

本题中给定一个方法，不同的子类行为不同，这是多态机制。

在计算机系统中，系统的(49)可以用 $MTTF / (1 + MTTF)$ 来度量，其中 MTTF 为平均无故障时间。

- (49) A. 可靠性 B. 可用性 C. 可维护性 D. 健壮性

【答案】A

【解析】

MTTF 是用来描述可靠性的指标。

修改现有软件系统的设计文档和代码以增强可读性，这种行为属于(50)维护。

- (50) A. 正确性 B. 适应性 C. 完善性 D. 预防性

【答案】D

【解析】

改正性维护：是指在使用过程中发现了隐蔽的错误后，为了诊断和改正这些隐蔽错误而修改软件的活动

适应性维护：是指为了适用变化了的环境而修改软件的活动

完善性维护：是指为了扩充或完善原有软件的功能或性能而修改软件的活动

预防性维护：是指为了提高软件的可维护性和可靠性、为未来的进一步改进打下基础而修改软件的活动

题干中修改文档和代码提高可读性，提高可读性利于提高可维护性，所以应该属于预防性维护。

以下不属于系统测试范畴的是(51)。

- (51) A. 单元测试 B. 安全测试 C. 强度测试 D. 性能测试

【答案】A

【解析】

单元测试是对程序模块进行的测试，不属于系统测试的范畴。

以下关于文档测试的说法中，不正确的是(52)。

- (52) A. 文档测试需要仔细阅读文档，检查每个图形

- B. 文档测试需要检查文档内容是否正确和完善
- C. 文档测试需要检查标记是否正确性
- D. 文档测试需要确保大部分示例经过测试

【答案】D

【解析】

文档中的示例应像用户一样载入和使用样例。如果是一段程序，就输入数据并执行它以每一个模板制作文件，确认它们的正确性。

软件测试的对象不包括_(53)。

- (53) A. 软件代码 B. 软件文档 C. 质量保证方法 D. 相关数据

【答案】C

【解析】

软件测试的对象为：程序、数据和文档。

测试用例的三要素不包括_(54)。

- (54) A. 输入 B. 预期输出 C. 执行条件 D. 实际输出

【答案】D

【解析】

IEEE 829 标准中列出测试用例应该包含的重要信息：标识符、测试项、输入说明、输出说明、环境要求、特殊过程要求、用例之间的依赖性。结合本题来看应该选择选项 D。

以下关于软件测试原则的叙述中，正确的是_(55)。

- ①测试开始得越早，越有利于发现缺陷
- ②测试覆盖率和测试用例数量成正比
- ③测试用例既需选用合理的输入数据，又需要选择不合理的输入数据
- ④应制定测试计划并严格执行，排除随意性
- ⑤采用合适的测试方法，可以做到穷举测试
- ⑥程序员应尽量测试自己的程序

- (55) A. ①②③④⑤⑥ B. ①②③④⑤ C. ①②③④ D. ①③④

【答案】D

【解析】

软件测试原则：所有的测试都应追溯到用户需求；应尽早并不断地进行测试；测试工作应避免由原开发软件的人或小组来承担（单元测试除外）；穷举测试是不可能的，测试需要终止；充分重视测试中的群集现象；严格按照测试计划来进行，避免随意性。

以下关于测试时机的叙述中，正确的是(56)。

- ①应该尽可能早地进行测试
- ②软件中的错误暴露得越迟，则修复和改正错误所花费的代价就越高
- ③应该在代码编写完成后开始测试
- ④项目需求分析和设计阶段不需要测试人员参与

(56) A. ①②③④ B. ①②③ C. ①② D. ①

【答案】C

【解析】

③④违背了测试应尽早开始的原则。

以下属于软件测试工具的是(57)。

- ①JTest ②LoadRunner
- ③Visual Studio ④JBuilder

(57) A. ①②③④ B. ①②③ C. ①② D. ①

【答案】C

【解析】

①Java 单元测试工具 JTest；②是一种预测系统行为和性能的负载测试工具；③④开发相关的工具

兼容性测试不包括(58)。

- (58) A. 软件兼容性测试 B. 硬件兼容性测试
- C. 数据兼容性测试 D. 操作人员兼容性测试

【答案】D

【解析】

兼容性测试包括有软件、硬件、数据、平台等兼容性测试。

根据输出对输入的依赖关系设计测试用例的黑盒测试方法是(59)。

- (59) A. 等价类划分法 B. 因果图法 C. 边界值分析法 D. 场景法

【答案】B

【解析】

因果图法：是从自然语言书写的程序规格说明的描述中找出因（输入条件）和果（输出或程序状态的改变），通过因果图转换为判断表。

以下关于边界值测试法的叙述中，不正确的是(60)。

- (60) A. 边界值分析法仅需考虑输入域边界，不用考虑输出域边界
B. 边界值分析法是对等价类划分方法的补充
C. 错误更容易发生在输入输出边界上而不是输入输出范围的内部
D. 测试数据应尽可能选取边界上的值

