

1. 单元测试-集成测试-系统测试-验收测试

功能测试一般默认在系统测试

单元测试是最小的测试单位，也称模块测试

软件开发周期，哪个阶段缺陷发现的最多-需求分析阶段

软件测试的对象-文档/代码

等价类：将测试分为：有效/无效类

有效等价类：对于程序来说有意义、合理的输入

无效等价类：对于程序来说无意义、不合理的输入

因果图法适合：多个控件之间有组合或制约关系

正交排列法的特点：每列数据均匀分布，每两列的有序数对也是均匀分布

m ：每个控件的取值个数

k ：控件的个数（正交表的列数）

n ：测试用例的个数（固定的）

k 值不合适：找最接近的大一点的，多余的列删除

M 值不合适：少数服从多数：控件取值相同的最多的；最大值： M ：控件取值数最大的值

6.5 正交排列的测试操作步骤

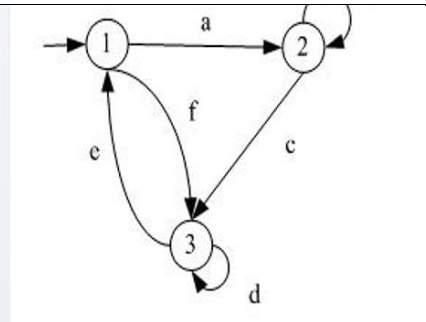
- 步骤1：分析需求，列出所有需要组合的**控件**以及每个控件的**取值**。
- 步骤2：选择一个合适的**正交表**（根据 m 和 k ）
- 步骤3：把**控件**及其取值映射到**正交表**中
 - ✓ 把正交表中的因子（**列标题**）分别替换成4个**控件名**
 - ✓ 把每列中的**状态**（每列中的1, 2, 3...取值）分别替换成这个控件的**取值**
- 9. 步骤4：根据最终的正交表，编写测试用例（不必要）

状态迁移图：用什么表示节点？用什么表示状态之间的迁移？

○ 表示节点

→ 表示状态间的迁移关系

在→ 标识迁移条件



白盒测试:

基本路径测试的目的?

基于控制流图, 计算程序的环路复杂性, 并导出基本可执行路径集合然后依据此设计测试用例。

什么是基本独立路径?

一条独立路径是指, 和其他的独立路径相比, 至少引入一个新处理语句或一个新判断的程序通路。V(G) 值正好等于该程序的独立路径的条数。

保证...执行?

设计出的测试用例要保证程序的每个可执行语句至少执行一次。

白盒测试的每种判定方法的原则:

1. 语句覆盖: 每条可执行语句 (; 结尾) 执行一次
2. 判定覆盖: 每条真假分支执行一次
3. 条件覆盖: 判定中每个条件真假值执行一次
4. 判定/条件覆盖: 判定中每个条件真假值至少取一次, 真假分支都执行一次
5. 条件组合覆盖: 判定中每个条件的真假值的所有组合至少取一次

边界值两边的点是边界吗?

边界值: 有效等价类与无效等价类的分界点

*边界值点两边的点称为次边界值

缺陷报告里的优先级别与严重程度代表什么

它们均分为 urgent, high, medium, low 四个等级

优先级: 表示处理和修正软件缺陷的先后顺序的指标, 即哪些缺陷需要优先修正, 哪些缺陷可以稍后修正。考虑因素: 严重程度, 影响范围, 人力资源, 修复代价。

严重程度: 是指 BUG 对软件质量的破坏程度, 即此 Bug 的存在将对软件的功能和性能产生怎样的影响。

特例是什么

测试的整个流程:

软件测试的主要流程:

1. 熟悉需求, 编写测试计划
2. 分析并设计、编写测试用例
3. 执行测试用例

4. 分析并记录测试结果
5. 发现缺陷，填写缺陷报告并跟踪管理缺陷
6. 编写测试总结报告

关于小数的等价类划分以及边界值的确定

若未明确说明小数范围内不包含整数则默认包含整数。

划分无效等价类时，需要考虑小数的位数不符合需求要求的位数的情况。

小数的次边界值确定要根据有效范围确定。

小数有专门用于位数的边界值，要测试有效位数以及两边的次边界。

A	B	C	D	E	F	G
控件名称	数据要求	有效等价类	有效边界值	无效等价类	无效边界值	所属用例
工资	1.00~30000.00之间的小数（或整数），可以为空，小数点后最多保留2位	1.00~30000.00之间的小数（或整数）	1.00 30000.00 1.01 29999.99 空	小于1.00的小数（或整数） 大于30000.00的小数（或整数） 特殊字符 字母 汉字	11位字符 0.99 30000.01	19 1,6-14,15-19,28-29 2 3 4 5 20 21 22 23 24 25 26 27

