

Ni 在 CPU 中，(1)不仅要保证指令的正确执行，还要能够处理异常事件。

- (1) A. 运算器                      B. 控制器                      C. 寄存器组                      D. 内部总线

【答案】B

【解析】本题考查计算机系统硬件方面的基础知识。

计算机中的 CPU 是硬件系统的核心，用于数据的加工处理，能完成各种算术、逻辑运算及控制功能。其中，控制器的作用是控制整个计算机的各个部件有条不紊地工作，它的基本功能就是从内存取指令和执行指令。

循环冗余校验码 (CRC) 利用生成多项式进行编码。设数据位为  $k$  位，校验位为  $r$  位，则 CRC 码的格式为 (2)。

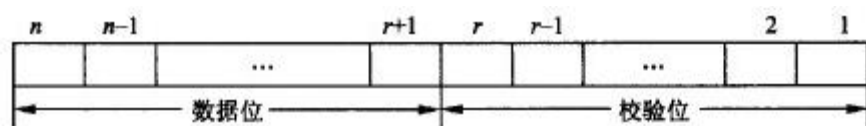
- (2) A.  $k$  个数据位之后跟  $r$  个校验位                      B.  $r$  个校验位之后跟  $k$  个数据位  
C.  $r$  个校验位随机加入  $k$  个数据位中                      D.  $r$  个校验位等间隔地加入  $k$  个数据位中

【答案】A

【解析】本题考查数据校验基础知识。

计算机系统运行时，各个部件之间要进行数据交换，为了确保数据在传送过程中正确无误，一是提高硬件电路的可靠性；二是提高代码的校验能力，包括查错和纠错。常用的三种校验码：奇偶校验码 (Parity Codes)、海明码 (Hamming Code) 和循环冗余校验 (Cyclic Redundancy Check, CRC) 码。

循环冗余校验码广泛应用于数据通信领域和磁介质存储系统中。它利用生成多项式为  $k$  个数据位产生  $r$  个校验位来进行编码，其编码长度为  $k+r$ 。CRC 的代码格式为：



以下关于数的定点表示和浮点表示的叙述中，不正确的是 (3)。

- (3) A. 定点表示法表示的数 (称为定点数) 常分为定点整数和定点小数两种  
B. 定点表示法中，小数点需要占用一个存储位  
C. 浮点表示法用阶码和尾数来表示数，称为浮点数  
D. 在总位数相同的情况下，浮点表示法可以表示更大的数

【答案】B

**【解析】** 本题考查数据表示基础知识。

各种数据在计算机中表示的形式称为机器数，其特点是采用二进制计数制，数的符号用 0、1 表示，小数点则隐含表示而不占位置。机器数对应的实际数值称为数的真值。

为了便于运算，带符号的机器数可采用原码、反码、补码和移码等不同的编码方法。

所谓定点数，就是表示数据时小数点的位置固定不变。小数点的位置通常有两种约定方式：定点整数（纯整数，小数点在最低有效数值位之后）和定点小数（纯小数，小数点在最高有效数值位之前）。

当机器字长为  $n$  时，定点数的补码和移码可表示  $2^n$  个数，而其原码和反码只能表示  $2^{n-1}$  个数（0 表示占用了两个编码），因此，定点数所能表示的数值范围比较小，运算中很容易因结果超出范围而溢出。

数的浮点表示的一般形式为： $N=2^E \times F$ ，其中  $E$  称为阶码， $F$  为尾数。阶码通常为带符号的纯整数，尾数为带符号的纯小数。浮点数的表示格式如下：

阶符	阶码	数符	尾数
----	----	----	----

很明显，一个数的浮点表示不是唯一的。当小数点的位置改变时，阶码也相应改变，因此可以用多种浮点形式表示同一个数。

浮点数所能表示的数值范围主要由阶码决定，所表示数值的精度则由尾数决定。

(4) 不属于按寻址方式划分的一类存储器。

- (4) A. 随机存储器    B. 顺序存储器    C. 相联存储器    D. 直接存储器

**【答案】** C

**【解析】** 本题考查存储系统的基础知识。

存储系统中的存储器，按访问方式可分为按地址访问的存储器和按内容访问的存储器；按寻址方式分类可分为随机存储器、顺序存储器和直接存储器。随机存储器（Random Access Memory, RAM）指可对任何存储单元存入或读取数据，访问任何一个存储单元所需的时间是相同的。顺序存储器（Sequentially Addressed Memory, SAM）指访问数据所需要的时间与数据所在的存储位置相关，磁带是典型的顺序存储器。直接存储器（Direct Addressed Memory, DAM）是介于随机存取和顺序存取之间的一种寻址方式。磁盘是一种直接存取存储器，它对磁道的寻址是随机的，而在一个磁道内，则是顺序寻址。相联存储器是一种按内容访问的存储器。其工作原理就是把数据或数据的某一部分作为关键字，将该关键字与存储器中的每一单元进行比较，从而找出存储器中所有与关键字相同的数据字。

在 I/O 设备与主机间进行数据传输时，CPU 只需在开始和结束时作少量处理，而无需干

预数据传送过程的是 (5) 方式。

- (5) A. 中断                      B. 程序查询                      C. 无条件传送                      D. 直接存储器存取

**【答案】D**

**【解析】** 本题考查计算机系统硬件方面的基础知识。

中断方式下的数据传送是当 I/O 接口准备好接收数据或准备好向 CPU 传送数据时,就发出中断信号通知 CPU。对中断信号进行确认后,CPU 保存正在执行的程序的现场, 转而执行提前设置好的 I/O 中断服务程序, 完成一次数据传送的处理。这样,CPU 就不需要主动查询外设的状态,在等待数据期间可以执行其他程序,从而提高了 CPU 的利用率。采用中断方式管理 I/O 设备,CPU 和外设可以并行地工作。

程序查询方式下,CPU 通过执行程序查询外设的状态,判断外设是否准备好接收数据或准备好了向 CPU 输入的数据。

直接内存存取 (Direct Memory Access, DMA) 方式的基本思想是通过硬件控制实现主存与 I/O 设备间的直接数据传送,数据的传送过程由 DMA 控制器 (DMAC) 进行控制,不需要 CPU 的干预。在 DMA 方式下,由 CPU 启动传送过程,即向设备发出“传送一块数据”的命令,在传送过程结束时,DMAC 通过中断方式通知 CPU 进行一些后续处理工作。

(6) 不属于系统总线。

- (6) A. ISA                      B. EISA                      C. SCSI                      D. PCI

**【答案】C**

**【解析】** 本题考查计算机系统的基础知识。

系统总线又称内总线或板级总线,在微机系统中用来连接各功能部件而构成一个完整的微机系统。系统总线包含有三种不同功能的总线,即数据总线 DB (Data Bus)、地址总线 AB (Address Bus) 和控制总线 CB (Control Bus)。

ISA (Industrial Standard Architecture) 总线标准是 IBM 公司 1984 年为推出 PC/AT 机而建立的系统总线标准,所以也叫 AT 总线。它是对 XT 总线的扩展,以适应 8/16 位数据总线要求。

EISA 总线是 1988 年由 Compaq 等 9 家公司联合推出的总线标准。它在 ISA 总线的基础上使用双层插座,在原来 ISA 总线的 98 条信号线上又增加了 98 条信号线,也就是在两条 ISA 信号线之间添加一条 EISA 信号线。在实用中,EISA 总线完全兼容 ISA 总线信号。

PCI (Peripheral Component Interconnect) 总线是当前最流行的总线之一,它是由 Intel

公司推出的一种局部总线。它定义了 32 位数据总线，且可扩展为 64 位。PCI 总线主板插槽的体积比原 ISA 总线插槽还小，支持突发读写操作，最大传输速率可达 132MB/S，可同时支持多组外围设备。PCI 局部总线不能兼容现有的 ISA、EISA、MCA (Micro Channel Architecture) 总线，但它不受制于处理器，是基于奔腾等新一代微处理器而发展的总线。

SCSI (Small Computer System Interface) 是一种用于计算机和智能设备之间（硬盘、软驱、光驱、打印机、扫描仪等）系统级接口的独立处理器标准。

下列安全协议中，与 TLS 最接近的协议是 (7)。

- (7) A. PGP                      B. SSL                      C. HTTPS                      D. IPSec

**【答案】B**

**【解析】** 本题考查安全协议方面的基础知识。

SSL (Secure Socket Layer, 安全套接层) 是 Netscape 于 1994 年开发的传输层安全协议，用于实现 Web 安全通信。1996 年发布的 SSL 3.0 协议草案已经成为一个事实上的 Web 安全标准。

TLS (Transport Layer Security, 传输层安全协议) 是 IETF 制定的协议，它建立在 SSL 3.0 协议规范之上，是 SSL 3.0 的后续版本。

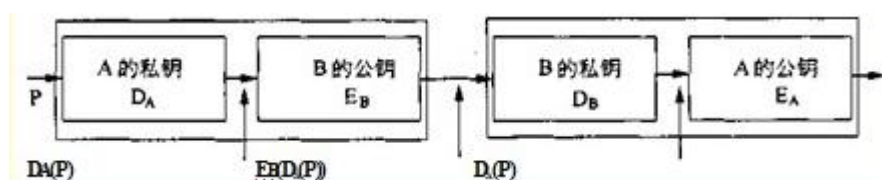
用户 B 收到用户 A 带数字签名的消息 M，为了验证 M 的真实性，首先需要从 CA 获取用户 A 的数字证书，并利用 (8) 验证该证书的真伪，然后利用 (9) 验证 M 的真实性。

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

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

**【答案】A    C**

**【解析】** 本题考查数字签名和数字证书方面的知识。



基于公钥的数字签名系统如下图所示：A 为了向 B 发送消息 P，A 用自己的私钥对 P 签名后再用 B 的公钥对签名后的数据加密，B 收到消息后先用 B 的私钥解密后在用 A 的公钥认证 A 的签名以及消息的真伪。

(9) 用户 B 收到用户 A 带数字签名的消息 M，为了验证 M 的真实性，首先需要从 CA 获取用户 A 的数字证书，验证证书的真伪需要用 CA 的公钥验证 CA 的签名，验证 M 的真实性需要用用户 A 的公钥验证用户 A 的签名。

M 软件公司的软件产品注册商标为 M，为确保公司在市场竞争中占据优势，对员工进行了保密约束。此情形下该公司不享有 (10)。

- (10) A. 商业秘密权      B. 著作权      C. 专利权      D. 商标权

**【答案】C**

**【解析】**本题考查知识产权基础知识。

关于著作权的取得，《计算机软件保护条例》规定：“软件著作权自软件开发完成之日起产生。”即软件著作权自软件开发完成之日起自动产生，不论整体还是局部，只要具备了软件的属性即产生软件著作权，既不要求履行任何形式的登记或注册手续，也无须在复制件上加注著作权标记，也不论其是否已经发表都依法享有软件著作权。软件开发经常是一项系统工程，一个软件可能会有很多模块，而每一个模块能够独立完成某一项功能。自该模块开发完成后就产生了著作权。软件公司享有商业秘密权。因为一项商业秘密受到法律保护的依据，必须具备构成商业秘密的三个条件，即不为公众所知悉、具有实用性、采取了保密措施。商业秘密权保护软件是以软件中是否包含着“商业秘密”为必要条件的。该软件公司组织开发的应用软件具有商业秘密的特征，即包含着他人不能知道到的技术秘密；具有实用性，能为软件公司带来经济效益；对职工进行了保密的约束，在客观上已经采取相应的保密措施。所以软件公司享有商业秘密权。商标权、专利权不能自动取得，申请人必须履行商标法、专利法规定的申请手续，向国家行政部门提交必要的申请文件，申请获准后即可取得相应权利。获准注册的商标通常称为注册商标。

X 软件公司的软件工程师张某兼职于 Y 科技公司，为完成 Y 科技公司交给的工作，做出了一项涉及计算机程序的发明。张某认为该发明是利用自己的业余时间完成的，可以以个人名义申请专利。此项专利申请权应归属 (11)。

- (11) A. 张某      B. X 软件公司      C. Y 科技公司      D. 张某和 Y 科技公司

**【答案】C**

**【解析】**本题考查考生知识产权方面的基础知识。

专利法意义上的发明人必须是：第一，直接参加发明创造活动。在发明创造过程中，只

负责组织管理工作或者是对物质条件的利用提供方便的人，不应当被认为是发明人；第二，必须是对发明创造的实质性特点作出创造性贡献的人。仅仅提出发明所要解决的问题而未对如何解决该问题提出具体意见的，或者仅仅从事辅助工作的人，不视为发明人或者设计人。有了发明创造不一定就能成为专利权人。发明人或设计人是否能够就其技术成果申请专利，还取决于该发明创造与其职务工作的关系。一项发明创造若被认定为职务发明创造，那么该项发明创造申请并获得专利的权利为该发明人或者设计人所属单位所有。根据专利法规定，职务发明创造分为两种情形：一是执行本单位的任务所完成的发明创造，二是主要是利用本单位的物质技术条件所完成的发明创造。《专利法实施细则》对“执行本单位的任务所完成的发明创造”和“本单位的物质技术条件”又分别作出了解释。所谓执行本单位的任务所完成的发明创造是指：①在本职工作中作出的发明创造；②履行本单位交付的本职工作之外的任务所作出的发明创造；③辞职、退休或者调动工作后一年内所作出的，与其在原单位承担的本职工作或原单位分配的任务有关的发明创造。职务发明创造的专利申请权属于发明人所在的单位，但发明人或者设计人仍依法享有发明人身份权和获得奖励报酬的权利。

以下图像文件格式中，(12)可以在单个文件中实现多个图片轮流显示或者构成简单的动画效果。

(12) A. BMP

B. JPEG

C. GIF

D. TIFF

**【答案】C**

**【解析】**本题考查多媒体方面的基础知识。

BMP 图像文件格式是 Windows 操作系统采用的图像文件格式，BMP 采用位映射存储格式，除了图像深度可选以外，一般不采用其他任何压缩，所以占用的存储空间较大。

JPEG 图像文件格式是一种有损压缩的静态图像文件存储格式，压缩比例可以选择，支持灰度图像、RGB 真彩色图像和 CMYK 真彩色图像。JPG 格式文件的压缩比例很高，非常适用于处理大量图像的场所。

TIFF 是电子出版 CD-ROM 中的一个图像文件格式。TIFF 格式非常灵活易变，它又定义了 4 类不同的格式：TIFF-B 适用于二值图像；TIFF-G 适用于黑白灰度图像；TIFF-P 适用于带调色板的彩色图像；TIFF-R 适用于 RGB 真彩图像。无论在视觉上还是其他方面，都能把任何图像编码成二进制形式而不丢失任何属性。

GIF 图像文件格式以数据块为单位来存储图像的相关信息，采用了 LZW 无损压缩算法按扫描行压缩图像数据。它可以在一个文件中存放多幅彩色图像，每一幅图像都由一个图像描

述符、可选的局部彩色表和图像数据组成。如果把存储于一个文件中的多幅图像逐幅读出来显示到屏幕上，可以像播放幻灯片那样显示或者构成简单的动画效果。GIF 定义了两数据存储方式，一种是按行连续存储，存储顺序与显示器的显示顺序相同；另一种是按交叉方式存储，由于显示图像需要较长的时间，使用这种方法存放图像数据，用户可以在图像数据全部收到之前这幅图像的全貌，而不觉得等待时间太长。

