

2015 年 5 月第 1 题

机器字长为 n 位的二进制数可以用补码来表示 (1) 个不同的有符号定点小数。

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

【答案】A

【解析】本题考查计算机系统的基础知识。

二进制数据在计算机系统表示方法是最基本的专业知识。补码本身是带符号位的，补码表示的数字中 0 是唯一的，不像原码有 +0 和 -0 之分，也就意味着 n 位进制编码可以表示 $2n$ 个不同的数。

2015 年 5 月第 2 题

计算机中 CPU 对其访问速度最快的是 (2)。

- (2) A. 内存 B. Cache C. 通用寄存器 D. 硬盘

【答案】C

【解析】本题考查计算机系统的基础知识。

计算机系统 CPU 内部对通用寄存器的存取操作是速度最快的，其次是 Cache，内存的存取速度再次，选项中访问速度最慢的就是作为外存的硬盘。它们共同组成分级存储体系来解决存储容量、成本和速度之间的矛盾。

2015 年 5 月第 3 题

Cache 的地址映像方式中，发生块冲突次数最小的是 (3)。

- (3) A. 全相联映像 B. 组相联映像 C. 直接映像 D. 无法确定

【答案】A

【解析】

Cache 工作时，需要拷贝主存信息到 Cache 中，就需要建立主存地址和 Cache 地址的映射关系。Cache 的地址映射方法主要有三种，即全相联映像、直接映像和组相联映像。其中全相联方式意味着主存的任意一块可以映像到 Cache 中的任意一块，其特点是块冲突概率低，Cache 空间利用率高，但是相联目录表容量大导致成本高、查表速度慢；直接映像方式是指主存的每一块只能映像到 Cache 的一个特定的块中，整个 Cache 地址与主存地址的低位部分完全相同，其特点是硬件简单，不需要相联存储器，访问速度快（无须地址变换），但是 Cache 块冲突概率高导致 Cache 空间利用率很低；组相联方式是对上述两种方式的折中处理，对

Cache 分组，实现组间直接映射，组内全相联，从而获得较低的块冲突概率、较高的块利用率，同时得到较快的速度和较低的成本。

2015 年 5 月第 4 题

计算机中 CPU 的中断响应时间指的是(4)的时间。

- (4)A.从发出中的请求到中断处理结束 B.从中断处理开始到中断处理结束
C.CPU 分析判断中断请求 D.从发出中断请求到开始进入中断处理程序

【答案】D

【解析】本题考查计算机组成的基础知识。

中断系统是计算机实现中断功能的软硬件总称。一般在 CPU 中设置中断机构，在外设接口中设置中断控制器，在软件上设置相应的中断服务程序。中断源在需要得到 CPU 服务时，请求 CPU 暂停现行工作转向为中断源服务，服务完成后，再让 CPU 回到原工作状态继续完成被打断的工作。中断的发生起始于中断源发出中断请求，中断处理过程中，中断系统需要解决一系列问题，包括中断响应的条件和时机，断点信息的保护与恢复，中断服务程序入口、中断处理等。中断响应时间，是指从发出中断请求到开始进入中断服务程序所需的时间。

2015 年 5 月第 5 题

总线宽度为 32bit，时钟频率为 200MHz，若总线上每 5 个时钟周期传送一个 32bit 的字，则该总线的带宽为(5)MB/s。

- (5)A.40 B.80 C.160 D.200

【答案】C

【解析】本题考查计算机系统的基础知识。

总线宽度是指总线的线数，即数据信号的并行传输能力，也体现总线占用的物理空间和成本；总线的带宽是指总线的最大数据传输率，即每秒传输的数据总量。总线宽度与时钟频率共同决定了总线的带宽。

$$32\text{bit} / 8 = 4 \text{ Byte},$$

$$200\text{MHz} / 5 \times 4 \text{ Byte} = 160 \text{ MB/s}$$

2015 年 5 月第 6 题

以下关于指令流水线性能度量的叙述中，错误的是(6)。

- (6)A.最大吞吐率取决于流水线中最慢一段所需的时间
- B.如果流水线出现断流，加速比会明显下降
- C.要使加速比和效率最大化应该对流水线各级采用相同的运行时间
- D.流水线采用异步控制会明显提高其性能

【答案】D

【解析】本题考查计算机系统结构的基础知识。

对指令流水线性能的度量主要有吞吐率，加速比和效率等指标。吞吐率是指单位时间内流水线所完成的任务数或输出结果的数量，最大吞吐率则是流水线在达到稳定状态后所得到的吞吐率，它取决于流水线中最慢一段所需的时间，所以该段成为流水线的瓶颈。流水线的加速比定义为等功能的非流水线执行时间与流水线执行时间之比，加速比与吞吐率成正比，如果流水线断流，实际吞吐率将会明显下降，则加速比也会明显下降。流水线的效率是指流水线的设备利用率，从时空图上看效率就是 n 个任务所占的时空区与 m 个段总的时空区之比。因此要使加速比和效率最大化应该流水线各级采用相同的运行时间。另外，流水线采用异步控制并不会给流水线性能带来改善，反而会增加控制电路的复杂性。

2015 年 5 月第 7 题

____(7)____ 协议在终端设备与远程站点之间建立安全连接。

- (7)A.ARP B.Telnet C.SSH D.WEP

【答案】C

【解析】

终端设备与远程站点之间建立安全连接的协议是 SSH。SSH 为 Secure Shell 的缩写，是由 IETF 制定的建立在应用层和传输层基础上的安全协议。SSH 是专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH 最初是 UNIX 上的程序，后来又迅速扩展到其他操作平台。

2015 年 5 月第 8 题

安全需求可划分为物理线路安全、网络安全、系统安全和应用安全。下面的安全需求中属于系统安全的是____(8)____，属于应用安全的是____(9)____。

- (8)A.机房安全 B.入侵检测 C.漏洞补丁管理 D.数据库安全
- (9)A.机房安全 B.入侵检测 C.漏洞补丁管理 D.数据库安全

【答案】C D

【解析】

机房安全属于物理安全，入侵检测属于网络安全，漏洞补丁管理属于系统安全，而数据库安全则是应用安全。

2015 年 5 月第 10 题

王某是某公司的软件设计师，每当软件开发完成后均按公司规定编写软件文档，并提交公司存档。那么该软件文档的著作权(10)享有。

- (10)A.应由公司 B.应由公司和王某共同
C.应由王某 D.除署名权以外，著作权的其他权利由王某

【答案】A

【解析】本题考查知识产权的基本知识。

依据著作权法第十一条、第十六条规定，职工为完成所在单位的工作任务而创作的作品属于职务作品。职务作品的著作权归属分为两种情况。

①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，如教师编写的教材，著作权应由作者享有，但法人或者其他组织在其业务范围内有优先使用的权利，期限为 2 年。

②由法人或者其他组织主持，代表法人或者其他组织意志创作，并由法人或者其他组织承担责任的职务作品，如工程设计、产品设计图纸及其说明、计算机软件、地图等职务作品，以及法律规定或合同约定著作权由法人或非法人单位单独享有的职务作品，作者享有署名权，其他权利由法人或者其他组织享有。

2015 年 5 月第 11 题

甲、乙两公司的软件设计师分别完成了相同的计算机程序发明，甲公司先于乙公司完成，乙公司先于甲公司使用。甲、乙公司于同一天向专利局申请发明专利。此情形下，(11)可获得专利权。

- (11)A.甲公司 B.甲、乙公司均 C.乙公司 D.由甲、乙公司协商确定谁

【答案】D

【解析】本题考查知识产权的基本知识。

当两个以上的申请人分别就同样的发明创造申请专利的，专利权授给最先申请的人。如

果两个以上申请人在同一天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,即均不授予专利权。我国专利法规定“两个以上的申请人分别就同样的发明创造申请专利的,专利权授予最先申请的人。”我国专利法实施细则规定:“同样的发明创造只能被授予一项专利。依照专利法第九条的规定,两个以上的申请人在同一日分别就同样的发明创造申请专利的,应当在收到国务院专利行政部门的通知后自行协商确定申请人。”

2015 年 5 月第 12 题

以下媒体中,(12)是感觉媒体。

- (12)A.音箱 B.声音编码 C.电缆 D.声音

【答案】D

【解析】本题考查多媒体的基本知识。

感觉媒体指直接作用于人的感觉器官,使人产生直接感觉的媒体,如引起听觉反应的声音,引起视觉反应的图像等。

2015 年 5 月第 13 题

微型计算机系统中,显示器属于(13)。

- (13)A.表现媒体 B.传输媒体 C.表示媒体 D.存储媒体

【答案】A

【解析】本题考查多媒体的基本知识。

表现媒体是指进行信息输入和输出的媒体,如键盘、鼠标、话筒,以及显示器、打印机、机等;表示媒体指传输感觉媒体的中介媒体,即用于数据交换的编码,如图像编码、文本编码和声音编码等;传输媒体指传输表示媒体的物理介质,如电缆、光缆、电磁波等;存储媒体指用于存储表示媒体的物理介质,如硬盘、光盘等。

2015 年 5 月第 14 题

(14)是表示显示器在纵向(列)上具有的像素点数目指标。

- (14)A.显示分辨率 B.水平分辨率 C.垂直分辨率 D.显示深度

【答案】C

【解析】本题考查多媒体的基本知识。

显示分辨率是指显示器上能够显示出的像素点数目,即显示器在横向和纵向上能够显示

出的像素点数目。水平分辨率表明显示器水平方向（横向）上显示出的像素点数目，垂直分辨率表明显示器垂直方向（纵向）上显示出的像素点数目。例如，显示分辨率为 1024X768 则表明显示器水平方向上显示 1024 个像素点，垂直方向上显示 768 个像素点，整个显示屏就含有 796432 个像素点。屏幕能够显示的像素越多，说明显示设备的分辨率越高，显示的图像质量越高。显示深度是指显示器上显示每个像素点颜色的二进制位数。

2015 年 5 月第 15 题

软件工程的基本要素包括方法、工具和（15）。

- (15)A.软件系统 B.硬件系统 C.过程 D.人员

【答案】C

【解析】本题考查软件工程的基本概念。

软件工程是一种层次化的技术，从底向上分别为质量、过程、方法和工具。任何工程方法必须以有组织的质量承诺为基础。软件工程的基础是过程，过程是将技术结合在一起的凝聚力，使得计算机软件能够被合理地及时地开发，过程定义了一组关键过程区域，构成了软件项目管理控制的基础；方法提供了建造软件在技术上需要“如何做”，它覆盖了一系列的任务。方法也依赖于一些基本原则，这些原则控制了每一个技术区域而且包含建模活动和其他描述技术；工具对过程和方法提供了自动或半自动的支持，如：计算机辅助软件工程（CASE）。软件工程的基本要素包括方法、工具和过程。

2015 年 5 月第 16 题

在（16）设计阶段选择适当的解决方案，将系统分解为若干个子系统，建立整个系统的体系结构。

- (16)A.概要 B.详细 C.结构化 D.面向对象

【答案】A

【解析】本题考查软件工程的基本概念。

软件设计的任务是基于需求分析的结果建立各种设计模型，给出问题的解决方案。从工程管理的角度，可以将软件设计分为两个阶段：概要设计阶段和详细设计阶段。结构化设计方法中，概要设计阶段进行软件体系结构的设计、数据设计和接口设计；详细设计阶段进行数据结构和算法的设计。面向对象设计方法中，概要设计阶段进行体系结构设计、初步的类设计/数据设计、结构设计；详细设计阶段进行构件设计。

结构化设计和面向对象设计是两种不同的设计方法，结构化设计根据系统的数据流图进

行设计，模块体现为函数、过程及子程序；面向对象设计基于面向对象的基本概念进行，模块体现为类、对象和构件等。

2015 年 5 月第 17 题

某项目包含的活动如下表所示，完成整个项目的最短时间为 (17) 周。不能通过缩短活动 (18) 的工期，来缩短整个项目的完成时间。

活动编号	工期（周）	直接前驱
A	3	-
B	5	A
C	1	B
D	3	A
E	5	D
F	4	C,E
G	3	C,E
H	4	F,G

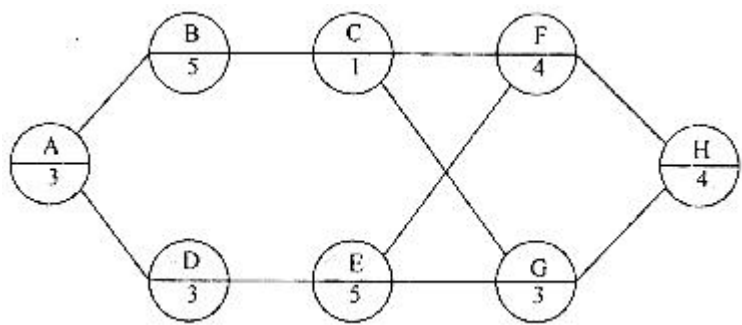
- (17)A.16 B.17 C.18 D.19
- (18)A.A B.B C.D D.F

【答案】D B

【解析】

本题考查软件项目管理的基础知识。

活动图是描述一个项目中各个工作任务相互依赖关系的一种模型，项目的很多重要特性可以通过分析活动图得到，如估算项目完成时间，计算关键路径和关键活动等。根据上表给出的数据，构建活动图，如下图所示。



从上图很容易看出，关键路径为 A-D-E-F-H,其长度为 19,关键路径上的活动均为关键活动。

2015 年 5 月第 19 题

风险的优先级通常是根据 (19) 设定。

- (19)A. 风险影响 (Risk Impact) B. 风险概率(Risk Probability)
C. 风险暴露(Risk Exposure) D. 风险控制(Risk Control)

【答案】C

【解析】本题考查软件项目管理的基础知识。

风险是一种具有负面后果的、人们不希望发生的事件。风险管理是软件项目管理的一项重要任务。在进行风险管理时，根据风险的优先级来确定风险控制策略，而优先级是根据风险暴露来确定的。风险暴露是一种量化风险影响的指标，等于风险影响乘以风险概率，风险影响是当风险发生时造成的损失。风险概率是风险发生的可能性。风险控制是风险管理的一个重要活动。

2015 年 5 月第 20 题

以下关于程序设计语言的叙述中，错误的是 (20)。

- (20)A. 程序设计语言的基本成分包括数据、运算、控制和传输等
B. 高级程序设计语言不依赖于具体的机器硬件
C. 程序中局部变量的值在运行时不能改变
D. 程序中常量的值在运行时不能改变

【答案】C

【解析】本题考查程序语言基础知识。

选项 A 涉及程序语言的一般概念，程序设计语言的基本成分包括数据、运算、控制和传输等。

选项 B 考查高级语言和低级语言的概念。对于程序设计语言高级语言和低级语言是指其相对于运行程序的机器的抽象程度。低级语言在形式上更接近机器指令，汇编语言就是与机器指令一一对应的。高级语言对底层操作进行了抽象和封装，其一条语句对应多条机器指令，使编写程序的过程更符合人类的思维习惯，并且极大简化了人力劳动。高级语言不依赖于具体的机器硬件。

