

# 测试理论第二章

## 目标

- 1.能够根据需求划分有效等价类和无效等价类
- 2.能够使用等价类方法设计（新浪邮箱登录案例）测试用例
- 3.能够说出等价类方法的适用场景
- 4.能够结合案例找出边界值的上点、离点和内点
- 5.能够使用边界值方法设计（新浪邮箱登录案例）测试用例
- 6.能够说出边界值方法的适用场景
- 7.能够使用判定表方法设计（文件修改案例）测试用例
- 8.能够说出判定表方法的使用场景
- 9.能够说出场景法的应用场景
- 10.能够说出错误推断法的应用场景

## 一 等价类划分法

### 1.等价类划分法的引入

案例：如何测试两个两位数整数之间的和（即-99到99之间数据求和）没有问题？

### 2.等价类划分法

- 等价类的概念：在所有测试数据中，具有某种共同特征的数据子集。
- 等价类划分为
  - 有效等价类：满足需求的数据子集
  - 无效等价类：不满足需求的数据子集

### 3.等价类划分法设计用例步骤

- 1.明确需求
- 2.确定有效和无效等价类
- 3.提取数据编写测试用例

案例1：验证QQ账号的合法性（要求：6~10位自然数）

案例2：验证某城市电话号码正确性

区号：非空或者是三位数字  
前缀码：非“0”且非“1”开头的三位数字  
后缀码：四位数字

案例3：新浪邮箱登录，要求输入（邮箱名）@sina.cn和（密码）

邮箱名为：4-16位字符，支持英文、数字、下划线（不能全是数字或者下划线）

密码：6-18位字符

## 4.适用场景

- 针对需要数据量大，有测试数据输入的地方
- 典型代表：页面级的输入框类测试

# 二 边界值分析法

## 1.边界值分析法的引入

案例：测试两个两位数整数之间的和（即-99到99之间数据求和）？

```
//判断输入的数据是否小于-99或者大于99，如果小于-99或大于99给出错误提示  
If Val(Text1.Text) >= 99 or Val(Text1.Text) <=- 99 Then  
    MsgBox ("输入的参数值必须大于-99同时小于99")  
Else if  
    Val(Text2.Text) >= 99 or Val(Text2.Text) <= -99 Then  
    MsgBox ("输入的参数值必须大于-99同时小于99") Then  
Else  
    Text3.Text = Val(Text1.Text) + (Text2.Text)  
End if  
End if
```

