

为实现程序指令的顺序执行，CPU (1)中的值将自动加 1。

- (1) A. 指令寄存器 (IR)                      B. 程序计数器 (PC)  
C. 地址寄存器 (AR)                      D. 指令译码器 (ID)

【答案】B

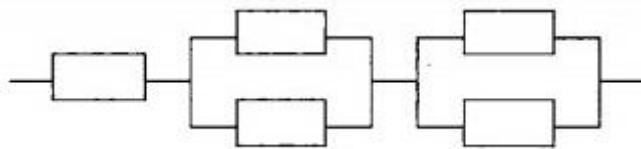
【解析】本题考查 CPU 中相关寄存器的基础知识。

指令寄存器 (IR)用来保存当前正在执行的指令。当执行一条指令时，先把它从内存取到数据寄存器 (DR)中，然后再传送至 IR。为了执行任何给定的指令，必须对操作码进行测试，以便识别所要求的操作。指令译码器 (ID)就是做这项工作的。指令寄存器中操作码字段的输出就是指令译码器的输入。操作码一经译码后，即可向操作控制器发出具体操作的特定信号。

地址寄存器 (AR)用来保存当前 CPU 所访问的内存单元的地址。由于在内存和 CPU 之间存在着操作速度上的差别，所以必须使用地址寄存器来保持地址信息，直到内存的读/写操作完成为止。

为了保证程序指令能够连续地执行下去，CPU 必须具有某些手段来确定下一条指令的地址。而程序计数器正起到这种作用，所以通常又称为指令计数器。在程序开始执行前，必须将它的起始地址，即程序的一条指令所在的内存单元地址送入 PC, 因此程序计数器(PC)的内容即是从内存提取的第一条指令的地址。当执行指令时，CPU 将自动修改 PC 的内容，即每执行一条指令 PC 增加一个量，这个量等于指令所含的字节数，以便使其保持的总是将要执行的下一条指令的地址。由于大多数指令都是按顺序来执行的，所以修改的过程通常只是简单的对 PC 加 1。

某计算机系统由下图所示的部件构成，假定每个部件的千小时可靠度都为 R，则该系统的千小时可靠度为(2)。



- (2) A.  $R+2R/4$                       B.  $R+R^2/4$                       C.  $R(1-(1-R)^2)$                       D.  $R(1-(1-R)^2)^2$

【答案】D

【解析】

本题考查系统可靠性方面的基础知识。

由子系统构成串联系统时，其中任何一个子系统失效就足以使系统失效，其可靠度等于各子系统可靠度的乘积；构成并联系统时，只要有一个子系统正常工作，系统就能正常工作。设每个子系统的可靠性分别以  $R_1, R_2, \dots, R_N$  表示，则整个系统的可靠度由下式求得：

$$R = 1 - (1 - R_1)(1 - R_2) \cdots (1 - R_N)$$

因此，本系统的可靠度为  $R(1 - (1 - R)^2)^2$ 。

以下关于计算机系统中断概念的叙述中，正确的是(3)。

- (3) A. 设备提出的中断请求和电源掉电都是可屏蔽中断  
B. 由 I/O 设备提出的中断请求和电源掉电都是不可屏蔽中断  
C. 由 I/O 设备提出的中断请求是可屏蔽中断，电源掉电是不可屏蔽中断  
D. 由 I/O 设备提出的中断请求是不可屏蔽中断，电源掉电是可屏蔽中断

【答案】C

【解析】本题考查中断基础知识。

按照是否可以被屏蔽，可将中断分为两大类：不可屏蔽中断（又叫非屏蔽中断）和可屏蔽中断。不可屏蔽中断源一旦提出请求，CPU 必须无条件响应，而对可屏蔽中断源的请求，CPU 可以响应，也可以不响应。典型的非屏蔽中断源的例子是电源掉电，一旦出现，必须立即无条件地响应，否则进行其他任何工作都是没有意义的。典型的可屏蔽中断源的例子是打印机中断，CPU 对打印机中断请求的响应可以快一些，也可以慢一些，因为让打印机等待是完全可以的。对于软中断，它不受中断允许标志位（IF 位）的影响，所以属于非屏蔽中断范畴。

与  $\overline{A} \oplus B$  等价的逻辑表达式是(4)。（ $\oplus$  表示逻辑异或， $+$  表示逻辑加）

- (4) A.  $A + \overline{B}$       B.  $A \oplus \overline{B}$       C.  $A \oplus B$       D.  $AB + \overline{AB}$

【答案】B

【解析】本题考查逻辑运算基础知识。

用真值表验证如下：

用真值表验证如下：

		选项 A	选项 B	选项 C	选项 D	$\overline{A \oplus B}$
A	B	$A + \overline{B}$	$A \oplus \overline{B}$	$A \oplus B$	$AB + \overline{AB}$	
0	0	1	1	0	1	1
0	1	0	0	1	0	0
1	0	1	0	1	0	0
1	1	1	1	0	1	1

从上表可知， $\overline{A \oplus B}$  与  $A + \overline{B}$ 、 $AB + \overline{AB}$  等价。

所以，选择 B 和 D

计算机指令一般包括操作码和地址码两部分，为分析执行一条指令，其(5)。

- (5) A. 操作码应存入指令寄存器 (IR)，地址码应存入程序计数器 (PC)  
B. 操作码应存入程序计数器 (PC)，地址码应存入指令寄存器 (IR)  
C. 操作码和地址码都应存入指令寄存器 (IR)  
D. 操作码和地址码都应存入程序计数器 (PC)

【答案】C

【解析】本题考查指令系统基础知识。

程序被加载到内存后开始运行，当 CPU 执行一条指令时，先把它从内存存储器取到缓冲寄存器 DR 中，再送入 IR 暂存，指令译码器根据 IR 的内容产生各种微操作指令，控制其他的组成部件工作，完成所需的功能。

程序计数器 (PC) 具有寄存信息和计数两种功能，又称为指令计数器。程序的执行分两种情况，一是顺序执行，二是转移执行。在程序开始执行前，将程序的起始地址送入 PC，该地址在程序加载到内存时确定，因此 PC 的内容即是程序第一条指令的地址。执行指令时，CPU 将自动修改 PC 的内容，以便使其保持的总是将要执行的下一条指令的地址。由于大多数指令都是按顺序来执行的，所以修改的过程通常只是简单地对 PC 加 1。当遇到转移指令时，后继指令的地址根据当前指令的地址加上一个向前或向后转移的位移量得到，或者根据转移指令给出的直接转移地址得到。

关于 64 位和 32 位微处理器，不能以 2 倍关系描述的是 (6)。

- (6) A. 通用寄存器的位数                      B. 数据总线的宽度  
C. 运算速度                                      D. 能同时进行运算的位数

【答案】C

**【解析】** 本题考查计算机系统基础知识。

计算机系统的运算速度受多种因素的影响,64 位微处理器可同时对 64 位数据进行运算,但不能说其速度是 32 位微处理器的 2 倍。

Outlook Express 作为邮件代理软件有诸多优点，以下说法中，错误的是(7)。

- (7) A. 可以脱机处理邮件  
B. 可以管理多个邮件账号  
C. 可以使用通讯簿存储和检索电子邮件地址  
D. 不能发送和接收安全邮件

**【答案】 D**

**【解析】**

Outlook Express 有以下一些优点:

- 可以脱机处理邮件，有效利用联机时间，降低了上网费用。
- 可以管理多个邮件账号，在同一个窗口中使用多个邮件账号。
- 可以使用通讯簿存储和检索电子邮件地址。
- 在邮件中添加个人签名或信纸。
- 发送和接收安全邮件。

杀毒软件报告发现病毒 Macro.Melissa，由该病毒名称可以推断病毒类型是(8)，这类病毒主要感染目标是(9)。

- (8) A. 文件型                  B. 引导型                  C. 目录型                  D. 宏病毒
- (9) A. EXE 或 COM 可执行文件                  B. Word 或 Excel 文件  
C. DLL 系统文件                  D. 磁盘引导区

**【答案】** D B

**【解析】** 本题考查计算机病毒方面的基础知识。

计算机病毒的分类方法有许多种，按照最通用的区分方式，即根据其感染的途径以及采用的技术区分，计算机病毒可分为文件型计算机病毒、引导型计算机病毒、宏病毒和目录型计算机病毒。

文件型计算机病毒感染可执行文件（包括 EXE 和 COM 文件）。

引导型计算机病毒影响软盘或硬盘的引导扇区。

目录型计算机病毒能够修改硬盘上存储的所有文件的地址。

宏病毒感染的对象是使用某些程序创建的文本文档、数据库、电子表格等文件，从文件

名可以看出 Macro.Melissa 是一种宏病毒，所以题中两空的答案是 D 和 B。

两个以上的申请人分别就相同内容的计算机程序的发明创造，先后向国务院专利行政部门提出申请，(10)可以获得专利申请权。

(10)A. 所有申请人均      B. 先申请人      C. 先使用人      D. 先发明人

**【答案】B**

**【解析】**本题考查知识产权基本知识

即专利管理部门授予专利权的基本原则。我国授予专利权采用先申请原则，即两个以上的申请人分别就同一项发明创造申请专利权的，专利权授予最先申请的人。如果两个以上申请人在同一日分别就同样的发明创造申请专利的，应当在收到专利行政管理部门的通知后自行协商确定申请人。如果协商不成，专利局将驳回所有申请人的申请，即所有申请人均不能取得专利权。所以，先申请人可以获得专利申请权。

王某是一名程序员，每当软件开发完成后均按公司规定完成软件文档，并上交公司存档，自己没有留存。因撰写论文的需要，王某向公司要求将软件文档原本借出复印，但遭到公司拒绝，理由是该软件文档属于职务作品，著作权归公司。以下叙述中，正确的是(11)。

- (11)A. 该软件文档属于职务作品，著作权归公司  
B. 该软件文档不属于职务作品，程序员享有著作权  
C. 该软件文档属于职务作品，但程序员享有复制权  
D. 该软件文档不属于职务作品，著作权由公司和程序员共同享有

**【答案】A**

**【解析】**本题考查知识产权知识

即软件知识产权归属。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、计算机软件都可能由于为执行工作任务而创作，属于职务作品。其著作权归属分两种情形：

(1)一般职务作品

一般职务作品的著作权由作者享有。单位或其他组织享有在其业务范围内优先使用的权利，期限为 2 年。单位的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单位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单位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。

(2)特殊的职务作品

所谓特殊职务作品一是指是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、产品设计图纸、地图、计算机软件等职务作品；二是指法律、行政法规规定或者合同约定著作权由法人或者其他组织享有的职务作品。对于特殊职务作品，作者享有署名权，其他权利由法人或非法人单位享有。

本题涉及软件知识产权，王某为完成公司指派的工作任务所开发的软件是职务软件，职务软件包括软件文档和源程序。该软件文档属于特殊职务作品，依据著作权法，对于特殊职务作品，除署名权以外，著作权的其他权利由公司享有。

在 ISO 制定并发布的 MPEG 系列标准中，(12)的音、视频压缩编码技术被应用到 VCD 中，(13)标准中的音、视频压缩编码技术被应用到 DVD 中，(14)标准中不包含音、视频压缩编码技术。

- |               |           |           |            |
|---------------|-----------|-----------|------------|
| (12)A. MPEG-1 | B. MPEG-2 | C. MPEG-7 | D. MPEG-21 |
| (13)A. MPEG-1 | B. MPEG-2 | C. MPEG-4 | D. MPEG-21 |
| (14)A. MPEG-1 | B. MPEG-2 | C. MPEG-4 | D. MPEG-7  |

【答案】A B D

【解析】

MPEG 是 Moving Picture Expert Group 的简称，最初是指由国际标准化组织（ISO）和国际电工委员会（IEC）联合组成的一个研究视频和音频编码标准的专家组。同时 MPEG 也用来命名这个小组所负责开发的一系列音、视频编码标准和多媒体应用标准。这个专家组至今为止已制定和制定中的标准包括 MPEG-1、MPEG-2、MPEG-4、MPEG-7 和 MPEG-21 标准。其中 MPEG-1、MPEG-2 和 MPEG-4 主要针对音、视频编码技术，而 MPEG-7 是多媒体内容描述接口标准，MPEG-21 是多媒体应用框架标准。

VCD 使用了 MPEG-1 标准作为其音、视频信息压缩编码方案，而 MPEG-2 标准中的音、视频压缩编码技术被应用到 DVD 中。

基于构件的软件开发，强调使用可复用的软件“构件”来设计和构建软件系统，对所需的构件进行合格性检验、(15)，并将它们集成到新系统中。

- |             |         |          |          |
|-------------|---------|----------|----------|
| (15)A. 规模度量 | B. 数据验证 | C. 适应性修改 | D. 正确性测试 |
|-------------|---------|----------|----------|

**【答案】C**

**【解析】**本题考查基于构件的软件开发基础知识。

基于构件的软件开发，主要强调在构建软件系统时复用已有的软件“构件”，在检索到可以使用的构件后，需要针对新系统的需求对构件进行合格性检验、适应性修改，然后集成到新系统中。

采用面向对象方法开发软件的过程中，抽取和整理用户需求并建立问题域精确模型的过程叫 (16)。

(16)A. 面向对象测试 B. 面向对象实现 C. 面向对象设计 D. 面向对象分析

**【答案】D**

**【解析】**本题考查面向对象软件开发过程的基础知识。

采用面向对象的软件开发，通常有面向对象分析、面向对象设计、面向对象实现。面向对象分析是为了获得对应用问题的理解，其主要任务是抽取和整理用户需求并建立问题域精确模型。面向对象设计是采用协作的对象、对象的属性和方法说明软件解决方案的一种方式，强调的是定义软件对象和这些软件对象如何协作来满足需求，延续了面向对象分析。面向对象实现主要强调采用面向对象程序设计语言实现系统。面向对象测试是根据规范说明来验证系统设计的正确性。