【答案】A

【解析】

边界值分析法需要考虑输入域的边界和输出域的边界。

一个程序的控制流图中有 6 个节点，10 条边，在测试用例数最少的情况下，确保程序中每个可执行语句至少执行一次所需要的测试用例数的上限是(61)。

- (61) A. 2 B. 4 C. 6 D. 8

【答案】C

【解析】

$V(g) = \text{边} - \text{节点} + 1 = 10 - 6 + 1 = 5$ ； $V(g)$ 表示实现基本路径覆盖测试用例的最大数量。

对于逻辑表达式 $(b1 \& b2) || in$ ，需要(62)个测试用例才能完成条件组合覆盖。

- (62) A. 2 B. 4 C. 8 D. 26

【答案】B

【解析】

多条件覆盖（MCC）：也称条件组合覆盖，设计足够的测试用例，使得使得每个判定中条件的各种可能组合都至少出现一次

本题中&被认为是位运算，则 MCC 的测试用例数应为：22=4

如果作为逻辑与运算，则有 3 个条件，符合 MCC 的测试用例数应为：23=8

测试执行过程的阶段不包括 (63)。

- (63) A. 初测期 B. 系统测试期 C. 细测期 D. 回归测试期

【答案】B

【解析】

测试执行过程的阶段为：初测期、细测期、和回归测试期。

初测期——测试主要功能和关键的执行路径，排除主要障碍。

细测期——依据测试计划和测试用例，逐一测试大大小小的功能、方方面面的特性、性能、用户界面、兼容性、可用性等等；预期可发现大量不同性质、不同严重程度的错误和问题。

回归测试期——系统已达到稳定，在一轮测试中发现的错误已十分有限；复查已知错误的纠正情况，未引发任何新的错误时，终结回归测试。

以下关于回归测试的叙述中，不正确的是 (64)。

- (64) A. 回归测试是为了确保改动不会带来不可预料的后果或错误
B. 回归测试需要针对修改过的软件成分进行测试
C. 回归测试需要能够测试软件的所有功能的代表性测试用例
D. 回归测试不容易实现自动化

【答案】A

【解析】

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

以下属于测试停止依据的是 (65)。

- ①测试用例全部执行结束 ②测试覆盖率达到要求
③测试超出了预定时间 ④查出了预定数目的故障
⑤执行了预定的测试方案 ⑥测试时间不足

- (65) A. ①②③④⑤⑥ B. ①②③④⑤ C. ①②③④ D. ①②③

【答案】B

【解析】

⑥测试时间不足不应作为测试停止的依据

以下关于测试方法的叙述中，不正确的是(66)。

- (66) A. 根据被测代码是否可见分为白盒测试和黑盒测试
- B. 黑盒测试一般用来确认软件功能的正确性和可操作性
- C. 静态测试主要是对软件的编程格式 M 结构等方面进行评估
- D. 动态测试不需要实际执行程序

【答案】D

【解析】

动态测试是指通过人工或使用工具运行程序进行检查、分析程序的执行状态和程序的外部表现。所以选项 D 是错误的描述。

以下关于性能测试的叙述中，不正确的是(67)。

- (67) A. 性能测试的目的是为了验证软件系统是否能够达到用户提出的性能指标
- B. 性能测试不用于发现软件系统中存在的性能瓶颈
- C. 性能测试类型包括负载测试，强度测试，容量测试等
- D. 性能测试常通过工具来模拟大量用户操作，增加系统负载

