

2014 年下半年

全国计算机技术与软件专业技术资格考试 2014 年下半年 软件设计师 上午试卷与解析

- 三总线结构的计算机总线系统由 (1) 组成。

(1) A. CPU 总线、内存总线和 IO 总线 B. 数据总线、地址总线和控制总线
C. 系统总线、内部总线和外部总线 D. 串行总线、并行总线和 PCI 总线

🔍 试题分析 总线线路都可以被归为三类：即数据总线、地址总线和控制总线。

参考答案：B

- 计算机采用分级存储体系的主要目的是为了解决 (2) 的问题。

(2) A. 主存容量不足 B. 存储器读写可靠性
C. 外设访问效率 D. 存储容量、成本和速度之间的矛盾

🔍 试题分析 为了解决对存储器要求容量大、速度快、成本低三者之间的矛盾，目前通常采用多级存储器体系结构，即使用高速缓冲存储器、主存储器和外存储器。高速缓冲存储器：高速存取指令和数据，存取速度快，但存储容量小。主存储器：主存存放计算机运行期间的大量程序和数据，存取速度较快，存储容量不大。外存储器：外存存放系统程序和大型数据文件及数据库，存储容量大，成本低。

参考答案：D

- 属于 CPU 中算术逻辑单元的部件是在 (3) 。

(3) A. 程序计数器 B. 加法器 C. 指令寄存器 D. 指令译码器

🔍 试题分析 算术逻辑运算单元 ALU (Arithmetic and Logic Unit) ALU 主要完成对二进制数据的定点算术运算（加减乘除）、逻辑运算（与或非异或）以及移位操作。

参考答案：B

- 内存按字节编址从 A5000H 到 DCFFFH 的区域其存储容量为 (4) 。

(4) A. 123KB B. 180KB C. 223KB D. 224KB

🔍 试题分析 $DCFFFH - A5000H + 1H = 38000H = 224KB$

参考答案：D

- 以下关于 RISC 和 CISC 的叙述中，不正确的是 (5) 。

(5) A. RISC 通常比 CISC 的指令系统更复杂
B. RISC 通常会比 CISC 配置更多的寄存器
C. RISC 编译器的子程序库通常要比 CISC 编译器的子程序库大得多
D. RISC 比 CISC 更加适合 VLSI 工艺的规整性要求

🔍 试题分析 1) 指令系统：RISC 设计者把主要精力放在那些经常使用的指令上，尽量使它们具有简单高效的特色。对不常用的功能，常通过组合指令来完成。而 CISC 计算机的指令系统比较丰富，有专用指令来完成特定的功能。因此，处理特殊任务效率较高。

2) 存储器操作：RISC 对存储器操作有限制，使控制简单化；而 CISC 机器的存储器操作指令多，操作直接。

3) 程序：RISC 汇编语言程序一般需要较大的内存空间，实现特殊功能时程序复杂，不易设计；而 CISC 汇编语言程序编程相对简单，科学计算及复杂操作的程序设计相对容易，效率较高。

4) 设计周期：RISC 微处理器结构简单，布局紧凑，设计周期短，且易于采用最新技术；CISC 微处理器结构复杂，设计周期长。

5) 应用范围: 由于 RISC 指令系统的确定与特定的应用领域有关, 故 RISC 机器更适合于专用机; 而 CISC 机器则更适合于通用机。

参考答案: A

- Flynn 分类法基于信息流特征将计算机分成 4 类, 其中 (6) 只有理论意义而无实例。

(6) A. SISD B. MISD C. SIMD D. MIMD

试题分析 按照 Flynn 分类法, 根据计算机中指令和数据的并行状况可把计算机分成:

单指令流单数据流 (SISD) ——传统的计算机包含单个 CPU, 它从存储在内存中的程序那里获得指令, 并作用于单一的数据流。

单指令流多数据流 (SIMD) ——单个的指令流作用于多于一个的数据流上。例如有数据 4、5 和 3、2, 一个单指令执行两个独立的加法运算: $4+5$ 和 $3+2$, 就被称为单指令流多数据流。SIMD 的一个例子就是一个数组或向量处理系统, 它可以对不同的数据并行执行相同的操作。

多指令流单数据流 (MISD) ——用多个指令作用于单个数据流的情况实际上很少见。这种冗余多用于容错系统。

多指令流多数据流 (MIMD) ——这种系统类似于多个 SISD 系统。实际上, MIMD 系统的一个常见例子是多处理器计算机, 如 Sun 的企业级服务器。

参考答案: B

- 网络系统中, 通常把 (7) 置于 DMZ 区。

(7) A. 网络管理服务器 B. Web 服务器
C. 入侵检测服务器 D. 财务管理服务器

试题分析 DMZ 是为了解决安装防火墙后外部网络不能访问内部网络服务器的问题, 而设立的一个非安全系统与安全系统之间的缓冲区, 这个缓冲区位于企业内部网络和外部网络之间的小网络区域内, 在这个小网络区域内可以放置一些必须公开的服务器设施, 如企业 Web 服务器、FTP 服务器和论坛等。

参考答案: B

- 以下关于拒绝服务攻击的叙述中, 不正确的是 (8)。

(8) A. 拒绝服务攻击的目的是使计算机或者网络无法提供正常的服务
B. 拒绝服务攻击是不断向计算机发起请求来实现的
C. 拒绝服务攻击会造成用户密码的泄露
D. DDOS 是一种拒绝服务攻击形式

试题分析 拒绝服务攻击即攻击者想办法让目标机器停止提供服务, 是黑客常用的攻击手段之一。其实对网络带宽进行的消耗性攻击只是拒绝服务攻击的一小部分, 只要能够对目标造成麻烦, 使某些服务被暂停甚至主机死机, 都属于拒绝服务攻击。拒绝服务攻击问题也一直得不到合理的解决, 究其原因这是因为这是由于网络协议本身的安全缺陷造成的, 从而拒绝服务攻击也成了攻击者的终极手法。攻击者进行拒绝服务攻击, 实际上让服务器实现两种效果: 一是迫使服务器的缓冲区满, 不接收新的请求; 二是使用 IP 欺骗, 迫使服务器把合法用户的连接复位, 影响合法用户的连接。

参考答案: C

- (9) 不是蠕虫病毒。

(9) A. 熊猫烧香 B. 红色代码 C. 冰河 D. 爱虫病毒

试题分析 熊猫烧香是一种经过多次变种的计算机蠕虫病毒, 2006 年 10 月 16 日由 25 岁的中国湖北武汉新洲区人李俊编写, 2007 年 1 月初肆虐中国大陆网络, 它主要透过网络下载的文件植入计算机系统。

“红色代码”病毒是一种新型网络病毒, 其传播所使用的技术可以充分体现网络时代网络安全与病毒的巧妙结合, 将网络蠕虫、计算机病毒、木马程序合为一体, 开创了网络病毒传播的新路, 可称之为划时代的病毒。

“冰河”木马病毒是国人编写的一种黑客性质的病毒, 感染该病毒之后, 黑客就可以通过网络远程

控制该电脑。主要用于远程监控。

“爱虫”病毒，是一种蠕虫病毒，这个病毒可以改写本地及网络硬盘上面的某些文件。用户机器染毒以后，邮件系统将会变慢，并可能导致整个网络系统崩溃。

参考答案: C

- 甲公司接受乙公司委托开发了一项应用软件，双方没有订立任何书面合同。在此情形下，(10)享有该软件的著作权。


(10) A. 甲公司 B. 甲、乙公司共同 C. 乙公司 D. 甲、乙公司均不

试题分析 接受任务开发软件的著作权归属一般按以下两条标准确定：①在合同中明确约定的，按照合同约定实行；②未明确约定的，著作权属于实际完成软件开发的单位。

参考答案: A

- 甲、乙软件公司于2013年9月12日就其财务软件产品分别申请“大堂”和“大唐”商标注册，两财务软件相似，且经协商双方均不同意放弃使用其申请注册的商标标识。此情形下，（11）获准注册。

(11) A. “大堂” B. “大堂”与“大唐”都能
C. “大唐” D. 由甲、乙抽签结果确定谁能

 **试题分析** 本案例中，两个商标发音和功能近似，存在侵权风险，根据软件知识产权管理法案，只能有一个获得注册，抽签决定。

参考答案: D

- 以下媒体中 (12) 是表示媒体, (13) 是表现媒体。

(12) A. 声音 B. 声音编码 C. 超声波 D. 喇叭

(13) A. 声音 B. 声音编码 C. 超声波 D. 喇叭

试题分析 表示媒体是为了加工、处理和传输感觉媒体而人为研究、构造出来的一种媒体，它有各种编码方式，如：文本编码、图像编码和声音编码等。

表现媒体是指进行信息输入和输出的媒体，如：键盘、鼠标、扫描仪、话筒和摄像机等输入媒体以及显示器、打印机和扬声器等输出媒体。

感觉媒体是指直接作用于人的感觉器官，使人产生直接感觉的媒体，如：引起听觉反应的声音，引起视觉反应的文本、图形和图像等。

存储媒体是指用于存储表示媒体的物理介质，如：硬盘、软盘、光盘和胶卷等。

传输媒体是指用于存储表示媒体的物理介质，如：电缆和光缆等。

参考答案: B D

- 显示深度、图像深度是图像显示的重要指标。当(14)时，显示器不能完全反映数字图像使用的全部颜色。

(14) A. 显示深度=图像深度 B. 显示深度>图像深度
C. 显示深度>=图像深度 D. 显示深度<图像深度

试题分析 图像深度图像文件中记录一个像素点所需要的位数，显示深度表示显示缓存中记录屏幕上一个点的位数，也即显示器可以显示的颜色数。因此，显示一幅图像时，屏幕上呈现的颜色效果与图文提供的颜色信息有关，也即与图像深度有关，同时也与显示器当前可容纳的颜色容量有关，也即与显示深度有关。