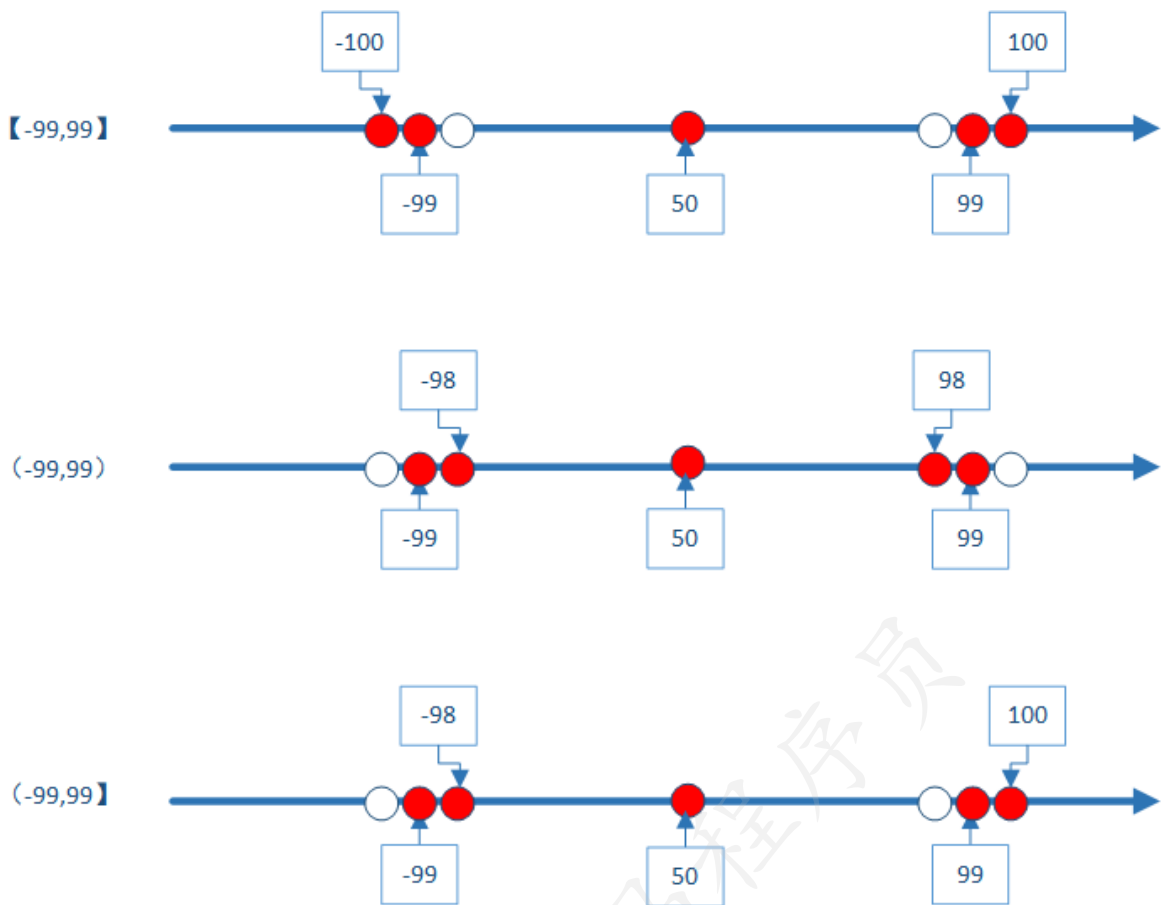
输入的数据包含99或者-99时，边界条件设置出错：代码中将">"写成了">="，将"<"写成了"<="

## 2.边界范围的确定

选取正好等于、刚好大于、搞好小于边界的值作为测试数据

- 上点：边界上的点（正好等于）
- 离点：距离上点最近的点（刚好大于、刚好下于）
- 内点：范围内的点（区间范围内的数据）

## 3.边界范围取值优化



结论：7个优化为5个点

上点：必选（不考虑区间开闭）

内点：必选（建议选择中间范围）

离点：开内闭外（考虑开区间，开区间选择内部离点，闭区间选择外部离点）

## 4.边界值法设计用例步骤

- 1.明确需求
- 2.确定有效和无效等价类
- 3.确定边界范围值
- 4.提取数据编写测试用例

案例1：通过边界值法验证标题长度的合法性（要求：标题长度大于0，小于等于30个字符）

案例2：通过边界值法验证QQ号码的合法性（要求：6~10位自然数）

案例3：新浪邮箱登录，要求输入（邮箱名）@sina.cn和（密码）

邮箱名为：4-16位字符，支持英文、数字、下划线（不能全是数字或者下划线）

密码：6-18位字符

## 5.适用场景

- 在等价类的基础上针对有边界范围的测试数据输入的地方
- 常见词语描述：大小、尺寸、重量、最大、最小、至多、至少等修饰词语
- 典型代表：有边界范围的输入框类测试

# 三 判定表法

## 1.判定表法的引入

案例：验证“若用户欠费或者关机，则不允许主被叫”功能的测试

等价类边界值分析法主要关注单个输入类条件的测试，并未考虑输入条件之间的各种组合、输入条件与输出结果之间有相互制约关系的测试。

## 2.判定表定义及组成部分

判定表：是一种以表格形式表达多条件逻辑判断的工具  
如上引例可以通过以下表格表示出：

条件	是否欠费	是	是	否	否
	是否关机	是	否	是	否
操作	是否允许主被叫	否	否	否	是

- 条件桩：列出问题中的所有条件。列出条件的次序无关紧要。
- 动作桩：列出问题中可能采取的操作。操作的排列顺序没有约束。
- 条件项：列出条件对应的取值。所有可能情况下的真假值。
- 动作项：列出条件项的各种取值情况下应该采取的动作结果。

