

希赛网, 专注于软考、PMP、通信考试的专业 IT 知识库和在线教育平台。希赛网在线题库, 提供历年考试真题、模拟试题、章节练习、知识点练习、错题本练习等在线做题服务, 更有能力评估报告, 让你告别盲目做题, 针对性地攻破自己的薄弱点, 更高效的备考。

希赛网官网: <http://www.educity.cn/>

希赛网软件水平考试网: <http://www.educity.cn/rk/>

希赛网在线题库: <http://www.educity.cn/tiku/>

2016 年下半年软设综合知识真题答案与解析: <http://www.educity.cn/tiku/tp19675.html>

## 2016 年下半年软件设计师考试上午真题 (参考答案)

- 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 ( ) 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 ( ) 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) A.  $-2^{64} \sim (1-2^{-8}) 2^{64}$
  - B.  $-2^{63} \sim (1-2^{-8}) 2^{63}$
  - C.  $-(1-2^{-8}) 2^{64} \sim (1-2^{-8}) 2^{64}$
  - 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 ( ) 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 ( )  $\Delta t$ 。
  - (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) 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) A. RSA
  - B. IDEA
  - C. RC4
  - D. MD5
  
- ( ) 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 ( ) 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
  
- ( ) 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，（ ）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，（ ）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（ ），改变它的信号幅度可以改变乐音的（ ）。

- (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

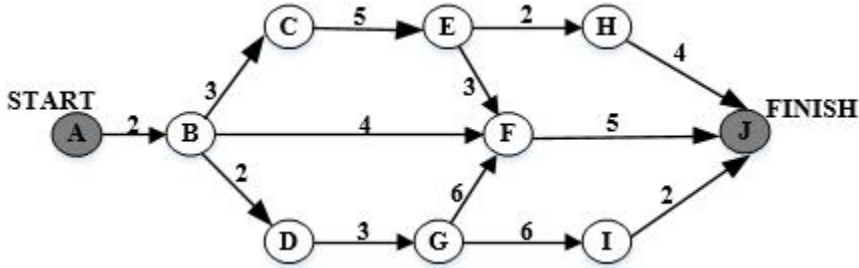
● 结构化开发方法中，（ ）主要包含对数据结构和算法的设计。

- (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，（ ）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（ ）天。活动 BC 和 BF 最多可以晚开始（ ）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，（ ）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. WOLVERTON  
 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) 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 的值决定是否计算 y 的值  
 D. x 为假，再根据 y 的值决定是否计算 z 的值

- 常用的函数参数传递方式有传值与传引用两种。( )。

- (21) 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。

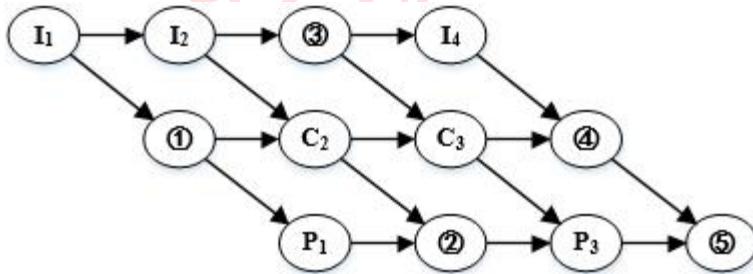
- 二维数组  $a[1..N, 1..N]$  可以按行存储或按列存储。对于数组元素  $a[i,j]$  ( $1 \leq i, j \leq N$ ), 当 ( ) 时, 在按行和按列两种存储方式下, 其偏移量相同。

- (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) 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

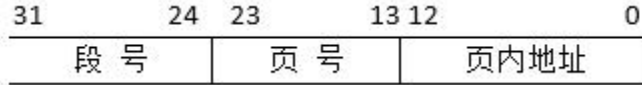
- 假设某计算机系统中只有一个 CPU、一台输入设备和一台输出设备, 若系统中有四个作业  $T_1$ 、 $T_2$ 、 $T_3$  和  $T_4$ , 系统采用优先级调度, 且  $T_1$  的优先级  $> T_2$  的优先级  $> T_3$  的优先级  $> T_4$  的优先级。每个作业  $T_i$  具有三个程序段: 输入  $I_i$ 、计算  $C_i$  和输出  $P_i$  ( $i=1, 2, 3, 4$ ), 其执行顺序为  $I_i \rightarrow C_i \rightarrow P_i$ 。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为 ( ), ③、④、⑤分别为 ( )。



- (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<sub>2</sub>、I<sub>3</sub>、C<sub>4</sub>
- C. I<sub>3</sub>、P<sub>3</sub>、P<sub>4</sub>
- D. I<sub>3</sub>、C<sub>4</sub>、P<sub>4</sub>

- 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) 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<sub>1</sub>、P<sub>2</sub> 又分别执行了 1 次 P(S) 操作，那么信号量 S 的值应为 ( )。

- (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 ( ) 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 ( ) 模型，主要是因为这种模型 ( )。

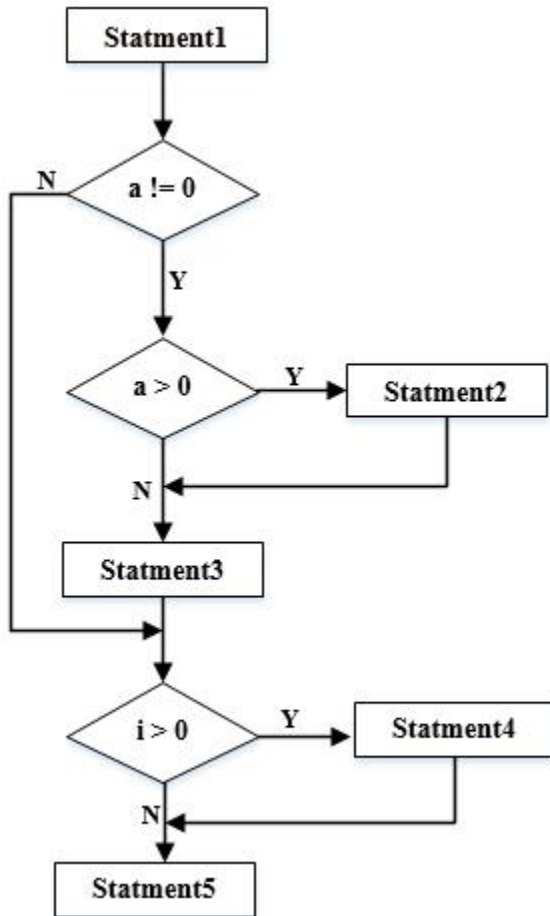
- (29) A. 瀑布
- B. 原型
- C. 增量
- D. 喷泉

- (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

- ( ) 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 ( ) 个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为 ( )。



