

常用的虚拟存储器由\_(1)\_两级存储器组成。

- (1) A. 主存-辅存      B. 主存-网盘      C. Cache-主存      D. Cache-硬盘

**【答案】A**

**【解析】**本题考查计算机系统存储系统基础知识。

在具有层次结构存储器的计算机中,虚拟存储器是为用户提供一个比主存储器大得多的可随机访问的地址空间的技术。虚拟存储技术使辅助存储器和主存储器密切配合, 对用户来说,好像计算机具有一个容量比实际主存大得多的主存可供使用,因此称为虚拟存储器。虚拟存储器的地址称为虚地址或逻辑地址。

中断向量可提供\_(2)\_。

- (2) A. I/O 设备的端口地址      B. 所传送数据的起始地址  
C. 中断服务程序的入口地址      D. 主程序的断点地址

**【答案】C**

**【解析】**本题考查计算机系统基础知识。

计算机在执行程序过程中,当遇到急需处理的事件时,暂停当前正在运行的程序, 转去执行有关服务程序,处理完后自动返回原程序,这个过程称为中断。

中断是一种非常重要的技术,输入输出设备和主机交换数据、分时操作、实时系统、计算机网络和分布式计算机系统中都要用到这种技术。为了提高响应中断的速度,通常把所有中断服务程序的入口地址(或称为中断向量)汇集为中断向量表。

为了便于实现多级中断嵌套,使用\_(3)\_来保护断点和现场最有效。

- (3) A. ROM      B. 中断向量表      C. 通用寄存器      D. 堆栈

**【答案】D**

**【解析】**本题考查计算机系统基础知识。

当系统中有多个中断请求时,中断系统按优先级进行排队。若在处理低级中断过程中又有高级中断申请中断,则高级中断可以打断低级中断处理,转去处理高级中断,等处理完高级中断后再返回去处理原来的低级中断,称为中断嵌套。实现中断嵌套用后进先出的栈来保护断点和现场最有效。

DMA 工作方式下,在\_(4)\_之间建立了直接的数据通路。

- (4) A. CPU 与外设      B. CPU 与主存      C. 主存与外设      D. 外设与外设

**【答案】C**

**【解析】**本题考查计算机系统基础知识。

计算机系统中主机与外设间的输入输出控制方式有多种，在 DMA 方式下，输入输出设备与内存直接相连，数据传送由 DMA 控制器而不是主机 CPU 控制。CPU 除了传送开始和结束时进行必要的处理外，不参与数据传送的过程。

地址编号从 80000H 到 BFFFFH 且按字节编址的内存容量为 (5) KB, 若用 16K\*4bit 的存储器芯片构成该内存，共需 (6) 片。

- (5) A. 128      B. 256      C. 512      D. 1024  
(6) A. 8      B. 16      C. 32      D. 64

**【答案】B C**

**【解析】**本题考查计算机系统基础知识。

从 80000H 到 BFFFFH 的编址单元共 3FFFF (即 218 )个，按字节编址的话，对应的容量为 28KB，即 256KB。若用 16K\*4bit 的芯片构成该内存，构成一个 16KB 存储器需要 2 片， $256 \div 16 = 16$ ，因此共需要 32 片。

利用报文摘要算法生成报文主要的目的是 (7)。

- (7) A. 验证通信对方的身份，防止假冒      B. 对传输数据进行加密，防止数据被窃听  
C. 防止发送方否认发送过数据      D. 防止发送的报文被篡改

**【答案】D**

**【解析】**本题考查报文摘要的知识

报文摘要是指单向哈希函数算法将任意长度的输入报文经计算得出固定位的输出。报文摘要是用来保证数据完整性的。传输的数据一旦被修改，那么计算出的摘要就不同，只要对比两次摘要就可确定数据是否被修改过。

防火墙通常分为内网、外网和 DMZ 三个区域，按照受保护程序，从高到低正确的排列次序为 (8)。

- (8) A. 内网、外网和 DMZ      B. 外网、内网和 DMZ  
C. DMZ、内网和外网      D. 内网、DMZ 和外网

**【答案】D**

**【解析】**本题考查防火墙的基础知识。

通过防火墙我们可以将网络划分为三个区域：安全级别最高的 LAN Area（内网），安全级别中等的 DMZ 区域和安全级别最低的 Internet 区域（外网）。三个区域因担负不同的任务而拥有不同的访问策略。通常的规则如下：

- ①内网可以访问外网：内网的用户需要自由地访问外网。在这一策略中，防火墙需要执行 NAT。
- ②内网可以访问 DMZ：此策略使内网用户可以使用或者管理 DMZ 中的服务器。
- ③外网不能访问内网：这是防火墙的基本策略，内网中存放的是公司内部数据，显然这些数据是不允许外网的用户进行访问的。如果要访问，就要通过 VPN 方式来进行。
- ④外网可以访问 DMZ：DMZ 中的服务器需要为外界提供服务，所以外网必须可以访问 DMZ。同时，外网访问 DMZ 需要由防火墙完成对外地址到服务器实际地址的转换。
- ⑤DMZ 不能访问内网：如不执行此策略，则当入侵者攻陷 DMZ 时，内部网络将不会受保护。
- ⑥DMZ 不能访问外网：此条策略也有例外，可以根据需要设定某个特定的服务器可以访问外网，以保证该服务器可以正常工作。

综上所述，防火墙区域按照受保护程度从高到低正确的排列次序应为内网、DMZ 和外网。

近年来，在我国出现各类病毒中，(9) 病毒通过木马形式感染智能手机。

- (9) A. 欢乐时光                      B. 熊猫烧香                      C. X 卧底                      D. CIH

**【答案】C**

**【解析】**本题考查病毒及其危害。

欢乐时光及熊猫烧香均为蠕虫病毒，CIH 则为系统病毒，这 3 者均以感染台式机或服务器为主，且产生较早；X 卧底则是新近产生的、通过木马形式传播、目标为智能手机的病毒。

王某是一名软件设计师，按公司规定编写软件文档，并上交公司存档。这些软件文档属于职务作品，且(10)。

- (10) A. 其著作权由公司享有
- B. 其著作权由软件设计师享有
- C. 除其署名权以外，著作权的其他权利由软件设计师享有
- D. 其著作权由公司和软件设计师共同享有

**【答案】A**

**【解析】**本题考查知识产权知识。

公民为完成法人或者其他组织工作任务所创作的作品是职务作品。职务作品可以是作品分类中的任何一种形式，如文字作品、电影作品、计算机软件等。职务作品的著作权归属分两种情形。

一般职务作品的著作权由作者享有。所谓一般职务作品是指虽是为完成工作任务而为，但非经法人或其他组织主持，不代表其意志创作，也不由其承担责任的职务作品。对于一般职务作品，法人或其他组织享有在其业务范围内优先使用的权利，期限为两年。优先使用权是专有的，未经单位同意，作者不得许可第三人以与法人或其他组织使用的相同方式使用该作品。在作品完成两年内，如单位在其业务范围内不使用，作者可以要求单位同意由第三人以与法人或其他组织使用的相同方式使用，所获报酬，由作者与单位按约定的比例分配。特殊的职务作品，除署名权以外，著作权的其他权利由法人或者其他组织（单位）享有。所谓特殊职务作品是指著作权法第16条第2款规定的两种情况：一是主要利用法人或者其他组织的物质技术条件创作，并由法人或者其他组织承担责任的工程设计、产品设计图、计算机软件、地图等科学技术作品；二是法律、法规规定或合同约定著作权由单位享有的职务作品。

甲经销商擅自复制并销售乙公司开发的OA软件光盘已构成侵权。丙企业在未知的情形下从甲经销商处购入10张并已安装使用。在丙企业知道了所使用的软件为侵权复制的情形下，以下说法正确的是(11)。

(11)A. 丙企业的使用行为侵权，须承担赔偿责任

B. 丙企业的使用行为不侵权，可以继续使用这10张软件光盘

C. 丙企业的使用行为侵权，支付合理费用后可以继续使用这10张软件光盘

D. 丙企业的使用行为不侵权，不需承担任何法律责任

**【答案】C**

**【解析】**本题考查知识产权知识。

我国计算机软件保护条例第30条规定“软件的复制品持有人不知道也没有合理理由应当知道该软件是侵权复制品的，不承担赔偿责任；但是，应当停止使用、销毁该侵权复制品。如果停止使用并销毁该侵权复制品将给复制品使用人造成重大损失的，复制品使用人可以在向软件著作权人支付合理费用后继续使用。”丙企业在获得软件复制品的形式上是合法的（向

经销商购买),但是由于其没有得到真正软件权利人的授权,其取得的复制品仍是非法的,所以丙企业的使用行为属于侵权行为。

丙企业应当承担的法律责任种类和划分根据主观状态来确定。首先,法律确立了著作权人的权利进行绝对的保护原则,即软件复制品持有人不知道也没有合理理由应当知道该软件是侵权复制品的,也必须承担停止侵害的法律责任,只是在停止使用并销毁该侵权复制品将给复制品使用人造成重大损失的情况下,软件复制品使用人可继续使用,但前提是必须向软件著作权人支付合理费用。其次,如果软件复制品持有人能够证明自己确实不知道并且也没有合理理由应当知道该软件是侵权复制品的,软件复制品持有人除承担停止侵害外,不承担赔偿责任。

软件复制品持有人一旦知道了所使用的软件为侵权复制品时,应当履行停止使用、销毁该软件的义务。不履行该义务,软件著作权人可以诉请法院判决停止使用并销毁侵权软件。如果软件复制品持有人在知道所持有软件是非法复制品后继续使用给权利人造成损失的,应当承担赔偿责任。

声音信号数字化过程中首先要进行\_(12)。

- (12)A. 解码                      B. D/A 转换                      C. 编码                      D. A/D 转换

**【答案】D**

**【解析】**本题考查多媒体基础知识。

声音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声音信号,即用二进制数字的编码形式来表示声音,通常将这一过程称为数字化过程。声音信号数字化过程中首先是将模拟信号转换成离散信号,即 A/D 转换(模数转换)。

以下关于 dpi 的叙述中,正确的是\_(13)。

- (13)A. 每英寸的 bit 数                      B. 存储每个像素所用的位数  
C. 每英寸像素点                      D. 显示屏上能够显示出的像素数目

**【答案】C**

**【解析】**本题考查多媒体基础知识。

dpi 是描述图像分辨率的单位,表示每英寸多少像素点,即组成一幅图像的像素密度。它实质上是图像数字化的采样间隔,由它确立组成一幅图像的像素数目。对同样大小的一幅图,如果组成该图像的图像像素数目越多,则说明图像的分辨率越高,图像看起来就越逼真。

相反，图像则显得越粗糙。因此，不同的分辨率会造成不同的图像清晰度。存储每个像素所用的位数是用来度量图像的分辨率的。像素深度确定彩色图像的每个像素可能有的颜色数，即确定彩色图像中可出现的最多颜色数。显示屏上能够显示出的像素数目是指显示分辨率

媒体可以分为感觉媒体、表示媒体、表现媒体、存储媒体、传输媒体，(14)属于表现媒体。

- (14) A. 打印机                      B. 硬盘                      C. 光缆                      D. 图像

**【答案】A**

**【解析】**本题考查多媒体基础知识。

表现媒体指实现信息输入和输出的媒体，如键盘、鼠标、扫描仪、话筒、摄像机等为输入媒体；显示器、打印机、喇叭等为输出媒体。硬盘属于存储媒体；光缆属于传输媒体；图像属于感觉媒体

统一过程模型是一种“用例和风险驱动，以架构为中心，迭代并且增量”的开发过程，定义了不同阶段及其制品，其中精化阶段关注(15)。

- (15) A. 项目的初始活动                      B. 需求分析和架构演进  
C. 系统的构建，产生实现模型                      D. 软件提交方面的工作，产生软件增量

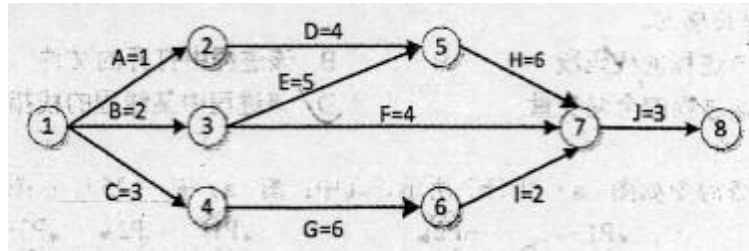
**【答案】B**

**【解析】**本题考查软件开发过程模型的基本概念。

统一过程模型是一种“用例和风险驱动，以架构为中心，迭代并且增量”的开发过程，由UML方法和工具支持，定义了不同阶段及其制品。

起始阶段专注于项目的初创活动。精化阶段理解了最初的领域范围之后，进行需求分析和架构演进。构建阶段关注系统的构建，产生实现模型。移交阶段关注于软件提交方面的工作，产生软件增量。产生阶段运行软件并监控软件的持续使用，提供运行环境的支持，提交并评估缺陷报告和变更请求。

在进行进度安排时，PERT图不能清晰地描述(16)，但可以给出哪些任务完成后才能开始另一些任务。某项目X包含任务A、B、...、J，其PERT图如下图所示（A=1表示任务A的持续时间是1天），则项目X的关键路径是(17)。



- (16) A. 每个任务从何时开始                      B. 每个任务到何时结束  
C. 各任务之间的并行情况                      D. 各任务之间的依赖关系

- (17) A. A-D-H-J                      B. B-E-H-J                      C. B-F-J                      D. C-G-I-J

**【答案】C B**

**【解析】** 本题考查项目管理及工具技术。

PERT 图可以清晰地表示各任务的开始时间和结束时间以及各任务之间的依赖关系，但是无法很好地表示各任务之间的并行情况。

根据关键路径法，计算出项目 X 中的关键路径为 B-E-H-J，关键路径长度为 16。

“软件产品必须能够在 3 秒内对用户请求作出响应”属于软件需求中的 (18)。

- (18) A. 功能需求                      B. 非功能需求                      C. 设计约束                      D. 逻辑需求