使用白盒测试方法时，应根据 (17) 和指定的覆盖标准确定测试数据。

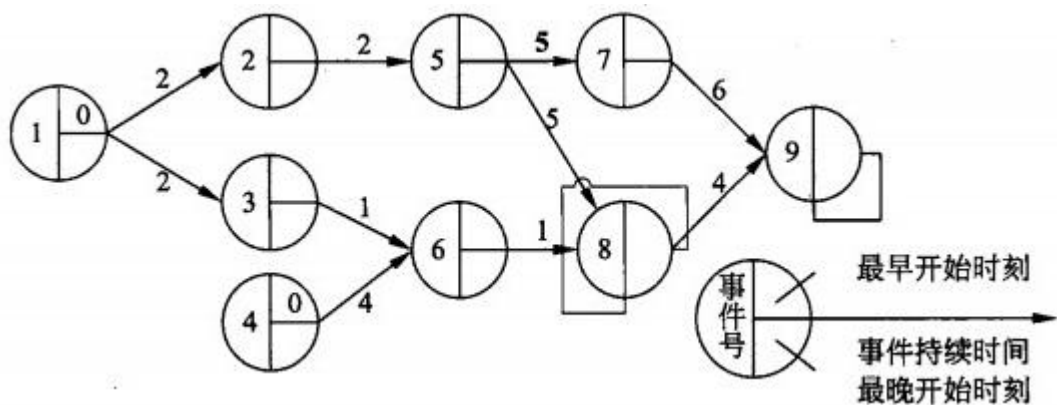
(17)A. 程序的内部逻辑 B. 程序结构的复杂性  
C. 使用说明书 D. 程序的功能

**【答案】A**

**【解析】**本题考查软件测试方法中白盒测试的基础知识。

白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的执行路径和过程进行测试，检查是否满足设计的需要。白盒测试常用的技术涉及不同覆盖标准，在测试时需根据指定的覆盖标准确定测试数据。

进度安排的常用图形描述方法有 Gantt 图和 PERT 图。Gantt 图不能清晰地描述 (18)：PERT 图可以给出哪些任务完成后才能开始另一些任务。下图所示的 PERT 图中，事件 6 的最晚开始时刻是 (19)。



- (18) A. 每个任务从何时开始                      B. 每个任务到何时结束  
C. 每个任务的进展情况                      D. 各任务之间的依赖关系

- (19) A. 0                      B. 3                      C. 10                      D. 11

【答案】D    C

【解析】本题考查软件项目计划基础知识。

软件项目计划的一个重要内容是安排进度,常用的方法有 Gantt 图和 PERT 图。Gantt 图用水平条状图描述,它以日历为基准描述项目任务,可以清楚地表示任务的持续时间和任务之间的并行,但是不能清晰地描述各个任务之间的依赖关系。PERT 图是一种网络模型,描述一个项目任务之间的关系。可以明确表达任务之间的依赖关系,即哪些任务完成后才能开始另一些任务,以及如期完成整个工程的关键路径。

若某整数的 16 位补码为 FFFFh (H 表示十六进制),则该数的十进制值为 (20)。

- (20) A. 0                      B. -1                      C. 216-1                      D. -216+1

【答案】B

【解析】

本题考查数据表示基础知识。

根据补码定义,数值  $X$  的补码记作  $[X]_{\text{补}}$ ,如果机器字长为  $n$ ,则最高位为符号位,0 表示正号,1 表示负号,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。

16 位补码能表示的数据范围为  $[-2^{15}, 2^{15} - 1]$ 。对于整数  $(2^{16} - 1)$  和  $(-2^{16} + 1)$ ,数据表示需要 16 位,再加一个符号位,共 17 位,因此不在 16 位补码能表示的数据范围之内。

在补码表示中,0 有唯一的编码:  $[+0]_{\text{补}} = 0000000000000000$ ,  $[-0]_{\text{补}} = 0000000000000000$ ,即  $0000_{\text{H}}$ 。

$[-1]_{\text{原}} = 1000000000000001$ ,  $[-1]_{\text{反}} = 1111111111111110$ ,因此 -1 的补码为  $[-1]_{\text{补}} = 1111111111111111 = \text{FFFF}_{\text{H}}$ 。



逻辑表达式“ $a \wedge b \vee c \wedge (b \vee x > 0)$ ”的后缀式为(21)。(其中 $\wedge$ 、 $\vee$ 分别表示逻辑与、逻辑或， $>$ 表示关系运算大于，对逻辑表达式进行短路求值)

- (21) A.  $abc b x 0 > \vee \wedge \wedge \vee$  B.  $ab \wedge c \vee b \wedge x 0 > \vee$   
C.  $ab \wedge cb \wedge x > 0 \vee \vee$  D.  $ab \wedge cb x 0 > \vee \wedge \vee$

【答案】D

【解析】

本题考查逻辑表达式的计算及程序语言处理基础知识。

“逻辑与运算”的优先级高于“逻辑或运算”。

“逻辑与运算”表达式“ $x \wedge y$ ”的短路求值逻辑是：若  $x$  为假，则可知“ $x \wedge y$ ”的值为假，无需再对  $y$  求值，因此只有在  $x$  为真时继续对  $y$  求值。

“逻辑或运算”表达式“ $x \vee y$ ”的短路求值逻辑是：若  $x$  为真，则可知“ $x \vee y$ ”的值为真，无需再对  $y$  求值，因此只有在  $x$  为假时继续对  $y$  求值。

对于逻辑表达式“ $a \wedge b \vee c \wedge (b \vee x > 0)$ ”，从运算符的优先级方面考虑，需先对“ $a \wedge b$ ”求值，然后对“ $c \wedge (b \vee x > 0)$ ”求值，最后进行“ $\vee$ ”运算，因此后缀式为“ $ab \wedge cb x 0 > \vee \wedge \vee$ ”。

编译程序对 C 语言源程序进行语法分析时，可以确定(22)。

- (22) A. 变量是否定义（或声明） B. 变量的值是否正确  
C. 循环语句的执行次数 D. 循环条件是否正确

【答案】A

【解析】本题考查程序语言基础知识。

对 C 源程序进行编译时，需建立符号表，其作用是记录源程序中各个符号（变量等）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。

如果系统采用信箱通信方式，当进程调用 Send 原语被设置成“等信箱”状态时，其原因是(23)。

- (23) A. 指定的信箱不存在 B. 调用时没有设置参数  
C. 指定的信箱中无信件 D. 指定的信箱中存满了信件

【答案】D

【解析】

因为 Send 原语是发送原语，如果系统采用信箱通信方式，那么当进程调用 Send 原语被

设置成“等信箱”状态时，意味着指定的信箱存满了信件，无可用空间。

若在系统中有若干个互斥资源 R，6 个并发进程，每个进程都需要 2 个资源 R，那么使系统不发生死锁的资源 R 的最少数目为 (24)。

(24) A. 6                      B. 7                      C. 9                      D. 12

**【答案】B**

**【解析】**

对于选项 A，操作系统为每个进程分配 1 个资源 R 后，若这 6 个进程再分别请求 1 个资源 R 时系统已无可供分配的资源 R，则这 6 个进程由于请求的资源 R 得不到满足而死锁。对于选项 B，操作系统为每个进程分配 1 个资源 R 后，系统还有 1 个可供分配的资源 R，能满足其中的 1 个进程的资源 R 要求并运行完毕释放占有的资源 R，从而使其他进程也能得到所需的资源 R 并运行完毕。

某进程有 5 个页面，页号为 0~4，页面变换表如下所示。表中状态位等于 0 和 1 分别表示页面不在内存或在内存。若系统给该进程分配了 3 个存储块，当访问的页面 3 不在内存时，应该淘汰表中页号为 (25) 的页面。假定页面大小为 4KB，逻辑地址为十六进制 2C25H，该地址经过变换后，其物理地址应为十六进制 (26)。

页号	页帧号	状态位	访问位	修改位
0	3	1	1	0
1	—	0	0	0
2	4	1	1	1
3	—	0	0	0
4	1	1	1	1

(25) A. 0                      B. 1                      C. 2                      D. 4

(26) A. 2C25H                      B. 4096H                      C. 4C25H                      D. 8C25H

**【答案】A C**

**【解析】**

根据题意，页面变换表中状态位等于 0 和 1 分别表示页面不在内存或在内存，所以 0、2 和 4 号页面在内存。当访问的页面 3 不在内存时，系统应该首先淘汰未被访问的页面，因为根据程序的局部性原理，最近未被访问的页面下次被访问的概率更小；如果页面最近都被访问过，应该先淘汰未修改过的页面。因为未修改过的页面内存与辅存一致，故淘汰时无需

写回辅存，使系统页面置换代价小。经上述分析，0、2 和 4 号页面都是最近被访问过的，但 2 和 4 号页面都被修改过而 0 号页面未修改过，故应该淘汰 0 号页面。

(26) 根据题意，页面大小为 4KB，逻辑地址为十六进制 2C25H 其页号为 2, 页内地址为 C25H，查页表后可知页帧号（物理块号）为 4, 该地址经过变换后，其物理地址应为页帧号 4 拼上页内地址 C25H，即十六进制 4C25H。

假设某磁盘的每个磁道划分成 9 个物理块，每块存放 1 个逻辑记录。逻辑记录 R0, R1, …, R8 存放在同一个磁道上，记录的安排顺序如下表所示：

物理块	1	2	3	4	5	6	7	8	9
逻辑记录	R0	R1	R2	R3	R4	R5	R6	R7	R8

如果磁盘的旋转速度为 27ms/周，磁头当前处在 R0 的开始处。若系统顺序处理这些记录，使用单缓冲区，每个记录处理时间为 3ms，则处理这 9 个记录的最长时间为 (27)：若对信息存储进行优化分布后，处理 9 个记录的最少时间为 (28)。

- (27) A. 54ms                      B. 108ms                      C. 222ms                      D. 243ms
- (28) A. 27ms                      B. 54ms                      C. 108ms                      D. 216ms

【答案】C B

【解析】

系统读记录的时间为  $27/9=3\text{ms}$ 。对第一种情况：系统读出并处理记录 R1 之后，将转到记录 R3 的开始处，所以为了读出记录 R2, 磁盘必须再转一圈，需要 27ms（转一圈）的时间。这样，处理 9 个记录的总时间应为 222ms。因为处理前 8 个记录（即 R1, R2, …, R8）的总时间再加上读 R9 时间： $8 \times 27\text{ms} + 6\text{ms} = 222\text{ms}$ 。

物理块	1	2	3	4	5	6	7	8	9
逻辑记录	R1	R6	R2	R7	R3	R8	R4	R9	R5

(28) 对于第二种情况，若对信息进行分布优化的结果如下所示：

从上表可以看出，当读出记录 R1 并处理结束后，磁头刚好转至 R2 记录的开始处，立即就可以读出并处理，因此处理 9 个记录的总时间为：

$9 \times (3\text{ms (读记录)} + 3\text{ms (处理记录)}) = 9 \times 6\text{ms} = 54\text{ms}$

对于一个大型软件来说，不加控制地变更很快会引起混乱。为有效地实现变更控制，需借助于配置数据库和基线的概念。(29)不属于配置数据库。

(29) A. 开发库                      B. 受控库                      C. 信息库                      D. 产品库

**【答案】C**

**【解析】** 本题考查软件变更管理和配置管理的基础知识。

软件变更控制是变更管理的重要内容，要有效进行变更控制，需要借助配置数据库和基线的概念。配置数据库一般包括开发库、受控库和产品库。

软件设计时需要遵循抽象、模块化、信息隐蔽和模块独立原则。在划分软件系统模块时，应尽量做到 (30)。

(30) A. 高内聚高耦合    B. 高内聚低耦合    C. 低内聚高耦合    D. 低内聚低耦合

**【答案】B**

**【解析】** 本题考查软件设计原则的基础知识。

软件设计时需要遵循抽象、模块化、信息隐蔽和模块独立原则。耦合性和内聚性是模块独立性的两个定性标准，在划分软件系统模块时，尽量做到高内聚、低耦合，提高模块的独立性。

能力成熟度集成模型 CMMI 是 CMM 模型的最新版本，它有连续式和阶段式两种表示方式。基于连续式表示的 CMMI 共有 6 个 (0~5) 能力等级，每个能力等级对应到一个一般目标以及一组一般执行方法和特定方法，其中能力等级 (31) 主要关注过程的组织标准化和部署。

(31) A. 1                      B. 2                      C. 3                      D. 4

**【答案】C**

**【解析】** 本题考查软件能力成熟度集成模型的基础知识。

能力成熟度集成模型 CMMI 是 CMM 模型的最新版本，基于连续式表述的 CMMI 共有 6 个 (0~5) 能力等级，对应于未完成级、已执行级、已管理级、已定义级、量化管理级、优化级。每个能力等级对应到一个一般目标，以及一组一般执行方法和特定方法。

能力等级 0 指未执行过程，表明过程域的一个或多个特定目标没有被满足；能力等级 1 指过程通过转化可识别的输入工作产品，产生可识别的输出工作产品，关注于过程域的特定目标的完成；能力等级 2 指过程作为已管理的过程制度化，针对单个过程实例的能力；能力等级 3 指过程作为已定义的过程制度化，关注过程的组织级标准化和部署；能力等级 4 指过程作为定量管理的过程制度化；能力等级 5 指过程作为优化的过程制度化，表明过程得到很好地执行且持续得到改进。

