若某条无条件转移汇编指令采用直接寻址，则该指令的功能是将指令中的地址码送入（1）。

(1)A.PC（程序计数器） B.AR（地址寄存器） C.AC（累加器） D.ALU（算逻运算单元）

【答案】A

【解析】本题考查指令系统基础知识。

直接寻址是指操作数存放在内存单元中，指令中直接给出操作数所在存储单元的地址。而跳转指令中的操作数即为要转向执行的指令地址。因此，应将指令中的地址码送入程序计数器（PC)，以获得下一条指令的地址，从而实现程序执行过程的自动控制功能。

若某计算机系统的 I/O 接口与主存采用统一编址，则输入输出操作是通过(2)指令来完成的。

(2)A.控制 B.中断 C.输入输出 D.访存

【答案】D

【解析】本题考查计算机系统中的输入输出系统基础知识。

常用的 I/O 接口编址方法有两种：一是与内存单元统一编址，二是单独编址。

与内存单元统一编址方式下，是将 I/O 接口中有关的寄存器或存储部件看作存储器单元，与主存中的存储单元统一编址。这样，内存地址和接口地址统一在一个公共的地址空间里，对I/O接口的访问就如同对主存单元的访问一样，可以用访问内存单元的指令访问I/O接口。I/O 接口单独编址是指通过设置单独的 I/O 地址空间，为接口中的有关寄存器或存储部件分配地址码，需要设置专门的 I/o 指令进行访问。这种编址方式的优点是不占用主存的地址空间，访问主存的指令和访问接口的指令不同，在程序中容易使用和辨认。

在程序的执行过程中，Cache 与主存的地址映像由(3)。

(3)A.专门的硬件自动完成 B.程序员进行调度

C.操作系统进行管理 D.程序员和操作系统共同协调完成

【答案】A

【解析】本题考查存储系统基础知识。

高速缓存(Cache)的出现主要有两个因素：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，其次就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静态存储器芯片构成 Cache，以尽可能发挥 CPU 的高速度。因此，必须用硬件来实现 Cache

的全部功能。

总线复用方式可以(4)。

(4)A.提高总线的传输带宽　　B.增加总线的功能

C.减少总线中信号线的数量　　D.提高 CPU 利用率

【答案】C

【解析】本题考查总线基础知识。

总线是一组能为多个部件分时共享的信息传送线，用来连接多个部件并为之提供信息交换通路，通过总线复用方式可以减少总线中信号线的数量，以较少的信号线传输更多的信息。

在 CPU 的寄存器中，(5)对用户是完全透明的。

(5)A.程序计数器　　B.指令寄存器　　C.状态寄存器　　D.通用寄存器

【答案】B

【解析】本题考查计算机系统基础知识。

寄存器组是 CPU 中的一个重要组成部分，它是 CPU 内部的临时存储空间。寄存器既可以用来存放数据和地址，也可以存放控制信息或 CPU 工作时的状态。在 CPU 中增加寄存器的数量，可以使 CPU 把执行程序时所需的数据尽可能地放在寄存器中，从而减少访问内存的次数，提高其运行速度。但是，寄存器的数目也不能太多，除了增加成本外，寄存器地址编码增加还会增加指令的长度。CPU 中的寄存器通常分为存放数据的寄存器、存放地址的寄存器、存放控制信息的寄存器、存放状态信息的寄存器和其他寄存器等类型。

程序计数器是存放指令地址的寄存器，其作用是：当程序顺序执行时，每取出一条指令，程序计数器(PC)内容自动增加一个值，指向下一条要取的指令。当程序出现转移时，则将转移地址送入 PC，然后由 PC 指向新的指令地址。

指令寄存器(IR)用于存放正在执行的指令，指令从内存取出后送入指令寄存器。其操作码部分经指令译码器送微操作信号发生器，其地址码部分指明参加运算的操作数的地址形成方式。在指令执行过程中，指令寄存器中的内容保持不变。

状态字寄存器(PSW)用于保存指令执行完成后产生的条件码，例如运算是否有溢出，结果为正还是为负，是否有进位等。此外，PSW 还保存中断和系统工作状态等信息。

通用寄存器组是 CPU 中的一组工作寄存器，运算时用于暂存操作数或地址。在程序中使用通用寄存器可以减少访问内存的次数，提高运算速度。

在汇编语言程序中，程序员可以直接访问通用寄存器以存取数据，可以访问状态字寄存器以获取有关数据处理结果的相关信息，可以通过相对程序计数器进行寻址，但是不能访问指令寄存器。

CPU 中译码器的主要作用是进行(6)。

(6)A.地址译码　　B.指令译码　　C.数据译码　　D.选择多路数据至 ALU

【答案】B

【解析】本题考查计算机系统基础知识。

CPU 中指令译码器的功能是对现行指令进行分析，确定指令类型和指令所要完成的操作以及寻址方式，并将相应的控制命令发往相关部件。

利用(7)可以获取某 FTP 服务器中是否存在可写目录的信息。

(7)A.防火墙系统　　B.漏洞扫描系统　C.入侵检测系统　D.病毒防御系统

【答案】B

【解析】本题考查网络安全方面网络攻击和防御相关的基础知识。

漏洞扫描技术是检测远程或本地系统安全脆弱性的一种安全技术。通过与目标主机 TCP/IP 端口建立连接并请求某些服务（如 TELNET、FTP 等)，记录目标主机的应答， 搜集目标主机相关信息(如匿名用户是否可以登录等)，从而发现目标主机某些内在的安全弱点。

通过内部发起连接与外部主机建立联系，由外部主机控制并盗取用户信息的恶意代码为(8)。

(8)A.特洛伊木马　　B.蠕虫病毒　　C.宏病毒　　D.CIH 病毒

【答案】A

【解析】本题考查网络安全方面病毒相关的基础知识。

典型网络病毒主要有宏病毒、特洛伊木马、蠕虫病毒、脚本语言病毒等。

宏病毒的传播方式通常如下：字处理程序 Word 在打开一个带宏病毒的文档或模板时，激活了病毒宏，病毒宏将自身复制至 Word 的通用（Normal)模板中，以后在打开或关闭文件时病毒宏就会把病毒复制到该文件中。

特洛伊木马是一种秘密潜伏且能够通过远程网络进行控制的恶意程序。控制者可以控制被秘密植入木马的计算机的一切动作和资源，是恶意攻击者窃取信息的工具。

蠕虫病毒的传播过程一般表现为：蠕虫程序驻于一台或多台机器中，它会扫描其他机器是否有感染同种计算机蠕虫，如果没有，就会通过其内建的传播手段进行感染，以达到使计算机瘫痪的目的。

从认证中心 CA 获取用户 B 的数字证书，该证书用(9)作数字签名:从用户 B 的数字证书中可以获得 B 的公钥。

(9)A.CA 的公钥　　B.CA 的私钥　　C.B 的公钥　　D.B 的私钥

【答案】B

【解析】本题考查数字证书的基础知识。

用户的数字证书由某个可信的证书发放机构（Certification Authority, CA)建立， 并由 CA 或用户将其放入公共目录中。在 X.509 标准中，一般格式的数字证书包含以下数据域：

•版本号：用于区分 X.509 的不同版本

•序列号：由同一发行者(CA)发放的每个证书的序列号是唯一的

•签名算法：签署证书所用的算法及其参数

•发行者：指建立和签署证书的 CA 的 X.509 名字

•有效期：包括证书有效期的起始时间和终止时间

•主体名：指证书持有者的名称及有关信息

•公钥：证书持有者的公钥以及其使用方法

•发行者 ID:任选的名字唯一地标识证书的发行者

•主体 ID:任选的名字唯一地标识证书的持有者

•扩展域：添加的扩充信息

•认证机构的签名：用 CA 私钥对证书的签名

从上述描述可知，数字证书用 CA 私钥做数字签名，从用户的数字证书中可以获得用户的公钥。

(10)指可以不经著作权人许可，不需支付报酬，使用其作品。

(10)A.合理使用　　B.许可使用　　C.强制许可使用　D.法定许可使用

【答案】A

【解析】本题考查知识产权方面的基础知识。

合理使用是指在特定的条件下,法律允许他人自由使用享有著作权的作品而不必征得著作权

人的同意，也不必向著作权人支付报酬，但应当在指明著作权人姓名、作品名称，并且不侵犯著作权人依法享有的合法权利的情况下对著作权人的作品进行使用。

许可使用是指著作权人将自己的作品以一定的方式、在一定的地域和期限内许可他人使用，并由此获得经济利益。

强制许可使用是指在一定条件下，作品的使用者基于某种正当理由，需要使用他人已发表的作品，经申请由著作权行政管理部门授权即可使用该作品，无需征得著作权人同意，但应向其支付报酬。

法定许可是指除著作权人声明不得使用外，使用人在未经著作权人许可的情况下， 向著作权人支付报酬，指明著作权人姓名、作品名称，并且不侵犯著作权人依法享有的合法权利的情况下进行使用。

王某是M国际运输有限公司计算机系统管理员。任职期间，王某根据公司的业务要求开发了“海运出口业务系统”，并由公司使用，随后，王某向国家版权局申请了计算机软件著作权登记，并取得了《计算机软件著作权登记证书》。证书明确软件名称是“海运出口业务系统V1.0”，著作权人为王某。以下说法中，正确的是（11）。

(11)A.海运出口业务系统V1.0的著作权属于王某

B.海运出口业务系统V1.0的著作权属于M公司

C.海运出口业务系统V1.0的著作权属于王某和M公司

D.王某获取的软件著作权登记证是不可以撤销的

【答案】B

【解析】本题考查知识产权方面的基础知识。

王某开发的软件（即“海运出口业务系统V1.0”）是在国际运输有限公司担任计算机系统管理员期间根据国际运输有限公司业务要求开发的，该软件是针对本职工作中明确指定的开发目标所开发的。根据《著作权法》第16条规定，公民为完成法人或者非法人单位工作任务所创作的作品是职务作品。认定作品为职务作品还是个人作品，应考虑两个前提条件：一是作者和所在单位存在劳动关系，二是作品的创作属于作者应当履行的职责。职务作品分为一般职务作品和特殊的职务作品：一般职务作品的著作权由作者享有，单位或其他组织享有在其业务范围内优先使用的权利，期限为二年；特殊的职务作品，除署名权以外，著作权的其他权利由单位享有。所谓特殊职务作品是指《著作权法》第16条第2款规定的两种情况：一是主要利用法人或者其他组织的物质技术条件创作，并由法人或者其他组织承担责任的工

程设计、产品设计图、计算机软件、地图等科 学技术作品；二是法律、法规规定或合同约定著作权由单位享有的职务作品。《计算机软 件保护条例》也有类似的规定，在第十三条中规定了三种情况，一是针对本职工作中明确指定的开发目标所开发的软件；二是开发的软件是从事本职工作活动所预见的结果或 者自然的结果；三是主要使用了法人或者其他组织的资金、专用设备、未公开的专门信 息等物质技术条件所开发并由法人或者其他组织承担责任的软件。王某在公司任职期间 利用公司的资金、设备和各种资料，且是从事本职工作活动所预见的结果。所以，其进行的软件开发行为是职务行为(只要满足上述三个条件之一)，其工作成果应由公司享有。 因此，该软件的著作权应属于国际运输有限公司，但根据法律规定，王某享有署名权。

根据《计算机软件保护条例》第 7 条规定，软件登记机构发放的登记证明文件是登 记事项的初步证明，只是证明登记主体享有软件著作权以及订立许可合同、转让合同的 重要的书面证据，并不是软件著作权产生的依据。因为，软件著作权是自软件开发完成之日起自动产生的，未经登记的软件著作权或软件著作权专有合同和转让合同仍受法律保护。因此，软件登记机构发放的登记证明并不是软件著作权最终归属的证明，如果有相反证明，软件著作权登记证是可以撤销的。该软件是王某针对本职工作中明确指定的开发目标所开发的，该软件的著作权应属于公司。明确真正的著作权人之后，软件著作权登记证书的证明力自然就消失了（只有审判机关才能确定登记证书的有效性)。

计算机通过 MIC （话筒接口)收到的信号是(12)。

(12)A.音频数字信号　　B.音频模拟信号　　C.采样信号　　D.量化信号

【答案】B

【解析】本题考查多媒体基础知识。

MIC（话筒）输出的是音频模拟信号，声卡从 MIC 获取音频模拟信号后，通过模数 转换器（ADC)，将声波振幅信号采样转换成一串数字信号并存储到计算机中。重放时，这些数字信号送到数模转换器（DAC),以同样的采样速度还原为模拟波形，放大后送到扬声器发声，这一技术称为脉冲编码调制技术(PCM)。

(13)既不是图像编码也不是视频编码的国际标准。

(13)A.JPEG　　B.MPEG　　C.ADPCM　　D.H.261

【答案】C

【解析】本题考查多媒体基础知识。

计算机中使用的图像压缩编码方法有多种国际标准和工业标准。目前广泛使用的编码及压缩标准有 JPEG、MPEG 和 H.261。

JEPG (Joint Photographic Experts Group)是一个由 ISO 和 IEC 两个组织机构（国际标准化组织）联合组成的一个专家组，负责制定静态和数字图像数据压缩编码标准，这个专家组开发的算法称为 JPEG 算法，并且成为国际上通用的标准。JPEG 是一个适用范围很广的静态图像数据压缩标准，既可用于灰度图像又可用于彩色图像。

MPEG (Moving Pictures Experts Group)动态图像压缩标准是一个由 ISO 和 IEC 两个组织机构联合组成的一个活动图像专家组制定的标准。1992 年提出 MPEG-1、MPEG-2 标准，用于实现全屏幕压缩编码及解码。MPEG-1 是针对传输率为 1Mbps 到 1.5Mbps 的普通电视质量的视频信号的压缩，MPEG-2 是对每秒 30 帧的 720X572 分辨率的视频信号进行压缩。1999 年发布了 MPEG-4 多媒体应用标准，目前推出了 MPEG-7 多媒体内容描述接口标准等。每个新标准的产生都极大地推动了数字视频的发展和更广泛的应用。

H.261 视频通信编码标准是由国际电话电报咨询委员会 CCITT (Consultative Committee on International Telephone and Telegraph)于 1998 年提出的电话/会议电视的建议标准，该标准又称为 PX64K 标准。CCITT 推出的 H.263 标准用于低位速率通信的电视图像编码。

多媒体制作过程中，不同媒体类型的数据收集、制作需要不同的软、硬件设备和技术手段，动画制作一般通过 (14)进行。

(14)A.字处理软件　　B.视频卡　　C.声卡　　D.图形/图像软件

【答案】D

【解析】本题考查多媒体基础知识。

在多媒体应用中，很重要的一个环节是制作所需要的各种媒体素材。声卡用于处理音频信息，它可以把话筒、录音机、电子乐器等输入的声音信息进行模数转换（A/D)、压缩等处理，也可以把经过计算机处理的数字化的声音信号通过还原（解压缩)、数模转换（D/A)后用音箱播放出来，或者用录音设备记录下来。音频卡的关键技术包括数字音频、音乐合成（FM 合成和波形表合面)、MIDI（乐器数字接口）和音效。数字音频部分具有 44.1kHz 的采样率，8 位以上的分辨率；具有录音和播放声音信号的功能；同时具有压缩采样信号的能力，最常用的压缩方法是自适应脉冲编码调制。数字音频的实现有不同的方法和芯片，大多数采用的是 CODEC 芯片，它具有硬件压缩功能。

