

海明校验码是在  $n$  个数据位之外增设  $k$  个校验位,从而形成一个  $k+n$  位的新的码字,使新的码字的码距比较均匀地拉大。 $n$  与  $k$  的关系是(1)。

- (1) A.  $2k - 1 \geq n + k$       B.  $2n - 1 \leq n + k$       C.  $n = k$       D.  $n-1 \leq k$

【答案】A

【解析】

本题考查校验码方面的基础知识。

海明码是一种多重(复式)奇偶检错编码。它将信息用逻辑形式编码,以便能够检错和纠错。用在海明码中的全部传输码字是由原来的信息和附加的奇偶校验位组成的。每一个这种奇偶位被编在传输码字的特定位置上。推导并使用长度为  $n$  位的码字的海明码,所需步骤如下:

(1) 确定最小的校验位数  $k$ , 将它们记成  $D_1$ 、 $D_2$ 、 $\dots$ 、 $D_k$ , 每个校验位符合不同的奇偶测试规定。

(2) 原有信息和  $k$  个校验位一起编成长为  $n+k$  位的新码字。选择  $k$  校验位(0 或 1)以满足必要的奇偶条件。

(3) 对所接收的信息作所需的  $k$  个奇偶检查。

(4) 如果所有的奇偶检查结果均正确,则认为信息无错误。如果发现有一个或多个错了,则错误的位由这些检查的结果来唯一地确定。

求海明码时的一项基本考虑是确定所需最少的校验位数  $k$ 。考虑长度为  $n$  位的信息,若附加了  $k$  个校验位,则所发送的总长度为  $n+k$ 。在接收器中要进行  $k$  个奇偶检查,每个检查结果或是真或是假。这个奇偶检查的结果可以表示成一个  $k$  位的二进制,它可以确定最多  $2^k$  种不同状态。这些状态中必有一个其所有奇偶测试都是真的,它便是判定信息正确的条件。于是剩下的  $(2^k-1)$  种状态,可以用来判定误码的位置。于是导出以下关系:

$$2^k - 1 \geq n + k$$

假设某硬盘由 5 个盘片构成(共有 8 个记录面),盘面有效记录区域的外直径为 30cm,内直径为 10cm,记录位密度为 250 位/mm,磁道密度为 16 道/mm,每磁道分 16 个扇区,每扇区 512 字节,则该硬盘的格式化容量约为(2)MB。

- (2) A.  $\frac{8 \times (30-10) \times 10 \times 250 \times 16}{8 \times 1024 \times 1024}$       B.  $\frac{8 \times (30-10) \times 10 \times 16 \times 16 \times 512}{2 \times 1024 \times 1024}$
- C.  $\frac{8 \times (30-10) \times 10 \times 250 \times 16 \times 16}{8 \times 1024 \times 1024}$       D.  $\frac{8 \times (30-10) \times 16 \times 16 \times 512}{2 \times 1024 \times 1024}$

【答案】B

【解析】

本题考查计算机系统硬件方面磁盘容量的计算。

硬盘容量分为非格式化容量和格式化容量两种，计算公式如下：

非格式化容量=面数×(磁道数/面)×内圆周长×最大位密度

格式化容量=面数×(磁道数/面)×(扇区数/道)×(字节数/扇区)

题目中给出硬盘的面数为 8，每面的磁道数为  $(30-10) \times 10 \div 2 \times 16$ ，每磁道扇区数为 16，每扇区 512 字节，因此其格式化容量为

$$\frac{8 \times (30-10) \times 10 \times 16 \times 16 \times 512}{2} \text{ B}$$

换算成 MB 单位时再除以  $1024 \times 1024$ 。

(3)是指按内容访问的存储器。

- (3) A. 虚拟存储器      B. 相联存储器      C. 高速缓存 (Cache)      D. 随机访问存储器

【答案】B

【解析】本题考查计算机系统存储器方面的基础知识。

计算机系统的存储器按所处的位置可分为内存和外存。按构成存储器的材料可分为磁存储器、半导体存储器和光存储器。按存储器的工作方式可分为读写存储器和只读存储器。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。按寻址方式可分为随机存储器、顺序存储器和直接存储器。

相联存储器是一种按内容访问的存储器。

处理机主要由处理器、存储器和总线组成，总线包括(4)

- (4) A. 数据总线、地址总线、控制总线      B. 并行总线、串行总线、逻辑总线  
C. 单工总线、双工总线、外部总线      D. 逻辑总线、物理总线、内部总线

【答案】A

【解析】本题考查计算机系统总线和接口方面的基础知识。

广义地讲，任何连接两个以上电子元器件的导线都可以称为总线。通常可分为 4 类：

- ①芯片内总线。用于在集成电路芯片内部各部分的连接。
- ②元件级总线。用于一块电路板内各元器件的连接。
- ③内总线，又称系统总线。用于构成计算机各组成部分 (CPU、内存和接口等) 的连接。
- ④外总线，又称通信总线。用计算机与外设或计算机与计算机的连接或通信。

连接处理机的处理器、存储器及其他部件的总线属于内总线，按总线上所传送的内容分为数据总线、地址总线和控制总线。

计算机中常采用原码、反码、补码和移码表示数据，其中， $\pm 0$  编码相同的是 (5)。

- (5) A. 原码和补码      B. 反码和补码      C. 补码和移码      D. 原码和移码

【答案】C

【解析】

本题考查计算机系统数据编码基础知识。

设机器字长为  $n$  (即采用  $n$  个二进制位表示数据)，最高位是符号位，0 表示正号，1 表示负号。

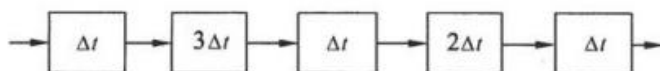
原码表示方式下，除符号位外， $n-1$  位表示数值的绝对值。因此， $n$  为 8 时， $[+0]_{\text{原}} = 0\ 0000000$ ， $[-0]_{\text{原}} = 1\ 0000000$ 。

正数的反码与原码相同，负数的反码则是其绝对值按位求反。 $n$  为 8 时，数值 0 的反码表示有两种形式： $[+0]_{\text{反}} = 0\ 0000000$ ， $[-0]_{\text{反}} = 11111111$ 。

正数的补码与其原码和反码相同，负数的补码则等于其反码的末尾加 1。在补码表示中，0 有唯一的编码： $[+0]_{\text{补}} = 0\ 0000000$ ， $[-0]_{\text{补}} = 00000000$ 。

移码表示法是在数  $X$  上增加一个偏移量来定义的，常用于表示浮点数中的阶码。机器字长为  $n$  时，在偏移量为  $2^{n-1}$  的情况下，只要将补码的符号位取反便可获得相应的移码表示。

某指令流水线由 5 段组成，第 1、3、5 段所需时间为  $\Delta t$ ，第 2、4 段所需时间分别为  $3\Delta t$ 、 $2\Delta t$ ，如下图所示，那么连续输入  $n$  条指令时的吞吐率 (单位时间内执行的指令个数)  $TP$  为 (6)。



- (6) A.  $\frac{n}{5 \cdot (3+2)\Delta t}$       B.  $\frac{n}{(3+3+2)\Delta t + 3(n-1)\Delta t}$
- C.  $\frac{n}{(3+2)\Delta t + (n-3)\Delta t}$       D.  $\frac{n}{(3+2)\Delta t + 5 \cdot 3\Delta t}$

【答案】B

【解析】

本题考查计算机系统流水线方面的基础知识。

吞吐率和建立时间是使用流水线技术的两个重要指标。吞吐率是指单位时间里流水线处理机流出的结果数。对指令而言，就是单位时间里执行的指令数。流水线开始工作时，需经过一定时间才能达到最大吞吐率，这就是建立时间。若  $m$  个子过程所用时间一样，均为  $\Delta t_0$ ，则建立时间  $T_0 = m\Delta t_0$ 。

本题目中，连续输入  $n$  条指令时，第 1 条指令需要的时间为  $(1+3+1+2+1)\Delta t$ ，之后，每隔  $3\Delta t$  便完成 1 条指令，即流水线一旦建立好，其吞吐率为最长子过程所需时间的倒数。综合  $n$  条指令的时间为  $(1+3+1+2+1)\Delta t + (n-1) \times 3\Delta t$ ，因此吞吐率为

$$\frac{n}{(3+3+2)\Delta t + 3(n-1)\Delta t}$$

下面关于漏洞扫描系统的叙述，错误的是 (7)。

- (7) A. 漏洞扫描系统是一种自动检测目标主机安全弱点的程序  
B. 黑客利用漏洞扫描系统可以发现目标主机的安全漏洞  
C. 漏洞扫描系统可以用于发现网络入侵者  
D. 漏洞扫描系统的实现依赖于系统漏洞库的完善

【答案】C

【解析】本题考查漏洞扫描系统的基本概念。

漏洞扫描系统是一种自动检测目标主机安全弱点的程序，漏洞扫描系统的原理是根据系统漏洞库对系统可能存在的漏洞进行一一验证。黑客利用漏洞扫描系统可以发现目标主机的安全漏洞从而有针对性的对系统发起攻击；系统管理员利用漏洞扫描系统可以查找系统中存在的漏洞并进行修补从而提高系统的可靠性。漏洞扫描系统不能用于发现网络入侵者，用于检测网络入侵者的系统称为入侵检测系统。

网络安全包含了网络信息的可用性、保密性、完整性和网络通信对象的真实性。其中 (8) 数字签名是对一一的保护。

- (8) A. 可用性      B. 保密性      C. 连通性      D. 真实性

【答案】D

【解析】本题考查网络安全方面的基础知识。

数字签名 (Digital Signature) 技术是不对称加密算法的典型应用。数字签名的应用过程是：数据源发送方使用自己的私钥对数据校验和或其他与数据内容有关的变量进行加密处理，完成对数据的合法“签名”；数据接收方则利用对方的公钥来解读收到的“数字签名”，

并将解读结果用于对数据完整性的检验，以确认签名的合法性。数字签名技术是在网络系统虚拟环境中确认身份的重要技术，完全可以代替现实过程中的“亲笔签字”，在技术和法律上有保证，可见数字签名是对签名真实性的保护。

计算机感染特洛伊木马后的典型现象是(9)

- (9)A. 程序异常退出
- B. 有未知程序试图建立网络连接
- C. 邮箱被垃圾邮件填满
- D. Windows 系统黑屏

**【答案】B**

**【解析】**本题考查计算机病毒相关知识。

特洛伊木马是一种通过网络传播的病毒，分为客户端和服务端两部分，服务端位于被感染的计算机，特洛伊木马服务端运行后会试图建立网络连接，所以计算机感染特洛伊木马后的典型现象是有未知程序试图建立网络连接。

关于软件著作权产生的时间，下面表述正确的是(10)。

- (10)A. 自作品首次公开发表时
- B. 自作者有创作意图时
- C. 自作品得到国家著作权行政管理部门认可时
- D. 自作品完成创作之日

**【答案】D**

**【解析】**本题考查知识产权中关于软件著作权方面的知识。

在我国，软件著作权采用“自动保护”原则。《计算机软件保护条例》第十四条规定：“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。

一般来讲，一个软件只有开发完成并固定下来才能享有软件著作权。如果一个软件一直处于开发状态中，其最终的形态并没有固定下来，则法律无法对其进行保护。因此，条例（法律）明确规定软件著作权自软件开发完成之日起产生。当然，现在的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。所以说，自该软件开发完成后就产生了著作权。

程序员甲与同事乙在乙家探讨甲近期编写的程序，甲表示对该程序极不满意，说要弃之重写，并将程序手稿扔到乙家垃圾筒。后来乙将甲这一程序稍加修改，并署乙名发表。以下说法正确的是 (11)。

- (11) A. 乙的行为侵犯了甲的软件著作权  
B. 乙的行为没有侵犯甲的软件著作权，因为甲已将程序手稿丢弃  
C. 乙的行为没有侵犯甲的著作权，因为乙已将程序修改  
D. 甲没有发表该程序并弃之，而乙将程序修改后发表，故乙应享有著作权

**【答案】A**

**【解析】** 本题考查知识产权中关于软件著作权方面的知识。

著作权因作品的完成而自动产生，不必履行任何形式的登记或注册手续，也不论其是否已经发表，所以甲对该软件作品享有著作权。乙未经甲的许可擅自使用甲的软件作品的行为，侵犯了甲的软件著作权。

PC 处理的音频信号主要是人耳能听得到的音频信号，它的频率范围是 (12)。

- (12) A. 300Hz~3400Hz      B. 20Hz~20kHz      C. 10Hz~20kHz      D. 20Hz~44kHz

**【答案】B**

**【解析】** 本题考查多媒体中关于音频信号方面的基础知识。

声音信号由许多频率不同的信号组成，通常称为复合信号，而把单一频率的信号称为分量信号。声音信号的一个重要参数就是带宽 (Bandwidth)，它用来描述组成声音的信号频率范围。

声音信号的频率是指声波每秒钟变化的次数，用 Hz 表示。人们把频率小于 20Hz 的声波信号称为亚音信号（也称次音信号）；频率范围为 20Hz~20kHz 的声波信号称为音频信号；高于 20kHz 的信号称为超音频信号（也称超声波）。

PC 处理的音频信号主要是人耳能听得到的音频信号 (audio)，它的频率范围是 20~20kHz。

可听声包括：

- 话音（也称语音）：人的说话声，频率范围通常为 300~3400Hz»
- 音乐：由乐器演奏形成（规范的符号化声音），其带宽可达到 20~20kHz。
- 其他声音：如风声、雨声、鸟叫声和汽车鸣笛声等，它们起着效果声或噪声的作用，其带宽范围也是 20~20kHz。



多媒体计算机图像文件格式分为静态图像文件格式和动态图像文件格式, (13)属于静态图像文件格式。

(13) A. MPG                      B. AVS                      C. JPG                      D. AVI

**【答案】C**

**【解析】**本题考查多媒体中关于文件格式方面的基础知识。

计算机中使用的图像文件格式大体上可分为图像文件格式和动态图像文件格式两大类, 每类又有很多种。JPEG 是由 ISO 和 IEC 两个组织机构联合组成的一个专家组, 负责制定静态和数字图像数据压缩编码标准, 这个专家组地区性的算法称为 JPEG 算法, 并且成为国际上通用的标准, 因此又称为 JPEG 标准。JPEG 是一个适用范围很广的静态图像数据压缩标准, 既可用于灰度图像又可用于彩色图像。MPEG 文件格式是运动图像压缩算法的国际标准, 它包括 MPEG 视频、MPEG 音频和 MPEG 系统(视频、音频同步)三个部分。MPEG 压缩标准是针对运动图像设计的, 其基本方法是: 单位时间内采集并保存第一帧信息, 然后只存储其余帧对第一帧发生变化的部分, 从而达到压缩的目的。MPEG 的平均压缩比为 50 : 1, 最高可达 200:1, 压缩效率非常高, 同时图像和音响的质量也非常好, 并且在 PC 上有统一的标准格式, 兼容性相当好。AVI 是 Microsoft 公司开发的一种符合 RIFF 文件规范的数字音频与视频文件格式, Windows、OS/2 等多数操作系统直接支持。AVI 格式允许视频和音频交错在一起同步播放, 支持 256 色和 RLE 压缩, 但 AVI 文件并未限定压缩标准。AVI 文件目前主要应用在中多媒体光盘上, 用来保存电影、电视等各种影像信息, 有时也出现在因特网上, 供用户下载、欣赏新影片的片段。