**【答案】B**

**【解析】** 本题考查软件需求分类基础知识。

软件需求是软件系统必须完成的事以及必须具备的品质。软件需求包括功能需求、非功能需求和设计约束三个方面的内容。功能需求是所开发的软件必须具备什么样的功能；非功能需求是指产品必须具备的属性或品质，如可靠性、性能、响应时间和扩展性等等；设计约束通常对解决方案的一些约束说明。“软件产品必须能够在 3 秒内对用户请求作出响应”主要表述软件的响应时间，属于非功能需求。

某项目为了修正一个错误而进行了修改。错误修正后，还需要进行 (19) 以发现这一修正是否引起原本正确运行的代码出错。

- (19) A. 单元测试                      B. 接受测试                      C. 安装测试                      D. 回归测试

**【答案】D**

**【解析】** 本题考查软件测试基础知识。

单元测试是在模块编写完成且无编译错误后进行，侧重于模块中的内部处理逻辑和数据结构；接受测试主要是用户为主的测试；安装测试是将软件系统安装在实际运行环境的测试；

回归测试是在系统有任何修改的情况下，需要重新对整个软件系统进行的测试。

以下关于解释程序和编译程序的叙述中，正确的是(20)。

- (20) A. 编译程序和解释程序都生成源程序的目标程序  
B. 编译程序和解释程序都不生成源程序的目标程序  
C. 编译程序生成源程序的目标程序，而解释程序则不然  
D. 编译程序不生成源程序的目标程序，而解释程序反之

**【答案】C**

**【解析】**本题考查程序语言翻译基础知识。

编译和解释方式是翻译高级程序设计语言的两种基本方式。

解释程序也称为解释器，它或者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。这两种语言处理程序的根本区别是：在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序（或其某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释器。解释器翻译源程序时不产生独立的目标程序，而编译器则需将源程序翻译成独立的目标程序。

以下关于传值调用与引用调用的叙述中，正确的是(21)。

- ①在传值调用方式下，可以实现形参和实参间双向传递数据的效果  
②在传值调用方式下，实参可以是变量，也可以是常量和表达式  
③在引用调用方式下，可以实现形参和实参间双向传递数据的效果  
④在引用调用方式下，实参可以是变量，也可以是常量和表达式

- (21) A. ①③                      B. ①④                      C. ②③                      D. ②④

**【答案】C**

**【解析】**本题考查程序语言知识。

调用函数和被调用函数之间交换信息的方法主要有两种：一种是由被调用函数把返回值返回给主调函数，另一种是通过参数传递信息。函数调用时实参与形参间交换信息的基本方法有传值调用和引用调用两种。

若实现函数调用时实参向形参传递相应类型的值，则称为是传值调用。这种方式下形参



不能向实参传递信息。实参可以是变量，也可以是常量和表达式。

引用调用的实质是将实参变量的地址传递给形参，因此，形参是指针类型，而实参必须具有左值。变量具有左值，常量没有左值。被调用函数对形参的访问和修改实际上就是针对相应实际参数所作的访问和改变，从而实现形参和实参间双向传递数据的效果。

在对高级语言源程序进行编译的过程中，为源程序中变量所分配的存储单元的地址属于(22)。

- (22) A. 逻辑地址      B. 物理地址      C. 接口地址      D. 线性地址

**【答案】A**

**【解析】**本题考查程序语言基础知识。

编译过程中为变量分配存储单元所用的地址是逻辑地址，程序运行时再映射为物理地址。

假设某分时系统采用简单时片轮转法，当系统中的用户数为  $n$ 、时间片为  $q$  时，系统对每个用户的响应时间  $T=(23)$ 。

- (23) A.  $n$       B.  $q$       C.  $n*q$       D.  $n+q$

**【答案】C**

**【解析】**

在分时系统中是将把 CPU 的时间分成很短的时间片轮流地分配给各个终端用户，当系统中的用户数为  $n$ 、时间片为  $q$  时，那么系统对每个用户的响应时间等于  $n*q$ 。

在支持多线程的操作系统中，假设进程 P 创建了若干个线程，那么(24)是不能被这些线程共享的。

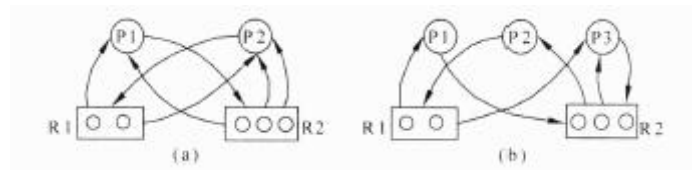
- (24) A. 该进程的代码段      B. 该进程中打开的文件  
C. 该进程的全局变量      D. 该进程中某线程的栈指针

**【答案】D**

**【解析】**

在同一进程中的各个线程都可以共享该进程所拥有的资源，如访问进程地址空间中的每一个虚地址；访问进程拥有已打开文件、定时器、信号量机构等，但是不能共享进程中某线程的栈指针。

进程资源图如图(a)和(b)所示, 其中: 图(a)中 (25); 图(b)中 (26)。



(25) A. P1 是非阻塞节点, P2 是阻塞节点, 所以该图不可以化简, 是死锁的

B. P1、P2 都是阻塞节点, 所以该图不可以化简, 是死锁的

C. P1、P2 都是非阻塞节点, 所以该图可以化简, 是非死锁的

D. P1 是阻塞节点, P2 是非阻塞节点, 所以该图不可以化简、是死锁的

(26) A. P1、P2、P3 都是非阻塞节点, 该图可以化简, 是非死锁的

B. P1、P2、P3 都是阻塞节点, 该图不可以化简, 是死锁的

C. P2 是阻塞节点, P1、P3 是非阻塞节点, 该图可以化简, 是非死锁的

D. P1、P2 是非阻塞节点, P3 是阻塞节点, 该图不可以化简, 是死锁的。

**【答案】B C**

**【解析】**

因为 R1 资源只有 2 个, P2 申请该资源得不到满足, 故进程 P2 是阻塞节点; 同样 R2 资源只有 3 个, P1 申请该资源得不到满足, 故进程 P1 也是阻塞节点。可见进程资源图 (a) 是死锁的, 该图不可以化简。

因为 R2 资源有 3 个, 已分配 2 个, P3 申请 1 个 R2 资源可以得到满足, 故进程 P3 可以运行完毕释放其占有的资源。这样可以使得 P1、P2 都变为非阻塞节点, 得到所需资源运行完毕, 因此, 进程资源图 (b) 是可化简的。

假设内存管理采用可变式分区分配方案, 系统中有五个进程 P1~P5, 且某一时刻内存使用情况如下图所示 (图中空白处表示未使用分区)。此时, 若 P5 进程运行完并释放其占有的空间, 则释放后系统的空闲区数应 (27); 造成这种情况的原因是 (28)。

分区号	进程
0	P1
1	P2
2	
3	P4
4	P3
5	
6	P5
7	

- (27) A. 保持不变                      B. 减 1                      C. 加 1                      D. 置零
- (28) A. 无上邻空闲区，也无下邻空闲区                      B. 有上邻空闲区，但无下邻空闲区
- C. 有上邻空闲区，但无下邻空闲区                      D. 有上邻空闲区，也有下邻空闲区

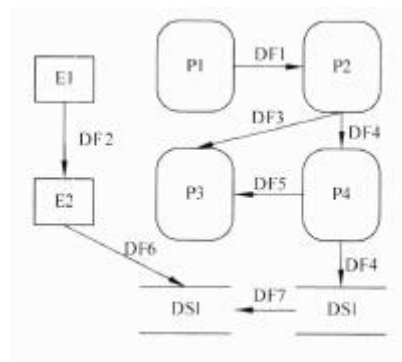
**【答案】B D**

**【解析】**

(27) 从图中不难看出，若 p5 进程运行完并释放其占有的空间，则由于其占用的分区有上邻空闲区，一旦释放后，就合并为一个空闲区，所以合并后系统空闲区数=3-1=2.

(28) 若 p5 进程运行完并释放其占有的空间，则由于其占用的分区既有上邻空闲区又有下邻空闲区，一旦释放后，就合并为一个空闲区，所以合并后系统空闲区数=3-1=2

在如下所示的数据流图中，共存在(29)个错误。



- (29) A. 4                      B. 6                      C. 8                      D. 9

**【答案】B**

**【解析】** 本题考查数据流图方面的基础知识。

结构化分析将数据和处理作为分析对象，数据的分析结果表示了现实世界中实体的属性及其之间的相互关系，而处理的结果则展现了系统对数据的加工和转换。面向数据流建模是目前仍然被广泛使用的方法之一，而 DFD 则是面向数据流建模中的重要工具，DFD 将系统建

模成输入—处理—输出的模型，即流入软件的数据对象，经由处理的转换，最后以结果数据对象的形式流出软件。在实际使用 DFD 进行数据流建模时，需要注意以下原则：

①加工处理和数据流的正确使用，如一个加工必须既有输入又有输出；数据流只能和加工相关，即从加工流向加工、数据源流向加工或加工流向数据源。

②每个数据流和数据存储都要在数据字典中有定义，数据字典将包括各层数据流图中数据元素的定义。

③数据流图中最底层的加工处理必须有加工处理说明。

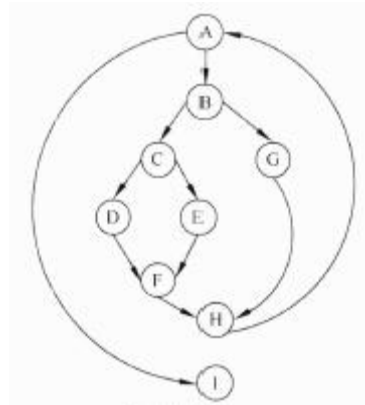
④父图和子图必须平衡，即父图中某加工的输入输出（数据流）和分解这个加工的子图的输入输出数据流必须完全一致，这种一致性不一定要求数据流的名称和个数一一对应，但它们在数据字典中的定义必须一致，数据流或数据项既不能多也不能少。

⑤加工处理说明和数据流图中加工处理涉及的元素保持一致。例如，在加工处理说明中，输入数据流必须说明其如何使用，输出数据流说明如何产生或选取，数据存储说明如何选取、使用或修改。

⑥一幅图中的图元个数控制在 7+2 以内。

在题目所示的 DFD 图中，数据流 DF2、DF6 和 DF7 的输入、输出均不是加工，这与“数据流只能和加工相关，即从加工流向加工、数据源流向加工或加工流向数据源”相违背。加工 P1 只有输出，没有输入；加工 P3 只有输入没有输出，这与“一个加工必须既有输入又有输出”相违背。数据流 DF4 经过加工 P4 之后没有发生任何改变，说明该数据对加工 P4 是没有作用的，根据数据守恒原理，这条数据流不应与 P4 有关联。综上，该 DFD 图中共有 6 个错误。

软件的复杂性主要体现在程序的复杂性。(30)是度量软件复杂性的一个主要参数。若采用 McCabe 度量法计算环路复杂性，则对于下图所示的程序图，其环路复杂度为(31)。



(30) A. 代码行数      B. 常量的数量      C. 变量的数量      D. 调用的库函数的数量

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

【答案】A C

【解析】本题考查软件复杂性方面的基础知识。

软件复杂性度量是软件度量的一个重要分支。对于软件复杂性度量的主要参数有：

- 规模，即总共的指令数，或源程序行数。
- 难度，通常由程序中出现的操作数的数目所决定的量来表示。
- 结构，通常用与程序结构有关的度量来表示。
- 智能度，即算法的难易程度。

软件复杂性主要表现在程序的复杂性。程序的复杂性主要指模块内程序的复杂性。

McCabe 度量法是一种基于程序控制流的复杂性度量方法。McCabe 复杂性度量又称为环路度量，它认为程序的复杂性很大程度上取决于控制的复杂性。单一的顺序程序结构最为简单，循环和选择所构成的环路越多，程序就越复杂。这种方法以图论为工具，先画出程序图，然后用该图的环路数作为程序复杂性的度量值。程序图是退化的程序流程图，也就是说，把程序流程图中每个处理符号都退化成一个结点，原来连接不同处理符号的流线变成连接不同点的有向弧，这样得到的有向图就叫做程序图。程序图仅描述程序内部的控制流程，完全不表现对数据的具体操作以及分支和循环的具体条件。

根据图论，在一个强连通的有向图  $G$  中，环的个数  $V(G)$  由以下公式给出：

$$V(G) = m - n + 2p$$

其中， $V(G)$  是有向图  $G$  中的环路数， $m$  是图  $G$  中弧的个数， $n$  是图  $G$  中的结点数， $P$  是  $G$  中的强连通分量个数。在一个程序中，从程序图的入口点总能到达图中的任何一个结点，因此，程序总是连通的，但不是强连通的。为了使程序图成为强连通图，从图的入口点到出口点加一条用虚线表示的有向边，使图成为强连通图，这样就可以使用上式计算环路复杂性了。对于题目中的程序图，其中结点数  $n = 9$ ，弧数  $m=11$ ， $p=1$ ，则有：

$$V(G) = m - n + 2p = 11 - 9 + 2 = 4$$

即 McCabe 环路复杂的度量值为 4。

(32) 不属于软件设计质量评审。

- (32) A. 功能与模块之间的对应关系      B. 软件规格说明是否符合用户的要求  
C. 软件是否具有可测试性      D. 软件是否具有有良好的可靠性