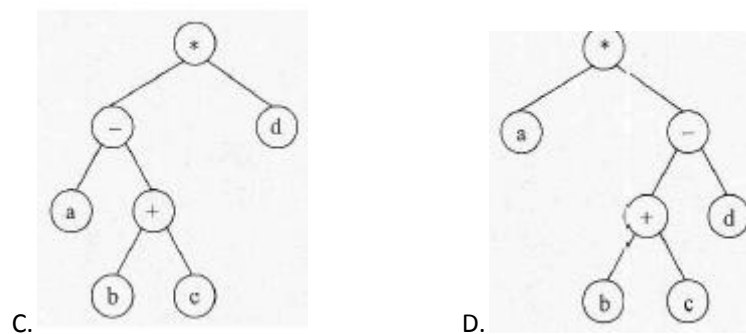
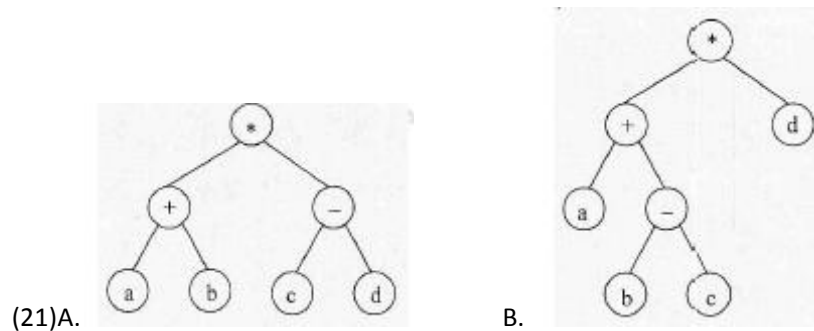
选项 C 考查局部变量的概念，凡是在函数内部定义的变量都是局部变量（也称作内部变量），包括在函数内部复合语句中定义的变量和函数形参表中说明的形式参数。局部变量只能在函数内部使用，其作用域是从定义位置起至函数体或复合语句体结束为止。局部变量的值通

常在其生存期内是变化的。

选项 D 考查常量的概念，程序中常量的值在运行时是不能改变的。

2015 年 5 月第 21 题

与算术表达式 “ $(a + (b - c)) * d$ ” 对应的树是 (21)。



【答案】B

【解析】本题考查程序语言与数据结构基础知识。

对算术表达式 “ $(a + (b - c)) * d$ ” 求值的运算处理顺序是：先进行 $b - c$ ，然后与 a 相加，最后再与 d 相乘。只有选项 B 所示的二叉树与其相符。

2015 年 5 月第 22 题

C 程序中全局变量的存储空间在 (22) 分配。

(22)A. 代码区 B. 静态数据区 C. 栈区 D. 堆区

【答案】B

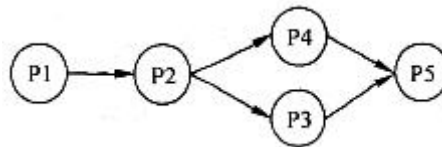
【解析】本题考查程序语言基础知识。

程序运行时的用户内存空间一般划分为代码区、静态数据区、栈区和堆区，其中栈区和

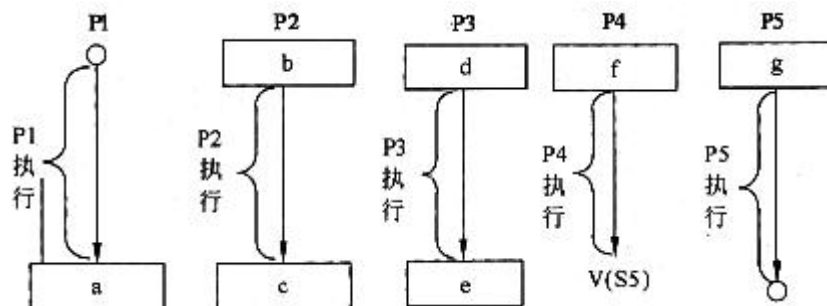
堆区也称为动态数据区。全局变量的存储空间在静态数据区。

2015 年 5 月第 23 题

进程 P1、P2、P3、P4 和 P5 的前趋图如下所示：



若用 PV 操作控制进程 P1、P2、P3、P4、P5 并发执行的过程，则需要设置 5 个信号量 S1、S2、S3、S4 和 S5，且信号量 S1~S5 的初值都等于零。下图中 a、b 和 c 处应分别填写 (23)；d 和 e 处应分别填写 (24)，f 和 g 处应分别填写 (25)。



(23) A. V(S1)、P(S1) 和 V(S2)V(S3)

B. P(S1)、V(S1) 和 V(S2)V(S3)

C. V(S1)、V(S2) 和 P(S1)V(S3)

D. P(S1)、V(S2) 和 V(S1)V(S3)

(24) A. V(S2) 和 P(S4)

B. P(S2) 和 V(S4)

C. P(S2) 和 P(S4)

D. V(S2) 和 V(S4)

(25) A. P(S3) 和 V(S4)V(S5)

B. V(S3) 和 P(S4)P(S5)

C. P(S3) 和 P(S4)P(S5)

D. V(S3) 和 V(S4)V(S5)

【答案】A B C

【解析】

试题 (23) 的正确的答案为 A。根据前驱图，P1 进程执行完需要通知 P2 进程，故需要利用 V(S1) 操作通知 P2 进程，所以空 a 应填 V(S1)；P2 进程需要等待 P1 进程的结果，故需要利用 P(S1) 操作测试 P1 进程是否运行完，所以空 b 应填 P(S1)；又由于 P2 进程运行结束需要利用 V(S2)、V(S3) 操作分别通知 P3、P4 进程，所以空 c 应填 V(S2)、V(S3)。

试题 (24) 的正确的答案为 B。根据前驱图，P3 进程运行前需要等待 P2 进程的结果，故需执行程序前要先利用 1 个 P 操作，根据排除法可选项只有选项 B 和选项 C。又因为 P3 进程运行结束后需要利用 1 个 V 操作通知 P5 进程，根据排除法可选项只有选项 B 满足要求。

试题 (25) 的正确的答案为 C。根据前驱图, P4 进程执行前需要等待 P2 进程的结果, 故空 f 处需要 1 个 P 操作; P5 进程执行前需要等待 P3 和 P4 进程的结果, 故空 g 处需要 2 个 P 操作。根据排除法可选项只有选项 C 能满足要求。

2015 年 5 月第 26 题

某进程有 4 个页面, 页号为 0~3, 页面变换表及状态位、访问位和修改位的含义如下图所示。若系统给该进程分配了 3 个存储块, 当访问前页面 1 不在内存时, 淘汰表中页号为(26)的页面代价最小。

页号	页帧号	状态位	访问位	修改位
0	6	1	1	1
1	—	0	0	0
2	3	1	1	1
3	2	1	1	0

状态位含义 {
=0 不在内存
=1 在内存

访问位含义 {
=0 未访问过
=1 访问过

修改位含义 {
=0 未修改过
=1 修改过

- (26) A.0 B.1 C.2 D.3

【答案】D

【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 3 号页面在内存。当访问的页面 1 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, 故淘汰时无须写回辅存, 使系统页面置换代价小。经上述分析, 0、2 和 3 号页面都是最近被访问过的, 但 0 和 2 号页面都被修改过而 3 号页面未修改过, 故应该淘汰 3 号页面。

2015 年 5 月第 27 题

嵌入式系统初始化过程主要有 3 个环节, 按照自底向上、从硬件到软件的次序依次为 (27)。系统级初始化主要任务是 (28)。

- (27) A. 片级初始化 → 系统级初始化 → 板级初始化
 B. 片级初始化 → 板级初始化 → 系统级初始化
 C. 系统级初始化 → 板级初始化 → 片级初始化

D.系统级初始化→片级初始化→板级初始化

(28)A.完成嵌入式微处理器的初始化

B.完成嵌入式微处理器以外的其他硬件设备的初始化

C.以软件初始化为主，主要进行操作系统的初始化

D.设置嵌入式微处理器的核心寄存器和控制寄存器工作状态

【答案】B C

【解析】本题考查应试者对嵌入式系统方面的基础知识。

嵌入式系统初始化过程可以分为3个主要环节，按照自底向上、从硬件到软件的次序依次为：片级初始化、板级初始化和系统级初始化。

片级初始化完成嵌入式微处理器的初始化，包括设置嵌入式微处理器的核心寄存器和控制寄存器、嵌入式微处理器核心工作模式和嵌入式微处理器的局部总线模式等。片级初始化把嵌入式微处理器从上电时的默认状态逐步设置成系统所要求的工作状态。这是一个纯硬件的初始化过程。

板级初始化完成嵌入式微处理器以外的其他硬件设备的初始化。另外，还需设置某些软件的数据结构和参数，为随后的系统级初始化和应用程序的运行建立硬件和软件环境。这是一个同时包含软硬件两部分在内的初始化过程。

系统初始化过程以软件初始化为主，主要进行操作系统的初始化。BSP将对嵌入式微处理器的控制权转交给嵌入式操作系统，由操作系统完成余下的初始化操作，包含加载和初始化与硬件无关的设备驱动程序，建立系统内存区，加载并初始化其他系统软件模块，如网络系统、文件系统等。最后，操作系统创建应用程序环境，并将控制权交给应用程序的入口。

2015年5月第29题

某公司计划开发一种产品，技术含量很高，与客户相关的风险也很多，则最适于采用(29)开发过程模型。

(29)A.瀑布

B.原型

C.增量

D.螺旋

【答案】D

【解析】本题考查软件过程模型的基础知识。

瀑布模型将软件生存周期各个活动规定为线性顺序连接的若干阶段的模型，规定了由前至后，相互衔接的固定次序，如同瀑布流水，逐级下落。这种方法是一种理想的开发模式，缺乏灵活性，特别是无法解决软件需求不明确或不准确的问题。

原型模型从初始的原型逐步演化成最终软件产品，特别适用于对软件需求缺乏准确认识的情况。

增量开发是把软件产品作为一系列的增量构件来设计、编码、集成和测试，可以在增量开发过程中逐步理解需求。

螺旋将瀑布模型与快速原型模型结合起来，并且加入两种模型均忽略了风险分析，适用于复杂的大型软件。

2015 年 5 月第 30 题

在敏捷过程的方法中 (30) 认为每一个不同的项目都需要一套不同的策略、约定和方法论。

(30)A.极限编程 (XP) B.水晶法(Crystal) C.并列争球法(Scrum) D.自适应软件开发(ASD)

【答案】B

【解析】本题考查敏捷方法的基础知识。

在 20 世纪 90 年代后期，一些开发人员抵制产格化软件开发过程，试图强调灵活性在快速有效的软件生产中的作用，提出了敏捷宣言，即个人和交互胜过过程和工具；可 以运行的软件胜过面面俱到的文档；与客户合作胜过合同谈判；对变化的反应胜过遵循计划。基于这些基本思想，有很多敏捷过程的典型方法，其中：

极限编程 XP 是激发开发人员创造性、使得管理负担最小的一组技术。

水晶法 Crystal 认为每一个不同的项目都需要一套不同的策略、约定和方法论。

并列争球法(Scrum)使用迭代的方法，其中把每 30 天一次的迭代称为个冲刺， 并按需求的优先级来实现产品多个自组织和自治小组并行地递增实现产品，协调是通过简短的日常情况会议进行。

自适应软件开发 (ASD)有六个基本的原则：

- ①在自适应软件开发中，有一个使命作为指导，它设立了项目的目标，但不描述如何达到这个目标；
- ②特征被视为客户键值的关键，因此，项目是围绕着构造的构件来组织并实现特征；
- ③过程中的迭代是很重要的，因此重做与做同样重要，变化也包含其中；
- ④变化不视为是一种更正，而是对软件开发实际情况的调整；
- ⑤确定的交付时间迫使开发人员认真考虑每一个生产版本的关键需求；
- ⑥风险也包含其中，它使开发人员首先跟踪最艰难的问题。

2015 年 5 月第 31 题

软件配置管理的内容不包括 (31)。

- (31)A.版本控制 B.变更控制 C.过程支持 D.质量控制

【答案】D

【解析】本题考查软件配置管理的基础知识。

软件配置管理 SCM 用于整个软件工程过程，其主要目标是标识变更、控制变更、确保变更正确的实现，报告变更。其主要内容包括版本管理、配置支持、变更支持、过程支持、团队支持、变化报告和审计支持等。

2015 年 5 月第 32 题

某模块实现两个功能：向某个数据结构区域写数据和从该区域读数据。该模块的内聚类型为 (32) 内聚。

- (32)A.过程 B.时间 C.逻辑 D.通信

【答案】D

【解析】本题考查软件设计的基础知识。

模块独立性是创建良好设计的一个重要原则，一般采用模块间的耦合和模块的内聚两个准则来进行度量。耦合程度越低，内聚程度越高，则模块的独立性越好。

存在多种模块内聚类型，从高到低依次为功能、通信、顺序、过程、时间、逻辑和偶然内聚。

一个模块内的几个操作是相关的，而且必须以特定的次序执行，则该模块的内聚类型为过程内聚。

模块的各个功能的执行与时间有关，通常要求所有功能必须在同一时间段内执行，则该模块的内聚类型为时间内聚。

若一个模块的各个部分只是通过代码的逻辑结构相关联，则该模块的内聚类型为逻辑内聚。

把访问或操作在同一数据结构的操作放在一个模块中，则该模块的内聚类型为通信内聚。

2015 年 5 月第 33 题

正式技术评审的目标是 (33)。

- (33)A.允许高级技术人员修改错误 B.评价程序员的工作效率

- D.记录程序员的错误情况并与绩效挂钩

【解析】 本题考查软件评审的基础知识。

软件评审的内容包括设计质量评审、程序质量评审和与运行环境接口的评审。评审的主要目标是为了发现软件中的错误。

自底向上的集成测试策略的优点包括 (34)。

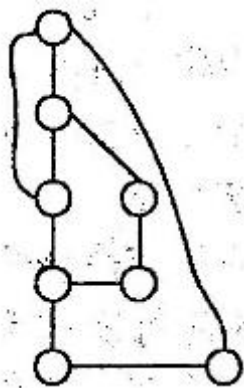
- B.不需要写驱动程序

- D.不需要进行回归测试

【解析】 本题考查软件测试的基础知识。

软件测试分为单元测试、集成测试和系统测试。集成测试把模块按系统设计说明书的要求组合起来进行测试。存在多种组合的集成测试策略：自底向上、自顶向下、一次性集成、明治集成等。其中，自底向上策略通过不断合并底层模块来测试更高层模块的方式进行。在该方法中，需要写驱动程序来调用待测试的底层模块，主要的设计问题需要到测试后期才能发现。由于从最底层的模块开始测试，因此过程中不需要写桩模块。当测试过程中发现错误时，需要进行回归测试。

采用 McCabe 度量法计算下列程序图的环路复杂性为 (35)。



- D.5

【答案】C

【解析】 本题考查软件度量的基础知识。

软件复杂性主要表现在程序的复杂性，而程序的复杂性主要指模块内程序的复杂性。McCabe 度量法是一种度量程序复杂性的方法，该方法用程序图来表示实际的程序，如上图所示，用环路数来表示程序复杂性，即 $V(G) = m - n + 2p$ ，其中 m 和 n 分别表示图中的边数和顶点数， p 为 1。本题中， $m=10, n=8$ ，因此 $V(G) = 10 - 8 + 2 = 4$ 。