1) 显示深度大于图像深度: 在这种情况下屏幕上的颜色能较真实地反映图像文件的颜色效果。显示的颜色完全取决于图像的颜色。

2) 显示深度等于图像深度: 这种情况下如果用真彩色显示模式来显示真彩色图像, 或者显示调色板与图像调色板一致时, 屏幕上的颜色能较真实地反映图像文件的颜色效果; 反之, 显示调色板与图像调色板不一致时, 显示色彩会出现失真。

3) 显示深度小于图像深度: 此时显示的颜色会出现失真。

参考答案: D

- 以下关于结构化开发方法的叙述中，不正确的是(15)。

(15) A. 总的指导思想是自顶向下，逐层分解
 B. 基本原则是功能的分解与抽象
 C. 与面向对象开发方法相比，更适合大规模、特别复杂的项目
 D. 特别适合于数据处理领域的项目

☞**试题分析** 结构化系统开发方法(Structured System Development Methodology)是目前应用得最普遍的一种开发方法。其基本思想是用系统的思想和系统工程的方法，按照用户至上的原则结构化、模块化，自顶向下对系统进行分析与设计。它不限于数据处理领域项目。

参考答案：C

- 模块 A、B 和 C 包含相同的 5 个语句，这些语句之间没有联系，为了避免重复，把这 5 个模块抽取出来组成模块 D。则模块 D 的内聚类型为(16)内聚。

(16) A. 功能 B. 通信 C. 逻辑 D. 巧合

☞**试题分析** 功能内聚：完成一个单一功能，各个部分协同工作，缺一不可。

顺序内聚：处理元素相同，而且必须顺序执行。

通信内聚：所有处理元素集中在一个数据结构的区域上。

过程内聚：处理元素相关，而且必须按待定的次序执行。

瞬时内聚：所包含的任务必须在同一时间间隔内执行(如初始化模块)。

逻辑内聚：完成逻辑上相关的一组任务。

偶然内聚：完成一组没有关系或松散关系的任务。

参考答案：D

- 某个项目在开发时采用了不成熟的前沿技术，由此而带来的风险属于(17)风险。

(17) A. 市场 B. 技术 C. 经济 D. 商业

☞**试题分析** 技术不成熟属于技术范围的风险。

参考答案：B

- 属于面向对象、解释型程序设计语言的是(18)。

(18) A. XML B. Python C. Prolog D. C++

☞**试题分析** XML：是一种用于标记电子文件使其具有结构性的标记语言。

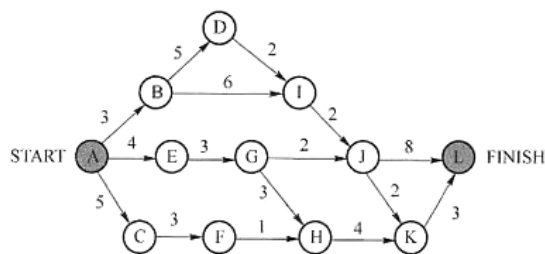
C++：是在 C 语言的基础上发展起来的，主要增加了类的功能，使其成为面向对象的程序设计语言。

Prolog：是以特殊的逻辑推理形式回答用户的查询，经常用于数据库和专家系统。

Python：是一种面向对象，解释型计算机程序设计语言。Python 语法简洁而清晰，具有丰富和强大的类库。它能够把用其他语言制作的模块（尤其是 C/C++）很轻松地联结在一起。

参考答案：B

- 下图是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示活动，边的权重表示活动的持续时间。则里程碑(19)在关键路径上。活动 GH 的松弛时间是(20)。



(19) A. B B. E C. C D. K

(20) A. 0 B. 1 C. 2 D. 3

☞**试题分析** 关键路径为 A-B-D-I-J-L

GH 活动最早开始时间是 7；最晚开始时间是 10，所以松弛时间是 3。

参考答案：A D

- 算术表达式“(a-b)*(c+d)”的后缀式是 (21)。

(21) A. ab-cd+* B. abcd-*+ C. ab-*cd+ D. ab-c+d*

☞**试题分析** 后缀式是波兰逻辑学家卢卡西维奇发明的一种表达方式，把运算符号写在运算对象的后面，例如把 a+b 写成 ab+，这种表示法的优点是根据运算对象和算符的出现次序进行计算，不需要使用括号。

参考答案：A

- 将高级语言源程序翻译成机器语言程序的过程，常引入中间代码。以下关于中间代码的叙述中，不正确的是 (22)。

(22) A. 中间代码不依赖于具体的机器 B. 使用中间代码可提高编译程序的可移植性
C. 中间代码可以用树或图表示 D. 中间代码可以用栈和队列表示

☞**试题分析** 中间代码是源程序的一种内部表示，或称中间语言。中间代码的作用是可使编译程序的结构在逻辑上更为简单明确，使用中间代码可提高编译程序的可移植性，常见的有逆波兰记号、四元式、三元式和树。

参考答案：D

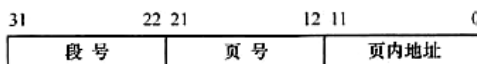
- 假设系统采用 PV 操作实现进程同步与互斥，若 n 个进程共享两台打印机，那么信号量 S 的取值范围为 (23)。

(23) A. -2~n B. -(n-1)~1 C. -(n-1)~2 D. -(n-2)~2

☞**试题分析** 信号量初值等于资源数量，即为 2，由于同时最多有 2 个进程访问打印机，其余进程必须处理等待状态，故 S 的最小值为-(n-2)。

参考答案：D

- 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 (24)。



(24) A. 最多可有 2048 个段，每个段的大小均为 2048 个页，页的大小为 2K
B. 最多可有 2048 个段，每个段最大允许有 2048 个页，页的大小为 2K
C. 最多可有 1024 个段，每个段的大小均为 1024 个页，页的大小为 4K
D. 最多可有 1024 个段，每个段最大允许有 1024 个页，页的大小为 4K

参考答案：D

☞**试题分析** 段号数量 $2^{10}=1024$ ，段内最大页数 $2^{10}=1024$ ，页大小 $2^{12}=4096$ 字节。

- 假设磁盘块与缓冲区大小相同，每个盘块读入缓冲区的时间为 $10\mu s$ ，由缓冲区送至用户区的时间是 $5\mu s$ ，系统对每个磁盘块数据的处理时间为 $2\mu s$ ，若用户需要将大小为 10 个磁盘块的 Doc 文件逐块从磁盘读入缓冲区，并送至用户区进行处理，那么采用单缓冲区需要花费时间为 (25) μs ；采用双缓冲区需要花费的时间为 (26) μs 。

(25) A. 100 B. 107 C. 152 D. 170

(26) A. 100 B. 107 C. 152 D. 170

☞**试题分析** 单缓冲区： $(10+5)\times 10+2=152$

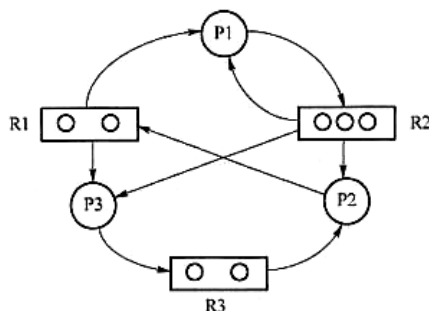
双缓冲区： $10\times 10+5+2=107$

参考答案：C B

- 在如下所示的进程资源图中， (27)；该进程资源图是 (28)。

(27) A. P1、P2、P3 都是阻塞节点
B. P1 是阻塞节点，P2、P3 是非阻塞节点
C. P1、P2 是阻塞节点，P3 是非阻塞节点

D. P1、P2 是非阻塞节点，P3 是阻塞节点



- (28) A. 可以化简的，其化简顺序为 P1->P2->P3
 B. 可以化简的，其化简顺序为 P3->P1->P2
 C. 可以化简的，其化简顺序为 P2->P1->P3
 D. 不可以化简的，因为 P1、P2、P3 申请的资源都不能得到满足

