

2017 年上半年

全国计算机技术与软件专业技术资格考试 2017 年上半年 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

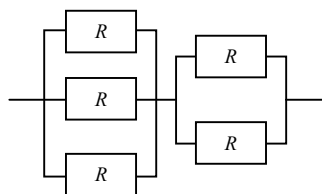
- 2017 年下半年全国计算机技术与软件专业技术资格考试日期是__(88)__月
__(89)__日。

(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 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 (1) 中。
 (1) A. 程序计数器 (PC) B. 累加寄存器 (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)。
 (8) A. RSA B. SHA-1 C. MD5 D. RC5
- 假定用户 A、B 分别在 I1 和 I2 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. I1、I2 互换私钥 D. I1、I2 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 (10) 享有。
 (10) A. 甲 B. 乙 C. 甲与乙共同 D. 软件设计师
- 根据《中华人民共和国商标法》，下列商品中必须使用注册商标的是 (11)。
 (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)。
 (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)。

(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 (14) 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 (15)。接口设计的任务主要是 (16)。

(15) A. 数据流图

B. E-R 图

C. 状态-迁移图

D. 加工规格说明

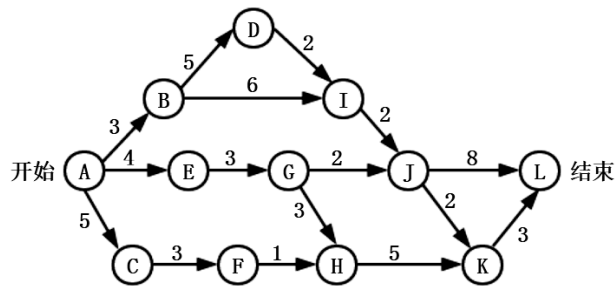
(16) A. 定义软件的主要结构元素及其之间的关系

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 (17) 天。活动 BD 和 HK 最早可以从第 (18) 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 (20)。

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)。

(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 (23) 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

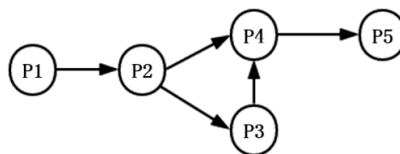
C. 14

D. 15

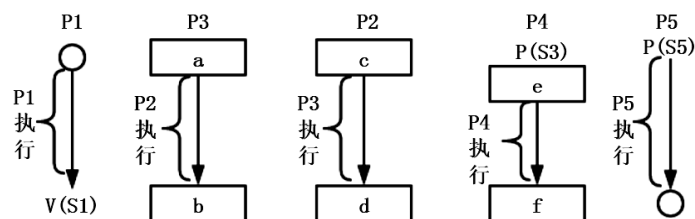
- 某计算机系统页面大小为 4K，进程的页面变换见下表。若进程的逻辑地址为 2D16H。该地址经过变换后，其物理地址应为 (25)。

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 (26)；c 和 d 处应分别填写 (27)；e 和 f 处应分别填写 (28)。

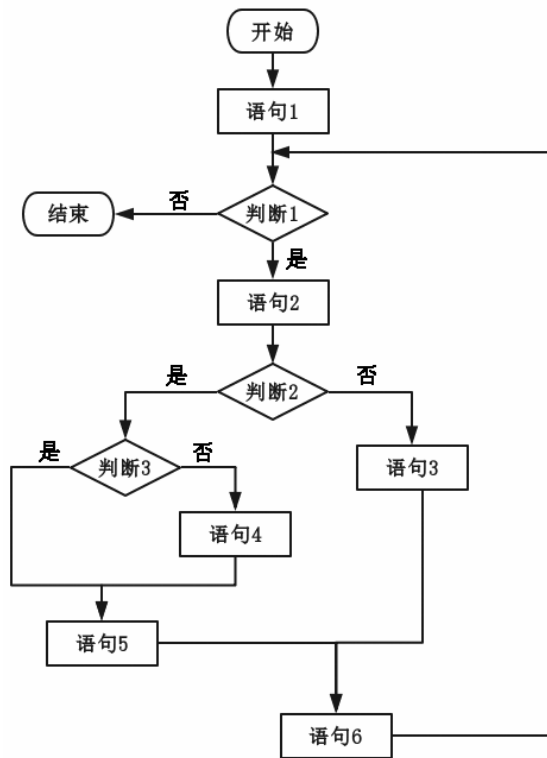


- (26) A. V(S1)和 P(S2)V(S3) B. P(S1)和 V(S2)V(S3)
 C. V(S1)和 V(S2)V(S3) D. P(S1)和 P(S2)V(S3)
- (27) A. P(S2)和 P(S4) B. V(S2)和 P(S4)
 C. P(S2)和 V(S4) D. V(S2)和 V(S4)
- (28) A. P(S4)和 V(S5) B. V(S5)和 P(S4)
 C. V(S4)和 P(S5) D. V(S4)和 V(S5)
- 以下关于螺旋模型的叙述中，不正确的是 (29)。
- (29) A. 它是风险驱动的，要求开发人员必须具有丰富的风险评估知识和经验
 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)。

(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

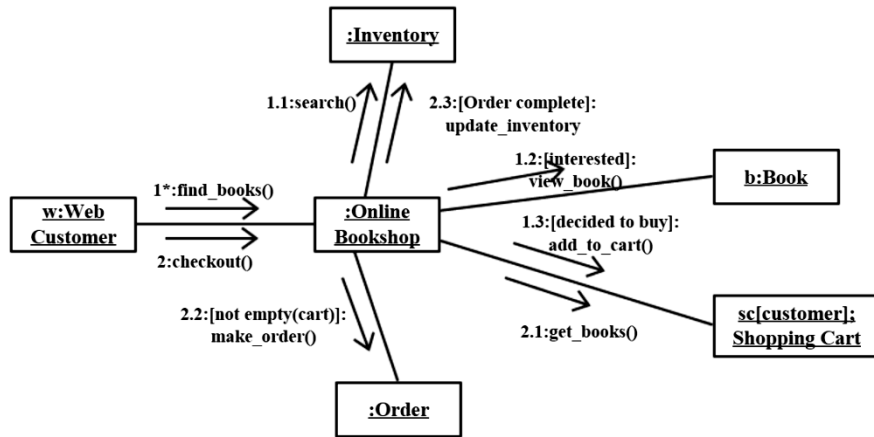
- C. 各层可以选择不同的开发语言进行并行开发
D. 系统安装、修改和维护均只在服务器端进行
- 在设计软件的模块结构时，(32)不能改进设计质量。
(32) A. 尽量减少高扇出结构 B. 尽量减少高扇入结构
C. 将具有相似功能的模块合并 D. 完善模块的功能
- 模块 A、B 和 C 有相同的程序块，块内的语句之间没有任何联系，现把改程序块取出来，形成新的模块 D，则模块 D 的内聚类型为(33)内聚。以下关于该内聚类型的叙述中，不正确的是(34)。
(33) A. 巧合 B. 逻辑 C. 时间 D. 过程
(34) A. 具有最低的内聚性 B. 不易修改和维护
C. 不易理解 D. 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径覆盖测试，至少需要(35)个测试用例。采用 McCabe 度量法计算其环路复杂度为(36)。



- (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(37)，使用它可能造成子类中存在(38)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)。
(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)。
(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