采用图像文件格式支持实现的显示效果与通过软件的实现方式相比，循环显示图像的数量、每幅图像显示的时间、透明度变化等参数受图像文件格式的限制，而通过软件方式实现显示效果的各项参数则可以灵活配置。

计算机终端用户因等待时间过长而放弃下载或显示图片/影像的问题，不能通过采用(13)来解决。

- |               |                |
|---------------|----------------|
| (13)A. 流媒体技术  | B. JPEG 累进编码方式 |
| C. GIF 图像文件格式 | D. AD/DA 变换方式  |

**【答案】D**

**【解析】**本题考查多媒体基础知识。

流媒体是指在网络中使用流式传输技术的连续时基媒体，而流媒体技术是指把连续的影像和声音信息经过压缩处理之后放到专用的流服务器上，让浏览者一边下载一边观看、收听，而不需要等到整个多媒体文件下载完成就可以即时观看和收听的技术。流媒体融合了多种网络以及音视频技术，在网络中要实现流媒体技术，必须完成流媒体的制作、发布、传播、播放等环节。

JPEG 累进（或增量、渐进、递增、progressive）编码模式，这样可以实现图像内容的方式传输，在浏览器上的直观效果是无需过久等待即可看到模糊的图像，然后图像显示内容由模糊逐渐变得清晰。

GIF 图像文件格式以数据块为单位来存储图像的相关信息，采用了 LZW 无损压缩算法按扫描行压缩图像数据。它可以在一个文件中存放多幅彩色图像，每一幅图像都由一个图像描述符、可选的局部彩色表和图像数据组成。如果把存储于一个文件中的多幅图像逐幅读出来显示到屏幕上，可以像播放幻灯片那样显示或者构成简单的动画效果。GIF 定义了两数据存储方式，一种是按行连续存储，存储顺序与显示器的显示顺序相同；另一种是按交叉方式存储，由于显示图像需要较长的时间，使用这种方法存放图像数据，用户可以在图像数据全部收到之前这幅图像的全貌，而不觉得等待时间太长。

图像信号是一种模拟信号，计算机要对它进行处理，必须将它转换成为数字图像信号，即用二进制数字的编码形式来表示图像，转换过程一般称为 A/D 转换（模数转换）。将图像播放出来，需进行 D/A 转换（数模转换）。

计算机处理模拟视频信号过程中首先要进行 (14)。

(14) A. A/D 变换      B. 数据压缩      C. D/A 变换      D. 数据存储

**【答案】A**

**【解析】** 本题考查多媒体基础知识。

视频信息是指活动的、连续的图像序列。一幅图像称为一帧，帧是构成视频信息的基本单元。全屏幕视频是指显示的视频图像充满整个屏幕，能以 30 帧/秒的速度刷新画面，使画面不会产生闪烁和不连贯的现象。电视机、激光视盘、摄像机等都可提供丰富多彩的模拟视频信号，常常需要把这些信号与计算机图形图像结合在一个共同的空间，通过处理达到最佳的效果，然后输出到计算机的显示器或其他电视设备上。模拟视频信号进入计算机，首先需要解决模拟视频信息的数字化问题。视频数字化的目的是将模拟信号经模数转换和彩色空间变换等过程，转换成计算机可以显示和处理的数字信号。由于电视和计算机的显示机制不同，因此要在计算机上显示视频图像需要作许多处理。例如，电视是隔行扫描，计算机的显示器通常是逐行扫描；电视是亮度（Y）和色度（C）的复合编码，而 PC 机的显示器工作在 RGB 空间；电视图像的分辨率和显示屏的分辨率也各不相同。这些问题在电视图像数字化过程中都需考虑。一般，对模拟视频信息进行数字化采取如下方式：

①先从复合彩色电视图像中分离出彩色分量，然后数字化。目前市场上的大多数电视信号都是复合的全电视信号，如录像带、激光视盘等存储设备上的电视信号。对这类信号的数字化，通常是将其分离成 YUV、YIQ 或 RGB 彩色空间的分量信号，然后用 3 个 A/D 转换器分别进行数字化。这种方式称为复合数字化。

②先对全彩色电视信号数字化，然后在数字域中进行分离，以获得 YUV、YIQ 或 RGB 分量信号。用这种方法对电视图像数字化时，只需一个高速 A/D 转换器。这种方式称为分量数字化。分量数字化方式是较多使用的一种方式。电视信号使用的彩色空间是 YUV 空间，即每幅彩色画面有亮度（Y）和色度（U、V）3 个分量，对这 3 个分量需分别进行取样和量化，得到一幅数字图像。由于人眼对色度信号的敏感程度远不如对亮度信号那么灵敏，所以色度信号的取样频率可以比亮度信号的取样频率低一些，以减少数字视频的数据量。



以下关于数据流图的叙述中，不正确的是\_(15)，

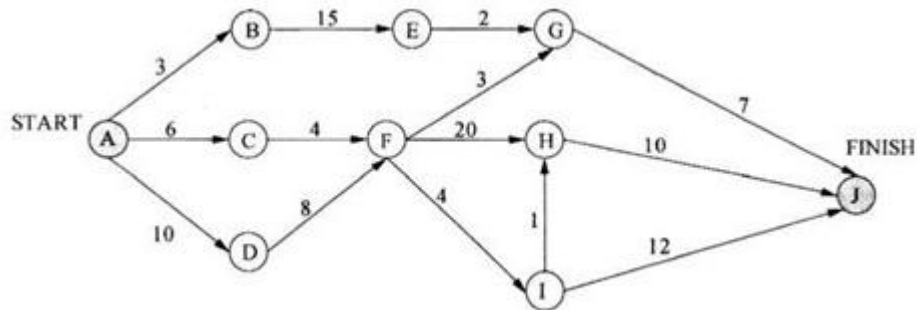
- (15)A. 每条数据流的起点或终点必须是加工  
B. 必须保持父图与子图平衡  
C. 每个加工必须有输入数据流，但可以没有输出数据流  
D. 应保持数据守恒

【答案】C

【解析】

数据流图是结构化分析方法的重要模型，用于描述系统的功能、输入、输出和数据存储等。在绘制数据流图中，每条数据流的起点或者终点必须是加工，即至少有一端是加工。在分层数据流图中，必须要保持父图与子图平衡。每个加工必须既有输入数据流又有输出数据流。必须要保持数据守恒。也就是说，一个加工所有输出数据流中的数据必须能从该加工的输入数据流中直接获得，或者是通过该加工能产生的数据。

某软件项目的活动图如下所示。图中顶点表示项目里程碑，连接顶点的边表示包含的活动，则里程碑\_(16)在关键路径上，活动FG的松弛时间为\_(17)。



- (16)A. B                      B. C                      C. D                      D. I  
(17)A. 19                      B. 20                      C. 21                      D. 24

【答案】C B

【解析】

(16)该活动图的关键路径为 ADFHL 关键路径长度为 48 天，因此里程碑 D 在关键路径上，B、C 和 I 步骤关键路径上。

(17) 活动 FG 的最早开始时间为第 18 天，最晚开始时间为第 38 天，因此松弛时间为 20 天。

在软件设计阶段，划分模块的原则是：一个模块的(18)。

- (18) A. 作用范围应该在其控制范围之内      B. 控制范围应该在其作用范围之内  
C. 作用范围与控制范围互不包含      D. 作用范围与控制范围不受任何限制

**【答案】A**

**【解析】**

模块的作用范围定义为受该模块内一个判定影响的模块集合，模块的控制范围为模块本身以及所有直接或间接从属于该模块的模块集合。其作用范围应该在控制范围之内。

定义风险参照水准是(19)活动常用的技术。

- (19) A. 风险识别      B. 风险预测      C. 风险评估      D. 风险控制

**【答案】C**

**【解析】**

定义风险参照水准是风险评估的一类技术，对于大多数软件项目来说成本、速度和性能是三种典型的风险参照水准。

可用于编写独立程序和快速脚本的语言是(20)。

- (20) A. Python      B. Prolog      C. Java      D. C#

**【答案】A**

**【解析】** 本题考查程序语言基础知识。

脚本语言又被称为扩建的语言，或者动态语言，是一种编程语言，通常以文本(如 ASCII)保存，只在被调用时进行解释或编译。Python 是一种脚本语言。

语言  $L = \{ambn \mid m \geq 0, n \geq 1\}$  的正规表达式是(21)

- (21) A.  $aa*bb*$       B.  $a*bb*$       C.  $aa*b*$       D.  $a*b*$

**【答案】B**

**【解析】** 本题考查程序语言基础知识。

$aa*bb*$  表示的字符串特点是：若干个 a 之后跟若干个 b，a 和 b 都至少出现 1 次。 $a*bb*$  表示的字符串特点是：若干个 a 之后跟若干个 b，a 可以不出现，b 至少出现 1 次。 $aa*b*$  表示的字符串特点是：若干个 a 之后跟若干个 b，a 至少出现 1 次，b 可以不出现。 $a*b*$  表示的字符串特点是：若干个 a 之后跟若干个 b，a 和 b 都可以不出现。语言  $L = \{ambn \mid m \geq 0, n \geq 1\}$

1} 中, 若干个 a 之后跟若干个 b, a 可以不出现, b 至少出现 1 次。

算术表达式  $(a-b)*c+d$  的后缀式是 (22) (-、+、\* 表示算术的减、加、乘运算, 运算符的优先级和结合性遵循惯例)。

(22) A.  $abcd - * +$                       B.  $ab - cd * +$                       C.  $ab - c * d +$                       D.  $abc - d * +$

**【答案】C**

**【解析】** 本题考查程序语言基础知识。

后缀式即逆波兰式, 是逻辑学家卢卡西维奇发明的一种表示表达式的方法。这种表示方式把运算符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。

$(a-b)*c+d$  的后缀式是  $ab-c*d+$ 。

某系统中仅有 5 个并发进程竞争某类资源, 且都需要 3 个该类资源, 那么至少有 (23) 个该类资源, 才能保证系统不会发生死锁。

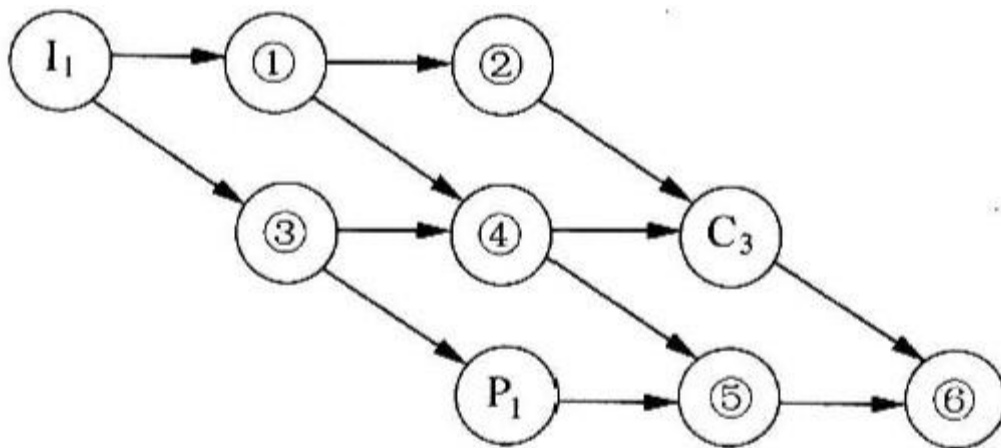
(23) A. 9                                      B. 10                                      C. 11                                      D. 15

**【答案】C**

**【解析】** 本题考查操作系统进程管理方面的基础知识。

假设系统为每个进程分配了 2 个资源, 对于选项 C, 系统还剩余 1 个资源, 能保证 5 个进程中的一个进程运行完毕。当该进程释放其占有的资源, 系统可用资源数为 3 个, 能保证未完成的 4 个进程中的 3 个进程运行完毕。当这 3 个进程释放其占有的资源, 系统可用资源数为 9 个, 显见能确保最后一个进程运行完。

某计算机系统有一个 CPU、一台输入设备和一台输出设备, 假设系统中有三个作业 T1、T2 和 T3, 系统采用优先级调度, 且 T1 的优先级 > T2 的优先级 > T3 的优先级。若每个作业具有三个程序段: 输入  $I_i$ 、计算  $C_i$  和输出  $P_i$  ( $i=1, 2, 3$ ), 执行顺序为  $I_i$ 、 $C_i$ 、 $P_i$ 。则这三个作业各程序段并发执行的前驱图如下所示。图中①、②分别为 (24), ③、④分别为 (25), ⑤、⑥分别为 (26)。

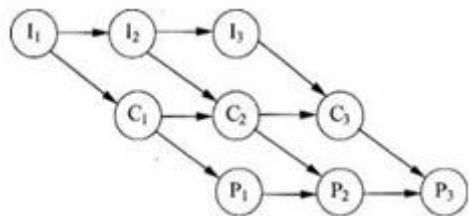


- (24) A. I2, c2      B. I2, I3      C. C1, P3      D. C2、P2  
 (25) A. C1、C2      B. I2、C1      C. I3、P3      D. C1、P2  
 (26) A. I3、C2      B. I2、C1      C. P2、P3      D. C1、P2

【答案】B A C

【解析】本题考查操作系统前驱图方面的基础知识。

前趋图是一个有向无循环图，由节点和有向边组成，节点代表各程序段的操作，而节点间的有向边表示两个程序段操作之间存在的前趋关系（“—>”）。程序段  $P_i$  和  $P_j$  的前趋关系可表示成  $P_i - P_j$ ，其中  $P_i$  是  $P_j$  的前趋， $P_j$  是  $P_i$  的后继，其含义是  $P_i$  执行结束后  $P_j$  才能执行。本题完整的前趋图如下图所示。



根据题意， $I_1$  执行结束后  $C_1$  才能执行， $C_1$  执行结束后  $P_1$  才能执行，因此  $I_1$  是  $C_1$ 、 $P_1$  的前趋， $C_1$  是  $P_1$  的前趋。可见，图中③应为  $C_1$ 。又因为计算机系统中只有一台输入设备，所以  $I_1$  执行结束后  $I_2$  和  $I_3$  才能执行，故  $I_1$  是  $I_2$  和  $I_3$  的前趋， $I_2$  是  $I_3$  的前趋。可见，图中①、②分别为  $I_2$ 、 $I_3$ 。

综上所述③应为  $C_1$ ，所以  $C_1$  是  $P_1$  的前趋，又因为计算机中只有一个 CPU，而且采用优先级调度，所以  $C_1$  是  $C_2$  的前趋，所以  $C_2$  是④。

由于计算机中只有一台输出设备，所以  $P_1$  执行结束后  $P_2$  才能执行，所以  $P_1$  是  $P_2$  的前趋， $P_2$  执行结束后  $P_3$  才能执行， $P_2$  是  $P_3$  的前趋，所以，⑤、⑥分别为  $P_2$  和  $P_3$

(25) 经上分析③应为  $C_1$ ，所以  $C_1$  是  $P_1$  的前驱。又因为计算机系统中只有一个 CPU，

而且系统采用优先级调度，所以 C1 是 C2 的前趋，C2 是 C3 的前趋。可见，图中④应为 C2。

(26) 因为计算机系统中只有一台输出设备，所以 P1 执行结束后 P2 才能执行，故 P1 是 P2 的前趋；p2 执行结束后 p3 才能执行，p2 是 p3 的前趋。可见，图中⑤、⑥分别为 p2、P3