计算机获取模拟视频信息的过程中首先要进行 (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】**本题考查多媒体中关于模拟视频信息处理方面的基础知识。

模拟视频信号进入计算机时, 首先需要解决模拟视频信息的数字化问题。与音频数字化一样, 视频数字化的目的是将模拟信号经 A/D 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 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：

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。

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。

视频信息数字化的过程比声音复杂一些，它是以一幅幅彩色画面为单位进行的。分量数字化方式是使用较多的一种方式。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，因此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。

数字图像数据的数据量大，而数字视频信息的数据量就更加突出。例如，每帧 352X240 像素点，图像深度 16 位的图像，其数据量约为 1.3Mb，每秒 30 帧，其数据量就高达 40Mb/s，这样大的数据量无论是传输、存储还是处理，都是极大的负担。

在采用面向对象技术构建软件系统时，很多敏捷方法都建议的一种重要的设计活动是 (15)，它是一种重新组织的技术，可以简化构件的设计而无需改变其功能或行为。

- (15) A. 精化                      B. 设计类                      C. 重构                      D. 抽象

**【答案】C**

**【解析】** 本题考查采用敏捷方法进行软件开发。

敏捷方法中，重构是一种重新组织技术，重新审视需求和设计，重新明确地描述它们以符合新的和现有的需求，可以简化构件的设计而无需改变其功能或行为。

一个软件开发过程描述了“谁做”、“做什么”、“怎么做”和“什么时候做”，RUP 用 (16) 来表述“谁做”。

- (16) A. 角色                      B. 活动                      C. 制品                      D. 工作流

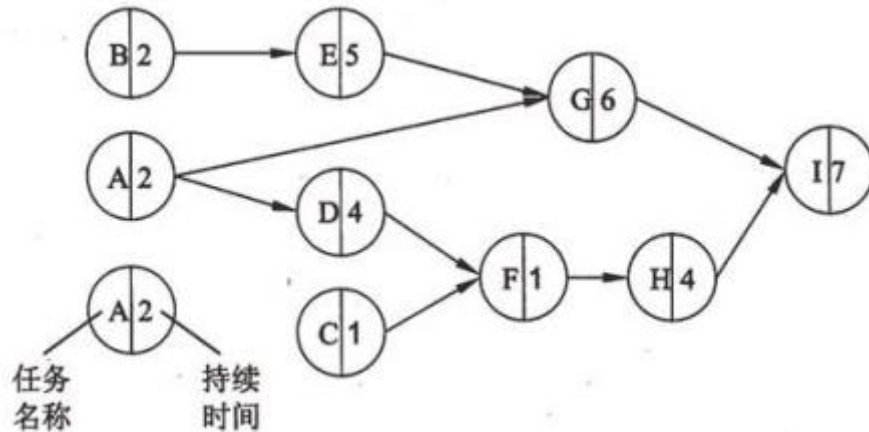
**【答案】A**



【解析】本题考查 RUP 对软件开发过程的描述。

RUP 应用了角色、活动、制品和工作流 4 种重要的模型元素，其中角色表述“谁做”，制品表述“做什么”，活动表述“怎么做”，工作流表述“什么时候做”。

某项目主要由 A~I 任务构成，其计划图（如下图所示）展示了各任务之间的前后关系以及每个任务的工期（单位：天），该项目的关键路径是 (17)。在不延误项目总工期的情况下，任务 A 最多可以推迟开始的时间是 (18) 天。



- (17) A. A→G→I    B. A→D→F→H→I    C. B→E→G→I    D. C→F→H→I
- (18) A. 0    B. 2    C. 5    D. 7

【答案】C B

【解析】本题考查项目计划的关键路径和松弛时间。

图中任务流 A→G→I 的持续时间为 15；任务流 A→D→F→H→I 的持续时间为 18；任务流 B→E→G→I 的持续时间为 20；任务流 C→F→H→I 的持续时间为 13。因此关键路径 B→E→G→I，其持续时间是 20。任务 A 处于任务流 A→G→I 和任务流 A→D→F→H→I 中，分别持续时间为 15 和 18，因此任务 A 的可延迟开始时间为 2。

软件风险一般包含 (19) 两个特性。

- (19) A. 救火和危机管理    B. 已知风险和未知风险
- C. 不确定性和损失    D. 员工和预算

【答案】C

【解析】本题考查软件风险的特性。

软件风险一般包括不确定性和损失两个特性，其中不确定性是指风险可能发生，也可能不发生；损失是当风险确实发生时，会引起的不希望的后果和损失。救火和危机管理是对不

适合但经常采用的软件风险管理策略。已知风险和未知风险是对软件风险进行分类的一种方式。员工和预算是在识别项目风险时需要识别的因素。

函数调用时，基本的参数传递方式有传值与传地址两种，(20)。

- (20)A. 在传值方式下，形参将值传给实参  
B. 在传值方式下，实参不能是数组元素  
C. 在传地址方式下，形参和实参间可以实现数据的双向传递  
D. 在传地址方式下，实参可以是任意的变量和表达式

**【答案】C**

**【解析】**本题考查程序语言基础知识。

函数调用时基本的参数传递方式有传值与传地址两种，在传值方式下是将实参的值传递给形参，因此实参可以是表达式（或常量），也可以是变量（或数组元素），这种信息传递是单方向的，形参不能再将值传回给实参。在传地址方式下，需要将实参的地址传递给形参，因此，实参必须是变量（数组名或数组元素），不能是表达式（或常量）。这种方式下，被调用函数中对把式参数的修改实际上就是对实际参数的修改，因此客观上可以实现数据的双向传递。

已知某高级语言源程序 A 经编译后得到机器 C 上的目标程序 B, 则(21)。

- (21)A. 对 B 进行反编译，不能还原出源程序 A  
B. 对 B 进行反汇编，不能得到与源程序 A 等价的汇编程序代码  
C. 对 B 进行反编译，得到的是源程序 A 的变量声明和算法流程  
D. 对 A 和 B 进行交叉编译，可以产生在机器 C 上运行的动态链接库

**【答案】A**

**【解析】**本题考查程序语言方面的基础知识。

编译是将高级语言源程序翻译成机器语言程序（汇编形式或机器代码形式），反编译是编译的逆过程。反编译通常不能把可执行文件还原成高级语言源代码，只能转换成功能上等价的汇编程序。

下面关于程序语言的叙述，错误的是(22)。

- (22)A. 脚本语言属动态语言，其程序结构可以在运行中改变

- B. 脚本语言一般通过脚本引擎解释执行，不产生独立保存的目标程序
- C. php、JavaScript 属于静态语言，其所有成分可在编译时确定
- D. C 语言属于静态语言，其所有成分可在编译时确定

**【答案】C**

**【解析】**本题考查程序语言基础知识。

动态语言是指程序在运行时可以改变其结构，例如新的函数可以被引进、已有的函数可以被删除等在结构上的变化等。动态语言的类型检查是在运行时进行的，其优点是方便阅读，不需要写非常多的与类型相关的代码；缺点是不方便调试，命名不规范时会读不懂、不利于理解等。

脚本语言代表一套与系统程序设计语言不同的协定。它们牺牲执行速度和与系统程序设计语言相关的类型长度而提供更高的编程创作能力和软件重用。脚本语言更适合在联系复杂的应用程序中进行胶着（粘合）。为了简化连接组件的工作，脚本语言被设计为无类型的，脚本语言一般是面向字符的，因为字符为许多不同的事物提供了一致的描述。

事实上，脚本语言都是动态语言，而动态语言都是解释型语言，不管它们是否是面向对象的语言。

在 Windows XP 操作系统中，用户利用“磁盘管理”程序可以对磁盘进行初始化、创建卷，(23)。通常将“C:\Windows\myprogram.exe”文件设置成只读和隐藏属性，以便控制用户对该文件的访问，这一级安全管理称之为(24)安全管理。

- (23)A. 但只能使用 FAT 文件系统格式化卷  
B. 但只能使用 FAT 32 文件系统格式化卷  
C. 但只能使用 NTFS 文件系统格式化卷  
D. 可以选择使用 FAT、FAT32 或 NTFS 文件系统格式化卷
- (24)A. 文件级            B. 目录级            C. 用户级            D. 系统级

**【答案】D A**

**【解析】**本题考查对 Windows XP 操作系统应用的掌握程度。

试题 (23)的正确答案是 D，因为 Windows XP 操作系统支持 FAT、FAT32 或 NTFS 文件系统，所以利用“磁盘管理”程序可以对磁盘进行初始化、创建卷，并可以选择使用 FAT、FAT32 或 NTFS 文件系统格式化卷。

试题 (24)的正确答案是 A。分析如下：文件级安全管理，是通过系统管理员或文件主

对文件属性的设置来控制用户对文件的访问。通常可设置以下几种属性：

- 只执行：只允许用户执行该文件，主要针对.exe和.com文件。
- 隐含：指示该文件为隐含属性文件。
- 索引：指示该文件是索引文件。
- 修改：指示该文件自上次备份后是否还被修改。
- 只读：只允许用户读该文件。
- 读/写：允许用户对文件进行读和写。
- 共享：指示该文件是可读共享的文件。
- 系统：指示该文件是系统文件。

用户对文件的访问，将由用户访问权、目录访问权限及文件属性三者的权限所确定。或者说是有有效权限和文件属性的交集。例如对于只读文件，尽管用户的有效权限是读/写，但都不能对只读文件进行修改、更名和删除。对于一个非共享文件，将禁止在同一时间内由多个用户对它们进行访问。通过上述四级文件保护措施，可有效地保护文件。因此将“C:\Windows\myprogram.exe”文件设置成只读和隐藏属性，以便控制用户对该文件的访问，这一级安全管理称之为文件级安全管理。

在移臂调度算法中，(25)算法可能会随时改变移动臂的运动方向。

- (25)A. 电梯调度和先来先服务      B. 先来先服务和最短寻找时间优先  
C. 单向扫描和先来先服务      D. 电梯调度和最短寻找时间优先

**【答案】B**

**【解析】**本题考查对磁盘调度方面基本知识掌握的程度。

因为先来先服务是谁先请求先满足谁的请求，而最短寻找时间优先是根据当前磁臂到要请求访问磁道的距离，谁短满足谁的请求，故先来先服务和最短寻找时间优先算法可能会随时改变移动臂的运动方向。

设系统中有R类资源m个，现有n个进程互斥使用。若每个进程对R资源的最大需求为w，那么当m、n、w取下表的值时，对于下表中的a~e五种情况，(26)两种情况可能会发生死锁。对于这两种情况，若将(27)，则不会发生死锁。

	a	b	c	d	e
m	2	2	2	4	4
n	1	2	2	3	3
w	2	1	2	2	3

(26) A. a 和 b

B. b 和 c

C. c 和 d

D. c 和 e

(27) A. n 加 1 或 w 加 1

B. m 加 1 或 w 减 1

C. m 减 1 或 w 加 1

D. m 减 1 或 w 减 1

【答案】D B

【解析】本题考查对操作系统死锁方面基本知识掌握的程度。

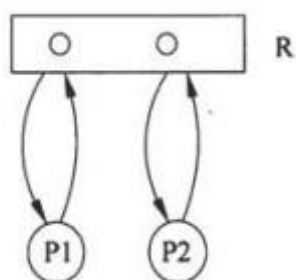
系统中同类资源分配不当会引起死锁。一般情况下，若系统中有  $m$  个单位的存储器资源，它被  $n$  个进程使用，当每个进程都要求  $w$  个单位的存储器资源，当  $m < nw$  时，可能会引起死锁。

试题 (26) 分析如下：

情况 a:  $m=2, n=1, w=2$ , 系统中有 2 个资源，1 个进程使用，该进程最多要求 2 个资源，所以不会发生死锁。

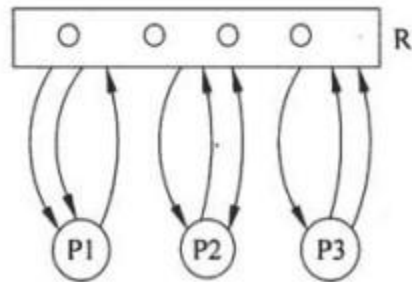
情况 b:  $m=2, n=2, w=1$ , 系统中有 2 个资源，2 个进程使用，每个进程最多要求 1 个资源，所以不会发生死锁。

情况 c:  $m=2, n=2, w=2$ , 系统中有 2 个资源，2 个进程使用，每个进程最多要求 2 个资源，此时，采用的分配策略是轮流地为每个进程分配，则第一轮系统先为每个进程分配 1 个，此时，系统中已无可供分配的资源，使得各个进程都处于等待状态导致系统发生死锁，这时进程资源图如下图所示。



情况 d:  $m=4, n=3, w=2$ , 系统中有 4 个资源，3 个进程使用，每个进程最多要求 2 个资源，此时，采用的分配策略是轮流地为每个进程分配，则第一轮系统先为每个进程分配 1 个资源，此时，系统中还剩 1 个资源，可以使其中的一个进程得到所需资源运行完毕，所以不会发生死锁。

情况 e:  $m=4$ ,  $n=3$ ,  $w=3$ , 系统中有 4 个资源, 3 个进程使用, 每个进程最多要求 3 个资源, 此时, 采用的分配策略是轮流地为每个进程分配, 则第一轮系统先为每个进程分配 1 个, 第二轮系统先为一个进程分配 1 个, 此时, 系统中已无可供分配的资源, 使得各个进程都处于等待状态导致系统发生死锁, 这时进程资源图如下图所示。



试题 (27) 分析如下:

对于 c 和 e 两种情况, 若将  $m$  加 1, 则情况 c:  $m=3$ ,  $n=2$ ,  $w=2$ , 系统中有 3 个资源, 2 个进程使用, 每个进程最多要求 2 个资源, 系统先为每个进程分配 1 个, 此时, 系统中还剩 1 个可供分配的资源, 使得其中的一个进程能得到所需资源执行完, 并释放所有资源使另一个进程运行完; 若将  $w$  减 1, 则情况 c:  $m=2$ ,  $n=2$ ,  $w=1$ , 系统中有 2 个资源, 两个进程各需一个, 系统为每个进程分配 1 个, 此时, 进程都能运行完, 显然不会发生死锁。情况 e 分析同理。

某文件系统采用链式存储管理方案, 磁盘块的大小为 1024 字节。文件 Myfile.doc 由 5 个逻辑记录组成, 每个逻辑记录的大小与磁盘块的大小相等, 并依次存放在 121、75、86、65 和 114 号磁盘块上。若需要存取文件的第 5120 字节处的信息, 应该访问 (28) 号磁盘块。

- (28) A. 75                      B. 85                      C. 65                      D. 114

**【答案】D**

**【解析】** 本题考查对操作系统文件系统空间管理方面基本知识掌握的程度。

根据题意每个逻辑记录的大小与磁盘块大小相等, 并依次存放在 121、75、86、65 和 114 号磁盘块上。而文件的第 5120 字节应该在 114 号磁盘块上。

软件能力成熟度模型 (CMM) 将软件能力成熟度自低到高依次划分为 5 级。目前, 达到 CMM 第 3 级 (已定义级) 是许多组织努力的目标, 该级的核心是 (29)。