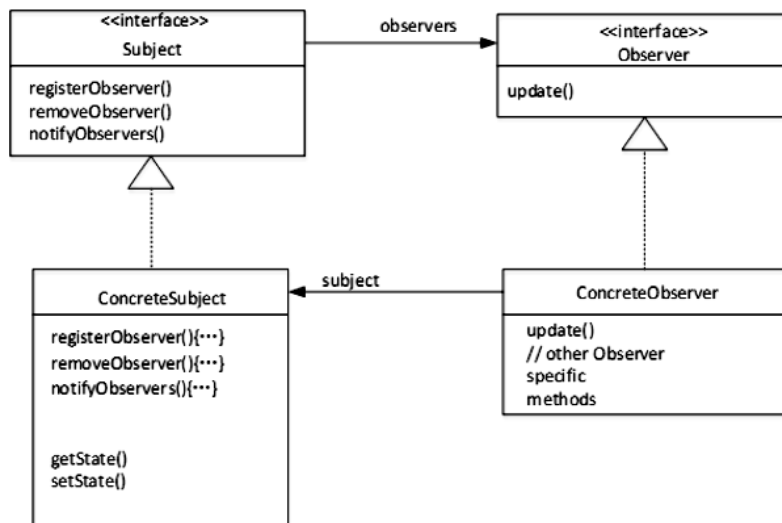
- C. 一个对象对应多张数据库表
D. 子类只能覆盖父类中非抽象的方法

- 以下 UML 图是 (41)，图中 `:Order` 和 `b:Book` 表示 (42)，`1*:find_books()` 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 (44) 知道其观察者，可以有任何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 (45)。



- (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

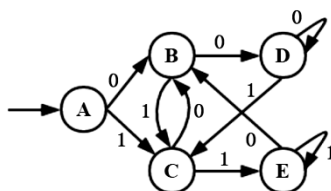
- 装饰器 (Decorator) 模式用于 (46)；外观 (Facade) 模式用于 (47)。

- 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)。



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

- 函数 main()、f() 的定义如下所示, 调用函数 f() 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() 函数中 "print(x)" 执行后输出的值为 (49)。

main()

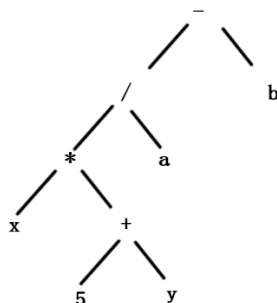
```
int x=5;
f(x+1,x);
print(x);
```

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)。



- (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 T1 对数据 D1 加了共享锁, 事务 T2、T3 分别对数据 D2、D3 加了排他锁, 则事务 T1 对数据 (51); 事务 T2 对数据 (52)。

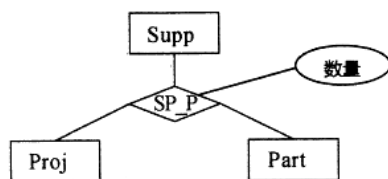
- (51) A. D2、D3 加排他锁都成功
 B. D2、D3 加共享锁都成功
 C. D2 加共享锁成功, D3 加排他锁失败
 D. D2、D3 加排他锁和共享锁都失败

- (52) A. D1、D3 加共享锁都失败
 B. D1、D3 加共享锁都成功
 C. D1 加共享锁成功, D3 加排他锁失败
 D. D1 加排他锁成功, D3 加共享锁失败

- 假设关系 $R \langle U, F \rangle$, $U = \{A1, A2, A3\}$, $F = \{A1A3 \rightarrow A2, A1A2 \rightarrow A3\}$, 则关系 R 的各候选关键字中必定含有属性 (53)。

- (53) A. A1 B. A2 C. A3 D. A2 A3

- 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 (54), 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 (55) GROUP BY 供应商号 (56) 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)。

(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 a1a2a3a4a5, 则通过栈 S 得到出栈序列 a2a4a5a3a1 的合法操作序列为 (58)。

(58) A. IIOIIOIOOO B. IOIOIOIOIO C. IOOIIIOIOIO 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)。

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

- 对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$, 当且仅当满足关系 $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ $\{i=1, 2, \dots, \lfloor n/2 \rfloor\}$ 时称其为小根堆(小顶堆)。以下序列中, (60) 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 (61) 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_1, X_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, 见式(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 的最短时间见式 (2)。

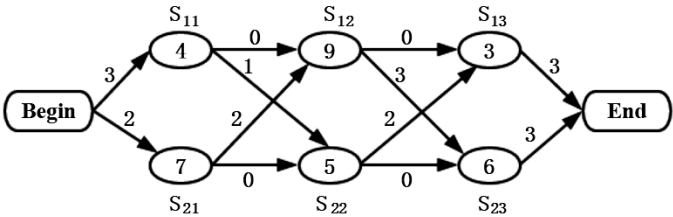
$$f_{1,j} = \begin{cases} e_1 + a_1, & j \\ \min (f_{1,j-1} + a_1, j + t_1, j-1, f_{2,j-1} + a_1, j + t_2, j-1) & \text{其他} \end{cases} \quad (1)$$

$$f_{\min} = \min (f_{1,n} + x_1, f_{2,n} + x_2) \quad (2)$$

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. $S_{11} \rightarrow S_{12} \rightarrow S_{13}$ B. $S_{11} \rightarrow S_{22} \rightarrow S_{13}$
 C. $S_{21} \rightarrow S_{12} \rightarrow S_{23}$ D. $S_{21} \rightarrow S_{22} \rightarrow S_{23}$

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)。

- (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)。

- (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 |

全国计算机技术与软件专业技术资格考试

2017 年上半年 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】

某医疗器械公司作为复杂医疗产品的集成商，必须保持高质量部件的及时供应。为了实现这一目标，该公司欲开发一个采购系统。系统的主要功能如下：

（1）检查库存水平。采购部门每天检查部件库存量，当特定部件的库存量降至其订货点时，返回低存量部件及库存量。

（2）下达采购订单。采购部门针对低存量部件及库存量提交采购请求，向其供应商（通过供应商文件访问供应商数据）下达采购订单，并存储于采购订单文件中。

（3）交运部件。当供应商提交提单并交运部件时，运输和接收（S/R）部门通过执行以下三步过程接收货物：

1）验证装运部件。通过访问采购订单并将其与提单进行比较来验证装运的部件，并将提单信息发给 S/R 职员。如果收货部件项目出现在采购订单和提单上，则已验证的提单和收货部件项目将被送去检验。否则，将 S/R 职员提交的装运错误信息生成装运错误通知发送给供应商。

2）检验部件质量。通过访问质量标准来检查装运部件的质量，并将已验证的提单发给检验员。如果部件满足所有质量标准，则将其添加到接受的部件列表用于更新部件库存。如果部件未通过检查，则将检验员创建的缺陷装运信息生成缺陷装运通知发送给供应商。

3）更新部件库存。库管员根据收到的（接受的）部件列表添加本次采购数量，与原有库存量累加来更新库存部件中的库存量。标记订单采购完成。

现采用结构化方法对该采购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。

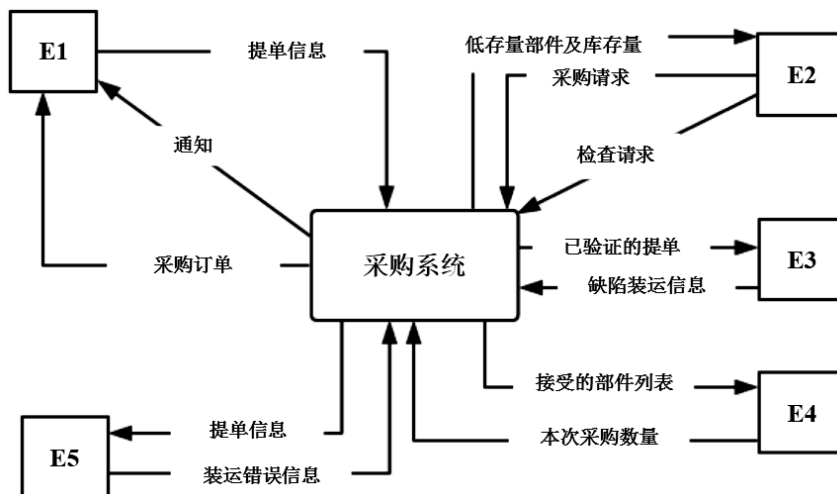


图 1 上下文数据流图

【问题 1】（5 分）

使用说明中的词语，给出图 1 中的实体 E1~E5

【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】（4 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】（2 分）

