



软件测试

黑盒测试技术 (功能测试)

——等价类划分法、边界值法

黄轶文

主要内容

- 1 功能测试
- 2 等价类划分法

1 功能测试

❖ 功能测试，依据产品设计规格说明书完成对产品功能进行操作，以验证系统是否满足用户的功能性需求

- 界面（UI）测试
- 数据输入/输出
- 操作（场景）
- （业务）逻辑
- 接口

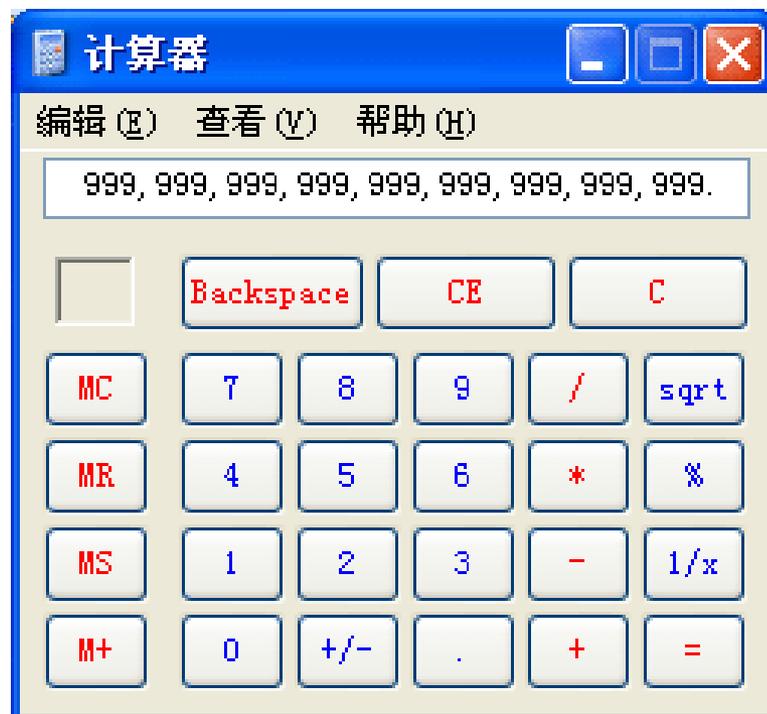
1 功能测试用例设计步骤

- ❖ 根据功能结构划分功能模块或子模块。
- ❖ 针对每一个功能模块，设计工作流程图或数据流图，确定逻辑路径、使用场景及其测试点。
- ❖ 针对各个测试点（条件、数据、路径、场景等），设计测试用例。从上到下的。
- ❖ 测试用例的评审和修改。

2 黑盒测试——测试用例的设计（1）

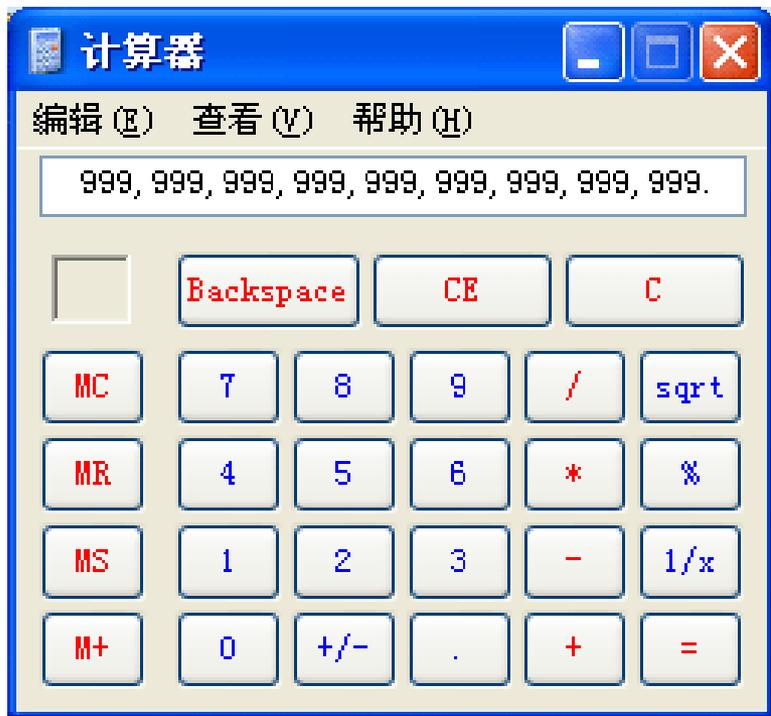
- **2.1 等价类划分法**
- 2.2 边界值分析法
- 2.3 因果图法
- 2.4 决策表方法
- 2.5 正交试验设计方法
- 2.6 场景法