- (29) A. 建立基本的项目管理和实践来跟踪项目费用、进度和功能特性

- B. 使用标准开发过程（或方法论）构建（或集成）系统
- C. 管理层寻求更主动地应对系统的开发问题
- D. 连续地监督和改进标准化的系统开发过程

**【答案】B**

**【解析】** 本题考查软件成熟度模型（CMM）的基本概念。建立基本的项目管理和实践来跟踪项目费用、进度和功能特性为可重复级的核心；

使用标准开发过程（或方法论）构建（或集成）系统为已定义级的核心；管理层寻求更主动地应对系统的开发问题为已管理级的核心；连续地监督和改进标准化的系统开发过程为优化级的核心。

RUP 在每个阶段都有主要目标，并在结束时产生一些制品。在(30)结束时产生“在适当的平台上集成的软件产品”。

- (30) A. 初启阶段      B. 精化阶段      C. 构建阶段      D. 移交阶段

**【答案】C**

**【解析】** 本题考查 RUP 中每个阶段产生的制品。

初启阶段结束时产生一个构想文档、一个有关用例模型的调查、一个初始的业务用例、一个早期的风险评估和一个可以显示阶段和迭代的项目计划等制品；精化阶段结束时产生一个补充需求分析、一个软件架构描述和一个可执行的架构原型等制品；构建阶段结束时的成果是一个准备交到最终用户手中的产品，包括具有最初运作能力的在适当的平台上集成的软件产品、用户手册和对当前版本的描述；移交阶段结束时产生移交给用户产品发布版本。

根据 ISO/IEC 9126 软件质量度量模型定义，一个软件的时间和资源质量特性属于(31)质量特性。

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

**【答案】B**

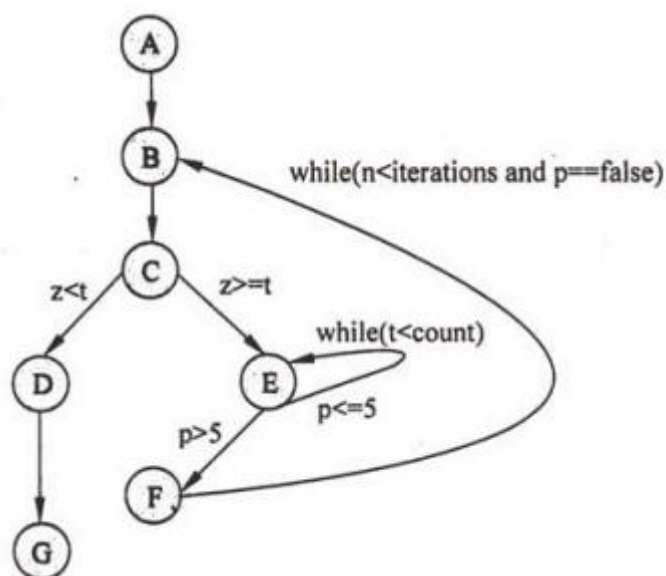
**【解析】** 本题考查 ISO/IEC 9126 软件质量度量模型中的质量特性。

效率质量特性包括时间特性和资源特性两个质量特性。

McCabe 度量法是通过定义环路复杂度，建立程序复杂性的度量，它基于一个程序模块的程序图中环路的个数。计算有向图 G 的环路复杂性的公式为： $V(G)=m-n+2$ ，其中 V(G) 是



有向图 G 中的环路个数，m 是 G 中的有向弧数，n 是 G 中的节点数。下图所示程序图的程序复杂度是 (32)。



- (32) A. 2                      B. 3                      C. 4                      D. 5

【答案】B

【解析】本题考查 McCabe 度量法。

要采用 McCabe 度量法度量程序复杂度，需要先画出程序流图，识别有向图中节点数  $n=7$  有向弧数  $m=8$ ，然后利用环路复杂性计算公式  $V(G)=m-n+2$  进行计算，得如图所示的程序图的程序复杂度是 3。

在开发信息系统时，用于系统开发人员与项目管理人员沟通的主要文档是 (33)。

- (33) A. 系统开发合同    B. 系统设计说明书    C. 系统开发计划    D. 系统测试报告

【答案】C

【解析】

本题考查开发文档的作用系统开发人员与项目管理人员在项目期内进行沟通的文档主要有系统开发计划、系统开发月报以及系统开发总结报告等项目管理文件。

软件工程每一个阶段结束前，应该着重对可维护性进行复审。在系统设计阶段的复审期间，应该从 (34) 出发，评价软件的结构和过程。

- (34) A. 指出可移植性问题以及可能影响软件维护的系统界面  
B. 容易修改、模块化和功能独立的目的

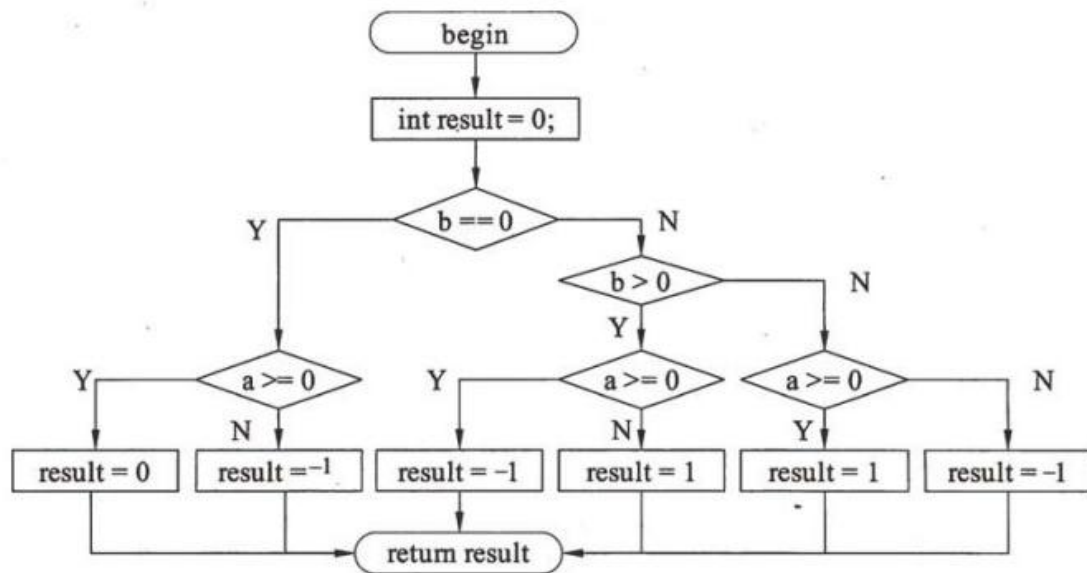
- C. 强调编码风格和内部说明文档
- D. 可测试性

【答案】B

【解析】本题考查软件复审基本概念。

可维护性是所有软件都应具有的基本特点，必须在开发阶段保证软件具有可维护的特点。在系统分析阶段的复审过程中，应该指出软件的可移植性问题以及可能影响软件维护的系统界面；在系统设计阶段的复审期间，应该从容易修改、模块化和功能独立的目的出发，评价软件的结构和过程；在系统实施阶段的复审期间，代码复审应该强调编码风格和内部说明文档这两个影响可维护性的因素。可测试性是可维护性的一个评价指标。

当用分支覆盖法对以下流程图进行测试时，至少需要设计 (35) 个测试用例。



- (35) A. 4                      B. 5                      C. 6                      D. 8

【答案】C

【解析】本题考查分支覆盖测试法。

采用分支覆盖法进行测试使得被测程序中每个判定表达式至少获得一次“真”值和“假”值，或者程序中的每一个取“真”分支和取“假”分支至少都通过一次。需要设计的测试用例需要确保每一个取“真”分支和取“假”分支至少都通过一次，即统计最多分支数。

某银行为了使其网上银行系统能够支持信用卡多币种付款功能而进行扩充升级，这需要对数据类型稍微进行一些改变，这一状况需要对网上银行系统进行 (36) 维护。

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

**【答案】B**

**【解析】**本题考查软件维护的内容。

为了使网上银行系统能够支持信用卡的多币种付款功能而扩充升级,是为了适应银行的市场环境和新的管理需求而提出的新的信息需求,因此需要适应性维护。

下面关于面向对象分析与面向对象设计的说法中,不正确的是(37)。

- (37) A. 面向对象分析侧重于理解问题  
B. 面向对象设计侧重于理解解决方案  
C. 面向对象分析描述软件要做什么  
D. 面向对象设计一般不关注技术和实现层面的细节

**【答案】D**

**【解析】**本题考查面向对象分析与设计的基本概念。

面向对象分析主要强调理解问题是什么,不考虑问题的解决方案,因此答案 A、C 是正确的。面向对象设计侧重问题的解决方案,并且需要考虑实现细节问题,因此选项 D 的说法是不正确的。

在面向对象分析与设计中,(38)是应用领域中的核心类,一般用于保存系统中的信息以及提供针对这些信息的相关处理行为;(39)是系统内对象和系统外参与者的联系媒介;(40)主要是协调上述两种类对象之间的交互。

- (38) A. 控制类                      B. 边界类                      C. 实体类                      D. 软件类  
(39) A. 控制类                      B. 边界类                      C. 实体类                      D. 软件类  
(40) A. 控制类                      B. 边界类                      C. 实体类                      D. 软件类

**【答案】C B A**

**【解析】**本题考查控制类、边界类和实体类的职责。

实体类主要负责数据和业务逻辑;边界类负责和用户进行交互,即用户界面;控制类则负责实体类和界面类的交互。

若类 A 仅在其方法 Method1 中定义并使用了类 B 的一个对象,类 A 其他部分的代码都不涉及类 B,那么类 A 与类 B 的关系应为(41);若类 A 的某个属性是类 B 的一个对象,并且类

A 对象消失时，类 B 对象也随之消失，则类 A 与类 B 的关系应为(42)。

(41)A. 关联                      B. 依赖                      C. 聚合                      D. 组合

(42)A. 关联                      B. 依赖                      C. 聚合                      D. 组合

**【答案】B D**

**【解析】**本题考查类间的关系。

类间关系可分为依赖、关联、聚合、组合和继承 5 种。按照上述顺序，类间关系依次增强，若类 A 的方法中仅仅使用了类 B 的对象，那么类 A 依赖于类 B。如果类 A 的部分是由类 B 的对象组成，并且类 A 控制类 B 的生命周期，那么类 A 与类 B 是组合关系。

当不适合采用生成子类的方法对已有的类进行扩充时，可以采用(43)设计模式动态地给一个对象添加一些额外的职责；当应用程序由于使用大量的对象，造成很大的存储开销时，可以采用(44)设计模式运用共享技术来有效地支持大量细粒度的对象；当想使用一个已经存在的类，但其接口不符合需求时，可以采用(45)设计模式将该类的接口转换成我们希望的接口。

(43)A. 命令 (Command)                      B. 适配器 (Adapter)

C. 装饰 (Decorate)                      D. 享元 (Flyweight)

(44)A. 命令 (Command)                      B. 适配器 (Adapter)

C. 装饰 (Decorate)                      D. 享元 (Flyweight)

(45)A. 命令 (Command)                      B. 适配器 (Adapter)

C. 装饰 (Decorate)                      D. 享元 (Flyweight)

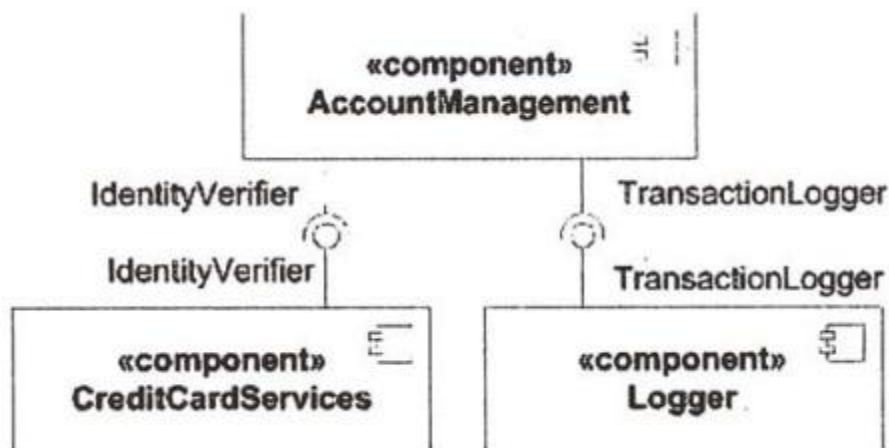
**【答案】C D B**

**【解析】**本题考查面向对象设计。

装饰模式主要的目的是在无法生成子类的情况下给一个对象动态地增加新的职责；

享元设计模式是共享大量细粒度的对象；适配器设计模式则是将已有的接口转换为系统希望的接口形式。

下图属于 UML 中的(46)，其中，AccountManagement 需要(47)。



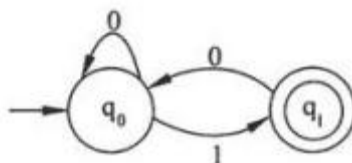
- (46) A. 组件图      B. 部署图      C. 类图      D. 对象图
- (47) A. 实现 IdentityVerifier 接口并被 CreditCardServices 调用  
 B. 调用 CreditCardServices 实现的 Identity Verifier 接口  
 C. 实现 IdentityVerifier 接口并被 Logger 调用  
 D. 调用 Logger 实现的 Identity Verifier 接口

【答案】A B

【解析】本题考查 UML 语言基础知识。

UML 语言是标准的建模语言，通过图形化的方式展现系统的模型。本题是 UML 中的组件图，其表示的含义是组件 AccountManagement 需要调用 CreditCardServices 组件和 Logger 组件分别实现的 Identity Verifier 接口和 TransactionLogger 接口。

下图所示有限自动机的特点是\_(48)。



- (48) A. 识别的 0、1 串是以 0 开头且以 1 结尾      B. 识别的 0、1 串中 1 的数目为偶数  
 C. 识别的 0、1 串中 0 后面必须是 1      D. 识别的 0、1 串中 1 不能连续出现

【答案】D

【解析】本题考查程序语言方面的基础知识。

对于题中自动机的状态图，先忽略状态  $q_0$  的自环（识别若干个 0），从初态  $q_0$  到终态  $q_1$ ，该自动机可识别的字符串为 1、101、10101、…，显然，该自动机识别的 0、1 串中 1 不能连续出现。

由 a、b 构造且仅包含偶数个 a 的串的集合用正规式表示为 (49)。

- (49) A.  $(a^*a)^*b^*$       B.  $(b^*(ab^*a)^*)^*$       C.  $(a^*(ba^*)^*b)^*$       D.  $(a|b)^*(aa)^*$

【答案】B

【解析】本题考查程序语言方面的基础知识。

在正规式中，符号\*表示重复若干次（包括 0 次），因此正规式“中的表达式  $(a^*a)^*$ ”，不能保证有偶数个 a。同理，“ $(a^*(ba^*)^*b)^*$ ”和“ $(a|b)^*(aa)^*$ ”中对 a 的个数也没有限制，而在“ $(ab^*a)^*$ ”中可以确保 a 的出现为偶数个。

