

# 2018 年下半年

## 全国计算机技术与软件专业技术资格考试 2018 年下半年 软件设计师 上午试卷

(考试时间 9:00~11:30 共 150 分钟)

请按下述要求正确填写答题卡

1. 在答题卡的指定位置上正确写入你的姓名和准考证号,并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。
2. 本试卷的试题中共有 75 个空格,需要全部解答,每个空格 1 分,满分 75 分。
3. 每个空格对应一个序号,有 A、B、C、D 四个选项,请选择一个最恰当的选项作为解答,在答题卡相应序号下填涂该选项。
4. 解答前务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项,如需修改,请用橡皮擦干净,否则会导致不能正确评分。

### 例题

- 2018 年下半年全国计算机技术与软件专业技术资格考试日期是\_\_(88)\_\_月\_\_(89)\_\_日。

(88) A. 10      B. 11      C. 12      D. 09

(89) A. 07      B. 08      C. 09      D. 10

因为考试日期是“11 月 09 日”,故(88)选 B,(89)选 C,应在答题卡序号 88 下对 B 填涂,在序号 89 下对 C 填涂。

- CPU 在执行指令的过程中, 会自动修改\_(1)\_, 以使其保存的总是将要执行的下一条指令的地址。

(1) A. 指令寄存器 B. 程序计数器  
C. 地址寄存器 D. 指令译码器

- 在微机系统中, BIOS (基本输入输出系统) 保存在\_(2)中。

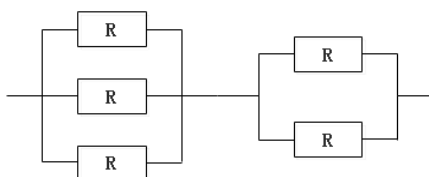
(2) A. 主板上的 ROM B. CPU 的寄存器  
C. 主板上的 RAM D. 虚拟存储器

- 采用  $n$  位补码 (包含一个符号位) 表示数据, 可以直接表示数值\_(3)。

(3) A.  $2^n$  B.  $-2^n$  C.  $2^{n-1}$  D.  $-2^{n-1}$

- 某系统由下图所示的部件构成, 每个部件的千小时可靠度都为  $R$ , 该系统的千小时可靠度为\_(4)。

(4) A.  $(3R+2R)/2$  B.  $R/3+R/2$   
C.  $[1-(1-R)^3][1-(1-R)^2]$  D.  $1-(1-R)^3-(1-R)^2$



- 以下关于采用一位奇校验方法的叙述中, 正确的是\_(5)。

(5) A. 若所有奇数位出错, 则可以检测出该错误但无法纠正错误  
B. 若所有偶数位出错, 则可以检测出该错误并加以纠正  
C. 若有奇数个数据位出错, 则可以检测出该错误但无法纠正错误  
D. 若有偶数个数据位出错, 则可以检测出该错误并加以纠正

- 下列关于流水线方式执行指令的叙述中, 不正确的是\_(6)。

(6) A. 流水线方式可提高单条指令的执行速度  
B. 流水线方式下可同时执行多条指令  
C. 流水线方式提高了各部件的利用率  
D. 流水线方式提高了系统的吞吐率

- DES 是\_(7)算法。

(7) A. 公开密钥加密 B. 共享密钥加密  
C. 数字签名 D. 认证

- 计算机病毒的特征不包括\_(8)。

(8) A. 传染性 B. 触发性 C. 隐蔽性 D. 自毁性

- MD5 是\_(9)算法, 对任意长度的输入计算得到的结果长度为\_(10)位。

(9) A. 路由选择 B. 摘要 C. 共享密钥 D. 公开密钥  
(10) A. 路由选择 B. 128 C. 140 D. 160

- 使用 Web 方式收发电子邮件时, 以下描述错误的是\_(11)。

(11) A. 无须设置简单邮件传输协议 B. 可以不设置账号密码登录  
C. 邮件可以插入多个附件 D. 未发送邮件可以保存到草稿箱

- 有可能无限期拥有的知识产权是\_(12)。

(12) A. 著作权 B. 专利权  
C. 商标权 D. 集成电路布图设计权

- \_(13)是构成我国保护计算机软件著作权的两个基本法律文件。

(13) A. 《软件法》和《计算机软件保护条例》  
B. 《中华人民共和国著作权法》和《计算机软件保护条例》

C. 《软件法》和《中华人民共和国著作权法》

D. 《中华人民共和国版权法》和《计算机软件保护条例》

- 某软件程序员接受一家公司（软件著作权人）委托开发完成一个软件，三个月后又接受另一家公司委托开发功能类似的软件，此程序员仅将受第一家公司委托开发的软件略作修改即提交给第二家公司，此种行为（14）。

(14) A. 属于开发者的特权

B. 属于正常使用著作权

C. 不构成侵权

D. 构成侵权

- 结构化分析的输出不包括（15）。

(15) A. 数据流图

B. 数据字典

C. 加工逻辑

D. 结构图

- 某航空公司拟开发一个机票预订系统，旅客预订机票时使用信用卡付款。付款通过信用卡公司的信用卡管理系统提供的接口实现。若采用数据流图建立需求模型，则信用卡管理系统是（16）。

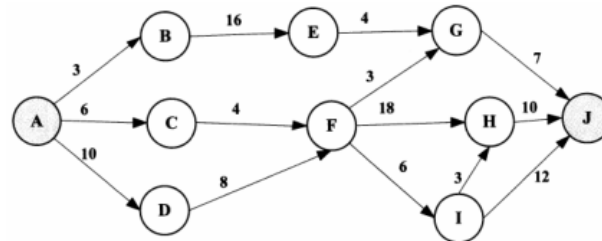
(16) A. 外部实体

B. 加工

C. 数据流

D. 数据存储

- 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示活动的持续时间（天），则完成该项目的最少时间为（17）天。活动 FG 的松弛时间为（18）天。



(17) A. 20

B. 37

C. 38

D. 46

(18) A. 9

B. 10

C. 18

D. 26

- 以下叙述中，（19）不是一个风险。

(19) A. 由另一个小组开发的子系统可能推迟交付，导致系统不能按时交付客户

B. 客户不清楚想要开发什么样的软件，因此开发小组采用原型开发模型帮助其确定需求

C. 开发团队可能没有正确理解客户的需求

D. 开发团队核心成员可能在系统开发过程中离职

- 对布尔表达式进行短路求值是指无须对表达式中所有操作数或运算符进行计算就可确定表达式的值。对于表达式 "a or ((c < d) and b)"，（20）时可进行短路计算。

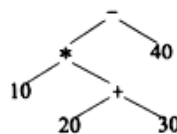
(20) A. d 为 true

B. a 为 true

C. b 为 true

D. c 为 true

- 下面二叉树表示的简单算术表达式为（21）。



(21) A.  $10 * 20 + 30 - 40$

B.  $10 * (20 + 30 - 40)$

C.  $10 * (20 + 30) - 40$

D.  $10 * 20 + (30 - 40)$

- 在程序运行过程中，（22）时涉及整型数据转换为浮点型数据的操作。

(22) A. 将浮点型变量赋值给整型变量

B. 将整型常量赋值给整型变量

C. 将整型变量与浮点型变量相加

D. 将浮点型常量与浮点型变量相加

- 某计算机系统中互斥资源 R 的可用数为 8，系统中有 3 个进程 P1、P2 和 P3 竞争 R，且每个进程都需要 i 个 R，该系统可能会发生死锁的最小 i 值为（23）。

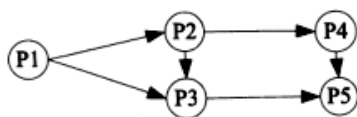
(23) A. 1

B. 2

C. 3

D. 4

- 进程 P1、P2、P3、P4 和 P5 的前趋图如下所示：



若用 PV 操作控制这 5 个进程的同步与互斥的程序如下，那么程序中的空①和空②处应分别为 (24)；空③和空④处应分别为 (25)；空⑤和空⑥处应分别为 (26)。

```

begin
  S1,S2,S3, S4, S5, S6: semaphore;    //定义信号量
  S1:=0; S2:=0; S3:=0; S4:=0; S5:=0; S6:=0;
  Cobegin
    process P1      process P2      process P3      process P4      process P5
    Begin           Begin           Begin           Begin           Begin
      P1 执行;      ②;              P(S2);          P(S4);          ⑥;
      V(S1)          P2 执行;      ③;              P4 执行;          P5 执行;
      ①;              V(S3);          ④;              ⑤;              end;
    end;            end;            end;            end;
  Coend;
end.
  