- (32) A. 3  
B. 4  
C. 6  
D. 8

- (33) A. 1  
B. 2  
C. 3  
D. 4

- 计算机系统的 ( ) 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) 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 ( ) 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 ( )。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 ( )。

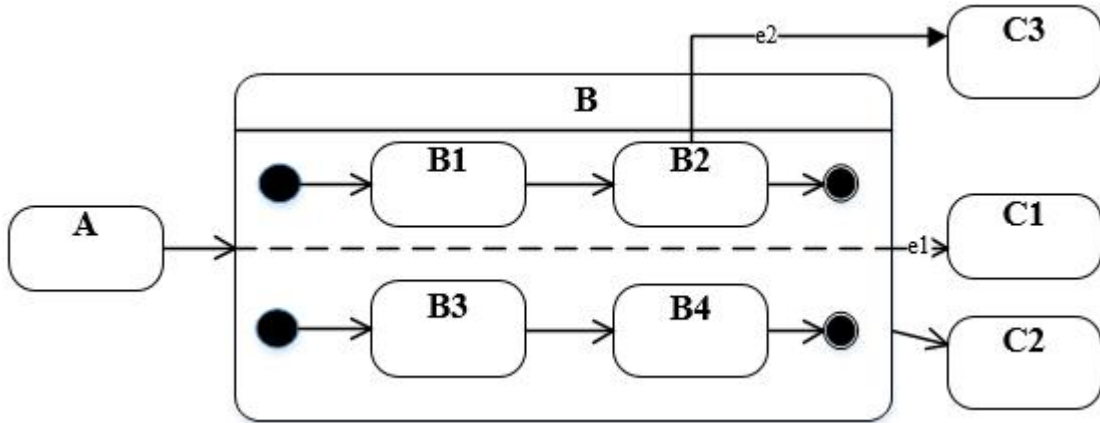
(38) A. 静态分配  
B. 动态分配  
C. 静态类型  
D. 动态绑定

- 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) 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, ( ) 时, 不一定会离开状态 B

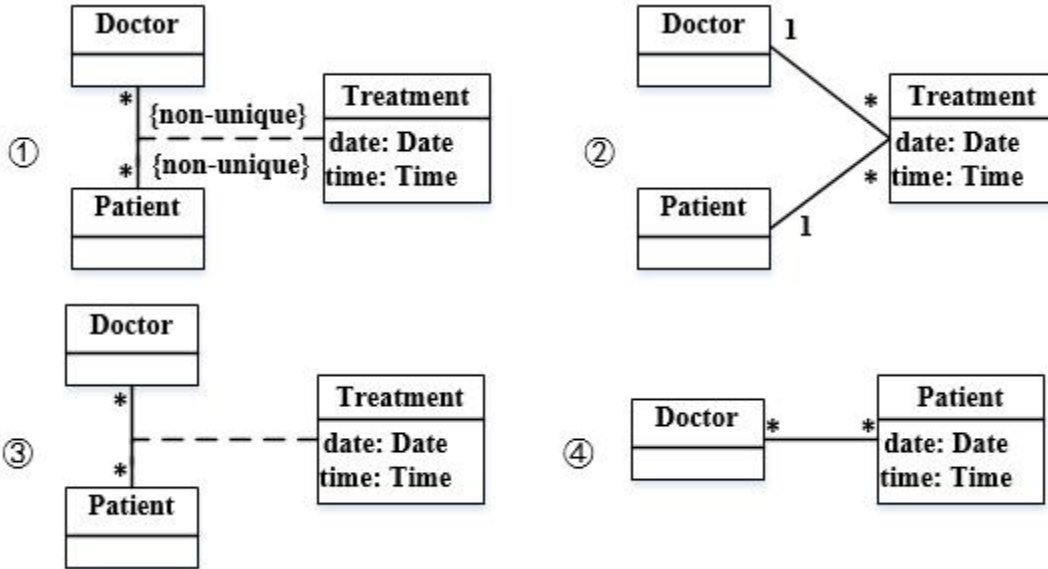


- (40) A. 状态 B 中的两个结束状态均达到
- B. 在当前状态为 B2 时, 事件 e2 发生
- C. 事件 e2 发生
- D. 事件 e1 发生

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

- (41) 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 ( )。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 ( )。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

● ( ) 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 ( ) 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它对象
- 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

● ( ) 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 ( ) 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) 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 ( ) 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 ( ) 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) 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 ( )。函数依赖集 F 中的 ( ) 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.  $A_2 \rightarrow A_4$

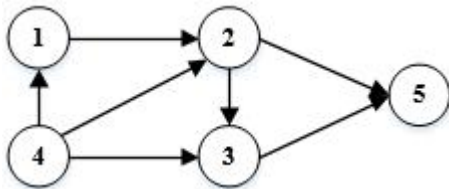
● 给定关系  $R(A, B, C, D)$  和关系  $S(A, C, E, F)$ ，对其进行自然连接运算  $R \bowtie S$  后的属性列为 ( ) 个；与  $\sigma_{R.B > S.E}(R \bowtie S)$  等价的关系代数表达式为 ( )。

- (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) 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，则在该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) 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（ ）个元素。

- (58) A. 1  
 B.  $(n-1)/2$   
 C.  $\log n$   
 D.  $n$

- (59) A. 0  
 B. 1  
 B.  $(n-1)/2$   
 D.  $n/2$

● 具有 3 个节点的二叉树有（ ）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) A. 对二叉排序树进行先序、中序和后序遍历，都得到结点关键字的有序序列  
 B. 含有  $n$  个结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结点的路径上，结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结点，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（ ）；编码“110001001101”的对应的字符序列为（ ）。

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:

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

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 ( )。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 ( )。

- (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 ( ), 该协议的报文封装在 ( )。

- (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) 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) 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) 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 ( ), 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 ( ) 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 ( ) fashion, and the complexity of the whole increases much more than linearly.

