

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/>

2011 年下半年软设综合知识真题答案与解析: <http://www.educity.cn/tiku/tp281.html>

## 2011 年下半年软件设计师考试上午真题 (参考答案)

- 若某条无条件转移汇编指令采用直接寻址, 则该指令的功能是将指令中的地址码送入\_\_ (1) \_\_。
  - (1) A. PC (程序计数器)
  - B. AR (地址寄存器)
  - C. AC (累加器)
  - D. ALU (算逻运算单元)
- 若某计算机系统的 I/O 接口与主存采用统一编址, 则输入输出操作是通过\_\_ (2) \_\_指令来完成的。
  - (2) A. 控制
  - B. 中断
  - C. 输入输出
  - D. 访存
- 在程序的执行过程中, Cache 与主存的地址映像由\_\_ (3) \_\_。
  - (3) A. 专门的硬件自动完成
  - B. 程序员进行调度
  - C. 操作系统进行管理
  - D. 程序员和操作系统共同协调完成
- 总线复用方式可以\_\_ (4) \_\_。
  - (4) A. 提高总线的传输带宽
  - B. 增加总线的功能
  - C. 减少总线中信号线的数量

D. 提高 CPU 利用率

● 在 CPU 的寄存器中, \_\_ (5) \_\_ 对用户是完全透明的。

- (5) A. 程序计数器
- B. 指令寄存器
- C. 状态寄存器
- D. 通用寄存器

● CPU 中译码器的主要作用是进行 \_\_ (6) \_\_。

- (6) A. 地址译码
- B. 指令译码
- C. 数据译码
- D. 选择多路数据至 ALU

● 利用 \_\_ (7) \_\_ 可以获取某 FTP 服务器中是否存在可写目录的信息。

- (7) A. 防火墙系统
- B. 漏洞扫描系统
- C. 入侵检测系统
- D. 病毒防御系统

● 通过内部发起连接与外部主机建立联系, 由外部主机控制并盗取用户信息的恶意代码为 \_\_ (8) \_\_。

- (8) A. 特洛伊木马
- B. 蠕虫病毒
- C. 宏病毒
- D. CIH 病毒

● 从认证中心 CA 获取用户 B 的数字证书, 该证书用 \_\_ (9) \_\_ 作数字签名; 从用户 B 的数字证书中可以获得 B 的公钥。

- (9) A. CA 的公钥
- B. CA 的私钥
- C. B 的公钥
- D. B 的私钥

● \_\_ (10) \_\_ 指可以不经著作权人许可, 不需支付报酬, 使用其作品。

- (10) A. 合理使用
- B. 许可使用
- C. 强制许可使用
- D. 法定许可使用

● 王某是 M 国际运输有限公司计算机系统管理员。任职期间, 王某根据公司的业务要求开发了“海运进出口业务系统”, 并由公司使用, 随后, 王某向国家版权局申请了计算机软件著作权登

记, 并取得了《计算机软件著作权登记证书》。证书明确软件名称是“海运出口业务系统 V1.0”, 著作权人为王某。以下说法中, 正确的是\_\_(11)\_\_。

- (11) A. 海运出口业务系统 V1.0 的著作权属于王某  
B. 海运出口业务系统 V1.0 的著作权属于 M 公司  
C. 海运出口业务系统 V1.0 的著作权属于王某和 M 公司  
D. 王某获取的软件著作权登记证是不可以撤消的