【答案】B

【解析】

性能测试用来保证产品发布后系统的性能能够满足用户需求。性能测试通常存在性能调优与性能评测两种性能测试策略。

性能评测主要包括：在真实环境下，检查系统服务等级的满足情况，评估并报告整个系统的性能；对系统的未来容量作出预测和规划。

性能评测是性能调优的基础，性能调优的步骤如下：查找形成系统瓶颈或者故障的根本原因；进行性能调整和优化；评估性能调整的效果。

不同加密机制或算法的用途、强度是不相同的，一个软件或系统中的加密机制使用是否合理，强度是否满足当前要求，需要通过测试来完成，通常(68)是测试的一个重要手段。

(68) A. 模拟加密 B. 模拟解密 C. 漏洞扫描 D. 算法强度理论分析

【答案】B

【解析】

加密机制是保护数据安全的重要手段，加密的基本过程就是对原来为明文的文件或数据，按某种算法进行处理，使其成为不可读的密文。由于不同加密机制的用途及强度不同，因此一个信息系统中加密机制使用是否合理，强度是否满足当前需要，需要通过测试来检验，通常模拟解密是测试的一个重要手段。

安全日志是软件产品的一种被动防范措施，是系统重要的安全功能，因此安全日志测试是软件系统安全性测试的重要内容，下列不属于安全日志测试基本测试内容的是 (69)。

(69) A. 对安全日志的完整性进行测试，测试安全日志中是否记录包括用户登录名称、时间、地址、数据操作行为以及退出时间等全部内容

B. 对安全日志的正确性进行测试，测试安全日志中记录的用户登录、数据操作等日志信息是否正确

C. 对日志信息的保密性进行测试，测试安全日志中的日志信息是否加密存储，加密强度是否充分

D. 对于大型应用软件系统，测试系统是否提供安全日志的统计分析能力

【答案】C

【解析】

日志应当记录所有用户访问系统的操作内容，包括登录用户名称、登录时间、浏览数据动作、修改数据动作、删除数据动作、退出时间、登录机器的 IP 等。

安全日志测试：根据业主要求或设计需求，对日志的完整性、正确性进行测试；测试安全日志是否包含了日志内容的全部项目，是否正确；对于大型应用软件，系统是否提供了安全日志的智能统计分析能；是否可以按照各种特征项进行日志统计，分析潜在的安全隐患，及时发现非法行为。

下到关于 DoS 攻击的描述中，错误的是 (70)。

(70) A. oS 攻击通常通过抑制所有或流向某一特定目的端的消息，从而使系统某一实体不能执行其正常功能，产生服务拒绝

B. DoS 攻击不需入目标系统，仅从外部就可实现攻击

C. 只要软件系统内部没有漏洞, DoS 攻击就不可能成功

D. 死亡之 Ping、Land 攻击、UDP 洪水、Smurf 攻击均是常见的 DoS 攻击手段

【答案】C

【解析】

服务拒绝: 当一个实体不能执行它的正常功能, 或它的动作妨碍了别的实体执行它们的正常功能的时候, 便发生服务拒绝。这种攻击不一定需要目标系统存在漏洞, 如 UDP 洪水。

Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a (71), open or closed. In this respect software systems differ profoundly from computers, buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more (72) than computers do. Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some (73) fashion, and the complexity of the whole increases much more than linearly.

The complexity of software is a(an) (74) property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence. Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving, properties from the models, and verifying those properties experimentally. This worked because the complexities (75) in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

(71) A. task

B. job

C. subroutine

D. program

- | | | | |
|-----------------|--------------|---------------|----------------|
| (72) A. states | B. parts | C. conditions | D. expressions |
| (73) A. linear | B. nonlinear | C. parallel | D. additive |
| (74) A. surface | B. outside | C. exterior | D. essential |
| (75) A. fixed | B. included | C. ignored | D. tabilized |

【答案】C A B D C

【解析】

软件实体规模上或许比其他任何人类创造的结构要更为复杂,因为没有两个部分是一样的(至少在语句级以上)。如果他们一样,我们便将这两个相似的部分合到一起,成为一个子程序,打开或关闭。在这一点上,软件系统与计算机、建筑或汽车经常混合使用一些重复的元件有很大区别。

数字计算机本身比人类所建造的大多数事物都要复杂,他们有超级多的状态。这使得对他们进行想象、描述和测试都很困难。软件系统的状态数目更是比计算机的超出几个数量级。同理,软件实体的扩展不单是这一批元件变大一点,它必将是大量不同元件都有增加。在大多数情况下,构件以非线性的方式相互作用,而整体的复杂性远超线性增加。

