# 软件测试核心概念

## 单选

1、开发人员接收到一个指派给自己的Bug后，认为自己的实现是符合需求的，此时该开发人员应该：（ ）  
A．找该bug的测试人员麻烦  
B．直接将bug改关闭  
C．跟提交该bug的人进行沟通，如果需求理解不能打成一致，找项目经理/需求管理者确定需求   
D．置之不理

2、在软件生命周期哪个阶段，软件缺陷修复费用最低（ ）  
A．产品发布  
B．需求分析（编制产品说明书）   
C．编码  
D．设计

3、单元测试中用来模拟被测模块调用者的模块是（ ）  
A．子模块  
B．父模块  
C．驱动模块  
D．桩模块

4、侧重于观察资源耗尽情况下的软件表现的系统测试被称为 （ ）   
A．容量测试  
B．压力测试  
C．强度测试  
D．性能测试

5、必须要用户参与的测试阶段是 （ ）   
A．验收测试  
B．确认测试  
C．单元测试  
D．集成测试

6、导致软件缺陷的最大原因是（ ）   
A．编码  
B．测试  
C．设计  
D．需求说明书

7、不属于集成测试步骤的是 （ ）   
A．记录集成测试结果  
B．指定集成计划  
C．回归测试  
D．执行集成测试

8、软件测试的目的（ ）  
A．表明软件的正确性  
B．评价软件质量   
C．尽可能发现软件中的错误   
D．判定软件是否合格

9、软件调试的目的是（ ）  
A、发现软件中隐藏的错误  
B、解决测试中发现的错误  
C、尽量不发现错误以便早日提交软件  
D、证明软件的正确性  
答案：B

10、软件测试的目的是（ ）  
A、发现软件中隐藏的错误  
B、解决软件中隐藏的错误  
C、尽量不发现错误以便早日提交软件  
D、证明软件的正确性  
答案：A

11、下列关于测试和调试的说法中正确的是（ ）   
A、测试和调试没有本质区别。目的都是为了发现软件系统中的错误。  
B、测试只是测试人员的职责，在整个测试活动中不需要开发人员的参与。？  
C、调试主要在软件的开发阶段进行。  
D、调试一般不能确定程序中潜在错误发生的原因  
答案：B

17、以下关于软件测试原则的说法中，错误的是（ ）。

A．在设计测试用例时，不但要包括合理的输入条件，还要包括不合理的输入条件  
B．测试过程中某模块中查出的错误越多，该模块残留的错误就越少  
C．坚持在软件开发各个阶段进行技术评审，才能在开发过程中尽早发现和预防错误  
D．在测试过程中要严格按照测试计划执行，以避免发生疏漏或重复无效的工作

答案：B

18、以下关于软件测试原则的说法中，错误的是（ ）。

A．应当把“尽早地和不断地进行软件测试”作为软件开发者的座右铭  
B．测试用例应该包括测试输入和与之对应的预期输出结果两部分  
C．程序员应避免检查自己的程序。如果由别人来进行测试，可能会更客观、更有效、更容易取得成功  
D．设计测试用例时，输入条件应当是合理的

答案：D

12、下面关于软件测试的说法，（ ）是错误的  
A．软件测试是程序测试   
B．软件测试贯穿于软件定义和开发的整个期间  
C．需求规格说明书、设计规格说明书都是软件测试的对象  
D．程序属于软件测试的对象

答案解析：A 软件包含程序、数据库、文档、服务，所以软件测试应该包含程序测试、数据库测试、文档测试、服务测试

13、某软件公司在招聘软件评测师时，应聘者甲向公司做如下保证：  
①经过自己测试的软件今后不会再出现问题；  
②在工作中对所有程序员一视同仁，不会因为在某个程序员编写的程序中发现的问 题多，就重点审查该程序，以免不利于团结；  
③承诺不需要其他人员，自己就可以独立进行测试工作；  
④发扬咬定青山不放松的精神，不把所有问题都找出来，决不罢休；  
你认为应聘者甲的保证（）。

A．①、④是正确的 B．②足正确的  
C．都是正确的 D．都不正确

答案：D

14、严重的软件缺陷的产生主要源自（ ）。  
A．需求  
B．设计 C．编码  
D．测试  
参考答案：A

15、软件测试的对象包括（ ）。  
A．程序和文档  
B．源程序、目标程序、数据库和相关文档   
C．程序、数据库、文档和服务  
D．目标程序、操作系统和平台软件

16、导致软件缺陷的原因有很多，①－④是可能的原因，其中最主要的原因包括（ ）。

①软件需求说明书编写的不全面，不完整，不准确，而且经常更改   
②软件设计说明书  
③软件操作人员的水平  
④开发人员不能很好的理解需求说明书和沟通不足  
A．①、②、③  
B．①、③ 　  
C．②、③   
D．①、④

答案：D

19、软件测试类型按照开发阶段划分是（ ）。  
A．需求测试、单元测试、集成测试、验证测试  
B．单元测试、集成测试、确认测试、系统测试、验收测试  
C．单元测试、集成测试、验证测试、确认测试、验收测试  
D．调试、单元测试、集成测试、用户测试

20、（ ）可以作为软件测试结束的标志。  
A．使用了特定的测试用例  
B．错误强度曲线下降到预定的水平  
C．查出了预定数目的错误  
D．按照测试计划中所规定的时间进行了测试

答案：B

21、以下不属于应用系统中的缺陷类型的是：（ ）

A．不恰当的需求解释 B．用户指定的错误需求

C．设计人员的习惯不好 D．不正确的程序规格说明

答案：B

22、降低缺陷费用最有效的方法是（ ）。

A．测试尽可能全面 B．尽可能早的开始测试

C．测试尽可能深入 D．让用户进行测试

答案：B

23、测试环境中不包括的内容是（ ）

A．测试所需人员 B．测试所需硬件

C．测试所需软件 D．测试所需场地

答案：A

25、用不同的方法可将软件测试分为白盒法和黑盒法，或者（ ）和静态测试。

A．白盒法 B．黑盒法

C．动态测试 D．灰盒法

答案：C

26、按照测试策略和过程，测试可以分为：（ ）

A．单元、白盒、确认、系统、验收

B．单元、集成、确认、系统、验收

C．白盒、黑盒、确认、系统、验收

D．白盒、集成、确认、系统、验收

27、软件产品与物质产品有很大的区别，软件产品是一种（）产品。

A.有形

B.消耗

C.逻辑3

D.文档

28.下列关于软件的说法正确的是（）

A.软件是通过定制进而生产制造出来的

B.软件没有磨损老化问题。

C.软件开发的成本很高

D.软件开发和运行必须依赖计算机环境

29.下列软件属于系统软件的是（）

A.操作系统

B.编译器

C.中间件

D.浏览器

30.下述软件属于支撑软件的是（）

A.财务管理软件

B.编译器

C.中间件

D.数据库

31.下述软件属于应用软件的是（）

A.财务管理软件

B.数据库

C.编译器

D.浏览器

32．软件测试的目的是（）。

A.试验性运行软件

B.发现软件错误

C.证明软件正确

D.修改软件中出现的错误

33. 被测试程序不在机器上运行，而是采用人工检测和计算机辅助静态分析的手段对程序进行检测，这种测试称为（）。

A.白盒测试

B.黑盒测试

C.静态测试

D.动态测试

34.软件调试的目的是（）。

A.发现错误

B.改正错误

C.改善软件的性能

D.挖掘软件的潜能

35. 调试应该由（）完成。

A.与源程序无关的程序员

B.编制该源程序的程序员

C.不了解软件设计的机构

D.设计该软件的机构

36.在下列描述中，关于测试与调试的说法错误的是（）。

A.测试是显示错误的行为，而调试是推理的过程

B.测试显示开发人员的错误，调试是开发人员为自己辩护

C.测试能预期和可控，调试需要想象、经验和思考

D.测试需要在了解软件设计的条件下完成，调试则需要完整的程序代码

37. 下面关于软件测试的原则说法正确的是（）

A.测试应该从大规模开始，逐渐转向小规模

B.测试计划应该在编码结束后开始制定

C.软件最好由编写该程序的程序员自己来测试

D.穷举测试是不可能的

38.判断题：测试只能保证尽可能多地发现错误，无法保证能够发现所有的错误。（对）

## Now

## 、多选

1、软件测试目的是什么？（ ）   
A．发现当前开发工作中所采用的软件过程的缺陷   
B．修正软件错误和缺陷提高软件质量  
C．为了证明软件没有错误   
D．对软件质量进行评度量和评估

2、软件测试是系统开发不可少的一部分，具有以下那些特征？（ ）   
A．可以是需求，而不仅仅是代码  
B．有助于在软件生命周期中尽早发现问题，以降低修复软件缺陷所需的成本  
C．既是静态活动也是动态活动   
D．用来预防软件失效

3、单元测试通过的标准是什么？（ ）   
A．分支覆盖率达到85%  
B．语句覆盖流程达到100%  
C．需求规格说明书中的需求必须全部实现并测试通过  
D．程序通过所有的单元测试用例

4、按照阶段划分，软件测试分为哪几类？  
A．验收测试   
B．集成测试  
C．单元测试  
D．系统测试

5、软件缺陷的常用状态有以下几种情况？  
A．已拒绝  
B．打开  
C．已解决  
D．提交

6、下列 （ ） 属于软件测试的原则  
A．测试无法显示潜伏的软件缺陷   
B．找到的缺陷越多，软件测缺陷就越少  
C．软件测试是有风险的行为  
D．完全的测试是不可能的

7、 验收测试合格通过的准则是（ ）   
A．立项审批表、需求分析文档、设计文档和编码实现一致。  
B．软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求  
C．所有测试项没有残余一级、二级和三级错误  
D．验收测试工件齐全

## 判断

1、软件测试在实际开发过程中可以做到穷尽测试。  
2、软件的质量特性有静态质量特性和动态质量特性。  
3、静态质量特性包括结构化的、可维护的、可测试的代码以及正确而又完整的文档。

## 简答题

1、简述软件测试工程师应具备的素质    
参考答案：    
1）软件测试人员具备的素质  
2）对软件测试工作有正确的认识  
3)具有很强的沟通能力、外交能力和学习能力  
4)掌握比较全面的技术  
5)测试中要做到“五心”（专心、细心、耐心、责任心和自信心）  
6）要有很强的记忆力，怀疑精神和洞察力  
7）具有探索、创新和挑战精神，努力追求完美  
2、简述软件缺陷的定义的五条规则。