● 计算机通过 MIC (话筒接口) 收到的信号是\_\_(12)\_\_。

- (12) A. 音频数字信号  
B. 音频模拟信号  
C. 采样信号  
D. 量化信号

● \_\_(13)\_\_ 既不是图像编码也不是视频编码的国际标准。

- (13) A. JPEG  
B. MPEG  
C. ADPCM  
D. H.261

● 多媒体制作过程中, 不同媒体类型的数据收集、制作需要不同的软、硬件设备和技术手段, 动画制作一般通过\_\_(14)\_\_进行。

A. 字处理软件 B. 视频卡 C. 声卡 D. 图形 / 图像软件

● 多媒体制作过程中, 不同媒体类型的数据收集、制作需要不同的软、硬件设备和技术手段, 动画制作一般通过 ( ) 进行。

(14) A. 字处理软件 B. 视频卡 C. 声卡 D. 图形 / 图像软件

● 确定软件的模块划分及模块之间的调用关系是\_\_(15)\_\_阶段的任务。

- (15) A. 需求分析  
B. 概要设计  
C. 详细设计  
D. 编码

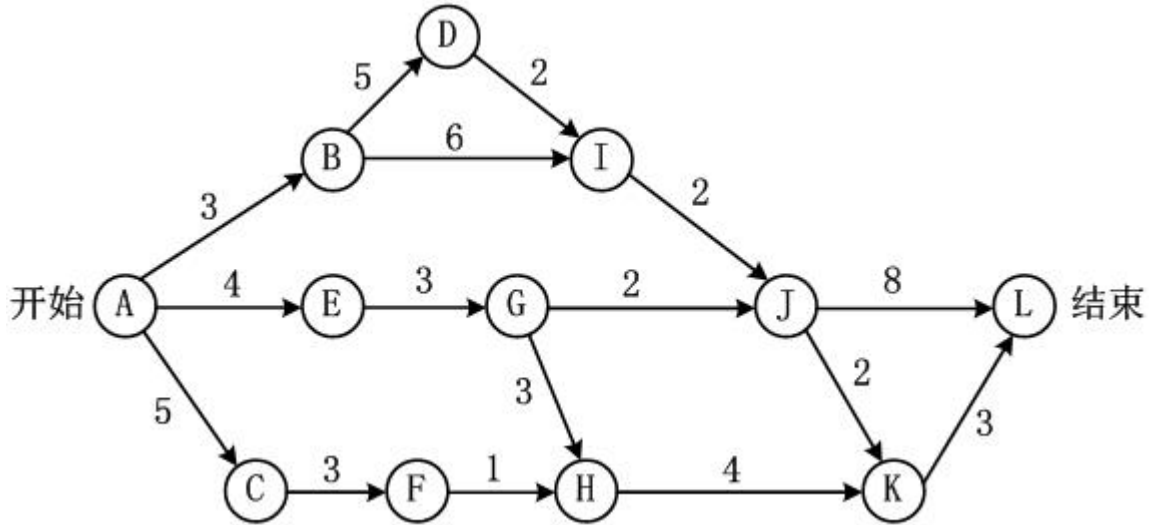
● 利用结构化分析模型进行接口设计时, 应以\_\_(16)\_\_为依据。

A. 数据流图 B. 实体--关系图 C. 数据字典 D. 状态--迁移图

● 利用结构化分析模型进行接口设计时, 应以 ( ) 为依据。

(16) A. 数据流图 B. 实体--关系图 C. 数据字典 D. 状态--迁移图

● 下图是一个软件项目的活动图, 其中顶点表示项目里程碑, 连接顶点的边表示包含的活动, 边上的值表示完成活动所需要的时间, 则关键路径长度为\_\_(17)\_\_。



- (17) A. 20  
 B. 19  
 C. 17  
 D. 16

● 甘特图 (Gantt 图) 不能\_\_(18)\_\_\_。

- (18) A. 作为项目进度管理的一个工具  
 B. 清晰地描述每个任务的开始和截止时间  
 C. 清晰地获得任务并行进行的信息  
 D. 清晰地获得各任务之间的依赖关系

● 以下关于风险管理的叙述中, 不正确的是\_\_(19)\_\_\_。

- (19) A. 仅根据风险产生的后果来对风险排优先级  
 B. 可以通过改变系统性能或功能需求来避免某些风险  
 C. 不可能去除所有风险, 但可以通过采取行动来降低或者减轻风险  
 D. 在项目开发过程中, 需要定期地评估和管理风险

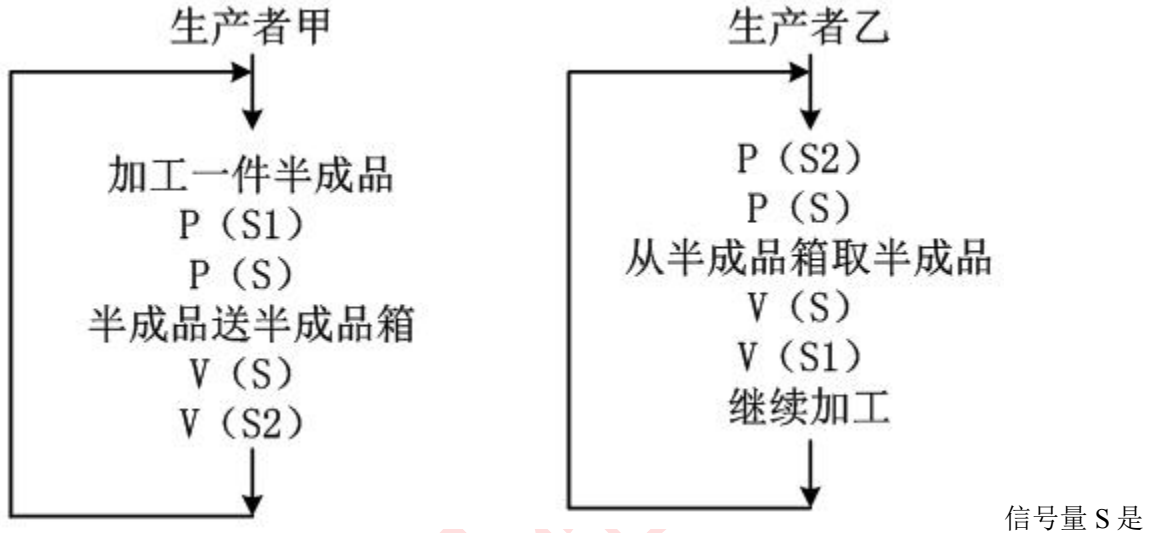
● 若 C 程序的表达式中引用了未赋初值的变量, 则\_\_(20)\_\_\_。

