

考点分析

第1章操作系统

1.1考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 操作系统的类型和结构。
- (2) 操作系统基本原理。
- (3) 网络操作系统及网络管理。

在这些知识点中，考查重点是操作系统的概念及特点、进程管理、存储管理、文件管理。具体考查知识点分布情况如表1-1所示。

表1-1历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题1~4	操作系统基本概念、前趋图与PV操作
2010年11月试题1~4	微内核操作系统、文件系统、页面置换算法
2011年11月试题1~4	操作系统接口、前趋图与PV操作
2012年11月试题1~4	PV操作与信息量、索引文件

试题1 (2009年11月试题1)

1.2 试题精解

试题1 (2009年11月试题1)

计算机系统中硬件层之上的软件通常按照三层来划分，如图1-1所示，图中①②③分别表示

(1) .

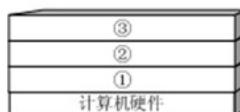


图 1-1 计算机结构图

- (1) A.操作系统、应用软件和其他系统软件
- B.操作系统、其他系统软件和应用软件
- C.其他系统软件、操作系统和应用软件
- D.应用软件、其他系统软件和操作系统

试题分析

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，如图1-2所示。

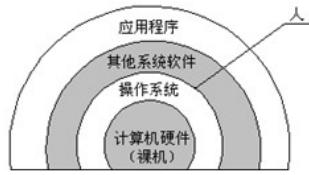


图 1-2 计算机结构图（完整）

从图1-2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，其他系统软件属于第二层，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（这些软件工作于操作系统之上，可服务于应用软件，所以有别于应用软件）；大量的应用软件属于第三层，例如希赛教育网上辅导平台，常见的一系列MIS系统等。其他系统软件和应用软件都是建立在操作系统基础之上的，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。

试题答案

(1) B

第1章：操作系统

试题2（2009年11月试题2~4）

试题2（2009年11月试题2~4）

某计算机系统中有一个CPU、一台扫描仪和一台打印机。现有三个图像任务，每个任务有三个程序段：扫描Si,图像处理Ci和打印Pi (i=1,2,3)。图1-3为三个任务各程序段并发执行的前趋图，其中，(2) 可并行执行，(3) 的直接制约，(4) 的间接制约。

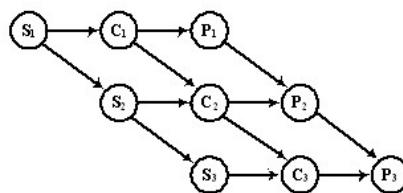


图1-3 前趋图

(2) A."C1S2","P1C2S3","P2C3" B."C1S1","S2C2P2","C3P3"

C."S1C1P1","S2C2P2","S3C3P3" D."S1S2S3","C1C2C3","P1P2P3"

(3) A.S1受到S2和S3、C1受到C2和C3、P1受到P2和P3

B.S2和S3受到S1、C2和C3受到C1、P2和P3受到P1

C.C1和P1受到S1、C2和P2受到S2、C3和P3受到S3

D.C1和S1受到P1、C2和S2受到P2、C3和S3受到P3

(4) A.S1受到S2和S3、C1受到C2和C3、P1受到P2和P3

B.S2和S3受到S1、C2和C3受到C1、P2和P3受到P1

C.C1和P1受到S1、C2和P2受到S2、C3和P3受到S3

D.C1和S1,受到P1、C2和S2受到P2、C3和S3受到P3

试题分析

如图1-3所示，当S1执行完毕后，计算C1与扫描S2可并行执行；C1与S2执行完毕后，打印P1、计算C2与扫描S3可并行执行；P1、C2与S3执行完毕后，打印P2与计算C3可并行执行。

根据题意，系统中有三个任务，每个任务有三个程序段，从前趋图中可以看出，系统要先进行扫描Si，然后再进行图像处理Ci，最后进行打印Pi，所以C1和P1受到S1直接制约、C2和P2受到S2的直接制约、C3和P3受到S3的直接制约。

系统中有一台扫描仪，因此S2和S3不能运行是受到了S1的间接制约。如果系统中有三台扫描仪，那么S2和S1能运行；同理，C2和C3受到C1的直接制约、P2和P3受到P1的间接制约。

试题答案

(2) A (3) C (4) B

第1章：操作系统

试题3 (2010年11月试题1)

试题3 (2010年11月试题1)

采用微内核结构的操作系统提高了系统的灵活性和可扩展性，(1)。

- (1) A.并增强了系统的可靠性和可移植性，可运行于分布式系统中
- B.并增强了系统的可靠性和可移植性，但不适用于分布式系统
- C.但降低了系统的可靠性和可移植性，可运行于分布式系统中
- D.但降低了系统的可靠性和可移植性，不适用于分布式系统

试题分析

现代操作系统大多拥有两种工作状态，分别是核心态和用户态。一般应用程序工作在用户态，而内核模块和最基本的操作系统核心工作在核心态。

微内核操作系统结构是20世纪80年代后期发展起来的。操作系统的一个发展趋势是将传统的操作系统代码放置到更高层，从操作系统中去掉尽可能多的东西，而只留下一个最小的核心，称之为微内核。通常的方法是将大多数操作系统功能由在用户态运行的服务器进程来实现。为了获取某项服务，用户进程（客户进程）将请求发送给一个服务器进程，服务器进程完成此操作后，把结果返回给用户进程。这样，服务器以用户进程的形式运行，而不是运行在核心态。因此，它们不能直接访问硬件，某个服务器的崩溃不会导致整个系统的崩溃。客户/服务器结构的另一个优点是它更适用于分布式系统。

微内核技术的主要优点如下：

- (1) 统一的接口，在用户态和核心态之间无需进程识别。
- (2) 可伸缩性好，能适应硬件更新和应用变化。
- (3) 可移植性好，所有与具体机器特征相关的代码，全部隔离在微内核中，如果操作系统要移植到不同的硬件平台上，只需修改微内核中极少代码即可。
- (4) 实时性好，微内核可以方便地支持实时处理。

(5) 安全可靠性高，微内核将安全性作为系统内部特性来进行设计，对外仅使用少量应用编程接口。

(6) 支持分布式系统，支持多处理器的体系结构和高度并行的应用程序。

虽然微内核操作系统具有诸多优点，但它并非完美无缺。例如，在运行效率方面，它就不如以前传统的操作系统。

试题答案

(1) A

第1章：操作系统

试题4（2010年11月试题2）

试题4（2010年11月试题2）

若操作系统文件管理程序正在将修改后的（2）文件写回磁盘时系统发生崩溃，对系统的影响相对较大。

(2) A. 用户数据 B. 用户程序 C. 系统目录 D. 空闲块管理

试题分析

操作系统为了实现“按名存取”，必须为每个文件设置用于描述和控制文件的数据结构，专门用于文件的检索，因此至少要包括文件名和存放文件的物理地址，该数据结构称为文件控制块（File Control Block, FCB），文件控制块的有序集合称为文件目录，或称为系统目录文件。若操作系统正在将修改后的系统目录文件写回磁盘时系统发生崩溃，则对系统的影响相对较大。

试题答案

(2) C

第1章：操作系统

试题5（2010年11月试题3~4）

试题5（2010年11月试题3~4）

某虚拟存储系统采用最近最少使用的（LRU）页面淘汰算法，假定系统为每个作业分配4个页面的主存空间，其中一个页面用来存放程序。现有某作业的程序如下：

```
Var A: Array[100,100] OF integer;  
i,j: integer;  
FOR i:=1 to 100 DO  
FOR j:=1 to 100 DO
```

A[i,j]:=0;

设每个页面可存放200个整数变量，变量i、j存放在程序页中。初始时，程序及i、j均已在内存，其余3页为空。若矩阵A按行序存放，那么当程序执行完后共产生（3）次缺页中断；若矩阵A按列序存放，那么当程序执行完后共产生（4）次缺页中断。

- (3) A.50B.100C.5000 D.10000
- (4) A.50B.100C.5000 D.10000

试题分析

虚拟存储管理的提出就是为了解决这一问题，应用程序在运行之前并不必全部装入内存，仅需将当前运行到的那部分程序和数据装入内存便可启动程序的运行，其余部分仍驻留在外存上。当要执行的指令或访问的数据不在内存时，再由操作系统通过请求调入功能将它们调入内存，以使程序能继续执行。如果此时内存已满，则还需通过置换功能，将内存中暂时不用的程序或数据调至外存上，腾出足够的内存空间后，再将要访问的程序或数据调入内存，使程序继续执行。这样，便可使一个大的用户程序能在较小的内存空间中运行，也可在内存中同时装入更多的进程使它们并发执行。从用户的角度看，该系统具有的内存容量比实际的内存容量大得多。将这种具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的存储器系统称为虚拟存储系统。

1. 局部性原理

虚拟存储管理能够在作业信息不全部装入内存的情况下保证作业正确运行，是利用了程序执行时的局部性原理。局部性原理是指程序在执行时呈现出局部性规律，即在一较短的时间内，程序的执行仅局限于某个部分。相应地，它所访问的存储空间也仅局限于某个区域。程序局部性包括时间局部性和空间局部性，时间局部性是指程序中的某条指令一旦执行，不久以后该指令可能再次执行。产生时间局部性的典型原因是由于程序中存在着大量的循环操作；空间局部性是指一旦程序访问了某个存储单元，不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址可能集中在一定的范围内，其典型情况是程序顺序执行。

2. 工作集

在虚拟存储管理中，可能会出现这种情况，即对于刚被替换出去的页，立即又要被访问，需要将它调入，因无空闲内存又要替换另一页，而后者是即将被访问的页，于是造成了系统需花费大量的时间忙于进行这种频繁的页面交换，致使系统的实际效率很低，严重时导致系统瘫痪，这种现象称为抖动现象。防止抖动现象有多种办法，例如，采取局部替换策略、引入工作集算法和挂起若干进程等。工作集是指在某段时间间隔内，进程实际要访问的页面的集合。引入虚拟内存后，程序只需有少量的内存就可运行，但为了使程序有效地运行，较少产生缺页，必须使程序的工作集全部在内存中。

3. 页面置换算法

当内存中没有空闲页面，而又有程序和数据需要从外存中装入内存运行时，就需要从内存中选出一个或多个页面淘汰出去，以便新的程序和数据装入运行，良好的页面置换算法应该淘汰那些被访问概率最低的页，将它们移出内存。

- (1) 随机淘汰算法。无法确定哪些页被访问的概率较低时，随机地选择某个页面，并将其换出。
- (2) 轮转算法。按照内存页面的编号，循环地换出内存中一个可以被换出的页，无论该页是刚换进来还是已驻留内存很长时间。
- (3) 先进先出算法（First In First Out,FIFO）。FIFO算法总是选择在内存驻留时间最长的一页将其淘汰。实现FIFO算法需要把各个已分配页面按页面分配时间顺序链接起来，组成FIFO队列，并

设置一置换指针，指向FIFO队列的队首页面。FIFO算法忽略了一种现象的存在，那就是在内存中停留时间最长的页往往也是经常要访问的页。将这些页淘汰，很可能刚置换出去，又请求调用该页，致使缺页中断太频繁，严重降低内存的利用率。

FIFO的另一个缺点是它可能会产生一种异常现象。一般来说，对于任一作业或进程，如果给它分配的内存页面数越接近于它所要求的页面数，则发生缺页的次数会越少。但使用FIFO算法时，有时会出现分配的页面数增多，缺页次数反而增加的现象，称为belady现象。

(4) 最近最久未使用算法 (Least Recently Used, LRU)。当需要淘汰某一页时，选择离当前时间最近的一段时间内最久没有使用过的页先淘汰。例如，考虑一个仅460个字节的程序的内存访问序列 (10,11,104,170,73,309,185,245,246,434,458,364)，页面的大小为100个字节，则460个字节应占5页，编号为0~4，第0页字节为0~99，第1页为100~199，依此类推。得到页面的访问序列是 (0,0,1,1,0,3,1,2,2,4,4,3)，可简化为 (0,1,0,3,1,2,4,3)。如果内存中有200个字节可供程序使用，则内存提供2个页帧供程序使用。按照FIFO算法，共产生6次缺页中断，如表1-2所示。

表1-2 FIFO算法缺页中断

0	1	0	3	1	2	4	3
0	0	0	3	3	3	4	4
	1	1	1	1	2	2	3
×	×		×		×	×	×

按照LRU算法，共产生7次缺页中断，如表1-3所示。

表1-3 LRU算法缺页中断

0	1	0	3	1	2	4	3
0	0	0	0	1	1	4	4
	1	1	3	3	2	2	3
×	×		×	×	×	×	×

(5) 最近没有使用页面置换算法 (No Used Recently, NUR)。在需要置换某一页时，从那些最近的一个时期内未被访问的页任选一页置换。只要在页表中增设一个访问位即可实现。当某页被访问时，访问位置为1，否则访问位置为0。系统周期性地对所有引用位清零。当需淘汰一页时，从那些访问位为零的页中选一页进行淘汰。

(6) 最优置换算法。选择那些永久不使用的，或者在最长时间内不再被访问的页面置换出去。因为要确定哪个页面是未来最长时间内不再被访问的，目前来说很难估计，所以，该算法通常用来评价其他算法。

(7) 时钟页面替换算法 (Clock)。使用页表中的引用位，将作业已调入内存的页面链成循环队列，用一个指针指向循环队列中的下一个将被替换的页面。其实现方法如下：一个页面首次装入内存时，其引用位置1；在内存中的任何一个页面被访问时，其引用位置1；淘汰页面时，存储管理从指针当前指向的页面开始扫描循环队列，把所遇到的引用位是1的页面的引用位清0，并跳过这个页面；把所遇到的引用位是0的页面淘汰掉，指针推进一步；扫描循环队列时，如果遇到的所有页面的引用位均为1，则指针就会绕整个循环队列一圈，将碰到的所有页面的引用位清0；指针停在起始位置，并淘汰掉这一页，然后指针推进一步。

在本题中，从题干可知，作业共有4个页面的主存空间，其中一个已被程序本身占用，所以在读取变量时可用的页面数只有3个。每个页面可存放200个整数变量，程序中A数组共有 $100 \times 100 = 10000$ 个变量。按行存放时，每个页面调入的200变量刚好是程序处理的200个变量，所以缺页次数为 $10000 / 200 = 50$ 。而按列存放时，虽然每个页面调取数据时，同样也读入了200个变

量，但这200个变量中，只有2个是近期需要访问的（如：第1个页面调入的是A[*],1与A[*],2，但程序近期需要访问的变量只有A[1,1]和A[1,2]），所以缺页次数为：10000/2=5000。

试题答案

(3) A (4) C

第1章：操作系统

试题6 (2011年11月试题1)

试题6 (2011年11月试题1)

操作系统为用户提供了两类接口：操作一级和程序控制一级的接口，以下不属于操作一级的接口是（1）。

(1) A.操作控制命令 B.系统调用 C.菜单 D.窗口

试题分析

操作系统是用户和计算机之间的接口，用户通过操作系统的帮助可以快速、有效和安全可靠地使用计算机各类资源。通常操作系统提供两类接口，分别是程序一级的接口（程序接口）和操作一级的接口（联机用户接口和脱机用户接口）。

用户与操作系统的接口通常是由“命令”和“系统调用”的形式表现出来的。命令是提供给用户在键盘终端上使用（命令接口），系统调用是用户在编程时使用（程序接口）。

在不同的系统中，系统调用的实现方式可能不同，但大体上都可以把系统调用的执行过程分成以下几步。

1、设置系统调用号和参数

在一个系统中，往往都设置了许多条系统调用命令，并赋予每条系统调用命令一个唯一的系统调用号。设置系统调用方式有2种方式：

(1) 直接将参数送入相应的寄存器中，这是最简单的一种方式。这种方式的主要问题是由于寄存器数量有限，从而限制了设置参数的数目。

(2) 参数表方式。将系统调用所需要的参数，放入一张参数表中，再将只想该参数表的指针放在某个规定的寄存器中。

2、系统调用命令的一般性处理

为了使不同系统调用能方便地转向相应的命令处理程序，在系统中配置了一张系统调用入口表。表中每个表目都对应一条系统调用命令，核心可利用系统调用号去查找该表，就可以找到相应命令处理程序的入口地址而去执行它。

3、系统调用命令处理程序的处理过程

为了提供系统调用的功能，操作系统内必须有事先编制好的实现这些功能的子程序或过程。这些程序是操作系统程序模块的一部分，且不能直接被用户程序调用。

程序员给定了系统调用名和参数之后是怎样得到系统服务的呢？这需要有一个类似于硬件终端处理的中断处理机构。当用户使用系统调用时，产生一条相应的指令，处理机在执行到该指令时发

生相应的中断，并发出有关信号给给该处理机构。该处理机构在收到了处理机发来的信号后，启动相关的处理程序去完成该系统调用所要求的功能。

在系统中为控制系统调用服务的机构称为陷阱处理机构。与此相对应，把由于系统调用引起处理中断的指令为陷阱指令。在操作系统中，每个系统调用都对应一个功能号。在陷阱指令中必须包括对应系统调用的功能号。而且，在有些陷阱指令中，还带有传递给陷阱处理机构和内部处理程序的有关参数。

为了实现系统调用，系统设计人员还必须为实现各种系统调用功能的子程序编造入口地址表，每个入口地址都与相应的系统子程序名对应起来。然后，由陷阱处理程序把陷阱指令中所包含的功能号与该入口地址表转念馆的有关项对应起来，从而由系统调用功能号驱动有关系统子程序执行。

由于在系统调用处理结束之后，用户程序还需利用系统调用的返回结果继续执行，因此，在进入系统调用处理之前，陷阱处理机构还需保存处理机现场。再者，在系统调用处理结束之后，陷阱处理机构还要回复处理机现场。在操作系统中，处理机的现场一般被保护在特定的内存区或寄存器中。

试题答案

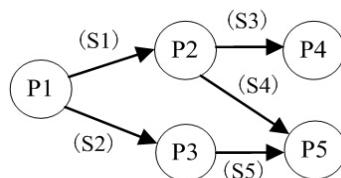
(1) B

第1章：操作系统

试题7 (2011年11月试题2~4)

试题7 (2011年11月试题2~4)

进程P1、P2、P3、P4和P5的前趋图如图1-4所示。



若用PV操作控制进程P1~P5并发执行的过程，则需要设置5个信号量S1、S2、S3、S4和S5，进程间同步所使用的信号量标注在图1-4中的边上，且信号量S1~S5的初值都等于零，初始状态下进程P1开始执行。图1-5中a、b和c处应分别填写 (2)；d和e处应分别填写 (3)，f和g处应分别填写 (4)。

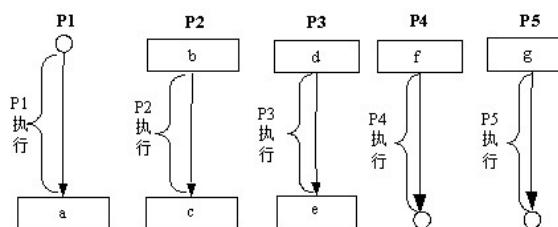


图1-5 PV操作示意图

(2) A. V(S1) V(S2)、P(S1) 和 V(S3) V(S4) B. P(S1) V(S2)、P(S1) 和 P(S2) V(S1)

- C.V (S1) V (S2)、P (S1) 和P (S3) P (S4) D.P (S1) P (S2)、V (S1) 和P (S3) V (S2)
- (3) A.P (S1) 和V (S5) B.V (S1) 和P (S5)
- C.P (S2) 和V (S5) D.V (S2) 和P (S5)
- (4) A.P (S3) 和V (S4) V (S5) B.P (S3) 和P (S4) P (S5)
- C.V (S3) 和V (S4) V (S5) D.V (S3) 和P (S4) P (S5)

试题分析

在多道程序系统中，由于资源共享与进程合作，使各进程之间可能产生两种形式的制约关系，一种是间接相互制约，例如，在仅有一台打印机的系统中，有两个进程A和B，如果进程A需要打印时，系统已将打印机分配给进程B，则进程A必须阻塞；一旦进程B将打印机释放，系统便将进程A唤醒，使之由阻塞状态变为就绪状态；另一种是直接相互制约，例如，输入进程A通过单缓冲区向进程B提供数据。当该缓冲区为空时，进程B不能获得所需的数据而阻塞，一旦进程A将数据送入缓冲区中，进程B就被唤醒。反之，当缓冲区满时，进程A就被阻塞，仅当进程B取走缓冲区中的数据时，才唤醒进程A。

进程同步主要源于进程合作，是进程之间共同完成一项任务时直接发生相互作用的关系，为进程之间的直接制约关系。在多道程序系统中，这种进程间在执行次序上的协调是必不可少的；进程互斥主要源于资源共享，是进程之间的间接制约关系。在多道程序系统中，每次只允许一个进程访问的资源称为临界资源，进程互斥要求保证每次只有一个进程使用临界资源。在每个进程中访问临界资源的程序段称为临界区，进程进入临界区要满足一定的条件，以保证临界资源的安全使用和系统的正常运行。

1. 信号量

信号量是一个二元组 (S, Q)，其中 S 是一个整形变量，初值为非负数，Q 为一个初始状态为空的等待队列。在多道程序系统中，信号量机制是一种有效的实现进程同步与互斥的工具。信号量的值通常表示系统中某类资源的数目，若它大于0，则表示系统中当前可用资源的数量；若它小于0，则表示系统中等待使用该资源的进程数目，即在该信号量队列上排队的PCB的个数。信号量的值是可变的，由PV操作来改变。

PV操作是对信号量进行处理的操作过程，而且信号量只能由PV操作来改变。P操作是对信号量减1，意味着请求系统分配一个单位资源，若系统无可用资源，则进程变为阻塞状态；V操作是对信号量加1，意味着释放一个单位资源，加1后若信号量小于等于0，则从就绪队列中唤醒一个进程，执行V操作的进程继续执行。

对信号量 S 进行 P 操作，记为 P (S)；对信号量 S 进行 V 操作，记为 V (S)。P (S) 和 V (S) 的处理过程如表1-4所示。

表1-4 P (S) 和 V (S) 的处理过程

P(S)	V(S)
<pre> S=S-1; if(S<0) { 当前进程进入等待队列 Q; 阻塞当前进程; } else 当前进程继续; </pre>	<pre> S=S+1; if(S<=0) { 从等待队列 Q 中取出一个进程 P; 进程 P 进入就绪队列; 当前进程继续; } else 当前进程继续; </pre>

2. 实现互斥模型

使用信号量机制实现进程互斥时，需要为临界资源设置一个互斥信号量S,其初值通常为1.在每个进程中将临界区代码置于P (S) 和V (S) 之间。必须成对使用PV原语，缺少P原语则不能保证互斥访问，缺少V原语则不能在使用临界资源之后将其释放。而且，PV原语不能次序颠倒、重复或遗漏。

3. 实现同步模型

使用信号量机制实现进程同步时，需要为进程设置一个同步信号量S,其初值通常为0.在进程需要同步的地方分别插入P (S) 和V (S) 。一个进程使用P原语时，则另一个进程往往使用V原语与之对应。具体怎么使用要根据实际情况决定，下面举个简单例子来加以说明。

有两个进程P1和P2,P1的功能是计算 $x=a+b$ 的值，a和b是常量，在P1的前面代码中能得到；P2的功能是计算 $y=x+1$ 的值。若这两个进程在并发执行，则有同步关系：P2要执行 $y=x+1$ 时必须等到P1已经执行完 $x=a+b$ 语句。P2进程可能会因为要等待x的值而阻塞，如果是这样的话，P1进程就要在计算出x的值后唤醒P2进程。因此，为了使P1和P2正常运行，用信号量来实现其同步的过程如表1-5所示。

表1-5 P1和P2的同步过程

P1	P2
$x=a+b;$ $V(S);$ \dots	\dots $P(S);$ $y=x+1;$ \dots

再举一个较为复杂的例子，以加深对PV操作的理解。设有两个并发进程Read和Print,Read负责从输入设备读入信息到一个容量为N的缓冲区，Print负责从缓冲区中取出信息送打印机输出。设置信号量mutex的初值为1,empty的初值为N,full的初值为0,则程序如表1-6所示。

表1-6 实现Read和Print的程序

Read	Print
$\begin{array}{l} \text{begin} \\ \text{P(empty);} \\ \text{P(mutex);} \\ \text{读入;} \\ \text{V(mutex);} \\ \text{V(full);} \\ \text{end} \end{array}$	$\begin{array}{l} \text{begin} \\ \text{P(full)} \\ \text{P(mutex);} \\ \text{输出} \\ \text{V(mutex);} \\ \text{V(empty)} \\ \text{end} \end{array}$

在本题中，从题目的前趋图，可以得知以下约束关系：

- (1) P1执行完毕，P2与P3才能开始；
- (2) P2执行完毕，P4才能开始；
- (3) P2与P3都执行完，P5才能开始。

分析清楚这种制约关系，解题也就容易了。

(1) 从“P1执行完毕，P2与P3才能开始”可以得知：P2与P3中的b与d位置，分别应填P (S1) 和P (S2)，以确保在P1执行完毕以前，P2与P3不能执行。当然当P1执行完毕时，应该要对此解锁，所以P1中的a位置应填V (S1) 与V (S2)。

(2) 从“P2执行完毕，P4才能开始”可以得知：P4的f位置，应填P (S3)，而P2的结束位置c应有V (S3)。

(3) 从" P2与P3都执行完，P5才能开始"可以得知：P5的g位置，应填P(S4)与P(S5)，而对应的P2的结束位置c应有V(S4)，结合前面的结论可知，c应填V(S3)与V(S4)。而e应填V(S5)。

试题答案

(2) A (3) C (4) B

第1章：操作系统

试题8 (2012年11月试题1~2)

试题8 (2012年11月试题1~2)

假设系统中有n个进程共享3台打印机，任一进程在任一时刻最多只能使用1台打印机。若用PV操作控制n个进程使用打印机，则相应信号量s的取值范围为 (1)；若信号量S的值为-3，则系统中有 (2) 个进程等待使用打印机。