统一过程（UP）定义了初启阶段、精化阶段、构建阶段、移交阶段和产生阶段，每个阶段以达到某个里程碑时结束，其中（32）的里程碑是生命周期架构。

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

**【答案】B**

**【解析】**本题考查软件开发过程的基础知识。

统一过程（UP）定义了初启阶段、精化阶段、构建阶段、移交阶段和产生阶段，每阶段达到某个里程碑时结束。其中初启阶段的里程碑是生命周期目标，精化阶段的里程碑是生命周期架构，构建阶段的里程碑是初始运作功能，移交阶段的里程碑是产品发布。

程序的三种基本控制结构是（33）。

- (33) A. 过程、子程序和分程序      B. 顺序、选择和重复  
C. 递归、堆栈和队列      D. 调用、返回和跳转

**【答案】B**

**【解析】**本题考查软件程序设计的基础知识。

程序的三种基本控制结构是顺序结构、选择结构和重复结构。

（34）不属于软件配置管理的活动。

- (34) A. 变更标识      B. 变更控制      C. 质量控制      D. 版本控制

**【答案】C**

**【解析】**本题考查软件配置管理的基础知识。

软件配置管理是一组管理整个软件生存期各阶段中变更的活动，主要包括变更标识、变更控制和版本控制。

一个功能模块 M1 中的函数 F1 有一个参数需要接收指向整型的指针，但是在功能模块 M2 中调用 F1 时传递了一个整型值，在软件测试中，（35）最可能测出这一问题。

- (35) A. M1 的单元测试      B. M2 的单元测试  
C. M1 和 M2 的集成测试      D. 确认测试

**【答案】C**

**【解析】**本题考查软件测试基础知识。

一个功能模块 M1 中的函数 F1 有一个参数需要接收指向整型的指针，但是在功能模块 M2 中调用 F1 时传递了一个整型值，这种模块之间传递参数的错误，在集成测试中最可能测试出来。

- 【答案】C

McCabe 度量法是一种基于程序控制流的复杂性度量方法。采用这种方法先画出程序图，然后采用公式  $V(G)=m-n+2$  计算环路复杂度。其中， $m$  是图  $G$  中弧的个数， $n$  是图  $G$  中的结点数。图中结点数为 7，边数为 11，所以环路复杂度为  $11-7+2=6$ 。

(37) A. 继承是父类和子类之间共享数据和方法的机制

B. 继承定义了一种类与类之间的关系

C. 继承关系中的子类将拥有父类的全部属性和方法

D. 继承仅仅允许单重继承，即不允许一个子类有多个父类

- 【答案】D

面向对象技术中，继承是父类和子类之间共享数据和方法的机制。这是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上来进行，把这个已经存在的类所定义的内容作为自己的内容，并加入若干新的内容。可以存在多重继承的概念，但不同的程序设计语言可以有自己规定。

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(38)。绑定是一个把过程调用和响应调用所需要执行的代码加以结合的过程。在一般的程序设计语言中，绑定是在编译时进行的，叫做(39)；而(40)则是在运行时进行的，即一个给定的过程调用和代码的结合直到调用发生时才进行的。

(38) A. 继承                      B. 多态                      C. 动态绑定                      D. 静态绑定

(39) A. 继承                      B. 多态                      C. 动态绑定                      D. 静态绑定

(40) A. 继承                      B. 多态                      C. 动态绑定                      D. 静态绑定

**【答案】** B   D   C

**【解析】** 本题考查面向对象中的基本概念。

在收到消息时，对象要予以响应。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。绑定是一个把过程调用和响应调用所需要执行的代码加以结合的过程。在一般的程序设计语言中，绑定是在编译时进行的，叫做静态绑定。动态绑定则是在运行时进行的，因此，一个给定的过程调用和代码的结合直到调用发生时才进行。

动态绑定是和类的继承以及多态相联系的。在继承关系中，子类是父类的一个特例，所以父类对象可以出现的地方，子类对象也可以出现。因此在运行过程中，当一个对象发送消息请求服务时，要根据接收对象的具体情况将请求的操作与实现的方法进行连接，即动态绑定。

(41)不是面向对象分析阶段需要完成的。

(41) A. 认定对象    B. 组织对象  
C. 实现对象及其相互关系    D. 描述对象间的相互作用

**【答案】** C

**【解析】** 本题考查面向对象分析的基本概念。

面向对象分析包含 5 个活动：认定对象、组织对象、描述对象间的相互作用、定义对象的操作、定义对象的内部信息。

认定对象是指：在应用领域中，按自然存在的实体确立对象。在定义域中，首先将自然存在的“名词”作为一个对象，这通常是研究问题定义域实体的良好开始。通过实体间的

关系寻找对象常常没有问题，而困难在于寻找（选择）系统关心的实质性对象。实质性对象是系统稳定性的基础。例如在银行应用系统中，实质性对象应包含客户账务、清算等，而门卫值班表不是实质性对象，甚至可不包含在该系统中。

组织对象含义是：分析对象间的关系，将相关对象抽象成类，其目的是为了简化关联对象，利用类的继承性建立具有继承性层次的类结构。抽象类时可从对象间的操作或一个对象是另一个对象的一部分来考虑；如房子由门和窗构成，门和窗是房子类的子类。由对象抽象类，通过相关类的继承构造类层次，所以说系统的行为和信息间的分析过程 是一种迭代表征过程。

描述对象间的相互作用是：描述出各对象在应用系统中的关系。如一个对象是另一个对象的一部分，一个对象与其他对象间的通信关系等。这样可以完整地描述每个对象的环境，由一个对象解释另一个对象，以及一个对象如何生成另一个对象，最后得到对象的界面描述。实现对象及其相互关系应该归入到系统的实现阶段，不属于分析阶段的任务。

以下关于面向对象设计的叙述中，错误的是(42)。

(42)A. 面向对象设计应在面向对象分析之前，因为只有产生了设计结果才可对其进行分析

B. 面向对象设计与面向对象分析是面向对象软件过程中两个重要的阶段

C. 面向对象设计应该依赖于面向对象分析的结果

D. 面向对象设计产生的结果在形式上可以与面向对象分析产生的结果类似，例如都可以使用 UML 表达

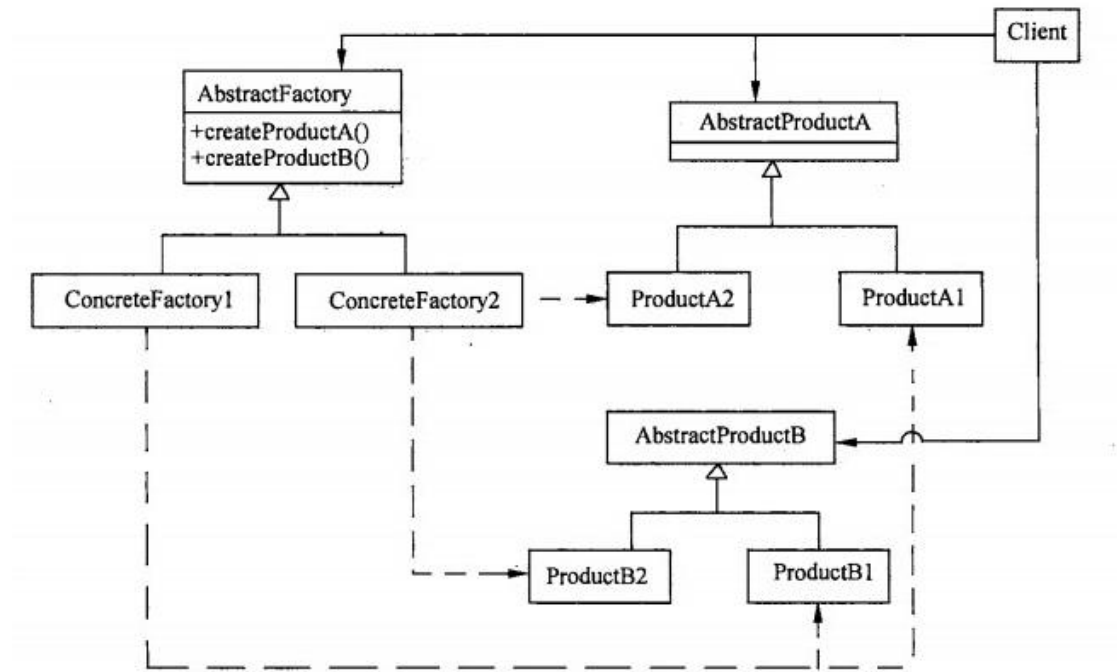
**【答案】A**

**【解析】**本题考查面向对象设计与面向对象分析的基本概念。

面向对象分析与设计是面向对象软件开发过程中的两个重要阶段，面向对象分析产生分析模型，该分析模型可以使用 UML 表达，面向对象设计以分析模型为基础，继续对分析模型进行精化，得到设计模型，其表达仍然可以采用 UML 建模语言。

下列 UML 类图表示的是(43)设计模式。关于该设计模式的叙述中，错误的是(44)。





(43) A. 工厂方法 B. 策略 C. 抽象工厂 D. 观察者

(44) A. 提供创建一系列相关或相互依赖的对象的接口，而无需指定这些对象所属的具体类

- B. 可应用于一个系统要由多个产品系列中的一个来配置的时候
- C. 可应用于强调一系列相关产品对象的设计以便进行联合使用的时候
- D. 可应用于希望使用已经存在的类，但其接口不符合需求的时候

【答案】C D

【解析】本题考查面向对象方法中的设计模式知识。

题中的类图是抽象工厂设计模式，该设计模式的意图是提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。使用抽象工厂设计模式的常见情形是：一个系统要独立于其产品的创建、组合和表示时；一个系统要由多个产品系列中的一个来配置时；当需要强调一系列相关的产品对象的设计以便进行联合使用时；当提供一个产品类库，而只想显示它们的接口而不是实现时。对于希望使用已经存在的类，但其接口不符合需求的情形，应当考虑桥接设计模式。

UML 类图中类与类之间的关系有五种：依赖、关联、聚合、组合与继承。若类 A 需要使用标准数学函数类库中提供的功能，那么类 A 与标准类库提供的类之间存在 (45) 关系；若类 A 中包含了其他类的实例，且当类 A 的实例消失时，其包含的其他类的实例也消失，则类

A 和它所包含的类之间存在 (46) 关系:若类 A 的实例消失时, 其他类的实例仍然存在并继续工作, 那么类 A 和它所包含的类之间存在 (47) 关系。

- |           |       |       |       |
|-----------|-------|-------|-------|
| (45)A. 依赖 | B. 关联 | C. 聚合 | D. 组合 |
| (46)A. 依赖 | B. 关联 | C. 聚合 | D. 组合 |
| (47)A. 依赖 | B. 关联 | C. 聚合 | D. 组合 |

【答案】A D C

【解析】本题考查 UML 类图之间类与类之间的关系。

UML 类图中类与类之间的关系有五种: 依赖、关联、聚合、组合与继承。依赖是几种关系中最弱的一种关系, 通常, 使用类库就是其中的一种关系。聚合与组合都表示整体和部分的联系。组合的程度比聚合高, 当整体对象消失时, 部分对象也随之消失, 则属于组合关系, 当整体对象消失而部分对象依然可以存在并继续被使用时, 则属于聚合关系。

以下关于高级语言程序的编译和解释的叙述中, 正确的是 (48)。

- (48)A. 编译方式下, 可以省略对源程序的词法分析、语法分析
- B. 解释方式下, 可以省略对源程序的词法分析、语法分析
- C. 编译方式下, 在机器上运行的目标程序完全独立于源程序
- D. 解释方式下, 在机器上运行的目标程序完全独立于源程序

【答案】C

【解析】本题考查语言处理基础知识。

编译和解释是语言处理的两种基本方式。编译过程包括词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等阶段, 以及符号表管理和出错处理模块。

解释过程在词法、语法和语义分析方面与编译程序的工作原理基本相同, 但是在运行用户程序时, 它直接执行源程序或源程序的内部形式。

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程序。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。

标记语言用一系列约定好的标记来对电子文档进行标记, 以实现对电子文档的语义、结构及格式的定义。(49)不是标记语言。

(49) A. HTML                      B. XML                      C. WML                      D. PHP

**【答案】D**

**【解析】** 本题考查程序语言基础知识。

HTML (Hypertext Marked Language, 超文本标记语言), 用于互联网的信息表示。用 HTML 编写的超文本文档称为 HTML 文档, 它能独立于各种操作系统平台 (如 UNIX, Windows 等)。HTML 文档是纯文本文档, 可以使用记事本、写字板等编辑工具来编写 HTML 文件, 其文件 (文档) 的扩展名是 .html 或 .htm, 它们需要通过 WWW 浏览器进行解释并显示出效果。

XML (Extensible Markup Language, 可扩展的标记语言) 1.0 标准于 1998 年 2 月 10 日发布, 被认为是继 HTML 和 Java 编程语言之后的又一个里程碑式的 Internet 技术。XML 丰富了 HTML 的描述功能, 可以描述非常复杂的 Web 页面, 如复杂的数字表达式、化学方程式等。XML 的特点是结构化、自描述、可扩展和浏览器自适应等。

用于 WAP 的标记语言就是 WML (Wireless Markup Language), 其语法跟 XML 一样, 是 XML 的子集。

PHP (Hypertext Preprocessor) 是一种在服务器端执行的、嵌入 HTML 文档的脚本语言, 其语言风格类似于 C 语言, 被网站编程人员广泛运用。

对于正规式  $0^*(10^*1)^*0^*$ , 其正规集中字符串的特点是 (50)。

- (50) A. 开头和结尾必须是 0                      B. 1 必须出现偶数次  
C. 0 不能连续出现                      D. 1 不能连续出现

**【答案】B**

**【解析】**

本题考查程序语言基础知识。

闭包运算符 “\*” 将其运算对象进行若干次连接, 因此  $0^*$  表示若干个 0 构成的串, 而  $(10^*1)^*$  则表示偶数个 1 构成的串。

