在程序执行过程中，Cache 与主存的地址映像由(1)。

(1)A.硬件自动完成　　B.程序员调度

C.操作系统管理　　D.程序员与操作系统协同完成

【答案】A

【解析】本题考查计算机系统基础知识。

Cache 的工作是建立在程序与数据访问的局部性原理上。经过对大量程序执行情况的结果分析：在一段较短的时间间隔内程序集中在某一较小的内存地址空间执行，这就是程序执行的局部性原理。同样，对数据的访问也存在局部性现象。

为了提高系统处理速度才将主存部分存储空间中的内容复制到工作速度更快的 Cache 中，同样为了提高速度的原因，Cache 系统都是由硬件实现的。

指令寄存器的位数取决于(2)。

(2)A.存储器的容量　B.指令字长　C.数据总线的宽度　D.地址总线的宽度

【答案】B

【解析】本题考查计算机系统基础知识。

指令寄存器是 CPU 中的关键寄存器，其内容为正在执行的指令，显然其位数取决于指令字长。

若计算机存储数据采用的是双符号位(00 表示正号、11 表示负号)，两个符号相同的数相加时，如果运算结果的两个符号位经(3)运算得 1，则可断定这两个数相加的结果产生了溢出。

(3)A.逻辑与　　B.逻辑或　　C.逻辑同或　　D.逻辑异或

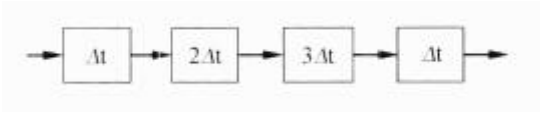
【答案】D

【解析】本题考查计算机系统基础知识。

当表示数据时并规定了位数后，其能表示的数值范围就确定了，在两个数进行相加运算的结果超出了该范围后，就发生了溢出。在二进制情况下，溢出时符号位将变反，即两个正数相加，结果的符号位是负数，或者两个负数相加，结果的符号位是正数。采用两个符号位时，溢出发生后两个符号位就不一致了，这两位进行异或的结果一定为 1。

某指令流水线由 4 段组成，各段所需要的时间如下图所示。连续输入 8 条指令时的吞吐

率(单位时间内流水线所完成的任务数或输出的结果数)为(4)。



(4)A.8/56△t　　B.8/32△t　　C.8/28△t　　D.8/24△t

【答案】C

【解析】本题考查计算机系统基础知识。

流水线的吞吐率指的是计算机中的流水线在特定的时间内可以处理的任务或输出数据的结果数量。流水线的吞吐率可以进一步分为最大吞吐率和实际吞吐率。该题目中要求解的是实际吞吐率，以流水方式执行 8 条指令的执行时间是 28△t，因此吞吐率为 8/28△t。

(5)不是 RISC 的特点。

(5)A.指令种类丰富　B.高效的流水线操作　C.寻址方式较少　D.硬布线控制

【答案】A

【解析】本题考查计算机系统基础知识。

RISC(Reduced Instruction Set Computer,精简指令集计算机)的主要特点是重叠寄存器窗口技术；优化编译技术。RISC 使用了大量的寄存器，如何合理分配寄存器、提高寄存器的使用效率及减少访存次数等，都应通过编译技术的优化来实现；超流水及超标量技术。为了进一步提高流水线速度而采用的技术；硬布线逻辑与微程序相结合在微程序技术中。

若某计算机字长为 32 位，内存容量为 2GB,按字编址，则可寻址范围为(6)。

(6)A.1024M　　B.1GB　　C.512M　　D.2GB

【答案】C

【解析】本题考查计算机系统基础知识。

内存容量 2GB=2*1024*1024*1024*8 位，按字编址时，存储单元的个数为 2*1024*1024*1024*8/32=512*1024*1024,即可寻址范围为 512MB。

下列网络攻击行为中，属于 DoS 攻击的是(7)。

(7)A.特洛伊木马攻击　　B.SYN Flooding 攻击　C.端口欺骗攻击 D.IP 欺骗攻击

【答案】B

【解析】本试题考查网络安全相关知识。

特洛伊木马是附着在应用程序中或者单独存在的一些恶意程序,它可以利用网络远程控制网络另一端的安装有服务端程序的主机，实现对被植入了木马程序的计算机的控制，或者窃取被植入了木马程序的计算机上的机密资料。

拒绝服务攻击通过网络的内外部用户来发动攻击。内部用户可以通过长时间占用系统的内存、CPU 处理时间使其他用户不能及时得到这些资源，而引起拒绝服务攻击；外部黑客也可以通过占用网络连接使其他用户得不到网络服务。SYN Flooding 攻击以多个随机的源主机地址向目的路由器发送 SYN 包，在收到目的路由器的 SYN ACK 后并不回应，于是目的路由器就为这些源主机建立大量的连接队列，由于没有收到 ACK一直维护着这些队列，造成了资源的大量消耗而不能向正常请求提供服务，甚至导致路由器崩溃。服务器要等待超时才能断开已分配的资源，所以 SYN Flooding 攻击是一种 DoS 攻击。

端口欺骗攻击是采用端口扫描找到系统漏洞从而实施攻击。

IP 欺骗攻击是产生的 IP 数据包为伪造的源 IP 地址，以便冒充其他系统或发件人的身份。

PKI 体制中，保证数字证书不被篡改的方法是(8)。

(8)A.用 CA 的私钥对数字证书签名　　B.用 CA 的公钥对数字证书签名

C.用证书主人的私钥对数字证书签名　　D.用证书主人的公钥对数字证书签名

【答案】A

【解析】本题考查 PKI 体制。

PKI 体制中，为保障数字证书不被篡改而且要发送到证书主人手中，需要用 CA 的私钥对数字证书签名，防伪造，不可抵赖。

下列算法中，不属于公开密钥加密算法的是(9)。

(9)A.ECC　　B.DSA　　C.RSA　　D.DES

【答案】D

【解析】本题考查加密算法的基础知识。

常用的加密算法依据所使用的秘钥数分为单钥和双钥加密体制,也称私钥和公钥加密算法。ECC、DSA 和 RSA 都属于公开密钥加密算法，DES 是典型的私钥加密体制。

矢量图是常用的图形图像表示形式，(10)是描述矢量图的基本组成单位。

(10)A.像素　　B.像素点　　C.图元　　D.二进制位

【答案】C

【解析】本题考查多媒体方面的基础知识。

矢量图形是用一系列计算机指令来描述和记录的一幅图的内容，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的曲面、光照、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。

视频信息是连续的图像序列，(11)是构成视频信息的基本单元。

(11)A.帧　　B.场　　C.幅　　D.像素

【答案】A

【解析】本题考查多媒体方面的基础知识。

视频信息是指活动的、连续的图像序列。一幅图像称为一帧，帧是构成视频信息的基本单元。

以下多媒体素材编辑软件中，(12)主要用于动画编辑和处理。

(12)A.WPS　　B.Xara3D　　C.PhotoShop　　D.Cool Edit Pro

【答案】B

【解析】本题考查多媒体编辑软件方面的知识。

多媒体编辑软件分为：文本工具、图形/图像工具、动画工具、视频工具、音频工具和播放工具。选项A“WPS”属于文本工具类软件，主要用于文字编辑和处理；选项B“Xara3D”属于动画工具类软件，主要用于动画编辑和处理；选项C“PhotoShop”属于图形/图像工具类软件，主要用于显示图形/图像、图形/图像编辑、图像压缩、围像捕捉、图形/图像素材库；选项D“Cool Edit Pro”属于音频工具类软件，主要用于音频播放、音频编辑、音频录制和声音素材库4个功能。

为说明某一问题，在学术论文中需要引用某些资料。以下叙述中，(13)是不正确的。

(13)A.既可引用发表的作品，也可引用未发表的作品

B.只能限于介绍、评论作品

C.只要不构成自己作品的主要部分，可适当引用资料

D.不必征得原作者的同意，不需要向他支付报酬

【答案】A

【解析】本题考查知识产权方面的基础知识。

选项A“既可引用发表的作品，也可引用未发表的作品”的说法显然是错误的。因为，为说明某一问题，在学术论文中需要引用某些资料必须是已发表的作品，但只能限于介绍、评论作品，只要不构成自己作品的主要部分，可适当引用资料，而不必征得原作者的同意，不需要向他支付报酬。

以下作品中，不适用或不受著作权法保护的是(14)。

(14)A.某教师在课堂上的讲课

B.某作家的作品《红河谷》

C.最高人民法院组织编写的《行政诉讼案例选编》

D.国务院颁布的《计算机软件保护条例》

【答案】D

【解析】本题考查知识产权方面的基础知识。

选项D“国务院颁布的《计算机软件保护条例》”的说法显然是错误的。因为，国务院颁布的《计算机软件保护条例》是国家为了管理需要制定的政策法规，故不适用著作权法保护。

以下关于数据流图中基本加工的叙述，不正确的是(15)。

(15)A.对每一个基本加工，必须有一个加工规格说明

B.加工规格说明必须描述把输入数据流变换为输出数据流的加工规则

C.加工规格说明必须描述实现加工的具体流程

D.决策表可以用来表示加工规格说明

【答案】C

【解析】本题考查结构化分析方法的基础知识。

分层的数据流图是结构化分析方法的重要组成部分。对数据流图中的每个基本加工，需

要有一个加工规格说明，描述把输入数据流变换为输出数据流的加工规则，但不需要描述实现加工的具体流程。可以用结构化语言、判定表和判定树来表达基本加工。

在划分模块时，一个模块的作用范围应该在其控制范围之内。若发现其作用范围不在其控制范围内，则(16)不是适当的处理方法。

(16)A.将判定所在模块合并到父模块中，使判定处于较高层次

B.将受判定影响的模块下移到控制范围内

C.将判定上移到层次较高的位置

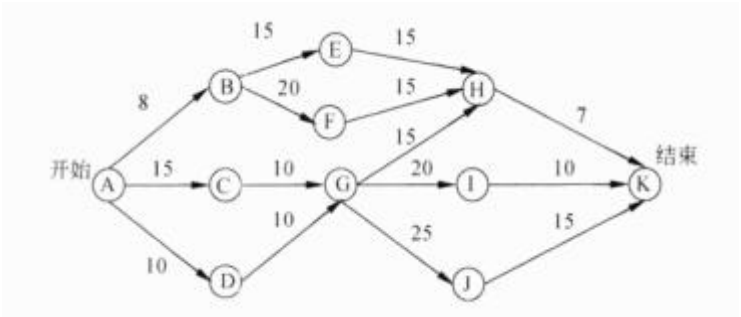
D.将父模块下移，使该判定处于较高层次

【答案】D

【解析】本题考查软件设计的基础知识。

模块的控制范围包括模块本身及其所有的从属模块。模块的作用范围是指模块一个判定的作用范围，凡是受这个判定影响的所有模块都属于这个判定的作用范围.，原则上一个模块的作用范围应该在其控制范围之内，若没有，则可以将判定所在模块合并到父模块中，使判定处于较高层次。

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，则里程碑(17)在关键路径上。若在实际项目进展中，活动 AD 在活动 AC 开始 3 天后才开始，而完成活动 DG 过程中，由于有临时事件发生，实际需要 15 天才能完成，则完成该项目的最短时间比原计划多了(18)天。



(17)A.B　　B.C　　C.D　　D.I

(18)A.8　　B.3　　C.5　　D.6

【答案】B　B

【解析】本题考查软件项目管理的基础知识。

根据关键路径法，计算出关键路径为 A—C—G—J—K，关键路径长度为 65。因此里程碑 C 在关键路径上，而里程碑 B、D 和 I 不在关键路径上。

若完成活动 DG 需要 15 天，则相当于 A--D--G--J--K 也是一个关键路径，而且活动 AD 推迟了三天才能完成，此时，完成项目的最短时间应该是 68 天，比原来的最短时间 65 天多了 3 天。

针对“关键职员在项目未完成时就跳槽”的风险，最不合适的风险管理策略是(19)。

(19)A.对每一个关键性的技术人员，要培养后备人员

B.建立项目组，以使大家都了解有关开发活动的信息

C.临时招聘具有相关能力的新职员

D.对所有工作组织细致的评审

【答案】C

【解析】本题考查软件项目管理的基础知识。

软件开发过程中不可避免会遇到风险，有效地管理软件风险对项目管理具有重要的意义。对不同的风险采取不同的风险管理策略。如对关键职员在项目未完成时就跳槽的风险，可以通过培养后备人员、让项目组人员了解开发信息、评审开发工作等来降低风险。通过临时招聘新职员，即使新职员具有相关的能力，由于对项目的开发进展、团队组成等多种情况不了解，并不能很好地降低风险。

程序运行过程中常使用参数在函数(过程)间传递信息，引用调用传递的是实参的(20)。

(20)A.地址　　B.类型　　C.名称　　D.值

【答案】A

【解析】本题考查程序语言基础知识。

进行函数调用时，常需要在调用环境中的数据传递给被调用函数，作为输入参数由被调用函数处理，基本的调用方式为值调用(或传值调用)和引用调用。其中，值调用方式下是将实参的值单向地传递给被调用函数的形参，引用调用方式下通过将实参的地址传递给形参，在被调用函数中通过指针实现对实参变量数据的间接访问和修改，从而达到将修改后的值“传回来”的效果。