The complexity of software is a(an) ( ) 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 ( ) 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. stabilized

## 2017年上半年软件设计师考试上午真题（参考答案）

- CPU 执行算术运算或者逻辑运算时，常将源操作数和结果暂存在（ ）中。

- (1) A. 程序计数器 (PC)  
B. 累加器 (AC)  
C. 指令寄存器 (IR)  
D. 地址寄存器 (AR)

- 要判断字长为 16 位的整数 a 的低四位是否全为 0，则\_\_ (2) \_\_

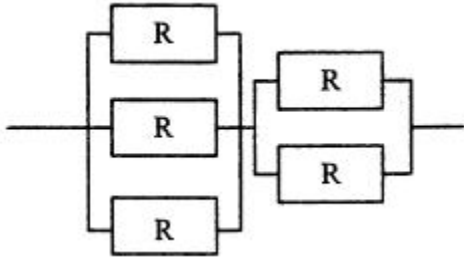
- (2) A. 将 a 与 0x000F 进行"逻辑与"运算，然后判断运算结果是否等于 0  
B. 将 a 与 0x000F 进行"逻辑或"运算，然后判断运算结果是否等于 F  
C. 将 a 与 0x000F 进行"逻辑异或"运算，然后判断运算结果是否等于 0  
D. 将 a 与 0x000F 进行"逻辑与"运算，然后判断运算结果是否等于 F

- 计算机系统中常用的输入/输出控制方式有无条件传送、中断、程序查询和 DMA 方式等。当采用\_\_ (3) \_\_方式时，不需要 CPU 执行程序指令来传送数据。

- (3) A. 中断  
B. 程序查询  
C. 无条件传送  
D. DMA

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





- (4) A.  $(1-R^3)(1-R^2)$   
B.  $(1-(1-R)^3)(1-(1-R)^2)$   
C.  $(1-R^3)+(1-R^2)$   
D.  $(1-(1-R)^3)+(1-(1-R)^2)$
- 已知数据信息为 16 位, 最少应附加\_\_(5)\_\_位校验位, 才能实现海明码纠错。
- (5) A. 3  
B. 4  
C. 5  
D. 6
- 以下关于 Cache (高速缓冲存储器)的叙述中, 不正确的是\_\_(6)\_\_
- (6) A. Cache 的设置扩大了主存的容量  
B. Cache 的内容是主存部分内容的拷贝  
C. Cache 的命中率并不随其容量增大线性地提高  
D. Cache 位于主存与 CPU 之间
  - HTTPS 使用\_\_(7)\_\_协议对报文进行封装

(7) A. SSH  
B. SSL  
C. SHA-1  
D. SET
  - 以下加密算法中适合对大量的明文消息进行加密传输的是 ( )

(8) A. RSA  
B. SHA-1  
C. MD5  
D. RC5
  - 假定用户 A、B 分别在  $I_1$  和  $I_2$  两个 CA 处取得了各自的证书, 下面\_\_(9)\_\_是 A、B 互信的必要条件。

- (9) A. A、B 互换私钥  
B.A、B 互换公钥  
C.I<sub>1</sub>、I<sub>2</sub> 互换私钥  
D.I<sub>1</sub>、I<sub>2</sub> 互换公钥

● 甲软件公司受乙企业委托安排公司软件设计师开发了信息系统管理软件，由于在委托开发合同中未对软件著作权归属作出明确的约定，所以该信息系统管理软件的著作权由( )享有。

- (10) A. 甲  
B.乙  
C.甲与乙共同  
D.软件设计师

● 根据我国商标法，下列商品中必须使用注册商标的是( )

- (11) A. 医疗仪器  
B.墙壁涂料  
C.无糖食品  
D.烟草制品

● 甲、乙两人在同一天就同样的发明创造提交了专利申请，专利局将分别向各申请人通报有关情况，并提出多种可能采用的解决办法。下列说法中，不可能采用( )

- (12) A. 甲、乙作为共同申请人  
B.甲或乙一方放弃权利并从另一方得到适当的补偿  
C. 甲、乙都不授予专利权  
D. 甲、乙都授予专利权

● 数字语音的采样频率定义为 8kHz，这是因为( )

- (13) A. 语音信号定义的频率最高值为 4kHz  
B.语音信号定义的频率最高值为 8kHz  
C.数字语音传输线路的带宽只有 8kHz  
D.一般声卡的采样频率最高为每秒 8k 次

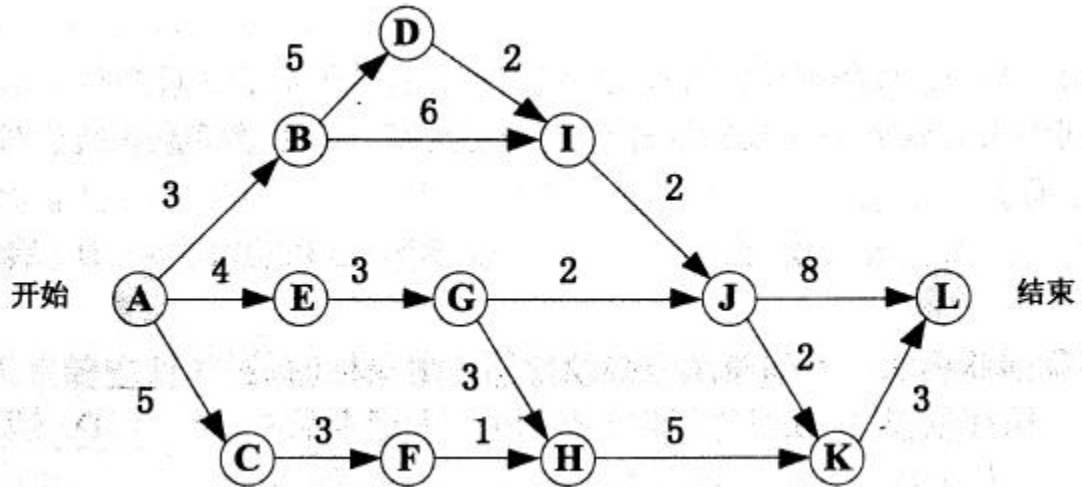
● 使用图像扫描仪以 300DPI 的分辨率扫描一幅 3×4 英寸的图片，可以得到( )像素的数字图像。

