

2016 年下半年

全国计算机技术与软件专业技术资格考试 2016 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

- 2016 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。

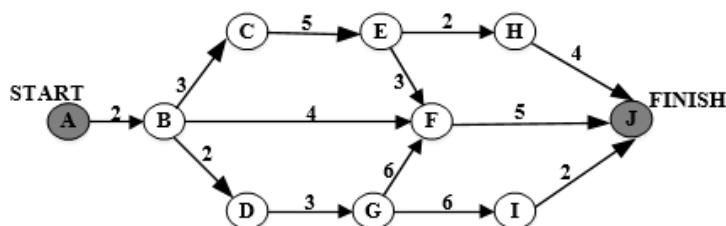
(88) A. 10 B. 11 C. 12 D. 09

(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。

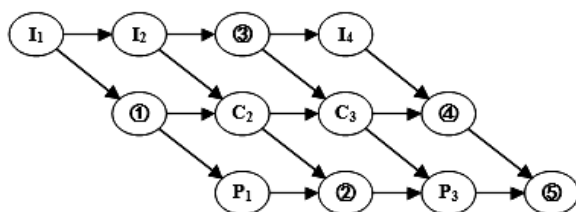
- 在程序运行过程中，CPU 需要将指令从内存中取出并加以分析和执行。CPU 依据 (1) 来区分在内存中以二进制编码形式存放的指令和数据。
- (1) A. 指令周期的不同阶段 B. 指令和数据的寻址方式
C. 指令操作码的译码结果 D. 指令和数据所在的存储单元
- 计算机在一个指令周期的过程中，为从内存读取指令操作码，首先要将 (2) 的内容送到地址总线上。
- (2) A. 指令寄存器 (IR) B. 通用寄存器 (GR)
C. 程序计数器 (PC) D. 状态寄存器 (PSW)
- 设 16 位浮点数，其中阶符 1 位、阶码值 6 位、数符 1 位、尾数 8 位。若阶码用移码表示，尾数用补码表示，则该浮点数所能表示的数值范围是 (3)。
- (3) A. $-2^{64} \sim (1-2^{-8})2^{64}$ B. $-2^{63} \sim (1-2^{-8})2^{63}$
C. $-2^{64} \sim (1-2^{-(1-2^{-8})})2^{64} \sim (1-2^{-8})2^{64}$ D. $-(1-2^{-8})2^{63} \sim (1-2^{-8})2^{63}$
- 已知数据信息为 16 位，最少应附加 (4) 位校验位，以实现海明码纠错。
- (4) A. 3 B. 4 C. 5 D. 6
- 将一条指令的执行过程分解为取指、分析和执行三步，按照流水方式执行，若取指时间 $t_{\text{取指}}=4\Delta t$ 、分析时间 $t_{\text{分析}}=2\Delta t$ 、执行时间 $t_{\text{执行}}=3\Delta t$ ，则执行完 100 条指令，需要的时间为 (5) Δt 。
- (5) A. 200 B. 300 C. 400 D. 405
- 以下关于 Cache 与主存间地址映射的叙述中，正确的是 (6)。
- (6) A. 操作系统负责管理 Cache 与主存之间的地址映射
B. 程序员需要通过编程来处理 Cache 与主存之间的地址映射
C. 应用软件对 Cache 与主存之间的地址映射进行调度
D. 由硬件自动完成 Cache 与主存之间的地址映射
- 可用于数字签名的算法是 (7)。
- (7) A. RSA B. IDEA C. RC4 D. MD5
- (8) 不是数字签名的作用。
- (8) A. 接收者可验证消息来源的真实性 B. 发送者无法否认发送过该消息
C. 接收者无法伪造或篡改消息 D. 可验证接收者合法性
- 在网络设计和实施过程中要采取多种安全措施，其中 (9) 是针对系统安全需求的措施。
- (9) A. 设备防雷击 B. 入侵检测
C. 漏洞发现与补丁管理 D. 流量控制
- (10) 的保护期限是可以延长的。
- (10) A. 专利权 B. 商标权 C. 著作权 D. 商业秘密权
- 甲公司软件设计师完成了一项涉及计算机程序的发明。之后，乙公司软件设计师也完成了与甲公司软件设计师相同的涉及计算机程序的发明。甲、乙公司于同一天向专利局申请发明专利。此情形下，(11) 是专利权申请人。
- (11) A. 甲公司 B. 甲、乙两公司
C. 乙公司 D. 由甲、乙公司协商确定的公司
- 甲、乙两厂生产的产品类似，且产品都使用“B”商标。两厂于同一天向商标局申请商标注册，且申请注册前两厂均未使用“B”商标。此情形下，(12) 能核准注册。
- (12) A. 甲厂 B. 由甲、乙厂抽签确定的厂
C. 乙厂 D. 甲、乙两厂
- 在 FM 方式的数字音乐合成器中，改变数字载波频率可以改变乐音的 (13)，改变它的信号幅度可以改变乐音的 (14)。
- (13) A. 音调 B. 音色 C. 音高 D. 音质
(14) A. 音调 B. 音域 C. 音高 D. 带宽

- 结构化开发方法中，(15) 主要包含对数据结构和算法的设计。
 (15) A. 体系结构设计 B. 数据设计
 C. 接口设计 D. 过程设计
- 在敏捷过程的开发方法中，(16) 使用了迭代的方法，其中，把每段时间（30 天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。
 (16) A. 极限编程 XP B. 水晶法
 C. 并列争球法 D. 自适应软件开发
- 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，边上的数字表示相应活动的持续时间（天），则完成该项目的最少时间为(17)天。活动 BC 和 BF 最多可以晚开始(18)天而不会影响整个项目的进度。



- (17) A. 11 B. 15 C. 16 D. 18
 (18) A. 0 和 7 B. 0 和 11 C. 2 和 7 D. 2 和 11
- 成本估算时，(19) 方法以规模作为成本的主要因素，考虑多个成本驱动因子。该方法包括三个阶段性模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。
 (19) A. 专家估算 B. Wolkerton
 C. COCOMO D. COCOMO II
 - 逻辑表达式求值时常采用短路计算方式。“&&”“||”“!” 分别表示逻辑与、或、非运算，“&&”“||”为左结合，“!” 为右结合，优先级从高到低为“!” “&&” “||”。对逻辑表达式 “x&& (y||!z)” 进行短路计算方式求值时，(20)。
 (20) A. x 为真，则整个表达式的值即为真，不需要计算 y 和 z 的值
 B. x 为假，则整个表达式的值即为假，不需要计算 y 和 z 的值
 C. x 为真，再根据 z 的值决定是否需要计算 y 的值
 D. x 为假，再根据 y 的值决定是否需要计算 z 的值
 - 常用的函数参数传递方式有传值与传引用两种，(21)。
 (21) A. 在传值方式下，形参与实参之间互相传值
 B. 在传值方式下，实参不能是变量
 C. 在传引用方式下，修改形参实质上改变了实参的值
 D. 在传引用方式下，实参可以是任意的变量和表达式
 - 二维数组 $a[1..N, 1..N]$ 可以按行存储或按列存储。对于数组元素 $a[i, j]$ ($1 \leq i, j \leq N$)，当(22) 时，在按行和按列两种存储方式下，其偏移量相同。
 (22) A. $i \neq j$ B. $i = j$ C. $i > j$ D. $i < j$
 - 实时操作系统主要用于有实时要求的过程控制等领域。实时系统对于来自外部的事件必须在(23)。
 (23) A. 一个时间片内进行处理
 B. 一个周转时间内进行处理
 C. 一个机器周期内进行处理
 D. 被控对象规定的时间内做出及时响应并对其进行处理

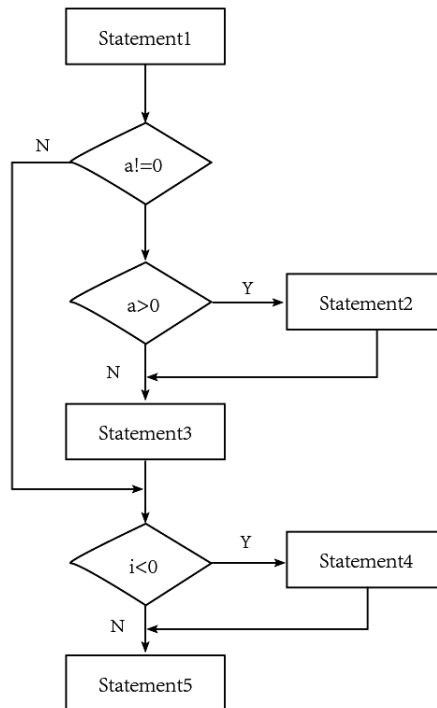
- 假设某计算机系统中只有一个 CPU、一台输入设备和一台输出设备，若系统中有四个作业 T1、T2、T3 和 T4，系统采用优先级调度，且 T1 的优先级>T2 的优先级>T3 的优先级>T4 的优先级。每个作业 T_i 具有三个程序段：输入 I_i 、计算 C_i 和输出 P_i ($i=1, 2, 3, 4$)，其执行顺序为 $I_i \rightarrow C_i \rightarrow P_i$ 。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为 (24)，③、④、⑤分别为 (25)。