- (20) A. 编译时一定会报告错误信息, 该程序不能运行  
 B. 可以通过编译并运行, 但运行时一定会报告异常  
 C. 可以通过编译, 但链接时一定会报告错误信息而不能运行  
 D. 可以通过编译并运行, 但运行结果不一定是期望的结果

● 若二维数组  $arr[1..M, 1..N]$  的首地址为  $base$ , 数组元素按列存储且每个元素占用  $K$  个存储单元, 则元素  $arr[i, j]$  在该数组空间的地址为\_\_(21)\_\_\_。

- (21) A.  $base + ((i-1)*M + j-1)*K$   
 B.  $base + ((i-1)*N + j-1)*K$   
 C.  $base + (j-1)*M + i-1)*K$   
 D.  $base + (j-1)*N + i-1)*K$

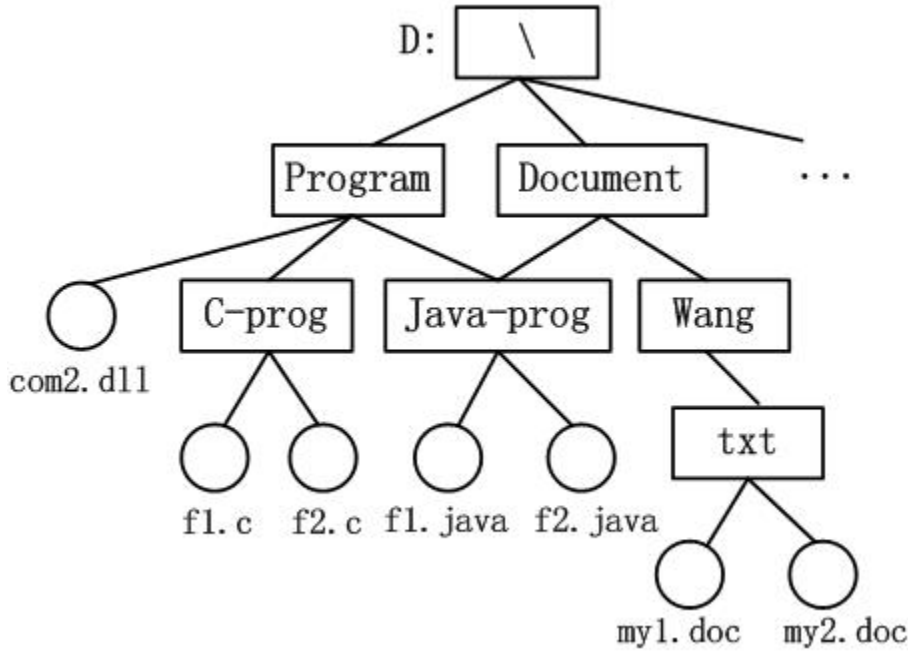
● 某企业生产流水线 M 共有两位生产者，生产者甲不断地将其工序上加工的半成品放入半成品箱，生产者乙从半成品箱取出继续加工。假设半成品箱可存放 n 件半成品，采用 PV 操作实现生产者甲和生产者乙的同步可以设置三个信号量 S、S1 和 S2，其同步模型如下图所示。



一个互斥信号量，初值为\_\_(22)\_\_; S1、S2 的初值分别为\_\_(23)\_\_。

- (22) A. 0  
 B. 1  
 C. n  
 D. 任意正整数
- (23) A. n、0  
 B. 0、n  
 C. 1、n  
 D. n、1

- 若某文件系统的目录结构如下图所示，假设用户要访问文件 f1.java，且当前工作目录为 Program，则该文件的全文件名为\_\_(24)\_\_\_，其相对路径为\_\_(25)\_\_\_。