```

- (24) A. V(S1)和 P(S2)                      B. P(S1)和 V(S2)  
       C. V(S1)和 V(S2)                    D. V(S2)和 P(S1)
- (25) A. V(S3)和 V(S5)                      B. P(S3)和 V(S5)  
       C. V(S3)和 P(S5)                    D. P(S3)和 P(S5)
- (26) A. P (S6) 和 P (S5) V (S6)            B. V (S5) 和 V (S5) V (S6)  
       C. V (S6) 和 P (S5) P (S6)           D. P (S6) 和 P (S5) P (S6)
- 某文件管理系统在磁盘上建立了位示图 (bitmap)，记录磁盘的使用情况。若磁盘上物理块的编号依次为 0、1、2、....；系统中的字长为 32 位，位示图中字的编号依次为 0、1、2、...，每个字中的一个二进制位对应文件存储器上的一个物理块，取值 0 和 1 分别表示物理块是空闲或占用。假设操作系统将 2053 号物理块分配给某文件，那么该物理块的使用情况在位示图中编号为 (27) 的字中描述。  
 (27) A. 32                      B. 33                      C. 64                      D. 65
  - 某操作系统文件管理采用索引节点法。每个文件的索引节点有 8 个地址项，每个地址项大小为 4 字节，其中 5 个地址项为直接地址索引，2 个地址项是一级间接地址索引，1 个地址项是二级间接地址索引，磁盘索引块和磁盘数据块大小均为 1KB。若要访问文件的逻辑块号分别为 1 和 518，则系统应分别采用 (28)。  
 (28) A. 直接地址索引和一级间接地址索引  
       B. 直接地址索引和二级间接地址索引  
       C. 一级间接地址索引和一级间接地址索引  
       D. 一级间接地址索引和二级间接地址索引
  - 某企业拟开发一个企业信息管理系统，系统功能与多个部门的业务相关。现希望该系统能够尽快投入使用，系统功能可以在使用过程中不断改善。则最适宜采用的软件过程模型为 (29)。  
 (29) A. 瀑布模型                      B. 原型模型  
       C. 演化 (迭代) 模型              D. 螺旋模型
  - 能力成熟度模型集成 (CMMI) 是若干过程模型的综合和改进。连续式模型和阶段式模型是 CMMI 提供的两种表示方法，而连续式模型包括 6 个过程域能力等级，其中 (30) 使用量化 (统计学)

手段改变和优化过程域，以应对客户要求的变化和持续改进计划中的过程域的功效。

- (30) A. CL2 (已管理的) B. CL3 (已定义级的)  
C. CL4 (定量管理的) D. CL5 (优化的)

- 在 ISO/IEC 9126 软件质量模型中，可靠性质量特性是指在规定时间内和规定的条件下，软件维持在其性能水平有关的能力，其质量特性不包括 (31)。

- (31) A. 安全性 B. 成熟性 C. 容错性 D. 易恢复性

- 以下关于模块化设计的叙述中，不正确的是 (32)。

- (32) A. 尽量考虑高内聚、低耦合，保持模块的相对独立性  
B. 模块的控制范围在其作用范围内  
C. 模块的规模适中  
D. 模块的宽度、深度、扇入和扇出适中

- 某企业管理信息系统中，采购子系统根据材料价格、数量等信息计算采购的金额，并给财务子系统传递采购金额、收款方和采购日期等信息，则这两个子系统之间的耦合类型为 (33) 耦合。

- (33) A. 数据 B. 标记 C. 控制 D. 外部

- 对以下的程序伪代码 (用缩进表示程序块) 进行路径覆盖测试，至少需要 (34) 个测试用例。采用 McCabe 度量法计算其环路复杂度为 (35)。

```
输入 x, y, z
语句1
if x > 0
    语句2
    if y > 0
        语句3
    Else
        语句4
Else
    语句5
    if z > 0
        语句6
    Else
        语句7
输出语句
```

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

- (35) A. 2 B. 3 C. 4 D. 5

- 某商场的销售系统所使用的信用卡公司信息系统的格式发生了更改，因此对该销售系统进行的修改属于 (36) 维护。

- (36) A. 改正性 B. 适应性 C. 改善性 D. 预防性

- 在面向对象方法中，继承用于 (37)。

- (37) A. 在已存在的类的基础上创建新类 B. 在已存在的类中添加新的方法  
C. 在已存在的类中添加新的属性 D. 在已存在的状态中添加新的状态

- (38) 多态是指操作 (方法) 具有相同的名称但在不同的上下文中所代表的含义不同。

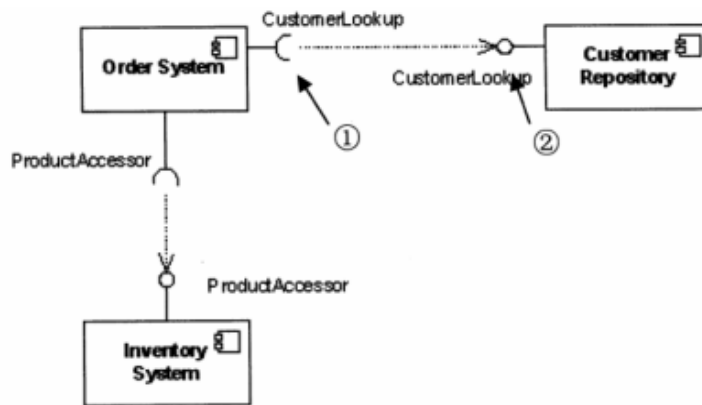
- (38) A. 参数 B. 包含 C. 过载 D. 强制

- 在某销售系统中，客户采用扫描二维码进行支付。若采用面向对象方法开发该销售系统，则客户类属于 (39) 类，二维码类属于 (40) 类。

- (39) A. 接口 B. 实体 C. 控制 D. 状态

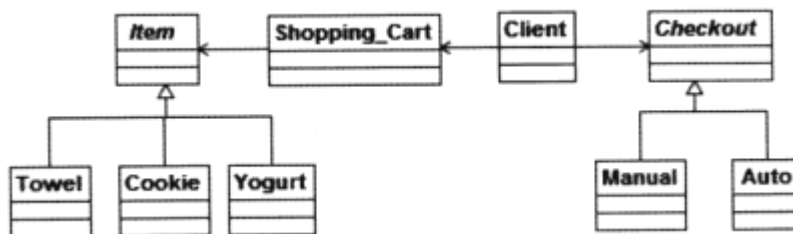
- (40) A. 接口 B. 实体 C. 控制 D. 状态

- 下图所示 UML 图为 (41)，用于展示 (42)。①和②分别表示 (43)。



- (41) A. 类图 B. 组件图 C. 通信图 D. 部署图
- (42) A. 一组对象、接口、协作和它们之间的关系  
B. 收发消息的对象的结构组织  
C. 组件之间的组织和依赖  
D. 面向对象系统的物理模型
- (43) A. 供接口和供接口 B. 需接口和需接口  
C. 供接口和需接口 D. 需接口和供接口

- 假设现在要创建一个简单的超市销售系统，顾客将毛巾、饼干、酸奶等物品（Item）加入购物车（Shopping\_Cart），在收银台（Checkout）人工（Manual）或自动（Auto）地将购物车中每个物品的价格汇总到总价格后结账。这一业务需求的类图（方法略）设计如下图所示，采用了（44）模式。其中（45）定义以一个 Checkout 对象为参数的 accept 操作，由子类实现此 accept 操作。此模式为（46），适用于（47）。

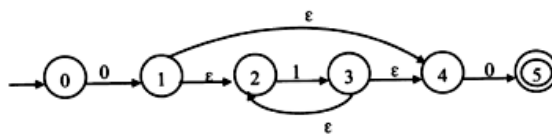


- (44) A. 观察者（Observer） B. 访问者（Visitor）  
C. 策略（Strategy） D. 桥接器（Bridge）
- (45) A. Item B. Shopping\_Cart  
C. Checkout D. Manual 和 Auto
- (46) A. 创建型对象模式 B. 结构型对象模式  
C. 行为型类模式 D. 行为型对象模式
- (47) A. 必须保存一个对象在某一个时刻的（部分）状态  
B. 想在不明确指定接收者的情况下向多个对象中的一个提交一个请求  
C. 需要对一个对象结构中的对象进行很多不同的并且不相关的操作  
D. 在不同的时刻指定、排列和执行请求

- 在以阶段划分的编译器中，（48）阶段的主要作用是分析程序中的句子结构是否正确。

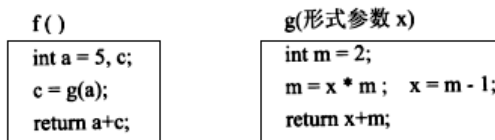
- (48) A. 词法分析 B. 语法分析 C. 语义分析 D. 代码生成

- 下图所示为一个不确定有限自动机（NFA）的状态转换图。该 NFA 可识别字符串（49）。



- (49) A. 0110      B. 0101      C. 1100      D. 1010

- 函数  $f$  和  $g$  的定义如下图所示。执行函数  $f$  时若采用引用 (call by reference) 方式调用函数  $g(a)$ , 则函数  $f$  的返回值为 (50)。



- (50) A. 14      B. 18      C. 24      D. 28

- 数据库系统中的视图、存储文件和基本表分别对应数据库系统结构中的 (51)。

- (51) A. 模式、内模式和外模式      B. 外模式、模式和内模式  
C. 模式、外模式和内模式      D. 外模式、内模式和模式

- 在分布式数据库中, (52) 是指用户或应用程序不需要知道逻辑上访问的表具体如何分块存储。

- (52) A. 逻辑透明      B. 位置透明  
C. 分片透明      D. 复制透明

- 设有关系模式  $R(A_1, A_2, A_3, A_4, A_5, A_6)$ , 函数依赖集  $F = \{A_1 \rightarrow A_3, A_1 A_2 \rightarrow A_4, A_5 A_6 \rightarrow A_1, A_3 A_5 \rightarrow A_6, A_2 A_5 \rightarrow A_6\}$ 。关系模式  $R$  的一个主键是 (53), 从函数依赖集  $F$  可以推出关系模式  $R$  (54)。