设文件索引节点中有 8 个地址项，每个地址项大小为 4 字节，其中 5 个地址项为直接地址索引，2 个地址项是一级间接地址索引，1 个地址项是二级间接地址索引，磁盘索引块和磁盘数据块大小均为 1KB。若要访问文件的逻辑块号分别为 5 和 518，则系统应分别采用(27)，而且可表示的单个文件最大长度是\_(28)KB。

(27)A. 直接地址索引和一级间接地址索引

B. 直接地址索引和二级间接地址索引

C. 一级间接地址索引和二级间接地址索引

D. 一级间接地址索引和一级间接地址索引

(28)A. 517

B. 1029

C. 16513

D. 66053

**【答案】C D**

**【解析】**本题考查操作系统文件管理方面的基础知识。

根据题意，磁盘索引块为 1KB 字节，每个地址项大小为 4 字节，故每个磁盘索引块可存放  $1024/4=256$  个物理块地址。又因为文件索引节点中有 8 个地址项，其中 5 个地址项为直接地址索引，这意味着逻辑块号为 0~4 的为直接地址索引；2 个地址项是一级间接地址索引，这意味着第一个地址项指出的物理块中存放逻辑块号为 5~260 的物理块号，第一个地址项指出的物理块中存放逻辑块号为 261~516 的物理块号；1 个地址项是二级间接地址索引，该地址项指出的物理块存放了 256 个间接索引表的地址，这 256 个间接索引表存放逻辑块号为 517~66052 的物理块号。

经上分析不难得出，若要访问文件的逻辑块号分别为 5 和 518，则系统应分别采用一级间接地址索引和二级间接地址索引。

(28) 单个文件的逻辑块号可以从 0~66052，而磁盘数据块大小为 1KB 字节，所以单个文件最大长度是 66053KB。

某开发小组欲开发一个规模较大、需求较明确的项目。开发小组对项目领域熟悉且该项目与小组开发过的某一项目相似，则适宜采用\_(29)开发过程模型。

(29)A. 瀑布

B. 演化

C. 螺旋

D. 喷泉

**【答案】A**

**【解析】**

项目规模大、开发小组对项目需求理解并了解相关领域，因此可以采用瀑布开发模型。演化模式适用于对软件需求缺乏准确认识的情况。螺旋模型在开发过程中加入风险分析。喷泉模型适合于面向对象的开发方法。

敏捷开发方法中，(30)认为每一种不同的项目都需要一套不同的策略、约定和方法论。

(30)A. 极限编程 (XP)

B. 水晶法 (Crystal)

C. 并列争球法 (Scrum)

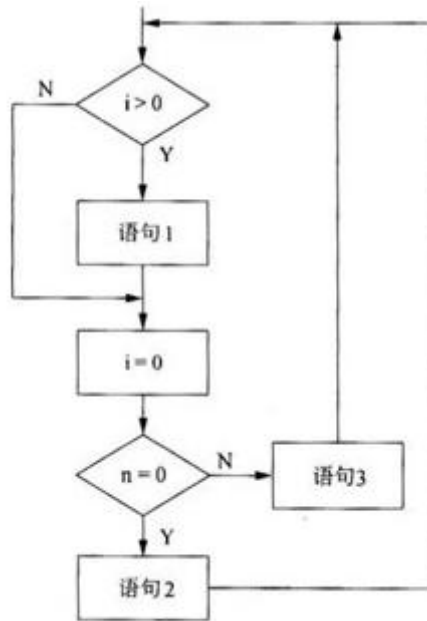
D. 自适应软件开发 (ASD)

**【答案】B**

**【解析】**

敏捷开发的总体目标是通过“尽可能早地、持续地对有价值的软件的交付”使客户满意。敏捷过程的典型方法很多，每一种方法基于一套原则，这些原则实现了敏捷方法所宣称的理念，即敏捷宣言。其中，极限编程 XP 是一种轻量级的软件开发方式，由价值观、原则、实践和行为 4 个部分组成，彼此相互依赖、关联，并通过行为贯穿于整个生存周期。水晶法 Crystal 认为每一个不同的项目都需要一套不同的策略、约定和方法论。并列争球法 Scrum 使用迭代的方法，并按需求的优先级来实现产品。自适应软件开发 ASD 有 6 个基本原则。

采用 McCabe 度量法计算下图的环路复杂件为(31)。



(31) A. 2

B. 3

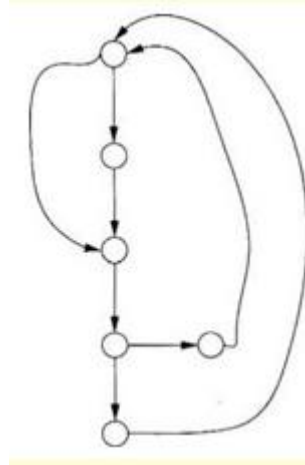
C. 4

D. 5

【答案】C

【解析】

题图可以用下图表示，图中顶点数 6，边数为 8，程序复杂度为  $m-n+2=8-6+2=4$ 。



在屏蔽软件错误的容错系统中，冗余附加技术的构成不包括\_(32)。

(32) A. 关键程序和数据的冗余存储及调用

B. 冗余备份程序的存储及调用

C. 实现错误检测和错误恢复的程序

D. 实现容错软件所需的固化程序试题

【答案】A

【解析】

冗余附加技术是指为实现结构、信息和时间冗余技术所需的资源和技术，包括程序、指

令、数据、存放和调动它们的空间和通道等。在屏蔽硬件错误的容错技术中，冗余附加技术包括：关键程序和数据的冗余及调用；检测、表决、切换、重构和复算的实现。在屏蔽软件错误的容错技术中，冗余附加技术包括：冗余备份程序的存储及调用；实现错误检测和错误恢复的程序；实现容错软件所需的固化程序。

以下关于文档的叙述中，不正确的是 (33)。

- (33)A. 文档仅仅描述和规定了软件的使用范围及相关的操作命令
- B. 文档是软件产品的一部分，没有文档的软件不能称之为软件产品
- C. 软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量
- D. 高质量文档对于发挥软件产品的效益有着重要的意义

**【答案】A**

**【解析】**

文档是系统建设过程的“痕迹”，是系统维护人员的指南，是开发人员与用户交流的工具。文档不仅仅描述和规定软件的适用范围及相关的操作命令。软件包括程序和文档，因此没有文档的软件不能称之为软件产品。软件文档的编制在软件开发中是相当重要的，高质量的文档对于发挥软件产品的效益有着重要的意义。

由于信用卡公司升级了其信用卡支付系统，导致超市的原有信息系统也需要做相应的修改工作，该类维护属于 (34)。

- (34)A. 正确性维护                      B. 适应性维护                      C. 完善性维护                      D. 预防性维护

**【答案】B**

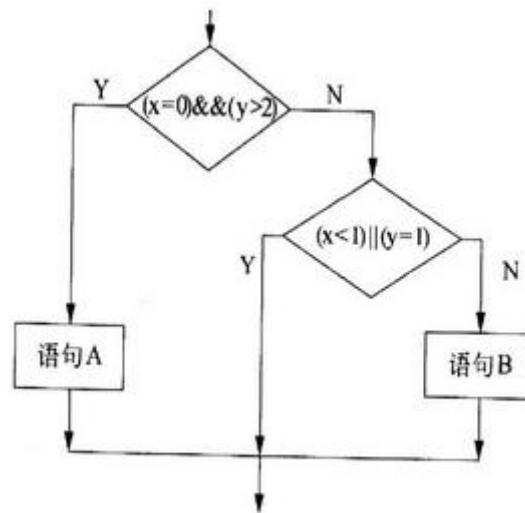
**【解析】**

软件维护一般包括下面四个方面。正确性维护是指改正在系统开发阶段已经发生而在系统测试阶段尚未发生的错误。适应性维护是指使应用软件适应信息技术变化和管理需求变化而进行的修改。完善性维护为扩充功能和改善性能而进行的修改。预防性维护是为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的编号，主动增加预防性的新的功能，以使应用系统适应各类变化而不会被淘汰。本题超市信息系统为了适应信用卡支付系统而做了相应的修改工作，是一个典型的适应性维护。

用白盒测试方法对下图所示的程序进行测试，设计了 4 个测试用例：①( $x = 0, y = 3$ )、



② ( $x=1, y=2$ )、③ ( $x=-1, y=2$ )和④ ( $x=3, y=1$ )。测试用例①②实现了 (35) 覆盖：若要完成路径覆盖，则可用测试用例 (36)。



- (35) A. 语句                      B. 条件                      C. 判定                      D. 路径
- (36) A. ①②                      B. ②③                      C. ①②③                      D. ①③④

【答案】A C

【解析】

白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的路径和过程进行测试，检查是否满足设计的需要。其常用的技术有逻辑覆盖、循环覆盖 和基本路径测试。

在逻辑覆盖中，语句覆盖是指选择足够的测试数据使被测试程序中每条语句至少执行一次。判定覆盖是指选择足够的测试数据使被测试程序中每个判定表达式至少获得一次“真”值和“假”值。条件覆盖是指构造一组测试用例，使得每一判定语句中每个逻辑条件的各种可能的值至少满足一次。路径覆盖是指覆盖被测程序中所有可能的路径。

(36) 本题的实例中，测试用例①会执行语句 A，测试用例②会执行语句 B，测试用例③和④不执行语句。因此测试用例①②可以完成语句覆盖，不能完成判定、条件和路径覆盖。要完成路径覆盖，需要测试用例①②③或测试用例①②④。

在面向对象技术中，对象具有以下特性： (37)。

①清晰的边界②良好定义的行为③确定的位置和数量④可扩展性

- (37) A. ②④                      B. ①②③④                      C. ①②④                      D. ①②

【答案】C

**【解析】** 本题考查面向对象的基本知识。

在面向对象技术中，对象是基本的运行时的实体，它既包括数据（属性），也包括作用于数据的操作（行为）。一个对象把属性和行为封装为一个整体，与其他对象之间有清晰的边界，有良好定义的行为和可扩展性。对象位置和数量，由使用其的对象或系统确定。

在面向对象技术中，(38) 说明一个对象具有多种形态，(39) 定义超类与子类的关系。

(38) A. 继承                      B. 组合                      C. 封装                      D. 多态

(39) A. 继承                      B. 组合                      C. 封装                      D. 多态

**【答案】** D    A

**【解析】** 本题考查面向对象的基本知识。

面向对象技术中，继承关系是一种模仿现实世界中继承关系的一种类之间的关系，是超类（父类）和子类之间共享数据和方法的机制。在定义和实现一个类的时候，可以在一个已经存在的类的基础上来进行，把这个已经存在的类所定义的内容作为自己的内容，并加入新的内容。组合表示对象之间整体与部分的关系。封装是一种信息隐藏技术，其目的是使对象（组件）的使用者和生产者分离，也就使地其他开发人员无需了解所要使用的软件组件内部的工作机制，只需知道如何使用组件，即组件提供的功能及其接口。多态（polymorphism）是不同的对象收到同一消息可以产生完全不同的结果的现象，使得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，达到同一消息就可以调用不同的方法，即多种形态。

如果要表示待开发软件系统中软件组件和硬件之间的物理关系，通常采用 UML 中的(40)。

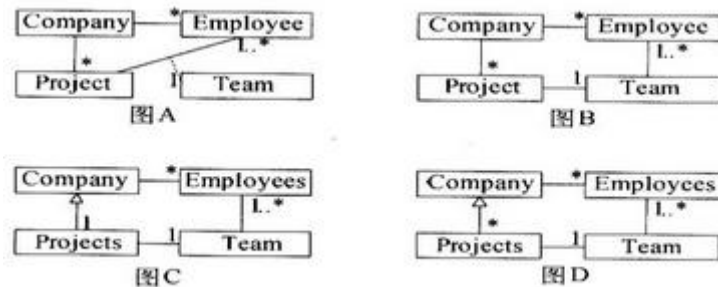
(40) A. 组件图                      B. 部署图                      C. 类图                      D. 网络图

**【答案】** B

**【解析】** 本题考查统一建模语言（UML）的基本知识。

UML2.0 中提供了多种图形。组件图（component diagram）展现了一组组件之间的组织和依赖，专注于系统的静态实现视图，与类图相关，通常把组件映射为一个或多个类、接口或协作。部署图（deployment diagram）展现了运行处理节点以及其中构件的配置。部署图给出了体系结构的静态实施视图。它与构件图相关，通常一个节点包含一个或多个构件。类图（class diagram）展现了一组对象、接口、协作和它们之间的关系，在开发软件系统时，类图用于对系统的静态设计视图建模。

对于场景：一个公司负责多个项目，每个项目（Project）由一个员工（Employee）团队（Team）来开发。下列 UML 概念图中，(41)最适合描述这一场景。



- (41) A. 图 A      B. 图 B      C. 图 C      D. 图 D

**【答案】B**

**【解析】** 本题考查采用统一建模语言（UML）进行面向对象建模的基本知识。

面向对象分析是从按对象分类的角度来创建对象领域的描述。领域的分解包括定义概念、属性和重要的关联。其结果可以被表示成领域模型，用一组显示领域概念或对象的图形来表示领域模型。通常首先发现和确定业务对象，然后组织对象并记录对象之间的主要概念关系。类图以图形化的用以描述对象及其关联关系。在该图中还包括多重性、关联关系、泛化/特化关系以及聚合关系。

从场景中先识别名词性术语，包括公司（Company）、项目（Project）、员工（Employee）、和团队（Team），再识别这些术语之间的关联关系。一个公司负责多个项目，公司和项目之间具有 1 对多的关联关系；一个项目由一个员工团队来开发，项目到团队的管理关系是 1，而一个（员工）团队是由多名员工组成，而且没有员工称不上是团队，所以一个团队至少和一个员工关联。

UML 中接口可用于 (42)。

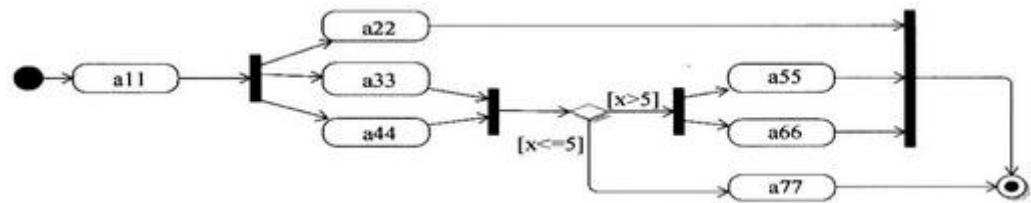
- (42) A. 提供构造型（stereotype）名称为《interface》的具体类  
B. Java 和 C++ 程序设计中，而 C# 程序设计中不支持  
C. 定义可以在多个类中重用的可执行逻辑  
D. 声明对象类所需要的服务

**【答案】D**

**【解析】** 本题考查统一建模语言（UML）建模的基本知识。

在采用 UML 进行面向对象系统建模时，会用 UML 中构造型（stereotype）名称为 «interface» 来表示接口这一概念，声明对象类所需要的服务，而服务具体如何执行，有实现它的具体类完成。