确定系统边界和关系规范化分别在数据库设计的 (51) 阶段进行。

- (51) A. 需求分析和逻辑设计                      B. 需求分析和概念设计  
C. 需求分析和物理设计                      D. 逻辑设计和概念设计

**【答案】A**

**【解析】**

需求分析阶段的任务是：对现实世界要处理的对象（组织、部门、企业等）进行详细调查，在了解现行系统的概况，确定新系统功能的过程中，确定系统边界、收集支持系统目标的基础数据及其处理方法。

逻辑设计阶段的任务之一是对关系模式进一步的规范化处理。因为生成的初始关系模式并不能完全符合要求，会有数据冗余、更新异常存在，这就需要根据规范化理论对关系模式进行分解，以消除冗余和更新异常。不过有时根据处理要求，可能还需要增加部分冗余以满足处理要求。逻辑设计阶段的任务就需要作部分关系模式的处理，分解、合并或增加冗余属性，提高存储效率和处理效率。

若关系 R、S 如下图所示，则关系代数表达式  $\pi_{1,3,7}(\sigma_{3<6}(R \times S))$  与 (52) 等价。

A	B	C	D
1	2	4	6
2	3	3	1
3	4	1	3

R

C	D	E
3	4	2
8	9	3

S

- (52) A.  $\pi_{A,C,E}(\sigma_{C<D}(R \times S))$       B.  $\pi_{A,R,C,E}(\sigma_{R.C<S.D}(R \times S))$   
 C.  $\pi_{A,S,C,S,E}(\sigma_{R.C<S.D}(R \times S))$       D.  $\pi_{R,A,R,C,R,E}(\sigma_{R.C<S.D}(R \times S))$

【答案】B

【解析】本题考查关系代数运算方面的基础知识。

本题要求关系代数表达式  $\pi_{1,3,7}(\sigma_{3<6}(R \times S))$  的结果集，其中， $R \times S$  的属性列名分别为：R.A，R.B，R.C，R.D，S.C，S.D 和 S.E，其结果如下表所示：

R.A	R.B	R.C	R.D	S.C	S.D	S.E
1	2	4	6	3	4	2
1	2	4	6	8	9	3
2	3	3	1	3	4	2
2	3	3	1	8	9	3
3	4	1	3	3	4	2
3	4	1	3	8	9	3
R×S						

$\sigma_{3<6}(R \times S)$  的含义是从  $R \times S$  结果集中选取第三个分量 (R.C) 小于第六个分量 (S.D) 的元组，故  $\sigma_{3<6}(R \times S)$  与  $\sigma_{R.C < S.D}(R \times S)$  等价。从上表中可以看出，满足条件的结果如下表所示：

R.A	R.B	R.C	R.D	S.C	S.D	S.E
1	2	4	6	8	9	3
2	3	3	1	3	4	2
2	3	3	1	8	9	3
3	4	1	3	3	4	2
3	4	1	3	8	9	3
$\sigma_{3<6}(R \times S)$						

某销售公司数据库的零件 P (零件号，零件名称，供应商，供应商所在地，库存量) 关系如下表所示，其中同一种零件可由不同的供应商供应，一个供应商可以供应多种零件。零件关系的主键为 (53)。

零件号	零件名称	供应商	供应商所在地	单价(元)	库存量
010023	P2	S1	北京市海淀区 58 号	22.80	380
010024	P3	S1	北京市海淀区 58 号	280.00	1350
010022	P1	S2	陕西省西安市雁塔区 2 号	65.60	160
010023	P2	S2	陕西省西安市雁塔区 2 号	28.00	1280
010024	P3	S2	陕西省西安市雁塔区 2 号	260.00	3900
010022	P1	S3	北京市新城区 65 号	66.80	2860
...	...	...	...	...	...

查询各种零件的平均单价、最高单价与最低单价之间差距的 SQL 语句为：

SELECT 零件号, (54)  
FROM P  
(55) ;

该关系存在冗余以及插入异常和删除异常等问题，为了解决这一问题需要将零件关系分解为 (56)。



素  $A[i]$  时, 最多与  $A$  中的 (57) 个元素进行比较。

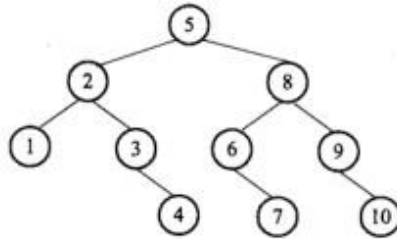
- (57) A.  $n$       B.  $\lfloor \log_2 n \rfloor - 1$       C.  $n/2$       D.  $\lfloor \log_2 n \rfloor + 1$

【答案】D

【解析】

本题考查数据结构基础知识。

二分查找是一种效率较高的查找方法, 在 10 个元素构成的有序表中进行二分查找的过程可用二分查找判定树表示, 如下图所示:



其中, 结点中的数字表示元素在表中的序号。以结点 10 为例, 它所在的位置说明若要查找表中的第 10 个元素, 则依次与第 5 个、第 8 个、第 9 个和第 10 个元素进行了比较。若有序表中有  $n$  个元素, 则对其进行二分查找的判定树的高度为  $\lfloor \log_2 n \rfloor + 1$  (与具有  $n$  个结点的完全二叉树高度一样), 因此, 查找过程中最多与  $\lfloor \log_2 n \rfloor + 1$  个元素进行比较。

设有如下所示的下三角矩阵  $A[0..8, 0..8]$ , 将该三角矩阵的非零元素 (即行下标不小于列下标的所有元素) 按行优先压缩存储在数组  $M[1..m]$  中, 则元素  $A[i, j]$  ( $0 \leq i \leq 8, j \leq i$ ) 存储在数组  $M$  的 (58) 中。

$$\begin{bmatrix} A_{0,0} & & & & & & & & \\ A_{1,0} & A_{1,1} & & & & & & & \\ \cdot & & \cdot & & & & 0 & & \\ \cdot & & & \cdot & & & & & \\ \cdot & & & & \cdot & & & & \\ A_{7,0} & A_{7,1} & A_{7,2} & \cdots & A_{7,7} & & & & \\ A_{8,0} & A_{8,1} & A_{8,2} & A_{8,3} & \cdots & & & & A_{8,8} \end{bmatrix}$$

- (58) A.  $M[\frac{i(i+1)}{2} + j + 1]$       B.  $M[\frac{i(i+1)}{2} + j]$   
C.  $M[\frac{i(i-1)}{2} + j]$       D.  $M[\frac{i(i-1)}{2} + j + 1]$

【答案】A

**【解析】**

本题考查数据结构基础知识。

如题图所示，按行方式压缩存储时， $A[i,j]$ 之前的元素数目为 $(1+2+\cdots+i+j)$ 个，数组M的下标从1开始，因此 $A[i,j]$ 的值存储在 $M[\frac{i(i+1)}{2}+j+1]$ 中。

若用n个权值构造一棵最优二叉树（哈夫曼树），则该二叉树的结点总数为(59)。

- (59) A.  $2n$                       B.  $2n-1$                       C.  $2n+1$                       D.  $2n+2$

**【答案】B**

**【解析】** 本题考查数据结构基础知识。

二叉树具有以下性质：度为2的结点（双分支结点）数比度为0（叶子结点）数正好少1。而根据最优二叉树（哈夫曼树）的构造过程可知，最优二叉树中只有度为2和0的结点，因此，其结点总数为 $2n-1$ 。

栈是一种按“后进先出”原则进行插入和删除操作的数据结构，因此，(60)必须用栈。

- (60) A. 实现函数或过程的递归调用及返回处理时  
B. 将一个元素序列进行逆置  
C. 链表结点的申请和释放  
D. 可执行程序的装入和卸载

**【答案】A**

**【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。将一个元素序列逆置时，可以使用栈也可以不用。链表结点的申请和释放次序与应用要求相关，不存在“先申请后释放”的操作要求。可执行程序的装入与卸载，也不存在“后进先出”的操作要求。对于函数的递归调用与返回，一定是后被调用执行的先返回。

对以下四个序列用直接插入排序方法由小到大进行排序时，元素比较数最少的是 (61)。

- (61) A. 89, 27, 35, 78, 41, 15                      B. 27, 35, 41, 16, 89, 70  
C. 15, 27, 46, 40, 64, 85                      D. 90, 80, 45, 38, 30, 25

**【答案】C**

**【解析】** 本题考查数据结构基础知识。



当序列基本有序时，直接插入排序过程中元素比较的次数较少，当序列为逆序时，元素的比较次数最多。

对于哈希表，如果将装填因子  $a$  定义为表中装入的记录数与表的长度之比，那么向表中加入新记录时，(62)。

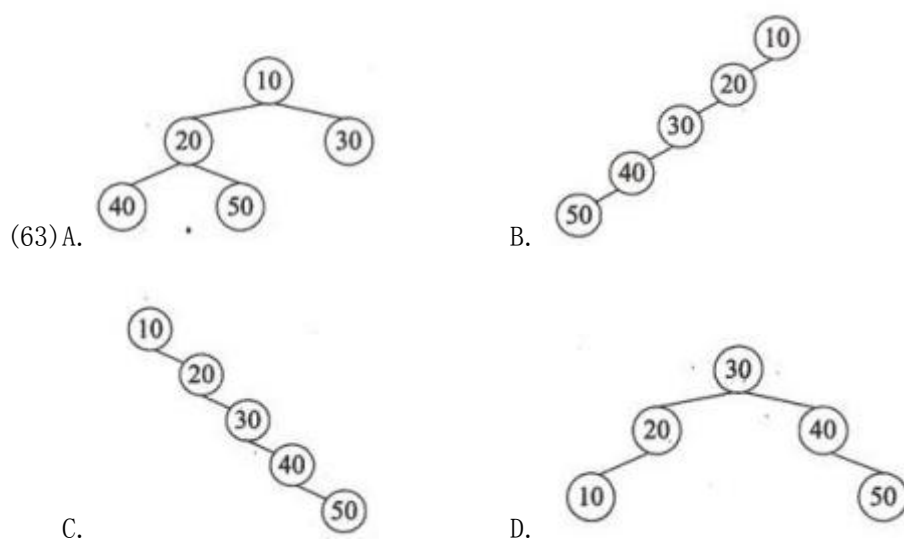
- (62) A.  $a$  的值随冲突次数的增加而递减    B.  $a$  越大发生冲突的可能性就越大  
C.  $a$  等于 1 时不会再发生冲突    D.  $a$  低于 0.5 时不会发生冲突

【答案】B

【解析】本题考查数据结构基础知识。

装填因子  $a$  表示了哈希表的装满程度，显然， $a$  越大发生冲突的可能性就越大。

用关键字序列 10、20、30、40、50 构造的二叉排序树（二叉查找树）为(63)。



【答案】C

【解析】本题考查数据结构基础知识。

根据关键字序列构造二叉排序树的基本过程是，若需插入的关键字大于树根，则插入到右子树上，若小于树根，则插入到左子树上，若为空树，则作为树根结点。

若某算法在问题规模为  $n$  时，其基本操作的重复次数可由下式表示，则该算法的时间复杂度为(64)。

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n-1) + n & n > 1 \end{cases}$$

(64) A.  $O(n)$

B.  $O(n^2)$

C.  $O(\log n)$

D.  $O(n \log n)$

**【答案】B**

**【解析】** 本题考查算法分析与设计基础知识。

根据题中给出的递归定义式进行推导, 可得  $T(n) = n + n-1 + \dots + 2 + 1$ , 因此时间复杂度为  $O(n^2)$ 。

若对一个链表最常用的操作是在末尾插入结点和删除尾结点, 则采用仅设尾指针的单向循环链表 (不含头结点) 时, (65)。

(65) A. 插入和删除操作的时间复杂度都为  $O(1)$

B. 插入和删除操作的时间复杂度都为  $O(n)$

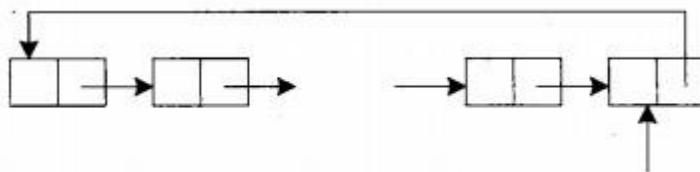
C. 插入操作的时间复杂度为  $O(1)$ , 删除操作的时间复杂度为  $O(n)$

D. 插入操作的时间复杂度为  $O(n)$ , 删除操作的时间复杂度为  $O(1)$

**【答案】C**

**【解析】** 本题考查数据结构与算法基础知识。

设尾指针的单向循环链表 (不含头结点) 如下图所示:



设结点的指针域为 `next`, 新结点的指针为 `s`, 则在尾指针所指结点后插入结点的操作为:

```
s->next = t->next; t->next = s; t = s;
```

也就是插入操作的时间复杂度为  $O(1)$ 。

要删除尾指针所指结点, 必须通过遍历操作找到尾结点的前驱结点, 其操作序列如下:

也就是说, 删除操作的时间复杂度为  $O(n)$ 。



通过该链接可以在 Internet 中发送电子邮件。

POP3 服务默认的 TCP 端口号是 (70)。

(70) A. 20                      B. 25                      C. 80                      D. 110

**【答案】D**

**【解析】**

本试题考查 POP3 服务器的配置。POP3 服务器默认端口为 110, 故选 D。

Observe that for the programmer, as for the chef, the urgency of the patron (顾客) may govern the scheduled completion of the task, but it cannot govern the actual completion. An omelette (煎鸡蛋), promised in two minutes, may appear to be progressing nicely. But when it has not set in two minutes, the customer has two choices—waits or eats it raw. Software customers have had (71) choices.

Now I do not think software (72) have less inherent courage and firmness than chefs, nor than other engineering managers. But false (73) to match the patron's desired date is much more common in our discipline than elsewhere in engineering. It is very (74) to make a vigorous, plausible, and job risking defense of an estimate that is derived by no quantitative method, supported by little data, and certified chiefly by the hunches of the managers.

