

以下关于 CPU 的叙述中，错误的是(1)。

- (1) A. CPU 产生每条指令的操作信号并将操作信号送往相应的部件进行控制
- B. 程序计数器 PC 除了存放指令地址，也可以临时存储算术/逻辑运算结果
- C. CPU 中的控制器决定计算机运行过程的自动化
- D. 指令译码器是 CPU 控制器中的部件

【答案】B

【解析】本题考查计算机硬件组成基础知识。

CPU 是计算机的控制中心，主要由运算器、控制器、寄存器组和内部总线等部件组成。控制器由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成，它是发布命令的“决策机构”，即完成协调和指挥整个计算机系统的操作。它的主要功能有：从内存中取出一条指令，并指出下一条指令在内存中的位置；对指令进行译码或测试，并产生相应的操作控制信号，以便启动规定的动作；指挥并控制 CPU、内存和输入输出设备之间数据的流动。

程序计数器 (PC) 是专用寄存器，具有寄存信息和计数两种功能，又称为指令计数器，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 的初始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执行的，因此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的地址得到。

以下关于 CISC (Complex Instruction Set Computer, 复杂指令集计算机) 和 RISC (Reduced Instruction Set Computer, 精简指令集计算机) 的叙述中，错误的是(2)。

- (2) A. 在 CISC 中，其复杂指令都采用硬布线逻辑来执行
- B. 采用 CISC 技术的 CPU，其芯片设计复杂度更高
- C. 在 RISC 中，更适合采用硬布线逻辑执行指令
- D. 采用 RISC 技术，指令系统中的指令种类和寻址方式更少

【答案】A

【解析】本题考查指令系统和计算机体系结构基础知识。

CISC (Complex Instruction Set Computer, 复杂指令集计算机) 的基本思想是：进一步增强原有指令的功能，用更为复杂的新指令取代原先由软件子程序完成的功能，实现软件

功能的硬件化，导致机器的指令系统越来越庞大而复杂。CISC 计算机一般所含的指令数目至少 300 条以上，有的甚至超过 500 条。

RISC (Reduced Instruction Set Computer, 精简指令集计算机) 的基本思想是：通过减少指令总数和简化指令功能，降低硬件设计的复杂度，使指令能单周期执行，并通过优化编译提高指令的执行速度，采用硬布线控制逻辑优化编译程序。在 20 世纪 70 年代末开始兴起，导致机器的指令系统进一步精炼而简单。

浮点数的一般表示形式为 $N = 2^E \times F$ ，其中 E 为阶码，F 为尾数。以下关于浮点表示的叙述中，错误的是 (3)。两个浮点数进行相加运算，应首先 (4)。

- (3) A. 阶码的长度决定浮点表示的范围，尾数的长度决定浮点表示的精度
B. 工业标准 IEEE754 浮点数格式中阶码采用移码、尾数采用原码表示
C. 规格化指的是阶码采用移码、尾数采用补码
D. 规格化表示要求将尾数的绝对值限定在区间 $[0.5, 1)$
- (4) A. 将较大的数进行规格化处理 B. 将较小的数进行规格化处理
C. 将这两个数的尾数相加 D. 统一这两个数的阶码

【答案】C D

【解析】 本题考查数据表示基础知识。

为了提高运算的精度，需要充分地利用尾数的有效数位，通常采取浮点数规格化形式，即规定尾数的最高数位必须是一个有效值，即 $1/2 < |F| < 1$ 。在尾数用补码表示时，规格化浮点数应满足尾数最高数位与符号位不同，即当 $1/2 \leq |F| < 1$ 时，应有 $0.1xx \dots x$ 形式；当 $-1 \leq M < -1/2$ 时，应有 $1.0xx \dots x$ 形式。

需要注意的是，当 $M = -1/2$ 时，对于原码来说是规格化数，而对于补码来说不是规格化数。

两个浮点数进行相加运算时，首先需要对阶（使它们的阶码一致），然后再进行尾数的相加处理。

以下关于校验码的叙述中，正确的是 (5)。

- (5) A. 海明码利用多组数位的奇偶性来检错和纠错
B. 海明码的码距必须大于等于 1
C. 循环冗余校验码具有很强的检错和纠错能力

D. 循环冗余校验码的码距必定为 1

【答案】A

【解析】本题考查校验码基础知识。

一个编码系统中任意两个合法编码（码字）之间不同的二进制位数称为这两个码字的码距，而整个编码系统中任意两个码字的最小距离就是该编码系统的码距。为了使一个系统能检查和纠正一个差错，码间最小距离必须至少是 3。

海明码是一种可以纠正一位差错的编码，是利用奇偶性来检错和纠错的校验方法。海明码的基本意思是给传输的数据增加 r 个校验位，从而增加两个合法消息（合法码字）的不同位的个数（海明距离）。假设要传输的信息有 m 位，则经海明编码的码字就有 $n=m+r$ 位。循环冗余校验码(CRC)编码方法是在 k 位信息码后再拼接 r 位的校验码，形成长度为 n 位的编码，其特点是检错能力极强且开销小，易于用编码器及检测电路实现。

在数据通信与网络中，通常 k 相当大，由一千甚至数千数据位构成一帧，而后采用 CRC 码产生 r 位的校验位。它只能检测出错误，而不能纠正错误。一般取 $r=16$ ，标准的 16 位生成多项式有 $CRC-16=x^{16} + x^{15} + x^2 + 1$ 和 $CRC-CCITT= x^{16} + x^{12} + x^5 + 1$ 。一般情况下， r 位生成多项式产生的 CRC 码可检测出所有的双错、奇数位错和突发长度小于等于 r 的突发错。用于纠错目的的循环码的译码算法比较复杂。

以下关于 Cache 的叙述中，正确的是 (6)。

- (6) A. 在容量确定的情况下，替换算法的时间复杂度是影响 Cache 命中率的关键因素
B. Cache 的设计思想是在合理成本下提高命中率
C. Cache 的设计目标是容量尽可能与主存容量相等
D. CPU 中的 Cache 容量应大于 CPU 之外的 Cache 容量

【答案】B

【解析】本题考查高速缓存基础知识。

Cache 是一个高速小容量的临时存储器，可以用高速的静态存储器(SRAM)芯片实现，可以集成到 CPU 芯片内部，或者设置在 CPU 与内存之间，用于存储 CPU 最经常访问的指令或者操作数据。Cache 的出现是基于两种因素：首先是由于 CPU 的速度和性能提高很快而主存速度较低且价格高，其次是程序执行的局部性特点。因此，才将速度比较快而容量有限的 SRAM 构成 Cache，目的在于尽可能发挥 CPU 的高速度。很显然，要尽可能发挥 CPU 的高速度，就必须用硬件实现其全部功能。

网络安全体系设计可从物理线路安全、网络安全、系统安全、应用安全等方面来进行，其中，数据库容灾属于 (7)。

- (7) A. 物理线路安全和网络安全 B. 应用安全和网络安全
C. 系统安全和网络安全 D. 系统安全和应用安全

【答案】D

【解析】

网络安全体系设计是逻辑设计工作的重要内容之一，数据库容灾属于系统安全和应用安全考虑范畴。

包过滤防火墙对数据包的过滤依据不包括 (8)。

- (8) A. 源 IP 地址 B. 源端口号 C. MAC 地址 D. 目的 IP 地址

【答案】C

【解析】 本题考查防火墙相关知识。

包过滤防火墙对数据包的过滤依据包括源 IP 地址、源端口号、目标 IP 地址和目标端口号。

某网站向 CA 申请了数字证书，用户通过 (9) 来验证网站的真伪。

- (9) A. CA 的签名 B. 证书中的公钥 C. 网站的私钥 D. 用户的公钥

【答案】A

【解析】 本题考查数字证书相关知识。

数字证书是由权威机构 CA 证书授权 (Certificate Authority) 中心发行的，能提供在 Internet 上进行身份验证的一种权威性电子文档，人们可以在因特网交往中用它来证明自己的身份和识别对方的身份。

数字证书包含版本、序列号、签名算法标识符、签发人姓名、有效期、主体名和主体公钥信息等并附有 CA 的签名，用户获取网站的数字证书后通过验证 CA 的签名来确认数字证书的有效性，从而验证网站的真伪。

在用户与网站进行安全通信时，用户发送数据时使用网站的公钥（从数字证书中获得）加密，收到数据时使用网站的公钥验证网站的数字签名，网站利用自身的私钥对发送的消息签名和对收到的消息解密。

下列智力成果中，能取得专利权的是(10)。

- (10)A. 计算机程序代码 B. 游戏的规则和方法
C. 计算机算法 D. 用于控制测试过程的程序

【答案】D

【解析】

数字证书是由权威机构 CA 证书授权 (Certificate Authority) 中心发行的，能提供在 Internet 上进行身份验证的一种权威性电子文档，人们可以在因特网交往中用它来证明自己的身份和识别对方的身份。

数字证书包含版本、序列号、签名算法标识符、签发人姓名、有效期、主体名和主体公钥信息等并附有 CA 的签名，用户获取网站的数字证书后通过验证 CA 的签名来确认数字证书的有效性，从而验证网站的真伪。

在用户与网站进行安全通信时，用户发送数据时使用网站的公钥（从数字证书中获得）加密，收到数据时使用网站的公钥验证网站的数字签名，网站利用自身的私钥对发送的消息签名和对收到的消息解密。

软件权利人与被许可方签订一份软件使用许可合同。若在该合同约定的时间和地域范围内，软件权利人不得再许可任何第三人以此相同的方法使用该项软件，但软件权利人可以自己使用，则该项许可使用是(11)。

- (11)A. 独家许可使用 B. 独占许可使用 C. 普通许可使用 D. 部分许可使用

【答案】A

【解析】

软件许可使用一般有独占许可使用、独家许可使用和普通许可使用三种形式。独占许可使用，许可的是专有使用权，实施独占许可使用后，软件著作权人不得将软件使用权授予第三方，软件著作权人自己不能使用该软件；独家许可使用，许可的是专有使用权，实施独家许可使用后，软件著作权人不得将软件使用权授予第三方，软件著作权人自己可以使用该软件；普通许可使用，许可的是非专有使用权，实施普通许可使用后，软件著作权人可以将软件使用权授予第三方，软件著作权人自己可以使用该软件。

多媒体中的“媒体”有两重含义，一是指存储信息的实体；二是指表达与传递信息的载

体。(12)是存储信息的实体。

- (12)A. 文字、图形、磁带、半导体存储器 B. 磁盘、光盘、磁带、半导体存储器
C. 文字、图形、图像、声音 D. 声卡、磁带、半导体存储器

【答案】B

【解析】

通常所说的“媒体(Media)”包括两重含义：一是指信息的物理载体，即存储和传递信息的实体，如手册、磁盘、光盘、磁带以及相关的播放设备等（本题只涉及存储信息）；二是指承载信息的载体，即信息的表现形式（或者说传播形式），如文字、声音、图像、动画和视频等，即 CCITT 定义的存储媒体和表示媒体。表示媒体又可以分为三种类型：视觉类媒体（如位图图像、矢量图形、图表、符号、视频和动画等）、听觉类媒体（如音响、语音和音乐等）和触觉类媒体（如点、位置跟踪，力反馈与运动反馈等）。视觉和听觉类媒体是信息传播的内容，触觉类媒体是实现人机交互的手段。

RGB8:8:8 表示一帧彩色图像的颜色数为(13)种。

- (13)A. 23 B. 28 C. 224 D. 2512

【答案】C

【解析】本题考查多媒体基础知识（图像深度）。

图像深度是指存储每个像素所用的位数，也是用来度量图像分辨率的。像素深度确定彩色图像的每个像素可能有的颜色数，或者确定灰度图像的每个像素可能有的灰度级数。如一幅图像的图像深度为 b 位，则该图像的最多颜色数或灰度级为 2^b 种。显然，表示一个像素颜色的位数越多，它能表达的颜色数或灰度级就越多。例如，只有 1 个分量的单色图像，若每个像素有 8 位，则最大灰度数目为 $2^8=256$ ；一幅彩色图像的每个像素用 R、G、B 三个分量表示，若 3 个分量的像素位数分别为 4、4、2，则最大颜色数目为 $2^4+2^4+2^2=210=1024$ ，就是说像素的深度为 10 位，每个像素可以是 210 种颜色中的一种。表示一个像素的位数越多，它能表达的颜色数目就越多，它的深度就越深。

位图与矢量图相比，位图(14)。

- (14)A. 占用空间较大，处理侧重于获取和复制，显示速度快
B. 占用空间较小，处理侧重于绘制和创建，显示速度较慢
C. 占用空间较大，处理侧重于获取和复制，显示速度较慢

D. 占用空间较小，处理侧重于绘制和创建，显示速度快

【答案】A

【解析】

矢量图形是用一系列计算机指令来描述和记录一幅图的内容，即通过指令描述构成一幅图的所有直线、曲线、圆、圆弧、矩形等图元的位置、维数和形状，也可以用更为复杂的形式表示图像中曲面、光照和材质等效果。矢量图法实质上是用数学的方式（算法和特征）来描述一幅图形图像，在处理图形图像时根据图元对应的数学表达式进行编辑和处理。在屏幕上显示一幅图形图像时，首先要解释这些指令，然后将描述图形图像的指令转换成屏幕上显示的形状和颜色。编辑矢量图的软件通常称为绘图软件，如适于绘制机械图、电路图的AutoCAD 软件等。这种软件可以产生和操作矢量图的各个成分，并对矢量图形进行移动、缩放、叠加、旋转和扭曲等变换。编辑图形时将指令转变成屏幕上所显示的形状和颜色，显示时也往往能看到绘图的过程。由于所有的矢量图形部分都可以用数学的方法加以描述，从而使得计算机可以对其进行任意放大、缩小、旋转、变形、扭曲、移动和叠加等变换，而不会破坏图像的画面。但是，用矢量图形格式表示复杂图像（如人物、风景照片），并且要求很高时，将需要花费大量的时间进行变换、着色和处理光照效果等。因此，矢量图形主要用于表示线框型的图画、工程制图和美术字等。