视频卡是基于 PC 的一种多媒体视频信号处理设备，用来支持视频信号的输入与输出。它可以采集视频源、声频源和激光视盘机、录像机、摄像机等设备的信息，经过编辑或特技等处理而产生非常精美的画面。还可以对这些画面进行捕捉、数字化、冻结、 存储、压缩、输出等操作。对画面的修整、像素显示调整、缩放功能等都是视频卡支持的标准功能。视频卡的功能是连接摄像机、VCR 影碟机、TV 等设备，以便获取、处理和表现各种动画和数字化视频媒体。

多媒体素材编辑软件用于采集、整理和编辑各种媒体数据。

文字处理软件的功能主要是文字处理，包括文字录入、编辑，文档编辑、排版、管理、打印、表格处理等功能，使用文字处理软件可以创建符合用户要求的、美观的文稿。常用的字处理软件有 WPS、Word、Notebook（记事本)、Writer（写字板）等。

图形/图像软件的主要功能包括显示和编辑图形/图像、图像压缩、图像捕捉、图形/ 图像素材库制作等。例如，Photoshop 用于图像的设计、编辑与处理，其功能强大，是使用最多的一种图形/图像工具软件；Xara3D 是一种 3D 图形软件，可用于制作高质量的三维动画。

确定软件的模块划分及模块之间的调用关系是(15)阶段的任务。

(15)A.需求分析　　B.概要设计　　C.详细设计　　D.编码

【答案】B

【解析】本题考查软件开发过程和软件开发阶段的基础知识。

需求分析确定软件要完成的功能及非功能性要求;概要设计将需求转化为软件的模块划分，确定模块之间的调用关系；详细设计将模块进行细化，得到详细的数据结构和算法；编码根据详细设计进行代码的编写，得到可以运行的软件，并进行单元测试。

利用结构化分析模型进行接口设计时，应以(16)为依据。

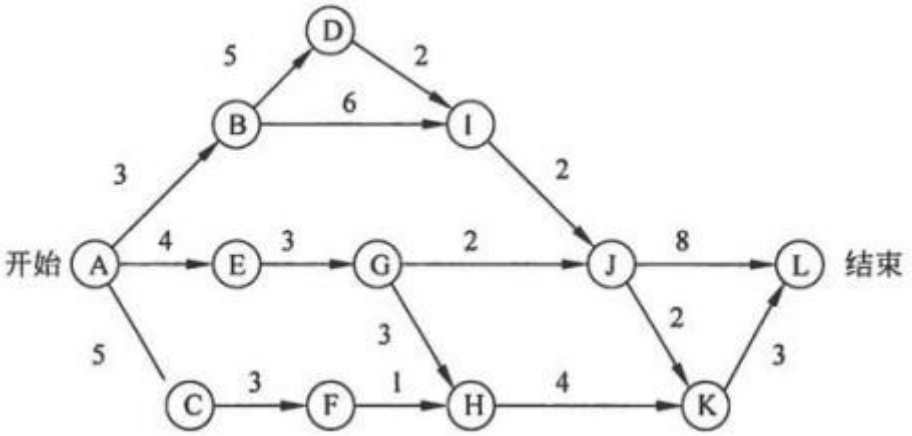
(16)A.数据流图　　B.实体--关系图　　C.数据字典　　D.状态--迁移图

【答案】A

【解析】本题考查结构化分析与设计基础知识。

软件设计必须依据软件的需求来进行,结构化分析的结果为结构化设计提供了最基本的输入信息，其关系为：根据加工规格说明和控制规格说明进行过程设计；根据数据字典和实体关系图进行数据设计；根据数据流图进行接口设计；根据数据流图进行体系结构设计。

下图是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的值表示完成活动所需要的时间，则关键路径长度为(17)。



(17)A.20　　B.19　　C.17　　D.16

【答案】A

【解析】本题考查软件项目管理的相关知识。

关键路径是从开始到结束的最长路径，也是完成项目所需要的最短时间。根据上述活动图，路径 A-B-D-I-J-L 是关键路径，其长度为 20。

甘特图（Gantt 图）不能(18)。

(18)A.作为项目进度管理的一个工具　　B.清晰地描述每个任务的开始和截止时间

C.清晰地获得任务并行进行的信息　D.清晰地获得各任务之间的依赖关系

【答案】D

【解析】

甘特图（Gantt 图）是进行项目进度管理的一个重要工具，它对项目进度进行描述，显示在什么地方活动是并行进行的，并用颜色或图标来指明完成的程度。使用该图，项目经理可以清晰的了解每个任务的开始和截止时间，哪些任务可以并行进行，哪些在关键路径上，但是不能很清晰的看出各任务之间的依赖关系。

以下关于风险管理的叙述中，不正确的是(19)。

(19)A.仅根据风险产生的后果来对风险排优先级

B.可以通过改变系统性能或功能需求来避免某些风险

C.不可能去除所有风险，但可以通过采取行动来降低或者减轻风险

D.在项目开发过程中，需要定期地评估和管理风险

【答案】A

【解析】本题考查风险管理知识。

风险是一种具有负面后果的、人们不希望发生的事件。项目经理必须进行风险管理， 以了解和控制项目中的风险。

风险可能会发生，因此具有一定的概率；风险产生的后果严重程度不一样，因此需要区分。在对风险进行优先级排序时，需要根据风险概率和后果来进行排序。在确定了风险之后，根据实际情况，可以通过改变系统的性能或功能需求来避免某些风险。在项目开发过程中，不可能去除所有风险，但是可以通过采取行动来降低或者减轻风险。而且风险需要定期地评估和管理。

若 C 程序的表达式中引用了未赋初值的变量，则(20)。

(20)A.编译时一定会报告错误信息，该程序不能运行

B.可以通过编译并运行，但运行时一定会报告异常

C.可以通过编译，但链接时一定会报告错误信息而不能运行

D.可以通过编译并运行，但运行结果不一定是期望的结果

【答案】D

【解析】本题考查程序语言翻译基础知识。

在编写 C/C++源程序时，为所定义的变量赋初始值是良好的编程习惯，而赋初值不是强制的要求，因此编译程序不检查变量是否赋初值。如果表达式中引用的变量从定义到使用始终没有赋值，则该变量中的值表现为一个随机数，这样对表达式的求值结果就是不确定的了。

若二维数组 arr[1..M，1..N]的首地址为 base，数组元素按列存储且每个元素占用 K 个存储单元，则元素 arr[i,j]在该数组空间的地址为(21)。

(21)A.base+((i-1)*M+j-1)*K　　B.base+((i-1)*N+j-1)*K

C.base+((j-1)*M+i-1)*K　　D.base+((j-1)*N+i-1)*K

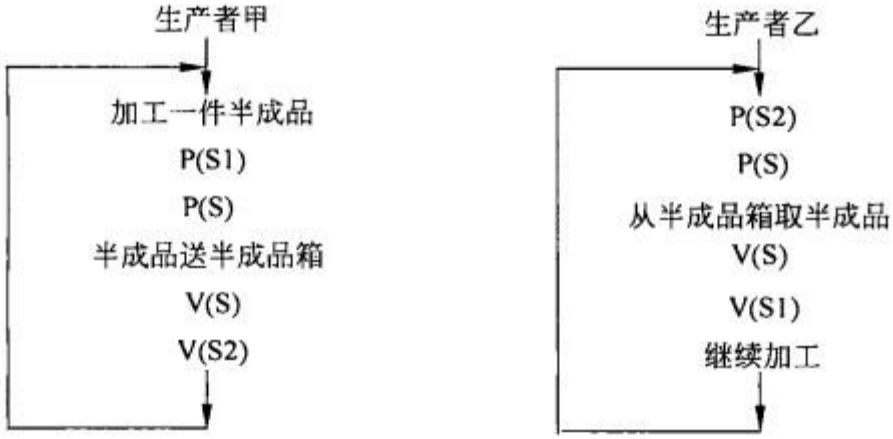
【答案】C

【解析】本题考查数组元素的存储知识。

二维数组 arr[1..M，1..N]的元素可以按行存储，也可以按列存储。按列存储时，元素的排列次序为，先是第一列的所有元素，然后是第二列的所有元素，最后是第 N 列的所 有

元素。每一列的元素则按行号从小到大依次排列。因此，对于元素 arr[i,j]，其存储位置如下计算：先计算其前面 j-1 列上的元素总数，为然后计算第 j 列上排列在 arr[i, j]之前的元素数目，为 i-1,因此 arr[i, j]的地址为 base+((j-1)*M+i-1)*K。

某企业生产流水线 M 共有两位生产者,生产者甲不断地将其工序上加工的半成品放入半成品箱，生产者乙从半成品箱取出继续加工。假设半成品箱可存放 n 件半成品，采用 PV 操作实现生产者甲和生产者乙的同步可以设置三个信号量 S、S1 和 S2,其同步模型如下图所示。



信号量 S 是一个互斥信号量，初值为 (22)； S1、S2 的初值分别为 (23).

(22)A.0　　B.1　　C.n　　D.任意正整数

(23)A.n、0　　B.0、n　　C.1、n　　D.n、1

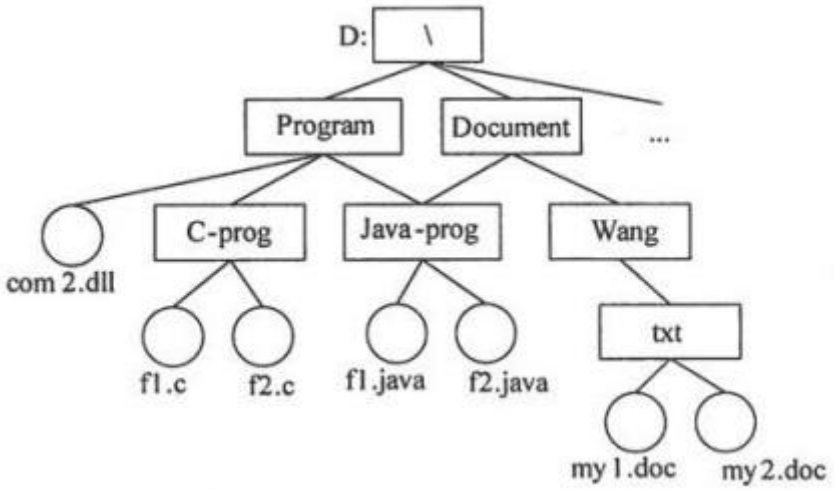
【答案】B　A

【解析】

由于信号量 S 是一个互斥信号量，表示半成品箱当前有无生产者使用，所以初值为 1。

信号量 S1 表示半成品箱容量，故其初值为 n。当生产者甲不断地将其工序上加工的半成品放入半成品箱时，应该先测试半成品箱是否有空位，故生产者甲使用 P(S1)。信号量 S2 表示半成品箱有无半成品，初值为 0。当生产者乙从半成品箱取出继续加工前应先测试半成品箱有无半成品，故生产者乙使用 P(S2)。

若某文件系统的目录结构如下图所示，假设用户要访问文件 f1.java，且当前工作目录为 Program,则该文件的全文件名为(24),其相对路径为(25)。



(24)A. f1. java　　B. \Document\Java-prog\f1. java

C. D:\Program\Java-prog\f1 . java　　D. \Program\Java-prog\f1 . java

(25)A. Java-prog\　　B. \Java-prog\

C. Program\Java-prog　　D. \Program\Java-prog\

【答案】C　A

【解析】

文件的全文件名应包括盘符及从根目录开始的路径名，所以从题图可以看出文件 f1. java 的全文件名为 D:\Program\Java-prog\f1. java。

文件的相对路径是当前工作目录下的路径名,所以从题图可以看出文件 f1. java 的相对路径名为 Java-prog\。

假设磁盘每磁道有 18 个扇区，系统刚完成了 10 号柱面的操作，当前移动臂在 13 号柱面上，进程的请求序列如下表所示。若系统采用 SCAN（扫描）调度算法，则系统响应序列为（26);若系统采用 CSCAN(单向扫描)调度算法，则系统响应序列为（27)。

请求序列	柱面号	磁头号	扇区号
①	15	8	9
②	20	6	5
③	30	9	6
④	20	10	5
⑤	5	4	5
⑥	2	7	4
⑦	15	8	1
⑧	6	3	10
⑨	8	7	9
⑩	15	10	4

(26)A.⑦⑩①②④③⑨⑧⑤⑥　　B.①⑦⑩②③④⑥⑤⑧⑨

C.⑦⑩①②④③⑥⑤⑧⑨　　D.①⑦⑩②③④⑧⑨⑥⑤

(27)A.⑦⑩①②④③⑨⑧⑤⑥　　B.①⑦⑩②③④⑥⑤⑧⑨

C.⑦⑩①②④③⑥⑤⑧⑨　　D.①⑦⑩②③④⑧⑨⑥⑤

【答案】A　C

【解析】

当进程请求读磁盘时，操作系统先进行移臂调度，再进行旋转调度。由于系统刚完成了 10 号柱面的操作，当前移动臂在 13 号柱面上，若系统采用 SCAN（扫描）调度算法，则系统响应柱面序列为 15→20→30→8→6→5→2。

按照旋转调度的原则进程在 15 号柱面上的响应序列为⑦→⑩→①，因为进程访问的是不同磁道上的不同编号的扇区，旋转调度总是让首先到达读写磁头位置下的扇区先进行传送操作。进程在 20 号柱面上的响应序列为②→④，或④→②。对于②和④可以任选一个进行读写，因为进程访问的是不同磁道上具有相同编号的扇区，旋转调度可以任选一个读写磁头位置下的扇区进行传送操作。

从上分析可以得出按照 SCAN（扫描）调度算法的响应序列为⑦⑩①②④③⑨⑧⑤⑥。

(27) 若系统采用 CSCAN（单向扫描）调度算法，在返程时是不响应用户请求的，因此系统的柱面响应序列为 15→20→30→2→5→6→8。

可见，按照 CSCAN（单向扫描）调度算法的响应序列为⑦⑩①②④③⑥⑤⑧⑨。

某程序设计语言规定在源程序中的数据都必须具有类型，然而，(28)并不是做出此规定的理由。

(28)A.为数据合理分配存储单元

B.可以定义和使用动态数据结构

C.可以规定数据对象的取值范围及能够进行的运算

D.对参与表达式求值的数据对象可以进行合法性检查

【答案】B

【解析】本题考查程序语言基础知识。

在机器层面上，所有的数据都是二进制形式的。应用领域中的数据可以有不同的形式、意义和运算，程序中的数据已经进行了抽象，不同类型的数据需要不同大小的存储空间，因此为程序中的数据规定类型后，可以更合理地安排存储空间。不同类型的数据其取值方式和运算也不同，引入类型信息后，在对源程序进行编译时就可以对参与表达式求值的数据对象可以进行合法性检查。