测试计算器



问题：如果不采用等级类方法？

连一个计算器程序的测试工作量都是惊人的！



能否找到某一个数据，对它的测试可以代表某一类数据（很多个数据）的测试？

当然。等价划分法。

等价类测试用例- Example



等价类1: 负实数

等价类2: 负整数

等价类3: 0

等价类4: 正整数

等价类5: 正实数

同区间数据运算、交叉区间混合运算

- 举例：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。

等价类法

❖ 定义：将程序可能的输入数据分成若干个子集，从每个子集选取一个代表性的数据作为测试用例

❖ 等价类划分法将不能穷举的测试数据进行合理分类，变成有限的、较少的若干数据来代表更为广泛的数据输入。

测试某等价类的代表值就是等效于对于这一类其他值的测试。

❖ 等价类是某个输入域的子集，在该子集中每个输入数据的作用是等效的。

❖ 在分析需求规格说明的基础上划分等价类，列出等价类表。

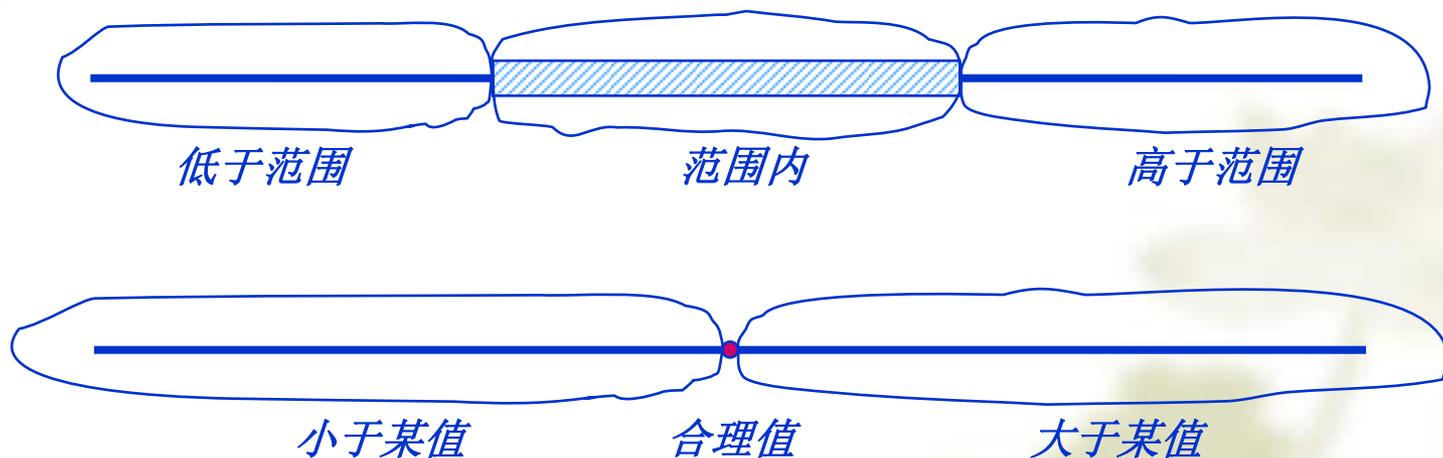
有效等价类和无效等价类

- ❖ 有效等价类是有意义的、合理的输入数据，可以检查程序是否实现了规格说明中所规定的功能和性能
- ❖ 无效等价类和有效等价类相反，即不满足程序输入要求或者无效的输入数据构成的集合