规则：判定表中贯穿条件项和动作项的一列就是一条规则

假设有n个条件，每个条件的取值有两个（0,1），全组合有 $2^n$ 种规则

## 3.判定表法设计用例步骤

- 1.明确需求
- 2.画出判定表
  - 列出条件桩和动作桩
  - 填写条件项，对条件进行全组合
  - 根据条件项的组合确定动作项
  - 简化、合并相似规则（有相同的动作）
- 3.根据规则编写测试用例

案例1：订购单检查

订购单的检查：

如果金额大于500元，又未过期，则发出批准单和提货单；  
如果金额大于500元，但过期了，则不发批准单与提货单；  
如果金额小于等于500元，则不论是否过期都发出批准单和提货单；  
在过期的情况下不论金额大小还需要发出通知单。

案例2：文件修改规则

如想对文件进行修改，需要遵守以下规则：  
输入的第一列字符必须是A或B，第二列字符必须是一个数字，  
如果第一列字符不正确，则给出信息L；  
如果第二列字符不正确，则给出信息M。  
如果两列字符输入正确，则修改文件成功，  
如果两列字符都输入错误，则给出信息L、M。

## 4.适用场景

- 有多个输入条件，多个输出结果，输入条件之间有组合关系，输入条件和输出结果之间有依赖（制约）关系