位图图像是指用像素点来描述的图。图像一般是用摄像机或扫描仪等输入设备捕捉实际场景画面，离散化为空间、亮度、颜色（灰度）的序列值，即把一幅彩色图或灰度图分成许许多多的像素（点），每个像素用若干二进制位来指定该像素的颜色、亮度和属性。位图图像在计算机内存中由一组二进制位组成，这些位定义图像中每个像素点的颜色和亮度。图像适合于表现比较细腻，层次较多，色彩较丰富，包含大量细节的图像，并可直接、快速地在屏幕上显示出来。但占用存储空间较大，一般需要进行数据压缩。

在采用结构化方法进行系统分析时，根据分解与抽象的原则，按照系统中数据处理的流程，用 (15) 来建立系统的逻辑模型，从而完成分析工作。

(15) A. E-R 图 B. 数据流图 C. 程序流程图 D. 软件体系结构

【答案】B

【解析】 本题考查结构化分析方法中图形工具的作用。

数据流图摆脱系统的物理内容，在逻辑上描述系统的功能、输入、输出和数据存储等，是系统逻辑模型的重要组成部分。

面向对象开发方法的基本思想是尽可能按照人类认识客观世界的方法来分析和解决问题，(16)方法不属于面向对象方法。

- (16) A. Booch B. Coad C. OMT D. Jackson

【答案】D

【解析】本题考查面向对象开发方法。

面向对象开发方法有 Booch 方法、Coad 方法和 OMT 方法。Jackson 方法是一种面向数据结构的开发方法。

确定构建软件系统所需要的人数时，无需考虑 (17)。

- (17) A. 系统的市场前景 B. 系统的规模
C. 系统的技术复杂性 D. 项目计划

【答案】A

【解析】本题考查项目管理内容。

在对软件开发资源进行规划时，为了确定构建软件系统所需的人数，需要考虑软件系统的规模、系统的技术复杂性、项目计划和开发人员的技术背景等方面，而与系统是否有市场前景无关。

一个项目为了修正一个错误而进行了变更。但这个错误被修正后，却引起以前可以正确运行的代码出错。(18)最可能发现这一问题。

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

【答案】C

【解析】

本题考查软件测试知识。回归测试是在软件发生变更之后进行的测试，以发现在变更时可能引起的其他错误。

风险预测从两个方面评估风险，即风险发生的可能性以及 (19)。

- (19) A. 风险产生的原因 B. 风险监控技术
C. 风险能否消除 D. 风险发生所产生的后果

【答案】D

【解析】

本题考查风险预测知识。风险预测从风险发生的可能性大小以及风险发生所产生的后果是否严重两个方面评估风险。

许多程序设计语言规定，程序中的数据都必须具有类型，其作用不包括_(20)。

- (20) A. 便于为数据合理分配存储单元
B. 便于对参与表达式计算的数据对象进行检查
C. 便于定义动态数据结构
D. 便于规定数据对象的取值范围及能够进行的运算

【答案】C

【解析】本题考查程序语言基础知识。

数据具有类型，便于编译程序在基础机器中完成对值的布局，同时还可用于检查表达式中对运算的应用是否正确。

以下关于 C/C++ 语言指针变量的叙述中，正确的是_(21)。

- (21) A. 指针变量可以是全局变量也可以是局部变量
B. 必须为指针变量与指针所指向的变量分配相同大小的存储空间
C. 对指针变量进行算术运算是没有意义的
D. 指针变量必须由动态产生的数据对象来赋值

【答案】A

【解析】本题考查程序语言基础知识。

变量是内存单元的抽象，用于在程序中表示数据。当变量存储的是内存单元地址时，称为指针变量，或者说指针变量指向了另一个变量。指针量可以定义在函数或复合语句内，也可以定义在所有的函数之外，即可以是全局变量，也可以是局部变量。需要区分指针变量与指针所指向的变量，无论指针变量指向何种变量，其存储空间大小都是一样的。当指针变量指向数组中的一个元素时，对指针变量进行算术运算可以使其指向同一个数组中的其他元素。

将高级语言源程序翻译为机器语言程序的过程中常引入中间代码。以下关于中间代码的叙述中，错误的是_(22)。

- (22) A. 不同的高级程序语言可以产生同一种中间代码
B. 使用中间代码有利于进行与机器无关的优化处理

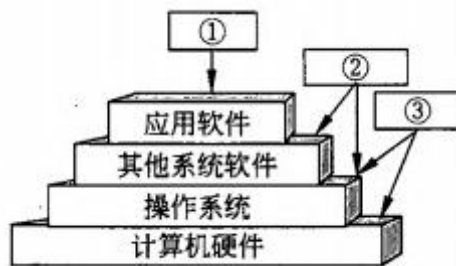
- C. 使用中间代码有利于提高编译程序的可移植性
- D. 中间代码与机器语言代码在指令结构上必须一致

【答案】D

【解析】本题考查程序语言基础知识。

“中间代码”是一种简单且含义明确的记号系统，与具体的机器无关，可以有若干种形式。可以将不同的高级程序语言翻译成同一种中间代码。由于与具体机器无关，使用中间代码有利于进行与机器无关的优化处理，以及提高编译程序的可移植性。

操作系统是裸机上的第一层软件，其他系统软件(如_(23)等)和应用软件都是建立在操作系统基础上的。下图①②③分别表示_(24)。



- (23)A. 编译程序、财务软件和数据库管理系统软件
 - B. 汇编程序、编译程序和 Java 解释器
 - C. 编译程序、数据库管理系统软件和汽车防盗程序
 - D. 语言处理程序、办公管理软件和气象预报软件
- (24)A. 应用软件开发者、最终用户和系统软件开发者
 - B. 应用软件开发者、系统软件开发者和最终用户
 - C. 最终用户、系统软件开发者和应用软件开发者
 - D. 最终用户、应用软件开发者和系统软件开发者

【答案】B D

【解析】本题考查操作系统基本概念。

财务软件、汽车防盗程序、办公管理软件和气象预报软件都属于应用软件，而选项 A、C 和 D 中含有这些软件。选项 B 中汇编程序、编译程序和数据库管理系统软件都属于系统软件。

计算机系统由硬件和软件两部分组成。通常把未配置软件的计算机称为裸机，直接使用裸机不仅不方便，而且将严重降低工作效率和机器的利用率。操作系统 (Operating System)

的目的是为了填补人与机器之间的鸿沟，即建立用户与计算机之间的接口而为裸机配置的一种系统软件。由图 1 可以看出，操作系统是裸机上的第一层软件，是对硬件系统功能的首次扩充。它在计算机系统中占据重要而特殊的地位，所有其他软件，如编辑程序、汇编程序、编译程序和数据库管理系统等系统软件，以及大量的应用软件都是建立在操作系统基础上的，并得到它的支持和取得它的服务。从用户角度看，当计算机配置了操作系统后，用户不再直接使用计算机系统硬件，而是利用操作系统所提供的命令和服务去操纵计算机，操作系统已成为现代计算机系统中必不可少的最重要的系统软件，因此把操作系统看作是用户与计算机之间的接口。因此，操作系统紧贴系统硬件之上，所有其他软件之下（是所有其他软件的共同环境）。

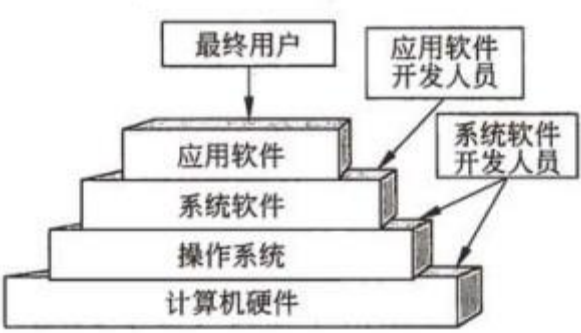
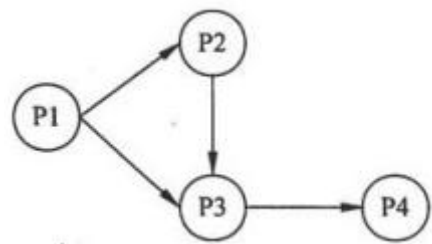
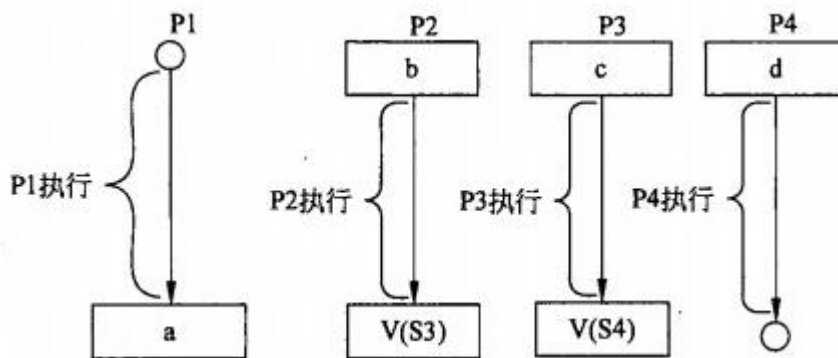


图 1 操作系统在计算机系统中的地位

进程 P1、P2、P3 和 P4 的前趋图如下：



若用 PV 操作控制这几个进程并发执行的过程，则需要设置 4 个信号量 S1、S2、S3 和 S4，且信号量初值都等于零。下图中 a 和 b 应分别填写 (25)，c 和 d 应分别填写 (26)。



- (25) A. P(S1) P(S2) 和 P(S3) B. P(S1) P(S2) 和 V(S1)
 C. V(S1) V(S2) 和 P(S1) D. V(S1) V(S2) 和 V(S3)
 (26) A. P(S1) P(S2) 和 P(S4) B. P(S2) P(S3) 和 P(S4)
 C. V(S1) V(S2) 和 V(S4) D. V(S2) V(S3) 和 V(S4)

【答案】C B

【解析】本题考查操作系统进程管理中PV操作方面的基本知识。

根据题意，进程P2、P3等待P1的结果，因此当P1执行完毕需要使用V操作通知P2、P3，即a处填V(S1) V(S2)，b处应填P(S1)。

根据题意，进程P3要执行需要测试P1、P2有没有消息，故应该在c处填P(S2)、P(S3)。当P3执行完毕需要使用V操作通知P4，即在d处填P(S4)。

若系统正在将(27)文件修改的结果写回磁盘时系统发生崩溃，则对系统的影响相对较大。

- (27) A. 空闲块 B. 目录 C. 用户数据 D. 用户程序

【答案】B

【解析】本题考查操作系统文件管理可靠性方面的基础知识。

影响文件系统可靠性因素之一是文件系统的一致性问题。很多文件系统是先读取磁盘块到主存，在主存进行修改，修改完毕再写回磁盘。例如读取某磁盘块，修改后再将信息写回磁盘前系统崩溃，则文件系统就可能会出现不一致性状态。如果这些未被写回的磁盘块是索引节点块、目录块或空闲块，特别是系统目录文件，那么对系统的影响相对较大，且后果也是不堪设想的。通常解决方案是采用文件系统的一致性检查，一致性检查包括块的一致性和文件的一致性检查。

软件能力成熟度模型（CMM）的第 4 级（已管理级）的核心是_(29)。

- (29)A. 建立基本的项目管理和实践来跟踪项目费用、进度和功能特性
B. 组织具有标准软件过程
C. 对软件过程和产品都有定量的理解和控制
D. 先进的新思想和新技术促进过程不断改进

【答案】C

【解析】本题考查成熟度等级（CMM）知识。在 CMM 的不同等级有不同的核心。

在可重复级，建立了基本的项目管理过程和实践来跟踪项目费用、进度和功能特性。在已定义级，所有项目都采用根据实际情况修改后得到的标准软件过程来开发和维护软件。在已管理级，收集对软件过程和产品质量的详细度量，对软件过程和产品都有定量的理解与控制。在优化级，过程的量化反馈和先进的新思想、新技术促使过程不断改进。

软件系统设计的主要目的是为系统制定蓝图，_(30)并不是软件设计模型所关注的。

- (30)A. 系统总体结构 B. 数据结构 C. 界面模型 D. 项目范围

【答案】D

【解析】本题考查软件系统设计知识。

系统设计为系统制定蓝图，软件设计模型关注新系统总体结构、代码设计、处理过程、数据结构和界面模型等。对项目范围的关注在软件设计之前。

ISO/IEC9126 软件质量模型中，可靠性质量特性包括多个子特性。一软件在故障发生后，要求在 90 秒内恢复其性能和受影响的数据，与达到此目的有关的软件属性为_(31)子特性。

