软件测试核心是编写测试用例

常用编写软件用例方法：

**1、等价类划分法**：

等价类划分法是一种典型的，重要的黑盒测试方法，是指某个输入域的子集合，在该子集合中，所有的输入数据对于揭需软件中的错误都是等效的

等价类划分：

**有效等价类**：满足用户需求的，是合理的，符合规则的等价类

**无效等价类**：不满足用户需求的，是不合理的，不符合规则的等价类

如何对单个输入项提取测试点？

保证正常功能后再考虑异常情况（输入长度，输入数据类型，是否为必填项等方面）

等价类划分法用例设计原则：

1）划分有效等价类和无效等价类，为每个等价类设置唯一的编号

2）设计测试用例尽可能多的覆盖尚未被覆盖的有效等价类，直到所有有效等价 类都被覆盖

3）针对每个尚未被覆盖的无效等价类设计测试用例，直到所有无效等价类被覆 盖

**例一：微信红包金额：**

1：划分有效等价类和无效等价类

|  |  |  |  |
| --- | --- | --- | --- |
| 有效等价类 | 编号 | 无效等价类 | 编号 |
| 长度范围 |  | | |
| 0.01~200 | 1 | 小于0..01 | 3 |
| 大于200 | 4 |
| 小数点后大于两位 | 5 |
| 数据类型 |  | | |
| 纯数字 | 2 | 包含非数字（中文，英文，符号） | 6 |

**2、边界值分析法：**

边界值划分法是对等价类划分法的补充，边界值通常从等价类的边缘值去找。边界值分析的基本思想：刚刚等于，刚刚大于，刚刚小于边界的值作为测试数据

**\*\*注意**：0是一个特殊值，在考虑边界值时需要考虑0以及负数.在考虑负数时，避免负数绝对值超出等价类范围否则无法判别因为数字原因还是符号原因对系统产生的影响

边界值作用：

从过往经验得知，大量错误发生在输入或输出范围的边界上而不是在输入范围内部。因此针对各种边界情况设计测试用例，方便测试人员发现更多错误

2：基本用例设计

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例设计 | 测试输入 | 测试覆盖 | 特殊边界值 |
| 输入正确红包金额0.01~200 | 0.01 | 1，2 | 0.01,0.02.199.99,200 |
| 0.02 |
| 199.99 |
| 200 |
| 金额输入小于0.01 | 0 | 3 | 0 |
| -100 |
| 输入正确红包金额大于200 | 200.01 | 4 | 200.01 |
| 小数点后三位 | 0.001 | 5 | 0.001 |
| 包含非数字类型 | 10a纸鸢！ | 6 |  |

**例二：针对密码8~16位，数字、字母、符号至少包含两种**

1.等价类划分：

|  |  |  |  |
| --- | --- | --- | --- |
| 有效等价类 | 编号 | 无效等价类 | 编号 |
| 长度（范围） |  | | |
| 8-16位 | 1 | 小于8位 | 7 |
| 大于16位 | 8 |
| 类型 |  | | |
| 数字加字母 | 2 | 纯数字 | 9 |
| 数字加符号 | 3 | 纯字符 | 10 |
| 字母加符号 | 4 | 纯符号 | 11 |
| 数字加字母加符号 | 5 | 非字母，数字，符号 | 12 |
| 空值 | 13 |

2：设计基本用例

边界值7,8,9,15,16,17

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例设计 | 测试输入 | 测试覆盖 | 特殊边界值 |
| 输入正确密码 |  | | |
| 输入8位数字加字母密码 | 1a2d3c4f | 1,2 | 8 |
| 输入9位数字加符号 | 12345678？ | 1,3 | 9 |
| 输入15位字母加符号 | aaaaaaaaaaaaaa？ | 1,4 | 15 |
| 输入16位数字加字母加符号 | 12346123461a？ | 1,5 | 16 |
| 输入错误密码 |  | | |
| 7位数字加字母 | 123456？ | 7 | 7 |
| 17位数字加符号 | 12345612345612345？ | 8 | 17 |
| 10位纯数字 | 0123456789 | 9 |  |
| 10位纯字符 | abcdeabcde | 10 |
| 10位纯符号 | ？？？？？？？？？？ | 11 |
| 10位包含中文 | zhiyuan纸233 | 12 |
| 空值 |  | 13 |