**【答案】A**

**【解析】**本题考查软件评审方面的基础知识。

通常，把“质量”理解为“用户满意程度”。为了使得用户满意，有两个必要条件：

- ①设计的规格说明书符合用户的要求，这称为设计质量；
- ②程序按照设计规格说明所规定的情况正确执行，这称为程序质量。设计质量评审的对象是在需求分析阶段产生的软件需求规格说明、数据需求规格说明，在软件概要设计阶段产生的软件概要设计说明书等。通常从以下几个方面进行评审：
  - ①评价软件的规格说明是否合乎用户的要求，即总体设计思想和设计方针是否明确；需求规格说明是否得到了用户或单位上级机关的批准；需求规格说明与软件的概要设计规格说明是否一致等。
  - ②评审可靠性，即是否能避免输入异常（错误或超载等）、硬件失效及软件失效所产生的失效，一旦发生应能及时采取代替手段或恢复手段。
  - ③评审保密措施实现情况，即是否对系统使用资格进行检查；是否对特定数据、特定功能的使用资格进行检查；在检查出有违反使用资格的情况后，能否向系统管理人员报告有关信息；是否提供对系统内重要数据加密的功能等。
  - ④评审操作特性实施情况，即操作命令和操作信息的恰当性，输入数据与输入控制语句的恰当性；输出数据的恰当性；应答时间的恰当性等。
  - ⑤评审性能实现情况，即是否达到所规定性能的目标值。
  - ⑥评审软件是否具有可修改性，可扩充性、可互换性和可移植性。
  - ⑦评审软件是否具有可测试性。
  - ⑧评审软件是否具有复用性。

在软件维护中，由于企业的外部市场环境和管理需求的变化而导致的维护工作，属于(33)维护。

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

**【答案】B**

**【解析】**本题考查软件维护方面的基础知识。

软件维护主要是指根据需求变化或硬件环境的变化对应用程序进行部分或全部的修改。修改时应充分利用源程序，修改后要填写程序修改登记表，并在程序变更通知书上写明新老程序的不同之处。

软件维护的内容一般有以下几个方面：

①正确性维护，是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。这方面的维护工作量要占整个维护工作量的 17%~21%。所发现的错误有的不太重要，不影响系统的正常运行，其维护工作可随时进行；而有的错误非常重要，甚至影响整个系统的正常运行，其维护工作必须制定计划，进行修改，并且要进行复查和控制。

②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。这方面的维护工作量占整个维护工作量的 18%~25%。由于目前计算机硬件价格的不断下降，各类系统软件层出不穷，人们常常为改善系统硬件环境和运行环境而产生系统更新换代的需求；企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。这些因素都将导致适应性维护工作的产生。进行这方面的维护工作也要像系统开发一样，有计划、有步骤地进行。

③完善性维护，这是为扩充功能和改善性能而进行的修改，主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。这些功能对完善系统功能是非常必要的。另外还包括对处理效率和编写程序的改进，这方面的维护占整个维护工作的 50%~60%，比重较大，也是关系到系统开发质量的重要方面。这方面的维护除了要有计划、有步骤地完成外，还要注意将相关的文档资料加入到前面相应的文档中去。

④预防性维护，为了改进应用软件的可靠性和可维护性，以及适应未来的软硬件环境的变化，应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被淘汰。比如将专用报表功能改成通用报表生成功能，以适应将来报表格式的变化。这方面的维护工作量占整个维护工作量的 4%左右。

在对软件系统进行评价时，需要从信息系统的组成部分、评价对象和经济学角度出发进行综合考虑以建立起一套指标体系理论架构。从信息系统评价对象出发，对于用户方来说，他们所关心的是(34)。

- |                   |              |
|-------------------|--------------|
| (34) A. 用户需求和运行质量 | B. 系统外部环境    |
| C. 系统内部结构         | D. 系统质量和技术水平 |

**【答案】A**

**【解析】**本题考查系统评价方面的基础知识。

在对软件系统进行评价时，需要从信息系统的组成部分、评价对象和经济学角度出发进行综合考虑，以建立起一套指标体系理论架构。

从信息系统的组成部分出发，信息系统是一个由人机共同组成的系统，所以可以按照运行效果和用户需求（人）、系统质量和技术条件（机）这两条线索构造指标。

从信息系统评价对象出发，对于用户方来说，他们所关心的是用户需求和运行质量；对开发方而言，他们所关心的是系统质量和技术水平。系统外部环境则主要通过社会效益指标来反映。

从经济学角度出发，分别按系统成本、系统效益和财务指标等 3 条线索建立指标。

在设计测试用例时，应遵循 (35) 原则。

- (35)A. 仅确定测试用例的输入数据，无需考虑输出结果
- B. 只需检验程是否执行应有的功能，不需要考虑程序是否做了多余的功能
- C. 不仅要设计有效合理输入，也要包含不合理、失效的输入
- D. 测试用例应设计得尽可能复杂

**【答案】C**

**【解析】**本题考查系统测试方面的基础知识。

系统测试是保证系统质量和可靠性的关键步骤，是对系统开发过程中的系统分析、系统设计和实施的最后复查。根据测试的概念和目的，在进行信息系统测试时应遵循以下基本原则：

①应尽早并不断地进行测试。测试不是在应用系统开发完之后才进行的。由于原始问题的复杂性、开发各阶段的多样性以及参加人员之间的协调等因素，使得在开发各个阶段都有可能出现错误。因此，测试应贯穿在开发的各个阶段，尽早纠正错误，消除隐患。

②测试工作应该避免由原开发软件的人或小组承担，一方面，开发人员往往不愿否认自己的工作，总认为自己开发的软件没有错误；另一方面，开发人员的错误很难由本人测试出来，很容易根据自己编程的思路来制定测试思路，具有局限性。测试工作应由专门人员进行，会更客观，更有效。

③设计测试方案的时候，不仅要确定输入数据，而且要根据系统功能确定预期输出结果。将实际输出结果与预期结果相比较就能发现测试对象是否正确。

④在设计测试用例时，不仅要设计有效合理的输入条件，也要包含不合理、失效的输入条件。测试的时候，人们往往习惯按照合理的、正常的情况进行测试，而忽略了对异常、不合理、意想不到的情况进行测试，而这些可能就是隐患。

⑤在测试程序时，不仅要检验程序是否做了该做的事，还要检验程序是否做了不该做的事。多余的工作会带来副作用，影响程序的效率，有时会带来潜在的危害或错误。



⑥严格按照测试计划来进行，避免测试的随意性。测试计划应包括测试内容、进度安排、人员安排、测试环境、测试工具和测试资料等。严格地按照测试计划可以保证进度，使各方面都得以协调进行。

⑦妥善保存测试计划、测试用例，作为软件文档的组成部分，为维护提供方便。

⑧测试用例都是精心设计出来的，可以为重新测试或追加测试提供方便。当纠正错误、系统功能扩充后，都需要重新开始测试，而这些工作重复性很高，可以利用以前的测试用例，或在其基础上修改，然后进行测试。

单元测试中，检查模块接口时，不需要考虑(36)。

(36)A. 测试模块的输入参数和形式参数的个数、属性、单位上是否一致

B. 全局变量在各模块中的定义和用法是否一致

C. 输入是否改变了形式参数

D. 输入参数是否使用了尚未赋值或者尚未初始化的变量

**【答案】D**

**【解析】**本题考查单元测试方面的基础知识。

单元测试也称为模块测试，在模块编写完成且无编译错误后就可以进行。单元测试侧重于模块中的内部处理逻辑和数据结构。单元测试主要检查模块的以下 5 个特征：模块接口、局部数据结构、重要的执行路径、出错处理和边界条件。

①模块接口。模块的接口保证了测试模块的数据流可以正确地流入、流出。在测试中应检查以下要点：

- 测试模块的输入参数和形式参数在个数、属性、单位上是否一致。
- 调用其他模块时所给出的实际参数和被调用模块的形式参数在个数、属性、单位上是否一致。
- 调用标准函数时所用的参数在属性、数目和顺序上是否正确。
- 全局变量在各模块中的定义和用法是否一致。
- 输入是否仅改变了形式参数。
- 开/关的语句是否正确。
- 规定的 I/O 格式是否与输入输出语句一致。
- 在使用文件之前是否已经打开文件或是用文件之后是否已经关闭文件。

②局部数据结构。在单元测试中，局部数据结构出错是比较常见的错误，在测试时应重点考

考虑以下因素：

- 变量的说明是否合适。
- 是否使用了尚未赋值或尚未初始化的变量。
- 变量的初始值或默认值是否正确。
- 变量名是否有错（例如：拼写错误）。

③重要的执行路径。在单元测试中，对路径的测试是最基本的任务。由于不能进行穷举测试，需要精心设计测试例子来发现是否有计算、比较或控制流等方面的错误。

•计算方面的错误：算术运算的优先次序不正确或理解错误；精度不够；运算对象的类型彼此不相容；算法错误；表达式的符号表示不正确等。

•比较和控制流的错误：本应相等的量由于精度造成不相等；不同类型进行比较；逻辑运算符不正确或优先次序错误；循环终止不正确（如多循环一次或少循环一次）、死循环；不恰当地修改循环变量；当遇到分支循环时，出口错误等。

④出错处理。好的设计应该能预测到出错的条件并且有对出错处理的路径。虽然计算机可以显示出错信息的内容，但仍需要程序员对出错进行处理，保证其逻辑的正确性，以便于用户维护。

⑤边界条件。边界条件的测试是单元测试的最后工作，也是非常重要的工作。软件容易在边界出现错误。

在多态的几种不同形式中，(37) 多态是一种特定的多态，指同一个名字在不同上下文中可代表不同的含义。

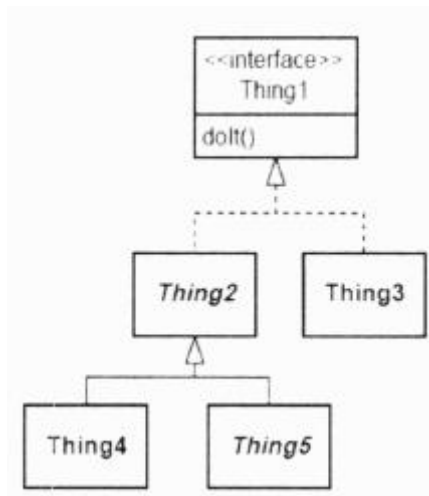
- (37) A. 参数                      B. 包含                      C. 过载                      D. 强制

**【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，不同的对象收到同一消息可以产生完全不同的结果，这一现象叫做多态（polymorphism）。在使用多态的时候，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。这样，同一消息就可以调用不同的方法。多态有参数多态、包含多态、过载多态和强制多态四类。参数多态是应用比较广泛的多态，被称为最纯的多态，包含多态在许多语言中都存在，最常见的例子就是子类型化，即一个类型是另一个类型的子类型。过载多态是同一个名字在不同的上下文中所代表的含义不同。

继承是父类和子类之间共享数据和方法的机制。以下关于继承的叙述中，不正确的是 (38)。有关于图中 doIt() 方法的叙述中，正确的是 (39)。



- (38) A. 一个父类可以有多个子类，这些子类都是父类的特例  
B. 父类描述了这些子类的公共属性和操作  
C. 子类可以继承它的父类（或祖先类）中的属性和操作而不必自己定义  
D. 子类中可以定义自己的新操作而不能定义和父类同名的操作
- (39) A. doIt() 必须由 Thing3 实现，同时可能由 Thing4 实现  
B. doIt() 必须由 Thing5 实现  
C. doIt() 必须由 Thing2、Thing3、Thing4、Thing5 实现  
D. doIt() 已经由 Thing1 实现，因为无需其它类实

**【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。父类定义公共的属性和操作，一个父类可以有多个子类，即多个特例。子类可以继承其父类或祖先类中的属性和操作作为自己的内容而不必自己定义，也可以覆盖这些操作，并加入新的内容。

接口是一种特殊的抽象机制，其中的操作不实现，需要由实现类来加以实现。对实现类为抽象类的，仍然可以保持操作为抽象，而如果是一个具体实现类，其中操作必须实现。题图中在接口 Thing1 中声明了 doIt()，在图中 Thing3 和 Thing2 作为 Thing1 的实现类，Thing3 为具体类，必须实现 doIt()；Thing2 可以保持 doIt() 为抽象操作，由其子类实现此操作，Thing4 为具体类，可以实现 doIt()，而 Thing5 仍然是抽象类，也可以保持 doIt() 为抽象操作。

以下关于 UML 部署图的叙述中，正确的是 (40)。

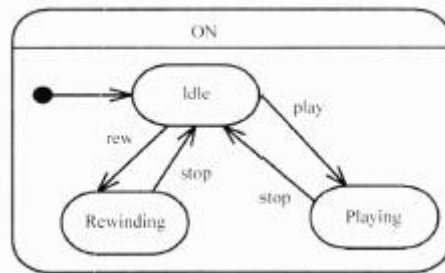
- (40) A. 因为一条消息总是有某种响应，所以部署组件之间的依赖是双向的  
B. 部署组件之间的依赖关系类似于包依赖  
C. 部署图不用于描述代码的物理模块  
D. 部署图不用于描述系统在不同计算机系统的物理分布

【答案】B

【解析】本题考查统一建模语言 (UML) 的基本知识。

UML2.0 中提供的部署图展现了运行处理节点以及其中组件的配置，描述代码的物理模块，用于描述系统在不同计算机系统的物理分布。部署图给出了体系结构的静态实施视图。它与组件图相关，通常一个节点包含一个或多个组件，其依赖关系类似于包图。

以下关于 UML 状态图的叙述中，不正确的是 (41)。对下图的描述，正确的是 (42)。



- (41) A. 用于描述一个对象在多个用例中的行为  
B. 用于某些具有多个状态的对象而不是系统中大多数或全部对象  
C. 用于描述多个对象之间的交互  
D. 可以用于用户界面或控制对象
- (42) A. ON 是一个并发状态  
B. 因为此状态图中没有终点 (final) 状态，所以此图是无效的  
C. play stop 和 rew 是动作  
D. ON 是超状态

