$	$P > 100$

1、对R取边界值：

$R = -1$ (无效)     $R = 0$ (无效)     $R = 1$

$R = 100$      $R = 101$ (无效)

2、对P取边界值

$P < R$ (无效)     $P = R$      $P = 100$      $P = 101$ (无效)

无效等价类	有效等价类	无效等价类
$R \leq 0$	$0 < R \leq 100$	$R > 100$
$P < R$	$R \leq P \leq 100$	$P > 100$

# 找零钱最佳组合

## □ 考虑输出——找零数额（N表示找零数额）

1、无效输入（不找零）：输出为相应错误提示信息。

$R > 100$

$R \leq 0$

$P > 100$

$P < R$

2、有效输入（找零）：

$0 < R \leq 100 \ \&\& \ R \leq P \leq 100$

此时考虑的输出：（假设 $N = P - R$ 正确,不考虑此种情况无效输出）

$0 \leq N < 5$

$5 \leq N < 10$

$10 \leq N < 50$

$50 \leq N < 100$

用边界值法，N分别取：0、4、5、9、10、49、50、99

## □ 设计测试用例，覆盖上述的输入/输出边界组合，如下：

1、对R取边界值：R = -1(无效) R = 0(无效)

R=1 R=100 R = 101(无效)

2、对P取边界值：P < R(无效) P=R

P = 100 P=101(无效)

# 找零钱最佳组合

测试用例	货品价格R	付款金额P	期望结果	说明
test1	-1	X	无效货品价格	无效边界
test2	0	X	无效货品价格	无效边界
test3	101	X	无效货品价格	无效边界
test4	100	99	无效付款	无效边界
test5	100	101	无效付款	无效边界
test6	1	100	N50=1, N10=4, N5=1, N1=4	R=1/N=99
test7	100	100	N50=0, N10=0, N5=0, N1=0	R=100/P=R P=100/N=0
Test8	96	100	N50=0, N10=0, N5=0, N1=4	N=4
Test9	95	100	N50=0, N10=0, N5=1, N1=0	N=5
test10	91	100	N50=0, N10=0, N5=1, N1=4	N=9
test11	90	100	N50=0, N10=1, N5=0, N1=0	N=10
Test12	51	100	N50=0, N10=4, N5=1, N1=4	N=49
Test13	50	100	N50=1, N10=0, N5=0, N1=0	N=50

# 边界值分析法

---

## □ 最坏情况测试

软件失效是由多故障引起的。

对每个变量首先进行包含最小值、略高于最小值、正常值、略低于最大值和最大值五元素集合的测试，然后对这些集合进行笛卡儿乘积计算，以生成测试用例。

如：三角形问题和NextDate函数问题的最坏情况测试有：  
 $5 \times 5 \times 5 = 125$ 种测试用例。

N个变量的最坏情况测试会产生 $5^n$ 个测试用例

N个变量的健壮最坏情况测试会产生 $7^n$ 个测试用例



# 错误推测法

---

## □ 错误推测法：

基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法。

- 例如，输入数据和输出数据为0的情况；输入表格为空格或输入表格只有一行。这些都是容易发生错误的情况。可选择这些情况下的例子作为测试用例。

# 错误推测法

---

- 例如，针对例10，采用错误推测法还可补充设计一些测试用例：
  - 1、 程序是否把空格作为回答
  - 2、 在回答记录中混有标准答案记录
  - 3、 除了标题记录外，还有一些的记录最后一个字符即不是2也不是3
  - 4、 有两个学生的学号相同
  - 5、 试题数是负数。

# 错误推测法

---

- 再如，测试一个对线性表（比如数组）进行排序的程序，可推测列出以下几项需要特别测试的情况：
  - 1) 输入的线性表为空表；
  - 2) 表中只含有一个元素；
  - 3) 输入表中所有元素已排好序；
  - 4) 输入表已按逆序排好；
  - 5) 输入表中部分或全部元素相同。