己知文法 G: S→A0|B1,A→S1|1, B→S0|0,其中 S 是开始符号。从 S 出发可以推导出(21)。

(21)A.所有由 0 构成的字符串　　B.所有由 1 构成的字符串

C.某些 0 和 1 个数相等的字符串　　D.所有 0 和 1 个数不同的字符串

【答案】C

【解析】本题考查程序语言基础知识。

用文法表示语言的语法规则时，推导是产生语言句子的基本方式。以题 H 中的文法为例，推导出 1010 的过程为 S→A0→S10→A010→1010，推导出 0110 的过程为 S→A0→S10→B110→0110,对于 0000、1111、1100、0011 等则推导不出。因为由 S 先推导出 A0 后，再去推导 A 则必然产生一个与 0 相邻(在 0 的左边)的 1，而由 S 先推导出 B1，则下一步必然要推导出一个与 1 相邻(在 1 的左边)的 0。这保证了当 1 出现时，马上就会出现 0,或者反之，且 0 和 1 的距离很近。分析更多的例子发现，仅有“某些 0 和 1 个数相等的字符串”是正确的。

算术表达式 a+(b-C)*d 的后缀式是(22) (-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例)。

(22)A.b c - d * a +　　B.a b c - d * +　　C.a b + c - d *　　D.a b c d - * +

【答案】B

【解析】本题考查程序语言基础知识。

后缀式的特点是将运算符号写在运算数的后面。对于表达式，其计算次序是相减、相乘、相加，其后缀式为“abc-d*+”。

假设系统采用 PV 操作实现进程同步与互斥，若有 n 个进程共享一台扫描仪，那么当信号量 S 的值为-3 时，表示系统中有(23)个进程等待使用扫描仪。

(23)A.0　　B.n-3　　C.3　　D.n

【答案】C

【解析】本题考查操作系统 PV 操作方面的基本知识。

系统采用 PV 操作实现进程的同步与互斥，当执行一次 P 操作表示申请一个资源，信号量 S 减 1，如果 S<0,其绝对值表示等待该资源的进程数。本题信号量 S 的值为-3,表示系统中有 3 个等扫描仪的进程。

假设段页式存储管理系统中的地址结构如下图所示，则系统中(24)。

(24)A.页的大小为 4K，每个段的大小均为 4096 个页，最多可有 256 个段

B.页的大小为 4K,每个段最大允许有 4096 个页，最多可有 256 个段

C.页的大小为 8K,每个段的大小均为 2048 个页，最多可有 128 个段

D.页的大小为 8K,每个段最大允许有 2048 个页，最多可有 128 个段

【答案】B

【解析】本题考查操作系统页式存储管理方面的基础知识。

从图中可见，页内地址的长度是 12 位，212=4096,即 4K;页号部分的地址长度是 12 位，每个段最大允许有 4096 个页;段号部分的地址长度是 8 位，28=256,最多可有 256 个段。

某文件管理系统采用位示图(bitmap)记录磁盘的使用情况。如果系统的字长为 32 位，磁盘物理块的大小为 4MB，物理块依次编号为：0、1、2、位示图字依次编号为：0、1、2、那么 16385 号物理块的使用情况在位示图中的第(25)个字中描述;如果磁盘的容量为 1000GB,那么位示图需要(26)个字来表示。

(25)A.128　　B.256　　C.512　　D.1024

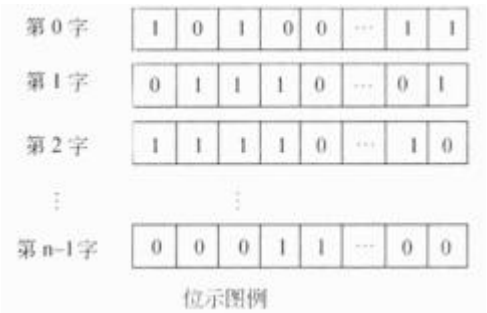
(26)A.1200　　B.3200　　C.6400　　D.8000

【答案】C　D

【解析】本题考查操作系统文件管理方面的基本知识。

文件管理系统是在外存上建立一张位示图(bitmap),记录文件存储器的使用情况。

每一位对应文件存储器上的一个物理块,取值 0 和 1 分别表示空闲和占用,如下图所示。



位示图例

由于系统中字长为 32 位，所以每个字可以表示 32 个物理块的使用情况。又因为文件存储器上的物理块依次编号为：0、1、2、位示图表示物理块的情况如下，从下图可见，16385 号物理块应该在位示图的第 512 个字中描述。

又因为磁盘物理块的大小为 4MB,1GB=1024M=256 个物理块，需要 8 个字表示，故磁盘的容量为 1000GB，那么位示图需要 1000×8=8000 个字表示。

假设系统中有三类互斥资源 R1、R2 和 R3,可用资源数分别为 10、5 和 3。在 T0 时刻系统中有 P1、P2、P3、P4 和 P5 五个进程，这些进程对资源的最大需求量和已分配资源数如下表所示，此时系统剩余的可用资源数分别为(27)。如果进程按(28)序列执行，那么系统状态是安全的。

资源 / 进程	最大需求量 R1	R2	R3	已分配资源数 R1	R2	R3
P1	5	3	1	1	1	1
P2	3	2	0	2	1	0
P3	6	1	1	3	1	0
P4	3	3	2	1	1	1
P5	2	1	1	1	1	0

(27)A.1、1 和 0　　B.1、1 和 1　　C.2、1 和 0　　D.2、0 和 1

(28)A.P1→P2→P4→P5→P3　　B.P5→P2→P4→P3→P1

C.P4→P2→P1→P5→P3　　D.P5→P1→P4→P2→P3

【答案】D　B

【解析】

试题(27)的正确答案是 D。因为，初始时系统的可用资源数分别为 10、5 和 3。在 T0 时刻已分配资源数分别为 8、5 和 2,因此系统剩余的可用资源数分别为 2、0 和 1。

试题(28)的正确答案是 B。安全状态是指系统能按某种进程顺序(P1，P2,…，Pn)，来为每个进程 Pi 分配其所需的资源，直到满足每个进程对资源的最大需求，使每个进程都可以顺利完成。如果无法找到这样的一个安全序列，则称系统处于不安全状态。

本题进程的执行序列已经给出，我们只需将四个选项按其顺序执行一遍，便可以判断出现死锁的三个序列。

资源 / 进程	最大需求量 R1	R2	R3	已分配资源数 R1	R2	R3	尚需资源数 R1	R2	R3
P1	5	3	1	1	1	1	4	2	0
P2	3	2	1	2	1	0	1	1	1
P3	6	1	1	3	1	0	3	0	1
P4	3	3	2	1	1	1	2	2	1
P5	2	1	1	1	1	0	1	0	1

P1→P2→P4→P5→P3 是不安全的序列。因为在该序列中，进程 P1 先运行，P1 尚需资源数为(4，2，0)，假设将资源 R1 分配 2 台给进程 P1，则系统剩余的可用资源数为(0，0，1)，将导致系统所有的进程都不能作上能完成标志“True”。

P5→P2→P4→P3→P1 是安全的序列。因为所有的进程都能作上能完成标志“True”，

如下表所示。

资源 / 进程	可用资源数 R1 R2 R3	已分配资源数 R1 R2 R3	尚需资源数 R1 R2 R3	可用+已分 R1 R2 R3	能否完成标志
P5	2 0 1	1 1 0	1 0 1	3 1 1	True
P2	3 1 1	2 1 0	1 1 1	5 2 1	True
P4	5 2 1	1 1 1	2 2 1	6 3 2	True
P3	6 3 2	3 1 0	3 0 1	9 4 2	True
P1	9 4 2	1 1 1	4 2 0	10 5 3	True

P5→P2→P4→P3→P1 具体分析如下： .

①进程 P5 运行，系统剩余的可用资源数为(2，0，1)，P5 尚需资源数为(1，0，1)，系统可进行分配，故进程 P5 能作上能完成标志“True”，释放 P5 占有的资源数(1，1，0)，系统可用资源数为(3，1，1)。

②进程 P2 运行，系统剩余的可用资源数为(3，1，1)，P2 尚需资源数为(1，1，1),系统可进行分配，故进程 P2 能作上能完成标志“True”，释放 P2 占有的资源数(2，1，0)，系统可用资源数为(5，2，1)。

③进程 P4 运行，系统剩余的可用资源数为(5，2，1)，P4 尚需资源数为(2，2，1)，系统可进行分配，故进程 P4 能作上能完成标志“True”，释放 P4 占有的资源数(1，1，1)，系统可用资源数为(6，3，2)。

④进程 P3 运行，系统剩余的可用资源数为(6，3，2)，P3 尚需资源数为(3，0，1)，系统可进行分配，故进程 P3 能作上能完成标志“True”，释放 P3 占有的资源数(3，1，0)，系统可用资源数为(9，4，2)。

⑤进程 P1 运行，系统剩余的可用资源数为(9，4，2)，P1 尚需资源数为(4，2，0)，系统可进行分配，故进程 P1 能作上能完成标志“True”，释放 P1 占有的资源数(1，1，1)，系统可

用资源数为(10，5，3)。

P4→P2→P1→P5→P3 是不安全的序列。因为在该序列中，进程 P4 先运厅，P4 尚需资源数为(2，2，1)，假设将资源 R1 分配 2 台给进程 P4,则系统剩余的可用资源数为(0,0，1)，将导致系统所有的进程都不能作上能完成标志“True”。

P5→P1→P4→P2→P3 是不安全的序列。因为在该序列中，进程 P5 先运行，系统剩余的可用资源数为(2，0，1),P5 尚需资源数为(1，0，1)，系统可进行分配，故进程 P5 能作上能完成标志“True”，释放 P5 占有的资源数(1，1，0)，系统可用资源数为(3,1，1)。进程 P1 运行，P1 尚需资源数为(4,2,0)，假设将资源 R1 分配 3 台给进程 P1，则系统剩余的可用资源数为(0，1，1)，将导致系统中的进程 P1、P2、P3 和 P4 都不能作上能完成标志“True”。

(29)开发过程模型最不适用于开发初期对软件需求缺乏准确全面认识的情况。

(29)A.瀑布　　B.演化　　C.螺旋　　D.增量

【答案】A

【解析】本题考查软件过程模型的基础知识。

瀑布模型将软件生存周期各个活动规定为线性顺序连接的若干阶段的模型,规定了由前至后，相互衔接的固定次序，如同瀑布流水，逐级下落。这种方法是一种理想的现象开发模式，缺乏灵活性，特别是无法解决软件需求不明确或不准确的问题。演化模型从初始的原型逐步演化成最终软件产品，特别适用于对软件需求缺乏准确认识的情况。螺旋将瀑布模型与快速原型模型结合起来，并且加入两种模型均忽略了的风险分析，适用于复杂的大型软件。增量开发是把软件产品作为一系列的增量构件来设计、编码、集成和测试，可以在增量开发过程中逐步理解需求。

(30)不是增量式开发的优势。

(30)A.软件可以快速地交付

B.早期的增量作为原型，从而可以加强对系统后续开发需求的理解

C.具有最高优先级的功能首先交付，随着后续的增量不断加入，这就使得更重要的功能得到更多的测试

D.很容易将客户需求划分为多个增量

【答案】D

【解析】本题考查过程模型的基础知识。

增量开发是把软件产品作为一系列的增量构件来设计、编码、集成和测试。每个构件由多个相互作用的模块构成，并且能够完成特定的功能。其优点包括：能在较短时间内向用户提交可完成-些有用的工作产品；逐步增加产品的功能可以使用户有较充裕的时间学习和适应新产品；项目失败的风险较低；优先级高的服务首先交付，使得最重要的系统服务将接受最多的测试。

在对程序质量进行评审时，模块结构是一个重要的评审项，评审内容中不包括(31)。

(31)A.数据结构　　B.数据流结构

C.控制流结构　　D.模块结构与功能结构之间的对应关系

【答案】A

【解析】本题考查软件质量的基础知识。

程序质量评审通常是从开发者的角度进行,与开发技术直接相关,考虑软件本身的结构、与运行环境的接口以及变更带来的影响等。其中，软件结构包括功能结构、功能的通用性、模块的层次性、模块结构和处理过程的结构，而模块结构包括控制流结构、数据流结构、模块结构与功能结构之间的对应关系。

SEI 能力成熟度模型(SEICMM)把软件开发企业分为 5 个成熟度级别，其中(32)重点关注产品和过程质量。

(32)A.级别 2:重复级　B.级别 3:确定级　C.级别 4:管理级　D.级别 5:优化级

【答案】C

【解析】本题考查软件过程和软件过程改进的基础知识。

CMM 是指软件开发能力成熟度模型，该模型给出了从混乱的个别的过程达到成熟的规范化过程的一个框架，分成 5 个等级，从 1 级到 5 级成熟度逐步提高。级别 1 为初始级，特点是混乱和不可预测；级别 2 为重复级级别，特点是项目得到管理监控和跟踪，有稳定的策划和产品基线；级别 3 为确定级级别，通过软件过程的定义和制度化确保对产品质量的控制；级别 4 为管理级级别，特点是产品质量得到策划，软件过程基于度量的跟踪；级别 5 为优化级，特点是持续的过程能力改进。