下列活动图中可以同时执行的活动是 (43)。



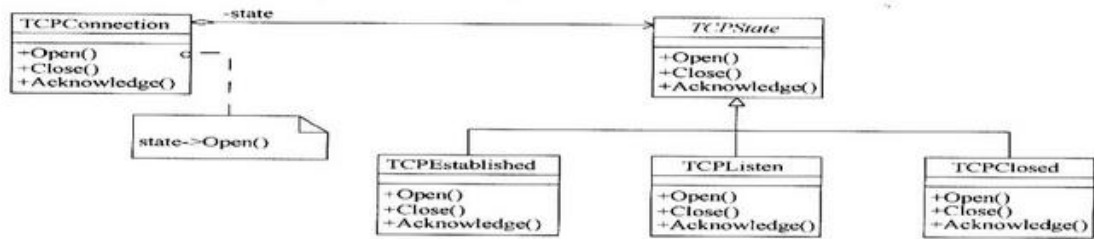
(43) A. a44 和 a66      B. a22, a33 和 a44      C. a11 和 a77      D. a66 和 a77

**【答案】B**

**【解析】** 本题考查统一建模语言（UML）的基本知识。

UML2.0 中提供的活动图（activity diagram）是一种特殊的状态图，它展现了在系统内从一个活动到另一个活动的流程。活动图专注于系统的动态视图，它对于系统的功能建模特别重要，并强调对象间的控制流程。通常用于在建模用例图之后，对复杂用例进行进一步细化。活动图中可以用条状图表示同步的起始点和结束点，其间的活动可以同时执行，如图中 a22、a33 和 a44，中 a11 执行完后，到同步起始，其后各自执行，同步结束后的活动必须等同步结束点之前的活动全部执行完之后才能继续，如 a33 和 a44 都结束后，才进入后续判定。

每种设计模式都有特定的意图。(44) 模式使得一个对象在其内部状态改变时通过调用另一个类中的方法改变其行为，使这个对象看起来如同修改了它的类。下图是采用该模式的有关 TCP 连接的结构图实例。该模式的核心思想是引入抽象类 (45) 来表示 TCP 连接的状态，声明不同操作状态的公共接口，其子类实现与特定状态相关的行为。当一个 (46) 对象收到其他对象的请求时，它根据自身的当前状态做出不同的反应。



(44) A. 适配器 (Adapter)

B. 命令 (Command)

C. 观察者 (Visitor)

D. 状态 (State)

(45) A. TCPConnection

B. state

C. TCPState

D. TCPEstablished

(46) A. TCPConnection

B. state

C. TCPState

D. TCPEstablished

**【答案】D C A**

**【解析】**本题考查设计模式的基本知识。

每种设计模式都有特定的意图。适配器 (Adapter) 模式将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。命令 (Command) 模式将一个请求封装为一个对象，从而使使用者可以采用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。观察者 (Visitor) 模式表示一个作用于某对象结构中的各元素的操作，是使用者可以在不改变各元素的类的前提下定义作用于这些元素的新操作。状态 (State) 模式是使得一个对象在其内部状态改变时通过调用另一个类中的方法改变其行为，使这个对象看起来如同修改了它的类。

(45) 题目中是一个标识网络连接的实例，网络连接类为 TCPConnection，其对象的状态处于不同状态之一：连接已建立 (Established)、正在监听 (Listening)、连接已关闭 (Closed)。当一个 TCPConnection 对象受到其他对象的请求时，它根据自身的当前状态作出不一致的响应。例如，一个 Open 请求的结果依赖于该连接是处于连接已关闭状态还是连接已建立状态。State 模式描述了 TCPConnection 如何做每一种状态下表现出不同的行为。这一模式思想是引入了一个称为 TCPState 的抽象类来表示网络的连接状态。TCPState 类为各表示不同的操作状态的子类声明了一个公共接口。TCPState 的子类实现与特定状态相关的行为。例如，TCPEstablished 和 TCPClosed 类分别实现了特定于 TCPConnection 的连接已建立状态和连接已关闭状态的行为。

(46) 题目中是一个标识网络连接的实例，网络连接类为 TCPConnection，其对象的状态处于不同状态之一：连接已建立 (Established)、正在监听 (Listening)、连接已关闭 (Closed)。当一个 TCPConnection 对象受到其他对象的请求时，它根据自身的当前状态出不

(47) A. 将 A 标识为 final                      B. 将 A 标识为 abstract

C. 将单例 (Singleton) 模式应用于 A        D. 将备忘 (Memento) 模式应用于 A

将类标识为 `final` 限制类不能再被继承；将设计为 `abstract` 表示类中定义出类提供什么服务，而有些具体服务需要通过其子类来实现；单例（Singleton）模式是指系统运行过程中，一个类只有一个对象实例；备忘模式是指不破坏封装性的前提下捕获一个对象的内部状态，并在该对象之外保存这个状态。

(48) A. 解释程序      B. 编译程序      C. 链接程序      D. 汇编程序

解释程序也称为解释器，它可以直接解释执行源程序，或者将源程序翻译成某种中间表示形式后再加以执行；而编译程序（编译器）则首先将源程序翻译成目标语言程序，然后在计算机上运行目标程序。汇编程序的功能是将汇编语言所编写的源程序翻译成机器指令程序。链接程序将各目标程序连接形成为可执行程序。

在对程序语言进行翻译的过程中，常采用一些与之等价的中间代码表示形式。常用的中间代码表示不包括\_(49)。

- (49) A. 树                      B. 后缀式                      C. 四元式                      D. 正则式

**【答案】D**

**【解析】**本考查程序语言基础知识。

从原理上讲，对源程序进行语义分析之后就可以直接生成目标代码，但由于源程序与目标代码的逻辑结构往往差别很大，特别是考虑到具体机器指令系统的特点，要使翻译一次到位很困难，而且用语法制导方式机械生成的目标代码往往是繁琐和低效的，因此有必要采用一种中间代码，将源程序首先翻译成中间代码表示形式，以利于进行与机器无关的优化处理。由于中间代码实际上也起着编译器前端和后端分水岭的作用，所以使用中间代码也有助于提高编译程序的可移植性。常用的中间代码有后缀式、三元式、四元式和树等形式。

以下关于程序错误的叙述中，正确的是\_(50)。

- A. 编译正确的程序必然不包含语法错误  
B. 编译正确的程序必然不包含语义错误  
C. 除数为 0 的错误可以在语义分析阶段检查出来  
D. 除数为 0 的错误可以在语法分析阶段检查出来

**【答案】A**

**【解析】**本题考查程序语言基础知识。

编译程序的工作过程可以分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等阶段。

用户编写的源程序不可避免地会有一些错误，这些错误大致可分为静态错误和动态错误。动态错误也称动态语义错误，它们发生在程序运行时，例如变量取零时作除数、引用数组元素下标错误等。静态错误是指编译阶段发现的程序错误，可分为语法错误和静态语义错误，如单词拼写错误、标点符号错、表达式中缺少操作数、括号不匹配等有关语言结构上的错误称为语法错误，而语义分析时发现的运算符与运算对象类型不合法等错误属于静态语义错误。

关系 R1 和 R2 如下图所示：

R1				R2			
A	B	C	D	C	D	E	F
a	d	c	e	a	e	c	a
c	b	a	e	a	e	a	b
d	e	c	e	c	e	b	c
e	f	d	a				

若进行  $R1R2$  运算，则结果集为 (51) 元关系，共有 (52) 个元组。

(51) A. 4                      B. 5                      C. 6                      D. 7

(52) A. 4                      B. 5                      C. 6                      D. 7

【答案】C A

【解析】本题考查数据库系统中关系代数运算方面的基础知识。

试题 (51) 的正确选项为 C。根据题意  $R1 \bowtie R2$  为自然联接，自然联接是一种特殊的等值联接，它要求两个关系中进行比较的分量必须是相同的属性，并且在结果集中将重复属性列去掉， $R2.C$  和  $R2.D$  去掉，故结果集为 6 元关系。

试题 (52) 的正确选项为 A。本题比较的条件为 “ $R1.C=R2.C \wedge R1.D=R2.D$ ”，从下图所示的  $R1 \bowtie R2$  的结果集中可见，共有 4 个元组满足条件，分别是第 3 个、第 4 个、第 5 个和第 9 个元组。

R1XR2							
R1A	R1B	R1C	R1D	R2C	R2D	R2E	R2F
a	d	c	e	a	e	c	a
a	d	c	e	a	e	a	b
a	d	c	e	c	e	b	c
c	b	a	e	a	e	c	a
c	b	a	e	a	e	a	b
c	b	a	e	c	e	b	c
d	e	c	e	a	e	c	a
d	e	c	e	a	e	a	b
d	e	c	e	c	e	b	c
e	f	d	a	a	e	c	a
e	f	d	a	a	e	a	b
e	f	d	a	c	e	b	c

设有关系模式  $R(E, N, M, L, Q)$ ，其函数依赖集为  $F=\{E \rightarrow N, EM \rightarrow Q, M \rightarrow L\}$ 。则关系模式  $R$  达到了 (53)；该关系模式 (54)。

(53) A. 1NF                      B. 2NF                      C. 3NF                      D. BCNF

(54) A. 无需进行分解，因为已经达到了 3NF

B. 无需进行分解，因为已经达到了 BCNF



- C. 尽管不存在部分函数依赖, 但还存在传递依赖, 所以需要进行分解
- D. 需要进行分解, 因为存在冗余、修改操作的不一致性、插入和删除异常

**【答案】A D**

**【解析】**本题考查关系数据库方面的基础知识。

试题 (53) 的正确选项为 A。根据题意 R 关系中的 EM 可决定该关系的所有属性, 所以 R 关系的主键为 EM; 又因为,  $EM \rightarrow Q$ , 而  $E \rightarrow N$ ,  $M \rightarrow L$ , 可以得出 N 和 L 都部分依赖于码, 所以, 该关系模式属于 1NF。

试题 (54) 的正确选项为 D。关系模式 R 属于 1NF, 1NF 存在冗余度大、修改操作的不一致性、插入异常和删除异常四个问题, 所以 R 需要进行分解。

已知关系模式: 图书 (图书编号, 图书类型, 图书名称, 作者, 出版社, 出版日期, ISBN), 图书编号唯一识别一本图书。建立“计算机”类图书的视图 Computer-BOOK, 并要求进行修改、插入操作时保证该视图只有计算机类的图书。实现上述要求的 SQL 语句如下:

```
CREATE (55)
AS SELECT 图书编号, 图书名称, 作者, 出版社, 出版日期
FROM 图书
WHERE 图书类型='计算机'
(56);
```

- (55) A. TABLE Computer-BOOK                      B. VIEW Computer-BOOK
- C. Computer-BOOK TABLE                      D. Computer-BOOK VIEW
- (56) A. FOR ALL                      B. PUBLIC
- C. WITH CHECK OPTION                      D. WITH GRANT OPTION

**【答案】B C**

**【解析】**本题考查数据库系统中关系代数运算方面的基础知识。

因为创建视图的语句格式如下:

```
CREATE VIEW 视图名 (列表名)
AS SELECT 查询子句
[WITH CHECK OPTION];
```

其中, WITH CHECK OPTION 表示对 UPDATE, INSERT, DELETE 操作时保证更新、插入或删除的行满足视图定义中的谓词条件 (即子查询中的条件表达式)。另外, 组成视图的属性列名或者全部省略或者全部指定。如果省略属性列名, 则隐含该视图由 SELECT 子查询目标列的主属性组成。

可见，完整的 Computer-BOOK 视图创建语句如下：

```
CREATE VIEW Computer-BOOK
AS SELECT 图书编号, 图书名称, 作者, 出版社, 出版日期
FROM 图书
WHERE 图书类型='计算机'
WITH CHECK OPTION;
```

在字符串的模式匹配过程中，如果模式串的每个字符依次和主串中一个连续的字符序列相等，则称为匹配成功。如果不能在主串中找到与模式串相同的子串，则称为匹配失败。在布鲁特-福斯模式匹配算法（朴素的或基本的模式匹配）中，若主串和模式串的长度分别为  $n$  和  $m$ （且  $n$  远大于  $m$ ），且恰好在主串末尾的  $m$  个字符处匹配成功，则在上述的模式匹配过程中，字符的比较次数最多为 (57)。

- (57) A.  $n*m$                       B.  $(n-m+1)*m$                       C.  $(n-m-1)*m$                       D.  $(n-m)*n$

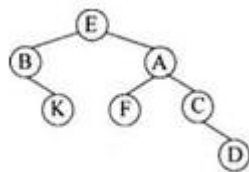
【答案】B

【解析】本题考查数据结构基础知识。

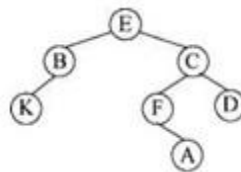
假设主串和模式串的长度分别为  $n$  和  $m$ ，位置序号从 0 开始计算。设从主串的第  $i$  个位置开始与模式串匹配成功，在前  $i$  趟匹配中（位置  $0 \sim i-1$ ），每趟不成功的匹配都是模式串的第一个字符与主串中相应的字符不相同，则在前  $i$  趟匹配中，字符的比较共进行了  $i$  次，而第  $i$  趟（从位置  $i$  开始）成功匹配的字符比较次数为  $m$ ，所以总的字符比较次数为  $i+m$ （ $0 \leq i \leq n-m$ ）。

而在最坏情况下，每一趟不成功的匹配都是模式串的最后一个字符与主串中相应的字符不相等，则主串中新一趟的起始位置为  $i-m+2$ 。若设从主串的第  $f$  个字符开始匹配时成功，则前  $i$  趟不成功的匹配中，每趟都比较了  $m$  次，总共比较了  $i \times m$  次，第  $i+1$  趟的成功匹配也比較了  $m$  次。因此，最坏情况下的比较次数为  $(n-m+1)*m$ 。

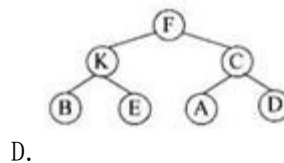
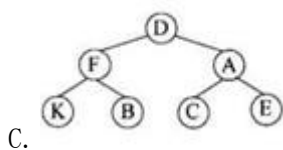
若某二叉树的后序遍历序列为 KBFDCAE，中序遍历序列为 BKEFACD，则该二叉树为 (58)。



(58) A.



B.