用 200 字以内文字，说明建模图 1 和图 2 时如何保持数据流图平衡。

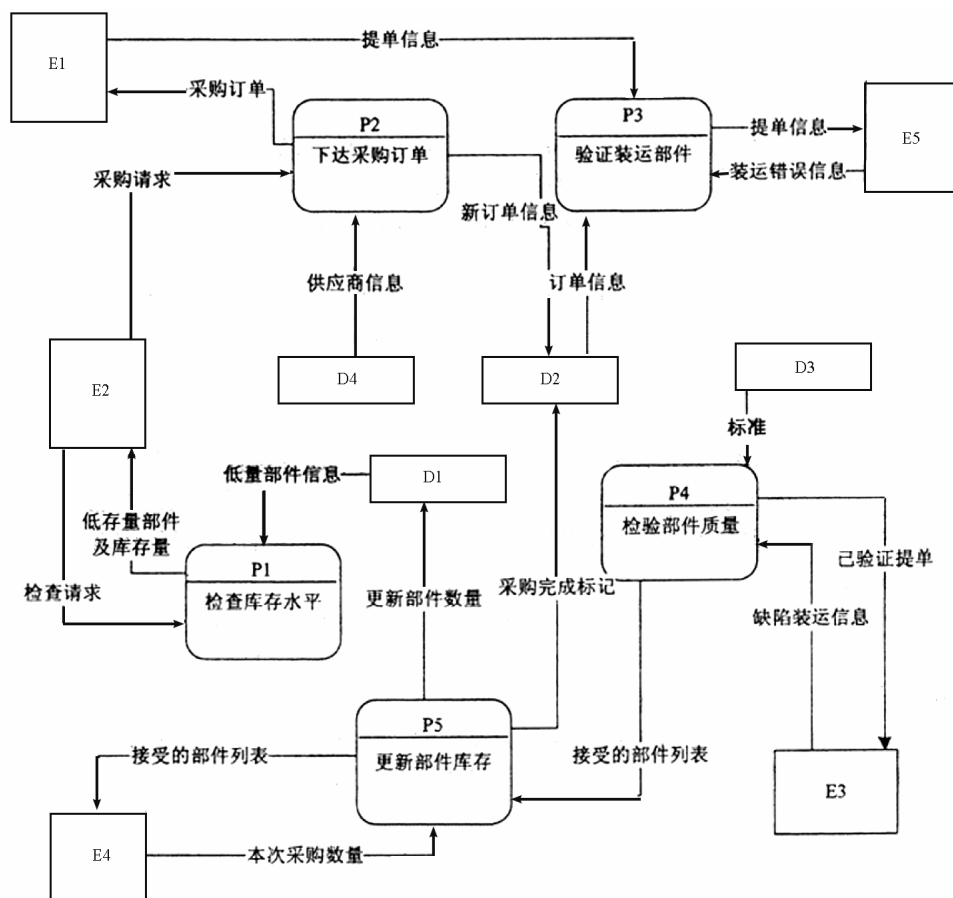


图 2 0 层数据流图

试题二（15 分）

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某房屋租赁公司拟开发一个管理系统用于管理其持有的房屋、租客及员工信息。请根据下述需求描述完成系统的数据库设计。

【需求描述】

- （1）公司拥有多幢公寓楼，每幢公寓楼有唯一的楼编号和地址。每幢公寓楼中有多套公寓，每套公寓在楼内有唯一的编号（不同公寓楼内的公寓号可相同）。系统需记录每套公寓的卧室数和卫生间数。
- （2）员工和租客在系统中有唯一的编号（员工编号和租客编号）。
- （3）对于每个租客，系统需记录姓名、多个联系电话、一个银行账号（方便自动扣房租）、一个紧急联系人的姓名及联系电话。
- （4）系统需记录每个员工的姓名、一个联系电话和月工资。员工类别可以是经理或维修工，也可兼任。每个经理可以管理多幢公寓楼。每幢公寓楼必须由一个经理管理。系统需记录每个维修工的业务技能，如：水暖维修、电工、木工等。
- （5）租客租赁公寓必须和公司签订租赁合同。一份租赁合同通常由一个或多个租客（合租）与该公寓楼的经理签订，一个租客也可租赁多套公寓。合同内容应包含签订日期、开始时间、租期、押金和月租金。

【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（不完整）如图 3 所示。

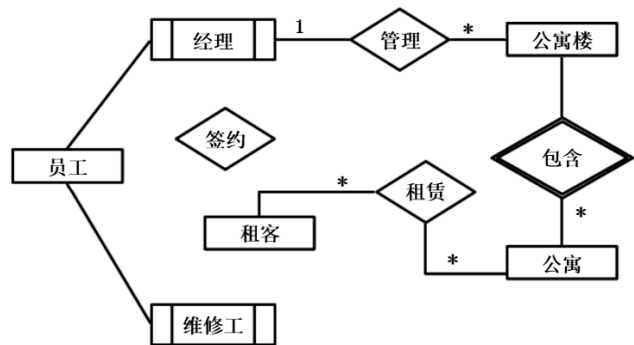


图 3 实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

联系电话（电话号码，租客编号）

租客（租客编号，姓名，银行账号，联系人姓名，联系人电话）

员工[员工编号，姓名，联系电话，类别，月工资，(a)]

公寓楼[(b)，地址，经理编号]

公寓（楼编号，公寓号，卧室数，卫生间数）

合同[合同编号，租客编号，楼编号，公寓号，经理编号，签订日期，起始日期，租期，(c)，押金]

【问题 1】（4.5 分）

补充图 3 中的“签约”联系所关联的实体及联系类型。

【问题 2】（4.5 分）

补充逻辑结构设计中的 (a)、(b)、(c) 三处空缺。

【问题 3】（6 分）

在租期内，公寓内设施如出现问题，租客可在系统中进行故障登记，填写故障描述，每项故障由系统自动生成唯一的故障编号，由公司派维修工进行故障维修，系统需记录每次维修的维修日期和维修内容。请根据此需求，对图 3 进行补充，并将所补充的 ER 图内容转换为一个关系模式，请给出该关系模式。

试题三（15 分）

阅读下列系统设计说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某玩具公司正在开发一套电动玩具在线销售系统，用于向注册会员提供端对端的玩具定制和销售服务。在系统设计阶段，“创建新订单 (New Order)”的设计用例详细描述见表 1，候选设计类分类见表 2，并根据该用例设计出部分类图如图 4 所示。

表 1 创建新订单 (New Order) 设计用例

用例名称	创建新订单 New Order
用例编号	ETM-R002
参与者	会员
前提条件	会员已经注册并成功登录系统

续表

典型事件流		(1) 会员 (C1) 单击“新的订单”按钮 (2) 系统列出所有正在销售的电动玩具清单及价格 (C2) (3) 会员单击复选框选择所需电动玩具并输入对应数量, 单击“结算”按钮 (4) 系统自动计算总价 (C3), 显示销售清单和会员预先设置个人资料的收货地址和支付方式 (C4) (5) 会员单击“确认支付”按钮 (6) 系统自动调用支付系统 (C5) 接口支付该账单 (7) 若支付系统返回成功标识, 系统生成完整订单信息持久存储到数据库订单表 (C6) 中 (8) 系统将以表格形式显示完整订单信息 (C7), 同时自动发送完整订单信息 (C8) 至会员预先配置的邮箱地址 (C9)
候选事件流	3a	1) 会员单击“定制”按钮 2) 系统以列表形式显示所有可以定制的电动玩具清单和定制属性 (如尺寸、颜色等) (C10); 3) 会员单击单选按钮选择所需要定制的电动玩具并填写所需要定制的属性要求, 单击“结算”按钮; 4) 回到步骤 (4)
	7a	1) 若支付系统返回失败标识, 系统显示会员当前默认支付方式 (C11) 让会员确认 2) 若会员单击“修改付款”按钮, 调用“修改付款”用例, 可以新增并存储为默认支付方式 (C12), 回到步骤 (4) 3) 若会员单击“取消订单”, 则该用例终止执行

表 2 候选设计类分类

接口类 (Interface, 负责系统与用户之间的交互)	(a)
控制类 (Control, 负责业务逻辑的处理)	(b)
实体类 (Entity, 负责持久化数据的存储)	(c)