【答案】C D

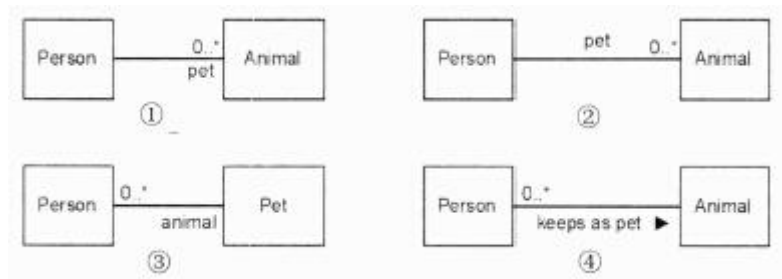
【解析】本题考查采用统一建模语言 (UML) 进行面向对象建模的基本知识。

UML2.0 中状态图主要用于描述对象、子系统、系统的生命周期。通过状态图可以了解到一个对象所能到达的所有状态以及对对象收到的事件 (消息、超时、错误、条件满足等) 对对象状态的影响等。针对具有可标记的状态和复杂的行为的对象构建状态图。状态可能有嵌

套的子状态，且子状态可以是一个状态图。

本题图示状态图中，ON 是一个超状态，它有 3 个子状态:Idle、Rewinding 和 Playing，这三个子状态之间在相关事件发生时状态之间进行迁移。

描述一些人 (Person) 将动物 (Animal) 养为宠物 (Pet) 的是图 (43)。



(43) A. ① B. ② C. ③ D. ④

【答案】A

【解析】本题考查统一建模语言 (UML) 的基本知识。

UML 中提供了类图，它以图形化的方式描述系统中对象及其关联关系。类图专注于系统的静态视图，它对于系统的领域内容建模特别重要。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。其中，关联关系是双向关系，即关联的对象双方彼此都能与对方通信。本题中人 (Person) 和动物 (Animal) 之间的关联关系，表示人可以将多只动物养为宠物 (Pet)，这时动物的角色是 pet，重数为 0..\*。

(44) 设计模式能使一个对象的状态发生改变时通知所有依赖它的监听者。(45) 设计模式限制类的实例对象只能有一个。适配器 (Adapter) 设计模式可以用于 (46)。用于为一个对象添加更多功能而不使用子类的是 (47) 设计模式。

(44) A. 责任链 B. 命令 C. 抽象工厂 D. 观察者

(45) A. 原型 B. 工厂方法 C. 单例 D. 生成器

(46) A. 将已有类的接口转换成和目标接口兼容

B. 改进系统性能

C. 将客户端代码数据转换成目标接口期望的合适的格式

D. 使所有接口不兼容可以一起工作

(47) A. 桥接 B. 适配器 C. 组合 D. 装饰器

【答案】D C A D

**【解析】** 本题考查设计模式的基本概念。

每种设计模式都有特定的意图，描述一个在我们周围不断重复发生的问题，以及该问题的解决方案的核心，使该方案能够重用而不必做重复劳动。

责任链 (Chain of Responsibility) 模式使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系，将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。

命令 (Command) 模式将一个请求封装为一个对象，从而使得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。

抽象工厂 (Abstract Factory) 模式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。

观察者 (Observer) 模式定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

原型 (Prototype) 模式用原型实例指定创建对象的种类，并且通过拷贝这个原型来创建新的对象。

工厂方法 (Factory Method) 定义一个用于创建对象的接口，让子类决定将哪一个类实例化，使一个类的实例化延迟到其子类。

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。

生成器 (Builder) 模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

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。

桥接 (Bridge) 模式将抽象部分与其实现部分分离，使它们都可以独立地变化。

组合 (Composite) 模式将对象组合成树形结构以表示“部分—整体”的层次结构，使得用户对单个对象和组合对象的使用具有一致性。

装饰器 (Decorator) 模式描述了以透明围栏来支持修饰的类和对象的关系，动态地给一个对象添加一些额外的职责，从增加功能的角度来看，装饰器模式相比生成子类更加灵活。

以下关于语言  $L = \{a^n b^n \mid n \geq 1\}$  的叙述中，正确的是 (48)。

(48) A. 可用正规式 “ $aa^*bb^*$ ” 描述，但不能通过有限自动机识别

B. 可用正规式 “ $ambm$ ” 表示，但可用有限自动机识别

- C. 不能用正规式表示，但可用有限自动机识别
- D. 不能用正规式表示，也不能通过有限自动机识别

**【答案】D**

**【解析】**本题考查程序语言基础知识。

$L = \{anbn | n \geq 1\}$  中的字符串特点是 a 的个数与 b 的个数相同，且所有的 a 都在 b 之前，该集合不是正规集，不能用正规式表示。

正规集可用正规式描述，用有限自动机识别。

编译过程中，对高级语言程序语句的翻译主要考虑声明语句和可执行语句。对声明语句，主要是将所需要的信息正确地填入合理组织的 (49) 中；对可执行语句，则是 (50)。

- (49) A. 符号表                      B. 栈                      C. 队列                      D. 树
- (50) A. 翻译成机器代码并加以执行                      B. 转换成语法树
- C. 翻译成中间代码或目标代码                      D. 转换成有限自动机

**【答案】A C**

**【解析】**本题考查程序语言翻译基础知识。

符号表的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。符号表的建立可以始于词法分析阶段，也可以放到语法分析和语义分析阶段，但符号表的使用有时会延续到目标代码的运行阶段。

编译过程中，在确认源程序的语法和语义之后，就可对其进行翻译，同时改变源程序的内部表示。对于声明语句，需要记录所遇到的符号的信息，因此应进行符号表的填查工作。对于可执行语句，需要翻译成中间代码或目标代码。

采用顺序表和单链表存储长度为 n 的线性序列，根据序号查找元素，其时间复杂度分别为 (51)。

- (51) A.  $O(1)$   $O(1)$                       B.  $O(1)$   $O(N)$                       C.  $O(N)$   $O(1)$                       D.  $O(N)$   $O(N)$

**【答案】B**

**【解析】**本题考查数据结构基础知识。

对于长度为 n 的线性序列，若采用顺序表（一维数组）存储，则每个元素的位序与存储该元素的数组元素下标有直接的对应关系，可进行随机查找，时间复杂度为  $O(1)$ ；若采用

单链表存储，则只能进行顺序访问，即必须从头指针出发，结合计数顺着指针链找到指定序号的元素，时间复杂度为  $O(n)$ 。

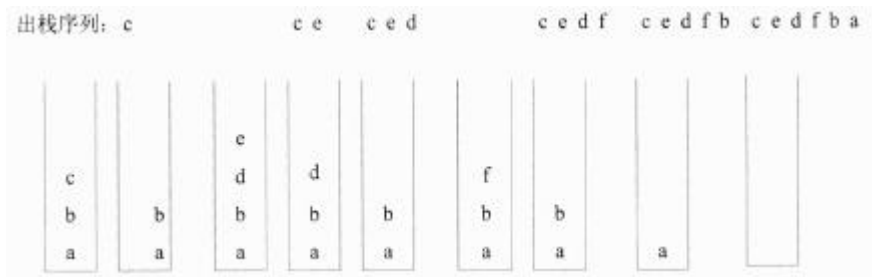
设元素序列 a, b, c, d, e, f 经过初始为空的栈 S 后，得到出栈序列 cedfba, 则栈 S 的最小容量为 (52)。

- (52) A. 3                      B. 4                      C. 5                      D. 6

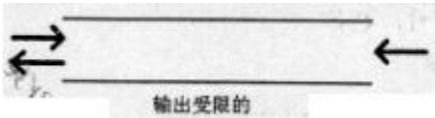
**【答案】B**

**【解析】** 本题考查数据结构基础知识。

栈是一种后进先出的数据结构。本问题中，根据元素入栈次序及出栈序列，每次需要出栈操作时栈的状态如下图所示，从中可以看出，栈中的元素个数最多时为 4。



输出受限的双端队列是指元素可以从队列的两端输入，但只能从队列的一端输出，如下图所示，若有 e1, e2, e3, e4 依次进入输出受限的双端队列，则得不到输出序列 (53)。



- (53) A. e4, e3, e2, e1    B. e4, e2, e1, e3    C. e4, e3, e1, e2    D. e4, e2, e3, e1

**【答案】D**

**【解析】** 本题考查数据结构基础知识。

该双端队列具有两个入口，所以 e1、e2、e3 进入队列后，从出口看可形成如下排列：

先出 e1，则得到 e1e2e3

先出 e2，则得到 e2e1e3

先出 e3，则得到 e3e1e2 或 e3e2e1

要在输出序列中首先得到 e4, 元素 e4 只能从出口端进入队列，结合前三个元素的可能排列，因此以 e4 打头的输出序列有：e4e1e2e3, e4e2e1e3, e4e3e1e2, e4e3e2e1。



在数据库系统中，视图是一个 (54)。

- (54) A. 真实存在的表，并保存了待查询的数据  
B. 真实存在的表，只有部分数据来源于基本表  
C. 虚拟表，查询时只能从一个基本表中导出  
D. 虚拟表，查询时可以从一个或者多个基本表或视图中导出

【答案】D

【解析】本题考查数据库系统概念方面的基本概念。

在数据库系统中，当视图创建完毕后，数据字典中存放的是视图定义。视图是从一个或者多个表或视图中导出的表，其结构和数据是建立在对应表的查询基础上的。和真实的表一样，视图也包括几个被定义的数据列和多个数据行，但从本质上讲，这些数据列和数据行来源于其所引用的表。因此，视图不是真实存在的基础表，而是一个虚拟表，视图所对应的数据并不实际地以视图结构存储在数据库中，而是存储在视图所引用的基本表中。

给定关系模式  $R(U, F)$ ，其中，属性集  $U = \{A, B, C, D, E, G\}$ ，函数依赖集  $F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D, AE \rightarrow G\}$ 。若将  $R$  分解为如下两个子模式 (55)，则分解后的关系模式保持函数依赖。

- (55) A.  $R_1(A, B, C)$  和  $R_2(D, E, G)$       B.  $R_1(A, B, C, D)$  和  $R_2(A, E, G)$   
C.  $R_1(B, C, D)$  和  $R_2(A, E, G)$       D.  $R_1(B, C, D, E)$  和  $R_2(A, E, G)$

【答案】B

【解析】本题考查关系模式和关系规范化方面的基础知识。

根据题意可以求出  $R_1(A, B, C, D)$  的函数依赖集  $F_1 = \{A \rightarrow B, A \rightarrow C, C \rightarrow D\}$ ， $R_2(A, E, G)$  的函数依赖集  $F_2 = \{AE \rightarrow G\}$ ，而  $F = F_1 + F_2$ ，所以分解后的关系模式保持函数依赖。

假设学生 **Students** 和教师 **Teachers** 的关系模式如下所示：

**Students**(学号，姓名，性别，类别，身份证号)

**Teachers**(教师号，姓名，性别，身份证号，工资)

其中，学生关系中的类别分为“本科生”和“研究生”两类。

a. 查询在读研究生的教师的平均工资、最高与最低工资之间差值的 SQL 语句如下：

```
SELECT _____ (56) _____  
FROM Students, Teachers  
WHERE _____ (57) _____;
```

b. 查询既是女教师，又是研究生且工资大于等于 3500 元的身份证号和姓名的 SQL 语句如下：

```
(SELECT 身份证号, 姓名
FROM Students
WHERE _____ (58) )
INTERSECT
(SELECT 身份证号, 姓名
FROM Teachers
WHERE _____ (59) )
```

(56) A. AVG(工资) AS 平均工资，MAX(工资) — MIN(工资) AS 差值

B. 平均工资 AS AVG(工资)，差值 AS MAX(工资) — MIN(工资)

C. AVG(工资) ANY 平均工资，MAX(工资) — MIN(工资) ANY 差值

D. 平均工资 ANY AVG(工资)，差值 ANY MAX(工资) — MIN(工资)

(57) A. Students. 身份证号=Teachers. 身份证号

B. Students. 类别='研究生'

C. Students. 身份证号=Teachers. 身份证号 AND Students. 类别='研究生'

D. Students. 身份证号=Teachers. 身份证号 OR Students. 类别='研究生'

(58) A. 工资>=3500

B. 工资>='3500'

C. 性别=女 AND 类别=研究生

D. 性别='女' AND 类别='研究生'

(59) A. 工资>=3500

B. 工资>='3500'

C. 性别=女 AND 类别=研究生

D. 性别='女' AND 类别='研究生'

**【答案】A C D A**

**【解析】**本题考查 SQL 方面的基础知识。

SQL 提供可为关系和属性重新命名的机制，这是通过使用具有“Old-name as new-name”形式的 as 子句来实现的。As 子句既可出现在 select 子句中，也可出现在 from 子句中。

查询在读研究生的教师的平均工资、最高与最低工资之间差值需要用条件“Students. 院系=Teachers. 院系 AND Students. 类别='研究生'”来限定。

因为第一条 SELECT 语句是从 Students 关系中查找女研究生的姓名和通信地址，故用条件“性别='女' AND 类别='研究生'”来限定；第二条 SELECT 语句查询是从 Teachers 关系中查找工资大于等于 3500 元的教师的姓名和通信地址，故用条件“工资>=3500”限定。又因为，第一条 SELECT 语句查询和第二条 SELECT 语句查询的结果集模式都为（姓名，通信地址），故可以用“INTERSECT”对它们取交集。

考虑下述背包问题的实例。有 5 件物品，背包容量为 100，每件物品的价值和重量如下表所示，并已经按照物品的单位重量价值从大到小排好序，根据物品单位重量价值大优先的策略装入背包中，则采用了 (60) 设计策略。考虑 0/1 背包问题（每件物品或者全部放入或者全部不装入背包）和部分背包问题（物品可以部分装入背包），求解该实例，得到的最大价值分别为 (61)。

物品编号	价值	重量
1	50	5
2	200	25
3	180	30
4	225	45
5	200	50

- (60) A. 分治                      B. 贪心                      C. 动态规划                      D. 回溯
- (61) A. 605 和 630              B. 605 和 605              C. 430 和 630                      D. 630 和 430