- (14) A. 300×300  
B.300×400  
C.900×4  
D.900×1200

● 在采用结构化开发方法进行软件开发时, 设计阶段接口设计主要依据需求分析阶段的( )。接口设计的任务主要是( )。

- (15) A. 数据流图  
 B.E-R 图  
 C.状态-迁移图  
 D.加工规格说明
- (16) A. 定义软件的主要结构元素及其之间的关系  
 B.确定软件涉及的文件系统的结构及数据库的表结构  
 C.描述软件与外部环境之间的交互关系, 软件内模块之间的调用关系  
 D.确定软件各个模块内部的算法和数据结构

● 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( )天。活动 BD 和 HK 最早可以从第( )天开始。(活动 AB、AE 和 AC 最早从第 1 天开始)



- (17) A. 17  
 B.18  
 C.19  
 D.20
- (18) A. 3 和 10  
 B.4 和 11  
 C.3 和 9  
 D.4 和 10

● 在进行软件开发时, 采用无程序员的开发小组, 成员之间相互平等;而主程序员负责制的开发小组, 由一个主程序员和若干成员组成, 成员之间没有沟通。在一个由 8 名开发人员构成的小组中, 无程序员组和主程序员组的沟通路径分别是\_\_(19)\_\_\_。

- (19) A. 32 和 8

- B.32 和 7
- C.28 和 8
- D.28 和 7

- 在高级语言源程序中, 常需要用户定义的标识符为程序中的对象命名, 常见的命名对象有( )  
①关键字 (或保留字) ②变量③函数④数据类型⑤注释

- (20) A. ①②③  
B.②③④  
C.①③⑤  
D.②④⑤

- 在仅由字符 a、b 构成的所有字符串中, 其中以 b 结尾的字符串集合可用正规式表示为( )

- (21) A. (b|ab)\*b  
B.(ab\*)\*b  
C.a\*b\*b  
D.(a|b)\*b

- 在以阶段划分的编译过程中, 判断程序语句的形式是否正确属于\_\_(22)\_\_阶段的工作。

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

- 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。若计算机系统的字长为 32 位, 磁盘的容量为 300GB, 物理块的大小为 4MB, 那么位示图的大小需要( )个字。

- (23) A. 1200  
B.2400  
C.6400  
D.9600

- 某系统中有 3 个并发进程竞争资源 R, 每个进程都需要 5 个 R, 那么至少有\_\_(24)\_\_个 R, 才能保证系统不会发生死锁。

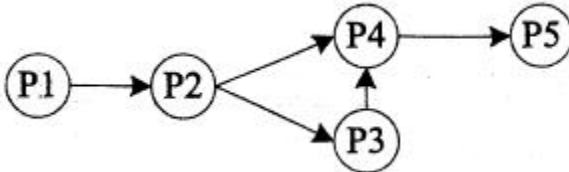
- (24) A. 12  
B.13  
C.14  
D.15

- 某计算机系统页面大小为 4K, 进程的页面变换表如下所示。若进程的逻辑地址为 2D16H。该地址经过变换后, 其物理地址应为( )

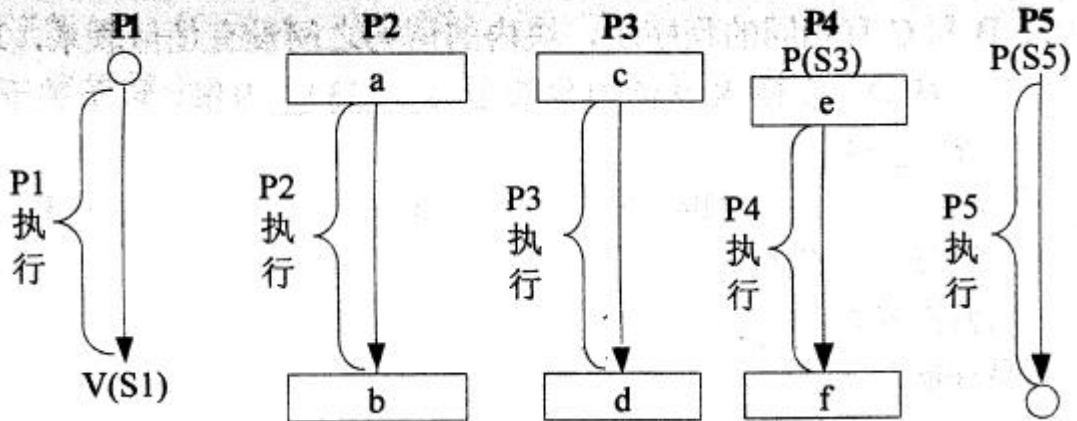
页号	物理块号
0	1
1	3
2	4
3	6

- (25) A. 2048H  
 B.4096H  
 C.4D16H  
 D.6D16H

• 进程 P1、P2、P3、P4 和 P5 的前趋图如下所示:



若用 PV 操作控制进程 P1、P2、P3、P4 和 P5 并发执行的过程, 需要设置 5 个信号量 S1、S2、S3、S4 和 S5, 且信号量 S1~S5 的初值都等于零。如下的进程执行图中 a 和 b 处应分别填写 ( ); c 和 d 处应分别填写 ( ); e 和 f 处应分别填写 ( )。



- (26) A. V(S1)和 P(S2)V(S3)  
 B.P(S1)和 V(S2)V(S3)  
 C.V(S1)和 V(S2)V(S3)  
 D.P(S1)和 P(S2)V(S3)

- (27) A. P(S2)和 P(S4)  
 B.V(S2)和 P(S4)  
 C.P(S2)和 V(S4)

D.V(S2)和 V(S4)

(28) A. P(S4)和 V(S5)

B.V(S5)和 P(S4)

C.V(S4)和 P(S5)

D.V(S4)和 V(S5)

● 以下关于螺旋模型的叙述中, 不正确的是( )