系统可维护性的评价指标不包括(33)。

(33)A.可理解性　　B.可测试性　　C.可移植性　　D.可修改性

【答案】C

【解析】本题考查软件质量的基础知识。

软件的可维护性是指纠正软件系统出现的错误和缺陷，以及为满足新的要求进行修改、扩充或压缩的容易程度，是软件开发阶段各个时期的关键目标。其中，可理解性、可测试性和可修改性是衡量可维护性的重要指标。

逆向工程从源代码或U标代码中提取设计信息，通常在原软件生命周期的(34)阶段进行。

(34)A.需求分析　　B.软件设计　　C.软件实现　　D.软件维护

【答案】D

【解析】本题考查软件工程的基础知识。

逆向工程从详细的源代码实现中抽取抽象规格说明，一般来说是在原软件交付用户使用之后进行的，即在原软件的维护阶段进行。

一个程序根据输入的年份和月份计算该年中该月的天数，输入参数包括年份(正整数)、月份(用1～12表示)。若用等价类划分测试方法进行测试，则(35)不是一个合适的测试用例(分号后表示测试的输出)。

(35)A.(2013,1;31)　B.(0,1;‘错误’)　C.(0,13;‘错误’)　D.(2000,-1;‘错误’)

【答案】C

【解析】本题考查软件测试的基础知识。

常用的测试技术包括白盒测试和黑盒测试。白盒测试是利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所在逻辑路径进行测试，又称为结构测试或逻辑驱动测试。黑盒测试根据程序的需求规格兑明书，检查程序的功能是否符合它的功能说明。

等价类划分是一类黑盒测试技术，该方法把输入数据分为若干个等价类，包括有效的和无效的等价类。基于等价类设计测试用例时，每个测试用例至多覆盖一个无效等价类，选项C包含两个无效等价类，故不是一个好的测试用例。

(36)不是单元测试主要检查的内容。

(36)A.模块接口　　B.局部数据结构　　C.全局数据结构　　D.重要的执行路径

【答案】C

【解析】本题考查软件测试的基础知识。

单元测试又称为模块测试，是针对软件设计的最小单元(程序模块)，进行正确性检验的测试。其目的在于发现个模块内不可能存在的各种问题和错误。单元测试需要从程序的内部结构出发设计测试用例。模块可以单独进行单元测试。单元测试测试以下几个方面：模块接口、局部数据结构、执行路径、错误处理和边界。

在领域类模型中不包含(37)。

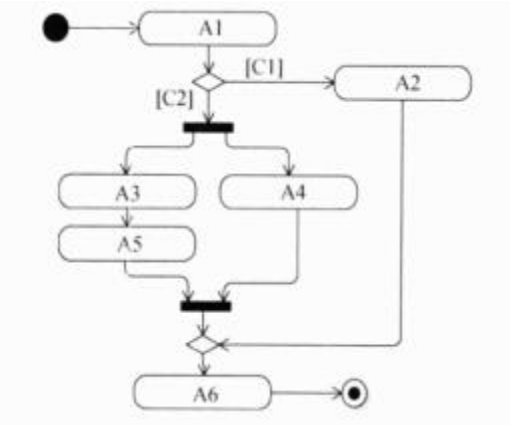
(37)A.属性　　B.操作　　C.关联　　D.领域对象

【答案】D

【解析】本题考查面向对象的基本知识。

定义领域模型是面向对象分析的关键步骤之一。领域模型是从按对象分类的角度来创建对象领域的描述，包括定义概念、属性和重要的关联，其结果用一组显示领域概念和对象的图形一一类图来组织，图中还包括多重性、关联关系、泛化/特化关系以及聚合关系等。

在执行如下所示的 UML 活动图时，能同时运行的最大线程数为(38)。



(38)A.4　　B.3　　C.2　　D.1

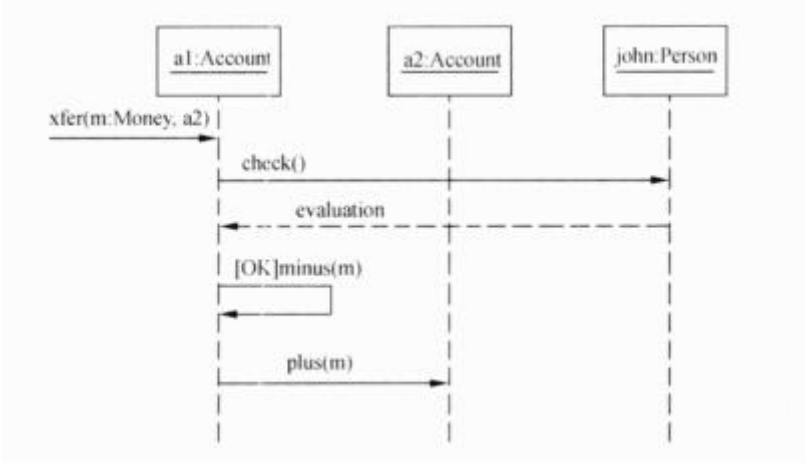
【答案】C

【解析】本题考查统一建模语言(UML)的基本知识。

UML 活动图用于构建系统的活动。建模用例执行过程中对象如何通过消息相互交互，将系统作为一个整体或者几个子系统进行考虑。对象在运行时可能会存在两个或多个并发运行的控制流，为了对并发控制流进行建模，UML 中引入同步的概念，用同步棒——黑色粗线条

表示并发分支与汇合。

下图所示的 UML 序列图中，(39)表示返回消息，Account 应该实现的方法有(40)。



(39)A.xfer　　B.check　　C.evaluation　　D.minus

(40)A.xfer()　　B.xfer()、plus()和 minus()

C.check()、plus()和 minus()　　D.xfer()、evaluation()、plus()和 minus()

【答案】C　B

【解析】本题考查统一建模语言(UML)的基本知识。

UML 序列图(Sequence Diagram)以二维图的形式显示对象之间交互的图，纵轴自上而下表示时间，横轴表示要交互的对象，主要体现对象间消息传递的时间顺序，强调参与交互的对象及其间消息交互的时序。序列图中包括的建模元素主要有：活动者(Actor)、对象(Object)、生命线(Lifeline)、控制焦点(Focus of control)和消息(Message)等。其中对象名标有下划线；生命线表示为虚线，沿竖线向下延伸；消息在序列图中标记为箭头；控制焦点由薄矩形表示。

消息是从一个对象的生命线到了一个对象生命线的箭头，用从上而下的时间顺序来安排。一般分为同步消息(→)，异步消息()和返回消息()。本题图中 evaluation 为返回消息，其他为同步消息。a1 和 a2 均为 Account 对象，所以 Account 应该实现了 xfer()、minus()和 plus()方法，Person 应该实现 check()方法。

在面向对象技术中，(41)定义了超类和子类之间的关系，子类中以更具体的方式实现从父类继承来的方法称为(42)，不同类的对象通过(43)相互通信。

(41)A.覆盖　　B.继承　　C.信息　　D.多态

(42)A.覆盖　　B.继承　　C.消息　　D.多态

(43)A.覆盖　　B.继承　　C.消息　　D.多态

【答案】B　A　C

【解析】本题考查面向对象技术的基础知识。

在面向对象技术中，继承关系是一种模仿现实世界中继承关系的一种类之间的关系，是超类(父类)和子类之间共享数据和方法的机制。在定义和实现一个类的时候，可以在一个已经存在的类的基础上来进行，子类可以继承其父类中的属性和操作作为自己的内容而不必自己定义，也可以用更具体地方式实现从父类继承来的方法，称为覆盖。不同的对象收到同一消息可以进行不同的响应，产生完全不同的结果，用户可以发送一个通用的消息，而实现细节则由接收对象自行决定，使得同一个消息就可以调用不同的方法，即一个对象具有多种形态，称为多态。不同类的对象通过消息相互通信。

(44)设计模式定义一系列算法，把它们一个个封装起来，并且使它们可相互替换。这一模式使得算法可独立于它的客户而变化。

(44)A.策略(Strategy)　　B.抽象工厂(Abstract Factory)

C.观察者(Visitor)　　D.状态(State)

【答案】A

【解析】本题考查设计模式的基本概念。

策略(Strategy)设计模式定义一系列算法，把它们一个个封装起来，并且使它们可相互替换。这一模式使得算法可独立于它的客户而变化。抽象工厂(Abstract Factory)模式提供一个创建一系列相关或相互依赖对象的接口，而无需指定他们具体的类。观察者(Observer)模式定义对象间的一种一对多的依赖关系，当一个对象的状恶发生改变时，所有依赖于它的对象都得到通知并被自动更新。状态(State)模式是使得一个对象在其内部状态改变时通过调用另一个类中的方法改变其行为，使这个对象看起来如同修改了它的类。

在发布-订阅(Publish-Subscribe)消息模型中，订阅者订阅一个主题后，当该主题有新消息到达时，所有订阅者都会收到通知。(45)设计模式最适合这一模型。

(45)A.适配器(Adapter)　　B.通知(Notifier)

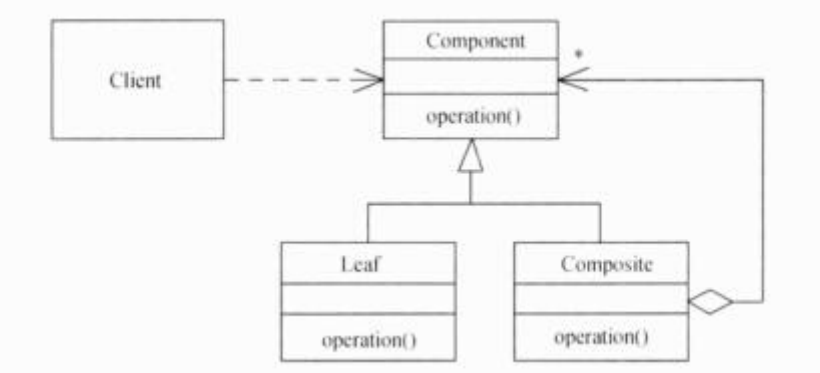
C.状态(State)　　D.观察者(Observer)

【答案】D

【解析】本题考查设计模式的基本概念。

适配器(Adapter)模式将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。观察者(Observer)模式定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新，其别名为发布-订阅(Publish-Subscribe)模式。状态(State)模式是使得一个对象在其内部状态改变时通过调用另一个类中的方法改变其行为，使这个对象看起来如同修改了它的类。

下图所示为(46)设计模式，适用于：(47)。



(46)A.组件(Component)　　B.适配器(Adapter)

C.组合(Composite)　　D.装饰器(Decorator)

(47)A.表示对象的部分-整体层次结构

B.不希望在抽象和它的实现部分之间有一个固定的绑定关系

C.在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责

D.使所有接口不兼容类可以一起工作

【答案】C　A

【解析】本题考查设计模式的基本概念。

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

适配器(Adapter)模式将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

组合(Composite)模式将对象组合成树形结构以表示“部分-整体”的层次结构，使得用户对单个对象和组合对象的使用具有一致性。组件 Component 为组合的对象声明接口，通常

定义父组件引用，Leaf 和 Composite 类可以继承这个引用以及管理这个应用的那些操作。装饰器(Decorator)模式描述了以透明围栏来支持修饰的类和对象的关系，动态地给一个对象添加一些额外的职责，从增加功能的角度来看，装饰器模式相比生成子类更加灵活。

将高级语言程序翻译为机器语言程序的过程中，常引入中间代码，其好处是(48)。

(48)A.有利于进行反编译处理　　B.有利于进行与机器无关的优化处理

C.尽早发现语法错误　　D.可以简化语法和语义分析

【答案】B

【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，可以有若干种形式，它们的共同特征是与具体的机器无关，此时所作的优化一般建立在对程序的控制流和数据流分析的基础之上，与具体的机器无关。

对高级语言源程序进行编译的过程中，有穷自动机(NFA 或 DFA)是进行(49)的适当工具。

(49)A.词法分析　　B.语法分析　　C.语义分析　　D.出错处理

【答案】A

【解析】本题考查程序语言基础知识。

语言中具有独立含义的最小语法单位是符号(单词)，如标识符、无符号常数与界限符等。词法分析的任务是把构成源程序的字符串转换成单词符号序列。

有限自动机是一种识别装置的抽象概念，它能准确地识别正规集。有限自动机分为两类：确定的有限自动机(DFA)和不确定的有限自动机(NFA)。

弱类型语言(动态类型语言)是指不需要进行变量/对象类型声明的语言。(50)属于弱类型语言。

(50)A.Java　　B.C/C++　　C.Python　　D.C#

【答案】C

【解析】本题考查程序语目基础知识。

弱/强类型指的是语言类型系统的类型检查的严格程度，动态类型和静态类型则指变量与类型的绑定方法。

静态类型指编译器在编译源程序期间执行类型检查，动态类型指编译器(虚拟机)在程序

运行时执行类型检查。简单地说，在声明了一个变量之后，不能改变其类型的语言，是静态语言；能够随时改变其类型的语言，是动态语言。

弱类型相对于强类型来说类型检查更不严格，比如说允许变量类型的隐式转换，允许强制类型转换等等。

若有关系R(A,B,C,D,E)和S(B,C,F,G),则R与S自然联结运算后的属性列有(51)个，与表达式π1,3,6,7(σ3<6(RS))等价的SQL语句如下：