试题分析 R1 已经全部分配给 P1 和 P3，所以 P2 再请求一个 R1 的时候，将进入阻塞状态。同理，R2 已经全部分配给 P1、P2 和 P3，当 P1 再请求一个 R2 时，将陷入阻塞。R3 还有一个未用资源，当 P3 申请时，可以顺利获得，故不会阻塞。

因为 P3 非阻塞且非孤立，所以可以化简。将其所用资源归还资源图后，P1 获得 R2，即可运行，然后也可以化简，最后 P2 可以运行。

参考答案：C B

- 以下关于增量模型的叙述中，正确的是 (29)。

- (29) A. 需求被清晰定义 B. 可以快速构造核心产品
 C. 每个增量必须要进行风险评估 D. 不适宜商业产品的开发

试题分析 增量模型融合了瀑布模型的基本成分(重复应用)和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时，第 1 个增量往往是核心的产品，即第 1 个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。

参考答案：B

- 以下关于 CMM 的叙述中，不正确的是 (30)。

- (30) A. CMM 是指软件过程能力成熟度模型
 B. CMM 根据软件过程的不同成熟度划分了 5 个等级，其中，1 级被认为成熟度最高，5 级被认为成熟度最低
 C. CMMI 的任务是将已有的几个 CMM 模型结合在一起，使之构造成为“集成模型”
 D. 采用更成熟的 CMM 模型，一般来说可以提高最终产品的质量

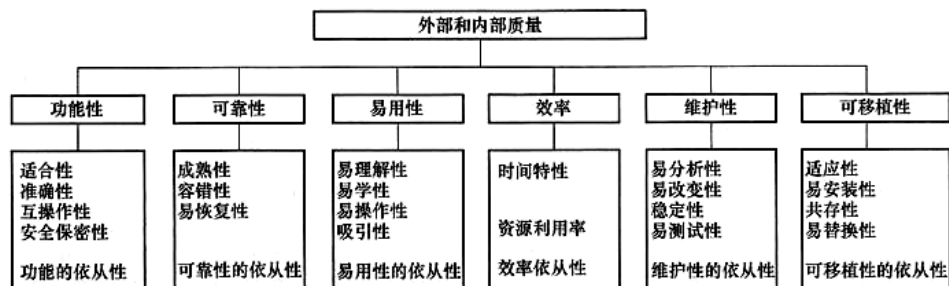
试题分析 CMM 模型指“能力成熟度模型”，其英文全称为 Capability Maturity Model for Software，英文缩写为 SW-CMM，简称 CMM。它是对于软件组织在定义、实施、度量、控制和改善其软件过程的实践中各个发展阶段的描述。CMM 的核心是把软件开发视为一个过程，并根据这一原则对软件开发和维护进行过程监控和研究，以使其更加科学化、标准化、使企业能够更好地实现商业目标。CMM 模型分为 5 个等级：①初始级；②可重复级；③已定义级；④定量管理级；⑤优化级。

参考答案：B

- 在 ISO/IEC 软件质量模型中，可靠性是指在规定的二段时间内和规定的条件下，软件维持在其性能水平的能力；其子特性不包括 (31)。

- (31) A. 成熟性 B. 容错性 C. 易恢复 D. 可移植性

● 试题分析 可靠性包含四个子特征：成熟性、容错性、易恢复和兼容性。



参考答案：D

- 在软件开发过程中，系统测试阶段的测试目标来自于 (32) 阶段。

(32) A. 需求分析 B. 概要设计 C. 详细设计 D. 软件实现

● 试题分析 系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。

参考答案：A

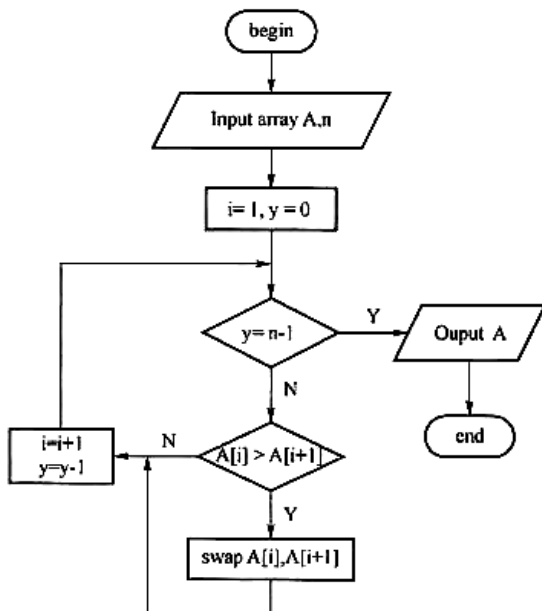
- 以下关于文档的叙述中，不正确的是 (33)。

(33) A. 项目相关人员可以通过文档进行沟通
B. 编写文档会降低软件开发的效率
C. 编写高质量文档可以提高软件开发的质量
D. 文档是软件的不可或缺的部分

● 试题分析 在软件的不同阶段，需要编写不同的文档。因为我们在实际开发过程中，会发现之前设计不合理的地方或者是考虑不周的地方，这就需要对之前的文档进行修改。

参考答案：B

- 下图所示的程序流程图中有 (34) 条不同的简单路径，采用 McCabe 度量法计算该程序图的环路复杂性为 (35)。



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

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

● **试题分析** 三条简单路径分别为 $\text{begin} \rightarrow y=n-1 \text{ 成立} \rightarrow \text{output A} \rightarrow \text{end}$; $\text{begin} \rightarrow y=n-1 \text{ 不成立} \rightarrow A[i] > A[i+1] \text{ 不成立} \rightarrow y=n-1 \text{ 成立} \rightarrow \text{end}$; $\text{begin} \rightarrow y=n-1 \text{ 不成立} \rightarrow A[i] > A[i+1] \text{ 成立} \rightarrow \text{swapA}[i], A[i+1] \rightarrow y=n-1 \text{ 成立} \rightarrow \text{end}$;

MacCabe 度量法公式为 $V(G)=m-n+2$, $V(G)$ 是有向图 G 中环路数, m 是图 G 中弧数, n 是图 G 中结点数。本题中 $m=10$, $n=9$, 故 $V(G)=3$ 。

参考答案: A A

- 以下关于软件维护和可维护性的叙述中, 不正确的是 (36)。

- (36) A. 软件维护要解决软件产品交付用户之后运行中发生的各种问题
B. 软件的维护期通常比开发期长得多, 其投入也大得多
C. 进行质量保证审查可以提高软件产品的可维护性
D. 提高可维护性是在软件维护阶段考虑的问题

● **试题分析** 在软件设计阶段就应该考虑软件的可维护性。

参考答案: D

- 类 (37) 之间存在着一般和特殊的关系。

- (37) A. 汽车与轮船 B. 交通工具与飞机
C. 轮船与飞机 D. 汽车与飞机

● **试题分析** 一般是统称, 特殊指特定具体的。

参考答案: B

- 多态分为参数多态、包含多态、过载多态和强制多态四种不同形式, 其中 (38) 多态在许多语言中都存在, 最常见的例子就是子类型化。

- (38) A. 参数 B. 包含 C. 过载 D. 强制

● **试题分析** 多态分为两种: 通用的多态和特定的多态。两者的区别是前者对工作的类型不加限制, 允许对不同类型的值执行相同的代码; 后者只对有限数量的类型有效, 而对不同类型的值可能要执行不同的代码。

通用的多态又分为参数多态 (parametric) 和包含多态 (Inclusion); 特定的多态分为过载多态 (overloading) 和强制多态 (coercion)。

强制多态: 编译程序通过语义操作, 把操作对象的类型强行加以变换, 以符合函数或操作符的要求。程序设计语言中基本类型的大多数操作符, 在发生不同类型的数据进行混合运算时, 编译程序一般都进行强制多态。

过载 (overloading) 多态: 同一个名 (操作符、函数名) 在不同的上下文中有不同的类型, 程序设计语言中基本类型的大多数操作符都是过载多态的。

参数多态: 采用参数化模板, 通过给出不同的类型参数, 使得一个结构有多种类型。

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

参考答案: B

- 在面向对象程序设计语言中, 对象之间通过 (39) 方式进行通信。以下关于好的面向对象程序设计语言的叙述中, 不正确的是 (40)。

- (39) A. 消息传递 B. 继承 C. 引用 D. 多态

- (40) A. 应该支持被封装的对象 B. 应该支持类写实例的概念
C. 应该支持通过指针进行引用 D. 应该支持继承和多态

● **试题分析** 对象间通过接口传递消息, 实现通信。B, C, D 为实现概念。

C 只针对部分语言, 如 C++, 不具有代表性。

参考答案: A C

- UML 中有 4 种事物: 结构事物、行为事物、分组事物和注释事物。类、接口、构建属于 (41) 事物; 依附于一个元素或一组元素之上对其进行约束或解释的简单符号为 (42) 事物。