软件的复杂的特点具有必然性,并非偶然。因此,对于软件实体的描述,剥离了它的复杂性往往就等于剥离了它的本质。过去这三个世纪,通过对复杂的现象构建出简化模型,从模型的属性再倒推,并通过实验验证这些属性,数学和物理科学取得了长足的进步。这之所以行之有效,是因为模型中忽略掉的复杂特性并不是现象中重要的本质属性。而当这些复杂性很重要时,这种方法就会失效。

试题一

阅读下列 C 程序，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【C 程序】

```
int count(int x,int z){  
    int y=0;  
    while(x>0){ //1  
        if(x==1) //2  
            y=7; //3  
        else{ //4  
            y=x+z+4;  
            if (y==7||y==21) //5, 6  
                x=1; //7  
        }  
        x--; //8  
    }  
    return y; //9  
}
```

【问题 1】(3 分)

请针对上述 C 程序给出满足 100%DC（判定覆盖）所需的逻辑条件。

$x > 0$; $x \leq 0$

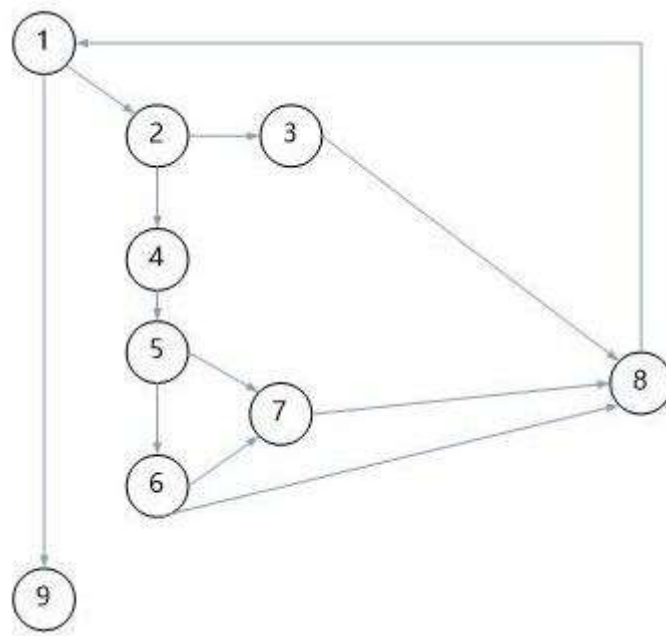
$x == 1$; $x \neq 1$

$y == 7$ 或者 $y == 21$; $y \neq 7$ 且 $y \neq 21$

解析：本题中的判定有 $x > 0$; $x == 1$; $y == 7 || y == 21$ ；三个判定点，所以要符合 100%判定覆盖的要求，就需要使得每个判定结果的真假值都出现 1 次，即： $x > 0$; $x \leq 0$; $x == 1$; $x \neq 1$;
 $y == 7$ 或者 $y == 21$; $y \neq 7$ 且 $y \neq 21$

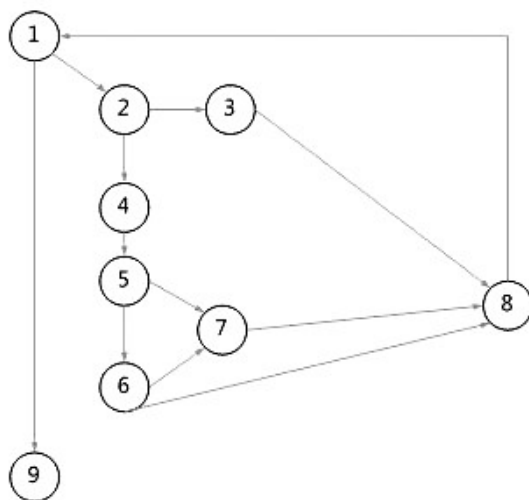
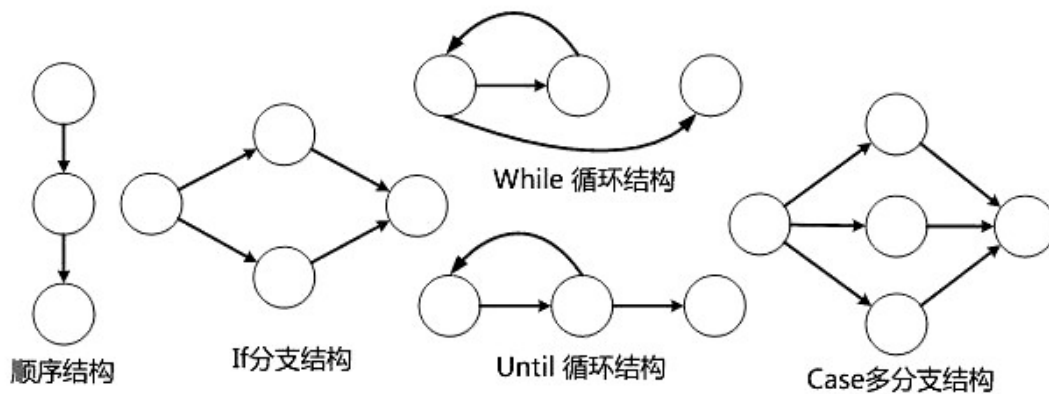
【问题 2】(7 分)