SELECT (52) FROM (53) WHERE (54);

(51)A.5　　B.6　　C.7　　D.9

(52)A.A,R.C,F,G　　B.A,C,S.B,S.F

C.A,C,S.B,S.C　　D.C.R.A, R.C, S.B,S.C

(53)A.R　　B.S　　C.RS　　D.R, S

(54)A.R.B=S.BANDR.C=S.CANDR.C<S.B　　B.R.B=S.BANDR.C=S.CANDR.C<S.F

C.C.R.B=S.BORR.C=S.CORR.C<S.B　　D.R.B=S.BORR.C=S.CORR.C<S.F

【答案】C　A　D　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π1,3,6,7(σ3<6(RS))中，自然联结RS运算后去掉右边重复的属性列名S.B、S.C后为：R.A、R.B、R.C、R.D、R.E、S.F和S.G,因此空(51)的正确答案为7。π1，3，6，7(σ3<6(RS))的含义是从RS结果集中选取R.C<S.F的元组，再进行R.A、R.C、S.F和S.G投影，因此，空(52)的正确答案为选项A。显然，空(53)的答案为R，S。

空(54)的正确答案为选项B。因为，自然联结RS需要用条件“WHERE R.B=S.B AND R.C=S.C”来限定，选取运算σ3<6需要用条件“WHERE R.C<S.F”来限定。

在分布式数据库系统中，(55)是指用户无需知道数据存放的物理位置。

(55)A.分片透明　　B.复制透明　　C.逻辑透明　　D.位置透明

【答案】D

【解析】本题考查分布式数据库基本概念。

分片透明是指用户或应用程序不需要知道逻辑上访问的表具体是怎么分块存储的。复制透明是指采用复制技术的分布方法，用尸不需要知道数据是复制到哪些节点，如何复制的。位置透明是指用户无需知道数据存放的物理位置，逻辑透明局部数据模型透明，是指用户或

应用程序无需知道局部场地使用的是哪种数据模型。

计算机系统的软硬件故障可能会造成数据库中的数据被破坏。为了防止这一问题，通常需要(56),以便发生故障时恢复数据库。

(56)A.定期安装 DBMS 和应用程序

B.定期安装应用程序，并将数据库做镜像

C.定期安装 DBMS，并将数据库作备份

D.定期将数据库作备份；在进行事务处理时，需要将数据更新写入日志文件

【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。具体的实现方法有多种，如：定期将数据库作备份；在进行事务处理时，对数据更新(插入、删除、修改)的全部有关内容写入日志文件；当系统正常运行时，按一定的时间间隔，设立检查点文件，把内存缓冲区内容还未写入到磁盘中去的有关状态记录到检查点文件中；当发生故障时,根据现场数据内容、日志文件的故障前映像和检查点文件来恢复系统的状态。

以下关于线性表存储结构的叙述，正确的是(57)。

(57)A.线性表采用顺序存储结构时,访问表中任意一个指定序号元素的时间复杂度为常量级

B.线性表采用顺序存储结构时,在表中任意位置插入新元素的运算时间复杂度为常量级

C.线性表采用链式存储结构时,访问表中任意一个指定序号元素的时间复杂度为常量级

D.线性表采用链式存储结构时,在表中任意位置插入新元素的运算时间复杂度为常量级

【答案】A

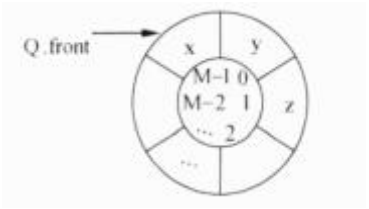
【解析】本题考查数据结构基础知识。

线性表进行顺序存储时，逻辑上相邻的元素，其物理位置也相邻，因此在已知第一个元

素存储位置和元素序号的情况下，可计算出表中任意指定序号元素的存储位置，即按照序号访问元素时随机的，该运算的时间复杂度为O(1)，也就是常量级。而插入元素时就需要移动一些元素了，在最坏情况下要移动表中的所有元素，因此该运算的时间复杂度为O(n)，其中n为线性表的长度。

线性表进行链式存储时，逻辑上相邻的元素，其物理位置不要求相邻，因此需要额外的存储空间表示元素之间的顺序关系。在链表上查找元素和插入元素的运算时间复杂度都为O(n)。

设循环队列Q的定义中有front和size两个域变量，其中front表示队头元素的指针，size表示队列的长度，如下图所示(队列长度为3,队头元素为x、队尾元素为z)。设队列的存储空间容量为M，则队尾元素的指针为(58)。



(58)A.(Q.front+Q.size-1)　　B.(Q.front+Q.size-1+M)%M

C.(Q.front-Q.size)　　D.(Q.front-Q.size+M)%M

【答案】B

【解析】本题考查数据结构基础知识。

根据题目中所给的示意图，Q.front为队头元素的指针，该指针加1后得到队列中的第2个元素(即y)的指针，由于队列中存储位置编号是在0～M-1之间循环的，队头指针加上1个增量后可能会超出该范围，应该用整除取余运算恢复一下，因此由Q.front可以算出队列尾部元素的指针为(Q.front+Q.size-1+M)%M。

在一个有向图G的拓扑序列中，顶点Vi排列在Vj之前，说明图G中(59)。

(59)A.一定存在弧(vj, vi)

B.一定存在弧

C.可能存在vi到vj的路径，而不可能存在Vj到v,的路径

D.可能存在vj到vi的路径，而不可能存在vi到vj的路径

【答案】C

【解析】本题考查数据结构基础知识。

对一个有向图 G 进行拓扑排序的方法如下。

①G 中选择一个入度为 0(没有前驱)的顶点且输出它；

②从网中删除该顶点及其与该顶点有关的所有弧；

③重复上述两步，直至网中不存在入度为 0 的顶点为止。

显然，若存在弧<vi，vj> ，则 vj 的入度就不为 0,而要删除该弧，则 vi 的入度应为 0,因此在拓扑序列中，vi 必然在'之前。另外，进行拓扑排序时，可能存在 vi 和 vj 的入度同时为 0 的情形，此时，在第①步可先输出 vi，后输出 vj。因此在拓扑序列中，顶点 vi 排列在 vj 之前，不一定存在弧<vi，vj>，一定不存在弧<vj，vi>,也一定不存在 vj 到 vi 的路径，而可能存在 vi 到 vj 的路径。

以下关于哈夫曼树的叙述，正确的是(60)。

(60)A.哈夫曼树一定是满二叉树，其每层结点数都达到最大值

B.哈夫曼树一定是平衡二叉树，其每个结点左右子树的高度差为-1、0 或 1

C.哈夫曼树中左孩子结点的权值小于父结点、右孩子结点的权值大于父结点

D.哈夫曼树中叶子结点的权值越小则距离树根越远、叶子结点的权值越大则距离树根越近

【答案】D

【解析】本题考查数据结构基础知识。

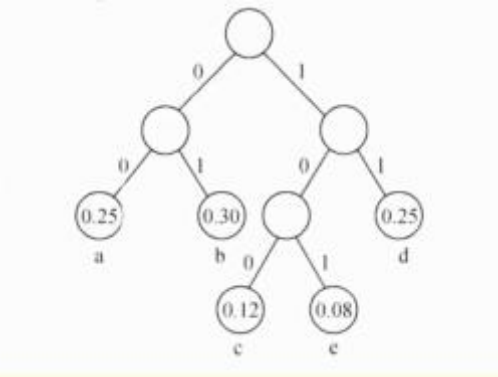
哈夫曼树是一类带权路径长度最短的树，根据一组权值构造出来。构造过埕为：

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。

根据权值集合{0.25,0.30,0.08,0.25,0.12}构造的哈夫曼树如下图所示，从中可以知道，哈夫曼树中叶子结点的权值越小则距离树根越远、叶子结点的权值越大则距离树根越近。



某哈希表(散列表)的长度为n，设散列函数为H(Key)=Key mod p，采用线性探测法解决冲突。以下关于p值的叙述中，正确的是(61)。

(61)A.p的值一般为不大于n且最接近n的质数　B.p的值一般为大于n的任意整数

C.p的值必须为小于n的合数　D.p的值必须等于n

【答案】A

【解析】本题考查数据结构基础知识。

在应用散列函数构造哈希表(或散列表)时，由于设计散列函数的目标是：作为一个压缩映像函数，它应具有较大的压缩性，以节省存储空间；哈希函数应具有较好的散列性，虽然冲突是不可避免的，但应尽量减少。题中所给是常用的除留余数法，P值一般为不大于n且最接近n的质数。

对n个基本有序的整数进行排序，若采用插入排序算法，则时间和空间复杂度分别为(62);若采用快速排序算法，则时间和空间复杂度分别为(63)。

(62)A.O(n2)和O(n)　B.O(n)和O(n)　C.O(n2)和O(1)　D.O(n)和O(1)

(63)A.O(n2)和O(n)　B.O(nlgn)和O(n)　C.O(n2)和O(1)　D.O(nlgn)和O(1)

【答案】D　C

【解析】本题考查算法分析的基础知识。

排序和查找是基本的计算问题，存在很多相关的算法，不同的算法适用于不同的场合。不同的数据输入特点相同的算法也有不同的计算时间。若数据基本有序，对插入排序算法而言，则可以在近似线性时间内完成排序，即O(n);而对于快速排序而已，则是其最坏情况，需要二次时间才能完成排序，即O(n2)。两个算法在排序时仅需要一个额外的存储空间，即空间复杂度均为常数时间复杂度O(1)。

在求解某问题时，经过分析发现该问题具有最优子结构性质，求解过程中子问题被重复求解，则采用(64)算法设计策略;若定义问题的解空间，以深度优先的方式搜索解空间，则采用(65)算法设计策略。

(64)A.分治　　B.动态规划　　C.贪心　　D.回溯

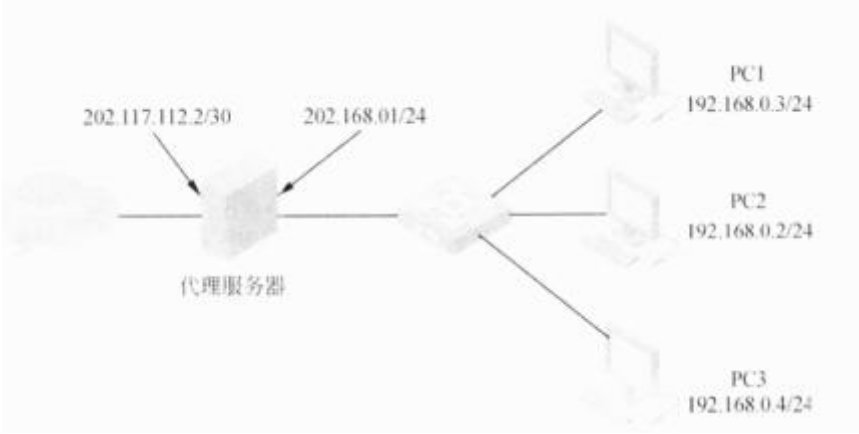
(65)A.动态规划　　B.贪心　　C.回溯　　D.分支限界

【答案】B　C

【解析】本题考查算法设计的基础知识

存在几种常用的算法设计策略：分治法、动态规划、贪心、回溯发和分支限界法等。其中，分治法一般用于将大问题分解为一个或多个规模较小的子问题，通常采用自顶向下的递归方法来求解。动态规划求解问题的特征是，问题具有最优子结构和重叠子问题，求解时一般采用自底向上的方法来进行。贪心法求解问题的特征是，问题具有最有子结构和贪心选择性质，求解时可以用自底向上或自顶向下的方法进行。回溯法和分支限界法是系统搜索解空间来求解问题的方法，一般先定义解空间，前者以深度优先的方式搜索，后者通常以广度优先的方式搜索。

某单位的局域网配置如下图所示，PC2 发送到 Internet 上的报文的源 IP 地址为(66)。



(66)A.192.168.0.2　　B.192.168.0.1　　C.202.117.112.1　　D.202.117.112.2

【答案】D

【解析】本试题考查局域网配置中 IP 地址设置相关问题。

PC2 发送到 Internet 上的报文经代理服务器转换后，源 IP 地址变成代理服务器的出口 IP 地址，即 202.117.112.2。

在 IPv4 向 IPv6 过渡期间,如果要使得两个 IPv6 结点可以通过现有的 IPv4 网络进行通信,则应该使用(67);如果要使得纯 IPv6 结点可以与纯 IPv4 结点进行通信,则需要使用(68)。

(67)A.堆栈技术　　B.双协议栈技术　　C.隧道技术　　D.翻译技术

(68)A.堆栈技术　　B.双协议栈技术　　C.隧道技术　　D.翻译技术

【答案】C　D

【解析】

如果要使得两个 IPv6 结点可以通过现有的 IPv4 网络进行通信，则应该使用隧道技术，如果要使得纯 IPv6 结点可以与纯 IPv4 结点进行通信，则需要使用翻译技术。

POP3 协议采用(69)模式进行通信,当客户机需要服务时，客户端软件与 POP3 服务器建立(70)连接。

(69)A.Browser/Server　　B.Client/Server　　C.PeertoPeer　　D.PeertoServer

(70)A.TCP　　B.UDP　　C.PHP　　D.IP

【答案】B　A

【解析】

POP3 协议采用 C/S 模式进行通信，POP3 需要 TCP 连接的支持，当客户机需要服务时，客户端软件与 POP3 服务器建立 TCP 连接。

There is nothing in this world constant but inconstancy. —SWIFT Project after project designs a set of algorithms and then plunges into construction of customer-deliverable software on a schedule that demands delivery of the first thing built.