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。经过正反的测试才能确保软件具有更高的可靠性。

确定等价类的方法

(1) 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。



例如程序要求输入为3位正整数 x 。

它们的有效等价类和无效等价类？

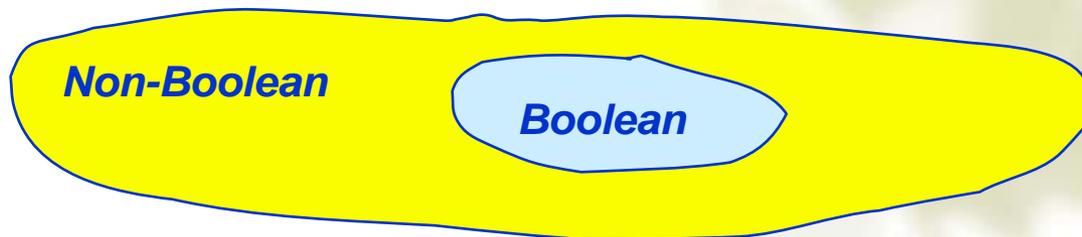
例如每个学生每学期只能选修1-3门课。

确定等价类的方法

(2) 在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情况下，可以确立一个有效等价类和多个无效等价类。



(3) 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



确定等价类的方法

(4) 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, 这种情况下可确立 n 个有效等价类和一个无效等价类。

枚举类型数据

例: 程序输入 x 取值于一个固定的枚举类型 $\{1,3,7,15\}$, 且程序中对这4个数值分别进行了处理, 则有效等价类为 $x=1$ 、 $x=3$ 、 $x=7$ 、 $x=15$, 无效等价类为 $x \neq 1,3,7,15$ 的值的集合。

确定等价类的方法

(5) 更复杂的情况是，输入数据只是要求符合某几个规则，这时可能存在多个有效等价类(符合规则)和若干个无效等价类(从不同角度违反规则)。例如，邮件地址和用户名的输入。

- 用户名要求输入26个英语字母和10个阿拉伯数字构成的、长度不超过20位。
- 有效的email地址，必须含有“@”，“@”后面的格式为x.y，email地址不能带有一些特殊符号，如“/\#&等。

等价类方法的应用步骤

- a) 数据分类，分出有效等价类和无效等价类
- b) 针对有效等价类，进一步分割，直至不能划分为止，形成等价类表，为每一等价类规定一个唯一的编号
- c) 就每一个具体的等价类，设计一个测试用例，直到所有有效等价类均被测试用例所覆盖
- d) 对无效等价类进行相同的处理

输入条件	有效等价类	无效等价类
...
...