以下关于喷泉模型的叙述中，不正确的是(29)。

(29)A.喷泉模型是以对象作为驱动的模型，适合于面向对象的开发方法

B.喷泉模型克服了瀑布模型不支持软件重用和多项开发活动集成的局限性

C.模型中的开发活动常常需要重复多次，在迭代过程中不断地完善软件系统

D.各开发活动（如分析、设计和编码）之间存在明显的边界试题

【答案】D

【解析】本题考查软件生存周期模型。

喷泉模型是典型的面向对象生命周期模型，是一种以用户需求为动力，以对象作为驱动的模型。该模型克服了瀑布模型不支持软件重用和多项开发活动集成的局限性。“喷泉” 一词本身体现了迭代和无间隙特性。迭代意味着模型中的开发活动常常需要重复多次，在迭代过程中不断地完善软件系统；无间隙是指在开发活动之间不存在明显的边界。

若全面采用新技术开发一个大学记账系统，以替换原有的系统，则宜选择采用(30)进行开发。

(30)A.瀑布模型　　B.演化模型　　C.螺旋模型　　D.原型模型

【答案】A

【解析】本题考查软件生存周期模型。

瀑布模型是将软件生存周期各个活动规定为依线性顺序连接的若干阶段的模型，它为软件的开发和维护提供了一种有效的管理模式，适合于软件需求很明确的软件项目的模型。本题中

开发的大学记账系统是基于原有系统开发的，要求采用新技术，而需求是明确的。

演化模型在获取一组基本的需求后，通过快速分析构造出该软件的一个初始可运行版本，然后逐步演化成为最终软件产品。原型模型快速构造软件的原型，在此基础上开发最终软件产品。这两类模型主要是针对需求不确定或者不清楚的情况下，进行项目开发建议采用的。而螺旋模型增加了风险分析。

将每个用户的数据和其他用户的数据隔离开，是考虑了软件的（31）质量特件。

(31)A.功能性　　B.可靠性　　C.可维护性　　D.易使用性

【答案】A

【解析】本题考查需求分析的相关知识。

要求将每个用户的数据和其他用户的数据隔离开，是安全性要求，而安全性质量子特性在 ISO/IEC 软件质量模型中属于功能性质量特性。

在软件评审中，设计质量是指设计的规格说明书符合用户的要求。设计质量的评审内容不包括（32）。

(32)A.软件可靠性　　B.软件的可测试性　　C.软件性能实现情况　D.模块层次

【答案】D

【解析】本题考查软件设计的相关知识。

为了使用户满意，软件应该满足两个必要条件：设计的规格说明书符合用户的要求，这称为设计质量；程序按照设计规格说明所规定的情况正确执行，这称为程序质量。

设计质量评审的对象是在需求分析阶段产生的软件需求规格说明、数据需求规格说明、在软件概要设计阶段产生的软件概要设计说明书等。主要从以下方面进行评审：软件的规格说明是否合乎用户的要求；可靠性；保密措施实现情况等；操作特性实施情况等；性能实现情况；可修改性、可扩充性、可互换性和可移植性；可测试性；可复用性。

针对应用在运行期的数据特点，修改其排序算法使其更高效，属于（33）维护。

(33)A.正确性　　B.适应性　　C.完善性　　D.预防性

【答案】C

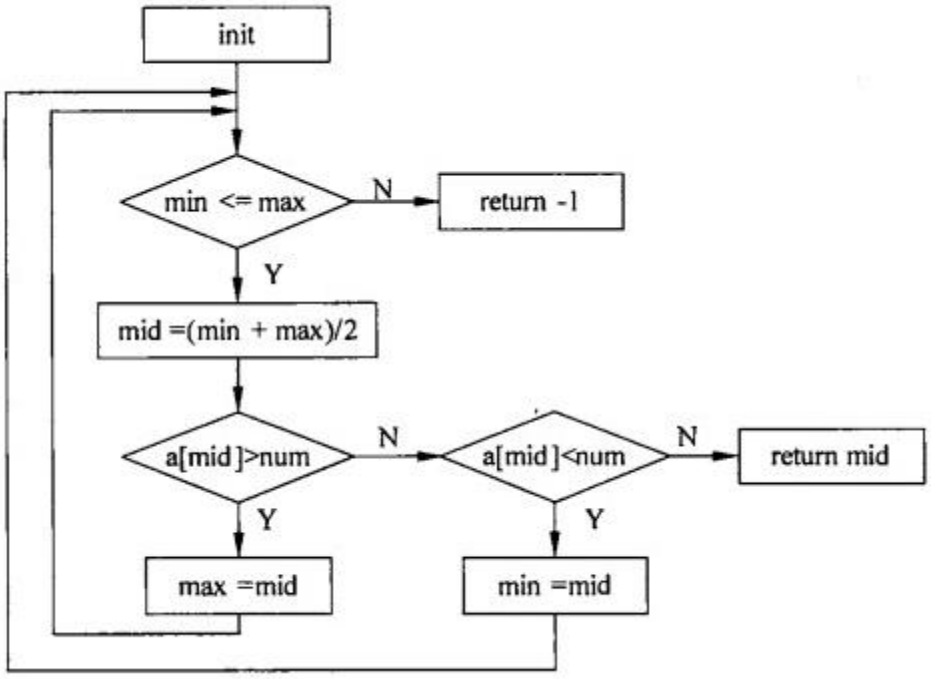
【解析】本题考查软件维护的相关知识。

软件维护的类型一般有四类：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;完善性维护是为扩充功能和改善性能而进行的修改;预防性维护是为了改进应用软件的可靠性和可维护性，为了适应未来变化的软硬件环境的变化，主动增加预防性的新的功能，以适应将来各类变化。

修改现有应用软件中的某个排序算法，提供其运行效率属于完善性维护。

下图所示的逻辑流实现折半查找功能，最少需要 (34) 个测试用例可以覆盖所有的可能路径。



(34)A.1　　B.2　　C.3　　D.4

【答案】B

【解析】本题考查软件测试的相关知识。

折半查找是在一组有序的数（假设为递增顺序）中查找一个数的算法，其思想是：将待查找的数与数组中间位置 mid 的数进行比较，若相等，则查找成功；若大于中间位置的数，则在后半部分进行查找；若小于中间位置的数，则在前半部分进行查找。直到查找成功，返回所查找的数的位置，或者失败，返回-1。设计一个查找成功的测试用例，可以覆盖除了 return-1 之外的所有语句和路径;设计一个查找失败的测试用例，可以覆盖除了 return mid 之外的所有语句和路径。因此，最少需要 2 个测试用例才可以覆盖所有的路径。

在某班级管理系统中，班级的班委有班长、副班长、学习委员和生活委员，且学生年龄在 15～25 岁。若用等价类划分来进行相关测试，则(35)不是好的测试用例。

(35)A.(队长，15)　　B.(班长，20)　　C.(班长，15)　　D.(队长，12)

【答案】D

【解析】本题考查软件测试的相关知识。

等价类划分是一类黑盒测试技术，将程序的输入域划分为若干等价类，然后从每个等价类中选取一个代表性数据作为测试用例。本题的等价类划分可以划分为三个等价类，一个有效等价类 I，即班委来自集合{班长，副班长，学习委员，生活委员}，年龄在 15～ 25; 一个无效等价类 II，即班委不来自集合{班长，副班长，学习委员，生活委员}，而年龄在 15～25; 一个无效等价类 III，即班委来自集合{班长，副班长，学习委员，生活委员}，而年龄不在 15～25。题中选项 A 来自等价类 II，选项 B 和选项 C 来自等价类 I， 而选项 D 则不属于任何等价类，因此不是一个好的测试用例。

进行防错性程序设计，可以有效地控制 (36)维护成本。

(36)A.正确性　　B.适应性　　C.完善性　　D.预防性

【答案】A

【解析】本题考查软件维护的相关知识。

软件维护的类型一般有四类：正确性维护、适应性维护、完善性维护和预防性维护。 防错性的程序设计可以减少在系统运行时发生错误，因此可以有效地控制正确性维护成本。

采用面向对象开发方法时，对象是系统运行时基本实体。以下关于对象的叙述中，正确的是(37)。

(37)A.对象只能包括数据（属性）　　B.对象只能包括操作（行为）

C.对象一定有相同的属性和行为　D.对象通常由对象名、属性和操作三个部分组成

【答案】D

【解析】本题考查面向对象的基本知识。

采用面向对象开发方法时，对象是系统运行时基本实体。它既包括数据（属性）， 也包括作用于数据的操作（行为）。一个对象通常可由对象名、属性和操作三部分组成。

一个类是(38)在定义类时，将属性声明为 private 的目的是(39)。

(38)A.一组对象的封装　　B.表示一组对象的层次关系

C.一组对象的实例　　D.一组对象的抽象定义

（39）A.实现数据隐藏，以免意外更改　　B.操作符重载

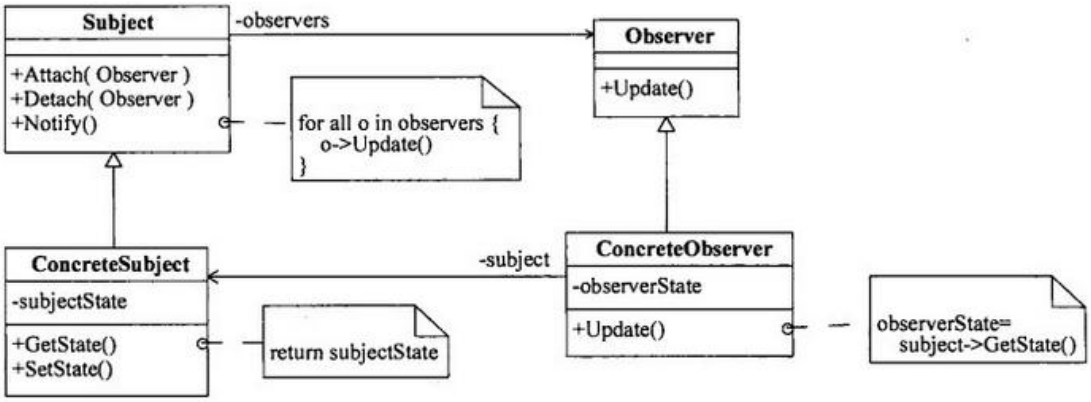
C.实现属性值不可更改　　D.实现属性值对类的所有对象共享

【答案】D　A

【解析】本题考查面向对象的基本知识。

面向对象技术中，将一组大体上相似的对象定义为一个类。把一组对象的共同特征加以抽象并存储在一个类中，是面向对象技术中的一个重要特点。一个所包含的方法和数据描述一组对象的共同行为和属性。在类定义时，属性声明 private 的目的是实现数据隐藏，以免意外更改。

(40)设计模式允许一个对象在其状态改变时，通知依赖它的所有对象。该设计模式的类图如下图，其中，(41)在其状态发生改变时，向它的各个观察者发出通知。



(40)A.命令（Command）　　B.责任链（Chain of Responsibility）

C.观察者(Observer)　　D.迭代器(Iterator)

(41)A.Subject　B.ConcreteSubject　C.Observer　D.ConcreteObserver

【答案】C　B

【解析】本题考查设计模式的基本知识。

命令（Command)模式通过将请求封装为一个对象，可将不同的请求对客户进行参数化。责任链（Chain of Responsibility)模式将多个对象的请求连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止，避免请求的发送者和接收者之间的耦合关系。观察者（Observer)模式定义对象之间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

在上述观察者模式的类图中，Subject（目标）知道其观察者，可以有任意多个观察者观察

同一个目标，提供注册和删除观察者对象的接口。Observer（观察者）为那些在目标发生改变时需获得通知的对象定义一个更新接口。ConcreteSubject（具体目标）将有关状态存入各 ConcreteObserver 对象，当它的状态发生改变时，向它的各个观察者发出通知。ConcreteObserver（具体观察者）维护一个指向 ConcreteSubject 对象的引用，存储有关状态，实现 Observer 的更新接口以使自身状态与目标的状态保持一致。

在面向对象软件开发中，封装是一种(42)技术，其目的是使对象的使用者和生产者分离。

(42)A.接口管理　　B.信息隐藏　　C.多态　　D.聚合

【答案】B

【解析】本题考查面向对象的基础知识。

在面向对象软件开发中，对象是软件系统中基本的运行时实体，对象封装了属性和行为。封装是一种信息隐藏技术，其目的是使对象的使用者和生产者分离，使对象的定义和实现分开。

欲动态地给一个对象添加职责，宜采用(43)模式。

(43)A.适配器（Adapter）　　B.桥接（Bridge）

C.组合（Composite）　　D.装饰器（Decorator）

【答案】D

【解析】本题考查设计模式的基本知识。

适配器(Adapter)模式是将类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。Bridge（桥接）模式将对象的抽象和其实现分离，从而可以独立地改变它们。组合(Composite)模式描述了如何构造一个类层次式结构。装饰器（Decorator)模式的意图是动态地给一个对象添加一些额外职责。在需要给某个对象而不是整个类添加一些功能时使用。这种模式对增加功能比生成 子类更加灵活。

(44)模式通过提供与对象相同的接口来控制对这个对象的访问。

(44)A.适配器（Adapter）　　B.代理（Proxy）

C.组合（Composite）　　D.装饰器（Decorator）

【答案】B

【解析】本题考查设计模式的基本知识。

适配器（Adapter)模式是将类的接口转换成客户希望的另外一个接口。代理（Proxy）模式通过提供与对象相同的接口来控制对这个对象的访问，以使得在确实需要这个对象时才对它

进行创建和初始化。组合（Composite)模式描述了如何构造一个类层次式结构。装饰器（Decorator)模式动态地给一个对象添加职责。

采用 UML 进行面向对象开发时，部署图通常在(45)阶段使用。

(45)A.需求分析　　B.架构设计　　C.实现　　D.实施

【答案】D

【解析】本题考查 UML 面向对象开发的基本知识。

UML2.0 提供多种视图，只有部署图描述系统的物理视图。部署图通常在实施阶段使用，以说明哪些组件或子系统部署于哪些结点。

业务用例和参与者一起描述(46)，而业务对象模型描述(47)。

(46)A.工作过程中的静态元素　　B.工作过程中的动态元素

C.工作过程中的逻辑视图　　D.组织支持的业务过程

(47)A.业务结构　　B.结构元素如何完成业务用例

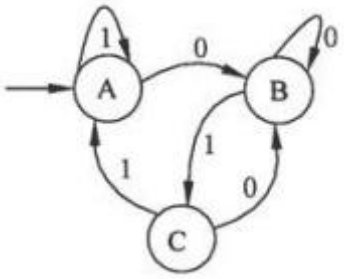
C.业务结构以及结构元素如何完成业务用例　　D.组织支持的业务过程

【答案】D　C

【解析】本题考查面向对象建模的基本知识。

在采用面向对象方法进行业务建模时，业务用例和参与者一起描述组织或企业所支持的业务过程。业务流程被定义为多个不同的业务用例，其中每个业务用例都代表业务中某个特定的工作流程。业务用例确定了执行业务时将要发生的事情，描述了一系列动作的执行，以及产生对特定业务主角具有价值的结果。业务对象模型从业务角色内部的观点定义了业务用例。该模型确定了业务人员及其处理和使用的对象之间应该具有的静态和动态关系，注重业务中承担的角色及其当前职责，既描述业务结构，又描述这些结构元素如何完成业务用例。