请画出上述程序的控制流图，并计算其控制流图的环路复杂度 $V(G)$ 。



$$V(g)=5$$

解析：控制流图是描述程序控制流的一种图示方法。其基本符号有圆圈和箭线：圆圈为控制流图中的一个结点，表示一个或多个无分支的语句；带箭头的线段称为边或连接，表示控制流。基本结构如下所示：



控制流程图的环路复杂性 $V(G)$ 等于：

- (1) 控制流程图中的区域个数。
- (2) 边数-结点数+2。
- (3) 判定数+1。

$V(g)=5$

【问题 3】(5 分)

请给出问题 2 中控制流图的线性无关路径。

线性无关路径是指包括一组以前没有处理的语句或条件的一条路径。从控制流图来看，一条线性无关路径是至少包含有一条在其他线性无关路径中从未有过的边的路径：

1: 1, 9

2: 1, 2, 3, 8, 1, 9

3: 1, 2, 4, 5, 7, 8, 1, 9

4: 1, 2, 4, 5, 6, 7, 8, 1, 9

5: 1, 2, 4, 5, 6, 8, 1, 9

试题二

阅读下列说明，回答问题 1 和问题 2，将解答填入答题纸的对应栏内。

【说明】

某航空公司的会员卡分为普卡、银卡、金卡和白金卡 4 个级别，会员每次搭乘该航空公司航班均可能获得积分，积分规则如表 2-1 所示。此外，银卡及以上级别会员有额外积分奖励，奖励规则如表 2-2 所示。

舱位	舱位代码	积分
头等舱	F	200%*K
	Z	150% *K
	A	125%*K+
公务舱	C	150%*K
	D/I	125%*K
	R	100%*K
经济舱	Y	125%*K
	B/H/K/L/m/V	100%*K
	Q/X/U/E	50%*k
	P/S/G/O/JN[FW	O

表 2-2 额外积分奖励规则

会员级别	普卡	银卡	金卡	白金卡
级别代码	F	S	G	p
额外积分奖励	0%	10%	25%	50%

公司开发了一个程序来计算会员每次搭乘航班历累积的积分，程序的输入包括会员级别

B、舱位代码 C 和飞行公里数 K，程序的输出为本次积分 S 。其中，B 和 C 三字母且大小写不敏感，K 为正整数，S 为整数（小数部分四舍五入）。

【问题 1】（7 分）

采用等价类型划分法对该程序进行测试，等价类表如 2-3 所示，请补充空（1）-（7）

表 2-3 等价类表

输入条件	有效等价类	编号	无效等价类	编号
会员等级 B	F	1	非字母	12
	S	2	非单个字母	13
	G	3	(5)	14
	(1)	4		
舱位代码 C	F	5	非字母	15
	(2)	6	(6)	16
	(3)	7		
	R/B/H/K/L/M/W	8		
	Q/X/U/E	9		
	P/S/G/O/I/V/N/T	10		
飞行公里数 K	(4)	11	非整数	17
			(7)	18