(29) A. 它是风险驱动的, 要求开发人员必须具有丰富的风险评估知识和经验

B.它可以降低过多测试或测试不足带来的风险

C.它包含维护周期, 因此维护和开发之间没有本质区别

D.它不适用于大型软件开发

● 以下关于极限编程(XP)中结对编程的叙述中, 不正确的是( )。

(30) A. 支持共同代码拥有和共同对系统负责

B. 承担了非正式的代码审查过程

C. 代码质量更高

D. 编码速度更快

● 以下关于 C/S (客户机/服务器)体系结构的优点的叙述中, 不正确的是\_\_(31)\_\_\_。

(31) A. 允许合理地划分三层的功能, 使之在逻辑上保持相对独立性

B.允许各层灵活地选用平台和软件

C. 各层可以选择不同的开发语言进行并行开发

D. 系统安装、修改和维护均只在服务器端进行

● 在设计软件的模块结构时, ( )不能改进设计质量。

(32) A. 尽量减少高扇出结构

B.模块的大小适中

C. 将具有相似功能的模块合并

D. 完善模块的功能

● 模块 A、B 和 C 有相同的程序块, 块内的语句之间没有任何联系, 现把改程序块取出来, 形成新的模块 D, 则模块 D 的内聚类型为( )内聚。以下关于该内聚类型的叙述中, 不正确的是( )。

(33) A. 巧合

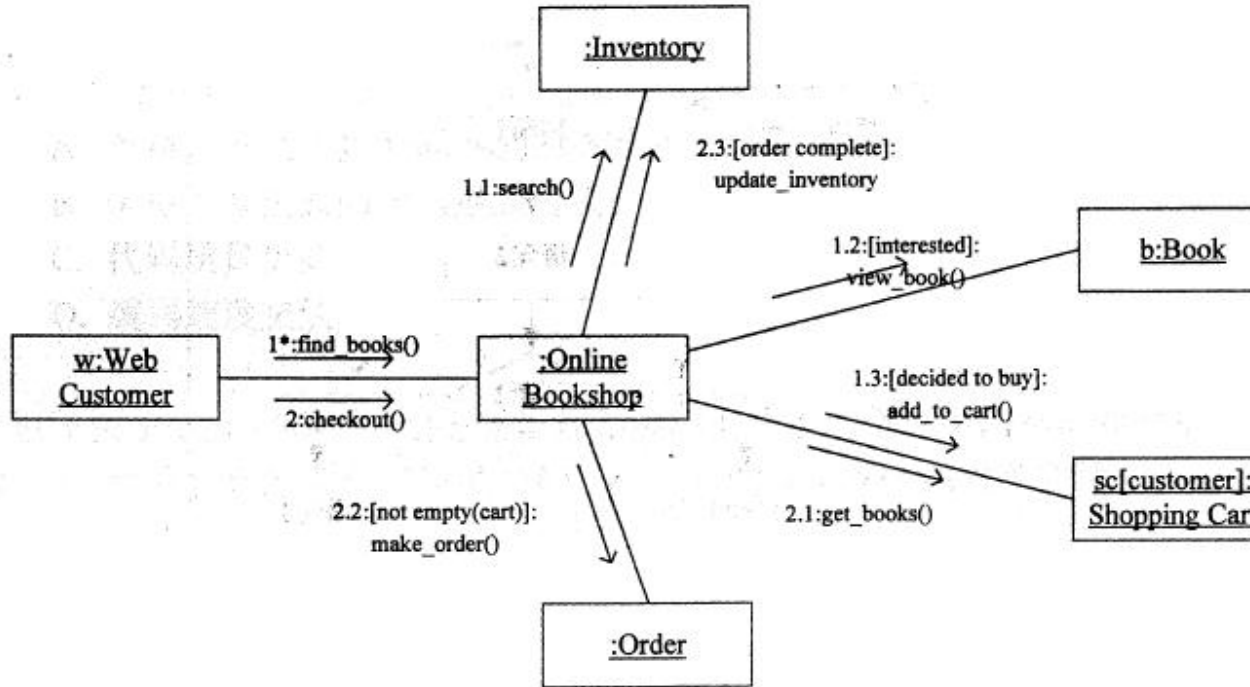
B.逻辑

C.时间

- D.过程
- (34) A. 具有最低的内聚性  
B.不易修改和维护  
C. 不易理解  
D. 不影响模块间的耦合关系
- 对下图所示的程序流程图进行语句覆盖测试和路劲覆盖测试, 至少需要( )个测试用例。采用McCabe度量法计算其环路复杂度为( )。
- (35) A. 2 和 3  
B.2 和 4  
C.2 和 5  
D.2 和 6
- (36) A. 1  
B.2  
C.3  
D.4
- 在面向对象方法中, 两个及以上的类作为一个类的超类时, 称为( ), 使用它可能造成子类中存在( )的成员。
- (37) A. 多重继承  
B.多态  
C.封装  
D.层次继承
- (38) A. 动态  
B.私有  
C.公共  
D.二义性
- 采用面向对象方法进行软件开发, 在分析阶段, 架构师主要关注系统的( )。
- (39) A. 技术  
B.部署  
C.实现  
D.行为
- 在面向对象方法中,多态指的是( )
- (40) A. 客户类无需知道所调用方法的特定子类的实现  
B. 对象动态地修改类  
C.一个对象对应多张数据库表  
D. 子类只能够覆盖父类中非抽象的方法