下图所示为一个有限自动机（其中，A 是初态、C 是终态)，该自动机识别的语言可用正规式 (48)表示。



(48)A.(0|1)*01　　B.1*0*10*1　　C.1*(0)*01　　D.1*(0|10)*1*

【答案】A

【解析】本题考查程序语言翻译基础知识。

分析题中所给自动机识别字符串的特点可知，该自动机识别的字符串必须以 01 结尾，而之前的 0 和 1 可以以任意方式组合，因此，正规式为(0 | 1)*01。

函数 t、f 的定义如下所示，其中，a 是整型全局变量。设调用函数 t 前 a 的值为 5, 则在函数 t 中以传值调用（call by value)方式调用函数 f 时，输出为(49)在函数 f 中以引用调用（callby reference)方式调用函数 f 时，输出为(50)。

```
t( ): int x = f(a);
      print a+x;

f(int r): a = r+1;   r = r * 2 ;
          return r;
```

(49)A.12　　B.16　　C.20　　D.24

(50)A.12　　B.16　　C.20　　D.24

【答案】B　D

【解析】本题考查函数调用时参数传递基础知识。

发生函数调用时,调用函数与被调用函数之间交换信息的主要方法有传值调用和引用调用两种。

若实现函数调用时实参向形式参数传递相应类型的值，则称为是传值调用。这种方式下形式参数不能向实际参数传递信息。

在 C 语言中，要实现被调用函数对实际参数的修改，必须用指针作形参。即调用时需要先对实参进行取地址运算，然后将实参的地址传递给指针形参。本质上仍属于传值调用。

引用是 C++中增加的数据类型，当形式参数为引用类型时，函数中对形参的访问和修改本质上就是针对相应实际参数变量所作的访问和改变。

本题中，传值调用方式下，表达式“X = f(a)”中调用 f 时，是将 a 的值（即 5)传给 n 这

样执行函数 f 时，r 的初始值为 5,经过“a = r+1”运算后，全局变量 a 的值从 5 变为 6,然后“r = r*2”将 r 的值改变为 10, “return r”将 10 返回并赋值给 x,因此执行“print a+x” 后输出了 16。

传值调用方式下，表达式“x = f(a)”中调用 f 时，r 则是 a 的引用（即 r 是 a 的别名），因此，经过“a = r+1”运算后，a 的值（也就是 r 的值）变为 6,然后“r = r*2”将 r 的值（也就是 a 的值）改变为 12, “return r”使得将 12 返回并赋值给 X，因此执行“print a+x”后输出了 24。

将 Students 表的插入权限赋予用户 UserA，并允许其将该权限授予他人，应使用的 SQL 语句为： GRANT (51) TABLE Students TO UserA (52);

(51)A. UPDATE　　B. UPDATE ON　　C. INSERT　　D. INSERT ON

(52)A. FOR ALL　　B. PUBLIC

C. WITH CHECK OPTION　　D. WITH GRANT OPTION

【答案】D　D

【解析】本题考查关系代数运算与 SQL 查询方面的基础知识。

授权语句的格式如下：

GRANT 〈权限〉 [,<权限>]...[ON<对象类型><对象名>]

TO <用户>[,<用户][WITH GRANT OPTION];

若在授权语句中指定了 “WITH GRANT OPTION”子句，那么，获得了权限的用户还可以将该权限赋给其他用户。

若有关系 R（A，B，C，D）和 S（C，D，E），则与表达式 $\pi_{3,4,7}(\sigma_{4<5}(R\times S))$ 等价的 SQL 语句如下：

SELECT　(53)　FROM　(54)　WHERE　(55)　;

(53)A. A，B，C，D，E　　B. C，D，E

C. R. A, R. B. R. C, R. D, S. E　　D. R. C, R. D. S. E

(54)A. R　　B. S　　C. R, S　　D. RS

(55)A. D<C　　B. R. D< S. C　　C. R. D<R. C　　D. S. D<R. C

【答案】D　C　B

【解析】

本题考查关系代数运算与 SQL 查询方面的基础知识。

在$\pi_{3,4,7}(\sigma_{4<5}(R\times S))$中，R×S 的属性列名分别为：R.A、 R.B、R.C、 R.D、S.C、S.D、和 S.E，$\pi_{3,4,7}(\sigma_{4<5}(R\times S))$的含义是从 R×S 结果集中选取 R.D<S.C 的元组，再进行 R.C、R.D 和 S.E 投影。

E-R 图转换为关系模型时，对于实体 E1 与 E2 间的多对多联系，应该将(56)。

(56)A.E1 的码加上联系上的属性并入 E2

B.E1 的码加上联系上的属性独立构成一个关系模式

C.E2 的码加上联系上的属性独立构成一个关系模式

D.E1 与 E2 码加上联系上的属性独立构成一个关系模式

【答案】D

【解析】本题考查 E-R 图向关系模型转换方面的基础知识。

E-R 图转换为关系模型时，对两个实体间的多对多联系应该转换为一个独立的关系模式，其方法是将两个实体的码加上联系的属性构成一个独立的关系模式。

在 KMP 模式匹配算法中，需要求解模式串 p 的 next 函数值，其定义如下（其中，j 是字符在模式串中的序号)。对于模式串“abaabaca”，其 next 函数值序列为(57)。

$$next[j]=\begin{cases}0 & j=1\\ \max\{k\mid 1<k<j,'p_1p_2\cdots p_{k-1}'='p_{j-k+1}p_{j-k+2}\cdots p_{j-1}'\} & \\ 1 & \text{其他情况}\end{cases}$$

(57)A.01111111　　B.01122341　　C.01234567　　D.01122334

【答案】B

【解析】

根据 next 函数的定义，“abaabaca”的 next 函数值为“01122341”。

对于线性表（由 n 个同类元素构成的线性序列)，采用单向循环链表存储的特点之一是(58)。

(58)A.从表中任意结点出发都能遍历整个链表

B.对表中的任意结点可以进行随机访问

C.对于表中的任意一个结点，访问其直接前驱和直接后继结点所用时间相同

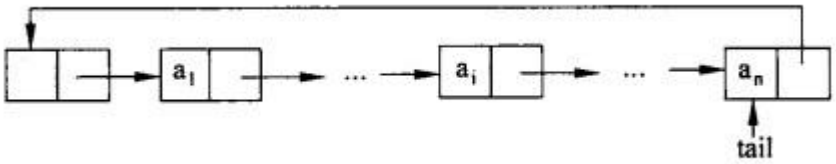
D.第一个结点必须是头结点

【答案】A

【解析】本题考查线性表的链表存储结构知识。

随机访问是指可由元素的序号和第一个元素存储位置的首地址计算得出该序号所对应元素的存储位置，这要求这一组元素必须连续地存储，链表存储结构中元素的存储位置是可以分散的，仅通过指针将逻辑上相邻而存储位置不要求相邻的元素链接起来， 而且只能顺着指针所指示的方向进行遍历。

单向循环链表中指针的指示方向是单方向地，其示意图如下所示，对于表中的任意一个元素，访问其直接后继的运算时间复杂度为O(1)，访问其直接前驱的运算时间复杂度为O(n)。链表中是否含有头结点要看具体的应用情况和运算要求，并没有必须设置的要求。



无向图中一个顶点的度是指图中与该顶点相邻接的顶点数。若无向图G中的顶点数为n，边数为e，则所有顶点的度数之和为（59）。

(59)A.n*e　　B.n+e　　C.2n　　D.2e

【答案】D

【解析】本题考查图结构的基础知识。

对于无向图中的两个顶点u和v，若存在边（u，v)，则该边为计算U的度和V的度各贡献一个值1，因此，所有顶点的度数之和为e的两倍。

一棵满二叉树，其每一层结点个数都达到最大值，对其中的结点从1开始顺序编号， 即根结点编号为1，其左、右孩子结点编号分别为2和3,再下一层从左到右的编号为4、5、6、7,依此类推，每一层都从左到右依次编号，直到最后的叶子结点层为止，则用（60） 可判定编号为m和n的两个结点是否在同一层。

(60)A. $\log_2 m = \log_2 n$　　B. $\lfloor \log_2 m \rfloor = \lfloor \log_2 n \rfloor$

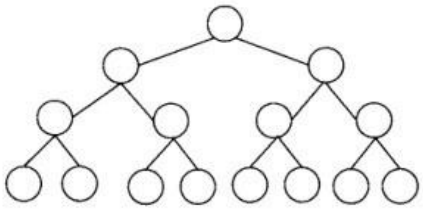
C. $\lfloor \log_2 m \rfloor + 1 = \lfloor \log_2 n \rfloor$　　D. $\lfloor \log_2 m \rfloor = \lfloor \log_2 n \rfloor + 1$

【答案】B

【解析】

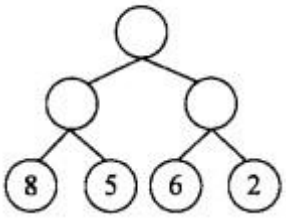
本题考查二叉树基础知识。

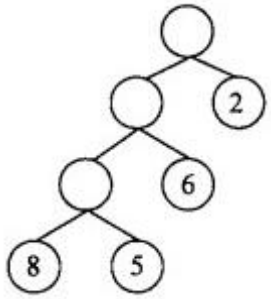
高度为 h 的满二叉树的结点个数为 2^h-1，高度为 4 的满二叉树如下图所示。

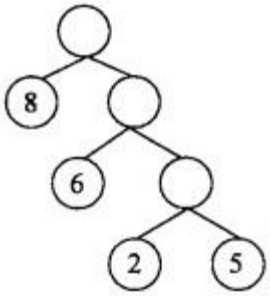


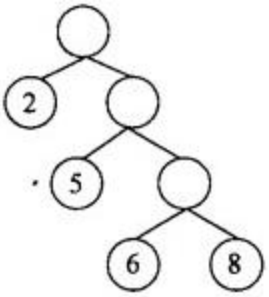
从该例可知，第一层结点的编号为 2^1-1，第二层的结点编号为 $2^1 \sim 2^2-1$，第三层的结点编号为 $2^2 \sim 2^3-1$，…，第 i 层的结点编号为 $2^{i-1} \sim 2^i-1$。因此，对于编号为 m 的结点，其所在层次为 $\lfloor \log_2 m \rfloor+1$，对于编号为 n 的结点，其所在层次为 $\lfloor \log_2 n \rfloor+1$，所以用 $\lfloor \log_2 m \rfloor = \lfloor \log_2 n \rfloor$ 可以判断这两个结点是否在同一层。

(61)是由权值集合{8, 5, 6, 2}构造的哈夫曼树（最优二叉树）。

(61)A. 

B. 

C. 

D. 

【答案】C

【解析】本题考查二叉树应用知识。

构造最优二叉树的哈夫曼算法如下：

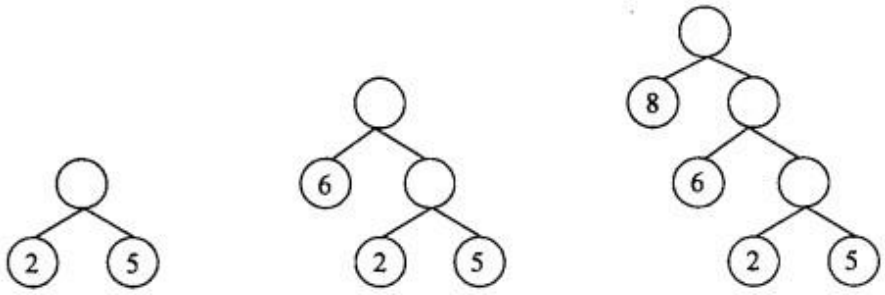
①根据给定的 n 个权值{w1,w2, ..., wn}，构成 n 棵二叉树的集合 F={T1, T2, ...,Tn}，其中每棵二叉树 Ti 中只有一个带权为 wi 的根结点，其左右子树均空。

②在 F 中选取两棵权值最小的二叉树作为左、右子树构造一棵新的二叉树，置新 构造二叉

树的根结点的权值为其左、右子树根结点的权值之和。

③从 F 中删除这两棵树，同时将新得到的二叉树加入到 F 中。

重复②、③，直到 F 中只含一棵树时为止。这棵树便是最优二叉树（哈夫曼树）。 根据题中给出的权值集合，构造哈夫曼树的过程如下图所示。



迪杰斯特拉（Dijkstra)算法用于求解图上的单源点最短路径。该算法按路径长度递增次序产生最短路径，本质上说，该算法是一种基于（62)策略的算法。

(62)A.分治　　B.动态规划　　C.贪心　　D.回溯

【答案】C

【解析】本题考查算法的设计策略。

单源点最短路径问题是指给定图 G 和源点 v0,求从 v0 到图 G 中其余各顶点的最短路径。迪杰斯特拉（Dijkstra)算法是一个求解单源点最短路径的经典算法，其思想是：把图中所有的顶点分成两个集合 S 和 T, S 集合开始时只包含顶点 v0,T 集合开始时包含图中除了顶点 v0 之外的所有顶点。凡是以 v0 为源点，已经确定了最短路径的终点并入 S 集合中，顶点集合 T 则是尚未确定最短路径的顶点集合。按各顶点与 v0 间最短路径长度递增的次序，逐个把 T 集合中的顶点加入到 S 集合中，使得从 v0 到 S 集合中各顶点的路径长度始终不大于从 v0 到 T 集合中各顶点的路径长度。该算法是以一种贪心的方式将 T 集合中的顶点加入到 S 集合中的，而且该贪心方法可以求得问题的最优解。

在有 n 个无序无重复元素值的数组中查找第 i 小的数的算法描述如下:任意取一个元素 r,用划分操作确定其在数组中的位置，假设元素 r 为第 k 小的数。若 i 等于 k，则返回该元素值；若 i 小于 k，则在划分的前半部分递归进行划分操作找第 i 小的数；否则在划分的后半部分递归进行划分操作找第 k-i 小的数。该算法是一种基于（63)策略的算法。

(63)A.分治　　B.动态规划　　C.贪心　　D.回溯

【答案】A

【解析】本题考查算法的设计策略。

从题干可以看出，划分操作与快速排序中的划分操作是一样的，确定某个元素如 r 的最终位置，划分后，在 r 之前的元素都小于 r,在 r 之后的元素都大于 r（假设无重复元素）。因此可以据此确定 r 是数组中第几小的数。题干所述的算法把找第 i 小的数转换为确定任意一个元素是第几小的数，然后根据这个结果再在依据该元素划分后得到的结果在前一部分还是后一部分来继续确定某个元素为第几小的数，重复这种处理，直到找到第 i 小的数。这是分治策略的一个典型应用。