1： P 2： Z/C 3： A/D/I /Y 4： 正整数

5： 非 FSGP 的字母 6： 非规定仓位的字母 7： 非正数

解析：等价类划分原则：

在输入条件规定了取值范围或值的个数的情况下，可以确定一个有效等价类和两个无效等价类

在输入条件规定了输入值得集合或者规定了“必须如何”的条件的情况下，可以确立一个有效等价类和一个无效等价类

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

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，可确定 n 个有效等价类和一个无效等价类

在规定了输入数据必须遵守的规则的情况下，可确定一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）

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地划分为更小的等价类

本题中会员等级的有效等价类应为：表格中指定的类：FSGP，由于每个等级是单独处理，

【问题 2】（13 分）

表 2-4

编号	输入			覆盖等价类 (编号)	预期输出 S
	B	C	K		
1	F	F	500	1,5,11	(1)
2	S	Z	(2)	2,6,11	825
3	G	A	500	(3)	781
4	P	(4)	500	4,8,11	750
5	(5)	Q	500	1,9,11	250
6	F	P	500	1,10,11	(6)
7	(7)	P	500	12,10,11	N/A
8	(8)	F	500	13,5,11	N/A
9	A	Z	500	14,6,11	N/A
10	S	(9)	500	2,15,11	N/A
11	S	(10)	500	2,16,11	N/A
12	S	Q	(11)	2,9,17	(12)
13	S	P	(13)	2,10,18	N/A

(无有效等价类，一个测试用例只能覆盖 1 个有效类)

试题三

阅读下列说明，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某证券交易所为了方便提供证券交易服务，欲开发一个基于 Web 的证券交易平台。其主要功能包括客户开户，记录查询、存取款、股票交易等。客户信息包括姓名、Email（必填且唯一）、地址等；股票交易信息包括股票代码（6 位数字编码的字符串）、交易数量（100 的整数倍）、买/卖价格（单位：元，精确到分）。

系统要支持：

(1) 在特定时朝内 3000 个用户并发时，主要功能的处理能力至少要达到 128 个请求/秒，平均数据量 2KB/请求；

(2) 页面中采用表单实现客户信息、交易信息等的提交与交互，系统前端采用 HTML5 实现。

【问题 1】（4 分）

在对此平台进行非功能测试时，需要测试哪些方面？

性能测试、安全性测试、兼容性测试、易用性测试

【问题 2】（5 分）

在满足系统要支持的(1)时，计算系统的通信吞吐量。

$3000 \times 128 \times 2KB = 750MB$

【问题 3】

每个字段的验证

字段的缺省值

表单中的输入

【问题 4】（8 分）

(1) 针对股票代码：111111、数量：10 万、当前价格：6.00，设计 4 个股票交易的测试输入。

(2) 设计 2 个客户开户的测试输入，以测试是否存在 XSS、SQL 注入。

1: 111111, 10 万, 6（正确输入）

2: 1222, 10 万, 6（代码错误）

3: 111111, 0 万, 6 (数量错误)

4: 111111, 10 万, 0 (价格错误)

21: 姓名: 张三 or 1=1-- Email: q123@q123.com 地址: xxxxxxx

22: 姓名: 张三 Email: q123@q123.com 地址: xxxxxxx<script>alert('测试信息')</script>

解析: (1) 本题中有 3 条件, 应该设计一个测试用例覆盖 3 个有效等价类; 然后针对每个条件的无效等价类各设计一个测试用例. (2) 针对客户信息包括姓名、Email (必填且唯一)、地址, 等输入, 任意挑选一个输入文本框, 在保障 SQL 能运行的前提下, 输入带有注入式攻击和 XSS 攻击的特征内容即可。

试题四

阅读下列说明，回答问题 1 至问题 5，将解答写在答题纸的对应栏内。

【说明】

图 4-1 是银行卡应用的部分类图，图中属性和操作前的“+”和“-”分别表示公有成员和私有成员。银行卡 Account 有两种类型，借记卡 SavingAccount 和信用卡 CreditAccount。