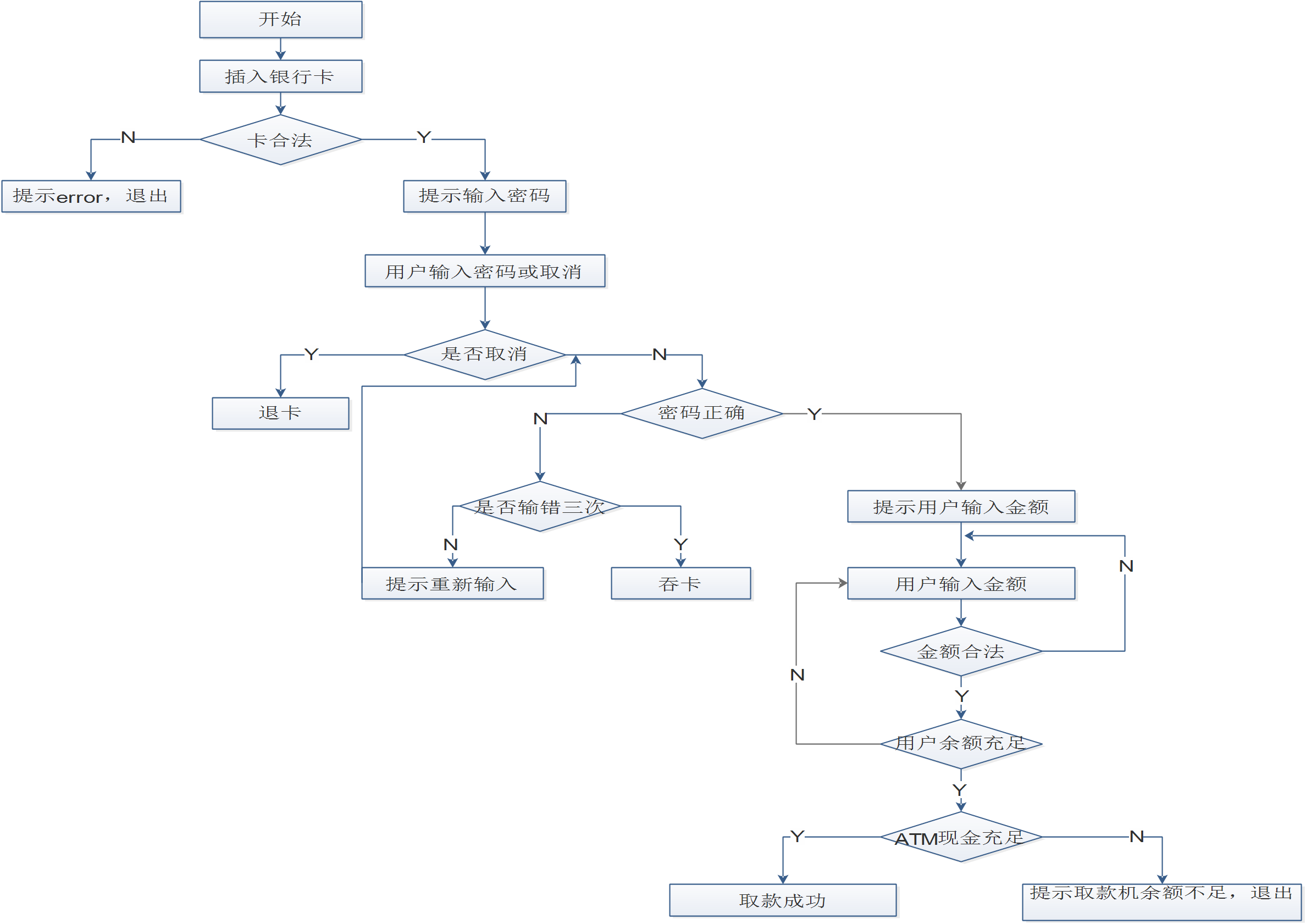
**3、场景法**

**1、什么是场景法：**

通过场景描述的业务流程（业务逻辑），也包括代码实现逻辑，设计用例来遍历场景，验证软件系统功能性测试

**\*\*：**场景法测试重点是测试流程，因为每个流程用一个用例验证即可。流程测试没问题不代表系统功能没问题，还需要针对单步的功能进行测试（等价类划分和边界值法）。经历过流程测试和单个功能点测试才算是充分的测试

例如针对ATM取款流程可以如下设计用例：



|  |  |
| --- | --- |
| 用例标题 | 覆盖路径 |
| 正常取款流程 | YNYYY |
| 卡不合法 | N |
| 用户取消取款 | YY |
| 密码输出三次 | YNNN |
| 密码输错超过三次吞卡 | YNNY |
| 用户输入非法金额 | YNYN |
| 用户余额不足 | YNYYN |
| ATM现金不足 | YNYYN |

**4、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，从而针对性的设计测试用例方 法

要素有三点：经验、知识、直觉

案例：某平台登录页面

根据错误推测法，先列举出可能出现结果错误情况：

1、用户名或密码对应关系验证

2、账号、密码为空

3、用户名、密码太长太短是否有提示

4、用户名、密码是否有对空格进行处理

5、密码错误限制次数

6、提交登录时，网络异常

7、多次点击提交操作但只提交一次

````````