- (53) A.  $A_1 A_4$       B.  $A_2 A_5$       C.  $A_3 A_4$       D.  $A_4 A_5$

- (54) A. 不存在传递依赖, 故  $R$  为 1NF  
B. 不存在传递依赖, 故  $R$  为 2NF  
C. 存在传递依赖, 故  $R$  为 3NF  
D. 每个非主属性完全函数依赖于主键, 故  $R$  为 2NF

- 给定关系  $R(A, B, C, D)$  和  $S(C, D, E)$ , 若关系  $R$  与  $S$  进行自然连接运算, 则运算后的元组属性列数为 (55); 关系代数表达式  $\pi_{1,4}(\sigma_{2=5}(R \bowtie S))$  与 (56) 等价。

- (55) A. 4      B. 5      C. 6      D. 7

- (56) A.  $\pi_{A,D}(\sigma_{C=D}(R \times S))$       B.  $\pi_{R.A,R.D}(\sigma_{R.B=S.C}(R \times S))$   
C.  $\pi_{A,R,D}(\sigma_{R.C=S.D}(R \times S))$       D.  $\pi_{R.A,R,D}(\sigma_{R.B=S.E}(R \times S))$

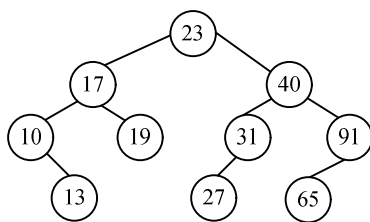
- 栈的特点是后进先出, 若用单链表作为栈的存储结构, 并用头指针作为栈顶指针, 则 (57)。

- (57) A. 入栈和出栈操作都不需要遍历链表  
B. 入栈和出栈操作都需要遍历链表  
C. 入栈操作需要遍历链表而出栈操作不需要  
D. 入栈操作不需要遍历链表而出栈操作需要

- 已知某二叉树的先序遍历序列为  $A B C D E F$ 、中序遍历序列为  $B A D C F E$ , 则可以确定该二叉树 (58)。

- (58) A. 是单支树 (即非叶子结点都只有一个孩子)  
B. 高度为 4 (即结点分布在 4 层上)  
C. 根结点的左子树为空  
D. 根结点的右子树为空

- 可以构造出下图所示二叉排序树 (二叉检索树、二叉查找树) 的关键码序列是 (59)。



- (59) A. 10 13 17 19 23 27 31 40 65 91      B. 23 40 91 17 19 10 31 65 27 13  
C. 23 19 40 27 17 13 10 91 65 31      D. 27 31 40 65 91 13 10 17 23 19

- 图 G 的邻接矩阵如下图所示 (顶点依次表示为  $v_0$ 、 $v_1$ 、 $v_2$ 、 $v_3$ 、 $v_4$ 、 $v_5$ )，G 是 (60)。对 G 进行广度优先遍历 (从  $v_0$  开始)，可能的遍历序列为 (61)。

$\infty$	18	17	$\infty$	$\infty$	$\infty$
$\infty$	$\infty$	$\infty$	20	16	$\infty$
$\infty$	19	$\infty$	23	$\infty$	$\infty$
$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	15
$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	12
$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

- (60) A. 无向图      B. 有向图      C. 完全图      D. 强连通图

- (61) A.  $v_0$ 、 $v_1$ 、 $v_2$ 、 $v_3$ 、 $v_4$ 、 $v_5$       B.  $v_0$ 、 $v_2$ 、 $v_4$ 、 $v_5$ 、 $v_1$ 、 $v_3$   
C.  $v_0$ 、 $v_1$ 、 $v_3$ 、 $v_5$ 、 $v_2$ 、 $v_4$       D.  $v_0$ 、 $v_2$ 、 $v_4$ 、 $v_3$ 、 $v_5$ 、 $v_1$

- 在一条笔直公路的一边有许多房子，现要安装消防栓，每个消防栓的覆盖范围远大于房子的面积，如下图所示。现求解能覆盖所有房子的最少消防栓数和安装方案 (问题求解过程中，可将房子和消防栓均视为直线上的点)。

该问题求解算法的基本思路为：从左端的第一栋房子开始，在其右侧  $m$  米处安装一个消防栓，去掉被该消防栓覆盖的所有房子。在剩余的房子中重复上述操作，直到所有房子被覆盖。算法采用的设计策略为 (62)；对应的时间复杂度为 (63)。



假设公路起点 A 的坐标为 0，消防栓的覆盖范围 (半径) 为 20m，10 栋房子的坐标为 (10, 20, 30, 35, 60, 80, 160, 210, 260, 300)，单位为 m。根据上述算法，共需要安装 (64) 个消防栓。以下关于该求解算法的叙述中，正确的是 (65)。

- (62) A. 分治      B. 动态规划      C. 贪心      D. 回溯  
(63) A.  $O(\lg n)$       B.  $O(n)$       C.  $(n \lg n)$       D.  $O(n^2)$   
(64) A. 4      B. 5      C. 6      D. 7  
(65) A. 肯定可以求得问题的一个最优解      B. 可以求得问题的所有最优解  
C. 对有些实例，可能得不到最优解      D. 只能得到近似最优解

- 使用 ADSL 接入 Internet，用户端需要安装 (66) 协议。

- (66) A. PPP      B. SLIP      C. PPTP      D. PPPoE

- 下列命令中，不能用于诊断 DNS 故障的是 (67)。

- (67) A. netstat      B. nslookup      C. ping      D. tracert



- 以下关于 TCP/IP 协议和层次对应关系的表示中, 正确的是 (68)。

(68) A.	HTTP	SNMP
	TCP	UDP
	IP	

B.	FTP	Telnet
	UDP	TCP
	ARP	

C.	HTTP	SMTP
	TCP	UDP
	IP	

D.	SMTp	FTP
	UDP	TCP
	ARP	

- 把 CSS 样式表与 HTML 网页关联, 不正确的方法是 (69)。

- (69) A. 在 HTML 文档的<head>标签内定义 CSS 样式  
B. 用@import 引入样式表文件  
C. 在 HTML 文档的<!-- -->标签内定义 CSS 样式  
D. 用<link>标签链接网上可访问的 CSS 样式表文件

- 使用 (70) 命令可以释放当前主机自动获取的 IP 地址。

- (70) A. `ipconfig/all` B. `ipconfig/reload`  
C. `ipconfig/release` D. `ipconfig/reset`

- The project workbook is not so much a separate document as it is a structure imposed on the documents that the project will be producing anyway.

All the documents of the project need to be part of this 71. This includes objectives ,external specifications , interface specifications , technical standards, internal specifications and administrative memoranda(备忘录).

Technical prose is almost immortal. If one examines the genealogy (手冊) of a customer manual for a piece of hardware or software, one can trace not only the ideas, but also many of the very sentences and paragraphs back to the first (72) proposing the product or explaining the first design. For the technical writer, the paste-pot is as mighty as the pen.

Since this is so, and since tomorrow's product-quality manuals will grow from today's memos, it is very important to get the structure of the documentation right. The early design of the project (73) ensures that the documentation structure itself is crafted, not haphazard. Moreover, the establishment of a structure molds later writing into segments that fit into that structure.

The second reason for the project workbook is control of the distribution of (74). The problem is not to restrict information, but to ensure that relevant information gets to all the people who need it.

The first step is to number all memoranda, so that ordered lists of titles are available and worker can see if he has what he wants. The organization of the workbook goes well beyond this to establish a tree-structure of memoranda. The (75) allows distribution lists to be maintained by subtree, if that is desirable.

- (71) A. structure      B. specification      C. standard      D. objective
- (72) A. objective      B. memoranda      C. standard      D. specification
- (73) A. title      B. list      C. workbook      D. quality
- (74) A. product      B. manual      C. document      D. information
- (75) A. list      B. document      C. tree-structure      D. number

# 全国计算机技术与软件专业技术资格考试

## 2018 年下半年 软件设计师 下午试卷

（考试时间 14:00～16:30 共 150 分钟）

请按下述要求正确填写答题纸

1. 本试卷共六道题，其中，试题（一）～试题（四）为必答题，试题（五）～试题（六）为选答题，满分 75 分。
2. 在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。
3. 在答题纸的指定位置填写准考证号、出生年月日和姓名。
4. 答题纸上除填写上述内容外只能写解答。
5. 解答时字迹务必清楚，字迹不清时，将不评分。

### 例题

2019 年上半年全国计算机技术与软件专业技术资格考试日期是\_\_（1）\_\_月\_\_（2）\_\_日。

因为正确的解答是“5 月 25 日”，故在答题纸的对应栏内写上“5”和“25”（参看下表）。

例题	解答栏
（1）	5
（2）	25

### 试题一（15 分）

阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

#### 【说明】

某房产中介连锁企业欲开发一个基于 Web 的房屋中介信息系统，以有效管理房源和客户，提高成交率。该系统的主要功能是：

（1）房源采集与管理。系统自动采集外部网站的潜在房源信息，保存为潜在房源。由经纪人联系确认的潜在房源变为房源，并添加出售/出租房源的客户。由经纪人或客户登记的出售/出租房源，系统将其保存为房源。房源信息包括基本情况、配套设施、交易类型、委托方式、业主等。经纪人可以对房源进行更新等管理操作。

（2）客户管理。求租/求购客户进行注册、更新，推送客户需求给经纪人，或由经纪人对求租/求购客户进行登记、更新。客户信息包括身份证号、姓名、手机号、需求情况、委托方式等。