使用等价类划分法测试的实例

电话号码在应用程序中也是经常能见到，我国固定电话号码一般由两部分组成。

- 地区码：以0开头的3位或者4位数字。
- 电话号码：以非0、非1开头的7位或者8位数字。

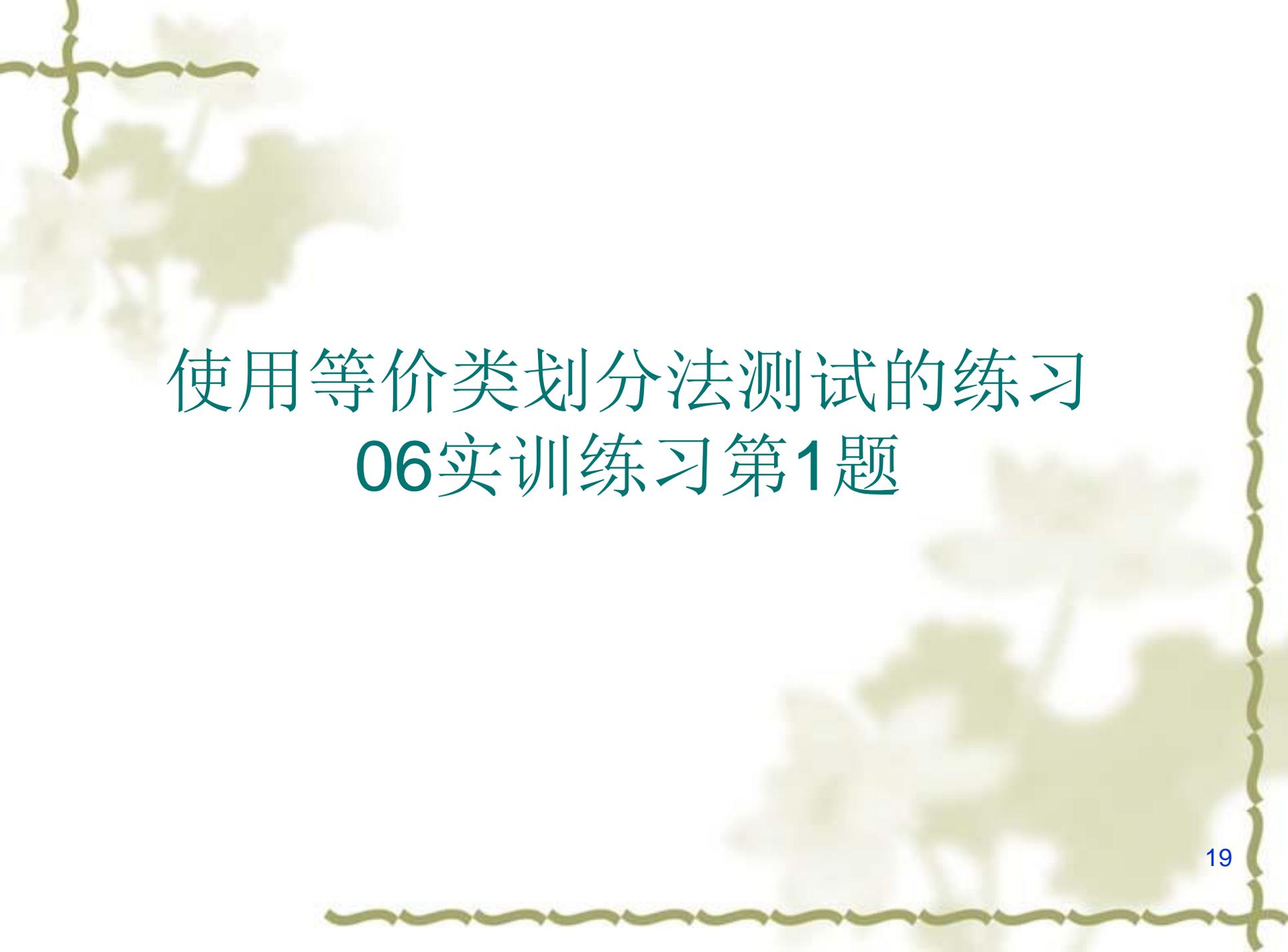
应用程序会接受一切符合上述规定的电话号码，而拒绝不符合规定的号码。在设计其测试用例时，就可用等价类划分法。

使用等价类划分法测试的实例(续)

表 6-1

电话号码的等价类方法应用

输入数据		有效等价类		无效等价类	
地区码	1	以 0 开头的 3 位区码	3	以 0 开头的小于 3 位的数字串	
	2	以 0 开头的 4 位区码	4	以 0 开头的大于 4 位的数字串	
			5	以非 0 开头的数字串	
			6	以 0 开头的含有非数字的字符串	
电话号码	7	以非 0、非 1 开头的 7 位号码	9	以 0 开头的数字串	
	8	以非 0、非 1 开头的 8 位号码	10	以 1 开头的数字串	
			11	以非 0、非 1 开头的小于 7 位数字串	
			12	以非 0、非 1 开头的大于 8 数字串	
			13	以非 0、非 1 开头的含有非法字符 7 或者 8 位字符串	
测试用例			01 81234567	覆盖 3	
			05511 6123456	覆盖 4	
			10 81234567	覆盖 5	
	010	6123456	覆盖 1、7	025g 81234567	覆盖 6
	025	81234567	覆盖 1、8	010 06123456	覆盖 9
	0551	7123456	覆盖 2、7	0551 1123456	覆盖 10
	0571	92345678	覆盖 2、8	0551 612345	覆盖 11
				0571 912345678	覆盖 12
			0571 912345ab	覆盖 13	



使用等价类划分法测试的练习

06实训练习第1题

使用等价类划分法测试的练习

1、某城市电话号码由三部分组成，分别是：

地区码——空白或三位数字；

前缀——非‘0’或‘1’开头的三位数字；

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电话号码。

(1) 分析程序的规格说明，列出输入域等价类表（包括有效和无效等价类）。

(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，要求包括输入数据和预期输出，并指出各个测试用例所覆盖的等价类编号。