**【答案】B C**

**【解析】** 本题考查算法设计和分析知识。

背包问题是典型的算法问题，包括两种形式，即 0-1 背包问题和部分背包问题。0-1 背包问题是指每个物品或者全部放在背包中或者不放在背包中，求解在特定背包容量下装入背包物品的最大价值。部分背包问题中，每个物品可以部分地放入背包中，求解在特定背包容量下装入背包物品的最大价值。

基于单位重量价值最大优先的策略来将物品放入背包中，本质上是一种贪心的策略。在该策略下求 0-1 背包问题，不能确保得到最优解，事实上在本题给出实例中是得不到最优解的。而对于部分背包问题，是可以得到最优解的。

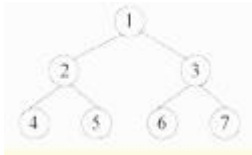
基于单位重量价值最大优先策略求解本题给出的实例。对于 0-1 背包问题，首先将物品 1、2 和 3 放入背包中，4 和 5 都不能再放入背包，此时背包重量为  $5+25+30=60$ ，获得价值  $50+200+180=430$ 。对于部分背包问题首先将物品 1、2 和 3 放入背包中，此时背包重量为 60，获得价值 430，此时还有剩余容量  $100-60=40$ ，可以将部分物品 4 放入背包，放入  $40/45=8/9$



【答案】D

【解析】本题考查数据结构基础知识。

用验证的方法求解，以高度为3的满二叉树（如下图所示）为例进行说明。



从中可以看出，若  $n=2m+1$ ，则结点  $n$  是  $m$  的右孩子结点。

以下关于哈希 (Hash, 散列) 查找叙述中，正确的是 (65)。

- (65) A. 哈希函数应尽可能复杂些，以消除冲突
- B. 构造哈希函数时应尽量使关键字的所有组成部分都能起作用
- C. 进行哈希查找时，不再需要与查找表中的元素进行比较
- D. 在哈希表中只能添加元素不能删除元素

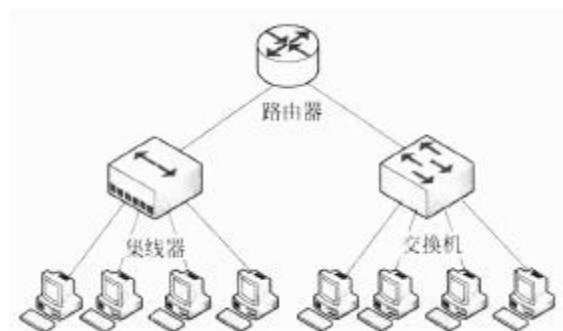
【答案】B

【解析】本题考查数据结构基础知识。

哈希表是通过一个以记录的关键字为自变量的函数（称为哈希函数）得到该记录的存储地址而构造的查找表，所以在哈希表中进行查找操作时，必须用同一哈希函数计算得到待查记录的存储地址，然后到相应的存储单元里去获得有关信息再判定查找是否成功。

冲突是指哈希函数将关键字不同的元素映射到了同一个存储地址。要减少冲突，就要设法使哈希函数尽可能均匀地把关键字映射到存储区的各个存储地址上，这样就可以提高查找效率。构造哈希函数时，一般应使关键字的所有组成部分都能起作用。

网络配置如下图所示，其中使用了一台路由器、一台交换机和一台集线器，对于这种配置，下面的论断中正确的是 (66)。



- (66) A. 2 个广播域和 2 个冲突域      B. 1 个广播域和 2 个冲突域  
C. 2 个广播域和 5 个冲突域      D. 1 个广播域和 8 个冲突域

**【答案】C**

**【解析】**

集线器连接的主机构成一个冲突域，交换机的每个端口属于一个冲突域，路由器连接的两部分网络形成两个广播域，所以共有两个广播域和 5 个冲突域。

把网络 117. 15. 32. 0/23 划分为 117. 15. 32. 0/27，得到的子网是 (67) 个，每个子网中可使用的主机地址是 (68) 个。

- (67) A. 4      B. 8      C. 16      D. 32  
(68) A. 30      B. 31      C. 32      D. 34

**【答案】C A**

**【解析】**

把网络 117. 15. 32. 0/23 划分为 117. 15. 32. 0/27，则子网掩码扩大了 4 位，所以得到的子网是 16 个。由于子网掩码为 27 位，所以主机地址只占 5 位，每个子网中可使用的主机地址是 30 个。

通常工作在 UDP 协议上的应用是 (69)。

- (69) A. 浏览网页      B. Telnet 远程登录      C. VoIP      D. 发送邮件

**【答案】C**

**【解析】**本试题考查各网络应用采用的下层传输协议。

浏览网页、Telnet 远程登录以及发送邮件应用均不允许数据的丢失，需要采用可靠的传输层协议 TCP，而 VoIP 允许某种程度上的数据丢失，采用不可靠的传输层协议 UDP。

随着网站知名度不断提高，网站访问量逐渐上升，网站负荷越来越重，针对此问题，一方面可通过升级网站服务器的软硬件，另一方面可以通过集群技术，如 DNS 负载均衡技术来解决，在 Windows 的 DNS 服务器中通过 (70) 操作可以确保域名解析并实现负载均衡。

- (70) A. 启用循环，启动转发器指向每个 Web 服务器  
B. 禁止循环，启动转发器指向每个 Web 服务器  
C. 禁止循环，添加每个 Web 服务器的主机记录

D. 启用循环，添加每个 Web 服务器的主机记录

**【答案】D**

**【解析】** 本题考查 Windows 的 DNS 服务器实现负载均衡的相关操作。

在 Windows 的 DNS 服务器中基于 DNS 的循环 (round robin), 只需要为同一个域名设置多个 ip 主机记录就可以了, DNS 中没有转发器的概念, 因此需要启用循环, 添加每个 Web 服务器的主机记录就可以确保域名解析并实现负载均衡。

So it is today. Schedule disaster, functional misfits, and system bugs all arise because the left hand doesn't know what the right hand is doing. As work (71), the several teams slowly change the functions, size, and speeds of their own programs, and the explicitly or implicitly (72) their assumptions about the inputs available and the uses to be made of outputs.

For example, the implementer of a program-overlaying function may run into problems and reduce speed relying on statistics that show how (73) this function will arise in application programs. Meanwhile, back at the ranch, his neighbor may be designing a major part of the supervisor so that it critically depends upon the speed of this function. This change in speed itself becomes a major specification change, and it needs to be proclaimed abroad and weighed from a system point of view.

How, then, shall teams (74) with one another? In as many ways as possible.

● Informally. Good telephone service and a clear definition of intergroup dependencies will encourage the hundreds of calls upon which common interpretation of written documents depends.

● Meetings. Regular project meetings, with one team after another giving technical briefings, are (75). Hundreds of minor misunderstandings get smoked out this way.

● Workbook. A formal project workbook must be started at the beginning.

(71) A. starts                      B. proceeds                      C. stops                      D. speeds

(72) A. change                      B. proceeds                      C. smooth                      D. hide

(73) A. frequently                      B. usually                      C. commonly                      D. rarely

(74) A. work

B. program

C. communicate

D. talk

(75) A. worthless

B. valueless

C. useless

D. invaluable

**【答案】 B A D C D**

**【解析】**

现在，其实也是这样的情况。因为左手不知道右手在做什么，所以进度灾难、功能的不合理和系统缺陷纷纷出现。随着工作的进行，许多小组慢慢地修改自己程序的功能、规模和速度，他们明确或者隐含地更改了一些有效输入和输出结果用法上的约定。

例如，程序覆盖（program-overlay）功能的实现者遇到了问题，并且统计报告显示了应用程序很少使用该功能。基于这些考虑，他降低了覆盖功能的速度。与此同时，整个开发队伍中，其他同事正在设计监控程序。监控程序在很大程度上依赖于覆盖功能，它在速度上的变化成为了主要的规格说明变更。因此需要从系统角度来考虑和衡量该变化，以及公开、广泛地发布变更结果。

那么，团队如何进行相互之间的交流沟通呢？通过所有可能的途径。

非正式途径，清晰定义小组内部的相互关系和充分利用电话，能鼓励大量的电话沟通，从而达到对所书写文档的共同理解。

会议，常规项目会议。会议中，团队一个接一个地进行简要的技术陈述。这种方式非常有用，能澄清成百上千的细小误解。

工作手册，在项目的开始阶段，应该准备正式的项目工作手册。



## 试题一

某慈善机构欲开发一个募捐系统,以跟踪记录为事业或项目向目标群体进行募捐而组织的集体性活动,该系统的主要功能如下所述。

(1) 管理志愿者。根据募捐任务给志愿者发送加入邀请、邀请跟进、工作任务;管理志愿者提供的邀请响应、支援站信息、工作时长、工作结果等。

(2) 确定募捐需求和收集所募捐赠(资金及物品)。根据需求提出的募捐任务、活动请求和募捐请求,获取所募集的资金和物品。

(3) 组织募捐活动。根据活动请求,确定活动时间范围。根据活动时间,搜索场馆,即向场馆发送场馆可用性请求,获得场馆可用性。然后根据活动时间和地点推广募捐活动,根据相应的活动信息举办活动,从募款机构获取资金并向其发放赠品。获取和处理捐赠,根据捐赠要求,提供所募集的捐赠;处理与处理人之间的交互。即:

录入捐赠人信息,处理后存入捐赠人信息表;从捐赠人信息表中查询捐赠人信息,向捐赠人发送捐赠请求,并将已联系的捐赠人存入已联系的捐赠人表。根据捐赠请求进行募集,募得捐赠后,将捐赠记录存入捐赠表;对捐赠记录进行处理后,存入已处理捐赠表,向捐赠人发送致谢函。根据已联系的捐赠人和捐赠记录进行跟进,将捐赠跟进情况发送给捐赠人。

现采用机构化方法对募捐系统进行分析与设计,获得如图 1-1、1-2 和 1-3 所示分层数据流图。

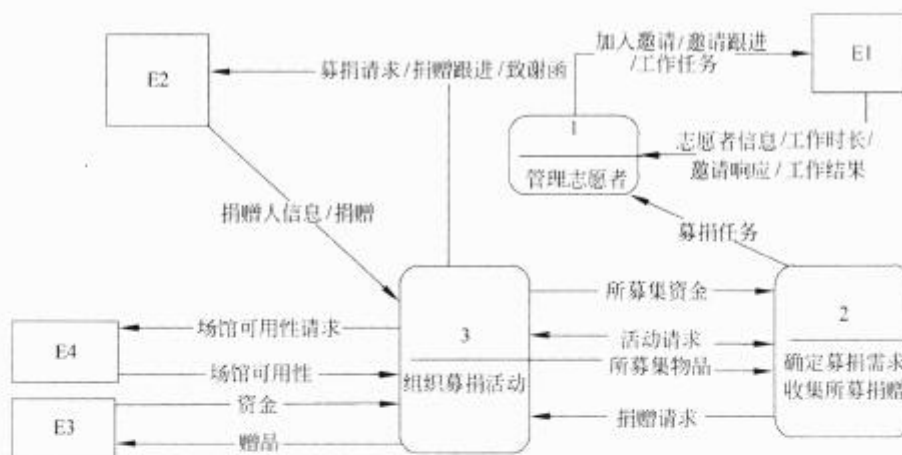


图 1-1 0 层数据流图

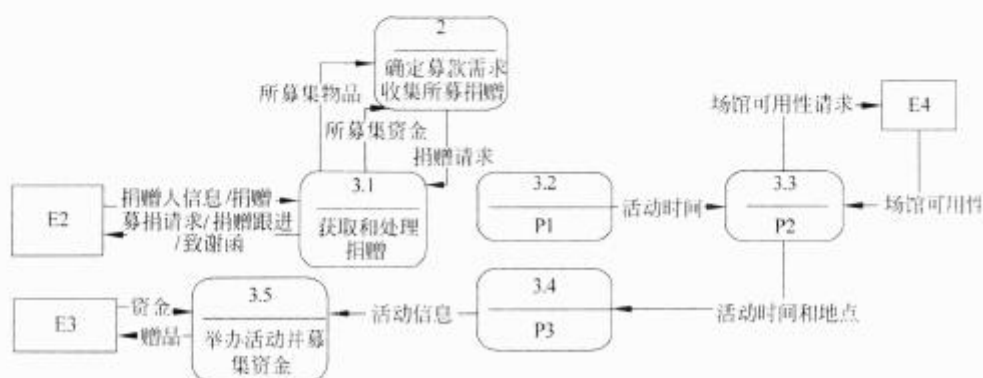


图 1-2 1 层数据流图

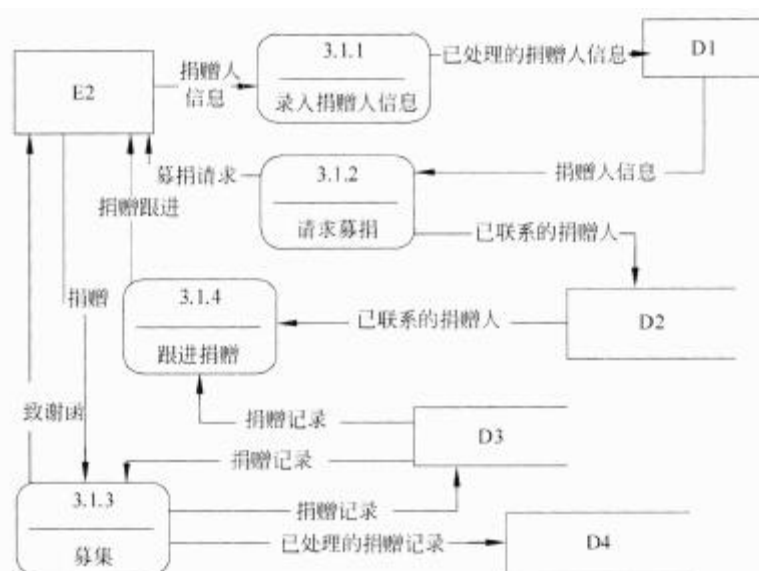


图 1-3 2 层数据流图

### 【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 志愿者 E2 捐赠人 E3 募捐机构 E4 场馆

本题采用结构化方法进行系统分析与设计，主要考查数据流图（DFD）的应用，是比较传统的题目，要求考生细心分析题目中所描述的内容。

DFD 是一种便于用户理解、分析系统数据流程的图形化建模工具，是系统逻辑模型的重要组成部分。顶层 DFD 一般用来确定系统边界，将待开发系统看作一个大的加工（处理），然后根据系统从哪些外部实体接收数据流，以及系统将数据流发送到哪些外部实体，建模出的顶层图中只有唯一的一个加工和一些外部实体，以及这两者之间的输入输出数据流。0 层 DFD 在顶层确定的系统外部实体以及与外部实体的输入输出数据流的基础上，将顶层 DFD 中的加工分解成多个加工，识别这些加工的输入输出数据流，使得所有顶层 DFD 中的输入数据流，经过这些加工之后变换成顶层 DFD 的输出数据流。根据 0 层 DFD 中的加工的复杂程度进一步建模加工的内容。