如上图，有效边界值中须指出或整数的可能，写有效边界值时，按小数情况分析，如有效边界值，若为整数，则应该是 1, 30000, 2, 30001. 但小数范围时，则取小数高一位，即 1.01 与 30000.01。无效等价类同理。并且写测试用例时即使 1 是整数，也需写出后面的两位，即 1.00。

云班课的选择題

单选题

SQA 是一项（ ）工作，侧重于对流程的（ ） 和（ ）。

- A. 管理 评审 监控
- B. 管理 评审 验证
- C. 技术性 评估 监控
- D. 管理 评估 监控

正确答案： A

单选题

软件测试的目的是

- A. 发现程序中的所有错误
- B. 尽可能多地发现程序中的错误
- C. 证明程序是正确的
- D. 调试程序

正确答案： B

单选题

3 分

软件开发周期中，产生缺陷最多的阶段是

- A. 需求分析阶段
- B. 概要设计阶段
- C. 详细设计阶段
- D. 编写代码阶段

正确答案： A

单选题

下列关于软件缺陷的说法中正确的一项是

- A. 在软件开发生命周期中，引入缺陷数量最多的阶段是编码阶段
- B. 缺陷管理的过程自测试人员提出缺陷开始至开发人员修改缺陷结束
- C. 开发人员在修改缺陷后需要验证缺陷，并且根据验证结果关闭或重新打开缺陷
- D. 在修改缺陷前，开发团队必须确认缺陷是否有效

正确答案： D

单选题

下列关于缺陷报告的说法中不正确的是

- A. 每个缺陷报告中只能描述一条缺陷，而不要包含多条缺陷；
- B. 缺陷报告中要提供复现缺陷的精确步骤
- C. 缺陷报告中需要提供发现缺陷的特定环境、特定设置的描述
- D. 缺陷报告需要简洁，因此不要包含图片、日志等附加信息。

正确答案： D

单选题

下列关于缺陷分类的说法中不正确的是

- A. 可以将缺陷分为功能缺陷、界面缺陷、警告、功能等几类
- B. 可以将缺陷按照严重程度可分为 Urgent/Very high/High/Medium/Low
- C. 可以将缺陷按照优先级分类，优先级的高低可以从用户需求角度来考虑
- D. 一般说来，低严重性缺陷(如界面缺陷)优先级较低

正确答案： D

单选题

关于缺陷报告的用途说法错误的是

- A. 记录缺陷
- B. 分类总结统计缺陷
- C. 监督评价管理开发人员的工作
- D. 跟踪管理缺陷

正确答案： C

单选题

缺陷的严重程度和优先级严格成正比

- A. 正确
- B. 错误

正确答案： B

单选题

关于缺陷报告的说法正确的是

- A. 为提高效率，多个类似的场景发现的同类缺陷可以报告在一个缺陷报告里
- B. 作为测试人员应该坚持原则，缺陷不修复坚决不予通过
- C. 发现了缺陷应该及时直接找到开发人员进行修复
- D. 在开发经理发现缺陷为假缺陷的情况下，可以将缺陷报告直接返回给测试人员