对 n 个元素值分别为–1、0 或 1 的整型数组 A 进行升序排序的算法描述如下：统计 A 中–1、0 和 1 的个数，设分别为 n_1、n_2 和 n_3，然后将 A 中的前 n_1 个元素赋值为–1，第 n_1+1 到 n_1+n_2 个元素赋值为 0，最后 n_3 个元素赋值为 1。该算法的时间复杂度和空间复杂度分别为__（64）__。

(64)A. $\Theta(n)$ 和 $\Theta(1)$　　B. $\Theta(n)$ 和 $\Theta(n)$

C. $\Theta(n^2)$ 和 $\Theta(1)$　　D. $\Theta(n^2)$ 和 $\Theta(n)$

【答案】A

【解析】

本题考查算法的分析技术。

算法首先遍历数组 A 中的所有元素，统计其中–1、0 和 1 的个数，其时间复杂度为 $\Theta(n)$，需要三个额外存储空间，因此空间复杂度 $\Theta(1)$。然后根据前面的统计结果对 A 数组的元素重新赋值，其时间复杂度为 $\Theta(n)$。因此算法的时间复杂度和空间复杂度分别为 $\Theta(n)$ 和 $\Theta(1)$。

设算法 A 的时间复杂度可用递归式 $T(n)=\begin{cases}\Theta(1) & ,n=1\\ 7T(n/2)+n^2 & ,n>1\end{cases}$ 表示，算法 B 的时间复杂度可用递归式 $T(n)=\begin{cases}\Theta(1) & ,n=1\\ aT(n/4)+n^2 & ,n>1\end{cases}$ 表示，若要使得算法 B 渐进地快于算法 A，则 a 的最大整数为__（65）__。

(65)A. 48　　B. 49　　C. 13　　D. 14

【答案】A

【解析】

本题考查算法的分析技术。

根据主定理，算法 A 的时间复杂度分析如下：a=7，b=2，$\log_b a=\log_2 7>2$，因此属于情况（1），时间复杂度为 $n^{\log_2 7}$。算法 B 的时间复杂度分析：a，b=4，$\log_b a=\log_4 a$，要使算法 B 快于算法 A，则需要 $\log_4 a< \log_2 7$，而 $\log_2 7= \log_4 49$，因此有 $\log_4 a< \log_4 49$，该式成立的最大 a 为 48，因此该题选 A。

A 类网络是很大的网络，每个 A 类网络中可以有（66）个网络地址。实际使用中必须把 A 类网络划分为子网，如果指定的子网掩码为 255.255.192.0,则该网络被划分为(67)个子网。

(66)A.210　　B.212　　C.220　　D.224

(67)A.128　　B.256　　C.1024　　D.2048

【答案】D　C

【解析】

A 类网络的地址掩码是 8 比特，剩余的 24 比特可表示主机地址，所以主机地址数为 224 个。如果为 A 类网络指定的子网掩码为 255.255.192.0,则其二进制表示为 11111111 11111111 11000000 00000000,实际上把 A 类网络划分为 210=1024 个子网。

TCP 是互联网中的（68）协议，使用（69）次握手协议建立连接。

(68)A.传输层　　B.网络层　　C.会话层　　D.应用层

(69)A.1　　B.2　　C.3　　D.4

【答案】A　C

【解析】

TCP 是互联网中的传输层协议，使用 3 次握手协议建立连接。这种建立连接的方法可以防止产生错误的连接，这种错误往往是由网络中存储的过期的分组引起的。TCP 使用的流量控制协议是可变大小的滑动窗口协议。

在 WINDOWS 系统中，为排除 DNS 域名解析故障，需要刷新 DNS 解析器缓存，应使用的命令是（70）。

(70)A.IPCONFIG/RENEW　　B.IPCONFIG/FLUSHDNS

C.NETSTAT -R　　D.ARP　- A

【答案】B

【解析】

本题考查 WEB 站点文档及相关知识。刷新和重置缓存的命令是 IPCONFIG /FLUSHDNS。

extreme programming (xp) is a discipline of software development with (71) of simplicity, communication, feedback and courage. successful software development is a team effort - not just the development team, but the larger team consisting of customer, management and developers. xp is a simple process that brings these people together and helps them to succeed together. xp is aimed primarily at object-oriented projects using teams of a dozen or fewer programmers in one location. the principles of xp apply to any (72) project that needs to deliver quality software rapidly and flexibly.

an xp project needs a(an) (73) customer to provide guidance. customers, programmers, managers, are all working (74) to build the system that's needed. customers - those who have software that needs to be developed - will learn simple, effective ways to (75) what they need, to be sure that they are getting what they need, and to steer the project to success.

(71)A.importance B.keys C.roles D.values

(72)A.small-sized B.moderately-sized

C.large-sized D.huge-sized

(73)A.part-time B.casual C.seldom D.full-time

(74)A.together B.by themselves C.separately D.alone

(75) A.tell B.know C.communicate D.feedback

【答案】D B D A C

【解析】

题目中描述敏捷开发方法极限编程（XP)。XP 强调简单、沟通、反馈和勇气 4 个核心价值（VALUES)，适合于需要快速和灵活交付的适当规模（MODERATELY-SIZED)的任何项目。XP 强调客户全职（FIILL-TIME)参与。客户和项目的其他成员工作在一起(TOGETHER)，以简单方式进行有效地沟通（COMMUNICATE)，以掌握项目按照需求向项目成功的方向进行。

极限编程（XP)是一种软件开发方法，其核心价值观是简单、沟通、反馈和勇气。 成功的软件开发是团队努力的结果—不仅仅指开发团队，而是包括了客户、管理人员和开发人员组成

的更大团队。XP 是一种将上述人员组织起来并帮助他们取得成功的简单的过程。XP 主要针对一个十几人或更少程序员组成的、在同一个场所工作的面向对象的项目团队。XP 原则适用于需要快速且灵活地交付高质量软件的中等规模项目组。

一个 XP 项目组需要一个全程参与的客户给予指导。客户、程序员和项目经理协同工作来构建需要的软件系统。客户，也就是需要软件的人，将学到简单而有效的沟通方法，来确保获得他们所需要的，从而引导项目走向成功。

试题一

某公司欲开发招聘系统以提高招聘效率，其主要功能如下：

(1)接受申请

验证应聘者所提供的自身信息是否完整，是否说明了应聘职位，受理验证合格的申请，给应聘者发送致谢信息。

(2)评估应聘者

根据部门经理设置的职位要求，审查已经受理的申请；对未被录用的应聘者进行谢绝处理，将未被录用的应聘者信息存入未录用的应聘者表，并给其发送谢绝决策；对录用的应聘者进行职位安排评价，将评价结果存入评价结果表，并给其发送录用决策，发送录用职位和录用者信息给工资系统。

现采用结构化方法对招聘系统进行分析与设计，获得如图 1-1 所示的顶层数据流图、图 1-2 所示 0 层数据流图和图 1-3 所示 1 层数据流图。

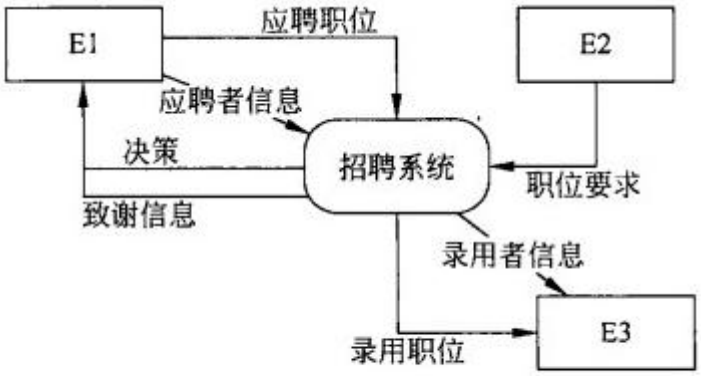


图 1-1　顶层数据流图

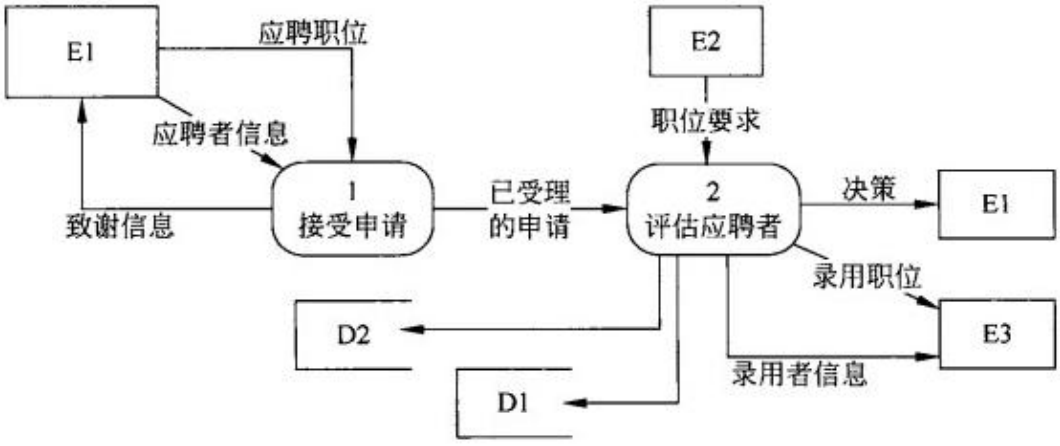


图 1-2　0 层数据流图

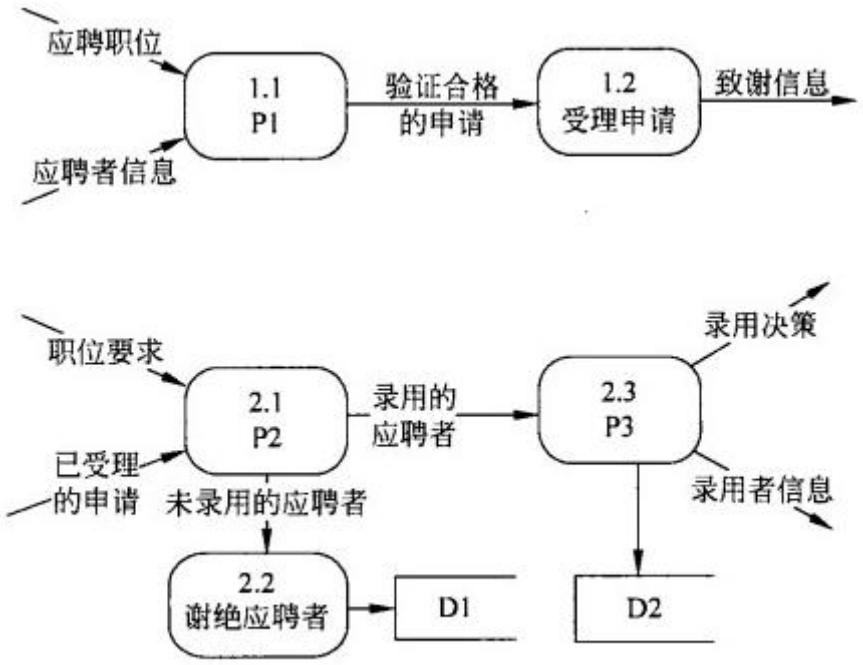


图 1-3　1 层数据流图

【问题 1】

使用说明中的术语，给出图中 E1～E3 所对应的实体名称。

本问题考查顶层 DFD。顶层 DFD 一般用来确定系统边界，将待开发系统看作一个加工，因此图中只有唯一的一个处理和一些外部实体，以及这两者之间的输入输出数据流。外部实体可以是使用系统的用户，也可以是为系统提供输入或接收系统输出的外部系统。本问题要求根据描述确定图中的外部实体。应仔细分析题目中描述;并结合已经在顶层数据流图中给出的数据流进行分析。从题目的说明中可以看出，与系统的交互者包括应聘者、部门经理和工资系统。分析说明中的描述可知，应聘者提供自身信息，并接收系统验证合格后的致谢信息等。部门经理设置职位要求。对录用者而言，将其录用职位和信息发送给工资系统。对应图 1-1 中数据流和实体的对应关系，可知 EI 为应聘者，E2 为部门经理，E3 为工资系统。

【问题 2】

使用说明中的术语，给出图中 D1～D2 所对应的数据存储名称。

本问题考查 DFD 中数据存储的确定。本题中涉及的数据存储只有 2 个，一个是存储未被录用的应聘者信息，即未录用的应聘者表：另一个是存储对录用的应聘者进行职位安排评价的评价结果，即评价结果表。可以确定图 1-2 中 D1 和 D2 为未录用的应聘者表和评价结果表，因为有一个处理与这两个数据存储相关,需要再对应图 1-3,可确认 D1 为未录用的应聘者表，D2 为评价结果表。

【问题 3】

使用说明和图中的术语，给出图 1-3 中加工 P1～P3 的名称。

本问题考查 1 层 DFD 中缺失的处理。从说明（1）中接受申请的描述功能，需先对应聘者信息进行验证，受理验证合格的申请，可知缺失的处理 P1 为验证信息。说明（2）中，根据职位要求，审查已经受理的申请，对录用者进行职位安排评价，可知缺失的处理 P2 为审查申请，P3 为职位安排评价。

【问题 4】

解释说明图 1-2 和图 1-3 是否保持平衡，若不平衡请按如下格式补充图 1-3 中数据流的名称以及数据流的起点或终点，使其平衡（使用说明中的术语或图中符号）。

数据流名称	起点	终点

不平衡。图 1-2 中加工的输入输出流与其子图 1-3 中的输入输出流的数量不同。

数据流名称	起　点
录用职位	P3 或 2.3 职位安排评价
已受理的申请	1.2 受理申请
谢绝决策	2.2 谢绝应聘者

本问题考查绘制分层 DFD 时的注意事项。在分层 DFD 中，需要保持父图与子图的平衡。即父图中某加工的输入输出数据流必须与其子图的输入输出数据流在数量和名字上相同，或者父图的一个输入（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。

本题中，图 1-2 中加工的输入输出流与其子图 1-3 中的输入输出流的数量不同。也无需将父图中一条数据流分解成子图中多条数据流，因此，补充子图中缺失的输入或输出数据流：录用职位、已受理的申请、谢绝决策。

试题二

某物流公司为了整合上游供应商与下游客户，缩短物流过程，降低产品库存，需要构建一个信息系统以方便管理其业务运作活动。

【需求分析结果】

（1）物流公司包含若干部门，部门信息包括部门号、部门名称、经理、电话和邮箱。一个部门可以有多名员工处理部门的日常事务，每名员工只能在一个部门工作。每个部门有一名经理，只需负责管理本部门的事务和人员。

（2）员工信息包括员工号、姓名、职位、电话号码和工资；其中，职位包括：经理、业务员等。业务员根据托运申请负责安排承运货物事宜，例如：装货时间、到达时间等。一个业务员可以安排多个托运申请，但一个托运申请只由一个业务员处理。

（3）客户信息包括客户号、单位名称、通信地址、所属省份、联系人、联系电话、银行账号，其中，客户号唯一标识客户信息的每一个元组。每当客户要进行货物托运时，先要提出货物托运申请。托运申请信息包括申请号、客户号、货物名称、数量、运费、出发地、目的地。其中，一个申请号对应唯一的一个托运申请；一个客户可以有多个货物托运申请，但一个托运申请对应唯一的一个客户号。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如图 2-1 所示。