## 填空

1. 软件测试的流程分为\_\_\_\_\_\_\_\_\_\_\_\_，设计测试，搭建测试环境，\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_五个主要的步骤。
2. 软件测试是通过\_\_\_\_\_\_\_\_\_\_\_ 和 \_\_\_\_\_\_\_\_\_\_\_\_ 的手段来运行或测试某个系统的过程，其目的在于检验被测软件系统是否满足规定的需求，或是弄清楚被测系统的\_\_\_\_\_\_\_\_\_\_\_\_与\_\_\_\_\_\_\_\_\_\_\_之间的差距。

# 白盒测试

## 单选

1、在下面所列举的逻辑测试覆盖中，测试覆盖最强的是（ ）

A．条件覆盖 B．条件组合覆盖

C．语句覆盖 D．判定条件覆盖

2、软件测试中白盒法是通过分析程序的（ ）来设计测试用例的。

A．应用范围 B．内部逻辑

C．功能 D．输入数据

答案：B

3、下列逻辑覆盖测试方法中，覆盖能力最弱的是（ ）

A．语句覆盖 B．判定覆盖

C．条件覆盖 D．条件组合覆盖

答案：A

4、选出属于白盒测试方法（ ）

A．测试用例覆盖 B．输入覆盖

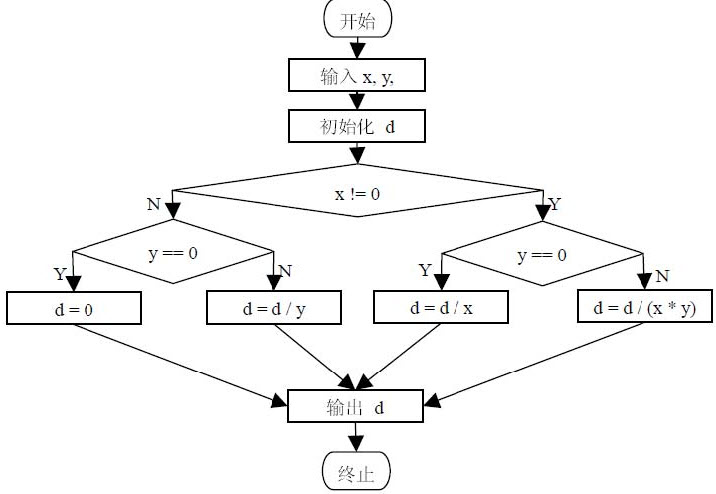
C．输出覆盖 D．条件覆盖

答案：D

5、对下面的个人所得税程序中满足语句覆盖测试用例的是（1），满足判定覆盖测试的用例是（2）。  
if (income<800) tarrate=0;  
else if (income<=1500) taxrate=0.05;  
else if (income<2000) taxrate=0.08;  
else taxrate=0.1;  
A．income=(800，1500，2000，2001) B．income=(800，801，1999，2000)  
C．income=(799，1499，2000，2001) D．income=(799，1500，1999，2000)

6、对下面的个人所得税程序中满足语句覆盖测试用例的是（1），满足判定覆盖测试的用例是（2）。  
if (income<800) tarrate=0;  
else if (income<=1500) taxrate=0.05;  
else if (income<2000) taxrate=0.08;  
else taxrate=0.1;  
A．income=(799，1500，1999，2001) B．income=(799，1501，2000，2001)  
C．income=(800，1500，2000，2001) D．income=(800，1499，2000，2001)

7、阅读下列流程图，当用判定覆盖法进行测试时，至少需要设计（）个测试用例。



A．2 B．4  
C．6 D．8

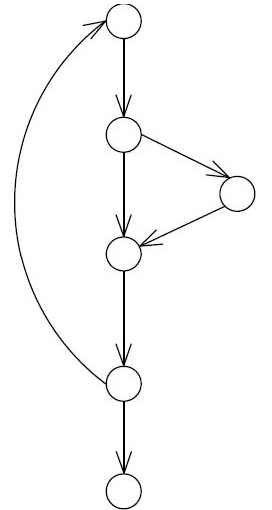
答案：B

8、下列控制流图的环路复杂度V（G）等于（ ）

A．4 B．5  
C．6 D．1

9、下列控制流图中有（ ）条线性无关（相互独立）的独立路径

A．4 B．5  
C．6 D．1



1. 如下图所示的控制流图的环路复杂度V（G）为（ ）。

A 4 B 5 C 6 D 7

11、条件组合覆盖是一种逻辑覆盖，它的含义是设计足够的测试用例，使得每个判定中条件的各种可能组合都至少出现一次，满足多条件覆盖级别的测试用例也是满足\_\_\_\_\_\_级别的。  
  
A．语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖  
B．判定覆盖、条件覆盖、判定/条件覆盖  
C．语句覆盖、判定覆盖、判定/条件覆盖  
D．路径覆盖、判定覆盖、条件覆盖、判定/条件覆盖

12、逻辑路径覆盖法是白盒测试用例的重要设计方法，其中语旬覆盖法是较为常用的方法，针对下面的语句段，采用语句覆盖法完成测试用例设计，测试用例见下表，对表中的空缺项(True或者False)，正确的选择是( )。

语句段：

if(A&&(B||C)) x=1；

else x=0；

用例表：



A. ①TRUE ②FALSE ③TRUE B. ①TRUE ②FALSE ③FALSE

C. ①FALSE ②FALSE ③TRUE D. ①TRUE ②TRUE ③FALSE

13、白盒测试是结构测试，被测对象基本上是源程序，以程序的（）为基础设计测试用例。

A.应用范围

B.功能

C.内部逻辑

D.输入数据

14、白盒测试的被测对象模型是（）。

A.程序流程图

B.控制流程图

C.事务流程图

D.模块结构图

15、下列几种逻辑覆盖测试按查错能力从弱到强排序正确的是（A）。

A.语句覆盖，分支覆盖，条件组合覆盖，路径覆盖

B.分支覆盖，条件组合覆盖，语句覆盖，路径覆盖

C.路径覆盖，条件组合覆盖，分支覆盖，语句覆盖

D.语句覆盖，路径覆盖，分支覆盖，条件组合覆盖

16、如果一个判定中的复合条件表达式为（A > 1）or（B <= 3），则为了达到100%的条件组合覆盖率，至少需要设计多少个测试用例（）。

A.1

B.2

C.3

D.4

## 简答题

1. 阐述白盒测试的各种方法。

参考答案：

白盒测试方法可以分为两大类：静态测试方法和动态测试方法，动态测试方法是设计一系列的测试用例，通过输入预先设定好的数据来动态地运行程序，从而达到发现程序错误的目的。静态测试方法则不在计算机上实际执行程序，而是以一些人工的模拟技术或使用测试软件对软件进行分析和测试。动态测试方法主要有逻辑覆盖、独立路径测试等。静态测试方法主要有静态结构分析、静态质量度量、代码检查方法等。

1. 简述独立路径测试的基本步骤。
2. 简述对单个循环结构和多个循环结构的测试方法
3. 简述白盒对判定的测试的6种覆盖指标及其各自的特点。

参考答案：

根据不同的测试要求，覆盖测试可以分为语句覆盖、判定覆盖、条件覆盖、判定／条件覆盖、条件组合覆盖和路径覆盖。

语句覆盖就是设计足够多的测试用例，使得被测试程序中的每条可执行语句至少被执行一次。

判定覆盖又称分支覆盖，就是设计足够多的测试用例，使得被测试程序中的每个判断的“真”、“假”分支至少被执行一次。

条件覆盖是指设计足够多的测试用例，使得被测试程序中的每个判断语句中的每个逻辑条件的可能值至少被满足一次。或者说设计足够多的测试用例，使得被测试程序中的每个逻辑条件的可能值至少被满足一次。

判定/条件覆盖是指设计足够多的测试用例，使得被测试程序中的每个判断本身的判定结果（真假）至少满足一次，同时，每个逻辑条件的可能值也至少被满足一次。即同时满足100%判定覆盖和100%条件覆盖的标准。在保证完成要求的情况下，测试用例的数目越少越好。

条件组合覆盖是指设计足够多的测试用例，使得被测试程序中的每个判断的所有可能条件取值的组合至少被满足一次。

路径覆盖就是设计足够多的测试用例，使得被测试程序中的每条路径至少被覆盖一次。

## 填空

1、对于循环嵌套结构的测试，有四种常用组合测试，分别为：内层最小循环次数，外层最大循环次数，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

2、定义/引用异常缺陷主要有三种情况，分别是变量在使用之前从未定义过，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

3、白盒测试主要的两个关注对象是\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

## 用例设计

### 1、 coverrageExample

**int** coverrageExample(**int** x, **int** y) {

**int** gift = 0;

**if** (x > 0 && y > 0) {

gift = x + y + 5; // 语句块1

} **else** {

gift = x + y - 5; // 语句块2

}

**if** (gift < 0) {

gift = 0; // 语句块3

}

**return** gift;// 语句块4

}

语句覆盖

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | 执行语句块 |
| Test Case1 | 2,2 | 9 | 语句块1，语句块4 |
| Test Case1 | -2,-2 | 0 | 语句块2，句块3，语句块4 |

如果&&修改成了||

判定覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | 判断1 | 判断2 |
| Test Case1 | 2,2 | 9 | T | F |
| Test Case2 | -2,-2 | 0 | F | T |

如果&&修改成了||

条件覆盖

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | C1 x>0 | C2 y>0 | C3 gift<0 |
| Test Case1 | 2,2 | 9 | T | T | F |
| Test Case2 | -2,-2 | 0 | F | F | T |

满足判定覆盖，不一定满足条件覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | C1 x>0 | C2 y>0 | C3 gift<0 | P1  x > 0 &&y > 0 | P2  gift<0 |
| Test Case1 | 2,2 | 9 | T | T | F | T | F |
| Test Case2 | -2,-2 | 0 | F | F | T | F | T |

判定条件覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | C1 x>0 | C2 y>0 | C3 gift<0 | P1  x > 0 &&y > 0 | P2  gift<0 |
| Test Case1 | 1,-1 | 0 | T | F | T | F | T |
| Test Case2 | -2,9 | 2 | F | T | F | F | F |

条件组合覆盖

x>0,y>0 组合1

x<=0,y>0组合2

x>0,y<=0组合3

x<=0,y<=0组合4

gift<0 组合5

gift>=0 组合6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | C1 x>0 | C2 y>0 | C3 gift<0 | 覆盖组合 |
| Test Case1 | 2,2 | 9 | T | T | F | 1,6 |
| Test Case2 | -2,-2 | 0 | F | F | T | 4,5 |
| Test Case3 | 2,-2 | 0 | T | F | T | 3,5 |
| Test Case4 | -2,2 | 0 | F | T | T | 2,5 |