（3）房源推荐。根据客户的需求情况（求购/求租需求情况以及出售/出租房源信息），向已登录的客户推荐房源。

（4）交易管理。经纪人对租售客户双方进行交易信息管理，包括订单提交和取消，设置收取中介费比例。财务人员收取中介费之后，表示该订单已完成，系统更新订单状态和房源状态，向客户和经纪人发送交易反馈。

（5）信息查询。客户根据自身查询需求查询房屋供需信息。

现采用结构化方法对房屋中介信息系统进行分析与设计，获得如图 1 所示的上下文数据流图和图 2 所示的 0 层数据流图。

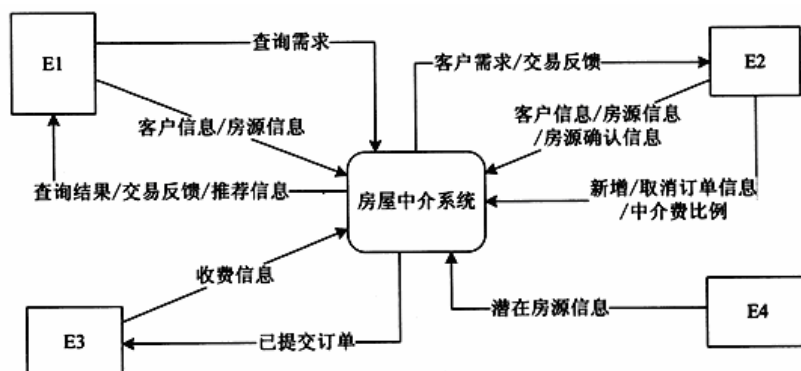


图 1 上下文数据流图

#### 【问题 1】（4 分）

使用说明中的词语，给出图 1 中的实体 E1~E4 的名称。

#### 【问题 2】（4 分）

使用说明中的词语，给出图 2 中的数据存储 D1~D4 的名称。

#### 【问题 3】（3 分）

根据说明和图中术语，补充图 2 中缺失的数据流及其起点和终点。

#### 【问题 4】（4 分）

根据说明中术语，给出图 1 中数据流“客户信息”“房源信息”的组成。

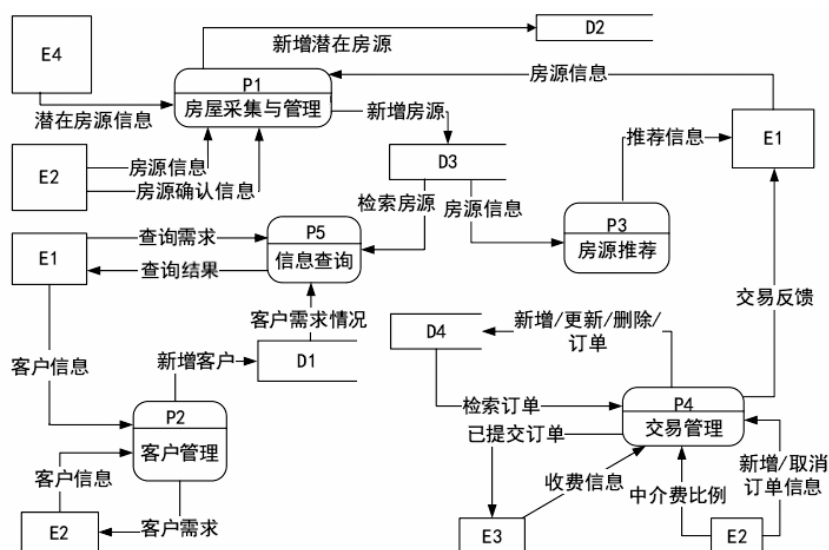


图2 0层数据流图

## 试题二（15分）

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】

某集团公司拥有多个分公司，为了方便集团公司对分公司各项业务活动进行有效管理，集团公司决定构建一个信息系统以满足公司的业务管理需求。

### 【需求分析】

（1）分公司关系需要记录的信息包括分公司编号、名称、经理、联系地址和电话。分公司编号唯一标识分公司信息中的每一个元组。每个分公司只有一名经理，负责该分公司的管理工作。每个分公司设立仅为本分公司服务的多个业务部门，如研发部、财务部、采购部、销售部等。

（2）部门关系需要记录的信息包括部门号、部门名称、主管号、电话和分公司编号。部门号唯一标识部门信息中的每一个元组。每个部门只有一名主管，负责部门的管理工作。每个部门有多名员工，每名员工只能隶属于一个部门。

（3）员工关系需要记录的信息包括员工号、姓名、隶属部门、岗位、电话和基本工资。其中，员工号唯一标识员工信息中的每一个元组。岗位包括：经理、主管、研发员、业务员等。

### 【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如图3所示。

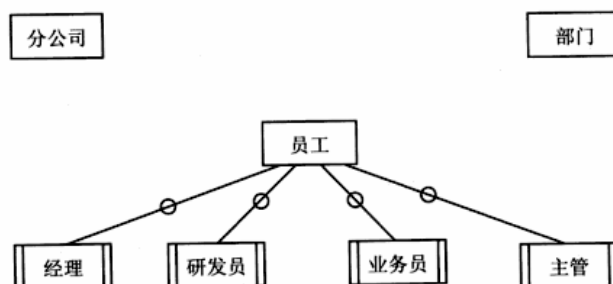


图3 实体联系图

【关系模式设计】

分公司[分公司编号，名称，(a)，联系地址，电话]

部门[部门号，部门名称，(b)，电话]

员工[员工号，姓名 (c)，电话，基本工资]

【问题 1】(4 分)

根据问题描述，补充 4 个联系，完善图 3 的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系类型为 1:1、1:n 和 m:n (或 1:1、1:\*和\*:\* )。

【问题 2】(5 分)

根据题意，将关系模式中的空 (a) ~ (c) 补充完整。

【问题 3】(4 分)

给出“部门”和“员工”关系模式的主键和外键。

【问题 4】(2 分)

假设集团公司要求系统能记录部门历任主管的任职时间和任职年限，那么是否需要在数据库设计时增设一个实体？为什么？

试题三 (15 分)

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】

社交网络平台 (SNS) 的主要功能之一是建立在线群组，群组中的成员之间可以互相分享或挖掘兴趣和活动。每个群组包含标题、管理员以及成员列表等信息。

社交网络平台的用户可以自行选择加入某个群组。每个群组拥有一个主页，群组内的所有成员都可以查看主页上的内容。如果在群组的主页上发布或更新了信息，群组中的成员会自动接收到发布或更新后的信息。

用户可以加入一个群组也可以退出这个群组。用户退出群组后，不会再接收到该群组发布或更新的任何信息。

现采用面向对象方法对上述需求进行分析和设计，得到如表 1 所示的类列表和如图 4 所示的类图。

表 1 类列表

类名	描述
SNSSubject	群组主页的内容
SNSGroup	社交网络平台中的群组（在主页上发布信息）
SNSObserver	群组主页内容的关注者
SNSUser	社交网络平台用户/群组成员
SNSAdmin	群组的管理员

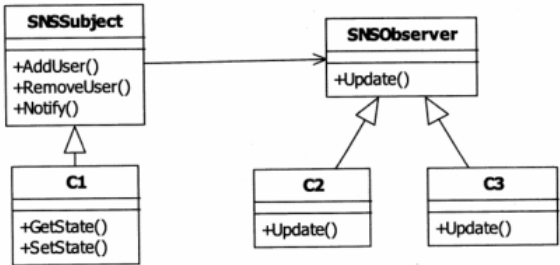


图 4 类图

【问题 1】(6 分)

根据说明中的描述，给出图 4 中 C1 ~C3 所对应的类名。

【问题 2】(6 分)

图 4 中采用了哪一种设计模式？说明该模式的意图及其适用场合。

【问题 3】(3 分)

现在对上述社交网络平台提出了新的需求：一个群体可以作为另外一个群体中的成员，例如群体 A 加入群体 B。那么，群体 A 中的所有成员就自动成为群体 B 中的成员。

若要实现这个新需求，需要对图 4 进行哪些修改？（以文字方式描述）

试题四 (15 分)

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

给定一个字符序列  $B = b_1b_2...b_n$ ，其中  $b_i \in \{A, C, G, U\}$ 。 $B$  上的二级结构是一组字符对集合  $S = \{(b_i, b_j)\}$ ，其中  $i, j \in \{1, 2, ..., n\}$ ，并满足以下四个条件：

- (1)  $S$  中的每对字符是 (A,U), (U,A), (C,G) 和 (G,C) 四种组合之一；
- (2)  $S$  中的每对字符之间至少有四个字符将其隔开，即  $i < j - 4$ ；
- (3)  $S$  中每一个字符（记为  $b_k$ ）的配对存在两种情况： $b_k$  不参与任何配对； $b_k$  和字符  $b_t$  配对，其中  $t < k - 4$ ；
- (4)（不交叉原则）若  $(b_i, b_j)$  和  $(b_k, b_t)$  是  $S$  中的两个字符对，且  $i < k$ ，则  $i < k < j < t$  不成立。