•

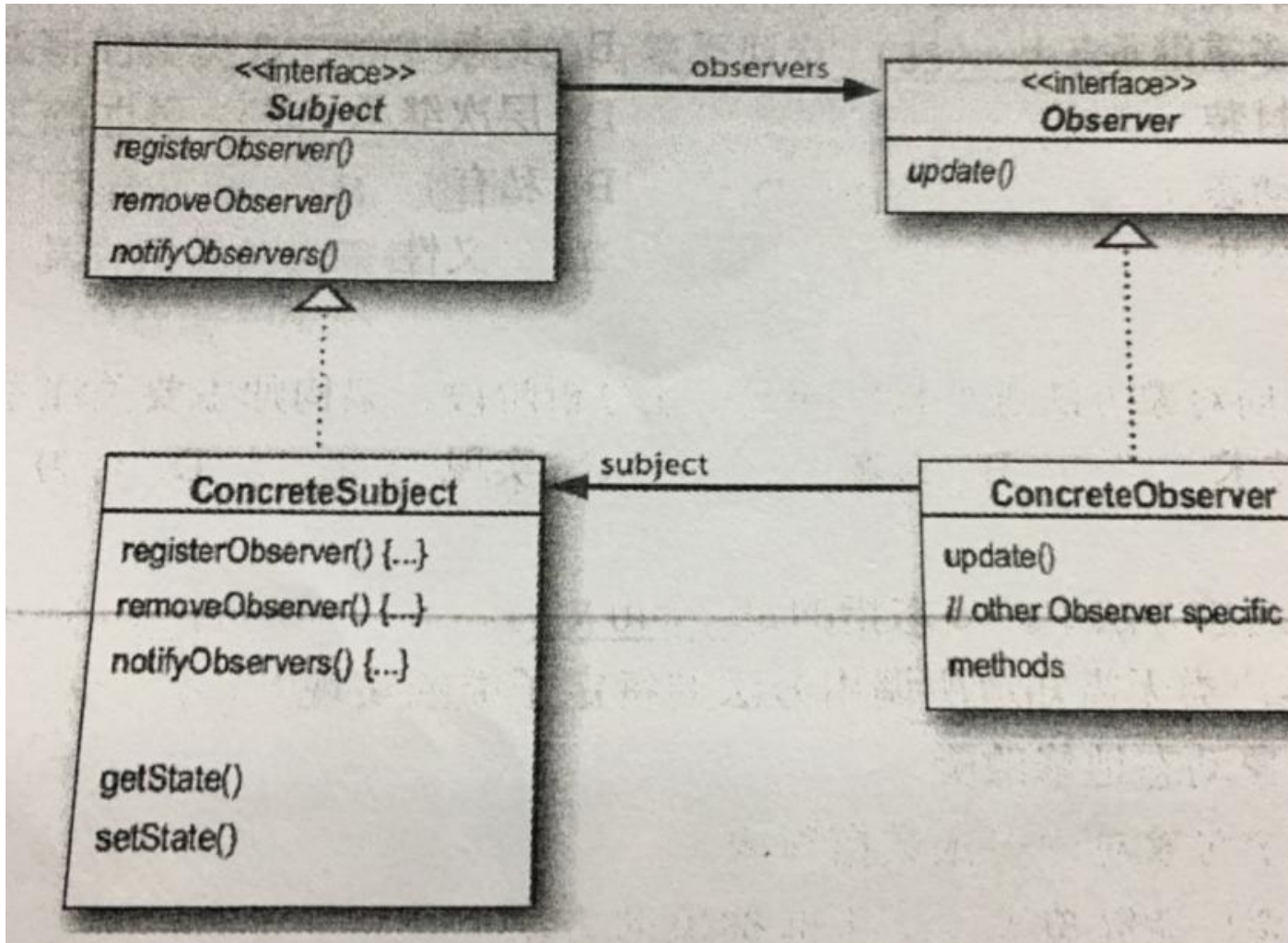
● 以下 UML 图是 (41), 图中 **:Order** 和 **b:Book** 表示 (42), **1\*:find** 和 **1.1:search()** 表示 (43)。



- (41) A. 序列图
- B. 状态图
- C. 通信图
- D. 活动图
- (42) A. 类
- B. 对象
- C. 流名称
- D. 消息
- (43) A. 类
- B. 对象
- C. 流名称
- D. 消息

● 下图所示为观察者(Observer)模式的抽象示意图, 其中( )知道其观察者, 可以有任意多个观察者观察同一个目标;提供住处和删除观察者对象的接口。此模式体现的最主要的特征是( )





- (44) A. Subject  
B. Observer  
C. ConcreteSubject  
D. ConcreteObserver
- (45) A. 类应该对扩展开放, 对修改关闭  
B. 使所要交互的对象尽量松耦合  
C. 组合优先于继承使用  
D. 仅与直接关联类交互

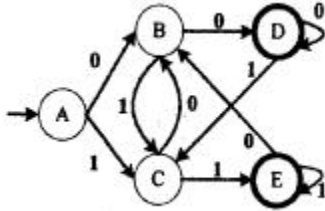
● 装饰器 (Decorator) 模式用于 ( ); 外观 (Facade) 模式用于 ( )。

- ① 将一个对象加以包装以给客户提供其希望的另外一个接口  
② 将一个对象加以包装以提供一些额外的行为  
③ 将一个对象加以包装以控制对这个对象的访问  
④ 将一系列对象加以包装以简化其接口

- (46) A. ①  
B. ②  
C. ③

- D.④  
 (47) A. ①  
 B.②  
 C.③  
 D.④

● 某确定的有限自动机 (DFA) 的状态转换图如下图所示 (A 是初态, D、E 是终态), 则该 DFA 能识别 ( )



- (48) A. 00110  
 B.10101  
 C.11100  
 D.11001

● 函数 main\_\_(49)\_\_\_、f\_\_(50)\_\_\_的定义如下所示, 调用函数们 f\_\_(51)\_\_\_时, 第一个参数采用传值 (call by value) 方式, 第二个参数采用传引用 (call by reference) 方式, main\_\_(52)\_\_\_ 函数中 "print(x)" 执行后输出的值为 ( )

**main( )**

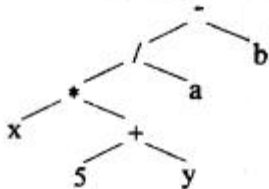
```
int x = 5;
f(x+1, x);
print(x);
```

**f(int x, int &a)**

```
x = x*x - 1;
a = x + a;
return;
```

- (49) A. 11  
 B.40  
 C.45  
 D.70

● 下图为一个表达式的语法树, 该表达式的后缀形式为 ( )



- (50) A. x 5 y + \* a / b -  
 B. x 5 y a b \* + / -  
 C. - / \* x + 5 y a b

D.  $x^5 * y + a/b$

● 若事务 T<sub>1</sub> 对数据 D<sub>1</sub> 加了共享锁, 事务 T<sub>2</sub>、T<sub>3</sub> 分别对数据 D<sub>2</sub>、D<sub>3</sub> 加了排它锁, 则事务 T<sub>1</sub> 对数据( ); 事务 T<sub>2</sub> 对数据( )

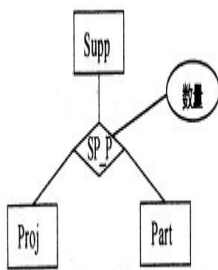
- (51) A. D<sub>2</sub>、D<sub>3</sub> 加排它锁都成功  
 B. D<sub>2</sub>、D<sub>3</sub> 加共享锁都成功  
 C. D<sub>2</sub> 加共享锁成功, D<sub>3</sub> 加排它锁失败  
 D. D<sub>2</sub>、D<sub>3</sub> 加排它锁和共享锁都失败