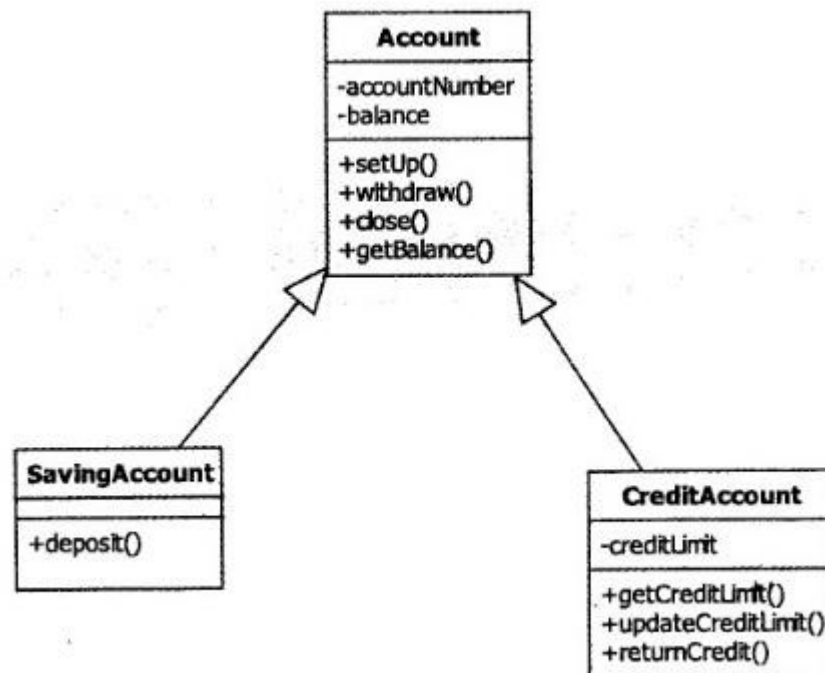


图 4-1 银行卡应用的部分类图

(1) 借记卡和信用卡都有卡号 `account Number` 和余额 `balance` 两个属性。借记卡的余额是正常余额，信用卡的余额是目前未还的金额，如果没有未还的金额，则为 0；有开户 `set Up`、取款 `withdraw`、查询余额 `get Balance` 和销户 `close` 四个方法。借记卡取钱时，要求取钱金额不能超过余额；而信用卡取钱金额不能超过信用额度，因此需要在子类中实现该方法。

(2) 借记卡可以存钱 `deposit`。

(3) 信用卡有信用额度 `creditLimit` 属性，可以查询信用额度 `getCreditLimit`、修改信用额度 `updateCreditLimit` 和还款 `returnCredit`。

现拟采用面向对象的方法进行测试。

【问题 1】(3 分)

面向对象单元测试的主要对象是什么？

是针对程序的函数、过程或完成某以特定功能的程序块。

【问题 2】(4 分)

在继承关系上，若某方法在测试父类时已经测试过，那么在什么情况下在子类中也需要测试？

1: 继承类的成员函数在子类中做了改动

2: 成员函数调用了改动过的成员函数的部分

【问题 3】(4 分)

要测试方法 `deposit()` 时，还需要调用什么方法？给出测试序列。

`getBalance()`

先测试 `getBalance()` 再测试 `deposit()`

【问题 4】(6 分)

方法 `withdraw` 在基类 `Account` 中定义，但在两个子类中有不同的实现。这是面向对象的什么机制？这种情况在测试时如何进行？

多态

只需要在原有的测试分析和基础上增加对测试用例中输入数据的类型的考虑；先测试基类，然后再分别依据输入数据设计不同的测试用例。

【问题 5】(3 分)

给出类 `SavingAccount` 的最小测试序列。

先测试 `Account` 类，然后测试 `SavingAccount` 类

试题五

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某嵌入式控制软件通过采集传感器数值来计算输出控制率，为了提高数据采集的可靠性，使用三余度采集方法进行三个通道的数据采集。

1. 三余度通道数据采集及处理要求：

(1) 三通道采集值 $In_U[0]$ 、 $In_U[1]$ 、 $In_U[2]$ 的正常范围为 $[-3.0, 3.0]V$ ，且任意两通道间差值不大于 $0.5V$ ；

(2) 如果某通道采集值超过正常范围或者因为通道采集值造成与其它通道差值大于 $0.5V$ ，则该通道数据不满足要求；

(3) 如果三通道值均满足要求，则取三通道中差值较小的两通道数据的平均值；

(4) 如果仅有一个通道数据不满足采集要求，取满足要求的两个通道数据的平均值；

(5) 如果多于一个通道数据不满足采集要求，取安全值 $0V$ 。

2. 对采集数值计算控制率的具体处理算法如下：

(1) 如果依据采集数据计算的控制率 $C1$ 与目前实际控制率 $C0$ 差值不大于 0.01 则使用本周期计算控制率 $C1$ 进行输出控制，否则使用目前实际控制率 $C0$ 输出控制。不上报传感器故障；