路径覆盖

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试输入x,y | 测试输出gift | C1  x>0 | C2  y>0 | C3  gift<0 | P1  x > 0 &&y > 0 | P2  gift<0 | 路径 |
| Test Case1 |  | 路径不可能 |  |  |  |  |  | a-b-d-f |
| Test Case2 | -2,2 | 0 | F | T | T | F | T | a-c-d-f |
| Test Case3 | 2,2 | 9 | T | T | F | T | F | a-b-e-f |
| Test Case4 | -2,9 | 2 | F | T | F | F | F | a-c-e-f |

### 2、闰年 独立路径测试

以下代码由Java语言书写，请按要求回答问题。

**int** GetMaxDay(**int** year, **int** month) {

**int** maxday = 0;

**if** (month >= 1 && month <= 12)

{

**if** (month == 2) {

**if** (year % 4 == 0) {

**if** (year % 100 == 0) {

**if** (year % 400 == 0)

maxday = 29;

**else**

maxday = 28;

} **else**

maxday = 29;

} **else**

maxday = 28;

}

**else** **if** (month == 4 || month == 6 || month == 9 || month == 11)

maxday = 30;

**else**

maxday = 31;

}

**return** maxday;

}

（1）请画出以上代码的控制流图。

（2）请计算上述控制流图的环复杂度V（G）（独立线性路径数）

（3）假设输入的取值范围是1000 < year < 2001,请使用基本路径测试法为变量year、month设计测试用例（写出year取值、month取值、maxday预期结果），使其满足基本路径覆盖要求。

（1）请画出以上代码的控制流图。 注意：需要将复合条件分开单独画成节点。

例如：month >= 1 && month <= 12分别用节点2-1,2-2表示

month == 4 || month == 6 || month == 9 || month == 11分别用节点21-1,21-2,21-3,21-4表示

2-1 2-2 4 8 10 11 17 14 6 21-1 21-2 21-3 20 21-4 22 24 25 26

（2）请计算上述控制流图的环复杂度V（G）（独立线性路径数） 区域法：V(G)=11

边与节点的关系：V(G)=27-18+2=11 判定节点的关系：V(G)=10+1=11

（3）假设输入的取值范围是1000 < year < 2001,请使用基本路径测试法为变量year、month设计测试用例（写出year取值、month取值、maxday预期结果），使其满足基本路径覆盖要求。 测试用例编号 1 2 3 4 5 year 1500 1500 1500 1500 1500 month 0 16 4 6 9 maxday 0 0 30 30 30 覆盖的路径 2-1,26 2-2,26 2-1,2-2,4，21-1，22,25,26 2-1,2-2,4，21-1，21-2，22,25,26 2-1,2-2,4，21-1,21-2，21-3，22,25,26 2-1,2-2,4，21-1,21-2,21-3，21-4，22,25,26 2-1,2-2,4，21-1,21-2,21-3，6 1500 11 30 7 1500 7 31 21-4，24,25,26 8 9 10 1201 1240 1300 2 2 2 28 29 28 2-1,2-2,4，6，20,25,26 2-1,2-2,4，6，8,17,20，25,26 2-1,2-2,4，6，8,10,14,17，20，25,26 2-1,2-2,4，6，8,10,11，14,17，20，25,26 11 2000 2 29

# 黑盒测试

## 单选

1．黑盒法是根据程序的（ ）来设计测试用例的。

A．应用范围 B．内部逻辑

C．功能 D．输入数据

答案：C

2、下列关于白盒测试与黑盒测试的说法中错误的是（ ）。

A．用白盒测试来验证单元的基本功能时，经常要用黑盒测试的思考方法来设计测试用例

B．仅仅通过白盒测试，或仅仅通过黑盒测试都不能全面系统地测试一个软件

C．白盒测试适用于软件测试的各个阶段

D．在黑盒测试中使用白盒测试的手段，常被称为“灰盒测试”

答案：C

3、关于黑盒测试与白盒测试的区别，下列说法正确的是（ ）。

A．白盒测试侧重于程序结构，黑盒测试侧重于功能

B．白盒测试可以使用自动测试工具，黑盒测试不能使用工具

C．白盒测试需要开发人员参与，黑盒测试不需要。

D．黑盒测试比白盒测试应用更广泛

答案：A

4、若一个通讯簿最多可以输入100条记录，则下列选项中设计的测试用例最优的是（ ）。

A．分别输入1、50、100条记录

B．分别输入0、1、50、99、100条记录

C．分别输入0、1、99、100、101条记录

D．分别输入0、1、50、99、100、101条记录

答案：D

5、( )是一种黑盒测试方法，它是把程序的输入域划分成若干部分，然后从每个部分中选取少数代表性数据当作测试用例。

A．等价类划分法 B．边界值分析法

C．因果图法 D．场景法

答案：A

6、黑盒测试是通过软件的外部表现来发现软件缺陷和错误的测试方法，具体的说，黑盒测试用例设计方法包括（ ）等

A．等价类划分法、因果图法、正交试验设计法、功能图法、路径覆盖法、语句覆盖法

B．等价类划分法、边界值分析法、判定表驱动法、场景法、错误推测法、路径覆盖法

C．因果图法、边界值分析法、判定表驱动法、场景法、独立路径法

D．场景法、因果图法、正交试验设计法、边界值分析法、等价类划分法

答案：B

7、错误推测法的基本想法是：（ ）

A．划分有效等价类和无效等价类

B．划分原因和结果

C．列出所有的可能情况

D．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，根据它们选择测试用例

答案：D

8、与设计测试用例无关的文档是（ ）

A．项目开发计划 B．需求规格说明书

C．设计说明书 D．源程序

答案：A

9、（ ）是为了高效率地发现软件缺陷而精心设计的少量测试数据。

A．测试用例 B．测试计划

C．测试报告 D．测试分析报告

10、在某大学学籍管理信息系统中，假设学生年龄的输入范围为15~45，则根据黑盒测试中的等价类划分技术，下面对数据范围的划分正确的是（ ）。

A 可划分为2个有效等价类，2个无效等价类

B 可划分为1个有效等价类，2个无效等价类

C 可划分为2个有效等价类，1个无效等价类

D 可划分为1个有效等价类，1个无效等价类

参考答案：B

1. 测试程序时，不可能遍历所有可能的输入数据，而只能是选择一个子集进行测试，那么最好的选择方法是\_\_\_\_\_\_。  
   A．随机选择  
   B．划分等价类  
   C．根据接口进行选择  
   D．根据数据大小进行选择

答案：B

1. 用边界值分析法，假定10＜=X＜100（X为正整数），那么X在测试中应取的边界值是（）。

A．X=10，X=100

B．X=10，X=11，X=99，X=100  
C．X=10，X=11，X=99，X=100

D．X=9，X=10，X=50，X=101

答案：A

[解析] 边界值分析方法设计测试用例时，应选取正好等于、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。  
13、（）方法根据输出对输入的依赖关系设计测试用例。

A．等价类划分法 B．边界值分析法

C．因果图法 D．场景法

答案：C

14、如果程序的功能说明中含有输入条件的组合情况，则一开始就可以选用（ ）和决策表

A．等价类划分法 B．边界值分析法

C．因果图法 D．场景法

答案：C

15、关于黑盒测试，下列说法错误的是（ ）

A．黑盒测试着重测试软件的功能需求

B．失败测试是为了纯粹为了破坏软件而设计和执行测试案例的

C．边界值测试是黑盒测试特有的技术，不适合于白盒测试

D．黑盒测试无法发现规格说明中的错误，不能进行充分的测试

16、黑盒测试是根据软件的（）来设计测试用例的。

A.应用范围

B.内部逻辑

C.功能

D.输入数据

17、黑盒测试不能发现下面的哪种类型的错误。（）

A.界面错误

B.性能错误

C.功能遗漏

D.是否存在冗余代码

18、下列方法中，不属于黑盒测试的是（）。

A.基本路径测试法

B.等价类测试法

C.边界值分析法

D.事务流测试法

19、在某大学学籍管理信息系统中，假设学生年龄的输入范围为16—40，则从数字范围层面按等价类划分技术，下面划分正确的是（）。

A.可划分为2个有效等价类，2个无效等价类

B.可划分为1个有效等价类，2个无效等价类

C.可划分为2个有效等价类，1个无效等价类

D.可划分为1个有效等价类，1个无效等价类

20、单选题：假定某参数的输入范围是0—10的整数，采用边界值测试技术，最少需要（）个测试用例。

A.1

B.2

C.3

D.4

21、如果要对一个三角形程序进行测试用例设计，三角形程序的功能是根据三条边的边长来判断三角形的类型，输出有4种情况，分别是：不等边三角形，等腰但非等边三角形，等边三角形，不能构成三角形。且要求边长必须是整数。请问，最多需要设计多少个测试用例？

A.30

B.100

C.1728

D.无穷尽

22、请问，对于同一个输入条件来说，针对不同边界点可以分别设置大小不同的邻域值吗？

**A.**不可以

**B.**可以，但不一定必须不同

**C.**可以，而且必须设置不同大小的邻域值

**D.**视不同情况而定

23、在某个等价类中取测试数据的时候，该如何取值？

**A.**取非边界值

**B.**取边界值

**C.**随便取值，不考虑是否是边界值

* **D.**边界和非边界值都要取

## 填空

1、正交表的的符号表示为Ln(qs)，其符号中n表示\_\_\_\_\_\_\_\_\_\_\_\_\_，q表示\_\_\_\_\_\_\_\_\_\_\_\_，s表示\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 。

2、决策表由条件桩， \_\_条件项\_， \_\_\_动作桩 ， \_\_\_\_\_动作项\_ 四个元素组成。

3、测试程序时，不可能遍历所有可能的输入数据，而只能是选择一个子集进行测试，那么最好的选择方法是（划分等价类)。

4、边界值分析法的测试用例来自于（等价类的边界）

5、因果图分析法适用于（输入条件的各种组合）情况

3、若输入条件规定了取值范围，且取值范围上、下限之间的数据是有意义的数据，则取值范围内的数据构成一个有效等价类，小于下限、或大于上限的所有数据分别构成两个无效等价类；