(41) A. 结构 B. 行为 C. 分组 D. 注释

(42) A. 结构 B. 行为 C. 分组 D. 注释

● **试题分析** 事物：模型中的基本成员。UML 中包括结构事物、行为事物、分组事物和注释事物。

1) 结构事物：模型中静态部分。

[类 Class]+[接口 Interface]+[协作 Collaboration]

+ [用例 UseCase]+[活动类]+[组件 Component]+[结点 Node]

2) 行为事物：模型中的动态部分。

[交互]+[状态机]

3) 分组事物：可以把分组事物看成是一个“盒子”，模型可以在其中被分解。目前只有一种分组事物，即包(Package)。结构事物、动作事物、甚至分组事物都有可能放在一个包中。包纯粹是概念上的，只存在于开发阶段，而组件在运行时存在。

4) 注释事物：注释事物是 UML 模型的解释部分。

参考答案：A D

● 一组对象以定义良好但是复杂的方式进行通信，产生的相互依赖关系结构混乱且难以理解。采用 (43) 模式，用一个中介对象来封装一系列的对象交互，从而使各对象不需要显式地相互引用，使其耦合松散，而且可以独立地改变它们之间的交互。此模式与 (44) 模式是相互竞争的模式，主要差别是：前者的中介对象封装了其他对象间的通信，而后者通过引入其他对象来分布通信。

(43) A. 解释器(Interpreter) B. 策略(Strategy)
C. 中介者(Mediator) D. 观察者(Observer)

(44) A. 解释器(Interpreter) B. 策略(Strategy)
C. 中介者(Mediator) D. 观察者(Observer)

● **试题分析** 中介者模式包装了一系列对象相互作用的方式，使得这些对象不必互相明显引用。从而使它们可以较松散地耦合。当这些对象中的某些对象之间的相互作用发生改变时，不会立即影响到其他的一些对象之间的相互作用。从而保证这些相互作用可以彼此独立地变化。在中介者模式中，所有的成员对象都可以协调工作，但是又不直接相互管理。这些对象都与一个处于中心地位的中介者对象发生紧密的关系，由这个中介者对象进行协调工作。这个协调者对象叫作中介者(Mediator)，而中介者所协调的成员对象称作同事(Colleague)对象。

在观察者模式中，一个目标物件管理所有相依于它的观察者物件，并且在它本身的状态改变时主动发出通知。这通常透过呼叫各观察者所提供的方法来实现。此种模式通常被用来实作事件处理系统。

参考答案：C D

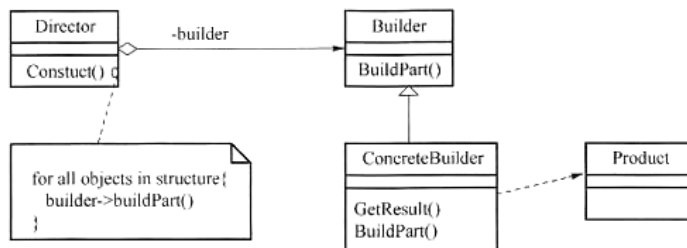
● UML 图中，一张交互图显示一个交互，由一组对象及其之间的关系组成，包含它们之间可能传递的消息。(45) 不是交互图。

(45) A. 序列图 B. 对象图 C. 通信图 D. 时序图

● **试题分析** 对象图(Object Diagram)是显示了一组对象和他们之间的关系。使用对象图来说明数据结构，类图中的类或组件等的实例的静态快照。

参考答案：B

● 下图所示为 (46) 设计模式，适用于 (47)。



- (46) A. 抽象工厂(Abstract Factory) B. 生成器(Builder)
 C. 工厂方法(Factory Method) D. 原型(Prototype)
- (47) A. 一个系统要由多个产品系列中的一个来配置时
 B. 当一个类希望由它的子类来指定它所创建的对象时
 C. 当创建复杂对象的算法应该独立于该对象的组成部分及其装配方式时
 D. 当一个系统应该独立于它的产品创建、构成和表示时

● **试题分析** 生成器是又称建造模式，是一种对象构建模式。它可以将复杂对象的建造过程抽象出来（抽象类别），使这个抽象过程的不同实现方法可以构造出不同表现（属性）的对象。该模式通常包含 Builder, ConcreteBuilder, Director 和 Product 四部分。

参考答案: B C

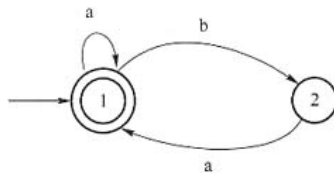
- 对高级语言源程序进行编译的过程可以分为多个阶段，分配寄存器的工作在 (48) 阶段进行。

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

● **试题分析** 目标代码生成阶段应考虑直接影响到目标代码速度的三个问题：一是如何生成较短的目标代码；二是如何充分利用计算机中的寄存器，减少目标代码访问存储单元的次數；三是如何充分利用计算机指令系统的特点，以提高目标代码的质量。

参考答案: D

- 以下关于下图所示有限自动机的叙述中，不正确的是 (49)。



- (49) A. 该自动机识别的字符串中 a 不能连续出现
 B. 自动机识别的字符串中 b 不能连续出现
 C. 自动机识别的非空字符串必须以 a 结尾
 D. 自动机识别的字符串可以为空串

● **试题分析** 图中 a 可代表两个步骤：状态 1-→1，状态 2-→1。如果两个 a 连续出现，则无法区分。

参考答案: A

- 对于大多数通用程序设计语言，用 (50) 描述其语法即可。

- (50) A. 正规文法 B. 上下文无关文法
 C. 上下文有关文法 D. 短语结构文法

● **试题分析** 上下文无关文法：形式语言理论中一种重要的变换文法，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文法，因此应用十分广泛。

参考答案: B

- 在数据库逻辑结构设计阶段，需要 (51) 阶段形成的 (52) 作为设计依据。

- (51) A. 需求分析 B. 概念结构设计 C. 物理结构设计 D. 数据库运行和维护
 (52) A. 程序文档、数据字典和数据流图 B. 需求说明文档、程序文档和数据流图
 C. 需求说明文档、数据字典和数据流图 D. 需求说明文档、数据字典和程序文档

● **试题分析** 数据库的设计过程，按照规范的设计方法，一般分为以下六个阶段：

- 1) 需求分析：分析用户的需求，包括数据、功能和性能需求；
- 2) 概念结构设计：主要采用 E-R 模型进行设计，包括画 E-R 图；
- 3) 逻辑结构设计：通过将 E-R 图转换成表，实现从 E-R 模型到关系模型的转换；
- 4) 数据库物理设计：主要是为所设计的数据库选择合适的存储结构和存取路径；

- 5) 数据库的实施：包括编程、测试和试运行；
6) 数据库运行与维护：系统的运行与数据库的日常维护。

参考答案：A C

- 给定关系模式 R(A, B, C, D, S(C, D, E, 与 $\pi_{1,3,5}(\sigma_{2='软件工程'}(R^0S))$)等价的 SQL 语句如下：

SELECT (53) FROM, SWHERE (54) ;

下列查询 B="信息"且 E="北京"的 A、B、E 的关系代数表达式中，查询效率最高的是 (55)。

(53) A. A,C,S,C B. A,B,E C. A,R,C,E D. A,R,C,S,D

(54) A. B='软件工程' ORR. C=S. CAND R. D=S. D
B. B='软件工程' ORR. C=S. CANDR. D=S. D
C. B='软件工程' ORR. C=S. CORR. D=S. D
D. B='软件工程' AND R. C=S. C AND R. D=S. D

(55) A. $\pi_{1, 2, 7}(\sigma_{2='信息', \wedge 3=5 \wedge 4=6 \wedge 7='北京'}(R \times S))$
B. $\pi_{1, 2, 7}(\sigma_{3=5 \wedge 4=6}(\sigma_{2='信息'}(R) \times \sigma_{5='北京'}(S)))$
C. $\pi_{1, 2, 7}(\sigma_{3=5 \wedge 4=6 \wedge 2=''}(R \times \sigma_{7=''}(S)))$
D. $\pi_{1, 2, 7}(\sigma_{3=5 \wedge 4=6 \wedge 7='北京'}(\sigma_{2='信息'}(R) \times (S)))$

☞ 试题分析 B 的最长查询语句最短，因此可以并行执行，提高效率。

参考答案：C D B

- 给定关系模式 R(U, F, U=(A, B, C, D, E, H)，函数依赖集 F={A->B, A->C, C->D, AE->H)。关系模式 R 的候选关键字为 (56)。

(56) A. AC B. AB C. AE D. DE

☞ 试题分析 通过 AE 关键字可以关联 A、B、C、D、E、H 所有属性，其他选择均不可。

参考答案：C

- 对于线性表，相对于顺序存储，采用链表存储的缺点是 (57)。

(57) A. 数据元素之间的关系需要占用存储空间，导致存储密度不高
B. 表中结点必须占用地址连续的存储单元，存储密度不高
C. 插入新元素时需要遍历整个链表，运算的时间效率不高
D. 删除元素时需要遍历整个链表，运算的时间效率不高