【答案】A

【解析】本题考查数据结构基础知识。

根据后序遍历序列 KBFDCAE, 可以确定根结点为 E, 然后根据中序遍历序列(为 BKEFACD), 可以确定 B、K 为左子树的结点, F、A、C、D 是右子树的结点。再根据左子树的后序遍历序列为 KB、中序遍历序列为 BK, 所以可以确定 B 是左子树的根结点, K 是结点 B 的右子树上。同理可推出其他结点的位置。

在 13 个元素构成的有序  $M[1 \cdots 13]$  中进行折半查找(向下取整), 若找到的元素为  $M[4]$ , 则被比较的元素依次为 (59)

(59) A.  $M[7]$ 、 $M[3]$ 、 $M[5]$ ,  $M[4]$

B.  $M[7]$ 、 $M[5]$ ,  $M[4]$

C.  $M[7]$ 、 $M[6]$ 、 $M[4]$

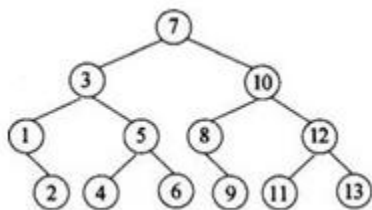
D.  $M[7]$ 、 $M[4]$

【答案】A

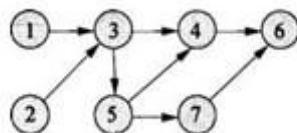
【解析】本题考查数据结构基础知识。

设查找表的元素存储在一维数组  $r[1..n]$  中, 在表中的元素已经按关键字递增方式排序的情况下, 进行折半查找的方法是: 首先将待查元素的关键字(key)值与表  $r$  中间位置上(下标为  $mid$ )记录的关键字进行比较, 若相等, 则查找成功; 若  $key > r[mid].key$ , 则说明待查记录只可能在后半个子表  $r[mid+1..ti]$  中, 下一步应在后半个子表中进行查找, 若  $key < r[mid].key$ , 说明待查记录只可能在前半个子表  $r[1..mid-1]$  中, 下一步应在  $r$  的前半个子表中进行查找, 通过逐步缩小范围, 直到查找成功或子表为空时失败为止。

折半查找的过程可以用一棵二叉树描述, 方法是以当前查找区间的中间位置序号作为根, 左半个子表和右半个子表中的记录序号分别作为根的左子树和右子树上的节点, 具有 13 个节点的折半查找判定树如下图所示。



拓扑排序是将有向图中所有顶点排成一个线性序列的过程，并且该序列满足：若在 AOV 网中从顶点  $v_i$  到  $v_j$  有一条路径，则顶点  $v_i$  必然在顶点  $v_j$  之前。对于下面所示的有向图，(60) 是其拓扑序列。



- (60) A. 1234576      B. 1235467      C. 2135476      D. 2134567

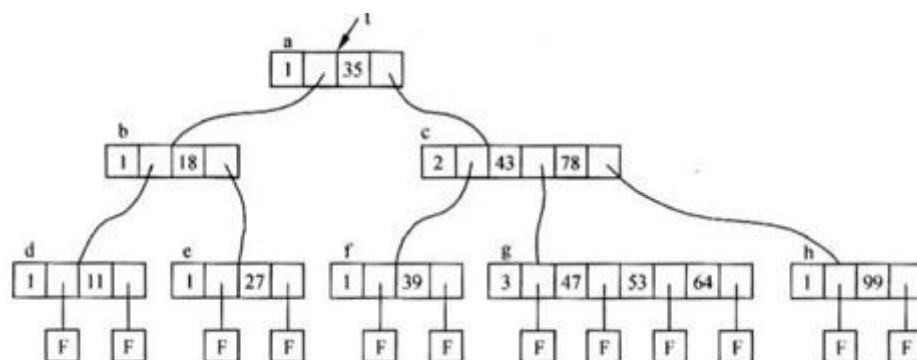
**【答案】C**

**【解析】** 本题考查数据结构基础知识。

拓扑排序是将 AOV 网中所有顶点排成一个线性序列的过程，并且该序列满足：若在 AOV 网中从顶点  $v_i$  到  $v_j$  有一条路径，则在该线性序列中，顶点  $v_i$  必然在顶点  $v_j$  之前。

题中所示有向图的拓扑序列有：1235476，2135476，1235746，2135746

下图所示为一棵 M 阶 B-树，M 最有可能的值为 (61)。



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

**【答案】D**

**【解析】** 本题考查数据结构基础知识。

在 m 阶 B-树的定义中，要求：

- ① 树中的每个节点至多有 M 棵子树
- ② 若根节点不是叶子节点，则至少有两棵子树。
- ③ 除根之外的所有非终端节点至少有  $\lceil M/2 \rceil$  棵子树。

因此，题图中所示 B-树最可能为 4 阶 B 树。

将数组 {1, 1, 2, 4, 7, 5} 从小到大排序，若采用 (62) 排序算法，则元素之间需要进行的比

较次数最少，共需要进行 (63) 次元素之间的比较。

(62) A. 直接插入                      B. 归并                      C. 堆                      D. 快速

(63) A. 5                      B. 6                      C. 7                      D. 8

**【答案】A B**

**【解析】**

输入数组 {1, 1, 2, 4, 7, 5} 基本有序 (从小到大)，在这种情况下，插入排序算法的时间复杂度为  $O(n)$ ，归并排序和堆排序的时间复杂度为  $O(n \lg n)$ ，而快速排序的时间复杂度为  $O(n^2)$ 。

(63) 用插入排序算法排序该输入数组，第二个元素 1 需要和第一个元素 1 进行 1 次比较，第三个元素 2 需要和第二个元素 1 进行 1 次比较，第四个元素 4 需要和第三个元素 2 进行 1 次比较，第五个元素 7 需要和第四个元素 4 进行 1 次比较，第六个元素 5 需要和第五个元素 7 进行 1 次比较，比 7 小，和元素 7 交换，在和第四个元素 4 进行 1 次比较，得到最终的排序结果。因此一共需要进行 6 次比较。

霍夫曼编码将频繁出现的字符采用短编码，出现频率较低的字符采用长编码。具体的操作过程为：i) 以每个字符的出现频率作为关键字构建最小优先级队列；ii) 取出关键字最小的两个结点生成子树，根节点的关键字为孩子节点关键字之和，并将根节点插入到最小优先级队列中，直至得到一颗最优编码树。

霍夫曼编码方案是基于 (64) 策略的。用该方案对包含 a 到 f 六个字符的文件进行编码，文件包含 100,000 个字符，每个字符的出现频率 (用百分比表示) 如下表所示，则与固定长度编码相比，该编码方案节省了 (65) 存储空间。

字符	a	b	c	d	e	f
出现频率	18	32	4	8	12	26

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

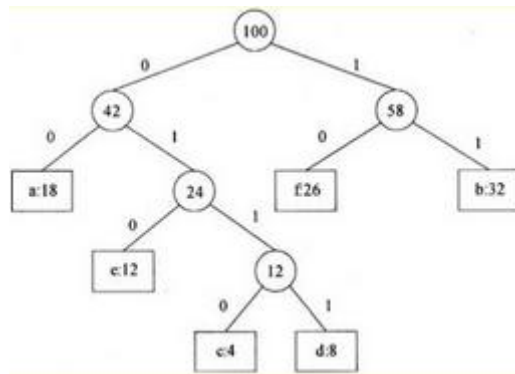
(65) A. 21%                      B. 27%                      C. 18%                      D. 36%

**【答案】B A**

**【解析】**

霍夫曼编码在构造最优编码树时，每次总是取出权值最小的两个结点来生成它们的父结点 (新的中间结点)，因此霍夫曼编码方案是基于贪心策略的。

(65) 对应该实例构造最优编码树如下图所示。



实例中包含 6 个字符，若用定长编码，则需要三位，对包含 100,000 个字符的文件，需要  $3 \times 100,000 = 300,000$  位的存储空间。而采用霍夫曼编码，则需要  $(18+26+32) \times 2 + 12 \times 3 + (4+8) \times 4 = 236,000$  位的存储空间，节省了 21% 的存储空间。

在 Windows 2003 Server 中启用配置 SNMP 服务时，必须以 (66) 身份登录才能完成 SNMP 服务的配置功能。

- (66) A. guest      B. 普通用户      C. administrator 组成员      D. user 组成员

**【答案】C**

**【解析】** 本题考查 Windows2003 中有关 SNMP 服务配置的操作权限。

Windows Server 2003 中配置 SNMP 服务时，必须以管理员身份或者 Administrators 组成员身份登录才能完成 SNMP 服务的配置功能。一般用户或者普通用户不能完成 SNMP 配置服务。

在 ASP 的内置对象中，(67) 对象可以修改 cookie 中的值。

- (67) A. request      B. response      C. application      D. session

**【答案】B**

**【解析】** 本题考查 ASP 的内置对象的基础知识。

在 ASP 的内置对象中，response 对象和 request 对象和 cookie 有关。其中，request 对象中的 Cookies 集合是服务器根据用户的请求，发出的所有 cookie 的值的集合，这些 Cookie 仅对相应的域有效，每个成员均为只读。Response 对象中的 Cookies 集合是服务器发回客户端的所有 Cookie 的值，这个集合为只写，所以只有 Response 对象可以修改 cookie 中的值。

分配给某公司网络的地址块是 220.17.192.0/20，该网络被划分为 (68) 个 C 类子网，

不属于该公司网络的子网地址是 (69)。

(68) A. 4                      B. 8                      C. 16                      D. 32

(69) A. 220. 17. 203. 0                      B. 220. 17. 205. 0

C. 220. 17. 207. 0                      D. 220. 17. 213. 0

**【答案】C D**

**【解析】**

220. 17. 192. 0 是一个 C 类网络地址，应该有 24 位子网掩码，现在仅采用 20 位子网掩码，少了 4 位，所以被划分成了 16 个子网。

(69) 这 16 个子网号的第三个字节都应该在 192+0~192+15 之间，由于 213 大于 192+15，所以 220. 17. 213. 0 不属于地址块 220. 17. 192. 0/20。

如果 DNS 服务器更新了某域名的 IP 地址，造成客户端域名解析故障，在客户端可以用两种方法解决此问题，其中一种是在 Windows 命令行下执行 (70) 命令。

(70) A. ipconfig /all                      B. ipconfig /renew

C. ipconfig/flushdns                      D. ipconfig/release

**【答案】C**

**【解析】** 本题考查 Windows 系统中 DNS 服务及相关配置命令的基础操作。

ipconfig 命令详解如下：

①具体功能

该命令用于显示所有当前的 TCP/IP 网络配置值、刷新动态主机配置协议 (DHCP) 和域名系统 (DNS) 设置。使用不带参数的 IPCONFIG 可以显示所有适配器的 IP 地址、子网掩码、默认网关。

②语法详解

```
ipconfig [/all] [/renew [adapter] [/release [adapter] [/flushdns] [/displaydns]
[/registerdns] [/showclassidpadapter] [/setclassidpadapter] [classID]
```

③参数说明

/all 显示所有适配器的完整 TCP/IP 配置信息。在没有该参数的情况下 IPCONFIG 只显示 IP 地址、子网掩码和各个适配器的默认网关值。适配器可以代表物理接口 (例如安装的网络适配器) 或逻辑接口 (例如拨号连接)。

/renew 更新所有适配器 (如果未指定适配器)，或特定适配器 (如果包含了 adapter 参数) 的

DHCP 配置。该参数仅在具有配置为自动获取 IP 地址的网卡的计算机上可用。要指定适配器名称，请输入使用不带参数的 IPCONFIG 命令显示的适配器名称。

/release[adapter] 发送 DHCPRELEASE 消息到 DHCP 服务器，以释放所有适配器(如果未指定适配器)或特定适配器(如果包含了 adapter 参数)的当前 DHCP 配置并丢弃 IP 地址配置。该参数可以禁用配置为自动获取 IP 地址的适配器的 TCP/IP。要指定适配器名称，请键入使用不带参数的 IPCONFIG 命令显示的适配器名称。

/flushdns 清理并重设 DNS 客户解析器缓存的内容。如有必要，在 DNS 疑难解答期间，可以使用本过程从缓存中丢弃否定性缓存记录和其他任何动态添加的记录。

DNS Client 服务为计算机解析和缓存 DNS 名称。为了要达到用最快速、最有效率的方式，让客户端能够迅速找到网域的验证服务，在 Win2000/XP 系统中，加入了 DNS 快取 (Cache) 的功能。当第一次在找到了目的主机的 IP 地址后，操作系统就会将所查询到的名称及 IP 地址记录在本机的 DNS 快取缓冲区中，下次客户端还需要再查询时，就不需要到 DNS 服务器上查询，而直接使用本机 DNS Cache 中的数据即可，所以你查询的结果始终是同一 IP 地址。这个服务关闭后，dns 还可以解析，但是本地无法储存 dns 缓存。

Computers will become more advanced and they will also become easier to use. Improved speed recognition will make the operation of a computer easier. Virtual reality (虚拟现实), the technology of (71) with a computer using all of the human senses, will also contribute to better human and computer (72). Other, exotic (奇异的) models of computation are being developed, including biological computing that uses living organisms, molecular computing that uses molecules with particular (73), and computing that uses DNA, the basic unit of heredity (遗传), to store data and carry out operations. These are examples of possible future computational platforms that, so far, are limited in abilities or are strictly (74). Scientists investigate them because of the physical limitations of miniaturizing circuits embedded in silicon. There are also (75) related to heat generated by even the tiniest of transistors.

- |                   |               |                 |               |
|-------------------|---------------|-----------------|---------------|
| (71)A. interact   | B. interactin | C. communica    | D. using      |
| (72)A. interfaces | B. behavior   | C. similarities | D. comparison |
| (73)A. software   | B. properties | C. programs     | D. hardware   |



(74) A. empirical                      B. real                      C. practical                      D. theoretical

(75) A. developments                      B. advantage                      C. limitations                      D. improvements

**【答案】** B   A   B   D   C

**【解析】** 本题考查对英语资料的阅读理解。

计算机将会变得越来越高级而且更易于使用。识别速度的提升将使计算机的操作更加容易。虚拟现实是使用人的感觉与计算机进行交互的一种技术，它也会使得人机界面更好。另外，各种新奇的计算模型正在不断发展，如生物计算使用人的器官、分子计算使用具有特定属性的分子、DNA 计算采用遗传的基本单元来存储数据和执行操作。到目前为止，这些都还属于未来计算平台，能力非常有限，并且只限于理论方面的研究。科学家们之所以研究这些计算模型，原因在于嵌入硅片中的微型芯片的体积的局限性，以及即使是非常小的晶体管所产生的热量方面的局限性。

试题一

- 某电子商务系统采用以数据库为中心的集成方式改进购物车的功能，详细需求如下：
- 1:加入购物车。顾客浏览商品，点击加入购物车，根据商品标识从商品表中读取商品信息，并更新购物车表。
  - 2:浏览购物车。顾客提交浏览购物车请求后，显示出购物车表中的商品信息。
  - 3:提交订单。顾客点击提交订单请求，后台计算购物车表中商品的总价（包括运费）加入订单表，将购物车表中的商品状态改为待付款，显示订单详情。若商家改变价格，则刷新后可看到更改后的价格。
  - 4:改变价格。商家查看订购自家商品的订单信息，根据特殊优惠条件修改价格，更新订单表中的商品价格。
  - 5:付款。顾客点击付款后，系统先根据顾客表中关联的支付账户，将转账请求（验证码、价格等）提交给支付系统（如信用卡系统）进行转账；然后根据转账结果返回支付状态并更改购物车表中商品的状态。
  - 6:物流跟踪。商家发货后，需按订单标识添加物流标识（物流公司、运单号）；然后可根据顾客或商家的标识以及订单标识，查询订单表中的物流标识，并从相应物流系统查询物流信息。
  - 7:生成报表。根据管理员和商家设置的报表选项，从订单表、商品表以及商品分类表中读取数据，调用第三方服务 Crystal Reports 生成相关报表。
  - 8:维护信息。管理员维护（增、删、改、查）顾客表、商品分类表和商品表中的信息。
- 现采用结构化方法实现上述需求，在系统分析阶段得到如图 1-1 所示的顶层数据流图和图 1-2 所示的 0 层数据流图。