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

试题分析

信号量是PV操作中的一种特殊变量，该变量的值指示一类资源的数量，当信号量的值为负数时，又能展示出目前系统中有多少个进程在等待该资源。

在本题中，系统有n个进程，有3台打印机。初始状态时，没有1个进程使用打印机，此时信号量s应为3，代表有3台打印机资源可用。而如果此时有1个进程占用了1台打印机，则信号量s变为2，代表目前只有2台打印机可用，依此类推。信号量的最小值为-(n-3)，即表示当前状态为：3个进程占用了3台打印机资源，而剩余的n-3个进程都在等待打印机资源。所以s的取值范围是：3,2,1,0,-1,...,-(n-3)。

有了前面的分析，接下来这一问就非常好回答了。信号量为-3，表示有3个进程在等待使用打印机。

试题答案

(1) B (2) D

第1章：操作系统

试题9 (2012年11月试题3~4)

试题9 (2012年11月试题3~4)

假设文件系统采用索引节点管理，且索引节点有8个地址项i-addr[0]~i-addr[7]，每个地址项大小为4字节，i-addr[0]~i-addr[4]采用直接地址索引，i-addr[5]和i-addr[6]采用一级间接地址索引，i-addr[7]采用二级间接地址索引。假设磁盘索引块和磁盘数据块大小均为1KB字节，文件File1的索引节点如图1-6所示。若用户访问文件File1中逻辑块号为5和261的信息，则对应的物理块号分别为（3）；101号物理块存放的是（4）。

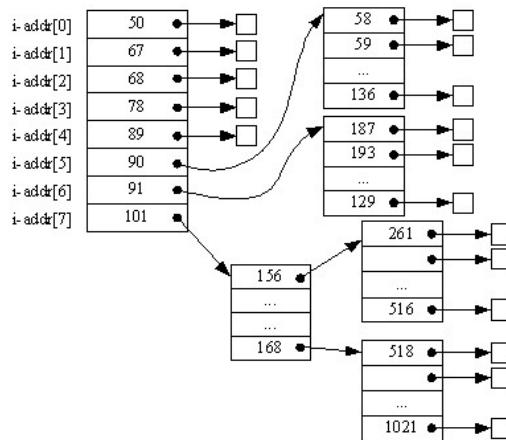


图1-6 索引文件示意图

(3) A.89和90 B.89和136 C.58和187 D.90和136

(4) A.File1的信息 B.直接地址索引表

C.一级地址索引表 D.二级地址索引表

试题分析

文件物理结构（物理文件）是指文件在存储介质上的组织方式，它依赖于物理的存储设备和存储空间，可以看作是相关物理块的集合。由于物理结构决定了信息在存储设备上的存放位置和方式，因此，信息的逻辑位置到物理位置的映射关系也是由物理结构决定的。常用的文件物理结构有顺序结构、链接结构和索引结构。

(1) 顺序结构（连续结构）。逻辑上连续的记录构成的文件分配到连续的物理块中。这种方式管理简单，存储速度快，空间利用率低，但文件记录插入或删除操作不方便，只能在文件末尾进行。

(2) 链接结构（串联结构）。将信息存放在非连续的物理块中，每个物理块均设有一个指针，指向其后续的物理块，从而使得存放同一文件的物理块链接成一个串联队列。链接方式又分为显式链接和隐式链接。显式链接的链接指针在专门的链接表中，隐式链接的指针在存放信息的物理块中。链接结构空间利用率高，且易于文件扩充，但查找效率比较低。

(3) 索引结构（随机结构）。为每个文件建立一个索引表，其中每个表项指出信息所在的物理块号，表目按逻辑记录编写顺序或按记录内某一关键字顺序排列。对于大文件，为检索方便，可以建立多级索引，还可以将文件索引表也作为一个文件（称为索引表文件）。该方式可以满足文件动态增长的要求且存取方便，但建立索引表增加了存储空间的开销，对于多级索引，访问时间开销较大。

例如，在UNIX系统中，文件的物理结构采用直接、一级、二级和三级间接索引技术，假如索引节点有13个地址项，并且规定地址项0~9采用直接寻址方法，地址项10采用一级间接寻址，地址项11采用二级间接寻址，地址项12采用三级间接寻址。每个盘块的大小为1KB，每个盘块号占4B，那

么，对于访问文件的第356168B处的数据来说，先进行简单换算 $356168/1024 \approx 348KB$ ，由于地址项0~9可直接寻址10个物理盘块，每个物理块大小为1KB，所以访问文件的前10KB范围的数据时是直接寻址。地址项10采用一次间接寻址，即地址项10里存放的是一级索引表的地址，因为每个盘块号占4B，所以该索引表可存放 $1024/4 = 256$ 个物理块的地址，所以当访问文件的10~266KB之间的数据时是一次间接寻址。由于要访问的数据是348KB，所以还有 $348 - 266 = 82KB$ 。显然地址项11足够存取这些数据，因此，最多就在地址项11而无须存取地址项12，即只需要二级间接寻址。

在本题中，索引节点共有8个地址项，共分3个梯度：直接索引，一级间接索引，二级间接索引。现在要求确认逻辑块号为5与261对应的物理块号（注意：块号是从0开始编址的）。在直接索引中，索引节点对应的物理块用于直接存放文件内容，节点中存放的地址便是物理块号的首地址，如0号逻辑块，它所对应的物理块号为50；1号逻辑块对应的物理块号为67；但5号逻辑块就已经到了一级间接索引了。在一级间接索引中，索引节点所对应的物理块并不是用于存储文件内容，而是存放物理块的地址，物理块的地址占4字节，所以一个块可以存放 $1024/4 = 256$ 个地址。5号逻辑块对应的是一级间接索引的第一个块，所以物理块号为58。依此类推，6号逻辑块对应的是59号物理块；由于 5 （直接索引的块数）+ 256 （1级间接索引中，1个物理块可容地址数）=261，这说明第91号物理块中的第1个地址，对应的是261号逻辑块（第262个逻辑块），即187号物理块对应块号为261的逻辑块。

接下来的问题比前一问更容易，从示意图可以看出，101号物理块对应的空间存储着一系列地址，而这些地址对应的物理块中存储的仍然是地址，再到下一层才是文件内容，所以101号物理块存放的是二级地址索引表。

试题答案

(3) C (4) D

第2章：数据库系统

考点分析

第2章数据库系统

2.1考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

(1) 数据库管理系统的类型、结构和性能评价。

(2) 常用的关系型数据库管理系统。

(3) 数据库模式。

(4) 数据库规范化。

(5) 分布式数据库系统，并行数据库系统。

(6) 数据仓库与数据挖掘技术。

(7) 数据库工程。

(8) 备份恢复。

从历年的考试情况来看，本章的考点比较分散，数据库的多个知识点都会涉及，但考查的难度都不大。具体考查知识点的分布情况如表2-1所示。

表2-1历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题5~8, 40	数据库设计、完整性约束、触发器、候选关键字、模式分解、分布式数据库
2010年11月试题5~8	数据库设计、完整性约束、SQL语言、关系代数
2011年11月试题5~8, 35, 36	实体联系、实体转关系模式、关系代数、数据库与数据仓库
2012年11月试题5~8, 44~48	数据库设计、范式、模式分解、事务处理

第2章：数据库系统

试题1 (2009年11月试题5)

2.2 试题精解

试题1 (2009年11月试题5)

在数据库设计的需求分析阶段应完成包括 (5) 在内的文档。

- (5) A. E-R图 B. 关系模式
C. 数据字典和数据流图 D. 任务书和设计方案

试题分析

基于数据库系统生命周期的数据库设计可分为五个阶段，分别是规划、需求分析、概念设计、逻辑设计和物理设计。

1. 规划

规划阶段的主要任务是进行建立数据库的必要性及可行性分析，确定数据库系统在企业和信息系统中的地位，以及各个数据库之间的联系。

2. 需求分析

需求分析的目标是通过调查研究，了解用户的数据和处理要求，并按一定格式整理形成需求说明书。需求说明书包括数据库所涉及的数据、数据的特征、使用频率和数据量的估计，例如，数据名、属性及其类型、主键属性、保密要求、完整性约束条件、更改要求、使用频率和数据量估计等。这些关于数据的数据称为元数据。在设计大型数据库时，这些数据通常由数据字典来管理。用数据字典管理元数据，有利于避免数据的重复或重名，以保持数据的一致性。同时，有利于提高数据库设计的质量，减轻设计者的负担。

3. 概念设计

概念设计也称为概念结构设计，其任务是在需求分析阶段产生的需求说明书的基础上，按照特定的方法将它们抽象为一个不依赖于任何DBMS的数据模型，即概念模型，通常用E-R图表示。概念模型使设计人员的注意力能够从复杂的实现细节中解脱出来，而只集中在最重要的数据的组织结构和处理模式上。为保证所设计的概念模型能正确、完全地反映用户的数据及其相互关系，便于进行所要求的各种处理，在本阶段设计中，可邀请用户参与。

在进行概念设计时，可先设计各个应用的视图，即各个应用所看到的数据及其结构，然后再进行视图集成，以形成一个单一的概念数据模型。

4. 逻辑设计

逻辑设计也称为逻辑结构设计，其任务是将概念模型转化为某个特定的DBMS上的逻辑模型。

设计逻辑结构时，首先为概念模型选定一个合适的逻辑模型（例如，关系模型、网状模型或层次模型），然后将其转化为特定DBMS支持的逻辑模型，最后对逻辑模型进行优化。

逻辑设计的目的是将概念设计阶段设计好的E-R图转换为与选用的具体机器上的DBMS所支持的数据模型相符合的逻辑结构，例如，系统如果使用的是关系型数据库，则逻辑设计阶段就需要将E-R图转换为关系模式。

5. 物理设计

物理设计也称为物理结构设计，其任务是对给定的逻辑模型选取一个最适合应用环境的物理结构，所谓数据库的物理结构，主要是指数据库在物理设备上的存储结构和存取方法。物理设计的步骤如下：

- (1) 设计存储记录结构，包括记录的组成、数据项的类型和长度，以及逻辑记录到存储记录的映射。
- (2) 确定数据存储安排。
- (3) 设计访问方法，为存储在物理设备上的数据提供存储和检索的能力。
- (4) 进行完整性和安全性的分析与设计。
- (5) 数据库程序设计。

试题答案

- (5) C

第 2 章：数据库系统

试题2 (2009年11月试题6)

试题2 (2009年11月试题6)

设有职务工资关系P（职务，最低工资，最高工资），员工关系EMP（员工号，职务，工资），要求任何一名员工，其工资值必须在其职务对应的工资范围之内，实现该需求的方法是（6）。

- (6) A.建立"EMP.职务"向"P.职务"的参照完整性约束
- B.建立"P.职务"向"EMP.职务"的参照完整性约束
- C.建立EMP上的触发器程序审定该需求
- D.建立P上的触发器程序审定该需求

试题分析

数据库的完整性是指数据库中数据的正确性和相容性。数据库完整性由各种各样的完整性约束来保证，完整性约束可以通过DBMS或应用程序来实现，基于DBMS的完整性约束作为关系模式的一部分存入数据库中。

1. 完整性约束条件

保证数据完整性的方法之一是设置完整性检查，即对数据库中的数据设置一些约束条件，这是数据的语义体现。完整性约束条件是指对数据库中数据本身的某些语法或语义限制、数据之间的逻

辑约束，以及数据变化时应遵守的规则等。所有这些约束条件一般均以谓词逻辑形式表示，即以具有真假值的原子公式和命题连接词（并且、或者、否定）所组成的逻辑公式表示。完整性约束条件的作用对象可以是关系、元组或属性三种。数据的完整性约束条件一般在关系模式中给出，并在运行时做检查，当不满足条件时立即向用户通报，以便采取措施。

数据库中数据的语法、语义限制与数据之间的逻辑约束称为静态约束，它反映了数据及其之间的固有逻辑特性，是最重要的一类完整性约束。静态约束包括静态属性级约束（对数据类型的约束、对数据格式的约束、对取值范围或取值集合的约束、对空值的约束、其他约束）、静态元组约束和静态关系约束（实体完整性约束、参照完整性约束、函数依赖约束、统计约束）。

数据库中的数据变化应遵守的规则称为数据动态约束，它反映了数据库状态变迁的约束。动态约束包括动态属性级约束（修改属性定义时的约束、修改属性值时的约束）、动态元组约束和动态关系约束。

完整性控制机制应该具有定义功能和检查功能，定义功能提供定义完整性约束条件的机制，检查功能检查用户发出的操作请求是否违背了完整性约束条件。如果发现用户的操作请求违背了约束条件，则采取一定的动作来保证数据的完整性。

2. 实体完整性

实体完整性要求主键中的任一属性不能为空，所谓空值是“不知道”或“无意义”的值。之所以要保证实体完整性，主要是因为在关系中，每个元组的区分是依据主键值的不同，若主键值取空值，则不能标明该元组的存在。例如，对于学生关系S (Sno, Sname, Ssex)，其主键为Sno，在插入某个元组时，就必须要求Sno不能为空。更加严格的DBMS，则还要求Sno不能与已经存在的某个元组的Sno相同。

3. 参照完整性

若基本关系R中含有与另一基本关系S的主键PK相对应的属性组FK（FK称为R的外键），则参照完整性要求，对R中的每个元组在FK上的值必须是S中某个元组的PK值，或者为空值。参照完整性的合理性在于，R中的外键只能对S中的主键引用，不能是S中主键没有的值。例如，对于学生关系S (Sno, Sname, Ssex) 和选课关系C (Sno, Cno, Grade) 两个关系，C中的Sno是外键，它是S的主键，若C中出现了某个S中没有的Sno，即某个学生还没有注册，却已有了选课记录，这显然是不合理的。

在实际应用中，对于参照完整性，需要明确外键能否接受空值的问题，以及在被参照关系中删除元组的问题。针对不同的应用，可以有不同的删除方式：

（1）级联删除。将参照关系中所有外键值与被参照关系中要删除元组主键值相同的元组一起删除。如果参照关系同时又是另一个关系的被参照关系，则这种删除操作会继续级联下去。

（2）受限删除。这是一般DBMS默认的删除方式。仅当参照关系中没有任何元组的外键值与被参照关系中要删除元组的主键值相同时，系统才可以执行删除操作，否则拒绝执行删除操作。

（3）置空删除。删除被参照关系的元组，并将参照关系中相应元组的外键值置为空值。

同样，还需要考虑在参照关系中插入元组的问题，一般可以采用以下两种方式：

（1）受限插入。仅当被参照关系中存在相应的元组时，其主键值与参照关系插入元组的外键值相同时，系统才执行插入操作，否则拒绝此操作。

（2）递归插入。首先向被参照关系中插入相应的元组，其主键值等于参照关系插入元组的外键值，然后向参照关系插入元组。

4. 用户定义的完整性

实体完整性和参照完整性适用于任何关系型DBMS.除此之外，不同的数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。

如果在一条语句执行完后立即检查，则称立即执行约束；如果在整个事务执行结束后再进行检查，则称延迟执行约束。完整性规则的五元组表示为 (D,O,A,C,P)，其中D表示约束作用的数据对象，O表示触发完整性检查的数据库操作，A表示数据对象必须满足的断言或语义约束，C表示选择A作用的数据对象值的谓词，P表示违反完整性规则时触发的过程。

5. 触发器

触发器是在关系型DBMS中应用得比较多的一种完整性保护措施，其功能比完整性约束要强得多。一般而言，在完整性约束功能中，当系统检查出数据中有违反完整性约束条件时，则仅给出必要提示以通知用户，仅此而已。而触发器的功能则不仅起到提示作用，还会引起系统自动进行某些操作，以消除违反完整性约束条件所引起的负面影响。

所谓触发器，其抽象的含义即是一个事件的发生必然触发（或导致）另外一些事件的发生，其中前面的事件称为触发事件，后面的事件称为结果事件。触发事件一般即为完整性约束条件的否定，而结果事件即为一组操作用以消除触发事件所引起的不良影响。目前，数据库中事件一般表示为数据的插入、修改、删除等操作。触发器除了有完整性保护功能外，还有安全性保护功能。

在本题中，需要达到的效果是EMP中的工资产生变化，则需要判断变化值是否在P关系规定的范围之内，三种完整性约束都无法达到目的，应在EMP上建立触发器。

试题答案

(6) C

第 2 章：数据库系统

试题3 (2009年11月试题7~8)

试题3 (2009年11月试题7~8)

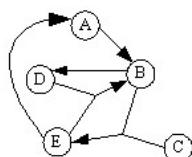
设关系模式R (U, F)，其中R上的属性集U={A, B, C, D, E},R上的函数依赖集F={A→B, DE→B, CB→E, E→A, B→D}. (7) 为关系R的候选关键字。分解 (8) 是无损连接，并保持函数依赖的。

(7) A.AB B.DE C.CE D.CB

(8) A.p={R1 (AC) , R2 (ED) , R3 (B) } B.p={R1 (AC) , R2 (E) , R3 (DB) }
C.p={R1 (AC) , R2 (ED) , R3 (AB) } D.p={R1, (ABC) , R2 (ED) , R3 (ACE) }

试题分析

候选关键字使用规范化理论中的图示法进行求解，对R关系模式画图如图2-1所示。



本图中C结点为0度结点，所以它必然被包含在候选关键字中，但仅有C结点并不能遍历全图，所以需要加入其他中间结点。若加入B结点，则 $BC \rightarrow E, E \rightarrow A, B \rightarrow D$ ，能遍历全图。与此同时，加入A、E都能起到同样的效果。所以关系R有三个候选键：BC、EC、AC。

接下来是判断模式分解过程中的无损连接与保持函数依赖的问题。这个问题相对来说比较复杂。如果逐个判断每个选项的无损连接与保持函数依赖，无疑工作量是很大的。所以我们可以先观察这些选项有什么特点，通过观察发现A与B选项都存在单字段的分解。在进行模式分解时，如果出现单字段，同时该字段未在其他分解的子关系模式中出现，并且函数依赖中有此字段的依赖关系，则说明此分解没有保持函数依赖。原因很简单，关于该字段的那个函数依赖，必然在分解中丢失了。所以A与B选项可以先排除。

然后判断C与D是否为无损连接。

对选项C构造初始的判定表如表2-2所示。

表2-2 模式分解C选项初始判定表

分解的关系模式	A	B	C	D	E
$R_1 (AC)$	a_1	b_{12}	a_3	b_{14}	b_{15}
$R_2 (ED)$	b_{21}	b_{22}	b_{23}	a_4	a_5
$R_3 (AB)$	a_1	a_2	b_{33}	b_{34}	b_{35}

由于 $A \rightarrow B$ ，属性A的第1行和第3行相同，可以将第1行 b_{12} 改为 a_2 ；又由于 $B \rightarrow D$ ，属性B的第1行和第3行相同，所以需要将属性D第1行 b_{14} 和第3行 b_{34} ，改为同一符号，即取行号值最小的 b_{14} 。修改后的判定表如表2-3所示。

表2-3 模式分解C选项修改判定表

分解的关系模式	A	B	C	D	E
$R_1 (AC)$	a_1	a_2	a_3	b_{14}	b_{15}
$R_2 (ED)$	b_{21}	b_{22}	b_{23}	a_4	a_5
$R_3 (AB)$	a_1	a_2	b_{33}	b_{14}	b_{35}

反复检查函数依赖集F，无法修改表2-3，所以选项C是有损连接的。

对选项D构造初始的判定表如表2-4所示。

表2-4 模式分解D选项初始判定表

分解的关系模式	A	B	C	D	E
$R_1 (ABC)$	a_1	a_2	a_3	b_{14}	b_{15}
$R_2 (ED)$	b_{21}	b_{22}	b_{23}	a_4	a_5
$R_3 (ACE)$	a_1	b_{22}	a_3	b_{34}	a_5

由于 $A \rightarrow B$ ，属性A的第1行和第3行相同，可以将第3行 b_{32} 改为 a_2 ； $E \rightarrow A$ ，属性E的第2行和第3行相同，可以将属性A第2行 b_{21} 改为 a_1 ； $AC \rightarrow E$ ，属性E的第2行和第3行相同，可以将属性E第1行 b_{15} 改为 a_5 ； $B \rightarrow D$ ，属性B的第1行和第3行相同，所以需要将属性D第1行 b_{14} 和第3行 b_{34} ，改为同一符号，即取行号值最小的 b_{14} 。 $E \rightarrow D$ ，属性E的第1~3行相同，可以将属性D第1行 b_{14} 和第3行 b_{34} 改为 a_4 。修改后的判定表如表2-5所示。

表2-5 模式分解D选项修改判定表

分解的关系模式	A	B	C	D	E
$R_1 (ABC)$	a_1	a_2	a_3	a_4	a_5
$R_2 (ED)$	a_1	b_{22}	b_{23}	a_4	a_5
$R_3 (ACE)$	a_1	a_2	a_3	a_4	a_5

由于表2-5第一行全为 a ，故分解无损。

所以本空应选D。

试题答案

(7) C (8) D

试题4 (2009年11月试题40)

试题4 (2009年11月试题40)

以下关于RDBMS数据分布的叙述中，错误的是（40）。

- (40) A.数据垂直分割是将不同表的数据存储到不同的服务器上
- B.数据水平分割是将不同行的数据存储到不同的服务器上
- C.数据复制是将数据的多个副本存储到不同的服务器上
- D.数据复制中由RDBMS维护数据的一致性

试题分析

数据分割和数据复制是数据分布的两种重要方式。数据分割是指将数据库中的表智能地分布在多个磁盘（或服务器）上，即可以将一个表的数据分布在不同的磁盘空间上，从而有效地提高并行处理的性能和高可用性。数据分割可以分为水平分割和垂直分割两种。

水平分割是将表中不同行的数据存储到不同的磁盘上。例如，当多个事务频繁访问数据表的不同行时，水平分割表，并消除新表中的冗余数据列。若个别事务要访问整个数据，则要用连接操作。水平分割会给应用增加复杂度，它通常在查询时需要多个表名，查询所有数据需要并操作。在许多数据库应用中，这种复杂性会超过它带来的优点，因为只要索引关键字不大，则在索引用于查询时，表中增加两到三倍数据量，查询时也就增加读一个索引层的磁盘次数。

垂直分割是将表中不同字段的数据存储到不同的磁盘上。例如，当多个事务频繁访问表的不同列时，可将该表垂直分成多个表。垂直分割可以使得数据行变小（因为列少了，一行数据就变小），一个数据页就能存放更多的数据，在查询时就会减少I/O次数。其缺点是需要管理冗余列，查询所有数据时需要连接操作。

数据分割增加了维护数据完整性的代价。

数据复制是为了提升数据访问效率而采用的一种增加数据冗余的方法，它将数据的多个副本存储到不同的服务器上，由RDBMS负责维护数据的一致性。

试题答案

- (40) A

试题5 (2010年11月试题5)

试题5 (2010年11月试题5)

在数据库设计的（5）阶段进行关系规范化。

- (5) A.需求分析 B.概念设计 C.逻辑设计 D.物理设计

试题分析

由试题1的分析可知，在数据库设计的逻辑设计阶段将E-R模型转换成关系模式，并使用规范化理论对关系模式进行优化处理，一般需要达到第三范式或BC范式。

试题答案

- (5) C

第2章：数据库系统

试题6 (2010年11月试题6~7)

试题6 (2010年11月试题6~7)

某数据库中有员工关系E（员工号，姓名，部门，职称，月薪）；产品关系P（产品号，产品名称，型号，尺寸，颜色）；仓库关系W（仓库号，仓库名称，地址，负责人）；库存关系I（仓库号，产品号，产品数量）。

a.若数据库设计中要求：

- ①仓库关系W中的“负责人”引用员工关系的员工号
- ②库存关系I中的“仓库号，产品号”唯一标识I中的每一个记录
- ③员工关系E中的职称“工程师”的月薪不能低于3500元

则①②③依次要满足的完整性约束是（6）。

- (6) A.实体完整性、参照完整性、用户定义完整性

- B.参照完整性、实体完整性、用户定义完整性
- C.用户定义完整性、实体完整性、参照完整性
- D.实体完整性、用户定义完整性、参照完整性

b.若需得到每种产品的名称和该产品的总库存量，则对应的查询语句为：

SELECT 产品名称, SUM(产品数量)

FROM P, I

WHERE P.产品号 = I.产品号 (7);

- (7) A.ORDER BY 产品名称 B.ORDER BY 产品数量

- C.GROUP BY 产品名称 D.GROUP BY 产品数量

试题分析

第(6)空考查数据库完整性约束。首先需要了解实体完整性、参照完整性和用户定义完整性的概念。

实体完整性：实体完整性要求主键中的任一属性不能为空，所谓空值是“不知道”或“无意义”的值。之所以要保证实体完整性，主要是因为在关系中，每个元组的区分是依据主键值的不同，若主键值取空值，则不能标明该元组的存在。例如，对于学生关系S(Sno,Sname,Ssex)，其主键为Sno，在插入某个元组时，就必须要求Sno不能为空。更加严格的DBMS，则还要求Sno不能与已经存在的某个元组的Sno相同。

参照完整性：若基本关系R中含有与另一基本关系S的主键PK相对应的属性组FK (FK称为R的外键)，则参照完整性要求，对R中的每个元组在FK上的值必须是S中某个元组的PK值，或者为空值。参照完整性的合理性在于，R中的外键只能对S中的主键引用，不能是S中主键没有的值。例如，对于学生关系S (Sno,Sname,Ssex) 和选课关系C (Sno,Cno,Grade) 两个关系，C中的Sno是外键，它是S的主键，若C中出现了某个S中没有的Sno,即某个学生还没有注册，却已有了选课记录，这显然是不合理的。

用户定义完整性：实体完整性和参照完整性适用于任何关系型DBMS.除此之外，不同的数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义完整性就是针对某一具体数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。

从以上概念说明可以看出"仓库关系W中的"负责人"引用员工关系的员工号"属于参照完整性，"库存关系I中的"仓库号，产品号"唯一标识I中的每一个记录"属于实体完整性，"员工关系E中的职称称为"工程师"的月薪不能低于3500元"属于用户自定义完整性。

第(7)空要求"得到每种产品的名称和该产品的总库存量",要达到该效果需要对数据表的数据进行分组统计。由于现在是要根据产品名称来统计库存量，所以分组的依据应是产品名称，所以应使用"GROUP BY 产品名称".