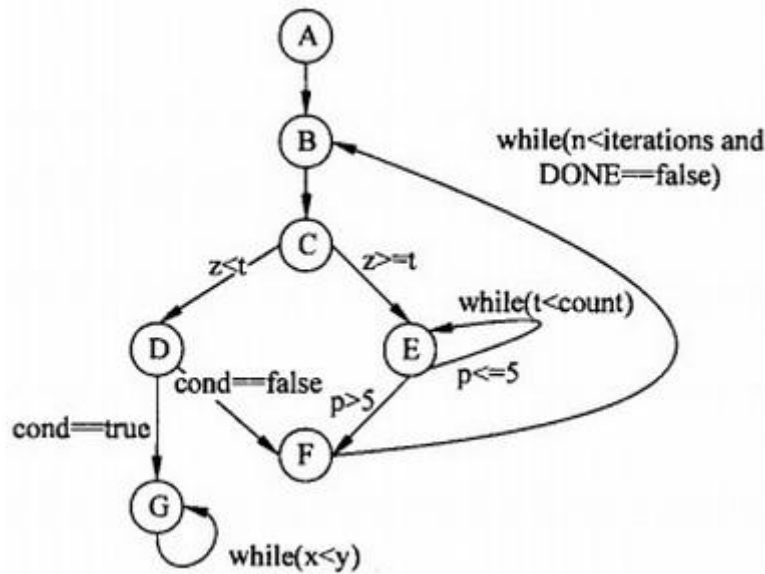
- (31)A. 容错性 B. 成熟性 C. 易恢复性 D. 易操作性

【答案】C

【解析】本题考查软件质量模型知识。

ISO/IEC 9126 软件质量模型中可靠性质量特性是指在规定的一段时间内和规定的条件下，软件维护其性能水平有关的能力。包括的子特性有成熟性、容错性和易恢复性。其中易恢复性是当在故障发生后，重新建立其性能水平并恢复直接受影响数据的能力，以及与为达到此目的所需的时间和有关的工作有关的软件属性。软件故障发生后，要在 90 秒内恢复其性能和受影响的数据，达到这一目的有关的属性即为易恢复性子特性。

某程序的程序图如下所示，运用 McCabe 度量法对其进行度量，其环路复杂度是(32)。



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

【答案】D

【解析】本题考查 McCabe 度量法。

McCabe 度量法是通过定义环路复杂度，建立程序复杂性的度量，它基于一个程序模块的程序图中环路的个数。计算有向图 G 的环路复杂性的公式为 $V(G) = m - n + 2$ ，其中 $V(G)$ 是有向图 G 中的环路个数， m 是 G 中的有向弧数， n 是 G 中的节点数。要采用 McCabe 度量法度量程序复杂度，需要先画出程序图，识别有向图中节点数 $n=7$ 和有向弧数 $m=10$ ，然后利用环路复杂性计算公式 $V(G) = m - n + 2$ 进行计算，得到图中程序图的程序复杂度是 5。

系统开发计划用于系统开发人员与项目管理人员在项目期内进行沟通，它包括(33)和预算分配表等。

- (33) A. PERT 图 B. 总体规划 C. 测试计划 D. 开发合同

【答案】A

【解析】本题考查系统开发计划文档知识。

用于系统开发人员与项目管理人员在项目期内进行沟通的文档主要有系统开发计划，包括工作任务分解表、PERT 图、甘特图和预算分配表等。总体规划和开发合同用于与系统分析人员在系统规划和系统分析阶段的沟通。测试计划用于系统测试人员与系统开发人员之间的沟通。

改正在软件系统开发阶段已经发生而系统测试阶段还没有发现的错误，属于(34)维护。

- (34) A. 正确性 B. 适应性 C. 完善性 D. 预防性

【答案】A

【解析】 本题考查软件维护知识。

软件维护的内容一般包括准确性维护、适应性维护、完善性维护和预防性维护。正确性维护是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。适应性维护是指使应用软件适应信息技术变化和管理需求变化而进行的修改。完善性维护主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。预防性维护是指为了适应未来的软硬件环境的变化，主动增加预防性的新功能，以使应用系统适应各类变化而不被淘汰。

某系统重用了第三方组件（但无法获得其源代码），则应采用_(35)对组件进行测试。

- (35) A. 基本路径覆盖 B. 分支覆盖 C. 环路覆盖 D. 黑盒测试

【答案】 D

【解析】 本题考查软件动态测试方法。

软件测试方法分为静态测试和动态测试。静态测试是被测试程序不在机器上运行而采用人工检测和计算机辅助静态分析手段对程序进行检测。动态测试是通过运行程序发现错误。对软件产品进行动态测试时可以采用黑盒测试和白盒测试法。黑盒测试法在完全不考虑软件的内部结构和特性的情况下，测试软件的外部特性。对于无法获得源代码的第三方组件，可采用黑盒对组件进行测试。基本路径覆盖、分支覆盖和环路覆盖需要根据程序的内部结构和逻辑进行测试。

极限编程（XP）由价值观、原则、实践和行为四个部分组成，其中价值观包括沟通、简单性、_(36)。

- (36) A. 好的计划 B. 不断的发布 C. 反馈和勇气 D. 持续集成

【答案】 C

【解析】 本题考查敏捷开发方法。

极限编程（XP）是敏捷开发的典型方法之一，是一种轻量级（敏捷）、高效、低风险、柔性、可预测的、科学的软件开发方法，它由价值观、原则、实践和行为 4 个部分组成。其中 4 大价值观为沟通、简单性、反馈和勇气。

以下关于类和对象的叙述中，错误的是_(37)。

- (37) A. 类是具有相同属性和服务的一组对象的集合

- B. 类是一个对象模板，用它仅可以产生一个对象
- C. 在客观世界中实际存在的是类的实例，即对象
- D. 类为属于该类的全部对象提供了统一的抽象描述

【答案】B

【解析】本题考查面向对象中类和对象的基本概念。

面向对象中，类是对象的模板，描述了一组对象共有的成员属性和方法。类是一组对象的抽象描述，通过类可以产生一个或多个对象，这些对象具有相同的属性和方法。

(38)是把对象的属性和服务结合成一个独立的系统单元，并尽可能隐藏对象的内部细节；(39)是指子类可以自动拥有父类的全部属性和服务；(40)是对象发出的服务请求，一般包含提供服务的对象标识、服务标识、输入信息和应答信息等。

- | | | | |
|-----------|-------|-------|-------|
| (38)A. 继承 | B. 多态 | C. 消息 | D. 封装 |
| (39)A. 继承 | B. 多态 | C. 消息 | D. 封装 |
| (40)A. 继承 | B. 多态 | C. 消息 | D. 封装 |

【答案】D A C

【解析】本题考查面向对象的基本概念。

封装的作用是把属性和服务结合成一个独立的系统单元，并隐藏对象的内部细节；而继承则使得子类可以自动拥有父类的全部属性和服务；消息则是对象发出的服务请求；多态是指面向对象中由重载或过载机制实现的结果。

以下关于面向对象分析的叙述中，错误的是(41)。

- (41)A. 面向对象分析着重分析问题域和系统责任
- B. 面向对象分析需要考虑系统的测试问题
- C. 面向对象分析忽略与系统实现有关的问题
- D. 面向对象分析建立独立于实现的系统分析模型

【答案】B

【解析】本题考查面向对象分析的基本概念。

面向对象分析主要回答软件系统需要解决什么问题，在面向对象分析阶段，并不考虑系统实现以及系统的测试问题，强调建立独立与系统实现的系统分析模型，其关注点仍然侧重于问题域。

以下关于面向对象设计的叙述中，错误的是(42)。

- (42)A. 高层模块不应该依赖于底层模块 B. 抽象不应该依赖于细节
C. 细节可以依赖于抽象 D. 高层模块无法不依赖于底层模块

【答案】D

【解析】本题考查面向对象设计的基本原则。

面向对象设计中存在很多原则，其中高层模块不应该依赖于底层模块就是其中一条。传统的结构化设计中，一般的高层模块需要调用或使用底层模块，导致高层模块必须依赖于底层模块，但借助面向对象中的抽象类、接口等一系列机制，可以使得高层模块仅仅和接口耦合，而和实际的底层模块没有直接关联。

采用(43)设计模式可保证一个类仅有一个实例；采用(44)设计模式可将对象组合成树形结构以表示“部分-整体”的层次结构，使用户对单个对象和组合对象的使用具有一致性；采用(45)设计模式可动态地给一个对象添加一些额外的职责。

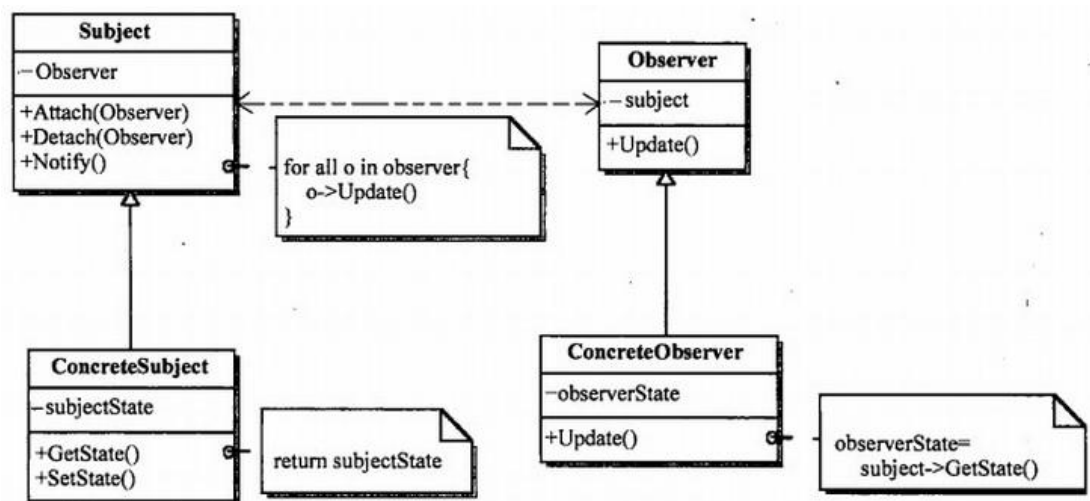
- (43)A. 命令 (Command) B. 单例 (Singleton)
C. 装饰 (Decorate) D. 组合 (Composite)
(44)A. 命令 (Command) B. 单例 (Singleton)
C. 装饰 (Decorate) D. 组合 (Composite)
(45)A. 命令 (Command) B. 单例 (Singleton)
C. 装饰 (Decorate) D. 组合 (Composite)

【答案】B D C

【解析】本题考查面向对象设计中的设计模式。

单例设计模式的目的是保证一个类仅能够生成一个对象；组合设计模式则用于表示“部分-整体”的层次结构，并且对部分和整体的使用具有一致性；装饰设计模式则动态地给一个对象增加一些额外的职责，无须改变类的设计和实现。

下列 UML 类图表示的是(46)设计模式。该设计模式中，(47)。



- (46) A. 备忘录 (Memento) B. 策略 (Strategy)
- C. 状态 (State) D. 观察者 (Observer)
- (47) A. 一个 Subject 对象可对应多个 Observer 对象
- B. Subject 只能有一个 ConcreteSubject 子类
- C. Observer 只能有一个 ConcreteObserver 子类
- D. 一个 Subject 对象必须至少对应一个 Observer 对象

【答案】D A

【解析】本题考查面向对象设计中的设计模式。

题中的图标是观察者设计模式，在该设计模式中，一个 Subject 类可以有多个 ConcreteSubject 子类，一个 Observer 类也可以有多个 ConcreteObserver 子类；一个 Subject 对象可以对应多个 Observer 对象。

以下关于编译系统对某高级语言进行翻译的叙述中，错误的是 (48)。

- (48) A. 词法分析将把源程序看作一个线性字符序列进行分析
- B. 语法分析阶段可以发现程序中所有的语法错误
- C. 语义分析阶段可以发现程序中所有的语义错误
- D. 目标代码生成阶段的工作与目标机器的体系结构相关

【答案】C

【解析】本题考查程序语言翻译基础知识。

词法分析是编译过程的第一阶段，其任务是对源程序从前到后（从左到右）逐个字符地扫描，从中识别出一个个的“单词”符号。语法分析的任务是在词法分析的基础上，根据语

言的语法规则将单词符号序列分解成各类语法单位，如“表达式”、“语句”和“程序”等。语义分析阶段主要检查源程序是否包含语义错误，并收集类型信息供后面的代码生成阶段使用。只有语法和语义都正确的源程序才能被翻译成正确的目标代码。目标代码生成是编译器工作的最后一个阶段。这一阶段的任务是把中间代码变换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码，这个阶段的工作与具体的机器密切相关。

源程序不可避免地会有一些错误，这些错误大致可分为语法错误和语义错误。语法错误是指语言结构上的使用错误，是指编译时所发现的程序错误，如单词拼写错误、标点符号错、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。

若一个程序语言可以提供链表的定义和运算，则其运行时的_(49)。

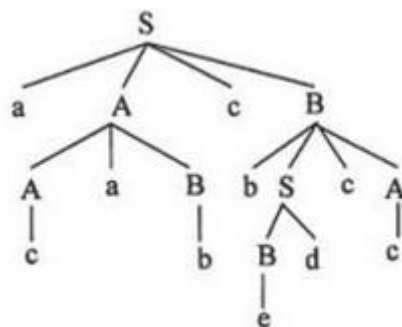
- (49) A. 数据空间适合采用静态存储分配策略 B. 数据空间必须采用堆存储分配策略
C. 指令空间需要采用栈结构 D. 指令代码必须放入堆区

【答案】B

【解析】 本题考查程序语言基础知识。

链表中的结点空间需要程序员根据需要申请和释放，因此，数据空间应采用堆存储分配策略。

由某上下文无关文法 $M[S]$ 推导出某句子的分析树如下图所示，则错误的叙述是_(50)。



- (50) A. 该文法推导出的句子必须以“a”开头
B. $acabcbdcc$ 是该文法推导出的一个句子
C. “ $S \rightarrow aAcB$ ”是该文法的一个产生式
D. a, b, c, d 属于该文法的终结符号集