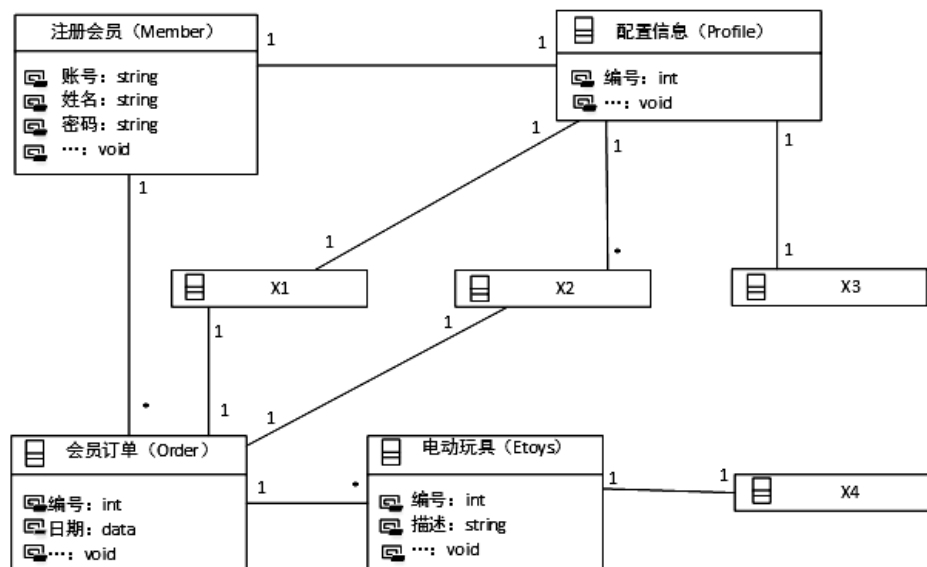


图 4 类图

在订单处理的过程中, 会员可以单击“取消订单”取消该订单。如果支付失败, 该订单将被标记为挂起状态, 可后续重新支付, 如果挂起超时 30min 未支付, 系统将自动取消该订单。订单支付成功后, 系统判断订单类型:

(1) 对于常规订单, 标记为备货状态, 订单信息发送到货运部, 完成打包后交付快递发货。

(2) 对于定制订单, 会自动进入定制状态, 定制完成后交付快递发货。会员在系统中单击“收货”按钮变为收货状态, 结束整个订单的处理流程。根据订单处理过程所设计的状态图如图 5 所示。

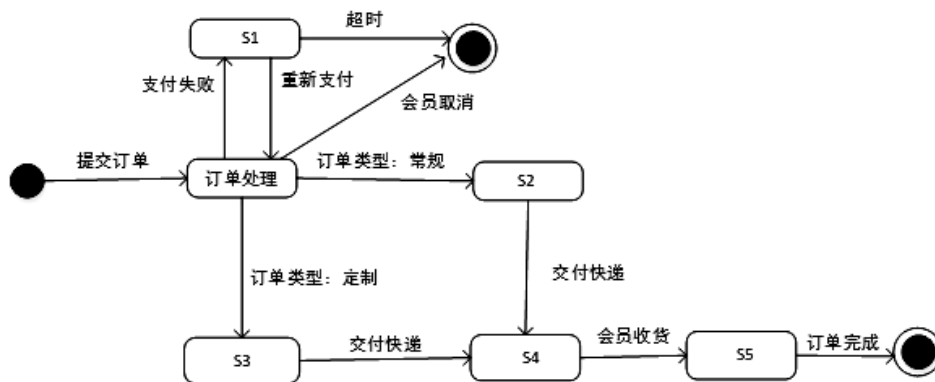


图5 订单状态图

【问题 1】（6 分）

根据表 1 中所标记的设计用例，请按照其类别将编号 C1~C12 分别填入表 2 中的（a）、（b）和（c）处。

【问题 2】（4 分）

根据创建新订单的用例描述，请给出图 4 中 X1~X4 处对应类的名称。

【问题 3】（5 分）

根据订单处理过程的描述，在图 5 中 S1~S5 处分别填入对应的状态名称。

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】

假币问题：有 n 枚硬币，其中有一枚是假币，已知假币的重量较轻。现只有一个天平，要求用尽量少的比较次数找出这枚假币。

【分析问题】

将 n 枚硬币分成相等的两部分：

（1）当 n 为偶数时，将前后两部分，即 $1 \cdots n/2$ 和 $n/2+1 \cdots 0$ ，放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币。

（2）当 n 为奇数时，将前后两部分，即 $1 \cdots (n-1)/2$ 和 $(n+1)/2+1 \cdots 0$ ，放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币；若两端重量相等，则中间的硬币，即第 $(n+1)/2$ 枚硬币是假币。

【C 代码】

下面是算法的 C 语言实现，其中：

coins[]: 硬币数组

first, last: 当前考虑的硬币数组中的第一个和最后一个下标

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

```
int getCounterfeitCoin(int coins[], int first, int last)
{
    int firstSum = 0, lastSum = 0;
    int i;
    if (first == last - 1) { //只剩两枚硬币
        if (coins[first] < coins[last])
```



```

        return first;
    return last;
}
if ( (1) ) { //偶数枚硬币
    for (i = first; i < first + (last - first) / 2 + 1; i++)
    {
        firstSum += coins[i];
    }
    for (i = first + (last - first) / 2 + 1; i < last + 1; i++) {
        lastSum += coins[i];
    }
    if (2)

    {
        return getCounterfeitCoin(coins, first, first + (last - first) / 2);
    }
    else
    {
        return getCounterfeitCoin(coins, first + (last - first) / 2 + 1, last);
    }
}
else
{ //奇数枚硬币
    for (i = first; i < first + (last - first) / 2; i++) {
        firstSum += coins[i];
    }
    for (i = first + (last - first) / 2 + 1; i < last + 1; i++) {
        lastSum += coins[i];
    }
    if (firstSum < lastSum) {
        return getCounterfeitCoin(coins, first, first + (last - first) / 2 - 1);
    }
    else if (firstSum > lastSum) {
        return getCounterfeitCoin(coins, first + (last - first) / 2 - 1, last);
    }
    else
        return (3);
}
}
}

```

【问题 1】（6 分）

根据题干说明，填充 C 代码中的空（1）～（3）。

【问题 2】（4 分）

根据题干说明和 C 代码，算法采用了 (4) 设计策略。

函数 getCounterfeitCoin 的时间复杂度为 (5) （用 O 表示）。

【问题 3】（5 分）

若输入的硬币数为 30，则最少的比较次数为 (6)，最多的比较次数为 (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】

某快餐厅主要制作并出售儿童套餐，一般包括主餐（各类比萨）、饮料和玩具，其餐品种类可能不同，但其制作过程相同。前台服务员（Waiter）调度厨师制作套餐。现采用生成器（Builder）模式实现制作过程，得到如图 6 所示的类图。

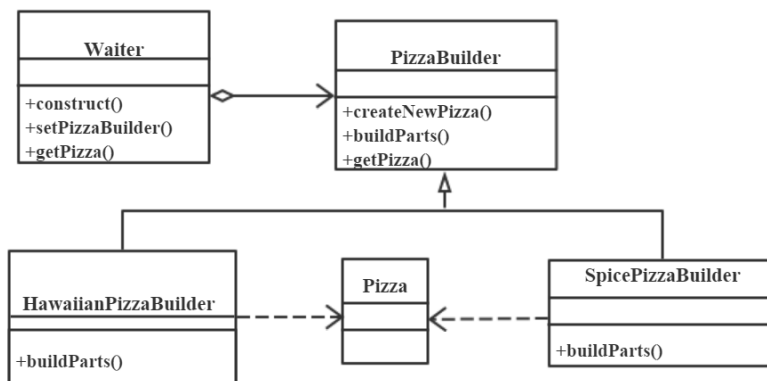


图 6 类图

【C++代码】

```
#include<iostream>
#include <string>
using namespace std;

class Pizza {
private:  string parts;
public:
    void setParts(string parts) { this->parts = parts; }
    string getParts() { return parts; }
};

class PizzaBuilder {
protected: Pizza*  pizza;
public:
    Pizza* getPizza() { return pizza; }
    void createNewPizza() { pizza = new Pizza(); }
    (1) ;
};

class HawaiianPizzaBuilder :public PizzaBuilder {
public:
    void buildParts() { pizza->setParts("cross +mild + ham&pineapple"); }
};

class SpicyPizzaBuider : public PizzaBuilder {
public:
    void buildParts() { pizza->setParts("pan baked +hot + ham&pineapple"); }
};
```

```

class Waiter{
private:
    PizzaBuilder* pizzaBuilder;
public:
    void setPizzaBuilder(PizzaBuilder* pizzaBuilder) {    //设置构建器
        (2) ;
    }
    Pizza* getPizza() { return pizzaBuilder->getPizza(); }
    void construct() {    //构建
        pizzaBuilder->createNewPizza();
        (3) ;
    }
};

int main()
{
    Waiter*waiter = new Waiter();
    PizzaBuilder*hawaiian_pizzabuilder = new HawaiianPizzaBuilder();
    (4) ;
    (5) ;
    cout << "pizza: " << waiter->getPizza()->getParts() << endl;
}