Clearly two solutions are needed. We need to develop and publicize productivity figures, bug-incidence figures, estimating rules, and so on. The whole profession can only profit from (75) such data. Until estimating is on a sounder basis, individual managers will need to stiffen their backbones and defend their estimates with the assurance that their poor hunches are better than wish derived estimates.

(71) A. no	B. the same	C. other	D. lots of
(72) A. testers	B. constructors	C. managers	D. architects
(73) A. tasks	B. jobs	C. works	D. scheduling
(74) A. easy	B. difficult	C. simple	D. painless
(75) A. sharing	B. excluding	C. omitting	D. ignoring

**【答案】B C D B A**

## 【解析】

观察一下编程人员，你可能会发现，同厨师一样，某项任务的计划进度，可能受限于顾客要求的紧迫程度，但紧迫程度无法控制实际的完成情况。就像约好在两分钟内完成一个煎蛋，看上去可能进行得非常好。但当它无法在两分钟内完成时，顾客只能选择等待或者生吃煎蛋。软件顾客的情况类似。

我现在并不认为软件经理内在的勇气和坚持不如厨师，或者不如其他工程经理。但为了满足顾客期望的日期而造成的不合理进度安排，在软件领域中却比其他的任何工程领域要普遍得多。而且，非量化方法的采用，少得可怜的数据支持，加上完全借助软件经理的直觉，这样的方式很难生产出健壮可靠和规避风险的估计。

显然我们需要两种解决方案。开发并推行生产率图表、缺陷率、估算规则等等，整个组织最终会从这些数据的共享上获益。或者在基于可靠基础的估算出现之前，项目经理需要挺直腰杆并坚持他们的估计，确信自己的经验和直觉总比从期望得出的估计要强得多。

试题一

某大型企业的数据中心为了集中管理、控制用户对数据的访问并支持大量的连接需求，欲构建数据管理中间件，其主要功能如下：

（1）数据管理员可通过中间件进行用户管理、操作管理和权限管理。用户管理维护用户信息，用户信息（用户名、密码）存储在用户表中；操作管理维护数据实体的标准操作及其所属的后端数据库信息，标准操作和后端数据库信息存放在操作表中；权限管理维护权限表，该表存储用户可执行的操作信息。

（2）中间件验证前端应用提供的用户信息。若验证不通过，返回非法用户信息；若验证通过，中间件将等待前端应用提交操作请求。

（3）前端应用提交操作请求后，中间件先对请求进行格式检查。如果格式不正确，返回格式错误信息；如果格式正确，则进行权限验证（验证用户是否有权执行请求的操作），若用户无权执行该操作，则返回权限不足信息，否则进行连接管理。

（4）连接管理连接相应的后台数据库并提交操作，连接管理先检查是否存在空闲的数据库连接，如果不存在，新建连接；如果存在，则重用连接。

（5）后端数据库执行操作并将结果传给中间件，中间件对收到的操作结果进行处理后，将其返回给前端应用。

现采用结构化方法对系统进行分析与设计，获得如图 1-1 所示的顶层数据流图和图 1-2 所示的 0 层数据流图。

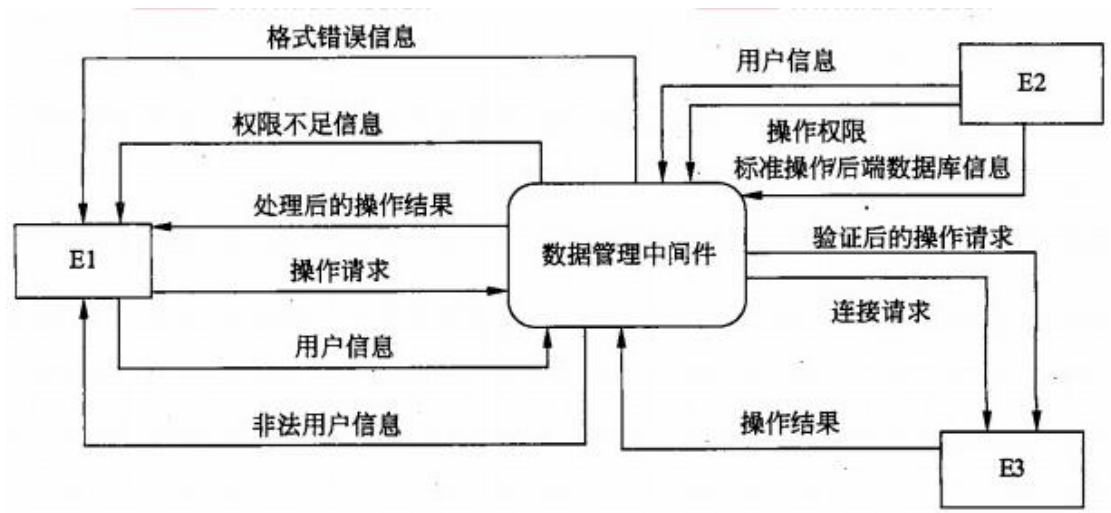


图 1-1 顶层数据流图

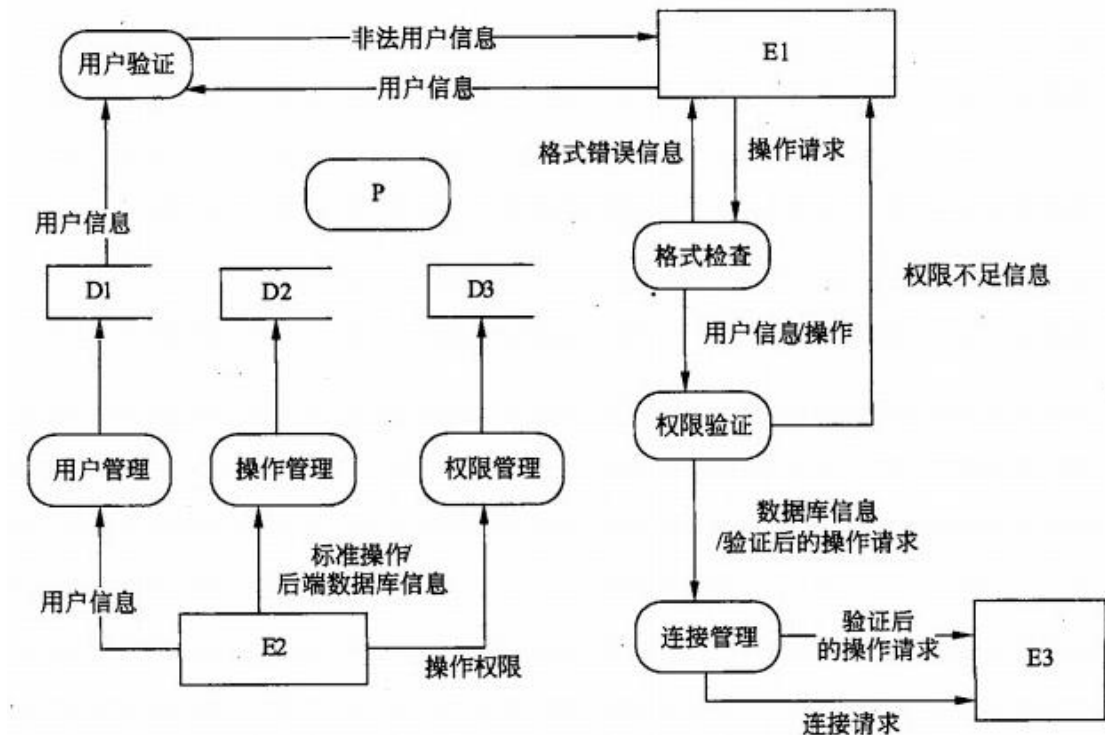


图 1-2 0 层数据流图

### 【问题 1】

使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

本问题考查顶层 DFD。顶层 DFD 一般用来确定系统边界，将待开发系统看作一个加工，因此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。题目要求根据描述确定图中的外部实体。分析题目中的描述，并结合已经在顶层数据流图中给出的数据流进行分析。题目中有信息描述：数据管理员可通过中间件进行用户管理、操作管理和权限管理；前端应用提交操作请求；连接管理连接相应的后台数据库并提交操作。由此可知该中间件系统有数据管理员、前端应用和后端数据库三个外部实体。从图 1-1 中数据流和实体的对应关系可知，E1 为前端应用，E2 为数据管理员，E3 为后端数据库。

### 【问题 2】

使用说明中的词语，给出图 1-2 中的数据存储 D1~D3 的名称。

本问题考查 0 层 DFD 中数据存储的确定。说明中描述：用户信息（用户名、密码）存储在用户表中；标准操作和后端数据库信息存放在操作表中；权限管理维护信息存放在权限表中。因此数据存储为用户表、操作表以及权限表。再根据图 1-2 可知 D1 的输入数据流从用户管理来，D2 的输入数据流从操作管理来，D3 的输入数据流从权限管理来，所以 D1 为用户表，D2 为操作表，D3 为权限表。

## 【问题 2】

给出图 1-2 中加工 P 的名称及其输入、输出流。

	名 称	起 点	终 点
输入流			P
输出流		P	

除加工 P 的输入与输出流外，图 1-2 还缺失了两条数据流，请给出这两条数据流的起点和终点。

起 点	终 点

注：名称使用说明中的词汇，起点和终点均使用图 1-2 中的符号或词汇。

P 的名称：操作结果处理

	名 称	起 点	终 点
输入流	操作结果	E3	P
输出流	处理后的操作结果	P	E1

缺少的数据流：

起 点	终 点
D2	权限验证
D3	权限验证

本问题考查 0 层 DFD 中缺失的加工和数据流。比较图 1-1 和图 1-2, 可知顶层 DFD 中的操作结果和处理后的操作结果没有在 0 层 DFD 中体现。再根据描述“后端数据库执行操作并将结果传给中间件，中间件对收到的操作结果进行处理后，将其返回给前端应用”可知，需要有操作结果处理，因此 P 为操作结果处理，其输入流为从后端数据库 E3 来的操作结果，输出结果为处理后的操作结果，并返回给前端应用 E1。

考查完 P 及其输入输出流之后，对图 1-2 的内部数据流进行考查，以找出缺失的另外 2 条数据流。从图中可以看出 D2 和 D3 只有输入流没有输出流，这是常见 DFD 设计时的错误，所以首先考查 D2 和 D3 的输出流。描述中有“权限验证是验证用户是否有权执行请求的操作，若用户有权执行该操作，进行连接管理；连接管理连接相应的后台数据库并提交操作；权限表存储用户可执行的操作信息”。因此，权限验证有从权限表 D3 来的输入数据流。而要连接后端数据库，需要数据库信息，从权限验证的输出流中包含有数据库信息可知，权限验证需要获取到数据库信息，所以还需从操作表 D2 来的输入流。



**【问题 3】**

在绘制数据流图时，需要注意加工的绘制。请给出三种在绘制加工的输入、输出时可能出现的错误。

在绘制数据流图的加工时，可能出现的输入、输出错误：

只有输入而无输出 或者 黑洞

只有输出而无输入 或者 奇迹

输入的数据流无法通过加工产生输出流 或者 灰洞

输入的数据流与输出的数据流名称相同

试题二

某学校拟发一套实验管理系统，对各课程的实验安排情况进行管理。

【需求分析】

一个实验室可进行多种类型不同的实验。由于实验室和实验员资源有限，需根据学生人数分批次安排实验室和实验员。一门课程可以为多个班级开设，每个班级每学期可以开设多门课程。一门课程的一种实验可以根据人数、实验室的可容纳人数和实验类型，分批次开设在多个实验室的不同时间段。一个实验室的一次实验可以分配多个实验员负责辅导实验，实验员给出学生的每次实验成绩。

(1) 课程信息包括：课程编号、课程名称、实验学时、授课学期和开课的班级等信息；实验信息记录该课程的实验进度信息，包括：实验名、实验类型、学时、安排周次等信息，如表 2-1 所示。

表 2-1 课程及实验信息

课程编号	15054037	课程名称	数字电视原理		实验学时	12
班级	电 0501,信 0501,计 0501	授课院系	机械与电气工程		授课学期	第三学期
序号	实验名		实验类	难度	学时	安排周次
1505403701	音视频 AD-DA 实验		验证性	1	2	3
1505403702	音频编码实验		验证性	2	2	5
1505403703	视频编码实验		演示性	0.5	1	9

(2) 以课程为单位制定实验安排计划信息，包括：实验地点，实验时间、实验员等信息，实验计划如表 2-2 所示。

表 2-2 实验安排计划

课程编号	15054037	课程名称	数字电视原理	安排学期	2009 年秋	总人数	220
实验编号	实验名	实验员	实验时间	地点	批次号	人数	
1505403701	音视频 AD-DA 实验	盛×，陈×	第 3 周周四晚上	实验三楼 310	1	60	
1505403701	音视频 AD-DA 实验	盛×，陈×	第 3 周周四晚上	实验三楼 310	2	60	
1505403701	音视频 AD-DA 实验	吴×，刘×	第 3 周周五晚上	实验三楼 311	3	60	
1505403701	音视频 AD-DA 实验	吴×	第 3 周周五晚上	实验三楼 311	4	40	
1505403702	音频编码实验	盛×，刘×	第 5 周周一下午	实验四楼 410	1	70	

(3) 由实验员给出每个学生每次实验的成绩，包括：实验名、学号、姓名、班级、实验成绩等信息，实验成绩如表 2-3 所示。

表 2-3 实验成绩

实验员： 盛×

实验名	音视频 AD-DA 实验	课程名	数字电视原理
学号	姓名	班级	实验成绩
030501001	陈民	信 0501	87
030501002	刘志	信 0501	78
040501001	张勤	计 0501	86