2015 年 5 月第 36 题

以下关于软件可维护性的叙述中，不正确的是“可维护性 (36)”。

- (36)A. 是衡量软件质量的一个重要特性
- B. 不受软件开发文档的影响
- C. 是软件开发阶段各个时期的关键目标
- D. 可以从可理解性、可靠性、可测试性、可行性、可移植性等方面进行度量

【答案】 B

【解析】 本题考查维护方面的基础知识。

软件交付给用户使用后到软件报废之前都属于软件维护阶段。软件系统的可维护性可以定义为：维护人员理解、改正、改动和改进该软件的难易程度。提供软件可维护性是开发软件系统所有步骤的关键目的，是衡量软件质量的一种重要特性，可以从可理解性、可靠性、可测试性、可行性、可移植性等方面进行度量。良好的软件开发文档可以有效地提高软件的可维护性。

2015 年 5 月第 37 题

对象、类、继承和消息传递是面向对象的 4 个核心概念。其中对象是封装 (37) 的整体。

- (37)A. 命名空间 B. 要完成任务 C. 一组数据 D. 数据和行为

【答案】 D

【解析】 本题考查面向对象的基本知识。

面向对象的 4 个核心概念是对象、类、继承和消息传递。其中，对象是基本的运行时的实体，它既包括数据（属性），也包括作用于数据的操作（行为）。所以，一个对象把属性和行为封装为一个整体。类定义了一组大体上相似的对象。一个类所包含的方法和数据描述一组对象的共同行为和属性。在进行类设计时，有些类之间存在一般和特殊关系，即一些类是某个类的特殊情况，某个类是一些类的一般情况，这就是继承关系。消息是对象之间进行

通信的一种构造，包含要求接收对象去执行某些活动的信息。

2015 年 5 月第 38 题

面向对象(38)选择合适的面向对象程序设计语言，将程序组织为相互协作的对象集合，每个对象表示某个类的实例，类通过继承等关系进行组织。

(38)A.分析 B.设计 C.程序设计 D.测试

【答案】C

【解析】本题考查面向对象的基本知识。

在采用面向对象技术开发系统时，主要步骤有面向对象分析、面向对象设计、面向对象程序设计和面向对象测试。面向对象分析主要包括：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。面向对象设计是设计分析模型和实现相应源代码。面向对象程序设计选择合适的面向对象程序设计语言，将程序组织为相互协作的对象集合，每个对象表示某个类的实例，类通过继承等关系进行组织。面向对象测试是尽可能早的开始进行系统测试，以发现系统中可能存在的错误并进行修复，进而保证系统质量。

2015 年 5 月第 39 题

一个类可以具有多个同名而参数类型列表不同的方法，被称为方法(39)。

(39)A.重载 B.调用 C.重置 D.标记

【答案】A

【解析】本题考查面向对象的基本知识。

在面向对象技术中，一个类可以具有多个同名而参数类型列表不同的方法，每个方法实现自己的特定行为，这种机制称为方法重载（Overloading）。对象协作过程中，对象给对象进行消息传递，通常通过方法调用进行。在继承关系中，子类重新定义父类中已经定义的方法，称为方法重置或覆盖（Overriding）。

2015 年 5 月第 40 题

UML 中有 4 种关系：依赖、关联、泛化和实现。(40)是一种结构关系，描述了一组链，链是对象之间的连接；(41)是一种特殊 / 一般关系，使子元素共享其父元素的结构和行为。

(40)A.依赖 B.关联 C.泛化 D.实现

(41)A.依赖 B.关联 C.泛化 D.实现

【答案】B C

【解析】本题考查统一建模语言（UML）的基本知识。

UML 是面向对象软件的标准化建模语言，由三个要素构成：UML 的基本构造块、支配这些构造块如何放置在一起的规则和运用与整个语言的一些公共机制。UML 的词汇表包含三种构造块：事物、关系和图。事物是对模型中最具有代表性的成分的抽象；关系把事物结合在一起；图聚集了相关的事物。其中关系包括 4 种：依赖、关联、泛化和实现。

依赖（Dependency）是两个事物间的语义关系，其中一个事物（独立事物）发生变化会影响另一个事物（依赖事物）的语义。关联（Association）是一种结构关系，它描述了一组链，链是对象之间的连接。泛化（Generalization）是一种特殊/一般关系，特殊元素（子元素）的对象可替代一般元素（父元素）的对象，用这种方法，子元素共享了父元素的结构和行为。实现（Realization）是类元之间的语义关系，其中一个类元指定了由另一个类元保证执行的契约。

2015 年 5 月第 42 题

UML 图中，对新开发系统的需求进行建模，规划开发什么功能或测试用例，采用（42）最适合。而展示交付系统的软件组件和硬件之间的关系的图是（43）。

- | | | | |
|----------|-------|-------|-------|
| (42)A.类图 | B.对象图 | C.用例图 | D.交互图 |
| (43)A.类图 | B.部署图 | C.组件图 | D.网络图 |

【答案】C B

【解析】本题考查统一建模语言（UML）的基本知识。

UML 中提供了多种建模系统需求的图，体现系统的静态方面和动态方面。

类图（Class Diagram）展现了一组对象、接口、协作和它们之间的关系。在面向对象系统的建模中，最常见的就是类图，它给出系统的静态设计视图。对象图（Object Diagram）展现了某一时刻一组对象以及它们之间的关系。对象图描述了在类图中所建立的事物的实例的静态快照，给出系统的静态设计视图或静态进程视图。用例图（Use Case Diagram）展现了一组用例、参与者（Actor）以及它们之间的关系。这个视图主要支持系统的行为，即该系统在它的周边环境的语境中所提供的外部可见服务。用例图用于对一个系统的需求进行建模，包括说明这个系统应该做什么（从系统外部的一个视点出发），而不考虑系统应该怎样做。交互图用于对系统的动态方面进行建模。一张交互图表现的是一个交互，由一组对象和它们之间的关系组成，包含它们之间可能传递的消息。交互图表现为序列图、通信图、交互概览图和

时序图，每种针对不同的目的，能适用于不同的情况。序列图是强调消息时间顺序的交互图；通信图是强调接收和发送消息的对象的结构组织的交互图；交互概览图强调控制流的交互图。

时序图 (Timing Diagram) 关注沿着线性时间轴、生命线内部和生命线之间的条件改变。

部署图 (Deployment Diagram) 是用来对面向对象系统的物理方面建模的方法，展现了运行时处理结点以及其中构件（制品）的配置。组件图 (Component Diagram) 展现了一组组件之间的组织和依赖。

2015 年 5 月第 44 题

下图所示为 (44) 设计模式，属于 (45) 设计模式，适用于 (46)。

(44) A. 代理 (Proxy) B. 生成器 (Builder) C. 组合 (Composite) D. 观察者 (Observer)

(45) A. 创建型 B. 结构型 C. 行为 D. 结构型和行为

(46) A. 表示对象的部分一整体层次结构时

B. 当一个对象必须通知其它对象，而它又不能假定其它对象是谁时

C. 当创建复杂对象的算法应该独立于该对象的组成部分及其装配方式时

D. 在需要比较通用和复杂的对象指针代替简单的指针时

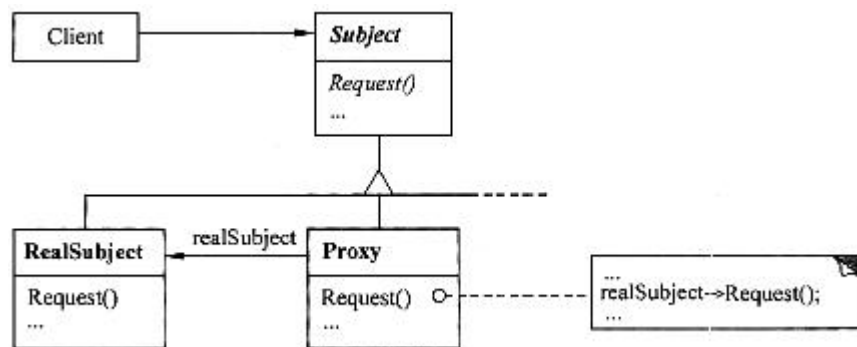
【答案】C B A

【解析】 本题考查设计模式的基本概念。

每一个设计模式都集中于一个特定的面向对象设计问题或设计要点，描述了什么时候使用它，在另一些设计约束条件下是否还能使用，以及使用的效果和如何取舍。按照设计模式的目的可以分为创建型、结构型和行为型三大类。创建型模式与对象的创建有关；结构型模式处理类或对象的组合；行为型模式对类或对象怎样交互和怎样分配职责进行描述。每种设计模式都有其适应性，描述适用于解决的问题场合。

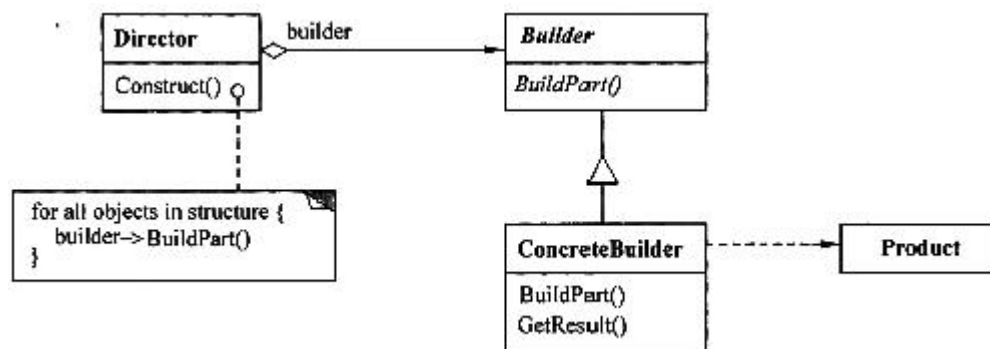
创建型模式包括 Factory Method、Abstract Factory、Builder、Prototype 和 Singleton。结构型模式包括 Adapter (类)、Adapter (对象)、Bridge、Composite、Decorator、Fafade、Flyweight 和 Proxy。行为型模式包括 Interpreter、Template Method、Chain of Responsibility、Command、Iterator、Mediator、Memento Observer State Strategy 和 Visitor。

Proxy 模式的结构图如下所示：



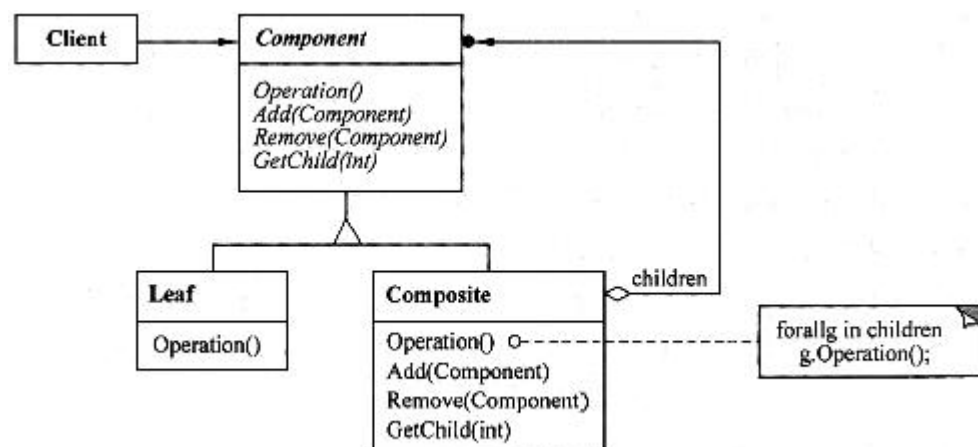
Proxy 模式适用于在需要比较通用和复杂的对象指针代替简单的指针的时候，常见情况有：
 远程代理（Remote Proxy）为一个对象在不同地址空间提供据不代表；虚代理（Virtual Proxy）
 根据需要创建开销很大的对象；保护代理（Protection Proxy）控制对原始对象的访问，用于对
 对象应该有不同的访问权限的时候；智能指引（Smart Reference）取代了简单的指针，它在访
 问对象时执行一些附加操作。

Builder 模式的结构图如下所示：



Builder 模式适用于当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方
 式时；当构造过程必须允许被构造的对象有不同的表示时。

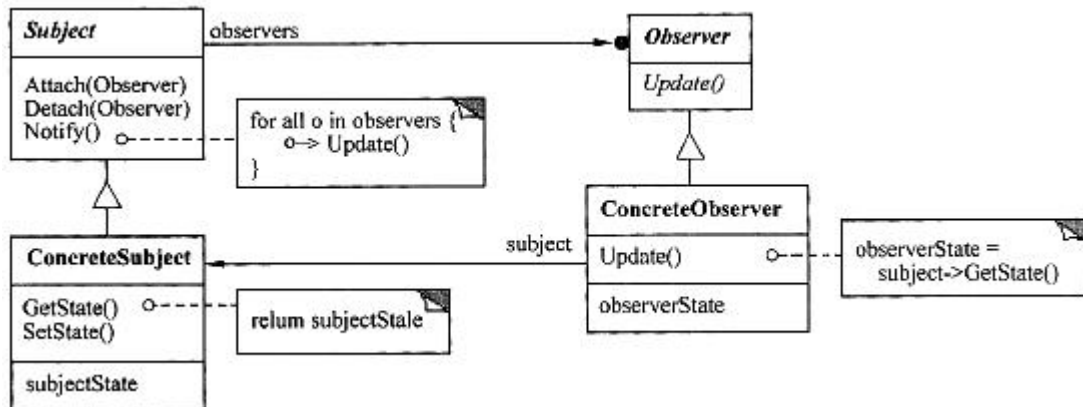
Composite 模式的结构图如下所示：



Composite 模式适用于：想表示对象的部分-整体层次结构；希望用户忽略组合对象与单个对

象的不同，用户将统一地使用组合结构中的所有对象。

Observer 模式的结构图如下所示：



Observer 模式适用于：.当一个抽象模型有两个方面，其中一个方面依赖于另一个方 面。将这两者封装在独立地对象中以使它们可以各自独立地改变和复用；当对一个对象的改变需要同时改变其他对象.而不知道具体有多少对象有待改变时；当一个对象必须通知其他对象，而它又不能假定其他对象是谁，即不希望这些对象是紧耦合的。

2015 年 5 月第 47 题

某些设计模式会引入总是被用作参数的对象。例如 (47) 对象是一个多态 accept 方法的参数。

(47)A.Visitor B.Command C.Memento D.Observer

【答案】A

【解析】本题考查设计模式的概念，对行为模式进行比较。

很多行为模式注重封装变化。当一个程序的某个方面的特征经常发生改变时，这些模式就定义一个封装这个方面的对象。这样，当该程序的其他部分依赖于这个方面时，它们都可以与此对象协作。一些模式引入总是被用作参数的对象。有些模式定义一些可作为令牌进行传递的对象，这些对象将在稍后被调用。