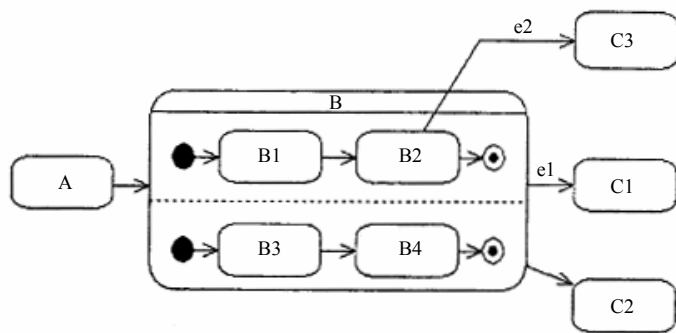
- (24) A. I_2, P_2 B. I_2, C_2
C. C_1, P_2 D. C_1, P_3
- (25) A. C_2, C_4, P_4 B. I_2, I_3, C_4
C. I_3, P_3, P_4 D. I_3, C_4, P_4
- 假设段页式存储管理系统中的地址结构如下图所示，则系统 (26)。

31	24	23	13	12	0
段 号		页 号		页内地址	

- (26) A. 最多可有 256 个段，每个段的大小均为 2048 个页，页的大小为 8K
B. 最多可有 256 个段，每个段最大允许有 2048 个页，页的大小为 8K
C. 最多可有 512 个段，每个段的大小均为 1024 个页，页的大小为 4K
D. 最多可有 512 个段，每个段最大允许有 1024 个页，页的大小为 4K
- 假设系统中有 n 个进程共享 3 台扫描仪，并采用 PV 操作实现进程同步与互斥。若系统信号量 S 的当前值为-1，进程 P_1 、 P_2 又分别执行了 1 次 P (S) 操作，那么信号量 S 的值应为 (27)。
- (27) A. 3 B. -3 C. 1 D. -1
- 某字长为 32 位的计算机的文件管理系统采用位示图 (bitmap) 记录磁盘的使用情况。若磁盘的容量为 300GB，物理块的大小为 1MB，那么位示图的大小为 (28) 个字。
- (28) A. 1200 B. 3200 C. 6400 D. 9600
- 某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不宜采用 (29) 模型，主要是因为这种模型 (30)。
- (29) A. 瀑布 B. 原型 C. 增量 D. 喷泉
(30) A. 不能解决风险 B. 不能快速提交软件
C. 难以适应变化的需求 D. 不能理解用户的需求
- (31) 不属于软件质量特性中的可移植性。
- (31) A. 适应性 B. 易安装性
C. 易替换性 D. 易理解性
- 对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要 (32) 个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为 (33)。
- (32) A. 3 B. 4 C. 6 D. 8
(33) A. 1 B. 2 C. 3 D. 4



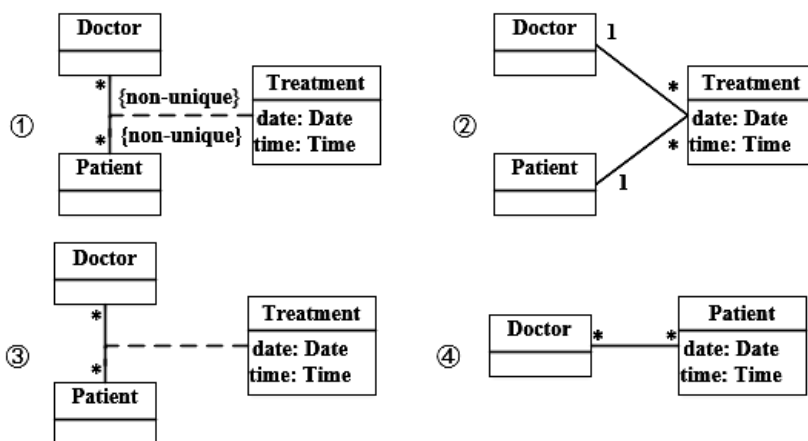
- 计算机系统的 (34) 可以用 $MTBF / (1+MTBF)$ 来度量, 其中 MTBF 为平均失效间隔时间。
 (34) A. 可靠性 B. 可用性 C. 可维护性 D. 健壮性
- 以下关于软件测试的叙述中, 不正确的是 (35)。
 (35) A. 在设计测试用例时应考虑输入数据和预期输出结果
 B. 软件测试的目的是证明软件的正确性
 C. 在设计测试用例时, 应该包括合理的输入条件
 D. 在设计测试用例时, 应该包括不合理的输入条件
- 某模块中有两个处理 A 和 B, 分别对数据结构 X 写数据和读数据, 则该模块的内聚类型为 (36) 内聚。
 (36) A. 逻辑 B. 过程 C. 通信 D. 内容
- 在面向对象方法中, 不同对象收到同一消息可以产生完全不同的结果, 这一现象称为 (37)。在使用时, 用户可以发送一个通用的消息, 而实现的细节则由接收对象自行决定。
 (37) A. 接口 B. 继承 C. 覆盖 D. 多态
- 在面向对象方法中, 支持多态的是 (38)。
 (38) A. 静态分配 B. 动态分配
 C. 静态类型 D. 动态绑定
- 面向对象分析的目的是为了获得对应用问题的理解, 其主要活动不包括 (39)。
 (39) A. 认定并组织对象 B. 描述对象间的相互作用
 C. 面向对象程序设计 D. 确定基于对象的操作
- 如下所示的 UML 状态图中, (40) 时, 不一定会离开状态 B。
 (40) A. 状态 B 中的两个结束状态均达到
 B. 在当前状态为 B2 时, 事件 e2 发生
 C. 事件 e2 发生
 D. 事件 e1 发生



- 以下关于 UML 状态图中转换（transition）的叙述中，不正确的是 （41）。

- （41） A. 活动可以在转换时执行也可以在状态内执行
 B. 监护条件只有在相应的事件发生时才进行检查
 C. 一个转换可以有事件触发器、监护条件和一个状态
 D. 事件触发转换

- 下图①②③④所示是 UML （42）。现有场景：一名医生（Doctor）可以治疗多位病人（Patient），一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时，需要存储治疗（Treatment）的日期和时间。以下①②③④图中 （43） 是描述此场景的模型。



- （42） A. 用例图 B. 对象图 C. 类图 D. 协作图
 （43） A. ① B. ② C. ③ D. ④

- （44） 模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下 （45） 情况适合选用该模式。

- ① 一个客户需要使用一组相关对象 ② 一个对象的改变需要改变其他对象
 ③ 需要使用一个算法的不同变体 ④ 许多相关的类仅仅是行为有异
 （44） A. 命令（Command） B. 责任链（Chain of Responsibility）
 C. 观察者（Observer） D. 策略（Strategy）

- （45） A. ①② B. ②③ C. ③④ D. ①④

- （46） 模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创建不同的表示。以下 （47） 情况适合选用该模式。

- ① 抽象复杂对象的构建步骤
 ② 基于构建过程的具体实现构建复杂对象的不同表示

③ 一个类仅有一个实例

④ 一个类的实例只能有几个不同状态组合中的一种

(46) A. 生成器 (Builder)

B. 工厂方法 (Factory Method)

C. 原型 (Prototype)

D. 单例 (Singleton)

(47) A. ①②

B. ②③

C. ③④

D. ①④

- 由字符 a、b 构成的字符串中, 若每个 a 后至少跟一个 b, 则该字符串集合可用正规式表示为 (48)。

(48) A. $(b|ab)^*$

B. $(ab^*)^*$

C. $(a^*b^*)^*$

D. $(a|b)^*$

- 乔姆斯基 (Chomsky) 将文法分为 4 种类型, 程序设计语言的大多数语法现象可用其中的 (49) 描述。

(49) A. 上下文有关文法

B. 上下文无关文法

C. 正规文法

D. 短语结构文法

- 运行下面的 C 程序代码段, 会出现 (50) 错误。

```
int k=0;
for(;k<100;);
{k++;}
```

(50) A. 变量未定义

B. 静态语义

C. 语法

D. 动态语义

- 在数据库系统中, 一般由 DBA 使用 DBMS 提供的授权功能为不同用户授权, 其主要目的是为了保证数据库的 (51)。

(51) A. 正确性

B. 安全性

C. 一致性

D. 完整性

- 给定关系模式 $R(U, F)$, 其中: U 为关系模式 R 中的属性集, F 是 U 上的一组函数依赖。假设 $U=\{A_1, A_2, A_3, A_4\}$, $F=\{A_1 \rightarrow A_2, A_1 A_2 \rightarrow A_3, A_1 \rightarrow A_4, A_2 \rightarrow A_4\}$, 那么关系 R 的主键应为 (52)。函数依赖集 F 中的 (53) 是冗余的。