【答案】A

【解析】

本题考查程序语言基础知识。

推导就是从文法的开始符号 S 出发，反复使用产生式，将产生式左部的非终结符替换为右部的文法符号序列（展开产生式用 \Rightarrow 表示），直到产生一个终结符的序列时为止。从题中给出的分析树可得到如下的一个最左推导过程 $S \Rightarrow aAcB \Rightarrow aAaBcB \Rightarrow acaBcB \Rightarrow acabcB \Rightarrow acabcBScA \Rightarrow acabcBbdcA \Rightarrow acabcBdcA \Rightarrow acabcBdccc$ ，因此， $acabcBdccc$ 是该文法推导出的一个句子。其中用到的产生式如下：

$S \rightarrow aAcB$

$S \rightarrow Bd$

$A \rightarrow AaB$

$A \rightarrow c$

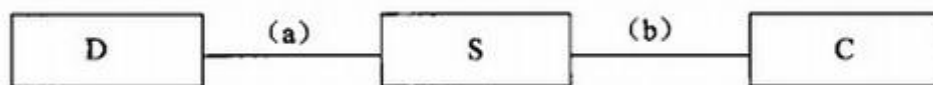
$B \rightarrow bScA$

$B \rightarrow b$

$B \rightarrow \varepsilon$

从起始符号 S 出发也可以如下推导： $S \Rightarrow Bd \Rightarrow \varepsilon d \Rightarrow d$ ，即该文法推导出的句子也可以为 d ，因此选项 A 错误。

假设有学生 S （学号，姓名，性别，入学时间，联系方式），院系 D （院系号，院系名称，电话号码，负责人）和课程 C （课程号，课程名）三个实体，若一名学生属于一个院系，一个院系有多名学生；一名学生可以选择多门课程，一门课程可被多名学生选择，则图中（a）和（b）分别为（51）联系。假设一对多联系不转换为一个独立的关系模式，那么生成的关系模式（52）。



(51) A. 1 * 和 1 *

B. 1 * 和 * 1

C. 1 * 和 * 1

D. * 1 和 * 1

(52) A. S 中应加入关系模式 D 的主键

B. S 中应加入关系模式 C 的主键

C. D 中应加入关系模式 S 的主键

D. C 中应加入关系模式 S 的主键

【答案】C A

【解析】

本题考查的是应试者对数据库系统中 E-R 模型方面的基础知识。

根据题意，学生 S 关系中一个学生属于一个院系，一个院系有多个学生，因此学生 S 和部门院系 D 之间是一个一对多的联系，故（a）应为 1 *；又因为一个学生可以选择多门课程，一门课程可由多名学生选择，所以学生 S 和课程 C 关系之间是一个多对多的联系，故（b）应为 * 1。

假设一对多联系（a）不转换为一个独立的关系模式，那么按照 E-R 模型向关系模型转换的规则，应当将一端的主键并入多端。所以，生成的关系模式学生 S 中应加入关系模式 D 的主键。

软硬件故障常造成数据库中的数据破坏。数据库恢复就是 (53)。

- (53) A. 重新安装数据库管理系统和应用程序
B. 重新安装应用程序，并将数据库做镜像
C. 重新安装数据库管理系统，并将数据库做镜像
D. 在尽可能短的时间内，把数据库恢复到故障发生前的状态

【答案】D

【解析】本题考查关系数据库事务处理方面的基础知识。

为了保证数据库中数据的安全可靠和正确有效，数据库管理系统 (DBMS) 提供数据库恢复、并发控制、数据完整性保护与数据安全性保护等功能。数据库在运行过程中由于软硬件故障可能造成数据被破坏，数据库恢复就是在尽可能短的时间内把数据库恢复到故障发生前的状态。

设有员工实体 Emp (员工号，姓名，性别，年龄，出生年月，联系方式，部门号)，其中“联系方式”要求记录该员工的手机号码和办公室电话，部门号要求参照另一部门实体 Dept 的主码“部门号”。Emp 实体中存在派生属性和多值属性：(54)；对属性部门号应该进行 (55) 约束；可以通过命令 (56) 修改表中的数据。

- (54) A. 年龄和出生年月 B. 年龄和联系方式
C. 出生年月和联系方式 D. 出生年月和年龄
(55) A. 非空主键 B. 主键 C. 外键 D. 候选键
(56) A. INSERT B. DELETE C. UPDATE D. MODIFY

【答案】B C C

【解析】本题考查数据库系统设计的基本概念和 SQL 基本命令。

概念模式设计阶段是根据对用户信息需要的分析设计 E-R 图，对于属性的分析，派生属性是指可以由其他属性经过运算得到的属性，因而派生属性产生冗余，通常不存储，如员工实体 Emp 的年龄；多值属性是指一个实体在该属性上会同时取多个属性值，这些值也都必须存储，如员工实体 Emp 的联系方式。数据约束也是描述用户信息的，根据参照完整性约束的定义，Emp 实体中的部门号属性应为外码，应该用 Foreign Key 进行外键约束。

标准 SQL 对数据库对象（基本表、索引和视图等）的创建、修改和删除定义了一组操作分别为 CREATE、ALTER 和 DROP；对基本表和视图中数据的插入、删除和修改定义了操作分别

为 INSERT、DELETE 和 UPDATE，这两组操作不能混淆。

已知一个二叉树的先序遍历序列为①、②、③、④、⑤，中序遍历序列为②、①、④、③、⑤，则该二叉树的后序遍历序列为(57)。对于任意一棵二叉树，叙述错误的是(58)。

- (57) A. ②、③、①、⑤、④ B. ①、②、③、④、⑤
C. ②、④、⑤、③、① D. ④、⑤、③、②、①

- (58) A. 由其后序遍历序列和中序遍历序列可以构造该二叉树的先序遍历序列
B. 由其先序遍历序列和后序遍历序列可以构造该二叉树的中序遍历序列
C. 由其层序遍历序列和中序遍历序列可以构造该二叉树的先序遍历序列
D. 由其层序遍历序列和后序遍历序列不能构造该二叉树的中序遍历序列

【答案】C B

【解析】 本题考查数据结构基础知识。

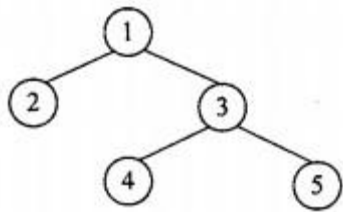
遍历运算是二叉树的基本运算，主要有先序、中序、后序和层序遍历。

先序遍历的基本方法：对于非空二叉树，先访问根结点，然后先序遍历根的左子树，最后先序遍历根的右子树。因此，若已知某二叉树的先序遍历序列，则可直接得到其树根结点。

中序遍历的基本方法：对于非空二叉树，先中序遍历根的左子树，然后访问根结点，最后中序遍历根的右子树。因此，若已知某二叉树的根结点，则可根据中序遍历序列将该二叉树左右子树上的结点划分开。

后序遍历的基本方法：对于非空二叉树，首先后序遍历根的左子树，接着后序遍历根的右子树，最后访问根结点。因此，若已知某二叉树的后序遍历序列，则可直接得到其树根结点。

题中给出的先序遍历序列为①、②、③、④、⑤，可知树根结点是①，据此再结合中序遍历序列②、①、④、③、⑤，可知②是根结点①左子树上的结点，由于是左子树上唯一的一个结点，因此②是根结点①的左孩子。对于右子树上的结点④、③、⑤，因右子树的先序遍历序列为③、④、⑤，因此③是根结点①的右孩子。依此类推，可知④是结点③的左孩子，⑤是结点③的右孩子。该二叉树如下图所示。



从二叉树的遍历过程可知，从先序遍历序列和后序遍历序列中无法将左子树和右子树上的结点区分开，因此，由某棵二叉树的先序遍历序列和后序遍历序列不能构造出该二叉树的中序遍历序列。

层序遍历二叉树的方法：设二叉树的根结点所在层数为 1，则层序遍历二叉树的操作定义为从树的根结点出发，首先访问第一层的结点（根结点），然后从左到右依次访问第二层上的结点，接着是第三层上的结点，依此类推，自上而下、自左至右逐层访问树中各层上的结点。

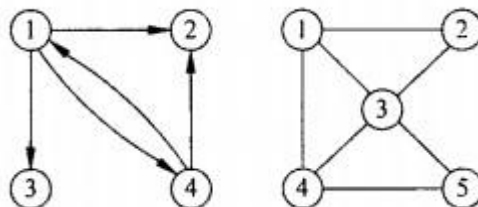
邻接矩阵和邻接表是图（网）的两种基本存储结构，对于具有 n 个顶点、 e 条边的图，
(59)。

- (59) A. 进行深度优先遍历运算所消耗的时间与采用哪一种存储结构无关
 B. 进行广度优先遍历运算所消耗的时间与采用哪一种存储结构无关
 C. 采用邻接表表示图时，查找所有顶点的邻接顶点的时间复杂度为 $O(n \cdot e)$
 D. 采用邻接矩阵表示图时，查找所有顶点的邻接顶点的时间复杂度为 $O(n^2)$

【答案】D

【解析】 本题考查数据结构基础知识。

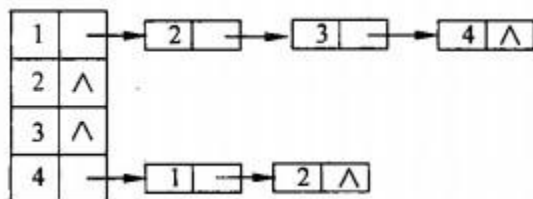
设某有向图和无向图如下所示。



下面的矩阵 A 是该有向图的邻接矩阵，B 为无向图的邻接矩阵。

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

上面有向图的邻接链表如下图所示。



图的遍历运算是按照某种策略访问图中的每一个顶点,实质上是通过边或弧找邻接点的过程,因此广度优先搜索遍历图和深度优先搜索遍历图的时间复杂度相同,其不同之处仅仅在于对顶点访问的次序不同。

单向链表中往往含有一个头结点,该结点不存储数据元素,一般令链表的头指针指向该结点,而该结点指针域的值域为第一个元素结点的指针。以下关于单链表头结点的叙述中,错误的是 (60)。

- (60) A. 若在头结点中存入链表长度值,则求链表长度运算的时间复杂度为 $O(1)$
- B. 在链表的任何一个元素前后进行插入和删除操作可用一致的方式进行处理
- C. 加入头结点后,代表链表的头指针不因为链表为空而改变
- D. 加入头结点后,在链表中进行查找运算的时间复杂度为 $O(1)$

【答案】D

【解析】 本题考查数据结构基础知识。

含有头结点的单链表如下图所示。



在链表中加入头结点后,查找表中某一元素仍然要从头指针出发,顺序找到目标元素或失败时找到表尾为止,时间复杂度与表长成正比。

对于长度为 m ($m>1$) 的指定序列, 通过初始为空的一个栈、一个队列后, 错误的叙述是 (61)。

- (61) A. 若入栈和入队的序列相同, 则出栈序列和出队序列可能相同
B. 若入栈和入队的序列相同, 则出栈序列和出队序列可以互为逆序
C. 入队序列与出队序列关系为 1:1, 而入栈序列与出栈序列关系是 1:n ($n\geq 1$)
D. 入栈序列与出栈序列关系为 1:1, 而入队序列与出队序列关系是 1:n ($n\geq 1$)

【答案】D

【解析】本题考查数据结构基础知识。

队列的元素按特点是先进先出。对于队列, 元素的进入次序和出队的次序相同, 例如, 入队的序列为 a、b、c, 则出队的序列也为 a、b、c。对于栈则不同, 栈的运算特点是后进先出。若入栈序列为 a、b、c, 则出栈序列可能为 a、b、c, a、c、b, b、a、c, b、c、a 或者 c、b、a, 而 c、a、b 则不行, 因此, 入栈序列与出栈序列关系为 1:1, 而入队序列与出队序列关系为 1:n ($n\geq 1$)。

字符串采用链表存储方式时, 每个结点存储多个字符有助于提高存储密度。若采用结点大小相同的链表存储串, 在串比较、求子串、串连接、串替换等串的基本运算中, (62)。

- (62) A. 进行串的比较运算最不方便 B. 进行求子串运算最不方便
C. 进行串连接最不方便 D. 进行串替换最不方便

【答案】D

【解析】本题考查数据结构基础知识。

在串比较、求子串、串连接、串替换运算中, 除了串替换外, 其他运算都不会改变串中的内容, 因此, 在链表存储方式下进行串替换最不方便。

某算法的时间复杂度表达式为 $T(n)=an^2+bn\lg n+cn+d$, 其中, n 为问题的规模, a 、 b 、 c 和 d 为常数, 用 O 表示其渐近时间复杂度为 (63)。

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

【答案】A

【解析】本题考查算法分析的基础知识。

在算法分析中, 符号 O 用于表示算法运行时间的上限。从定义上说, 对一个函数 $g(n)$, $O(g(n))$ 表示函数集合:

$\{f(n): \text{存在正常数 } c \text{ 和 } n_0, \text{使得对所有的 } n \geq n_0, \text{有 } 0 \leq f(n) \leq c g(n)\}$

根据上述定义,可以知道表达式 $T(n)=an^2+bn\lg n+cn+d$ 在函数集合 $O(n^2)$ 中。对此问题,简单的做法是忽略 n 的低阶项和最高阶项的常系数,故答案应为 $O(n^2)$ 。

以下关于快速排序算法的描述中,错误的是 (64)。在快速排序过程中,需要设立基准元素并划分序列来进行排序。若序列由元素 $\{12, 25, 30, 45, 52, 67, 85\}$ 构成,则初始排列为 (65) 时,排序效率最高 (令序列的第一个元素为基准元素)。

