

测试用例设计

深 圳 市 门 道 信 息 咨 询 有 限 公 司

Shenzhen MT Information Consulting Co . , LTD

版权所有.侵权必究

目录

[1. 测试用例设计规范 4](#_Toc25595505)

[1.1 什么是测试用例 4](#_Toc25595506)

[1.2 测试用例包含的要素 4](#_Toc25595507)

[1.3 测试用例 － 参考 5](#_Toc25595508)

[1.4 测试用例误区 7](#_Toc25595509)

[2. 黑盒用例设计技术（重点） 7](#_Toc25595510)

[2.1 为什么要设计如此多方法 7](#_Toc25595513)

[2.2 等价类（重点） 7](#_Toc25595514)

[2.2.1 概念 7](#_Toc25595519)

[2.2.2 设计步骤 7](#_Toc25595520)

[2.2.3 示例 8](#_Toc25595521)

[2.2.4 练习 10](#_Toc25595522)

[2.3 边界值（重点） 10](#_Toc25595523)

[2.3.1 概念 10](#_Toc25595525)

[2.3.2 与等价划分的区别 10](#_Toc25595526)

[2.3.3 设计步骤 10](#_Toc25595527)

[2.3.4 示例： 10](#_Toc25595528)

[2.4 错误推测法（重点） 11](#_Toc25595529)

[2.4.1 概念 11](#_Toc25595531)

[2.4.2 示例 11](#_Toc25595532)

[2.4.3 练习 11](#_Toc25595533)

[2.5 场景法（重点） 11](#_Toc25595534)

[2.5.1 概念 11](#_Toc25595536)

[2.5.2 事件流 12](#_Toc25595537)

[2.5.3 设计步骤 12](#_Toc25595538)

[2.5.4 示例 12](#_Toc25595539)

[2.6 因果图（了解） 13](#_Toc25595540)

[2.6.1 概念 13](#_Toc25595542)

[2.6.2 案例 14](#_Toc25595543)

[2.7 判定表（了解） 15](#_Toc25595544)

[2.7.1 概念 15](#_Toc25595546)

[2.7.2 设计步骤 15](#_Toc25595547)

[2.7.3 示例 15](#_Toc25595548)

[2.8 正交试验（了解） 15](#_Toc25595549)

[2.9.1 概念 15](#_Toc25595552)

[2.9.2 设计步骤 16](#_Toc25595553)

[3. 设计方法综合策略 16](#_Toc25595554)

# 测试用例设计规范

## 什么是测试用例

* 测试用例是一份测试文档，其目的是确定系统的某个特性是否正常工作。
* 测试用例是软件测试团队的主要工作成果之一。
* 测试用例的质量与写该用例的测试人员的水平关系极大。
* 执行测试用例是将这些用例逐个在被测的软件上执行，并判断其结果是否和预期相符。

## 测试用例包含的要素

* 用例编号

用例编号：给用例定一个具体编号，用以唯一标识一条用例。

编号要唯一

技巧：

* 1. 项目名\_模块名\_编号
  2. 模块名\_编号
  3. 编号
* 模块

该用例所属模块名，一个模块下有一个或多个功能。某些大模块还分子模块，具体分法根据项目业务和测试用例的组织来确定，一般没有严格的规定。

* 功能

该用例所涉及的功能，每个功能下有一条或多条用例

* 用例标题

有的公司也叫测试目的

标题不能重复

标题是用例评审时的重要评审点，因此要不断练习，语言精炼

* 优先级（用例等级）

一般分为高、中、低

高：核心流程、冒烟用例

中：一般流程、异常流程

低：界面、兼容

注意：不同的公司会有不同的优先级标识，如：1、2、3

* 预置条件（前置条件）

一般不填写，除用例必须在特殊情况，特殊条件下才能执行时填写。

* 不需要填写：
  1. 需要登录后才能点击某个连接或进入某个界面
  2. 需要准备一个正确数据才能登录
  3. 需要添加数据才能执行查询
* 需要填写：
  1. 需要某种特定网络环境
  2. 需要有某些权限才能执行用例
  3. 需要在某个用例执行后才执行本用例
* 步骤

描述具体如何操作的过程

执行人会根据步骤执行，因此编写后一定要有可执行性，即执行人拿到后不会因为读不懂或看不明白而问用例设计者

一般包含:

* 1. 进入页面步骤，即路径
  2. 输入了哪些数据
  3. 执行了哪些操作
* 预期结果

按照测试步骤执行后，期望得到一个什么输出或者结果

有的公司一对一，有的公司多对一。如：一个步骤一个预期结果；多个步骤一个预期结果

* 编写者

编写该条用例的测试人员

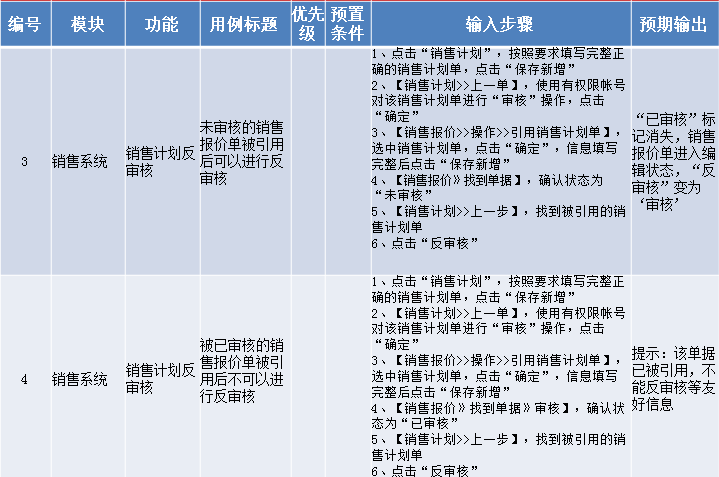
* 备注

用例需要做特殊说明时使用

一般不填写任何内容

## 测试用例 － 参考





练习：

* 登录成功
* 登录失败
* 购买商品流程
* 由于库存不足，购买商品失败流程

## 测试用例误区

1. 实际结果不属于测试用例的组成部分
2. 用例由于条件不足，数据不全，不具备测试等原因，在填写执行结果时除了通过和不通过，还有一个状态：未执行
3. 上述元素仅是用例公有部分，实际工作中各公司用例模板上会有差异，如：有的公司还有额外一些字段（环境、URL、开发者、参考资料等）

# 黑盒用例设计技术（重点）



## 为什么要设计如此多方法

当我们测试一个输入框，条件是：6-20位数字、字母、下划线时，如果把所有满足/不满足的情况全部测试一遍（穷举），那么测几年都不够，面对一个系统，时间就更不可取，因此我们需要一个切实可行的办法把穷举变得可测

## 等价类（重点）



### 概念

等价类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（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

### 设计步骤

1. 划分为有效和无效两个部分

有效：合理的、有意义的、接收后按照正常流程执行的数据所构成的集合；利用有效等价类可检验是否实现了需求所规定的功能

无效：不合理的、没有意义的、接收后按照非法流程执行或无法继续执行的数据所构成的集合；利用无效等价类可检验在异常状态下程序是否给出了友好提示或作出了相应判断

1. 列出所有的有效类和无效类

6大规则：

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
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,可确立一个有效等价类和一个无效等价类
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
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的情况下,可确立n个有效等价类和一个无效等价类
5. 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）
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的划分为更小的等价类
7. 设计足够少的测试用例，使其足够多的覆盖有效类
8. 设计足够多的测试用例，使其一一覆盖无效类

### 示例

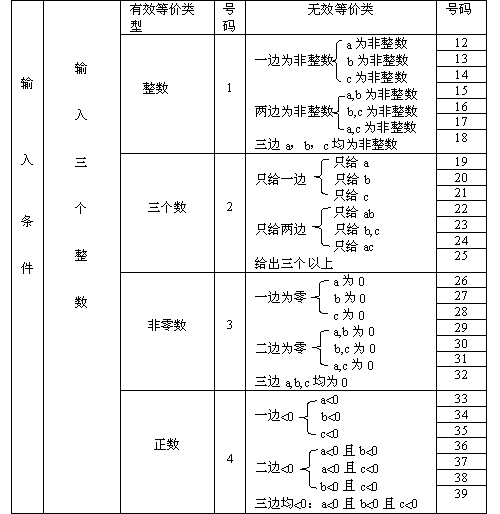
1、三角形问题（重点）

某程序规定：“输入三个整数 a 、 b 、 c 分别作为三边的边长构成三角形。通过程序判定所构成的三角形的类型，当此三角形为一般三角形、等腰三角形及等边三角形时，分别作计算 … ”。用等价类划分方法为该程序进行测试用例设计。

输入条件：

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

列出等价类



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）

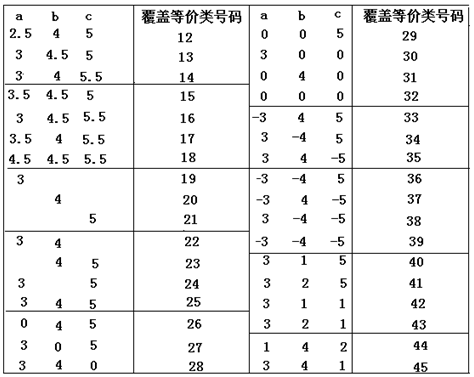
4 4 5 （1）--（7），（8）

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



### 练习

一个用户名框，需求为：6-20位数字、字母、下划线

## 边界值（重点）



### 概念

* 边界值分析法是对输入或输出的边界值进行测试的一种黑盒测试方法。
* **边界值分析方法是对等价类划分方法的补充，其测试用例来自等价类的边界。**
* **错误隐藏在角落里，问题聚集在边界上。**