```

程序的输出结果为:

pizza:cross+mild+ham&pineapple

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】

某快餐厅主要制作并出售儿童套餐，一般包括主餐（各类比萨）、饮料和玩具，其餐品种类可能不同，但其制作过程相同。前台服务员（Waiter）调度厨师制作套餐。现采用生成器（Builder）模式实现制作过程，得到如图 7 所示的类图。

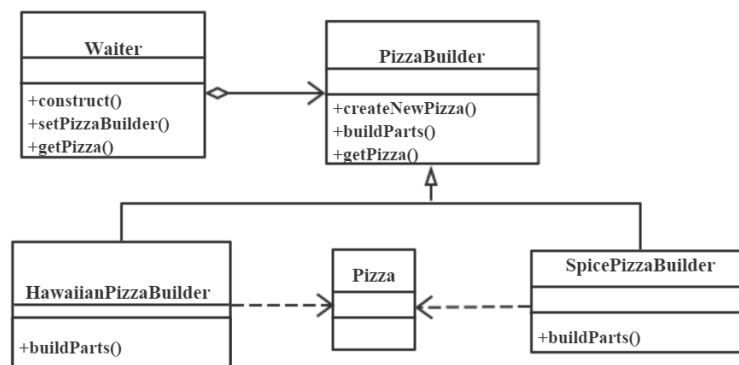


图 7 类图

【Java 代码】

```

class Pizza {
    private String parts;
    public void setParts(String parts) { this.parts = parts; }
}

```

```

        public String toString() { return this.parts; }
    }
    abstract class PizzaBuilder {
        protected Pizza pizza;
        public Pizza getPizza() { return pizza; }
        public void createNewPizza() { pizza = new Pizza(); }
        public (1);
    }
    class HawaiianPizzaBuilder extends PizzaBuilder {
        public void buildParts() {
            pizza.setParts("cross + mild + ham&pineapple");
        }

    class SpicyPizzaBuilder extends PizzaBuilder {
        public void buildParts() { pizza.setParts("pan baked + hot + pepperoni&salami"); }
    }
    class Waiter {
        private PizzaBuilder pizzaBuilder;
        public void setPizzaBuilder(PizzaBuilder pizzaBuilder) { //设置构建器
            (2);
        }
        public Pizza getPizza(){ return pizzaBuilder.getPizza(); }

        public void construct() { //构建
            pizzaBuilder.createNewPizza();
            (3);
        }
    }
    class FastFoodOrdering {
        public static void main(String[] args) {
            Waiter waiter = new Waiter();
            PizzaBuilder hawaiian_pizzabuilder = new HawaiianPizzaBuilder();
            (4);
            (5);
            System.out.println("pizza: " + waiter.getPizza());
        }
    }
}

```

程序的输出结果为:

Pizza:cross + mild + ham&pineapple

全国计算机技术与软件专业技术资格考试

2017 年上半年 软件设计师 上午试卷解析

(1) 参考答案: B

☞ 试题分析 选项 A 程序计数器 (PC) 是存放执行指令的地方, 计算之前就要用到。

选项 B 累加寄存器, 用来暂时存放算术逻辑运算部件 ALU 运算的结果信息。

选项 C 指令寄存器 (IR) 保存当前正在执行的一条指令。

选项 D 地址寄存器 (AR) 用来保存当前 CPU 所要访问的内存单元的地址。

(2) 参考答案: A

☞ 试题分析 在逻辑运算中, 设 A 和 B 为两个逻辑变量, 当且仅当 A 和 B 的取值都为“真”时, A 与 B 的值为“真”; 否则 A 与 B 的值为“假”。当且仅当 A 和 B 的取值都为“假”时, A 或 B 的值为“假”; 否则 A 或 B 的值为“真”。当且仅当 A、B 的值不同时, A 异或 B 为“真”, 否则 A 异或 B 为“假”。对于 16 位二进制整数 a, 其与 0000000000001111 (即十六进制数 000F) 进行逻辑与运算后, 结果的高 12 位都为 0, 低 4 位则保留 a 的低 4 位, 因此, 当 a 的低 4 位全为 0 时, 上述逻辑与运算的结果等于 0。

(3) 参考答案: D

☞ 试题分析 由于 DMA 方式是在 DMA 控制器硬件的控制下实现数据的传送, 不需要 CPU 执行程序, 故这种方式传送的速度最快。另外三种都是通过 CPU 执行某一段程序, 实现计算机内存与外设间的数据交换。

(4) 参考答案: B

☞ 试题分析 串联系统可靠性公式为: $R=R_1 \times R_2 \times \cdots \times R_n$

并联系统可靠性公式为: $R=1-(1-R_1) \times (1-R_2) \times \cdots \times (1-R_n)$

(5) 参考答案: C

☞ 试题分析 在海明码中, 用 K 代表其中有效信息位数, r 表示添加的校验码位, 它们之间的关系满足: $2^r \geq K+r+1$ 。

本题中 K=16, 则要求 $2^r \geq 16+r+1$, 根据计算可以得知 r 的最小值为 5。

(6) 参考答案: A

☞ 试题分析 Cache 存储器用来存放主存的部分拷贝 (副本)。它是按照程序的局部性原理选取出来的最常使用或不久的将来仍将使用的内容。

(7) 参考答案: B

☞ 试题分析 HTTPS (全称: Hyper Text Transfer Protocol over Secure Socket Layer), 是以安全为目的的 HTTP 通道, 简单讲是 HTTP 的安全版。HTTPS 在 HTTP 的基础上加入了 SSL 协议, SSL 依靠证书来验证服务器的身份, 并为浏览器和服务器之间的通信加密。

(8) 参考答案: D


☞ 试题分析 选项 A 的 RSA 是非对称加密算法; 选项 B 的 SHA-1 与选项 C 的 MD5 属于信息摘要算法; 选项 D 的 RC-5 属于非对称加密算法。这些算法中 SHA-1 与 MD5 是不能用来加密数据的, 而 RSA 由于效率问题, 一般不直接用于大量的明文加密。

(9) 参考答案: D

☞ 试题分析 两个证书发放机构 I1 和 I2 互换公钥是 A、B 互信的必要条件。选项 A 和 C 的私钥是不能互换的。选项 B 中要 AB 互信, 其信任基础是建立在 CA 之上的, 如果仅交换 AB 的公钥并不能

解决信任的问题。


(10) 参考答案: A

 **试题分析** 根据《著作权法》第十七条的规定,受委托创作的作品,著作权归属由委托人和受托人通过合同约定。合同未作明确约定或者没有订立合同的,著作权属于受托人。


(11) 参考答案: D

 **试题分析** 根据我国法律法规的规定必须使用注册商标的是烟草类商品。


(12) 参考答案: D

 **试题分析** 根据“同一的发明创造只能被授予一项专利”的规定,在同一天,两个不同的人就同样的发明创造申请专利的,专利局将分别向各申请人通报有关情况,请他们自己去协商解决这一问题,解决的方法一般有两种,一种是两申请人作为一件申请的共同申请人;另一种是其中一方放弃权利并从另一方得到适当的补偿。