在建分层 DFD 时，根据需求情况可以将数据存储建模在不同层次的 DFD 中，注意在绘制下层数据流图时要保持父图与子图平衡。父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名称上相同，或者父图中的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流。

本问题给出 0 层 DFD，要求根据描述确定图中的外部实体。分析题目中描述，并结合已在图中给出的数据流进行分析。从题目的说明中可以看出，与系统交互实体包括志愿者、捐赠人、募款机构和场馆，这四个作为外部实体。

### 【问题 2】

在建模 DFD 时，需要对有些复杂加工（处理）进行进一步精化，图 1-2 为图 1-1 中处理 3 的进一步细化的 1 层数据图，图 1-3 为图 1-2 中 3.1 进一步细化的 2 层数据流图。补全图 1-2 中加工 P1、P2 和 P3 的名称和图 1-2 与图 1-3 中缺少的数据流。

P1：确定活动时间范围

P2：搜索场馆

P3：推广募捐活动

数据流名称	起点	终点
所募集资金	3.5 或 举办活动并募集资金	2
活动请求	2	3.2 或 确定活动时间范围
捐赠请求	2	3.1.3 募集
所募集捐赠	3.1.3 或 募集	2
或		
所募集资金	3.1.3 或 募集	2
所募集物品	3.1.3 或 募集	2

注：数据流没有次序要求；表中 2 处可以是“确定募捐需求收集所募捐赠”。

本题考查分层 DFD 的加工分解，以及父图与子图的平衡。图 1-2 中对图 1-1 的加工 3 进行进一步分解，根据说明 (3) 中对加工 3 的描述对图 1-2 进行分析。首先需要确定活动时间范围，其输入数据流是活动请求，输出流为活动时间。然后是搜索场馆，其输入流为活动时间，输出活动时间和地点，同时向场馆发送的场馆可用性请求和获得的场馆可用性分别作为输入和输出数据流。在确定活动时间和地点的基础上推广募捐活动，活动时间和地点是其输入流，活动信息作为其输出流，流向举办活动并募集资金，从募款机构获取资金并向其发放赠品，加工 2 收集募得的资金和物品，因此 3.5 还需要将所募集资金作为输出流。获取和处理捐赠（资金和物品）时以捐赠请求作为其输入流，输出流为所募集的捐赠，因为既有资金又有物品，而从募款机构募得的只有资金，将图 1-1 中加工 3 流向加工 2 的数据流，分为所募集资金和所募集物品，而 3.5 的输出流中只有所募集资金。

因此，P1 为确定活动时间范围，P2 为搜索场馆，P3 为推广募捐活动。图 1-2 中缺失了从 2 到 3.3 的活动时间和从 3.5 到 2 的所募集资金这两条数据流。

题目给出处理和捐赠人之间的交互进一步描述，对 3.1 进一步建模下层数据流图（图 1-3）。分解加工 3.1，确定相关数据流。其中根据加工 2 的捐赠请求进行募集，所募捐赠需要返回给加工 2。

根据父图与子图的平衡原则，图 1-3 中此处也缺失了捐赠请求和所募集资金和所募集物品。

### 【问题 3】

使用说明中的词语，给出图 1-3 中的数据存储 D1~D4 的名称。

D1：捐赠人信息表 D2：已联系的捐赠人表

D3：捐赠表 D4：已处理捐赠表

本问题考查 2 层 DFD 中数据存储的确定。本案例中，数据存储的描述都是在这一部分描述给出，所以数据存储建模在此层体现。

试题二

【说明】

某电视台拟开发一套信息管理系统，以方便对全台的员工、栏目、广告和演播厅等进行管理。

【需求分析】

（1）系统需要维护全台员工的详细信息、栏目信息、广告信息和演播厅信息等。员工的信息包括：工号、姓名、性别、出生日期、电话、住址等。栏目信息主要包括：栏目名称、播出时间、时长等。广告信息主要包括：广告编号、价格等。演播厅信息包括：房间号、房间面积等。

（2）电视台根据调度单来协调各个栏目、演播厅和场务。一档栏目只会占用一个演播厅，但会使用多名场务来进行演出协调。演播厅和场务可以被多个栏目循环使用。

（3）电视台根据栏目来插播广告。每档栏目可以插播多条广告，每条广告也可以在多档栏目插播。

（4）一档栏目可以有多个主持人，但一名主持人只能主持一档栏目。

（5）一名编辑人员可以编辑多条广告，一条广告只能由一名编辑人员编辑。

【概念模型设计】

根据需求阶段收集的信息设计的实体联系图（不完整）如图 2-1 所示。

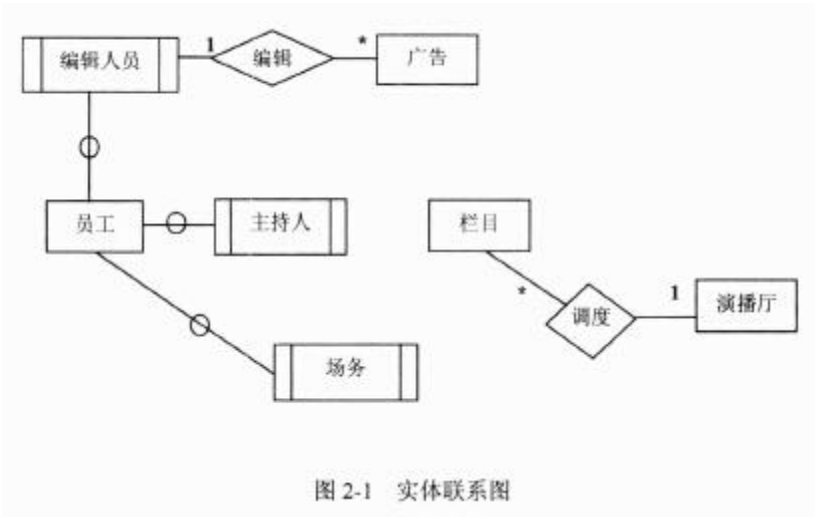


图 2-1 实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

演播厅（房间号，房间面积）

栏目（栏目名称，播出时间，时长）

广告（广告编号，销售价格，（1））

员工（工号，姓名，性别，出生日期，电话，住址）

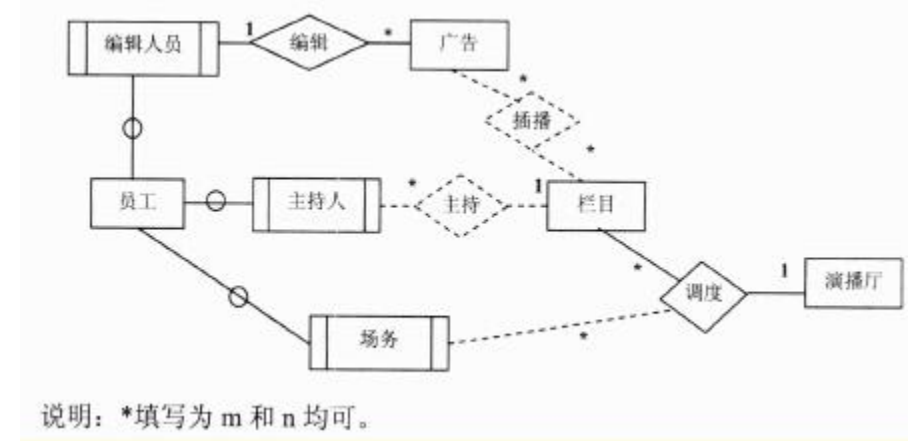
主持人（主持人工号，（2））

插播单（（3），播出时间）

调度单（（4））

【问题1】

补充图 2-1 中的联系和联系的类型。



本题考查数据库设计，属于比较传统的题目，考查点也与往年类似。

本问题考查数据库的概念结构设计，题目要求补充完整实体联系图中的联系和联系的类型。

根据题目的需求描述可知，一个栏目可以插播多条广告，而多条广告也可以在多个栏目中播放，因此栏目和广告之间存在“插播”联系，联系的类型为多对多（\*:\*，或 m:n）。

根据题目的需求描述可知，一个栏目可以有多个主持人，而一个主持人只能主持一档栏目，因此栏目和主持人之间存在“主持”联系，联系的类型为一对多（1:\*，或 1:n）。

根据题目的需求描述可知，一个栏目需要使用多名场务来进行演出协调，场务可以被多个栏目循环使用，因此演播厅、栏目和场务之间存在“调度”联系，联系的类型为 1 对多对多（1:\*:\*，或 1:m:n）。

【问题2】

根据图 2-1，将逻辑结构设计阶段生成的关系模式中的空（1）~（4）补充完整，并用下划线指出空（1）~（4）所在关系模式的主键。

广告（广告编号，销售价格，编辑人员工号）  
主持人（主持人工号，栏目名称）  
插播单（栏目名称，广告编号，播出时间）  
调度单（栏目名称，房间号，场务工号）

本问题考查数据库的逻辑结构设计，题目要求补充完整各关系模式，并给出各关系模式的主键。

根据实体联系图和需求描述，广告记录广告编号、销售价格和编辑人员工号。所以，对于“广告”关系模式，需补充属性“广告编号”。广告编号为广告的主键。

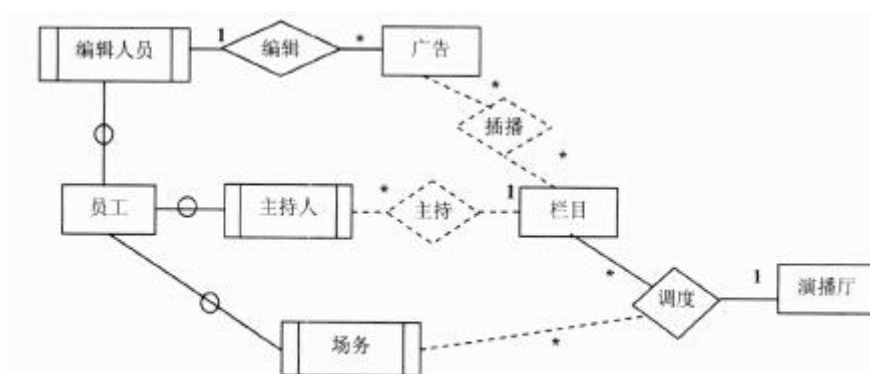
根据实体联系图和需求描述，主持人记录主持人工号和所属的栏目名称。所以，对于“主持人”关系模式，需补充属性“主持人工号”。主持人工号为主持人的主键。

根据实体联系图和需求描述，插播单需要记录栏目名称、广告编号和播出的时间。所以，对于“插播单”关系模式，需补充属性“栏目名称”和“广告编号”。栏目名称和广告编号联合作为插播单的主键。

根据实体联系图和需求描述，调度单需要记录栏目名称、房间号和参与的场务工号。所以，对于“调度单”关系模式，需补充属性“栏目名称”、“房间号”和“场务工号”。栏目名称、房间号和场务工号联合作为插播单的主键。

### 【问题 3】

现需要记录广告商信息，增加广告商实体。一个广告商可以提供多条广告，一条广告只由一个广告商提供。请根据该要求，对图 2-1 进行修改，画出修改后的实体间联系和联系的类型。



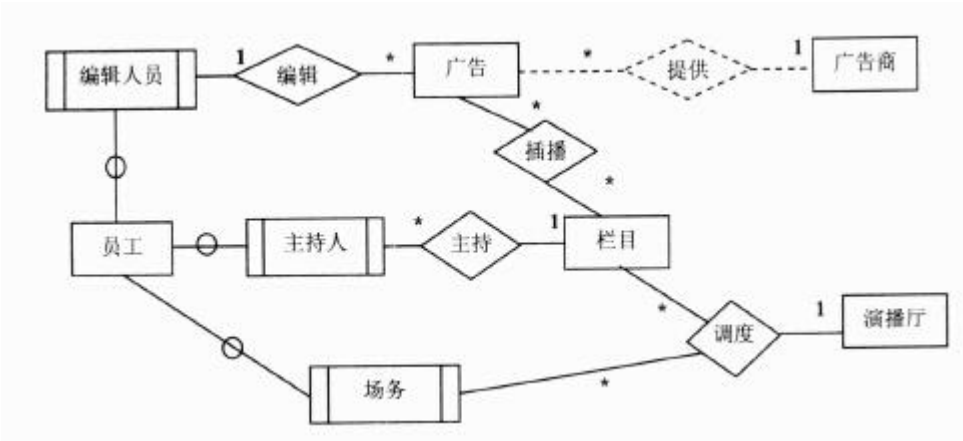
说明：\*填写为 m 和 n 均可。

本问题考查数据库的概念结构设计，根据新增的需求增加实体联系图中的实体的联系和联系的类型。

根据问题描述，一个广告商可以提供多条广告，一条广告只由一个广告商提供。则须在广告商实体和广告实体之间存在“提供”联系，联系的类型为 1 对多（1:\*, 或 1:n）