(52) A. A_1

B. $A_1 A_2$

C. $A_1 A_3$

D. $A_1 A_2 A_3$

(53) A. $A_1 \rightarrow A_2$

B. $A_1 A_2 \rightarrow A_3$

C. $A_1 \rightarrow A_4$

D. $A_2 \rightarrow A_4$

- 给定关系 $R(A, B, C, D)$ 和关系 $S(A, C, E, F)$, 对其进行自然连接运算 $R \bowtie S$ 后的属性列为 (54) 个; 与 $\sigma_{R.B=S.E}(R \bowtie S)$ 等价的关系代数表达式为 (55)。

(54) A. 4

B. 5

C. 6

D. 8

(55) A. $\sigma_{2>7}(R \times S)$

B. $\pi_{1,2,3,4,7,8}(\sigma_{1=5^2>7^3=6}(R \times S))$

C. $\sigma_{2>7'}(R \times S)$

D. $\pi_{1,2,3,4,7,8}(\sigma_{1=5^2>7'^3=6}(R \times S))$

- 下列查询 $B=$ “大数据” 且 $F=$ “开发平台”, 结果集属性列为 A、B、C、F 的关系代数表达式中, 查询效率最高的是 (56)。

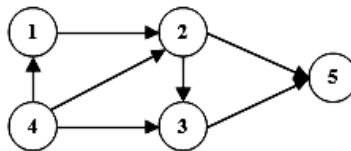
(56) A. $\pi_{1,2,3,8}(\sigma_{2='大数据' \wedge 1=5 \wedge 3=6 \wedge 8='开发平台'}(R \times S))$

B. $\pi_{1,2,3,8}(\sigma_{1=5 \wedge 3=6 \wedge 8='开发平台'}(\sigma_{2='大数据'}(R) \times S))$

C. $\pi_{1,2,3,8}(\sigma_{2='大数据' \wedge 1=5 \wedge 3=6}(R \times \sigma_{4='开发平台'}(S)))$

D. $\pi_{1,2,3,8}(\sigma_{1=5 \wedge 3=6}(\sigma_{2='大数据'}(R) \times \sigma_{4='开发平台'}(S)))$

- 拓扑序列是有向无环图中所有顶点的一个线性序列, 若有向图中存在弧 $\langle v, w \rangle$ 或存在从顶点 v 到 w 的路径, 则在有向图的任一拓扑序列中, v 一定在 w 之前。下面有向图的拓扑序列是 (57)。



(57) A. 41235

B. 43125

C. 42135

D. 41325

- 设有一个包含 n 个元素的有序线性表。在等概率情况下删除其中的一个元素, 若采用顺序存储结构, 则平均需要移动 (58) 个元素; 若采用单链表存储, 则平均需要移动 (59) 个元素。

- (58) A. 1 B. $(n-1)/2$ C. $\text{Log}n$ D. n
 (59) A. 0 B. 1 C. $(n-1)/2$ D. $n/2$

● 具有 3 个节点的二叉树有 (60) 种形态。

- (60) A. 2 B. 3 C. 5 D. 7

● 以下关于二叉排序树（或二叉查找树、二叉搜索树）的叙述中，正确的是 (61)。

- (61) A. 对二叉排序树进行先序、中序和后序遍历，都得到节点关键字的有序序列
 B. 含有 n 个节点的二叉排序树高度为 $\lfloor \log_2 n \rfloor + 1$
 C. 从根到任意一个叶子节点的路径上，节点的关键字呈现有序排列的特点
 D. 从左到右排列同层次的节点，其关键字呈现有序排列的特点

● 下表为某文件中字符的出现频率，采用霍夫曼编码对下列字符编码，则字符序列“bee”的编码为 (62)；编码“110001001101”对应的字符序列为 (63)。

字 符	a	b	c	d	e	f
频率/%	45	13	12	16	9	5

- (62) A. 10111011101 B. 10111001100 C. 001100100 D. 110011011
 (63) A. Bad B. bee C. Face D. bace

● 两个矩阵 $A_{m \times n}$ 和 $B_{n \times p}$ 相乘，用基本的方法进行，则需要的乘法次数为 $m \times n \times p$ 。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定 $M_i, M_{(i+1)}, \dots, M_j$ 多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用 $m[i, j]$ 表示，其递归式定义为

$$m[i, j] = \begin{cases} 0 & i \geq j \\ \min_{i \leq k \leq j} \{m[i, k] + m[k+1, j] + p_{i-1}p_kp_j\} & i < j \end{cases}$$

其中 i, j 和 k 为矩阵下标，矩阵序列中 M_i 的维度为 $(p_{i-1}) \times p_i$ 。采用自底向上的方法实现该算法来确定 n 个矩阵相乘的顺序，其时间复杂度为 (64)。若四个矩阵 M_1, M_2, M_3, M_4 相乘的维度序列为 2、6、3、10、3，采用上述算法求解，则乘法次数为 (65)。

- (64) A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n^3 \lg n)$
 (65) A. 156 B. 144 C. 180 D. 360

● 以下协议中属于应用层协议的是 (66)，该协议的报文封装在 (67)。

- (66) A. SNMP B. ARP C. ICMP D. X.25
 (67) A. TCP B. IP C. UDP D. ICMP

● 某公司内部使用 wb.xyz.com.cn 作为访问某服务器的地址，其中 wb 是 (68)。

- (68) A. 主机名 B. 协议名 C. 目录名 D. 文件名

● 如果路由器收到了多个路由协议转发的关于某个目标的多条路由，那么决定采用哪条路由的策略是 (69)。

- (69) A. 选择与自己路由协议相同的 B. 选择路由费用最小的
 C. 比较各个路由的管理距离 D. 比较各个路由协议的版本

● 与地址 220.112.179.92 匹配的路由表的表项是 (70)。

- (70) A. 220.112.145.32/22 B. 220.112.145.64/22
 C. 220.112.147.64/22 D. 220.112.177.64/22

● 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. stabilize |

全国计算机技术与软件专业技术资格考试

2016 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

2016 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，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，欲开发一证券交易平台，该平台的主要功能如下：

（1）开户。根据客户服务助理提交的开户信息，进行开户，并将客户信息存入客户记录中，账户信息（余额等）存入账户记录中。

（2）存款。客户可以向其账户中存款，根据存款金额修改账户余额。

（3）取款。客户可以从其账户中取款，根据取款金额修改账户余额。

（4）证券交易。客户和经纪人均可进行证券交易（客户通过在线方式，经纪人通过电话），将交易信息存入交易记录中。

（5）检查交易。平台从交易记录中读取交易信息，将交易明细返回给客户。

现采用结构化方法对该证券交易平台进行分析与设计，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。

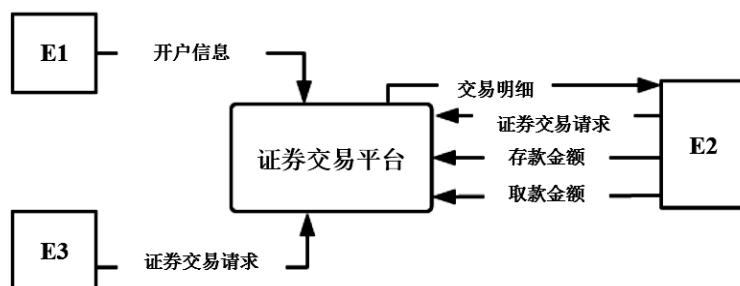


图 1 上下文数据流图

【问题 1】（3 分）

使用说明中的词语，给出图 1 中的实体 E1~E3 的名称。

【问题 2】（3 分）

使用说明中的词语，给出图 2 中的数据存储 D1~D3 的名称。

【问题 3】（4 分）

根据说明和图中的术语，补充图 2 中缺失的数据流及其起点和终点。

【问题 4】（5 分）

实际的证券交易通常是在证券交易中心完成的，因此，该平台的“证券交易”功能需将交易信息传递给证券交易中心。针对这个功能需求，需要对图 1 和图 2 进行哪些修改，请用 200 字以内的文字加以说明。