(2)若输入条件规定了“必须如何”的条件，则满足必须条件的数据构成一个有效等价类，其他数据构成一个无效等价类；

4、划分等价类的标准：完备性、无冗余性

5、对于有两个输入条件的情况，如果x1条件有3个有效等价类有，x2条件有两个有效等价类，根据弱覆盖标准的要求，最少需要多少个测试用例？

## 用例设计

### 1、保费计算（等价类）

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%。

保险率和以下参数关系如下：

年龄：数字0~150

性别：字母组合，区分大小写，（FEMALE/MALE）

婚姻：字母组合(已婚/未婚)

抚养人数：数字1~9

其中前三项为必填项，最后一项为选填项。

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，具体规则如表1所示。

表1 保险公司计算保费费率的规则

|  |  |  |
| --- | --- | --- |
| 年龄 | 20~39岁 | 6点 |
| 40~59岁 | 4点 |
| 60岁以上及20岁以下 | 2点 |
| 性别 | MALE | **5点** |
| FEMALE | 3点 |
| 婚姻 | 已婚 | 3点 |
| 未婚 | 5点 |
| 抚养人数 | 1人扣0．5点  最多扣3点  （四舍五入取整） | |

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，10分）和测试用例表（10分）。

考点：等价类测试方法

参考答案

步骤1：确定输入

输入：年龄、性别、婚姻、抚养人数

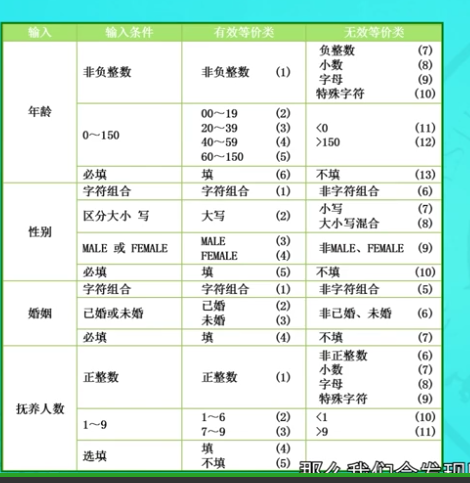
步骤2：确定每个输入项的输入条件

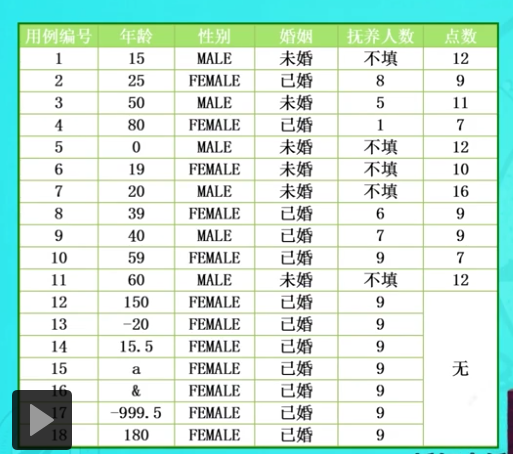
● 年龄：正整数，0~150、必填  
● 性别：字母组合，区分大小写，FEMALE/MALE、必填

● 婚姻：字母组合、已婚/未婚、必填  
● 抚养人数：正整数、1~9、选填

步骤3：对每个输入进行等价类分析，得到等价类表

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 数据要求 | 有效等价类 | 无效等价类 |
| 年龄 | 正整数 | 正整数 | 负整数 |
|  |  |  | 小数 |
|  |  |  | 字母 |
|  |  |  | 中文 |
|  |  |  | 其他字符 |
|  | 0~150 | 0~19 | <0 |
|  |  | 20~39 | >150 |
|  |  | 40~59 |  |
|  |  | 60~150 |  |
|  | 必填 | 填 | 不填 |
|  |  |  |  |
| 性别 | 字母组合 | 字母组合 | 非字母组合 |
|  | 区分大小写， | 大写 | 小写 |
|  |  |  | 大小写混合 |
|  | FEMALE/MALE | FEMALE | 非FEMALE、MALE |
|  |  | MALE |  |
|  | 必填 | 填 | 不填 |
|  |  |  |  |
| 抚养人数 | 正整数 |  | 负整数 |
|  |  |  | 小数 |
|  |  |  | 字母 |
|  |  |  | 中文 |
|  |  |  | 其他字符 |
|  | 1~9 | 1~6 | <1 |
|  |  | 7~9 | >9 |
|  | 选填 | 填 |  |
|  |  | 不填 |  |





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 输入条件 | 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 年龄 |  | 20<=x<=39 | （1） | 非正整数 | （12） |
|  | 40<=x<=59 | （2） |
|  | 0<=x<=19 | （3） |
|  | 60<=x<=150 | (16) |
| 性别 |  | ‘M’ | （4） | 非‘M’and非‘F’ | （13） |
|  | ‘F’ | （5） |
| 婚姻 |  | ‘已婚’ | （6） | 非‘已婚’and  非‘未婚’ | （14） |
|  | ‘未婚’ | （7） |
| 抚养人数 |  | null | （8） | 非 null and  非正整数 | （15） |
|  | 1<=x<=2 | （9） |
|  | 3<=x<=4 | （10） |
|  | x>=5 | （11） |

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入参数 | 覆盖等价类 | 输出参数 |
| 1 | （25，‘M’，‘已婚’，null） | （1）（4）（6）（8） | 保险率为0.6% |
| 2 | （44，‘F’，‘未婚’，2） | （2）（5）（7）（9） | 保险率为0.6% |
| 3 | （5，‘F’，‘已婚’，4） | （3）（5）（6）（10） | 保险率为0.1% |
| 4 | （88‘F’，‘已婚’，9） | （16（5）（6）（11） | 保险率为0.1% |
| 5 | （AA，‘M’，‘已婚’，null） | （12） | 输入错误 |
| 6 | （65，‘H’，‘未婚’，5） | （13） | 输入错误 |
| 7 | （78，‘F’，‘假婚’，null） | （14） | 输入错误 |
| 8 | （88，‘M’，‘已婚’，齐全） | （15） | 输入错误 |

### 2、注册（等价类+边界值）

利用等价类+边界值方式设计测试用例



参考答案：

步骤1：确定输入

步骤2：确定每个输入项的输入条件

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 数据要求 | 有效等价类 | 无效等价类 |
| 用户名 | [6~18] | [6~18] | <6 |
|  |  |  | >18 |
|  | 字母、数字、下划线 | 字母 | 特殊字符 |
|  |  | 数字 |  |
|  |  | 下划线 |  |
|  |  | 字母+数字+下划线 |  |
|  | 以字母开头 | 以字母开头 | 数字开头 |
|  |  |  | 下划线开头 |
| 密码 | [6~16] |  |  |
|  |  |  |  |
|  | 区分大小写 |  |  |

### 3、一个档案管理系统（等价类）

一个档案管理系统，某文本框要求用户输入以年月表示的日期。假设日期限定在2001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”。

解：

（1）划分等价类

|  |  |  |
| --- | --- | --- |
| **输入等价类** | **有效等价类** | **无效等价类** |
| 日期类型及长度 | 6位数字字符 | 非数字字符 >6位数字字符 <6位数字字符 |
| 年份范围 | 2001<year<2049 | year<2001 year>2049 |
| 月份范围 | 1<mouth<12 | mouth<1 mouth>12 |

 （2）等价类编号

|  |  |  |
| --- | --- | --- |
| **输入等价类** | **有效等价类** | **无效等价类** |
| 日期类型及长度 | (1)   6位数字字符 | (4)   非数字字符 (5)   >6位数字字符 (6)   <6位数字字符 |
| 年份范围 | (2)   2001<year<2049 | (7)   year<2001 (8)   year>2049 |
| 月份范围 | (3)   1<mouth<12 | (9)   mouth<1 (10)   mouth>12 |

（3）为有效等价类设计测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **输入数据** | **期望结果** | **覆盖范围** |
| 1 | 20161001 | 输入有效 | 等价类(1)(2)(3) |

 （4）为无效等价类设计测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **输入数据** | **期望结果** | **覆盖范围** |
| 1 | 20161001 | 输入有效 | 等价类(1)(2)(3) |
| 2 | 2016Oct1 | 输入无效 | 等价类(4) |
| 3 | 201610011 | 输入无效 | 等价类(5) |
| 4 | 2016100 | 输入无效 | 等价类(6) |
| 5 | 20001001 | 输入无效 | 等价类(7) |
| 6 | 20501001 | 输入无效 | 等价类(8) |
| 7 | 20160001 | 输入无效 | 等价类(9) |
| 8 | 20161301 | 输入无效 | 等价类(10) |

**2.分析**

**what?**

* 等价类划分法：
  + 是把所有可能的输入数据（即程序输入域）**划分**为若干部分（子集），然后从每一个子集中选取少量具有**代表性**的数据作为测试用例。
* 有效等价类：
  + 符合需求说明的，合理的，有意义的输入数据的集合。
  + 可检验程序是否实现了规格需求中所规定的功能。
* 无效等价类：
  + 不符合需求说明的，不合理的，无意义的输入数据的集合。
  + 检查程序异常处理情况，而且可以检查被测程序功能是否符合需求。

**why?**

* 等价类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中错误都是等效的。
* **合理的假定**：测试某等价类的代表值就等同于对这一类其他值的测试。

**how?**

1. 依据常用的原则划分等价类
2. 为每一个等价类规定唯一的编号
3. 设计一个新的测试用例，使其**尽可能多**的覆盖尚未被覆盖的有效等价类，重复这一步，直到所有有效等价类都被覆盖为止。
4. 设计一个新的测试用例，使其覆盖**一个**尚未被覆盖的无效等价类，重复这一步，直到所有的无效等类都被覆盖为止。

### 8、strtoint（等价类，边界值）+研究生考试题目

假设有一个把数字串转变成整数的函数。运行程序的计算机字长16位，用二进制补码表示整数。这个函数是用C语言编写的，它的说明如下：

int strtoint(char shortstr[]) 其中,参数shortstr[]定义如下：

char shortstr[6]；

被处理的数字串是右对齐的，也就是说，如果数字串比6个字符短，则在它的左边补空格。如果数字串是负的，则负号和最高位数字紧相邻(负号在最高位数字左边一位)。考虑到编译程序固有的检错功能，测试时不需要使用长度不等于6的数组做输入参数，更不需要使用任何非字符数组类型的输入参数。