In most projects, the first system built is (71) usable. It may be too slow, too big, awkward to use, or all three. There is no (72) but to start again, smarting but smarter, and build a redesigned version in which these problems are solved. The discard and (73) may be done in one lump, or it may be done piece-by-piece. But all large-system experience shows that it will be done. Where a new system concept or new technology is used, one has to build a system to throw away, for even the best planning is not so omniscient (全知的) as to get it right the first time.

The management question, therefore, is not whether to build a pilot system and throw it away. You will do that. The only question is whether to plan in advance to build a (74) , or to promise to deliver the throwaway to customers. Seen this way, the answer is mi.ch clearer. Delivering that throwaway to customers buys time, but it does so only at the (75) of agony (极大痛苦)for the user, distraction for the builders while they do the redesign, and a bad reputation for the product that the best redesign will find hard to live down.

(71)A.almost	B.often	C.usually	D.barely
(72)A.alternative	B.need	C.possibility	D.solution
(73)A.design	B.redesign	C.plan	D.build
(74)A.throwaway	B.system	C.software	D.product
(75)A.worth	B.value	C.cost	D.invaluable

【答案】D A B A C

【解析】

不变只是愿望，变化才是永恒。——SWIFT

一个接一个的软件项目都是一开始设计算法，然后将算法应用到待发布的软件中，接着根据时间进度把第一次开发的产品发布给客户。

对于大多数项目，第一个开发的系统并不适用。它可能太慢、太大、难以使用，或者三者兼有。要解决所有的问题，除了重新开始以外，没有其他的办法——即开发一个更灵巧或者更好的系统。系统的丢弃和重新设计可以一步完成，也可以一块块地实现。所有大型系统的经验都显示，这是必须完成的步骤。而且，新的系统概念或新技术会不断出现，因此开发的系统必须被抛弃,但即使是最优秀的项目计划也不能无所不知地在最开始就解决这些问题。因此，管理上的问题不再是“是否构建一个实验性的系统，然后抛弃它”，你必须这样做。现在的问题是“是否预先计划抛弃原型的开发，或者是否将该原型发布给用户”。从这个角度看待问题，答案更加清晰。将原型发布给用户，虽然可以获得时间，但是其代价高昂——对于用户，使用极度痛苦；对于重新开发的人员，分散了精力；对于产品，影响了声誉，即使是最好的再设计也难以挽回名声。

因此，为舍弃而计划，无论如何，你一定要这样做。

试题一

某大学欲开发一个基于 Web 的课程注册系统，该系统的主要功能如下：

1. 验证输入信息

(1) 检查学生信息：检查学生输入的所有注册所需信息。如果信息不合法，返回学生信息不合法提示；如果合法，输出合法学生信息。

(2) 检查学位考试结果：检查学生提供的学位考试结果。如果不合法，返回学位考试结果不合法提示；如果合法，检查该学生注册资格。

(3) 检查学生注册资格：根据合法学生信息和合法学位考试结果，检查该学生对欲选课程的注册资格。如果无资格，返回无注册资格提示；如果有注册资格，则输出注册学生信息（包含选课学生标识）和欲注册课程信息。

2. 处理注册申请

(1) 存储注册信息：将注册学生信息记录在学生库。

(2) 存储所注册课程：将选课学生标识与欲注册课程进行关联，然后存入课程库。

(3) 发送注册通知：从学生库中读取注册学生信息，从课程库中读取所注册课程信息，给学生发送接受提示；给教务人员发送所注册课程信息和已注册学生信息。

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

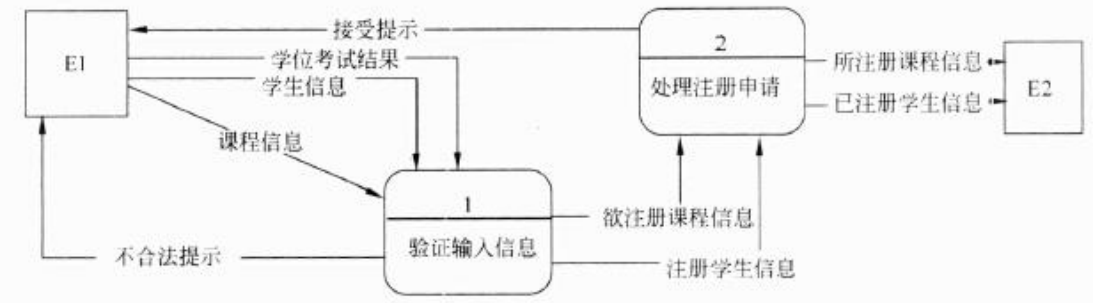


图 1-1　0 层数据流图

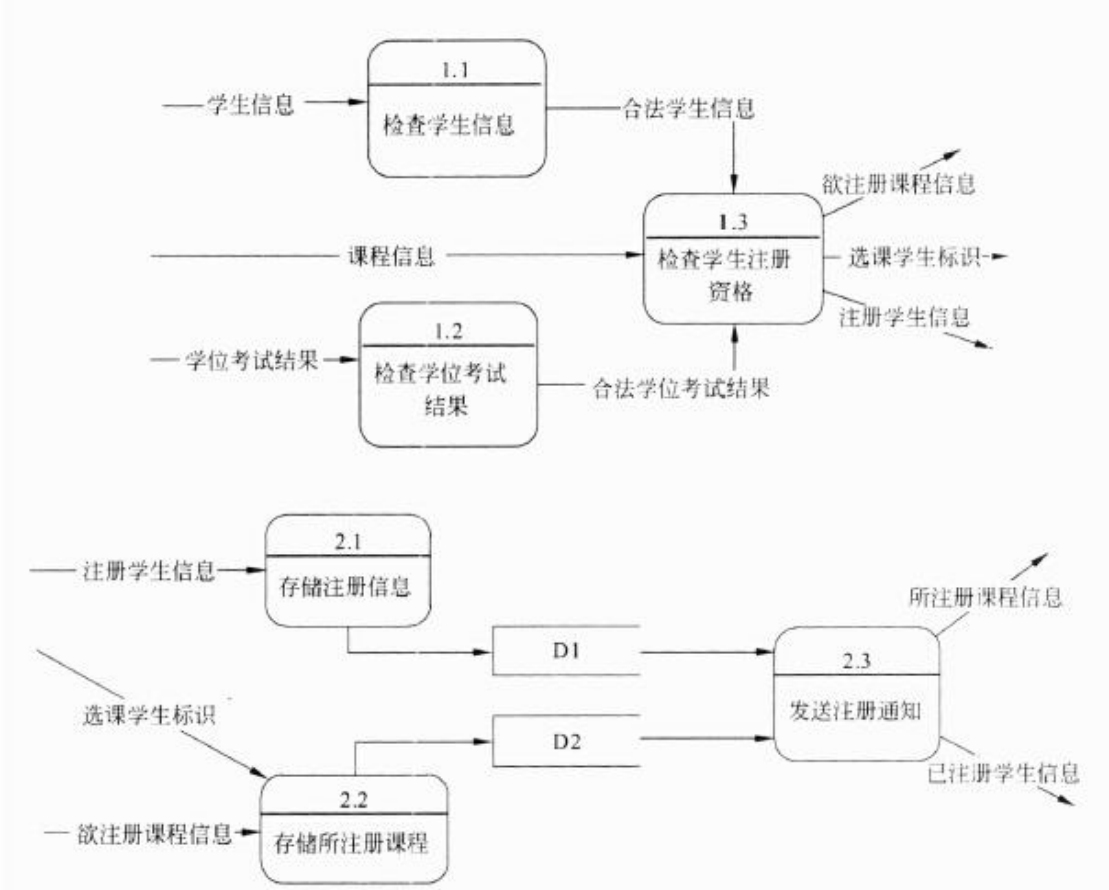


图 1-2　1 层数据流图

【问题 1】

使用说明中的词语，给出图 1-1 中的实体 E1 和 E2 的名称。

E1：学生

E2：教务人员

本问题考查 0 层 DFD，要求确定外部实体。不难看出，在 0 层 DFD 中，系统主要功能“验证输入信息”和“处理注册申请”，涉及与系统交互的外部实体有“学生”提供输入信息，发送注册通知功能给“教务人员”发送所注册的课程信息和已注册的学生信息，从而即可确定 E1 为“学生”实体，E2 为“教务人员”实体。

【问题 2】

使用说明中的词语，给出图 1-2 中的数据存储 D1 和 D2 的名称。

D1：学生库

D2：课程库

本问题要求确定 1 层数据流图中的数据存储。分析说明中和数据存储有关的描述，不难发现，说明 2.(1)存储注册信息明确说明“将注册学生信息记录在学生库”，可知 D1 为学生库；说明 2.(2)存储所注册课程中明确说明“然后存入课程库”，可知 D2 为课程库。

【问题 3】

根据说明和图中术语，补充图 1-2 中缺失的数据流及其起点和终点。

数 据 流	起 点	终 点
学生信息不合法提示	1.1 或 检查学生信息	E1 或 学生
无注册资格提示	1.3 或 检查学生注册资格	E1 或 学生
学位考试结果不合法提示	1.2 或 检查学位考试结果	E1 或 学生
接受提示	2.3 或 发送注册通知	E1 或 学生

本问题要求补充缺失的数据流及其起点和终点。细心的考生可能会发现，对照图 1-1 和图 1-2 的输入数据流，数量和名称均相同，所以缺失的数据流是输出数据流或者处理之间的数据流。考查图 1-1 中输出至 E1 的数据流，有“接受提示”和“不合法提示”，而图 1-2 中没有这两条数据流，可以确定缺失的数据流包括这两条或者其分解的数据流。

考查说明 1 中的 3 个子功能，

1.(1)检查学生信息完成检查学生输入的所有注册所需信息。如果信息不合法，返回学生信息不合法提示。

1.(2)检查学位考试结果完成检查学生提供的学位考试结果。如果不合法，返回学位考试结果不合法提示。

1.(3)检查学生注册资格完成根据合法学生信息和合法学位考试结果，检查该学生对欲选课程的注册资格。如果无资格，返回无注册资格提示。

对应图 1-1 中的处理 1 验证输入信息的输出数据流“不合法提示”，不难发现，在图 1-2 中，处理 1.1 缺少了到实体学生的输出数据流“学生信息不合法提示”；处理 1.2 缺少了到实体学生的输出数据流“无注册资格提示”；处理 1.3 缺少了到实体学生的输出数据流“学位考试结果不合法提示”。

再考查图 1-1 中处理 2,其输出数据流有三条，而图 1-2 中对图 1-1 中处理 2 的分解中，只包含了“所注册课程信息”和“已注册学生信息”两条数据流，缺失了“接受提示”。说明 2.(3)中发送注册通知功能完成从学生库中读取注册学生信息，从课程库中读取所注册课程信息，给学生发送接受提示；给教务人员发送所注册课程信息和已注册学生信息。所以，缺失的“接受提示”的起点是处理 2.3 发送注册通知，终点是 E1 学生。

【问题 4】

根据补充完整的图 1-1 和图 1-2,说明上层的哪些数据流是由下层的哪些数据流组合而成。

图 1-1 中不合法提示分解为图 1-2 中的三条数据流的组合：学生信息不合法提示、无注册资格提示、学位考试结果不合法提示。
图 1-1 中注册学生信息对应图 1-2 中注册学生信息和选课学生标识。

本问题考查数据流的分解与组合。仔细分析【说明】中的文字并与图 1-1 的对照，可以发现在图 1-1 中不合法提示在图 1-2 中没有出现。事实上，从前述【问题 3】缺失数据流的分析中，已经发现，图 1-2 中对于说明中的功能出现了“学生信息不合法提示”、“无注册资格提示”和“学位考试结果不合法提示”三条数据流，说明图 1-1 中的数据流“不合法提示”是由这三条数据流组合而成。同样，2.(2)存储所注册课程将选课学生标识与欲注册课程进行关联，然后存入课程库，图 1-1 中注册学生信息在图 1-2 中进一步分出注册学生信息和选课学生标识，即图 1-1 中注册学生信息是注册学生信息和选课学生标识的并集。