☞ 试题分析 链接需要额外的空间存储结点之间的连接关系，因此存储密度不高，但其优点是插入和删除单个元素的时间复杂度为 O(1)。

参考答案：A

- 若一个栈初始为空，其输入序列是 1, 2, 3, ..., n-1, n，其输出序列的第一个元素是 k(1≤k≤n/2)，则输出序列的最后一个元素是 (58)。

(58) A. 1 B. n C. n-1 D. 不确定的

☞ 试题分析 因为题目中没指出出栈的顺序，因此输出的最后一个元素是不确定的。

参考答案：D

- 某个二叉查找树（即二叉排序树）中进行查找时，效率最差的情形是该二叉查找树是 (59)。

(59) A. 完全二叉树 B. 平衡二叉树 C. 单枝树 D. 满二叉树

☞ 试题分析 单枝树极度不平衡，查找的平均时间复杂度为 O(N)。

参考答案：C

- 在字符串的 KMP 模式匹配算法中，需先求解模式串的 next 函数值，其定义如下式所示，j 表示模式串中字符的序号（从 1 开始）。若模式串 p 为 "abaac"，则其 next 函数值为 (60)。

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

(60) A. 01234 B. 01122 C. 01211 D. 01111

☞**试题分析** 根据公式依次推导即可。

参考答案: B

- 快速排序算法在排序过程中,在待排序数组中确定一个元素为基准元素,根据基准元素把待排序数组划分成两个部分,前面一部分元素值小于基准元素,而后面一部分元素值大于基准元素。然后再分别对前后两个部分进一步进行划分。根据上述描述,快速排序算法采用了 (61) 算法设计策略。已知确定着基准元素操作的时间复杂度为 $O(n)$,则快速排序算法的最好和最坏情况下的时间复杂度为 (62)。

(61) A. 分治 B. 动态规划 C. 贪心 D. 回溯

(62) A. $O(n)$ 和 $O(n\lg n)$ B. $O(n)$ 和 $O(n^2)$
C. $O(n\lg n)$ 和 $O(n\lg n)$ D. $O(n\lg n)$ 和 $O(n^2)$

☞**试题分析** 将数据分成若干份,每份单独处理后再合并,其思想为分治。

理想情况下,快速排序每次将数据划分为规模相近的两部分,并递归至不可再划分,因此其时间复杂度为 $O(n\lg n)$ 。在最坏情况下,每次划分都极不均匀,如一个类别中仅有一个元素,另一个类别中包含剩余所有元素。这时划分的复杂度为 $O(n)$,“次操作的总复杂度为 $O(n^2)$ ”。

参考答案: A D

- 对一待排序序列分别进行直接插入排序和简单选择排序,若待排序序列中有两个元素的值相同,则 (63) 保证这两个元素在排序前后的相对位置不变。

(63) A. 直接插入排序和简单选择排序都可以 B. 直接插入排序和简单选择排序都不能
C. 只有直接插入排序可以 D. 只有简单选择排序可以

☞**试题分析** 它的工作原理是通过构建有序序列,对于未排序数据,在已排序序列中从后向前扫描,找到相应位置并插入。因为元素扫描顺序固定,所以对于相同元素,可以保证后出现的元素一定在之前出现元素的后面。

参考答案: C

- 已知一个文件中出现的各个字符及其对应的频率如下表所示。若采用定长编码,则该文件中字符的码长应为 (64)。若采用 Huffman 编码,则字符序列“face”的编码应为 (65)。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

(64) A. 2 B. 3 C. 4 D. 5

(65) A. 110001001101 B. 001110110011
C. 101000010100 D. 010111101011

☞**试题分析** ①有 6 个不同字母,需要采用 3 位二进制进行编码。

②哈夫曼静态编码:它对需要编码的数据进行两遍扫描:第一遍统计原数据中各字符出现的频率,利用得到的频率值创建哈夫曼树,并必须把树的信息保存起来,即把字符 $0 \sim 255(2^8=256)$ 的频率值以 $2 \sim 4$ BYTES 的长度顺序存储起来,(用 4Bytes 的长度存储频率值,频率值的表示范围为 $0 \sim 2^{32}-1$,这已足够表示大文件中字符出现的频率了。)以便解压时创建同样的哈夫曼树进行解压;第二遍则根据第一遍扫描得到的哈夫曼树进行编码,并把编码后得到的码字存储起来。

参考答案: B A

- PPP 中的安全认证协议是 (66),它使用三次握手的会话过程传送密文。

(66) A. MDS B. PAP C. CHAP D. NCP

☞**试题分析** 1) 安全认证介绍

①PPP 的 NCP 可以承载多种协议的三层数据包。

②PPP 使用 LCP 控制多种链路的参数(建立、认证、压缩、回拨)。

2) PPP 的认证类型


- ①PPP 的 pap 认证是通过二次握手建立认证（明文不加密）。
- ②PPP 的 chap 挑战握手认证协议，通过三次握手建立认证（密文采用 MEDS 加密）。
- ③PPP 的双向验证，采用的是 chap 的主验证风格。
- ④PPP 的加固验证，采用的是两种（pap、chap）验证同时使用。

参考答案：C

- ICMP 协议属于因特网中的 (67) 协议，ICMP 协议数据单元封装在 (68) 中传送。

(67) A. 数据链路层 B. 网络层 C. 传输层 D. 会话层

(68) A. 以太帧 B. TCP 段 C. UDP 数据报 D. IP 数据报

 **试题分析** ICMP 是 TCP/IP 协议族的一个子协议，属于网络层协议，主要用于在主机与路由器之间传递控制信息，包括报告错误、交换受限控制和状态信息等，和 TCP、UDP、IGMP 等协议一样，ICMP 协议使用 IP 数据报传送数据。

参考答案：B D


- DHCP 客户端可以从 DHCP 服务器获得 (69)。

(69) A. DHCP 服务器的地址和 Web 服务器的地址

B. DNS 服务器的地址和 DHCP 服务器的地址

C. 客户端地址和邮件服务器地址

D. 默认网关的地址和邮件服务器地址

 **试题分析** DNS 服务器地址一般由网络宽带服务商提供，一般是当地主要机房的服务器解析地址。它无法通过 DHCP 客户端得到。

参考答案：(69) B


- 分配给某公司网络的地址块是 210.115.192.0/20，该网络可以被划分为 (70) 个 C 类子网。

(70) A. 4

B. 8

C. 16

D. 32

 **试题分析** 子网掩码的前 20 位为 1，还有 4 位可以表示不同的子网，故最多可以划分为 $2^4=16$ 个子网。

参考答案：C

- Teams are required for most engineering projects. Although some small hardware or software products can be developed by individuals, the scale and complexity of modern systems is such, and the demand for short schedules so great, that it is no longer (71) for one person to do most engineering jobs. Systems development is a team (72), and the effectiveness of the team largely determines the (73) of the engineering.

Development teams often behave much like baseball or basketball teams. Even though they may have multiple specialties, all the members work toward (74). However, on systems maintenance and enhancement teams, the engineers often work relatively independently, much like wrestling and track teams.

A team is (75) just a group of people who happen to work together. Teamwork takes practice and it involves special skills. Teams require common processes; they need agreed-upon goals; and they need effective guidance and leadership. The methods for guiding and leading such teams are well known, but they are not obvious.

(71) A. convenient B. existing

C. practical

D. real

(72) A. activity

B. job

C. process

D. application

(73) A. size

B. quality

C. scale

D. complexity

(74) A. multiple objectives

B. different objectives

C. a single objective

D. independent objectives

(75) A. relatively

B. /

C. only

D. more than

参考答案：C A B C D

全国计算机技术与软件专业技术资格考试

2014 年下半年 软件设计师 下午试卷与解析

试题一（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某大型披萨加工和销售商为了有效管理生产和销售情况，欲开发一披萨信息系统，其主要功能如下：

（1）销售。处理客户的订单信息，生成销售订单，并将其记录在销售订单表中。销售订单记录了订购者、所订购的披萨、期望的交付日期等信息。

（2）生产控制。根据销售订单以及库存的披萨数量，制定披萨生产计划（包括生产哪些披萨、生产顺序和生产量等），并将其保存在生产计划表中。

（3）生产。根据生产计划和配方表中的披萨配方，向库存发出原材料申领单，将制作好的披萨的信息存入库存表中，以便及时进行交付。

（4）采购。根据所需原材料及库存量，确定采购数量，向供应商发送采购订单，并将其记录在采购订单表中；得到供应商的供应量，将原材料数量记录在库存表中，在采购订单表中标记已完成采购的订单。