- (24) A. f1.java  
 B. \Document\Java-prog\f1.java  
 C. D:\Program\Java-prog\f1.java  
 D. \Program\Java-prog\f1.java
- (25) A. Java-prog\  
 B. \Java-prog\  
 C. Program\Java-prog  
 D. \Program\Java-prog\  
 希赛网在软考

- 假设磁盘每磁道有 18 个扇区，系统刚完成了 10 号柱面的操作，当前移动臂在 13 号柱面上，进程的请求序列如下表所示。若系统采用 SCAN（扫描）调度算法，则系统响应序列为\_\_(26)\_\_\_；若系统采用 CSCAN（单向扫描）调度算法，则系统响应序列为\_\_(27)\_\_\_。

请求序列	柱面号	磁头号	扇区号
①	15	8	9
②	20	6	5
③	30	9	6
④	20	10	5
⑤	5	4	5
⑥	2	7	4
⑦	15	8	1
⑧	6	3	10
⑨	8	7	9
⑩	15	10	4

- (26) A. ⑦⑩①②④③⑨⑧⑤⑥

- B. ①⑦⑩②③④⑥⑤⑧⑨
  - C. ⑦⑩①②④③⑥⑤⑧⑨
  - D. ①⑦⑩②③④⑧⑨⑥⑤
- (27) A. ⑦⑩①②④③⑨⑧⑤⑥
- B. ①⑦⑩②③④⑥⑤⑧⑨
  - C. ⑦⑩①②④③⑥⑤⑧⑨
  - D. ①⑦⑩②③④⑧⑨⑥⑤

● 某程序设计语言规定在源程序中的数据都必须具有类型, 然而, \_\_ (28) \_\_ 并不是做出此规定的理由。

- (28) A. 为数据合理分配存储单元
- B. 可以定义和使用动态数据结构
  - C. 可以规定数据对象的取值范围及能够进行的运算
  - D. 对参与表达式求值的数据对象可以进行合法性检查

● 以下关于喷泉模型的叙述中, 不正确的是 \_\_ (29) \_\_。

- (29) A. 喷泉模型是以对象作为驱动力的模型, 适合于面向对象的开发方法
- B. 喷泉模型克服了瀑布模型不支持软件重用和多项开发活动集成的局限性
  - C. 模型中的开发活动常常需要重复多次, 在迭代过程中不断地完善软件系统
  - D. 各开发活动 (如分析、设计和编码) 之间存在明显的边界

● 若全面采用新技术开发一个大学记账系统, 以替换原有的系统, 则宜选择采用 \_\_ (30) \_\_ 进行开发。

- (30) A. 瀑布模型
- B. 演化模型
  - C. 螺旋模型
  - D. 原型模型

● 将每个用户的数据和其他用户的数据隔离开, 是考虑了软件的 \_\_ (31) \_\_ 质量特性。

- (31) A. 功能性
- B. 可靠性
  - C. 可维护性
  - D. 易使用性

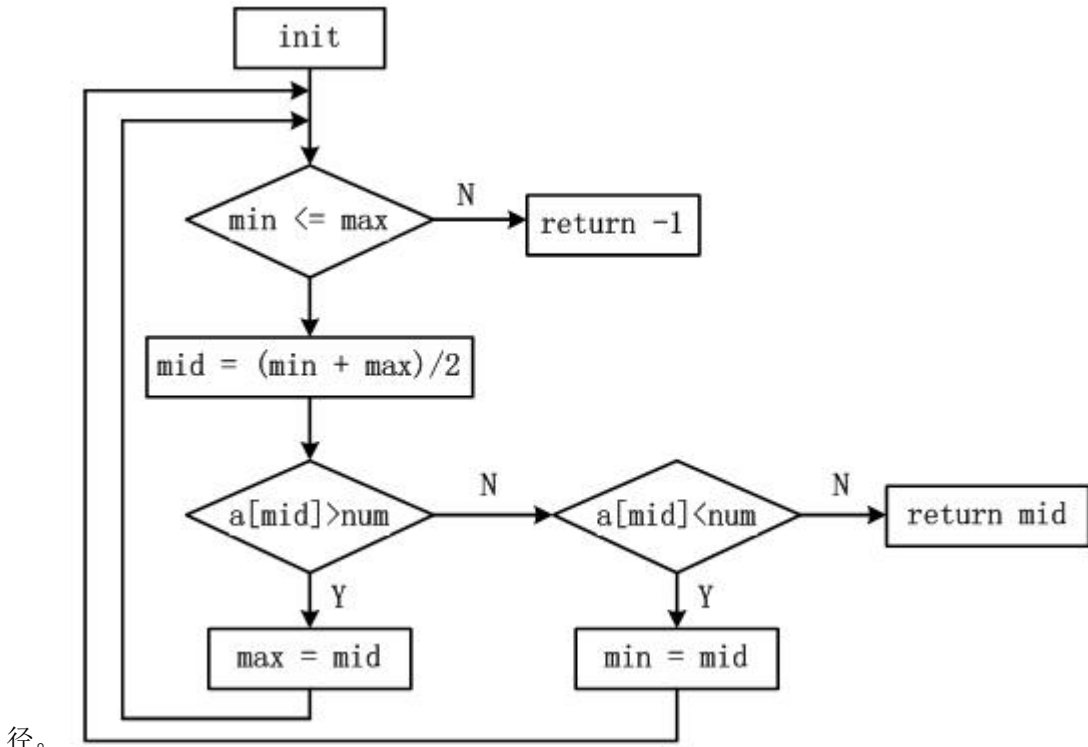
● 在软件评审中, 设计质量是指设计的规格说明书符合用户的要求。设计质量的评审内容不包括 \_\_ (32) \_\_。