正确答案： D

单选题

在软件开发周期中造成缺陷最多的阶段是：

- A. 编码阶段
- B. 需求分析阶段
- C. 概要设计阶段
- D. 详细设计阶段

正确答案： B

单选题

因果图设计，首先需要根据程序规格说明，分析并确定（ ）。

- A. 原因
- B. 结果
- C. 中间节点
- D. 原因和结果

正确答案： D

单选题

因果图中，a，b，c为原因条件，表示这三个条件有且仅有一个1的关系是（ ）。

- A. 互斥
- B. 唯一
- C. 包含
- D. 要求

正确答案： B

单选题

因果图最终转换成判定表，设计测试用例。

- A. 正确

B. 错误

正确答案： A

单选题

判定表由四部分组成，下面()部分不属于这四部分之一。

A. 条件桩

B. 动作项

C. 条件项

D. 结果桩

正确答案： D

单选题

判定表有 n 个条件，每个条件都取真、假值，则产生 () 个规则。

A. n

B. $2n$

C. $2 \times n$

D. 2^n

正确答案： D

单选题

判定表分析法可以进行完备的测试，是因为它能把所有条件的组合罗列出来，并针对每种组合执行不同的动作。

A. 正确

B. 错误

正确答案： A

单选题

开发桩模块和驱动模块经常是集成测试阶段我们需要完成的内容

A. 正确

B. 错误

正确答案： A

单选题

由少数几个经验丰富的测试人员用少量时间对软件的主要核心功能进行测试，决定是否接受作为最新的内侧版本的测试叫 ()

A. 验收测试

B. 冒烟测试

C. 确认测试

D. 猴子测试

正确答案： B

单选题

以下说法正确的是

A. 验收测试一般都是有用户自己亲自进行的。

B. 集成测试以黑盒测试为主，以《详细设计说明书》为依据进行测试

C. 系统测试重点关注各个组成部分组成系统之后各个接口的性能

D. 进入系统测试阶段之前需要进行确认测试

正确答案： D

多选题

以下属于系统测试内容的是

- A. 功能测试
- B. 易用性测试
- C. 兼容性测试
- D. 性能测试

正确答案：ABCD

单选题

下列不属于性能测试的性能指标的一项是（ ）

- A. 并发用户数
- B. 界面布局的美观度
- C. 页面响应时间
- D. 网络吞吐量

正确答案：B

单选题

在单元测试中利用工具对源代码中的缺陷进行走查、评审属于：

- A. 静态白盒测试
- B. 静态黑盒测试
- C. 动态白盒测试
- D. 动态黑盒测试

正确答案：A

单选题

对需求规格说明书的审查活动属于：

- A. 静态白盒测试
- B. 静态黑盒测试
- C. 动态白盒测试
- D. 动态黑盒测试

正确答案：B

多选题

这个 Web 系统实际并发用户数是指系统中同时进行（ ）操作的用户数

- A. 浏览页面
- B. 填写订单
- C. 提交订单
- D. 查询订单

正确答案：CD

单选题

因果图的符号分成两大类，基本符号表示的是（ ）

- A. 输入和输入之间的关系
- B. 输入和输出之间的关系
- C. 输出和输出之间的关系
- D. 以上都不对

正确答案：B

多选题

因果图的符号分成两大类，约束符号表示的是（ ）

- A. 输入和输入之间的关系
- B. 输入和输出之间的关系

- C. 输出和输出之间的关系
- D. 以上都不对

正确答案： AC

多选题

判定表是由以下哪些项组成（）

- A. 条件桩
- B. 动作桩
- C. 条件项
- D. 动作项

正确答案： ABCD

单选题

关于因果图和判定表以下说法正确的是（）

- A. 判定表是将部分关键的组合列出来并不是全集
- B. 因果图和判定表分别是两个独立进行测试的方法
- C. 因果图法更便于表达图形之间的制约关系
- D. 判定表法可以将组合和制约关系都表示出来