- (52) A. D<sub>1</sub>、D<sub>3</sub> 加共享锁都失败  
 B. D<sub>1</sub>、D<sub>3</sub> 加共享锁都成功  
 C. D<sub>1</sub> 加共享锁成功, D<sub>3</sub> 加排它锁失败  
 D. D<sub>1</sub> 加排它锁成功, D<sub>3</sub> 加共享锁失败

● 假设关系 R<U,F>, U = {A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub>}, F = {A<sub>1</sub>A<sub>3</sub> → A<sub>2</sub>, A<sub>1</sub>A<sub>2</sub> → A<sub>3</sub>}, 则关系 R 的各候选关键字中必定含有属性( )。

- (53) A. A<sub>1</sub>  
 B. A<sub>2</sub>  
 C. A<sub>3</sub>  
 D. A<sub>2</sub> A<sub>3</sub>

● 在某企业的工程项目管理系统的数据库中供应商关系 Supp、项目关系 Proj 和零件关系 Part 的 E-R 模型和关系模式如下:



Supp (供应商号, 供应商名, 地址, 电话)

Proj (项目号, 项目名, 负责人, 电话)

Part (零件号, 零件名)

其中, 每个供应商可以为多个项目供应多种零件, 每个项目可由多个供应商供应多种零件。SP P 需要生成一个独立的关系模式, 其联系类型为 ( )

给定关系模式 SP P (供应商号, 项目号, 零件号, 数量) 查询至少供应了 3 个项目 (包含 3 项) 的供应商, 输出其供应商号和供应零件数量的总和, 并按供应商号降序排列。

```
SELECT 供应商号, SUM(数量) FROM( )
GROUP BY 供应商号
( )
```

ORDER BY 供应商号 DESC;

- (54) A. \*.\*:\*  
B.1:\*:\*  
C.1:1:\*  
D.1:1:1

- (55) A. Supp  
B.Proj  
C.Part  
D.SP P

- (56) A. HAVING COUNT(项目号)>2  
B.WHERE COUNT(项目号)>2  
C.HAVING COUNT(DISTINCT(项目号))>2  
D.WHERE COUNT(DISTINCT(项目号))>3

- 以下关于字符串的叙述中, 正确的是 ( )

- (57) A. 包含任意个空格字符的字符串称为空串  
B. 字符串不是线性数据结构  
C. 字符串的长度是指串中所含字符的个数  
D. 字符串的长度是指串中所含非空格字符的个数

- 已知栈 S 初始为空, 用 I 表示入栈、O 表示出栈, 若入栈序列为 a<sub>1</sub>a<sub>2</sub>a<sub>3</sub>a<sub>4</sub>a<sub>5</sub>, 则通过栈 S 得到出栈序列 a<sub>2</sub>a<sub>4</sub>a<sub>5</sub>a<sub>3</sub>a<sub>1</sub> 的合法操作序列( )

- (58) A. IIIOIOIOOO  
B.IOIOIOIOIO  
C.IOOIOIOIO  
D.IIOOIOIOOO

- 某二叉树的先序遍历序列为 ABCDEF, 中序遍历序列为 BADCFE, 则该二叉树的高度(即层数)为( )

- (59) A. 3  
B.4  
C.5  
D.6

- 对于 n 个元素的关键字序列 {k<sub>1</sub>, k<sub>2</sub>, ..., k<sub>n</sub>} , 当且仅当满足关系 k<sub>i</sub> ≤ k<sub>2i</sub> 且 k<sub>i</sub> ≤ k<sub>2i+1</sub> {i=1..2...[n/2]} 时称其为小根堆(小顶堆)。以下序列中, ( )不是小根堆。

- (60) A. 16,25,40,55,30,50,45  
B.16,40,25,50,45,30,55  
C.16,25,39,41,45,43,50  
D.16,40,25,53,39,55,45

● 在 12 个互异元素构成的有序数组  $a[1..12]$  中进行二分查找(即折半查找, 向下取整), 若待查找的元素正好等于  $a[9]$ , 则在此过程中, 依次与数组中的( )比较后, 查找成功结束。

- (61) A.  $a[6]$ 、 $a[7]$ 、 $a[8]$ 、 $a[9]$   
 B.  $a[6]$ 、 $a[9]$   
 C.  $a[6]$ 、 $a[7]$ 、 $a[9]$   
 D.  $a[6]$ 、 $a[8]$ 、 $a[9]$

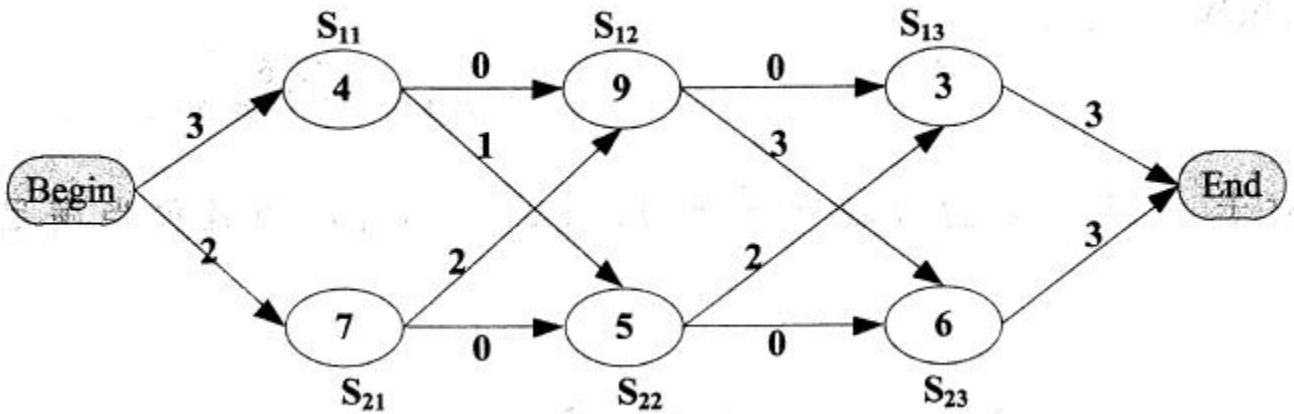
● 某汽车加工工厂有两条装配线 L1 和 L2, 每条装配线的工位数为  $n$  ( $S_{ij}$ ,  $i=1$  或  $2$ ,  $j=1, 2, \dots, n$ ), 两条装配线对应的工位完成同样的加工工作, 但是所需要的时间可能不同 ( $a_{ij}$ ,  $i=1$  或  $2$ ,  $j=1, 2, \dots, n$ )。汽车底盘开始到进入两条装配线的时间 ( $e_1, e_2$ ) 以及装配后到结束的时间( $X_1X_2$ )也可能不相同。从一个工位加工后流到下一个工位需要迁移时间( $t_{ij}$ ,  $i=1$  或  $2$ ,  $j=2, \dots, n$ )。现在要以最快的时间完成一辆汽车的装配, 求最优的装配路线。  
 分析该问题, 发现问题具有最优子结构。以 L1 为例, 除了第一个工位之外, 经过第  $j$  个工位的最短时间包含了经过 L1 的第  $j-1$  个工位的最短时间或者经过 L2 的第  $j-1$  个工位的最短时间, 如式(1)。装配后到结束的最短时间包含离开 L1 的最短时间或者离开 L2 的最短时间如式(2)。