在 Visitor 模式中，一个 Visitor 对象是一个多态的 accept 操作的参数，这个操作作用于该 Visitor 对象访问的对象。在 Command 模式中，令牌代表一个请求；在 Memento 模式中，它代表在一个对象在某个特定时刻的内部状态。在这两种情况下，令牌都可以有一个复杂的内部表示，但客户并不会意识到这一点。在 Observer 模式中，通过引入 Observer 和 Subject 对象来分布通信。

2015 年 5 月第 48 题

对高级语言源程序进行编译或解释的过程可以分为多个阶段，解释方式不包含(48)阶段。

- (48)A.词法分析 B.语法分析 C.语义分析 D.目标代码生成

【答案】D

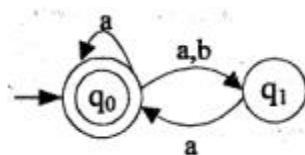
【解析】本题考查程序语言基础知识。

用某种高级语言或汇编语言编写的程序称为源程序，源程序不能直接在计算机上执行。汇编语言源程序需要用一个汇编程序将其翻译成目标程序后才能执行。高级语言源程序则需要对应的解释程序或编译程序对其进行翻译，然后在机器上运行。

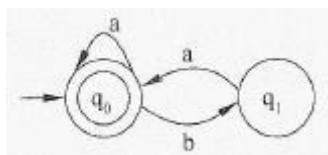
解释程序也称为解释器,它或者直接解释执行源程序，或者将源程序翻译成某种中间代码后再加以执行；而编译程序（编译器）则是将源程序翻译成目标语言程序，然后在计算机上运行目标程序。这两种语言处理程序的根本区别是：在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序（或其某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释程序。简单来说，在解释方式下，翻译源程序时不生成独立的目标程序，而编译器则将源程序翻译成独立保存的目标程序。

2015 年 5 月第 49 题

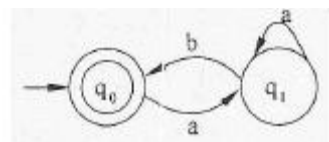
某非确定的有限自动机(NFA)的状态转换图如下图所示（ q_0 既是初态也是终态），与该 NFA 等价的确定的有限自动机(DFA)是(49)。

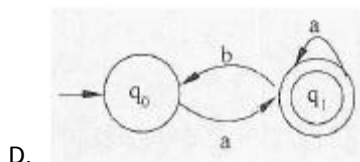
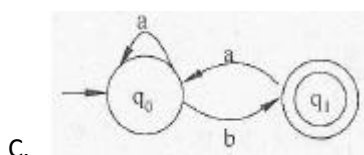


(49)A.



B.





【答案】A

【解析】本题考查程序语言基础知识。

对高级语言源程序进行编译（或解释）方式的翻译过程中，词法分析采用有限自动机作为计算模型。有限自动机分为确定的有限自动机（DFA）和不确定的有限自动机（NFA）。可将一个从 NFA 转换为等价的最小化 DFA。

题中的 NFA 的功能是识别空串以及 b 不能连续出现（即每个 b 后至少含有 1 个 a。）的 a、b 字符串，若是非空串，则以 a 结尾。

选项 A 识别的是空串以及每个 b 后至少含有 1 个 a 的 a、b 字符串，若是非空串，则以 a 结尾。

选项 B 识别空串以及 b 不能连续出现且以 b 结尾的 a、b 字符串。

选项 C 识别 b 不能连续出现且以 b 结尾的 a、b 字符串，不能识别空串。

选项 D 识别 b 不能连续出现且以 a 结尾的 a、b 字符串，不能识别空串。

2015 年 5 月第 50 题

递归下降分析方法是一种 (50) 方法。

(50)A. 自底向上的语法分析

B. 自上而下的语法分析

C. 自底向上的词法分析

D. 自上而下的词法分析

【答案】B

【解析】本题考查程序语言基础知识。

对高级语言源程序进行编译（或解释）方式的翻译过程中，语法分析的任务是根据语言的语法规则，分析单词串是否构成短语和句子，即表达式、语句和程序等基本语言结构，同时检查和处理程序中的语法错误。程序设计语言的绝大多数语法规则可以采用上下文无关文法进行描述。语法分析方法有多种，根据产生语法树的方向，可分为自底向上和自顶向下两类。递归下降分析法和预测分析法是常用的自顶向下分析法。算符优先分析法和 LR 分析法属于自底向上的语法分析方法。

2015 年 5 月第 51 题

若关系 R(H, L, M, P)的主键为全码(All-key), 则关系 R 的主键应 (51)。

(51)A.为 HLMP

B.在集合{H, L, M, P}中任选一个

C.在集合{HL, HM, HP, LM, LP, MP}中任选一个

D.在集合{HLM, HLP, HMP, LMP}中任选一个

【答案】A

【解析】本题考查关系数据库系统中关系的键的基本概念。

在关系数据库系统中,全码(All-key)是指关系模型的所有属性组是这个关系模式的候选键, 本题所有属性组为 HLMP,故本题的正确选项为 A。

2015 年 5 月第 52 题

给定关系模式 R(A1, A2, A3, A4)上的函数依赖集 $F=\{A_1A_3 \rightarrow A_2, A_2 \rightarrow A_3\}$ 。若将 R 分解为 $\rho=\{(A_1, A_2), (A_1, A_3)\}$, 则该分解是 (52) 的。

(52)A.无损联接且不保持函数依赖

B.无损联接且保持函数依赖

C.有损联接且保持函数依赖

D.有损联接且不保持函数依赖

【答案】D

【解析】本题考查关系数据库规范化理论方面的基础知识。

试题 (52) 正确答案为 D。设 $U_1=\{A_1, A_2\}$, $U_2=\{A_1, A_3\}$, 那么可得出:
 $(U_1 \cap U_2) \rightarrow (U_1 - U_2) = A_1 \rightarrow A_2$, $(U_1 \cap U_2) \rightarrow (U_2 - U_1) = A_1 \rightarrow A_3$,
而 $A_1 \rightarrow A_2, A_1 \rightarrow A_3 \notin F^+$, 所以分解 ρ 是有损连接的。
又因为 $F_1=F_2=\emptyset$, $F^+ \neq (F_1 \cup F_2)^+$, 所以分解不保持函数依赖。

2015 年 5 月第 53 题

(53) 算法采用模拟生物进化的三个基本过程“繁殖(选择) -> 交叉(重组) -> 变异(突变)”。

(53)A.粒子群

B.人工神经网络

C.遗传

D.蚁群

【答案】C

【解析】本题考查数据挖掘基本概念方面的基础知识。

数据挖掘就是应用一系列技术从大型数据库或数据仓库中提取人们感兴趣的信息和知识,这些知识或信息是隐含的,事先未知而潜在有用的,提取的知识表示为概念、规则、规律、模式等形式。也可以说,数据挖掘是一类深层次的数据分析。常见和应用最广泛的数据挖掘方法如下。

①决策树:决策树方法是利用信息论中的互信息(信息增益)寻找数据库中具有最大信息量的属性字段,建立决策树的一个结点,再根据该属性字段的不同取值建设树的分支;在每个分支子集中重复建立树的下层结点和分支的过程。国际上最早的、也是最有影响的决策树方法是 Quiulan 研究的 ID3 方法。

②神经网络:神经网络方法是模拟人脑神经元结构,完成类似统计学中的判别、回归、聚类等功能,是一种非线性的模型,主要有三种神经网络模型:前馈式网络、反馈式网络和自组织网络。人工神经网络最大的长处是可以自动地从数据中学习,形成知识,这些知识有些是我们过去未曾发现的,因此它具有较强的创新性。神经网络的知识体现在网络连接的权值上,神经网络的学习主要表现在神经网络权值的逐步计算上。

③遗传算法:遗传算法是模拟生物进化过程的算法,它由三个基本过程组成:繁殖(选择)、交叉(重组)、变异(突变)。采用遗传算法可以产生优良的后代,经过若干代的遗传,将得到满足要求的后代即问题得解。

④关联规则挖掘算法:关联规则是描述数据之间存在关系的规则,形式为“ $A_1A_2...A_n \Rightarrow B_1B_2...B_n$ ”。一般分为两个步骤:求出大数据项集、用大数据项集产生关联规则。

除了上述的常用方法外,还有粗集方法,模糊集合方法, Bayesian Belief Netords, 最邻近算法(K-nearest Neighbors Method, kNN)等。

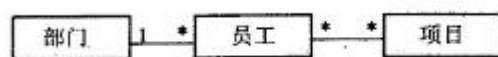
2015 年 5 月第 54 题

部门、员工和项目的关系模式及它们之间的 E-R 图如下所示,其中,关系模式中带实下划线的属性表示主键属性。图中:

部门(部门代码,部门名称,电话)

员工(员工代码,姓名,部门代码,联系方式,薪资)

项目(项目编号,项目名称,承担任务)



若部门和员工关系进行自然连接运算，其结果集为(54)元关系。由于员工和项目之间关系之间的联系类型为(55)，所以员工和项目之间的联系需要转换成一个独立的关系模式，该关系模式的主键是(56)。

- (54) A.5 B.6 C.7 D.8
- (55) A.1 对 1 B.1 对多 C.多对 1 D.多对多
- (56) A. (项目名称, 员工代码) B. (项目编号, 员工代码)
- C. (项目名称, 部门代码) D. (项目名称, 承担任务)

【答案】C D B

【解析】本题考查关系数据库 E-R 模型的相关知识。

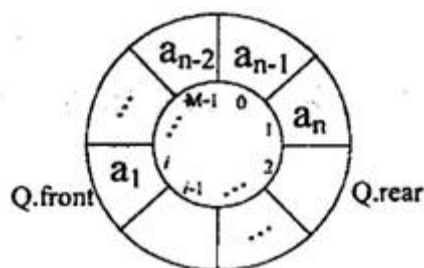
试题 (54) 的正确答案是 C。根据题意部门和员工关系进行自然连接运算，应该去掉一个重复属性“部门代码”，所以自然连接运算的结果集为 7 元关系。

(55) 在 E-R 模型中，用 1 1 表示 1 对 1 联系，用 1 * 表示 1 对多联系，用 * * 表示多对多联系。

(56) 因为员工和项目之间是一个多对多的联系，多对多联系的向关系模式转换的规则是：多对多联系只能转换成一个独立的关系模式，关系模式的名称取联系的名称，关系模式的属性取该联系所关联的两个多方实体的主键及联系的属性，关系的码是多方实体的主键构成的属性组。由于员工关系的主键是员工代码，项目关系的主键是项目编号，因此，根据该转换规则试题 (56) 员工和项目之间的联系的关系模式的主键是 (员工代码, 项目编号)。

2015 年 5 月第 57 题

设某循环队列 Q 的定义中有 front 和 rear 两个域变量，其中，front 指示队头元素的位置，rear 指示队尾元素之后的位置，如下图所示。若该队列的容量为 M，则其长度为(57)。



- (57) A. (Q.rear - Q.front + 1) B. (Q.rcar - Q.front + M)
- C. (Q.rear - Q.front + 1) % M D. (Q.rear - Q.front + M) % M

【答案】D

【解析】本题考查数据结构基础知识。

根据图示,可以计算得知 $(Q.rear-Q.front+M)\%M$ 为队列中的元素个数(即队列长度)。

2015 年 5 月第 58 题

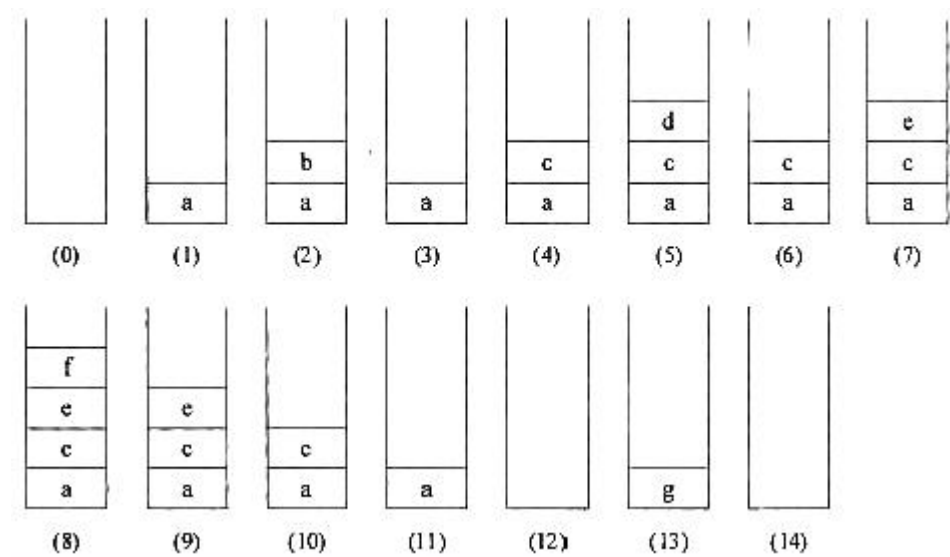
设栈 S 和队列 Q 的初始状态为空,元素 $a b c d e f g$ 依次进入栈 S 。要求每个元素出栈后立即进入队列 Q ,若 7 个元素出队列的顺序为 $b d f e c a g$,则栈 S 的容量最小应该是 (58)。

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

【答案】B

【解析】本题考查数据结构基础知识。

根据队列的特点,元素出队的顺序与入队的顺序相同,因此,可知这 7 个元素的出栈顺序为 $b d f e c a g$ 。对于入栈序列 $a b c d e f g$,得到出栈序列 $b d f e c a g$ 的操作过程为: $push(a)$ 入)、 $push(b)$ 入)、 $pop(b)$ 出)、 $push(c)$ 入)、 $push(d)$ 入)、 $pop(d)$ 出)、 $push(e)$ 入)、 $push(f)$ 入)、 $pop(f)$ 出)、 $pop(e)$ 出)、 $pop(c)$ 出)、 $pop(a)$ 出)、 $push(g)$ 入)、 $pop(g)$ 出),如下图所示,从中可知栈 S 中元素最多时为 4。因此, S 的容量最小为 4。



2015 年 5 月第 59 题

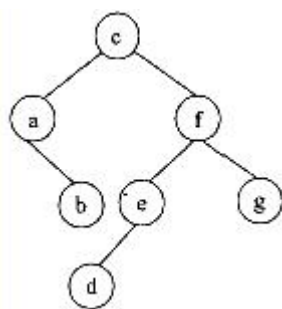
某二叉树的先序遍历序列为 $a b c d e f g$,中序遍历序列为 $a b c d e f g$,则该二叉树是(59)。

