

# 测试用例设计方法（一）

等价类划分和边界值分析

讲师：路都行



# 内容

**1、等价类划分方法**

**2、边界值分析方法**



# chapter 1

## 等价类划分方法

# 什么是等价类划分

## ■ 概念

- 等价类划分就是把所有可能的输入数据,即程序的输入域划分成若干部分,从每一部分中选取少数有代表性的数据做为测试用例, 代表性数据等同于该类中的其他值
- 案例: 乌鸦喝水

## ■ 分类 **注意: 设计测试用例时, 要同时考虑有效等价类和无效等价类设计**

- **有效等价类**: 对于程序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**: 对于程序规格说明来说是不合理的、无意义的输入数据构成的集合
- **作用域**: 输入数据、输出数据

# 等价类划分的方法

- 按区间划分
- 按数值划分
- 按数值集合划分
- 按限制条件或规划划分
- 按处理方式划分
- **等价类划分重要的是：**
  - 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。

# 等价类划分的原则

## ■ 1. 输入条件的取值范围，可以划分出一个有效等价类和两个无效等价类

■ 例如：在程序的规格说明中，对输入条件有一句话：

“..... 输入数值范围可以从1到999 .....” ， 则：

➤ **有效等价类：** “ $1 \leq \text{输入数值} \leq 999$ ”

➤ **无效等价类：**

“输入数值  $< 1$ ” 或 “输入数值  $> 999$ ”



# 等价类划分的原则

- 2. 如果输入条件规定了输入值的集合，或者是规定了“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
  - 例如：对用户名规定为“以字母打头的字符串”，那么：
    - **有效等价类**：所有以字母打头的构成
    - **无效等价类**：不以字母打头的字符串

# 等价类划分的原则

- 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。
  - 例如：简历录入时有一栏填写是否党员，那么：
    - 一个有效等价类：{True, False}
    - 一个无效等价类：{非布尔量}



# 等价类划分的原则

- 4. 如果规定了输入数据的一组值(假设N个), 而且程序要对每个输入值分别进行处理。
  - 每个允许的输入值是一个有效等价类 (即N个有效的)
  - 这组值确立一个无效等价类, 它是所有不允许的输入值的集合。
- 例如: 在教师分房方案中规定对教授、副教授、讲师和助教分别计算分数, 做相应的处理。那么:
  - 4个有效等价类: 分别为教授、副教授、讲师和助教
  - 一个无效等价类: 所有不符合以上身份的人员输入值的集合

# 等价类划分的原则

## ■ 5. 如果规定了输入数据必须遵守的规则，则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)

■ 案例1：语言规定 “一个语句必须以分号 ‘;’ 结束”。这时可以确定：

- **一个有效等价类：** “以‘;’ 结束”
- **若干个无效等价类：** “以 ‘:’ 结束”、“以 ‘,’ 结束”、“以 ‘.’结束”、“以LF结束” 等

■ 案例2：以原则5分析原则1中的案例，可以得到：

- **有效等价类：** 输入数值
- **无效等价类：** 输入英文、特殊字符等非数值

# 等价类划分的原则

- 6. 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地划分为更小的等价类
  - 例如：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"。

# 等价类划分的原则

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

# 确立测试用例

- 在确立了等价类之后，建立等价类表，列出所有划分出的等价类
- 例如：

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

等价类表

# 确立测试用例

- 从已列出的等价类表中按以下原则选择测试用例
  - 为每个等价类规定一个唯一的编号
  - 设计一个新的测试用例，使其尽可能多地覆盖尚未覆盖的有效等价类。  
重复这一步，最后使得所有有效等价类均被测试用例所覆盖
  - 设计一个新的测试用例，使其只覆盖一个无效等价类。重复这一步使所有无效等价类均被覆盖

# 案例一

## ■ 新员工信息注册

- 员工姓名可以是 [a-z][A-Z]所组成的任意字符。
- 年龄的范围是1-150。
- 性别可以是以下两种之一： {Female, male}.

## 案例一-创建等价类表

输入条件	有效等价类	ID	无效等价类	ID
Name	[a-z][A-Z]	1	{1}	5
Age	1<age<150	2	{0}	6
			{151}	7
Gender	Female	3	abc	8
	male	4		



# 案例一-构建测试案例

- 为每一个等价规定一个唯一编号。
- 使用测试案例尽可能多的覆盖有效等价类。
- 使用单独的一个测试案例覆盖单独的一个无效等价类。
- 最后，直到所有的有效等价类和无效等价类均被覆盖。

## 案例一-构建测试用例

输入条件	有效等价类	ID	无效等价类	ID
Name	[a-z][A-Z]	1	{1}	5
Age	1<age<150	2	{0}	6
			{151}	7
Gender	Female	3	abc	8
	male	4		