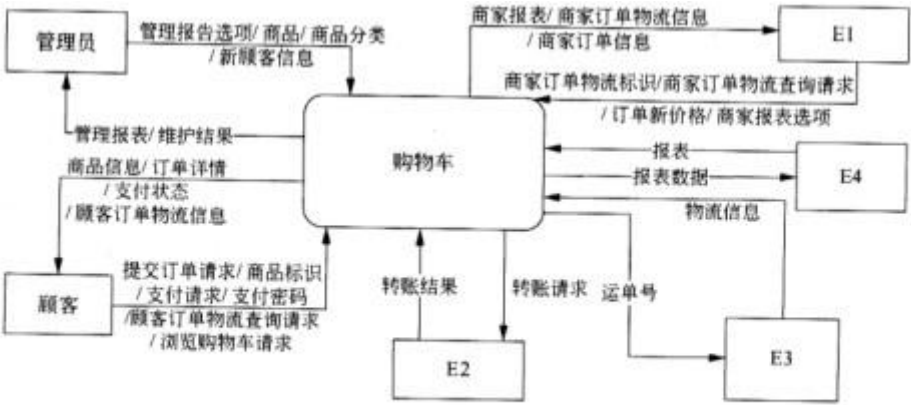


图 1-1 顶层数据流图

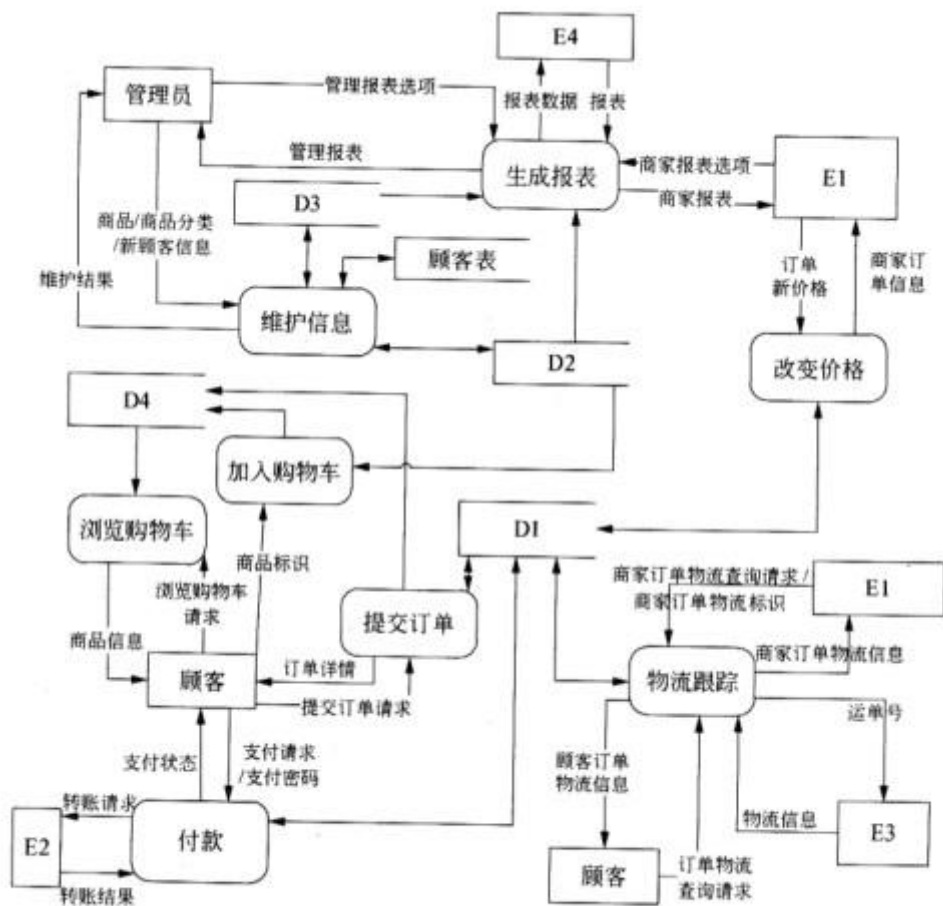


图 1-2 0 层数据流图

### 【问题 1】

使用说明中的词语，给出图 1-1 中的实体 E1~E4 的名称。

E1: 商家. E2: 支付系统. E3: 物流系统. E4: Crystal Report 或第三方服务.

解析: 本问题考查顶层 DFD。顶层 DFD 一般用来确定系统边界，将待开发系统看作一个加工，图中只有唯一的一个处理和—些外部实体，以及这两者之间的输入输出数据流。题目要求根据描述确定图中的外部实体。外部实体可以是和系统交互的人，以及和系统交互的外部系统或服务。分析题目中的描述，并结合已经在顶层数据流图中给出的数据流进行分析。分析题目中的说明，管理员维护系统中信息，顾客和商家是系统的主要使用者；商家查看订购自家商品的订单信息，根据特殊优惠条件修改价格，更新订单表中的商品价格，还可以添加物流标识并进行物流跟踪；使用支付系统进行支付，通过物流系统进行物流跟踪，以及第三方服务 Crystal Report 生成报表。可以看出，和系统的交互者包括管理员、顾客、商家三类人，支付系统、物流系统和 Crystal Report 三种外部系统。

对应图 1-1 中数据流和实体的对应关系，管理员和顾客已经给出，可知 E1 为商家，E2 为支

付系统，E3 为物流系统，E4 为第三方服务 Crystal Report。

【问题 2】

使用说明中的词语，给出图 1-2 中的数据存储 D1~D4 的名称。

D1：订单表. D2:商品表. D3:商品分类表. D4:购物车表.

本问题考查 0 层 DFD 中数据存储的确定。根据说明中所描述的处理和相关数据存储之间的连接关系，判定每个数据存储。加入购物车和浏览购物车分别读取和更新购物车表中的数据；改变价格和提交订单要读取和更新订单表中的数据；维护信息时需要维护商品表和商品分类表，生成报街要读取商品表和商品分类表，加入购物车时，需要读取商品表中的商品信息。根据描述和图 1-2 中的数据存储的输入输出数据流提示，可知：D1 为订单表，D2 为商品表，D3 为商品分类表，D4 为购物车表。

【问题 3】

图 1-2 中缺失了数据流，请用说明或图 1-2 中的词语，给出其起点和终点。

图 1-2 中缺少的数据流：	
起点	终点
付款	D4 或购物车表
D4 或购物车表	提交订单
顾客表	付款
D1 或订单表	生成报表

本问题考查绘制 0 层 DFD 时是否将本层该绘制的数据流全部绘制出。对照顶层数 据流图和 0 层数据流图，检查是否和外部实体之间的数据流一致；仔细对照说明中的描述和图 1-2 中给出的数据流，检查是否遗漏掉信息。说明中：提交订单处理时，后台计算购物车表中的商品的总价，即需要读出购物车表中的相关价格进行计算，读取出其中数据；付款需要读取顾客表中关联的支付账户，并向支付系统提交转账请求，然后根据转账结果更改购物车表中商品的状态；生成报告时根据管理员和商家设置的报告选项，从订单表、商品表以及商品分类表中读取数据，再调用第三方服务 Crystal Reports 生成相关报告。将这些说明和图 1-2 进行对照，发现缺少了从付款到购物车表（D4）、从购物车表到提交订单、从顾客表到付款，以及从订单表（D1）到生成报表等 4 条数据流。

【问题 4】

根据说明，给出数据流“转账请求”、“顾客订单物流查询请求”和“商家订单物流查询请求”的各组成数据项。

转账请求=验证码+价格+账号信息 顾客订单物流查询请求=顾客标识+订单标识

商家订单物流查询请求=商家标识+{订单标识}

解析：本问题考查在绘制数据流图时数据流的数据项组成。数据流图描述了系统的分解，但它并没有给出图中各成分的说明。通常采用数据字典为数据流图中的每个数据流、文件、处理，以及组成数据流或文件的数据项做出说明。对于数据流，通常列出该数据流的各组成数据项，并采用数据字典定义式中出现的符号进行表示，如“=”表示“被定义为”，“+”表示“与” “{……}”广表示其中数据可以有多个等等。本试题说明中：付款时，需根据顾客表中关联的支付账户将转账请求（验证码、价格等）提交给支付系统；物流跟踪时，根据顾客和商家的标识以及订单标识进行查询，而且在改变价格时商家查看订购自家商品的订单信息，可知商家可以查询一批订单。可以看出，提交给支付系统的请求中包含支付账户、验证码与价格；顾客订单查询请求中有顾客标识、订单标识；商家订单查询请求中有商家标识、订单标识（一批订购自家商品的订单标识）。因此“转账请求=支付账户+验证码+价格”；“商家订单物流查询请求=物流标识+{订单标识}”；“顾客订单物流标识=物流标识+订单标识”。

试题二

某会议策划公司为了方便客户，便于开展和管理各项业务活动，需要构建一个基于网络的会议预定系统。

【需求分析】

会议策划公司设有受理部、策划部和其他部门。部门信息包括部门号、部门名称、部门主管、电话和邮箱号。每个部门有多名员工处理部门的日常事务，每名员工只能在一个部门工作。每个部门有一名主管负责管理本部门的事务和人员。

员工信息包括员工号、姓名、部门号、职位、联系方式和工资；其中，职位包括主管、业务员、策划员等。业务员负责受理会议申请。若申请符合公司规定，则置受理标志并填写业务员的员工号。策划部主管为已受理的会议申请制定策划任务，包括策划内容、参与人数、要求完成时间等。一个已受理的会议申请对应一个策划任务，一个策划任务只对应一个已受理的会议申请，但一个策划任务可由多名策划员参与执行，且一名策划员可以参与多项策划任务。

客户信息包括客户号、单位名称、通信地址、所属省份、联系人、联系电话、银行账号。其中，一个客户号唯一标识一个客户。一个客户可以提交多个会议申请，但一个会议申请对应唯一的一个客户号。

会议申请信息包括申请号、开会日期、会议地点、持续天数、会议人数、预算费用、会议类型、酒店要求、会议室要求、客房类型、客房数、联系人、联系方式、受理标志和业务员的员工号等。客房类型有豪华套房、普通套房、标准间、三人间等，且申请号和客房类型决定客房数。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如下：

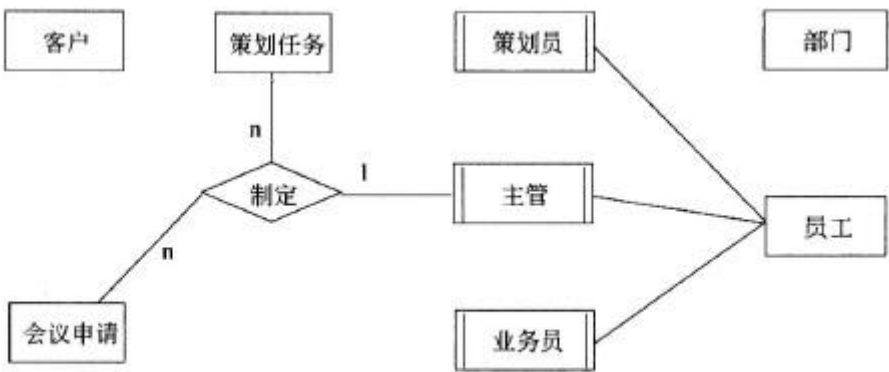


图 2-1 实体联系图

### 【关系模式设计】

部门（部门号，部门名称，主管，电话，邮箱号）

员工（员工号，姓名，（a），联系方式，工资）

客户（客户号，单位名称，通信地址，所属省份，联系人，联系电话，银行账号）

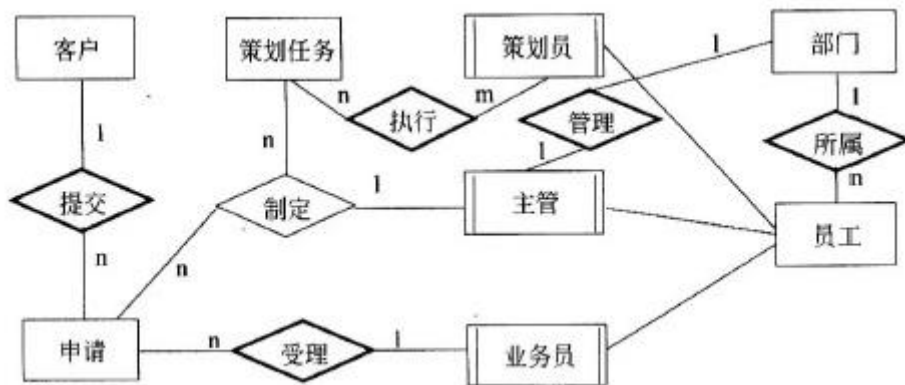
会议申请（（b），开会日期，会议地点，持续天数，会议人数，预算费用，会议类型，酒店要求，会议室要求，客房数，联系人，联系方式，受理标志，员工号）

策划任务（（c），策划内容，参与人数，要求完成时间）

执行策划（（d），实际完成时间）

### 【问题 1】

根据问题描述，补充五个联系、联系的类型，完善图 2-1 的实体联系图



根据题意，一个客户可以提交多个会议申请，但一个会议申请对应唯一的一个客户号，故应在客户和会议申请之间增加一个 1 :n 的“提交”联系；由于业务员负责受理会议申请，若申请符合公司规定则置受理标志并填写业务员的员工号，因此业务员和会议申请之间有一个 1 :n 的“受理”联系；由于一个已受理的会议申请对应一个策划任务，一个策划任务只对应一个已受理的会议申请，但一个策划任务可由多名策划员参与执行，且一名策划员可以参与多项策划任务，因此策划任务和策划员之间有一个 n : m 的“执行”联系；由于每个部门有多名员工处理部门的日常事务，每名员工只能在一个部门工作，因此部门和员工之间有一个 1 :n 的“所属”联系；又由于每个部门有一名主管负责管理本部门的事务和人员，而该主管也是一名员工，因此主管和部门之间有一个 1 : 1 的“管理”联系。

### 【问题 2】

根据实体联系图，将关系模式中的空（a）~（d）补充完整（1 个空缺处可能有多个数据项）。对会议申请、策划任务和执行策划关系模式，用下划线和#分别指出各关系模式的主键和外键。

a) 部门号，职位

b) 申请号，客房类型，客户号

c) 申请号，员工号

d) 申请号，员工号

关系模式为：

会议申请(申请号，客房类型，客户号#，开会日期，会议地点，持续天数，会议人数，预算费用，会议类型，酒店要求，会议室要求，客房数，联系人，联系方式，受理标志，员工号#)

策划任务(申请号#，员工号#，策划内容，参与人数，要求完成时间)

执行策划(申请号#，员工号#，实际完成时间)

解析：根据题意，在员工关系模式中，部门与员工之间是一个 1:n 的联系，需要将 1 端（即部门）的码“部门号”并入员工关系；又因为每个员工担任相应职位，故员工关系模式欢迎添加“职位”属性；可见，空 (a) 应填写“部门号，职位”。

在会议申请关系模式中，由于申请号、客房类型、客户号为主键，故空 (b) 应填写“申请号，客房类型，客户号”；在策划任务关系模式中，申请号、员工号为主键，故空 (c) 应填写“申请号，客户号”；由于一个策划任务可由多名策划员参与执行，且一名策划员可以参与多项策划任务，故在执行策划关系模式中，执行策划又由于一个业务员可以安排多个托运申请，申请号、员工号为主键，故空 (d) 应填写“申请号，客户号”。

会议申请关系模式的主键为“申请号，客房类型”，因为，申请号、客房类型能唯一标识该关系模式的每一个元组。会议申请关系模式的外键为客户号及员工号，因为，客户号及员工号分别为客户及员工关系模式的主键，故为该关系模式的外键。