(59)A.完全二叉树 B.最优二叉树 C.平衡二叉树 D.满二叉树

【答案】C

【解析】本题考查数据结构基础知识。

根据题中所给的遍历序列，可知其对座的二叉树如下图所示。



2015 年 5 月第 60 题

对某有序顺序表进行折半查找时，(60) 不可能构成查找过程中关键字的比较序列。

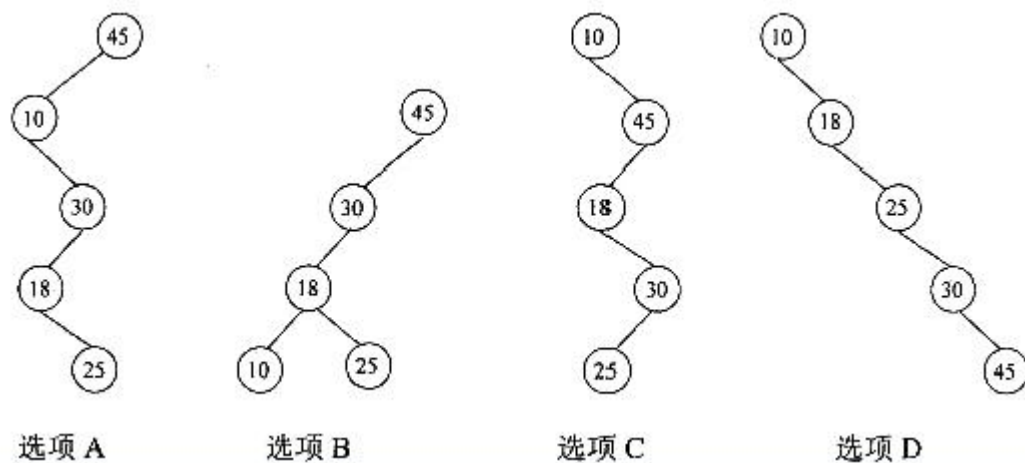
(60) A. 45, 10, 30, 18, 25 B. 45, 30, 18, 25, 10 C. 10, 45, 18, 30, 25 D. 10, 18, 25, 30, 45

【答案】B

【解析】本题考查数据结构基础知识。

进行折半查找时，首先与表中间位置上的元素进行比较，若待查找的元素大于中间元素，则接下来在后半区（是比中间元素更大者组成的有序子表）进行折半查找，否则在前半区（是比中间元素更小者组成的有序子表）进行折半查找。二分查找过程可用二分查找判定树来描述，即大于中间元素时走右分支，小于中间元素时走左分支，等于时查找成功结束。

四个选项的比较序列在二分查找判定树上的查找路径表示为：



显然，选项 B 是不可能的查找路径。

2015 年 5 月第 61 题

用某排序方法对一元素序列进行非递减排序时，若该方法可保证在排序前后排序码相同者的相对位置不变，则称该排序方法是稳定的。简单选择排序法排序方法是不稳定的，(61)

可以说明这个性质。

(61)A.21 48 21* 63 17 B.17 21 21* 48 63 C.63 21 48 21* 17 D.21* 17 48 63 21

【答案】A

【解析】本题考查数据结构基础知识。

对选项 A 进行简单选择排序时，第一趟需交换 17 和 21，导致 21 与 21*的相对位置 发生变化，最后的非递减序列为 17 21* 21 48 63,说明简单选择排序是不稳定的排序方法。

2015 年 5 月第 62 题

优先队列通常采用(62)数据结构实现，向优先队列中插入一个元素的时间复杂度为(63)。

(62)A.堆 B.栈 C.队列 D.线性表

(63)A. $\Theta(n)$ B. $\Theta(1)$ C. $\Theta(\lg n)$ D. $\Theta(n^2)$

【答案】A C

【解析】本题考查算法设计与分析的相关知识。

优先队列是一种常用的数据结构，通常用堆实现。对应于大顶堆和小顶堆，存在最大优先队列和最小优先队列。以最大优先队列为例，优先队列除了具有堆上的一些操作，如调整堆、构建堆之外，还有获得优先队列的最大元素，抽取出优先队列的最大元素，向优先队列插入一个元素和增大优先队列中某个元素的值。其中除了获得优先队列的最大元素的时间复杂度为 $\Theta(1)$ 之外，其他几个操作的时间复杂度均为二叉树的高度，即 $\Theta(\lg n)$ 。

2015 年 5 月第 64 题

在 n 个数的数组中确定其第 $i(1 \leq i \leq n)$ 小的数时，可以采用快速排序算法中的划分思想，对 n 个元素划分，先确定第 k 小的数，根据 i 和 k 的大小关系,进一步处理，最终得到第 i 小的数。划分过程中，最佳的基准元素选择的方法是选择待划分数组的(64)元素。此时，算法在最坏情况下的时间复杂度为（不考虑所有元素均相等的情况）(65)。

(64)A.第一个 B.最后一个 C.中位数 D.随机一个

(65)A. $\Theta(n)$ B. $\Theta(\lg n)$ C. $\Theta(n \lg n)$ D. $\Theta(n^2)$

【答案】C A

【解析】本题考查算法设计与分析的相关知识。

中位数的含义：将一组数据按照由小到大（或由大到小）的顺序排列，如果数据的个数是奇数，则处于中间位置的数就是这组数据的中位数；如果数据的个数是偶数，则中间两个数据

的平均数就是这组数据的中位数。根据题干的描述,选择的基准元素将数组分得越均匀越好,因此中位数是最佳选择。

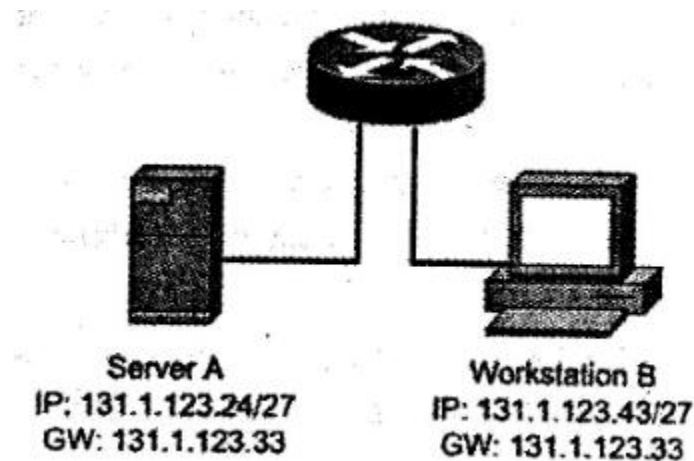
对于该问题,若每次都是选择中位数作为基准元素,则时间复杂度的递归式为:

$$T(n) = T(n/2) + cn$$

求解该递归式,得到 $T(n) = \Theta(n)$ 。

2015 年 5 月第 66 题

在下图所示的网络配置中,发现工作站 B 无法与服务器 A 通信。(66) 故障影响了两者互通。



(66)A.服务器 A 的 IP 地址是广播地址

B.工作站 B 的 IP 地址是网络地址

C.工作站 B 与网关不属于同一子网

D.服务器 A 与网关不属于同一子网

【答案】D

【解析】

服务器 A 的 IP 地址 131.1.123.24/27: 10000011.00000001.01111011.00011000 服务器 A 的地址不是广播地址。

服务器 A 的网关地址 131.1.123.33: 10000011.00000001.01111011.00100001 这个地址与服务器 A 的地址不属于同一个子网。

工作站 B 的 IP 地址 131.1.123.43/27: 10000011.00000001.01111011.00101011 这个地址不是网络地址。

工作站 B 的网关地址 131.1.123.33: 10000011.00000001.01111011.00100001 工作站 B 与网关属于同一个子网。

2015 年 5 月第 67 题

以下关于 VLAN 的叙述中，属于其优点的是_(67)。

- (67)A.允许逻辑地划分网段 B.减少了冲突域的数量
C.增加了冲突域的大小 D.减少了广播域的数量

【答案】A

【解析】

把局域网划分成多个不同的 VLAN，使得网络接入不再局限于物理位置的约束，这样就简化了在网络中增加、移除和移动主机的操作，特别是动态配置的 VLAN，无论主机在哪里，它都处于自己的 VLAN 中。VLAN 内部可以相互通信，VLAN 之间不能直接通信，必须经过特殊设置的路由器才可以连通。这样做的结果是，通过在较大的局域网中创建不同的 VLAN，可以抵御广播风暴的影响，也可以通过设置防火墙来提高网络的安全性。VLAN 并不能直接增强网络的安全性。

2015 年 5 月第 68 题

以下关于 URL 的叙述中，不正确的是_(68)。

- (68)A.使用 www.abc.com 和 abc.com 打开的是同一页面
B.在地址栏中输入 www.abc. com 默认使用 http 协议
C.www.abc.com 中的“www”是主机名
D.www.abc.com 中的“abc.com”是域名

【答案】A

【解析】本题考查 URL 的使用和格式的基本知识。

URL 由三部分组成：资源类型、存放资源的主机域名、资源文件名。

URL 的一般语法格式为（带方括号[]的为可选项）：

protocol :// hostname[:port] / path /filename

其中，protocol 指定使用的传输协议，最常见的是 HTTP 或者 HTTPS 协议，也可以有其他协议，如 file、ftp、,gopher、mms、ed2k 等；hostname 是指主机名，即存放资源的服务域名或者 IP 地址；port 是指各种传输协议所使用的默认端口号，该选项是可选选项，例如 http 的默认端口号为 80,一般可以省略，如果为了安全考虑，可以更改默认的端口号，这时，该选项是必选的；path 是指路径，有一个或者多个“/”分隔，一般用来表示主机上的一个目录或者文件地址；filename 是指文件名，该选项用于指定需要打开 的文件名称。

一般情况下，一个 URL 可以采用“主机名.域名”的形式打开指定页面，也可以单独使用“域名”来打开指定页面，但是这样实现的前提是需进行相应的设置和对应。

2015 年 5 月第 69 题

DHCP 协议的功能是 (69)；FTP 使用的传输层协议为 (70)。

(69)A.WINS 名字解析 B.静态地址分配 C.DNS 名字登录 D.自动分配 IP 地址

(70)A.TCP B.IP C.UDP D.HDLC

【答案】D A

【解析】本题考查 DHCP 和 FTP 两个应用协议。

DHCP 协议的功能是自动分配 IP 地址；FTP 协议的作用是文件传输，使用的传输层协议为 TCP。

2015 年 5 月第 70 题

Why Have Formal Documents?

Finally, writing the decisions down is essential. Only when one writes do the gaps appear and the (71) protrude(突出).The act of writing turns out to require hundreds of mini-decisions,and it is the existence of these that distinguishes clear,exact policies from fuzzy ones.

Second,the documents will communicate the decisions to others. The manager will be continually amazed that policies he took for common knowledge are totally unknown by some member of his team . Since his fundamental job is to keep everybody going in the (72) direction, his chief daily task will be communication, not decision-making,and his documents will immensely (73) this load.

Finally,a manager,s documents give him a data base and checklist. By reviewing them (74) he sees where he is, and he sees what changes of emphasis or shifts in direction are needed.

The task of the manager is to develop a plan and then to realize it. But only the written plan is precise and communicable. Such a plan consists of documents on what,when, how much,where,and who.This small set of critical documents (75) much of the manager,s work. If their comprehensive and critical nature is recognized in the beginning, the manager can approach them as friendly tools rather than annoying busywork. He will set his direction much

more crisply and quickly by doing so.

- | | | | |
|-----------------------|-----------------|----------------|----------------|
| (71)A.inconsistencies | B.consistencies | C.steadiness | D.adaptability |
| (72)A.other | B.different | C.another | D.same |
| (73)A.extend | B.broaden | C.lighten | D.release |
| (74)A.periodically | B.occasionally | C.infrequently | D.rarely |
| (75)A.decides | B.encapsulates | C.realizes | D.recognizes |

【答案】A D C A B

【解析】

为什么要有正式的文档？

首先，书面记录决策是必要的。只有记录下来，分歧才会明朗，矛盾才会突出。书写这项活动需要上百次的细小决定，正是由于它们的存在，人们才能从令人迷惑的现象中得到清晰、确定的策略。

第二，文档能够作为同其他人的沟通渠道。项目经理常常会不断发现，许多理应被普遍认同的策略，完全不为团队的一些成员所知。正因为项目经理的基本职责是使每个人都向着相同的方向前进，所以他的主要工作是沟通，而不是做出决定。这些文档能极大地减轻他的负担。

最后，项目经理的文档可以作为数据基础和检查列表。通过周期性的回顾，他能清楚项目所处的状态，以及哪些需要重点进行更改和调整。

项目经理的任务是制订计划，并根据计划实现。但是只有书面计划是精确和可以沟通的。计划中包括了时间、地点、人物、做什么、资金。这些少量的关键文档封装了一些项目经理的工作。如果一开始就认识到它们的普遍性和重要性，那么就可以将文档作为工具友好地利用起来，而不会让它成为令人厌烦的繁重任务。通过遵循文档开展工作，项目经理能更清晰和快速地设定自己的方向。

试题一

阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某大学为进一步推进无纸化考试，欲开发一考试系统。系统管理员能够创建包括专业方向、课程编号、任课教师等相关考试基础信息，教师和学生进行考试相关的工作。系统与考试有关的主要功能如下。

- (1)考试设置。教师制定试题（题目和答案），制定考试说明、考试时间和提醒时间等考试信息，录入参加考试的学生信息，并分别进行存储。
- (2)显示并接收解答。根据教师设定的考试信息，在考试有效时间内向学生显示考试说明和题目，根据设定的考试提醒时间进行提醒，并接收学生的解答。
- (3)处理解答。根据答案对接收到的解答数据进行处理，然后将解答结果进行存储。
- (4)生成成绩报告。根据解答结果生成学生个人成绩报告，供学生查看。
- (5)生成成绩单。对解答结果进行核算后生成课程成绩单供教师查看。
- (6)发送通知。根据成绩报告数据，创建通知数据并将通知发送给学生；根据成绩单数据，创建通知数据并将通知发送给教师。