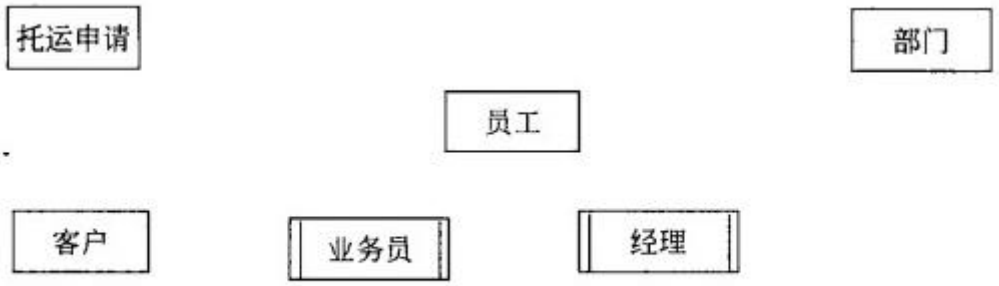


图 2-1　实体联系图

【关系模式设计】

部门（部门号，部门名称，经理，电话，邮箱）

员工（员工号，姓名，职位，电话号码，工资，(A)）

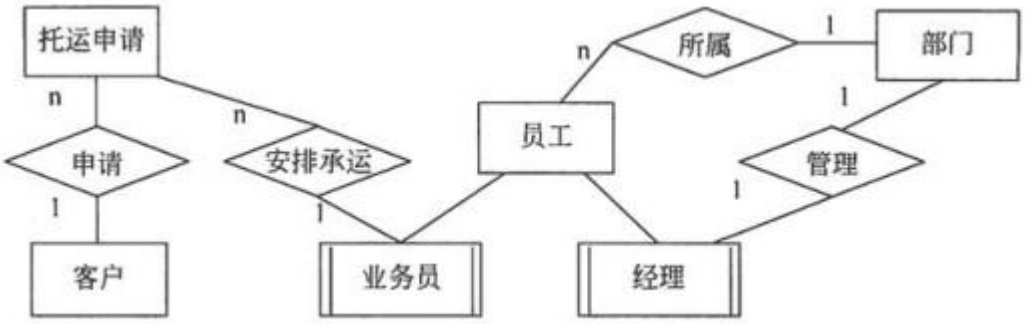
客户（(B) 单位名称，通信地址，所属省份，联系人，联系电话，银行账号）

托运申请（(C)，货物名称，数量，运费，出发地，目的地）

安排承运（(D)，装货时间，到达时间，业务员）

【问题 1】

根据问题描述，补充四个联系、联系的类型，以及实体与子实体的联系，完善图 2-1 所示的实体联系图。



两个实体集之间的联系类型分为三类：一对一（1：1)联系、一对多（1：N)联系 和多对多（M：N)联系。

根据题意，每名员工只能在一个部门工作，所以部门和员工之间有一个 1：N 的“所属”联系；由于每个部门有一名经理，只需负责管理本部门的事务和人员，因此部门和经理之间有一个 1：1 的“管理”联系；由于一个业务员可以安排多个托运申请，但一个托运申请只由一个业务员处理，故业务员和托运申请之间有一个 1 ：N 的“托运”联系； 又由于一个客户可以有多个货物托运申请，但一个托运申请对应唯一的一个客户号，故客户和托运申请之间有一个 1：N 的“申请”联系。

【问题 2】

根据实体联系图，将关系模式中的空（A)～（D)补充完整。分别指出部门、员工和安排承运关系模式的主键和外键。

（a）	部门号	
（b）	客户号	
（c）	申请号，客户号	
（d）	申请号	
部门	主键：部门号	外键：经理
员工	主键：员工号	外键：部门号
安排承运	主键：申请号	外键：业务员

根据题意，部门和员工之间有一个 1：N 的“所属”联系需要将一端的码并入多端，故员工关系模式中的空（A）应填写部门号；在客户关系模式中，客户号为主键，故空（B）应填写客户号；在托运申请关系模式中；申请号、客户号为主键，故空（C）应填写申请号、客户

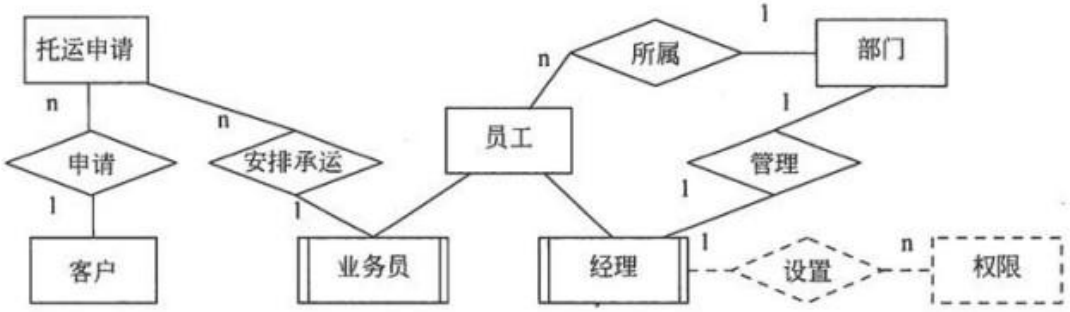
号；又由于一个业务员可以安排多个托运申请，但一个托运申请只由一个业务员处理，因此在安排承运关系模式中，申请号为主键，故空（D）应填写申请号。

部门关系模式中的部门号为主键，经理为外键；因为经理来自员工关系。员工关系模式中的员工号为主键，部门号为外键，因为部门号来自部门关系。安排承运关系模式中的申请号为主键，业务员为外键，因为业务员来自员工关系。

【问题 3】

若系统新增需求描述如下：

为了数据库信息的安全性，公司要求对数据库操作设置权限管理功能，当员工登录系统时，系统需要检查员工的权限。权限的设置人是部门经理。为满足上述需要，应如何修改（或补充）图 2-1 所示的实体联系图，请给出修改后的实体联系图和关系模式。



根据题意，权限的设置人是部门经理，因此，需要建立一个权限关系模式，以及经理到权限之间的 1 ： N 的“设置”联系。

试题三

PAY&DRIVE 系统（开多少付多少）能够根据驾驶里程自动计算应付的费用。

系统中存储了特定区域的道路交通网的信息。道路交通网由若干个路段（ROAD SEGMENT）构成，每个路段由两个地理坐标点（NODE）标定，其里程数（DISTANCE）是已知的。在某些地理坐标点上安装了访问控制（ACCESSCONTROL）设备，可以自动扫描行驶卡（CARD）。行程（TRAJECTORY)由一组连续的路段构成。行程的起点（ENTRY）和终点（EXIT）都装有访问控制设备。

系统提供了 3 种行驶卡。常规卡（REGULAR CARD）有效期（VALID PERIOD）为一年，可以在整个道路交通网内使用。季卡（SEASONCARD)有效期为三个月，可以在整个道路交通网内使用。单次卡（MINITRIP CARD)在指定的行程内使用，且只能使用一次。其中，季卡和单次卡都是预付卡（PREPAIDCARD)，需要客户（CUSTOMER）预存一定的费用。

系统的主要功能有：客户注册、申请行驶卡、使用行驶卡行驶等。

使用常规卡行驶，在进入行程起点时，系统记录行程起点、进入时间（DATE OF ENTRY）等信息。在到达行程终点时，系统根据行驶的里程数和所持卡的里程单价（UNIT PRICE）计算应付费用，并打印费用单（INVOICE)。

季卡的使用流程与常规卡类似,但是不需要打印费用单,系统自动从卡中扣除应付费用。单次卡的使用流程与季卡类似,但还需要在行程的起点和终点上检查行驶路线是否符合该卡所规定的行驶路线。

现采用面向对象方法开发该系统，使用 UML 进行建模。构建出的用例图和类图分别如图 3-1 和图 3-2 所示。

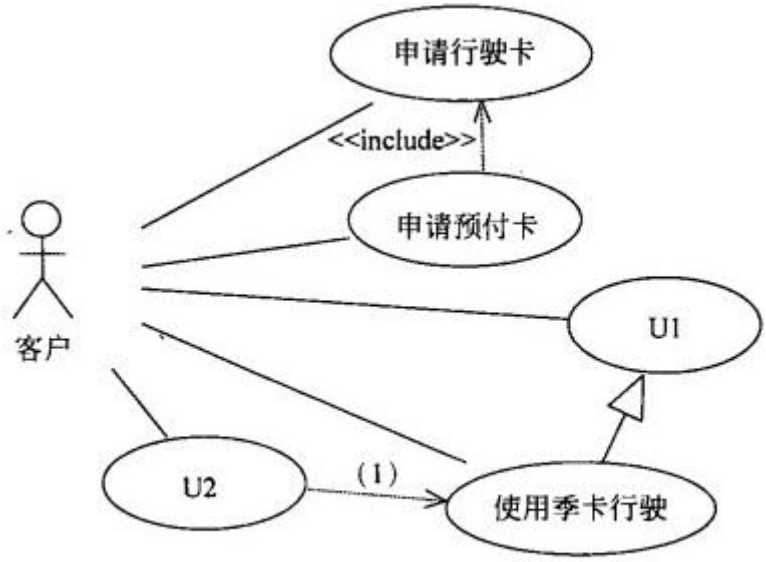


图 3-1　用例图

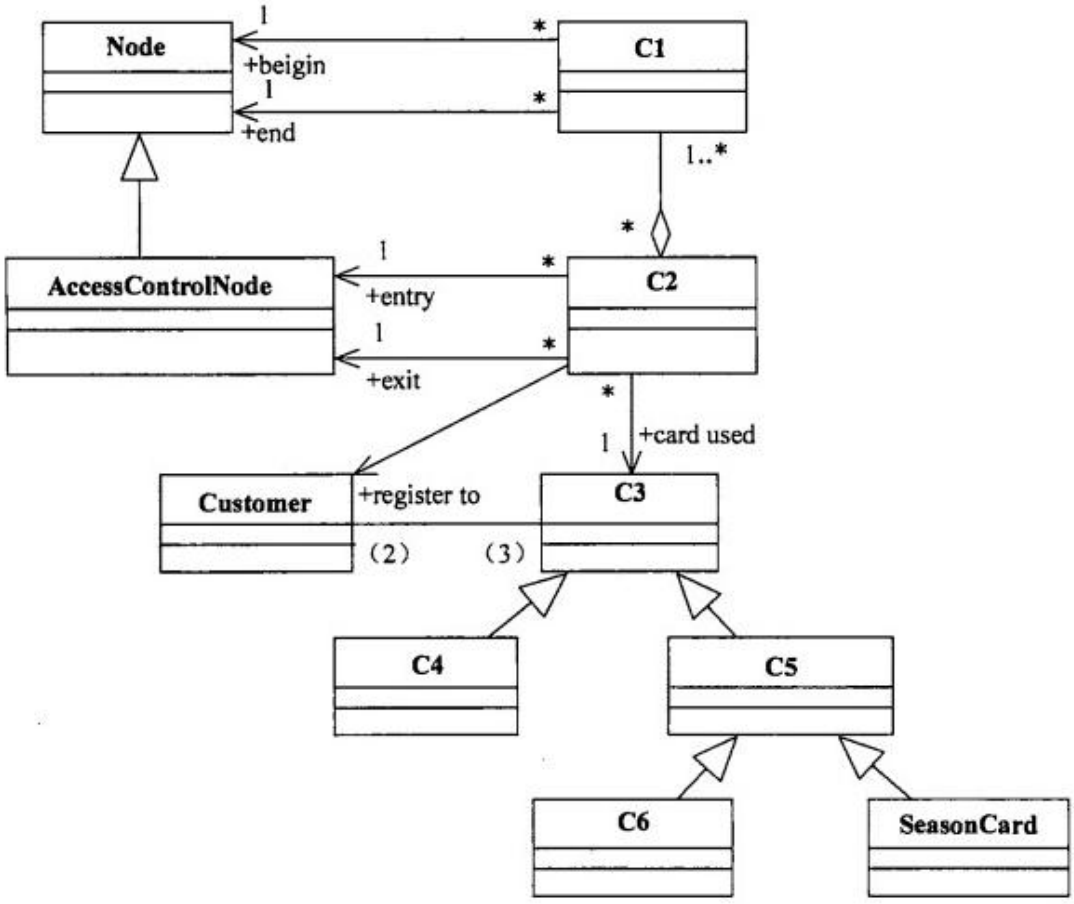


图 3-2　类图

【问题 1】

根据说明中的描述，给出图 3-1 中 U1 和 U2 所对应的用例，以及（1）所对应的关系。

U1：使用常规卡行驶

U2：使用单次卡行驶

（1）：EXTEND

本问题要求将图 3-1 所给出的用例图补充完整。用例图的构成要素有：参与者、用例以及用例之间的关系。图中缺少了两个用例，以及一个用例关系。解答此题时，首先应从说明中找到所有的用例。

用例表示系统的一个单一业务功能。从题目的描述中可以看出，系统的主要功能就是申请行驶卡，以及使用行驶卡行驶。由于行驶卡分为三种，所以在说明中详细描述了三种行驶卡的使用方法。再结合用例图来看，缺少的两个用例与用例“使用季卡行驶”有关联关系，由此可以推断出，需要补充的这两个用例必定与另两种行驶卡相关，分别为“使用常规卡行驶”和“使用单次卡行驶”。

下面需要解决的问题是这两个用例与 U1 和 U2 的对应关系。这就需要仔细考查一下用例图所给出的用例关系。由图 3-1 可知，U1 和“使用季卡行驶”之间是泛化 (GENERALIZATION)关系。当多个用例共同拥有一种类似的结构和行为时，可以将它们的共性抽象为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。根据说明中的“季卡的使用流程与常规卡类似，但是不需要打印费用单，系统自动从卡中扣除应付费用”可知，U1 应该对应着用例“使用常规卡行驶”。由此不难得出 U2 对应着用例“用单次卡行驶”。

现在图中只剩下(1)处的用例关系没有确定。用例之间的关系在用例图上只有三种：包含(INCLUDE)、扩展(EXTEND)和泛化(GENERALIZATION)。

包含关系是指当多个用例中存在相同事件流时，可以把这些公共事件流抽象成为公共用例，这个公共用例称为抽象用例，而原始用例称为基础用例。基础用例和抽象用例之间是包含关系。

如果一个用例明显地混合了两种或两种以上的不同场景，则可以将这个用例分为一个基本用例和多个扩展用例。扩展关系用"《EXTEND》"表示，箭头指向基本用例。

包含关系和扩展关系的区别在于，抽象用例中的事件流一定要插入到基本用例中去，并且插入点只有一个，通常抽象用例不能脱离基本用例而独立存在。扩展用例的事件流往往可以抽象为基本用例的备选事件流，在扩展关系中，可以根据一定的条件来决定是否将扩展用例的