1、输入域等价类表

输入条件	有效等价类	编号	无效等价类	编号
地区码	空白	1	有非数字字符	5
	3 位数字	2	少于 3 位数字	6
			多于 3 位数字	7
前缀	200~999	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

2、根据等价类表设计的测试用例

测试用例 编号	输入数据			预期输出	覆盖等价类 编号
	地区码	前缀	后缀		
1	空白	123	4567	接受 (有效)	1, 3, 4
2	123	805	9876	接受 (有效)	2, 3, 4
3	20A	123	4567	拒绝 (无效)	5, 3, 4
4	33	234	5678	拒绝 (无效)	6, 3, 4
5	1234	234	4567	拒绝 (无效)	7, 3, 4
6	123	2B3	1234	拒绝 (无效)	2, 8, 4
7	123	013	1234	拒绝 (无效)	2, 9, 4
8	123	123	1234	拒绝 (无效)	2, 10, 4
9	123	23	1234	拒绝 (无效)	2, 11, 4
10	123	2345	1234	拒绝 (无效)	2, 12, 4
11	123	234	1B34	拒绝 (无效)	2, 3, 13
12	123	234	34	拒绝 (无效)	2, 3, 14
13	123	234	23345	拒绝 (无效)	2, 3, 15

思考与练习

假如某个系统的注册用户名要求由字母开头，后跟字母或数字的任意组合构成，有效字符数不超过6个

□ 有效等价类？

□ 无效等价类？



□ 测试用例？

输入数据	有效等价类	编号	无效等价类	编号
用户名	0<个全字母<=6	1	由数字开头构成的字符串	3
	0<字母开头+数字<=6	2	由字母开头构成的字符串，并含有特殊字符	4
			由字母开头构成的字符串且长度超过6的集合	5
			空字符串	6

例题：根据下面给出的规格说明，利用等价类划分的方法，给出足够的测试用例。“一个程序读入3个整数，把这3个数值看作一个三角形的3条边的长度值。这个程序要打印出信息，说明这个三角形是不等边的、是等腰的、还是等边的”。

分析：我们可以设三角形的3条边分别为A，B，C。如果它们能够构成三角形的3条边，必须满足：

$A > 0$ ， $B > 0$ ， $C > 0$ ，且 $A + B > C$ ， $B + C > A$ ， $A + C > B$ 。

如果是等腰的，还要判断 $A = B$ ，或 $B = C$ ，或 $A = C$ 。

如果是等边的，则需判断是否 $A = B$ ，且 $B = C$ ，且 $A = C$ 。

表 5-2 等价类表

输入条件	有效等价类	无效等价类
是否三角形的 3 条边	$(A > 0)$, (1)	$(A \leq 0)$, (7)
	$(B > 0)$, (2)	$(B \leq 0)$, (8)
	$(C > 0)$, (3)	$(C \leq 0)$, (9)
	$(A+B > C)$, (4)	$(A+B \leq C)$, (10)
	$(B+C > A)$, (5)	$(B+C \leq A)$, (11)
	$(A+C > B)$ (6)	$(A+C \leq B)$ (12)
是否等腰三角形	$(A=B)$, (13)	$(A \neq B)$ and $(B \neq C)$ and $(C \neq A)$, (16)
	$(B=C)$, (14)	
	$(C=A)$, (15)	
是否等边三角形	$(A=B)$ and $(B=C)$ and $(C=A)$, (17)	$(A \neq B)$, (18)
		$(B \neq C)$, (19)
		$(C \neq A)$, (20)

表 5-3 测试用例

topSage.co

序号	[A, B, C]	覆盖等价类	输出
1	[3, 4, 5]	(1), (2), (3), (4), (5), (6)	一般三角形
2	[0, 1, 2]	(7)	不能构成三角形
3	[1, 0, 2]	(8)	
4	[1, 2, 0]	(9)	
5	[1, 2, 3]	(10)	
6	[1, 3, 2]	(11)	
7	[3, 1, 2]	(12)	
8	[3, 3, 4]	(1), (2), (3), (4), (5), (6), (13)	等腰三角形
9	[3, 4, 4]	(1), (2), (3), (4), (5), (6), (14)	
10	[3, 4, 3]	(1), (2), (3), (4), (5), (6), (15)	
11	[3, 4, 5]	(1), (2), (3), (4), (5), (6), (16)	非等腰三角形
12	[3, 3, 3]	(1), (2), (3), (4), (5), (6), (17)	是等边三角形
13	[3, 4, 4]	(1), (2), (3), (4), (5), (6), (14), (18)	非等边三角形
14	[3, 4, 3]	(1), (2), (3), (4), (5), (6), (15), (19)	
15	[3, 3, 4]	(1), (2), (3), (4), (5), (6), (13), (20)	