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

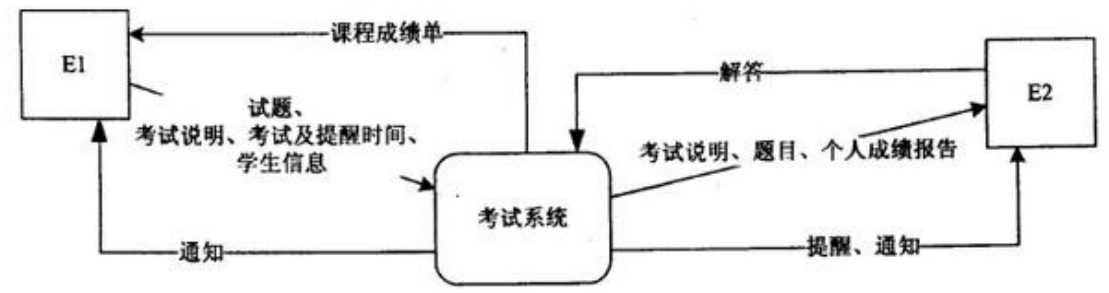


图 1-1 上下文数据流图

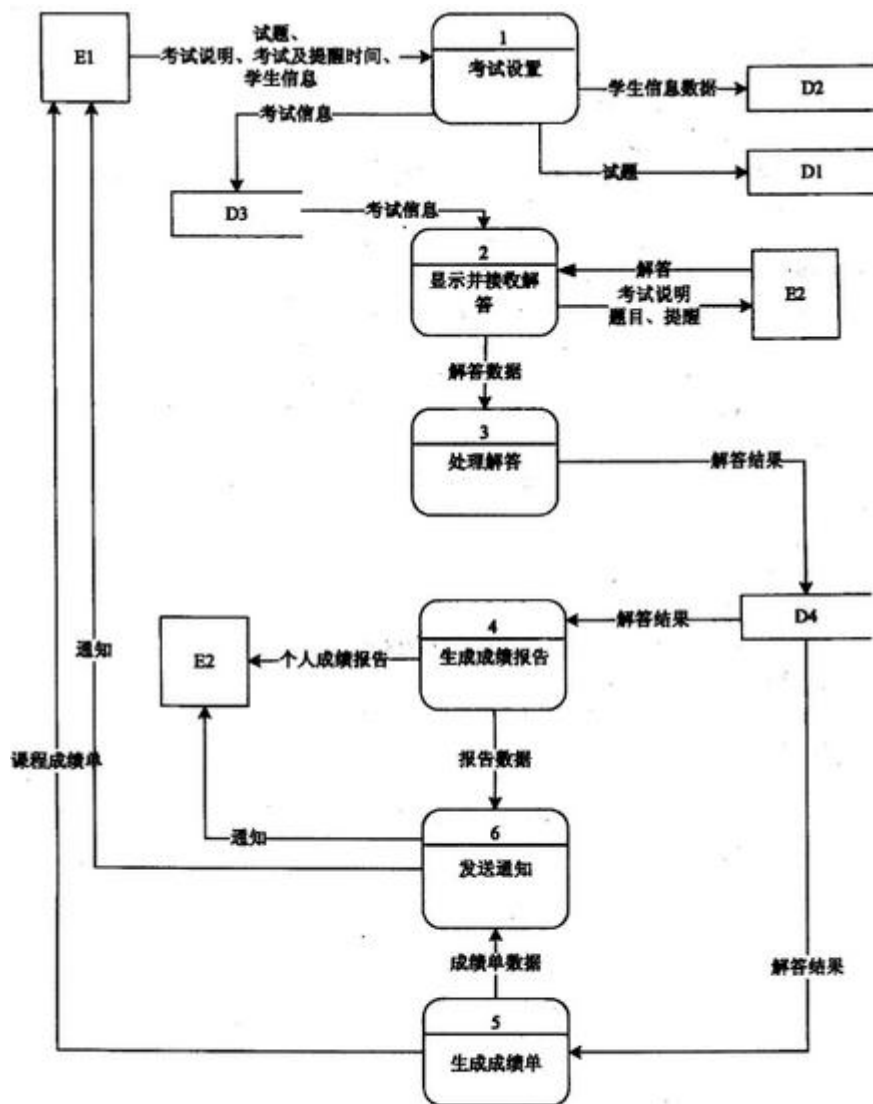


图 1-2 0 层数据流图

【问题 1】（2 分）

使用说明中的词语，给出图 1-1 中的实体 E1~E2 的名称。

E1: 教师;

E2: 学生。

解析: 本题考查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, 与往年考点类似, 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\上下文 DFD (顶层 DFD)通常用来确定系统边界，将待开发系统本身看作一个大的加工（处理），然后根据谁为系统提供数据流，谁使用系统提供的的数据流，来确定外部实体。建模出的上下文 DFD 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。在上下文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，建模 0 层 DFD，将上下文 DFD 中的加工进一步分解成多个加工，识别这些加工的输入输出数据流，使得所有上下文 DFD 中的输入数据流，经过这些加工之后变换成上下文 DFD 的输出数据流。根据 0 层 DFD 中加工的复杂程度进一步建模加工的内容。

在建模分层 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，注意要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字上相同，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。

本问题考查上下文 DFD,要求确定外部实体。考察系统的主要功能，不难发现，针对系统与考试有关的主要功能，涉及到教师和学生，系统管理员不在与考试有关的主要功能中涉及，另外没有提到其他与系统交互的外部实体。根据描述（1）中“教师制定试题等考试信息”等信息，描述（2）中“根据教师设定的考试信息，在考试有效时间内向学生显示考试说明和题目”，从而即可确定 E1 为“教师”实体，E2 为“学生”实体。

【问题 2】（4 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1~D4 的名称。

D1: 试题（表）或题目和答案（表）

D2: 学生信息（表）

D3: 考试信息（表）

D4: 解答结果（表）

解析: 本问题要求确定 0 层数据流图中的数据存储。分析说明中和数据存储有关的描述，说明中（1）中“教师制定试题（题目和答案），制定考试说明、考试时间和提醒时间等考试信息，录入参加考试的学生信息，并分别进行存储”，可知 D1、D2 和 D3 为试题、学生信息和考试信息，再从图 1-2 中流入 D2 的数据流名称“学生信息数据”，确定 D2 是学生信息，

流入 D1 的数据流名称为“试题”，确定 D1 为试题，流入 D3 的数据流名称为考试信息，确定 D3 为考试信息。说明中 (3)根据答案对接收到的解答数据进行处理，然后将解答结果进行存储，确定 D4 是解答结果。其他描述中对数据存储的使用更多说明，进一步确定 D1~D4 满足上述分析。

【问题 3】(4 分)

根据说明和图中词语，补充图 1-2 中缺失的数据流及其起点和终点。

数 据 流	起 点	终 点
答案	D1 或试题（表）或题目和答案（表）	3 或处理解答
题目	D1 或试题（表）或题目和答案（表）	2 或显示并接收解答

解析：本问题要求补充缺失的数据流及其起点和终点。通过不同层的 DFD 以及说明中描述和图之间的对应#系加以确定。首先对照图 1-1 和图 1-2 的输入、输出数据流，发现数据流的数量和名称均相同，所以，需进一步考查说明中的功能描述和图 1-1 中的数据流的对应关系，以确定缺失的是加工之间还是加工与数据存储之间的数据流。

说明 (2)显示并接收解答，需要“根据教师设定的考试信息，在考试有效时间内向学生显示考试说明和题目”，对照图 1-2 可以看出，加工 2 缺少所要显示的题目的输入源，即缺失输入流“题目”，题目存储于数据存储“试题”中，因此，缺少的数据流为从题目 (D1)到加工 2 显示并接收解答的题目。说明 (3)处理解答，需要“根据答案对 接收到的解答数据进行处理”，对照图 1-2 可以看出，加工 3 “处理解答”缺少输入流“答案”，而从说明 (1)中可以看出“答案”存储在试题（题目和答案）数据存储中 (D1)， 因此确定缺失的一条数据流“答案”，从 D1 或试题到加工 3 或处理解答。

【问题 4】(5 分)

图 1-2 所示的数据流图中，功能(6)发送通知包含创建通知并发送给学生或老师。请分解图 1-2 中加工(6)，将分解出的加工和数据流填入答题纸的对应栏内。（注：数据流的起点和终点须使用加工的名称描述）

分解为加工：发送通知和加工：创建通知

数 据 流	起 点	终 点
报告数据	生成成绩报告	创建通知
成绩单数据	生成成绩单	创建通知
通知数据	创建通知	发送通知

解析：

本问题考查建模分层 DFD 时的分解粒度。在说明（6）发送通知中，“根据成绩报告数据，创建通知数据并将通知发送给学生；根据成绩单数据，创建通知数据并将通知发送给教师。”说明功能（6）发送通知包含创建通知并发送给学生或老师。在图 1-2 中建模为一个加工，完成的功能是依据不同的输入数据流创建通知，然后发送给相应的外部实体老师或学生，因此为了进一步清晰每个加工的职责，需对图 1-2 中原有加工 6 进行分解，分解为“创建通知”和“发送通知”。创建通知针对输入数据流“报告数据”和“成绩单数据”，这两条数据流保持原有的起点，终点即为创建通知。创建通知产生出“通知数据”“通知数据”作为加工“发送通知”的输入流，进一步根据通知数据是针对哪个外部实体而发送“通知”给相应的学生或者教师。至此，对图 1-2 中原有加工 6 的分解完成。

试题二

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某省针对每年举行的足球联赛，拟开发一套信息管理系统，以方便管理球队、球员、主教练、主裁判、比赛等信息。

【需求分析】

(1)系统需要维护球队、球员、主教练、主裁判、比赛等信息。

球队信息主要包括：球队编号、名称、成立时间、人数、主场地址、球队主教练。

球员信息主要包括：姓名、身份证号、出生日期、身高、家庭住址。

主教练信息主要包括：姓名、身份证号、出生日期、资格证书号、级别。

主裁判信息主要包括：姓名、身份证号、出生日期、资格证书号、获取证书时间、级别。

(2)每支球队有一名主教练和若干名球员。一名主教练只能受聘于一支球队，一名球员只能效力于一支球队。每支球队都有自己的唯一主场场地，且场地不能共用。

(3)足球联赛采用主客场循环制，一周进行一轮比赛，一轮的所有比赛同时进行。

(4)一场比赛有两支球队参加，一支球队作为主队身份、另一支作为客队身份参与比赛。一场比赛只能有一名主裁判，每场比赛有唯一的比赛编码，每场比赛都记录比分和日期。

【概念结构设计】

根据需求分析阶段的信息，设计的实体联系图（不完整）如图 2-1 所示。

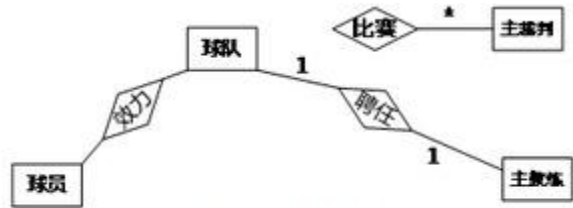


图2-1 实体联系图

【逻辑结构设计】

根据概念结构设计阶段完成的实体联系图，得出如下关系模式（不完整）：

球队（球队编号，名称，成立时间，人数，主场地址）

球员（姓名，身份证号，出生日期，身高，家庭住址， (1) ）

主教练（姓名，身份证号，出生日期，资格证书号，级别， (2) ）

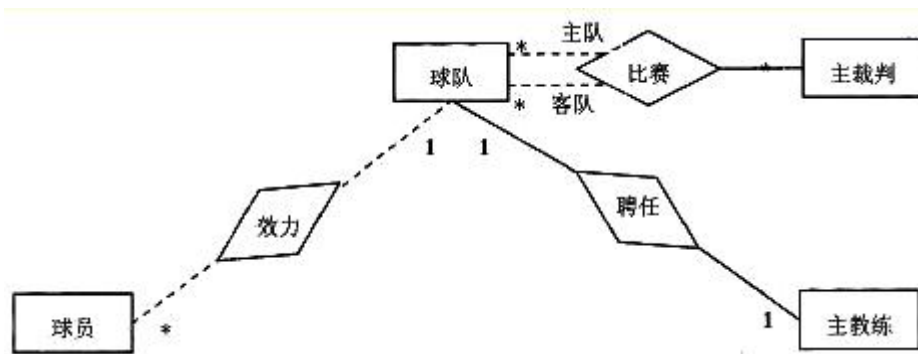
主裁判（姓名，身份证号，出生日期，资格证书号，获取证书时间，级别）

比赛（比赛编码，主队编号，客队编号，主裁判身份证号，比分，日期）

【问题 1】（6 分）

补充图 2-1 中的联系和联系的类型。

图 2-1 中的联系“比赛”应具有的属性是哪些？



比赛联系应具有的属性包括：比赛编码，比分，日期。

解析：本题考查数据库概念结构设计及向逻辑结构转换的掌握。

此类题目要求考生认真阅读题目，根据题目的需求描述，给出实体间的联系。

根据题意由“一名球员只能效力于一支球队”可知球队和球员之间为 1:* 联系。由“一场比赛有两支球队参加，一支球队作为主队身份、另一支作为客队身份参与比赛”可知球队分别按照“主队”和“客队”两种角色参与“比赛”的联系。“比赛”应具有的属性：比赛编码，出分和日期。

【问题 2】（4 分）

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)~(2)补充完整。

(1) 球队编号；

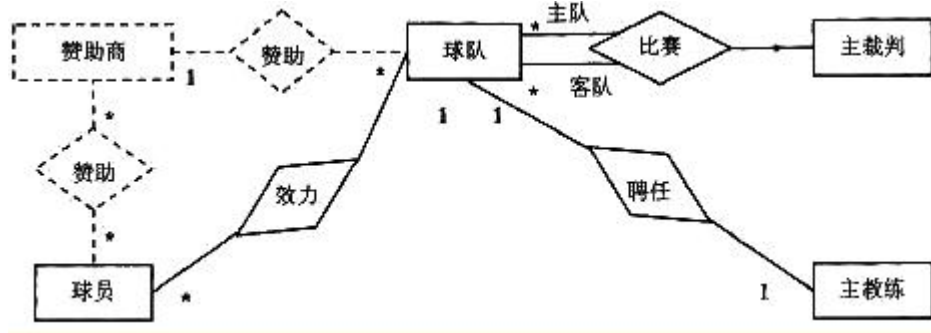
(2) 球队编号。

解析：根据问题 1 分析可知球队和球员之间为 1:* 联系，所以在球员关系里应该包括球队的主键，即“球队编号”。根据“每支球队有一名主教练，一名主教练只能受聘于一支球队”可知球队和教练之间为 1:1 联系，而球队关系已经给定，所以需要在主教练关系中包含球队的主键，即“球队编号”。

【问题 3】（5 分）

现在系统要增加赞助商信息，赞助商信息主要包括赞助商名称和赞助商编号。

赞助商可以赞助某支球队，一支球队只能有一个赞助商，但赞助商可以赞助多支球队。赞助商也可以单独赞助某些球员，一名球员可以为多个赞助商代言。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。



解析：根据题意由“赞助商可以赞助某支球队，一支球队只能有一个赞助商，但赞助商可以赞助多支球队”可知赞助商和球队之间为 1:* 联系。由“赞助商也可以单独赞助某些球员，一名球员可以为多个赞助商代言”可知赞助商和球员之间为 1:* 联系。

试题三

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某物品拍卖网站为参与者提供物品拍卖平台，组织拍卖过程，提供在线或线下交易服务。

网站主要功能描述如下：

(1) 拍卖参与者分为个人参与者和团体参与者两种。不同的团体也可以组成新的团体参与拍卖活动。网站记录每个参与者的名称。

(2) 一次拍卖中，参与者或者是买方，或者是卖方。

(3) 一次拍卖只拍出来自一个卖方的一件拍卖品；多个买方可以出价；卖方接受其中一个出价作为成交价，拍卖过程结束。

(4) 在拍卖结算阶段，买卖双方可以选择两种成交方式：线下成交，买卖双方在事先约定好的成交地点，当面完成物价款的支付和拍卖品的交付；在线成交，买方通过网上支付平台支付物价款，拍卖品由卖方通过快递邮寄给买方。

一次拍卖过程的基本事件流描述如下：

(1) 卖方在网站上发起一次拍卖，并设置本次拍卖的起拍价。

(2) 确定拍卖标的以及拍卖标的保留价（若在拍卖时间结束时，所有出价均低于拍卖标的保留价，则本次拍卖失败）。

(3) 在网站上发布本次拍卖品的介绍。

(4) 买方参与拍卖，给出竞拍价。

(5) 卖方选择接受一个竞拍价作为成交价，结束拍卖。

(6) 系统记录拍卖成交价，进入拍卖结算阶段。

(7) 卖方和买方协商拍卖品成交方式，并完成成交。

现采用面向对象方法对系统进行分析与设计，得到如表 3-1 所示的类列表以及如图 3-1 所示的类图，类中关键属性与方法如表 3-2 所示。

序号	类名	说明
C1	SellerRole	一次拍卖中的卖方
C2	Item	拍卖品
C3	Auction	拍卖活动
C4	Sale	拍卖结算
C5	AuctionParticipant	拍卖参与者
C6	Interchange	成交方式
C7	OneParticipant	个人参与者
C8	OfflinePay	线下成交
C9	CompositeParticipant	团体参与者
C10	OnlinePay	在线成交
C11	Bid	拍卖标的
C12	BuyerRole	一次拍卖中的买方