长期的测试工作经验告诉我们，大量的错误是发生在输入或输 出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。

* **应当选取正好等于，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作为测试数据。**

### 与等价划分的区别

* + 1. 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。
    2. 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。

### 设计步骤

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
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
3. 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
4. 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

### 示例：



* 5M（正好等于）
* 4.9M（略小于最大值）
* 5.1M（刚刚大于）
* 3M（正常值）
* 0.1M（略高于最小值）
* 10M （远大于最大值）

## 错误推测法（重点）



### 概念

1. 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。
2. 错误推测方法的基本思想：列举出程序中所有可能有的错误和容易发生错误的特殊情况，根据他们选择测试用例。

### 示例

一个密码输入框

1. 空
2. 混合
3. 左右空格
4. 条数据完全一致（在不允许一致的情况下）
5. 极限

### 练习

一个用户名框，需求为：6-20位数字、字母、下划线（按照边界值设计）

## 场景法（重点）

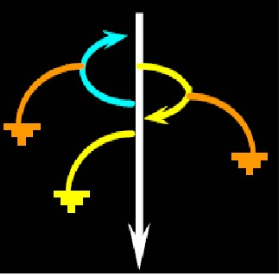
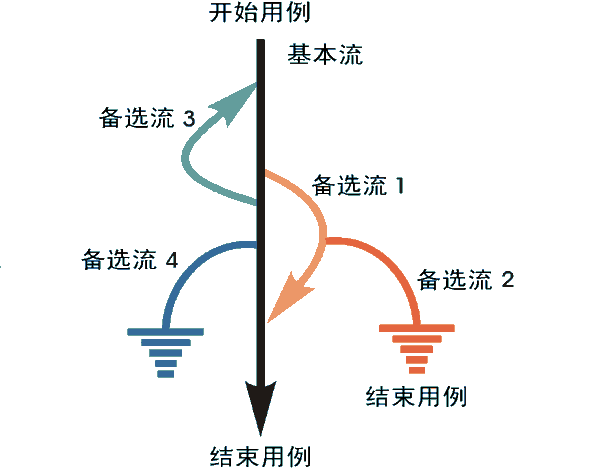


### 概念

也叫流程分析法，是将软件系统的某个流程看成路径，用路径分析的方法来设计测试用例。

根据流程的顺序依次进行组合，使得流程的各个分支都能走到

### 事件流

* 事件流组成
  + 基本流
    - 仅有一个基本流，白色箭头
    - 是经过用例的最简单的路径，指每个步骤都“正常”运作所行进的路线。
  + 备选流
    - 可以有多个
    - 可选或备选情况
    - 异常事件
* 基本流和备选流的区别

|  |  |  |
| --- | --- | --- |
| **对象** | **基本流** | **备选流** |
| 测试重要性 | 重要 | 次要 |
| 数目 | 1条 | 1条或多条 |
| 初始节点 | 初始状态 | 基本流或其他备选流 |
| 终止节点 | 终止状态 | 基本流或者其他终止状态 |
| 业务流程完整性 | 完整 | 不完整，仅为片段 |
| 能否构成场景 | 能 | 不能，需要基本流 |

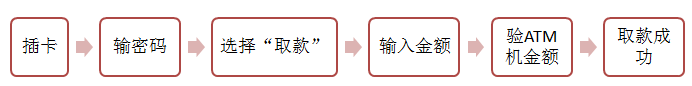
### 设计步骤

1. 分析被测业务，找出基本流及各项备选流。
2. 根据基本流和各项备选流生成不同的场景。
3. 对每一个场景生成相应的测试用例。

### 示例

ATM取款

* 基本流



* 备选流

1、插卡

* + 1. 卡插反了
    2. 不是银行卡
    3. 消磁卡
    4. 黑名单卡
    5. 已冻结卡
    6. 已注销卡

2、输密码

* + 1. 1次错误
    2. 2次错误
    3. 3次错误

3、选择“取款”

* + 1. 选择其他业务

4、输入金额

* + 1. 规则：单笔100倍，单笔最高5000，单日最多2W
    2. 根据这个规则，继续设计一堆用例

5、验ATM

* + 1. ATM钱不足
    2. ATM没钱



## 因果图（了解）



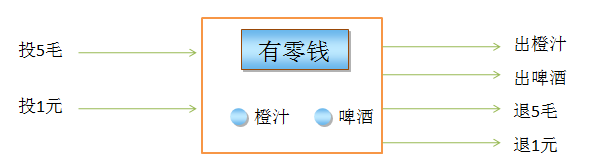
### 概念

1. 前面介绍的等价类划分方法和边界值分析方法都是着重考虑输入条件，但未考虑输入条件之间的联系, 相互组合等
2. 这就需要利用因果图方法将所有的组合进行过滤后设计用例
3. 实际工作中，因果图直接转判定表，这种方法实用性不大