(64) A. 快速排序算法是不稳定的排序算法

B. 快速排序算法在最坏情况下的时间复杂度为 $O(n\lg n)$

C. 快速排序算法是一种分治算法

D. 当输入数据基本有序时,快速排序算法具有最坏情况下的时间复杂度

(65) A. 45, 12, 30, 25, 67, 52, 85

B. 85, 67, 52, 45, 30, 25, 12

C. 12, 25, 30, 45, 52, 67, 85

D. 45, 12, 25, 30, 85, 67, 52

【答案】B A

【解析】

本题考查快速排序算法。

快速排序算法是一种经典的排序算法,其基本思想是选择一个基准元素(通常选择第一个元素或者最后一个元素),通过一趟排序将待排序序列分成两部分,一部分比基准元素小,一部分大于等于基准元素,此时基准元素在其排好序后的正确位置;然后再递归地排序划分的两部分,因此本质上快速排序是一种分治算法。由于在排序的过程中,各元素与基准元素比较大小,若小于基准元素则与基准元素交换位置,因此该算法是不稳定的排序算法。当每一趟排序进行后,选择的基准元素恰好最大或者最小时,就把序列分成极端不均衡的两部分,即一部分为空,另一部分为待排序序列的元素个数减 1,此时算法处于最坏情况,其时间复杂度为 $O(n^2)$ 。当输入数据基本有序或者所有元素值相等时,不论选择第一个元素还是最后一个元素作为基准元素,都恰好把序列分成极端不均衡的两部分,快速排序算法具有最坏情况下的时间复杂度。

对于选项 A,以 45 作为基准元素进行第一趟划分,先从后向前找出比 45 小的元素,67、52、85 这三个元素保持不动,找到 25,将其与 45 交换后,第一趟划分完成,序列为 25,12,30,45,67,52,85。第二趟先对子序列 25,12,30 进行划分,使得 25 与 12 对调,形成子序列 12,25,30;然后对 67,52,85 进行划分,使得 67 与 52 交换,形成子序列 52,67,85。至此,整个排序过程完成。期间,第一趟划分中元素的比较次数为 6 次、交换 1 次,第二趟划分中元素的比较次数共 4 次、交换次数为 2 次,因此,排序过程中比较次数共 10 次,交换次数为 3 次。

下列网络互连设备中,属于物理层的是 (66),属于网络层的是 (67)。

- (66) A. 中继器 B. 交换机 C. 路由器 D. 网桥
- (67) A. 中继器 B. 交换机 C. 路由器 D. 网桥

【答案】A C

【解析】

中继器是物理层设备，其作用是对接收的信号进行再生放大，以延长传输的距离。网桥是数据链路层设备，可以识别 MAC 地址，进行帧转发。交换机是由硬件构成的多端口网桥，也是一种数据链路层设备。路由器是网络层设备，可以识别 IP 地址，进行数据包的转发。

下图是 HTML 文件 test.html 在 IE 中的显示效果，实现图中①处效果的 HTML 语句是 (68)，实现图中②处效果的 HTML 语句是 (69)，实现图中③处效果的 HTML 语句是 (70)。



- (68) A. <TITLE>我的主页</TITLE> B. <HEAD>我的主页</HEAD>
- C. <BODY>我的主页</BODY> D. <H1>我的主页</H1>
- (69) A. <HR> B. <LINE></LINE> C. <CELL></CELL> D. <TR></TR>
- (70) A. Welcome B. Welcome
- C. <I>Welcome</I> D. <H>Welcome</H>

【答案】A A C

【解析】本题考查 HTML 标记相关知识。

①处显示的信息为文档的标题，采用<TITLE></TITLE>标记。②处为一条分隔线，采用<HR>标记。③处字体 Welcome 是斜体，采用<I></I>标记。

Why is (71) fun? What delights may its practitioner expect as his reward? First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. Second is the pleasure of making things that are useful to other people. Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning. Fourth is the joy of always learning, which springs from the (72) nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes (73) , sometimes theoretical, and

sometimes both. Finally, there is the delight of working in such a tractable medium. The (74) , like the poet, works only slightly removed from pure thought-stuff. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures.

Yet the program (75) , unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

- | | | | |
|--------------------|--------------|------------------|----------------|
| (71)A. programming | B. composing | C. working | D. writing |
| (72)A. repeating | B. basic | C. non-repeating | D. advance |
| (73)A. semantic | B. practical | C. lexical | D. syntactical |
| (74)A. poet | B. architect | C. doctor | D. programmer |
| (75)A. construct | B. code | C. size | D. scale |

【答案】A C B D A

【解析】

编程为什么有趣？作为回报，其从业者期望得到什么样的快乐？首先是一种创建事物的纯粹快乐。如同小孩在玩泥巴时感到愉快一样，成年人喜欢创建事物，特别是自己进行设计。其次，快乐来自于开发对其他人有用的东西。第三是整个过程体现出魔术般的力量——将相互啮合的零部件组装在一起，看到它们精妙地运行，得到预先所希望的结果。第四是学习的乐趣，来自于这项工作的非重复特性。人们所面临的问题，在某个或其他方面总有些不同，

因而解决问题的人可以从中学习新的事物：有时是实践上的，有时是理论上的，或者兼而有之。最后，乐趣还来自于工作在如此易于驾驭的介质上。程序员就像诗人一样，几乎仅仅工作在单纯的思考中，凭空地运用自己的想象来建造自己的“城堡”。很少有这样的介质——创造的方式如此灵活，如此易于精炼和重建，如此容易地实现概念上的设想。

然而程序毕竟同诗歌不同，它是实实在在的东西；可以移动和运行，能独立产生可见的输出；能打印结果，绘制图形，发出声音，移动支架。编程非常有趣，在于它不仅满足了我们内心深处进行创造的渴望，而且还愉悦了每个人内在的情感。

试题一

现准备为某银行开发一个信用卡管理系统 CCMS, 该系统的基本功能为:

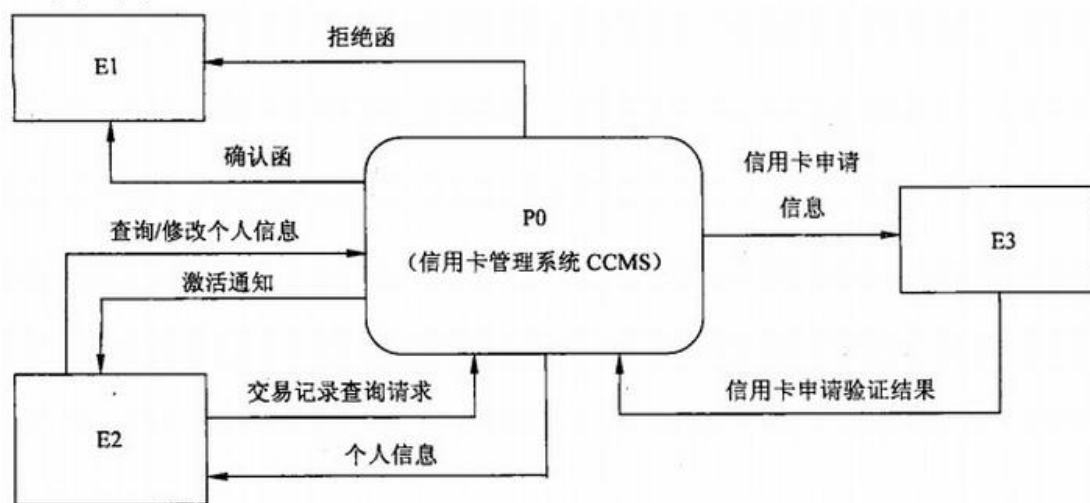
1. 信用卡申请。非信用卡客户填写信用卡申请表, 说明所要申请的信用卡类型及申请者的基本信息, 提交 CCMS。如果信用卡申请被银行接受, CCMS 将记录该客户的基本信息, 并发送确认函给该客户, 告知客户信用卡的有效期限及信贷限额; 否则该客户 将会收到一封拒绝函。非信用卡客户收到确认函后成为信用卡客户。

2. 信用卡激活。信用卡客户向 CCMS 提交激活请求, 用信用卡号和密码激活该信用卡。激活操作结束后, CCMS 将激活通知发送给客户, 告知客户其信用卡是否被成功激活。

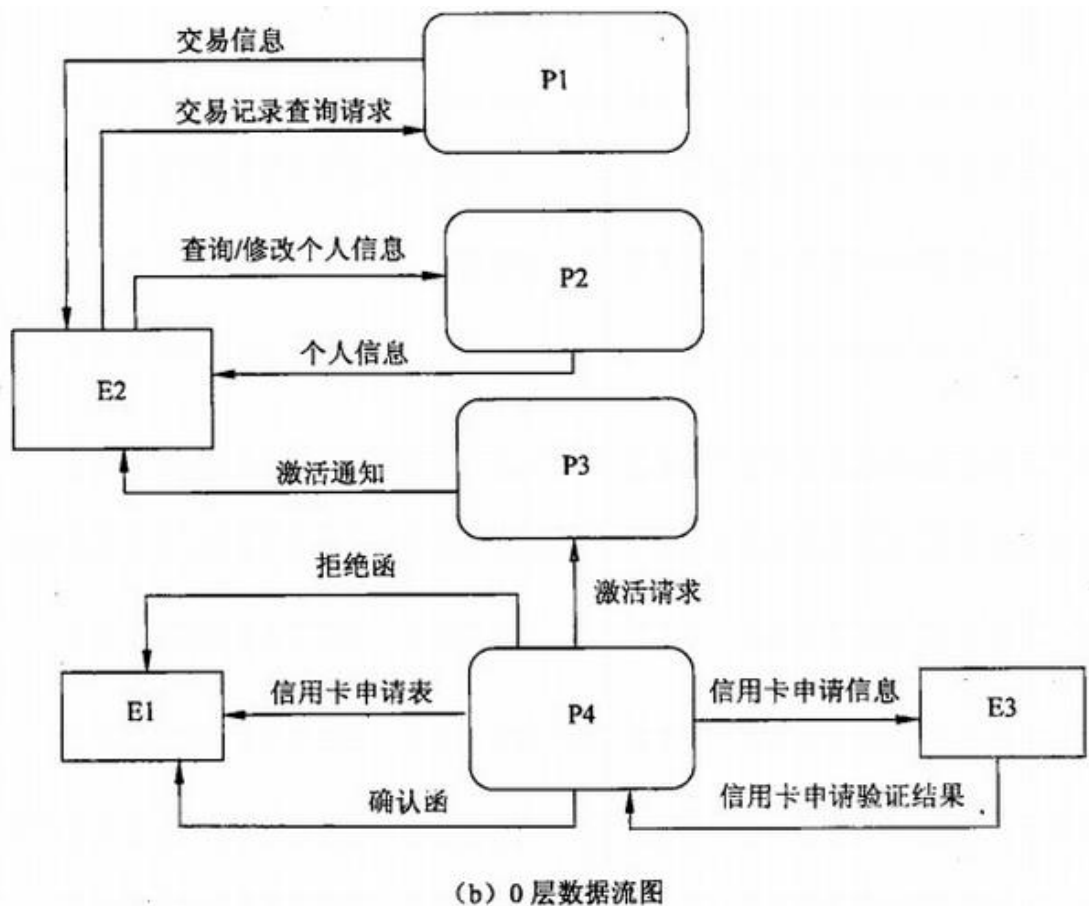
3. 信用卡客户信息管理。信用卡客户的个人信息可以在 CCMS 中进行在线管理。每位信用卡客户可以在线查询和修改个人信息。

4. 交易信息查询。信用卡客户使用信用卡进行的每一笔交易都会记录在 CCMS 中。信用卡客户可以通过 CCMS 查询并核实其交易信息 (包括信用卡交易记录及交易额)。

下图 (a) 和 (b) 分别给出了该系统的顶层数据流图和 0 层数据流图的初稿。



(a) 顶层数据流图



【问题 1】

根据说明，将图 (a) 中的 E1~E3 填充完整。

根据题目中的说明，可以很容易找到与 CCMS 系统进行信息交互的角色有非信用卡客户、信用卡客户以及银行。下面要做的事情是在上图 (a) 中找到对应的位置。

根据图 (a) 给出的输入和输出数据流，可知 E1 表示非信用卡客户；E2 表示信用卡客；E3 表示银行。

【问题 2】

图 (a) 中缺少三条数据流，根据说明，分别指出这三条数据流的起点和终点。(注：数据流的起点和终点均采用图中的符号和描述)

起 点	终 点	名 称
E1	P0	信用卡申请表
E2	P0	激活请求
P0 或 信用卡管理系统 CCMS	E2	信用卡交易信息

注：每条数据流的起点和终点全部答对方可给 1 分

这道题目主要考查父图与子图的平衡问题。对照上图 (a) 和 (b) 可以发现，数据流“信

用卡申请表”、“激活请求”、“信用卡交易信息”出现在图（b）中，却没有出现在图（a）中。下一步只要正确地标出这三条数据流的起点和终点就可以了。

【问题 3】

图（b）中有两条数据流是错误的，请指出这两条数据流的名称，并改正。（注：数据流的起点和终点均采用图中的符号和描述）

数据流名称	改正后数据流起点	改正后数据流终点
激活请求	E2	P3
信用卡申请表	E1	P4

数据流的错误主要有与错误的加工相连接、没有经过任何的加工、数据流方向错误等。在图（b）中，并没有出现任何的数据流没有经过加工，那错误就在于与数据流相连接的加工有问题或者数据流方向错误。

这样，可以找两条有错误的数据流“激活请求”和“信用卡申请表”。从图（a）中可知，“激活请求”是从系统流向外部实体 E2 的，而在图（b）中，“激活请求”却出现在两个加工之间。数据流“信用卡申请表”是在问题 2 中补充找到的数辑滩，它应该从外部实体 E1 流向 CCMS 系统。