设某语言的语法规则用上下文无关文法  $G=(N, T, P, S)$  表示，其中 N 是非终结符号的集合，T 是终结符号的集合，P 是产生式集合，S 是开始符号，令  $V=NUT$ ，那么符合该语言的句子是 (50)。

- (50) A. 从 S 出发推导的、仅包含 T 中符号的符号串  
B. 从 N 中符号出发推导的、仅包含 T 中符号的符号串  
C. 从 S 出发推导的、包含 V 中符号的符号串  
D. 从 N 中符号出发推导的、包含 V 中符号的符号串

【答案】A

【解析】本题考查程序语言方面的基础知识。

一个文法的语言是该文法能产生的句子的集合。一个文法产生的句子是从文法开始符号出发推导出的所有终结符号串。

采用二维表格结构表达实体类型及实体间联系的数据模型是 (51)。

- (51) A. 层次模型      B. 网状模型      C. 关系模型      D. 面向对象模型

【答案】C

【解析】本题考查对数据库数据模型方面基本知识掌握的程度。

不同的数据模型具有不同的数据结构形式。目前最常用的数据结构模型有层次模型 (hierarchical model)、网状模型 (network model)、关系模型 (relational Model) 和面向对象数据模型 (object oriented model)。其中层次模型和网状模型统称为非关系模型。非关系模型的数据库系统在 20 世纪 70 年代非常流行，在数据库系统产品中占据了主导地位。到了 20 世纪 80 年代，逐渐被关系模型的数据库系统取代，但某些地方，由于历史的原因，

目前层次和网状数据库系统仍在使用。

关系模型是目前最常用的数据模型之一。关系数据库系统采用关系模型作为数据的组织方式，在关系模型中用二维表格结构表达实体集以及实体集之间的联系，其最大特色是描述的一致性。关系模型是由若干个关系模式组成的集合。一个关系模式相当于一个记录型，对应于程序设计语言中类型定义的概念。关系是一个实例，也是一张表，对应于程序设计语言中变量的概念。给定变量的值随时间可能发生变化；类似地，当关系被更新时，关系实例的内容也随时间发生了变化。

假设员工关系 EMP（员工号，姓名，部门，部门电话，部门负责人，家庭住址，家庭成员，成员关系）如下表所示。如果一个部门可以有多名员工，一个员工可以有多个家庭成员，那么关系 EMP 属于 (52)，且 (53) 问题：为了解决这一问题，应该将员工关系 EMP 分解为 (54)。

员工号	姓名	部门	部门电话	部门负责人	家庭住址	家庭成员	成员关系
0011	张晓明	开发部	808356	0012	北京海淀区 1 号	张大军	父亲
0011	张晓明	开发部	808356	0012	北京海淀区 1 号	胡敏铮	母亲
0011	张晓明	开发部	808356	0012	北京海淀区 1 号	张晓丽	妹妹
0012	吴俊	开发部	808356	0012	上海昆明路 15 号	吴胜利	父亲
0012	吴俊	开发部	808356	0012	上海昆明路 15 号	王若垚	母亲
0021	李立丽	市场部	808358	0021	西安雁塔路 8 号	李国庆	父亲
0021	李立丽	市场部	808358	0021	西安雁塔路 8 号	罗明	母亲
0022	王学强	市场部	808356	0021	西安太白路 2 号	王国钧	父亲
0031	吴俊	财务部	808360		西安科技路 18 号	吴鸿翔	父亲

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

(53) A. 无冗余、无插入异常和删除异常

B. 无冗余，但存在插入异常和删除异常

C. 存在冗余，但不存在修改操作的不一致

D. 存在冗余、修改操作的不一致，以及插入异常和删除异常

(54) A. EMP1（员工号，姓名，家庭住址）

EMP2（部门，部门电话，部门负责人）

EMP3（员工号，家庭成员，成员关系）

B. EMP1（员工号，姓名，部门，家庭住址）

EMP2（部门，部门电话，部门负责人）

EMP3（员工号，家庭成员，成员关系）



C. EMP1 (员工号, 姓名, 家庭住址)

EMP2 (部门, 部门电话, 部门负责人, 家庭成员, 成员关系)

D. EMP1 (员工号, 姓名, 部门, 部门电话, 部门负责人, 家庭住址)

EMP2 (员工号, 家庭住址, 家庭成员, 成员关系)

**【答案】A D B**

**【解析】**本题考查应试者对范式、模式分解知识的掌握程度。

试题 (52) 考查的是范式的基础知识。员工关系 EMP 属于第一范式的原因是因为其主键是 (员工号, 家庭成员), 非主属性部门名, 负责人, 电话存在对主键的部分函数依赖。所以正确的答案是 A。

试题 (53) 正确的答案是 D, 因为表中存在冗余、修改操作的不一致, 以及插入异常和删除异常。

试题 (54) 正确的答案是 B, 因为对一个给定的关系模式进行分解, 使得分解后的模式是否与原来的模式等价有如下三种情况:

①分解具有无损连接性;

②分解要保持函数依赖;

③分解既要无损连接性, 又要保持函数依赖。

选项 A 是错误的, 因为将原关系模式分解成 EMP1 (员工号, 姓名, 家庭住址), EMP2 (部门, 部门电话, 部门负责人) 和 EMP3 (员工号, 家庭成员, 成员关系) 三个关系模式, 分解后的关系模式既是有损连接, 又不能保持函数依赖。因为此时给定员工号已无法查找所在的部门, 如下表所示。

EMP1			EMP2		
员工号	姓名	家庭住址	员工号	家庭成员	成员关系
0011	张晓明	北京海淀区 1 号	0011	张大军	父亲
0012	吴俊	上海昆明路 15 号	0011	胡敏铮	母亲
0021	李立丽	西安雁塔路 8 号	0011	张晓丽	妹妹
0022	王学强	西安太白路 2 号	0012	吴胜利	父亲
0031	吴俊	西安科技路 18 号	0012	王若垚	母亲
EMP3			0021	李国庆	父亲
部门	部门电话	部门负责人	0021	罗明	母亲
开发部	808356	0012	0022	王国钧	父亲
市场部	808358	0021	0031	吴鸿翔	父亲
财务部	808360				

选项 B 是正确的，因为将原关系模式分解成 EMP1（员工号，姓名，部门，家庭住址），EMP2（部门，部门电话，部门负责人）和 EMP3（员工号，家庭成员，成员关系）既具有无损连接性，又保持了函数依赖。如下表所示。

EMP1				EMP2		
员工号	姓名	部门	家庭住址	员工号	家庭成员	成员关系
0011	张晓明	开发部	北京海淀区 1 号	0011	张大军	父亲
0012	吴俊	开发部	上海昆明路 15 号	0011	胡敏铮	母亲
0021	李立丽	市场部	西安雁塔路 8 号	0011	张晓丽	妹妹
0022	王学强	市场部	西安太白路 2 号	0012	吴胜利	父亲
0031	吴俊	财务部	西安科技路 18 号	0012	王若垚	母亲
EMP3				0021	李国庆	父亲
部门	部门电话	部门负责人		0021	罗明	母亲
开发部	808356	0012		0022	王国钧	父亲
市场部	808358	0021		0031	吴鸿翔	父亲
财务部	808360					

选项 C 是错误的，因为将原关系模式分解成 EMP1（员工号，姓名，家庭住址）和 EMP2（部门，部门电话，部门负责人，家庭成员，成员关系）两个关系模式，分解后的关系模式既有损连接，又不能保持函数依赖。例如，给定员工号无法查找所在的部门，无法查找其家庭成员等信息。如下表所示。

EMP1		
员工号	姓名	家庭住址
0011	张晓明	北京海淀区 1 号
0012	吴俊	上海昆明路 15 号
0021	李立丽	西安雁塔路 8 号
0022	王学强	西安太白路 2 号
0031	吴俊	西安科技路 18 号

EMP2				
部门	部门电话	部门负责人	家庭成员	成员关系
开发部	808356	0012	张大军	父亲
开发部	808356	0012	胡敏铮	母亲
开发部	808356	0012	张晓丽	妹妹
开发部	808356	0012	吴胜利	父亲
开发部	808356	0012	王若垚	母亲
市场部	808358	0021	李国庆	父亲
市场部	808358	0021	罗明	母亲
市场部	808356	0021	王国钧	父亲
财务部	808360		吴鸿翔	父亲

选项 D 是错误的，因为将原关系模式分解成 EMP1（员工号，姓名，部门，部门电话，部门负责人，家庭住址）和 EMP2（员工号，家庭住址，家庭成员，成员关系）两个关系模式，分解后的关系模式存在冗余和修改操作的不一致性。例如，EMP1 中某员工的家庭住址从“陕西省西安市太白路 2 号”修改为“陕西省西安市雁塔路 18 号”，而 EMP2 中该员工的家庭住址未修改，导致修改操作的不一致性。又如，EMP2 中某员工的家庭成员有 5 个，那么其家庭住址就要重复出现 5 次，导致数据的冗余。

关系 R、S 如下图所示，关系代数表达式  $\pi_{3,4,5}(\sigma_{1<6}(R \times S)) = \underline{(55)}$ ，对关系 R、S 进行自然连接后的属性列数和元组个数分别为  $\underline{(56)}$ 。

A	B	C
1	2	4
3	4	5
4	5	9
5	6	6

R

A	B	C
5	3	3
4	6	1
9	8	3
6	9	1

S

A	B	C
1	2	4
5	3	3

(55) A.

A	B	C
5	3	4
9	8	4

B.

A	B	C
5	3	3
9	8	3

C.

A	B	C
1	2	4
3	4	5

D.

(56) A. 3 和 0

B. 3 和 2

C. 6 和 0

D. 6 和 2

【答案】B A

【解析】

本题考查对关系代数运算方面的基础知识。  
本题要求关系代数表达式  $\pi_{3,4,5}(\sigma_{1<6}(R \times S))$  的结果集，其中， $R \times S$  的属性列名分别为 R.A，R.B，R.C，S.A，S.B 和 S.C，其结果如下表所示：

R.A	R.B	R.C	S.A	S.B	S.C
1	2	4	5	3	3
1	2	4	4	6	1
1	2	4	9	8	3
1	2	4	6	9	1
3	4	5	5	3	3
3	4	5	4	6	1
3	4	5	9	8	3
3	4	5	6	9	1
4	5	9	5	3	3
4	5	9	4	6	1
4	5	9	9	8	3
4	5	9	6	9	1
5	6	6	5	3	3
5	6	6	4	6	1
5	6	6	9	8	3
5	6	6	6	9	1

R×S

$\sigma_{1<6}(R \times S)$  的含义是从  $R \times S$  结果集中选取第一个分量（R.A）小于第 6 个分量（S.C）的元组，从上表中可以看出，满足条件的是第一和第三个元组，其结果如下表所示：

R.A	R.B	R.C	S.A	S.B	S.C
1	2	4	5	3	3
1	2	4	9	8	3

$\sigma_{1<6}(R \times S)$

$\pi_{3,4,5}(\sigma_{1<6}(R \times S))$  的含义是从  $\sigma_{1<6}(R \times S)$  结果集中选取第三列、第四列和第五列，其结果如图（a）所示；表中 S.A、S.B 和 R.C 的属性名不重复，所以可以用图（b）表示。

S.A	S.B	R.C
5	3	4
9	8	4

$\pi_{3,4,5}(\sigma_{1<6}(R \times S))$

图（a）

A	B	C
5	3	4
9	8	4

$\pi_{3,4,5}(\sigma_{1<6}(R \times S))$

图（b）

从上面分析可见，试题（55）的正确答案是 B。

试题（56）的正确答案是 A，因为根据自然连接要求，两个关系中进行比较的分量必须是相同的属性组，并且在结果中将重复属性列去掉，故  $R \bowtie S$  后的属性列数应为 3；又因为自然连接是一种特殊的等值连接，即  $R$  关系中的 A、C、D 属性与  $S$  关系中的 A、C、D 属性进行等值连接没有符合条件的元组，故其结果集的元组个数为 0。

下面关于查找运算及查找表的叙述，错误的是（57）。

- (57) A. 哈希表可以动态创建
- B. 二叉排序树属于动态查找表
- C. 二分查找要求查找表采用顺序存储结构或循环链表结构
- D. 顺序查找方法既适用于顺序存储结构，也适用于链表结构

**【答案】C**

**【解析】** 本题考查数据结构方面的基础知识。

哈希表和二叉排序树都可以在查找过程中动态创建，属于动态查找表。顺序查找方法按照设定的次序依次与查找表中元素的关键字进行比较，在顺序存储结构和链表结构上都可以实现该查找过程。二分查找需要对中间元素进行快速定位，在链表结构上无法实现。

下面关于图（网）的叙述，正确的是（58）。

- (58) A. 连通无向网的最小生成树中，顶点数恰好比边数多 1
- B. 若有向图是强连通的，则其边数至少是顶点数的 2 倍
- C. 可以采用 AOV 网估算工程的工期
- D. 关键路径是 AOE 网中源点至汇点的最短路径

**【答案】A**

**【解析】** 本题考查数据结构方面的基础知识。

在有向图中，若以顶点表示活动，用有向边表示活动之间的优先关系，则称这样的有向图为以顶点表示活动的网（Activity On Vertex Network, AOV 网）。

若在带权有向图 G 中以顶点表示事件，以有向边表示活动，边上的权值表示该活动持续的时间，则这种带权有向图称为用边表示活动的网（Activity On Edge Network, AOE 网）。通常在 AOE 网中列出了完成预定工程计划所需进行的活动、每项活动的计划完成时间、要发生哪些事件以及这些事件和活动间的关系，从而可以分析该项工程是否实际可行并估计工程完成的最短时间，分析出哪些活动是影响工程进度的关键。进一步可以进行人力、物力的调度和分配，以达到缩短工期的目的。

根据生成树的定义，有  $n$  个顶点的连通图的生成树中恰好有  $n-1$  条边。

下面关于二叉排序树的叙述，错误的是 (59)。

(59) A. 对二叉排序树进行中序遍历，必定得到节点关键字的有序序列

B. 依据关键字无序的序列建立二叉排序树，也可能构造出单支树

C. 若构造二叉排序树时进行平衡化处理，则根节点的左子树节点数与右子树节点数的差值一定不超过 1

D. 若构造二叉排序树时进行平衡化处理，则根节点的左子树高度与右子树高度的差值一定不超过 1

**【答案】C**

**【解析】** 本题考查数据结构方面的基础知识。

二叉排序树或者是一棵空树，或者是具有如下性质的二叉树：

①若它的左子树非空，则其左子树上所有节点的关键字均小于根节点的关键字；

②若它的右子树非空，则其右子树上所有节点的关键字均大于根节点的关键字；

③左、右子树本身就是两棵二叉排序树。

由上述定义可知，二叉排序树是一个有序表，对二叉排序树进行中序遍历，可得到一个关键字递增排序的序列。

对于给定的关键字序列，可从空树开始，逐个将关键字插入树中来构造一棵二叉排序树。其过程是：每读入一个关键字值，就建立一个新节点。若二叉排序树非空，则将新节点的关键字与根节点的关键字相比较，如果小于根节点的值，则插入到左子树中，否则插入到右子树中；若二叉排序树为空树，则新节点作为二叉排序树的根节点。