(4) 学生的实验课程总成绩根据每次实验的成绩以及每次实验的难度来计算。

### 【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。

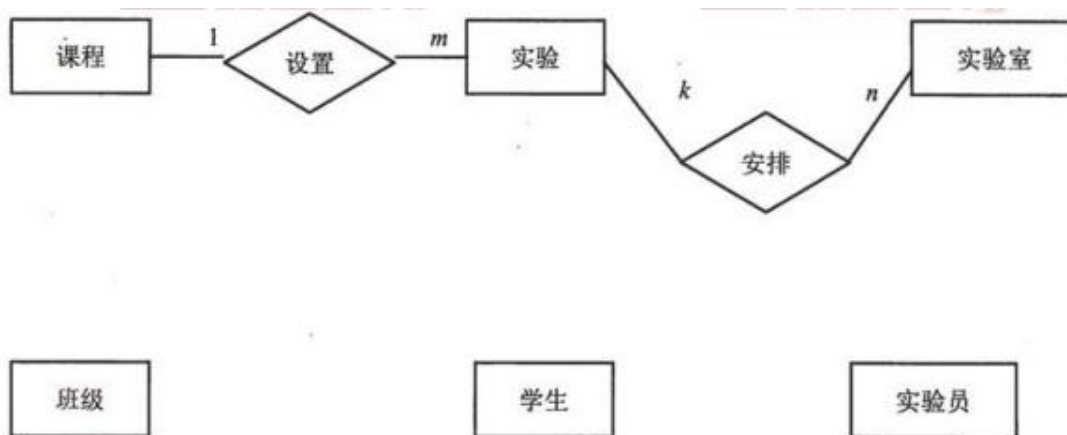


图 2-1 实体联系图

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）：

课程（课程编号，课程名称，授课院系，实验学时）

班级（班级号，专业，所属系）

开课情况（          (1)          ，授课学期）

实验（          (2)          ，实验类型，难度，学时，安排周次）

实验计划（          (3)          ，实验时间，人数）

实验员（          (4)          ，级别）

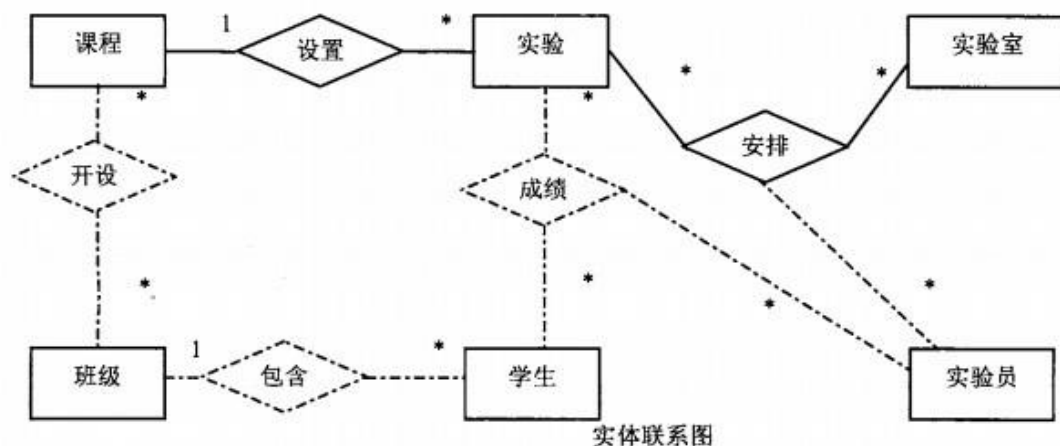
实验室（实验室编号，地点，开放时间，可容纳人数，实验类型）

学生（          (5)          ，姓名，年龄，性别）

实验成绩（          (6)          ，实验成绩，评分实验员）

### 【问题 1】

补充图 2-1 中的联系和联系的类型。



根据题意，由“一门含实验的课程可以开设给多个班级，每个班级每学期可以开设多门含实验的课程”可知课程和班级之间的开设关系为  $m:n$  联系。由“一个实验室的一次实验可以分配多个实验员负责辅导实验”可知实验、实验室与实验员之间的安排关系为  $k:n:m$  联系。由“实验员给出学生的每次实验成绩”可知实验、学生与实验员之间的成绩关系为联系。班级和学生之间的包含关系为  $1:n$  联系。

### 【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)~(6) 补充完整并用下划线指出这六个关系模式的主键。

- (1) 课程编号, 班级号
- (2) 实验编号, 课程编号
- (3) 实验编号, 批次号, 安排学期, 实验室编号, 实验员编号
- (4) 实验员编号, 实验员姓名
- (5) 学号, 班级号
- (6) 实验编号, 学号

其他关系模式主键:

课程 (课程编号, 课程名称, 授课院系, 实验学时)

班级 (班级号, 专业, 所属系)

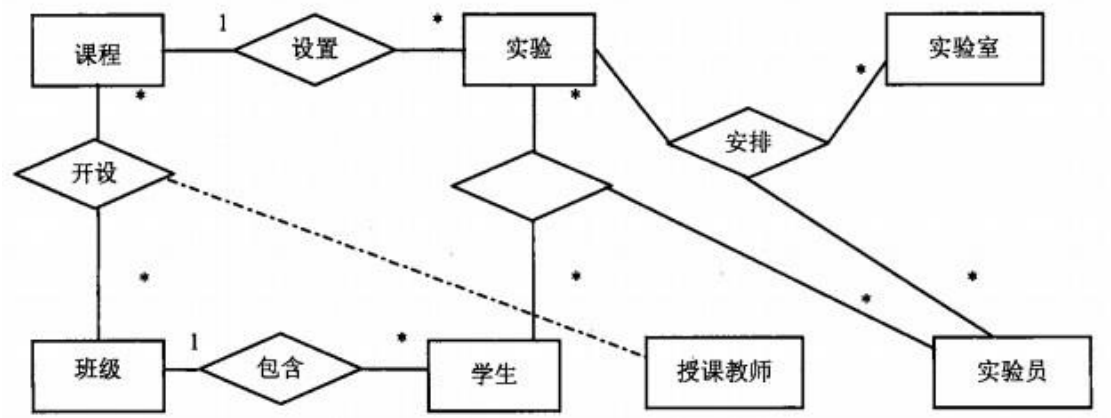
实验室 (实验室编号, 地点, 开放时间, 可容纳人数, 实验课类型)

实验室(实验室编号, 地点, 开放时间, 可容纳人数, 实验课类型)根据题意可知课程编号是课程的主键, 班级号是班级的主键。从表 2-1 可知, 开课情况是体现课程与班级间的  $m:n$  联系, 因此开课情况关系模式应该包含课程编号和班级号, 并共同作为主键。一门课程包含多次实验, 实验与课程之间是  $m:1$  关系, 因此, 根据表 2-1, 实验关系模式应包含实验编号和课程编号, 并且以实验编号为主键, 以课程编号为外键。在制定试验计划时, 每个班的每次实验可能按实验室被分成多个批次, 每个批次的实验会有若干名实验员来辅导学生实

实验并打分。实验员关系模式应该记录实验员编号和实验员姓名，并以实验员编号为主键。实验室编号是实验室的主键。从表 2-2 可见，实验计划关系模式应记录实验编号、批次号和授课学期，并且共同作为主键。从表 2-3 可见，实验成绩关系模式记录每个学生的每次实验成绩，应包含学号和实验编号，并共同作为主键。

【问题 3】

如果需要记录课程的授课教师，新增加“授课教师”实体。请对图 2-1 进行修改，画出修改后的实体间联系和联系的类型。



由于授课教师负责给若干个班级开设若干门课程，因此，课程、班级和授课教师之间的开设关系是联系。

试题三

某运输公司决定为新的售票机开发车票销售的控制软件。图 3-1 给出了售票机的面板示意图以及相关的控制部件。

售票机相关部件的作用如下所述：

- (1) 目的地键盘用来输入行程目的地的代码（例如，200 表示总站）。
- (2) 乘客可以通过车票键盘选择车票种类（单程票、多次往返票和座席种类）。
- (3) 继续/取消键盘上的取消按钮用于取消购票过程，继续按钮允许乘客连续购买多张票。
- (4) 显示屏显示所有的系统输出和用户提示信息。
- (5) 插卡口接受 MCard（现金卡），硬币口和纸币槽接受现金。
- (6) 打印机用于输出车票。

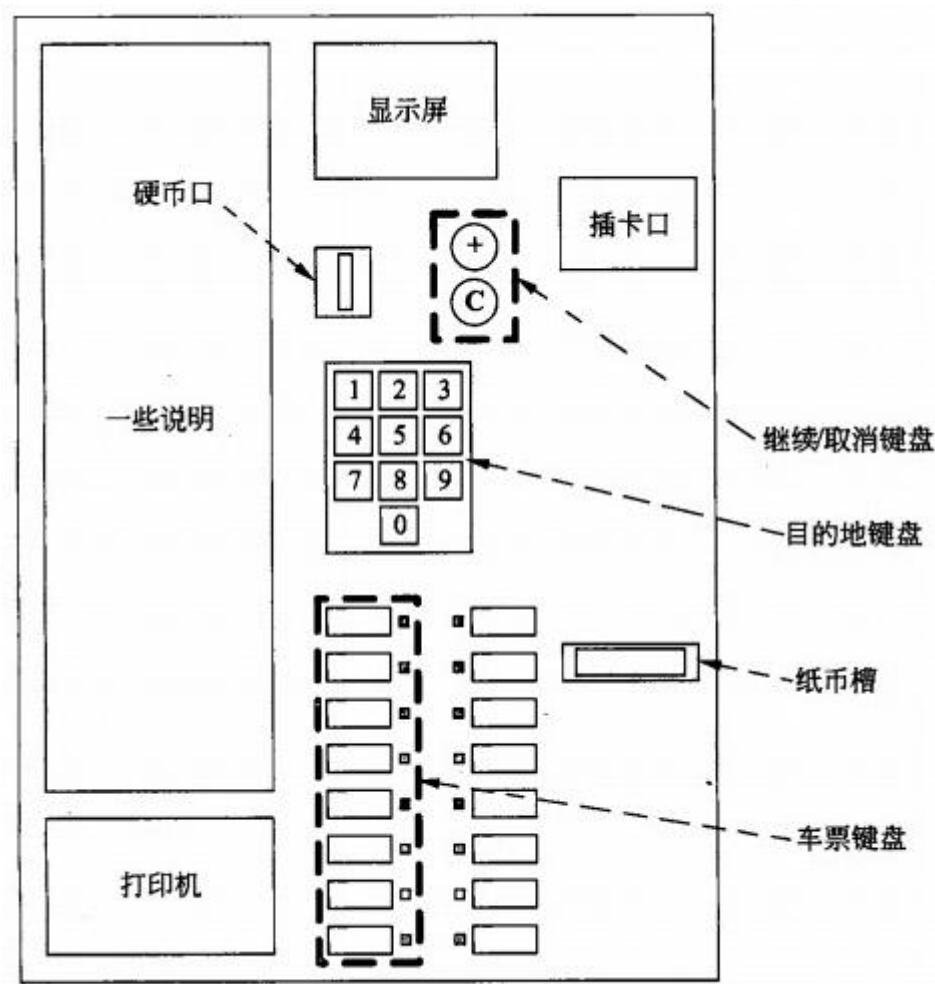


图 3-1 售票机面板示意图

假设乘客总是支付恰好需要的金额而无需找零，售票机的维护工作（取回现金、放入空白车票等）由服务技术人员完成。

系统采用面向对象方法开发/使用 UML 进行建模。系统的顶层用例图和类图分别如图 3-2 和图 3-3 所示。

系统采用面向对象方法开发/使用 UML 进行建模。系统的顶层用例图和类图分别 如图 3-2 和图 3-3 所示。

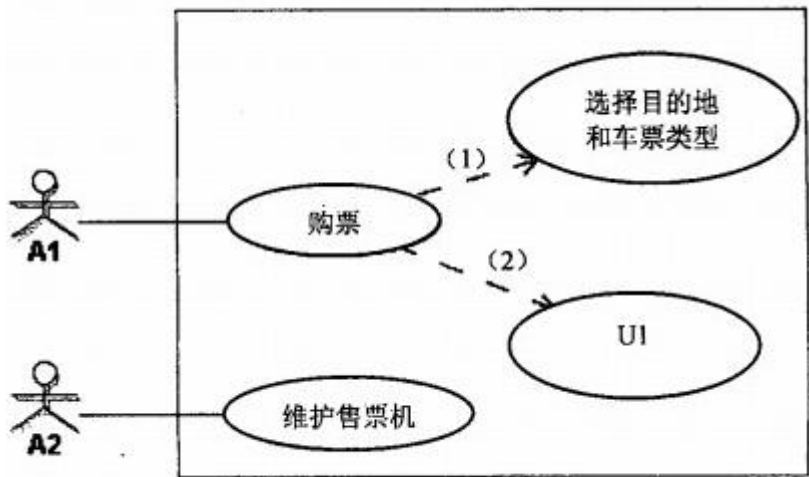


图 3-2 顶层用例图

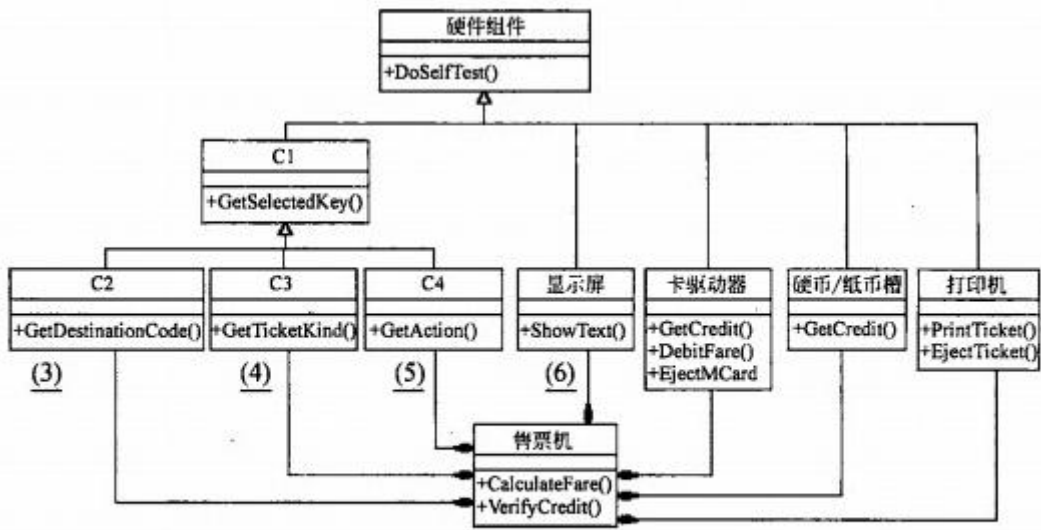


图 3-3 类图

【问题 1】

根据说明中的描述，给出图 3-2 中 A1 和 A2 所对应的参与者，U1 所对应的用例，以及 (1)、(2) 处所对应的关系。

A1: 乘客 A2:服务技术人员

U1: 支付 (1) «include» (2) «include»