- (32) A. 软件可靠性
- B. 软件的可测试性
  - C. 软件性能实现情况
  - D. 模块层次

- 针对应用在运行期的数据特点, 修改其排序算法使其更高效, 属于\_\_(33)\_\_维护。

(33) A. 正确性  
 B. 适应性  
 C. 完善性  
 D. 预防性

- 下图所示的逻辑流实现折半查找功能, 最少需要\_\_(34)\_\_个测试用例可以覆盖所有的可能路



(34) A. 1  
 B. 2  
 C. 3  
 D. 4

- 在某班级管理系统中, 班级的班委有班长、副班长、学习委员和生活委员, 且学生年龄在 15~25 岁。若用等价类划分来进行相关测试, 则\_\_(35)\_\_不是好的测试用例。

(35) A. (队长, 15)  
 B. (班长, 20)  
 C. (班长, 15)  
 D. (队长, 12)

- 进行防错性程序设计, 可以有效地控制\_\_(36)\_\_维护成本。

(36) A. 正确性  
 B. 适应性  
 C. 完善性



D. 预防性

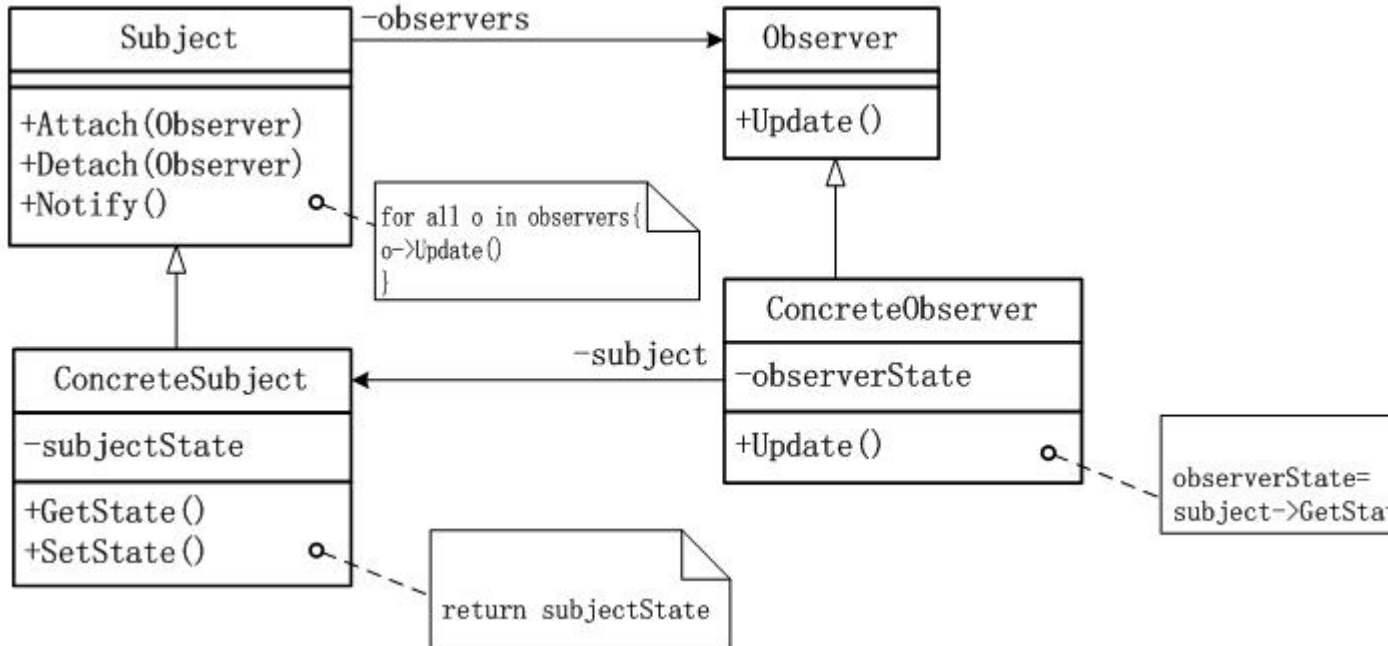
● 采用面向对象开发方法时，对象是系统运行时基本实体。以下关于对象的叙述中，正确的是\_\_(37)\_\_\_。