试题答案

(6) B (7) C

第2章：数据库系统

试题7 (2010年11月试题8)

试题7 (2010年11月试题8)

若对关系R (A,B,C,D) 和S (C,D,E) 进行关系代数运算，则表达式 $\pi_{3,4,7}(\sigma_{4<5}(R \times S))$ 与 (8) 等价。

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

试题分析

解题的关键在于了解关系代数式子中的数字下标所表达的意思。在 $\pi_{3,4,7}(\sigma_{4<5}(R \times S))$ 中，首先是R与S做笛卡尔积运算，运算结果为RS (R.A,R.B,R.C,R.D,S.C,S.D,S.E)，然后进行选择操作，操作条件为：4<5,此时RS中的第4列为：R.D,第5列为S.C,所以"4<5"相当于"R.D < S.C",最后进行投影操作，操作条件为：3,4,7,他们所对应的列为：R.C,R.D,S.E,列的写法可以进行精简，如S.E,可以直接写为E,但需要注意，由于R与S两个关系都有C和D字段，所以R.C和R.D不能简写为：C,D.

试题答案

(8) B

试题8 (2011年11月试题5~7)

试题8 (2011年11月试题5~7)

某企业工程项目管理数据库的部分关系模式如下所示，其中带实下画线的表示主键，虚下画线的表示外键。

供应商（供应商号，名称，地址，电话，账号）
项目（项目号，负责人，开工日期）
零件（零件号，名称，规格，单价）
供应（项目号，零件号，供应商号，供应量）
员工（员工号，姓名，性别，出生日期，职位，联系方式）

其中供应关系是（5）的联系。若一个工程项目可以有多个员工参加，每个员工可以参加多个项目，则项目和员工之间是（6）联系。对项目和员工关系进行设计时，（7）设计成一个独立的关系模式。

(5) A.2个实体之间的1:n B.2个实体之间的n:m

C.3个实体之间的1:n:m D.3个实体之间的k:n:m

(6) A.1:1 B.1:n C.n:m D.n:1

(7) A.多对多的联系在向关系模型转换时必须

B.多对多的联系在向关系模型转换时无须

C.只需要将一端的码并入多端，所以无须

D.不仅需要将一端的码并入多端，而且必须

试题分析

题目虽然有多个问题，但实际上只考查了一个知识点--实体之间的联系。

供应关系中，有属性：项目号，零件号，供应商号。这些属于分别来自供应商、项目、零件这三个关系，并且，一个供应商可以向多个项目供应零件，一个供应商可以供应多种零件，一个项目可以由多个供应商供应零件，一个项目可以使用多种零件，而一种零件可以由多个不同供应商来提供，一种零件可用于不同项目。这说明供应关系涉及3个实体，这3个实体之间的关系是k:n:m。

从题目的描述“若一个工程项目可以有多个员工参加，每个员工可以参加多个项目”可以得知，项目和员工的关系是n:m。

在实体转关系模式过程中，存在3种类型的联系，他们的处理方式如下：

1:1联系：在两个关系模式中的任意一个模式中，加入另一个模式的键和联系类型的属性；

1:n联系：在n端实体类型对应的关系模式中加入1端实体类型的键和联系类型的属性；

m:n联系：将联系类型也转换成关系模式，属性为两端实体类型的键加上联系类型的属性。

试题中是m:n联系，所以需要把联系单独转成一个关系模式。

试题答案

(5) D (6) C (7) A

试题9 (2011年11月试题8)

试题9 (2011年11月试题8)

给定学生S (学号 , 姓名 , 年龄 , 入学时间 , 联系方式) 和选课SC (学号 , 课程号 , 成绩) 关系 , 若要查询选修了1号课程的学生学号、姓名和成绩 , 则该查询与关系代数表达式 (8) 等价。

- (8) A. $\pi_{1,2,8}(\sigma_{l=6 \wedge 7=1}(S \bowtie SC))$ B. $\pi_{1,2,7}(\sigma_{6=1}(S \bowtie SC))$
C. $\pi_{1,2,7}(\sigma_{l=6}(S \bowtie SC))$ D. $\pi_{1,2,8}(\sigma_{7=1}(S \bowtie SC))$

试题分析

解答本题需要对关系代数中的自然连接有一定了解。自然连接操作会自动以两个关系模式中共有属性值相等作为连接条件 , 对于连接结果 , 将自动去除重复的属性。所以在本题中 , 连接条件为两个表的学号相等 , 当连接操作完成以后 , 形成的结果表 , 有属性"学号 , 姓名 , 年龄 , 入学时间 , 联系方式 , 课程号 , 成绩", 此时要选择1号课程的学生记录 , 应使用条件 $6=1$, 其含义是表中的第6个属性值为"1"。

试题答案

- (8) B

第 2 章 : 数据库系统

试题10 (2011年11月试题35~36)

试题10 (2011年11月试题35~36)

企业战略数据模型可分为两种类型 : (35) 描述日常事务处理中的数据及其关系 ;

(36) 描述企业管理决策者所需信息及其关系。

- (35) A. 元数据模型 B. 数据库模型 C. 数据仓库模型 D. 组织架构模型

- (36) A. 元数据模型 B. 数据库模型 C. 数据仓库模型 D. 组织架构模型

试题分析

企业中使用的数据模型分两大类 , 一类针对于处理日常事务的应用系统 , 即数据库。另一类针对于高层决策分析 , 即数据仓库。

试题答案

- (35) B (36) C

第 2 章 : 数据库系统

试题11 (2012年11月试题5~6)

试题11 (2012年11月试题5~6)

在数据库设计的需求分析阶段应当形成 (5), 这些文档可以作为 (6) 阶段的设计依据。

- (5) A. 程序文档、数据字典和数据流图
B. 需求说明文档、程序文档和数据流图
C. 需求说明文档、数据字典和数据流图
D. 需求说明文档、数据字典和程序文档
- (6) A. 逻辑结构设计 B. 概念结构设计
C. 物理结构设计 D. 数据库运行和维护

试题分析

对于题目所考查的内容, 我们可以用一个图来完美地诠释, 如图2-2所示。

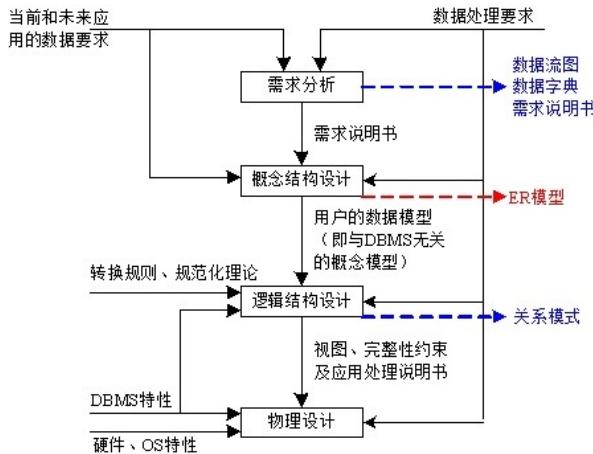


图2-2 数据库开发流程示意图

试题答案

- (5) C (6) B

第2章：数据库系统

试题12 (2012年11月试题7~8)

试题12 (2012年11月试题7~8)

某商场商品数据库的商品关系模式P (商品代码, 商品名称, 供应商, 联系方式, 库存量), 函数依赖集F={商品代码→商品名称, (商品代码, 供应商)→库存量, 供应商→联系方式}.商品关系模式P达到 (7); 该关系模式分解成 (8) 后, 具有无损连接的特性, 并能够保持函数依赖。

- (7) A.1NF B.2NF C.3NF D.BCNF
- (8) A.P1 (商品代码, 联系方式), P2 (商品名称, 供应商, 库存量)
B.P1 (商品名称, 联系方式), P2 (商品代码, 供应商, 库存量)
C.P1 (商品代码, 商品名称, 联系方式), P2 (供应商, 库存量)
D.P1 (商品代码, 商品名称), P2 (商品代码, 供应商, 库存量), P3 (供应商, 联系方式)

试题分析

要分析一个关系模式的范式，第一步应找出该关系模式的主键，接下来需要判断关系模式是否消除了非主属性对主键的部分依赖、传递依赖，这样便可得出结论。

首先可采用图示法求关系模式的主键（若不懂图示法，可参看《系统架构设计师考试考点突破、案例分析、试题实战一本通》）。将关系模式P₁使用图示法表达，如图2-3所示。

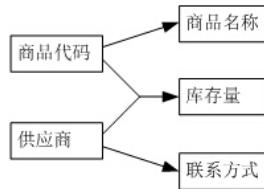


图2-3 P关系函数依赖示意图

从图中可以看出，只有商品代码与供应商的组合键才能遍历全图，所以只有他们的组合能充当主键。

由于（商品代码，供应商）是主键，而又有函数依赖：商品代码→商品名称，这便形成了部分依赖。所以在此关系模式中，部分依赖未消除，其范式只能达到1NF。

接下来的这个问题，对于经验丰富的考生，看完这几个选项，应该是能直接判断出应选D的。因为D是最佳分拆方案，能达到的范式级别很高。

如无经验，可考虑先分析是否为保持函数依赖的分解。

A选项分析：P₁（商品代码，联系方式）未保持任何原有的函数依赖。而P₂（商品名称，供应商，库存量）也未保持任何原有的函数依赖。

B选项分析：P₁（商品名称，联系方式）未保持任何原有的函数依赖。P₂（商品代码，供应商，库存量）只保持了函数依赖：（商品代码，供应商）→库存量。这样整体来说，就丢失了两个函数依赖关系。

C选项分析：P₁（商品代码，商品名称，联系方式）保持了函数依赖：商品代码→商品名称，P₂（供应商，库存量）未保持任何原有的函数依赖。这样整体来说，就丢失了两个函数依赖关系。

D选项分析：P₁（商品代码，商品名称）保持了函数依赖：商品代码→商品名称，P₂（商品代码，供应商，库存量）保持了函数依赖：（商品代码，供应商）→库存量，P₃（供应商，联系方式）保持了函数依赖：供应商→联系方式。这样将各个分拆关系的函数依赖整合起来，能构成原关系的函数依赖，所以这个分解是保持了函数依赖的分解。

试题答案

(7) A (8) D

试题13（2012年11月试题44~48）

试题13（2012年11月试题44~48）

在数据库系统中，“事务”是访问数据库并可能更新各种数据项的一个程序执行单元。为了保证数据完整性，要求数据库系统维护事务的原子性、一致性、隔离性和持久性。针对事务的这4种特性，考虑以下的架构设计场景。

假设在某一个时刻只有一个活动的事务，为了保证事务的原子性，对于要执行写操作的数据项，数据库系统在磁盘上维护数据库的一个副本，所有的写操作都在数据库副本上执行，而保持原始数据库不变，如果在任一时刻操作不得不中止，系统仅需要删除副本，原数据库没有受到任何影响。这种设计策略称为（44）。

事务的一致性要求在没有其他事务并发执行的情况下，事务的执行应该保证数据库的一致性。数据库系统通常采用（45）机制保证单个事务的一致性。

事务的隔离性保证操作并发执行后的系统状态与这些操作以某种次序顺序执行（即可串行化执行）后的状态是等价的。两阶段锁协议是实现隔离性的常见方案，该协议（46）。

持久性保证一旦事务完成，该事务对数据库所做的所有更新都是永久的，如果事务完成后系统出现故障，则需要通过恢复机制保证事务的持久性。假设在日志中记录所有对数据库的修改操作，将一个事务的所有写操作延迟到事务提交后才执行，则在日志中（47）。当系统发生故障时，如果某个事务已经开始，但没有提交，则该事务应该（48）。

（44）A.主动冗余 B.影子拷贝

C.热备份 D.多版本编程

（45）A.逻辑正确性检查 B.物理正确性检查

C.完整性约束检查 D.唯一性检查

（46）A.能够保证事务的可串行化执行，可能发生死锁

B.不能保证事务的可串行化执行，不会发生死锁

C.不能保证事务的可串行化执行，可能发生死锁

D.能够保证事务的可串行化执行，不会发生死锁

（47）A.无需记录“事务开始执行”这一事件

B.无需记录“事务已经提交”这一事件

C.无需记录数据项被事务修改后的新值

D.无需记录数据项被事务修改前的原始值

（48）A.重做 B.撤销

C.什么都不做 D.抛出异常后退出

试题分析

在数据库系统中，运行的基本工作单位是事务，事务是用户定义的一个数据库操作序列，这些操作序列要么全做，要么全不做，是一个不可分割的工作单位。事务具有以下特性：

（1）原子性（Atomicity）。事务是数据库的逻辑工作单位，事务的原子性保证事务包含的一组更新操作是原子不可分的，也就是说，这些操作是一个整体，不能部分的完成。

（2）一致性（Consistency）。一致性是指使数据库从一个一致性状态变到另一个一致性状态。例如，在转账的操作中，各账户金额必须平衡。一致性与原子性是密切相关的，一致性在逻辑上不是独立的，它由事务的隔离性来表示。

（3）隔离性（Isolation）。隔离性是指一个事务的执行不能被其他事务干扰，即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。它要求即使有多个事务并发执行，但看上去每个事务按串行调度执行一样。这一性质也称为可串行性，也就是说，系统允许的任何交错操作调度等价于一个串行调度。

（4）持久性（Durability）。持久性也称为永久性，是指事务一旦提交，改变就是永久性的，无论发生何种故障，都不应该对其有任何影响。

题干中第1个架构设计场景描述了数据库设计中为了实现原子性和持久性的最为简单的策略：“影子拷贝”。该策略假设在某一个时刻只有一个活动的事务，首先对数据库做副本（称为影子副本），并在磁盘上维护一个dp_pointer指针，指向数据库的当前副本。对于要执行写操作的数据项，数据库系统在磁盘上维护数据库的一个副本，所有的写操作都在数据库副本上执行，而保持原始数据库不变，如果在任一时刻操作不得不中止，系统仅需要删除新副本，原数据库副本没有受到任何影响。

题干中的第2个架构设计场景主要考查考生对事务一致性实现机制的理解。事务的一致性要求在没有其他事务并发执行的情况下，事务的执行应该保证数据库的一致性。数据库系统通常采用完整性约束检查机制保证单个事务的一致性。

题干中的第3个架构设计场景主要考查数据库的锁协议。两阶段锁协议是实现事务隔离性的常见方案，该协议通过定义锁的增长和收缩两个阶段约束事务的加锁和解锁过程，能够保证事务的串行化执行，但由于事务不能一次得到所有需要的锁，因此该协议可能会导致死锁。

题干中的第4个架构设计场景主要考查数据库的恢复机制，主要描述了基于日志的延迟修改技术（deferred-modification technique）的设计与恢复过程。该技术通过在日志中记录所有对数据库的修改操作，将一个事务的所有写操作延迟到事务提交后才执行，日志中需要记录“事务开始”和“事务提交”时间，还需要记录数据项被事务修改后的newValue，无需记录数据项被事务修改前的originalValue。当系统发生故障时，如果某个事务已经开始，但没有提交，则该事务对数据项的修改尚未体现在数据库中，因此无需做任何恢复动作。

试题答案

(44) B (45) C (46) A (47) D (48) C

第3章：计算机硬件基础及嵌入式系统设计

考点分析

第3章计算机硬件基础及嵌入式系统设计

3.1考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 嵌入式系统的特点。
- (2) 嵌入式系统的硬件组成与设计。
- (3) 嵌入式系统应用软件及开发平台。
- (4) 嵌入式系统网络。
- (5) 嵌入式系统数据库。
- (6) 嵌入式操作系统与实时操作系统。

计算机硬件基础及嵌入式系统设计部分从历年的考试情况来看，考点主要集中在以下方面：存储、串/并转换、寄存器、传输等方面。虽然计算机硬件基础部分在考纲中未做要求，但每次考试仍

会考查，所以本章也会有相关主题的试题。另外值得注意的是，嵌入式系统相关知识不仅在上午综合知识会考查，案例分析也会考查。具体考查知识点分布情况如表3-1所示。

表3-1历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题9~12	中断、存储器、串并、并/串转换
2010年11月试题9~12	寄存器、内存容量计算、磁盘文件读取、异步传输
2011年11月试题9~12	CISC与RISC、Cache特点、虚拟存储器、总线
2012年11月试题10~12	宿主机与目标机、硬件抽象层、实时操作系统

第3章：计算机硬件基础及嵌入式系统设计

试题1 (2009年11月试题9~10)

3.2 试题精解

试题1 (2009年11月试题9~10)

嵌入式系统中采用中断方式实现输入/输出的主要原因是 (9) .在中断时，CPU断点信息一般保存到 (10) 中。

- (9) A.速度最快 B.CPU不参与操作
C.实现起来比较容易 D.能对突发事件做出快速响应
- (10) A.通用寄存器 B.堆
C.栈 D.I/O接口

试题分析

在一般的操作系统中，输入/输出方式主要有以下几种：

(1) 程序控制方式：CPU直接利用I/O指令编程，实现数据的I/O.CPU发出I/O命令，命令中包含了外设的地址信息和所要执行的操作，相应的I/O系统执行该命令并设置状态寄存器；CPU不停地（定期地）查询I/O系统以确定该操作是否完成。由程序主动查询外设，完成主机与外设间的数据传送，方法简单，硬件开销小。

(2) 程序中断方式：CPU利用中断方式完成数据的I/O,当I/O系统与外设交换数据时，CPU无需等待也不必去查询I/O的状态，当I/O系统完成了数据传输后则以中断信号通知CPU.然后CPU保存正在执行程序的现场，转入I/O中断服务程序完成与I/O系统的数据交换。再然后返回原主程序继续执行。与程序控制方式相比，中断方式因为CPU无需等待而提高了效率。在系统中具有多个中断源的情况下，常用的处理方法有：多中断信号线法、中断软件查询法、雏菊链法、总线仲裁法和中断向量表法。

(3) DMA方式：使用DMA控制器（DMAC）来控制和管理数据传输。DMAC和CPU共享系统总线，并且具有独立访问存储器的能力。在进行DMA时，CPU放弃对系统总线的控制而由DMAC控制总线；由DMAC提供存储器地址及必须的读/写控制信号，实现外设与存储器之间进行数据交换。DMAC获取总线方式主要有三种，分别是暂停方式、周期窃取（挪用）方式和共享方式。

(4) 通道：通道是一种通过执行通道程序管理I/O操作的控制器，它使主机与I/O操作之间达到更高的并行程度。在具有通道处理机的系统中，当用户进程请求启动外设时，由操作系统根据I/O要求构造通道程序和通道状态字，将通道程序保存在主存中，并将通道程序的首地址放到通道地址字

中，然后执行“启动I/O”指令。按照所采取的传送方式，可将通道分为字节多路通道、选择通道和数组多路通道三种。

(5) 输入/输出处理机 (IOP)：也称为外围处理机 (PPU)，它是一个专用处理机，也可以是一个通用的处理机，具有丰富的指令系统和完善的中断系统。专用于大型、高效的计算机系统处理外围设备的I/O，并利用共享存储器或其他共享手段与主机交换信息。从而使大型、高效的计算机系统更加高效地工作。与通道相比，IOP具有比较丰富的指令系统，结构接近于一般的处理机，有自己的局部存储器。

嵌入式系统中采用中断方式实现输入/输出的主要原因是能对突发事件做出快速响应。在中断时，CPU断点信息一般保存到栈中。

试题答案

(9) D (10) C

第3章：计算机硬件基础及嵌入式系统设计

试题2 (2009年11月试题11)

试题2 (2009年11月试题11)

在嵌入式系统设计时，下面几种存储结构中对程序员是透明的是 (11)。

(11) A.高速缓存 B.磁盘存储器 C.内存 D.flash存储器

试题分析

四个选项中，高速缓存就是Cache，它处于内存与CPU之间，是为了提高访问内存时的速度而设置的，这个设备对于程序员的程序编写是完全透明的。

磁盘存储器与flash存储器都属于外设，在存储文件时，需要考虑到该设备的情况，因为需要将文件内容存于相应的设备之上。

内存是程序员写程序时需要考虑的，因为内存的分配与释放是经常要用到的操作。

试题答案

(11) A

第3章：计算机硬件基础及嵌入式系统设计

试题3 (2009年11月试题12)

试题3 (2009年11月试题12)

系统间进行异步串行通信时，数据的串/并和并/串转换一般是通过 (12) 实现的。

(12) A.I/O指令 B.专用的数据传送指令

- C.CPU中有移位功能的数据寄存器 D.接口中的移位寄存器

试题分析

一般来说，嵌入式系统通常采用接口中的移位寄存器来实现数据的串/并和并/串转换操作。

试题答案

(12) D

第 3 章：计算机硬件基础及嵌入式系统设计

试题4 (2010年11月试题9)

试题4 (2010年11月试题9)

计算机执行程序时，在一个指令周期的过程中，为了能够从内存中读指令操作码，首先是将 (9) 的内容送到地址总线上。

- (9) A.程序计数器PC B.指令寄存器IR
C.状态寄存器SR D.通用寄存器GR

试题分析

指令系统中的每一条指令都有一个操作码，它表示该指令应进行什么性质的操作。不同的指令用操作码这个字段的不同编码来表示，每一种编码代表一种指令。组成操作码字段的位数一般取决于计算机指令系统的规模。

通用计算机系统的基本指令有数据传送类指令、运算类指令、程序控制类指令、I/O指令、处理机控制和调试指令。

(1) 数据传送类指令。数据传送类指令是最基本的指令类型，主要用于实现寄存器与寄存器之间、寄存器与主存单元之间，以及两个主存单元之间的数据传送。数据传送指令的种类由三个主要因素决定，分别是数据存储设备的种类、数据传送单位和采用的寻址方式。数据传送类指令又可以细分为一般传送指令、堆栈操作指令和数据交换指令。一般传送指令具有数据复制的性质，即数据从源地址传送到目的地址，而源地址中的内容保持不变；堆栈操作指令分为进栈 (PUSH) 和出栈 (POP) 两种，在程序中它们往往是成对出现的；数据交换指令是双方向数据传送指令，即将源操作数与目的操作数相互交换位置。

(2) 运算类指令。运算类指令又分为算术运算指令、逻辑运算指令和移位指令，其中移位指令又可分为算术移位、逻辑移位和循环移位。运算类指令在整个指令系统中应该占有比较大的比重 (例如，超过 30%)。如果所占比重过小，就会影响整个计算机系统的性能。设计运算类指令，主要考虑操作种类、数据表示、数据长度、数据存储设备，以及它们的组合。在对这些因素进行组合时，必须考虑指令的执行时间、使用频率、硬件实现的复杂程度等多方面的情况。

(3) 程序控制指令。程序控制类指令用于控制程序的执行顺序，并使程序具有测试、分析与判断的能力，主要包括三类，分别是转移指令 (包括无条件转移和有条件转移) 、程序调用和返回指令、循环控制指令。其中，前两类指令在一般计算机中是必须具备的，最后一类指令用于对循环程序进行优化。

(4) I/O指令。I/O指令用来实现主机与外部设备之间的信息交换，包括I/O数据、主机向外设备控制命令或外设向主机报告工作状态等。I/O指令通常比较简单，采用单一的直接寻址方式，数据字长一般以字节为单位。在多用户或多任务环境下，I/O指令属于特权指令。当程序需要进行I/O操作时，用系统调用进入操作系统，由操作系统对设备统一进行管理。

(5) 处理机控制和调试指令。在一般的计算机系统中，处理机有两个状态，分别是管态和用户态，或称主态和从态。这两个状态需要互相切换，在这两个状态下所能使用的指令应该有所区别。在一般通用计算机系统中，按照指令的使用权限，可以把指令分为两大类，分别是一般指令和特权指令。只有系统管理程序能够使用，一般用户程序不能使用的指令称为特权指令，主要包括处理机状态的设置和管理、系统硬件和软件资源的管理、进程的管理等。只有在管态下才能够使用特权指令；在用户态下，只能使用一般指令。

程序计数器 (PC) 用于记录需要执行的下一条指令操作码的地址，所以在读指令操作码时，应将程序计数器的内容送到地址总线上。

试题答案

(9) A

第 3 章：计算机硬件基础及嵌入式系统设计

试题5 (2010年11月试题10)

试题5 (2010年11月试题10)

内存按字节编址，利用 $8K \times 4bit$ 的存储器芯片构成 $84000H$ 到 $8FFFFH$ 的内存，共需 (10) 片。

(10) A.6 B.8 C.12 D.24

试题分析

本题的题型在软考中较为常见，其难度在于计算时需要注意技巧，如果不注意技巧，将浪费大量时间于无谓的计算过程。 $8FFFFH - 84000H + 1 = (8FFFFH + 1) - 84000H = 90000H - 84000H = C000H$ ，化为十进制为 48K。由于内存是按字节编址，所以存储容量为： $48K \times 8bit, 48K \times 8bit / (8K \times 4bit) = 12$ 。

试题答案

(10) C

第 3 章：计算机硬件基础及嵌入式系统设计

试题6 (2010年11月试题11)

试题6 (2010年11月试题11)

某磁盘磁头从一个磁道移至另一个磁道需要10ms.文件在磁盘上非连续存放，逻辑上相邻数据块的平均移动距离为10个磁道，每块的旋转延迟时间及传输时间分别为100ms和2ms,则读取一个100块的文件需要 (11) ms的时间。