策划任务关系模式的主键为申请号，因为，申请号能唯一标识该关系模式的每一个元组，故申请号为该关系模式的主键。策划任务关系模式的外键为员工号，因为，员工号为员工关系模式的主键，故为该关系模式的外键。

执行策划关系模式的主键为“申请号，员工号”，因为，申请号 k 员工号能唯一标识该关系模式的每一个元组，故“申请号，员工号”为该关系模式的主键。执行策划关系模式的外键为申请号及员工号，因为，申请号和员工号分别为会议申请和员工关系模式的主键，故为该关系模式的外键。

### 【问题 3】



请说明关系模式“会议申请”存在的问题及解决方案。

会议申请存在数据冗余及数据修改的不一致性问题，应该将关系模式分解为如下两个模式：

会议申请 1（申请号，客户号，开会日期，会议地点，持续天数，会议人数，预算费用，会议类型，酒店要求，会议室要求，联系人，联系方式，受理标志，员工号）

会议申请 2（申请号，客房类型，客房数）。

关系模式“会议申请”存在数据冗余及数据修改的不一致性问题，应该将关系模式分解。

试题三

某城市的各国家公园周边建造了许多供游客租用的小木屋和营地，为此，该城市设置了一个中心售票处和若干个区域售票处。游客若想租用小木屋或营地，必须前往中心售票处进行预定并用现金支付全额费用。所有的预定操作全部由售票处的工作人员手工完成。现欲开发一信息系统，实现小木屋和营地的预定及管理功能，以取代手工操作。该系统的主要功能描述如下：

- 1. 管理预定申请。游客可以前往任何一个售票处提出预定申请。系统对来自各个售票处的预定申请进行统一管理。
- 2. 预定。预定操作包含登记游客预定信息、计算租赁费用、付费等步骤。
- 3. 支付管理。游客付费时可以选择现金和信用卡付款两种方式。使用信用卡支付可以享受 3%的折扣，现金支付没有折扣。
- 4. 游客取消预定。预定成功之后，游客可以在任何时间取消预定，但需支付赔偿金，剩余部分则退还给游客。赔偿金的计算规则是，在预定入住时间之前的 48 小时内取消，支付租赁费用 10%的赔偿金；在预定入住时间之后取消，则支付租赁费用 50%的赔偿金。
- 5. 自动取消预定。如果遇到恶劣天气（如暴雨、山洪等），系统会自动取消所有的预定，发布取消预定消息，全额退款。
- 6. 信息查询。售票处工作人员查询小木屋和营地的预定情况和使用情况，以判断是否能够批准游客的预定申请。

现采用面向对象方法开发上述系统，得到如表 3-1 所示的用例列表和表 3-2 所示的类列表。对应的用例图和类图分别如图 3-1 和 3-2 所示。

表 3-1 用例列表

用 例 名	说 明	用 例 名	说 明
ManageInquiries	管理预定申请	ManageCashPayment	现金支付
MakeReservation	预定	ManageCrCardPayment	信用卡支付
ManagePayment	支付管理	GetDiscount	计算付款折扣
CancelReservation	游客取消预定	AutoCancelReservation	系统自动取消预定
CheckAvailability	信息查询	CalculateRefund	计算取消预定的赔偿金
PublishMessage	发布取消预定消息		

表 3-2 类列表

类 名	说 明	类 名	说 明
NationalPark	国家公园	Customer	游客
Reservation	预定申请	ReservationItem	预定申请内容
TicketingOfficer	售票处	CampSite	营地
Bungalow	小木屋	Payment	付款
Discount	付款折扣	CashPayment	现金支付
CreditCardPayment	信用卡支付	Rate	租赁费用

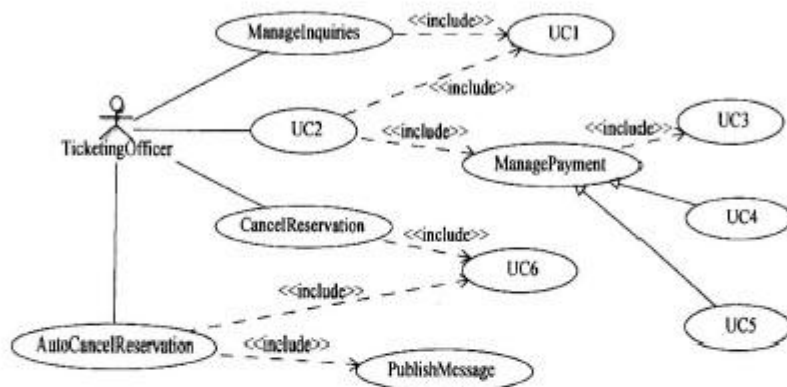


图 3-1 用例图

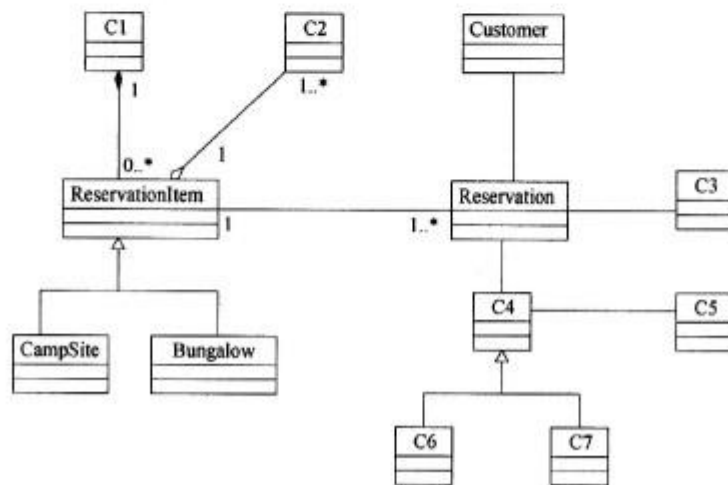


图 3-2 类图

## 【问题 1】

根据说明中的描述与表 3-1，给出图 3-1 中 UC1~UC6 处所对应的用例名称。

UC1 : CheckAvailablity

UC3 : GetDiscount

UC5 : ManageCrCardPayment

UC2: MakeReservation

UC4 : ManageCashPayment

UC6 : CalculateRefund

UC4 和 UC5 可以互换。

本题要求将图 3-1 所给出的用例图补充完整。题目说明中已经给出了所有可能的用例的列表（如表 3-1 所示）。这就省去了寻找用例的步骤，只需要依据用例列表中给出的用例，在说明中确定用例与 Actor 之间的关系即可将图补充完整。

用例图的构成要素有：参与者（Actor）、用例（Usecase）以及用例之间的关系。题目中的信息系统的主要用户是售票处的工作人员（TicketingOfficer），所以在图 3-1 中只给出了 1 个参与者。由说明可知，售票处工作人员利用该系统可以实现 6 个主要的功能：管理预定申请（ManageInquiries）、预定（MakeReservation）、支付管理（ManagePayment）、游客取消预定（CancelReservation）、自动取消预定（AutoCancelReservation）和信息查询（CheckAvailability）。其中“管理预定申请”、“支付管理”、“游客取消预定”、“自动取消预定”和“支付管理”均已经出现在图 3-1 中。支付租赁费用是预定过程中的一个必要步骤，而 UC2 和“支付管理”之间又是“include”关系，可以推断出 UC2 应该对应用例“预定（MakeReservation）”。那么用例“管理预定申请”和“预定”具有的相同步骤就是 UC1 所对应的用例，由此推断出 UC1 对应用例“信息查询（CheckAvailability）”。

由功能“支付管理”的说明可知，它具备两个能力：管理支付方式（信用卡或现金）以及计算折扣。UC4 和 UC5 与用例“支付管理”之间是概括关系，说明 UC4 和 UC5 是支付方式两个特化，所以 UC4 为“现金支付（MangeCashPayment）”，UC5 为“信用卡支付（ManageCrCardPayment）”。UC3 对应“计算折扣（GetDiscount）”。

这时用例列表中只剩下用例 CalcuateRefimd（计算取消预定的赔偿金）没有出现在图中了，那么它就是 UC6 对应的用例。从图 3-1 来看，UC6 应该表示用例“游客取消预定（CancelReservation）”和“自动取消预定（AutoCancelReservation）”中包含的公共事件流。不管是哪种类型的取消预定，都需要计算赔偿金，以决定退还给用户的费用，所以 UC6 对应用例 CalcuateRefimd。

## 【问题 2】

根据说明中的描述与表 3-2, 给出图 3-2 中 C1~C7 处所对应的类名。

C1:NationalPark . C2:Rate . C3:TicketingOfficer .C4:Payment . C5:Discount .  
C6;CashPaymen. C7 :CreditCardPayment C6 和 C7 可以互换。

本题考察的是类图建模。题目中已经给出了类的列表，要求考生根据说明指出每个类在类图中的位置。在解题时，可以同时参考用例图中给出的信息。

先整体地看一下类图，寻找其中是否包含继承、聚集或组装等这些层次结构，这是快速确定部分类的关键。在图 3-2 中有一个继承结构：C4、C6 和 C7。在图 3-1 中，用例之间也有一个概括的关系，这就提示我们，C4、C6 和 C7 这 3 个类一定与支付功能相关。在表 3-2 中寻找与支付功能相关的类：Payment、CashPayment 和 CreditCardPayment。下来就是确定这 3 个类中，哪个是父类。很明显，Payment 应该作为父类。因此 C4 对应 Payment，C6 对应 CashPayment，C7 对应 CreditCardPayment（C6 和 C7 可以互换支付管理中还有一项计算折扣的能力，类列表中的类 Discount 表示付款折扣，而与 C5 与 C4 之间具有关联关系，所以 C5 应该对应类 Discount。

C1、C2 分别与类“ReservationItem”之间具有组装和聚集的关系，而从说明中可知，具有这种整体部分关系的只有公园、预定及租赁费用之间，所以 C1 对应 NationalPark，C2 对应 Rate。最后的一个类 C3 对应 TicketingOfficer，即用例图中的 Actor。

### 【问题 3】

对于某些需求量非常大的小木屋或营地，说明中功能 4 的赔偿金计算规则，不足以弥补取消预定所带来的损失。如果要根据预定的时段以及所预定场地的需求量，设计不同层次的赔偿金计算规则，需要对图 3-2 进行怎样的修改？（请用文字说明）

解答 1:增加一个新的类，该类与类 ReservationItem 之间有关联关系。或解答 2:修改 Rate 类，使其具有计算赔偿金的功能。

在面向对象方法中，好的类模型对需求的变化应该具有一定的适应性。本题考察的就是这一点。根据题目，现在对原有的赔偿金计算规则要进行修正。除了考虑取消预定的时间之外，同时要考虑所预定的小木屋或营地的地段以及需求量。修正类模型时通常两种基本方式，一种是修改已有的类，使其适应新的需求；第二种是增加一个新的类来完成新的需求，但是需要同时考虑新增加的类与已有类之间的关系。这道题目两种修改方法都可以采用。

若要修改已有的类，需要首先了解哪个类与现在的新需求是有相关性的。新需求针对的是赔偿金，赔偿金又与租赁费用相关，所以要找原先与租赁费用相关的那个类，即 Rate。解决方案之一就是修改 Rate，使其能够按照新的规则计算赔偿金。

第二种修改方式，增加一个专门计算赔偿金的类。按照新的计算规则，这个类就与游客的每

次预定内容相关，因此这个新增加的类应该与类 ReservationItem 之间有关联关系。

#### 试题四

设有  $n$  个货物要装入若干个容量为  $C$  的集装箱以便运输, 这  $n$  个货物的体积分别为  $\{s_1, s_2, \dots, s_n\}$ , 且有  $s_i < C (1 \leq i \leq n)$ 。为节省运输成本, 用尽可能少的集装箱来装运这  $n$  个货物。

下面分别采用最先适宜策略和最优适宜策略来求解该问题。

最先适宜策略 (firstfit) 首先将所有的集装箱初始化为空, 对于所有货物, 按照所给的次序, 每次将一个货物装入第一个能容纳它的集装箱中。

最优适宜策略 (bestfit) 与最先适宜策略类似, 不同的是, 总是把货物装到能容纳它且目前剩余容量最小的集装箱, 使得该箱子装入货物后闲置空间最小。

#### 【C 代码】

下面是这两个算法的 C 语言核心代码。

(1) 变量说明

$n$ : 货物数

$C$ : 集装箱容量

$s$ : 数组, 长度为  $n$ , 其中每个元素表示货物的体积, 下标从 0 开始

$b$ : 数组, 长度为  $n$ ,  $b[i]$  表示第  $i+1$  个集装箱当前已经装入货物的体积, 下标从 0 开始

$i, j$ : 循环变量

$k$ : 所需的集装箱数

$\min$ : 当前所用的各集装箱装入了第  $i$  个货物后的最小剩余容量

$m$ : 当前所需要的集装箱数

$\text{temp}$ : 临时变量

(2) 函数 firstfit

```
int firstfit() {
    int i, j;
    k = 0;
    for(i = 0; i < n; i++) {
```

```

for(i = 0; i < n; i++){
    (1);
    while(C - b[j] < s[i]){
        j++;
    }
    (2);
    k = k > (j+1) ? k : (j+1) ;
}
return k;
}

```

### (3) 函数 bestfit

```

int bestfit() {
    int i, j, min, m, temp;
    k = 0;
    for(i = 0; i < n; i++) {
        b[i] = 0;
    }
    for(i = 0; i < n; i++) {
        min = C;
        m = k + 1;
        for(j = 0; j < k + 1; j++) {
            temp = C - b[j] - s[i];
            if(temp > 0 && temp < min){
                (3);
                m = j;
            }
        }
        (4);
        k = k > (m+1) ? k : (m+1) ;
    }
    return k;
}

```



```

for(i = 0; i < n; i++){
    (1);
    while(C - b[j] < s[i]){
        j++;
    }
    (2);
    k = k > (j+1) ? k : (j+1) ;
}
return k;
}

```

### (3) 函数 bestfit

```

int bestfit() {
    int i, j, min, m, temp;
    k = 0;
    for(i = 0; i < n; i++) {
        b[i] = 0;
    }
    for(i = 0; i < n; i++) {
        min = C;
        m = k + 1;
        for(j = 0; j < k + 1; j++) {
            temp = C - b[j] - s[i];
            if(temp > 0 && temp < min){
                (3);
                m = j;
            }
        }
        (4);
        k = k > (m+1) ? k : (m+1) ;
    }
    return k;
}

```

### 【问题 1】

根据【说明】和【C 代码】，填充 C 代码中的空 (1)～(4)。

(1)  $j = 0$

(2)  $b[j] = b[j] + s[i]$  及其等价形式

(3)  $\min = \text{temp}$

(4)  $b[m] = b[m] + s[i]$  及其等价形式

根据最先适宜算法思想，每取出一个货物，从第一个集装箱开始判断该货物是否能放入集装箱，若能则放入，因此空 (1) 填  $j = 0$ 。while 循环判断，若货物不能放入集装箱，则考虑下一个集装箱。不满足 while 循环中的条件，说明货物能放入集装箱，因此空 (2) 填  $b[j] =$

$b[j] + S[i]$ 。根据最优适宜算法思想，每取出一个货物，从第一个集装箱开始，确定能放入该货物且剩余容量最小的集装箱，并把该货物放入该集装箱中。if 条件判断，若找到了比能放入货物且剩余容量更小的集装箱，则剩余容量最小值改为当前的集装箱的剩余容量，因此空(3)填  $min = temp$ 。确定了集装箱后，把货物装入集装箱中，空(4)  $b[m] = b[m] + s[i]$ 。