【问题 4】

根据说明，将图（b）中 P1~P4 的处理名称填充完整。

这道题要求将图（b）中的加工补充完整。加工的名称在说明中 B 经明确给出了：信用卡申请、信用卡激活、信用卡客户信息管理以及交易信息查询。下一步需要根据图（b）中给出的数据流关系将这 4 个加工对号入座即可。这样可以得到 P1 表示交易信息查询；P2 表示信用卡客户信息管理；P3 表示信用卡激活；P4 表示信用卡申请。

试题二

某公司拟开发一多用户电子邮件客户端系统，部分功能的初步需求分析结果如下：

(1) 邮件客户端系统支持多个用户，用户信息主要包括用户名和用户密码，且系统中的用户名不可重复。

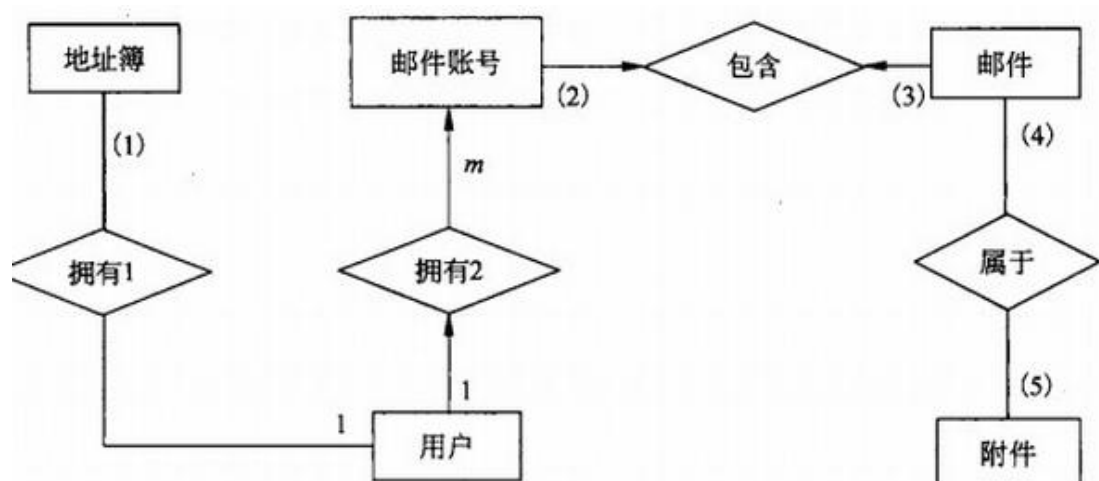
(2) 邮件账号信息包括邮件地址及其相应的密码，一个用户可以拥有多个邮件地址（如 user1@123.com）。

(3) 一个用户可拥有一个地址簿，地址簿信息包括联系人编号、姓名、电话、单位地址、邮件地址 1、邮件地址 2、邮件地址 3 等信息。地址簿中一个联系人只能属于一个用户，且联系人编号唯一标识一个联系人。

(4) 一个邮件账号可以含有多封邮件，一封邮件可以含有多个附件。邮件主要包括邮件号、发件人地址、收件人地址、邮件状态、邮件主题、邮件内容、发送时间、接收时间。其中，邮件号在整个系统内唯一标识一封邮件，邮件状态有已接收、待发送、已发送和已删除 4 种，分别表示邮件是属于收件箱、发件箱、已发送箱和废件箱。一封邮件可以发送给多个用户。附件信息主要包括附件号、附件文件名、附件大小。一个附件只属于一封邮件，附件号仅在一封邮件内唯一。

【问题 1】

根据以上说明设计的 E-R 图如下图所示，请指出地址簿与用户、电子邮件账号与邮件、邮件与附件之间的联系类型。

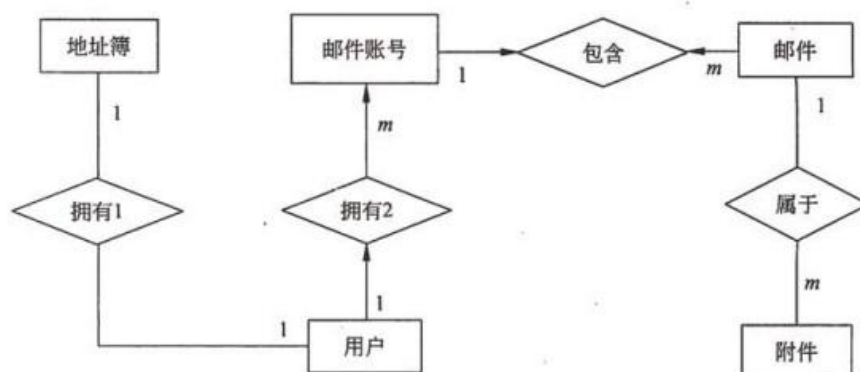


电子邮件客户端系统 E-R 图

- (1) 1
 (2) 1
 (3) m 或 n 或 $*$
 (4) 1
 (5) m 或 n 或 $*$

两个实体模型之间的联系可以分为三类：一对一联系（1:1）、一对多联系（1: n ）和多对多联系（ $m:n$ ）。

根据题意，地址簿与用户之间应该是一个 1:1 的联系，空（1）应填 1。电子邮件账号与邮件之间应该是一个 1: m 的联系，故空（2）和空（3）应分别填写 1 和 m 。邮件与附件之间应该是一个 1: m 的联系，故空（4）和空（5）应分别填写 1 和 m 。得到的 E-R 图如下图所示。



【问题 2】

该邮件客户端系统的主要关系模式如下，请填补（a）~（c）的空缺部分。

用户（用户名，用户密码）

地址簿（（a），联系人编号，姓名，电话，单位地址，邮件地址 1，邮件地址 2，邮件地址 3）

邮件账号（邮件地址，邮件密码，用户名）

邮件（（b），收件人地址，邮件状态，邮件主题，邮件内容，发送时间，接收时间）

附件（（c），附件号，附件文件名，附件大小）

(a) 用户名

(b) 邮件号，发件人地址

(c) 邮件号

空（a）分析：根据题意可知邮件客户端系统支持多个用户，用户信息主要包括用户名和用户密码，且系统中的用户名不可重复，“用户名”可以作为用户关系模式主键。地址簿关系模

式中与用户关系模式是一个 1 : 1 的联系，必须将任一方的主键加入另一方， 以建立它们之间的联系，故空 (a)处应填写“用户名”。

空 (b)分析：根据题意可知邮件号在整个系统内唯一标识一封邮件，故邮件关系模式必须有属性“邮件号”，另外一封邮件需要填写“发件人地址”，故空 (b)处应填写“邮件号，发件人地址”。

空 (c)分析：根据题意可知邮件和附件是一个 1 :m 的联系，按照 E-R 模型向关系模型的转换规则对于 1 : m 的联系应将 1 端的主键并入多端，故空 (c)处应填写“邮件号”。

【问题 3】

(1) 请指出问题 2 中给出的地址簿、邮件和附件关系模式的主键，如果关系模式存在外键请指出。

(2) 附件属于弱实体吗？请用 50 字以内的文字说明原因。

(1)

关 系 模 式	主 键	外 键
地址簿	联系人编号	用户名
邮件	邮件号	发件人地址或收件人地址
附件	邮件号，附件号	邮件号

(2) 附件属于弱实体，因为附件的存在必须以邮件的存在为前提，即附件总是依附于某邮件。

(1)地址簿关系模式的主键为“联系人编号”，外键为“用户名”，因为“用户名” 是参考用户关系模式的“用户名”主键。邮件关系模式的主键为“邮件号”，外键为“发件人地址”或“收件人地址”，因为当用户向其他人发邮件的时候，“发件人地址”是参考邮件账号关系模式的“邮件地址”的主键；当用户收邮件的时候，“收件人地址”是参考邮件账号关系模式的“邮件地址”的主键。附件关系模式的主键为“邮件号，附件号”，外键为“邮件号”，因为该“邮件号”参考邮件关系模式的“邮件号”的主键。

(2) 附件属于弱实体，因为如果没有邮件，附件也就不存在。

试题三

某企业为了方便员工用餐，为餐厅开发了一个订餐系统（COS: Cafeteria Ordering System），企业员工可通过企业内联网使用该系统。

企业的任何员工都可以查看菜单和今日特价。

系统的顾客是注册到系统的员工，可以订餐（如果未登录，需先登录）、注册工资支付、预约规律的订餐，在特殊情况下可以覆盖预订。

餐厅员工是特殊顾客，可以进行备餐、生成付费请求和请求送餐，其中对于注册工资支付的顾客生成付费请求并发送给工资系统。

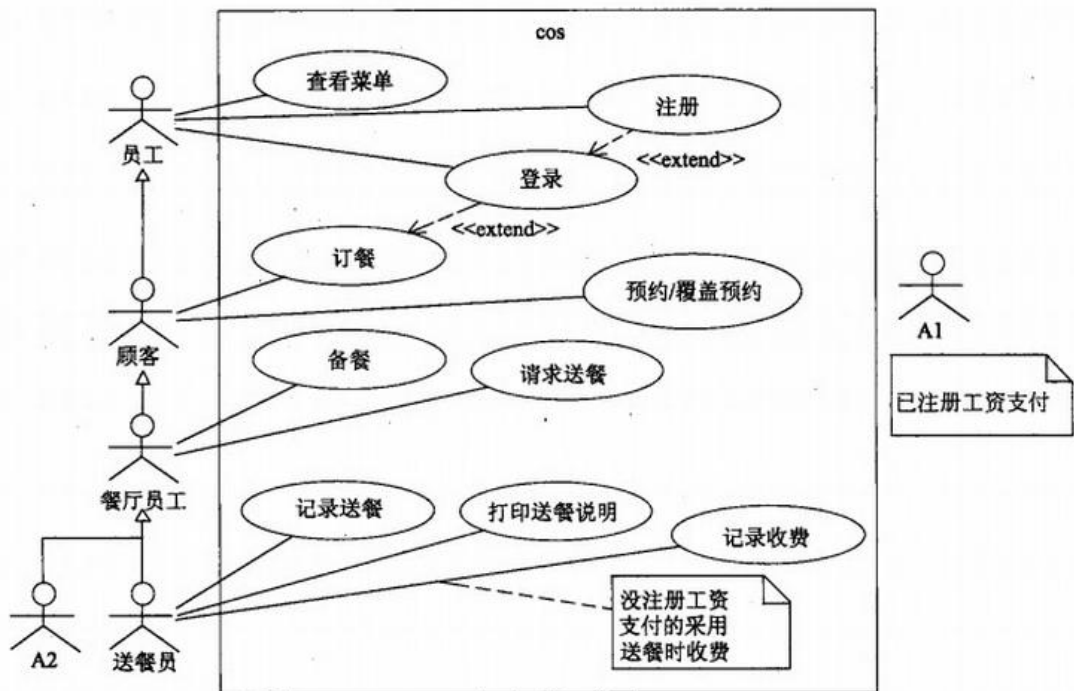
菜单管理员是餐厅特定员工，可以管理菜单。

送餐员可以打印送餐说明，记录送餐信息（如送餐时间）以及记录收费（对于没有注册工资支付的顾客，由送餐员收取现金后记录）。

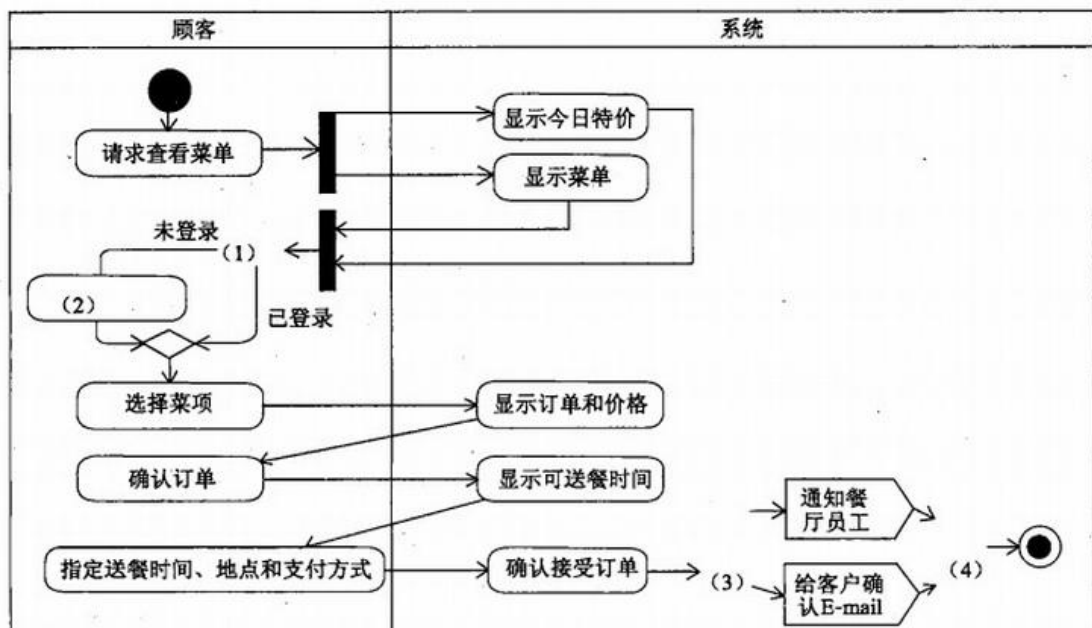
顾客订餐过程如下：

1. 顾客请求查看菜单；
2. 系统显示菜单和今日特价；
3. 顾客选菜；
4. 系统显示订单和价格；
5. 顾客确认订单；
6. 系统显示可送餐时间；
7. 顾客指定送餐时间、地点和支付方式；
8. 系统确认接受订单，然后发送 E-mail 给顾客以确认订餐，同时发送相关订餐信息通知给餐厅员工。