正确答案： C

多选题

请选出以下适合局部测试的测试方法

- A. 因果图和判定表法
- B. 等价类和边界值法
- C. 状态转换法
- D. 正交排列法

正确答案： ABD

简答题

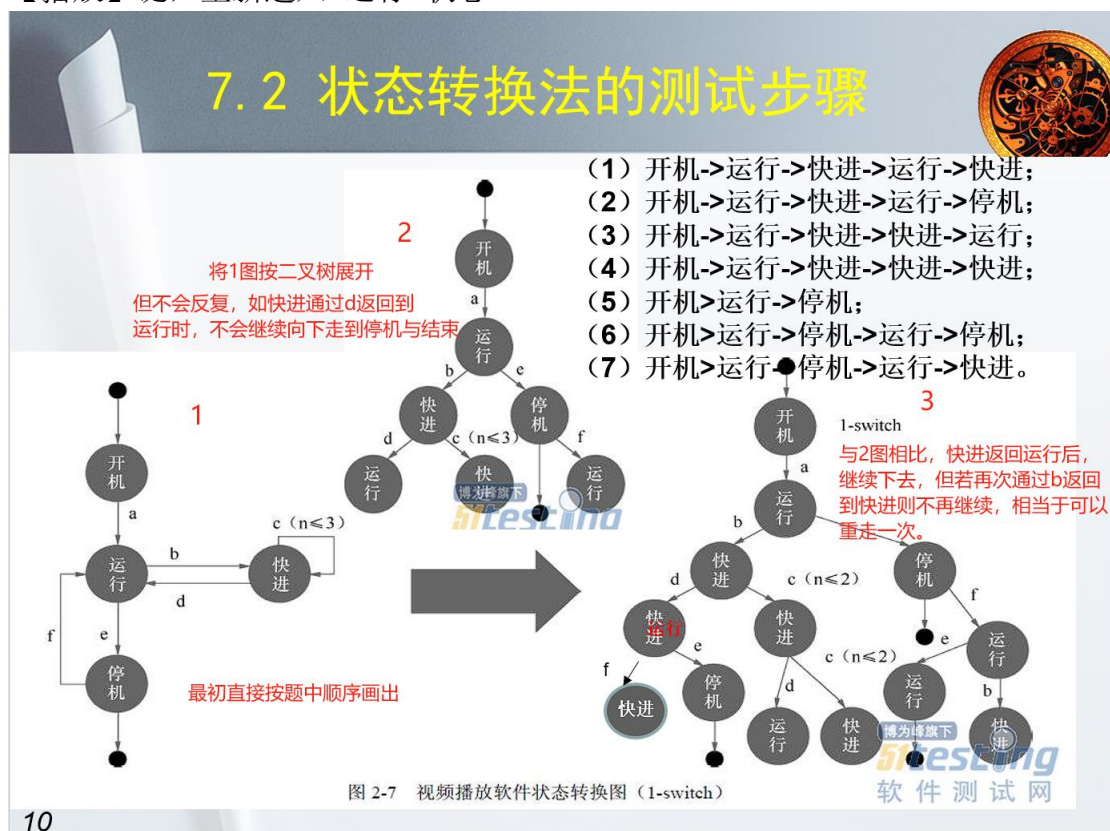
1. 状态转换法核心思想与使用场景

核心思想：

通过引入状态图（State Diagram）来描述软件测试对象和软件测试数据、对象状态之间的关系

例子

打开视频播放机，系统处于“开机”状态，单击【运行】键，系统处于“运行”状态；单击【停机】键，播放结束，系统处于停机状态；在“运行”状态单击【快进】键，进入“快进”状态，【快进】键最多可以按4次，分别为2倍数、4倍数、8倍数和16倍数前进；快进状态单击【停止】键返回“运行”状态，停机状态单击【播放】键，重新进入“运行”状态。