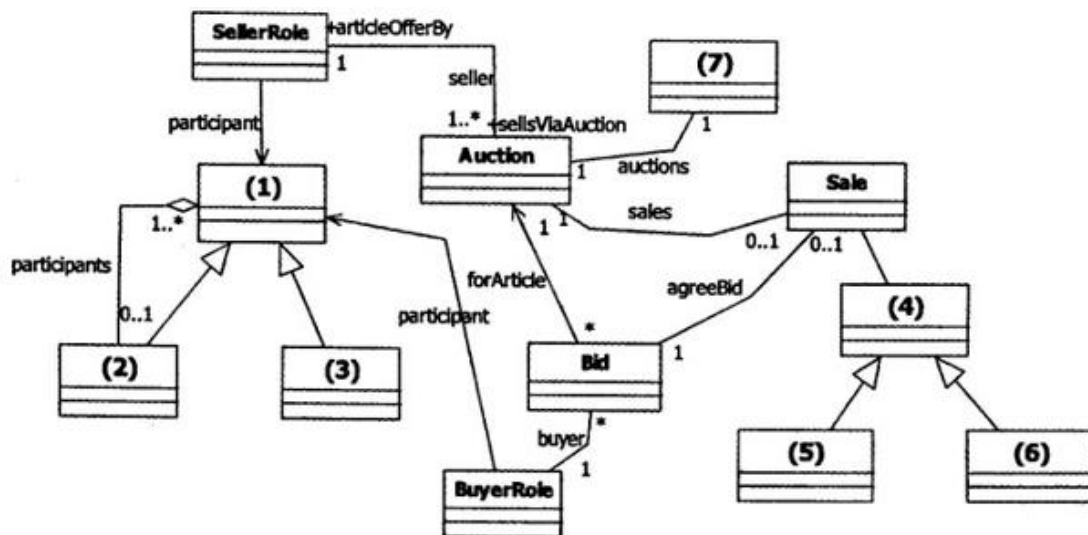


图 3-1 类图

表3-2 关键属性与方法列表

序号	名称	说明
M1	name	属性名, 用户名
M2	description	属性名, 拍卖品描述
M3	minBidPrice	属性名, 拍卖的起拍价
M4	agreePrice	属性名, 拍卖成交价
M5	bidPrice	属性名, 拍卖标的保留价
M6	address	属性名, 线下成交地点
M7	sellerAccount	属性名, 卖方网上支付账户名
M8	buyerAddress	属性名, 买方邮寄地址
M9	placeBidForAuction	方法名, 为拍卖品出竞拍价
M10	sellNewArticle	方法名, 发起一次拍卖

【问题1】（7分）

根据说明中的描述，给出图3-1中(1)~(7)所对应的类名（类名使用表3-1中给出的序号）。

	类		类
(1)	C5 (AuctionParticipant)	(5)	C8 (OfflinePay)
(2)	C9 (CompositeParticipant)	(6)	C10 (OnlinePay)
(3)	C7 (OneParticipant)	(7)	C2 (Item)
(4)	C6 (Interchange)		

注：(5)和(6)的类名可互换。

解析：本题属于经典的考题，主要考查面向对象分析方法与设计的基本概念。在建模方面，本题中只涉及到了UML类图。类图上的考点也是比较常规的对类的识别以及类中属性及方法的确定，题目难度不大。

图3-1共需要确定7个类，可以先从图中几个特殊关系处入手，即(1)~(3)和(4)~(6)。先来分析(1)~(3)，这是一个继承+聚集的结构，而且联系的名称“participants”是一个比较明显的提示，说明这个层次结构是与【说明】中的功能描述(1)相对应的。表3-1，与之相关的类是C5 (AuctionParticipant)、C7 (OneParticipant)和C9 (CompositeParticipant)。C7、C9是特殊的参与者，所以(1)处应该为C5；(2)处应该为C9，这个聚集关系针对着【说明】中的“不同的团体也可以组成新的团体参与拍卖活动”需求；(3)处为C7。

结合【说明】和表3-1，另外一组具有“一般-特殊”关系的类只有C6 (Interchange)、C8 (OfflinePay)和C10 (OnlinePay)。显而易见，C8和C10是C6的两种具体方式，所以(4)处应该为C6，(5)、(6)处分别为C8和C10。

这样(7)处对应的类只能是Item了。结合【说明】和表3-1可知，(7)处对应的类表达的应该是拍卖中的拍卖品，所以(7)处应该是C2。

【问题2】（5分）

根据说明中的描述，确定表3-2中的属性/方法分别属于哪个类（类名、方法/属性名使用表3-1、3-2中给出的序号）。

属性/方法序号	所属类的序号	属性/方法序号	所属类的序号
M1	C5	M6	C8
M2	C2	M7	C10
M3	C3	M8	C10
M4	C4	M9	C12
M5	C11	M10	C1

【问题 3】（3 分）

在图 3-1 采用了何种设计模式？以 100 字以内文字说明采用这种设计模式的原因。

在图 3-1 中使用了 Composite 模式。

以树形结构表示个人参与者和团体参与者之间的“部分-整体”关系，使得对单个对象和组合对象的使用具有一致性。

解析：在【说明】部分有一个很明显的提示：“拍卖参与者分为个人参与者和团体参与者两种。不同的团体也可以组成新的团体参与拍卖活动”。这里很清晰地表达了一种“部分 - 整体”的层次关系，这种关系非常适合于采用 Composite (组合) 设计模式来表达。

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

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】

n-皇后问题是在 n 行 n 列的棋盘上放置 n 个皇后，使得皇后彼此之间不受攻击，其规则是任意两个皇后不在同一行、同一列和相同的对角线上。

拟采用以下思路解决 n-皇后问题：第 i 个皇后放在第 i 行。从第一个皇后开始，对每个皇后，从其对应行（第 i 个皇后对应第 i 行）的第一列开始尝试放置，若可以放置，确定该位置，考虑下一个皇后；若与之前的皇后冲突，则考虑下一列；若超出最后一列，则重新确定上一个皇后的位置。重复该过程，直到找到所有的放置方案。

【C 代码】

下面是算法的 C 语言实现。

(1)常量和变量说明

pos: 一维数组，pos[i]表示第 i 个皇后放置在第 i 行的具体位置

count: 统计放置方案数

i, j, k: 变量

N: 皇后数

(2)C 程序

```
#include <stdio.h>

#include <math.h>

#define N4

/*判断第 k 个皇后目前放置位置是否与前面的皇后冲突*/
in isplace(int pos[], int k) {
    int i;
    for(i=1; i<k; i++) {
        if( (1) || fabs(i-k) == fabs(pos[i] - pos[k])) {
            return();
        }
    }
    return 1;
}
```

```

int main() {

    int i,j,count=1;

    int pos[N+1];

    //初始化位置

    for(i=1; i<=N; i++) {

        pos[i]=0;

    }

    (2) ;

    while(j>=1) {

        pos[j]= pos[j]+1;

        /*尝试摆放第 i 个皇后*/

        while(pos[j]<=N&& (3) _){

            pos[j]= pos[j]+1;

        }

        /*得到一个摆放方案*/

        if(pos[j]<=N&&j== N) {

            printf("方案%d: ",count++);

            for(i=1; i<=N; i++){

                printf("%d ",pos[i]);

            }

            printf("\n");

        }

        /*考虑下一个皇后*/

        if(pos[j]<=N&& (4) ) {

            j=j+1;

        } else{ //返回考虑上一个皇后

            pos[j]=0;

            (5) ;

        }

    }

```

```
}  
    return 1;  
}
```

【问题 1】（10 分）

根据以上说明和 C 代码，填充 C 代码中的空（1）～（5）。

- （1）`pos[i]==pos[k]` 或其等价形式
（2）`j=1`
（3）`! isplace(pos,j)` 或其等价形式
（4）`j<N` 或其等价形式
（5）`j=j-1` 或其等价形式

解析：本题考查算法设计和 C 程序设计语言的相关知识。

此类题目要求考生认真阅读题目，理解算法思想，并思考将算法思想转化为具体的程序设计语言的代码。

根据题干描述。空（1）所在的代码行判断皇后合法放置的约束条件，即不在同一行，这通过把第 i 个皇后放在第 i 行实现，条件“`fabs(i-k)==fabs(pos[i]-pos[k])`”判断的是当前摆放的皇后是否与之前摆放的皇后在同一对角线上。因此，空（1）判断的是当前摆放的皇后是否和之前摆放的皇后在同一列上，即应填入“`pos[i]==pos[k]`”。

根据算法思想和主函数上下文，空（2）处应该考虑第 1 个皇后，即初始化 j 为 1，空（2）填写“`j=1`”。空（3）所在的行是判断放置第 j 个皇后的位置是否合适，“`pos[j] <= N`”表示在该行的合法列上，但还需要进一步判断是否与前面的皇后有冲突，根据满足条件后的语句，尝试放入下一列，因此空（3）处填入“`!isplace(pos, j)`”。根据前面的注释，空（4）所在的行是考虑下一个皇后，其条件是，当前皇后找到了合适的位置，而且还存在下一个皇后，因此空（4）处应填入“`j < N`”。根据下面的注释，若当前皇后没有找到合适的位置，则应回溯，即再次考虑上一个皇后的位置，因此空（5）处填入“`j=j-1`”。

【问题 2】（2 分）

根据以上说明和 C 代码，算法采用了（6）设计策略。

(6) 回溯法

解析：从上述题干的叙述和 C 代码很容易看出，从第一个皇后开始，对每个皇后总是从第 1 个位置开始尝试，找到可以放置的合法位置；若某个皇后在对应的行上没有合法位置，则回溯到上一个皇后，尝试将上一个皇后放置另外的位置。这是典型的深度优先的系统搜索方式，即回溯法的思想。

【问题 3】(3 分)

上述 C 代码的输出为：(7) 。

(7)

方案 1：2 4 1 3；

方案 2：3 1 4 2。

试题五

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

【说明】

某图书管理系统中管理着两种类型的文献：图书和论文。现在要求统计所有馆藏文献的总页码（假设图书馆中有一本 540 页的图书和两篇各 25 页的论文，那么馆藏文献的总页码就是 590 页）。采用 Visitor（访问者）模式实现该要求，得到如图 5-1 所示的类图。

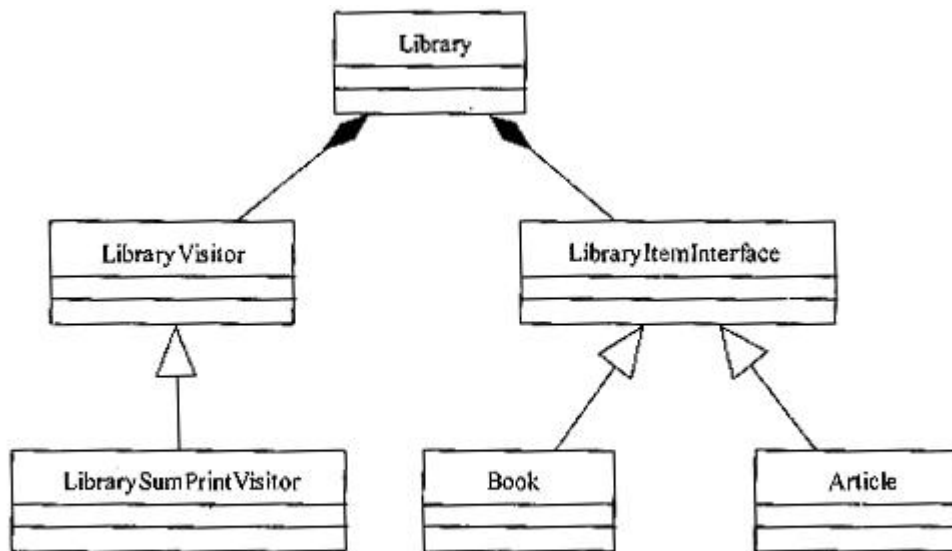


图 5-1 Visitor 模式类图

【C++ 代码】

```
class LibraryVisitor;
```

```
class LibraryItemInterface{
```

```
public:
```

```
    (1) :
```

```
};
```

```
class Article : public LibraryItemInterface {
```

```
private:
```

```
    string m_title; //论文名
```

```
    string m_author; //论文作者
```

```
    int m_start_page;
```

```
    int m_end_page;
```

```

public:
    Article(string p_author, string p_title, int p_start_page,int p_end_page );

    int getNumber()fPages();

    void accept(Library Visitor* visitor);

};

```

```

class Book : public LibraryItemInterface {

private:
    string m_title; //书名
    string m_author; //作者
    int m_pages; //页数

public:
    Book(string p_author, string p_title, int p_pages);

    int getNumber()fPages();

    void accept(LibraryVisitor* visitor);

};

```

```

class LibraryVisitor {

public:
    (2) ;
    ( 3) ;
    virtual void printSum() = 0;

};

```

```

class LibrarySumPrintVisitor : public LibraryVisitor { //打印总页数

private:
    int sum;

public:
    LibrarySumPrintVisitor();

    void visit(Book* p_book);

```

```

        void visit(Article* p_article);

        void printSum();

};

// visitor.cpp

int Article::getNumber()fPages(){

    return m_end_page - m_start_page;

}

void Article::accept(LibraryVisitor* visitor) { (4) ;}

Book::Book(string p_author, string p_title, int p_pages ) {

    m_title = p_title;

    m_author = p_author;

    m_pages = p_pages;

}

int Book::getNumberOfPages(){ return m_pages; }

void Book::accept(LibraryVisitor* visitor){ (5) ; }

    //其余代码省略

}