事件流插入到基本用例的事件流中，并且插入点可以有多个。

根据以上分析可知，（1）处的用例关系选择“《EXTEND》”最为合适。

【问题 2】

根据说明中的描述，给出图 3-2 中缺少的 C1～C6 所对应的类名以及（2）～（3） 处所对应的多重度（类名使用说明中给出的英文词汇）。

C1： ROADSEGMENT

C2： TRAJECTORY

C3： CARD

C4： REGULARCARD

C5： PREPAIDCARD

C6： MINITRIPCARD

（2）1

（3）1..3

本问题考查的是类图建模。解题的重点在于根据类图中提供的类及类之间的关联关系，推断出剩余的类。

可以先观察一下类图。可以看到，需要补充的类基本上集中在两个结构上：聚集结构（类 C1 和 C2）以及继承结构（类 C3～C6）。继承结构是比较容易辨识的类之间的关联关系，图上给出了其中的一个子类 SEASONCARD。以这个类为线索，回到说明中寻找与类 SEASONCARD 相关的其他类。从说明中可知，“系统提供了 3 种卡”，常规卡、季卡、单次卡，而“季卡和单次卡都是预付卡”。这些描述暗示，“季卡”、“单次卡”与“预付卡” 之间存在着特殊/一般关系，即“IS-A”关系，这是继承结构的典型标志。由此可以得出类 C5 和 C6 应该分别对应 PREPAIDCARD（预付卡）和 MINITRIPCARD（单次卡）。根据 C5 和 C6 所对应的类，可以推断出，C4 和 C3 必定也是与行驶卡相关的类。三种卡中，已经有两种卡有了对应的类，还剩下一种卡即“常规卡”。而“常规卡”只能是与“预付卡” 同层次的概念，所以只能对应于 C4，C3 表示的是能代表所有这几种卡的公共概念。所 以 C3 和 C4 应分别对应于 CARD 和 REGULARCARD。确定了 C3 之后，就可以识别出（2）和（3）处的多重度。CUSTOMER 和 CARD 之间是持有和被持有的关系，由于系统中只有 3 种卡，所以一个客户最多只能有 3 种卡，所

以(3)处应填 1..3。而对于任何一张卡来说，只能有唯一地一个所属人，因此(2)处应填 1。现在还剩下类 C1 和 C2 没有确定。由于这两个类之间是聚集关系，所以需要在说明中寻找具有“部分-整体”关系的概念。由说明中的“行程（TRAJECTORY)由一组连续的路段构成”可知，C1 和 C2 应分别对应于 ROADSEGMENT 和 TRAJECTORY。

【问题 3】

根据说明中的描述,给出 ROAD SEGMENT、TRAJECTORY 和 CARD 所对应的类的关键属性（属性名使用说明中给出的英文词汇)。

ROADSEGMENT 的属性：DISTANCE

RAJECTORY 的属性：ENTRY、EXIT、DATEOFENTRY

CARD 的属性：UNITPRICE、VALIDPERIOD

本问题考查类的关键属性的识别。由说明中给出的描述可知，类 ROADSEGMENT 的属性至少应包括 DISTANCE;类 TRAJECTORY 的属性至少应包括 ENTRY、EXIT 和 DATEOFENTRY;类 CARD 的属性至少应包括 UNITPRICE、VALIDPERIOD。

试题四

设某一机器由 N 个部件组成，每一个部件都可以从 M 个不同的供应商处购得。供应商 J 供应的部件 I 具有重量 WIJ 和价格 CIJ。设计一个算法，求解总价格不超过上限 CC 的最小重量的机器组成。

采用回溯法来求解该问题：

首先定义解空间。解空间由长度为 N 的向量组成，其中每个分量取值来自集合{1,2，•••，M} 将解空间用树形结构表示。

接着从根结点开始，以深度优先的方式搜索整个解空间。从根结点开始，根结点成为活结点，同时也成为当前的扩展结点。向纵深方向考虑第一个部件从第一个供应商处购买，得到一个新结点。判断当前的机器价格(C11)是否超过上限(CC)，重量(W11) 是否比当前已知的解（最小重量）大，若是，应回溯至最近的一个活结点；若否，则该新结点成为活结点，同时也成为当前的扩展结点，根结点不再是扩展结点。继续向纵深方向考虑第二个部件从第一个供应商处购买，得到一个新结点。同样判断当前的机器价格(C11+C21)是否超过上限(CC)，重量(W11+W21)是否比当前已知的解(最小重量)大。若是，应回溯至最近的一个活结点；若否，则该新结点成为活结点，同时也成为当前的扩展结点，原来的结点不再是扩展结点。以这种方式递归地在解空间中搜索，直到找到所要求的解或者解空间中已无活结点为止。

【C 代码】

下面是该算法的 C 语言实现。

（1）变量说明

n：机器的部件数

m：供应商数

cc：价格上限

w[][]：二维数组，w[i][j]表示第 j 个供应商供应的第 i 个部件的重量

c[][]：二维数组，c[i][j]表示第 j 个供应商供应的第 i 个部件的价格

bestW：满足价格上限约束条件的最小机器重量

bestC：最小重量机器的价格

bestX[]：最优解，一维数组，bestX[i]表示第 i 个部件来自哪个供应商

cw：搜索过程中机器的重量

cp：搜索过程中机器的价格

x[]：搜索过程中产生的解，x[i]表示第 i 个部件来自哪个供应商

i：当前考虑的部件，从 0 到 n–1

j：循环变量

（2）函数 backtrack

```
int n = 3;
int m = 3;
int cc = 4;
int w[3][3] = {{1,2,3},{3,2,1},{2,2,2}};
int c[3][3] = {{1,2,3},{3,2,1},{2,2,2}};
int bestW = 8;
int bestC = 0;
int bestX[3] = {0,0,0};
int cw = 0;
int cp = 0;
int x[3] = {0,0,0};
int backtrack(int i){
```

```
    int j = 0;
    int found = 0;
    if(i > n - 1){  /*得到问题解*/
        bestW = cw;
        bestC = cp;
        for(j = 0; j < n; j++){
            ____(1)____;
        }
return 1;
    }
    if(cp <= cc){ /*有解*/
        found = 1;
    }
    for(j = 0; ___(2)___; j++){
        /*第 i 个部件从第 j 个供应商购买*/
____(3)____;
        cw = cw + w[i][j];
        cp = cp + c[i][j];
if(cp <= cc &&___(4)___){ /*深度搜索，扩展当前结点*/
            if(backtrack(i + 1)){   found = 1;  }
        }
        /*回溯*/
        cw = cw - w[i][j];
        ___(5)___ ;
    }
    return found;
}
```

(1) BESTX[J] = X[J]

(2) J <M

(3) X[I] = J

(4) CW<BESTW

(5) CP = CP-C[I][J]

本题考查算法的设计和分析技术中的回溯法。

回溯法是一种系统搜索问题解的方法，在包含问题所有解的解空间树中，按照深度优先的策略，从根结点出发搜索解空间树。算法在到达解空间树的任一结点时，总是先判断该结点是否肯定不包含问题的解。若肯定不包含，则跳过对以该结点为根的子树的系统搜索，逐层向其祖先结点回溯；否则进入该子树，继续按深度优先的策略进行搜索。回溯法在求问题的最优解时，要回溯到根，且根结点的所有子树都已经被搜索遍才结束。

根据上述思想和题干说明，对实例：部件数 N = 3,厂商数 M=3,具体的重量和价格 如表 4-1 所示。

表 4-1 每个部件的重量和价格

		n					
		1		2		3	
		w_1	c_1	w_2	c_2	w_3	c_3
m	1	1	1	2	2	3	3
	2	3	3	2	2	1	1
	3	2	2	2	2	2	2

构造该实例的解空间树如图 4-1 所示。

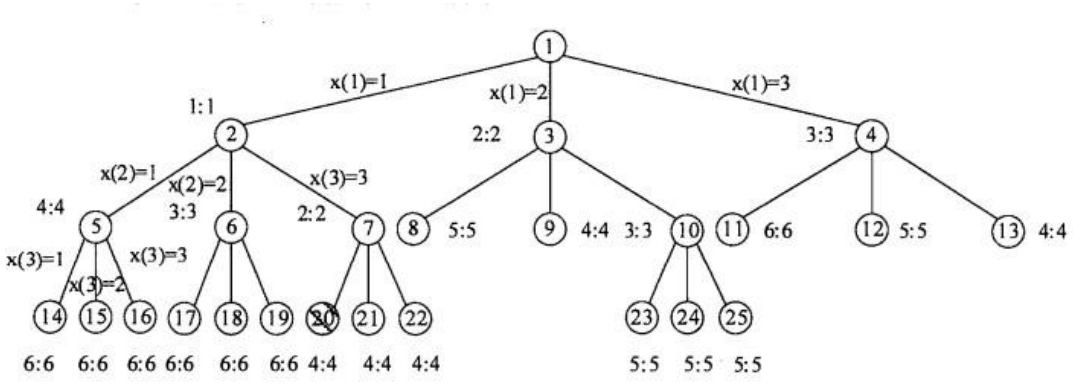


图 4-1 中结点编号表示生成该结点的顺序。边上的编号表示哪个部件选择哪个厂商， 如 X(2)=1，表示第 2 个部件来自厂商 1。结点旁边的两个数字表示当前解或部分解对应的重量和价格，如 2: 2 表示重量为 2，价格为 2。从图 4-1 可以看出，最优解是结点 20 表示的解，即 X(1)=1, X(2)=3, X(3)=1,即第 1 个部件来自厂商 1，第 2 个部件来自厂商 3,第 3 个部件

来自厂商1,总的价格和重量分别为4和4。当然,本实例的最优解还可 以是 X(1)=1,X(2)=3,X(3)=2 和 X(1)=1, X(2)=3, X(3)=3,分别对应解空间树上的21号和22号结点。

代码中的空(1)处是得到问题解之后，将搜索过程中产生的重量CW、价格CP和解X放到最终重量BESTW、价格BESTC和解BESTX中，因此空格(1)处填写BESTX[J] = X[J]。空(2)处的FOR循环是考虑第I个部件选择哪个厂商，因此J从0到M-1依次检查，此处应填J<M。对搜索过程中产生的重量CW、价格CP和解X的值进行设置，因此空(3) 处应填X[I]=J，表示第I个部件选择厂商J。空(4)是判断当前结点是否要扩展，若当前获得的价格比目前最优解更优，且重量没有超过当前得到的最优重量，即CP<=CC 且 CW<BESTW,则扩展当前结点，否则回溯。在回溯过程中，需要把原来选择的部件的价格和重量从搜索过程中产生的重量CW和价格CP中去掉,因此空(5)应填CP = CP - C[I][J]。

试题五

某大型商场内安装了多个简易的纸巾售卖机，自动出售 2 元钱一包的纸巾，且每次仅售出一包纸巾。纸巾售卖机的状态图如图 5-1 所示。

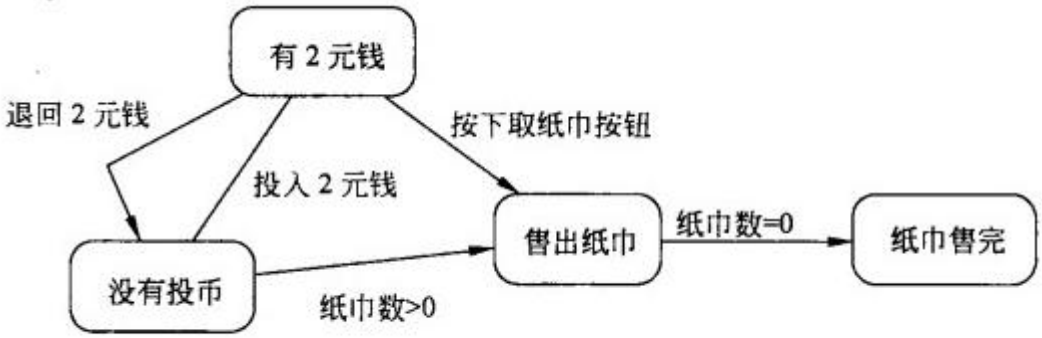


图 5-1　纸巾售卖机状态图

采用状态（STATE)模式来实现该纸巾售卖机，得到如图 5-2 所示的类图。其中类 STATE 为抽象类，定义了投币、退币、出纸巾等方法接口。类 SOLDSTATE、SOLDOUTSTATE> NOQUARTERSTATE 和 HASQUARTERSTATE 分别对应图 5-1 中纸巾售卖机的 4 种状态：售出纸巾、纸巾售完、没有投币、有 2 元钱。

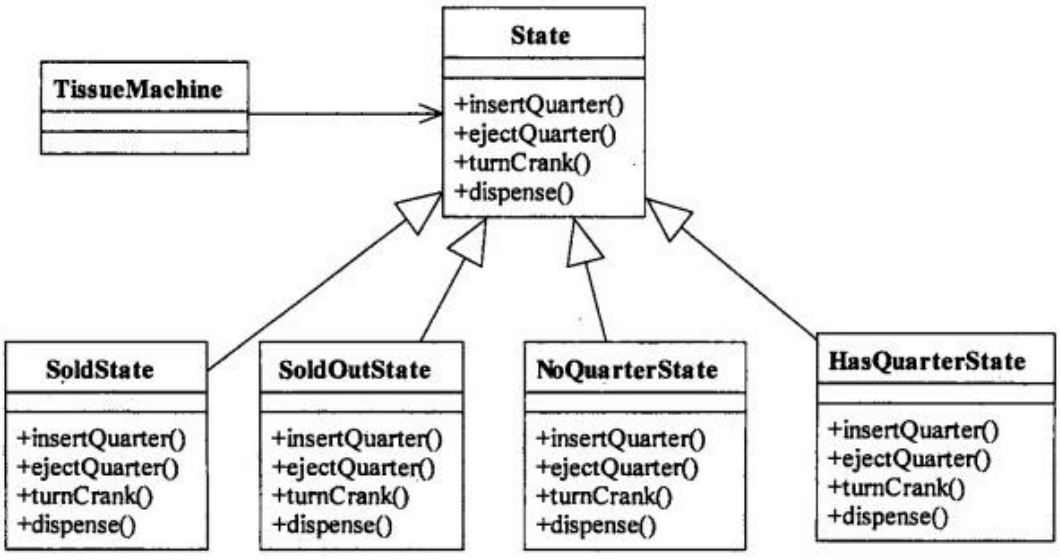


图 5-2　类图

【问题 1】

【C++代码】

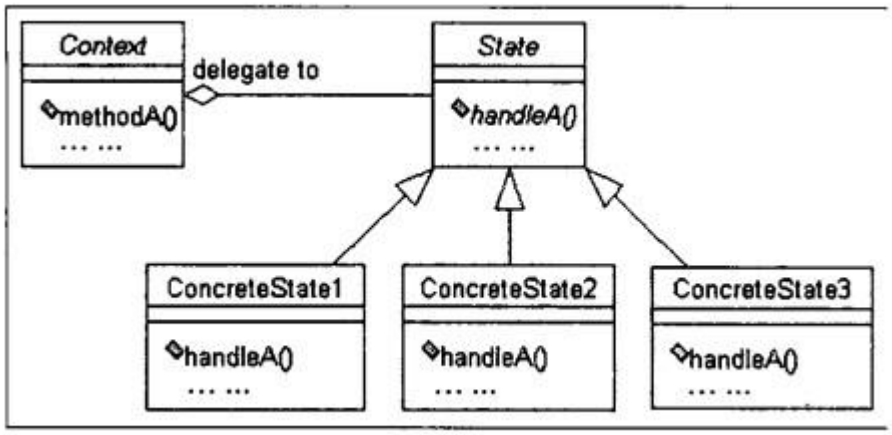
```
#include <iostream>
using namespace std;
// 以下为类的定义部分
class TissueMachine;                    // 类的提前引用

class State {
public:
   virtual void insertQuarter() = 0;//投币
   virtual void ejectQuarter() = 0; //退币
   virtual void turnCrank()= 0;      //按下“出纸巾”按钮
   virtual void dispense() = 0;      //出纸巾
};
/* 类 SoldOutState、NoQuarterState、HasQuarterState、SoldState 的定义省略，
每个类中均定义了私有数据成员 TissueMachine* tissueMachine; */
class TissueMachine {
private:
   __(1)__ *soldOutState, *noQuarterState, *hasQuarterState,*soldState,
*state ;
   int count;                           //纸巾数
public:
  TissueMachine(int numbers);
  void setState(State* state);
  State* getHasQuarterState();
  State* getNoQuarterState();
  State* getSoldState();
  State* getSoldOutState();
  int getCount();
  //其余代码省略
};
```

```
//以下为类的实现部分
void NoQuarterState ::insertQuarter() {
tissueMachine->setState(   (2)   );
}
void HasQuarterState ::ejectQuarter() {
tissueMachine->setState(   (3)   );
}
void SoldState ::dispense() {
    if(tissueMachine->getCount() > 0) {
      tissueMachine->setState(   (4)   );
}
else {
      tissueMachine->setState(   (5)   );
}
}  //其余代码省略
```