(2) 如果连续 3 个周期计算的控制率 $C1$ 与目前实际控制率 $C0$ 差值大于 0.01 ，则上报传感器三级故障，连续超差计数清零，使用目前实际控制率 $C0$ 输出控制；如果已经连续 3 个周期控制率超差，并上报三级故障，但第 4 个周期计算的控制率 $C1$ 与目前实际控制率 $C0$ 差值不大于 0.01 ，则清除三级故障上报；

(3) 如果累计大于等于 10 个周期计算的控制率 $C1$ 与目前实际控制率 $C0$ 差值大于 0.01 ，则上报传感器二级故障，使用目前实际控制率输出控制；

(4) 如果累计大于等于 100 个周期计算的控制率 $C1$ 与目前实际控制率 $C0$ 差大于 0.01 ，则上报传感器一级故障，并清除二级故障，并切断输出控制（输出安全值 0 ）

(5) 如果低级故障和高级故障同时发生，则按高级故障处理。

【问题 1】（9 分）

为了测试采集算法，在不考虑测量误差的情况下，设计了表 5-1 所示的测试用例，请填写该表中的空 (1) ~ (6)。

表 5-1 测试用例

序号	输 入			输出 Out_A1
	In_U[0]	In_U[1]	In_U[2]	预期输出（保留两位小数）
1	0.0V	0.0V	0.0V	0.00V
2	2.0V	2.3V	1.8V	(1)
3	1.5V	1.6V	1.3V	(2)
4	2.8V	2.6V	2.0V	(3)
5	-3.0V	-3.1V	-2.8V	(4)
6	2.0V	1.4V	2.6V	(5)
7	3.1V	2.8V	3.2V	(6)

1: 1.9 2: 1.55 3: 2.7

4: -2.90 5: 0 6: 0

解析：（1）信道之差不大于 0.5，所以三个信道都满足要求，按规则（3）处理，即取 1.8 和 2.0 的平均值

（2）信道之差不大于 0.5，所以三个信道都满足要求，按规则（3）处理，即取 1.5 和 1.6 的平均值

（3）信道之差 2.8 与 2.0 直接大于 0.5，2.6 与 2.0 之间大于 0.5，所以 2.0 不符合要求，按规则（4）处理，即取 2.6 和 2.8 的平均值

（4）信道之差不大于 0.5，-3.1 超过【-3.0，3.0】的范围，不满足要求，按规则（4）处理，即取-3.0 和-2.8 的平均值

（5）由于任意两个信道之间的差值都超过 0.5，所以所有信道都不符合要求，按规则（5）处理，即取值为 0

（6）由于 3.1 和 3.2 都超过【-3.0，3.0】的范围，按规则（5）处理，即取值为 0

【问题 2】（9 分）

为了测试控制率计算算法，在不考虑测量误差的情况下，设计了表 5-2 所示的用例，请完善其中的空（1）～（6）。

表 5-2 测试用例

序号	前置条件		输 入		输出（预期结果）	
	控制率超差 连续计数	控制率超差 累计计数	计算控制率 C_1	实际控制率 C_0	输出控制率	上报故障
1	0	0	1.632	1.638	1.632	无
2	0	0	1.465	1.454	(1)	无
3	(2)	6	2.358	2.369	2.369	三级故障
4	1	(3)	1.569	1.557	1.557	二级故障
5	2	9	2.221	2.234	2.234	(4)
6	0	99	1.835	1.822	(5)	一级故障
7	2	99	2.346	2.357	0	(6)

1: 1.454 2: 2 3: 9

4: 二级故障 5: 0 6: 一级故障

解析: (1) 按照规则 1, $1.465 - 1.454 = 0.011$ 大于 0.01, 所以应该为实际控制率

(2) 是三级故障, 三级故障要求连续 3 个周期, 所以前置条件应该为 2

(3) 是二级故障, 二级故障要求累计 10 个周期, 所以前置条件应该为 9

(4) 前置累计次数已经达到 9, 在加上本次, 就连续次数达到 10, 所以应该为二级故障

(5) 是一级故障, 一级故障的输出值为 0

(6) 由于输出值为 0, 应属于一级故障

【问题 3】(2 分)

测试人员在设计测试用例进行采集算法测试时, 发现本项目的三余度采集值的具体处理算法存在 1 处缺陷, 请指出此处缺陷。

缺陷: 当三个通道的数据只差不超过 0.5, 而且两两之间的差值有两个相等时, 存在取值问题。如: 测试数据为: 2.0 2.2 2.4 这时, 存在取值问题, 是取 2.1 还是 2.3?