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一次订餐的活动图初稿分别如下图（a）和（b）所示。



(a) COS 系统顶层用例图



(b) 一次订餐的活动图

【问题 1】

根据说明中的描述，给出图 (a) 中 A1 和 A2 所对应的参与者。

识别参与者时，考查和系统交互的人员和外部系统。本题中，与系统交互的人员包括员工、连册到系统的员工（顾客）、餐厅员工、菜单管理员、送餐员以及工资系统。由“菜单管理员是餐厅特定员工”以及图中 A2 和图中餐厅员工之间的“是一种”关系可知，A2 为菜单管理员；图中还缺少描述中与工资系统的交互，由“……并发送给工资系统”可

知，A1 为工资系统。

【问题 2】

根据说明中的描述，给出图 (a)中缺少的四个用例及其所对应的参与者。

用 例 名	参 与 者
查看当日特价	员工
注册工资支付	顾客和工资系统（或顾客和 A1）
生成付费请求	餐厅员工和工资系统（或餐厅员工和 A1）
管理菜单	菜单管理员（或 A2）

（注：4 行的顺序可以不同，但是每行必须对应）

考查用例及其和参与者之间的关系时，通过判断哪一个特定参与者发起或者触发了与系统的哪些交互，来识别用例并建立和参与者之间的关联。

本题中，由“任何员工都可以查看菜单和今日特价”可知，图中缺少用例查看今日特价，对应参与者是员工；由“系统的顾客是……，注册工资支付、……”可知，图中缺少用例注册工资支付，对应参与者是顾客和工资系统；由“餐厅员工是……，可以进行备餐、生成付费请求……发送给工资系统”可知，图中缺少用例“生成付费请求”，对应的参与者是餐厅员工和工资系统；由“菜单管理员是餐厅特定员工，可以管理菜单”可知，图中缺少用例管理菜单，对应的参与者是菜单管理员。

需要注意的是，在注册工资支付所对应的参与者中，虽然没有明确说明要和工资系统交互，但是由“对于注册工资支付的顾客生成付费请求并发送给工资系统”可知，工资支付是由工资系统控制，所以注册也需要和工资系统交互。

【问题 3】

根据说明中的描述，给出图 (b) 中 (1)～(4)处对应的活动名称或图形符号。

在顾客订餐过程的描述中，在“顾客选菜”之前，图中缺少符号和活动。由说明中顾客“可以订餐（如果未登录，需先登录）”可以判断，在系统“显示菜单和今日特价”之后“顾客选菜”之前，需要判断（判定符号◇）当前用户身份是否为顾客，如果不是，需先登录；由“……发送 E-mail 给顾客以确认订餐，同时发送相关订餐信息通知给餐厅员工”可知，发送 E-mail 和通知餐厅员工为并行活动，需要在前后有同步条（或纵向）。

【问题 4】

指出图 (a)中员工和顾客之间是什么关系，并解释该关系的内涵。

泛化关系（一般/特殊关系、继承关系）。泛化关系描述了一个参与者可以完成另一个参与者

同样的任务，并可补充额外的角色功能。

参与者之间的关系表示子类型“是一种”父类型，即泛化关系。其中父类型通常是一个抽象泛化的参与者，可以完成子类型可完成的共同行为，每个具体的子类型继承它，可以完成父类型参与者同样的任务，并可以补充额外的角色功能。

试题四

0-1 背包问题可以描述为：有 n 个物品，对 $i = 1, 2, \dots, n$ ，第 i 个物品价值为 v_i ，重量为 w_i (v_i 和 w_i 为非负数)，背包容量为 W (W 为非负数)，选择其中一些物品装入背包，使装入背包物品的总价值最大，即 $\max \sum_{i=1}^n v_i x_i$ ，且总重量不超过背包容量，即 $\sum_{i=1}^n w_i x_i \leq W$ ，其中， $x_i \in \{0, 1\}$ ， $x_i=0$ 表示第 i 个物品不放入背包， $x_i=1$ 表示第 i 个物品放入背包。

【问题 1】

用回溯法求解此 0-1 背包问题，请填充下面伪代码中 (1)~(4) 处空缺。

回溯法是一种系统的搜索方法。在确定解空间后，回溯法从根结点开始，按照深度优先策略遍历解空间树，搜索满足约束条件的解。对每一个当前结点，若扩展该结点已经不满足约束条件，则不再继续扩展。为了进一步提高算法的搜索效率，往往需要设计一个限界函数，判断并剪枝那些即使扩展了也不能得到最优解的结点。现在假设已经设计了 $\text{BOUND}(v, w, k, W)$ 的函数，其中 v 、 w 、 k 和 W 分别表示当前已经获得的价值、当前背包的重量、已经确定是否选择的物品数和背包的总容量。对应于搜索树中的某个结点，该函数值表示确定了部分物品是否选择之后，对剩下的物品在满足约束条件的前提下进行选择可能获得的最大价值，若该价值小于等于当前已经得到的最优解，则该结点无需再扩展。

下面给出 0-1 背包问题的回溯算法伪代码。

函数参数说明如下：

W ：背包容量； n ：物品个数； w ：重量数组； v ：价值数组； fw ：获得最大价值时背包的重量； fp ：背包获得的最大价值； X ：问题的最优解。

变量说明如下：

CW ：当前的背包重量； cp ：当前获#的价值； k ：当前考虑的物品编号； Y ：当前已获得的部分解。

```

BKNAP( W,n,w,v,fw,fp,X )
1  cw ← cp ← 0
2  (1)
3  fp ← -1
4  while true
5      while k ≤ n and cw + w[k] ≤ W do
6          (2)
7          cp ← cp + v[k]
8          Y[k] ← 1
9          k ← k + 1
10     if k > n then
11         if fp < cp then
12             fp ← cp
13             fw ← cw
14             k ← n
15             X ← Y
16     else Y(k) ← 0
17     while BOUND(cp,cw,k,W) ≤ fp do
18         while k ≠ 0 and Y(k) ≠ 1 do
19             (3)
20             if k = 0 then return
21             Y[k] ← 0
22             cw ← cw - w[k]
23             cp ← cp - v[k]
24     (4)

```

- (1) $k \leftarrow 1$ 或其等价形式 (2) $cw \leftarrow cw + w[k]$ 或其等价形式
 (3) $k \leftarrow k-1$ 或其等价形式 (4) $k \leftarrow k+1$ 或其等价形式

本题考查的是用回溯法求解 0-1 背包问题。回溯法有两类算法框架：非递归形式和递归形式，本题采用非递归形式表示。理解回溯法的基本思想和这两类算法框架是正确解答本题的根本要求。

回溯法从第一项物品开始考虑是否应该装入背包中，因此当前考虑的物品编号 k 从 1 开始，即 $k \leftarrow 1$ 。然后逐项往后检查，若能全部放入背包则将该项放入背包，此时背包的重量应该是当前的重量加上当前考虑物品的重量，即 $cw \leftarrow cw + w[k]$ ，当然背包中物品的价值也为当前的价值加上当前考虑物品的价值。若已经考虑完了所有的物品，则得到一个解，判断该解是否为当前最优，若为最优，则将该解的信息放入变量 fp 、 fw 和 X 中。若还没有考虑完所有的物品，意味着有些物品不能放入背包，此时先判断若不将当前的物品放入背包中，则其余物品放入背包是否可能得到比当前最优解更优的解，若得不到则回溯；否则继续考虑其余的物品。

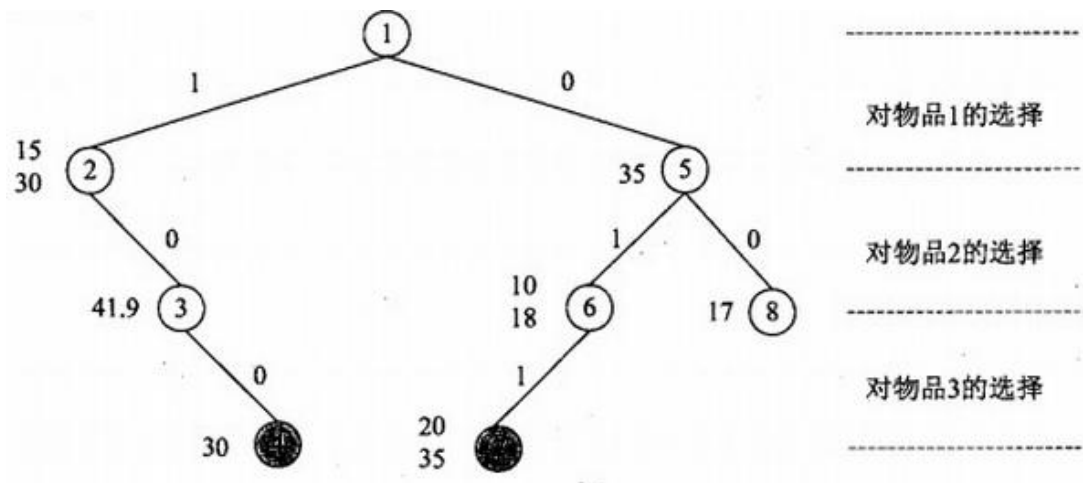
【问题 2】

考虑下表所示的实例，假设有 3 个物品，背包容量为 22。

0-1 背包问题实例

	物品 1	物品 2	物品 3
重量	15	10	10
价值	30	18	17
单位价值	2	1.8	1.7

下图是根据上述算法构造的搜索树，其中结点的编号表示了搜索树生成的顺序，边上的数字 1/0 分别表示选择/不选择对应物品。除了根结点之外，每个左孩子结点旁边的上下两个数字分别表示当前背包的重量和已获得的价值，右孩子结点旁边的数字表示扩展了该结点后最多可能获得的价值。为获得最优解，应该选择物品 (5)，获得的价值 为 (6)。



对于上述实例，若采用穷举法搜索整个解空间，则搜索树的结点数为 (7)，而用了上述回溯法，搜索树的结点数为 (8)。

(5) 物品 2 和物品 3

(6) 35

(7) 15

(8) 8

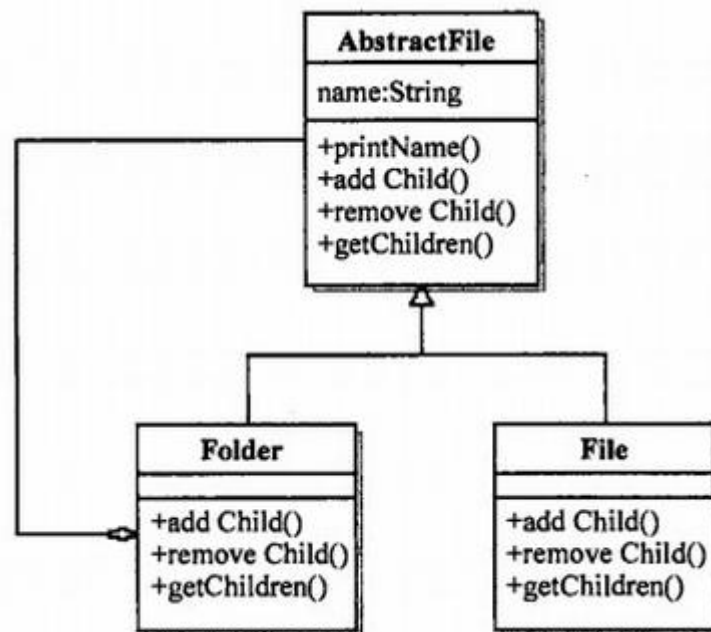
根据问题 1 中给出的伪代码运行该实例，可以很容易得到此 0-1 背包问题的最优解，应该选择物品 2 和物品 3，此时背包的重量为 $10+10=20$ ，获得的价值为 $17+18=35$ 。

若采用穷举法搜索整个解空间，即要构造一颗完全二叉树，此时搜索树的结点数应为 $2^4-1=15$ ，而采用了上述回溯法，搜索树的结点数仅为 8 个，如上图所示。

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。

现欲构造一文件/目录树，采用组合 (Composite) 设计模式来设计，得到的类图如下图所示：



【问题 1】

【C++ 代码】

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

class AbstractFile {
protected :
    string name; // 文件或目录名称
public:
    void printName(){cout << name;} // 打印文件或目录名称
    virtual void addChild(AbstractFile *file)=0;
    // 给一个目录增加子目录或文件
    virtual void removeChild(AbstractFile *file)=0;
    // 删除一个目录的子目录或文件
    virtual list<AbstractFile*> *getChildren()=0;
    // 获得一个目录的子目录或文件
};
```