$B$  的具有最大可能字符对数的二级结构  $S$  被称为最优配对方案，求解最优配对方案中的字符对数的方法如下：

假设用  $C(i, j)$  表示字符序列  $b_i b_{i+1} ... b_j$  的最优配对方案（即二级结构  $S$ ）中的字符对数，则  $C(i, j)$  可以递归定义为：

$$C(i, j) = \begin{cases} \max(C(i, j-1), \max(C(i, t-1) + 1 + C(t+1, j-1))) & \text{若 } b_t R b_j \text{ 匹配且 } i < j-4 \\ 0 & \text{否则} \end{cases}$$

下面代码是算法的 C 语言实现，其中

n：字符序列长度

B[]：字符序列

C[][]：最优配对数量数组

【C 代码】

```
#include<stdio.h>
#include<stdlib.h>
#define LEN 100
/*判断两个字符是否配对*/
int isMatch(char a, char b)
{
    if ((a == 'A' && b == 'U') || (a == 'U' && b == 'A'))
        return 1;
    if ((a == 'C' && b == 'G') || (a == 'G' && b == 'C'))
        return 1;
    return 0;
}

/*求最大配对数*/
int RNA_2(char B[LEN], int n){
    int i, j, k, t;
    int max;
    int C[LEN][LEN] = { 0 };

```

```

for (k = 5; k <= n - 1; k++)
{
    for (i = 1; i <= n - k; i++)
    {
        j = i + k;
        (1);
        for ((2); t <= j - 4; t++)
        {
            if ((3) && max < C[i][t - 1] + 1 + C[t + 1][j - 1])
                max = C[i][t - 1] + 1 + C[t + 1][j - 1];
        }
        C[i][j] = max;
        printf("c[%d][%d]=%d--", i, j, C[i][j]);
    }
}

return (4);
}

```

**【问题 1】** (8 分)

根据题干说明，填充 C 代码中的空 (1) ~ (4)。

**【问题 2】** (4 分)

根据题干说明和 C 代码，算法采用的设计策略为 (5)。

算法的时间复杂度为 (6)，(用 O 表示)。

**【问题 3】** (3 分)

给定字符序列 ACCGGUAGU，根据上述算法求得最大字符对数为 (7)。

**试题五 (15 分)**

阅读下列说明和 C++ 代码，将应填入 (1) ~ (5) 处的字句写在答题纸的对应栏内。

**【说明】**

某航空公司的会员积分系统将其会员划分为普卡 (Basic)、银卡 (Silver) 和金卡 (Gold) 三个等级。非会员 (NonMember) 可以申请成为普卡会员。会员的等级根据其一年内累积的里程数进行调整。描述会员等级调整的状态图如图 5 所示。现采用状态 (State) 模式实现上述场景，得到如图 6 所示的类图。

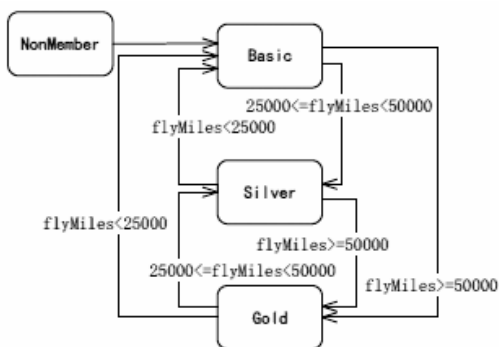


图 5 会员等级调整状态图

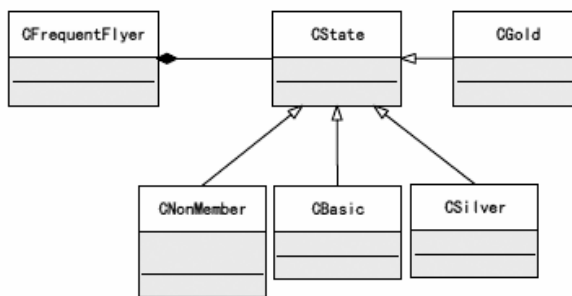


图 6 状态模式类图

**【C++ 代码】** 代码修改点做了备注

```

#include <iostream>
using namespace std;

```

```

class FrequentFlyer; class CBasic; class CSilver; class CGold; class CNoCustomer; //提前引用
class CState
{
private:
    int flyMiles;    //里程数
public:
    (1);
    //根据累积里程数调整会员等级
};
class FrequentFlyer
{
friend class CBasic;
friend CSilver;           //此处应在 CSilver 前面少了 class
friend class CGold;
private:                  //此处应该改为 public 题目有误
    CState *state; CState *nocustomer; CState *basic; CState *silver; CState *gold;
    double flyMiles;
public:
    FrequentFlyer()
    {
        flyMiles = 0;
        setState(nocustomer);
    }
    void setState(CState *state)
    {
        this->state = state;
    }
    void travel(int miles)
    {
        double bonusMiles = state->travel(miles, this);
        flyMiles = flyMiles + bonusMiles;
    }
};
class CNoCustomer :public CState    //非会员
{
public:
    double travel(int miles, FrequentFlyer *context)
    {
        cout << "Your travel will not account for points\n";    //不累积里程数
        return miles;
    }
};
class CBasic:public CState    //普卡会员
{
public:
    double travel(int miles, FrequentFlyer* context)
    {
        if (context->flyMiles >= 25000 && context->flyMiles < 50000)

```



```

        (2) ;
    if (context->flyMiles<25000)
        (3) ;
    return miles + 0.5*miles;    //累积里程数
}
};
class CGold :public CState      //金卡会员
{
public:
    double travel(int miles, FrequentFlyer* context)
    {
        if (context->flyMiles >= 25000 && context->flyMiles < 50000)
            (4) ;
        if (context->flyMiles < 25000)
            (5) ;
        return miles + 0.5*miles;    //累积里程数
    }
};
class CSilver :public CState    //银卡会员
{
    double travel(int miles, FrequentFlyer* context)
    {
        if (context->flyMiles<25000)
            context->setState(context->basic);
        if (context->flyMiles >= 50000)
            context->setState(context->gold);
        return (miles + 0.25*miles);    //累积里程数
    }
};
};

```

#### 试题六（15 分）

阅读下列说明和 Java 代码，将应填入（1）～（5）处的字句写在答题纸的对应栏内。

##### 【说明】

某航空公司的会员积分系统将其会员划分为普卡（Basic）、银卡（Silver）和金卡（Gold）三个等级。非会员（NonMember）可以申请成为普卡会员。会员的等级根据其一年内累积的里程数进行调整。描述会员等级调整的状态图如图 7 所示。现采用状态（State）模式实现上述场景，得到如图 8 所示的类图。

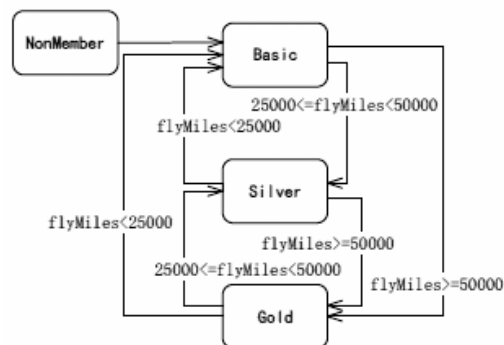


图 7 会员等级调整状态图

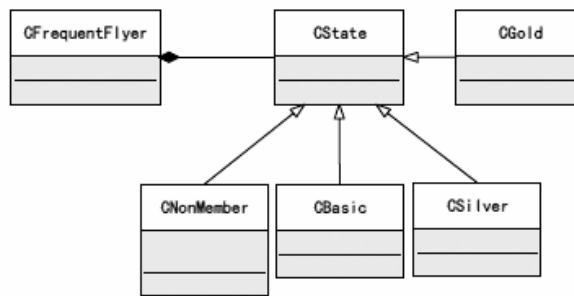


图 8 状态模式类图

【Java 代码】

```

import java.util.*;

abstract class CState
{
    public int flyMiles;    //里程数
    public (1) ;    //根据累积里程数调整会员等级
}

class CNoCustomer extends CState    //非会员
{
    public double travel(int miles, FrequentFlyer context)
    {
        System.out.println("Your travel will not account for points");
        return miles;    //不累积里程数
    }
}

class CBasic extends CState    //普卡会员
{
    public double travel(int miles, FrequentFlyer context)
    {
        if (context.flyMiles >= 25000 && context.flyMiles < 50000)
            (2) ;
        if (context.flyMiles >= 50000)
            (3) ;
        return miles;
    }
}

class CGold extends CState    //金卡会员
{
    public double travel(int miles, FrequentFlyer context)
    {
        if (context.flyMiles >= 25000 && context.flyMiles < 50000)
            (4) ;
        if (context.flyMiles <= 25000)
            (5) ;
        return miles + 0.5*miles;    //累积里程数
    }
}
  
```

```

    }

class CSilver extends CState           //银卡会员
{
    public double travel(int miles, FrequentFlyer context)
    {
        if (context.flyMiles <= 25000)
            context.setState(new CBasic());
        if (context.flyMiles > 50000)
            context.setState(new CGold());
        return (miles + 0.25*miles);    //累积里程数
    }
}

class FrequentFlyer
{
    CState state;
    double flyMiles;
    public FrequentFlyer()
    {
        state = new CNoCustomer();
        flyMiles = 0;
        setState(state);
    }
    public void setState(CState state){this.state = state;}
    public void travel(int miles)
    {
        double bonusMiles = state.travel(miles, this);
        flyMiles = flyMiles + bonusMiles;
    }
}