$$f_{1,j} = \begin{cases} e_1 + a_{1,j} & \text{若 } j=1 \\ \min(f_{1,j-1} + a_{1,j} + t_{1,j-1}, f_{2,j-1} + a_{1,j} + t_{2,j-1}) & \text{其他} \end{cases} \quad (1)$$

$$f_{\min} = \min(f_{1,n} + X_1, f_{2,n} + X_2) \quad (2)$$

由于在求解经过 L1 和 L2 的第  $j$  个工位的最短时间均包含了经过 L1 的第  $j-1$  个工位的最短时间或者经过 L2 的第  $j-1$  个工位的最短时间, 该问题具有重复子问题的性质, 故采用迭代方法求解。

该问题采用的算法设计策略是 (62), 算法的时间复杂度为 (63)  
 以下是一个装配调度实例, 其最短的装配时间为 (64), 装配路线为 (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.  $O(n^2)$
- D.  $O(n \lg n)$
- (64) A. 21
- B. 23
- C. 20
- D. 26
- (65) A. S11→S12→S13
- B. S11→S22→S13
- C. S21→S12→S23
- D. S21→S22→S23

● 在浏览器地址栏输入一个正确的网址后, 本地主机将首先在\_\_(66)\_\_\_查询该网址对应的 IP 地址。

- (66) A. 本地 DNS 缓存
- B. 本机 hosts 文件
- C. 本地 DNS 服务器
- D. 根域名服务器

● 下面关于 Linux 目录的描述中, 正确的是\_\_(67)\_\_\_

- (67) A. Linux 只有一个根目录, 用 "/root" 表示
- B. Linux 中有多个根目录, 用 "/" 加相应目录名称表示
- C. Linux 中只有一个根目录, 用 "/" 表示
- D. Linux 中有多个根目录, 用相应目录名称表示

● 以下关于 TCP/IP 协议栈中协议和层次的对应关系正确的是\_\_(68)\_\_\_

- A. 

TFTP	Telnet
UDP	TCP
ARP	
- B. 

RIP	Telnet
UDP	TCP
ARP	
- C. 

HTTP	SNMP
TCP	UDP
IP	
- D. 

SMTP	FTP
UDP	TCP
IP	

● 在异步通信中, 每个字符包含 1 位起始位、7 位数据位和 2 位终止位, 若每秒钟传送 500 个字符, 则有效数据速率为\_\_(69)\_\_\_

- (69) A. 500b/s

- B.700b/s
- C.3500b/s
- D.5000b/s

- 以下路由策略中, 依据网络信息经常更新路由的是( )

- (70) A. 静态路由  
B. 洪泛式  
C. 随机路由  
D. 自适应路由

- The beauty of software is in its function, in its internal structure, and in the way in which it is created by a team. To a user, a program with just the right features presented through an intuitive and ( 71 ) interface is beautiful. To a software designer, an internal structure that is partitioned in a simple and intuitive manner, and that minimizes internal coupling is beautiful. To developers and managers, a motivated team of developers making significant progress every week, and producing defect-free code, is beautiful. There is beauty on all these levels.

our world needs software--lots of software. Fifty years ago software was something that ran in a few big and expensive machines. Thirty years ago it was something that ran in most companies and industrial settings. Now there is software running in our cell phones, watches, appliances, automobiles, toys, and tools. And need for new and better software never ( 72 ). As our civilization grows and expands, as developing nations build their infrastructures, as developed nations strive to achieve ever greater efficiencies, the need for more and more Software ( 73 ) to increase. It would be a great shame if, in all that software, there was no beauty.

We know that software can be ugly. We know that it can be hard to use, unreliable, and carelessly structured. We know that there are software systems whose tangled and careless internal structures make them expensive and difficult to change. We know that there are software systems that present their features through an awkward and cumbersome interface. We know that there are software systems that crash and misbehave. These are ( 74 ) systems. Unfortunately, as a profession, software developers tend to create more ugly systems than beautiful ones.

There is a secret that the best software developers know. Beauty is cheaper than ugliness. Beauty is faster than ugliness. A beautiful software system can be built and maintained in less time, and for less money, than an ugly one. Novice software developers don't understand this. They think that they have to do everything fast and quick. They think that beauty is ( 75 ). No! By doing things fast and quick, they make messes that make the software stiff, and hard to understand. Beautiful systems are flexible and easy to understand. Building them and maintaining them is a joy. It is ugliness that is impractical. Ugliness will slow you down and make your software expensive and brittle. Beautiful systems cost the least build and maintain, and are delivered soonest.

- (71) A. Simple  
B. Hard  
C. Complex  
D. duplicated

- (72) A. happens  
B. exists  
C. stops  
D. starts

- (73) A. starts

- B.continues
  - C.appears
  - D.stops
- (74) A. practical
- B.useful
  - C.beautiful
  - D.ugly
- (75) A. impractical
- B.perfect
  - C.time-wasting
  - D.practical

希赛网在线题库