试题三



【说明】

某城市拟开发一个基于 Web 的城市黄页，公开发布该城市重要的组织或机构（以下统称为客户）的基本信息，方便城市生活。该系统的主要功能描述如下：

- （1）搜索信息：任何使用 Internet 的网络用户都可以搜索发布在城市黄页中的信息，例如客户的名称、地址、联系电话等。
- （2）认证：客户若想在城市黄页上发布信息，需要通过系统的认证。认证成功后，该客户成为系统授权用户。
- （3）更新信息：授权用户登录系统之后，可以更改自己的在城市黄页中的相关信息，例如变更联系电话等。
- （4）删除客户：对于拒绝继续在城市黄页上发布信息的客户，由系统管理员删除该客户的相关信息。

系统采用面向对象方法进行开发，在开发过程中认定出如表 3-1 所示的类。系统的用例图和类图分别如图 3-1 和 3-2 所示。

表 3-1 类列表	
类名	说明
InternetClient	网络用户
CustomerList	客户集，维护城市黄页上的所有客户信息
Customer	客户信息，记录单个客户的信息
RegisteredClient	授权用户
Administrator	系统管理员

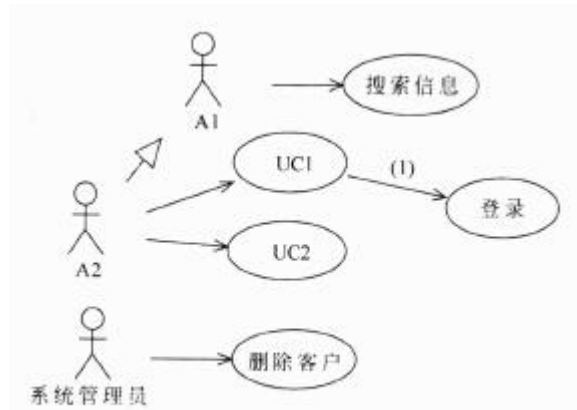


图 3-1 系统用例图

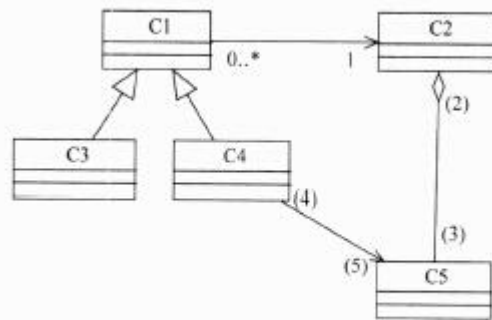


图 3-2 系统类图

### 【问题 1】

根据说明中的描述，给出图 3-1 中的 A1 和 A2 处所对应的参与者、UC1 和 UC2 处所对应的用例以及（1）处的关系。

A1：网络用户或 InternetClient      A2：授权用户或 RegisteredClient

UC1：更新信息 UC2:认证

(1)：«include»

本题属于经典的考题，主要考查面向对象分析方法以及 UML 的用例图和类图的相关应用。

本问题要求将图 3-1 所给出的用例图补充完整。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。

本题背景描述简单，再结合图 3-1 中给出的两个用例“搜索信息”和“删除客户”，很容易确定出“认证”、“更新信息”就是需要补充的用例。下面只需要确定参与者 A1、A2 以及这

两个用例与 A1、A2 之间的关系即可将图补充完整。在说明中出现了两类角色：客户和系统管理员。系统管理员已经作为参与者明确标识在 3-1 中了。那么 A1 和 A2 只能从“客户”这个角色中产生。明显地，“客户”在说明中被细分成了两类：网络用户和授权用户，而认证用户是一种特殊的网络用户。因此，A1 应该对应“网络用户”，A2 应该对应“授权客户”。要成为授权用户，必须首先经过认证。成为授权用户之后，不仅可以搜索信息，还具有更新信息的权限。更新信息时要求首先登录系统，因此“登录”是“更新信息”操作中所包含的一个必须步骤。由此可以确定 UC1 对应“更新信息”，UC2 对应“认证”，而 (1) 处的关系应该是«include»

## 【问题 2】

根据说明中的描述，给出图 3-2 中的 C1~C5 所对应的类名（表 3-1 中给出的类名）和 (2)~(5) 处所对应的多重度。

C1: IntemetClient      C2: CustomerList      C3: RegisteredClient

C4:Administrator    C5: Customer

(2)1 (3)0..\* (4) 0..1 (5) 0..1

本问题考查类建模。表 3-1 已经给出了类列表，这样对照图 3-2 寻找对应关系即可。图 3-2 中有两个明显的结构：继承（C1、C3 和 C4 之间）和聚集（C2 和 C5 之间）。

先确定聚集关系相关的类。由表 3-1 可以找出两个明显具有部分整体概念的类：CustomerList 和 Customer, 由类的说明可以看出，CustomerList 表示整体概念，Customer 表示部分概念。由此可以确定 C2 对应 CustomerList，C5 对应 Customer。同时可以确定出这两个类之间的多重度，即 (2) 处应为 1，(3) 处为 0..\*。

下面来确定继承关系相关的类。图 3-1 中已经出现了一个继承关系(A1 和 A2 之间)，这就给出了一个明显的提示：图 3-2 中的继承关系与这两类角色相关。回到表 3-1 中，发现了 3 个与角色相关的类：IntemetClient（网络用户）、RegisteredClient（授权用户）和 Administrator（系统管理员）。由于已经确定了 C5 是 Customer, 而能够对 Customer 进行操作的只有系统管理员。因此具有继承关系的这 3 个类应分别是：C1-网络用户，C3-授权用户，C4-系统管理员。(4)、(5) 处的多重度也可以确定下来，均为 0..1。

### 【问题 3】

认定类是面向对象分析中非常关键的一个步骤。一般首先从问题域中得到候选类集合，再根据相应的原则从该集合中删除不作为类的，剩余的就是从问题域中认定出来的类。简要说明选择题选类的原则，以及对候选类集合进行删除的原则。

选择候选类时通常考虑的是问题域中自然存在的名词。

具有下列特征的候选类需要删除：含义相近（冗余）、含义不明确的对象、暗示实现方式的、表示属性或特征、有动词含义的名词（表示行为和方法）。

本问题考查面向对象分析过程中认定类/对象的过程。通常分为两个步骤进行，首先将问题域（需求分析）中所有自然存在的名词都选出来，构成候选类集合。然后针对这个候选类集合，将满足以下原则的名词从候选类集合中删除：含义相近（冗余）、含义不明确的对象、暗示实现方式的、表示属性或特征、有动词含义的名词（表示行为和方法）。最后剩余的就是所认定的类/对象。

#### 试题四

阅读下列说明和 C 代码, 回答问题 1 至问题 3, 将解答写在答题纸的对应栏内。

##### 【说明】

设有  $m$  台完全相同的机器运行  $n$  个独立的任务, 运行任务  $i$  所需要的时间为  $t_i$ , 要求确定一个调度方案, 使得完成所有任务所需要的时间最短。

假设任务已经按照其运行时间从大到小顺序。算法基于最长运行时间作业优先的策略; 按顺序先把每个任务分配到一台机器上, 然后将剩余的任务依次放入空闲的机器。

##### 【C 代码】

下面是算法的 C 语言实现。

(1) 常量和变量说明

$m$ : 机器数

$n$ : 任务数

$t[]$ : 输入数组, 长度为  $n$ , 其中每个元素表示任务的运行时间, 下标从 0 开始

$s[][]$ : 二维数组, 长度为  $m \times n$ , 下标从 0 开始, 其中元素  $s[i][j]$  表示机器  $i$  运行的任务  $j$  的编号

$d[]$ : 数组, 长度为  $m$ , 其中元素  $d[i]$  表示机器  $i$  的运行时间, 下标从 0 开始

$count[]$ : 数组, 长度为  $m$ , 下标从 0 开始, 其中元素  $count[i]$  表示机器  $i$  的运行任务数

$i$ : 循环变量

$j$ : 循环变量

$k$ : 临时变量

$max$ : 完成所有任务的时间

$min$ : 临时变量

(2) 函数 `schedule`

```
void schedule() {  
    int i, j, k, max=0  
    for(i=0; i<m; i++) {  
        d[i]=0;  
        for(j=0; j<n; j++) {  
            s[i][j]=0;  
        }  
    }
```

```

    }

    for(i=0;i<m;i++) {                //分配前 m 个任务

        s[i][0]=j;

        (1)                ;

        count[i]=1;

    }

    for(                (2)                、 i<n,i++) {                //分配后 n-m 个任务

        int min=d[0];

        k=0;

        for(j=1;j<m;j++) {                //确定空闲机器

            if(min>d[j]) {

                min=d[j];

                k=j;                // 机器 K 空闲

            }

        }

        (3)                ;

        count[k]=count[k]+1;

        d[k]=d[k]+t[i];

    }

    for(i=0;i<m;i++){                //确定完成所有任务需要的时间

        if(                (4)                ) {

            max=d[i];

        }

    }

}

```

### 【问题 1】

根据说明和 c 代码，填充 C 代码中的空 (1) ~ (4)。

(1)  $d[i] = t[i]$       (2)  $i=m$       (3)  $s[k][count[k]] = i$       (4)  $max < d[i]$

本题考查算法设计与分析技术以及算法的 C 语言实现，是比较传统的题目，要求考生细心分

析题目中所描述的内容。

根据题中说明和代码注释，算法首先初始化数组  $d$  和  $s$  中的元素；然后将  $m$  个任务分配到  $m$  台机器上，此时将任务  $0, 1, \dots, m-1$  分别分配到机器  $0, 1, \dots, m-1$  上，同时设置  $d$ 、 $s$  和  $count$  数组中的相关元素的值，故空格 (1) 填写  $d[i] = t[i]$ ；接下来将剩下的  $n-m$  个任务分配到  $m$  台机器上，从任务  $m$  开始，因此空格 (2) 填写  $i=m$ ，确定首先空闲的机器  $k$ ，将当前尚未分配的第一个任务分配到机器  $k$  上，并设置  $d$ 、 $s$  和  $count$  数组中的相关元素的值，故空格 (3) 填写  $s[k][count[k]] = i$ ；最后确定从任务开始到结束所需要的时间，从所有机器的运行时间中选择运行时间最长的机器的运行时间，即最大的  $d[i]$ ，因此空格 (4) 填写  $\max\{d[i]\}$ 。

### 【问题 2】

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

(5) 贪心 (6)  $O(mn)$

根据上述 C 代码，算法有两处两重循环，时间复杂度为  $O(mn)$ ，有两处一重循环，时间复杂度为  $O(m)$ ，因此整个算法的时间复杂度为  $O(mn)$ 。

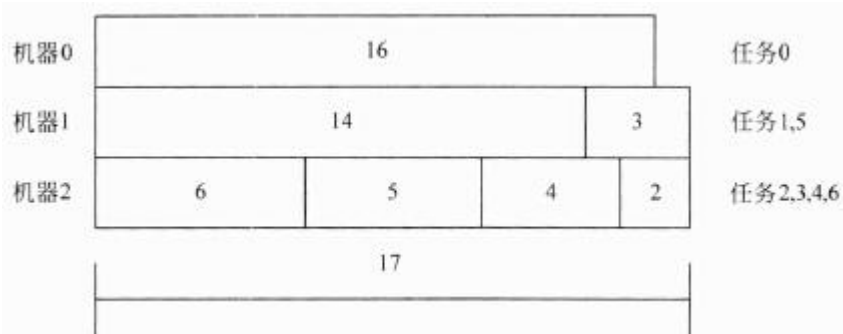
### 【问题 3】

考虑实例  $m=3$  (编号  $0 \sim 2$ )， $n=7$  (编号  $0 \sim 6$ )，各任务的运行时间为  $\{16, 14, 6, 5, 4, 3, 2\}$ 。则在机器 0、1 和 2 上运行的任务分别为 (7)、(8) 和 (9) (给出任务编号)。从任务开始运行到完成所需要的时间为 (10) 。

(7) 0 (8) 1, 5 (9) 2, 3, 4, 6 (10) 17

根据说明中的算法思想和 C 代码，首先将任务 0、1 和 2 分配到机器 0、1 和 2 上运行，运行时间分别为 16、14 和 6。由于任务 2 的时间最短，故任务 3 在机器 2 上运行，机器 2 上的运行时间为  $6+5=11$ ，仍然是时间最短，任务 4 继续在机器 2 上运行，机器 2 上的运行时间为  $11+4=15$ 。此时机器 1 上的运行时间最短，任务 5 在机器 1 上运行，机器 1 上的运行时间为  $14+3=17$ 。此时机器 2 上的运行时间最短，任务 6 在机器 2 上运行，机器 2 上的运行时间为  $15+2=17$ 。所以任务分配完成。根据此分配，在机器 0 上运行任务 0，运行时间为 16；机器 1 上运行任务 1 和 5，运行时间为 17；在机器 2 上运行任务 2，3，4 和 6，运行时间为 17。