```

# 全国计算机技术与软件专业技术资格考试

## 2018 年下半年 软件设计师 上午试卷解析

(1) 参考答案: B

☞ 试题分析 程序计数器 pc 是用于存放下一条指令所在单元的地址的地方。

指令寄存器是临时放置从内存里面取得的程序指令的寄存器,用于存放当前从主存储器读出的正在执行的一条指令。

地址寄存器用来保存当前 CPU 所访问的内存单元的地址。

指令译码器从内存中取出的一条指令经数据总线送往指令寄存器中。

(2) 参考答案: A

☞ 试题分析 BIOS 是一个程序,是固化在主板的 BIOS ROM 芯片里面的,它相当于硬件底层的一个操作系统,控制和保存着硬件的输入输出工作。

(3) 参考答案: D

☞ 试题分析 对于有  $n$  位的整数补码,其取值范围是  $-(2^{n-1}) \sim (2^{n-1})-1$ ,以 8 位的整数补码举例,其有效取值范围是  $-(2^7) \sim (2^7)-1$ ,也就是  $-128 \sim 127$ 。拿 C 项计算值为 128 越界了,只有 D 项符合。其他选项都越界了。

(4) 参考答案: C

☞ 试题分析

串行系统的可靠度  $R=R_1 \times R_2 \times \cdots \times R_n$

并联系统的可靠度为  $R=1-(1-R_1) \times (1-R_2) \times \cdots \times (1-R_n)$

(5) 参考答案: C

☞ 试题分析 奇校验:如果以二进制数据中 1 的个数是奇数为依据,则是奇校验。

偶校验:如果以二进制数据中 1 的个数是偶数为依据,则是偶校验。

如果传输过程中包括校验位在内的奇数个数据位发生改变,那么奇校验位将出错,表示传输过程有错误发生,但是由于没有办法确定哪一位出错,所以它不能进行错误校正。

(6) 参考答案: A

☞ 试题分析 流水线技术是指在程序执行时,多条指令重叠进行操作的一种任务分解技术。把一个任务分解为若干顺序执行的子任务,不同的子任务由不同的执行机构来负责执行,而这些执行机构可以同时并行工作。

(7) 参考答案: B

☞ 试题分析 共享密钥加密算法又称对称加密算法,即用同一个密钥去加密和解密数据。

常用的算法包括 DES、3DES、IDEA、AES 等。

(8) 参考答案: D

☞ 试题分析 计算机病毒具有隐蔽性、传染性、潜伏性、触发性和破坏性等特性。

因此不包括选项 D 自毁性。

(9)(10) 参考答案: B B

☞ 试题分析 常见的摘要算法有:MD5、SHA 等。

MD5:信息摘要算法第五版,输入以 512 位为分组,进行处理,产生一个 128 位的输出。

(11) 参考答案: B

☞ 试题分析 谈到收发这两个关键字,使用 Web 方式收发邮件是必须设置账号密码登录的。如果只是简单地查看可以不设置账号密码。

(12) 参考答案: C

☞ 试题分析 根据我国《著作权法》第二十条的规定,作者的署名权、修改权、保护作品完整权的保护期不受限制。

但是商标权可以每 10 年无限续期。所以也存在无限期拥有。但是题目中提到“有可能”,所以应该选择 C, A 选项为无限期。

(13) 参考答案: B

☞ 试题分析 我国保护计算机软件著作权的两个基本法律文件是《中华人民共和国著作权法》和《计算机软件保护条例》。

(14) 参考答案: D

☞ 试题分析 本题目指明了第一家软件公司为软件著作权人,所以再把略作修改的软件交给第二家,构成侵权行为。

(15) 参考答案: D

☞ 试题分析 《软件设计师教程》(第 5 版) P325 页:结构化方法的分析结果由以下几部分组成:一套分层的数据流图、一本数据词典、一组小说明(也称加工逻辑说明)、补充材料。因此 D 选项不包括在内。

(16) 参考答案: A

☞ 试题分析 数据流图的成分包括:数据存储、数据流、加工、外部实体。其中,数据流、加工、数据存储是用于系统内部数据处理的模型,而外部实体指系统以外,又和系统有联系的人或事物或软件系统。所以正确选项为 A。

(17)(18) 参考答案: D C

☞ 试题分析 最少完成时间是指完成项目工期的最长时间。因此, ADFHJ 为项目活动图的关键路径,等于 46 天。FG 的松弛时间为 18 天。46-7-3 和 46-10-18 相比,差值为 18,所以此选项为 18。

(19) 参考答案: B

☞ 试题分析 一般而言,风险与不确定性有关,若某一事件的发生存在着两种或两种以上的可能性,即可认为该事件存在风险。

但是选项 B 已经确定客户不清楚需求,所以是确定事情,不存在风险。

(20) 参考答案: B

☞ 试题分析 此题考察的是短路问题。or 代表或的意思, and 代表并且的意思。所以此题 a 为 true 后面的就可以不用执行下去,直接判定为真。所以形成短路效应。

(21) 参考答案: C

☞ 试题分析 此题考察的是二叉树排序问题。根据图示可知最后计算的符号为-号,其次为\*号,最后为+号,所以答案为 C 选项。

(22) 参考答案: C

☞ 试题分析 此题 B、D 选项为同数据类型,对于 A 选项是将 float 数据类型转成 int 数据类型, C 选项是将 int 类型和 float 类型进行相加,实现了 int 类型转成 float 类型。

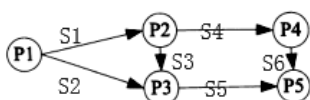
(23) 参考答案: D

☞ 试题分析 如果是 3,每个进程都需要 3 个资源。按照给每个最大资源分配会是 3, 3, 2 这种情况,那么有两个进程资源就够了。

结束后释放给最后那个进程 8 个资源足够,如果 i 是 4, 3, 3, 2 这种必定是死锁了。

(24) ~ (26) 参考答案: D B C

☞ 试题分析

首先标明信号  ,然后由题可知正确选项为 D、B、C。

(27) 参考答案: C

☞ 试题分析  $2053/32=64.156$ , 并且位示图是从 0 号开始, 所以选择 C 选项。

(28) 参考答案: B

☞ 试题分析 试题分析直接地址索引的地址项有 5 个, 所以对应的可访问逻辑块号为第 0 块~第 4 块, 所以逻辑块号为 1 的块, 可通过直接地址索引来访问。由于每个块的大小为 1KB, 每个地址项占 4 个字节, 因此, 所以每个块可以容纳的地址项数量为  $1\text{KB}/4\text{ Byte}=256$ , 由此可知, 2 个一级间接地址索引的地址项可访问的最大逻辑块号为  $256*2+4=516$  号; 1 个二级间接地址索引的地址项可访问的最大逻辑块号为  $256*256+516=66052$ ; 可见, 要访问逻辑块号为 518 的块, 必须得采用二级间接地址索引。

(29) 参考答案: C

☞ 试题分析 增量模型在各个阶段并不交付一个可运行的完整产品, 而是交付满足客户需求的一个子集的可运行产品。整个产品被分解成若干个构件, 开发人员逐个构件地交付产品, 这样做的好处是软件开发可以较好地适应变化, 客户可以不断地看到所开发的软件, 从而降低开发风险。

(30) 参考答案: D

☞ 试题分析

CL0: 未完成的, 未执行或未达到 CL1 定义的所有目标。

CL1: 已执行的, 共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品, 以实现支持过程域的特定目标。

CL2: 已管理的, 共性目标集中于已管理的过程的制度化。

CL3: 已定义级的, 共性目标集中于已定义过程的制度化。

CL4: 定量管理的, 共性目标集中于可定量管理的过程的制度化。

CL5: 优化的使用量化(统计学)手段改变和优化过程域, 以对付客户要求的可持续改进计划中的过程域的功效。

(31) 参考答案: A

☞ 试题分析 软件质量模型如下图所示, 其中质量子特性不包括安全性。



(32) 参考答案: B

☞ 试题分析 模块化设计的原则应该是尽量考虑高内聚、低耦合。

1) 模块独立。模块的独立性原则表现在模块完成独立的功能, 与其他模块的联系应该尽可能地简单, 各个模块具有相对的独立性。

2) 模块的规模要适当。模块的规模不能太大, 也不能太小。如果模块的功能太强, 可读性就会较差, 若模块的功能太弱, 就会有很多的接口。读者需要通过较多的程序设计来进行经验的积累。

3) 分解模块时要注意层次。在进行多层次任务分解时, 要注意对问题进行抽象化。在分解初期,

可以只考虑大的模块，在中期，再逐步进行细化，分解成较小的模块进行设计。

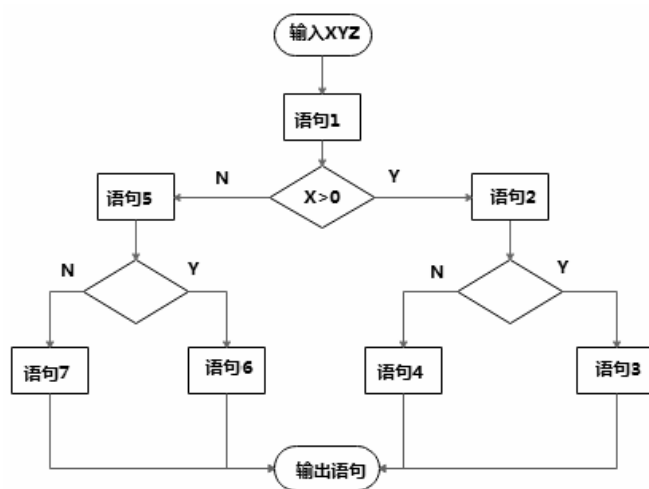
(33) 参考答案: A

🔍 试题分析

- 数据耦合指两个模块之间有调用关系，传递的是简单的数据值，相当于高级语言中的值传递。
- 标记耦合指两个模块之间传递的是数据结构。
- 控制耦合指一个模块调用另一个模块时，传递的是控制变量，被调用模块通过该控制变量的值有选择地执行模块内的某一功能。因此，被调用模块应具有多个功能，哪个功能起作用受调用模块控制。
- 外部耦合指模块间通过软件之外的环境联结（如 I/O 将模块耦合到特定的设备、格式、通信协议上）。