（5）运送。根据销售订单将披萨交付给客户，并记录在交付记录表中。

（6）财务管理。在披萨交付后，为客户开具费用清单，收款并出具收据：依据完成的采购订单给供应商支付原材料费用并出具支付细节；将收款和支付记录存入收支记录表中。

（7）存储。检查库存的原材料、披萨和未完成订单，确定所需原材料。

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

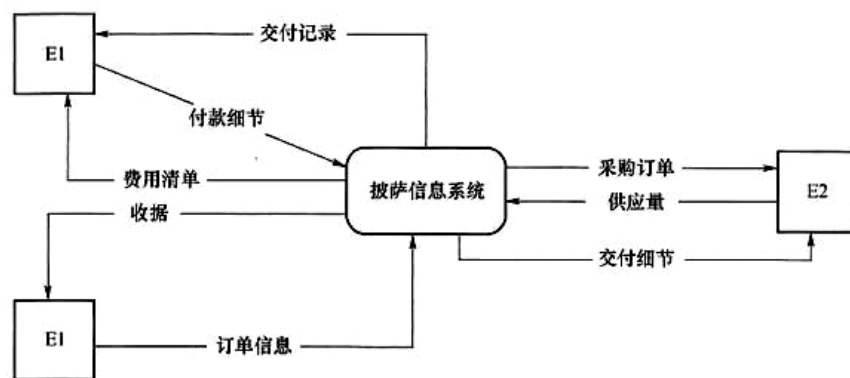


图 1 上下数据流图

【问题 1】（5 分）

根据说明中的词语，给出图 1 中的实体 E1~E2 的名称。

答案：E1：客户；E2：供应商

【解析】

该题以披萨信息系统为载体来考查学生对数据流图知识点的掌握程度。从题目的问答形式上来看，和往年相似，要求补充外部实体、补充缺失数据流、找出外部存储。

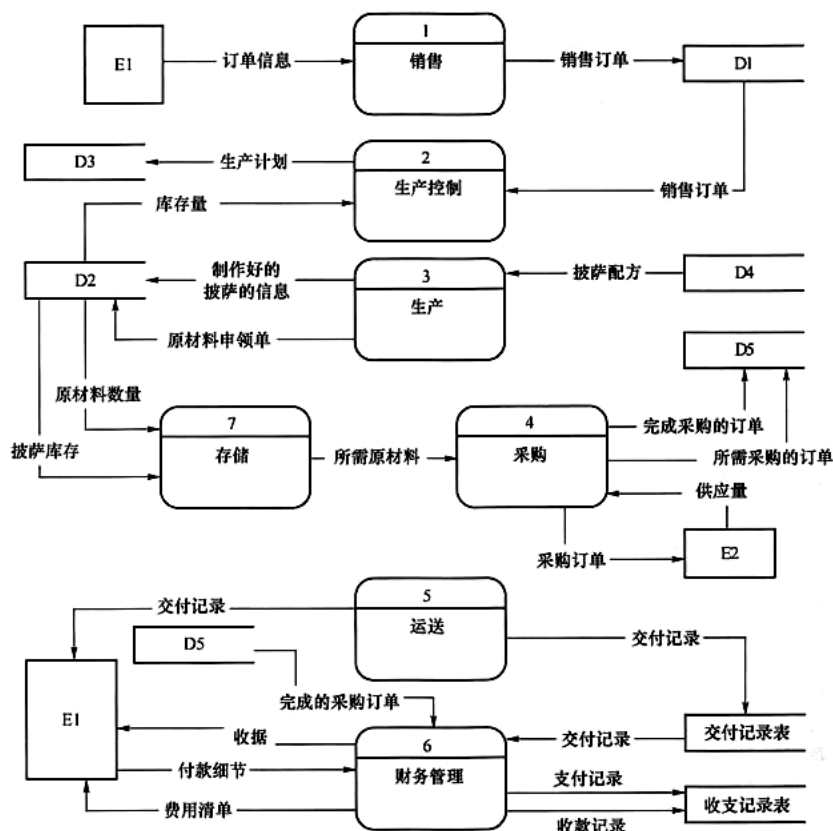


图 2 0 层数据流图

根据 0 层数据流中财务管理为客户开具费用清单数据流可知，E1 实体为客户；从向供应商发送采购订单、得到供应商的供应量可知，E2 实体为供应商。

【问题 2】（5 分）

根据说明中的词语，给出图 2 中的数据存储 D1~D5 的名称。

答案：D1：销售订单表；D2：库存表；D3：生产计划表；D4：配方表；D5：采购订单表

【解析】

根据题中对更功能的描述，以及 0 层数据流：生产控制，制定生产计划，并将其保存在生产计划表中可知，D3 为生产计划表；生产，将制作好的披萨的信息存入库存表中，可知 D2 为库存表；采购，在采购订单表中标记已完成的订单可知 D5 为采购订单表；生产，由“根据生产计划和配方表中的披萨配方，向库存发出原料申请”可知 D4 为配方表；根据“处理客户的订单信息，生成销售订单。并将其记录在销售订单表中”可知 D1 为销售订单表。

【问题 3】（5 分）

根据说明中的词语，补充图 2 中缺失的数据流及其起点和终点。

答案：数据流名称：支付细节；起点：财务管理；终点：E2

数据流名称：销售订单；起点：销售订单表；终点：5 运送

数据流名称：生产计划；起点：D3；终点：3 生产

数据流名称：库存量；起点：D2；终点：4 采购

数据流名称：原材料数量；起点：4 采购；终点：D2

数据流名称：未完成订单；起点：销售订单表；终点：7 存储。

【解析】

本题考查补充分层 DFD 中的数据流。在分层 DFD 中，需要保持父图和子图的平衡，即父图中某加

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和名字上相同，或者父图的一个输入(或输出)数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是附图中的一个数据流。

由“财务管理”段中“依据完成的采购订单给供应商支付原材料费用并出具支付细节”的描述，存在一个起点为财务管理，重点为供应商即 E2 的数据流，即支付细节数据流；由“运送”段中“根据销售订单将披萨交付给客户，并记录在交付记录表中”可知存在一个由“销售订单”指向“运送”的数据流，即销售订单数据流；由“采购”段中“根据所需原材料及库存量，确定采购数量”可知存在由“库存表”指向“采购”的数据流，数据流为库存量；根据“存储”段中“检查库存的原材料、披萨和未完成订单，确定所需原料”可知，存在一个由“订单表”指向“存储”的数据流，数据流为未完成的订单。

试题二（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某集团公司在全国不同城市拥有多个大型超市，为了有效管理各个超市的业务工作，需要构建一个超市信息管理系统。

[需求分析结果]

超市信息包括：超市名称、地址、经理和电话，其中超市名称唯一确定超市关系的每一个元组。每个超市只有一名经理。

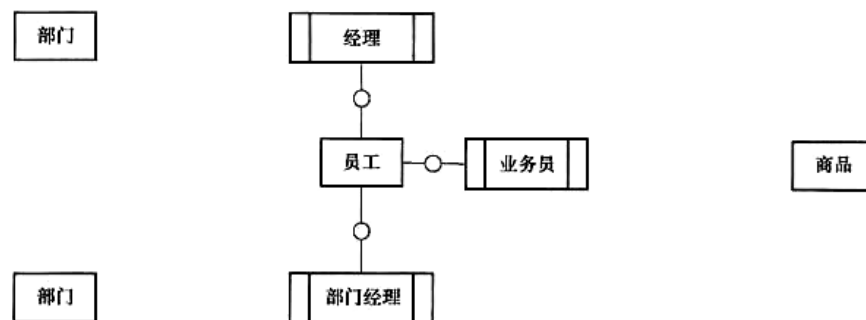
超市设有计划部、财务部、销售部等多个部门，每个部门只有一名部门经理，有多名员工，每个员工只属于一个部门。部门信息包括：超市名称、部门名称、部门经理和联系电话。超市名称、部门名称唯一确定部门关系的每一个元组。

员工信息包括：员工号、姓名、超市名称、部门名称、职位、联系方式和工资。其中，职位信息包括：经理、部门经理、业务员等。员工号唯一确定员工关系的每一个元组。

商品信息包括：商品号、商品名称、型号、单价和数量。商品号唯一确定商品关系的每一个元组。一名业务员可以负责超市内多种商品的配给，一种商品可以由多名业务员配给。

[概念模型设计]

根据需求分析阶段收集的信息，设计的实体联系图和关系模式（不完整）如下图：



实体联系图

[关系模式设计]

超市(超市名称、经理、地址、电话)

部门((a)，部门经理、联系电话)

员工((b)，姓名、联系方式、职位、工资)

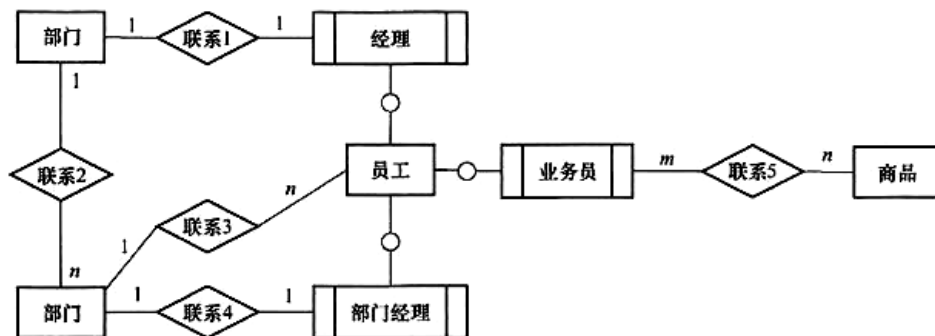
商品(商品号、商品名称、型号、单价、数量)

配给((c)，配给时间、配给数量、业务员)

【问题 1】（5 分）

根据问题描述，补充四个联系，完善图上图的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型分为 1:1、1:n 和 m:n(或 1:1、1:*和*:*).