试题二（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】

某宾馆为了有效地管理客房资源，满足不同客户需求，拟构建一套宾馆信息管理系统，以方便宾馆管理及客房预订等业务活动。

【需求分析结果】

该系统的部分功能及初步需求分析的结果如下：

（1）宾馆有多个部门，部门信息包括部门号、部门名称、电话、经理。每个部门可以有多名员工，每名员工只属于一个部门；每个部门只有一名经理，负责管理本部门。

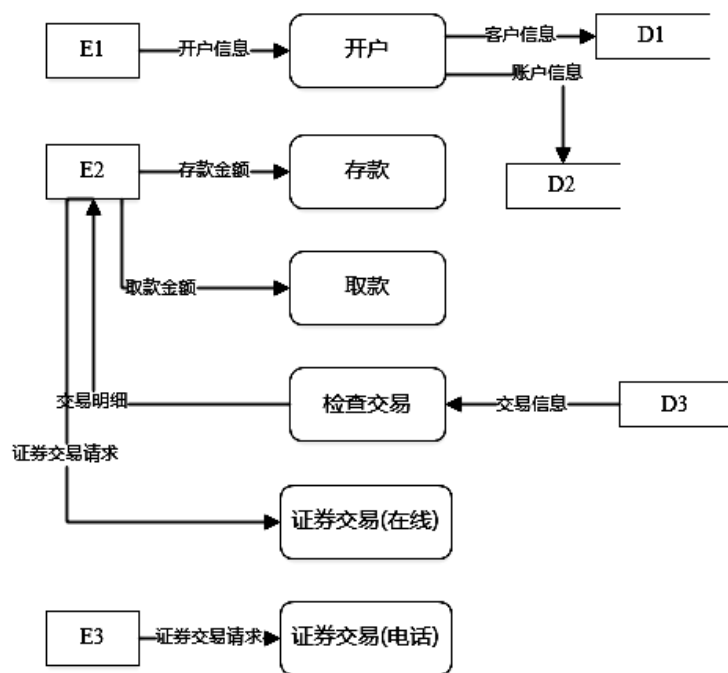


图 2 0 层数据流图

(2) 员工信息包括员工号、姓名、岗位、电话、工资，其中，员工号唯一标识员工关系中的一个元组，岗位有经理、业务员。

(3) 客房信息包括客房号（如 1301、1302 等）、客房类型、收费标准、入住状态（已入住/未入住），其中客房号唯一标识客房关系中的一个元组，不同客房类型具有不同的收费标准。

(4) 客户信息包括客户号、单位名称、联系人、联系电话、联系地址，其中客户号唯一标识客户关系中的一个元组。

(5) 客户预订客房时，需要填写预订申请。预订申请信息包括申请号、客户号、入住时间、入住天数、客房类型、客房数量，其中，一个申请号唯一标识预订申请中的一个元组；一位客户可以有多个预订申请，但一个预订申请对应唯一的一位客户。

(6) 当客户入住时，业务员根据客户的预订申请负责安排入住客房事宜。安排信息包括客房号、姓名、性别、身份证号、入住时间、天数、电话，其中客房号、身份证号和入住时间唯一标识一次安排。一名业务员可以安排多个预订申请，一个预订申请只由一名业务员安排，而且可安排多间同类型的客房。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图如图 3 所示。

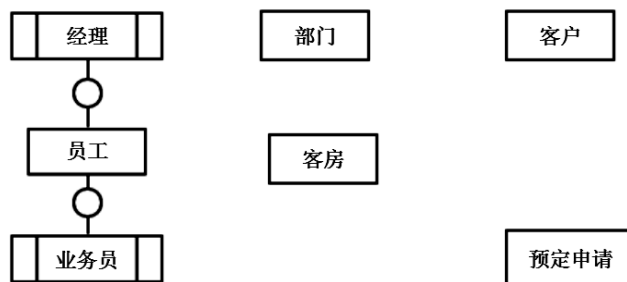


图 3 实体联系图

【关系模式设计】删除了预定申请中的申请号

部门（部门号，部门名称，经理，电话）

员工[员工号，(a)，姓名，岗位，电话，工资]

客户[(b)，联系人，联系电话，联系地址]

客房（客房号，客房类型，收费标准，入住状态）

预订申请[(c)，入住时间，天数，客房类型，客房数量]

安排[申请号，客房号，姓名，性别，(d)，天数，电话，业务员]

【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，联系的类型为 1:1、1:n 和 m:n（或 1:1、1:*和*:*）。

【问题 2】（8 分）

（1）根据题意，将关系模式中的空（a）～（d）补充完整，并填入答题纸对应的位置上。

（2）给出“预订申请”和“安排”关系模式的主键和外键。

【问题 3】（3 分）

【关系模式设计】中的“客房”关系模式是否存在规范性问题，请用 100 字以内文字解释你的观点（若存在问题，应说明如何修改“客房”关系模式）。

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】

某种出售罐装饮料的自动售货机（Vending Machine）的工作过程描述如下：

（1）顾客选择所需购买的饮料及数量。

（2）顾客从投币口向自动售货机中投入硬币（该自动售货机只接收硬币）。硬币器收集投入的硬币并计算其对应的价值。如果所投入的硬币足够购买所需数量的这种饮料且饮料数量足够，则推出饮料，计算找零，顾客取走饮料和找回的硬币；如果投入的硬币不够或者所选购的饮料数量不足，则提示用户继续投入硬币或重新选择饮料及数量。

（3）一次购买结束之后，将硬币器中的硬币移走（清空硬币器），等待下一次交易。自动售货机还有一个退币按钮，用于退还顾客所投入的硬币。已经成功购买饮料的钱是不会被退回的。

现采用面向对象方法分析和设计该自动售货机的软件系统，得到如图 4 所示的用例图，其中，“购买饮料”的用例规约描述如下。

参与者：顾客。

主要事件流：

- 1) 顾客选择需要购买的饮料和数量，投入硬币。
- 2) 自动售货机检查顾客是否投入足够的硬币。
- 3) 自动售货机检查饮料储存仓中所选购的饮料是否足够。
- 4) 自动售货机推出饮料。
- 5) 自动售货机返回找零。

备选事件流：

- 2a. 若投入的硬币不足，则给出提示并退回到 1)。
- 3a. 若所选购的饮料数量不足，则给出提示并退回到 1)。

根据用例“购买饮料”得到自动售货机的 4 个状态：“空闲”状态、“准备服务”状态、“可购买”状态以及“饮料出售”状态，对应的状态图如图 5 所示。

所设计的类图如图 6 所示。

【问题 1】（6 分）

根据说明中的描述，使用说明中的术语，给出图 5 中的 S1～S4 所对应的状态名。

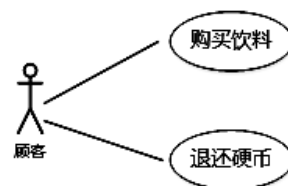


图 4 用例图

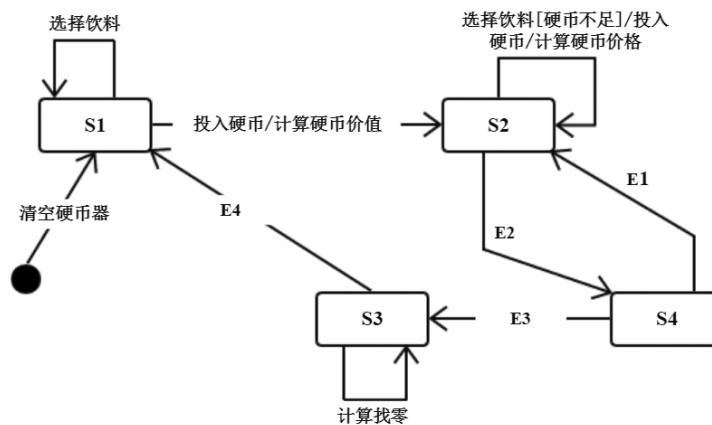


图 5 状态图

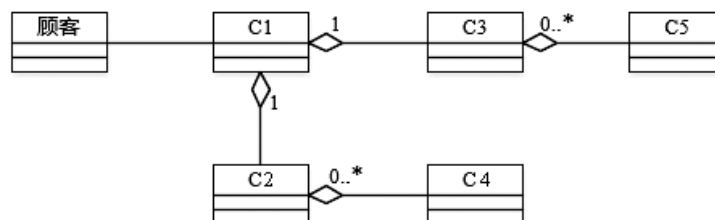


图 6 类图

【问题 2】（4 分）

根据说明中的描述，使用说明中的术语，给出图 5 中的 E1~E4 所对应的事件名。

【问题 3】（5 分）

根据说明中的描述，使用说明中的术语，给出图 6 中 C1~C5 所对应的类名。

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】

模式匹配是指给定主串 t 和子串 s ，在主串 t 中寻找子串 s 的过程，其中 s 称为模式。如果匹配成功，返回 s 在 t 中的位置，否则返回 -1。

KMP 算法用 $next$ 数组对匹配过程进行了优化。KMP 算法的伪代码描述如下：

（1）在串 t 和串 s 中，分别设比较的起始下标 $i=j=0$ 。

（2）如果串 t 和串 s 都还有字符，则循环执行下列操作：

1) 如果 $j=-1$ 或者 $t[i]=s[j]$ ，则将 i 和 j 分别加 1，继续比较 t 和 s 的下一个字符。

2) 否则，将 j 向右滑动到 $next[j]$ 的位置，即 $j=next[j]$ 。

（3）如果 s 中所有字符均已比较完毕，则返回匹配的起始位置（从 1 开始）；否则返回 -1。

其中， $next$ 数组根据子串 s 求解。求解 $next$ 数组的代码已由 get_next 函数给出。

【C 代码】

1. 常量和变量说明

t , s : 长度为 lt 和 ls 的字符串

$next$: $next$ 数组，长度为 ls

2. C 程序

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```

#include <string.h>
//求 next[] 的值
void get_next(int *next, char *s, int ls) {
    int i = 0, j = -1;
    next[0] = -1;    //初始化 next[0]
    while (i < ls) { //还有字符
        if (j == -1 || s[i] == s[j]) { //匹配
            j++;
            i++;
            if (s[i] == s[j])
                next[i] = next[j];
            else
                next[i] = j;
        }
        else
            j = next[j];
    }
}