(34) (35) 参考答案: B C

🔍 试题分析



由公式可知  $V(G)=m-n+2$ ，得到  $14-12+2=4$ 。

(36) 参考答案: B

🔍 试题分析

更正性。由于系统测试不可能揭露系统存在的所有错误，因此在系统投入运行后频繁的实际应用过程中，就有可能暴露出系统内隐藏的错误。

适应性维护。适应性维护是为了使系统适应环境的变化而进行的维护工作。

完善性维护。在系统的使用过程中，用户往往要求扩充原有系统的功能，增加一些在软件需求规范书中没有规定的功能与性能特征，以及对处理效率和编写程序的改进。

预防性维护。系统维护工作不应总是被动地等待用户提出要求后才进行，应进行主动的预防性维护，即选择那些还有较长使用寿命，目前尚能正常运行，但可能将要发生变化或调整的系统进行维护，目的是通过预防性维护为未来的修改与调整奠定更好的基础。

(37) 参考答案: A

🔍 试题分析

本题考查的是继承的定义：继承是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上进行。因此本题选择 A 选项。

(38) 参考答案: C

🔍 试题分析

参数多态：应用广泛、最纯的多态。

包含多态：同样的操作可用于一个类型及其子类型。包含多态一般需要进行运行时的类型检查。

过载多态：同一个名（操作符、函数名）在不同的上下文中有不同的类型。

强制多态：编译程序通过语义操作，把操作对象的类型强行加以变换，以符合函数或操作符的要求。

本题应该选择 C 选项过载多态。

(39) (40) 参考答案: B A

☞ 试题分析 类可以分为三种: 实体类、接口类(边界类)和控制类。

实体类的对象表示现实世界中真实的实体, 如人、物等。

接口类(边界类)的对象为用户提供一种与系统合作交互的方式, 分为人和系统两大类, 其中人的接口可以是显示屏、窗口、Web 窗体、对话框、菜单、列表框、其他显示控制、条形码、二维码或者用户与系统交互的其他方法。系统接口涉及到把数据发送到其他系统, 或者从其他系统接收数据。

控制类的对象用来控制活动流, 充当协调者。

(41) (42) (43) 参考答案: B C D

☞ 试题分析 请参考《软件设计师教程》(第5版) P375 组件图。

末端有一个完整的圆周的接口符号代表组件提供接口。

在末端只有半个圆的接口符号代表组件需求接口。

(44) ~ (47) 参考答案: B A D C

☞ 试题分析 访问者模式包含如下角色:

Visitor: 抽象访问者。

ConcreteVisitor: 具体访问者。

Element: 抽象元素。

ConcreteElement: 具体元素。

ObjectStructure: 对象结构。

(48) 参考答案: B

☞ 试题分析

1) 词法分析阶段: 识别出一个个的单词, 删掉无用的信息, 报告分析时的错误。

2) 语法分析阶段: 语法分析器以单词符号作为输入, 分析单词符号是否形成符合语法规则的语法单位, 如表达式、赋值、循环等, 按语法规则分析检查每条语句是否有正确的逻辑结构。语法分析构造一棵语法树。

3) 语义分析阶段: 主要检查源程序是否存在语义错误, 并收集类型信息供后面的代码生成阶段使用, 如: 赋值语句的右端和左端的类型不匹配, 表达式的除数是否为零等。

(49) 参考答案: A

☞ 试题分析 由题目的图中可知以 0 开始以 0 结尾, 所以只有 A 符合。

(50) 参考答案: D

☞ 试题分析

代入可知  $m=2*5; x=10-1;$

`return x+m=19`

$19+9=28$

(51) 参考答案: D

☞ 试题分析

本题考察的是数据库体系结构: 三层模式。对于题干给出的视图、存储文件、基本表分别对应: 视图-外模式, 存储文件-内模式, 基本表-模式。因此本题选择 D 选项。

(52) 参考答案: C

☞ 试题分析 本题考查的是分布式数据库相关知识。

1) 逻辑透明性(局部映像透明性): 它是最低层次的透明性, 该透明性提供数据到局部数据库的映像, 即用户不必关心局部 DBMS 支持哪种数据模型、使用哪种数据操纵语言, 数据模型和操纵语言的转换是由系统完成的。因此, 局部映像透明性对异构型和同构异质的分布式数据库系统是非常重要的。

2) 位置透明性: 用户不必知道所操作的数据放在何处, 即数据分配到哪个或哪些站点存储对用户是透明的。因此, 数据分片模式的改变, 如把数据从一个站点转移到另一个站点将不会影响应用程序, 因而应用程序不必改写。



3) 分片透明性: 用户不必关心数据是如何分片, 他们对数据的操作在全局关系上进行, 即关心如何分片对用户是透明的, 因此, 当分片改变时应用程序可以不变。

4) 复制透明性: 用户不用关心数据库在网络中的各个节点的复制情况, 被复制的数据的更新都由系统自动完成。

(53) (54) 参考答案: B D

试题分析 把每个选项代入求它的闭包, A2A5 能推出所有属性的为主键。

第二空根据第一空可知, R 的一个主键为 A2A5, 由函数依赖集 F 可知, 存在  $A2A5 \rightarrow A6$ ,  $A5A6 \rightarrow A1$ ,  $A1 \rightarrow A3$ , 这里存在传递函数依赖, 故 A、B 选项均不正确, C 选项本身不正确, 存在非主属性对候选键的传递函数依赖, 是不满足 3NF 的。因此本题选择 D 选项。

(55) (56) 参考答案: B D

试题分析

对于自然连接只要把重复的列去掉就可以, 所以为 5。

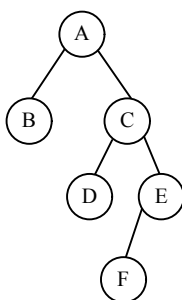
与第二空等价的为 D 选项。

(57) 参考答案: A

试题分析 本题用单链表作为栈的存储结构, 因为栈的操作是先进后出, 因此无论是入栈还是出栈, 都只对栈顶元素操作, 而在单链表中用头指针作为栈顶指针, 此时无论是出栈还是入栈, 都只需要对头指针指向的栈顶指针操作即可, 不需要遍历链表。

(58) 参考答案: B

试题分析



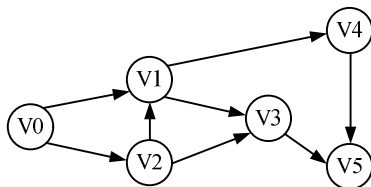
根据先序遍历和中序遍历序列可以构造出二叉树高度为 4, 所以选择 B 选项。

(59) 参考答案: B

试题分析 根据排序二叉树的构造过程可知, A 选项的根节点为 10, D 选项的根节点为 27, 因此可以排除。对于 C 选项, 构造根节点的子结点, 可知 19 为其左孩子结点, 与图不符。本题只有 B 选项可以构造出图示的排序二叉树。

(60) (61) 参考答案: B A

试题分析



(62) ~ (65) 参考答案: A B B C

试题分析 对于第一个空, 本题使用的是分治法。

1) 分治法特征: 对于一个规模为  $n$  的问题, 若该问题可以容易地解决 (比如说规模  $n$  较小) 则直

接解决；否则将其分解为  $k$  个规模较小的子问题，这些子问题互相独立且与原问题形式相同，递归地解这些子问题，然后将各子问题的解合并得到原问题的解。

2) 动态规划法：在求解问题中，对于每一步决策，列出各种可能的局部解，再依据某种判定条件，舍弃那些肯定不能得到最优解的局部解，在每一步都经过筛选，以每一步都是最优解来保证全局是最优解。本题情景没有列出所有的可能解进行筛选，因此，本题不属于动态规划法。

3) 回溯法：回溯法是一种选优搜索法，按选优条件向前搜索，以达到目标。但当搜索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择。这种走不通就退回再走的技术就是回溯法。本题情景没有探索和回退的过程，因此，本题不属于回溯法。

4) 贪心法：总是做出在当前来说是最好的选择，而并不从整体上加以考虑，它所做的每步选择只是当前步骤的局部最优选择，但从整体来说不一定是最优的选择。由于它不必为了寻找最优解而穷尽所有可能解，因此其耗费时间少，一般可以快速得到满意的解，但得不到最优解。在本题情景中，没有给出每步选择的局部最优判断条件，因此，本题不属于贪心法。

舍弃已被覆盖的房子，可以将问题的规模逐步缩小，形成规模较小的子问题，而这些问题的求解与原问题的求解过程相同，因此本题属于分治法的算法思想。

由于本题的算法过程，是依次与各个房子进行判断，当所有房子都被比较之后，则问题结束，因此时间复杂度与房子的个数相关，本问题的时间复杂度应该趋于现象，为  $O(n)$ 。