答案：



【解析】

本题考查数据库系统中实体联系模型和关系模式设计方面的应用知识，属于比较传统的题目，考察点也和往年类似。

本题考查数据库的概念结构设计。两个实体集间的联系分为三类：一对一(1:1)，一对多(1:n)和多对多(m:n)。

根据题意，“每个部门只有一名部门经理”部门和部门经理之间是 1:1 关系，每个部门有多名员工，即部门和员工之间是 1:n 的关系，“超市只有一名经理”可知表示超市的部门和经理之间是 1:1 的关系，“超市有多个部门”可知超市和部门之间是 1:n 的关系。由“一名业务员可以负责超市内多种商品的配给，一种商品可以由多名业务员配给”可知业务员和商品之间是多对多的关系，即 m:n。

根据以上分析，即可完成题中的实体联系图。

【问题 2】（5 分）

(1) 根据实体联系图，将关系模式中的空(a)~(c)补充完整；

(2) 给出部门和配给关系模式的主键和外键。

答案：

(1) (a)超市名称、部门名称主键：(超市名称、部门名称)外键：超市名称、部门经理

(b)员工号、超市名称、部门名称

(c)商品号

(2) 主键：(商品号、业务员、配给时间)外键：业务员、商品号

【解析】

本题考查数据库的逻辑结构设计，题目要求补充完整各关系模式，并给出部门和配给关系模式的主键和外键。

根据问题一画完整的实体联系图和需求描述，员工关系模式包括员工号、姓名、部门名称、超市名称、职位、联系方式、工资等，因此(b)处应填员工号、超市名称和部门名称；部门关系模式包括超市名称、部门名称、部门经理和联系电话等，由此可知(a)处填超市名称和部门名称。配给关系模式包括商品号、业务员、配给时间、配给数量、业务员等，由此可知，(c)处应填商品号、业务员。部门关系模式的主键应为(超市名称和部门名称)，外键名称为(超市名称、部门经理)；配给关系模式的主键应为(商品号、业务员、配给时间)，外键应为(业务员、商品号)。

【问题 3】（5 分）

(1) 超市关系的地址可以进一步分为邮编、省、市、街道，那么该属性是属于简单属性还是复合属性?请用 100 字以内文字说明。

(2) 假设超市需要增设一个经理的职位，那么超市与经理之间的联系类型应修改为(d)，超市关系

应修改为(e)。

答案：

(1) 超市关系中的地址属于复合属性。所谓复合属性就是指属性中含有多种信息，可以进一步拆分的属性，地址可以拆分成多个简单属性，符合这一特征。

(2) d 是 1:n e 超市名称、地址、电话

【解析】

超市关系的地址可以进一步分为邮编、省、市、街道，则超市关系中的地址属于复合属性。所谓复合属性就是指属性中含有多种信息，可以进一步拆分的属性，地址可以拆分成多个简单属性，符合这一特征。

假设超市需要增设一个经理的职位，那么超市与经理之间的联系类型应该是一个超市有多个经理，为 1:n 的关系，超市关系应包含超市名称、地址、电话。

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某公司欲开发一个管理选民信息的软件系统。系统的基本需求描述如下：

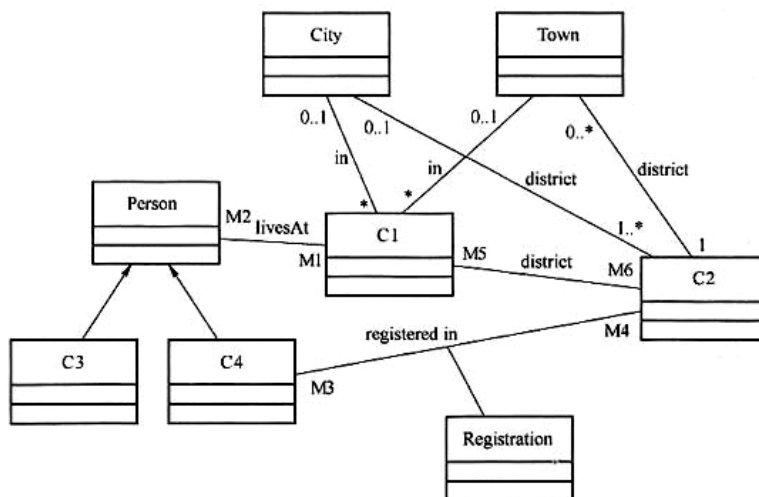
每个人(Person)可以是一个合法选民(Eligible)或者无效的选民(Ineligible)。

每个合法选民必须通过该系统对其投票所在区域(即选区，Riding)进行注册(Registration)。每个合法选民仅能注册一个选区。

选民所属选区由其居住地址(Address)决定。假设每个人只有一个地址，地址可以是镇(Town)或者城市(City)。

某些选区可能包含多个镇；而某些较大的城市也可能包含多个选区。

现采用面向对象方法对该系统进行分析与设计，得到如下图所示的初始类图。



类图

【问题 1】（5 分）

根据说明中的描述，给出上图中 C1~C4 所对应的类名（类名使用说明中给出的英文词汇）。

答案：

C1:Address C2:Riding C3:Ineligible C4:Eligible

【解析】

由“每个人可以是一个合法选民或者无效选民”可知 C3 和 C4 是这两者中的一个，由 C4 和 C2 关

联可知, C4 位合法选民 (Eligible), 则 C3 为无效选民 (Ineligible)。由 city 和 Town 共同指向 C1 以及描述“选民所属选区由其居住地址 (Address) 决定。假设每个人只有一个地址, 地址可以是镇 (Town) 或者城市 ((City))”可知, C1 应该为 Address。由描述“每个合法选民必须通过该系统对其投票所在区域 (即选区, Riding) 进行注册 (Registration)”以及 C4 指向 C2 和 cit-y、town 同时指向 C2 可知, C2 为 Riding, 即选区。

【问题 2】(5 分)

根据说明中的描述, 给出上图中 M1~M6 处的多重度。

答案: M1:1,M2:*,M3:*,M4:1,M5:*,M6:1

【解析】

UML 中关联的多重度是指一个类的实例能够与另一个类的多少个实例相关联。具体的取值意义有:

0..1:0 个或 1 个

1:只能 1 个

0..*:0 个或多个

*:0 个或多个

1..*:1 个或多个

由描述“每个人只有一个地址”可知, M1 为 1; 一个地址可以有 0 个或多个人, 因此 M2 为*; 一个选区可以有 0 个或多个选民, 一个选民只在一个选区投票, 因此, M3 为*, M4 为 1; 由“某些选区可能包含多个镇”可知每个选区可包含 0 个或多个地址, M5 应为*, 每个选民只有一个地址, 每个地址属于一个选区, 因此 M6 为 1。

【问题 3】(5 分)

现对该系统提出了以下新需求:

(1)某些人拥有在多个选区投票的权利, 因此需要注册多个选区;

(2)对手满足(1)的选民, 需要划定其“主要居住地”, 以确定他们应该在哪个选区进行投票。

为了满足上述需求, 需要对上图所示的类图进行哪些修改?请用 100 字以内文字说明。

答案: 将 M1 与 M4 由 1, 修改为 1..*

【解析】

若对系统提出新的要求:

(1) 某些人拥有在多个选区投票的权利, 因此需要注册多个选区;

(2) 对手满足(1)的选民, 需要划定其“主要居住地”, 以确定他们应该在哪个选区进行投票。则将 M1 与 M4 由 1, 修改为 1..*。

试题四 (15 分)

阅读下列说明和 C 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

计算一个整数数组 a 的最长递增子序列长度的方法描述如下:

假设数组 a 的长度为 n, 用数组 b 的元素 $b[i]$ 记录以 $a[i](0 \leq i < n)$ 为结尾元素的最长递增子序列的长度为 $\max_{0 \leq i < n} \{b[i]\}$; 其中 $b[i]$ 满足最优子结构, 可递归定义为:

$$\begin{cases} b[0] = 1 \\ b[i] = \max_{\substack{0 \leq k \leq i \\ a[k] \leq a[i]}} \{b[k]\} + 1 \end{cases}$$