试题二

某快递公司为了方便管理公司物品运送的各项业务活动，需要构建一个物品运送信息管理系统。

【需求分析结果】

(1) 快递公司有多个分公司，分公司信息包括分公司编号、名称、经理、办公电话和地址。每个分公司可以有多名员工处理分公司的日常业务，每名员工只能在一个分公司工作。每个分公司由一名经理负责管理分公司的业务和员工，系统需要记录每个经理的任职时间。

(2) 员工信息包括员工号、姓名、岗位、薪资、手机号和家庭地址。其中，员工号唯一标识员工信息的每一个元组。岗位包括经理、调度员、业务员等。业务员根据客户提交的快件申请单进行快件受理事宜，一个业务员可以受理多个客户的快件申请，一个快件申请只能由一个业务员受理。调度员根据已受理的申请单安排快件的承运事宜，例如：执行承运的业务员、运达时间等。一个业务员可以执行调度员安排的多个快件的承运业务。

(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。当客户要寄快件时，先要提交快件申请单，申请号由系统自动生成。快件申请信息包括申请号、客户号、发件人、发件人电话、快件名称、运费、发出地、收件人、收件人电话、收件地址。其中，一个申请号对应唯一的一个快件申请，一个客户可以提交多个快件申请，但一个快件申请由唯一的一个客户提交。

【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（图 2-1)和关系模式（不完整）如下：

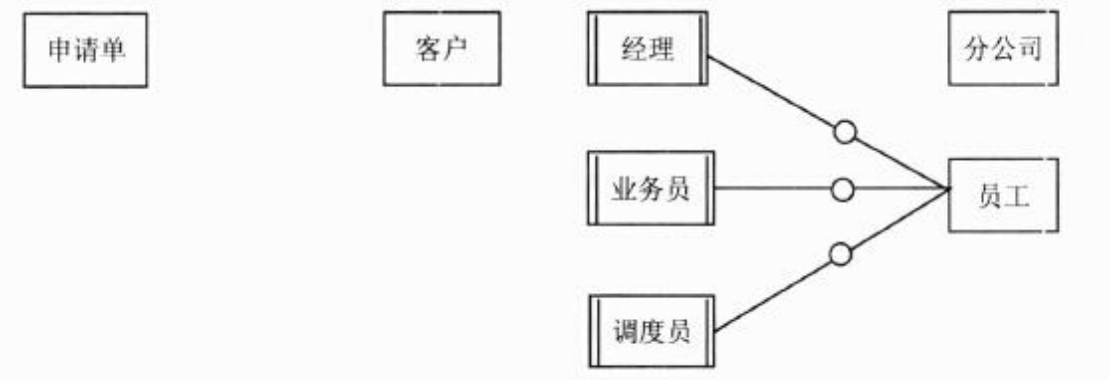


图 2-1　实体联系图

【关系模式设计】

分公司（分公司编号，名称，经理，办公电话，地址）

员工（员工号，姓名，(a),岗位，薪资，手机号，家庭地址）

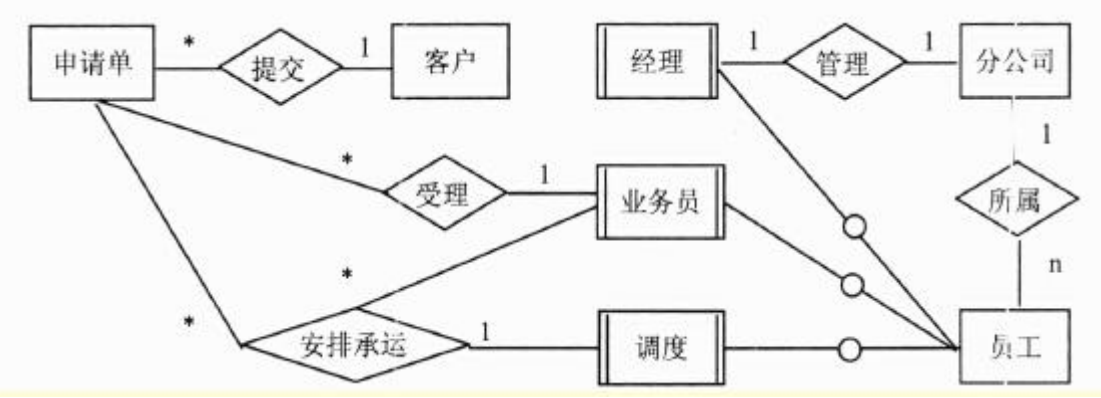
客户（客户号，单位名称，通信地址，所属省份，联系人，联系电话，银行账号）申请单（(b)，发件人，发件人电话，发件人地址，快件名称，运费，收件人，收件人电话，收件地址，受理标志，业务员）

安排承运（（c），实际完成时间，调度员）

【问题 1】

根据问题描述，补充五个联系，完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3、联系 4 和联系 5 代替，联系的类型分为 1:1、1:n 和 m:n(或 1:1、1：*和*:*)。

图中的*可表示为 m 或 n，对联系名称可不做要求，但不能出现重名。



由“每个分公司有一位经理”可知分公司与经理之间的管理联系类型为 1:);由“每个分公司有多名员工处理日常事务，每个员工属于一个分公司”可知分公司与员工间的所属联系类型为 1:*;并且员工是经理的超类型，经理是员工的子类型。

由“一个客户可以有多个快件申请，但一个快件申请对应唯一的一个客户”可知，客户与申请单之间的提交联系类型为 1:*。

由“业务员根据客户提交的快件申请单进行快件受理事宜，一个业务员可以受理多个客户的快件申请，一个快件申请只能由一个业务员受理”可知业务员与申请单之间的受理联系类型为 1:*。

由“调度根据已受理的申请单安排快件的承运事宜，例如：执行承运的业务员、运达时间等；一个业务员可以执行调度安排的多个快件的承运业务。”可知，调度、业务员和申请单之间的承运联系类型为 1:*:*。

【问题 2】

(1) 根据实体联系图，将关系模式中的空(a)～（c)补充完整。

(2) 给出员工、申请单和安排承运关系模式的主键和外键。

(1) (a)分公司编号

(b) 申请号，客户号

(c) 申请号，业务员

关 系 模 式	主　　键	外　　键
员工	员工号	分公司编号
申请单	申请号	客户号，业务员
安排承运	申请号	业务员，调度员

(1)完整的关系模式如下：

分公司（分公司编号，名称，办公电话，地址）

员工(员工号，姓名，分公_司_编_号，岗位，薪资，手机号，家庭地址）

客户(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)

申请单(申请号，客方号，发件人，发件人电话，发件人地址，快件名称，运费，收件人，收件人电话，收件地址，受理标志，业务员）

安排承运(申请号，亚务炅，实际完成时间，调度员）

(2)员工、申请单和安排承运关系模式的主键和外键的分析如下：

在申请单信息中，申请号由系统自动生成，不会重复，可作为申请单的主键属性，外键为客户号，业务员；在员工信息中，员工号唯一标识员工信息的每一个元组，故为员工关系的主键属性，外键为分公司编号；安排承运关系模式的主键为申请号，外键为业务员和调度员。

【问题 3】

(1)客户关系的通信地址可以进一步分为邮编、省、市、街道，那么该属性是否属于简单属性，为什么？请用 100 字以内的文字说明。

(2)假设分公司需要增设一位经理的职位，那么分公司与经理之间的联系类型应修改为（d），分公司的主键应修改为 （e）。

(1)该属性不属于简单属性。因为简单属性是原子的、不可再分的，复合属性是可以细分为更小的部分（即划分为别的属性)，本题客户关系的通信地址可以进一步分为邮编、

省、市、街道，所以属于复合属性。

(2) (d)1:n

(e)分公司编号，经理

(1) 客户的通信地址属性不属于简单属性。因为根据题意，客户关系的通信地址可以进一步分为邮编、省、市、街道，而简单属性是原子的、不可再分的，复合属性可以细分为更小的部分（即划分为别的属性)。由于客户的通信地址可以进一步分为邮编、省、市、街道，故属于复合属性。

(2) 根据题意，分公司需要增设一位经理的职位，那么分公司与经理之间的联系类型应修改为1:n，分公司的主键应修改为分公司编号，经理。

试题三

某航空公司会员积分系统（CFequentFlyer)的主要功能描述如下：

乘客只要办理该航空公司的会员卡，即可成为普卡会员（CBasic)。随着飞行里程数的积累，可以从普卡会员升级到银卡会员(CSilver)或金卡会员(CGold)。非会员(CNonMember)不能累积里程数..

每年年末，系统根据会员在本年度累积的里程数对下一年会员等级进行调整。普卡会员在一年内累积的里程数若满 25,000 英里但不足 50,000 英里，则自动升级为银卡会员；若累积的里程数在 50,000 英里以上，则自动升级为金卡会员。银卡会员在一年内累积的里程数若在 50,000 英里以上，则自动升级为金卡会员。

若一年内没有达到对应级别要求的里程数，则自动降低会员等级。金卡会员一年内累积的里程数若不足 25,000 英里，则自动降级为普卡会员；若累积的里程数达到 25,000 英里，但是不足 50,000 英里，则自动降级为银卡会员。银卡会员一年内累积的里程数若不足 25,000 英里，则自动降级为普卡会员。

采用面向对象方法对会员积分系统进行分析与设计，得到如图 3-1 所示的状态图和图 3-2 所示的类图。

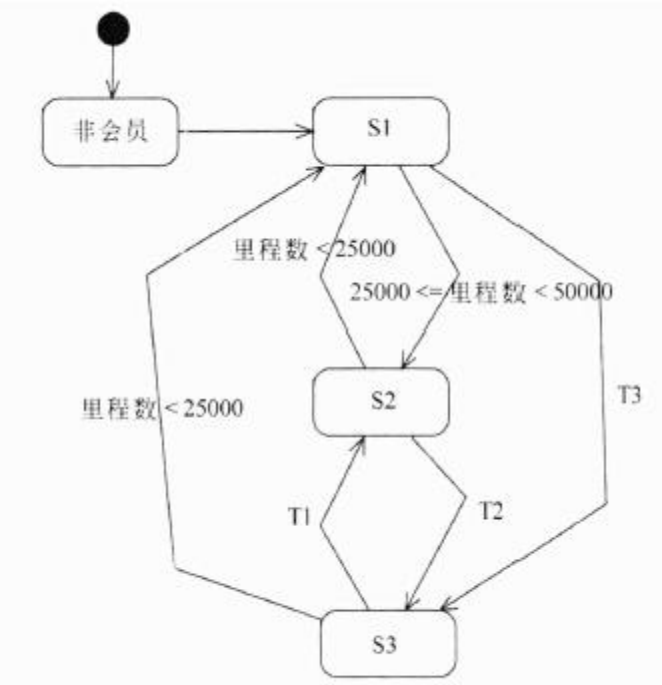


图 3-1　状态图

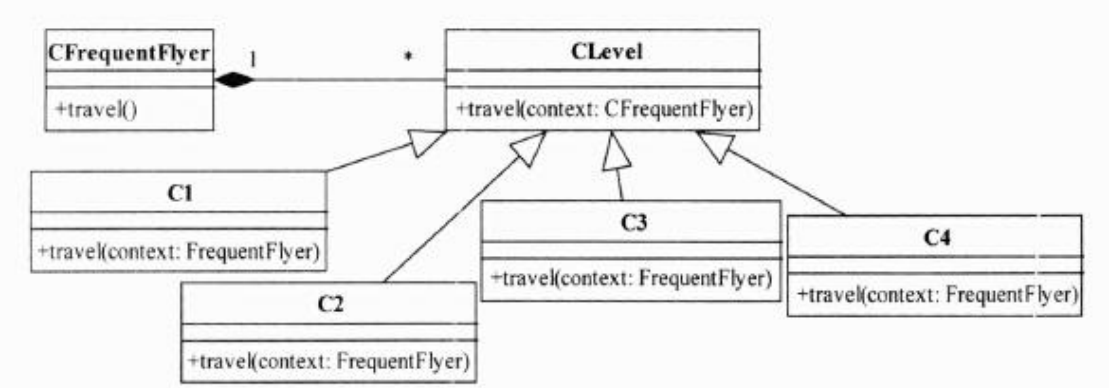


图 3-2　类图

【问题 1】

根据说明中的描述，给出图 3-1 中 S1～S3 处所对应的状态以及 T1~T3 处所对应的迁移的名称。

S1：普卡、普卡会员

S2：银卡、银卡会员

S3：金卡、金卡会员

T1:25000<=里程数<50000

T2:里程数>=50000

T3:里程数>=50000

UML 中的状态图主要用于描述一个对象在其生存期间的动态行为，表现一个对象所经历的装填序列，引起状态转移的事件以及因状态转移而伴随的动作。图中给出的是会员的状态图。图中要求填充 SI、S2、S3 这三个状态以及它们之间的变迁关系。本题中会员有三种状态：普卡、金卡和银卡。根据说明，办理会员卡之后即可成为普卡会员，所以 S1 可以判定为普卡会员。当“里程数满 25,000 英里但不足 50,000 英里，则自动升级为银卡会员”，所以 S2 应为银卡会员，那么 S3 就应该是金卡会员。T1、T2 就是 S2 和 S3 之间的转换原则。T3 是 S1->S2 的转换原则。由说明可知，S2->S3(T2)：里程数在 50,000 英里以上;S3->S3(T1)：里程数达到 25,000 英里，但是不足 50,000 英里；S1->S3(T3)：累积的里程数在 50,000 英里以上。