Test Case	Data	CID
Case 1	Alex,25,Female	1,2,3
Case 2	Abc ,26,male	1,2,4
Case 3	!@#, 26,male	2, 4, 5
Case 4	Abc , 0, male	1\4\6
Case 5	Abc\151\male	1\4\7
Case 6	Abc\24\abc	1\2\8

## 案例二

- 有一个程序读入3个整数，把这3个整数作为一个三角形的3条边的长度值。这个程序要打印出信息，说明这3个整数是否能组成三角形，如果能组成三角形，说明这个三角形是不等边、等腰或等边三角形。

## 案例二-分析

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

**如果三条边满足条件 (1) ~ (4) , 则输出下列四种情况之一：**

- 如果不满足条件 (5) , 则程序输出为 “ 非三角形 ”
- 如果三条边相等即满足条件 (7) , 则程序输出为 “ 等边三角形 ”
- 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 ”
- 如果三条边都不相等, 则程序输出为 “ 不等边三角形 ”

## 案例二-分析

- 可以设定3条边分别为A,B,C。如果他们能构成三角形的3条边，必须满足：

$$0 < A, 0 < B, 0 < C,$$

$$\text{且 } A+B > C, B+C > A, A+C > B$$

如果是**等腰**，要判断 $A=B$ ,或 $B=C$ ,或 $A=C$

如果是**等边**，要判断 $A=B$ ,且 $B=C$ ,且 $A=C$

## 案例二-创建等价类表


等价类表示例

输入条件	有效等价类	无效等价类
是否是三角形的 3 条边	$0 < A$ (1) $0 < B$ (2) $0 < C$ (3) $A + B > C$ (4) $B + C > A$ (5) $A + C > B$ (6)	$A \leq 0$ (7) $B \leq 0$ (8) $C \leq 0$ (9) $A + B \leq C$ (10) $B + C \leq A$ (11) $A + C \leq B$ (12)
是否等腰三角形	$A = B$ (13) $C = B$ (14) $A = C$ (15)	$(A \neq B) \text{ AND } (B \neq C) \text{ AND } (C \neq A)$ (16)
是否等边三角形	$(A = B) \text{ AND } (B = C) \text{ AND } (C = A)$ (17)	$A \neq B$ (18) $A \neq C$ (19) $C \neq B$ (20)

## 案例二-构建测试用例

### 测试用例

序号	『A,B,C』	覆盖等价类	输出
1	『3, 4, 5』	1、2、3、4、5、6、16	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、20	等腰三角形
9	『3, 4, 4』	1、2、3、4、5、6、14、18	
10	『3, 4, 3』	1、2、3、4、5、6、15、19	
11	『3, 3, 3』	1、2、3、4、5、6、17	等边三角形



# chapter 2

## 边界值分析方法



# 边界值分析

- 从测试工作经验得知，大量的错误是发生在输入或输出范围的边界上，而不是在输入范围的内部
- 针对各种边界情况设计测试用例，可以查出更多的错误

比如，在做三角形计算时，要输入三角形的三个边长：A、B和C。我们应注意到这三个数值应当满足： $A > 0$ 、 $B > 0$ 、 $C > 0$ 、 $A + B > C$ 、 $A + C > B$ 、 $B + C > A$ ，才能构成三角形。但如果把六个不等式中的任何一个大于号“ $>$ ”错写成大于等于号“ $\geq$ ”，那就不能构成三角形。

**问题恰出现在容易被疏忽的边界附近**

# 边界的含义

- **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,稍高于其边界值及稍低于其边界值的一些特定情况**
- **边界值分析方法**
  - 选取正好等于, 刚刚大于, 或刚刚小于边界的值做为测试数据的方法
- **边界值分析是等价类划分方法的补充, 所有测试阶段都可使用**

# 边界值分析原则

- 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值，以及刚刚超越这个范围的边界值作为测试的输入数据。

例如：

输入值的范围是“1~9”，则可以选择“1”、“9”、“0.9”、“9.1”作为测试输入数据。

# 边界值分析原则

- 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。

例如：

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。

# 边界值分析原则

- 3) 根据规格说明的每个输出条件，使用原则 1)

例如：某程序的功能是计算折扣量，最低折扣是0元，最高折扣是1000元。则设计一些测试用例，使他们刚好产生0元和1000元的结果。

- 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，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) 分析规格说明，找出其他可能的边界条件。

# 案例一

```
Void CMyClass::Grow(int years)
{
    mAge += years;

    if(mAge<10)
        mPhase = "儿童";
    else if(mAge<20)
        mPhase = "少年";
    else if(mAge<45)
        mPhase = "青年";
    else if(mAge<60)
        mPhase = "中年";
    else
        mPhase = "老年";
}
```

## 案例一-分析

序号	表达式	略小	边界	略大
1	$X < 10$	9	10	11
2	$X < 20$	19	20	21
3	$X < 45$	44	45	46
4	$X < 60$	59	60	61