使用场景：

软件测试对象的输出和行为方式不仅受当前输入数据的影响，同时还与软件测试对象之前的执行情况、之前的事件或以前的输入数据等有关。

2. 场景法的测试思想（ATM 机的例子）

测试思想：

先用场景法对业务逻辑（整体）作测试

再对控件做细节测试《等价类、边界值、因果图、判定表》

适用：大部分业务逻辑较为复杂的软件系统

场景法基于业务逻辑

测试人员要把自己当作最终用户《尽可能真实的模拟用户在使用此软件的操作场景》

基本流、备选流、基本流程

基本流（有效流，正确流）

备选流（无效流）

步骤 1：熟悉、分析需求、整理出业务逻辑，列出基本流和备选流

步骤 2：根据基本流和备选流生成相应的场景分析表

步骤 3：根据场景设计测试用例

举例：ATM

1. 插卡 2. 验证卡 3. 选择取款业务 4. 输入金额

…取款业务的基本流就是对于“取款”这一操作的一套正确流程

备选流即这一流程中各环节可能出现的错误。如在验证卡时，该卡可能无效或者不是本行的卡，那么“卡无效”就是一个备选流。

3. 缺陷报告的主要作用是什么？

①缺陷报告是开发人员与测试人员沟通的重要工具

②记录缺陷

③跟踪管理缺陷

④对缺陷进行分类和统计

*缺陷报告的注意事项：

一个缺陷对应一个报告。

保证再现。（详细描述步骤）

完整统一（发现该缺陷时的环境与设置）

对严重性和优先级划分准确客观。

缺陷的定义：

1. 需求要求的功能未实现

2. 实现了没要求的功能

3. 出现了需求中明确指出不应该出现的问题

4. 软件没有实现需求未明确说明，但应该实现的问题

5. 软件难以理解，不易操作，运行缓慢等，站在用户角度上所有认为不好的问题。

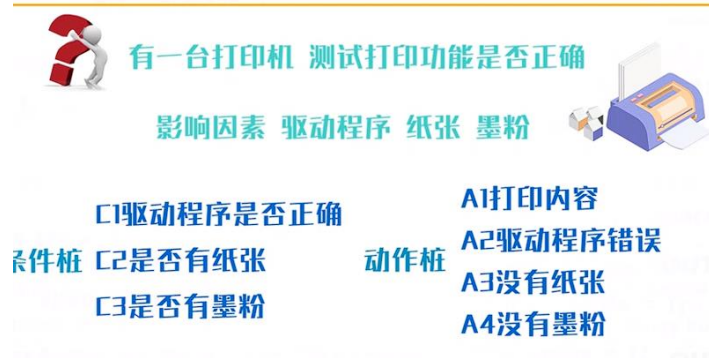
小缺陷：“不可重现”的缺陷

*判定表法

确定条件桩、动作桩...

条件桩：即对于该功能而言的所有必要条件

动作桩：即该功能的所有可能的结果



优化用例的过程（将可以合并的合并）

如下图左，通过条件桩之间的排列组合，可以得出不同条件下所对应的结果可能性，其中 1 代表该项所对应的值为真，0 代表为假。若有 n 个条件项，则有 2^n 种可能的组合。

	ID	R1	R2	R3	R4	R5	R6	R7	R8		ID	项目名称	R1	R2	R4/6	R3/5/7/8
条件项	C1	1	0	1	1	0	0	1	0	条件项	C1	驱动程序是否正确	1	0	-	-
	C2	1	1	0	1	0	1	0	0		C2	是否有纸张	1	1	1	0
	C3	1	1	1	0	1	0	0	0		C3	是否有墨粉	1	1	0	-
动作项	A1	1	0	0	0	0	0	0	0	动作项	A1	打印内容	1	0	0	0
	A2	0	1	0	0	0	0	0	0		A2	提示驱动程序不对	0	1	0	0
	A3	0	0	1	0	1	0	1	1		A3	提示没有纸张	0	0	0	1
	A4	0	0	0	1	0	1	0	0		A4	提示没有墨粉	0	0	1	0

优化时，根据实际情况来判断，如 R4 与 R6，它们的条件桩除了 C1 以外，均为没有墨粉，而结果也均为没有墨粉。故将 R4 与 R6 合并，因为这说明 C1 对它们这个组合的结果完全没有影响。同理，根据左表，结果 A3（没有纸张）的组合有 R3\R5\R7\R8，与上述过程一样，将它们合并，并将对结果不是关键性因素的条件桩取“-”。

*等价类与等价值
阶梯式——优化编号时更为方便。参考下图。

步骤1：分析需求，划分等价类

A) 有效等价类：

- 1) -99—99 的整数

B) 无效等价类

- 2) 不填（为空）
- 3) 非整数
- 4) >99的整数
- 5) <-99的整数

步骤2：细化等价类

依据的不再是字面上的需求，而是根据数据在内存或者数据库中存储的类型、格式等

- 非整数细化：
 - ✓ 小数
 - ✓ 字母
 - ✓ 汉字
 - ✓ 符号
- 99—99（有效等价类）
 - 正数和负数单独测试

步骤3：建立等价类表

有效等价类	
编号	数据要求
1	-99-(-1)的整数
2	0-99的整数

无效等价类	
编号	数据要求
3	为空
4	小数
5	字母
6	汉字
7	符号
8	>99的整数
9	<-99的整数

步骤4：编写测试用例

从每个等价类中至少选一个代表数据进行测试。

用例编号	测试目的	测试步骤	预期结果
1	在“第二个数”正确的情况下，“第一个数”为-99-(-1)的整数，可以正确计算	步骤： 1、在“第一个数”文本框中输入：-52 2、在“第二个数”文本框中输入：13 3、点击“计算”按钮	步骤3：在“结果”文本框中显示：-39