- ❧ 测试同一个复杂程序的两个测试员，可能会制定出两组不同的等价区间。
- ❧ 在两数相加用例中，测试 $1+13$ 和 $1+999999999$ 似乎有点不同。这是一种直觉，一个是普通加法，而另一个似乎有些特殊，这个直觉是对的。程序对1和最大数值相加的处理和对两个小一些的数值相加的处理有所不同。后者必须处理溢出情况。因为软件操作可能不同，所以这两个用例属于不同的等价区间。（与编程经验有些关系）



2.2 边界值设计方法

边界值设计方法

- 程序的很多错误发生在输入或输出范围的边界上，因此针对各种边界情况设置测试用例，可以更有效地发现缺陷。
- 设计方法：
 - ❖ 确定边界情况（输入或输出等价类的边界）
 - ❖ 选取正好等于、刚刚大于或刚刚小于边界值作为测试数据

如何确定边界值?

项	边界值附近数据	测试用例的设计思路
字符	起始-1个字符/结束+1个字符	假设一个文本输入区域要求允许输入1到255个字符，输入1个和255个字符作为有效等价类；不输入字符（0个）和输入256个字符作为无效等价类。
数值范围	开始位-1/结束位+1	如数据输入域为1~999，其最小值为1，而最大值为999，则0、1000则刚好在边界值附近。从边界值方法来看，要测试4个数据：0、1、999、1000
空间	比零空间小一点/比满空间大一点	如测试数据的存储，使用比剩余磁盘空间大几K的文件作为测试的边界条件附近值。

如何确定边界值?

- ❖ 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。



- ❖ 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。



如何确定边界值？

- ❖ 如果软件规格说明给出的输入/输出域是有序集合，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。
- ❖ 如果程序中使用了内部数据结构，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。

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

一些特殊的边界值

- 隐藏已知文件类型的扩展名
- 用彩色显示加密或压缩的 NTFS 文件
- 在标题栏显示完整路径
- 在单独的进程中打开文件夹窗口
- 在登录时还原上一个文件夹窗口
- 在地址栏中显示完整路径
- 在文件夹提示中显示文件大小信息
- 在我的电脑上显示控制面板
- 在资源管理器文件夹列表中显示简单文件夹查看
- 自动搜索网络文件夹和打印机

- 隐藏已知文件类型的扩展名
- 用彩色显示加密或压缩的 NTFS 文件
- 在标题栏显示完整路径
- 在单独的进程中打开文件夹窗口
- 在登录时还原上一个文件夹窗口
- 在地址栏中显示完整路径
- 在文件夹提示中显示文件大小信息
- 在我的电脑上显示控制面板
- 在资源管理器文件夹列表中显示简单文件夹查看
- 自动搜索网络文件夹和打印机

全选、选9项、不选、选1

实例- web页面

8. Attendee Privileges

Select privileges that you want **attendees** to have when meeting begins:

Privileges:

- Save
- Print
- Annotate
- View participants list
- Control next or previous images
- View thumbnails
- Control applications, Web browser or desktop remotely
- View any document
- View any page
- Contact operator privately

Participant in private chat with:

- Host
- Presenter
- All attendees

Navigation: Cancel **7** Back Next **9**

Progress: **1** Required Information
2 Date & Time
3 Teleconference
4 Invite Attendees
5 Registration
6 Agenda & Welcome
8 **Attendee Privileges**
9 Review

START

测试用例:

正常的选择: 随机选择一些选项

边界值: 选择所有选项

边界值: 一项都不选

边界值: 只选一项

边界值分析法测试用例（续）

❖ **例1:** 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:

{ $\langle x_1 \text{nom}, x_2 \text{min} \rangle$, $\langle x_1 \text{nom}, x_2 \text{min}+ \rangle$,
 $\langle x_1 \text{nom}, x_2 \text{nom} \rangle$, $\langle x_1 \text{nom}, x_2 \text{max} \rangle$,
 $\langle x_1 \text{nom}, x_2 \text{max}- \rangle$, $\langle x_1 \text{min}, x_2 \text{nom} \rangle$,
 $\langle x_1 \text{min}+, x_2 \text{nom} \rangle$, $\langle x_1 \text{max}, x_2 \text{nom} \rangle$, $\langle x_1 \text{max}-$,
 $x_2 \text{nom} \rangle$ }

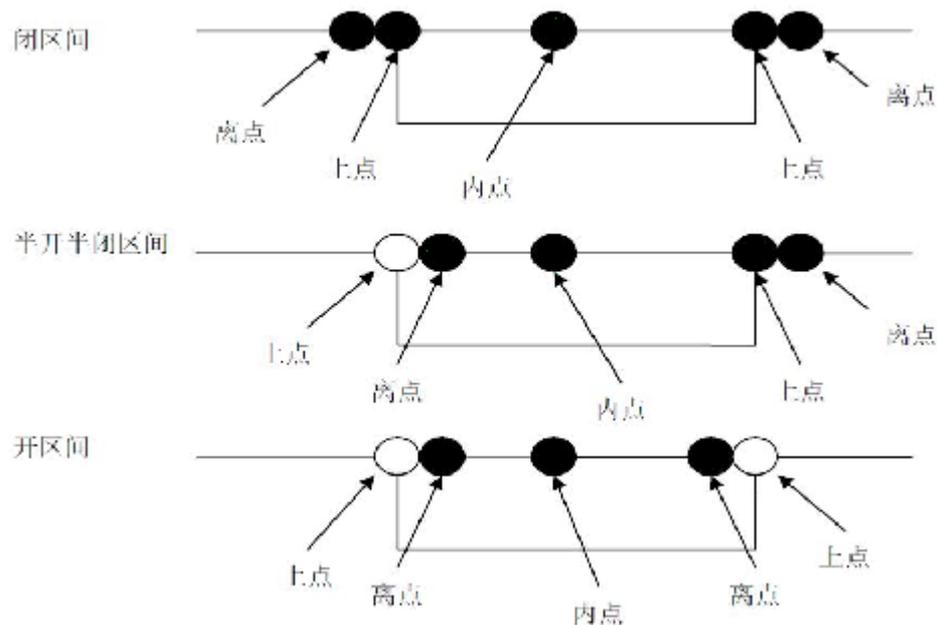
边界值分析法测试用例（续）

- ❖ **例2:** 有二元函数 $f(x,y)$ ，其中 $x \in [1,12]$ ， $y \in [1,31]$ 。采用边界值分析法设计的测试用例的数据是：
{ $\langle 6,1 \rangle$, $\langle 6,2 \rangle$, $\langle 6,31 \rangle$, $\langle 6,30 \rangle$, $\langle 6,15 \rangle$, $\langle 1,15 \rangle$, $\langle 2,15 \rangle$, $\langle 12,15 \rangle$, $\langle 11,15 \rangle$ }
- **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。

❖ 实训06： 第2题

边界值分析法测试用例（续）

- ❖ 了解一下边界点的定义，边界点分为上点、内点和离点。



上点，就是边界上的点，不管它是开区间还是闭区间，就是说，如果该点是封闭的，那上点就在域范围内，如果该点是开放的，那上点就在域范围外；

内点，就是在域范围内的任意一个点；

离点，就是离上点最近的一个点，如果边界是封闭的，那离点就是域范围外离上点最近的点，如果边界是开放的，那离点就是域范围内离上点最近的点。

边界值分析法测试用例（续）

闭区间 $[50, 100]$ 的上点为50和100，离点是49和101，在域范围内的都是内点；

半开半闭区间 $(50, 100]$ 的上点为50和100，离点是51和101，在域范围内的都是内点；

开区间 $(50, 100)$ 的上点为50和100，离点是51和99，在域范围内的都是内点；

等价类和边界值方法的结合

- ❖ 在进行等价类分析时，往往先要确定边界。如果不能确定边界，就很难定义等价类所在的区域。只有边界值确定下来，才能划分出有效等价类和无效等价类。边界确定清楚了，等价类就自然产生了。
- ❖ 边界值分析方法是等价类划分法的补充。在测试中，会将两者方法结合起来共同使用
- ❖ 如表6-3所示的例子
个人所得税处理程序：