# 四 场景法

## 1.场景法介绍

场景法，也可以叫流程图法，是用流程图描述用户的使用场景，然后通过覆盖流程路径来设计测试用例。

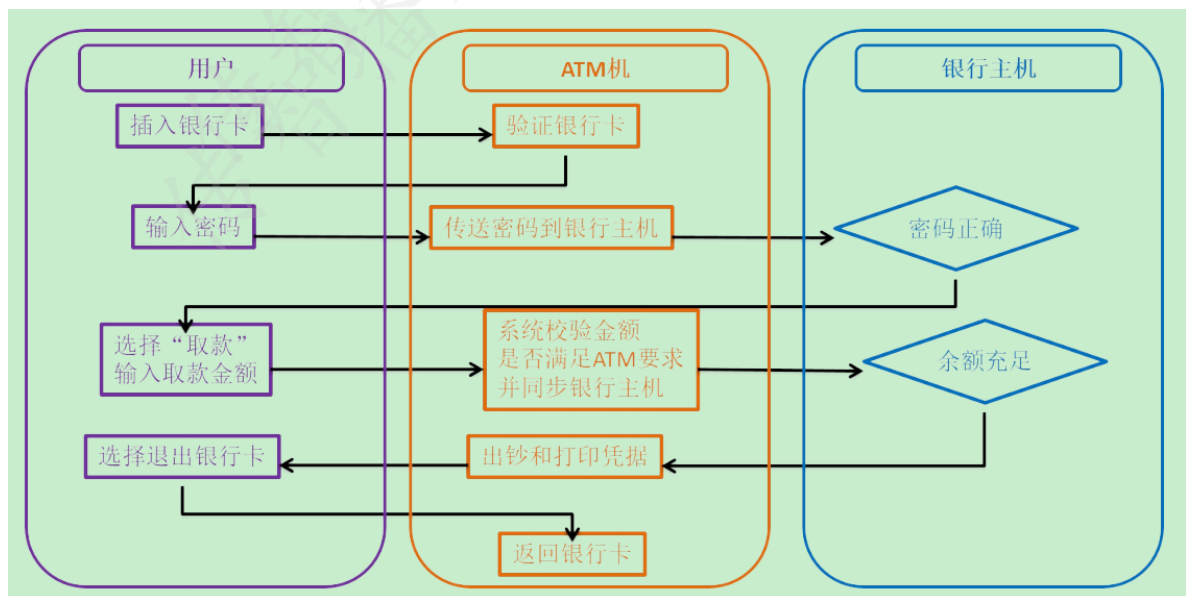
## 2.场景法的意义

- 用户使用角度：用户平时使用的不是单个功能，而是多个功能组合起来进行使用
- 测试人员角度：平时测试的都是单个功能点进行测试，容易忽略多个功能的组合测试

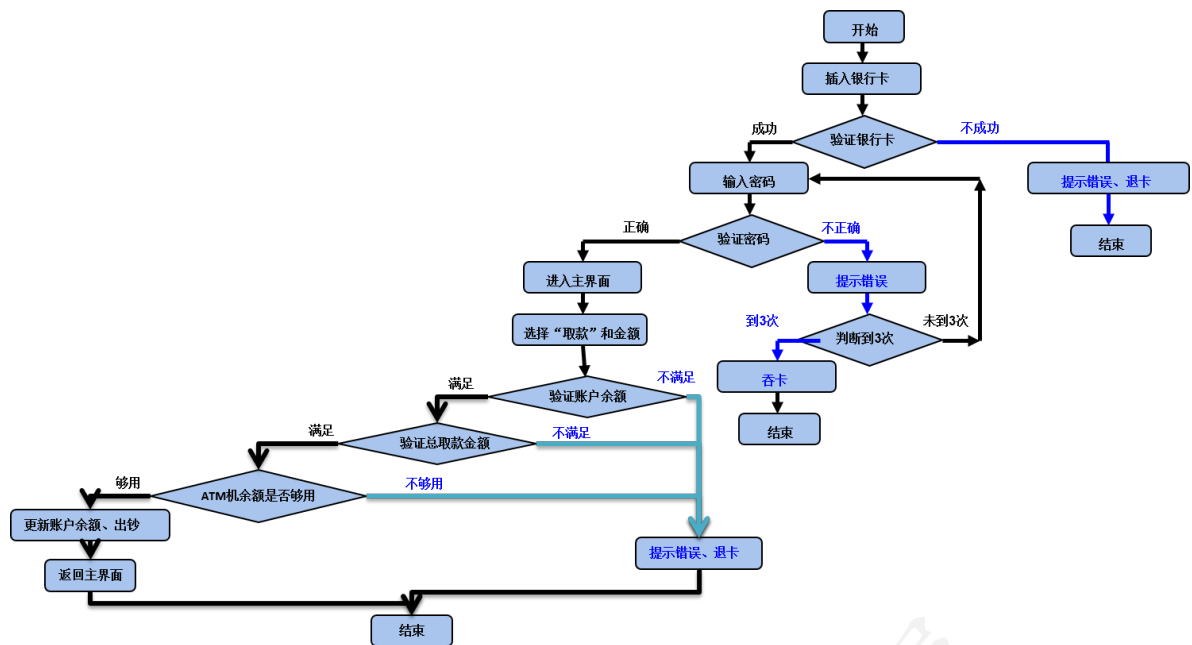
## 3.场景法的适用场景

对于多个功能之间的组合逻辑测试，可以使用场景法

案例：ATM机取款流程



ATM机的取款流程图



## 五 错误推测法

### 1.错误推测法的定义

通过经验和直觉推测系统可能出现问题的地方，一般由经验丰富的测试人员使用。

### 2.错误推测法的设计思想

- 凭人们对过去所作测试结果的分析，列举出可能出现问题的清单，根据清单测试来发现缺陷

### 3.错误推测法的适用场景

- 所有正常测试结束后，通过错误推断法再测试之前问题较多的模块
- 时间紧，任务量大，根据之前项目类似经验找出易出错的模块重点测试