本问题考查用例图。用例图用于确定系统边界，识别与系统交互的参与者，通过判断参与者发起的用例，建立和参与者之间的关联，然后再确认用例之间的关系。

本题中对售票机的描述为“乘客可以通过车票键盘选择车票种类（单程票、多次往返票和座席种类）；售票机的维护工作（取回现金、放入空白车票等）由服务技术人员完成”。由此可知，图 3-1 中 A1 为乘客，A2 为服务技术人员。

对购票用例，要选择目的地和车票类型、通过插卡口进行支付才可完成购票。因此 U2 为支付。

在考查用例之间的关系时，购票过程可以取消，也允许乘客连续购买多张票，因此，购票时可以包含多次选择目的地和车票类型、支付，即购票用例包含（关系《include》）选择目的地和车票类型以及支付。

### 【问题 2】

根据说明中的描述，给出图 3-3 中缺少的 C1~C4 所对应的类名以及（3）~（6）处所对应的多重度。

C1：键盘 C2：目的地键盘 C3：车票键盘 C4：继续/取消键盘

（3）~（6）：1

本问题考查类图。类图设计的重点是类的抽象和继承关系以及多重度。售票机的面板由多个控制部件组成。根据说明这些控制部件有目的地键盘、车票键盘和继续/取消键盘、显示屏、卡驱动器、硬币/纸币槽、打印机。图 3-3 中只有前 3 个部件在图中没有给出，而要填如 4 个类。从图中已经抽象出的硬件组件，给出了抽象的思路，从而可以把键盘抽象出来。由 C1 与 C2、C3、C4 的继承关系中 C1 为基类，可知 C1 为键盘。由 C2、C3 和 C4 给出的方法名称可知，C2 为目的地键盘获取目的地代码，C3 为车票键盘选择产品类型，C4 为继续/和取消动作。

本题中的重复度比较简单。从图 3-1 售票机的图示中可以看出，一个售票机只包含一个目的地键盘、一个车票键盘和一个继续/取消键盘，因此（3）~（6）均为 1。

### 【问题 3】

图 3-3 中的类图设计采用了中介者（Mediator）设计模式，请说明该模式的内涵。

本问题考查设计模式。设计模式题目虽然比较难，但是本题题目中已经给出了所采用的



设计模式为 Mediator 模式，只需说明设计模式的内涵即可，也比较容易。使用 Mediator 模式，可以使各个对象间的耦合松散，只需关心和 Mediator 的关系，使多对多的关系变成了一对多的关系，可以降低系统的复杂性，提高可修改扩展性。

试题四

对有向图进行拓扑排序的方法是：

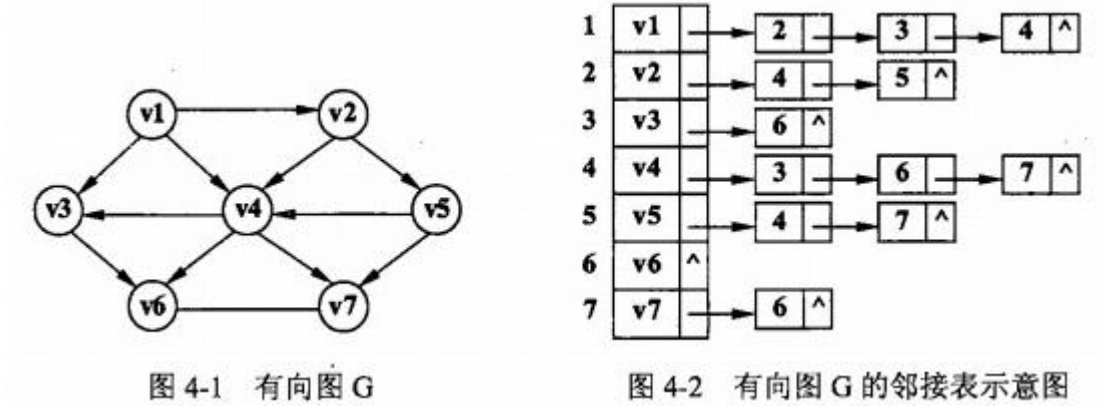
- (1) 初始时拓扑序列为空；
- (2) 任意选择一个入度为 0 的顶点，将其放入拓扑序列中，同时从图中删除该顶点 以及从该顶点出发的弧；
- (3) 重复 (2)，直到不存在入度为 0 的顶点为止（若所有顶点都进入拓扑序列则完 成拓扑排序，否则由于有向图中存在回路无法完成拓扑排序）。

函数 `int* TopSort(LinkedDigraph G)` 的功能是对有向图 `G` 中的顶点进行拓扑排序，返回拓 扑序列中的顶点编号序列，若不能完成拓扑排序，则返回空指针。其中，图 `G` 中的顶点从 1 开始依次编号，顶点序列为 `v1, v2, ..., vn`, 图 `G` 采用邻接表表示，其数据类型定义如下：

```
#define MAXVNUM 50
typedef struct ArcNode{
    int adjvex;
    struct ArcNode *nextarc;
}ArcNode;
typedef struct AdjList{
    char vdata;
    ArcNode *firstarc;
}AdjList;
typedef struct LinkedDigraph{
    int n;
    AdjList Vhead[MAXVNUM];
}LinkedDigraph;
```

/\* 最大顶点数 \*/  
/\* 表结点类型 \*/  
/\* 邻接顶点编号 \*/  
/\* 指示下一个邻接顶点 \*/  
  
/\* 头结点类型 \*/  
/\* 顶点的数据信息 \*/  
/\* 指向邻接表的第一个表结点 \*/  
  
/\* 图的类型 \*/  
/\* 图中顶点个数 \*/  
/\* 所有顶点的头结点数组 \*/

例如，某有向图 `G` 如图 4-1 所示，其邻接表如图 4-2 所示。



函数 `TopSort` 中用到了队列结构（`Queue` 的定义省略），实现队列基本操作的函数原型 如下表所示：

函数原型	说 明
void InitQueue(Queue *Q)	初始化队列（构造一个空队列）
bool IsEmpty(Queue Q)	判断队列是否为空，若是则返回 true，否则返回 false
void EnQueue(Queue *Q, int e)	元素入队列
void DeQueue(Queue *Q, int *p)	元素出队列

### 【C 代码】

```

int *TopSort(LinkedDigraph G) {
    ArcNode *p;                /* 临时指针，指示表结点 */
    Queue Q;                    /* 临时队列，保存入度为 0 的顶点编号 */
    int k = 0;                  /* 临时变量，用作数组元素的下标 */
    int j = 0, w = 0;           /* 临时变量，用作顶点编号 */
    int *topOrder, *inDegree;
    topOrder = (int *)malloc((G.n+1) * sizeof(int));
                                /* 存储拓扑序列中的顶点编号 */
    inDegree = (int *)malloc((G.n+1) * sizeof(int));
                                /* 存储图 G 中各顶点的入度 */
    if (!inDegree || !topOrder) return NULL;

    (1);                        /* 构造一个空队列 */

    for ( j = 1; j <= G.n; j++ ) { /* 初始化 */
        topOrder[j] = 0;    inDegree[j] = 0;
    }
}

```

```

}

for (j = 1; j <= G.n; j++) /* 求图 G 中各顶点的入度 */
for( p = G.Vhead[j].firstarc; p; p = p->nextarc )
    inDegree[p-> adjvex] += 1;

for (j = 1; j <= G.n; j++) /* 将图 G 中入度为 0 的顶点保存在队列中 */
if ( 0 == inDegree[j] )    EnQueue(&Q,j);

while (!IsEmpty(Q)) {
    (2); /* 队头顶点出队列并用 w 保存该顶点的编号 */
    topOrder[k++] = w;
    /* 将顶点 w 的所有邻接顶点的入度减 1 (模拟删除顶点 w 及从该顶点出发的弧的操作) */
    for(p = G.Vhead[w].firstarc; p; p = p->nextarc) {
        (3) -= 1;
        if (0 == (4))    EnQueue(&Q, p->adjvex);
    } /* for */
} /* while */

free(inDegree);

if ( (5) )
    return NULL;
return topOrder;
} /*TopSort*/

```

### 【问题 1】

根据以上说明和 C 代码，填充 C 代码中的空 (1)～(5)。

- (1) InitQueue(&Q)
- (2) DeQueue(&Q, &w)
- (3) inDegree[p-> adjvex]      或其等价形式
- (4) inDegree[p->adjvex]      或其等价形式
- (5) k<G.n 或 k!=G.n          或其等价形式

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

对 AOE 网进行拓扑排序的方法如下：

- ① 在 AOE 网中选择一个入度为零（没有前驱）的顶点且输出它；
- ② 从网中删除该顶点及其与该顶点有关的所有边；
- ③ 重复上述两步，直至网中不存在入度为零的顶点为止。

在拓扑排序过程中，需要将入度为 0 的顶点临时存储起来。函数中用一个队列暂存入度为 0 且没有进入拓扑序列的顶点。显然，空（1）处应填入 `InitQueue(&Q)`。

进行拓扑排序之前，应先求出网中每个顶点的入度并存入数组 `inDegree[]` 中，从而将“从网中删除该顶点及其与该顶点有关的所有边”的操作转换为“相关顶点的入度减 1”，一旦发现某个顶点的入度变为 0，就将其编号压入堆栈。从而将选择入度为 0 的顶点操作转化为令队头所代表的顶点出队。

根据注释，空（2）处应填入 `DeQueue(&Q,&w)`，实现队头元素出队列的处理。

题中图采用邻接表存储结构，当指针  $p$  指向  $v_i$  邻接表中的结点时， $p \rightarrow \text{adjvex}$  表示  $v_i$  的一个邻接顶点，删除  $v_i$  至顶点  $p \rightarrow \text{adjvex}$  的弧的操作实现为顶点  $p \rightarrow \text{adjvex}$  的入度减 1，因此，空（3）处应填入 `inDegree[p->adjvex]`，当顶点  $p \rightarrow \text{adjvex}$  的入度为 0 时，需要将其加入队列，因此空（4）处也应填入 `inDegree[p->adjvex]`。

空（5）处判断是否所有顶点都加入了拓扑序列，算法中变量  $k$  用于对加入序列的顶点计数，因此，空（5）处应填入“ $k < Gn$ ”或“ $k != Gn$ ”。

## 【问题 2】

对于图 4-1 所示的有向图 G，写出函数 `TopSort` 执行后得到的拓扑序列。若将函数 `TopSort` 中的队列改为栈，写出函数 `TopSort` 执行后得到的拓扑序列。

队列方式：v1v2v5v4v3v7v6 或者 1254376

栈方式：v1v2v5v4v7v3v6 或者 1254736

使用栈和队列的差别在于拓扑序列中顶点的排列次序可能不同。对于本题中的有向图，在使用队列的方式下：

- (1) 开始时仅顶点  $v_1$  的入度为 0，因此顶点  $v_1$  入队；
- (2) 队头顶点  $v_1$  出队，并进入拓扑序列，然后删除从顶点  $v_1$  出发的弧后，仅使顶点  $v_2$  的入度为 0，因此顶点  $v_2$  入队；
- (3) 队头顶点  $v_2$  出队，并进入拓扑序列，然后删除从顶点  $v_2$  出发的弧后，仅使顶点  $v_5$  的入度为 0，因此顶点  $v_5$  入队；
- (4) 队头顶点  $v_5$  出队，并进入拓扑序列，然后删除从顶点  $v_5$  出发的弧后，仅使顶点  $v_4$  的入度为 0，因此顶点  $v_4$  入队；

(5)队头顶点 v4 出队，并进入拓扑序列，然后删除从顶点 v4 出发的弧后，仅使顶点 v3 和 v7 的入度为 0，因此顶点 v3 和 v7 依次入队；

(6)队头顶点 v3 出队，并进入拓扑序列，然后删除从顶点 v3 出发的弧后，没有产生新的入度为 0 的顶点；

(7)队头顶点 v7 出队，并进入拓扑序列，然后删除从顶点 v7 出发的弧后，使顶点 v6 的入度为 0，因此顶点 v6 入队；

(8)队头顶点 v6 出队，并进入拓扑序列，然后删除从顶点 v6 出发的弧后，没有产生新的入度为 0 的顶点，队列已空，因此结束拓扑排序过程，得到的拓扑序列为 v1v2v5v4v3v7v6。