int kmp(int *next, char *t, char *s, int lt, int ls)
{
    int i = 0, j = 0;
    while (i < lt && (1)) {
        if (j == -1 || (2)) {
            i++;
            j++;
        }
        else
            (3);
    }
    if (j >= ls)
        return (4);
    else
        return -1;
}

```

【问题 1】（8 分）

根据题干说明，填充 C 代码中的空（1）～（4）。

【问题 2】（2 分）

根据题干说明和 C 代码，分析出 KMP 算法的时间复杂度为（5）（主串和子串的长度分别为 lt 和 ls，用 O 符号表示）。

【问题 3】（5 分）

根据 C 代码，字符串“BBABBCAC”的 next 数组元素值为（6）（直接写素值，之间用逗号隔开）。若主串为“AABBCBBABBCACCD”，子串为“BBABBCAC”，则函数 KMP 的返回值是（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】

某发票（Invoice）由抬头（Head）部分、正文部分和脚注（Foot）部分构成。现采用装饰（Decorator）模式实现打印发票的功能，得到如图 7 所示的类图。

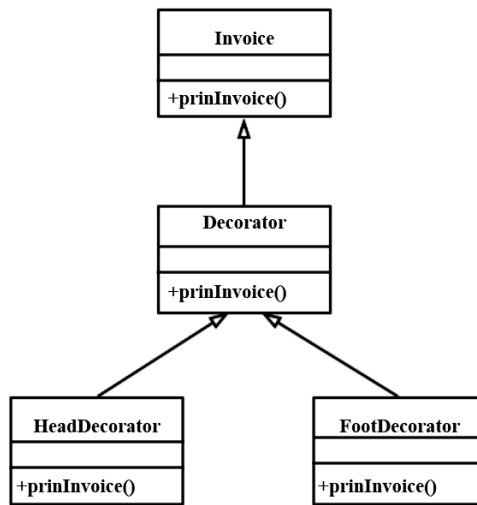


图 7 类图

【C++代码】

```

#include <iostream>
using namespace std;
class Invoice{
public:
    (1) {
        cout << "This is the content of the invoice!" << endl;
    }
};
class Decorator : public Invoice {
    Invoice *ticket;
public:
    Decorator(Invoice *t)      { ticket = t; }
    void printInvoice(){
        if (ticket != NULL)
            (2);
    }
};
class HeadDecorator : public Decorator{
public:
    HeadDecorator(Invoice*t) : Decorator(t) {}
    void printInvoice() {
        cout << "This is the header of the invoice! " << endl;
        (3);
    }
};
class FootDecorator : public Decorator{
public:
    FootDecorator(Invoice *t) : Decorator(t) {}
    void printInvoice(){
        (4);
        cout << "This is the footnote of the invoice!" << endl;
    }
};
int main(void) {

```



```

Invoice t;
FootDecorator f(&t);
HeadDecorator h(&f);
h.printInvoice();
cout << "-----" << endl;
FootDecorator a(NULL);
HeadDecorator b((5));
b.printInvoice();
return 0;
}

```

程序的输出结果为：

```

This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
-----
This is the header of the invoice!
This is the footnote of the invoice!

```

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】

某发票（Invoice）由抬头（Head）部分、正文部分和脚注（Foot）部分构成。现采用装饰（Decorator）模式实现打印发票的功能，得到如图 8 所示的类图。

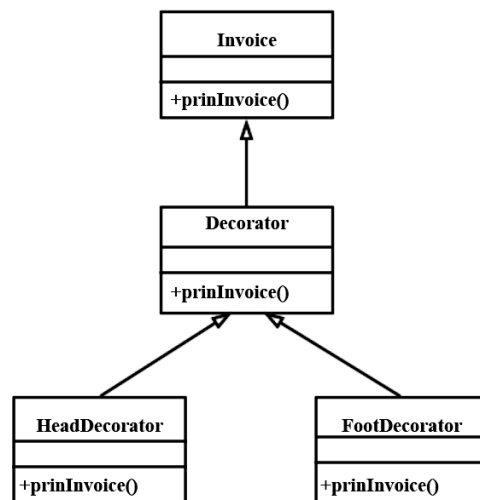


图 8 类图

【Java 代码】

```

class invoice{
    public void printInvoice(){
        System.out.println("This is the content of the invoice!");
    }
}
class Decorator extends invoice {
    protected Invoice ticket;
    public Decorator(Invoice t){
        ticket = t;
    }
}

```

```

    }
    public void printInvoice(){
        if (ticket != null)
            (1);
    }
}
class HeadDecorator extends Decorator{
    public HeadDecorator(Invoice t){
        super(t);
    }
    public void printInvoice(){
        System.out.println("This is the header of the invoice! ");
        (2);
    }
}
class FootDecorator extends Decorator {
    public FootDecorator(Invoice t){
        super(t);
    }
    public void printInvoice(){
        (3);
        System.out.println("This is the footnote of the invoice! ");
    }
}
class test{
    public static void main(String[] args){
        Invoice t = new Invoice();
        Invoice ticket;
        ticket = (4);
        ticket.printInvoice();
        System.out.println("-----");
        ticket = (5);
        ticket.printInvoice();
    }
}

```

程序的输出结果为:

```

This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
-----
This is the header of the invoice!
This is the footnote of the invoice!

```

全国计算机技术与软件专业技术资格考试

2016 年下半年 软件设计师 上午试卷解析

(1) 参考答案: A

☞ 试题分析 指令和数据都存储在内存中, 传统计算机 CPU 在执行过程中根据指令周期的不同阶段来区分是指令还是数据, 取指周期取出的是指令, 执行周期取出的是数据。

(2) 参考答案: C

☞ 试题分析 PC (程序计数器) 是用于存放下一条指令所在单元的地址。当执行一条指令时, 处理器首先需要从 PC 中取出指令在内存中的地址, 通过地址总线寻址获取。

(3) 参考答案: B

☞ 试题分析 尾数的位数越多, 表示的精度越高。

阶码的位数越多, 表示的数值范围越大。

最大正数: $+(1-2^{-M}) \times 2^{(2^R-1)}$, 最小负数 $-1 \times 2^{(2^R-1)}$

M 表示尾数位数, R 表示阶码位数

(4) 参考答案: C

☞ 试题分析 海明码的构造方法是: 在数据位之间插入 k 个校验位, 通过扩大码距来实现检错和纠错。设数据位是 n 位, 校验位是 k 位, 则 n 和 k 必须满足以下的关系。 $2K \geq K+n+1$, 当 $n=16$ 时, K 取 5。

(5) 参考答案: D

☞ 试题分析 计算公式: 第一条指令执行时间+(指令数-1)*各指令段执行时间中最大的执行时间。

$4\Delta t + 3\Delta t + 2\Delta t + (100-1) * 4\Delta t = 405\Delta t$

(6) 参考答案: D

☞ 试题分析 在程序的执行过程中, Cache 与主存的地址映射是由硬件自动完成的。

(7) 参考答案: A

☞ 试题分析 IDEA 算法和 RC4 算法都对称加密算法, 只能用来进行数据加密; MD5 算法是消息摘要算法, 只能用来生成消息摘要, 无法进行数字签名; RSA 算法是典型的非对称加密算法, 主要具有数字签名和验签的功能。

(8) 参考答案: D

☞ 试题分析 数字签名是信息的发送者才能产生的别人无法伪造的一段数字串, 这段数字串同时也是对信息的发送者发送信息真实性的一个有效证明; 不能验证接收者的合法性。

(9) 参考答案: C

☞ 试题分析 在网络设计和实施过程中要采取多种安全措施, 其中漏洞发现与补丁管理是针对系统安全需求的措施, 其他几项都不属于。

(10) 参考答案: B

☞ 试题分析 专利权可分为: 发明专利 (保护期限 20 年), 新型实用设计专利 (保护期限 10 年), 外观设计专利 (保护期限 10 年), 专利期满后专利权终止, 因此专利期限是不可延长的。

商标权 (有效期限 10 年) 到期后可以无限续期, 每次续期有效期还是 10 年, 有效期满未续期的会被注销商标。

著作权保护期限要分开说: 作者的署名权, 修改权和保护作品完整权的保护期限是没有限制的。作品的发表权, 财产权的保护权是作者的终身和死后 50 年, 该权利不可延长。

商业秘密权, 法律上没有规定, 只要商业秘密未泄露出去则一直受法律保护。


(11) 参考答案: D

☞ 试题分析 在审查过程中, 对于不同的申请人同日 (指申请日, 有优先权的指优先权日) 就同


样的发明创造分别提出专利申请，并且这两件申请符合授予专利权的其他条件的，应当根据《专利法》实施细则第四十一条第一款的规定，通知申请人自行协商确定申请人。同一天申请的情况处理方式如下：

- 1、两申请人作为一件申请的共同申请人；
- 2、其中一方放弃权利并从另一方得到适当的补偿。如果双方协商不成的，则两件申请都不授予专利权。

(12) 参考答案：B

 **试题分析** 按照《中华人民共和国商标法》的规定，第 29 条，以及实施条例 19 条规定，同一天申请的，初步审定并公告使用在先的。驳回其他人的申请。均未使用或无法证明的，各自协商，不愿协商或者协商不成的，抽签决定，不抽签的，视为放弃。

(13) (14) 参考答案：A C


 **试题分析** 声音的三个主观属性，即音量（响度）、音调、音色（音品）。

音调主要由声音的频率决定。

音色是声音的特色，根据不同的音色，即使在同一音高和同一声音强度的情况下，也能区分出是不同乐器或人声发出的。

音高即音的高度，是人耳对声音调子高低的主观感觉。主要取决于频率的高低与响度的大小。

(15) 参考答案：D


 **试题分析** 体系结构设计：主要是指要开发的系统中包含哪些部件，这些部件与部件之间的关系就是体系结构的设计。

数据设计：也称为数据库设计，主要包含数据库的设计和这个数据所包含的核心表的设计。

接口设计：用于子系统和模块之间或者内部系统和外部系统之间的各种交互，如功能描述，输入输出的定义，错误处理的设计。

过程设计：也称为模块详细设计，主要是详细模块的实现算法，以及模块所使用的数据结构。


(16) 参考答案：C

 **试题分析** 以上四种方法都属于敏捷开发方法。


极限编程 XP：近似螺旋的开发方法，把整个开发过程分解为相对比较小而简单的周期，通过大家积极的沟通反馈，开发人员和客户都比较清楚当前的开发进度、需要解决的问题等，根据这些实际情况去调整开发过程，这是极限编程的思想。

并列争球法：就是我们通常所说的 Scrum。Scrum 是一个增量、迭代的开发过程。在这个框架中，整个开发过程由若干个短的迭代周期组成，一个短的迭代周期称为一个 Sprint，每个 Sprint 的建议长度是 2 到 4 周。在 Scrum 中，使用产品 Backlog 来管理产品的需求，产品团队总是先开发对客户具有较高价值的需求。挑选的需求在 Sprint 计划会议上经过讨论、分析和估算得到相应的任务列表，我们称它为 Sprint backlog。在每个迭代结束时，Scrum 团队将递交潜在的可交付的产品增量。

(17) (18) 参考答案：D A

 **试题分析** 找出关键路径并计算出关键路径，总工期，各活动总时差，得总工期为 18 天，BC 总时差为 0 天，BF 总时差为 7 天。

(19) 参考答案：D

 **试题分析** 专家估算：根据专家的行业经验和历史数据对软件开发过程的成本进行估算。

Wolverton：也叫 loc 方法，通过执行的源代码行数来进行成本估算，估算准确性低，现在已经不用了。

COCOMO：构造性成本模型，是一种参数化的成本估算方法。例如通过软件的难度、规模等作为参数进行成本估算。

COCOMO II：COCOMO II 是对 COCOMO 作出的改进版，把最新软件开发方法考虑在内。COCOMO II 由三个不同的计算模型组成：

- 1) 应用组合模型：适用于使用现代 GUI 工具开发的项目。
- 2) 早期开发模型：适用于在软件架构确定之前对软件进行粗略的成本和事件估算，包含了一系列

新的成本和进度估算方法。基于功能点或者代码行。

3) 结构化后期模型: 是 COCOMO II 中最详细的模型。它使用在整体软件架构已确定之后。包含最新的成本估算、代码行计算方法。

(20) 参考答案: B

☞**试题分析** 在进行逻辑与“&&”运算时, 只有当两个操作数的值为真, 最后的结果才会为真。因此一旦 x 的值为假, 整个运算表达式的值则为假。

(21) 参考答案: C

☞**试题分析** 传值调用最显著的特征就是被调用的函数内部对形参的修改不影响实参的值。引用调用是将实参的地址传递给形参, 使得形参的地址就是实参的地址。

(22) 参考答案: B

☞**试题分析**

$i \setminus j$	$j=1$	$j=2$	$j=3$
$i=1$	1	2	3
$i=2$	4	5	6
$i=3$	7	8	9

按行存储: 123456789

按列存储: 147258369

可以看到, 当 $i=j$ 时其偏移量相同。

(23) 参考答案: D

☞**试题分析** 实时操作系统是指系统能及时响应外部事件的请求, 在规定的时间内完成对该事件的处理, 并控制所有实时任务协调一致地运行。

(24) (25) 参考答案: C D

☞**试题分析** 题目告诉我们一共有 3 个设备, 分别是一个 CPU、一台输入设备和一台输出设备, 其实输入设备对应程序段输入 I_i , 而 CPU 对应程序段计算 C_i , 输出设备对应程序段输出 P_i 。而每个作业都分为这三段, 各段间有个顺序关系。再结合图中已经给出的节点, 我们不难发现, 第一行是输入, 第二行是计算, 而第三行是节点数输出节点。因此可以知道①、②分别为 C_1 、 P_3 , ③、④、⑤分别为 I_3 、 C_4 、 P_4 。

(26) 参考答案: B

☞**试题分析** 有改动, 页内地址为 13 位, 页号地址为 11 位, 段号地址为 8 位。根据公式, 可以分别计算段号, 页号以及页内地址最大的寻址空间。存储管理系统中的地址长度均表示为最大的寻址空间。

8 位段号, 11 位页号, 13 位页内地址, 所以 $2^8=256$ 段, 每段有 $2^{11}=2048$ 页, 页的大小为 $2^3=8K$ 。

(27) 参考答案: B

☞**试题分析** $P(S)$ 操作是申请资源, 是减量操作; $V(S)$ 操作是释放资源, 是增量操作。所以执行 2 次 $P(S)$ 后 S 值为-3。

(28) 参考答案: D

☞**试题分析** 位示图(bitmap): 一位表示一个物理块的状态, 即 1 位表示 1MB 的使用与否, 300G 的磁盘有 $300 \times 1024=307200$ 个物理块, 需要 307200 位, $307200/32=9600$ 字。(32 位系统字长就是 32 位)。