显然，若关键字初始序列已经有序，则构造出的二叉排序树一定是单枝树（每个节点只有一个孩子）。

为了使在二叉排序树上进行的查找操作性能最优，构造二叉排序树时需进行平衡化处理，使每个节点左、右子树的高度差的绝对值不超过 1。

下面关于栈和队列的叙述，错误的是 (60)。

(60) A. 栈和队列都是操作受限的线性表

B. 队列采用单循环链表存储时，只需设置队尾指针就可使入队和出队操作的时间复杂度都为  $O(1)$



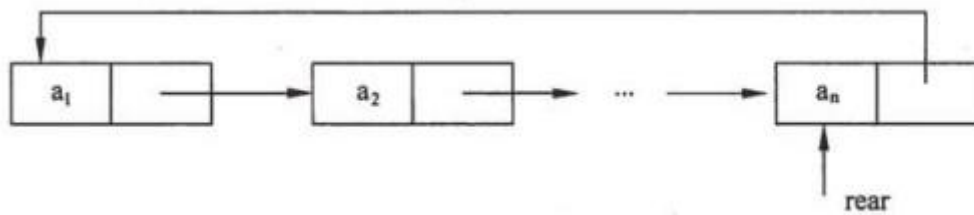
- C. 若队列的数据规模  $n$  可以确定, 则采用顺序存储结构比链式存储结构效率更高
- D. 利用两个栈可以模拟一个队列的操作, 反之亦可

【答案】D

【解析】本题考查数据结构方面的基础知识。

栈和队列都是操作受限的线性表: 栈仅在表尾插入和删除元素; 队列仅在表头删除元素、在表尾插入元素。

采用单循环链表表示队列的示意图如下图所示:



```
q=rear->next;  //q 指向队头元素所在节点
rear->next=q->next;
free(q);
```

入队时初始队列为空、出队后队列变为空要进行特殊处理。

入队操作和出队操作均与队列长度无关, 因此其时间复杂度都为  $O(1)$ 。

队列是先入先出的线性表, 栈是后进先出的线性表。一个线性序列经过队列结构后只能得到与原序列相同的元素序列, 而经过一个栈结构后则可以得到多种元素序列。用两个栈可以模拟一个队列的入队和出队操作。

下面关于二叉树的叙述, 正确的是 (61)。

- (61) A. 完全二叉树的高度  $h$  与其节点数  $n$  之间存在确定的关系
- B. 在二叉树的顺序存储和链式存储结构中, 完全二叉树更适合采用链式存储结构
- C. 完全二叉树中一定不存在度为 1 的节点
- D. 完全二叉树中必定有偶数个叶子节点

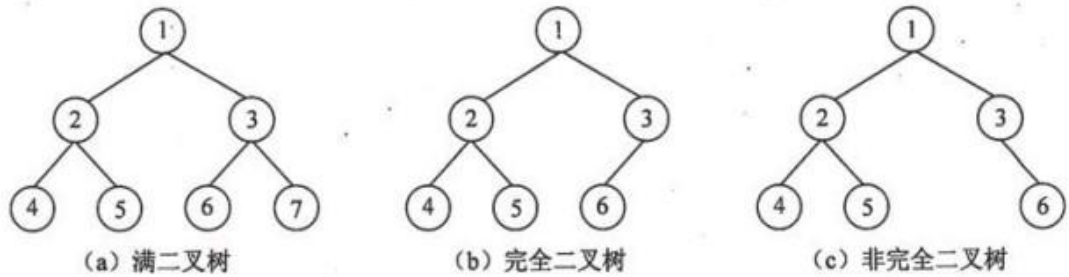
【答案】A

【解析】本题考查数据结构方面的基础知识。

根据其定义, 一棵完全二叉树除了最后一层外, 其余层的节点数都是满的, 最后一层的



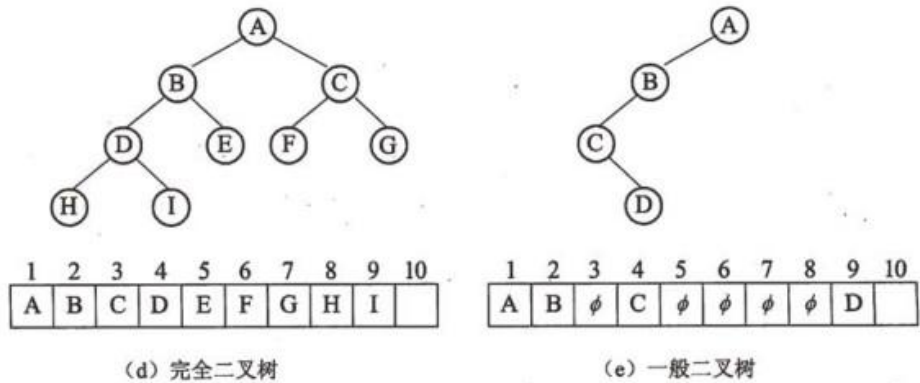
节点也必须自左至右排列，例如图 (a) 是高度为 3 的满二叉树，图 (b) 是完全二叉树，图 (c) 不是完全二叉树。



二叉树采用顺序存储结构时，对于编号为  $i$  的节点，则有：

- 若  $i=1$  时，该节点为根节点，无双亲；
- 若  $i>1$  时，该节点的双亲节点为  $\lfloor i/2 \rfloor$ ；
- 若  $2i \leq n$ ，则该节点的左孩子编号为  $2i$ ，否则无左孩子；
- 若  $2i+1 \leq n$ ，则该节点的右孩子编号为  $2i+1$ ，否则无右孩子。

图 (d) 为具有 10 个节点的完全二叉树及其顺序存储结构，图 (e) 为某非完全二叉树的顺序存储结构，从中可以看出，完全二叉树适合采用顺序存储结构。



可以推导出具有  $n$  个节点的完全二叉树的深度为  $\lfloor \log_2 n \rfloor + 1$ 。

设  $L$  为广义表，将  $\text{head}(L)$  定义为取非空广义表的第一个元素， $\text{tail}(L)$  定义为取非空广义表除第一个元素外剩余元素构成的广义表。若广义表  $L = ((x, y, z), a, (u, t, w))$ ，则从  $L$  中取出原子项  $y$  的运算是 (62)。

- (62) A.  $\text{head}(\text{tail}(\text{tail}(L)))$

C.  $\text{head}(\text{tail}(\text{head}(L)))$

B.  $\text{tail}(\text{head}(\text{head}(L)))$

D.  $\text{tail}(\text{tail}(\text{head}(L)))$

**【答案】C**

**【解析】** 本题考查数据结构方面的基础知识。

广义表是函数式语言中使用的一种数据结构。根据广义表取表头和取表尾的定义，对于广义表  $L = ((x, y, z), a, (u, t, w))$ ，运算  $\text{head}(L) = (x, y, z)$ ，而  $\text{tail}(L) = (a, (u, t, w))$ ，

因此原子项  $y$  应从  $\text{head}(L)$  中取，对  $(x, y, z)$  取表头可得到原子项  $x$ ，因此从  $L$  中取出原子项  $y$  的运算为  $\text{head}(\text{tail}(\text{head}(L)))$ 。

现有 16 枚外形相同的硬币，其中有一枚比真币的重量轻的假币，若采用分治法找出这枚假币，至少比较 (63) 次才能够找出该假币。

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

**【答案】B**

**【解析】** 本题考查算法基础知识。

用分治法找假币的过程为：先将 16 枚硬币对等分为 2 堆（各 8 枚）并比较其重量，假币在较轻的那一堆中；然后将 8 枚硬币对等分为 2 堆（各 4 枚）并比较其重量，假币在较轻的那一堆中；再将 4 枚硬币对等分为 2 堆（各 2 枚）并比较其重量，假币在较轻的那一堆中；最后比较两个硬币的重量，找出假币。因此，至少比较 4 次才能够找出该假币。

以下的算法设计方法中，(64) 以获取问题最优解为目标。

(64) A. 回溯方法              B. 分治法              C. 动态规划              D. 递推

**【答案】C**

**【解析】** 本题考查算法基础知识。

回溯法的实质是在包含问题的所有解的解空间树中，按照深度优先的策略，从根节点出发搜索解空间树。若进入某子节点的子树后没有找到解（或者需要找出全部解），则需要从子节点回退（回溯）至父节点，从而可以选择其他子节点进行搜索。回溯法有“通用的解题法”之称，用它可以系统地搜索一个问题的所有解或任一解。

分治与递归就像一对孪生兄弟，经常同时应用于算法设计之中。分治的思路是将一个难以直接解决的大问题分解成一些规模较小的相同问题，以便各个击破，分而治之。

如果规模为  $n$  的问题可分解成  $k$  个子问题， $1 < k < n$ ，这些子问题互相独立且与原问题相同。动态规划算法与分治法类似，其基本思想也是将待求解问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解。与分治法不同的是，适合于用动态规划法求解的问题，经分解得到的子问题往往不是独立的。若用分治法来解这类问题，则相同的子问题会被求解多次，以至于最后解决原问题需要耗费指数级时间。动态规划算法通常用于求解具有某种最优性质的问题。在这类问题中，可能会有许多可行解，每个解都对应于一个值，我们希望找到具有最优值（最大值或最小值）的那个解。

归并排序采用的算法设计方法属于 (65)。

- (65) A. 归纳法                      B. 分治法                      C. 贪心法                      D. 回溯方法

**【答案】B**

**【解析】** 本题考查算法基础知识。

以 2-路归并排序为例进行说明。2-路归并是指将两个有序序列合并成一个有序序列，其基本过程为：从两个序列中各取一个元素，进行比较，输出较小的元素，从较小元素所在序列取下一个元素，与未输出的那个元素比较，输出较小者。依此类推，直到输出序列包含了两个初始有序序列的全部元素。

对于一个初始无序的序列，可以先将其等分为两个无序的子序列，对这两个子序列再次二分，重复该过程，直到分出的子序列中仅包含一个元素时（一个元素自然是有序的）为止，然后再反复进行 2-路归并的过程，最后完成排序。

一个 B 类网络的子网掩码为 255. 255. 224. 0, 则这个网络被划分成了 (66) 个子网。

- (66) A. 2                              B. 4                              C. 6                              D. 8

**【答案】D**

**【解析】**

子网掩码 255. 255. 224. 0 的二进制表示为 11111111. 11111111. 11100000. 00000000, 比正常的 B 类子网掩码为 255. 255. 0. 0 多出了 3 位“1”，所以把 B 类网络划分成了 8 个子网。

在 Windows 系统中设置默认路由的作用是 (67)。

- (67) A. 当主机接收到一个访问请求时首先选择的路由  
B. 当没有其他路由可选时最后选择的路由  
C. 访问本地主机的路由  
D. 必须选择的路由

**【答案】B**

**【解析】**

Windows Server 2003 的路由类型有 5 种，见下表。当 Windows 服务器收到一个 IP 数据包时，先查找主机路由，再查找网络路由（直连网络和远程网络），这些路由查找失败时，最后才查找默认路由。

路由类型

路由类型	说明
直连网络 ID (Directly attached network ID)	用于直接连接的网络, Interface (或 next hop) 可以为空
远程网络 ID (Remote network ID)	用于不直接连接的网络, 可以通过其他路由器到达这种网络 Interface 字段是本地路由器的 IP 地址
主机路由 (Host route)	到达特定主机的路由, 子网掩码为 255.255.255.255
默认路由 (Default route)	无法找到确定路由时使用的路由, 目标网络和网络掩码都是 0.0.0.0
持久路由 (Persistent route)	利用 route add -p 命令添加的表项, 每次初始化时, 这种路由都会加入 Windows 的注册表中, 同时加入路由表

HTML<body>元素中, (68)属性用于定义超链接被鼠标点击后所显示的颜色。

- (68)A. alink                      B. background                      C. bgcolor                      D. vlink

【答案】D

【解析】本题考查的是 HTML 基本标签的使用。

alink 用于设置正在被击中的链接的颜色。 vlink 用于设置已使用的链接的颜色。

background 用于设置背景图片的 URL。 bgcolor 用于设置文档整体背景颜色。

HTML 中<tr>标记用于定义表格的 (69)。

- (69)A. 行                      B. 列                      C. 单元格                      D. 标题

【答案】A

【解析】本题考查 HTML 标记中 tr 标记的使用。

<tr>标记用于定义表格中的一行。 <col>标记用于定义表格中一个或多个列的属性值。

<td>标记用于定义表格中的一个单元格。 <title>标记用于定义文档标题。

以下不符合 XML 文档语法规范的是 (70)。

- (70)A. 文档的第一行必须是 XML 文档声明                      B. 文档必须包含根元素  
C. 每个开始标记必须和结束标记配对使用                      D. 标记之间可以交叉嵌套

【答案】D

【解析】本题考查 XML 语法的基础知识。

XML 文件的第一行必须是声明该文件是 XML 文件以及它所使用的 XML 规范版本。在文件

的前面不能够有其他元素或者注释。所有的 XML 文档必须有一个根元素。XML 文档中的第一个元素就是根元素。所有 XML 文档都必须包含一个单独的标记来定义，所有其他元素都必须成对地在根元素中嵌套。XML 文档有且只能有一个根元素。所有的元素都可以有子元素，子元素必须正确地嵌套在父元素中。在 XML 中规定，所有标识必须成对出现，有一个开始标识，就必须有一个结束标识，否则将被视为错误。

For nearly ten years, the Unified Modeling Language (UML) has been the industry standard for visualizing, specifying, constructing, and documenting the (71) of a software-intensive system. As the (72) standard modeling language, the UML facilitates communication and reduces confusion among project (73). The recent standardization of UML 2.0 has further extended the language's scope and viability. Its inherent expressiveness allows users to (74) everything from enterprise information systems and distributed Web-based applications to real-time embedded systems.

The UML is not limited to modeling software. In fact, it is expressive enough to model (75) systems, such as workflow in the legal system, the structure and behavior of a patient healthcare system, software engineering in aircraft combat systems, and the design of hardware.

To understand the UML, you need to form a conceptual model of the language, and this requires learning three major elements: the UML's basic building blocks, the rules that dictate how those building blocks may be put together, and some common mechanisms that apply throughout the UML.