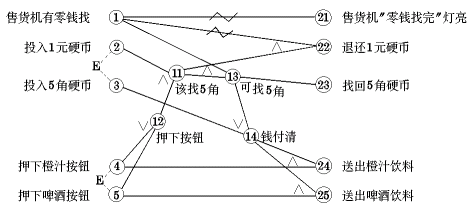
### 案例

有一个处理单 价为5角钱的饮料的自动售货机软件测试用例的设计。 其规格说明如下：若投钱或1元钱的硬币，押下〖橙汁〗或〖啤酒〗的按钮，则相应的入5角饮料就送出来。若售货机没有零钱找，则一个显示〖零钱找完〗的红灯亮，这时在投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则显示〖零钱找完〗的红灯灭，在送出饮料的同时退还5角硬币。

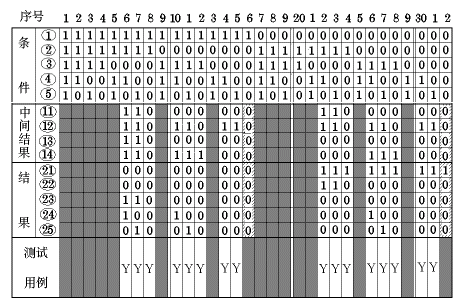
找出因果：



根据公式，确定组合



转化为判定表



## 判定表（了解）



### 概念

1. 前面因果图方法中已经用到了判定表。
2. 判定表（Decision Table）是分析和表达多逻辑条件下执行不同操作的方法。它可以把复杂的逻辑关系和多种条件组合的情况表达得既具体又明确。



### 设计步骤

1、找出所有的因和果。

2、确定因果存在的组合有多少种。假如有n个条件，每个条件有两个取值（0,1），故有2n种组合。

3、列出所有组合，得到初始判定表。

4、竖着执行，把初始判定表中不可能的情况全部去除，剩下的就是最终判定表。

### 示例

象棋走马

## 正交试验（了解）



### 概念

1. 利用因果图来设计测试用例时， 作为输入条件的原因与输出结果之间的因果关系，有时很难从软件需求规格说明中得到。
2. 往往因果关系非常庞大，以至于据此因果图而得到的测试用例数目多的惊人，给软件测试带来沉重的负担，为了有效地，合理地减少测试的工时与费用，可利用正交实验设计方法进行测试用例的设计。

### 设计步骤

1、提取功能说明，构造因子--状态表

把影响实验指标的条件称为因子，而影响实验因子的条件叫因子的状态。首先要根据被测试软件的规格说明书找出影响其功能实现的操作对象和外部因素，把他们当作因子，而把各个因子的取值当作状态。

2、加权筛选,生成因素分析表

对因子与状态的选择可按其重要程度分别加权，可根据各个因子及状态的作用大小，出现频率的大小以及测试的需要，确定权值的大小。

3、利用正交表构造测试数据集

正交表的推导依据Galois理论（这里省略，需要时可查数理统计方面的教材）。

# 设计方法综合策略

* 1. 设计方法如何选择

1. 当遇到需要输入数据时，首先进行等价类和边界值，这是减少工作量和提高测试效率最有效方法
2. 用错误推测法再追加一些测试用例
3. 对照程序逻辑，如果存在组合，用判定表设计用例
4. 对于业务流清晰的功能，利用场景法贯穿整个测试案例过程，再针对不同节点综合使用其他测试方法
5. 测试用例的设计方法不是单独存在的
6. 在实际测试中，往往是综合使用各种方法才能有效提高测试效率和测试覆盖度
   1. 测试用例的粒度
7. 粒度，指的是粗细程度。粒度大，就是说一个用例所涵盖的关注内容比较多，反之同理。
8. 用例的粒度大，则总的用例数就少，用例看起来也简洁
9. 用例的粒度小，则单条用例关注的测试点很集中，不容易遗漏，并且执行需要的时间比较好估计
10. 实际过程中是根据测试时间、项目阶段、主观/客观因素综合决定的
    1. 测试用例的更新
11. 测试用例并不可能一开始就写得很完美，可能也有写错的，可能也有遗漏的测试点。
12. 随着软件的版本不断更新，软件本身的需求和规格以及设计都可能在不断的变更。
13. 随着测试的不断开展，测试人员对产品的理解逐渐加深。