- (37) A. 对象只能包括数据（属性）  
 B. 对象只能包括操作（行为）  
 C. 对象一定有相同的属性和行为  
 D. 对象通常由对象名、属性和操作三个部分组成

● 一个类是\_\_(38)\_\_\_。在定义类时，将属性声明为 private 的目的是\_\_(39)\_\_\_。

- (38) A. 一组对象的封装  
 B. 表示一组对象的层次关系  
 C. 一组对象的实例  
 D. 一组对象的抽象定义  
 (39) A. 实现数据隐藏，以免意外更改  
 B. 操作符重载  
 C. 实现属性值不可更改  
 D. 实现属性值对类的所有对象共享

● \_\_\_\_(40)\_\_\_设计模式允许一个对象在其状态改变时，通知依赖它的所有对象。该设计模式的类图如下图，其中，\_\_(41)\_\_\_在其状态发生改变时，向它的各个观察者发出通知。



- (40) A. 命令 (Command)  
 B. 责任链 (Chain of Responsibility)  
 C. 观察者 (Observer)  
 D. 迭代器 (Iterator)

- (41) A. Subject  
B. ConcreteSubject  
C. Observer  
D. ConcreteObserver

● 在面向对象软件开发中, 封装是一种\_\_(42)\_\_技术, 其目的是使对象的使用者和生产者分离。

- (42) A. 接口管理  
B. 信息隐藏  
C. 多态  
D. 聚合

● 欲动态地给一个对象添加职责, 宜采用\_\_(43)\_\_模式。

- (43) A. 适配器(Adapter)  
B. 桥接(Bridge)  
C. 组合( Composite)  
D. 装饰器(Decorator)

● \_\_\_\_(44)\_\_\_模式通过提供与对象相同的接口来控制对这个对象的访问。

- (44) A. 适配器( Adapter)  
B. 代理(Proxy)  
C. 组合( Composite)  
D. 装饰器(Decorator)

● 采用 UML 进行面向对象开发时, 部署图通常在\_\_(45)\_\_阶段使用。

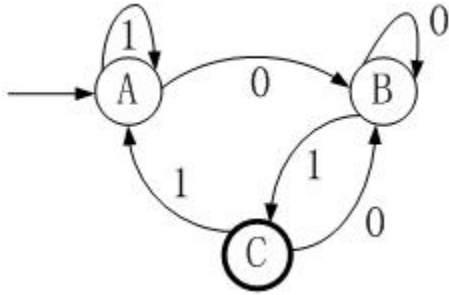
- (45) A. 需求分析  
B. 架构设计  
C. 实现  
D. 实施

● 业务用例和参与者一起描述\_\_(46)\_\_\_, 而业务对象模型描述\_\_(47)\_\_\_。

- (46) A. 工作过程中的静态元素  
B. 工作过程中的动态元素  
C. 工作过程中的逻辑视图  
D. 组织支持的业务过程

- (47) A. 业务结构  
B. 结构元素如何完成业务用例  
C. 业务结构以及结构元素如何完成业务用例  
D. 组织支持的业务过程

- 下图所示为一个有限自动机 (其中, A 是初态、C 是终态), 该自动机识别的语言可用正规式\_\_(48)\_\_表示。



- (48) A. (0|1)\*01  
 B. 1\*0\*10\*1  
 C. 1\*(0)\*01  
 D. 1\*(0|10)\*1\*

- 函数 t、f 的定义如下所示, 其中, a 是整型全局变量。设调用函数 t 前 a 的值为 5, 则在函数 t 中以传值调用(call by value)方式调用函数 f 时, 输出为\_\_(49)\_\_; 在函数 t 中以引用调用(call by reference)方式调用函数 f 时, 输出为\_\_(50)\_\_。