- |                     |                 |              |                 |
|---------------------|-----------------|--------------|-----------------|
| (71)A. classes      | B. components   | C. sequences | D. artifacts    |
| (72)A. real         | B. legal        | C. de facto  | D. illegal      |
| (73)A. investors    | B. developers   | C. designers | D. stakeholders |
| (74)A. model        | B. code         | C. test      | D. modify       |
| (75)A. non-hardware | B. non-software | C. hardware  | D. software     |

**【答案】** D C D A B

**【解析】**

近十年来，统一建模语言（UML）已经成为工业标准，它可用来可视化、规范化说明、构

建以及文档化软件密集系统中的开发制品。作为事实上的工业标准，UML 能够方便项目相关人员的沟通并减少理解上的二义问题。UML 2.0 标准扩宽了该语言的应用范围，它所具有的表达能力能够让用户对企业信息系统、分布式 Web 系统和嵌入式实时系统进行建模。

UML 不仅能够对软件系统进行建模，实际上，它具有足够的能力去对法律系统中的工作流、病人监护系统中的结构和行为、飞行战斗系统和硬件系统进行建模。

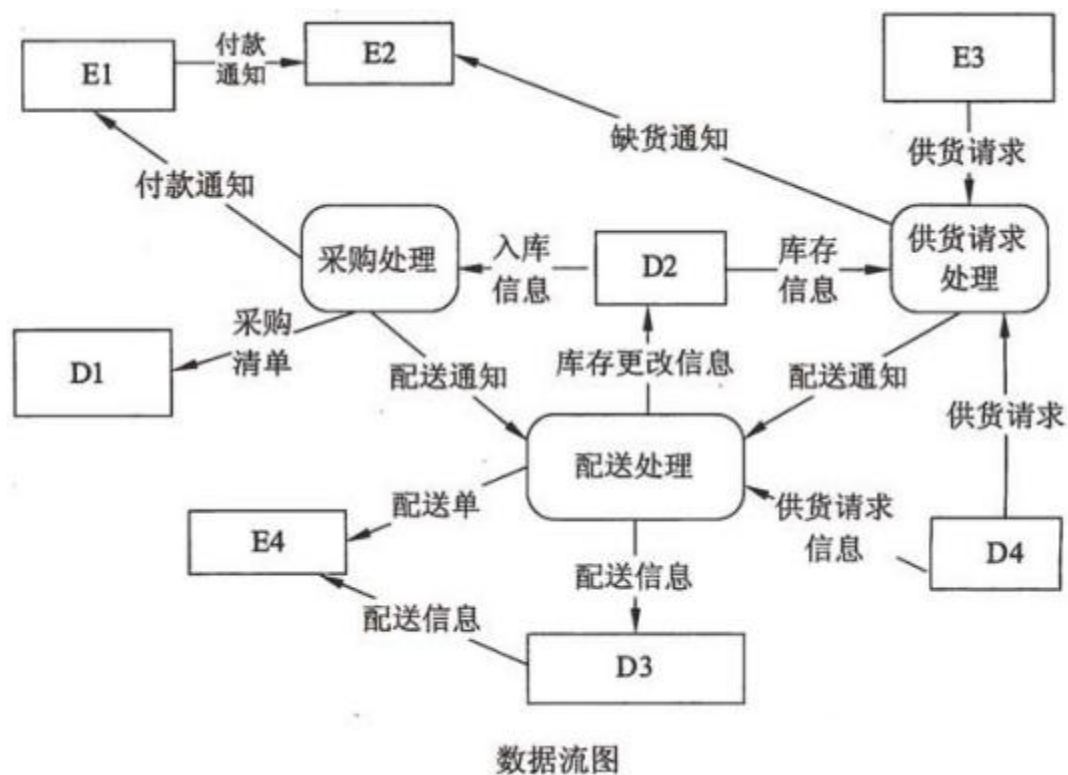
为了理解 UML, 需要具备该语言的概念模型，这需要学习三个主要元素：UML 的基本构造块，基本构造块的关系规则和应用这些构造块与规则的通用机制。

### 试题一

假设某大型商业企业由商品配送中心和连锁超市组成，其中商品配送中心包括采购、财务、配送等部门。为实现高效管理，设计了商品配送中心信息管理系统，其主要功能描述如下：

1. 系统接收由连锁超市提出的供货请求，并将其记录到供货请求记录文件。
2. 在接到供货请求后，从商品库存记录文件中进行商品库存信息查询。如果库存满足供货请求，则给配送处理发送配送通知；否则，向采购部门发出缺货通知。
3. 配送处理接到配送通知后，查询供货请求记录文件，更新商品库存记录文件，并向配送部门发送配送单，在配送货品的同时记录配送信息至商品配送记录文件。
4. 采购部门接到缺货通知后，与供货商洽谈，进行商品采购处理，合格商品入库，并记录采购清单至采购清单记录文件、向配送处理发出配送通知，同时通知财务部门给供货商支付货款。

该系统采用结构化方法进行开发，得到待修改的数据流图如下图所示。



#### 【问题1】

使用【说明】中的词语，给出上图中外部实体 E1 至 E4 的名称和数据存储 D1 至 D4 的名称。



E1: 财务部门

E2: 采购部门

### E3: 连锁超市

E4: 配送部門

D1: 采购清单记录文件 D2: 商品库存记录文件

D3: 商品配送记录文件 D4: 供货请求记录文件

本题考查 DFD 的分析与设计，问题一主要考查 DFD 中的外部实体和数据存储，由于在题干中已经提到“系统接收由连锁超市提出的供货请求，并将其记录到供货请求记录文件”，因此可以明确出“连锁超市”外部实体和“供货请求记录文件”数据存储；对应到 DFD 图中为 E3 和 D4。描述中的第二项提出“从商品库存记录文件中进行商品库 存信息查询。如果库存满足供货请求，则给配送处发送配送通知；否则，向采购部门发出缺货通知”，因为配送通知需要发送到采购部门，因此采购部门将成为系统的外部实体；同时，商品库存记录文件能够提供库存信息，所以 DFD 图中 E2 和 D2 分别为采购部门和商品配送记录文件。第三项需求“配送处理接到配送通知后，查询供货请求记录文件，更新商品库存记录文件，并向配送部门发送配送单，在配送货品的同时记录配送信息至 商品配送记录文件”，所以配送处理需要查询供货请求记录文件，更新商品库存记录文件与商品配送记录文件，因此 D3 为商品配送记录文件；采购处理需要记录采购清单同时通知财务部门，所以 E1 应该为财务部门，D1 为采购清单记录文件，剩下的 E4 则为配送部门。

DFD 中出现的错误数据流为：E1 到 E2，E1 与 E2 的数据流不属于系统的范围；D3 到 E4，多余的数据流；D2 到采购处理，数据流方向错误；D4 到供货请求处理，数据流方向错误。

需要补充的数据流为：E2 到采购处理，因为 E2 是采购部门，采购部门需要给采购处提供入库商品信息；采购处到 D2 需要一条数据流，因为采购处理需要更改库存信息；供货请求处理到 D4 需要一条数据流，因为供货请求处理需要记录供货请求信息。

### 【问题 2】

以上数据流图中存在四处错误数据流，请指出各自的起点和终点；若将上述四条错误数据流删除，为保证数据流图的正确性，应补充三条数据流，请给出所补充数据流的起点和终点。（起点和终点请采用上述数据流图中的符号或名称）

错误数据流

起点	终点

补充的数据流

起点	终点

错误数据流

起点	终点
E1	E2
D3	E4
D2	采购处理
D4	供货请求处理

补充的数据流

起点	终点
E2	采购处理
采购处理	D2
供货请求处理	D4

本题考查 DFD 的分析与设计，问题一主要考查 DFD 中的外部实体和数据存储，由于在题干中已经提到“系统接收由连锁超市提出的供货请求，并将其记录到供货请求记录文件”，因此可以明确出“连锁超市”外部实体和“供货请求记录文件”数据存储；对应到 DFD 图中为 E3 和 D4。描述中的第二项提出“从商品库存记录文件中进行商品库存信息查询。如果库存满足供货请求，则给配送处发送配送通知；否则，向采购部门发出缺货通知”，因为配送通知需要发送到采购部门，因此采购部门将成为系统的外部实体；同时，商品库存记录文件能够提供库存信息，所以 DFD 图中 E2 和 D2 分别为采购部门和商品配送记录文件。第三项需求“配送处理接到配送通知后，查询供货请求记录文件，更新商品库存记录文件，并向配送

部门发送配送单，在配送货品的同时记录配送信息至 商品配送记录文件”，所以配送处理需要查询供货请求记录文件，更新商品库存记录文件与商品配送记录文件，因此 D3 为商品配送记录文件；采购处理需要记录采购清单同时通知财务部门，所以 E1 应该为财务部门，D1 为采购清单记录文件，剩下的 E4 则为配送部门。DFD 中出现的错误数据流为：E1 到 E2, E1 与 E2 的数据流不属于系统的范围；D3 到 E4, 多余的数据流；D2 到采购处理，数据流方向错误；D4 到供货请求处理，数据流方向错误。需要补充的数据流为：E2 到采购处理，因为 E2 是采购部门，采购部门需要给采购处提供入库商品信息；采购处到 D2 需要一条数据流，因为采购处理需要更改库存信息；供货请求处理到 D4 需要一条数据流，因为供货请求处理需要记录供货请求信息。

试题三

某集团公司拥有多个大型连锁商场，公司需要构建一个数据库系统以方便管理其业务运作活动。

【需求分析结果】

1. 商场需要记录的信息包括商场编号（编号唯一），商场名称，地址和联系电话。某商场信息如下表所本。

商场信息表

商场编号	商场名称	地址	联系电话
PS2101	淮海商场	淮海中路 918 号	021-64158818
PS2902	西大街商场	西大街时代盛典大厦	029-87283220
PS2903	东大街商场	碑林区东大街 239 号	029-87450287
PS2901	长安商场	雁塔区长安中路 38 号	029-85264953

2. 每个商场包含有不同的部门，部门需要记录的信息包括部门编号（集团公司分配），部门名称，位置分布和联系电话。某商场的部门信息如下表所示。

部门信息表

部门编号	部门名称	位置分布	联系电话
DT002	财务部	商场大楼六层	82504342
DT007	后勤部	商场地下副一层	82504347
DT021	安保部	商场地下副一层	82504358
DT005	人事部	商场大楼六层	82504446
DT001	管理部	商场裙楼三层	82504668

3. 每个部门雇用多名员工处理日常事务，每名员工只能隶属于一个部门（新进员工在培训期不隶属于任何部门）。员工需要记录的信息包括员工编号（集团公司分配），姓名，岗位，电话号码和工资。员工信息如下表所示。

员工信息表

员工编号	姓名	岗位	电话号码	工资
XA3310	周 超	理货员	13609257638	1500.00
SH1075	刘 飞	防损员	13477293487	1500.00
XA0048	江雪花	广播员	15234567893	1428.00
BJ3123	张正华	部门主管	13345698432	1876.00

4. 每个部门的员工中有一名是经理，每个经理只能管理一个部门，系统需要记录每个经理的任职时间。

### 【概念模型设计】



实体联系图

### 【关系模式设计】

商场（商场编号，商场名称，地址，联系电话）

部门（部门编号，部门名称，位置分布，联系电话，(a)）

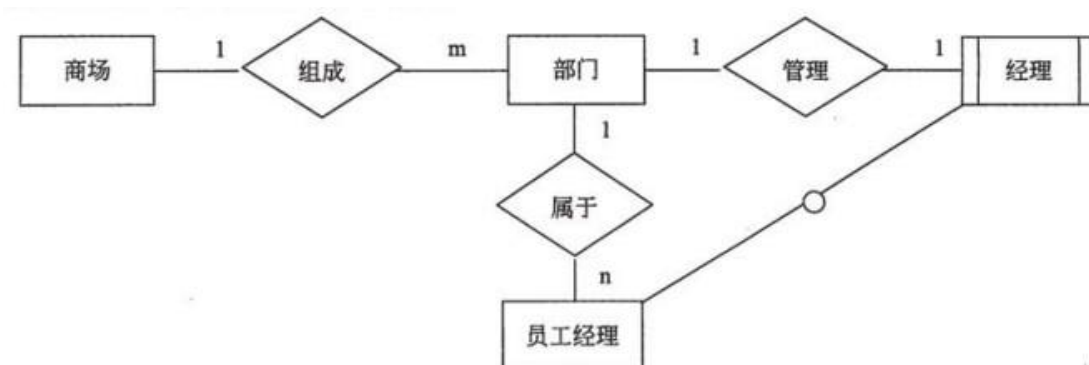
员工：（员工编号，员工姓名，岗位，电话号码，工资，(b)）

经理((c), 任职时间)

### 【问题 1】

根据问题描述，补充四个联系，完善图 2-1 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。

（图中的 m、n 也可用\*表示，对联系名称可不作要求，但不能出现重名）



由“每个商场包含有不同的部门”可知商场与部门间为 1:m 联系；由“每个部门雇用了多名员工处理日常事务”可知部门与员工间为 1:n 联系；由“每个部门的员工中有一个经理……每个经理只能管理一个部门”可知部门与经理间为 1:1 联系，并且员工是经理的超类型，经理是员工的子类型。

### 【问题 2】

根据实体联系图，将关系模式中的空 (a)～(c) 补充完整，并分别给出部门、员工和经理关系模式的主键和外键。

(a) 商场编号

(b) 部门编号

(c) 员工编号

部门关系模式的主键：部门编号 外键：商场编号

员工关系模式的主键：员工编号 外键：部门编号

经理关系模式的主键：员工编号 外键：员工编号

商场的属性信息中，商场编号由集团公司分配，不会重复，可作为商场的主键属性；部门的属性信息中，部门编号由集团公司分配，不会重复，可作为部门的主键属性，商场与部门的联系需要通过将商场的主键（商场编号）加入到部门中来表达；员工的属性信息中，员工编号由集团公司分配，不会重复，可作为员工的主键属性，部门与员工的联系需要通过将部门的主键（部门编号）加入到员工中来表达；经理除了包含员工的属性信息外，还需要任职时间属性。完整的模式如下：

商场（商场编号，商场名称，地址，联系电话）

部门（部门编号，部门名称，位置分布，联系电话，商场编号）

员工（员工编号，姓名，岗位，电话号码，工资，部门编号）

经理（员工编号，任职时间）

### 【问题 3】

为了使商场有紧急事务时能联系到轮休的员工，要求每位员工必须且只能登记一位紧急联系人的姓名和联系电话，不同的员工可以登记相同的紧急联系人。则在图 2-1 中 还需添加的实体是 (1)，该实体和图 2-1 中的员工存在 (2) 联系(填写联系类型)。 给出该实体的关系模式。

(d) 紧急联系人

(e) 1:n

关系模式：紧急联系人（员工编号，姓名，联系电话）

员工的紧急联系人信息通过添加紧急联系人关系来实现，由“每位员工必须且只能登记一位紧急联系人的姓名和联系电话”，但可能存在多位员工登记同一位家属，可知员工与家属间为 n:1 联系；由“不同员工可以登记相同的紧急联系人”可知，员工编号可作为家属的主键属性。所以需要添加的关系模式如下：

紧急联系人（员工编号，姓名，联系电话）

### 试题三

某银行计划开发一个自动存提款机模拟系统(ATM System)。系统通过读卡器(CardReader)读取ATM卡;系统与客户(Customer)的交互由客户控制台(Customer-Console)实现;银行操作员(Operator)可控制系统的启动(System Startup)和停止(System Shutdown);系统通过网络和银行系统(Bank)实现通信。

当读卡器判断用户已将ATM卡插入后,创建会话(Session)。会话开始后,读卡器进行读卡,并要求客户输入个人验证码(PIN)。系统将卡号和个人验证码信息送到银行系统进行验证。验证通过后,客户可从菜单选择如下事务(Transaction):

1. 从ATM卡账户取款(Withdraw);
2. 向ATM卡账户存款(Deposit);
3. 进行转账(Transfer);
4. 查询(Inquire)ATM卡账户信息。

一次会话可以包含多个事务,每个事务处理也会将卡号和个人验证码信息送到银行系统进行验证。若个人验证码错误,则转个人验证码错误处理(Invalid PIN Process)。每个事务完成后,客户可选择继续上述事务或退卡。选择退卡时,系统弹出ATM卡,会话结束。系统采用面向对象方法开发,使用UML进行建模。系统的顶层用例图如图3-1所示,一次会话的序列图(不考虑验证)如图3-2所示。