(29) (30) 参考答案: A C

☞**试题分析** 对于较大型软件系统的需求往往难以在前期确定, 所以瀑布模型最不适合。瀑布模型不适用需求多变或早期需求不确定的开发过程。瀑布模型难以适应变化的需求。

(31) 参考答案: D

☞**试题分析** 软件质量特性有六个: 功能性、可靠性、易用性、效率、维护下、可移植性。其中可移植性包括: 适应性、易安装性、共存性、易替换性、可移植性依存性。

而易理解性是易用性中的一种。

(32)(33) 参考答案: C D

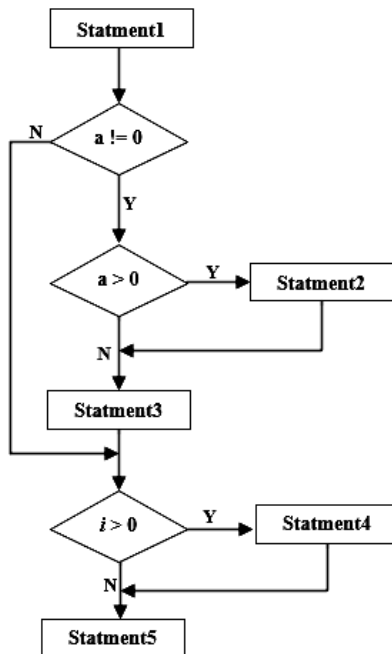
☞ 试题分析 路径覆盖: 使程序中每条路径都至少执行一次。

如图, 每个判断都需要两个用例, 因此需要 6 个测试用例。

有二种方法计算环路复杂度:

1) 环路复杂度 $V(G)=E-N+2$, 其中, E 是流程图中边的条数, N 是节点数。

2) 封闭空间数加 1, 即可得到环路复杂度。



(34) 参考答案: A

☞ 试题分析 MTBF: 称为平均故障间隔时间, 用于衡量产品可靠性的一个指标。

可靠性: 在规定的时间内, 产品保持正常功能的一种能力。

(35) 参考答案: B

☞ 试题分析 测试用例 (Test Case) 是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果, 以便测试某个程序路径或核实是否满足某个特定需求。在设计测试用例时, 应该包括合理的输入条件和不合理的输入条件。

测试的目的是为了发现尽可能多的缺陷, 不是为了说明软件中没有缺陷。

成功的测试在于发现了迄今尚未发现的缺陷。

(36) 参考答案: C

☞ 试题分析 通信内聚: 模块内各功能部分使用了相同的输入数据或产生相同的输出数据。

1) 功能内聚: 指模块内所有元素共同完成某一功能, 联系紧密, 缺一不可, 是最强的内聚类型。

2) 顺序内聚: 指一个模块中各个处理元素都密切相关于同一功能且必须顺序执行, 前一功能元素输出是下一功能元素的输入。即一个模块完成多个功能, 这些模块又必须顺序执行。

3) 通信内聚: 指模块内所有处理元素都在同一个数据结构上操作, 或者指各处理使用相同的输入数据或者产生相同的输出数据。

4) 过程内聚: 构件或者操作的组合方式是, 允许在调用前面的构件或操作之后, 马上调用后面的构件或操作, 即使两者之间没有数据进行传递。

5) 时间内聚: 把需要同时执行的动作组合在一起形成的模块为时间内聚模块, 所有的动作需在同

一个时间段内执行。

6) 逻辑内聚: 把几种相关的功能组合在一起, 每次被调用时, 由传送给模块参数来确定该模块应完成哪一种功能。

7) 偶然内聚: 模块内各部分之间没有联系, 或者有联系, 这种联系也很松散, 是内聚度最低的模块。

(37) 参考答案: D

☞**试题分析** 多态是面向对象程序设计的一个重要特征。指同一操作作用于不同的对象, 可以有不同的解释, 产生不同的执行结果。

继承是指它可以使用现有类的所有功能, 并在无需重新编写原来的类的情况下对这些功能进行扩展。

(38) 参考答案: D

☞**试题分析** 绑定是一个把过程调用和响应调用而需要执行的代码加以结合的过程。在一般程序设计语言中绑定是在编译时进行的, 叫做静态绑定。动态绑定则是在运行时进行的, 因此, 一个给定的过程调用和代码结合直到调用发生时才进行。

动态绑定是和类的继承以及多态相联系的。在继承关系中, 子类是父类的一个特例, 所以父类对象可以出现的地方, 子类也可以出现。因此在运行过程中, 当一个对象发送消息请求服务时, 要根据接受对象的具体情况将请求的操作和实现的方法进行连接, 即动态绑定。

(39) 参考答案: C

☞**试题分析** 面向对象分析包括 5 个活动: 认定对象、组织对象、描述对象之间的相互作用、定义对象的操作、定义对象的内部信息。

(40) 参考答案: C

☞**试题分析** 事件 e2 发生时, 如果当前状态是 B2 状态, 他就会离开状态 B 转入 C3, 如果当前状态不是 B2, 那么 e2 发生时是不会离开 B 状态的。因此事件 e2 发生时不一定会离开状态 B。

(41) 参考答案: C

☞**试题分析** 转移 (Transitions) 是两个状态之间的一种关系, 表示对象将在源状态 (Source State) 中执行一定的动作, 并在某个特定事件在发生而且某个特定的警戒条件满足时进入目标状态 (Target State)。

转移有两个状态: 原状态和目标状态。

(42) (43) 参考答案: C C

☞**试题分析** 类图是最常用的 UML 图, 显示出类、接口以及它们之间的静态结构和关系; 包含 3 个组成部分。类名、属性 (attributes)、类提供的方法。类名不能省略, 其他组成部分可以省略。

对象图描述的是参与交互的各个对象在交互过程中某一时刻的状态。对象图可以被看作是类图在某一时刻的实例。

1) 允许存在多个相同的元素数据, 显然不合理。

2) 根据试题描述, 病人和医生之间应该有关联关系, 显然该图不符合。

3) 符合题目中的描述。

4) 只表达了医生和病人的关系, 没有表达存储治疗的时间和日期, 缺少一个关联类, 因此, 该图也不符合。

(44) (45) 参考答案: D C

☞**试题分析** 命令模式: 将一个请求封装为一个对象, 从而使你可用不同的请求对客户进行参数化。

责任链: 使多个对象都有机会处理请求, 从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链, 并沿着这条链传递该请求, 直到有一个对象处理它为止。

观察者模式: 定义对象间的一种一对多的依赖关系, 当一个对象的状态发生改变时, 所有依赖于它的对象都得到通知并被自动更新。

策略模式: 定义一系列的算法, 把每一个算法封装起来, 并且是它们可互相替换。本模式使得算法

可独立于使用它的客户而变化。

(46) (47) 参考答案: A A

试题分析 建造者模式 (生成器模式): 将一个复杂对象的构建与它的表示分离, 使得同样的构建过程可以创建不同的表示。建造者模式将部件和其组装过程分开, 一步一步创建一个复杂的对象。

工厂方法: 也叫虚拟构造器模式, 它定义一个用于创建对象的接口, 让子类决定实例化哪一个类, 使一个类的实例化延迟到其子类。

原型模式: 用原型实例指定创建对象的种类, 并且通过复制这些原型创建新的对象。

单例模式: 单例模式确保某一个类只有一个实例, 而且自行实例化并向整个系统提供这个实例, 这个类称为单例类, 它提供全局访问的方法。

(48) 参考答案: A

试题分析 正规式中 | 表示或的意思, * 表示 * 前的字符或字符串出现了 0 次或多次。

(49) 参考答案: B

试题分析 文法类型: 0 型、1 型、2 型、3 型。

上下文无关文法: 形式语言理论中一种重要的变化文法, 用来描述上下文无关语言, 在乔姆斯基分层中称为 2 型文法。由于程序设计语言的语法基本上都是上下文无关的文法, 因此应用十分广泛。

(50) 参考答案: D

试题分析 错误可分为静态错误和动态错误两类。

动态错误, 也被称作动态语义错误, 是指源程序中的逻辑错误, 它们发生在程序运行的时候, 如死循环, 变量取值为零。

静态错误又可分为语法错误和静态语义错误。

(51) 参考答案: B

试题分析 DMBS 是数据库管理系统, 主要用来保证数据库的安全性和完整性。而 DBA 通过授权功能为不同用户授权, 主要的目的是为了保证数据的安全性。

(52) (53) 参考答案: A C

试题分析 依据 $U1 = \{A1, A2, A3, A4\}$, $F = \{A1 \rightarrow A2, A1A2 \rightarrow A3, A1 \rightarrow A4, A2 \rightarrow A4\}$ 构造出依赖关系图之后, A1 是入度为 0 的节点, 且从 A1 出发能遍历全图, 因此 A1 为主键。

$A1 \rightarrow A2, A2 \rightarrow A4$ 利用传递率: $A1 \rightarrow A4$, 因此 $A1 \rightarrow A4$ 是冗余。

(54) (55) 参考答案: C B

试题分析 (54) 关系 $R(A, B, C, D)$ 和关系 $S(A, C, E, F)$, 进行 $R \bowtie S$ (自然连接) 后只需将重复的属性给去掉即可, 所以得到有 6 个。

(55) 首先将关系 R 中的元素分别标记为 1 至 4, 关系 S 中的元素标记为 5 至 8, 然后将各选项进行比较可知 B 选项为正确答案。

(56) 参考答案: D

试题分析 π 表示关系代数投影操作。 σ 表示关系代数选择操作。

避免查询一开始就进行笛卡尔积操作, 尽量对数据进行筛选过滤后再进行笛卡尔积操作, 减少参与笛卡尔积运算的数据量。

(57) 参考答案: A

试题分析 有向图的拓扑序列通俗一点来讲, 就是依次遍历没有前驱节点的节点。因为在某一时刻没有前驱节点的节点有可能存在多个, 所以一个图的拓扑排序可能有多个。

4 号节点没有前驱, 所以拓扑排序的第一个元素是 4。当 4 访问完了就可以访问 1, 1 号访问完了就可以访问 2, 2 号访问完了就可以访问 3 或 5。所以拓扑排序的结果为 41235。

(58) (59) 参考答案: B A

试题分析 (58) 顺序表存储, 最好的情况是删除最后一个元素, 此是无需移动任何元素, 最差的情况是删除第一个元素, 此时需要移动 $n-1$ 个元素, 所以平均需要移动 $(n-1)/2$ 。