### 【问题 2】

根据【说明】和【C 代码】，该问题在最先适宜和最优适宜策略下分别采用了 (5) 和 (6) 算法设计策略，时间复杂度分别为 (7) 和 (8) (用 0 符号表示)。

(5) 贪心 (6) 贪心 (7)  $O(n^2)$  (8)  $O(n^2)$

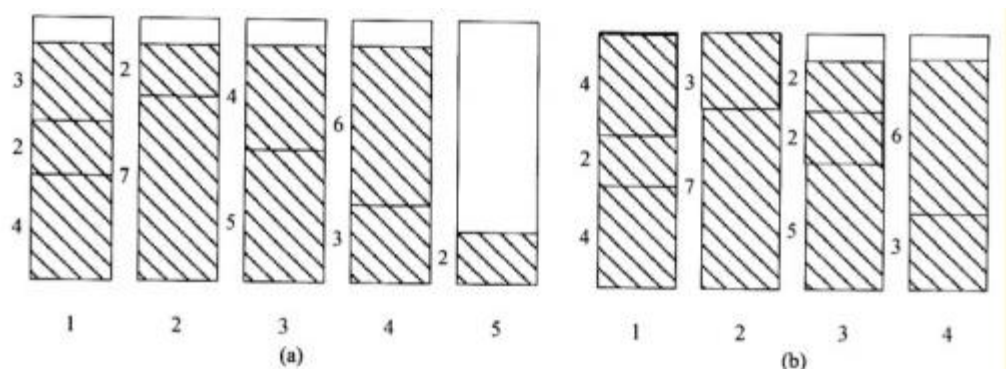
最先适宜算法总是把货物放入第一个能放入的集装箱，最优适宜算法总是把货物放入能容纳该货物且剩余容量最小的集装箱，因此都是基于贪心策略进行的，空(5)和(6)填贪心。函数 `firstfit` 中的 for 循环考虑  $n$  个货物，其中嵌套了 while 循环，最多的集装箱数为  $n$ ，因此时间复杂度为  $O(n^2)$ 。函数 `bestfit` 中的 for 循环考虑  $n$  个货物，其中嵌套了 for 循环检查每个集装箱的剩余容量，最多的集装箱数为  $n$ ，因此时间复杂度为  $O(n^2)$ 。

### 【问题 3】

考虑实例  $n = 10$ ,  $C = 10$ , 各个货物的体积为  $\{4, 2, 7, 3, 5, 4, 2, 3, 6, 2\}$ 。该实例在最先适宜和最优适宜策略下所需的集装箱数分别为 (9) 和 (10)。考虑一般的情况，这两种求解策略能否确保得到最优解？ (11) (能或否)

(9) 5 (10) 4 (11) 否

对实例  $n = 10$ ,  $C = 10$ ,  $S = \{4, 2, 7, 3, 5, 4, 2, 3, 6, 2\}$ , 根据最先适宜和最优适宜算法，其具体的装箱方案分别如下图(a)和(b)所示。



因此最先适宜和最优适宜方法所需的集装箱数分别为 5 和 4, 装箱问题是一个非常难的问题，

这两种贪心策略不能确保得到最优解，即最少的装箱数。

## 试题五

现欲开发一个软件系统，要求能够同时支持多种不同的数据库，为此采用抽象工厂模式设计该系统。以 SQL Server 和 Access 两种数据库以及系统中的数据库表 Department 为例，其类图如图 5-1 所示。

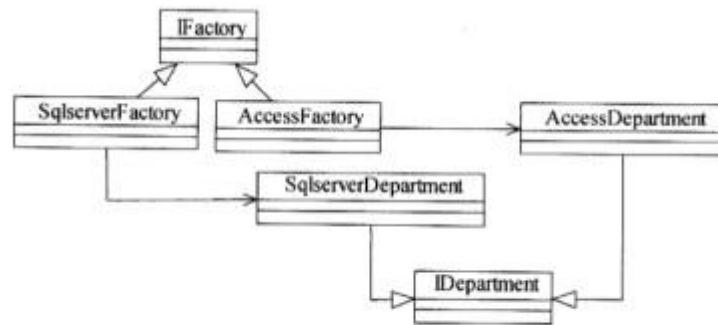


图 6-1 类图

### 【问题 1】

#### 【C++代码】

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

class Department { /* 代码省略 */ };
class IDepartment {
public:
```

```

        (1) = 0;
        (2) = 0;
};

class SqlserverDepartment : (3) {
public:
    void Insert(Department* department) {
        cout << "Insert a record into Department in SQL Server!\n";
        // 其余代码省略
    }
    Department GetDepartment(int id) {
        /* 代码省略 */
    }
};

class AccessDepartment : (4) {
public:
    void Insert(Department* department) {
        cout << "Insert a record into Department in ACCESS!\n";
        // 其余代码省略
    }
    Department GetDepartment(int id) {
        /* 代码省略 */
    }
};

(5) {
public:
    (6) = 0;
};

class SqlServerFactory : public IFactory {
public:
    IDepartment* CreateDepartment() { return new SqlserverDepartment(); }
};

class SqlServerFactory : public IFactory {
public:
    IDepartment* CreateDepartment() { return new SqlserverDepartment(); }
    // 其余代码省略
};

class AccessFactory : public IFactory {
public:
    IDepartment* CreateDepartment() { return new AccessDepartment(); }
    // 其余代码省略
};

```

1、virtual void Insert(Department\* department)

2、virtual Department GetDepartment(int id)

3、public IDepartment

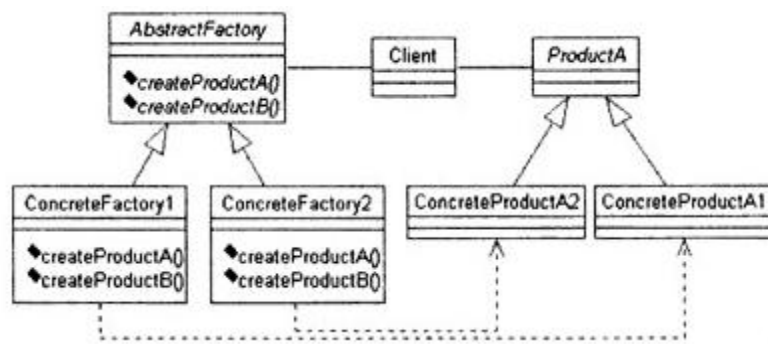
4、public IDepartment

5、class IFactory

6、virtual IDepartment\* CreateDepartment()

本题考查抽象工厂（Abstract Factory）模式的概念及应用。

Abstract Factory 模式的意图是，提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。Abstract Factory 模式的结构如下图所示。



其中，类 AbstractFactory 声明一个创建抽象产品对象的操作接口；类 ConcreteFactory 实现创建具体产品对象的操作；类 productA 为一类产品对象声明一个接口；类 ConcreteProduct 具有 2 个功能：定义一个将被相应的具体工厂创建的产品对象；实现 ProductA 接口；类 Client 仅使用由 AbstractFactory 和 AbstractProduct 类声明的接口。在以下情况可以使用 AbstractFactory 模式：

- (1) 一个系统要独立于它的产品的创建、组合和表示时；
- (2) 一个系统要由多个产品系统中的一个来配置时；
- (3) 当要强调一系列相关的产品对象的设计一边进行联合使用时；
- (4) 提供一个产品类库，而只想显示它们的接口而不是实现时。

题目利用抽象工厂模式来解决在同一个软件系统中支持多种不同数据库的问题，这也是软件开发中比较常见的情形。其中的类 IFactory 相当于上图中的类 AbstractFactory；类 IDepartment 相当于上图中的类 ProductA。本题中只给出了一个产品类。

下面来分析程序。

第 (1)、(2) 空出现类 IDepartment 的定义中。类 IDepartment 的作用是为一类产品对象声明一个接口，在 C++ 中通常都采用抽象类来定义这种抽象操作接口。C++ 中的抽象类是包含

了至少一个纯虚拟函数的类。纯虚拟函数的语法是：`virtual <函数名>(<参数列表>)=0;`；在程序中已经出现了纯虚拟函数的标志“=0”，因此（1）、（2）空应该都是纯虚拟函数。下面来确定纯虚拟函数的原型。这需要去考察类 `IDepartment` 的子类，因此纯虚拟函数是在父类中定义，在子类中实现。由类图 5-1 可知，类 `IDepartment` 的子类分别是 `SqlserverDepartment` 和 `AccessDepartment`。至此可以提前确定（3）、（4）空的内容了，即其所对应的父类。因此（3）、（4）空都应该填写 `public IDepartment`。我们看类 `SqlserverDepartment` 中的方法，分别为 `:Insert` 和 `GetDepartment`，而在类 `AccessDepartment` 中也出现了这两个方法，且接口完全一致。所以这两个方法就应该是类 `IDepartment` 中所定义的抽象接口。由此可知，（1）空应该填写 `virtual void Insert(Department* department)`，（2）空应该填写 `virtual Department GetDepartment(int id)`。

空（5）和（6）分别缺失在类的名称以及该类中的方法。由图 5-1 和代码可知，缺少类 `IFactory` 的定义，所以（5）空处应该填写 `class IFactory`。那么类 `IFactory` 应包含的方法是什么？类 `IFactory` 的作用是声明一个创建抽象产品对象的操作接口，这个接口一定会同时出现在 `IFactory` 的子类 `SqlServerFactory` 和 `AccessFactory` 中，即 `CreateDepartment`。（6）处同样应该是一个纯虚拟函数，所以（6）空处应该填写 `virtual IDepartment* CreateDepartment()`。

试题六

现欲开发一个软件系统，要求能够同时支持多种不同的数据库，为此采用抽象工厂模式设计该系统。以 SQL Server 和 Access 两种数据库以及系统中的数据库表 Department 为例，其类图如图 6-1 所示。

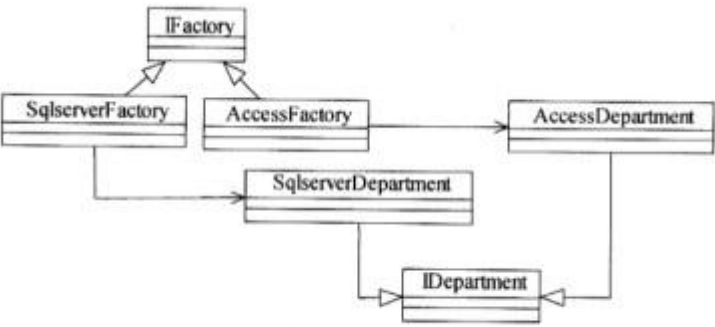


图 6-1 类图

【问题 1】

```
class Department { /* 代码省略 */ }

interface IDepartment {
    (1) _____;
    (2) _____;
}

class SqlserverDepartment (3) _____ {
    public void Insert(Department department) {
        System.out.println("Insert a record into Department in SQL Server!");
        // 其余代码省略
    }
    public Department GetDepartment(int id) {
        /* 代码省略 */
    }
}

class AccessDepartment (4) _____ {
    public void Insert(Department department) {
        System.out.println("Insert a record into Department in ACCESS!");
        // 其余代码省略
    }
}
```



```

    }
    public Department GetDepartment(int id) {
/* 代码省略 */
    }
}

    (5)
    (6);
}

class SqlServerFactory implements IFactory {
    public IDepartment CreateDepartment() {
        return new SqlserverDepartment();
    }
    // 其余代码省略
}

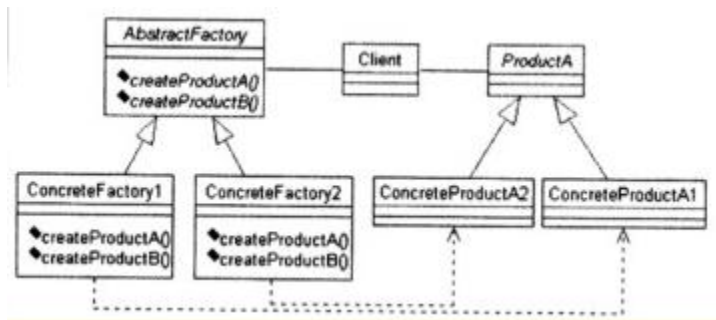
class AccessFactory implements IFactory {
    public IDepartment CreateDepartment() {
        return new AccessDepartment();
    }
    // 其余代码省略
}

```

1. void Insert(Department department)
2. Department GetDepartment(int id)
3. implements IDepartment
4. implements IDepartment
5. interface IFactory
6. IDepartmentCreateDepartment()

本题考查抽象工厂（Abstract Factory）模式的概念及应用。

Abstract Factory 模式的意图是，提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。Abstract Factor 模式的结构如下图所示。



其中，类 AbstractFactory 声明一个创建抽象产品对象的操作接口；类 ConcreteFactory 实现创建具体产品对象的操作；类 ProductA 为一类产品对象声明一个接口；类 ConcreteProduct 具有 2 个功能：定义一个将被相应的具体工厂创建的产品对象；实现 ProductA 接口；类 Client 仅使用由 AbstractFactory 和 AbstractProduct 类声明的接口。在以下情况可以使用 Abstract Factory 模式：

- (1) 一个系统要独立于它的产品的创建、组合和表示时；
- (2) 一个系统要由多个产品系统中的一个来配置时；
- (3) 当要强调一系列相关的产品对象的设计一边进行联合使用时；
- (4) 提供一个产品类库，而只想显示它们的接口而不是实现时。

题目利用抽象工厂模式来解决在同一个软件系统中支持多种不同数据库的问题，这也是软件开发中比较常见的情形。其中的类 IFactory 相当于上图中的类 AbstractFactory；类 IDepartment 相当于上图中的类 ProdcutA。本题中只给出了一个产品类。

下面来分析程序。

第 (1)、(2) 空出现类 IDepartment 的定义中。类 IDepartment 的作用是为一类产品对象声明一个接口，在 Java 中通常都采用 Inteface 来定义这种抽象操作接口。要确定接口的原型，需要去考察实现接口 IDepartment 的类。由类图 6-1 可知，实现接口 IDepartment 的类分别是 SqlserverDepartment 和 AccessDepartment。至此可以提前确定 (3)、(4) 空的内容了。

(3)、(4) 空都应该填写 implements IDepartment。我们看类 SqlserverDepartment 中的方法，分别为：Insert 和 GetDepartment，而在类 AccessDepartment 中也出现了这两个方法，且接口完全一致。所以这两个方法就应该是 IDepartment 中所定义的抽象接口。由此可知，(1) 空应该填写 void Insert(Department department)，(2) 空应该填写 Department GetDepartment(int id)。

空(5)和(6)分别缺失在类的名称以及该类中的方法。由图 6-1 和代码可知，缺少类 IFactory 的定义。由代码 “class SqlServerFactory implements IFactory” 和 “class

“AccessFactory implements IFactory” 可知，IFactory 也是一个接口。所以 (5) 空处应该填写 Interface IFactory。那么类 IFactory 应包含的方法是什么？类 IFactory 的作用是声明一个创建抽象产品对象的操作接口，这个接口一定会同时出现在类 SqlServerFactory 和 AccessFactory 中，即 CreateDepartment。(6) 处同样应该是一个纯虚拟函数，所以 (6) 空处应该填写 IDepartment CreateDepartment()。