个人月收入 - x	税率
$x < 0$	无效输入
$x \leq 1600$	0%
$1600 < x < 2100$	5%
$500 \leq x < 3600$	10%
$3600 \leq x < 6600$	15%
$6600 \leq x < 21600$	20%
$21600 \leq x < 41600$	25%
.....	
> 101600	45%

实训06第3题

NextDate函数包含三个变量month、day、year,函数的输出为输入日期的下一个日期,例如,输入为1989年5月16日,则函数的输出为1989年5月17日。要求输入变量month、day、year均为整数,设定变量year的取值范围为 $1980 \leq \text{year} \leq 2100$.请结合等价类方法和边界值分析方法来设计NextDate函数的测试用例.

❖ NextDate问题陈述

- 输入：三个整数 m,d,y ，代表月份、日期和年，
- 输出：输入日期后面的那个日期。 m,d,y 满足：
 - a1. $m \geq 1, m \leq 12$
 - a2. $d \geq 1, d \leq 31$
 - a3. $y \geq 1980, y \leq 2100$

1.3 测试用例的选择——边界值分析

假设:

➤ 输入变量之间相互独立。

测试用例: 使所有变量取正常值, 仅使一个变量取极值

测试用例: 不考虑函数性质和变量的语义含义

Test Case	Input Values		
	Month	Day	Year
ID-1	1	15	1992
ID-2	2	15	1992
ID-3	6	15	1992
ID-4	11	15	1992
ID-5	12	15	1992
ID-6	6	1	1992
ID-7	6	2	1992
ID-3	6	15	1992
ID-8	6	30	1992
ID-9	6	31	1992
ID-10	6	15	1980
ID-11	6	15	1981
ID-3	6	15	1992
ID-12	6	15	2099
ID-13	6	15	2100

分析

1、 year $1980 \leq \text{year} \leq 2100$

 平年? 闰年?

2、 month $1 \leq \text{month} \leq 12$

 大月? 小月? 2月? 12月?

3、 day $1 \leq \text{day} \leq 31$

$1 \leq \text{day} \leq 27?$ $=28?$ $=29?$ $=30?$ $=31?$

等价类列表

输入类型	有效等价类	无效等价类
year范围	1. $1980 \leq \text{year} \leq 2100$ 是闰年 2. $1980 \leq \text{year} \leq 2100$ 是平年	3. $\text{year} < 1980$ 4. $\text{year} > 2100$
month范围	5. month为4、6、9、11月 6. month为1、3、5、7、8、10月 7. month为2月 8. month为12月	9. $\text{month} < 1$ 10. $\text{month} > 12$
day范围	11. $1 \leq \text{day} \leq 27$ 12. day为28 13. day为29 14. day为30 15. day为31	16. $\text{day} < 1$ 17. $\text{day} > 31$ 18. day为29(平年2月) 19. day为30(闰年2月) 20. day为31 (小月)

测试用例

用例	Year	Month	Day	预期输出	覆盖等价类
1	2101	11	28	年份错误	4,5,12
2	1979	8	29	年份错误	3,6,13
3	2099	-1	31	月份错误	2,9,15
4	2008	13	30	月份错误	1,10,14
5	2009	2	-1	日期错误	2,7,16
6	2012	12	32	日期错误	1,8,17
7	2013	12	31	2014-01-01	1,8,15
8	2008	12	27	2008-12-28	1,8,11
9	2008	2	30	日期错误	1,7,19
10	1981	2	29	日期错误	2,8,18
11	1980	5	30	2008-05-31	1,6,14
12	2008	4	31	日期错误	1,5,20
13	2008	6	29	2008-06-30	1,5,13
14	2008	2	28	2008-02-29	1,7,12

案例2:

❖ 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:

投保额 × 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	

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

(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。

等价类表

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 60~99岁	3	小于1	12
			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 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