(59) 链表存储: 直接将需要删除元素的前趋 next 指针指向后继元素结点即可, 无需移动元素, 所

以移动元素个数为 0。

(60) 参考答案: C

☞ 试题分析 N 个节点 ($N \geq 2$) 的二叉树有 $A[N] = \sum_{M=0}^{N-1} (A[M] * A[N-M-1])$ 种形态。

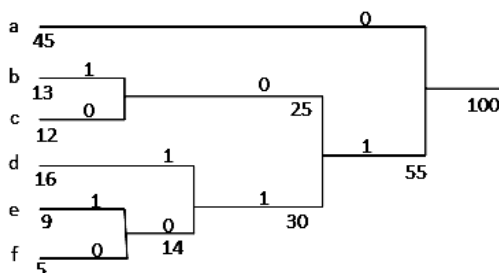
(61) 参考答案: D

☞ 试题分析 二叉排序树又称为二叉查找树。或者是一颗空树, 或者是具有下列性质的二叉树:

- 1) 若左子树不空, 则左子树上所有节点的值均小于它的根节点的值。
- 2) 若右子树不为空, 则右子树上所有的节点的值均大于或等于它的根节点的值。
- 3) 左、右子树也分别为二叉排序树。

(62) (63) 参考答案: A C

☞ 试题分析 考察构造霍夫曼编码树:



如上图所示, 霍夫曼编码是依次把概率最小的两个元素分别置 0 (更小的为 0) 和 1 (相对大的为 2), 然后从右往左写出 1/0 序列。即:

f:1100 e:1101 d:111 c:100 b:101 a:0

(64) (65) 参考答案: C B

☞ 试题分析

方法一, 矩阵链乘法:

一个给定的矩阵序列 $A_1 A_2 \dots A_n$ 计算连乘乘积, 有不同的结合方法, 并且在结合时, 矩阵的相对位置不能改变, 只能相邻结合。根据矩阵乘法的公式, 10×100 和 100×5 的矩阵相乘需要做 $10 \times 100 \times 5$ 次标量乘法。那么对于维数分别为 10×100 、 100×5 、 5×50 的矩阵 A 、 B 、 C , 用 $(A * B) * C$ 来计算需要 $10 \times 100 \times 5 + 10 \times 5 \times 50 = 7500$ 次标量乘法; 而 $A * (B * C)$ 则需要 $100 \times 5 \times 50 + 10 \times 100 \times 50 = 75000$ 次标量乘法。

四个矩阵分别为:

2×6 6×3 3×10 10×3

先计算: $M_1 * M_2$ 以及 $M_3 * M_4$, 计算次数分别为: $2 \times 6 \times 3 = 36$, $3 \times 10 \times 3 = 90$ 。

然后结果相乘, 计算次数为: $2 \times 3 \times 3 = 18$ 。

方法二, 四个矩阵分别为:

2×6 6×3 3×10 10×3

先计算: $M_1 * M_2$ 及 $M_3 * M_4$, 计算次数分别为: $2 \times 6 \times 3 = 36$, $3 \times 10 \times 3 = 90$ 。

然后结果相乘, 计算次数为: $2 \times 3 \times 3 = 18$ 。

(66) (67) 参考答案: A C

☞ 试题分析 ARP 和 ICMP 是网络层协议, X.25 是标准的接口协议, 只有 SNMP 是应用层协议。SNMP 协议的报文是封装在 UDP 协议中传送。

(68) 参考答案: A

☞ 试题分析 URL 由三部分组成: 协议类型, 域名, 路径及文件名

(69) 参考答案: C

☞ 试题分析 管理距离是各种路由协议的优先权, 当多个路由协议转发的关于某个目标的多条路由, 使用管理距离选择权限最高的路由, 管理距离小的优先级最高。

路由费用是一种路由协议对于到达目标网络的各种可能路径的成本衡量。

(70) 参考答案: D

☞ 试题分析 因为都是/22, 我们只看前 22 位即可, 又因为选项中前两个数(16 位)一样, 所以我们只对比第三个数, 将第三个数转换位二进制, 对比其前 6 位。

179 → 1011 0011

145 → 1001 0001

147 → 1001 0011

177 → 1011 0001

22 位掩码, 除去前 2 组 16 位, 还剩余 6 位, 从左向右匹配 177 一致。

(71) ~ (75) 参考答案: C A B D C

☞ 试题分析 翻译:

软件实体比人类构造出的其他实体要复杂得多, 因为软件实体没有两个部分是相同的(至少就表现层面来说)。如果有相同的, 则我们将这两个部分合成一个, 即一个 (71), 统一打开或关闭。在这方面, 软件系统与计算机、建筑物或汽车等系统有很大的不同, 在这些系统中, 重复的元素比比皆是。

数字计算机本身比人们建造的大多数东西都要复杂; 它们有很多种状态。这使得构思、描述和测试工作变得很难。而软件系统的 (72) 比整个计算机系统都要多。

同样地, 一个软件实体的扩展不仅仅是大规模地重复使用同样的元素, 增加不同元素的数量也是非常必要的。在大多数情况下, 这些元素以某种 (73) 的方式相互作用, 整个过程的复杂程度是非线性的。

软件的复杂性是一个 (74) 属性, 而不是一个偶然的属性。因此, 抽走了复杂性的软件实体也没有了本质精髓。近三个世纪以来, 数学和物理科学取得了巨大的进步, 可以把复杂问题简化成简单的模型, 再从模型中得到一些特性, 并通过实验验证这些特性。这种方式之所以有效是因为模型中 (75) 的复杂性并不是本质精髓, 也没有什么作用。

软件开发中的许多常见问题都源自于这种本质复杂度, 并且这些问题都呈非线性增长。并且不仅是技术问题出自于本质复杂度, 管理问题也是如此。

- | | | | |
|-------------|---------|---------|--------|
| (71) A. 任务 | B. 工作 | C. 子程序 | D. 程序 |
| (72) A. 状态 | B. 部分 | C. 条件 | D. 表示 |
| (73) A. 线性 | B. 非线性的 | C. 平行的 | D. 附加的 |
| (74) A. 表面的 | B. 外部的 | C. 内部的 | D. 本质的 |
| (75) A. 固定的 | B. 包含的 | C. 被忽视的 | D. 使稳定 |

全国计算机技术与软件专业技术资格考试

2016 年下半年 软件设计师 下午试卷解析

试题一

【问题 1】

E1: 客户服务助理; E2: 客户; E3: 经纪人。

【问题 2】

D1: 客户记录; D2: 账户记录; D3: 交易记录。

【问题 3】

数据流名称: 修改账户余额, 起点: 存款, 终点: D2。

数据流名称: 修改账户余额, 起点: 取款, 终点: D2。

数据流名称: 交易信息存入交易记录 (在线), 起点: 证券交易 (在线), 终点: D3。

数据流名称: 交易信息存入交易记录 (电话), 起点: 证券交易 (电话), 终点: D3。

【问题 4】

图 1-1 增加外部实体“证券交易中心”, 增加“证券交易平台”到“证券交易中心”, 数据流: 交易信息。

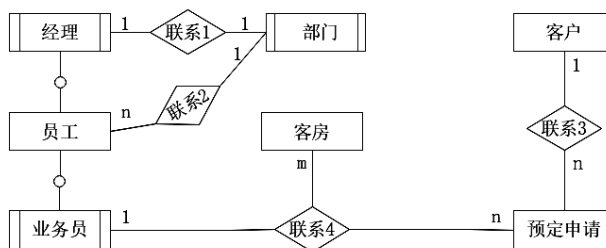
图 1-2 增加外部实体“证券交易中心”, 增加“证券交易 (在线)”到“证券交易中心”, 数据流: 交易信息。

图 1-2 增加“证券交易 (电话)”到“证券交易中心”, 数据流: 交易信息。

试题二

【问题 1】

- (1) 经理与部门之间存在 1:1 的联系。
- (2) 部门与员工之间存在 1:n 的联系。
- (3) 客户与预订申请之间存在 1:n 的联系。
- (4) 业务员、客房、预订申请之间存在 1:m:n 的联系。



【问题 2】

(1) (a) 部门号; (b) 客户号、单位名称; (c) 申请号、客户号; (d) 身份证号、入住时间。

(2) “预订申请”关系模式中的主键是申请号, 外键是客户号。

“安排”关系模式中的主键是: 客房号、身份证号、入住时间, 外键是: 申请号、客房号、业务员。

【问题 3】

根据试题中的描述, 客房信息中客房号是唯一标识客房关系的一个元组, 即可以作为唯一的主键。在客房关系模式中, 不存在其他部分依赖关系, 但客户号→类型→收费标准, 存在传递函数依赖, 所以

冗余，添加异常，修改异常，删除异常均存在。

试题三

【问题 1】

S1：空闲；S2：准备服务；S3：饮料出售；S4：可购买。

【问题 2】

E1：饮料数量不足；E2：硬币数量足够；E3：推出饮料；E4：返回找零。

【问题 3】

C1：自动售货机；C2：硬币器；C3：饮料储存仓；C4：硬币；C5：饮料。

试题四

【问题 1】

- (1): $j < ls$;
- (2): $t[i] == s[j]$;
- (3): `get_next(next, s, ls);`
`j = next[j];`
- (4): $i + 1 - ls$;

【问题 2】

- (5) $O(ls + lt)$

【问题 3】

- (6) $[-1, -1, 1, -1, -1, 2, 0, 0]$
- (7) 6

试题五

参考答案:

- (1) `virtual void printInvoice()`
- (2) `ticket->printInvoice()`
- (3) `Decorator::printInvoice()`
- (4) `Decorator::printInvoice()`
- (5) `&a`

试题六

参考答案:

- (1) `ticket.printInvoice()`
- (2) `ticket.printInvoice()`
- (3) `ticket.printInvoice()`
- (4) `new FootDecorator(new HeadDecorator(t))`
- (5) `new FootDecorator(new HeadDecorator(new Decorator(null)))`