- (11) A.10200 B.11000 C.11200 D.20200

试题分析

在硬盘中，信息分布呈以下层次：记录面、圆柱面、磁道和扇区，如图3-1所示。

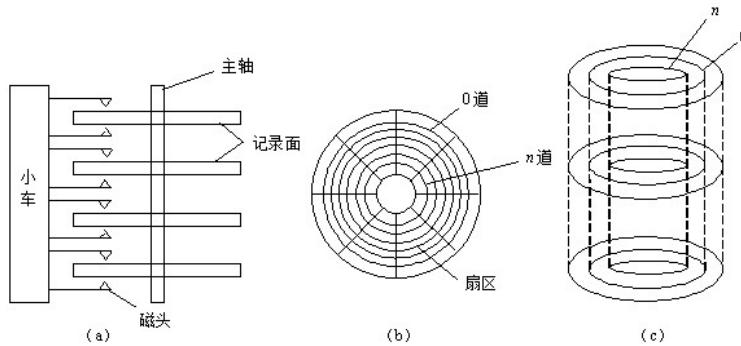


图3-1 硬盘信息分布示意图

一台硬盘驱动器中有多个磁盘片，每个盘片有两个记录面，每个记录面对应一个磁头，所以记录面号就是磁头号，如图3-1 (a) 所示。所有的磁头安装在一个公用的传动设备或支架上，磁头一致地沿盘面径向移动，单个磁头不能单独地移动。在记录面上，一条条磁道形成一组同心圆，最外圈的磁道为0号，往内则磁道号逐步增加，如图3-1 (b) 所示。在一个盘组中，各记录面上相同编号(位置)的各磁道构成一个柱面，如图3-1 (c) 所示。若每个磁盘片有m个磁道，则该硬盘共有m个柱面。

引入柱面的概念是为了提高硬盘的存储速度。当主机要存入一个较大的文件时，若一条磁道存不完，就需要存放在几条磁道上。这时，应首先将一个文件尽可能地存放在同一柱面中。如果仍存放不完，再存入相邻的柱面内。

通常将一条磁道划分为若干个段，每个段称为一个扇区或扇段，每个扇区存放一个定长信息块(例如，512个字节)，如图3-1 (b) 所示。一条磁道划分多少扇区，每个扇区可存放多少字节，一般由操作系统决定。磁道上的扇区编号从1开始，不像磁头或柱面编号从0开始。

主机向硬盘控制器送出有关寻址信息，硬盘地址一般表示为：驱动器号、柱面(磁道)号、记录面(磁头)号、扇区号。通常，主机通过一个硬盘控制器可以连接几台硬盘驱动器，所以需送出驱动器号。调用磁盘常以文件为单位，故寻址信息一般应当给出文件起始位置所在的柱面号与记录面号(这就确定了具体磁道)、起始扇区号，并给出扇区数(交换量)。

硬盘称的容量是指格式化容量，即用户实际可以使用的存储容量，而非格式化容量是指磁记录介质上全部的磁化单元数，格式化容量一般约为非格式化容量的60%~70%.格式化存储容量的计算公式是：

$$\text{存储容量} = n \times t \times s \times b$$

其中：n为保存数据的总记录面数，t为每面磁道数，s为每道的扇区数，b为每个扇区存储的字节数。

硬盘转速是硬盘主轴电机的旋转速度，它是决定硬盘内部传输速率的关键因素之一，在很大程度上直接影响到硬盘的速度。硬盘转速以每分钟多少转(RPM)来表示，RPM值越大，内部传输速率就越快，访问时间就越短，硬盘的整体性能也就越好。

记录密度是指硬盘存储器上单位长度或单位面积所存储的二进制信息量，通常以道密度和位密度表示。道密度是指沿半径方向上单位长度中的磁道数目，位密度是指沿磁道方向上单位长度中所记录的二进制信息的位数。

硬盘的存取时间主要包括三个部分。第一部分是指磁头从原先位置移动到目的磁道所需要的时间，称为寻道时间或查找时间；第二部分是指在到达目的磁道以后，等待被访问的记录块旋转到磁头下方的等待时间；第三部分是信息的读写操作时间。由于寻找不同磁道和等待不同记录块所花的时间不同，所以通常取它们的平均值。因为读/写操作时间比较快，相对于平均寻道时间 T_s 和平均等待时间 T_w 来说，可以忽略不计。所以，磁盘的平均存取时间 T_a 为：

$$T_a \approx T_s + T_w = \frac{t_{s\min} + t_{s\max}}{2} + \frac{t_{w\min} + t_{w\max}}{2}$$

硬盘缓存存在的目的是为了解决硬盘内部与接口数据之间速度不匹配的问题，它可以提高硬盘的读/写速度。

硬盘的数据传输速率分为内部数据传输速率和外部数据传输速率。内部数据传输速率是指磁头与硬盘缓存之间的数据传输速率，它的高低是评价一个硬盘整体性能的决定性因素。外部数据传输速率指的是系统总线与硬盘缓存之间的数据传输速率，外部数据传输速率与硬盘接口类型和缓存大小有关。

硬盘接口是硬盘与主机系统间的连接部件，不同的硬盘接口决定着硬盘与计算机之间的连接速度，在整个系统中，硬盘接口的优劣直接影响着程序运行快慢和系统性能好坏。

在本题中，首先需要寻道，即将磁头定位于目标磁道，然后通过磁盘的旋转，定位于要读取的信息之上，最后读取并传输数据。所以读取每个数据需要耗费的平均时间为： $10 \times 10 + 100 + 2 = 202\text{ms}$ ，100块的文件需要 20200ms 。

试题答案

(11) D

第 3 章：计算机硬件基础及嵌入式系统设计

试题7 (2010年11月试题12)

试题7 (2010年11月试题12)

计算机系统中，在 (12) 的情况下一般应采用异步传输方式。

- (12) A.CPU访问内存 B.CPU与I/O接口交换信息
C.CPU与PCI 总线交换信息 D.I/O接口与打印机交换信息

试题分析

根据外部设备与I/O模块交换数据的方式，系统接口可以分为串行接口和并行接口两种。串行接口一次只能传送1位信息，而并行接口一次就可传送多位信息。

串行通信又可分为异步通信方式和同步通信方式两种。同步通信是一种连续串行传送数据的通信方式，一次通信只传送一帧信息。在异步通信中，数据通常以字符或者字节为单位组成字符帧传送。字符帧由发送端逐帧发送，通过传输线被接收设备逐帧接收。发送端和接收端可以由各自的时钟来控制数据的发送和接收，这两个时钟源彼此独立，互不同步。

异步通信在发送字符时，所发送的字符之间的时间间隔可以是任意的。接收端必须时刻做好接收的准备，发送端可以在任意时刻开始发送字符，因此必须在每一个字符的开始和结束的地方加上标志，即加上开始位和停止位，以便使接收端能够正确地将每一个字符接收下来。异步通信的好处是通信设备简单、便宜，但传输效率较低；同步通信要求收发双方具有同频同相的同步时钟信号，只需在传送报文的最前面附加特定的同步字符，使收发双方建立同步，此后，便在同步时钟的控制下逐位发送和接收。

在本题中，CPU访问内存通常是同步方式，CPU与I/O接口交换信息通常是同步方式，CPU与PCI总线交换信息通常是同步方式，I/O接口与打印机交换信息则通常采用基于缓存池的异步方式。

试题答案

(12) D

第3章：计算机硬件基础及嵌入式系统设计

试题8 (2011年11月试题9)

试题8 (2011年11月试题9)

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 (9) .

- (9) A. 在CISC中，复杂指令都采用硬布线逻辑来执行
- B. 一般而言，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

试题分析

在计算机系统结构发展的过程中，指令系统的优化设计有两个截然相反的方向，一个是增强指令的功能，设置一些功能复杂的指令，把一些原来由软件实现的、常用的功能改用硬件的指令系统来实现，这种计算机系统称为复杂指令集计算机 (Complex Instruction Set Computer, CISC) ；另一个是尽量简化指令功能，只保留那些功能简单，能在在一个节拍内执行完成指令，较复杂的功能用一段子程序来实现，这种计算机系统称为精简指令集计算机 (Reduced Instruction Set Computer, RISC) 。

1. CISC指令系统的特点

CISC指令系统的主要特点如下：

- (1) 指令数量众多。指令系统拥有大量的指令，通常有100~250条左右。
- (2) 指令使用频率相差悬殊。最常使用的一些比较简单的指令，仅占指令总数的20%，但在程序中出现的频率却占80%。而大部分复杂指令却很少使用。
- (3) 支持很多种寻址方式。支持的寻址方式通常为5~20种。
- (4) 变长的指令。指令长度不是固定的，变长的指令增加指令译码电路的复杂性。
- (5) 指令可以对主存单元中的数据直接进行处理。典型的CISC通常都有指令能够直接对主存单元中的数据进行处理，其执行速度较慢。

(6) 以微程序控制为主。CISC的指令系统很复杂，难以用硬布线逻辑（组合逻辑）电路实现控制器，通常采用微程序控制。

2. 目标程序的优化

目标程序是由指令直接组成的，是要在处理机中直接执行的，因此，面向目标程序优化指令系统是提高计算机系统性能的最直接的方法。优化目标程序的目的主要有两个，一是缩短程序的长度，即减少程序的空间开销；另一个是缩短程序的执行时间，即减少程序的时间开销。

优化目标程序的主要途径是增强指令的功能，包括数据传送指令、运算类指令和程序控制指令。具体方法是，对大量的程序及其执行情况进行统计分析，找出那些使用频率高，执行时间长的指令和指令串。对于那些使用频率高的指令，用硬件加快其执行，就能缩短整个程序的执行时间。对于那些使用频率高的指令串，用一条新的指令来代替它，这样，不但能缩短整个程序的执行时间，而且能缩短整个程序的长度，从而减少程序的空间开销。

3. 对高级语言和编译程序的支持

高级语言和一般计算机的机器语言的语义差距比较大，通常用高级语言编写的程序经编译程序编译后生成的目标程序，与直接用机器语言或汇编语言编写的程序相比，时间和空间的开销都要大一个数量级。因此，改进指令系统，增加对高级语言和编译程序的支持，缩小高级语言与机器语言之间的差距，就能提高整个计算机系统的性能。

面向高级语言和编译程序增强指令系统的途径主要有两个。一是增强对高级语言和编译程序支持的指令的功能，增强体系结构的规整性，减少体系结构中各种例外情况；二是设计高级语言计算机，在这种计算机中，用高级语言编写的程序不需要经过编译，直接由机器的硬件来执行。例如，LISP计算机和PROLOG计算机等。

4. 操作系统的优化实现

任何一种计算机系统都必须有操作系统的支撑才能工作，而操作系统又必须用指令系统来实现。有关操作系统的知识，请阅读第3章。

5. CISC指令系统的缺陷

CISC指令系统主要存在如下三个方面的问题：

(1) 80-20规律。在CISC中，各种指令的使用频率相差很悬殊，大量的统计数字表明，大约有20%的指令使用频率比较大，占据了80%的处理机时间。换句话说，有80%的指令只在20%的处理机运行时间内才被用到。

(2) 超大规模（甚大规模、极大规模）集成电路技术的发展引起的问题。超大规模集成电路工艺要求规整性，而在CISC中，为了实现大量的复杂指令，控制逻辑极不规整，给超大规模集成电路工艺造成很大困难。在CISC中，大量使用微程序技术以实现复杂的指令系统。由于超大规模集成电路的集成度迅速提高，使得生产单芯片处理机成为可能。在单芯片处理机内，希望采用规整的硬布线控制逻辑，不希望用微程序。

(3) 软硬件的功能分配问题。在CISC中，为了支持目标程序的优化，支持高级语言和编译程序，增加了许多复杂的指令，用一条指令来代替一串指令。这些复杂指令简化了目标程序，缩小了高级语言与机器指令之间的语义差距。然而，为了实现这些复杂的指令，不仅增加了硬件的复杂程度，而且使指令的执行周期大大加长。例如，为了支持编译程序的对称性要求，一般的运算类指令都能直接访问主存，从而使指令的执行周期数增加，数据的重复利用率降低。

RISC不是简单地把指令系统进行简化，而是通过简化指令的途径使计算机的结构更加简单、合理，以减少指令的执行周期数，从而提高运算速度。

1. RISC指令系统的特点

RISC要求指令系统简化，操作在单周期内完成，指令格式力求一致，寻址方式尽可能减少，并提高编译的效率，最终达到加快机器处理速度的目的。RISC指令系统的主要特点如下：

- (1) 指令数量少。优先选取使用频率最高的一些简单指令和一些常用指令，避免使用复杂指令。只提供了LOAD（从存储器中读数）和STORE（把数据写入存储器）两条指令对存储器操作，其余所有的操作都在CPU的寄存器之间进行。
- (2) 指令的寻址方式少。通常只支持寄存器寻址方式、立即数寻址方式和相对寻址方式。
- (3) 指令长度固定，指令格式种类少。因为RISC指令数量少、格式少、相对简单，其指令长度固定，指令之间各字段的划分比较一致，译码相对容易。
- (4) 以硬布线逻辑控制为主。为了提高操作的执行速度，通常采用硬布线逻辑（组合逻辑）来构建控制器。
- (5) 单周期指令执行，采用流水线技术。因为简化了指令系统，很容易利用流水线技术，使得大部分指令都能在一个机器周期内完成。少数指令可能会需要多周期，例如，LOAD/STORE指令因为需要访问存储器，其执行时间就会长一些。
- (6) 优化的编译器：RISC的精简指令集使编译工作简单化。因为指令长度固定、格式少、寻址方式少，编译时不必在具有相似功能的许多指令中进行选择，也不必为寻址方式的选择而费心，同时易于实现优化，从而可以生成高效率执行的机器代码。
- (7) CPU中的通用寄存器数量多，一般在32个以上，有的可达上千个。

大多数RISC采用了Cache方案，使用Cache来提高取指令的速度。而且，有的RISC使用两个独立的Cache来改善性能。一个称为指令Cache，另一个称为数据Cache。这样，取指令和取数据可以同时进行，互不干扰。

2. RISC的核心思想

计算机执行一个程序所用的时间可表示为：

$$P = I \times CPI \times T$$

其中，P是执行这个程序所使用的总时间，I是这个程序所需执行的总的指令条数，CPI是每条指令执行的平均周期数，T是一个周期的时间长度。

(1) 由于RISC的指令都比较简单，CISC中的一条复杂指令所完成的功能在RISC中可能要用几条指令才能实现。对于同一个源程序，分别编译后生成的动态目标代码，显然RISC的要比CISC的多。但是，由于CISC中复杂指令使用的频率很低，程序中使用的绝大多数指令都是与RISC一样的简单指令，因此，实际上的统计结果表明，RISC的I长度只比CISC的长20%~40%。

(2) 由于CISC一般是用微程序实现的，一条指令往往要用好几个周期才能完成，一些复杂指令所要的周期数就更多。根据统计，大多数CISC的指令平均执行周期数CPI在4~10之间；而RISC的大多数指令都是单周期执行的，它们的CPI应该是1，但是，由于RISC中还有LOAD和STORE指令，也还有少数复杂指令，所以，CPI要略大于1。

(3) 由于RISC一般采用硬布线逻辑实现，指令要实现的功能都比较简单，所以，CISC的T通常是RISC的3倍左右。

综合以上三点，可以大致计算出，RISC的处理速度要比同规模的CISC提高3~5倍。其中的关键在于RISC的指令平均执行周期数CPI减小了，这正是RISC设计思想的精华。减小CPI是多个方面共同努力的结果。在硬件方面，采用硬布线控制逻辑，减少指令和寻址方式的种类，使用固定的指令格式，采用LOAD/STORE结构，指令执行过程中设置多级流水线等，软件方面十分强调优化编译技术的作用。

3. RISC的关键技术

RISC要达到很高的性能，必须有相应的技术支持。目前，在RISC处理机中采用的主要技术有如下几种：

(1) 延时转移技术。在RISC处理机中，指令一般采用流水线方式工作。流水线技术所面临的一个问题就是转移指令的出现，这时，有可能使流水线断流。其中一个解决办法是在转移指令之后插入一条有效的指令，而转移指令好象被延迟执行了，因此，把这种技术称为延迟转移技术。有关流水线方面的知识，将在6.5节中介绍。

(2) 指令取消技术。采用指令延时技术，遇到条件转移指令时，调整指令序列非常困难，在许多情况下找不到可以用来调整的指令。有些RISC处理机采用指令取消技术。在使用指令取消技术的处理机中，所有转移指令和数据变换指令都可以决定下面待执行的指令是否应该取消。如果指令被取消，其效果相当于执行了一条空操作指令，不影响程序的运行环境。为了提高程序的执行效率，应该尽量少取消指令，以保持指令流水线处于充满状态。

(3) 重叠寄存器窗口技术。在处理机中设置一个数量比较大的寄存器堆，并把它划分成很多个窗口。每个过程使用其中相邻的三个窗口和一个公共的窗口，而在这些窗口中，有一个窗口是与前一个过程共用，还有一个窗口是与下一个过程共用的。与前一个过程共用的窗口可以用来存放前一个过程传送给本过程的参数，同时也存放下一个过程的计算结果；与下一个过程共用窗口可以用来存放下一个过程的参数和存放下一个过程传送给本过程的计算结果。

(4) 指令流调整技术。为了使RISC处理机中的指令流水线高效率地工作，尽量不断流。编译器必须分析程序的数据流和控制流，当发现指令流有断流可能时，要调整指令序列。对有些可以通过变量重新命名来消除的数据相关，要尽量消除。这样，可以提高流水线的执行效率，缩短程序的执行时间。

(5) 逻辑实现以硬件为主，固件为辅。RISC主要采用硬布线逻辑来实现指令系统。对于那些必须的复杂指令，也可用固件实现。

在实际应用中，商品化的RISC机器并不是纯粹的RISC。为了满足应用的需要，实用的RISC除了保持RISC的基本特色之外，还必须辅以一些必不可少的复杂指令，例如，浮点运算、十进制运算指令等。所以，这种机器实际上是在RISC基础上实现了RISC与CISC的完美结合。

本题考查CISC与RISC的区别，这是一种常见的考题，两者的区别总结如表3-2所示。

表3-2 CISC与RISC的区别

指令系统类型	指令	寻址方式	实现方式	其他
CISC（复杂）	数量多，使用频率差别大，可变长格式	支持多种	微程序控制技术	
RISC（精简）	数量少，使用频率接近，定长格式，大部分为单周期指令，操作寄存器，只有Load/Store操作内存	支持方式少	增加了通用寄存器；硬布线逻辑控制为主；采用流水线	优化编译，有效支持高级语言

试题答案

(9) A

以下关于Cache的叙述中，正确的是（10）。

- (10) 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的成本下提高命中率
- 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

试题分析

Cache的功能是提高CPU数据输入输出的速率，突破所谓的“冯·诺依曼瓶颈”，即CPU与存储系统间数据传送带宽限制。高速存储器能以极高的速率进行数据的访问，但因其价格高昂，如果计算机的内存完全由这种高速存储器组成，则会大大增加计算机的成本。通常在CPU和内存之间设置小容量的Cache。Cache容量小但速度快，内存速度较低但容量大，通过优化调度算法，系统的性能会大大改善，仿佛其存储系统容量与内存相当而访问速度近似Cache。

Cache通常采用相联存储器（Content Addressable Memory, CAM）。CAM是一种基于数据内容进行访问的存储设备。当对其写入数据时，CAM能够自动选择一个未用的空单元进行存储；当要读出数据时，不是给出其存储单元的地址，而是直接给出该数据或者该数据的一部分内容，CAM对所有存储单元中的数据同时进行比较，并标记符合条件的所有数据以供读取。由于比较是同时、并行进行的，所以，这种基于数据内容进行读写的机制，其速度比基于地址进行读写的方式要快很多。

1. Cache基本原理

使用Cache改善系统性能的依据是程序的局部性原理。程序访问的局部性有两个方面的含义，分别是时间局部性和空间局部性。时间局部性是指如果一个存储单元被访问，则可能该单元会很快被再次访问。这是因为程序存在着循环。空间局部性是指如果一个存储单元被访问，则该单元邻近的单元也可能很快被访问。这是因为程序中大部分指令是顺序存储、顺序执行的，数据一般也是以向量、数组、树、表等形式簇聚地存储在一起的。

根据程序的局部性原理，最近的、未来要用的指令和数据大多局限于正在用的指令和数据，或是存放在与这些指令和数据位置上邻近的单元中。这样，就可以把目前常用或将要用到的信息预先放在Cache中。当CPU需要读取数据时，首先在Cache中查找是否有所需内容，如果有，则直接从Cache中读取；若没有，再从内存中读取该数据，然后同时送往CPU和Cache。如果CPU需要访问的内容大多都能在Cache中找到（称为访问命中），则可以大大提高系统性能。

如果以 h 代表对Cache的访问命中率（“ $1-h$ ”称为失效率，或者称为未命中率）， t_1 表示Cache的周期时间， t_2 表示内存的周期时间，以读操作为例，使用“Cache+主存储器”的系统的平均周期为 t_3 。则：

$$t_3 = t_1 \times h + t_2 \times (1-h)$$

系统的平均存储周期与命中率有很密切的关系，命中率的提高即使很小也能导致性能上的较大改善。

例如，设某计算机主存的读/写时间为100ns，有一个指令和数据合一的Cache，已知该Cache的读/写时间为10ns，取指令的命中率为98%，取数的命中率为95%。在执行某类程序时，约有1/5指令需要存/取一个操作数。假设指令流水线在任何时候都不阻塞，则设置Cache后，每条指令的平均访存时间约为：

$$(2\% \times 100\text{ns} + 98\% \times 10\text{ns}) + 1/5 \times (5\% \times 100\text{ns} + 95\% \times 10\text{ns}) = 14.7\text{ns}$$

2. 映射机制

当CPU发出访存请求后，存储器地址先被送到Cache控制器以确定所需数据是否已在Cache中，若命中则直接对Cache进行访问。这个过程称为Cache的地址映射（映像）。在Cache的地址映射中，主存和Cache将均分成容量相同的块（页）。常见的映射方法有直接映射、全相联映射和组相联映射。

（1）直接映射。直接映射方式以随机存取存储器作为Cache存储器，硬件电路较简单。直接映射是一种多对一的映射关系，但一个主存块只能被复制到Cache的一个特定位置上去。例如，某Cache容量为16KB（即可用14位表示），每块的大小为16B（即可用4位表示），则说明其可分为1024块（可用10位表示）。主存地址的最低4位为Cache的块内地址，然后接下来的中间10位为Cache块号。如果主存地址为1234E8F8H（一共32位），那么，最后4位就是1000（对应十六进制数的最后一位“8”），而中间10位，则应从E8F（1110 1000 1111）中获取，得到10 1000 1111。因此，主存地址为1234E8F8H的单元装入的Cache地址为10 1000 1111 1000。

直接映射的关系可以用下列公式来表示：

$$K = I \bmod C$$

其中，K为Cache的块号，I为主存的页号，C为Cache的块数。

直接映射方式的优点是比较容易实现，缺点是不够灵活，有可能使Cache的存储空间得不到充分利用。例如，假设Cache有8块，则主存的第1页与第17页同时复制到Cache的第1块，即使Cache其他块空闲，也有一个主存页不能写入Cache。

（2）全相联映射。全相联映射使用相联存储器组成的Cache存储器。在全相联映射方式中，主存的每一页可以映射到Cache的任一块。如果淘汰Cache中某一块的内容，则可调入任一主存页的内容，因而较直接映射方式灵活。

在全相联映射方式中，主存地址不能直接提取Cache块号，而是需要将主存页标记与Cache各块的标记逐个比较，直到找到标记符合的块（访问Cache命中），或者全部比较完后仍无符合的标记（访问Cache失败）。因此，这种映射方式速度很慢，失掉了高速缓存的作用，这是全相联映射方式的最大缺点。如果让主存页标记与各Cache标记同时比较，则成本又太高。全相联映像方式因比较器电路难于设计和实现，只适用于小容量的Cache。

（3）组相联映射。组相联映射是直接映射和全相联映射的折中方案。它将Cache中的块再分成组，通过直接映射方式决定组号，通过全相联映射的方式决定Cache中的块号。在组相联映射方式中，主存中一个组内的页数与Cache的分组数相同。

例如，容量为64块的Cache采用组相联方式映像，每块大小为128个字，每4块为一组，即Cache分为 $64/4=16$ 组。若主存容量为4096页，且以字编址。首先，根据主存与Cache块的容量需一致，即每个内存页的大小也是128个字，因此共有 128×4096 个字（219个字），即主存地址需要19位。因为Cache分为16组，所以主存需要分为 $4096/16=256$ 组（每组16页），即28组，因此主存组号需8位。

按照上述划分方法，主存每一组的第1页映射到Cache的第1组，主存每一组的第2页映射到Cache的第2组，以此类推。因为主存中一个组内的页数与Cache的分组数相同，所以主存每一组的最后一页映射到Cache的最后一组。

要注意的是，有关组相联映射的划分方法不止一种。例如，另外一种方式是主存不分组，而是根据下列公式直接进行映射：

$$J = I \bmod Q$$

其中，J为Cache的组号，I为主存的页号，Q为Cache的组数。

在组相联映射中，由于Cache中每组有若干可供选择的块，因而它在映像定位方面较直接映像方式灵活；每组块数有限，因此付出的代价不是很大，可以根据设计目标选择组内块数。

3. 替换算法

当Cache产生了一次访问未命中之后，相应的数据应同时读入CPU和Cache.但是当Cache已存满数据后，新数据必须替换（淘汰）Cache中的某些旧数据。最常用的替换算法有以下三种：

（1）随机算法。这是最简单的替换算法。随机法完全不管Cache块过去、现在及将来的使用情况，简单地根据一个随机数，选择一块替换掉。

（2）先进先出（First In and First Out,FIFO）算法。按调入Cache的先后决定淘汰的顺序，即在需要更新时，将最先进入Cache的块作为被替换的块。这种方法要求为每块做一记录，记下它们进入Cache的先后次序。这种方法容易实现，而且系统开销小。其缺点是可能会把一些需要经常使用的程序块（如循环程序）替换掉。

（3）近期最少使用（Least Recently Used,LRU）算法。LRU算法是把CPU近期最少使用的块作为被替换的块。这种替换方法需要随时记录Cache中各块的使用情况，以便确定哪个块是近期最少使用的块。LRU算法相对合理，但实现起来比较复杂，系统开销较大。通常需要对每一块设置一个称为“年龄计数器”的硬件或软件计数器，用以记录其被使用的情况。