【C 代码】

下面是算法的 C 语言实现。

(1) 常量和变量说明

a: 长度为 n 的整数数组, 待求其最长递增子序列

b: 长度为 n 的数组, $b[i]$ 记录以 $a[i](0 \leq i < n)$ 为结尾元素的最长递增子序列的长度, 其中 $0 \leq i < n$

len: 最长递增子序列的长度

i,j: 循环变量

temp: 临时变量

(2) C 程序

```
#include <stdio.h>
int maxL(int*b, int n) {
    int i, temp=0;
    for(i=0; i<n; i++) {

        if(b[i]>temp)
            temp=b[i];
    }
    return temp;
}
int main() {
    int n, a[100], b[100], i, j, len;
    scanf("%d", &n);
    for(i=0; i<n; i++) {
        scanf("%d", &a[i]);
    }
    (1) ;
    for(i=1; i<n; i++) {
        for(j=0, len=0; (2) ; j++) {
            if( (3) && len<b[j])
                len=b[j];
        }
        (4) ;
    }
    Printf("len:%d\n", maxL(b,n));
    printf("\n");
}
```

【问题 1】(8 分)

根据说明和 C 代码, 填充 C 代码中的空 (1) ~ (4)。

答案: (1) b[0]=1 (2) j<i (3) a[j]<=a[i] (4) b[i]=len+1

【问题 2】(4 分)

根据说明和 C 代码, 算法采用了 (5) 设计策略, 时间复杂度为 (6) (用 O 符号表示)。

答案: (5) 动态规划法 (6) O(n²)

【问题 3】(5 分)

已知数组 a={3, 10, 5, 15, 6, 8}, 据说明和 C 代码, 给出数组 b 的元素值。

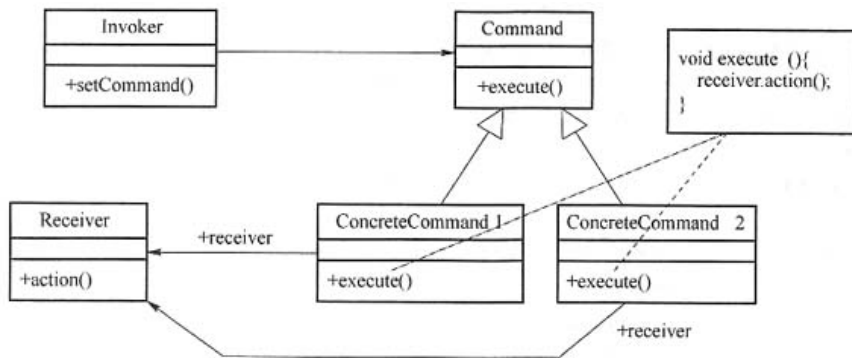
答案: B={1, 2, 2, 3, 3, 4}

试题五 (15 分)

阅读下列说明和 C++ 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某灯具厂商欲生产一个灯具遥控器, 该遥控器具有 7 个可编程的插槽, 每个插槽都有开关按钮, 对应着一个不同的灯。利用该遥控器能够统一控制房间中该厂商所有品牌灯具的开关, 现采用 Command(命令)模式实现该遥控器的软件部分。Command 模式的类图如下图所示。



Command 模式类图

【C++代码】

```

class Light {
public:
    Light(string name) { /* 代码省略 */ }
    void on() { /* 代码省略 */ } // 开灯
    void off() { /* 代码省略 */ } // 关灯
};

class Command {
public:
    (1) ;
};

class LightOnCommand:public Command { // 开灯命令
private:
    Light* light;
public:
    LightOnCommand(Light* light) { this->light=light; }
    void execute() { (2) ; }
};

class LightOffCommand:public Command { // 关灯命令
private:
    Light *light;
public:
    LightOffCommand(Light* light) { this->light=light; }
    void execute() { (3) ; }
};

class RemoteControl{ // 遥控器
private:
    Command* onCommands[7];
    Command* offCommands[7];
public:
    RemoteControl() { /* 代码省略 */ }
    void setCommand(int slot, Command* onCommand, Command* offCommand) {
        (4) =onCommand;
        (5) =offCommand;
    }
    void onButtonWasPushed(int slot) { (6) ; }
    void offButtonWasPushed(int slot) { (7) ; }
};

```

```

};

int main() {
    RemoteControl* remoteControl=new RemoteControl();
    Light* livingRoomLight=new Light("Living Room");
    Light*kitchenLight=new Light("kitchen");
    LightOnCommand*livingRoomLightOn=newLightOnCommand(livingRoomLight);
    LightOffCommand* livingRoomLightOff=newLightOffCommand(livingRoomLight);
    LightOnCommand*kitchenLightOn=new LightOnCommand(kitchenLight);
    LightOffCommand* kitchenLightOff=new LightOffCommand(kitchenLight);
    remoteControl->setCommand(0, livingRoomLightOn, livingRoomLightOff);
    remoteControl->setCommand(1, kitchenLightOn, kitchenLightOff);
    remoteControl->onButtonWasPushed(0);
    remoteControl->offButtonWasPushed(0);
    remoteControl->onButtonWasPushed(1);
    remoteControl->offButtonWasPushed(1);
    /* 其余代码省略 */
    return 0;
}

```

答案:

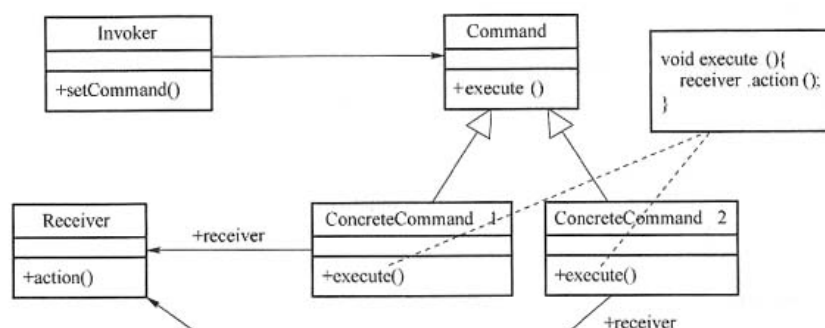
- (1) virtual void execute()=0
- (2) light->on()
- (3) light->off()
- (4) onCommands[slot]
- (5) offCommands[slot]
- (6) onCommands[slot]->execute()
- (7) offCommands[slot]->execute()

试题六（15 分）

阅读下列说明和 Java 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某灯具厂商欲生产一个灯具遥控器，该遥控器具有 7 个可编程的插槽，每个插槽都有开关按钮，对应着一个不同的灯。利用该遥控器能够统一控制房间中该厂商所有品牌灯具的开关，现采用 Command(命令)模式实现该遥控器的软件部分。Command 模式的类图如下图所示。



Command 模式的类图

【Java 代码】

```
class Light {
    public Light() {}
    public Light(String name) { /* 代码省略 */ }
    public void on() { /* 代码省略 */ } // 开灯
    public void off() { /* 代码省略 */ } // 关灯
    // 其余代码省略
}
____(1)____ {
    public void execute();
}
class LightOnCommand implements Command { // 开灯命令
    Light light;
    public LightOnCommand(Light light) { this.light=light; }
    public void execute() { ____ (2) ____; }
}
class LightOffCommand implements Command { // 关灯命令
    Light light;
    public LightOffCommand(Light light) { this.light=light; }
    public void execute() { ____ (3) ____; }
}
class RemoteControl { // 遥控器
    Command[] onCommands=new Command[7];
    Command[] offCommands=new Command[7];
    public RemoteControl() { /* 代码省略 */ }
    public void setCommand(int slot, Command onCommand, Command offCommand) {
        ____ (4) ____=onCommand;
        ____ (5) ____=offCommand;
    }
    public void onButtonWasPushed(int slot) {
        ____ (6) ____;
    }
    public void offButtonWasPushed(int slot){
        ____ (7) ____ ;
    }
}
class RemoteLoader {
    public static void main(String[] args) {
        RemoteControl remoteControl=new RemoteControl();
        Light livingRoomLight=new Light("Living Room");
        Light kitchenLight=new Light("kitchen");
        LightOnCommand livingRoomLightOn=new LightOnCommand(livingRoomLight);
        LightOffCommand livingRoomLightOff=new LightOffCommand(livingRoomLight);
        LightOnCommand kitchenLightOn=new LightOnCommand(kitchenLight);
        LightOffCommand kitchenLightOff=new LightOffCommand(kitchenLight);
        remoteControl.setCommand(0, livingRoomLightOn, livingRoomLightOff);
        remoteControl.setCommand(1, kitchenLightOn, kitchenLightOff);
        remoteControl.onButtonWasPushed(0);
        remoteControl.offButtonWasPushed(0);
    }
}
```



```
        remoteControl.onButtonWasPushed(1);  
        remoteControl.offButtonWasPushed(1);  
    }  
}
```

答案:

- (1) interface Command
- (2) light.on()
- (3) light.off()
- (4) onCommands[slot]
- (5) offCommands[slot]
- (6) onCommands[slot].execute()
- (7) offCommands[slot].execute()