【问题 2】

根据说明中的描述，给出图 3-2 中 C1～C4 所对应的类名（类名使用说明中给出的英文

词汇）。

C1：CNonMember

C2：CBasic

C3：CSilver

C4：CGold

（C1～C4 的次序可以互换）

由图 3-2 可知，需要补充的是继承结构中的子类。根据题目说明，能够具有一般/特殊关系的只有不同级别的会员。所以 C1~C4 依次应该是：CNonMember、CBasic,CSilver,CGold。

【问题 3】

图 3-2 所示的类图中使用了哪种设计模式？在这种设计模式下，类 CFrecuentFlyer 必须具有的属性是什么？C1～C4 中的 travel 方法应具有什么功能？

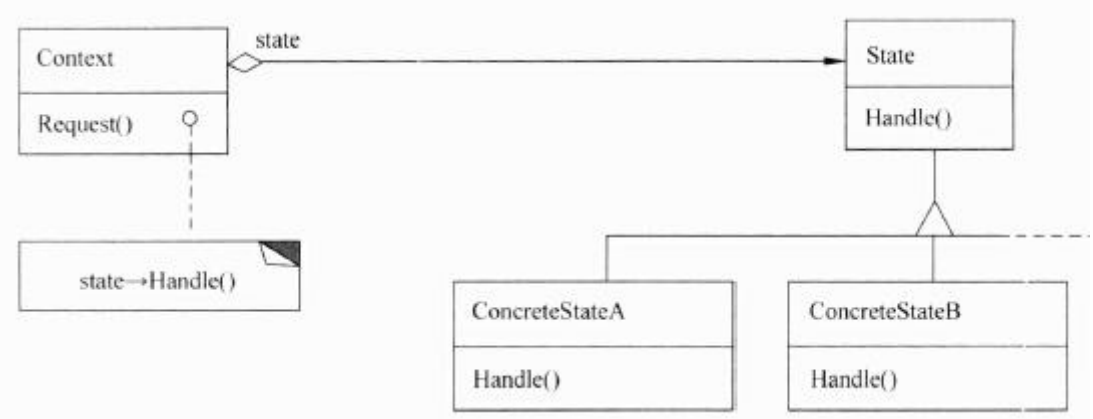
使用了 State 模式（状态模式）。

类 CFrequentFlyer 必须具有的属性：CLevel 的对象。

travel 方法的功能：计算飞行里程数，根据里程数判断是否需要调整会员级别（跳转到不同的状态）。

本题在设计类时使用到了状态模式。

状态模式允许对象在内部状态变化时，变更其行为，并且修改其类。状态模式的类图如下所示。



其中：

•环境类（Context):定义客户感兴趣的接口。维护一个 ConcreteState 子类的实例，这个实例定义当前状态。

•抽象状态类（State):定义一个接口以封装与 Context 的一个特定状态相关的行为。

•具体状态类（ConcreteState):每一子类实现一"与 Context 的一个状态相关的行为。

图 3-2 中的类 CFrequentFlyer 对应上图中的环境类，因此类 CFrequentFlyer 应该有一个 CLevel 类的对象。

travel 方法的功能：计算飞行里程数，根据里程数判断是否需要调整会员级别（跳转到不同的状态)。

试题四

某工程计算中要完成多个矩阵相乘（链乘）的计算任务。

两个矩阵相乘要求第一个矩阵的列数等于第二个矩阵的行数，计算量主要由进行乘法运算的次数决定。采用标准的矩阵相乘算法，计算 Am×n*Bn×p，需要 m*n*p 次乘法运算。

矩阵相乘满足结合律，多个矩阵相乘，不同的计算顺序会产生不同的计算量。以矩阵 A110×100，A2100x5,A35x50 三个矩阵相乘为例，若按（A1*A2)*A3 计算，则需要进行 10*100*5+10*5*50=7500 次乘法运算；若按 Al*(A2*A3) 计算，则需要进行 100*5*50+10*100*50=75000 次乘法运算。可见不同的计算顺序对计算量有很大的影响。

矩阵链乘问题可描述为：给定 n 个矩阵<A1，A2,…，An>，矩阵 Ai 的维数为 pMXPi，其中 i=1，2,…，n。确定一种乘法顺序，使得这 n 个矩阵相乘时进行乘法的运算次数最少。

由于可能的计算顺序数量非常庞大，对较大的 n，用蛮力法确定计算顺序是不实际的。经过对问题进行分析，发现矩阵链乘问题具有最优子结构，即若 A1*A2*…*An 的一个最优计算顺序从第 k 个矩阵处断开，即分为 A1*A2*“，*Ak 和 Ak+1*Ak--2*“，*An 两个子问题，则该最优解应该包含 A1*A2*-，*Ak 的一个最优计算顺序和 Ak+PAk+St-*An 的一个最优计算顺序。据此构造递归式，

$$cost[i][j]=\begin{cases}0 & \text{if } i=j\\ \min_{i\le k<j}\{cost[i][k]+cost[k+1][j]+p_i*p_{k+1}*p_{j+1}\} & \text{if } i<j\end{cases}$$

其中，cost[i][j]表示 Ai+1*Ai+2*……*Aj+1 的最优计算的计算代价。最终需要求解 cost[0][n-1]。

【C 代码】

算法实现采用自底向上的计算过程。首先计算两个矩阵相乘的计算量，然后依次计算 3 个矩阵、4 个矩阵……n 个矩阵相乘的最小计算量及最优计算顺序。下面是该算法的 C 语言实现。

(1)主要变量说明

n：矩阵数

seq[]：矩阵维数序列

cost[][]：二维数组，长度为 n*n,其中元素 cost[i]U]表示 Ai+1*Ai+2*…… *Aj+1 的最优计算的计算代价

trace[][]：二维数组，长度为 n*n,其中元素 trace[i][j]表示 Ai+1*Ai+2*，″*Aj+1 的

最优计算对应的划分位置，即 k

(2)函数 cmm

```
#define    N 100
int cost[N][N];
int trace[N][N];
int cmm(int n, int seq[]){
    int tempCost;
    int tempTrace;
    int i, j, k, p;
    int temp;
    for(i = 0; i < n; i++){  cost[i][i] = 0;    }
    for(p = 1; p < n; p++){
        for(i = 0;  (1)  ; i++){
            (2)  ;
            tempCost = -1;
            for(k = i; k < j; k++){
            temp =  (3)  ;
                if(tempCost == -1 || tempCost > t
                    tempCost = temp;
                    (4)  ;
                }
            }
            cost[i][j] = tempCost;
            trace[i][j] = tempTrace;
      }
   }
    return cost[0][n - 1];
}
```

【问题 1】

根据以上说明和 C 代码，填充 C 代码中的空（1)～（4)。

(1) i<n-p

(2) j=i+p

(3) cost[i][k]+cost[k+1][j]+seq[i]*seq[k+1]*seq[j+1]

(4) tempTrace=k

本问题考查算法的实现。C 程序中主要部分是三重循环，循环变量 p 定义了求解问题的规模，

因为是自底向上，因此，p 的值应该是从 1 到 n-1，即从规模为 1 的问题一直到规模为 n-1 的问题。循环变量 i 是要求解的子问题的起始，从 0 开始，最大为 n-p-1，故(1)处应填 n-p。确定了 i 和 p 之后，下来就要确定 j 了，显然，空(2)处为 j=i+p。循环变量 k 是问题 Ai*Ai+1*__，*Aj 的划分位置，对每一个 k，都要计算需要的计算成本，可以根据递归式来填写，空（3）处为 cost[i][k]+cost[k+1][j]+seq[i]*seq[k+1]*seq[j+1].确定每个问题 Ai*Ai+1*…*Aj 的划分位置 k 之后，要把这个 k 值记住，放在变量 tempTrace 中，即空(4)处填写 tempTrace=k。

【问题 2】

根据以上说明和 C 代码，该问题采用了（5）算法设计策略，时间复杂度为(6)(用 0 符号表示)。

(5) 动态规划

(6) 0(n3)

本问题考查算法的设计策略和时间复杂度，从题干说明可以很容易看出，问题具有最优子结构和重叠子问题，采用自底向上的方法求解，这些都是动态规划的典型特点，因此采用的是动态规划设计策略。从上述 C 程序很容易分析出，程序中没有递归，存在三重循环，故时间复杂度为 0(n3)。

【问题 3】

考虑实例 n=6,各个矩阵的维数：A1 为 5*10,A2 为 10*3,A3 为 3*12,A4 为 12*5,A5 为 5*50,A6 为 50*6，即维数序列为 5,10,3,12,5,50,6。则根据上述 C 代码得到的一个最优计算顺序为（7)(用加括号方式表示计算顺序)，所需要的乘法运算次数为（8)。

(7) ((A1A2)((A3A4)(A5A6)))

(8) 2010

本问题考查算法的应用。通过一个具体实例可以更容易理解问题和求解方法。可以根据问题 1 中的程序执行来求解。启发式的思路是先把维度最大的消掉，如 A;;*A6 相乘之后，维度

50 就没有了，所以考虑这两个矩阵先相乘；然后是 A3*A4 相乘之后，维度 12 就没有了，所以考虑这两个矩阵相乘；接着，A1*A2 相乘之后，维度 10 就没有了，

所以考虑这两个矩阵相乘 这样可以确定相乘的顺序((A1A2)((A3A4)(A5A<5))),需要的计算开销分别是 5*50*6=1500,3*12*5=180,5*10*3=150,3*5*6=90，5*3*6=90,把上述值相加，即 1500+180+150+90+90=2010。

试题五

欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。以绘制直线和圆形为例，对应的绘图程序如表 5-1 所示。

表 5-1　不同的绘图程序

	DP1	DP2
绘制直线	draw_a_line(x1,y1,x2,y2)	drawline(x1,x2,y1,y2)
绘制圆	draw_a_circle(x, y, r)	drawcircle(x, y, r)

该绘图软件的扩展性要求，将不断扩充新的图形和新的绘图程序。为了避免出现类爆炸的情况，现采用桥接（Bridge)模式来实现上述要求，得到如图 5-1 所示的类图。

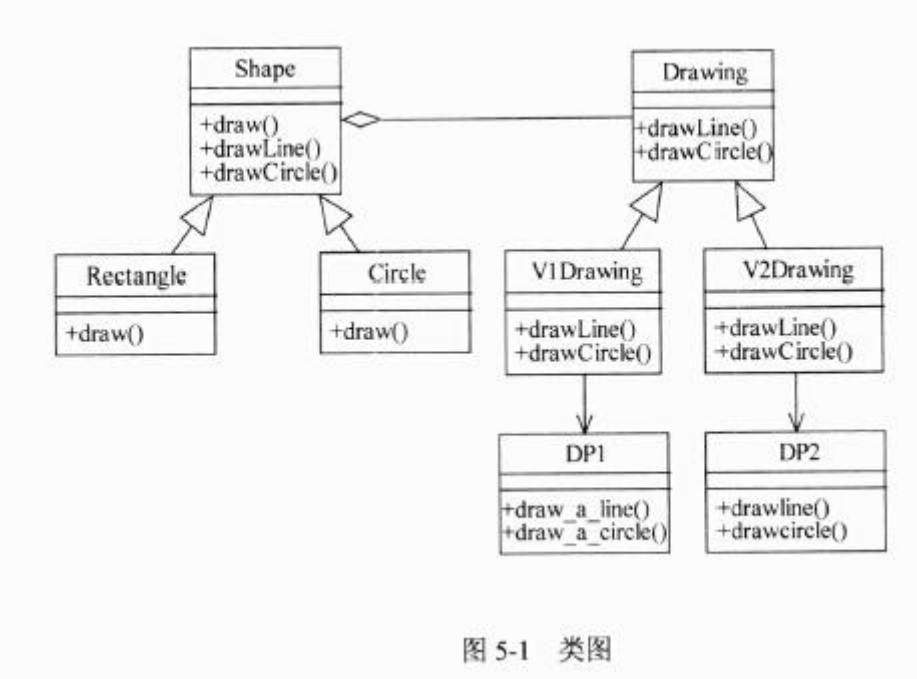


图 5-1　类图

【C++代码】

```
class DP1 {
public:
   static void draw_a_line(double x1, double y1, double x2, double y2) { /*
代码省略 */  }
   static void draw_a_circle(double x, double y, double r) {  /* 代码省略
*/  }
};
class DP2 {
public:
   static void drawline(double x1, double x2, double y1, double y2) {  /*
代码省略 */  }
   static void drawcircle(double x, double y, double r) {  /* 代码省略 */  }
};
class Drawing {
public:
   ____(1)____;
   ____(2)____;
};
class V1Drawing : public Drawing {
public:
   void drawLine(double x1, double y1, double x2, double y2) { /* 代码
省略 */ }
   void drawCircle(double x, double y, double r) {  ____(3)____;  }
};
class V2Drawing : public Drawing {
public:
   void drawLine(double x1, double y1, double x2, double y2) { /* 代码
省略 */ }
   void drawCircle(double x, double y, double r) {  ____(3)____;  }
};
class V2Drawing : public Drawing {
public:
   void drawLine(double x1, double y1, double x2, double y2) { /* 代码
省略 */ }
   void drawCircle(double x, double y, double r) {  ____(4)____;  }
};
class Shape {
public:
   ____(5)____;
   Shape(Drawing *dp) { _dp = dp; }
   void drawLine(double x1, double y1, double x2, double y2) {
       _dp->drawLine(x1, y1, x2, y2);  }
   void drawCircle(double x, double y, double r) { _dp->drawCircle(x, y,
r); }
private:  Drawing *_dp;
};
```

```
class Rectangle : public Shape {
public:
  void draw() {  /* 代码省略 */  }
// 其余代码省略
};

class Circle : public Shape {
private:  double _x, _y, _r;
public:
  Circle(Drawing *dp, double x, double y, double r) :____(6)____ {  _x =
x; _y = y; _r = r;  }
  void draw() { drawCircle(_x, _y, _r); }
};
```