分析这个程序的规格说明，可以划分出如下等价类：

有效输入的等价类有

(1) 1～6个数字字符组成的数字串(最高位数字不是零)；

(2) 最高位数字是零的数字串；

(3) 最高位数字左邻是负号的数字串；

无效输入的等价类有

(4) 空字符串(全是空格)；

(5) 左部填充的字符既不是零，也不是空格；

(6) 最高位数字右面由数字和空格混合组成；

(7) 最高位数字右面由数字和其他字符混合组成；

(8) 负号与最高位数字之间有空格；

合法输出的等价类有

(9) 在计算机能表示的最小负整数和零之间的负整数；

(10) 零；

(11) 在零和计算机能表示的最大正整数之间的正整数；

非法输出的等价类有

(12) 比计算机能表示的最小负整数还小的负整数；

(13) 比计算机能表示的最大正整数还大的正整数。

因为所用的计算机字长16位，用二进制补码表示整数，所以能表示的最小负整数是-32768，能表示的最大正整数是32767。

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 预期输出 | 覆盖的等价类 |
| 1 | ‘1’ | 1 | (1),(11) |
| 2 | ‘000001’ | 1 | (2),(11) |
| 3 | ‘-001’ | -1 | (3),(9) |
| 4 | ‘ ’ | 无效输入 | (4) |
| 7 | ‘++++1’ | 无效输入 | (5) |
| 8 | ‘1 3 ’ | 无效输入 | (6) |
| 9 | ‘1a3’ | 无效输入 | (7) |
| 10 | ‘- 23’ | 负号位置错误 | (8) |
| 11 | ‘0000’ | 0 | (10) |
| 12 | ‘-47561’ | 无效输入 | (12) |
| 13 | ‘134567’ | 无效输入 | (13) |
| 14 | ‘-32769’ | 无效输入 | 边界值测试 |
| 15 | ‘-32768’ | -32768 | 边界值测试 |
| 16 | ‘-32767’ | -32767 | 边界值测试 |
| 17 | ‘32766’ | 32766 | 边界值测试 |
| 18 | ‘32767’ | 32767 | 边界值测试 |
| 19 | ‘32768’ | 无效输入 | 边界值测试 |

### 4：8位计算机（考研题-等价类）

某一种8位计算机，其十六进制常数的定义是0x或者0X开头的十六进制整数，取值范围为-7f~7f（字母不区分大小写），如0x11,0x2A，-0X3c，使用的等价类划分设计测试用例。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | | 无效等价类 | |
| 开头字符 | 由0x或0X开头 | （1） | 以字母开头 以非0数字开头 | （2）（3） |
| 数值字符 | 数字或A—F的字母 | （4） | A—F以外的字母 | （5） |
| 数值字符个数 | ≥1个 | （6） | 0个 | （7） |
| 数值 | ≥-7f且≤7f | （8） | ＜－7f  ＞7f | （9）（10） |

     用例1：0x7F，      覆盖等价类（1）（4）（6）（8）  
     用例2：-0Xb，      覆盖等价类（1）（4）（6）（8）  
     用例3：0X0，       覆盖等价类（1）（4）（6）（8）  
     用例4：0x，        覆盖等价类（1）（7）  
     用例5：A7，        覆盖等价类（2）  
     用例6：-1A，       覆盖等价类（3）  
     用例7：0X8h，      覆盖等价类（1）（5）  
     用例8：0x80，      覆盖等价类（1）（4）（10）  
     用例9：-0XaB，     覆盖等价类（1）（4）（9）

### 2、两个字符的输入（因果图）

某软件规格说明中包含这样的要求：输入的第一个字符必须是A或B，第二个字符必须是一个数字，在此情况下进行文件的修改；但如果第一个字符不正确，则给出信息L；如果第二个字符不是数字，则给出信息M。

1. 写出文字说明（5分）
2. 画出因果图（5）
3. 写出相应的决策表（5分）
4. 写出测试用例（5分）

考点：因果图测试方法

1、分析程序的规格说明，列出原因和结果。

原因：C1----第一个字符是A

C2----第一个字符是B

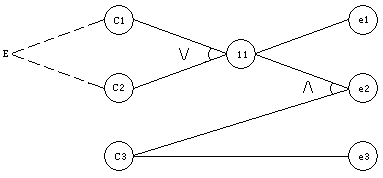
C3----第二个字符是一个数字

11为中间节点是导出结果的进一步原因。

结果：e1----给出信息L

e2----修改文件

e3----给出信息M



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 规则  选项 | | 1 | | 2 | 3 | | 4 | | 5 | | 6 | 7 | 8 |
| 条件 | C1 | | 1 | 1 | | 1 | | 1 | | 0 | 0 | 0 | 0 |
| C2 | | 1 | 1 | | 0 | | 0 | | 1 | 1 | 0 | 0 |
| C3 | | 1 | 0 | | 1 | | 0 | | 1 | 0 | 1 | 0 |
| 11 | |  |  | | 1 | | 1 | | 1 | 1 | 0 | 0 |
| 动作 | e1 | |  |  | | 0 | | 0 | | 0 | 0 | 1 | 1 |
| e2 | |  |  | | 1 | | 0 | | 1 | 0 | 0 | 0 |
| e3 | |  |  | | 0 | | 1 | | 0 | 1 | 0 | 1 |
| 不可能 | | 1 | 1 | |  | |  | |  |  |  |  |
| 测试用例 | |  | |  | | A5 | | A# | | B9 | B? | X2 | Y% |

### 3、航空（决策表）

假设中国某航空公司规定：

中国去欧美的航线所有座位都有食物供应。每个座位都可以播放电影。

中国去非欧美的国外航线都有食物供应，只有商务舱可以播放电影。

中国国内的航班的商务舱有食物供应，但是不可以播放电影 。

中国国内的航班的经济舱除非飞行时间大于2小时就有食物供应，但是不可以播放电影。

使用决策表法设计测试用例。

考点：决策表测试方法

解：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 航  线 | 中国去欧美？ | 1 | 0 | 0 | 0 | 0 | 0 |  |
| 中国去非欧美？ | 0 | 1 | 1 | 0 | 0 | 0 |  |
| 中国国内的航班？ | 0 | 0 | 0 | 1 | 1 | 1 |  |
| 经济舱？ | --- | 0 | 1 | 0 | 1 | 1 |  |
| 商务舱？ | --- | 1 | 0 | 1 | 0 | 0 |  |
| 飞行时间>2小时？ | --- | --- | --- | --- | 1 | 0 |  |
| 服  务 | 食物供应 | 1 | 1 | 1 | 1 | 1 | 0 |  |
| 播放电影 | 1 | 1 | 0 | 0 | 0 | 0 |  |

测试用例

|  |  |  |
| --- | --- | --- |
| 序号 | 输入数据 | 预期输出 |
| 1 | 美国 | 有食物和电影 |
| 2 | 日本的商务舱 | 有食物有电影 |
| 3 | 去韩国的经济舱 | 有食物无电影 |
| 4 | 去北京的商务舱 | 有食物无电影 |
| 5 | 去上海的经济舱 | 无食物无电影 |
| 6 |  |  |

### 4、中国象棋（因果图）

分析中国象棋中走马的实际情况（下面未注明的均指的是对马的说明）

1、如果落点在棋盘外，则不移动棋子；2、如果落点与起点不构成日字型，则不移动

棋子；3、如果落点处有自己方棋子，则不移动棋子；4、如果在落点方向的邻近交叉点有棋

子（绊马腿），则不移动棋子；5、如果不属于1-4 条，且落点处无棋子，则移动棋子；6、

如果不属于1-4 条，且落点处为对方棋子(非老将)，则移动棋子并除去对方棋子；7 如果不

属于1-4 条，且落点处为对方老将，则移动棋子，并提示战胜对方，游戏结束。

写出文字说明（5分）

画出因果图（5）

写出相应的决策表（10分）

考点：因果图测试方法

解：根据分析明确原因和结果

原因：

1、落点在棋盘上；

2、落点与起点构成日字；

3、落点处为自己方棋子；

4、落点方向的邻近交叉点有棋子；

5、落点处无棋子；

6、落点处为对方棋子（非老将）；

7、落点处为对方老将。

结果：

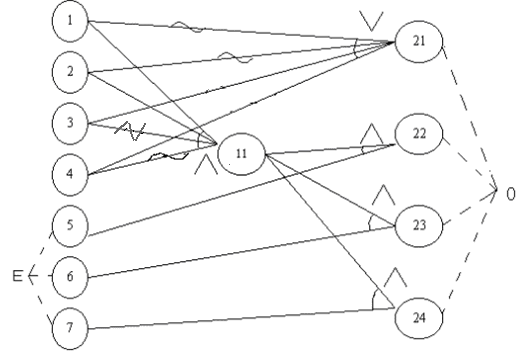
21、不移动棋子；

22、移动棋子；

23、移动棋子，并除去对方棋子；

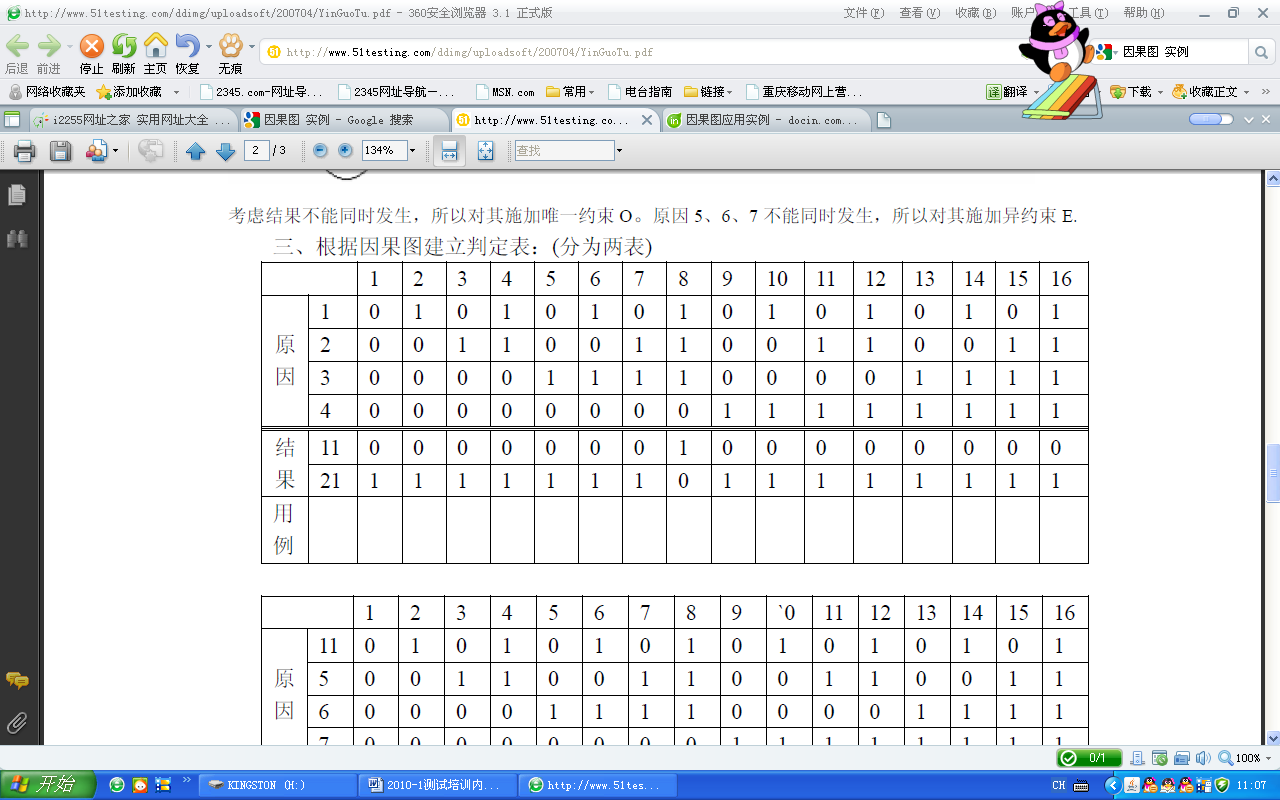
24、移动棋子，并提示战胜对方，结束游戏。

添加中间节点11，目的是作为导出结果的进一步原因，简化因果图导出的判定表



考虑结果不能同时发生，所以对其施加唯一约束O。原因5、6、7 不能同时发生，所以对其施加异约束E.