```

【问题 1】（15 分）

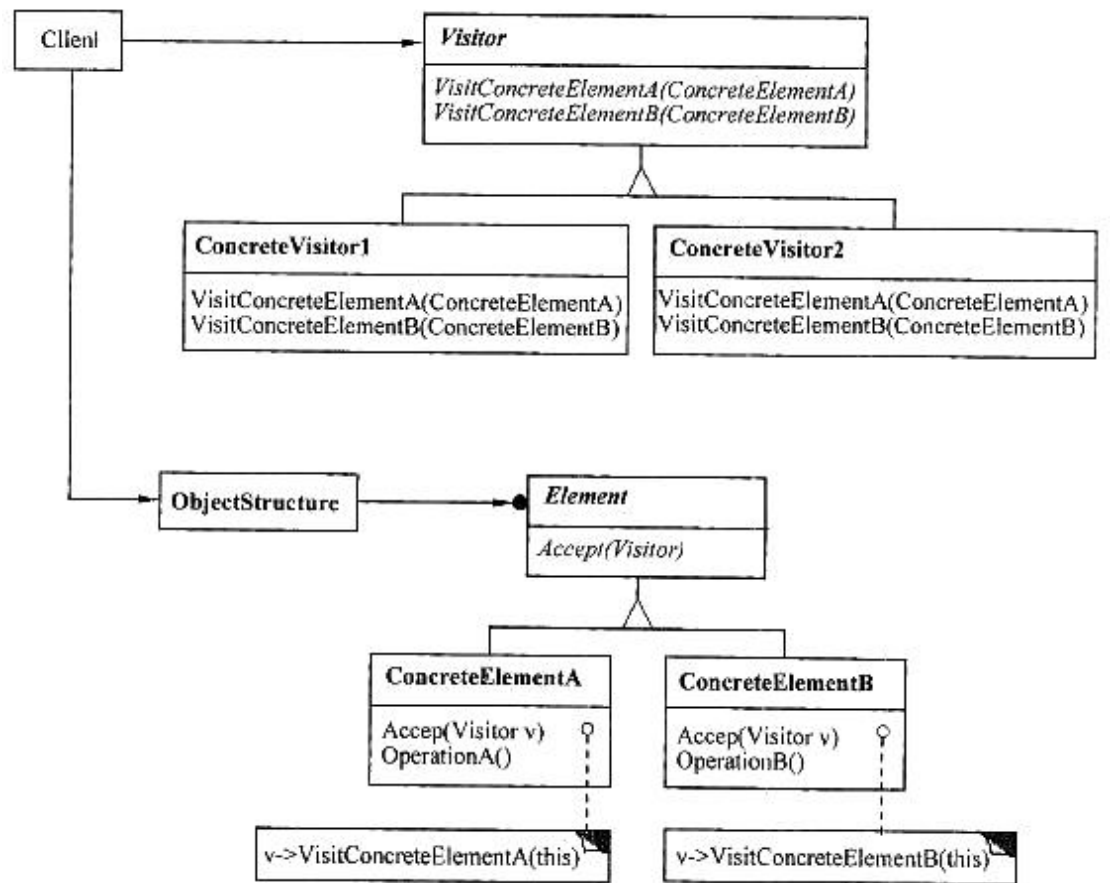
阅读上述说明和 C++代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

- (1) virtual void accept(LibraryVisitor*visitor)=0
- (2) virtual void visit(Book* p_book)=0
- (3) virtual void visit(Article* p_article)=0
- (4) visitor->visit(this)
- (5) visitor->visit(this)

解析：本题考查 Visitor (访问者) 模式的基本概念和应用。

访问者模式是行为设计模式中的一种。行为模式不仅描述对象或类的模式，还描述它们之间的通信模式。这些模式刻画了在运行时难以跟踪的复杂的控制流。

访问者模式表示一个作用于某对象结构中的各元素的操作。它使在不改变各元素的类的前提下可以定义作用于这些元素的新操作。此模式的结构图如下图所示。



Visitor (访问者) 为该对象结构中 **ConcreteElement** 的每一个类声明一个 **visit** 操作。该操作的名字和特征标识了发送 **visit** 请求给该访问者的哪个类。这使得访问者可以确定正被访问元素的具体的类。这样访问者就可以通过该元素的特定接口直接访问它。

Concrete Visitor (具体访问者) 实现每个有 **Visitor** 声明的操作，每个操作实现本算法的一部分，而该算法片段乃是对应于结构中对象的类。**ConcreteVisitor** 为该算法提供了上下文并存储它的局部状态。这一状态常常在遍历该结构的过程中累积结果。

Element (元素) 定义以一个访问者为参数的 **Accept** 操作。

ConcreteElement (具体元素) 实现以一个访问者为参数的 **Accept** 操作。

ObjectStructure (对象结构) 能枚举它的元素；可以提供一个高层的接口以允许该访问者访问它的元素；可以是一个组合或者一个集合，如一个列表或一个无序集合。

本题中类 **Library** 对应着上图中的 **Client**，**Library Visitor** 对应着 **Visitor**，**LibrarySumPrintVisitor**

对应着 Concrete Visitor。LibraryItemInterface 对应着上图中的元素部分。下面可以结合程序代码来完成程序填空了。

LibraryItemInterface 在本题中充当着 Element 的作用,其中应定义以一个访问者为参数的 Accept 操作。对照其两个子类 Article 和 Book 的代码,可以得知该操作的原型是 void accept(LibraryVisitor* visitor)。由于该操作的具体实现在子类 Article 和 Book 中,所以这里适用于采用 C++ 中的纯虚拟函数来实现。由此可以得知,(1)处应填写“virtual void accept(LibraryVisitor* visitor) = 0”。

(2) 和 (3)空与类 LibraryVisitor 有关。由前文分析已知,Library Visitor 对应着访问者模式中的 Visitor,其作用是为类 LibrarySumPrintVisitor 声明 Visit 操作。类 LibrarySumPrintVisitor 需要访问两种不同的元素,每种元素应该对应不同的 visit 操作。再结合类 LibrarySumPrintVisitor 的定义部分,可以得知 (2)和 (3)处应给出分别以 Book 和 Article 为参数的 visit 方法,同样采用纯虚拟函数机制。因此(2)和 (3)处分别为“virtual void visit(Book* p_book) = 0”、“virtual void visit(Article* p_article) = 0”。

(4)和 (5)处考查的是 accept 接口的实现。由访问者模式的结构图可以看出,在 Book 和 Article 中 accept 方法的实现均为 visitor->visit(this)。

试题六（共 15 分）

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

【说明】

某图书管理系统中管理着两种类型的文献：图书和论文。现在要求统计所有馆藏文献的总页码（假设图书馆中有一本 540 页的图书和两篇各 25 页的论文，那么馆藏文献的总页码就是 590 页）。采用 Visitor（访问者）模式实现该要求，得到如图 6-1 所示的类图。

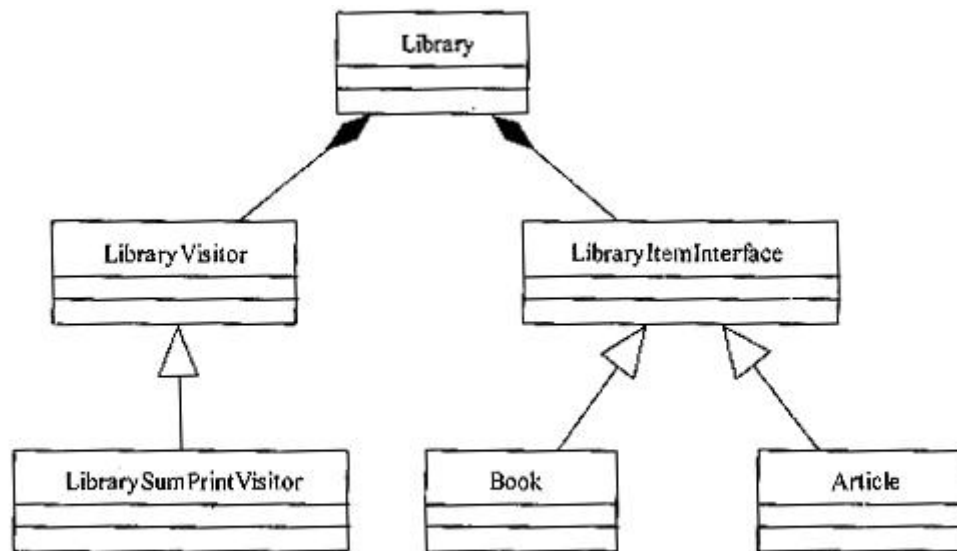


图 5-1 Visitor 模式类图

【Java 代码】

```
import java.util.*;
```

```
interface LibraryVisitor {
```

```
    (1)    :
```

```
    (2)    :
```

```
    void printSum();
```

```
}
```

```
class LibrarySumPrintVisitor implements LibraryVisitor { //打印总页数
```

```
    private int sum = 0;
```

```
    public void visit(Book p_book) {
```

```
        sum = sum + p_book.getNumberOfPages();
```

```

    }

    public void visit(Article p_article) {

        sum = sum + p_article.getNumberOfPages();

    }


    public void printSum(){

        System.out.println("SUM = " + sum);

    }

}

interface LibraryItemInterface {

    (3) ;

}

class Article implements LibraryItemInterface{

    private String m_title; //论文名 。

    private String m_author; //论文作者

    private int m_start_page;

    private int m_end_page;

    public Article(String p_author, String p_title,int p_start_page,int p_end_page){

        m_title=p_title;

        m_author= p_author;

        m_end_page=p_end_page;

    }

    public int getNumbelOfPages(){

        rctum m_end_page - m_start_page;

    }

    public void acccpt(LibraryVisitor Visitor){

        (4) :

    }

}

```



```
class Book implements LibraryItemInterface{

    private String m_title; //书名

    private String m_author; //书作者

    private int m_pages; //页教

    public Book(String p_author, String p_title,int p_pages){

        m_title= p_title;

        m_author= p_author;

        m_pages= p_pages;

    }

    public int getNumberOfPages(){

        return m_pages;

    }

    public void accept(LibraryVisitor visitor){

        (5)____;

    }

}
```

【问题 1】（15 分）

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

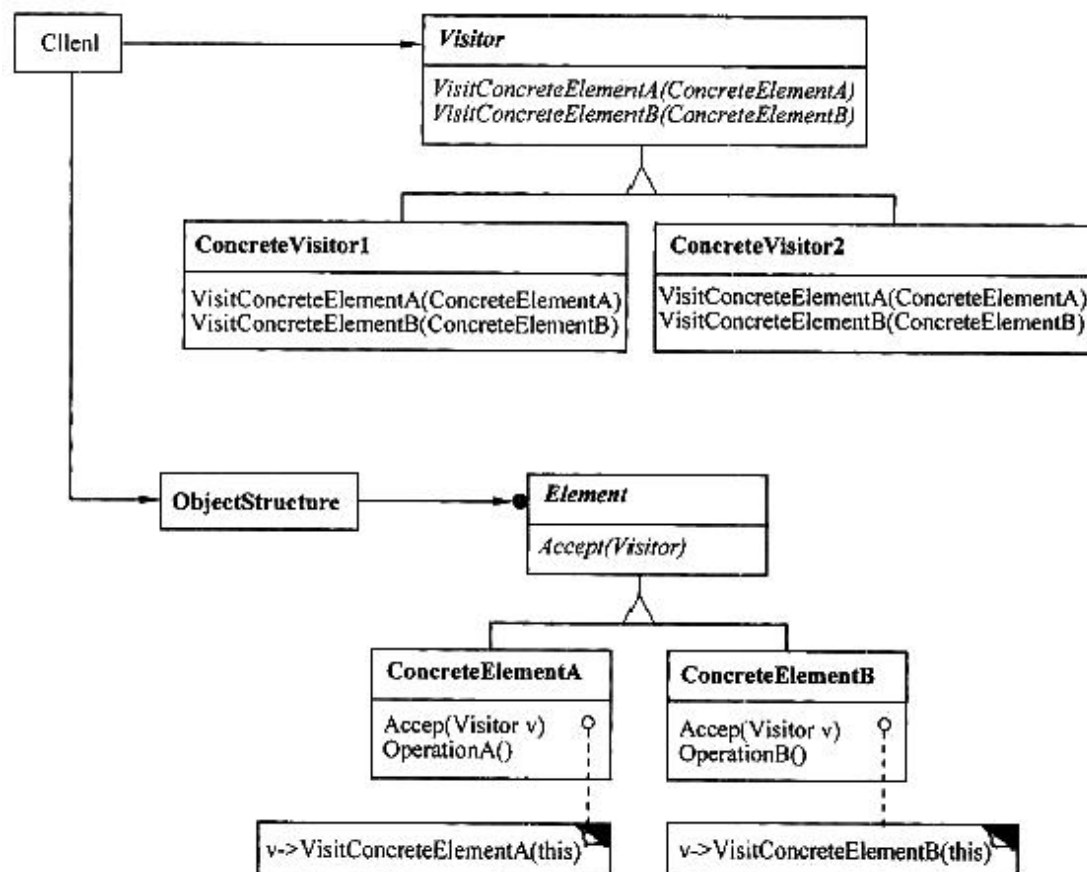
- (1) void visit(Book p_book)
- (2) void visit(Article p_article)
- (3) void accept(LibraryVisitor visitor)
- (4) visitor.visit(this)
- (5) visitor.visit(this)

解析：本题考查 Visitor (访问者) 模式的基本概念和应用。

访问者模式是行为设计模式中的一种。行为模式不仅描述对象或类的模式，还描述它们之间

的通信模式。这些模式刻画了在运行时难以跟踪的复杂的控制流。

访问者模式表示一个作用于某对象结构中的各元素的操作。它使在不改变各元素的类的前提下可以定义作用于这些元素的新操作。此模式的结构图如下图所示。



Visitor (访问者) 为该对象结构中 **ConcreteElement** 的每一个类声明一个 **Visit** 操作。该操作的名字和特征标识了发送 **Visit** 请求给该访问者的哪个类。这使得访问者可以确定正被访问元素的具体的类。这样访问者就可以通过该元素的特定接口直接访问它。

Concrete Visitor (具体访问者) 实现每个有 **Visitor** 声明的操作，每个操作实现本算法的一部分，而该算法片段乃是对应于结构中对象的类。**ConcreteVisitor** 为该算法提供了上下文并存储它的局部状态。这一状态常常在遍历该结构的过程中累积结果。

Element (元素) 定义以一个访问者为参数的 **Accept** 操作。

ConcreteElement (具体元素) 实现以一个访问者为参数的 **Accept** 操作。

ObjectStructure (对象结构) 能枚举它的元素；可以提供一个高层的接口以允许该访问者访问它的元素；可以是一个组合或者一个集合，如一个列表或一个无序集合。

本题中类 **Library** 对应着上图中的 **Client**，**Library Visitor** 对应着 **Visitor**，**LibrarySumPrintVisitor** 对应着 **Concrete Visitor**。**LibraryItemInterface** 对应着上图中的元素部分。下面可以结合程序

代码来完成程序填空了。

(1)和(2)空与类 `LibraryVisitor` 有关。由前文分析已知, `LibraryVisitor` 对应着访问者模式中的 `Visitor`, 其作用是为类 `LibrarySumPrintVisitor` 声明 `Visit` 操作。类 `LibrarySumPrintVisitor` 需要访问 2 种不同的元素, 每种元素应该对应不同的 `visit` 操作。再结合类 `LibrarySumPrintVisitor` 的定义部分, 可以得知 (2)和 (3)处应给出分别以 `Book` 和 `Article` 为参数的 `visit` 方法。因此 (1)和(2)处分别为 “`void visit(Book p_book)`”、“`void visit(Article p_article)`”。

`LibraryItemInterface` 在本题中充当着 `Element` 的作用, 其中应定义以一个访问者为参数的 `Accept` 操作。对照实现该接口的两个类 `Article` 和 `Book` 的代码, 可以得知该操作的原型是 `void accept(LibraryVisitor visitor)`。由此可以得知, (3)处应填写“ `void accept(LibraryVisitor visitor)` ”。(4)和(5)处考查的是 `accept` 接口的实现。由访问者模式的结构图可以看出, 在 `Book` 和 `Article` 中 `accept` 方法的实现均为 `visitor.visit(this)`。