4. 写操作

因为需要保证缓存在Cache中的数据与内存中的内容一致，相对读操作而言，Cache的写操作比较复杂，常用的有以下几种方法。

（1）写直达（write through）。当要写Cache时，数据同时写回内存，有时也称为写通。当某一块需要替换时，也不必把这一块写回到主存中去，新调入的块可以立即把这一块覆盖掉。这种方法实现简单，而且能随时保持主存数据的正确性，但可能增加多次不必要的主存写入，会降低存取速度。

（2）写回（write back）。CPU修改Cache的某一块后，相应的数据并不立即写入内存单元，而是当该块从Cache中被淘汰时，才把数据写回到内存中。在采用这种更新策略的Cache块表中，一般有一个标志位，当一块中的任何一个单元被修改时，标志位被置“1”.在需要替换掉这一块时，如果标志位为“1”,则必须先把这一块写回到主存中去之后，才能再调入新的块；如果标志位为“0”,则这一块不必写回主存，只要用新调入的块覆盖掉这一块即可。这种方法的优点是操作速度快，缺点是因为主存中的字块未随时修改而有可能出错。

（3）标记法。对Cache中的每一个数据设置一个有效位。当数据进入Cache后，有效位置“1”；而当CPU要对该数据进行修改时，数据只需写入内存并同时将该有效位置“0”.当要从Cache中读取数据时需要测试其有效位，若为“1”则直接从Cache中取数，否则，从内存中取数。

试题答案

（10）B

试题10 (2011年11月试题11)

虚拟存储器发生页面失效时，需要进行外部地址变换，即实现（11）的变换。

- (11) A.虚地址到主存地址 B.主存地址到Cache地址
- C.主存地址到辅存物理地址 D.虚地址到辅存物理地址

试题分析

虚拟存储 (virtual storage) 是指把多个存储介质模块 (例如，硬盘、RAID等) 通过一定的手段集中管理，形成统一管理的存储池 (storage pool)，为用户提供大容量、高数据传输性能的存储系统。存储虚拟化是将实际的物理存储实体与存储的逻辑表示实现分离，使用虚拟存储技术，应用服务器只与分配给它们的逻辑卷 (虚卷) 交互，而不用关心其数据是在哪个物理存储实体上。

1. 虚拟存储的分类

按照拓扑结构的不同，虚拟存储可以分为两种方式，分别是对称式和非对称式。对称式虚拟存储技术是指虚拟存储控制设备与存储软件系统、交换设备集成成为一个整体，内嵌在网络数据传输路径中；非对称式虚拟存储技术是指虚拟存储控制设备独立于数据传输路径之外。

按照实现原理的不同，虚拟存储也可以分为两种方式，分别是数据块虚拟与虚拟文件系统。数据块虚拟存储方式着重解决数据传输过程中的冲突和延时问题，利用虚拟的多端口并行技术，为多个用户提供极高的带宽，最大限度上减少延时与冲突的发生；虚拟文件系统存储方式着重解决大规模网络中文件共享的安全机制问题。通过对不同的站点指定不同的访问权限，保证网络文件的安全。在实际应用中，数据块虚拟存储方式以对称式拓扑结构为表现形式，虚拟文件系统存储方式以非对称式拓扑结构为表现形式。

2. 虚拟存储的实现方式

虚拟存储要解决的关键问题是逻辑卷与物理存储实体之间的映射关系，这种映射关系可以在计算机层解决，也可以在存储设备层解决，还可以在存储网络层解决。

(1) 主机级的虚拟化。由安装在应用服务器上的卷管理软件完成存储的虚拟化。基于主机端的虚拟存储几乎都是通过纯软件的方式实现的，这种实现机制不需要引入新设备，也不影响现有存储系统的基本架构，所以实现成本比较低。但是，这种机制难以克服的困难是平台依赖性太强，开发商要为每一种操作系统平台甚至每一个版本，开发一套软件产品。同时，由于存储管理由主机解决，增加了主机的负担。

(2) 存储设备级的虚拟化。由存储设备的控制器实现存储的虚拟化。这种虚拟存储一般是存储厂商实施的，很可能使用厂商独家的存储产品。这种实现机制主要通过大规模的RAID子系统和多个I/O通道连接到服务器上，智能控制器提供访问控制、缓存和其他 (例如，数据复制等) 管理功能。这种方式的优点在于效率高、性能好，管理员对设备有完全的控制权，而且通过与服务器系统分开，可以将存储的管理与多种服务器操作系统隔离，并且可以很容易地调整硬件参数。但是，在现实中，厂商往往都只提供对自身产品的支持，不能解决异构存储环境中的虚拟化问题。

(3) 网络级的虚拟化。由加入SAN的专用装置实现存储虚拟化。这种机制可以管理不同厂商的存储设备，实现SAN中所有设备的统一管理，具有较好的开放性。

不管采用何种虚拟存储技术，其目的都是为了提供一个高性能、安全、稳定、可靠、可扩展的存储网络平台，满足系统的要求。根据综合的性能价格比来说，一般情况下，在主机级和存储设备级的虚拟化技术能够保证系统的数据处理能力要求时，则可优先考虑，因为这两种虚拟存储技术构建方便、管理简单、维护容易、产品相对成熟、性能价格比高。在需要将多个异构的存储设备集成

为一个或多个存储池时，则需要使用网络级的虚拟化技术，以达到充分利用存储容量、集中管理存储、降低存储成本的目的。

3. 虚拟存储的特点

虚拟存储技术是为了解决复杂、烦琐的存储管理而产生的，但是，随着信息技术的发展，虚拟存储在很多方面表现出优秀的性能：

(1) 虚拟存储提供了一个大容量存储系统集中管理的手段，由网络中的一个环节进行统一管理，避免了由于存储设备扩充所带来的管理方面的麻烦。例如，增加新的存储设备时，只需要管理员对存储系统进行较为简单的配置更改，客户端无需任何操作。

(2) 虚拟存储可以大大提高存储系统整体访问带宽。存储系统是由多个存储模块组成的，而虚拟存储系统可以很好地进行负载平衡，把每一次数据访问所需的带宽合理地分配到各个存储模块上，这样，系统的整体访问带宽就增大了。例如，一个存储系统中有4个存储模块，每一个存储模块的访问带宽为50MBps，则这个存储系统的总访问带宽就可以接近各存储模块带宽之和，即200MBps。

(3) 虚拟存储技术为存储资源管理提供了更好的灵活性和兼容性，可以将不同类型的存储设备集中管理使用，保护用户的已有投资。

(4) 虚拟存储技术可以通过管理软件，为网络系统提供一些其他的有用功能，例如，无需服务器的远程镜像和数据快照等。

(5) 虚拟存储技术将计算机的应用系统与存储设备分离，使各种不同的存储设备看上去具有标准的存储特性，应用系统不需要关心数据存储的具体设备，从而减轻了应用系统的负担。

由于虚拟存储具有上述特点，正逐步成为共享存储管理的主流技术，在数据镜像、数据复制、实时数据恢复、应用集成等方面有着广泛的应用。

虚拟存储区的容量与物理主存大小无关，而受限于计算机的地址结构和可用磁盘容量。其页面的置换依据相应的页面置换算法进行，当页面失效时，需要进行数据交换，此时涉及逻辑地址（虚地址）到辅存物理地址的变换。

试题答案

(11) D

第3章：计算机硬件基础及嵌入式系统设计

试题11 (2011年11月试题12)

试题11 (2011年11月试题12)

挂接在总线上的多个部件，(12)。

- (12) A. 只能分时向总线发送数据，并只能分时从总线接收数据
- B. 只能分时向总线发送数据，但可同时从总线接收数据
- C. 可同时向总线发送数据，并同时从总线接收数据
- D. 可同时向总线发送数据，但只能分时从总线接收数据

试题分析

总线是一组能为多个部件分时共享的公共信息传送线路。共享是指总线上可以挂接多个部件，各个部件之间相互交换的信息都可以通过这组公共线路传送；分时是指同一时刻只允许有一个部件向总线发送信息，如果出现两个或两个以上部件同时向总线发送信息，势必导致信号冲突。当然，在同一时刻，允许多个部件同时从总线上接收相同的信息。

1.总线的分类

按总线相对于CPU或其他芯片的位置可分为内部总线和外部总线两种。在CPU内部，寄存器之间和算术逻辑部件ALU与控制部件之间传输数据所用的总线称为内部总线；外部总线是指CPU与内存RAM、ROM和输入/输出设备接口之间进行通信的通路。由于CPU通过总线实现程序取指令、内存/外设的数据交换，在CPU与外设一定的情况下，总线速度是制约计算机整体性能的最大因素。

按总线功能来划分，又可分为地址总线、数据总线、控制总线三类，人们通常所说的总线都包括这三个组成部分，地址总线用来传送地址信息，数据总线用来传送数据信息，控制总线用来传送各种控制信号。例如，ISA (Industrial Standard Architecture,工业标准结构) 总线共有98条线，其中数据线有16条、地址线24条，其余为控制信号线、接地线和电源线。

按总线在微机系统中的位置，可分为机内总线和机外总线两种。上面所说的总线都是机内总线，而机外总线是指与外部设备接口相连的，实际上是一种外设的接口标准。例如，目前计算机上流行的接口标准IDE (Integrated Drive Electronics,电子集成驱动器) 、SCSI、USB (Universal Serial Bus,通用串行总线) 和IEEE (Institute of Electrical and Electronics Engineers,美国电气电子工程师协会) 1394等，前两种主要是与硬盘、光驱等设备接口相连，后面两种新型外部总线可以用来连接多种外部设备。

计算机的总线按其功用来划分，主要有局部总线、系统总线、通信总线三种类型。其中局部总线是在传统的ISA总线和CPU总线之间增加的一级总线或管理层，它的出现是由于计算机软硬件功能的不断发展，系统原有的ISA或EISA (Extended ISA,扩展的ISA) 等已远远不能适应系统高传输能力的要求，而成为整个系统的主要瓶颈；系统总线是计算机系统内部各部件（插板）之间进行连接和传输信息的一组信号线，例如，ISA、EISA、MCA (Micro Channel Architecture,微通道结构) 、VESA (Video Electronic Standard Association,视频电子标准协会) 、PCI (Peripheral Component Interconnect,外设组件互连) 、AGP (Accelerate Graphical Port,加速图形接口) 等；通信总线是计算机系统之间或计算机系统与其他系统（例如，远程通信设备、测试设备等）之间进行通信的一组信号线。

按照总线中数据线的多少，可分为并行总线和串行总线。并行总线是含有多条双向数据线的总线，它可以实现一个数据的多位同时传输，总线中数据线的数量决定了可传输一个数据的最大位数（一般为8的倍数）。由于可以同时传输数据的各位，所以并行总线具有数据传输速率高的优点。但由于各条数据线的传输特性不可能完全一致，当数据线较长时，数据各位到达接收端时的延迟可能不一致，会造成传输错误，所以并行总线不宜过长，适合近距离连接。大多数的系统总线属于并行总线；串行总线是只含有一条双向数据线或两条单向数据线的总线，可以实现一个数据的各位按照一定的速度和顺序依次传输。由于按位串行传输数据对数据线传输特性的要求不高，在长距离连线情况下仍可以有效地传送数据，所以串行总线的优势在于远距离通信。但由于数据是按位顺序传送的，所以在相同的时钟控制下，数据传输速率低于并行总线。大多数的通信总线属于串行总线。

2.总线标准

总线标准是指计算机部件各生产厂家都需要遵守的总线要求，从而使不同厂家生产的部件能够互换。总线标准主要规定总线的机械结构规范、功能结构规范和电气规范。总线标准可以分为正式

标准和工业标准，其中正式标准是由IEEE等国际组织正式确定和承认的标准；工业标准也称为事实标准，是首先由某一厂家提出，然后得到其他厂家广泛使用的标准。

3. 总线的性能指标

通常，总线规范中会详细描述总线各方面的特性，包括物理特性、功能特性、电气特性和时间特性。物理特性又称机械特性，它规定了总线的线数，以及总线的插头、插座的形状、尺寸和信号线的排列方式等要素；功能特性描述总线中每一根线的功能；电气特性定义了每根线上信号的传递方向及有效电平范围；时间特性规定了每根线在什么时间有效以及不同信号之间相互配合的时间关系。

总线的主要性能指标主要有以下几个：

(1) 总线宽度。总线宽度指的是总线的线数，它决定了总线所占的物理空间和成本。对总线宽度最直接的影响是地址线和数据线的数量。主存空间和I/O空间的扩充使地址线数量的增加，并行传输要求有足够的数据线。例如，32位的PCI总线允许寻址的主存空间的大小为 $2^{32}=4G$ 个单元。

(2) 总线带宽。总线带宽定义为总线的最大数据传输速率，即每秒传输的字节数。在同步通信中，总线的带宽与总线时钟密不可分，总线时钟频率的高低决定了总线带宽的大小：

$$\text{总线带宽} = \text{总线宽度} \times \text{总线频率}$$

总线的实际带宽还会受到总线长度（总线延迟）、总线负载、总线收发器性能等多方面因素的影响。例如，假设某系统总线在一个总线周期中并行传输4字节信息，一个总线周期占用2个时钟周期，总线时钟频率为10MHz。此时，时钟周期 $T=1/10M=0.1\mu s$ ，总线周期 $=2T=0.2\mu s$ ，则总线带宽为 $4/0.2=20MB/s$ 。

(3) 总线负载。总线负载是指连接在总线上的最大设备数量。大多数总线的负载能力是有限的。

(4) 总线分时复用。总线分时复用是指在不同时段利用总线上同一个信号线传送不同信号，例如，地址总线和数据总线共用一组信号线。采用这种方式的目的是减少总线数量，提高总线的利用率。

(5) 总线猝发传输。猝发式数据传输是一种总线传输方式，即在一个总线周期中可以传输存储地址连续的多个数据。

除了以上提到的性能指标外，总线是否具有即插即用功能，是否支持总线设备的热插拔，是否支持多主控设备，是否具有错误检测能力，是否依赖于特定CPU等，也是评价总线性能的指标。

试题答案

(12) B

试题12 (2012年11月试题10)

试题12 (2012年11月试题10)

以下关于嵌入式系统开发的叙述，正确的是 (10)。

(10) A.宿主机与目标机之间只需要建立逻辑连接

- B.宿主机与目标机之间只能采用串口通信方式
- C.在宿主机上必须采用交叉编译器来生成目标机的可执行代码
- D.调试器与被调试程序必须安装在同一台机器上

试题分析

在嵌入式系统开发过程中，有三种不同的开发模式，这三种开发模式就会涉及本题所述的宿主机与目标机（调试程序运行的机器称为宿主机，被调试程序运行的机器称为目标机）。下面将详细说明这三种开发模式。

本机开发：本机开发也就是在目标机（在嵌入式系统中通常把嵌入式系统或设备简称为目标机）中直接进行操作系统移植及应用程序的开发。在这种方式下进行开发，首先就得在目标机中安装操作系统，并且具有良好的人机开发界面。

交叉开发：意思就是在一台宿主机（在嵌入式系统中通常把通用PC称为宿主机）上进行操作系统的裁剪，以及编写应用程序，在宿主机上应用交叉编译环境编译内核及应用程序，然后把目标代码下载到目标机上运行。这就需要在宿主机上安装、配置交叉编译环境（交叉开发工具链），使其能够编译成在目标机上运行的目标代码。

模拟开发：建立在交叉开发环境基础之上。除了宿主机和目标机以外，还得提供一个在宿主机上模拟目标机的环境，使得开发好的内核和程序直接在这个环境下运行以验证其正确性，这就不需要将每次的修改都下载到目标机中，待程序正确后再下载到目标机上运行。这样就可以达到在没有目标机的情况下调试软件的目的。比较著名的模拟开发环境有SkyEye，它能够模拟如ARM等处理器的开发环境。模拟硬件环境是一项比较复杂的工程，所以多数商业嵌入式系统的开发采用的是交叉开发模式。

从以上解释可以看出，宿主机与目标机可能在一台机器上，也可能在不同机器上。宿主机与目标机之间既要有逻辑连接，还要有物理连接。至于通信方式，串口只是其中一种标准，还可采用其他方式。

试题答案

(10) C

第 3 章：计算机硬件基础及嵌入式系统设计

试题13 (2012年11月试题11)

试题13 (2012年11月试题11)

以下关于嵌入式系统硬件抽象层的叙述，错误的是（ 11 ）。

- (11) A.硬件抽象层与硬件密切相关，可对操作系统隐藏硬件的多样性
- B.硬件抽象层将操作系统与硬件平台隔开
- C.硬件抽象层使软硬件的设计与调试可以并行
- D.硬件抽象层应包括设备驱动程序和任务调度

试题分析

针对不同的硬件平台，操作系统通常建立在一个抽象硬件层上，该抽象层位于底层硬件和内核之间，为内核提供各种方便移植的宏定义接口，在不同的平台间移植时，只需要修改宏定义即可。在硬件抽象层中，封装了与特定硬件有关的各种类型定义、数据结构和各种接口。硬件抽象层提供的接口包括I/O接口、中断处理、异常处理、Cache处理和对称多处理等。

根据抽象程度的不同，硬件抽象层的结构可以分为以下三个级别：

(1) 系统结构抽象层。该层抽象了CPU核的特征，包括中断的传递、异常处理、上下文切换和CPU的启动等。

(2) 处理器变种抽象层。该层抽象了CPU变种的特征，例如，cache、内存管理部件、浮点处理器和片上部件（存储器、中断控制器）等。

(3) 平台抽象层。该层抽象了不同平台的特征，例如，片外器件定时器和I/O寄存器等。

每个EOS都有一个内核，大多数内核都包含调度器、内核对象和内核服务三个公共构件。其中调度器是EOS的心脏，提供一组算法决定何时执行哪个任务；内核对象是特殊的内核构件，帮助创建嵌入式应用；内核服务是内核在对象上执行的操作或通用操作。

试题答案

(11) D

第3章：计算机硬件基础及嵌入式系统设计

试题14（2012年11月试题12）

试题14（2012年11月试题12）

(12) 不是反映嵌入式实时操作系统实时性的评价指标。

- (12) A.任务执行时间 B.中断响应和延迟时间
C.任务切换时间 D.信号量混洗时间

试题分析

影响嵌入式操作系统实时性的6个主要因素如下。

① 常用系统调用平均运行时间：即系统调用效率，是指内核执行常用的系统调用所需的平均时间。

② 任务切换时间：任务切换时间是指事件引发切换后，从当前任务停止运行、保存运行状态（CPU寄存器内容），到装入下一个将要运行的任务状态、开始运行的时间间隔。任务切换时间也称为上下文切换时间，是指CPU控制权由运行态的任务转移给另外一个就绪任务所需要的时间，包括在进行任务切换时，保存和恢复任务上下文所花费的时间，以及选择下一个待运行任务的调度时间。该指标与微处理器的寄存器数目和系统结构有关。相同的操作系统在不同的微处理器上运行时，所花费的时间可能不同。

③ 线程切换时间：线程是可被调度的最小单位。在嵌入式系统的应用系统中，很多功能是以线程的方式执行的，所以线程切换时间同样是考察的一个要点。测试方法及原理与任务切换类似，不再介绍。

④ 任务抢占时间：任务抢占时间是高优先级的任务从正在运行的低优先级任务中获得系统控制权所消耗的时间。

⑤ 信号量混洗时间：信号量混洗时间是指从一个任务释放信号量到另一个等待该信号量的任务被激活的时间延迟。在嵌入式系统中，通常有许多任务同时竞争某一共享资源，基于信号量的互斥访问保证了任一时刻只有一个任务能够访问公共资源。信号量混洗时间反映了与互斥有关的时间开销，是RTOS实时性的一个重要指标。

⑥ 中断处理相关时间：中断延迟时间是指从中断发生到系统获知中断的时间，主要受系统最大关中断时间的影响，关中断的时间越长，中断延迟也就越长。最大关中断时间包含两个方面，一是内核最大关中断时间，即内核在执行临界区代码时关闭中断，二是应用关中断时间。关中断最大时间是这两种关中断时间的最大值。

中断响应时间是指从中断发生到开始执行用户中断服务例程的时间；中断恢复时间是指用户中断服务例程结束回到被中断的代码之间的时间。对于可抢占式调度，中断恢复时间还要加上进行任务切换和恢复新的任务上下文的时间；任务响应时间是指从任务对应的中断产生到该任务真正开始运行的时间。

试题答案

(12) A

第4章：数据通信与计算机网络

考点分析

第4章数据通信与计算机网络

4.1 考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 数据通信的基本知识。
- (2) 开放系统互连参考模型。
- (3) 常用的协议标准。
- (4) 网络互连与常用网络设备。
- (5) 计算机网络的分类与应用。

从历年的考试情况来看，本章的考点主要集中在网络设计相关的知识，偏宏观而非技术细节。

具体考查知识点分布情况如表4-1所示。

表4-1 历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题13~15	网络层次化设计、物理网络设计、网络存储
2010年11月试题13~15	网络层次化设计、逻辑网络设计、网络系统生命周期
2011年11月试题13~15	网络层次化设计、综合布线、逻辑网络设计
2012年11月试题13~15	网络控制、域名服务器、网络存储

试题1 (2009年11月试题13)

4.2 试题精解

试题1 (2009年11月试题13)

以下关于网络核心层的叙述中，正确的是（13）。

- (13) A.为了保障安全性，应该对分组进行尽可能多的处理
- B.在区域间高速地转发数据分组
- C.由多台二、三层交换机组成
- D.提供多条路径来缓解通信瓶颈

试题分析

为了能够更好地分析与设计复杂的大型互连网络，在计算机网络设计中，主要采用分层（分级）设计模型，它类似于软件工程中的结构化设计。通过一些通用规则来设计网络，就可以简化设计、优化带宽的分配和规划。在分层设计中，引入了三个关键层的概念，分别是核心层、汇聚层和接入层。

通常将网络中直接面向用户连接或访问网络的部分称为接入层，将位于接入层和核心层之间的部分称为分布层或汇聚层。接入层的目的是允许终端用户连接到网络，因此，接入层交换机具有低成本和高端口密度特性。

汇聚层是核心层和接入层的分界面，完成网络访问策略控制、数据包处理、过滤、寻址，以及其他数据处理的任务。汇聚层交换机是多台接入层交换机的汇聚点，它必须能够处理来自接入层设备的所有通信量，并提供到核心层的上行链路，因此，汇聚层交换机与接入层交换机比较，需要更高的性能，更少的接口和更高的交换速率。

网络主干部分称为核心层，核心层的主要目的在于通过高速转发通信，提供优化、可靠的骨干传输结构，因此，核心层交换机应拥有更高的可靠性，性能和吞吐量。核心层为网络提供了骨干组件或高速交换组件，在纯粹的分层设计中，核心层只完成数据交换的特殊任务。需要根据网络需求的地理距离、信息流量和数据负载的轻重来选择核心层技术，常用的技术包括ATM、100Base-Fx和千兆以太网等。在主干网中，考虑到高可用性的需求，通常会使用双星（树）结构，即采用两台同样的交换机，与汇聚层交换机分别连接，并使用链路聚合技术实现双机互联。

试题答案

- (13) B

试题2 (2009年11月试题14)

试题2 (2009年11月试题14)

网络开发过程中，物理网络设计阶段的任务是（14）。

- (14) A. 依据逻辑网络设计的功能要求，确定设备的具体物理分布和运行环境
B. 分析现有网络和新网络的各类资源分布，掌握网络所处状态
C. 根据需求规范和通信规范，实施资源分配和安全规划
D. 理解网络应该具有的功能和性能，最终设计出符合用户需求的网络

试题分析

网络的生命周期至少包括网络系统的构思计划、分析设计、实时运行和维护的过程。对于大多数网络系统来说，由于应用的不断发展，这些网络系统需要不断重复设计、实施、维护的过程。

完成网络规划之后，将进入网络系统的设计阶段，这个阶段通常包括确定网络总体目标和设计原则，进行网络总体设计和拓扑结构设计，确定网络选型和进行网络安全设计等方面的内容。

(1) 确定网络总体目标。明确采用哪些网络技术和标准，构筑一个满足哪些应用的多大规模的网络，包括是否分期实施、网络的实施成本和运行成本等方面的问题。

(2) 确定总体设计原则。对主要设计原则进行选择和权衡，并确定其在方案设计中的优先级。网络设计的一些基本原则实用性原则、开放性原则、高可用性/可靠性原则、安全性原则、先进性原则、易用性原则和可扩展性原则等。

(3) 通信子网设计。通信子网设计包括拓扑结构与网络总体规划、分层的设计，以及远程接入访问的设计。其中，拓扑结构与网络总体规划是整个网络设计的基础，通常应该结合费用、灵活性和可靠性三个方面综合考虑。

(4) 资源子网设计。资源子网设计主要考虑服务器的接入和子网连接的问题。服务器通常是网络中的核心设备，包括为全网服务的服务器和为部门业务服务的服务器两类，每类服务器可以采用不同的接入方式。

(5) 设备选型。设备选型包括网络设备和各个层次的交换机的选择策略。网络设备的选型应考虑厂商选择原则（尽可能选取同一厂商，选择产品线全、技术认证队伍力量强、市场占有率为高的网络设备品牌）、扩展性原则（主要预留扩展，低端够用即可），根据方案实际需要选型（性能、端口类型和端口密度等），选择性价比高、质量好的设备；核心交换机的选型策略是设备应具备高性能和高速率、定位准确、便于升级和扩展、高可靠性、强大网络控制能力、良好可管理性等特点；汇聚层/接入层交换机的选型策略是应具备灵活性、高性能，在满足要求的基础上尽量便宜、易用、简单，具备一定的QoS和控制能力，支持多级网络管理等特点。

(6) 网络操作系统与服务器资源设备。选择服务器时，首先要看具体的网络应用，然后确定网络操作系统，再进行服务器选型。网络操作系统的选型要点是，要结合服务器的性能和兼容性、安全因素、价格因素、第三方软件和市场占有率为等方面进行综合考查。根据需要，还应配置服务器集群或双机容错系统等，以便实现更好的性能。

(7) 网络安全设计。网络安全设计的基本原则有木桶原则、整体性原则、有效性与实用性原则、等级性原则、设计为本原则、自主和可控性原则、安全有价原则等。网络安全设计与实施的步骤是，确定面临的攻击和风险，明确安全策略，建立安全模型，选择并实现安全服务，对安全产品的选型进行测试。不同的网络安全技术所控制的范围不一样。