根据因果图建立判定表：(分为两表)



6、某网站用户注册时，要求用户必须输入用户名、密码及确认密码，对每一项的输入条件的要求如下：

用户名要求4~12位，只能使用英文字母、数字、“-”、“\_”这几种字符组合，并且首字母必须为字母或数字；

密码要求为6~12位，只能使用英文字母、数字以及“-”、“\_”、这几种字符组合，并且区分大小写

试用等价类划分方法为其设计测试用例。

1. 划分等价类

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 输入条件 | 无效等价类 | 编号 |
| 用户名 | 4~12位 | 1 | 用户名 | 少于4位 | 8 |
| 首字母必须为字母 | 2 | 多于12位 | 9 |
| 首字母必须为数字 | 3 | 首字母除字母或数字以为的字符 | 10 |
| **英文字母、数字、“-”、“\_”组合** | 4 | 组合中含有英文字母、数字、“-”、“\_”之外的字符 | 11 |
| 密码 | 6~12位 | 5 | 密码 | 少于6位 | 12 |
| 英文字母、数字、“-”、“\_”组合 | 6 | 多于12位 | 13 |
|  |  | 组合中含有英文字母、数字、“-”、“\_”之外的字符 | 14 |
| 确认密码 | 内容与密码相同 | 7 | 确认密码 | 内容与密码不同 | 15 |
|  |  |  | 内容与密码相同，但大小写不同 | 16 |

1. 设计测试用例

### 6、超市销售库存表（决策表）

**如果某产品销售好并且库存低，则继续销售，并增加该产品的进货；**

**如果该产品销售好，但库存量不低，则继续销售；**

**如果该产品销售不好，但库存量低，则产品下架；**

**如果该产品销售不好，且库存量不低，如有空货架，则继续销售，如果没有空货架，则该产品下架。**

**超市库存决策表的构造过程：**

1. **确定规则的个数**

**3个条件（销售、库存、有空货架），每个条件有两种取值，故有23=8种规则**

1. **列出所有的条件桩和动作桩**

**条件： 动作：**

**c1：销售好？ a1：增加进货**

**c2：库存低？ a2：继续销售**

**c3：有无空货架 a3： 产品下架**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **规则**  **选项** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 条件 | c1:销售好？ | T | T | T | T | F | F | F | F |
| c2:库存低？ | T | T | F | F | T | T | F | F |
| c3:有空货架？ | T | F | T | F | T | F | T | F |
| 动作 | a1:增加进货 | √ | √ |  |  |  |  |  |  |
| a2:继续销售 | √ | √ | √ | √ |  |  | V |  |
| a3:产品下架 |  |  |  |  | √ | √ |  | V |

合并

### 7、房产中介费的收取（决策表）

某房产中介公司的中介费政策如下：如果房屋销售总价少于10万元，那么基础中介费将是销售额的2%；

如果房屋销售总价大于10万元（含10万），但小于100万，那么基础中介费将是销售额的1.5%，外加1000元；

如果房屋销售总价大于100万元（含100万），那么基础中介费将是销售额的1%，外加1500元；

另外房屋销售单价和销售的套数对中介费也有影响。

如果单击低于1万元/m2元，则外加基础中介费的5%，此外若是老顾客，则减免外加基础中介费；

如果单击大于1万元/m2元（含1万），但低于2万元/m2元，则外加基础中介费的2.5%，若是老顾客，则减免外加基础中介费；

如果单击在2万元/m2元（含2万）以上，则减免外加基础中介费，若是老顾客，则减去基础中介费的5%；

### 9、库存（决策表）

以下列问题为例给出构造决策表的具体过程。

**如果某产品销售好并且库存低，则继续销售，并增加该产品的进货；**

**如果该产品销售好，但库存量不低，则继续销售；**

**如果该产品销售不好，但库存量低，则产品下架；**

**如果该产品销售不好，且库存量不低，如有空货架，则继续销售，如果没有空货架，则该产品下架。**

**超市库存决策表的构造过程：**

1. **输入域，确定规则的个数**

**3个条件（销售、库存、有空货架），每个条件有两种取值，故有23=8种规则**

**条件：**

**c1：销售好？**

**c2：库存低？**

**c3：有无空货架**

1. **输出域**

**动作：**

**a1：增加进货**

**a2：继续销售**

**a3： 产品下架**

1. **得到初始决策表**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **规则**  **选项** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 条件 | c1:销售好？ | T | T | T | T | F | F | F | F |
| c2:库存低？ | T | T | F | F | T | T | F | F |
| c3:有空货架？ | T | F | T | F | T | F | T | F |
| 动作 | a1:增加进货 | √ | √ |  |  |  |  |  |  |
| a2:继续销售 | √ | √ | √ | √ |  |  | V |  |
| a3:产品下架 |  |  |  |  | √ | √ |  | V |

1. **合并相似规则或者相似动作**

**如果表中有两条以上规则具有相同的动作，并且在条件项之间存在极为相似的关系，则可以合并。**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **规则**  **选项** | | **1、2** | **3、4** | **5、6** | **7** | **8** |
| 条件 | c1:销售好？ | T | T | F | F | F |
| c2:库存低？ | T | F | T | F | F |
| c3:有空货架？ | —— | —— | —— | T | F |
| 动作 | a1:增加进货 | √ |  |  |  |  |
| a2:继续销售 | √ | √ |  | √ |  |
| a3:产品下架 |  |  | √ |  | √ |

设计测试用例

|  |  |  |
| --- | --- | --- |
| **序号** | **输入条件** | **预期结果** |
| **supermaket\_01** | **产品销售好**  **库存低** | **继续销售**  **增加进货** |
| **supermaket\_02** | **产品销售好**  **库存不低** | **继续销售** |
| **supermaket\_03** | **产品销售不好**  **库存低** | **产品下架** |
| **supermaket\_04** | **产品销售不好**  **库存不低**  **有空货架** | **继续销售** |
| **supermaket\_05** | **产品销售不好**  **库存不低**  **无空货架** | **产品下架** |

### 10、城市的电话号码

城市的电话号码由三部分组成。这两部分的名称和内容分别是：

地区码：空白或三位数字

前缀：除0或者1以外开头的任意三位数字

后缀：认识4位数字

电话号码：以非0、非1开头的七位或者八位数字。



### 11、用户在线购物（场景法）。

选购物品后，进行在线购买。这是需要使用账号登录，登录成功进行付款交易，交易成功后生成订单，完成整个购物过程。

（ 1）确定基本流，备选流

基本流：选购—登录—付款—生成订单

备选流1：用户名不存在

备选流2：密码错误

备选流3：用户账户余额不足

备选流4：用户账户没钱

（2）根据基本流和备选流确定场景

场景1：购物成功（基本流）

场景2：用户名不存在（基本流，备选流1）

场景3：密码错误（基本流，备选流2）

场景4：账户余额不足（基本流，备选流3）

场景5：账户没钱（基本流，备选流4）

（3）每一个场景生成对应的测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试用例ID** | **场景** | **用户名** | **密码** | **账户余额** | **预期结果** |
| 1 | 1.成功购物 | V | V | V | 成功购物 |
| 2 | 2.用户名不存在 | I | n/a | n/a | 提示账号不存在 |
| 3 | 3.密码错误 | V | I | n/a | 提示密码错误，返回登录步骤 |
| 4 | 4.余额不足 | V | V | I | 提示账户余额不足，请充值 |
| 5 | 5.账户没钱 | V | V | I | 提示账户没钱，请充值 |

* V 表示这个条件必须是有效才能执行基本流
* I 表示在该种条件下激活备选流
* n/a 表示这个条件不使用测试用例

（4）设计测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试用例ID** | **场景** | **用户名** | **密码** | **账户余额** | **预期结果** |
| 1 | 1.成功购物 | User | Psw | 1024 | 成功购物 |
| 2 | 2.用户名不存在 | 1 | n/a | n/a | 提示账号不存在 |
| 3 | 3.密码错误 | User | 1 | n/a | 提示密码错误，返回登录步骤 |
| 4 | 4.余额不足 | User | Psw | 1 | 提示账户余额不足，请充值 |
| 5 | 5.账户没钱 | User | Psw | 0 | 提示账户没钱，请充值 |

**2.分析**

**what?**

分析软件应用场景，从用户角度出发，从场景角度设计测试用例，是一种面向用户的测试用例设计方法。

* 基本流：经过用例的最简单路径（正常流程）
* 备选流：一个备选流可以从基本流开始，在某个特定条件下执行，然后重新加入基本流中；也可以起源于另一个备选流，或终止用例。不在加入备选流中。（一般为错误流程）