```

class File : public AbstractFile {
public :
    File(string name) { __ (1) __ = name; }
    void addChild(AbstractFile *file) { return ; }
    void removeChild(AbstractFile *file) { return ; }
    __ (2) __ getChildren() { return __ (3) __; }
};

class Folder :public AbstractFile {
private :
    list <AbstractFile*> childList; // 存储子目录或文件
public :
    Folder(string name) { __ (4) __ = name; }
    void addChild(AbstractFile *file) { childList.push_back(file); }
    void removeChild(AbstractFile *file) { childList.remove(file); }
    list<AbstractFile*> *getChildren() { return __ (5) __; }
};

void main() {
    // 构造一个树形的文件/目录结构
    AbstractFile *rootFolder = new Folder("c:\\");
    AbstractFile *compositeFolder = new Folder("composite");
    AbstractFile *windowsFolder = new Folder("windows");
    AbstractFile *file = new File("TestComposite.java");
    rootFolder->addChild(compositeFolder);
    rootFolder->addChild(windowsFolder);
    compositeFolder->addChild(file);
}

```

(1) this->name

(2) list<AbstractFile* >*

(3) NULL

(4) this->name

(5) &childList

本题考查基本面向对象设计中设计模式的运用能力。

组合设计模式主要是表达整体和部分的关系，并且对整体和部分对象的使用无差别。题目中 AbstractFile 是 File 类和 Folder 类的父类，它抽象了两个类的共有属性和行为，在后续

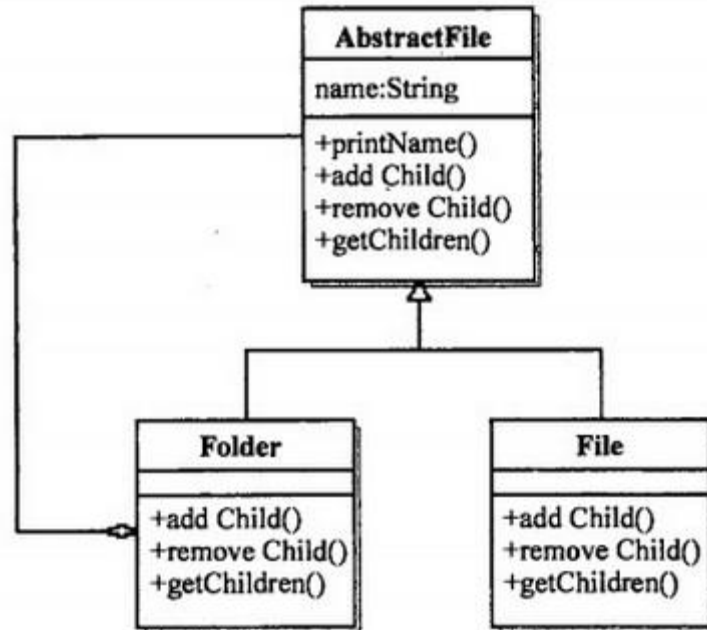
main 方法的使用中，不论是 File 对象还是 Folder 对象，都可被当作 AbstractFile 对象来使用。另外，由于 Folder 对象可以聚合其他的 Folder 对象和 File 对象，等价于 Folder 对象可以聚合另一个 AbstractFile 对象。

在类 File 和类 Folder 的构造函数中都需要记录文件或目录的名称，因此空（1）和 空（4）处主要是设置对象的名称。因为 File 对象不再聚合其他的对象，所以 File 对象没有孩子节点，因此，空（3）处应该返回 NULL 。getChildren() 方法继承自 AbstractFile 类，因此其返回类型也应保持一致。对于空（5），要求返回 Folder 对象的孩子对象，因此返回其成员 childList 的地址。

试题六

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

现欲构造一文件/目录树，采用组合 (Composite) 设计模式来设计，得到的类图如下图所示：



【问题 1】

【Java 代码】

```
import java.util.ArrayList;
import java.util.List;
(1) class AbstractFile {
    protected String name;
    public void printName() {System.out.println(name);}
    public abstract boolean addChild(AbstractFile file);
    public abstract boolean removeChild(AbstractFile file);
    public abstract List<AbstractFile> getChildren();
}
class File extends AbstractFile {
    public File(String name) { this.name = name; }
    public boolean addChild(AbstractFile file) { return false; }
    public boolean removeChild(AbstractFile file) { return false; }
    public List<AbstractFile> getChildren() { return (2); }
}
class Folder extends AbstractFile {
    private List <AbstractFile> childList;
    public Folder(String name) {
        this.name = name;
        this.childList = new ArrayList<AbstractFile>();
    }
}
```

```

public boolean removeChild(AbstractFile file) { return childList.
remove(file); }
public ____ (3) ____ <AbstractFile> getChildren() { return ____ (4) ____; }
}
public class Client {
    public static void main(String[] args) {
        // 构造一个树形的文件/目录结构
        AbstractFile rootFolder = new Folder("c:\\");
        AbstractFile compositeFolder = new Folder("composite");
        AbstractFile windowsFolder = new Folder("windows");
        AbstractFile file = new File("TestComposite.java");
        rootFolder.addChild(compositeFolder);
        rootFolder.addChild(windowsFolder);
        compositeFolder.addChild(file);

        // 打印目录文件树
        printTree(rootFolder);
    }
    private static void printTree(AbstractFile ifile) {
        ifile.printName();
        List <AbstractFile> children = ifile.getChildren();
        if(children == null) return;
        for (AbstractFile file:children) {
            ____ (5) ____;
        }
    }
}

```

该程序运行后输出结果为：

```

c:\
composite
TestComposite.java
Windows

```

(1) abstract

(2) null

(3) List

(4) childList

(5) printTree(file)

本题考查基本面向对象设计中设计模式的运用能力。

组合设计模式主要是表达整体和部分的关系，并且对整体和部分对象的使用无差别。题目中 AbstractFile 是 File 类和 Folder 类的父类，它抽象了两个类的共有属性和行为，在后续 main 方法的使用中，不论是 File 对象还是 Folder 对象，都可被当作 AbstractFile 对象来使用。另外，由于 Folder 对象可以聚合其他的 Folder 对象和 File 对象，等价于 Folder 对象可以聚合另一个 AbstractFile 对象。

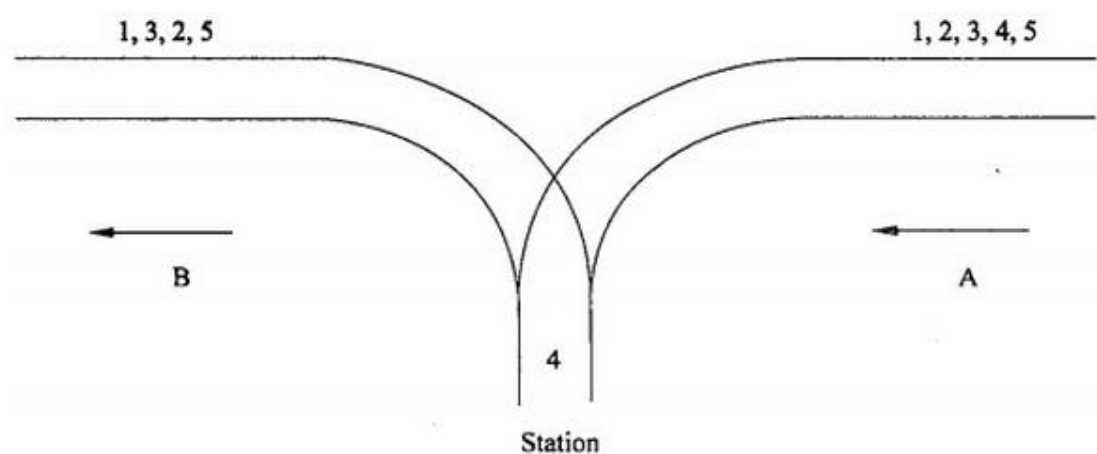
题目中 AbstractFile 类应该为抽象类，因此其修饰符应该包括 abstract，空(2)处返回 File 类对象的孩子，但 File 类对象没有孩子节点，因此其返回值应该为 NULL。getChildren 方法是继承自抽象父类 AbstractFile，所以其返回类型应该和父类的定义保持一致，空(4)处返回存储孩子节点的集合对象 childList。该程序的运行能够打印出文件目录树，因此空(5)处应该为打印方法的调用。

【试题七】

阅读以下说明和 C 程序，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

现有 n ($n < 1000$) 节火车车厢，顺序编号为 1, 2, 3, ..., n ，按编号连续依次从 A 方向的铁轨驶入，从 B 方向铁轨驶出，一旦车厢进入车站 (Station) 就不能再回到 A 方向的铁轨上；一旦车厢驶入 B 方向铁轨就不能再回到车站，如下图所示，其中 Station 为栈结构，初始为空且最多能停放 1000 节车厢。



下面的 C 程序判断能否从 B 方向驶出预先指定的车厢序列，程序中使用了栈类型 STACK，关于栈基本操作的函数原型说明如下：

```
void InitStack(STACK *s): 初始化栈
void Push(STACK *s, int e): 将一个整数压栈，栈中元素数目增 1
void Pop(STACK *s): 栈顶元素出栈，栈中元素数目减 1
int Top(STACK s): 返回非空栈的栈顶元素值，栈中元素数目不变
int IsEmpty(STACK s): 若是空栈则返回 1，否则返回 0
```

【问题 1】

【C 程序】

```
#include<stdio.h>

/*此处为栈类型及其基本操作的定义, 省略*/

int main( ){
    STACK station;
    int state[1000];
    int n;                                /*车厢数*/
    int begin, i, j, maxNo;               /*maxNo 为 A 端正待入栈的车厢编号*/
    printf("请输入车厢数: ");
    scanf("%d",&n);
    printf("请输入需要判断的车厢编号序列 (以空格分隔): ");
    if (n < 1) return -1;
    for (i = 0; i<n; i++) /* 读入需要驶出的车厢编号序列, 存入数组 state[] */
        scanf("%d",&state[i]);
    (1); /*初始化栈*/
    maxNo = 1;
    for(i = 0; i < n;){/*检查输出序列中的每个车厢号 state[i]是否能从栈中获取*/
        if ((2)) /*当栈不为空时*/
            if (state[i] == Top(station)){ /*栈顶车厢号等于被检查车厢号*/
                printf("%d ",Top(station));
                Pop(&station); i++;
            }
        else
            if ((3)){
                printf("error\n");
                return 1;
            }
        else {
            begin = (4);
            for(j = begin+1; j<=state[i]; j++) {
                Push(&station, j);
            }
        }
    }
}
```

```

else { /*当栈为空时*/
    begin = maxNo;
    for(j = begin; j<=state[i]; j++){
        Push(&station, j);
    }
    maxNo = (5);
}
}
printf("OK");
return 0;
}

```

- (1) InitStack(&station)
- (2) !IsEmpty(station)
- (3) state[i] < Top(station)
- (4) Top(station)
- (5) j, 或 state[i]+1

本题考查栈数据结构的应用和 C 程序设计基本能力。

栈的运算特点是后进先出。在本题中，入栈序列为 1、2、…、 $n-1$ 、 n ，出栈序列保存在 state[] 数组中，state[0] 记录出栈序列的第 1 个元素，state[1] 记录出栈序列的第 2 个元素，依此类推。程序采用模拟元素入栈和出栈的操作过程来判断出栈序列是否恰当。需要注意的是，对于栈，应用时不一定是所有元素先入栈后，再逐个进行出栈操作，也不一定是进入一个元素紧接着就出来一个元素，而是栈不满且输入序列还有元素待进入就可以进栈，只要栈不空，栈顶元素就可以出栈，从而使得唯一的一个入栈序列可以得到多个出栈序列。当然，在栈中有多个元素时，只能让栈顶的元素先出栈，栈中其他的元素才能从顶到底逐个出栈。本题中入栈序列和出栈序列的元素为车厢号。

空（1）处对栈进行初始化，根据题干中关于栈基本操作的说明，调用 InitStack 初始化栈，由于形参是指针参数，因此实参应为地址量，即应填入 “InitStack(&station)”。

当栈不空时，就可以令栈顶车厢出栈，空（2）处应填入 “!IsEmpty(station)”。

栈顶车厢号以 Top(station) 表示，若栈顶车厢号等于出栈序列的当前车厢号 state[i]，说明截至到目前为止，出栈子序列 state[0]~state[i] 可经过栈运算获得。由于进栈时小编号的车厢先于大编号的车厢进入栈中，因此若栈顶车厢号大于出栈序列的当前车厢号 state[i]，则对于 state[i] 记录的车厢，若它还在栈中，则此时无法出栈，因为它不在栈顶，所以出错；若它已先于当前的栈顶车厢出栈，则与目前的出栈序列不匹配，仍然出错，因此空（3）处应填入 “state[i] < Top(station)”。

若栈顶车厢号小于出栈序列的当前车厢号 `state[i]`，则说明 `state[i]` 记录的车厢还没有进入栈中，因此从入栈序列（A 端）正待进入的车厢（即比栈顶车厢号正好大 1）开始，直到 `state[i]` 记录的车厢号为止，这些车厢应依次进入栈中。程序中用以下代码实现此操作：

```
for(j = begin+1; j<=state[i]; j++) {  
    Push(&station, j);  
}
```

显然，`begin` 应获取栈顶车厢号的值，即空（4）处应填入 “`Top(station)`”。

还有一种情况，就是待考查的出栈序列还没有结束而栈空了，则说明需要处理入栈序列，使其车厢入栈。程序中用 `maxNo` 表示 A 端正待入栈的车厢编号，相应的处理如下面代码所示：

```
begin = maxNo;  
for(j = begin; j<=state[i]; j++){
```

若栈顶车厢号小于出栈序列的当前车厢号 `state[i]`，则说明 `state[i]` 记录的车厢还没有进入栈中，因此从入栈序列（A 端）正待进入的车厢（即比栈顶车厢号正好大 1）开始，直到 `state[i]` 记录的车厢号为止，这些车厢应依次进入栈中。程序中用以下代码实现此操作：

```
for(j = begin+1; j<=state[i]; j++) {  
    Push(&station, j);  
}
```

显然，`begin` 应获取栈顶车厢号的值，即空（4）处应填入 “`Top(station)`”。

还有一种情况，就是待考查的出栈序列还没有结束而栈空了，则说明需要处理入栈序列，使其车厢入栈。程序中用 `maxNo` 表示 A 端正待入栈的车厢编号，相应的处理如下面代码所示：

```
begin = maxNo;  
for(j = begin; j<=state[i]; j++){  
  
    Push(&station, j);  
}
```

接下来，A 端正待入栈的车厢编号等于 `j` 或 `state[i]+1`，即空（5）处应填入 `j` 或 “`state[i]+1`”。

如果驶出的车厢编号序列是经由栈获得的，则程序运行时输出该序列及字符串 “OK”，否则输出 “error” 而结束。