控件名称	数据要求	有效等价类	有效边界值	无效等价类	无效边界值	所属用例				
姓名	3~10个字符，不能为空	3~10个字符	3位字符 10位字符 4位字符 9位字符	空 小于3位字符 大于10位字符	5001	14				
工资	1.00~30000.00之间的小数（或整数），可以为空，小数点后最多保留2位	1.00~30000.00之间的小数（或整数）	1.00 30000.00 1.01 29999.99 空	小于1.00的小数（或整数） 大于30000.00的小数（或整数） 特殊字符 字母 汉字	2位字符 11位字符	16-14.15-19.28-29 2 3 4 5 20 21 22 23 24 25 26 27				

*白盒测试的各种覆盖测试用例

基本路径法:

对程序，从开始分析，每个判断语句进行一次分叉，每个片段仅取其末尾。如下的程序片段：

```

1 void CaculateTeacherSalary()
2 {
3     int j;
4     int j=0;
5     printf("输入要计算的软件学院教师编号: \n");
6     fflush(stdin);
7     scanf("%d",&num);
8     for(j=0;j<MAXNUM;j++)
9     {
10        if(Teacher[j].TeacherNo==num) //确定是否为输入的教师号
11        {
12            j=1; //先赋值，在后面让j同0比较
13            printf("输入保险金额:");
14            fflush(stdin);
15            scanf("%f",&baoxianjin);
16            printf("输入月效益:");
17            fflush(stdin);
18            scanf("%f",&xiaoyi);
19            TeacherSalary[j]=(Teacher[j].TeacherBaseSalary+2*Teacher[j].TeacherMonthWorkDays*xiaoyi
20            printf("%04d 号软件学院教师的薪水为:%lf 元每月\n",
21            Teacher[j].TeacherNo,TeacherSalary[j]);
22            break; //找到该教师后，直接跳出循环
23        }
24        if(j==0)
25            printf("未找到! \n");
26    }

```

如左图，开始即开始，直接写到第 8 行，判断语句，则画棱形，数字 8，若符合判断，则写到第 10 行判断语句，随后一直直接写到第 21 行结束。若不是判断语句则以方块画图。

环路复杂度计算如下：

$V(G)=E-N+2$ ，E 为边数，N 为节点数

$V(G)=P+1$ ，P 为判定节点数

5 种逻辑覆盖...

基于给出的例子来写出能够充分覆盖的用例

① 语句覆盖

- 每条可执行语句（“；”结尾）执行一次

被测代码

```

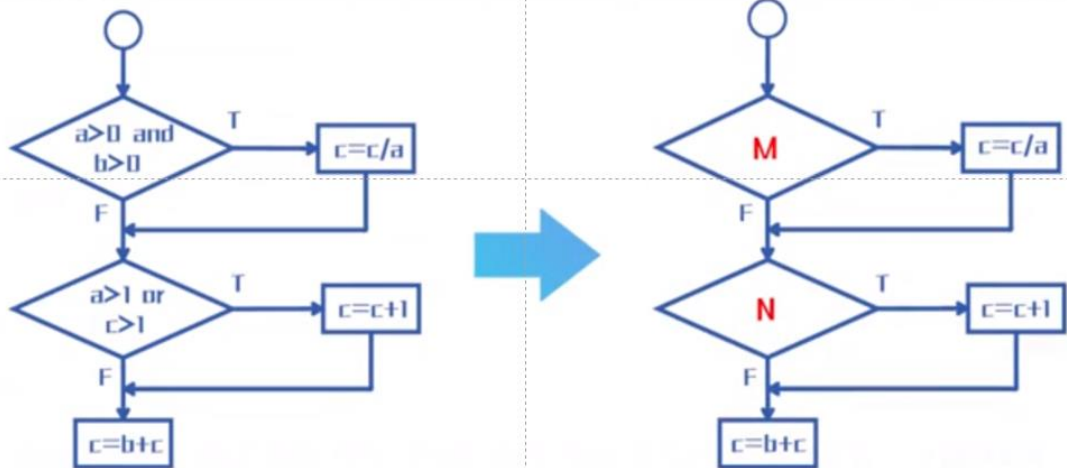
1. int a, b;
2. duple c;
3. if (a>0 OR b>0)
4. c = c/a;
5. if (a>1 AND b>1)
6. c = c+1;
7. c = b+c;