(1)State
(2)tissueMachine->getHasQuarterState()
(3)tissueMachine->getNoQuarterState()
(4)tissueMachine->getNoQuarterState()
(5)tissueMachine->getSoldOutState()

本题考查状态（State)模式的概念及应用。
状态模式是一种对象的行为型模式，允许一个对象在其内部状态改变时改变它的行为，对象看起来似乎修改了它的类。状态模式的类图如下所示：



状态模式主要解决的是控制一个对象转换的条件表达式过于复杂的情况。把状态的判断逻辑转移到表示不同状态的一系列类当中，可以把复杂的判断逻辑简化。状态模式的好处是将与特定状态相关的行为局部化，并且将不同状态的行为分割开来。

题目利用状态模式来实现一个简易的纸巾售卖机。售卖机的状态转换图已经在题目中给出，类 SoldState、SoldOutState、NoQuarterState 和 HasQuarterState 分别用来表示售卖机的 4 种不同状态，对应于状态模式中的 ConcreteState1，...ConcreteStateN。题目所设置的填空，主要集中在状态转换上。因此解答该题时，要求在理解状态模式内涵的基础上，依据纸巾售卖机的状态转换原则，给出正确的状态设置。

空(1)出现在类 TissueMachine 的数据成员定义部分。状态模式封装了状态的转换过程，但是它需要枚举可能的状态，因此需要确定状态种类。因此在类 TissueMachine 中需定义出所有可能的状态对象。根据所给出的对象名称及说明中的描述，可知(1)处应填入的类名为 State。

空(2)～(5)都是与状态转换相关的，要求填写类 TissueMachine 中的方法 setState 在不同调用处的实际参数。根据方法的名称及调用方式，可以推断出这个方法的功能就是设置自动售卖机的当前状态。要填出这些空，只要对照图 5-1 的状态转换图，根据状态转换的条件确定出当前状态及下一状态即可。

空(2)出现在方法 insertQuarter 内，即给纸巾售卖机投入 2 元钱。根据状态图，“投入 2 元钱”之后，售卖机应转换到“有 2 元钱”的状态。“有 2 元钱”对应的状态的类为 ″HasQuarterState”，所以空(2)处应填类 HasQuarterState 的对象。由于 hasQuarterState 是类 TissueMachine 的私有数据成员，不能直接访问，所以只能通过调用相关的 get 方法来获取该对象。由此得出(2)应填 tissueMachine->getHasQuarterState()。

同理，空(3)表示的状态是从“有 2 元钱”状态，经历“退回 2 元钱”事件之后的状态，及“没有投币”状态。所以空(3)处应填 tissueMachine->getNoQuarterStat()。

空(4)和(5)处分别表示卖出一包纸巾之后，售卖机应该转换到的下一个状态。 这个跟售卖机中的纸巾数有关，如果还有纸巾，则转换到“没有投币”状态，如果没有纸巾了，则转换到“纸巾售完”状态，因此，空(4)处应填 tisSUeMachine->getNoQuarterState(),空(5)处应填 tissueMachine->getSoldOutState()。

试题六

某大型商场内安装了多个简易的纸巾售卖机，自动出售 2 元钱一包的纸巾，且每次仅售出一包纸巾。纸巾售卖机的状态图如图 6-1 所示。

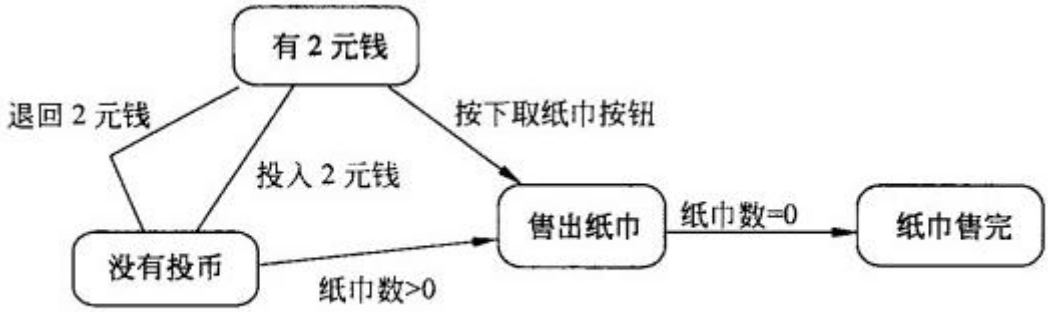


图 6-1　纸巾售卖机状态图

采用状态（State)模式来实现该纸巾售卖机，得到如图 6-2 所示的类图。其中类 State 为抽象类，定义了投币、退币、出纸巾等方法接口。类 SoldState、SoldOutState、NoQuarterState 和 HasQuarterState 分别对应图 6-1 中纸巾售卖机的 4 种状态：售出纸巾、纸巾售完、没有投币、有 2 元钱。

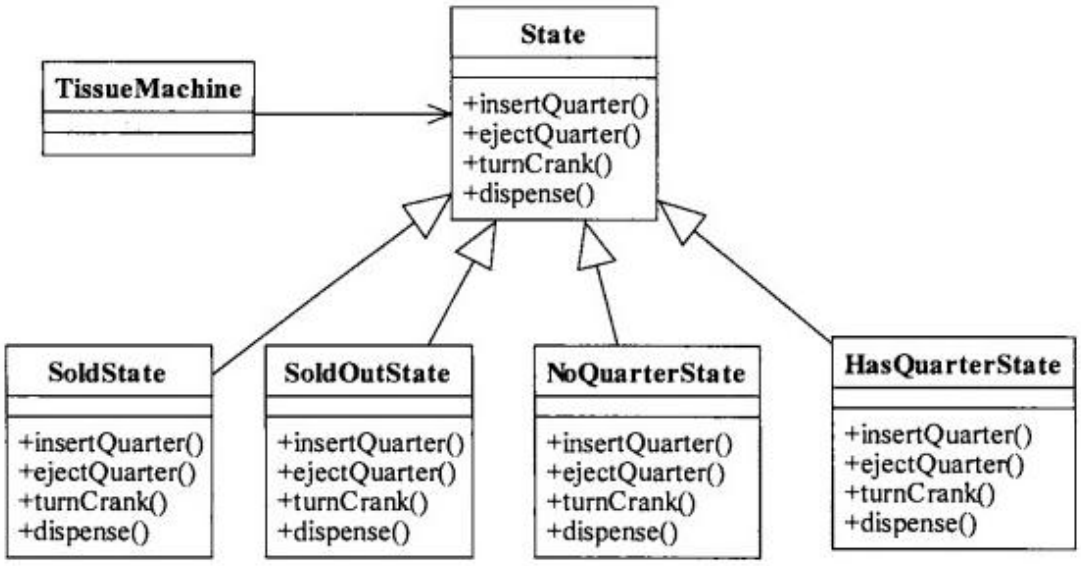


图 6-2　类图

【问题 1】

【Java 代码】

```
import java.util.*;
interface State {
  public void insertQuarter();        //投币
  public void ejectQuarter();         //退币
  public void turnCrank();            //按下“出纸巾”按钮
  public void dispense();             //出纸巾
}
class TissueMachine {
   (1)  soldOutState, noQuarterState, hasQuarterState, soldState,
state;
  state = soldOutState;
  int count = 0;                      //纸巾数
  public TissueMachine(int numbers) { /* 实现代码省略 */ }
public State getHasQuarterState()  { return hasQuarterState;  }
  public State getNoQuarterState()   { return noQuarterState;  }
  public State getSoldState()        { return soldState;       }
  public State getSoldOutState()     { return soldOutState;    }
  public int getCount()            { return count;          }
//其余代码省略
}

class NoQuarterState implements State {
  TissueMachine tissueMachine;
  public void insertQuarter() {
    tissueMachine.setState(  (2)  );
  }
//构造方法以及其余代码省略
}
class HasQuarterState implements State {
  TissueMachine tissueMachine;
  public void ejectQuarter() {
    tissueMachine.setState(  (3)  );
  }
//构造方法以及其余代码省略
}
class SoldState implements State {
  TissueMachine tissueMachine;
  public void dispense() {
    if(tissueMachine.getCount() > 0)  {
```

```
        tissueMachine.setState(__(4)__);
    } else {
        tissueMachine.setState(__(5)__); }
    }
}
```

(1)State

(2)tissueMachine.getHasQuarterStat()

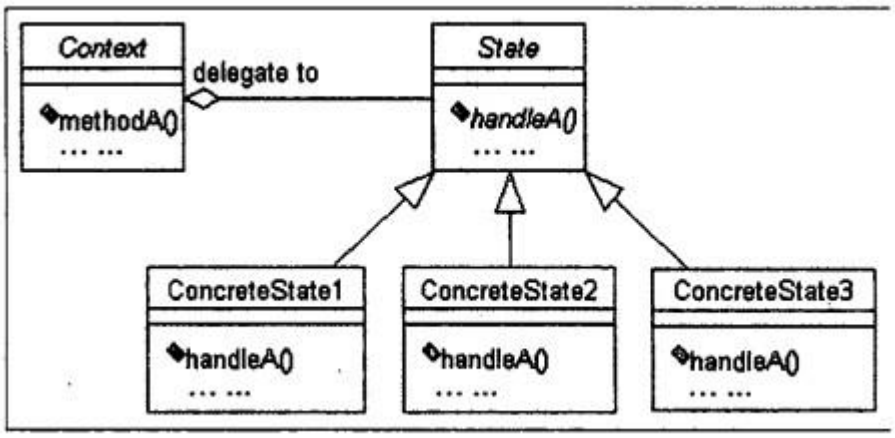
(3)tissueMachine.getNoQuarterState()

(4)tissueMachine.getNoQuarterState()

(5)tissueMachine.getSoldOutState()

本题考查状态(State)模式的概念及应用。

状态模式是一种对象的行为型模式，允许一个对象在其内部状态改变时改变它的行为，对象看起来似乎修改了它的类。状态模式的类图如下所示：



状态模式主要解决的是控制一个对象转换的条件表达式过于复杂的情况。把状态的判断逻辑转移到表示不同状态的一系列类当中，可以把复杂的判断逻辑简化。状态模式的好处是将与特定状态相关的行为局部化，并且将不同状态的行为分割开来。

题目利用状态模式来实现一个简易的纸巾售卖机。售卖机的状态转换图已经在题目中给出，类 SoldState、SoldOutState、NoQuarterState 和 HasQuarterState 分别用来表示售卖机的 4 种不同状态，对应于状态模式中的 ConcreteState1，...ConcreteStateN。题目所设置的填空，主要集中在状态转换上。因此解答该题时，要求在理解状态模式内涵的基础上，

依据纸巾售卖机的状态转换原则，给出正确的状态设置。

空(1)出现在类 TissueMachine 的数据成员定义部分。状态模式封装了状态的转换过程，但是它需要枚举可能的状态，因此需要实现确定状态种类。因此在类 TissueMachine 中需定义出所有可能的状态对象。根据所给出的对象名称及说明中的描述，可知(1)处 应填入的类名为 State。

空(2)～(5)要求填写类 TissueMachine 中的方法 setState 在不同调用处的实际参数。这里的一个难点在于题目中没有显示地给出方法 setState 的原型及语义，这要求考生根据面向对象程序设计风格及说明中给出的应用场合来推断 setState 的内涵及原型，主要是确定其参数列表。

在面向对象程序设计中，为了做到封装，通常都会把数据成员定义为私有的。私有的数据成员对象不能直接访问，因此在类中都会提供 2 组访问私有数据成员的方法，分别为 get...方法和 set...方法(…代表对应的数据成员名称)。get...方法表示获取私有数据成员的值，其返回值类型为对应的数据成员的类型；set...方法表示对数据成员进行赋值，所要赋的值通常通过参数传递进去，方法的返回值类型通常为 void。根据面向对象 程序设计的这些特点，以及状态模式的内涵及应用场合，可以推断出 setState 方法的功能就是设置纸巾售卖机的当前状态。纸巾售卖机在任一时刻只能处于一个唯一的状态， 由状态模式可知，纸巾售卖机的状态都是用状态对象表示的，由此就可以确定出，setState 方法的参数只要一个就可以了，就是表示纸巾售卖机下一状态的状态对象。

经过以上分析之后，可以明确空(2)～(5)空所填的内容都应与状态转换相关。因此要填充这些空,只要对照图 5-1 的状态转换图,根据状态转换的条件确定出当前状态及下一状态即可。

空(2)出现在方法 insertQuarter 内，即给纸巾售卖机投入 2 元钱。根据状态图，“投入 2 元钱”之后，售卖机应转换到“有 2 元钱”的状态。“有 2 元钱”对应的状态的类为“HasQuarterState”，所以空(2)处应填写类 HasQuarterState 的对象。由此得出（2)应填 tissueMachine.getHasQuarterState()。

同理，空（3)表示的状态是从“有 2 元钱”状态，经历“退回 2 元钱”事件之后 的状态，及“没有投币”状态。所以空（3)处应填 tissueMachine.getNoQuarterState()。

空（4)和（5)处分别表示卖出一包纸巾之后，售卖机应该转换到的下一个状态。 这个跟售卖机中的纸巾数有关，如果还有纸巾，则转换到“没有投币”状态，如果没有纸巾了，则转换到“纸巾售完”状态，因此，空(4)处应填 tissueMachine.getNoQuarterState()，空（5)处应填 tissueMachine.getSoldOutState()。