对于第三个空，关于对应序列 (10, 20, 30, 35, 60, 80, 160, 210, 260, 300)：

第一轮放置：在第一座房子  $x = 10$  的右侧 20m 处安装一个消火栓，可以覆盖 10, 20, 30, 35 这 4 栋房子；

第二轮放置：去掉前 4 栋房子，在第 5 栋房子  $x = 60$  的右侧 20m 处安装一个消火栓，可以覆盖 60、80 这 2 栋房子；

第三轮放置：去掉前面已覆盖的房子，在第 7 栋房子  $x = 160$  的右侧 20m 处安装一个消火栓，只可以覆盖 160 这一栋房子；


第四轮放置：去掉前面已覆盖的房子，在第 8 栋房子  $x = 210$  的右侧 20m 处安装一个消火栓，可以覆盖 210 这一栋房子

第五轮放置：去掉前面已覆盖的房子，在第 9 栋房子  $x = 260$  的右侧 20m 处安装一个消火栓，可以覆盖 260、300 这 2 栋房子；


房子全部覆盖完毕，因此共需安装 5 个消火栓。

对于第四个空，对于得到一个最优解是动态规划的特点，可以得到问题所有的最优解，是回溯法的特征，可以排除 A、B 选项。对于 C、D 选项，C 的语法更为合理一些。

(66) 参考答案：D

 **试题分析** PPPoE (Point-to-Point Protocol Over Ethernet)，以太网上的点对点协议，是将点对点协议 (PPP) 封装在以太网 (Ethernet) 框架中的一种网络隧道协议。提供用户身份验证、用户管理以及数据加密等功能。

(67) 参考答案：A

 **试题分析** netstat 是控制台命令，是一个监控 TCP/IP 网络的非常有用的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态信息。

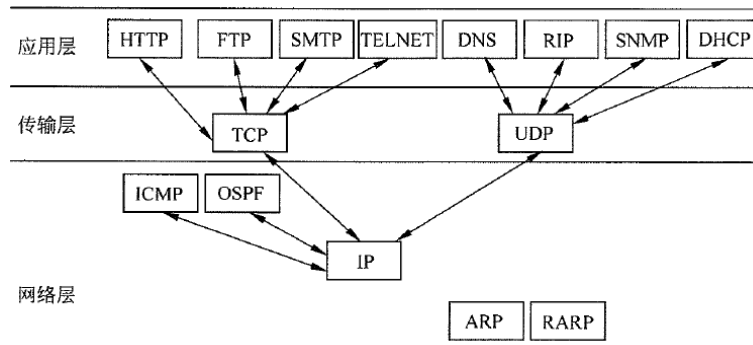
nslookup 可以指定查询的类型，可以查到 DNS 记录的生存时间，还可以指定使用哪个 DNS 服务器进行解释。在已安装 TCP/IP 协议的计算机上面均可以使用这个命令。

利用 “ping” 命令可以检查网络是否连通，可以很好地帮助我们分析和判定网络故障。

tracert (跟踪路由) 是路由跟踪实用程序，用于确定 IP 数据包访问目标所采取的路径。Tracert 命令使用 IP 生存时间 (TTL) 字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由。

(68) 参考答案: A

试题分析



(69) 参考答案: C

试题分析

<!-- -->是 HTML 注释的表示方式, 在这里定义 CSS 样式无效。

(70) 参考答案: C

试题分析

```
选项:
/?          显示此帮助消息。
/all        显示完整配置信息。
/release    释放指定适配器的 IPv4 地址。
/release6   释放指定适配器的 IPv6 地址。
/renew       更新指定适配器的 IPv4 地址。
/renew6      更新指定适配器的 IPv6 地址。
/flushdns   清除 DNS 解析程序缓存。
/registerdns 刷新所有 DHCP 租用并重新注册 DNS 名称。
/displaydns 显示 DNS 解析程序缓存的内容。
/showclassid 显示适配器允许的所有 DHCP 类 ID。
/setclassid  修改 DHCP 类 ID。
/showclassid6 显示适配器允许的所有 IPv6 DHCP 类 ID。
/setclassid6 修改 IPv6 DHCP 类 ID。
```

(71) ~ (75) 参考答案: A B C D C

试题分析

项目工作手册与其说是一个孤立的文档, 倒不如说它是一种加在文档上的 (71), 项目会按照项目工作手册来实施。

项目所有的文档都是项目工作手册的一部分, 其中包括目标、外部说明、接口说明、技术标准、内部说明、以及管理备忘录。

技术文档是其中至关重要的部分。如果客户通过用户手册去搜寻硬件或软件的信息, 他不止能找到简单的描述还能找到关于项目计划与初步设计方面的 (72)。对于技术文档工程师来说糊锅跟笔一样重要。

鉴于项目质量手册出自出自于这些备忘录, 那么建立正确的文档结构尤为重要。早期的项目 (73) 可以确保文档的结构被精心制作而不是被粗制滥造。更进一步说文档结构的建立也给后续的书写工作提供一个模板。

再者, 项目工作手册也能控制 (74) 的分发, 它不是严格限制分发而是把信息传递给需要它的人。

第一步要给所有备忘录进行编号, 形成列表, 工作手册再把这些备忘录建立一个树形结构。如有必要则在此 (75) 基础上形成分发列表的子结构。

- |            |        |         |       |
|------------|--------|---------|-------|
| (71) A. 结构 | B. 说明  | C. 标准   | D. 目标 |
| (72) A. 目标 | B. 备忘录 | C. 标准   | D. 说明 |
| (73) A. 标题 | B. 列表  | C. 工作手册 | D. 质量 |
| (74) A. 产品 | B. 手册  | C. 文档   | D. 信息 |
| (75) A. 列表 | B. 文档  | C. 树形结构 | D. 号码 |

# 全国计算机技术与软件专业技术资格考试

## 2018 年下半年 软件设计师 下午试卷解析

### 试题一

【问题 1】 E1 客户          E2 经纪人          E3 财务人员          E4 外部网站

【问题 2】 D1 客户信息表/客户记录表          D2 潜在房源表

D3 房源表                                  D4 订单记录表

【问题 3】

序号	起点	终点	缺失数据流
1	交易管理 P4	经纪人 E2	交易反馈信息
2	交易管理 P4	房源表 D3	更新房源状态
3	客户信息表 D1	房源推荐 P3	客户信息

【问题 4】

房源信息包括基本情况、配套设施、交易类型、委托方式、主业等。

客户信息包括身份证号、姓名、手机号、需求情况、委托方式等。

### 试题二

【问题 1】

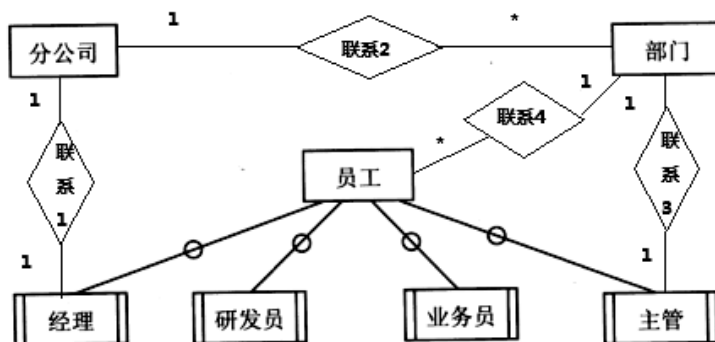


图 2-1 实体联系图

【问题 2】

(a) 经理

(b) 主管号，分公司编号

(c) 隶属部门，岗位

【问题 3】

部门      主键：部门号      外键：分公司编号，主管号

员工      主键：员工号      外键：隶属部门

**【问题 4】**

不需要增加实体。因为它可以直接归属到联系当中，它的联系可以直接写成关系模式，所以不需要增加实体。

**试题三**

**【问题 1】**

C1=SNSGroup      C2=SNSUser      C3=SNSAdmin (C2 可以和 C3 互换位置)

**【问题 2】**

采用了观察者模式。改模式的意图当被观察者发生改变的时候会给观察者发送消息通知，让它们随之发生改变。适用场合一般适用于一个被观察者改变时观察者也随之改变的场合。

**【问题 3】**

可以在 SNSSubject 下面增加一个被观察者对象，然后它可以在观察者对象这里增加一个加入另外群体的一个方法，以实现接收被观察者发送的通知。

**试题四**

**【问题 1】**

- (1)  $\max = C[i][j-1]$
- (2)  $t=0$
- (3)  $\text{isMatch}(B[t], B[j])$
- (4)  $C[i][j-1]$

**【问题 2】**

- (5) 动态规划    (6)  $O(n^3)$

**【问题 3】**

- (7) 2

**试题五**

- (1) `virtual double travel(int miles, FrequentFlyer* context)=0`
- (2) `context->setState(context-> CSilver)`
- (3) `context->setState(context-> Cbasic)`
- (4) `context->setState(context-> CSilver)`
- (5) `context->setState(context-> Cbasic)`

**试题六**

- (1) `abstract double travel(int miles, FrequentFlyer context)`
- (2) `context.setState(new CSilver())`
- (3) `context.setState(new C Gold ())`
- (4) `context.setState(new CSilver())`
- (5) `context.setState(new C Basic ())`