因此从任务开始到结束的时间为 17。



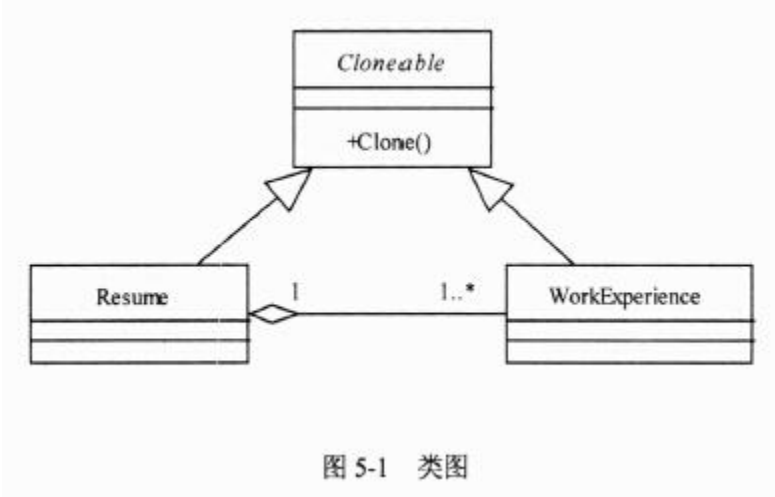


试题五

【说明】

现要求实现一个能够自动生成求职简历的程序。简历的基本内容包括求职者的姓名、性别、年龄及工作经历等。希望每份简历中的工作经历有所不同，并尽量减少程序中的重复代码。

现采用原型（Prototype）模式来实现上述要求，得到如图 5-1 所示的类图。



【C++代码】

```
#include<string>

using namespace std;

class Cloneable {
public;

    (1)          ;

};

class WorkExperience:public Cloneable {    //工作经历
private;

    string workDate;

    string company;

public;

    Cloneable* Clone() {

        (2)          ;

        obj->workDate=this->workDate;

        obj->company=this->company;
```

```

        return obj;

    }

    //其余代码省略

};

class Resume:public Cloneable {           //简历
private;

    string name;    string sex;    string age;

    WorkExperience* work;

    Resume(WorkExperience* work) {

        this->work=                (3)                ;

    }

public;

    Resume(string name){    /* 实现略 */    }

    void SetPersonallInfo(string sex, string age;){                /* 实现略
*/                }

    void SetWorkExperience(string workDate, string company){    /* 实现略
*/    }

    Cloneable* Clone() {

        (4)                ;

        obj->name=this->name;

        obj->sex=this->sex;

        obj->age=this->age;

        return obj;

    }

};

int main() {

    Resume *a=new Resume( “张三” );

    a->SetPersonallInfo( “男” , “29” );

    a->SetWorkExperience( “1998~2000” , “XXX 公司” );

    Resume *b=                (5)                ;

```

```
b->SetWorkExperience(“2001~2006”, “YYY 公司” );  
return 0;  
}
```

### 【问题1】

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。

- (1) virtual Cloneable\* Clone() = 0
- (2) WorkExperience \*obj = new WorkExperience()
- (3) (WorkExperience \*)work->Clone()
- (4) Resume \*obj = new Resume(this->work)
- (5) (Resume \*)a->Clone()

本题考查原型 (Prototype) 模式的概念及应用。

Prototype 模式是一种对象创建型模式。Prototype 模式通过给出一个原型对象来指明所要创建的对象类型，然后通过复制这个原型对象的方法，创建出更多同类型的对象。原型模式又可以分为两种：浅克隆和深克隆。浅克隆仅仅复制所考虑的对象，而不复制它所引用的对象，也就是其中的成员对象并不复制；深克隆除了对象本身被复制外，对象包含的引用也被复制，即成员对象也被复制。

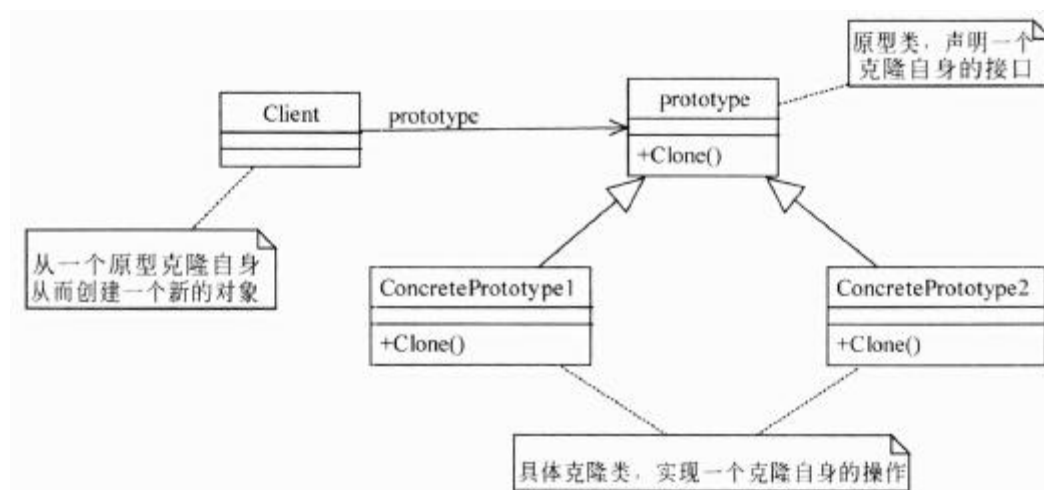
原型模式的优点是：

- (1) 向客户隐藏制造新实例的复杂性；
- (2) 提供让客户能够产生未知类型对象的选项；
- (3) 在某些环境下，复制对象比创建新对象更有效。

在一个复杂的类层次中，当系统必须从其中的许多类型创建新对象时，可以考虑原型模式。

使用原型模式的缺点是：对象的复制有时相当复杂。

Prototype 模式的结构如下图所示。



题目利用原型模式来实现自动生成多份求职简历，而且每份简历的工作经历有所差别。题目中使用的是原型模型中的深克隆。题目中的类 Cloneable 对应上图中的类 Prototype，Resume 和 WorkExperience 分别是两个具体克隆类。由于 WorkExperience 是 Resume 的一部分，所以在 Resume 中有一个 WorkExperience 类型的成员对象。

下面来分析程序。

第 (1) 空出现在类 Cloneable 的定义中。类 Cloneable 的作用是声明一个克隆自身的接口，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;
```

下面来确定纯虚拟函数的原型。由类图 5-1 以及 Cloneable 的子类 WorkExperience 中定义的成员函数 Clone() 可知，(1) 空应该填写 `virtual Cloneable* Clone() = 0`。空 (2) 出现在类 WorkExperience 成员函数 Clone() 的定义中。WorkExperience 作为具体克隆类，应实现继承自父类的纯虚拟函数 Clone()。根据已给出的代码的提示，在 Clone() 方法中，应再创建一个 WorkExperience 对象，而这个对象的名称就是 obj。因此

(2) 空处应填 `WorkExperience *obj = new WorkExperience()`

本题采用的是深克隆，除了对象本身被复制外，成员对象也被复制。work 是 Resume 中的成员对象，所以 (3) 空是对成员对象的克隆，因此应调用 work 对象的克隆操作，即

`(WorkExperience *)work->Clone()`。

第 (4) 空出现在 Resume 的 Clone() 方法中，功能与第 (2) 空类似，完成 Resume 对象的克隆，所以第 (4) 空应填 `Resume *obj = new Resume(this->work)`。

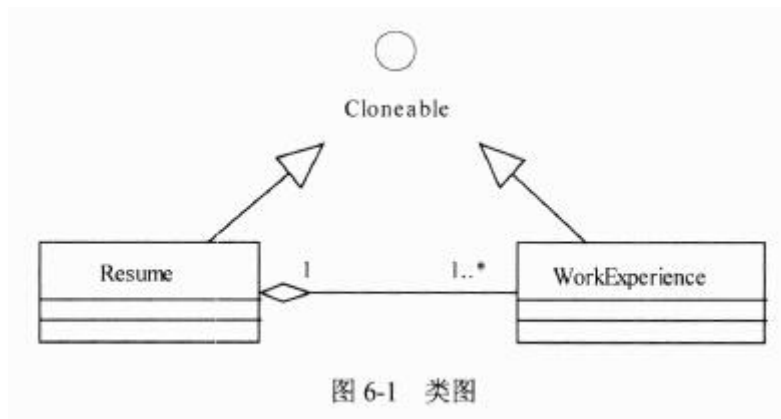
第 (5) 空是对原型模式的应用。用简历 a 克隆出简历 b，即调用 a 的克隆操作，所以第 (5) 空应填 `(Resume *)a->Clone()`

## 试题六

### 【说明】

现要求实现一个能够自动生成求职简历的程序。简历的基本内容包括求职者的姓名、性别、年龄及工作经历等。希望每份简历中的工作经历有所不同，并尽量减少程序中的重复代码。

现采用原型（Prototype）模式来实现上述要求，得到如图 6-1 所示的类图。



### 【Java 代码】

```
class WorkExperience          (1)          Cloneable {          //工作经历

    private string workDate;

    private string company;

    public Object Clone() {

                                (2)                                ;

        obj.workDate=this.workDate;

        obj.company=this.company;

        return obj;

    }

}

class Resume                  (3)          Cloneable {          //简历

    private string name;

    private string sex;

    private string age;
```

```

private WorkExperience work;

public Resume(string name) {
    this.name=name;          work=new SetWorkExperience();
}

private Resume(WorkExperience work) {
    this.work=          (4)          ;
}

public void SetPersonalInfo(string sex, string age;) {    /* 代码略 */    }
public void SetWorkExperience(string workDate, string company) {    /* 代码
略 */    }

public Object Clone() {
    Resume obj=          (5)          ;
    // 其余代码省略
    return obj;
}
}

class WorkResume {
    public static void main(string[]arg) {
        Resume a=new Resume( “张三” );
        a.SetPersonalInfo( “男”, “29” );
        a.SetWorkExperience( “1998~2000”, “XXX 公司” );
        Resume b=          (6)          ;
        b.SetWorkExperience( “2001~2006”, “YYY 公司” );
    }
}

```

### 【问题 1】

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。

- (1) implements
- (2) `WorkExperience obj = new WorkExperience()`
- (3) implements
- (4) `(WorkExperience)work.Clone()`
- (5) `new Resume(this.work)`
- (6) `(Resume)a.Clone()`

本题考查原型 (Prototype) 模式的概念及应用。

Prototype 模式是一种对象创建型模式。Prototype 模式通过给出一个原型对象来指明所要创建的对象类型，然后通过复制这个原型对象的方法，创建出更多同类型的对象。原型模式又可以分为两种：浅克隆和深克隆。浅克隆仅仅复制所考虑的对象，而不复制它所引用的对象，也就是其中的成员对象并不复制；深克隆除了对象本身被复制外，对象包含的引用也被复制，即成员对象也被复制。

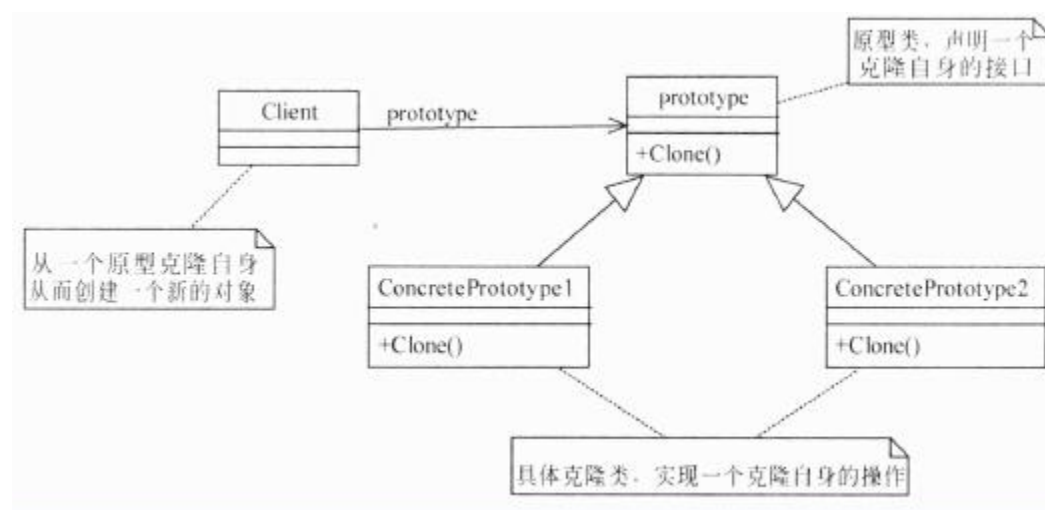
原型模式的优点是：

- (1) 向客户隐藏制造新实例的复杂性；
- (2) 提供让客户能够产生未知类型对象的选项；
- (3) 在某些环境下，复制对象比创建新对象更有效。

在一个复杂的类层次中，当系统必须从其中的许多类型创建新对象时，可以考虑原型模式。

使用原型模式的缺点是：对象的复制有时相当复杂。

Prototype 模式的结构如下图所示。



题目利用原型模式来实现自动生成多份求职简历，而且每份简历的工作经历有所差别。题目中使用的是原型模型中的深克隆。题目中的类 Cloneable 对应上图中的类 Prototype，Resume 和 WorkExperience，分别是两个具体克隆类。由于 WorkExperience 是 Resume 的一部分，所以在 Resume 中有一个 WorkExperience 类型的成员对象。

下面来分析程序。

Cloneable 是 Java 中的一个接口，其中已经定义 f Clone() 接口，可以直接在程序中使用。

对于两个具体克隆类 Resume 和 WorkExperience 来说，只要实现 Cloneable 即可。因此第

(1)、(3)空均填写 implements。

空 (2) 出现在类 WorkExperience 成员函数 Clone() 的定义中。WorkExperience 作为具体克隆类，应实现 Cloneable 中的方法 Clone()。根据已给出的代码的提示，在 Clone() 方法中，应再创建一个 WorkExperience 对象，而这个对象的名称就是 obj。因此 (2) 空处应填

WorkExperience obj = new WorkExperience() 。

本题采用的是深克隆，除了对象本身被复制外，成员对象也被复制。work 是 Resume 中的成员对象，所以 (3) 空是对成员对象的克隆，因此应调用 work 对象的克隆操作，即 (WorkExperience)work.Clone() 。

第 (4) 空出现在 Resume 的 Clone() 方法中，功能与第 (2) 空类似，完成 Resume 对象的克隆，所以第 (4) 空应填 new Resume(this.work)。

第 (5) 空是对原型模式的应用。用简历 a 克隆出简历 b，即调用 a 的克隆操作，所以第 (5) 空应填 (Resume)a.Clone()。