#### 【问题1】

根据【说明】中的描述,给出图3-1中A1和A2所对应的参与者,U1至U3所对应的用例,以及该图中空(1)所对应的关系。(U1至U3的可选用例包括:Session、Transaction, Insert Card、Invalid PIN Process 和 Transfer)

A1: Customer

A2: Bank

U1: Session

U2: Invalid PIN Process

U3: Transaction

(1): «extend»

构建用例图时,常用的方式是先识别参与者,然后确定用例以及用例之间的关系。识别参与



者时，考查和系统交互的人员和外部系统。本题中，与系统交互的人员包括客户（Customer）和银行操作员（Operator），与本模拟系统交互的外部系统包括银行。系统（Bank）。

考查用例时，通过判断哪一个特定参与者发起或者触发了与系统的哪些交互，来识别用例并建立和参与者之间的关联。考查用例之间的关系时，《include》（包含）定义了用例之间的包含关系，用于一个用例包含另一个用例的行为的建模；如果可以从一个用例的执行中，在需要时转向执行另一个用例，执行完返回之前的用例继续执行，用例间即存在《extend》关系。

本题中，客户一旦插卡成功，系统就创建会话（Session），会话中可以执行用户从菜单选择的 Withdraw、Deposit、Transfer 和 Inquire 等事务（Transaction）。由图中 U3 和 Withdraw 之间的扩展关系，可知 U3 为 Transaction；又由 U1 和 U3 之间的《include》关系，得知 U1 为 Session，进而判定图中 A1 为 Customer，A2 为 Bank。每个事务处理 也会将卡号和个人验证码信息送到银行系统进行验证，若个人验证码错误，则转个人验证码错误处理（Invalid PIN Process，图中 U2），所以（1）处应填《extend》。

## 【问题 2】

根据【说明】中的描述，使用消息名称列表中的英文名称，给出图 3-2 中 6~9 对应的消息。

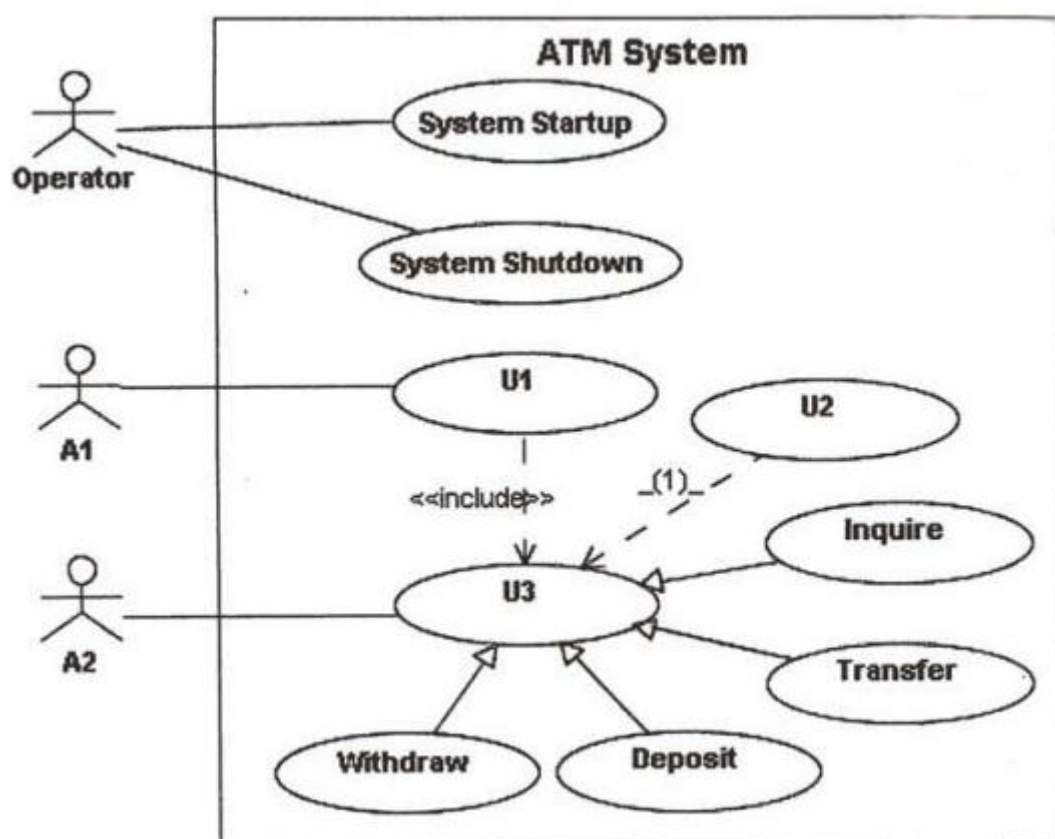


图 3-1 ATM 系统顶层用例图

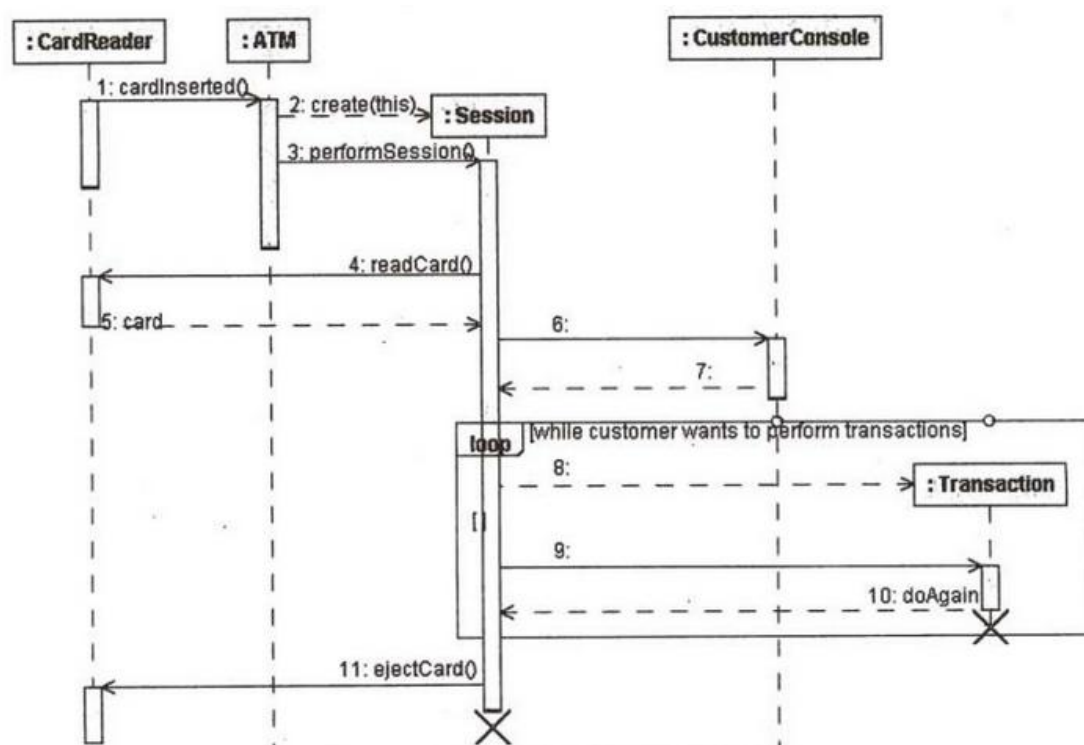


图 3-2 一次会话的序列图（无验证消息）

可能的消息名称列表			
名 称	说 明	名 称	说 明
cardInserted()	ATM 卡已插入	performTransaction()	执行事务
performSession()	执行会话	readCard()	读卡
readPIN()	读取个人验证码	PIN	个人验证码信息
creat(atm, this, card, pin)	为当前会话创建事务	create(this)	为当前 ATM 创建会话
card	ATM 卡信息	doAgain	执行下一个事务
ejectCard()	弹出 ATM 卡		

6: readPIN()

7: PIN

8: creat(atm, this, card, pin)

9: performTransaction()

序列图是场景的图形化表示，描述了以时间顺序组织的对象之间的交互活动。构造序列图时遵循如下指导原则：确定顺序图的范围，描述这个用例场景或一个步骤；绘制参与者和接口类，如果范围包括这些内容的话；沿左边列出用例步骤；对控制器类及必须在顺序中协作的每个实体类，基于它拥有的属性或已经分配给它的行为绘制框；为持续类和系统类绘制框；绘制所需消息，并把每条消息指到将实现响应消息的责任的类上；添加活动条指示每个对象实例的生命期；为清晰起见，添加所需的返回消息；如果需要，为循环、可选步骤和替代步骤等添加框架。

本题中，根据说明中的描述，从 ATM 机判断卡已插入 (cardInserted()) 开始会话，即为当前 ATM 创建会话 (create(this)) 并开始执行会话 (performSession()); 读卡器读卡 (readCard()) 获得 ATM 卡信息 (card)，然后从控制台读取个人验证码输入 (readPIN(), 图中标号 6 处) 并获得个人验证码信息 (PIN, 图中标号 7 处); 然后根据用户选择启动并执行事务，即为当前会话创建事务 (creat(atm, this, card, pin), 图中标号 8 处) 和执行事务 (performTransaction(), 图中标号 9 处); 可以选择继续执行某个事务 (doAgain) 循环，或者选择退卡 (ejectCard())。

### 【问题 3】

解释图 3-1 中用例 U3 和用例 Withdraw、Deposit 等四个用例之间的关系及其内涵。

Transaction 是一个抽象泛化用例，具有其他事务类型共有的属性和行为，每个具体的事务类型继承它，并实现适合自己的特定的操作。

用例之间的继承关系表示子类型“是一种”父类型。其中父类型通常是一个抽象泛化用例，具有子类型共有的属性和行为，每个具体的子类型继承它，并实现适合自己的特定的操作。本题中 Transaction 和 Withdraw、Deposit 等四个用例之间的关系即为继承关系，Transaction 即是一个抽象泛化用例，具有其他事务类型共有的属性和行为，每个具体的事务类型继承它，并实现适合自己的特定的操作。

#### 试题四

现需在某城市中选择一个社区建一个大型超市,使该城市的其他社区到该超市的距离总和最小。用图模型表示该城市的地图,其中顶点表示社区,边表示社区间的路线,边上的权重表示该路线的长度。

现设计一个算法来找到该大型超市的最佳位置:即在给定图中选择一个顶点,使该顶点到其他各顶点的最短路径之和最小。算法首先需要求出每个顶点到其他任一顶点的最短路径,即需要计算任意两个顶点之间的最短路径;然后对每个顶点,计算其他各顶点到该顶点的最短路径之和;最后,选择最短路径之和最小的顶点作为建大型超市的最佳位置。

##### 【问题 1】

本题采用 Floyd-Warshall 算法求解任意两个顶点之间的最短路径。已知图  $G$  的顶点集合为  $V = \{1, 2, \dots, n\}$ ,  $W = \{w_{ij}\}_{n \times n}$  为权重矩阵。设  $d_{ij}^{(k)}$  为从顶点  $i$  到顶点  $j$  的一条最短路径的权重。当  $k = 0$  时,不存在中间顶点,因此  $d_{ij}^{(0)} = w_{ij}$ ; 当  $k > 0$  时,该最短路径上所有的中间顶点均属于集合  $\{1, 2, \dots, k\}$ 。若中间顶点包括顶点  $k$ , 则  $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$ ; 若中间顶点不包括顶点  $k$ , 则  $d_{ij}^{(k)} = d_{ij}^{(k-1)}$ 。于是得到如下递归式。

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & k = 0 \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & k > 0 \end{cases}$$

因为对于任意路径,所有的中间顶点都在集合  $\{1, 2, \dots, n\}$  内,因此矩阵  $D^{(n)} = \{d_{ij}^{(n)}\}_{n \times n}$  给出了任意两个顶点之间的最短路径,即对所有  $i, j \in V$ ,  $d_{ij}^{(n)}$  表示顶点  $i$  到顶点  $j$  的最短路径。

```

LOCATE -SHOPPINGMALL(W, n)
1  D(0) = W
2  for _____ (1)
3      for i = 1 to n
4          for j = 1 to n
5              if  $d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$ 
6                  _____ (2)
7              else
8                  _____ (3)
9  for i = 1 to n
10     SP[i] = 0
11     for j = 1 to n
12         _____ (4)
13 min_SP = SP[1]
14 _____ (5)
15 for i = 2 to n

```

(1)  $k = 1 \text{ to } n$                       (2)  $d_{ij}^{(k)} = d_{ij}^{(k-1)}$                       (3)  $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$   
 (4)  $SP[i] = SP[i] + d_{ij}^{(n)}$                       (5)  $\min\_v = 1$                       (6)  $\min\_v$

本问题考查算法流程。第(1)空表示主循环，k 是循环控制变量，故第(1)空填  $k = 1 \text{ to } n$ 。第(2)和(3)空根据题意和递归式，可分别得到答案为  $d_{ij}^{(k)} = d_{ij}^{(k-1)}$  和  $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$ 。计算了任意两个顶点之间的最短路径之后，对每个顶点，开始统计其到所有其他顶点的最短路径之和，因此第(4)空填  $SP[i] = SP[i] + d_{ij}^{(n)}$ 。第13和第14行初始化，假设最小的到所有其他顶点的最短路径之和为第一个顶点的最小路径之和，大型超市的最佳位置为第一个顶点，故第(5)空填  $\min\_v = 1$ 。最后要求返回大型超市的最佳位置，即到所有其他顶点的最短路径之和最小的顶点，故第(6)空填  $\min\_v$ 。

## 【问题2】

【问题1】中伪代码的时间复杂度为(7)(用O符号表示)。

(7)  $O(n^3)$

本问题考查【问题1】中的伪代码第2~8行，计算任意两点之间的最短路径，有三重循环，故时间复杂度为  $O(n^3)$ 。第9~12行，计算每个点到任意其他点的最短路径之和，有两重循环，故时间复杂度为  $O(n^2)$ 。第15~18行，在所有点的最短路径之和中找到最小的最短路径之和，时间复杂度为  $O(n)$ 。故算法总的时间复杂度为  $O(n^3)$ 。

## 试题五

阅读下列说明和 C 函数代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

### 【说明】

对二叉树进行遍历是二叉树的一个基本运算。遍历是指按某种策略访问二叉树的每个节点，且每个节点仅访问一次的过程。函数 InOrder() 借助栈实现二叉树的非递归中序遍历运算。

设二叉树采用二叉链表存储，节点类型定义如下：

```
typedef struct BtNode{
    ElemType data;           /*节点的数据域，ElemType 的具体定义省略*/
    struct BtNode *lchild,*rchild; /*节点的左、右孩子指针域*/
}BtNode, *BTree;
```

在函数 InOrder() 中，用栈暂存二叉树中各个节点的指针，并将栈表示为不含头节点的单向链表（简称链栈），其节点类型定义如下：

```
typedef struct StNode{      /*链栈的节点类型*/
    BTree elem;             /*栈中的元素是指向二叉链表节点的指针*/
    struct StNode *link;
}StNode;
```