使用栈保存入度为 0 的顶点时，前 4 步都是一样的，因为每次仅有一个元素进栈，因此出栈序列与入栈序列一致。到第 5 步时，v3 和 v7 依次入栈后，出栈时的次序为 v7 和 v3，因此得到的拓扑序列为 v1v2v5v4v7v3 v6。

### 【问题 3】

设某有向无环图的顶点个数为 n、弧数为 e，那么用邻接表存储该图时，实现上述拓扑排序算法的函数 TopSort 的时间复杂度是 (6)。

若有向图采用邻接矩阵表示（例如，图 4-1 所示有向图的邻接矩阵如图 4-3 所示），且将函数 TopSort 中有关邻接表的操作修改为针对邻接矩阵的操作，那么对于有 n 个顶点、e 条弧的有向无环图，实现上述拓扑排序算法的时间复杂度是 (7)。

	v1	v2	v3	v4	v5	v6	v7
v1	0	1	1	1	0	0	0
v2	0	0	0	1	1	0	0
v3	0	0	0	0	0	1	0
v4	0	0	1	0	0	1	1
v5	0	0	0	1	0	0	1
v6	0	0	0	0	0	0	0
v7	0	0	0	0	0	1	0

图 4-3 有向图 G 的邻接矩阵

(6)  $O(n)$

(7)  $O(n^2)$

以邻接表为存储结构时，计算各顶点入度的时间复杂度为  $O(e)$ ，建立零入度顶点队列

的时间复杂度为  $O(n)$ 。在拓扑排序过程中，（图中无环情况下）每个顶点进出队列各 1 次，入度减 1 的操作在 while 循环中共执行  $e$  次，所以总的时间复杂度为  $O(n+e)$ 。

以邻接矩阵为存储结构时，计算各顶点入度时需要遍历整个矩阵，因此时间复杂度为  $O(n^2)$ ，建立零入度顶点队列的时间复杂度为  $O(n)$ 。在拓扑排序过程中，（图中无环情况下）每个顶点进出队列各 1 次，实现入度减 1 操作时需遍历每个顶点的行向量 1 遍（时间复杂度为  $O(n)$ ），所以总的时间复杂度为  $O(n^2)$ 。

试题五

某软件公司现欲开发一款飞机飞行模拟系统，该系统主要模拟不同种类飞机的飞行特征与起飞特征。需要模拟的飞机种类及其特征如表 5-1 所示。

表 5-1

飞 机 种 类	起 飞 特 征	飞 行 特 征
直升机 (Helicopter)	垂直起飞 (VerticalTakeOff)	亚音速飞行 (SubSonicFly)
客机 (AirPlane)	长距离起飞 (LongDistanceTakeOff)	亚音速飞行 (SubSonicFly)
歼击机 (Fighter)	长距离起飞 (LongDistanceTakeOff)	超音速飞行 (SuperSonicFly)
鹞式战斗机 (Harrier)	垂直起飞 (VerticalTakeOff)	超音速飞行 (SuperSonicFly)

为支持将来模拟更多种类的飞机，采用策略设计模式 (Strategy) 设计的类图如图 5-1 所示。 ..

图 5-1 中,AirCraft 为抽象类,描述了抽象的飞机,而类 Helicopter、AirPlane、Fighter 和 Harrier 分别描述具体的飞机种类，方法 fly0 和 takeOff () 分别表示不同飞机都具有飞行特征和起飞特征；类 FlyBehavior 与 TakeOffffleavior 为抽象类，分别用于表示抽象的飞行行为与起飞行为；类 SubSonicFly 与 SuperSonicFly 分别描述亚音速飞行和超音速飞行的行为；类 VerticalTakeOff 与 LongDistanceTakeOff 分别描述垂直起飞与长距离起飞的行为。

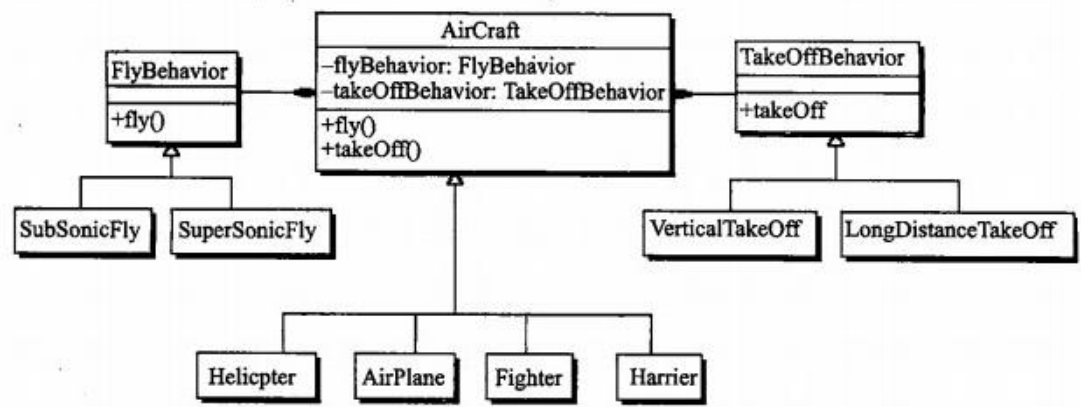


图 5-1 类图

【问题 1】



## 【C++代码】

```
#include<iostream>
using namespace std;
class FlyBehavior {
public: virtual void fly() = 0;
};
class SubSonicFly:public FlyBehavior{
public: void fly(){ cout << "亚音速飞行 !" << endl; }
};
class SuperSonicFly:public FlyBehavior{
public: void fly(){ cout << "超音速飞行 !" << endl; }
};

class TakeOffBehavior {
public: virtual void takeOff() = 0;
};
class VerticalTakeOff:public TakeOffBehavior{
public: void takeOff(){ cout << "垂直起飞 !" << endl; }
};
class LongDistanceTakeOff:public TakeOffBehavior {
public: void takeOff (){ cout << "长距离起飞 !" << endl; }
};

class AirCraft{
protected:
    (1) ;
```

```

    ____ (2) ____;
public:
    void fly(){ ____ (3) ____; }
    void takeOff() { ____ (4) ____; };
};
class Helicopter: public AirCraft {
public:
    Helicopter () {
        flyBehavior = new ____ (5) ____;
        takeOffBehavior = new ____ (6) ____;
    }
    ____ (7) ____ {
        if(!flyBehavior) delete flyBehavior;
        if(!takeOffBehavior) delete takeOffBehavior;
    }
};
//其他代码省略

```

- (1) FlyBehavior \* flyBehavior
- (2) TakeOffBehavior \* takeOffBehavior
- (3) flyBehavior->fly()
- (4) takeOffBehavior->takeOff()\_\_
- (5) SubSonicFly()
- (6) VerticalTakeOff()
- (7) ~Helicopter()

注：空 (1) 与空 (2) 答案可互换

本题目考查了设计模式中的策略设计模式，实际上与 2007 年上半年考核内容相同。

从本题的叙述中可以看出，存在 4 种不同的飞机类型，但每种飞机类型的起飞特征和飞行特征并不完全相同，这就使得我们很难采用比较直接的方法来实现重用。例如，定义一个抽象的飞机类，实现飞机的起飞特征，然后 4 种飞机直接重用该特征。但是我们可以观察到，尽管飞机的起飞特征和飞行特征有所不同，有一点可以肯定的是，每一种飞机都具备了飞行特征和起飞特征。因此，可以抽象出一个飞机类，其中含有飞行特征与起飞特征，但关于两个特征的实现要单独抽取出来，所以又形成了 FlyBehavior 类和 TakeOffBehavior 类分别表示抽象的飞行和起飞特征，而这两个类的子类则分别实现不同的起飞和飞行特征，最终转化为，在创建一个具体的飞机时，给其配上不同的起飞 特征和飞行特征即可。

本题中的空(1)和空(2)应该填写成员变量，根据类图可以得知，此处应该表示的是飞行和起飞特征变量，在 C++中可以采用指针来表示。空(3)和空(4)处需要实现飞行与起飞特征，但 Aircraft 是抽象的类，所以把实现代理给指针变量。Helicopter 类需要指定由父类继承而来的成员变量的初始值，因为 Helicopter 的特征是垂直起飞和亚音速飞行，因此生成这两个特征的对象，分别赋值给 flyBehavior 和 takeOffBehavior 变量。



试题六

某软件公司现欲开发一款飞机飞行模拟系统，该系统主要模拟不同种类飞机的飞行特征与起飞特征。需要模拟的飞机种类及其特征如表 6-1 所示。

表 6-1		
飞 机 种 类	起 飞 特 征	飞 行 特 征
直升机（Helicopter）	垂直起飞（VerticalTakeOff）	亚音速飞行（SubSonicFly）
客机（AirPlane）	长距离起飞（LongDistanceTakeOff）	亚音速飞行（SubSonicFly）
歼击机（Fighter）	长距离起飞（LongDistanceTakeOff）	超音速飞行（SuperSonicFly）
鹞式战斗机（Harrier）	垂直起飞（VerticalTakeOff）	超音速飞行（SuperSonicFly）

为支持将来模拟更多种类的飞机，采用策略设计模式（Strategy）设计的类图如图 6-1 所示。

图 6-1 中,AirCraft 为抽象类,描述了抽象的飞机,而类 Helicopter、AirPlane、Fighter 和 Harrier 分别描述具体的飞机种类，方法 fly() 和 takeOff() 分别表示不同飞机都具有飞行特征和起飞特征；类 FlyBehavior 与 TakeOffBehavior 为抽象类，分别用于表示抽象的飞行行为与起飞行为；类 SubSonicFly 与 SuperSonicFly 分别描述亚音速飞行和超音速飞行 的行为；类 VerticalTakeOff 与 LongDistanceTakeOff 分别描述垂直起飞与长距离起飞的 行为。

【Java 代码】

```
interface FlyBehavior {
    public void fly();
};

class SubSonicFly implements FlyBehavior{
    public void fly(){ System.out.println("亚音速飞行 !"); }
};

class SuperSonicFly implements FlyBehavior{
    public void fly(){ System.out.println("超音速飞行 ! " ); }
};
```

【问题 1】

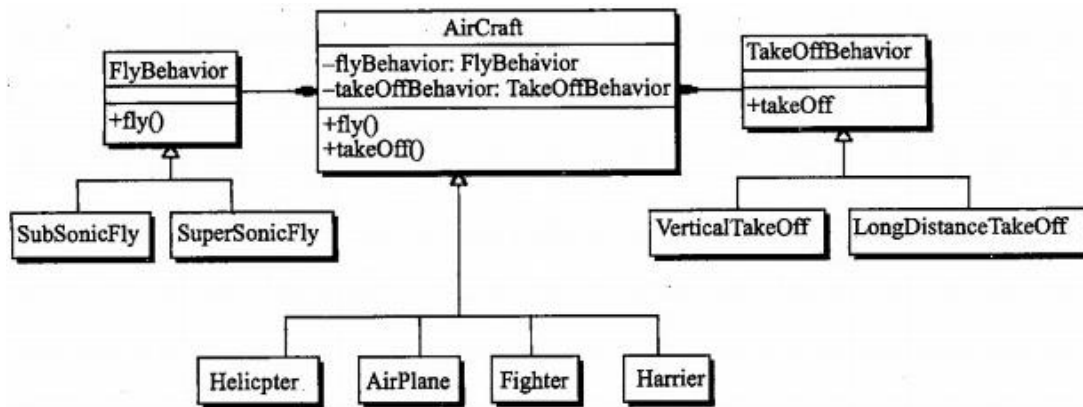


图 6-1 类图

```

interface TakeOffBehavior {
    public void takeOff();
};

class VerticalTakeOff implements TakeOffBehavior {
    public void takeOff () { System.out.println("垂直起飞！"); }
};

class LongDistanceTakeOff implements TakeOffBehavior {
    public void takeOff() { System.out.println("长距离起飞！"); }
};

abstract class AirCraft {
    protected (1);
    protected (2);
    public void fly(){ (3); }
    public void takeOff() { (4); };
};

class Helicopter (5) AirCraft{
    public Helicopter () {
        flyBehavior = new (6);
        takeOffBehavior = new (7);
    }
};

//其他代码省略

```

考在线  
W.RKPASS.CN

- (1) FlyBehavior flyBehavior
- (2) TakeOffBehavior takeOffBehavior
- (3) flyBehavior.fly()
- (4) takeOffBehavior.takeOff()
- (5) extends

(6) SubSonicFly()

(7) VerticalTakeOff()

本题目考查了设计模式中的策略设计模式,实际上与 2007 年上半年 Java 题目的考核内容相同。

从本题的叙述中可以看出,存在四种不同的飞机类型,但每种飞机类型的起飞特征和飞行特征并不完全相同,这就使得我们很难采用比较直接的方法来实现重用。例如'定义一个抽象的飞机类,实现飞机的起飞特征,然后四种飞机直接重用该特征。但是,我们可以观察到,尽管飞机的起飞特征和飞行特征有所不同,有一点可以肯定的是,每一种飞机都具备了飞行特征和起飞特征。因此,可以抽象出一个飞机类,其中含有飞行特征与起飞特征,但关于两个特征的实现要单独抽取出来,所以又形成了 FlyBehavior 类和 TakeOffBehavior 类,分别表示抽象的飞行和起飞特征,而这两个类的子类则分别实现不同的起飞和飞行特征,最终转化为,在创建一个具体的飞机时,给其配上不同的起飞特征和飞行特征即可。

本题中的空(1)和空(2)应该填写成员变量,根据类图可以得知,此处应该表示的是飞行和起飞特征变量。空(3)和空(4)处需要实现飞行与起飞特征,但 Aircraft 是抽象的类,所以把实现代理给指针变量。Helicopter 类需要指定由父类继承而来的成员变量的初始值,因为 Helicopter 的特征是垂直起飞和亚音速飞行,因此生成这两个特征的对象,分别赋值给 flyBehavior 和 takeOffBehavior 变量。