【问题 1】

阅读说明和 C++代码，将应填入（n)处的字句写在答题纸的对应栏内。

(1) virtual void draw Line (doublex1,doubley1,doublex2,doubley1) = 0

(2) virtua lvoid draw Circle (donblex,doubley,doubler) = 0

(3) DPI::draw_a—circle(x,y,r)

(4) DP2::draw circle(x,y,r)

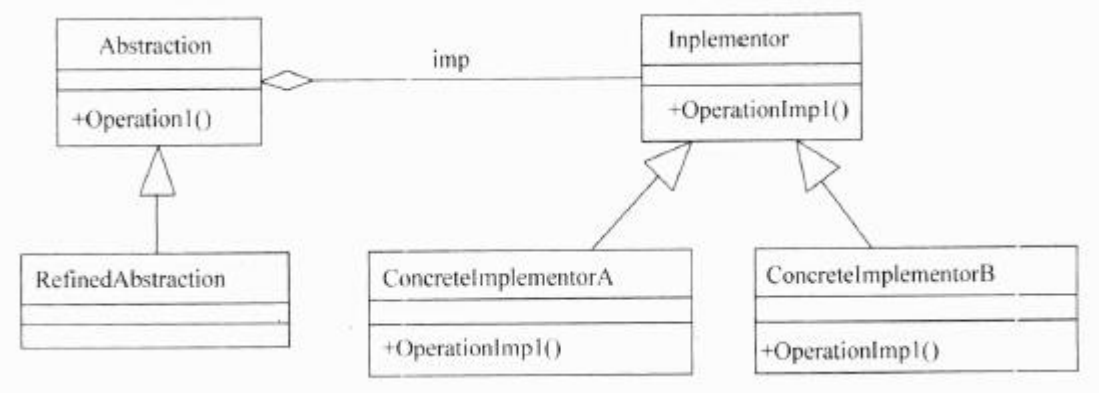
(5) virtua lvoid draw()=0

(6) Shape(dp)

本题考查桥接（Bridge)模式的概念及应用。

Bridge 模式可以将复杂的组件分成两个独立的、

的抽象和内部实现。改变组件的这两个层次结构很简单，以至于它们可以相互独立地变化。当具有抽象的层次结构和相应的实现层次结构时，Bridge 模式是非常有用的。除了可以将抽象和实现组合成许多不同的类，该模式还可以动态组合的独立类的形式实现这些抽象和实现。下图所示是 Bridge 模式的类图。



在以下情况中，应该使用 Bridge 模式：

•想避免在抽象及其实现之间存在永久的绑定；

•抽象及其实现可以使用子类进行扩展；

•抽象的实现被改动应该对客户端没有影响，也就是说，不需要重新编译代码。

本题中，类 Shape 对应上图中的 Abstraction,表示抽象部分；类 Drawing 对应

Implementor,表示实现部分。这两个类的子类分别表示具体的抽象部分和实现部分。在 C++中，Drawing 可以用抽象类来实现，将其中的方法定义为纯虚拟函数。因此（1)、(2)分别应为“ virtualvoiddrawLine(doublex1,doubley1,doublex2,doubley1)=0 ”、‘virtualvoiddrawCircle(doublex,doubley,doubler)=0”。

VIDrawing 是绘图实现类之一，它采用的绘图程序由是 DPI 所提供的。DPI 中的方法均为静态方法，必须用类名来引用。因此（3)处应为 DP1::draw__a_circle(x，y,r)。同理（4)处应为“DP2::drawcircle(x，y，r)”。

由类图可以看出，Shape 类中定义的方法 draw 在其子类中被重置了，而 Shape 表示的是抽象部分，可以将 draw 方法定义为纯虚拟函数。所以，(5)应该为“virtaalvoiddraw()=0”。

空(6)处考查继承结构中子类构造函数的定义。构造子类对象时，需要调用基类的构造函数，这可以通过初始化列表显式指明需要调用的基类的构造函数。在本题中，Shape 类只定义了一个构造函数，因此（6)应该为“Shape(dp)”。

试题六

欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。以绘制直线和圆形为例，对应的绘图程序如表 6-1 所示。

表 6-1　不同的绘图程序

	DP1	DP2
绘制直线	draw_a_line(x1,y1,x2,y2)	drawline(x1,x2,y1,y2)
绘制圆	draw_a_circle(x, y, r)	drawcircle(x, y, r)

该绘图软件的扩展性要求，将不断扩充新的图形和新的绘图程序。为了避免出现类爆炸的情况，现采用桥接（Bridge)模式来实现上述要求，得到如图 6-1 所示的类图。

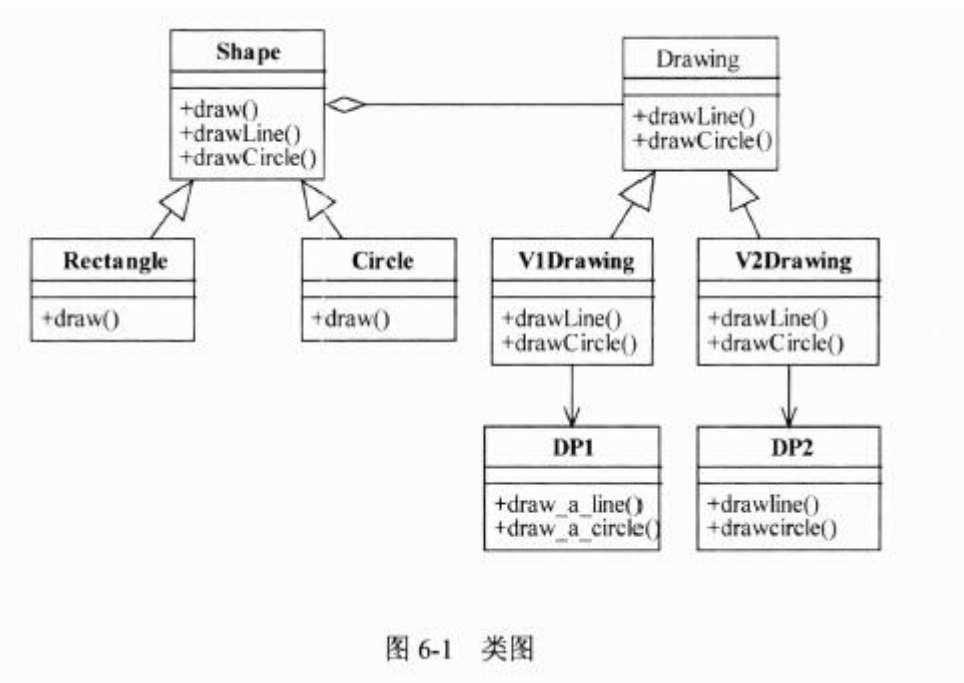


图 6-1　类图

【Java 代码】

```
    (1)     Drawing {
     (2)    ;
     (3)    ;
}

class DP1{
   static public void draw_a_line(double x1, double y1, double x2, double
y2)
{/*代码省略 */ }
   static public void draw_a_circle (double x, double y, double r) { /*
代码省略 */ }
}

class DP2{
   static public void drawline(double x1, double y1, double x2, double y2)
{/*代码省略 */ }
   static public void drawcircle (double x, double y, double r) {/*代码
省略 */ }
}

class V1Drawing implements Drawing {
   public void drawLine(double x1, double y1, double x2, double y2) { /*
代码省略 */ }
```

```
    public void drawCircle(double x, double y, double r) {      (4)     ;  }
}

class V2Drawing implements Drawing {
    public void drawLine(double x1, double y1, double x2, double y2) { /*
代码省略 */ }
    public void drawCircle(double x, double y, double r) {      (5)     ;  }
}

abstract class Shape {
  private Drawing _dp;
      (6)    ;
  Shape(Drawing dp) { _dp = dp; }
  public void drawLine(double x1, double y1, double x2, double y2) {
      _dp.drawLine(x1, y1, x2, y2);  }
  public void drawCirle(double x, double y, double r) { _dp.drawCircle(x,
y, r);  }
}

class Rectangle extends Shape {
    private double _x1, _x2, _y1, _y2;
    public Rectangle (Drawing dp, double x1, double y1, double x2, double
y2)
{ /* 代码省略 */ }
    public void draw() { /* 代码省略 */ }
}

class Circle extends Shape {
    private double _x, _y, _r;
    public Circle(Drawing dp, double x, double y, double r) { /* 代码省略
*/ }
    public void draw() { drawCirle(_x, _y, _r);    }
}
```

【问题 1】

阅读说明和 Java 代码，将应填入（n)处的字句写在答题纸的对应栏内。

(1) interface

(2) void draw Line(doublex1, doubley1,doublex2,doubley2)

(3) void draw Circle(doublex,doubley,doubler)

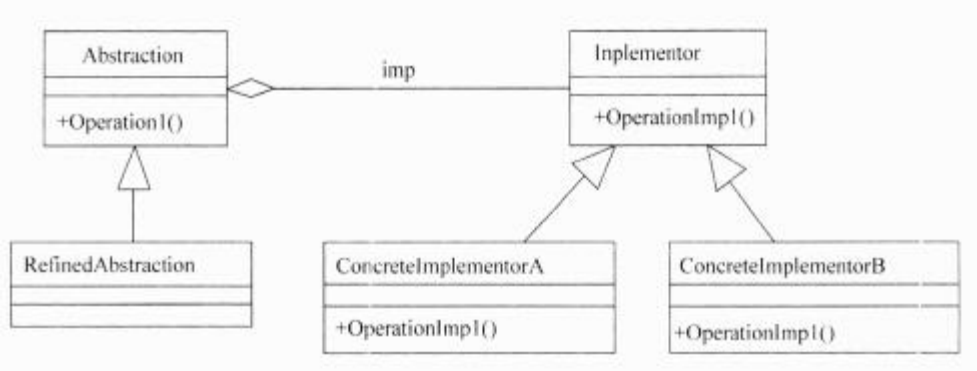
(4) DPI.draw_a_circle(x,y,r)

(5) DP2.draw circle(x,y,r)

(6) abstract public void draw()

本题考查桥接（Bridge)模式的概念及应用。

Bridge 模式可以将复杂的组件分成两个独立的但又相关的继承层次结构：功能性的抽象和内部实现。改变组件的这两个层次结构很简单，以至于它们可以相互独立地变化。当具有抽象的层次结构和相应的实现层次结构时，Bridge 模式是非常有用的。除了可以将抽象和实现组合成许多不同的类，该模式还可以以动态组合的独立类的形式实现这些抽象和实现。下图所示是 Bridge 模式的类图。



在以下情况中，应该使用 Bridge 模式：

•想避免在抽象及其实现之间存在永久的绑定；

•抽象及其实现可以使用子类进行扩展；

•抽象的实现被改动应该对客户端没有影响，也就是说，不需要重新编译代码。

本题中，类 Shape 对应上图中的 Abstraction，表示抽象部分；类 Drawing 对应 Implementor，表示实现部分。这两个类的子类分别表示具体的抽象部分和实现部分。类 Drawing 为具体的实现类提供统一接口，在 Java 中可以使用接口来实现。因此(1)、(2)、(3) 分别应为“interface”、“void drawLine (doublexl,doubleyl,doublex2,doubley2)”、“void draw Circle (doublex,doubley,doubler)’’。

VIDrawing 是绘图实现类之一，它采用的绘图程序由是 DPI 所提供的。3 此（4)处应为“DPl.draw_a_circle(x，y,r)”。同理（5)处应为“DP2.drawcircle(x,y，r:>”。

由类图可以看出，Shape 类中定义的方法 draw 在其子类中被重置了，而 Shape 表示的是抽象部分，可以将 draw 方法定义为抽象函数。所以，(6)应该为“abstract public void draw()’