**why?**

从用户角度出发，是一种面向用户的测试用例设计方法。

**how?**

1.根据需求，描述出程序的基本流以及各项备选流

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

4.对生成的测试用例重新复审，去掉多余的测试用例

5.测试用例确定后，为每一个测试用例确定测试数据值

### 某系统业务单据处理规则如下；（因果图）

解：

（1）列出可能的输入、输出并编号

输入：

* C1:单据处于提交审批状态
* C2:单据数据完整率达到80%
* C3:单据经过业务员确认

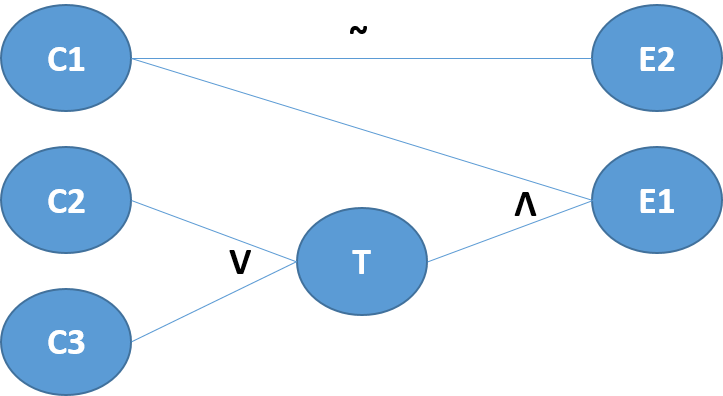
输出：

* E1:处理
* E2:不处理

（2）找出输入输出的对应关系

* 若单据不处于提交审批状态，则不处理
* 若单据处于提交审批状态且数据完整率达到 80%，则处理
* 若单据处于提交审批状态且经过业务员确认，则处理

（3）画出因果图



（4）将因果图转换为判定表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **条件** | C1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|  | C2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
|  | C3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| **中间结果** | T | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| **动作** | E1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
|  | E2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

（5）将判定表转化为测试用例 (略)

**2.分析**

**what?**

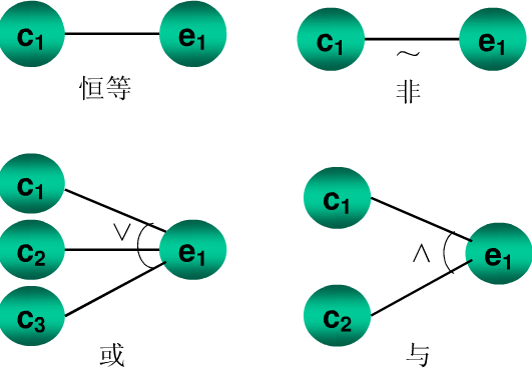
因果图法就是从需求中找出因（输入条件）果（输出结果或程序状态改变），通过分析输入条件之间的关系（组合关系、约束关系等）以及输入与输出之间的关系，制成因果图，转化为判定表，最后生成测试用例。

**why?**

等价类划分法和边界值分析法只考虑了输入条件，但是没有考虑输入条件之间的组合、制约关系，而实际输入之间存在着相互依赖关系。

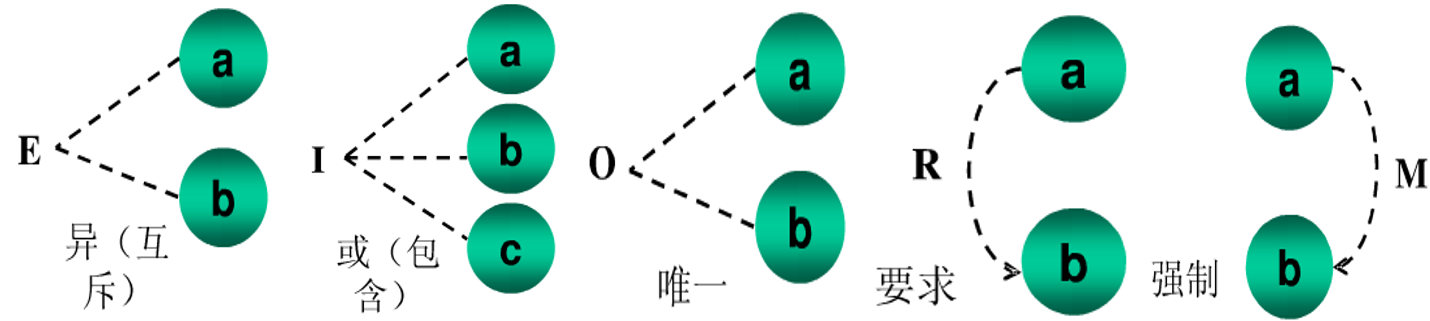
**how?**

**--因果图中的符号：**



* Ci 表示原因
* Ei 表示结果
* 恒等：原因结果同时出现
* 与：原因都出现，结果才出现；原因任意一个不出现，结果不出现
* 或：原因任意一个出现，结果出现；原因多不出现，结果不出现
* 非：原因不出现，结果出现；原因出现，结果不出现

**--因果图中的约束：**



输入条件

* E 表示a、b两个原因不能同时成立
* I 表示a、b、c中至少有一个条件成立
* O 表示a、b条件中有且仅有一个成立
* R 表示当a出现时b也必须出现

输出条件

* M 表示结果a是1，则结果b强制为0

**--因果图法步骤：**

1.分析所有可能的输入输出，并赋予标识符

2.找出输入输出之间的关系

3.根据关系画出因果图

4.将因果图转换为判定表

5.根据判定表生成测试用例

* 对于处于提交审批状态的单据，数据完整率达到80%以上或已经过业务员确认，则进行处理

### 14、电子不停车收费系统（ETC）（场景法）

**下面是对电子不停车收费系统（ETC）的基本流和备选流的描述**

**表1-1基本流**

|  |  |
| --- | --- |
| **编号** | **步骤描述** |
| A1 | 用例开始，ETC准备就绪，自动栏杆放下 |
| A2 | ETC与车辆通信，读取车辆信息 |
| A3 | 对车辆拍照 |
| A4 | 根据公式计算通行费用 |
| A5 | 查找关联账户信息，确认账户余额大于通行费用 |
| A6 | 从账户中扣除该费用 |
| A7 | 显示费用信息 |
| A8 | 自动栏杆打开 |
| A9 | 车辆通过 |
| A10 | 自动栏杆放下，ETC回到就绪状态 |

**表1-2备选流**

|  |  |  |
| --- | --- | --- |
| **编号** | **步骤名称** | **步骤描述** |
| B | 读取车辆信息出错 | 在基本流A2步骤，ETC读取车辆信息错误（重复读取五次），不够五次则返回A2；否则显示警告信息后退出基本流 |
| C | 账户不存在 | 在基本流A5步骤，在银行系统中不存在该账户信息，退出基本流 |
| D | 账户余额不足 | 在基本流A5步骤，账户余额小于通行费用，显示账户余额不足警告，退出基本流 |
| E | 账户状态异常 | 在基本流A5步骤，账户已销户、冻结或由于其他原因而无法使用，显示账户状态异常信息，退出基本流 |

**【问题1】**

使用场景法设计测试用例，指出所涉及到的基本流和备选流。基本流用A字母编号表示，备选流用表1-2中对应的字母标号表示。

例如：

T01：A

T02：A、B

**【问题2】**

针对问题1设计的测试用例，一次将初次读取车辆信息、最终读取车辆信息、账户号码、账户余额和账户状态等信息填入下述测试用例表中。表中行代表各个测试用例，列代表测试用例的输入值，用V表示有效数据元素，用I表示无效数据元素，n/a表示不适用，例如T01表示“成功通过”用例。

**表1-3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例编号** | **场景** | **初次读取车辆信息** | **最终读取车辆信息** | **账户号码** | **账户余额** | **账户状态** | **预期结果** |
| T01 | A | V | n/a | V | V | V | 扣除通行费，车辆顺利通过，用例结束 |
| T02 | A、B | I | I | n/a | n/a | n/a |  |
| T03 |  |  |  |  |  |  |  |
| T04 |  |  |  |  |  |  |  |
| T05 |  |  |  |  |  |  |  |
| T06 |  |  |  |  |  |  |  |
| T07 |  |  |  |  |  |  |  |
| T08 |  |  |  |  |  |  |  |

**参考答案：**

**【问题1】**

**T01：A**

**T02：A、B**

**T03：A、C**

**T04：A、D**

**T05：A、E**

**T06：A、B、C**

**T07：A、B、D**

**T08：A、B、E**

**【问题2】**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例编号** | **场景** | **初次读取车辆信息** | **最终读取车辆信息** | **账户号码** | **账户余额** | **账户状态** | **预期结果** |
| T01 | A | V | n/a | V | V | V | 扣除通行费，车辆顺利通过，用例结束 |
| T02 | A、B | I | I | n/a | n/a | n/a | 连续5次读取失败，显示警告信息，用例结束 |
| T03 | **A、C** | V | n/a | I | n/a | n/a | 账户不存在，提示警告信息，用例结束 |
| T04 | **A、D** | V | n/a | V | I | n/a | 账户余额不足，提示警告信息，用例结束 |
| T05 | **A、E** | V | n/a | V | V | I | 账户状态异常，提示警告信息，用例结束 |
| T06 | **A、B、C** | I | V | I | n/a | n/a | 多次读取车辆信息，最终成功；账户不存在，提示警告信息，用例结束 |
| T07 | **A、B、D** | I | V | V | I | n/a | 多次读取车辆信息，最终成功；账户余额不足，提示警告信息，用例结束 |
| T08 | **A、B、E** | I | V | V | V | I | 多次读取车辆信息，账户状态异常，提示警告信息，用例结束 |

### 15、网上银行支付交易系统（场景法）

场景法是黑盒测试中重要的测试用例设计方法。目前多数软件系统都是用事件触发来控制业务流程，事件触发时的情景便形成了场景，场景的不同触发顺序构成用例。场景法通过场景描述业务流程(包括基本流(基本流程)和备选流(分支流程))，设计用例遍历软件系统功能，验证其正确性。

下面是对网上银行支付交易系统的基本流和备选流的描述：

**表2-1基本流**

|  |  |  |
| --- | --- | --- |
| **编号** | **步骤名称** | **步骤描述** |
| A1 | 网上订购商品 | 用户登陆网站，订购所需商品，点击网上银行支付 |
| A2 | 输入银行卡信息 | 输入银行卡号和密码 |
| A3 | 校验银行卡信息 | 系统对银行卡号和密码进行校验 |
| A4 | 金额验证1 | 系统确认订单金额不大于卡内余额 |
| A5 | 金额验证2 | 系统确认订单金额不大于银行卡网上可支付额度 |
| A6 | 银行卡扣款 | 支付成功，系统从银行卡中扣除相应金额，返回订单号 |