<pre>t(): int x = f(a);       print a+x;</pre>	<pre>f(int r): a = r+1; r = r*2           return r;</pre>
--	---

- (49) A. 12  
 B. 16  
 C. 20  
 D. 24
- (50) A. 12  
 B. 16  
 C. 20  
 D. 24

- 将 Students 表的插入权限赋予用户 UserA, 并允许其将该权限授予他人, 应使用的 SQL 语句为:

GRANT\_\_(51)\_\_TABLE Students TO UserA\_\_(52)\_\_\_;

- (51) A. UPDATE  
 B. UPDATEON  
 C. INSERT  
 D. INSERT ON
- (52) A. FORALL  
 B. PUBLIC  
 C. WITH CHECK OPTION  
 D. WITH GRANT OPTION

- 若有关系 R(A, B, C, D)和 S(C, D, E), 则与表达式 $\pi_{3, 4, 7}(\sigma_{4 < 5}(R \times S))$ 等价的 SQL 语句如下:

SELECT \_\_ (53) \_\_ FROM \_\_ (54) \_\_ WHERE \_\_ (55) \_\_;

- (53) A. A, B, C, D, E  
 B. C, D, E  
 C. R.A, R.B, R.C, R.D, S.E  
 D. R.C, R.D, S.E
- (54) A. R  
 B. S  
 C. R, S  
 D. RS
- (55) A. D < C  
 B. R.D < S.C  
 C. R.D < R.C  
 D. S.D < R.C

- E-R 图转换为关系模型时, 对于实体 E<sub>1</sub> 与 E<sub>2</sub> 间的多对多联系, 应该将 \_\_ (56) \_\_。

- (56) A. E<sub>1</sub> 的码加上联系上的属性并入 E<sub>2</sub>  
 B. E<sub>1</sub> 的码加上联系上的属性独立构成一个关系模式  
 C. E<sub>2</sub> 的码加上联系上的属性独立构成一个关系模式  
 D. E<sub>1</sub> 与 E<sub>2</sub> 码加上联系上的属性独立构成一个关系模式

- 在 KMP 模式匹配算法中, 需要求解模式串 p 的 next 函数值, 其定义如下 (其中, j 为模式串中字符的序号)。对于模式串“abaabaca”, 其 next 函数值序列为 \_\_ (57) \_\_。

$$next[j] = \begin{cases} 0 & j=1 \\ \max \{k \mid 1 < k < j, 'p_1 p_2 \dots p_{k-1}' = 'p_{j-k+1} p_{j-k+2} \dots p_{j-1}'\} & \text{其他情况} \\ 1 & \end{cases}$$

- (57) A. 01111111  
 B. 01122341  
 C. 01234567  
 D. 01122334

- 对于线性表 (由 n 个同类元素构成的线性序列), 采用单向循环链表存储的特点之一是 \_\_ (58) \_\_。

- (58) A. 从表中任意结点出发都能遍历整个链表  
 B. 对表中的任意结点可以进行随机访问  
 C. 对于表中的任意一个结点, 访问其直接前驱和直接后继结点所用时间相同  
 D. 第一个结点必须是头结点

- 无向图中一个顶点的度是指图中与该顶点相邻接的顶点数。若无向图 G 中的顶点数为 n, 边数为 e, 则所有顶点的度数之和为 \_\_ (59) \_\_。

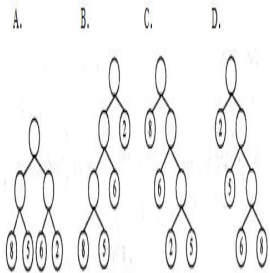
- (59) A. n\*e

- B.  $n+e$
- C.  $2n$
- D.  $2e$

● 一棵满二叉树，其每一层结点个数都达到最大值，对其中的结点从1开始顺序编号，即根结点编号为1，其左、右孩子结点编号分别为2和3，再下一层从左到右的编号为4、5、6、7，依此类推，每一层都从左到右依次编号，直到最后的叶子结点层为止，则用\_\_ (60) \_\_可判定编号为  $m$  和  $n$  的两个结点是否在同一层。

- A.  $\log_2 m = \log_2 n$
- B.  $[\log_2 m] = [\log_2 n]$
- C.  $[\log_2 m] + 1 = [\log_2 n]$
- D.  $[\log_2 m] = [\log_2 n] + 1$