在设计阶段上，可以将网络设计细分为逻辑结构设计和物理结构设计两个阶段。

网络逻辑结构设计是体现网络设计核心思想的关键阶段，在这一阶段根据需求规范和通信规范，选择一种比较适宜的网络逻辑结构，并基于该逻辑结构实施后续的资源分配规划、安全规划等内容。

物理网络设计是对逻辑网络设计的物理实现，通过对设备的具体物理分布、运行环境等的确定，确保网络的物理连接符合逻辑连接的要求。在这一阶段，网络设计者需要确定具体的软硬件、连接设备、布线和服务。

试题答案

(14) A

第 4 章：数据通信与计算机网络

试题3 (2009年11月试题15)

试题3 (2009年11月试题15)

希赛公司欲构建一个网络化的开放式数据存储系统，要求采用专用网络连接并管理存储设备和存储管理子系统。针对这种应用，采用 (15) 存储方式最为合适。

- (15) A.内置式存储 B.DAS
C.SAN D.NAS

试题分析

目前，主流的网络存储技术主要有三种，分别是直接附加存储 (Direct Attached Storage,DAS) 、网络附加存储 (Network Attached Storage,NAS) 和存储区域网络 (Storage Area Network,SAN) 。

1. 直接附加存储

DAS是将存储设备通过SCSI (Small Computer System Interface,小型计算机系统接口) 电缆直接连到服务器，其本身是硬件的堆叠，存储操作依赖于服务器，不带有任何存储操作系统。因此，有些文献也把DAS称为SAS (Server Attached Storage,服务器附加存储) 。

DAS的适用环境为：

- (1) 服务器在地理分布上很分散，通过SAN或NAS在它们之间进行互连非常困难时；
(2) 存储系统必须被直接连接到应用服务器 (例如，Microsoft Cluster Server或某些数据库使用的"原始分区") 上时；
(3) 包括许多数据库应用和应用服务器在内的应用，它们需要直接连接到存储器上时。

由于DAS直接将存储设备连接到服务器上，这导致它在传递距离、连接数量、传输速率等方面都受到限制。因此，当存储容量增加时，DAS方式很难扩展，这对存储容量的升级是一个巨大的瓶颈；另一方面，由于数据的读取都要通过服务器来处理，必然导致服务器的处理压力增加，数据处理和传输能力将大大降低；此外，当服务器出现宕机等异常时，也会波及到存储数据，使其无法使用。目前DAS基本被NAS所代替。

2. 网络附加存储

采用NAS技术的存储设备不再通过I/O总线附属于某个特定的服务器，而是通过网络接口与网络直接相连，由用户通过网络访问。NAS存储系统的结构如图4-1所示。

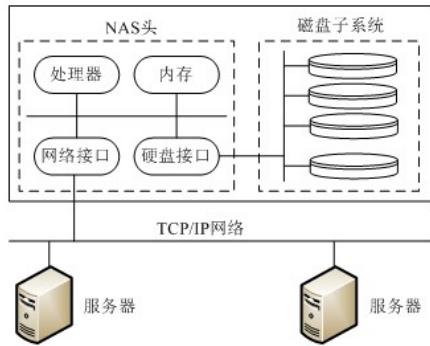


图4-1 NAS存储系统的结构

NAS存储设备类似于一个专用的文件服务器，它去掉了通用服务器的大多数计算功能，而仅仅提供文件系统功能，从而降低了设备的成本。并且为方便存储设备到网络之间以最有效的方式发送数据，专门优化了系统硬件体系结构。NAS以数据为中心，将存储设备与服务器分离，其存储设备在功能上完全独立于网络中的主服务器，客户机与存储设备之间的数据访问不再需要文件服务器的干预，同时它允许客户机与存储设备之间进行直接的数据访问，所以不仅响应速度快，而且数据传输速率也很高。

NAS技术支持多种TCP/IP网络协议，主要是NFS (Net File System, 网络文件系统) 和CIFS (Common Internet File System, 通用Internet文件系统) 来进行文件访问，所以NAS的性能特点是进行小文件级的共享存取。在具体使用时，NAS设备通常配置为文件服务器，通过使用基于Web的管理界面来实现系统资源的配置、用户配置管理和用户访问登录等。

NAS存储支持即插即用，可以在网络的任一位置建立存储。基于Web管理，从而使设备的安装、使用和管理更加容易。NAS可以很经济地解决存储容量不足的问题，但难以获得满意的性能。

3. 存储区域网络

SAN是通过专用交换机将磁盘阵列与服务器连接起来的高速专用子网。它没有采用文件共享存取方式，而是采用块 (block) 级别存储。SAN是通过专用高速网将一个或多个网络存储设备和服务器连接起来的专用存储系统，其最大特点是将存储设备从传统的以太网中分离了出来，成为独立的存储区域网络SAN的系统结构如图4-2所示。

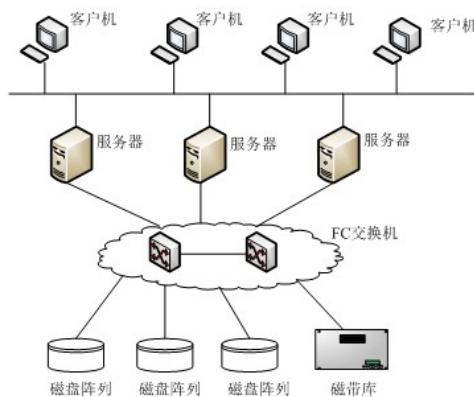


图4-2 SAN存储系统的结构

根据数据传输过程采用的协议，其技术划分为FC SAN和IP SAN.另外，还有一种新兴的IB SAN技术。

(1) FC SAN.FC (Fiber Channel, 光纤通道) 和SCSI接口一样，最初也不是为硬盘设计开发的接口技术，而是专门为网络系统设计的，随着存储系统对速度的需求，才逐渐应用到硬盘系统中。光纤通道的主要特性有：热插拔性、高速带宽、远程连接、连接设备数量大等。它是当今最昂贵和复杂的存储架构，需要在硬件、软件和人员培训方面进行大量投资。

FC SAN由三个基本的组件构成，分别是接口（SCSI、FC）、连接设备（交换机、路由器）和协议（IP、SCSI）。这三个组件再加上附加的存储设备和服务器就构成一个SAN系统。它是专用、高速、高可靠的网络，允许独立、动态地增加存储设备，使得管理和集中控制更加简化。

FC SAN有两个较大的缺陷，分别是成本和复杂性，其原因就是因为使用了FC。在光纤通道上部署SAN，需要每个服务器上都要有FC适配器、专用的FC交换机和独立的布线基础架构。这些设施使成本大幅增加，更不用说精通FC协议的人员培训成本。

（2）IP SAN.IP SAN是基于IP网络实现数据块级别存储方式的存储网络。由于设备成本低，配置技术简单，可共享和使用大容量的存储空间，因而逐渐获得广泛的应用。

在具体应用上，IP存储主要是指iSCSI（Internet SCSI）。作为一种新兴的存储技术，iSCSI基于IP网络实现SAN架构，既具备了IP网络配置和管理简单的优势，又提供了SAN架构所拥有的强大功能和扩展性。iSCSI是连接到一个TCP/IP网络的直接寻址的存储库，通过使用TCP/IP协议对SCSI指令进行封装，可以使指令能够通过IP网络进行传输，而过程完全不依赖于地点。

iSCSI优势的主要表现在于，首先，建立在SCSI、TCP/IP这些稳定和熟悉的标准上，因此安装成本和维护费用都很低；其次，iSCSI支持一般的以太网交换机而不是特殊的光纤通道交换机，从而减少了异构网络和电缆；最后，iSCSI通过IP传输存储命令，因此可以在整个Internet上传输，没有距离限制。

iSCSI的缺点在于，存储和网络是同一个物理接口，同时协议本身的开销较大，协议本身需要频繁地将SCSI命令封装到IP包中以及从IP包中将SCSI命令解析出来，这两个因素都造成了带宽的占用和主处理器的负担。但是，随着专门处理iSCSI指令的芯片的开发（解决主处理器的负担问题），以及10G以太网的普及（解决带宽问题），iSCSI将有着更好的发展。

（3）IB SAN.IB（InfiniBand,无限带宽）是一种交换结构I/O技术，其设计思路是通过一套中心机构（IB交换机）在远程存储器、网络以及服务器等设备之间建立一个单一的连接链路，并由IB交换机来指挥流量。这种结构设计得非常紧密，大大提高了系统的性能、可靠性和有效性，能缓解各硬件设备之间的数据流量拥塞。而这是许多共享总线式技术没有解决好的问题，因为在共享总线环境中，设备之间的连接都必须通过指定的端口建立单独的链路。

IB主要支持两种环境：模块对模块的计算机系统（支持I/O模块附加插槽）；在数据中心环境中的机箱对机箱的互连系统、外部存储系统和外部局域网/广域网访问设备。IB支持的带宽比现在主流的I/O载体（例如，SCSI、FC等）还要高，另外，由于使用IPv6的报头，IB还支持与传统Internet/Intranet设施的有效连接。用IB技术替代总线结构所带来的最重要的变化就是建立了一个灵活、高效的数据中心，省去了服务器复杂的I/O部分。

IB SAN采用层次结构，将系统的构成与接入设备的功能定义分开，不同的主机可通过HCA（Host Channel Adapter,主机通道适配器）、RAID等网络存储设备利用TCA（Target Channel Adapter,目标通道适配器）接入IB SAN。

IB SAN主要具有如下特性：可伸缩的Switched Fabric互连结构；由硬件实现的传输层互连高效、可靠；支持多个虚信道；硬件实现自动的路径变换；高带宽，总带宽随IB Switch规模成倍增长；支持SCSI远程DMA（Direct Memory Access,直接内存存取）协议；具有较高的容错性和抗毁性，支持热拔插。

5. 网络存储技术的选择

网络存储技术的目的都是为了扩大存储能力，提高存储性能。这些存储技术都能提供集中化的数据存储并有效存取文件；都支持多种操作系统，并允许用户通过多个操作系统同时使用数据；都

可以从应用服务器上分离存储，并提供数据的高可用性；同时，都能通过集中存储管理来降低长期的运营成本。

因此，从存储的本质上来看，它们的功能都是相同的。事实上，它们之间的区别正在变得模糊，所有的技术都在用户的存储需求下接受挑战。在实际应用中，需要根据系统的业务特点和要求（例如，环境要求、性能要求、价格要求等）进行选择。

试题答案

(15) C

第4章：数据通信与计算机网络

试题4（2010年11月试题13）

试题4（2010年11月试题13）

大型局域网通常划分为核心层、汇聚层和接入层，以下关于各个网络层次的描述中，不正确的是（13）。

- (13) A.核心层进行访问控制列表检查
- B.汇聚层定义了网络的访问策略
- C.接入层提供局域网络接入功能
- D.接入层可以使用集线器代替交换机

试题分析

接入层的目的是允许终端用户连接到网络，因此，接入层交换机具有低成本和高端口密度特性。

汇聚层是核心层和接入层的分界面，完成网络访问策略控制、数据包处理、过滤、寻址，以及其他数据处理的任务。汇聚层交换机是多台接入层交换机的汇聚点，它必须能够处理来自接入层设备的所有通信量，并提供到核心层的上行链路，因此，汇聚层交换机与接入层交换机比较，需要更高的性能，更少的接口和更高的交换速率。

核心层的主要目的在于通过高速转发通信，提供优化、可靠的骨干传输结构，因此，核心层交换机应拥有更高的可靠性，性能和吞吐量。核心层为网络提供了骨干组件或高速交换组件，在纯粹的分层设计中，核心层只完成数据交换的特殊任务。需要根据网络需求的地理距离、信息流量和数据负载的轻重来选择核心层技术，常用的技术包括ATM、100Base-Fx和千兆以太网等。在主干网中，考虑到高可用性的需求，通常会使用双星（树）结构，即采用两台同样的交换机，与汇聚层交换机分别连接，并使用链路聚合技术实现双机互联。

通过以上的分析可以得知“核心层进行访问控制列表检查”的说法是不正确的。

试题答案

(13) A

试题5 (2010年11月试题14)

试题5 (2010年11月试题14)

网络系统设计过程中，逻辑网络设计阶段的任务是（14）。

- (14) A. 依据逻辑网络设计的要求，确定设备的物理分布和运行环境
- B. 分析现有网络和新网络的资源分布，掌握网络的运行状态
- C. 根据需求规范和通信规范，实施资源分配和安全规划
- D. 理解网络应该具有的功能和性能，设计出符合用户需求的网络

试题分析

在逻辑网络设计阶段，需要描述满足用户需求的网络行为以及性能，详细说明数据是如何在网絡上阐述的，此阶段不涉及网络元素的具体物理位置。

网络设计者利用需求分析和现有网络体系分析的结果来设计逻辑网络结构。如果现有的软件、硬件不能满足新网络的需求，现有系统就必须升级。如果现有系统能继续运行使用，可以将它们集成到新设计中来。如果不集成旧系统，网络设计小组可以找一个新系统，对它进行测试，确定是否符合用户的需求。

此阶段最后应该得到一份逻辑网络设计文档，输出的内容包括以下几点：

- ① 逻辑网络设计图；
- ② IP地址方案；
- ③ 安全方案；
- ④ 具体的软件、硬件、广域网连接设备和基本的服务；
- ⑤ 雇佣和培训新网络员工的具体说明；
- ⑥ 初步对软件、硬件、服务、网络雇佣员工和培训的费用估计。

物理网络设计阶段的任务是如何实现确定的逻辑网络结构。在这一阶段，网络设计者需要确定具体的软件、硬件、连接设备、服务和布线。

如何购买和安装设备，由网络物理结构这一阶段的输出作指导，所以网络物理设计文档必须尽可能详细、清晰，输出的内容如下：

- ① 物理网络图和布线方案；
- ② 设备和部件的详细列表清单；
- ③ 软件、硬件和安装费用的估计；
- ④ 安装日程表，用以详细说明实际和服务中断的时间以及期限；
- ⑤ 安装后的测试计划；
- ⑥ 用户培训计划。

试题答案

C

试题6 (2010年11月试题15)

试题6 (2010年11月试题15)

网络系统生命周期可以划分为5个阶段，实施这5个阶段的合理顺序是 (15) .

- (15) A.需求规范、通信规范、逻辑网络设计、物理网络设计、实施阶段
- B.需求规范、逻辑网络设计、通信规范、物理网络设计、实施阶段
- C.通信规范、物理网络设计、需求规范、逻辑网络设计、实施阶段
- D.通信规范、需求规范、逻辑网络设计、物理网络设计、实施阶段

试题分析

一个网络系统从构思开始，到最后被淘汰的过程被称为网络的生命周期；一般来说网络的生命周期至少包括网络系统的构思计划、分析设计、实时运行和维护的过程；对于大多数网络系统来说，由于应用的不断发展，这些网络系统需要不断重复设计、实施、维护的过程。

因此，网络系统的生命周期和软件工程中的软件生命周期非常类似，首先是一个循环迭代的过程，每次循环迭代的动力都来自于网络应用需求的变更；其次每次循环过程中，都存在需求分析、规划设计、实施调试和运营维护等阶段。有些网络仅仅经过一个周期就被淘汰，而有些网络在存活过程中经过多次循环周期，一般来说，网络规模越大、投资越多，则其可能经历的循环周期也越多。

1. 网络生命周期的迭代模型

网络生命周期的迭代模型的核心思想是网络应用驱动理论和成本评价机制，当网络系统无法满足用户的需求时，就必须进入到下一个迭代周期，经过迭代周期后，网络系统将能够满足用户的网络需求；成本评价机制决定是否结束网络系统的生命周期，当对已有投资的再利用成本小于新建系统的成本时，网络系统可以进入下一次迭代周期，而再利用成本大于新建成本时，就必须舍弃迭代，终结当前网络系统，新建网络系统。网络生命周期的迭代模型如图4-3所示。

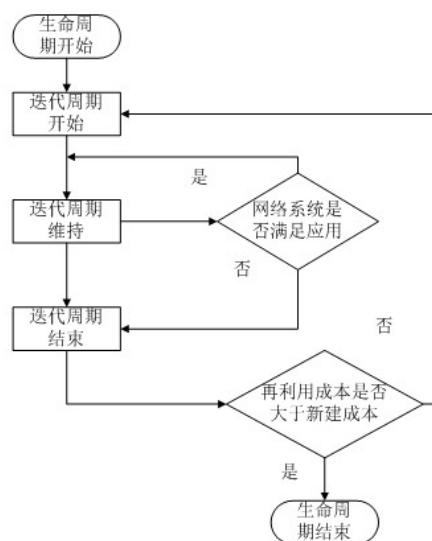


图4-3 网络生命周期的迭代模型

2. 迭代周期的构成

每一个迭代周期，都是一个网络重构的过程，不同的网络设计方法中，对迭代周期的划分方式是不同的；这些划分方式侧重点不同，拥有不同的网络文档模板，但是实施后的效果都是满足了用户的网络需求；目前没有哪个迭代周期可以完美描述所有项目的开发构成，但是常见的构成方式主要有三种。

(1) 四阶段周期

如图4-4所示，四个阶段分别为构思与规划阶段、分析与设计阶段、实施与构建阶段和运行与维护阶段，这四个阶段之间有一定的重叠，保证了两个阶段之间的交接工作，同时也赋予了网络工程设计的灵活性。

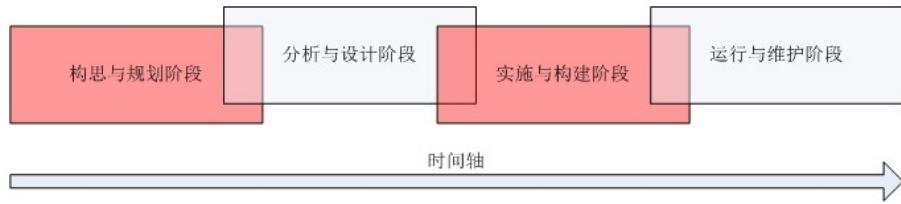


图4-4 四阶段周期

构思与规划阶段的主要工作是对明确网络设计或改造的需求，同时对新网络的建设目标进行明确；分析与设计阶段的工作在于根据网络的需求进行设计，并形成特定的设计方案；实施与构建阶段的工作在于根据设计方案进行设备购置、安装、调试，形成可试用的网络环境；运行维护阶段提供网络服务，并实施网络管理。

四阶段周期的特点是，能够快速适应新的需求，强调网络建设周期中的宏观管理，灵活性较强。四阶段周期的长处在于工作成本较低、灵活性高，适用于网络规模较小、需求较为明确、网络结构简单的网络工程。

(2) 五阶段周期

五阶段周期是较为常见的迭代周期划分方式，将一次迭代划分为五个阶段。

需求规范；

通信规范；

逻辑网络设计；

物理网络设计；

实施阶段。

在五个阶段中，由于每个阶段都是一个工作环节，每个环节完毕后才能进入到下一个环节，类似于软件工程中的“瀑布模型”，形成了特定的工作流程。如图4-5所示。

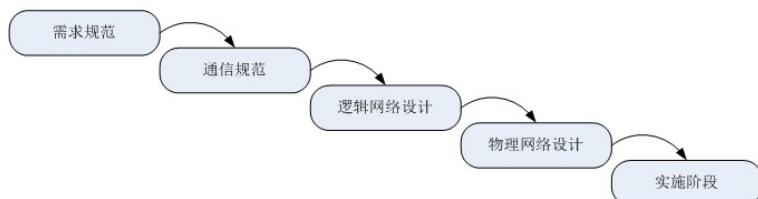


图4-5 五阶段周期

按照这种流程构建网络，在下一个阶段开始之前，前面的每个阶段的工作必须已经完成。一般情况下，不允许返回到前面的阶段，如果出现前一阶段的工作没有完成就开始进入下一个阶段，则会对后续的工作造成较大的影响，甚至产生工期拖后和成本超支。

主要优势在于所有的计划在较早的阶段完成，该系统的所有负责人对系统的具体情况以及工作进度都非常清楚，更容易协调工作。

五阶段周期的缺点是比较死板，不灵活。因为往往在项目完成之前，用户的需求经常会发生变化，这使得已开发的部分需要经常修改，从而影响工作的进程，所以基于这种流程完成网络设计时，用户的需求确认工作非常重要。

五阶段周期由于存在较为严格的需求和通信分析规范，并且在设计过程中充分考虑了网络的逻辑特性和物理特性，因此较为严谨，适用于网络规模较大，需求较为明确，在一次迭代过程中需求变更较小的网络工程。

(2) 六阶段周期

六阶段周期是对五阶段周期的补充，是对其缺乏灵活性的改进，通过在实施阶段前后增加相应的测试和优化过程，提高网络建设工作中对需求变更的适应性。

六个阶段分别由需求分析、逻辑设计、物理设计、设计优化、实施及测试、监测及性能优化组成，如图4-6所示。

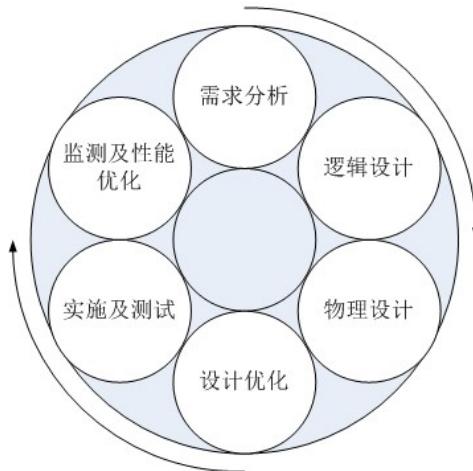


图4-6 六阶段周期

需求分析阶段，网络分析人员通过与用户和技术人员进行交流来获取对新的或升级系统的商业和技术目标，然后归纳出当前网络的特征，分析出当前和将来的网络通信量、网络性能，包括流量、负载、协议行为和服务质量要求。

逻辑设计阶段，主要完成网络的逻辑拓扑结构、网络编址、设备命名、交换及路由协议选择、安全规划、网络管理等设计工作，并且根据这些设计产生对设备厂商、服务提供商的选择策略。

物理设计阶段，根据逻辑设计的成果，选择具体的技术和产品，使得逻辑设计成果符合工程设计规范。

设计优化阶段，该阶段完成在实施阶段前的方案优化，通过召开专家研讨会、搭建试验平台、网络仿真等多种形式，找出设计方案中的缺陷，并进行方案优化。

实施及测试阶段，该阶段根据优化后的方案进行设备的购置、安装、调试与测试，通过测试和试用，发现网络环境与设计方案的偏离，纠正实施过程中的错误，甚至可能导致修改网络设计方案。

监测及性能优化阶段，该阶段是网络的运营和维护阶段，通过网络管理、安全管理等技术手段，对网络是否正常运行进行实时监控，一旦发现问题，通过优化网络设备配置参数，达到优化网络性能的目的；一旦发现网络性能已经无法满足用户需求，则进入下一次迭代周期。

六阶段周期偏重于网络的测试和优化，侧重于网络需求的不断变更，由于其严格的逻辑设计和物理设计规范，使得该模式适合于大型网络的建设工作。

试题答案

(15) A

试题7 (2011年11月试题13)

试题7 (2011年11月试题13)

核心层交换机应该实现多种功能，下面选项中，不属于核心层特性的是（13）。

- (13) A.高速连接 B.冗余设计
- C.策略路由 D.较少的设备连接

试题分析

请参考试题4的分析。

试题答案

- (13) C

第4章：数据通信与计算机网络

试题8 (2011年11月试题14)

试题8 (2011年11月试题14)

建筑物综合布线系统中的垂直子系统是指（14）。

- (14) A.由终端到信息插座之间的连线系统 B.楼层接线间的配线架和线缆系统
- C.各楼层设备之间的互连系统 D.连接各个建筑物的通信系统

试题分析

结构化布线系统是一个能够支持任何用户选择的话音、数据、图形图像应用的电信布线系统。

系统应能支持话音、图形、图像、数据多媒体、安全监控、传感等各种信息的传输，支持UTP、光纤、STP、同轴电缆等各种传输载体，支持多用户多类型产品的应用，支持高速网络的应用。

结构化布线系统具有以下特点：

- (1) 实用性。支持多种数据通信、多媒体技术及信息管理系统等，适应现代和未来技术的发展。
- (2) 灵活性。任意信息点能够连接不同类型的设备，如计算机、打印机、终端、服务器等；
- (3) 开放性。能够支持任何厂家的任意网络产品，支持任意网络结构，如总线形、星形、环型等；
- (4) 模块化。所有的接插件都是积木式的标准件，方便使用、管理和扩充；
- (5) 扩展性。实施后的结构化布线系统是可扩充的，以便将来有更大需求时，很容易将设备安装接入；
- (6) 经济性。一次性投资，长期受益，维护费用低，使整体投资达到最少。

结构化布线系统分为六个子系统：工作区子系统、水平干线子系统、垂直干线子系统、设备间子系统、管理间子系统、建筑群子系统，如图4-7所示。

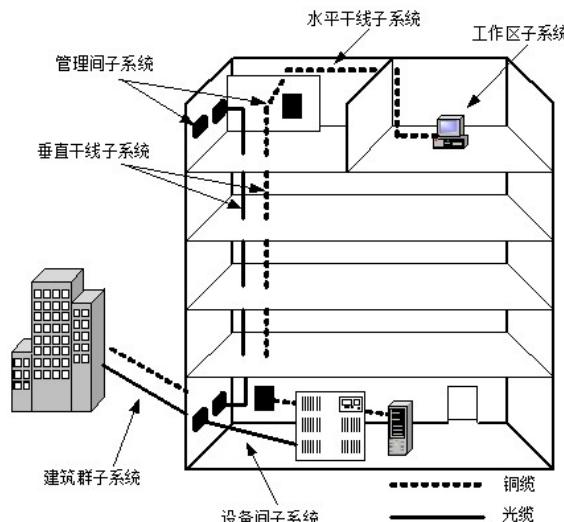


图4-7 结构化布线系统示意图

(1) 工作区子系统

工作区布线子系统由终端设备连接到信息插座的连线（或软线）组成，包括装配软线、适配器和连接所需的扩展软线，并在终端设备和I/O之间搭桥，如图4-8所示。

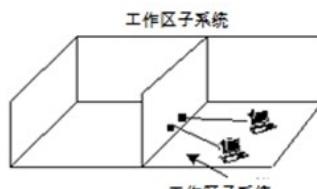


图4-8 工作区子系统

工作区子系统的布线通常是非永久性的，但在设计阶段可根据用户的需要增加或改变，既便于连接也易于管理。工作区子系统中的信息插座类型选择应根据网络系统的规模和终端设备的种类、数量而定。