(13) 参考答案: A

 **试题分析** 根据尼奎斯特取样定理:如果取样速率大于模拟信号最高频率的 2 倍,则可以用得到的样本中恢复原来的模拟信号。


(14) 参考答案: D

 **试题分析** DPI 为像素/英寸,可以得到 $(300 \times 3) \times (300 \times 4) = 900 \times 1200$ 。

(15)(16) 参考答案: A C


 **试题分析** 接口设计的主要依据是数据流图,接口设计的任务主要是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。定义软件的主要结构元素及其之间的关系是架构阶段的任务;确定软件涉及的文件系统的结构及数据库的表结构是数据存储设计阶段的任务;确定软件各个模块内部的算法和数据结构是详细设计阶段的任务。

(17)(18) 参考答案: D B


 **试题分析** 项目的工期是从开始到结束持续时间最长的工作。题目中持续时间最长的是 ABDIJL,需要时间 20 天。

BD 活动在 AB 活动结束后便可以开始,所以最早开始时间为 4。HK 活动需要在 AEGH 与 ACFH 两条路径上的活动均完成之后,才能开始,所以最早开始时间为 11。


(19) 参考答案: D

 **试题分析** 沟通渠道 $=N(N-1)/2=28$,其中 N 是指参加沟通的人数。由于成员之间没有沟通,所以沟通路径为 7。


(20) 参考答案: B

 **试题分析** 常见的命名对象有:变量、函数、数据类型。

(21) 参考答案: D

 **试题分析** 首先所有选项都是以 b 结尾的,但只有选项 D 中的 $(a|b)^*$ 可以表示 $\{\epsilon, a, b, aa, ab, \dots\}$ 由字符 a、b 构成的所有字符串。


(22) 参考答案: B

 **试题分析** 词法分析阶段:输入源程序,对构成源程序的字符串进行扫描和分解,识别出一个个的单词,删掉无用的信息,报告分析时的错误。


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。


(23) 参考答案: B

 **试题分析** $300 \times 1024 / 4 / 32 = 2400$


(24) 参考答案: B

 试题分析 首先给每个进程分配所需资源数减 1 个资源, 然后系统还有 1 个资源, 则不可能发生死锁。即: $3 \times 4 + 1 = 13$ 个。


(25) 参考答案: C

 试题分析 逻辑地址=页号+页内地址, 为 32 位。物理地址=物理块号+物理地址的页内地址。其中页内地址=物理地址的页内地址。页面大小为 4K, 占 0~11 位 ($4k=4096=2^{12}$), 也就是页内地址有 12 位, 所以十六进制数中的 D16H 是页内地址, 因为 D16H 转化成二进制是 1101 0001 0110, 正好是 12 位, 那么逻辑页号就为 2。查表可知对应的物理块号为 4, 所以物理地址为 4D16H。

(26) (27) (28) 参考答案: B C A


 试题分析 图中有五条箭线, 因此需要设置五个信号量, 按照从小到大的顺序分别是: P1→P2 是 S1, P2→P3 是 S2, P2→P4 是 S3, P3→P4 是 S4, P4→P5 是 S5。每个进程开始的时候执行 P 操作 (P1 没有 P 操作, 只有 V 操作), 结束的时候执行 V 操作, 如 P2 开始的时候执行 P (S1) 的操作, 结束时执行 V (S2) 的操作。其他同理。

(29) 参考答案: C 或 D

 试题分析 螺旋模型是一种演化软件开发过程模型, 它兼顾了快速原型的迭代的特征以及瀑布模型的系统化与严格监控。螺旋模型最大的特点在于引入了其他模型不具备的风险分析, 使软件在无法排除重大风险时有机会停止, 以减小损失。同时, 在每个迭代阶段构建原型是螺旋模型用以减小风险的途径。螺旋模型更适合大型的昂贵的系统级的软件应用。

此题目有点问题, 应该 C 和 D 都是错误的, 所以选择 C 或 D 都会给分。


(30) 参考答案: D

 试题分析 XP 提倡结对编程, 代码所有权是归于整个开发队伍。其中的结对编程就是一种对代码的审查过程, XP 主要解决代码质量低的问题, 编码速度不能改变。

(31) 参考答案: D

 试题分析 选项 D 是 B/S 结构的特点。

(32) 参考答案: D

 试题分析 在结构化设计中, 系统由多个逻辑上相对独立的模块组成, 在模块划分时需要遵循如下原则:

1) 模块的大小要适中。系统分解时需要考虑模块的规模, 过大的模块可能导致系统分解不充分, 其内部可能包括不同类型的功能, 需要进一步划分, 尽量使得各个模块的功能单一; 过小的模块将导致系统的复杂度增加, 模块之间的调用过于频繁, 反而降低了模块的独立性。一般来说, 一个模块的大小使其实现代码在 1~2 页纸之内, 或者其实现代码行数在 50~200 行之间, 这种规模的模块易于实现和维护。

2) 模块的扇入和扇出要合理。一个模块的扇出是指该模块直接调用的下级模块的个数; 扇出大表示模块的复杂度高, 需要控制和协调过多的下级模块。扇出过大一般是因为缺乏中间层次, 应该适当增加中间层次的控制模块; 扇出太小时可以把下级模块进一步分解成若干个子功能模块, 或者合并到它的上级模块中去。一个模块的扇入是指直接调用该模块的上级模块的个数; 扇入大表示模块的复用程度高。设计良好的软件结构通常顶层扇出比较大, 中间扇出较少, 底层模块则有大扇入。一般来说, 系统的平均扇入和扇出系数为 3 或 4, 不应该超过 7, 否则会增大出错的概率。

3) 深度和宽度适当。深度表示软件结构中模块的层数, 如果层数过多, 则应考虑是否有些模块设计过于简单, 看能否适当合并。宽度是软件结构中同一个层次上的模块总数的最大值, 一般说来, 宽度越大系统越复杂, 对宽度影响最大的因素是模块的扇出。在系统设计时, 需要权衡系统的深度和宽度, 尽量降低系统的复杂性, 减少实施过程的难度, 提高开发和维护的效率。

(33) (34) 参考答案: A D

 试题分析 内聚按强度从低到高有以下几种类型:

1) 偶然内聚。如果一个模块的各成分之间毫无关系,则称为偶然内聚,也就是说模块完成一组任务,这些任务之间的关系松散,实际上没有什么联系。

2) 逻辑内聚。几个逻辑上相关的功能被放在同一模块中,则称为逻辑内聚。如一个模块读取各种不同类型外设的输入。尽管逻辑内聚比偶然内聚合理一些,但逻辑内聚的模块各成分在功能上并无关系,即使局部功能的修改有时也会影响全局,因此这类模块的修改也比较困难。

3) 时间内聚。如果一个模块完成的功能必须在同一时间内执行(如系统初始化),但这些功能只是因为时间因素关联在一起,则称为时间内聚。

4) 通信内聚。如果一个模块的所有成分都操作同一数据集或生成同一数据集,则称为通信内聚。

5) 顺序内聚。如果一个模块的各个成分和同一个功能密切相关,而且一个成分的输出作为另一个成分的输入,则称为顺序内聚。


6) 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。

模块完成多个需要按一定的步骤一次完成的功能。(过程相关—控制耦合)。例如:在用程序流程图设计模块时,若将程序流程图中的一部分划出各自组成模块,便形成过程内聚。

7) 信息内聚。模块完成多个功能,各个功能都在同一数据结构上操作,每一项功能有一个唯一的入口点。这个模块将根据不同的要求,确定该模块执行哪一个功能。由于这个模块的所有功能都是基于同一个数据结构(符号表),因此,它是一个信息内聚的模块。