```

a=2,b=1,c=6,expected=5

② 判定覆盖

- 每条真假分支执行一次



$a=2, b=1, c=6$, $\text{expected}=5$

$a=-2, b=1, c=-6$, $\text{expected}=-5$

③ 条件覆盖

- 判定中每个条件真假值执行一次

判定条件M:

条件 $a > 0$: 取T时为T1, 取F时为F1;

条件 $b > 0$: 取T时为T2, 取F时为F2;

$a=2, b=-1, c=-2$, $\text{expected}=-2$

取值满足: $a > 0, b \leq 0, a > 1, c \leq 1$

条件取值: T1, F2, T3, F4

判定条件N:

条件 $a > 1$: 取T时为T3, 取F时为F3;

条件 $c > 1$: 取T时为T4, 取F时为F4。

$a=-1, b=2, c=3$, $\text{expected}=6$

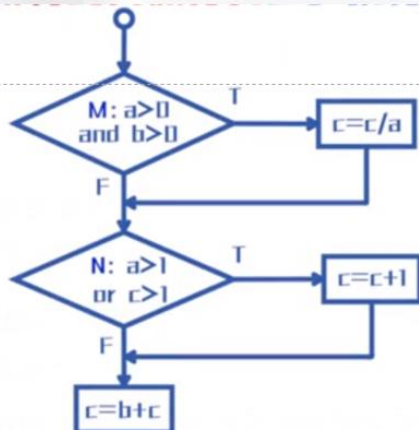
取值满足: $a \leq 0, b > 0, a \leq 1, c > 1$

条件取值: F1, T2, F3, T4

④ 判定/条件覆盖

- 判定中每个条件真值至少取一次

真假分支都执行一次



$a=2, b=1, c=6, \text{expected}=5$

取值满足: $a>0, b>0, a>1, c>1$

判定取值: $M=T, N=T$

条件取值: $T1, T2, T3, T4$

$a=-1, b=-2, c=-3, \text{expected}=-5$

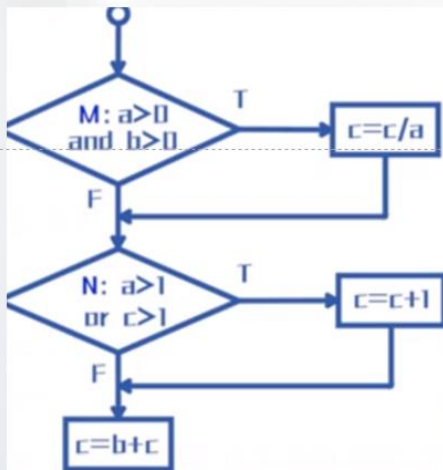
取值满足: $a\leq 0, b\leq 0, a\leq 1, c\leq 1$

判定取值: $M=F, N=F$

条件取值: $F1, F2, F3, F4$

⑤ 条件组合覆盖

判定中每个条件真值的所有组合至少取一次



1、 $\{T1, T2\}$

2、 $\{T1, F2\}$

3、 $\{F1, T2\}$

4、 $\{F1, F2\}$

5、 $\{T3, T4\}$

6、 $\{T3, F4\}$

7、 $\{F3, T4\}$

8、 $\{F3, F4\}$

$a=2, b=1, c=6, \text{expected}=5$

条件取值: $T1, T2, T3, T4$

覆盖组合: $\{T1, T2\}, \{T3, T4\}$

$a=2, b=-1, c=-2, \text{expected}=-2$

条件取值: $T1, F2, T3, F4$

覆盖组合: $\{T1, F2\}, \{T3, F4\}$

$a=-1, b=2, c=3, \text{expected}=6$

条件取值: $F1, T2, F3, T4$

覆盖组合: $\{F1, T2\}, \{F3, T4\}$

$a=-1, b=-2, c=-3, \text{expected}=-5$

条件取值: $F1, F2, F3, F4$

覆盖组合: $\{F1, F2\}, \{F3, F4\}$