工作区子系统中的布线、信息插座通常安装在工作间四周的墙壁下方，也有的安装在用户的办公桌上，而无论安装在何处，应以方便、安全、不易损坏为目标。

工作区子系统的布线，实质上相当于通信线路的布线，终端包括计算机、电话机、传真机、有线电视机等设备，应使这些终端设备与信息插座（通信引出端）相连。

(2) 水平干线子系统

水平干线子系统也称为水平布线子系统，它的作用是将干线子系统线路延伸到用户工作区，水平子系统区别于干线子系统的区别是：水平布线系统处于同一楼层，并端接在信息插座或区域布线的中转点上；水平布线子系统一端端接于信息插座上，另一端端接在干线接线间或设备机房的管理配线架上；如图4-9所示。



图4-9 水平布线子系统

在进行水平布线时，传输媒体中间不宜有转折点，两端应直接从配线架连接到工作区插座。水平布线的布线通道有两种：一种是暗管预埋、墙面引线方式，另一种是地下管槽、地面引线方式；前者适用于多数建筑系统，一旦铺设完成，不易更改和维护；后者适合于少墙、多柱的环境，更改和维护方便。

(3) 管理间子系统

管理间子系统也称为管理子系统，由交连、互连和配线架和信息插座式配线架以及相关跳线组成。管理点为连接其它子系统提供连接手段，交连和互连允许将通信线路定位或重新定位到建筑物的不同部分，以便能更容易地管理通信线路。

通过卡接或插接式跳线，交叉连接允许将端接在配线架一端的通信线路与端接于另一端配线架上的线路相连。插入线为重新安排线路提供一种简易的方法，而且不需要安装跨接线时使用的专用工具。如图4-10所示。

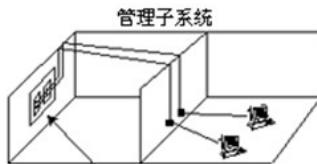


图4-10 管理子系统

对于楼层较少的楼栋来说，管理子系统可以不采用配线间的方式，而是采用悬挂式配线柜。

对于大多数楼栋来说，每一楼层至少有一个配线间，用于放置交换机、集线器和配线架等交叉连接设备。配线架等交叉设备通过水平布线子系统连接至各工作区的信息插座；集线器或交换机与交叉设备之间通过短线缆互连，这些短线被称为跳线；通过跳线的调整，可以方便地形成工作区的信息插座和交换机端口的连接切换。

同时，主干子系统将根据其分布式结构独立地连接到每一个配线间，大多数情况下，管理子系统的配线间至少拥有一条以上的主干线缆。

(4) 垂直干线子系统

垂直干线子系统一般简称为干系子系统，是建筑物内网络系统的中枢，实现各楼层的水平子系统之间的互联。干线子系统提供建筑物的干线（馈电线）电缆的路由；通常由垂直大对数铜缆或光缆组成，一端端接于设备机房的主配线架上，另一端通常端接在楼层接线间的各个管理分配线架上。如图4-11所示。

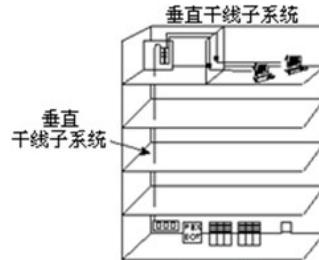


图4-11 干线子系统

主干子系统在设计时，对于旧式建筑，主要采用楼层牵引管等方式铺设，对于新式建筑物，主要利用建筑物的线井进行铺设。

(5) 设备间子系统

设备间子系统由设备间中的跳线电缆、适配器组成，实现中央主配线架与各种不同设备的互连，如PBX、网络设备和监控设备等与主配线架之间的连接。通常设备间子系统设计与网络具体应用有关，相对独立于通用的结构布线系统。如图4-12所示。

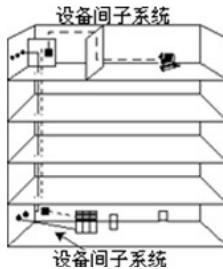


图4-12 设备间子系统

设备间子系统是一幢建筑物中集中存放的各种设备。如主机、电话专用程控交换机、调制解调器的安放柜、网络服务器和局域网络集线器等。就结构化局域网而言，这种设备间应包括一个主要的交叉连接器。

在选择设备间位置时，既要考虑到连接方便的要求，也要兼顾对电磁干扰的要求。考虑到结构化布线的投资、施工安装与维护等，设备室通常选择在一幢建筑物的中部楼层。设备室的供电要求也格外严格，通常要配备不间断电源，还要有备份电源。

(6) 建筑群子系统

建筑群子系统将一个建筑物中的电缆延伸到建筑群的另外一些建筑物中的通信设备和装置上；该子系统是整个布线系统中的一部分，并支持提供楼群之间通信设施所需的硬件，其中有导线电缆、光缆和防止电缆的浪涌电压进入建筑物的电气保护设备。如图4-13所示。

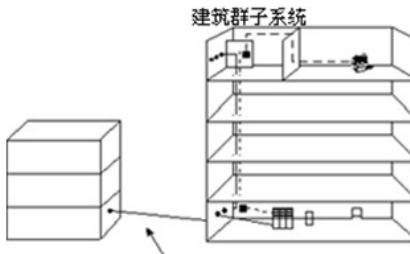


图4-13 建筑群子系统

建筑群子系统主要由连接楼栋的线缆构成，在设计时，应尽量使用地下管道敷设方式，管道内敷设的铜缆或光缆应遵循电话管道和入孔的各项设计规定。此外安装时至少应预留1~2个备用管孔，以供扩充之用。

建筑群子系统采用直埋沟内敷设时，如果在同一个沟内埋入了其他的图像、监控电缆，应设立明显的共用标志。

试题答案

(14) C

试题9 (2011年11月试题15)

试题9 (2011年11月试题15)

网络设计过程包括逻辑网络设计和物理网络设计两个阶段，下面的选项中，(15) 应该属于逻辑网络设计阶段的任务。

- (15) A.选择路由协议 B.设备选型 C.结构化布线 D.机房设计

试题分析

逻辑网络设计包括：网络结构设计、物理层技术选择、局域网技术选择与应用、广域网技术选择与应用、地址设计与命名模型、路由选择协议、网络管理、网络安全、逻辑网络设计文档。

物理网络设计的内容包括：设备选型、结构化布线、机房设计及物理网络设计相关的文档规范（如：软硬件清单，费用清单）。

试题答案

- (15) A

第 4 章：数据通信与计算机网络

试题10 (2012年11月试题13)

试题10 (2012年11月试题13)

以下关于网络控制的叙述，正确的是（13）。

- (13) A.由于TCP的窗口大小是固定的，所以防止拥塞的方法只能是超时重发
B.在前向纠错系统中，当接收端检测到错误后就要请求发送端重发出错分组
C.在滑动窗口协议中，窗口的大小以及确认应答使得可以连续发送多个数据
D.在数据报系统中，所有连续发送的数据都可以沿着预先建立的虚通路传送

试题分析

TCP采用可变大小的滑动窗口协议进行流量控制。在前向纠错系统中，当接收端检测到错误后就根据纠错编码的规律自行纠错；在后向纠错系统中，接收方会请求发送方重发出错分组。IP协议不预先建立虚电路，而是对每个数据报独立地选择路由并一站一站地进行转发，直到送达目标地。

试题答案

- (13) C

第 4 章：数据通信与计算机网络

试题11 (2012年11月试题14)

试题11 (2012年11月试题14)

以下关于域名服务器的叙述，错误的是（14）。

- (14) A.本地缓存域名服务不需要域名数据库
B.顶级域名服务器是最高层次的域名服务器
C.本地域名服务器可以采用递归查询和迭代查询两种查询方式

D.权限服务器负责将其管辖区内的主机域名转换为该主机的IP地址

试题分析

可提供域名服务的包括本地缓存、本地域名服务器、权限域名服务器、顶级域名服务器以及根域名服务器。DNS主机名解析的查找顺序是，先查找客户端本地缓存，如果没有成功，则向DNS服务器发出解析请求。

本地缓存是内存中的一块区域，保存着最近被解析的主机名及其IP地址映像。由于解析程序缓存常驻内存中，所以比其他解析方法速度快。

当一个主机发出DNS查询报文时，这个查询报文就首先被送往该主机的本地域名服务器。本地域名服务器离用户较近，当所要查询的主机也属于同一个本地ISP时，该本地域名服务器立即就能将所查询的主机名转换为它的IP地址，而不需要再去询问其他的域名服务器。

每一个区都设置有域名服务器，即权限域名服务器，它负责将其管辖区内的主机域名转换为该主机的IP地址。在其上保存有所管辖区内的所有主机域名到IP地址的映射。

顶级域名服务器负责管理在本顶级域名服务器上注册的所有二级域名。当收到DNS查询请求时，能够将其管辖的二级域名转换为该二级域名的IP地址。或者是下一步应该找寻的域名服务器的IP地址。

根域名服务器是最高层次的域名服务器。每一个根域名服务器都要存有所有顶级域名服务器的IP地址和域名。当一个本地域名服务器对一个域名无法解析时，就会直接找到根域名服务器，然后根域名服务器会告知它应该去找哪一个顶级域名服务器进行查询。

试题答案

(14) B

第 4 章：数据通信与计算机网络

试题12 (2012年11月试题15)

试题12 (2012年11月试题15)

以下关于网络存储的叙述，正确的是 (15) .

- (15) A.DAS支持完全跨平台文件共享，支持所有的操作系统
- B.NAS通过SCSI连接至服务器，通过服务器网卡在网络上传输数据
- C.FC SAN的网络介质为光纤通道，而IP SAN使用标准的以太网
- D.SAN设备有自己的文件管理系统，NAS中的存储设备没有文件管理系统

试题分析

请参考试题3的分析。

试题答案

(15) C

考点分析

第5章系统安全性与保密性设计

5.1 考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

(1) 信息安全与保密。

加密和解密

身份认证 (数字签名、密钥、口令)

访问控制

安全保密管理 (防泄漏、数字水印)

安全协议 (SSL、PGP、IPSec)

系统备份与恢复

防治病毒

(2) 安全性规章与保护私有信息规则

信息系统安全法规与制度

计算机防病毒制度

保护私有信息规则

从历年的考试情况来看，本章的考点主要集中在与安全相关的基本概念以及网络安全方面。具体考查知识点分布情况如表5-1所示。

表5-1 历年考查知识点分布情况表

试 题	考查知识点
2009 年 11 月试题 64~65	安全策略、对称加密算法
2010 年 11 月试题 64~65	网络安全漏洞、ARP
2011 年 11 月试题 64~65	SNMPv3、PGP
2012 年 11 月试题 64~65	PKI

试题1 (2009年11月试题64)

5.2 试题精解

试题1 (2009年11月试题64)

信息安全策略应该全面地保护信息系统整体的安全，网络安全体系设计是网络逻辑设计工作的重要内容之一，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安全体系的设计与规划。其中，数据库的容灾属于 (64) 的内容。

(64) A.物理线路安全与网络安全 B.网络安全与系统安全

C.物理线路安全与系统安全 D.系统安全与应用安全

试题分析

网络安全体系是建设网络工程的有效内容之一，不管网络工程规模如何，都应该存在一个可扩展的总体安全体系框架。对于不同的网络工程，允许建设不同的安全体系框架，图5-1就是一个较为常见、可行的安全体系框架，设计人员在进行网络安全需求收集时，可以依据该框架进行安全需求的明确。



图5-1 安全体系框架的示例

在图5-1的安全体系框架中，安全管理体系是整个安全架构的基础，使安全问题可控可管；以安全技术为核心的技术措施（包括机房及物理线路安全、网络安全、系统安全、应用安全、安全信任体系等），使安全手段更加可靠；以容灾与恢复为目标的后备保障措施，对付重大灾难性事件的网络重建；以安全运维支持服务作为外部支撑条件，使安全问题能够及时有效地解决。

基于以上框架，对于网络安全的需求，设计人员应该协助网络管理人员，对安全管理体系、运营服务体系、数据容灾与恢复、安全信任体系等方面的需求进行确定。同时，对于技术措施需求，可以借鉴表5-2的内容进行明确。

表5-2 技术措施需求表

技术措施层次	需求项目	需求项目	需求项目	需求项目
机房及物理线路安全需求	机房安全	计算机通信线路安全	骨干线路冗余防护	主要设备的防雷击措施
网络安全需求	安全区域划分	安全区域级别	区域内部安全策略	区域边界安全策略
	路由设备安全	网闸	防火墙	入侵检测
	抗DDOS	VPN	流量管理	网络监控与审计
	网络监控与审计	访问控制		
系统安全需求	身份认证	帐户管理	主机系统配置管理	漏洞发现与补丁管理
	内核加固	病毒防护	桌面安全管理	系统备份与恢复
	系统监控与审计	访问控制		
应用安全需求	数据库安全	邮件服务安全	Web服务安全	应用系统定制安全

试题答案

(64) D

公司总部与分部之间需要传输大量数据，在保障数据安全的同时又要兼顾密钥算法效率，最适合的加密算法是（65）。

- (65) A.RC-5 B.RSA C.ECC D.MD5

试题分析

加密是指对数据进行编码变换，使其看起来毫无意义，但同时却仍可以保持其可恢复的形式的过程。在这个过程中，被变换的数据称为明文，它可以是一段有意义的文字或者数据，变换后的数据称为密文。加密机制有助于保护信息的机密性和完整性，有助于识别信息的来源，它是最广泛使用的安全机制。

传统加密算法主要包括代换加密和置换加密两种。代换加密的方法是，首先构造一个或多个密文字母表，然后用密文字母表中的字母或字母组来代替明文字母或字母组，各字母或字母组的相对位置不变，但其本身改变了；置换加密将明文中的字母重新排列，字母本身不变，但其位置改变了。传统加密算法具有设计简单的特点，曾经得到大量使用。但它们的一个主要弱点是，算法和密钥密切相关。攻击者可以根据字母的统计和语言学知识，相对容易破译密文，尤其是在计算机技术高度发展的今天，可以充分利用计算机进行密文分析。

从20世纪60年代开始，随着电子技术、计算机技术的迅速发展，以及结构代数、可计算性和计算复杂性理论等学科的研究，密码学又进入了一个新的发展时期。在20世纪70年代后期，对称加密算法和非对称加密算法的出现，成为了现代密码学发展史中的两个重要里程碑。

1. 对称加密算法

对称加密算法也称为私钥加密算法，是指加密密钥和解密密钥相同，或者虽然不同，但从其中的任意一个可以很容易地推导出另一个。其优点是具有很高的保密强度，但密钥的传输需要经过安全可靠的途径。对称加密算法有两种基本类型，分别是分组密码和序列密码。分组密码是在明文分组和密文分组上进行运算，序列密码是对明文和密文数据流按位或字节进行运算。

常见的对称加密算法包括瑞士的国际数据加密算法（International Data Encryption Algorithm, IDEA）、美国的数据加密标准（Data Encryption Standard, DES）和RC-5。

DES是一种迭代的分组密码，明文和密文都是64位，使用一个56位的密钥以及附加的8位奇偶校验位。攻击DES的主要技术是穷举法，由于DES的密钥长度较短，为了提高安全性，就出现了使用112位密钥对数据进行三次加密的算法（3DES），即用两个56位的密钥K1和K2，发送方用K1加密，K2解密，再使用K1加密；接收方则使用K1解密，K2加密，再使用K1解密，其效果相当于将密钥长度加倍。

IDEA是在DES的基础上发展起来的，类似于3DES。IDEA的明文和密文都是64位，密钥长度为128位。

RC-5是一种使用可变参数的分组迭代密码体制，其中可变参数为分组长（为2倍字长w位），密钥长（按字节数计b）和迭代轮数（以RC-5-w/r/b）。它面向字结构，便于软件和硬件的实现，适用于不同字长的微处理器。通过字长、密钥长和迭代轮数三个参数的配合，可以在安全性和速度上进行灵活的选择。RC-5加密效率高，适合于加密大量的数据。

RC-5由R.Rivest设计，是RSA实验室的一个产品。RC-5还引入了一种新的密码基本变换数据相依旋转（Data-Dependent Rotations）方法，即一个中间的字是另一个中间的低位所决定的循环移位结果，以提高密码强度，这也是RC-5的新颖之处。

2. 非对称加密算法

非对称加密算法也称为公钥加密算法，是指加密密钥和解密密钥完全不同，其中一个为公钥，另一个为私钥，并且不可能从任何一个推导出另一个。它的优点在于可以适应开放性的使用环境，可以实现数字签名与验证。

最常见的非对称加密算法是RSA，该算法的名字以发明者的名字命名：Ron Rivest, Adi Shamir 和 Leonard Adleman。RSA算法的密钥长度为512位。RSA算法的保密性取决于数学上将一个大数分解为两个素数的问题的难度，根据已有的数学方法，其计算量极大，破解很难。但是加密/解密时要进行大指数模运算，因此加密/解密速度很慢，主要用在数字签名中。

在本题中，RC-5是对称加密算法，RSA和ECC属于非对称加密算法，MD5为消息摘要算法。

试题答案

(65) A

第 5 章：系统安全性与保密性设计

试题3 (2010年11月试题64)

试题3 (2010年11月试题64)

所谓网络安全漏洞是指 (64) 。

- (64) A. 用户的误操作引起的系统故障
- B. 网络节点的系统软件或应用软件在逻辑设计上的缺陷
- C. 网络硬件性能下降产生的缺陷
- D. 网络协议运行中出现的错误

试题分析

网络安全漏洞通常是指网络节点的系统软件或应用软件在逻辑上的缺陷。

漏洞是在硬件、软件、协议的具体实现或系统安全策略上存在的缺陷，从而可以使攻击者能够在未授权的情况下访问或破坏系统。是受限制的计算机、组件、应用程序或其他联机资源的无意中留下的不受保护的入口点。例如，在Intel Pentium芯片中存在的逻辑错误，在Sendmail早期版本中的编程错误，在NFS协议中认证方式上的弱点，在UNIX系统管理员设置匿名FTP服务时配置不当的问题都可能被攻击者使用，威胁到系统的安全。这些都可以认为是系统中存在的安全漏洞。

漏洞会影响到很大范围的软硬件设备，包括作系统本身及其支撑软件，网络客户和服务器软件，网络路由器和安全防火墙等。换而言之，在这些不同的软硬件设备中都可能存在不同的安全漏洞问题。在不同种类的软、硬件设备，同种设备的不同版本之间，由不同设备构成的不同系统之间，以及同种系统在不同的设置条件下，都会存在各自不同的安全漏洞问题。

漏洞问题是与时间紧密相关的。一个系统从发布的那一天起，随着用户的深入使用，系统中存在的漏洞会被不断暴露出来，这些早先被发现的漏洞也会不断被系统供应商发布的补丁软件修补，或在以后发布的新版系统中得以纠正。而在新版系统纠正了旧版本中具有漏洞的同时，也会引入一些新的漏洞和错误。因而随着时间的推移，旧的漏洞会不断消失，新的漏洞会不断出现。漏洞问题也会长期存在。

因此，脱离具体的时间和具体的系统环境来讨论漏洞问题是毫无意义的。只能针对目标系统的操作系统版本、其上运行的软件版本以及服务运行设置等实际环境来具体谈论其中可能存在的漏洞及其可行的解决办法。同时应该看到，对漏洞问题的研究必须要跟踪当前最新的计算机系统及其安全问题的最新发展动态。这一点如同对计算机病毒发展问题的研究相似。如果在工作中不能保持对新技术的跟踪，就没有谈论系统安全漏洞问题的发言权，即使是以前所作的工作也会逐渐失去价值。

目前，计算机系统安全的分级标准一般都是依据"橘皮书"中的定义。橘皮书正式名称是"受信任计算机系统评量基准" (Trusted Computer System Evaluation Criteria)。橘皮书中对可信任系统的定义是这样的：一个由完整的硬件及软件所组成的系统，在不违反访问权限的情况下，它能同时服务于不限定个数的用户，并处理从一般机密到最高机密等不同范围的信息。橘皮书将一个计算机系统可接受的信任程度加以分级，凡符合某些安全条件、基准规则的系统即可归类为某种安全等级。橘皮书将计算机系统的安全性能由高而低划分为A、B、C、D四大等级。其中：

(1) D级：最低保护 (Minimal Protection)，凡没有通过其他安全等级测试项目的系统即属于该级，如DOS,Windows个人计算机系统。

(2) C级：自主访问控制 (Discretionary Protection)，该等级的安全特点在于系统的客体 (如文件、目录) 可由该系统主体 (如系统管理员、用户、应用程序) 自主定义访问权。例如：管理员可以决定系统中任意文件的权限。当前Unix、Linux、Windows NT等作系统都为此安全等级。

(3) B级：强制访问控制 (Mandatory Protection)，该等级的安全特点在于由系统强制对客体进行安全保护，在该级安全系统中，每个系统客体 (如文件、目录等资源) 及主体 (如系统管理员、用户、应用程序) 都有自己的安全标签 (Security Label)，系统依据用户的安全等级赋予其对各个对象的访问权限。

(4) A级：可验证访问控制 (Verified Protection)，而其特点在于该等级的系统拥有正式的分析及数学式方法可完全证明该系统的安全策略及安全规格的完整性与一致性。

可见，根据定义，系统的安全级别越高，理论上该系统也越安全。可以说，系统安全级别是一种理论上的安全保证机制。是指在正常情况下，在某个系统根据理论得以正确实现时，系统应该可以达到的安全程度。试题答案

答案：(64) B

第5章：系统安全性与保密性设计

试题4 (2010年11月试题65)

试题4 (2010年11月试题65)

ARP攻击造成网络无法跨网段通信的原因是 (65) 。

- (65) A.发送大量ARP报文造成网络拥塞
- B.伪造网关ARP报文使得数据包无法发送到网关
- C.ARPI攻击破坏了网络的物理连通性
- D.ARPI攻击破坏了网关设备

试题分析

ARP (Address Resolution Protocol, 地址解析协议) 是一个位于 TCP/IP 协议栈中的底层协议，对应于数据链路层，负责将某个 IP 地址解析成对应的 MAC 地址。ARP 协议的基本功能就是通过目标设备的 IP 地址，查询目标设备的 MAC 地址，以保证通信的进行。ARP 攻击就是通过伪造 IP 地址和 MAC 地址实现 ARP 欺骗，能够在网络中产生大量的 ARP 通信量使网络阻塞，攻击者只要持续不断的发出伪造的 ARP 响应包就能更改目标主机 ARP 缓存中的 IP-MAC 条目，造成网络中断或中间人攻击。

ARP 攻击主要是存在于局域网网络中，局域网中若有一台计算机感染 ARP 病毒，则感染该 ARP 病毒的系统将会试图通过“ARP 欺骗”手段截获所在网络内其它计算机的通信信息，并因此造成网内其它计算机的通信故障。

某机器 A 要向主机 B 发送报文，会查询本地的 ARP 缓存表，找到 B 的 IP 地址对应的 MAC 地址后，就会进行数据传输。如果未找到，则 A 广播一个 ARP 请求报文（携带主机 A 的 IP 地址 Ia-- 物理地址 Pa ），请求 IP 地址为 Ib 的主机 B 回答物理地址 Pb. 网上所有主机（包括 B ）都收到 ARP 请求，但只有主机 B 识别自己的 IP 地址，于是向 A 主机发回一个 ARP 响应报文。其中就包含有 B 的 MAC 地址，A 接收到 B 的应答后，就会更新本地的 ARP 缓存。接着使用这个 MAC 地址发送数据（由网卡附加 MAC 地址）。因此，本地高速缓存的这个 ARP 表是本地网络流通的基础，而且这个缓存是动态的。

ARP 欺骗的中毒现象表现为：使用局域网时会突然掉线，过一段时间后又会恢复正常。例如，客户端状态频频变红，用户频繁断网，IE 浏览器频繁出错，以及一些常用软件出现故障等。如果局域网中是通过身份认证上网的，会突然出现可认证，但不能上网的现象（无法 ping 通网关），重启机器或在 MS-DOS 窗口下运行命令 arp -d 后，又可恢复上网。

ARP 病毒只需成功感染一台电脑，就可能导致整个局域网都无法上网，严重的甚至可能带来整个网络的瘫痪。该病毒发作时除了会导致同一局域网内的其他用户上网出现时断时续的现象外，还会窃取用户密码。例如，盗取 QQ 密码、盗取各种网络游戏密码和账号去做金钱交易，盗窃网上银行账号来做非法交易活动等，这是病毒或木马的惯用伎俩，给用户造成了很大的不便和巨大的经济损失。

试题答案

(65) B

第 5 章：系统安全性与保密性设计

试题 5 (2011 年 11 月试题 64)

试题 5 (2011 年 11 月试题 64)

在网络管理中要防止各种安全威胁。在 SNMP v3 中，无法预防的安全威胁是 (64) 。

- (64) A. 篡改管理信息：通过改变传输中的 SNMP 报文实施未经授权的管理操作
- B. 通信分析：第三者分析管理实体之间的通信规律，从而获取管理信息
- C. 假冒合法用户：未经授权的用户冒充授权用户，企图实施管理操作
- D. 消息泄露：SNMP 引擎之间交换的信息被第三者偷听