8) 功能内聚。模块的所有成分对于完成单一的功能都是必须的,则称为功能内聚。

(35)(36) 参考答案: A D


 **试题分析** 覆盖 2 条路径就能达到语句覆盖的要求,用 2 个测试用例即可。路径覆盖需要把程序中的 3 条路径均覆盖一遍,需要 3 个用例。

整个程序流程图转化为节点图之后,一共 11 个节点,13 条边,根据环路复杂度公式有: $13-11+2=4$ 。


(37)(38) 参考答案: A D

 **试题分析** 多重继承是编程语言中的概念,多重继承指的是一个类可以继承另外一个类,而另外一个类又可以继承别的类,比如 A 类继承 B 类,而 A 类又可以继承 C 类,这就是多重继承。多重继承可能造成混淆的情况,出现二义性的成员。


(39) 参考答案: D

 **试题分析** 采用面向对象方法进行软件开发,分析阶段,架构师主要关注系统的行为,即系统应该做什么。


(40) 参考答案: A

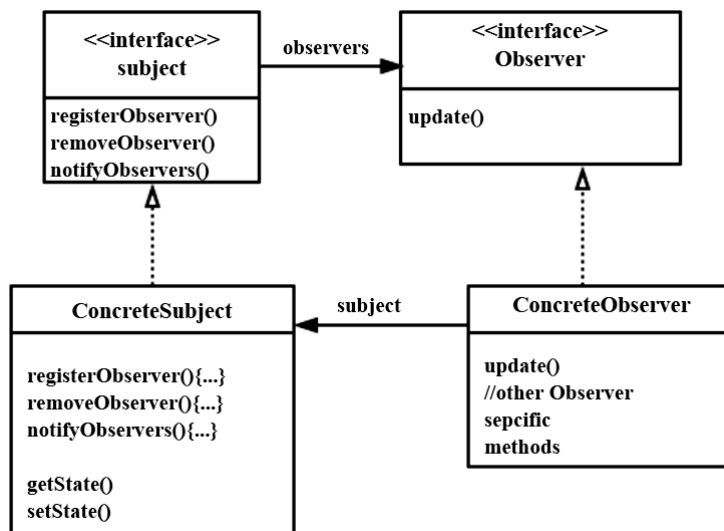
 **试题分析** 多态按字面的意思就是“多种状态”。在面向对象语言中,接口的多种不同的实现方式即为多态。例如不同的足球运动员在都进行射门这个动作时,会产生不同的结果。在运行时,可以通过指向基类的指针,来调用实现派生类中的方法。也就是说客户类其实在调用方法时,并不需要知道特定子类的实现,都会用统一的方式来调用。

(41)(42)(43) 参考答案: C B D

 **试题分析** 通信图也叫作协作图,描述的是对象和对象之间的关系,即一个类操作的实现。简而言之就是,对象和对象之间的调用关系,体现的是一种组织关系。“协作”作为一个结构事物用于表达静态结构和动态行为的概念组合,表达不同事物相互协作完成一个复杂功能。文本框中的“:”号,表示一个对象,“:”号前的部分是对象名,“:”号后面的部分是类名,对象之间连线上方的箭头所标识的是对象之间通信的消息。

(44)(45) 参考答案: C A

 **试题分析** 观察者将自己注册到事件,那么具体的事件就知道了自己的观察者。观察者和事件都有自己的抽象,当实现具体的观察者和事件的时候都要实现相应接口,所以对扩展是开放的。



(46) (47) 参考答案: B D

☞**试题分析** 装饰模式是一种对象结构型模式，可动态地给一个对象增加一些额外的职责，就增加对象功能来说，装饰模式比生成子类实现更为灵活。通过装饰模式，可以在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责；当需要动态地给一个对象增加功能，这些功能可以再动态地被撤销时可使用装饰模式；当不能采用生成子类的方法进行扩充时也可使用装饰模式。

外观模式是对象的结构模式，要求外部与一个子系统的通信必须通过一个统一的外观对象进行，为子系统中的一组接口提供一个一致的界面，外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

(48) 参考答案: C

☞**试题分析** 解析路径为: ACEEBDD。

(49) 参考答案: B

☞**试题分析** 传值与传引用的区别是: 值传递时，实参被拷贝了一份，只能在函数体内使用。因此，a 代表的其实就是 x 本身，只有 a 的变化才能导致 main 函数里面的 x 值的变化。

(50) 参考答案: A

☞**试题分析** 表达式语法树后缀形式，就是对树进行后序遍历（左右根），结果为: x5y+*a/b-。

(51) (52) 参考答案: D C

☞**试题分析** 若事务 T1 对数据 D1 加上共享锁，则其他事务只能再对 D1 加共享锁，而不能加排他锁。

若事务 T2、T3 对数据 D2、D3 加上排他锁，其他事务不能再对 D2、D3 加任何锁。

(53) 参考答案: A

☞**试题分析** 候选关键字是 A1A3, A1A2, 必有的属性是 A1。

(54) (55) (56) 参考答案: A D C

☞**试题分析** 从“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”中判断 SP_P 的联系类型是: 多对多的关系。而选择供应商号和供应零件的数量只能从新的关系模式 SP_P 中选择。

查询条件 Where 与 Having 的区别: Where 是针对单条记录的判断条件，而 Having 是针对分组之后的判断条件。同时，由于考虑到项目号可能重复，所以要加 Distinct 去掉重复的项目。

(57) 参考答案: C

☞**试题分析** 在程序设计中，字符串 (string) 为符号或数值的一个连续序列，如符号串 (一串字符) 或二进制数字串 (一串二进制数字)。选项 A 中的空格也是属于一种字符，都是空格的字符串叫空

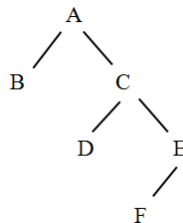
格串，而不是空串；选项 B 的字符串是属于线性结构的；选项 C 字符串的长度是指字符串所有字符个数的总和（包括空格）。

(58) 参考答案：A

☞ 试题分析 操作步骤：a1a2 入，a2 出，a3a4 入，a4 出，a5 入，a5 出，a3 出，a1 出。

(59) 参考答案：B

☞ 试题分析 先序遍历是根左右的方式，中序遍历是左根右的方式。最终的结果如下图所示：



(60) 参考答案：D

☞ 试题分析 按照条件“ $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ ”要求，代入四个选项。以选项 A 为例，当 $i=1$ 时， $K_1(16) < K_2(25)$ ，且 $K_1(16) < K_3(40)$ …依此类推，可得选项 D 不满足要求。

(61) 参考答案：B

☞ 试题分析 二分查找法的原则是：首先，假设表中元素是按升序排列，将表中间位置记录的关键字与查找关键字比较，如果两者相等，则查找成功；否则利用中间位置记录将表分成前、后两个子表，如果中间位置记录的关键字大于查找关键字，则进一步查找前一子表，否则进一步查找后一子表。重复以上过程，直到找到满足条件的记录，使查找成功，或直到子表不存在为止，此时查找不成功。

(62) (63) (64) (65) 参考答案：B B A B

☞ 试题分析 动态规划算法与分治法不同的是，适合于用动态规划求解的问题，经分解得到子问题往往不是互相独立的。若用分治法来解这类问题，则分解得到的子问题数目太多，有些子问题被重复计算了很多次。如果能够保存已解决的子问题的答案，而在需要时再找出已求得的答案，这样就可以避免大量的重复计算，节省时间。可以用一个表来记录所有已解的子问题的答案。不管该子问题以后是否被用到，只要它被计算过，就将其结果填入表中。这就是动态规划法的基本思路。本题中的时间复杂度为 $O(n)$ 。

贪心选择是指所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到。这是贪心算法可行的第一个基本要素，也是贪心算法与动态规划算法的主要区别。

回溯算法实际上一个类似枚举的搜索尝试过程，主要是在搜索尝试过程中寻找问题的解，当发现已不满足求解条件时，就“回溯”返回，尝试别的路径。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。但当探索到某一步时，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为回溯法，而满足回溯条件的某个状态的点称为“回溯点”。

求最短的装配时间与装配路线只需要将选项按照公式代入计算（将图上每条路径上的所有数字相加）可得最短路线为 $S11 \rightarrow S22 \rightarrow S13$ ，时间为 21。