假设从栈顶到栈底的元素为  $e_n$ 、 $e_{n-1}$ 、 $\dots$ 、 $e_1$ ，则不含头节点的链栈示意图如图 5-1 所示。

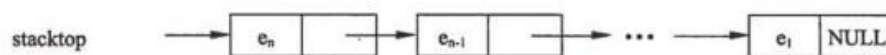


图 5-1 链栈示意图

### 【问题 1】

### 【C 函数】

```
int InOrder(BTree root)      /*实现二叉树的非递归中序遍历*/
{
    BTree ptr;               /*ptr 用于指向二叉树中的节点*/
    StNode *q;               /*q 暂存链栈中新创建或待删除的节点指针*/
    StNode *stacktop = NULL; /*初始化空栈的栈顶指针 stacktop*/
    ptr = root;              /*ptr 指向二叉树的根节点*/
    while ( __ (1) __ || stacktop != NULL) {
        while (ptr != NULL) {
            q = (StNode *)malloc(sizeof(StNode));
            if (q == NULL)
                return -1;
        }
    }
}
```



```

    q->elem = ptr;
    (2) ;
    stacktop = q;          /*stacktop 指向新的栈顶*/
    ptr = (3) ;           /*进入左子树*/
}

    q = stacktop;
    (4) ;                 /*栈顶元素出栈*/
    visit(q);             /*visit 是访问节点的函数，其具体定义省略*/
    ptr = (5) ;           /*进入右子树*/
    free(q);              /*释放原栈顶元素的节点空间*/
}
return 0;
}/*InOrder*/

```

- (1) ptr != NULL, 或 ptr !=0, 或 ptr
- (2) q->link = stacktop
- (3) ptr->lchild
- (4) stacktop = stacktop->link. 或 stacktop = q->link
- (5) q->elem->rchild

本题考查基本数据结构和 C 语言程序设计能力。

对非空二叉树进行中序遍历的方法是：先中序遍历根节点的左子树，然后访问根节点，最后中序遍历根节点的右子树。用递归方式描述的算法如下：

```

void In_Order_Traversing(BiTree root)
{ //root 是指向二叉树根节点的指针
    if (root != NULL) {
        In_Order_Traversing(root->LeftChild);
        visit(root);
        In_Order_Traversing(root->RightChild);
    }
}

```

从以上算法的执行过程可知，从树根出发进行遍历时，递归调用 In\_Order\_Traversing(root->LeftChild)使得遍历过程沿着左孩子分支一直走向下层节点，直到到达二叉树中最左下方的节点（设为 f）的空左子树为止，然后返回 f 节点，再由递归调用 In\_Order\_Traversing(root->RightChild)进入 f 的右子树，并重复以上过程。在递归算法执行过程中，辅助实现递归调用和返回处理的控制栈实际上起着保存从根节点到当前节点的路径信息。

用非递归算法实现二叉树的中序遍历时，可以由一个循环语句实现从指定的根节点出发，沿着左孩子分支一直到头（到达一个没有左子树的节点）的处理，从根节点到当前节点的路径信息（节点序列）可以明确构造一个栈来保存。



本题目的难点在于将栈的实现和使用混合在一起来处理，而且栈采用单链表存储结构。下面分析题中给出的代码。

空 (1) 是遍历的条件之一，由于另外一个条件 `stacktop != NULL` 初始时是不成立的，因此空 (1) 所表示的条件必须满足，由于是对非空二叉树进行遍历，显然该条件代表二叉树非空，即 `ptr != NULL` 或其等价表示形式。

临时指针 `ptr` 初始时指向整个二叉树的根节点，此后用以下代码表示一直沿左孩子指针链向下走的处理，临时指针 `q` 用于在链栈中加入新元素时使用。处理思路是：若当前节点有左子树，则将当前节点的指针存入栈中，然后进入当前节点的左子树。入栈时，先申请元素在链栈中的节点空间，然后设置节点数据域的值（即当前节点的指针），最后将新申请的节点加入链栈首部。

```
while (ptr != NULL) {
    q = (StNode *)malloc(sizeof(StNode)); /*为新入栈的元素创建节点*/
    if (q == NULL)                        /*若创建新节点失败，则退出*/
        return -1;
    q->elem = ptr;                        /*在栈顶保存指向当前节点的指针*/
    q->link = stacktop;                  /*新节点加入栈顶*/
    stacktop = q;                        /*更新栈顶指针，即 stacktop 指向新的栈顶*/
    ptr = ptr->lchild;                   /*进入当前节点的左子树*/
}
```

当上述过程进入一棵空的子树时（`ptr` 为空指针），循环结束。此后，应该从空的子树返回其父节点并进行访问。由于进入空的左子树前已将其父节点指针压入栈中，因此，栈顶元素即为该父节点，对应的处理就是弹栈。相应地，在链栈中要删除表头节点并释放节点空间：

```
q = stacktop;                          /*q 指向链栈中需要删除的节点，即栈顶元素*/
stack = stacktop->link;                 /*栈顶元素出栈*/
visit(q);                              /*访问节点*/
free(q);                               /*释放节点空间*/
```

由于还需要通过 `q` 指针进入被删除节点的右子树，因此，释放节点空间的操作 `free(q)` 操作之前，使 `ptr` 指向 `q` 所指节点的右子树指针，以得到被删除节点的数据域信息，即空 (5) 所在语句 `ptr = q->elem->rchild`

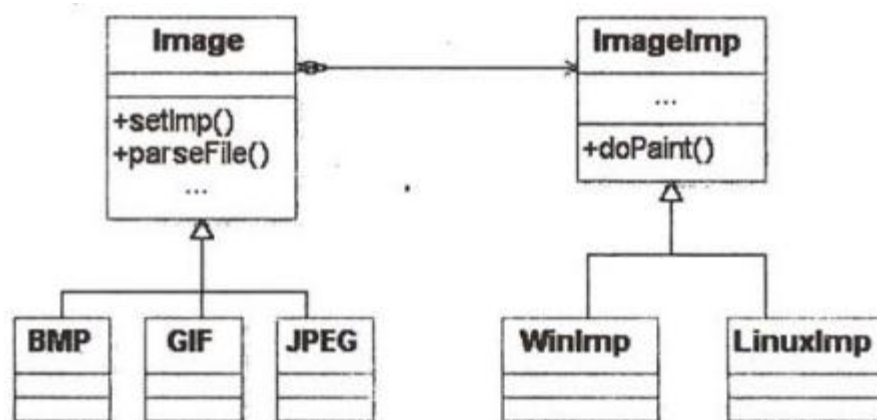
指针是 C 语言中灵活且非常强大的工具，是否熟练掌握 C 语言的判断条件之一就是对指针的理解和使用。软件设计师需要熟练掌握这些内容。

## 试题六

阅读下列说明和 C++ 代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

### 【说明】

现欲实现一个图像浏览系统，要求该系统能够显示 BMP、JPEG 和 GIF 三种格式的文件，并且能够在 Windows 和 Linux 两种操作系统上运行。系统首先将 BMP、JPEG 和 GIF 三种格式的文件解析为像素矩阵，然后将像素矩阵显示在屏幕上。系统需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，采用桥接 (Bridge) 设计模式进行设计，所得类图如下图所示。



类图

采用该设计模式的原因在于：系统解析 BMP、GIF 与 JPEG 文件的代码仅与文件格式相关，而在屏幕上显示像素矩阵的代码则仅与操作系统相关。

### 【C++代码】

```
class Matrix{ //各种格式的文件最终都被转化为像素矩阵
    //此处代码省略
};

class ImageImp{
public:
    virtual void doPaint(Matrix m) = 0; //显示像素矩阵 m
};

class WinImp : public ImageImp{
public:
    void doPaint(Matrix m){ /*调用 Windows 系统的绘制函数绘制像素矩阵*/ }
};
```

### 【问题 1】

```

class LinuxImp : public ImageImp{
public:
    void doPaint(Matrix m){ /*调用 Linux 系统的绘制函数绘制像素矩阵*/ }
};

class Image {
public:
    void setImp(ImageImp *imp){ (1) = imp;}
    virtual void parseFile(string fileName) = 0;
protected:
    (2) *imp;
};

class BMP : public Image{
public:
    void parseFile(string fileName){
        //此处解析 BMP 文件并获得一个像素矩阵对象 m
        (3) ;// 显示像素矩阵 m
    }
};

class GIF : public Image{
    //此处代码省略
};

class JPEG : public Image{
    //此处代码省略
};

void main(){
    //在 Windows 操作系统上查看 demo.bmp 图像文件
    Image *image1 = (4) ;
    ImageImp *imageImp1 = (5) ;
    (6) ;

    image1->parseFile("demo.bmp");

}

```

现假设该系统需要支持 10 种格式的图像文件和 5 种操作系统，不考虑类 Matrix，若采用桥接设计模式则至少需要设计(7) 个类。

- (1) this->imp
- (2) ImageImp
- (3) imp->doPaint(m)
- (4) newBMP()
- (5) new WinImp()
- (6) image 1 ->setImp(imageImp1)
- (7) 17

根据题目描述，在设计该图像显示系统时主要分为两个步骤：一是读取各种文件并将文件内容转换成像素矩阵，因为各种图片格式不同，因此需要针对每一种图片格式编写文件读取代码，而该代码与操作系统平台无关。将像素矩阵显示到屏幕上时，由于和操作系统相关，因此需要把该代码和读取文件代码相分离。设计中的 Image 类表示抽象的图像概念，Image 类中就包含了读取文件接口和设置实现平台接口；Image 的子类 BMP、GIF 和 JPEG 分别负责读取各种不同格式的文件；ImageImp 的主要任务是将像素矩阵显示在屏幕上，因此，它存在两个子类，分别实现 Windows 系统和 Linux 系统上的图像显示代码。空缺（1）处主要是设置将在哪个平台上进行实现，因此该处应该存储参数所传递的对象，由于该类的成员变量也是 imp，与参数相同，因此需要填写 this->imp；同理，该成员变量的类型和参数的类型应该保持相同，空（2）处应该填写 ImageImp；空（3）处需要根据 imp 成员变量存储的实现对象来显示图像；在空（4）处需要生成一个 BMP 对象；由于需要在 Windows 平台上实现，因此空（5）处需要生成一个 WinImp 对象，同时，还需设置该 BMP 对象，应采用 WinImp 对象来实现显示。采用桥接模式能够将文件分析代码和图像显示代码分解在不同的类层次结构中，如果不考虑中间使用的 Matrix 等类，那么最后需要设计的类包括 2 个父类，对应文件格式子类，对应操作系统平台类，因此 10 种图像格式和 5 种操作系统需要 17 个类。

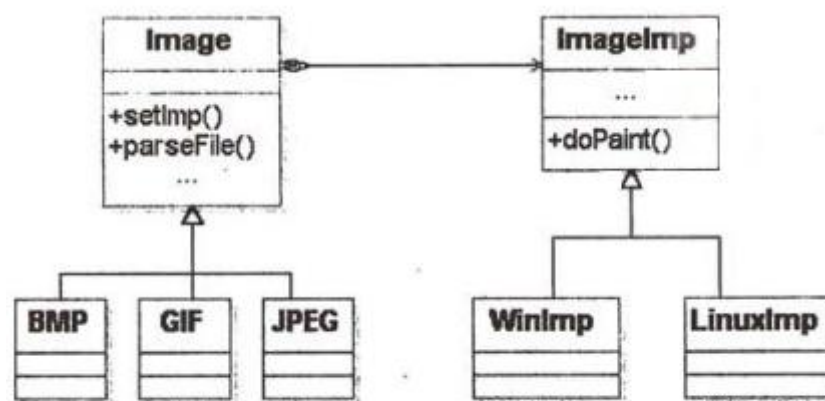


## 试题七

阅读下列说明和 Java 代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

### 【说明】

现欲实现一个图像浏览系统，要求该系统能够显示 BMP、JPEG 和 GIF 三种格式的文件，并且能够在 Windows 和 Linux 两种操作系统上运行。系统首先将 BMP、JPEG 和 GIF 三种格式的文件解析为像素矩阵，然后将像素矩阵显示在屏幕上。系统需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，采用桥接 (Bridge) 设计模式进行设计，所得类图如下图所示。



类图

采用该设计模式的原因在于：系统解析 BMP、GIF 与 JPEG 文件的代码仅与文件格式相关，而在屏幕上显示像素矩阵的代码则仅与操作系统相关。

### 【问题 1】

## 【Java 代码】

```
class Matrix{    //各种格式的文件最终都被转化为像素矩阵
    //此处代码省略
};

abstract class ImageImp{
    public abstract void doPaint(Matrix m);    //显示像素矩阵 m
};

class WinImp extends ImageImp{
    public void doPaint(Matrix m){            /*调用 Windows 系统的绘制函数绘制像素矩阵*/
};

class LinuxImp extends ImageImp{
    public void doPaint(Matrix m){/*调用 Linux 系统的绘制函数绘制像素矩阵*/
};

abstract class Image {
    public void setImp(ImageImp imp){
        (1) = imp; }
    public abstract void parseFile(String fileName);
    protected (2) imp;
};

class BMP extends Image{
    public void parseFile(String fileName){
        //此处解析 BMP 文件并获得一个像素矩阵对象 m
        (3); // 显示像素矩阵 m
    }
};

class GIF extends Image{
    //此处代码省略
};

class JPEG extends Image{
    //此处代码省略
};

public class javaMain{
    public static void main(String[] args){
        //在 Windows 操作系统上查看 demo.bmp 图像文件
    }
}
```

```

        Image image1 = ____ (4) ____;
        ImageImp imageImp1 = ____ (5) ____;
        ____ (6) ____;
        image1.parseFile("demo.bmp");
    }
}

```

- (1) this.imp
- (2) ImageImp
- (3) imp.doPaint(m)
- (4) newBMP()
- (5) new WinImp()
- (6) image1.setImp(imageImp1)
- (7) 17

根据题目描述，在设计该图像显示系统时主要分为两个步骤：一是读取各种文件并将文件内容转换为像素矩阵，因为各种图片格式不同，因此需要针对每一种图片格式编写文件读取代码，而该代码与操作系统平台无关。将像素矩阵显示到屏幕上时，由于和操作系统相关，因此需要把该代码和读取文件代码相分离。设计中的 Image 类表示抽象的图像概念，Image 类中就包含了读取文件接口和设置实现平台接口；Image 的子类 BMP、GIF 和 JPEG 分别负责读取各种不同格式的文件；ImageImp 的主要任务是将像素矩阵显示在屏幕上，因此，它存在两个子类，分别实现 Windows 系统和 Linux 系统上的图像显示代码。空缺 (1) 处主要是设置将在哪个平台上进行实现，因此该处应该存储参数所传递的对象，由于该类的成员变量也是 imp，与参数相同，因此需要填写 this.imp；同理，该成员变量的类型和参数的类型应该保持相同，空 (2) 处应该填写 ImageImp；空 (3) 处需要根据 imp 成员变量存储的实现对象来显示图像；在空 (4) 处需要生成一个 BMP 对象；由于需要在 Windows 平台上实现，因此空 (5) 处需要生成一个 WinImp 对象，同时，还需设置该 BMP 对象，应采用 WinImp 对象来实现显示。采用桥接模式能够将文件分析代码和图像显示代码分解在不同的类层次结构中，如界不考虑中间使用的 Matrix 等类，那么最后需要设计的类包括 2 个父类，对应文件格式 & 目的子类，对应操作系统数目的平台类，因此 10 种图像格式和 5 种操作系统需要 17 个类。