● \_\_ (61) \_\_ 一是由权值集合 {8, 5, 6, 2} 构造的哈夫曼树（最优二叉树）。



● 迪杰斯特拉(Dijkstra)算法用于求解图上的单源点最短路径。该算法按路径长度递增次序产生最短路径，本质上说，该算法是一种基于\_\_ (62) \_\_策略的算法。

- (62)
- A. 分治
  - B. 动态规划
  - C. 贪心
  - D. 回溯

● 在有  $n$  个无序无重复元素值的数组中查找第  $i$  小的数的算法描述如下：任意取一个元素  $r$ ，用划分操作确定其在数组中的位置，假设元素  $r$  为第  $k$  小的数。若  $i$  等于  $k$ ，则返回该元素值；若  $i$  小于  $k$ ，则在划分的前半部分递归进行划分操作找第  $i$  小的数；否则在划分的后半部分递归进行划分操作找第  $k-i$  小的数。该算法是一种基于\_\_ (63) \_\_策略的算法。

- (63)
- A. 分治
  - B. 动态规划
  - C. 贪心
  - D. 回溯

● 对  $n$  个元素值分别为-1、0或1的整型数组  $A$  进行升序排序的算法描述如下：统计  $A$  中-1、0和1的个数，设分别为  $n_1$ 、 $n_2$ 和  $n_3$ ，然后将  $A$  中的前  $n_1$  个元素赋值为-1，第  $n_1+1$  到  $n_1+n_2$  个元素赋值为0，最后  $n_3$  个元素赋值为1。该算法的时间复杂度和空间复杂度分别为\_\_ (64) \_\_。

- A.  $\Theta(n)$ 和 $\Theta(1)$                       B.  $\Theta(n)$ 和 $\Theta(n)$   
 C.  $\Theta(n^2)$ 和 $\Theta(1)$                       D.  $\Theta(n^2)$ 和 $\Theta(n)$

- 设算法 A 的时间复杂度可用递归式  $T(n) = \begin{cases} \Theta(1) & , n=1 \\ 7T(n/2)+n^2 & , n>1 \end{cases}$  表示, 算法 B 的时间复杂度可用递

归式  $T(n) = \begin{cases} \Theta(1) & , n=1 \\ aT(n/4)+n^2 & , n>1 \end{cases}$  表示, 若要使得算法 B 渐进地快于算法 A, 则 a 的最大整数为 ( )。

- (65) A. 48  
 B. 49  
 C. 13  
 D. 14

- A 类网络是很大的网络, 每个 A 类网络中可以有 (66) 个网络地址。实际使用中必须把 A 类网络划分为子网, 如果指定的子网掩码为 255.255.192.0, 则该网络被划分为 (67) 个子网。

- (66) A.  $2^{10}$   
 B.  $2^{12}$   
 C.  $2^{20}$   
 D.  $2^{24}$   
 (67) A. 128  
 B. 256  
 C. 1024  
 D. 2048

- TCP 是互联网中的 (68) 协议, 使用 (69) 次握手协议建立连接。

- (68) A. 传输层  
 B. 网络层  
 C. 会话层  
 D. 应用层  
 (69) A. 1  
 B. 2  
 C. 3  
 D. 4

- 在 Windows 系统中, 为排除 DNS 域名解析故障, 需要刷新 DNS 解析器缓存, 应使用的命令是 (70)。

- (70) A. ipconfig/renew  
 B. ipconfig/flushdns  
 C. netstat -r

D. arp -a

● Extreme Programming (XP) is a discipline of software development with\_\_(71)\_\_of simplicity, communication, feedback and courage. Successful software development is a team effort - not just the development team, but the larger team consisting of customer, management and developers. XP is a simple process that brings these people together and helps them to succeed together. XP is aimed primarily at object-oriented projects using teams of a dozen or fewer programmers in one location. The principles of XP apply to any\_\_(72)\_\_project that needs to deliver quality software rapidly and flexibly.

An XP project needs a \_\_(73)\_\_customer to provide guidance. Customers, programmers, managers, are all working \_\_(74)\_\_ to build the system that's needed. Customers - those who have software that needs to be developed - will learn simple, effective ways to \_\_(75)\_\_ what they need, to be sure that they are getting what they need, and to steer the project to success.

- (71) A. importance  
B. keys  
C. roles  
D. values
- (72) A. small-sized  
B. moderately-sized  
C. large-sized  
D. huge-sized
- (73) A. part-time  
B. casual  
C. seldom  
D. full-time
- (74) A. together  
B. by themselves  
C. separately  
D. alone
- (75) A. tell  
B. know  
C. communicate  
D. feedback