(66) 参考答案：B

☞ 试题分析 本地主机将首先在本机 hosts 文件中查询该网址对应的 IP 地址。

(67) 参考答案：C

☞ 试题分析 Linux 中只有一个根目录，用“/”表示。


(68) 参考答案：C

☞ 试题分析 TCP 与 UDP 是基于 IP 协议的；SMTP 是基于 TCP 协议的。

(69) 参考答案：C

☞ 试题分析 总的速率率为 $(1+7+2) * 500 = 5000$ ，其中有效数据是 7 位，那么有效数据速率为 $5000 * 7 / 10 = 3500 \text{ b/s}$ 。

(70) 参考答案: D


 **试题分析** 静态路由是指由用户或网络管理员手工配置的路由信息。当网络的拓扑结构或链路的状态发生变化时,网络管理员需要手工去修改路由表中相关的静态路由信息。

随机路由使用前向代理来收集网络中的有限全局信息即当前节点到其源节点的旅行时间,并以此来更新节点的旅行时间表。

洪泛路由是一种简单的路由算法,将收到的封包,往所有的可能连结路径上递送,直到封包到达为止。

动态路由就是自适应路由选择算法,是指路由器能够自动地建立自己的路由表,并且能够根据实际情况的变化适时地进行调整。

(71)~(75) 参考答案: A C B D A

 **试题分析**

翻译:软件的优点在于其功能,内部结构以及由团队创建的方式。对于用户来说,通过直观和(71)界面呈现的正确功能的程序是出色的软件。对于软件设计师来说,分割的内部结构是一种简单而直观的方式,最小化内部耦合是美观的。对于开发人员和经理来说,一个积极的开发团队每周都取得重大进展,并且生产无缺陷的代码是件美好的事情。

我们的世界需要大量软件。五十年前,软件是在大多数公司和工业环境中运行的。现在软件存在于我们的手机,手表,电器,汽车,玩具和工具中。并且对新的和更好的软件的需求永远不会(72)。随着我们文明的发展和壮大,随着发展中国家建设基础设施,发达国家努力实现更高的效率,越来越多的软件需求(73)增长。如果在所有的软件中没有美存在的话,这将是一个很大的耻辱。

我们知道软件可能很难使用,有不可靠,粗心大意的结构。我们知道这些结构使得它们变得昂贵和难以改变。我们知道有一些软件系统通过尴尬和繁琐的界面来呈现其功能。我们知道有软件系统崩溃和捣乱行为。这些都是(74)系统。不幸的是,作为专业人士,软件开发人员开发出难用的系统多过开发好用的系统。

这是优秀的软件开发者知道的秘密。好用的软件比难用的更便宜、更快。一个好用的软件系统相对于一个难用的系统来说,建立和维护要花的时间与金钱会少得多。很多新手软件开发人员不明白这一点。他们认为做每一个事情必须快速,更快速。他们认为软件之美是(75)。不!快速开发使软件变得僵硬,难以理解而好用的系统灵活易懂,使得开发和维护工作成为一种快乐。难用的软件不切实际,会减慢速度,会使软件昂贵而脆弱。美观的系统成本最低,建立和维护成本最低,交货时间也最短。

- | | | | |
|--------------|--------|----------|--------|
| (71) A. 简单 | B. 困难 | C. 复杂 | D. 复制品 |
| (72) A. 发生 | B. 存在 | C. 停止 | D. 开始 |
| (73) A. 开始 | B. 持续 | C. 出现 | D. 停止 |
| (74) A. 实用的 | B. 有用的 | C. 美丽的 | D. 丑陋的 |
| (75) A. 不实用的 | B. 完美的 | C. 浪费时间的 | D. 实用的 |

全国计算机技术与软件专业技术资格考试

2017 年上半年 软件设计师 下午试卷解析

试题一

【问题 1】

E1: 供应商

E2: 采购部门

E3: 检验员

E4: 库管员

E5: S/R 职员

【问题 2】

D1: 库存表

D2: 采购订单表

D3: 质量标准表

D4: 供应商表

【问题 3】

数据流名称: 检查库存信息

起点: P1 (检查库存水平)

终点: D1 (部件库存表)

数据流名称: 产品送检

起点: P3 (验证装运部件)

终点: P4 (校验部件质量)

数据流名称: 装运错误通知

起点: P3 (验证装运部件)

终点: E1 (供应商)

数据流名称: 缺陷装运通知

起点: P4 (校验部件质量)

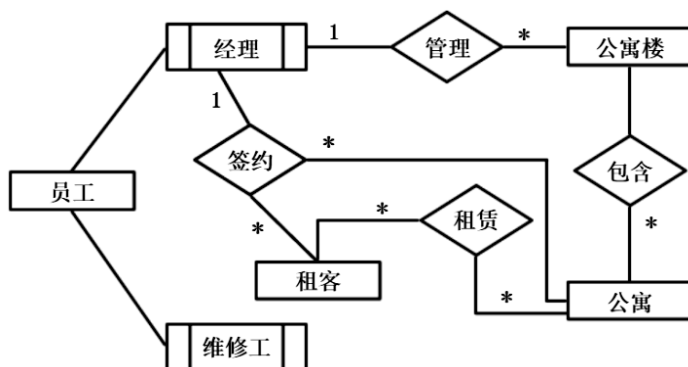
终点: E1 (供应商)

【问题 4】

父图中某个加工的输入输出数据流必须与其子图的输入输出数据流在数量上和内容上保持一致。父图的一个输入（或输出）数据流对应子图中几个输入（或输出）数据流，而子图中组成的这些数据流的数据项全体正好是父图中的这一个数据流。

试题二

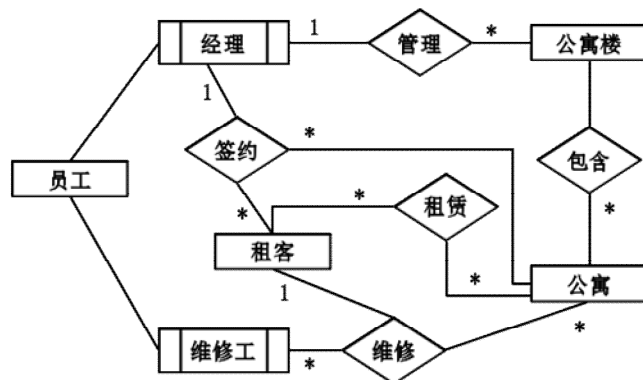
【问题 1】



【问题 2】

(a) 业务技能 (b) 楼编号 (c) 月租金

【问题 3】



新增维修关系，维修工维修公寓，关系模式为维修情况

维修情况（故障编号，员工编号，楼编号，公寓号，维修日期，维修内容）

试题三

【问题 1】

(a): C4、C5、C7、C10、C11

(b): C3、C8

(c): C1、C2、C6、C9、C12

【问题 2】

X1: 收货地址

X2: 支付方式

X3: 邮箱地址

X4: 定制属性

【问题 3】

S1: 订单挂起

S2: 订单备货

S3: 订单定制

S4: 订单发货

S5: 订单收货

试题四

【问题 1】

(1) $\text{first} + (\text{last} - \text{first}) / 2 + 1$ 或 $(\text{first} + \text{last}) / 2 + 1$

(2) $\text{firstSum} < \text{lastSum}$

(3) $\text{first} + (\text{last} - \text{first}) / 2$ 或 $(\text{first} + \text{last}) / 2$

【问题 2】

(4) 分治法 (5) $O(n \log n)$

【问题 3】

(6) 2 (7) 4

试题五

(1) `virtual void buildParts()`

(2) `this->pizzaBuilder=pizzaBuilder`

(3) `pizzaBuilder->buildParts()`

- (4) waiter->setPizzaBuilder(hawaiian_pizzabuilder)
- (5) waiter->construct()

试题六

- (1) abstract void buildParts();
- (2) this.pizzaBuilder=pizzaBuilder
- (3) pizzaBuilder.buildParts()
- (4) waiter.setPizzaBuilder(hawaiian_pizzabuilder)
- (5) waiter.construct()