**表2-1备选流**

|  |  |  |
| --- | --- | --- |
| **编号** | **步骤名称** | **步骤描述** |
| B | 密码不正确 | 在基本流A3步骤中，密码不正确（且密码输入尚未超过三次），重新加入基本流A2；否则退出基本流 |
| C | 银行卡内余额不足 | 在基本流A4步骤中，系统判断银行卡内余额不足以支付订单，退出基本流 |
| D | 银行卡网上可支付额度不够 | 在基本流A5步骤中，系统判断银行卡网上可支付额度小于订单金额，退出基本流 |

注：假定输入的银行卡号是正确的；不考虑备选流内循环情况。

**【问题1】**

使用场景法设计测试用例，指出所涉及到的基本流和备选流。基本流用A字母编号表示，备选流用表2-2中对应的字母标号表示。

**【问题2】**

请针对问题1设计的测试用例，依次将银行卡号、初次输入密码、最终输入密码、卡内余额、银行卡可支付额度等信息填入下述测试用例表中。表中行代表各个测试用例，列代表测试用例的输入值，用V表示有效数据元素，用I表示无效数据元素，n/a表示不适用，例如T01表示“成功支付”用例。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例编号** | **场景** | **银行卡号** | **初次输入密码** | **最终输入密码** | **卡内余额** | **银行卡可支付额度** | **预期结果** |
| T01 | A | V | V | n/a | V | V | 扣除购物金额，购物成功，用例结束 |
| T02 |  |  |  |  |  |  |  |
| T03 |  |  |  |  |  |  |  |
| T04 |  |  |  |  |  |  |  |
| T05 |  |  |  |  |  |  |  |
| T06 |  |  |  |  |  |  |  |

**参考答案**

**【问题1】**

T01：A

T02：A、B

T03：A、C

T04：A、D

T05：A、B、C

T06：A、B、D

**【问题2】**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例编号** | **场景** | **银行卡号** | **初次输入密码** | **最终输入密码** | **卡内余额** | **银行卡可支付额度** | **预期结果** |
| T01 | A | V | V | n/a | V | V | 扣除购物金额，购物成功，用例结束 |
| T02 | A、B | V | I | I | n/a | n/a | 连续3次密码错误，显示警告信息，用例结束 |
| T03 | A、C | V | V | n/a | I | n/a | 卡内余额不足，提示警告信息，用例结束 |
| T04 | A、D | V | V | n/a | V | I | 网上支付额度不够，提示警告信息，用例结束 |
| T05 | A、B、C | V | I | V | I | n/a | 多次输入密码，最终成功；卡内余额不足，提示警告信息，用例结束 |
| T06 | A、B、D | V | I | V | V | I | 多次输入密码，最终成功；网上支付额度不够，提示警告信息，用例结束 |

# 单元测试

## 单选

1、单元测试的测试用例主要根据（ ）的结果来设计。

A．需求分析 B．源程序

C．概要设计 D．详细设计

答案：B

2、单元测试的测试目的是（ ）

A．保证每个模块能正常工作 B．保证每个组件能正常工作

C．确保缺陷得到解决 D．使程序正常运行

答案：C

3、Junit是（ ）语言的单元测试框架。

A．C B．.Net

C．Java D．C++

答案：C

4、在单元测试中由被测模块调用，用以代替被测模块所调用的子模块被称作（）。

A.驱动模块

B.桩模块

C.骨架模块

D.响应模块

5、单元测试主要针对模块的几个基本特征进行测试，该阶段不能完成的测试是（）。

A.系统功能

B.局部数据结构

C.重要的执行路径

D.错误处理

## 填空

1. 在单元测试中，为了能够单独对某个可测单元进行测试，常需要设计两个模块，分别是\_\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_。
2. 有限状态机的三种表示形式分别为：状态树，\_\_\_\_\_\_\_\_\_\_\_\_ ， \_\_\_\_\_\_\_\_\_\_\_ 。
3. 因果图中有四种基本的因果关系，分别是恒等，\_\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_\_。

# 集成测试

1、集成测试分为渐增组装测试和（ ）

A．非渐增组装测试 B．确认测试

C．单元测试 D．测试计划

答案：A

2、集成测试中使用的辅助模块分为驱动模块和（ ）

A．传入模块 B．主模块

C．桩模块 D．传出模块

答案：C

3、驱动模块模拟的是( )

A．子模块 B．第一模块

C．底层模块 D．主程序

答案：D

4、集成测试既可以将所有模块组装到一起然后进行测试，也可以在组装的过程中边连接边测试。对

# 系统测试

1、用户文档测试中不包括的是（）

A．用户需求说明 B．操作指南

C．用户手册 D．随机帮助

答案：A

2、下列（ ）情况需要考虑引入自动化测试。

A．需要重复执行很多次的测试 B．只执行一次的测试

C．不重要的测试 D．很快有回报的测试

答案：A

3、WEB应用中一般采用架构的是（ ）

A．C/S结构 B．B/S结构

C．集群结构 D．服务器互备份

答案：B

4、下列哪个不是WEB测试的测试内容？（ ）

A．功能测试 B．兼容性测试

C．界面测试 D．用户测试

答案：D

5、导航测试属于（ ）

A．功能测试 B．性能测试

C．可用性/可靠性测试 D．压力测试

答案：C

6、Web测试的一个重要特征是（ ）

A．图片 B．文字

C．链接 D．视频

答案：C

7、确保测试与（ ）无关，不要依赖使用过期的数据进行测试。

A．时间 B．地点

C．软硬件资源 D．测试人员

答案：A

8、单元测试计划、集成测试计划、确认测试计划分别在（ ）阶段制定。

A.详细设计、需求分析、总体设计

B.总体设计、详细设计、需求分析

C.需求分析、总体设计、详细设计

D.详细设计、总体设计、需求分析

# 验收测试

## 单选

1、下列关于alpha 测试的描述中正确的是：（ ）

A．alpha 测试需要用户代表参加 B．alpha 测试需要开发人员参加

C．alpha 测试是集成测试的一种 D．alpha 测试是验收测试的一种

答案：B

2．验收测试的测试用例主要根据（ ）的结果来设计。

A．需求分析 B．源程序

C．概要设计 D．详细设计

答案：A

3、以用户为主导的测试成为（ ）

A．验收测试 B．白盒测试

C．黑盒测试 D．系统测试

# 软件开发与测试模型

1、以下哪个不是软件测试模型（ ）  
A．X模型 B．K模型  
C．V模型 D．W模型

2、测试驱动开发的简称是（ ）

A．ADD B．TTD

C．TDD D．TDT

答案：C

# 测试过程管理

1、软件工程中描述生存周期的瀑布模型一般包括计划、( )、设计、编码、测试、维护等几个阶段。

A．需求分析 B．需求调查

C．可行性分析 D．问题定义

答案：A

2、缺陷的生命周期可以这样划分new、confirmed、fixed、closed、reopen

每一个缺陷都是由测试人员发现并提交的，这个状态标注为new/新建，缺陷被移交后，由相应的负责人进行接受，即confirmed/确认状态，相应的开发人员解决了缺陷后，状态修改为fixed/解决，并且将其发给测试人员进行（回归测试），防止产生其他错误，如果确定已经解决，缺陷的状态修改为closed/关闭，否则需要返还给该缺陷的开发人员进行修正。有的缺陷在以前的版本中已经关闭，但是新的版本又重新出现，则需要将其状态修改为reopen（重新打开）

# 综合

判断题：

1. 软件测试就是为了验证软件功能实现的是否正确，是否完成既定目标的活动，所以软件测试在软件工程的后期才开始具体的工作。 （F）
2. 广义的软件测试由“确认”、“验证”、“测试”3个方面组成。T
3. 软件测试等于程序测试。F
4. 测试人员在测试过程中发现一处问题，如果问题影响不大，而自己又可以修改，应立即将此问题正确修改，以加快、提高开发的进程。（F）
5. 软件测试的目的是尽可能多的找出软件的缺陷。（T ）
6. 要充分注意软件测试中的群集现象。（T ）
7. 好的测试方案是极可能发现迄今为止尚未发现的错误。（T）
8. 测试人员要坚持原则，缺陷未修复完坚决不予通过。（F ）
9. 发现错误多的模块，残留在模块中的错误也多。( T )
10. 测试人员说：“没有可运行的程序，我无法进行测试工作”。 F
11. 负载测试是验证要检验的系统的能力最高能达到什么程度。（F）
12. 白盒测试方法比黑盒测试方法好。 F
13. 在软件开发过程中，若能尽早暴露其中的错误，则为修复和改进错误所花费的代价就会降低。T
14. 产品质量检验阶段是质量管理的发展阶段之一。
15. V模型不能适应较大的需求变化。T
16. W模型能够较早地发现缺陷。T
17. 验收测试是由最终用户来实施的。（F ）
18. 单元测试能发现约80% 的软件缺陷。（T ）
19. 我们可以人为的使得软件不存在配置问题。（F ）
20. 集成测试计划在需求分析阶段末提交。（F ）
21. Beta 测试是验收测试的一种。（T）
22. 自底向上集成需要测试员编写驱动程序。T
23. 为了快速完成集成测试，采用一次性集成方式是适宜的。 F
24. alpha 测试不是验收测试的一种。F
25. 必须要求用户参与的测试阶段是单元测试。F
26. 项目立项前测试人员不需要提交任何工件。（T）
27. 测试需求一定要写得很细很细。 F
28. 测试需求就是软件需求。 F
29. 测试组负责软件质量。F
30. 代码评审是检查源代码是否达到模块设计的要求。（F）
31. 代码评审员一般由测试员担任。（F ）
32. 一个程序中所含有的路径数与程序的复杂程度有着直接的关系。T
33. 静态检查就是看代码。F
34. 等价类法和边界值着重考虑输入条件，而不考虑输入条件的各种组合，也不考虑输入条件之间的相互制约关系。T
35. 在实际测试中，边界值分析法和等价类划分法经常结合使用。T
36. 因果图法是建立在决策表法基础上的一种白盒测试方法。F
37. 黑盒测试法又叫功能测试或数据驱动测试。T
38. 测试用例是在测试执行后写的。 F
39. 测试用例一般是由测试设计人员写的。T
40. 在设计测试用例时，应包括合理的输入条件和不合理的输入条件。 T
41. QTP是一种自动化测试工具。T
42. 测试自动化是万能的。F
43. LoadRunner是性能测试工具。 T
44. 采用自动化测试有可能延误项目进度。T
45. 面向对象的软件测试是采用面向对象的概念和原则，用结构化的方法构建。F
46. Web网站测试需要考虑数据库测试。 T
47. Web网站测试不需要考虑功能测试。F
48. 敏捷测试不需要写测试计划。 F
49. Junit是一个开源的Java单元测试框架。T
50. WAS是单元测试工具。 F