试题分析

SNMP v3把对网络协议的安全威胁分为主要的和次要的两类。标准规定安全模块必须提供防护的两种主要威胁如下。

① 修改信息：就是某些未经授权的实体改变了进来的SNMP报文，企图实施未经授权的管理操作，或者提供虚假的管理对象。

② 假冒：即未经授权的用户冒充授权用户的标识，企图实施管理操作。

必须提供防护的两种次要威胁如下。

① 修改报文流：由于SNMP协议通常是基于无连接的传输服务，重新排序报文流、延迟或重放报文的威胁都可能出现。这种威胁的危害性在于通过报文流的修改可能实施非法的管理操作。

② 消息泄露：SNMP引擎之间交换的信息可能被偷听，对于这种威胁的防护应采取局部的策略。

不必提供防护的威胁包括如下。

(1) 拒绝服务：因为在很多情况下拒绝服务和网络失效是无法区别，所以可以由网络管理协议来处理，安全子系统不必采取措施。

(2) 通信分析：即由第三者分析管理实体之间的通信规律，从而获取需要的信息。由于通常都是由少数管理站来管理整个网络的，所以管理系统的通信模式是可预见的，防护通信分析就没有多大作用了。

试题答案

(64) B

第5章：系统安全性与保密性设计

试题6 (2011年11月试题65)

试题6 (2011年11月试题65)

下面安全协议中，用来实现安全电子邮件的协议是 (65)。

(65) A. IPSec B. L2TP C. PGP D. PPTP

试题分析

IPSec是一个工业标准网络安全协议，为IP网络通信提供透明的安全服务，保护TCP/IP通信免遭窃听和篡改，可以有效抵御网络攻击，同时保持易用性。IPSec有两个基本目标，分别是保护IP数据包安全和为抵御网络攻击提供防护措施。IPSec结合密码保护服务、安全协议组和动态密钥管理三者来实现上述两个目标。

IPSec是针对IPv4和IPv6的，其主要特征是可以支持IP级所有流量的加密和/或认证，增强所有分布式应用的安全性。IPSec在IP层提供安全服务，使得系统可以选择所需要的安全协议，确定该服务所用的算法，并提供安全服务所需任何加密密钥。

IPSec基于一种端对端的安全模式。这种模式有一个基本前提假设，就是假定数据通信的传输媒介是不安全的，因此，通信数据必须经过加密，而掌握加解密方法的只有数据的发送端和接收端，两者各自负责相应的数据加解密处理，而网络中其他只负责转发数据的路由器或计算机无须支持IPSec。使用IPSec可以显著地减少或防范以下几种网络攻击：

(1) Sniffer.Sniffer可以读取数据包中的任何信息，对抗Sniffer最有效的方法就是对数据进行加密。IPSec的封装安全有效负载（Encapsulating Security Payload,ESP）协议通过对IP包进行加密来保证数据的私密性。

(2) 数据篡改。IPSec用密钥为每个IP包生成一个数字检查和，该密钥为且仅为数据的发送方和接收方共享。对数据包的任何篡改，都会改变检查和，从而可以让接收方得知包在传输过程中遭到了修改。

(3) 身份欺骗，盗用口令，应用层攻击。IPSec的身份交换和认证机制不会暴露任何信息，不给攻击者有可趁之机，双向认证在通信双方之间建立信任关系，只有可信赖的系统才能彼此通信。

(4) 中间人攻击。IPSec结合双向认证和共享密钥，足以抵御中间人攻击。

(5) 拒绝服务攻击。IPSec使用IP包过滤法，依据IP地址范围和协议，甚至特定的协议端口号来决定哪些数据流需要受到保护，哪些数据流可以被允许通过，哪些需要拦截。

PPTP (Point to Point Tunneling Protocol,点对点隧道协议) 是一种支持多协议虚拟专用网络的网络技术，它工作在第二层（数据链路层）。通过该协议，远程用户能够通过装有点对点协议的系统安全访问公司网络，并能拨号连入本地 ISP，通过Internet安全链接到公司网络。PPTP协议假定在PPTP客户机和PPTP服务器之间有连通并且可用的IP网络。因此如果PPTP客户机本身已经是IP网络的组成部分，那么即可通过该IP网络与PPTP服务器取得连接；而如果PPTP客户机尚未连入网络（例如，在Internet拨号用户的情形下），PPTP客户机必须首先拨打NAS以建立IP连接。这里所说的PPTP客户机也就是使用PPTP协议的VPN客户机，而PPTP服务器亦即使用PPTP协议的VPN服务器。

PPTP 只能通过PAC和PNS来实施，其它系统没有必要知道 PPTP.拨号网络可与 PAC 相连接而无需知道 PPTP.标准的 PPP 客户机软件可继续在隧道 PPP 链接上操作。

PPTP 使用GRE的扩展版本来传输用户PPP包。这些增强允许为在PAC和PNS之间传输用户数据的隧道提供低层拥塞控制和流控制。这种机制允许高效使用隧道可用带宽并且避免了不必要的重发和缓冲区溢出。PPTP 没有规定特定的算法用于底层控制，但它确实定义了一些通信参数来支持这样的算法工作。PPTP控制连接数据包包括一个IP报头，一个TCP报头和PPTP控制信息。

L2TP是一种工业标准的Internet隧道协议，功能大致和PPTP协议类似，比如同样可以对网络数据流进行加密。不过也有不同之处，比如PPTP要求网络为IP网络，L2TP要求面向数据包的点对点连接；PPTP使用单一隧道，L2TP使用多隧道；L2TP提供包头压缩、隧道验证，而PPTP不支持。

L2TP 扩展了 PPP 模型，允许第二层和 PPP 终点处于不同的由包交换网络相互连接的设备来。通过 L2TP,用户在第二层连接到一个访问集中器（如：调制解调器池、ADSL DSLAM 等），然后这个集中器将单独得的 PPP 帧隧道到 NAS.这样，可以把 PPP 包的实际处理过程与 L2 连接的终点分离开来。对于这样的分离，其明显的一个好处是，L2 连接可以在一个（本地）电路集中器上终止，然后通过共享网络如帧中继电路或英特网扩展逻辑 PPP 会话，而不用在 NAS 上终止。从用户角度看，直接在 NAS 上终止 L2 连接与使用 L2TP 没有什么功能上的区别。L2TP 协议也用来解决"多连接联选组分离"问题。多连接 PPP,一般用来集中 ISDN B 通道，需要构成多连接捆绑的所有通道在一个单网络访问服务器（NAS）上组合。因为 L2TP 使得 PPP 会话可以出现在接收会话的物理点之外的位置，它用来使所有的通道出现在单个的 NAS 上，并允许多链接操作，即使是在物理呼叫分散在不同物理位置的 NAS 上的情况下。

PGP (Pretty Good Privacy) 是一个基于RSA的邮件加密软件，可以用它对邮件保密以防止非授权者阅读，它还能对邮件加上数字签名，从而使收信人可以确信邮件发送者。PGP的基本原理是，先用对称密钥加密传送的信息，再将该对称加密密钥以接收方的公钥加密，组成数字信封，并

将此密钥交给公正的第三方保管；然后，将此数字信封传送给接收方。接收方必须先以自己的私钥将数字信封拆封，以获得对称解密密钥，再以该对称解密密钥解出真正的信息，兼顾方便与效率。

PGP还可用于文件存储的加密。PGP承认两种不同的证书格式，分别是PGP证书和X.509证书。其中，一份PGP证书包括版本号、证书持有者的公钥、证书持有者的信息（例如，姓名、用户ID、照片等）、证书拥有者的数字签名、证书的有效期和密钥首选的对称加密算法等内容。

试题答案

(65) C

第 5 章：系统安全性与保密性设计

试题7 (2012年11月试题64~65)

试题7 (2012年11月试题64~65)

下图所示PKI系统结构中，负责生成和签署数字证书的是（64），负责验证用户身份的是（65）。

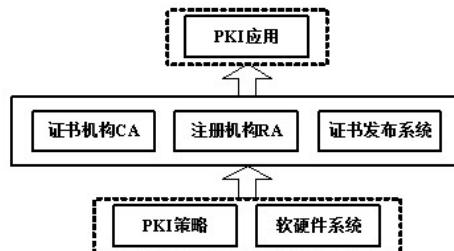


图5-1 PKI系统结构

(64) A.证书机构CA B.注册机构RA C.证书发布系统 D.PKI策略

(65) A.证书机构CA B.注册机构RA C.证书发布系统 D.PKI策略

试题分析

公钥基础设施（Public Key Infrastructure,PKI）是一个采用公钥概念和技术来提供安全服务的具有普适性的安全基础设施，是目前网络安全建设的基础与核心。PKI由公开密钥密码技术、数字证书、证书发放机构（CA）和关于公开密钥的安全策略等基本成分共同组成的。

PKI采用证书进行公钥管理，通过第三方的可信任机构（认证中心，即CA），把用户的公钥和用户的其他标识信息捆绑在一起，其中包括用户名和电子邮件地址等信息，以在Internet网上验证用户的身份。PKI把公钥密码和对称密码结合起来，在Internet网上实现密钥的自动管理，保证网上数据的安全传输。

因此，从大的方面来说，所有提供公钥加密和数字签名服务的系统，都可归结为PKI系统的一部分，PKI的主要目的是通过自动管理密钥和证书，为用户建立起一个安全的网络运行环境，使用户可以在多种应用环境下方便的使用加密和数字签名技术，从而保证网上数据的机密性、完整性、有效性。数据的机密性是指数据在传输过程中，不能被非授权者偷看；数据的完整性是指数据在传输过程中不能被非法篡改；数据的有效性是指数据不能被否认。

一个有效的PKI系统必须是安全的和透明的，用户在获得加密和数字签名服务时，不需要详细地了解PKI的内部运作机制。在一个典型、完整和有效的PKI系统中，除证书的创建和发布，特别是证书的撤销，一个可用的PKI产品还必须提供相应的密钥管理服务，包括密钥的备份、恢复和更新等。没有一个好的密钥管理系统，将极大影响一个PKI系统的规模、可伸缩性和在协同网络中的运行成本。在一个企业中，PKI系统必须有能力为一个用户管理多对密钥和证书；能够提供安全策略编辑和管理工具，如密钥周期和密钥用途等。

PKI作为一组在分布式计算系统中利用公钥技术和X.509证书所提供的安全服务，企业或组织可利用相关产品建立安全域，并在其中发布密钥和证书。在安全域内，PKI管理加密密钥和证书的发布，并提供诸如密钥管理（包括密钥更新，密钥恢复和密钥委托等）、证书管理（包括证书产生和撤销等）和策略管理等。PKI产品也允许一个组织通过证书级别或直接交叉认证等方式来同其他安全域建立信任关系。这些服务和信任关系不能局限于独立的网络之内，而应建立在网络之间和Internet之上，为电子商务和网络通信提供安全保障，所以具有互操作性的结构化和标准化技术成为PKI的核心。

PKI在实际应用上是一套软硬件系统和安全策略的集合，它提供了一整套安全机制，使用户在不知道对方身份或分布地很广的情况下，以证书为基础，通过一系列的信任关系进行通讯和电子商务交易。

一个典型的PKI系统包括PKI策略及软硬件系统、证书机构CA、注册机构RA、证书发布系统和PKI应用等。

（1）PKI策略及软硬件系统

PKI安全策略建立和定义了一个组织信息安全方面的指导方针，同时也定义了密码系统使用的处理方法和原则。它包括一个组织怎样处理密钥和有价值的信息，根据风险的级别定义安全控制的级别。一般情况下，PKI中有两种类型的策略：一是证书策略，用于管理证书的使用，例如，可以确认某一CA是在Internet上的公有CA，还是某一企业内部的私有CA；另外一种就是CPS（Certificate Practice Statement）。一些由商业证书发放机构（CCA）或者可信的第三方操作的PKI系统需要CPS，它实际上是一些操作过程的详细文档，描述了如何在实践中增强和支持安全策略，包括CA是如何建立和运作的，证书是如何发行、接收和废除的，密钥是如何产生、注册的，以及密钥是如何存储的，用户是如何得到它的等。

（2）证书机构

证书机构CA是PKI的信任基础，它管理公钥的整个生命周期，其作用包括：发放证书、规定证书的有效期和通过发布证书废除列表（CRL）确保必要时可以废除证书。

（3）注册机构

注册机构RA提供用户和CA之间的一个接口，它获取并认证用户的身份，向CA提出证书请求。它主要完成收集用户信息和确认用户身份的功能。这里指的用户，是指将要向认证中心（即CA）申请数字证书的客户，可以是个人，也可以是集团或团体、某政府机构等。注册管理一般由一个独立的注册机构（即RA）来承担。它接受用户的注册申请，审查用户的申请资格，并决定是否同意CA给其签发数字证书。注册机构并不给用户签发证书，而只是对用户进行资格审查。因此，RA可以设置在直接面对客户的业务部门，如银行的营业部、机构认识部门等。当然，对于一个规模较小的PKI应用系统来说，可把注册管理的职能由认证中心CA来完成，而不设立独立运行的RA。但这并不是取消了PKI的注册功能，而只是将其作为CA的一项功能而已。PKI国际标准推荐由一个独立的RA来完成注册管理的任务，可以增强应用系统的安全。

（4）证书发布系统

证书发布系统负责证书的发放，如可以通过用户自己，或是通过目录服务。目录服务器可以是一个组织中现存的，也可以是PKI方案中提供的。

试题答案

(64) A (65) B

第 6 章：信息化基础

考点分析

第6章信息化基础

6.1 考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

(1) 信息系统工程总体规划

总体规划目标、范围
总体规划的方法论
信息系统的组成
信息系统的实现

(2) 政府信息化与电子政务

电子政务的概念、内容和技术形式
中国政府信息化的策略和历程
电子政务建设的过程模式和技术模式

(3) 企业信息化与电子商务

企业信息化的概念、目的、规划、方法
ERP的主要模块和主要算法
企业业务流程重组 (BPR)
CRM、PDM在企业的应用
知识管理
企业应用集成
全程供应链管理的思想
商业智能
电子商务的类型、标准

(4) 信息资源管理

(5) 国际和国内有关信息化的标准、法律和规定

从历年考试情况来看，本章的考点主要集中在以下方面：企业应用集成的方式、商业智能、企业门户以及信息化相关的一些概念。具体考查知识点分布情况如表6-1所示。

表6-1 历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题18~20	商业智能、企业应用集成、EDI
2010年11月试题18~21	企业门户、CRM、企业应用集成
2011年11月试题18~21, 37~38	数据集成、企业应用集成、企业信息化建设、企业信息化方法
2012年11月试题18~24	信息化“三流”、CRM、企业信息资源集成管理、企业应用集成

第6章：信息化基础

试题1 (2009年11月试题18)

6.2 试题精解

试题1 (2009年11月试题18)

商业智能是指利用数据挖掘、知识发现等技术分析和挖掘结构化的、面向特定领域的存储与数据仓库的信息。它可以帮助用户认清发展趋势、获取决策支持并得出结论。以下(18)活动，并不属于商业智能范畴。

- (18) A.某大型企业通过对产品销售数据进行挖掘，分析客户购买偏好
- B.某大型企业查询数据仓库中某种产品的总体销售数量
- C.某大型购物网站通过分析用户的购买历史记录，为客户进行商品推荐
- D.某银行通过分析大量股票交易的历史数据，做出投资决策

试题分析

一般现代化的业务操作通常都会产生大量的数据，例如订单、库存、交易帐目、通话记录和客户资料等。如何利用这些数据增进对业务情况的了解，帮助人们在业务管理及发展上作出及时、正确的判断，也就是说，怎样从业务数据中提取有用的信息，然后根据这些信息来采用明智的行动，这就是商业智能 (Business Intelligence, BI) 的功能。

目前，商业智能产品及解决方案大致可分为数据仓库产品、数据抽取产品、OLAP产品、展示产品，以及集成这几种产品的针对某个应用的整体解决方案等。

1. BI的技术应用

BI系统主要包括数据预处理、建立数据仓库、数据分析和数据展现四个主要阶段。

数据预处理是整合企业原始数据的第一步，它包括数据的抽取 (Extraction) 、转换

(Transformation) 和加载 (Load) 三个过程 (ETL过程) ；建立数据仓库则是处理海量数据的基础；数据分析是体现系统智能的关键，一般采用OLAP和数据挖掘两大技术。OLAP不仅进行数据汇总/聚集，同时还提供切片、切块、下钻、上卷和旋转等数据分析功能，用户可以方便地对海量数据进行多维分析。数据挖掘的目标则是挖掘数据背后隐藏的知识，通过关联分析、聚类和分类等方法建立分析模型，预测企业未来发展趋势和将要面临的问题；在海量数据和分析手段增多的情况下，数据展现则主要保障系统分析结果的可视化。

一般认为，数据仓库、OLAP和数据挖掘技术是BI的三大组成部分。

2. BI的实施步骤

实施BI系统是一项复杂的系统工程，整个项目涉及企业管理、运营管理、信息系统、数据仓库、数据挖掘和统计分析等众多门类的知识。因此，用户除了要选择合适的BI软件工具外，还必须按照正确的实施方法才能保证项目得以成功。BI项目的实施步骤大致如下：

(1) 需求分析。需求分析是BI实施的第一步，在其他活动开展之前，必须明确地定义企业对BI的期望和需求，包括需要分析的主题、各主题可能查看的维度、需要发现企业哪些方面的规律等。

(2) 数据仓库建模。通过对企业需求的分析，建立企业数据仓库的逻辑模型和物理模型，并规划好系统的应用架构，将企业各类数据按照分析主题进行组织和归类。

(3) 数据抽取。数据仓库建立后，必须将数据从业务系统中抽取到数据仓库中，在抽取的过程中还必须将数据进行转换和清洗，以适应分析的需要。

(4) 建立BI分析报表。BI分析报表需要专业人员按照用户制订的格式进行开发，用户也可自行开发。

(5) 用户培训和数据模拟测试。对用户进行培训，在实际环境中对BI系统进行测试，以便发现和修改问题。

(6) 系统改进和完善。在用户使用一段时间后，可能会提出更多的、更具体的需求，这时，就需要按照上述步骤对BI系统进行重构或完善。

BI主要体现在“智能”上，即通过对大量数据的分析，得到趋势变化等重要知识，并为决策提供支持。选项A、C、D都是对数据进行分析，获得知识的过程；选项B仅仅是获取数据，并没有对数据进行分析，因此不属于BI范畴。

试题答案

(18) B

第6章：信息化基础

试题2（2009年11月试题19）

试题2（2009年11月试题19）

企业应用集成通过采用多种集成模式构建统一标准的基础平台，将具有不同功能和目的且独立运行的企业信息系统联合起来。其中，面向（19）的集成模式强调处理不同应用系统之间的交互逻辑，与核心业务逻辑相分离，并通过不同应用系统之间的协作共同完成某项业务功能。

(19) A.数据 B.接口 C.过程 D.界面

试题分析

最初的企业应用集成（Enterprise Application Integration, EAI）EAI仅指企业内部不同应用系统之间的互连，以期通过应用集成实现数据在多个系统之间的同步与共享。伴随着EAI技术的不断发展，它所被赋予的内涵变得越来越丰富。现在的EAI具有更为广义的内涵，它已经被扩展到业务集成的范畴。

对于要实施EAI的企业而言，EAI是分层次的，但对于如何划分和规范EAI层次的定义，业界并没有一个统一的标准。针对不同的企业，EAI的内容和层次可能就会存在一定的差异；对于各EAI厂商，基于EAI理解的侧重点不同，也可以说出不同的答案。当前，从最普遍的意义上来说，EAI可以包括表示集成、数据集成、控制集成和业务流程集成等多个层次和方面。

1. 表示集成

表示集成也称为界面集成，这是比较原始和最浅层次的集成，但又是常用的集成。这种方法把用户界面作为公共的集成点，把原有零散的系统界面集中在一个新的界面中。其模型如图6-1所示。

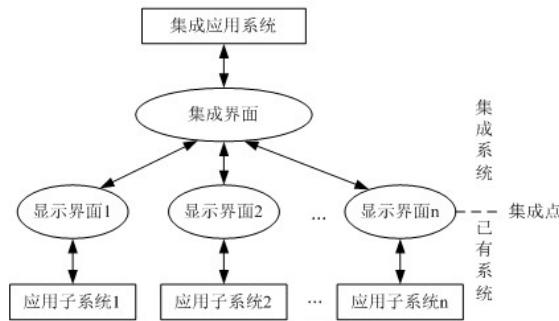


图6-1 表示集成示意图

表示集成是黑盒集成，无需了解程序与数据库的内部构造。常用的集成技术主要有屏幕截取和输入模拟技术。表示集成通常应用于以下几种情况：

- (1) 在现有的基于终端的应用系统上配置基于PC的用户界面。
 - (2) 为用户提供一个看上去统一,但是由多个系统组成的应用系统。
 - (3) 当只可能在显示界面上实现集成时。

从图6-1中可以看出，表示集成的实现是很简单的，也是很不彻底的，只是做了一层“外装修”，而额外多出来的集成界面也将可能成为系统的性能瓶颈。

2. 数据集成

为了完成控制集成和业务流程集成，必须首先解决数据和数据库的集成问题。在集成之前，必须首先对数据进行标识并编成目录，另外还要确定元数据模型，保证数据在数据库系统中分布和共享。因此，数据集成是白盒集成，其模型如图6-2所示。

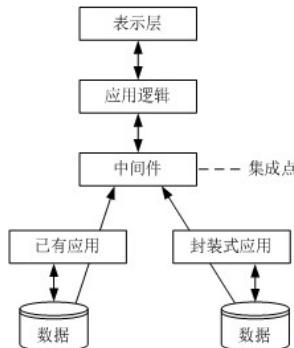


图6-2 数据集成示意图

有很多不同的中间件工具可以用于数据集成。例如，批量文件传输，即以特定的或是预定的方式在原有系统和新开发的应用系统之间进行文件传输；用于访问不同类型数据库系统的ODBC标准接口；向分布式数据库提供连接的数据库访问中间件技术等。

通常在以下情况下，将会使用数据集成：

- (1) 需要对多种信息源产生的数据进行综合分析和决策。
 - (2) 要处理一些多个应用共享需要访问的公用信息库。
 - (3) 当需要从某数据源获得数据来更新另一个数据源时，特别是它们之间的数据格式不相同

相对而言，数据集成比表示集成要更加灵活。但是，当业务逻辑经常发生变化时，数据集成就会面临困难。

3 控制集成

控制集成也称为功能集成或应用集成，是在业务逻辑层上对应用系统进行集成的。控制集成的集成点存于程序代码中，集成处可能只需简单使用公开的API就可以访问，当然也可能需要添加附加的代码来实现。控制集成是黑盒集成，其模型如图6-3所示。

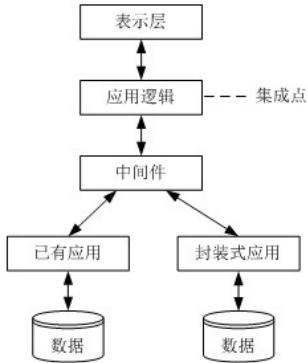


图6-3 控制集成示意图

实现控制集成时，可以借助于远程过程调用或远程方法调用、面向消息的中间件、分布式对象技术和事务处理监控器来实现。控制集成与表示集成、数据集成相比，灵活性更高。表示集成和数据集成适用的环境下，都适用于控制集成。但是，由于控制集成是在业务逻辑层进行的，其复杂度更高一些。而且，很多系统的业务逻辑部分并没有提供API，这样，集成难度就会更大。

4. 业务流程集成

业务流程集成也称为过程集成，这种集成超越了数据和系统，它由一系列基于标准的、统一数据格式的工作流组成。当进行业务流程集成时，企业必须对各种业务信息的交换进行定义、授权和管理，以便改进操作、减少成本、提高响应速度。

业务流程集成不仅要提供底层应用支撑系统之间的互连，同时要实现存在于企业内部的应用之间，本企业和其他合作伙伴之间的端到端的业务流程的管理，它包括应用集成、B2B集成、自动化业务流程管理、人工流程管理、企业门户，以及对所有应用系统和流程的管理和监控等。

目前，市场主流的集成模式有三种，分别是面向信息的集成技术、面向过程的集成技术和面向服务的集成技术。

在数据集成的层面上，信息集成技术仍然是必选的方法。信息集成采用的主要数据处理技术有数据复制、数据聚合和接口集成等。其中，接口集成仍然是一种主流技术。它通过一种集成代理的方式实现集成，即为应用系统创建适配器作为自己的代理，适配器通过其开放或私有接口将信息从应用系统中提取出来，并通过开放接口与外界系统实现信息交互，而假如适配器的结构支持一定的标准，则将极大地简化集成的复杂度，并有助于标准化，这也是面向接口集成方法的主要优势来源。标准化的适配器技术可以使企业从第三方供应商获取适配器，从而使集成技术简单化。

面向过程的集成技术其实是一种过程流集成的思想，它不需要处理用户界面开发、数据库逻辑、事务逻辑等，而只是处理系统之间的过程逻辑，和核心业务逻辑相分离。在结构上，面向过程的集成方法在面向接口的集成方案之上，定义了另外的过程逻辑层；而在该结构的底层，应用服务器、消息中间件提供了支持数据传输和跨过程协调的基础服务。对于提供集成代理、消息中间件以及应用服务器的厂商来说，提供用于业务过程集成是对其产品的重要拓展，也是目前应用集成市场的重要需求。

基于SOA (Service Oriented Architecture,面向服务架构) 和Web Service (Web服务) 技术的应用集成是业务集成技术上的一次重要的变化，被认为是新一代的应用集成技术。集成的对象是一个个的Web服务或者是封装成Web服务的业务处理。Web服务技术由于是基于最广为接受的、开放的技术标准，支持服务接口描述和服务处理的分离、服务描述的集中化存储和发布、服务的自动

查找和动态绑定以及服务的组合，成为新一代面向服务的应用系统的构建和应用系统集成的基础设施。

试题答案

(19) C

第 6 章：信息化基础

试题3 (2009年11月试题20)

试题3 (2009年11月试题20)

电子数据交换 (EDI) 是电子商务活动中采用的一种重要的技术手段。以下关于EDI的叙述中，错误的是 (20) 。

- (20) A.EDI的实施需要一个公认的标准和协议，将商务活动中涉及的文件标准化和格式化
- B.EDI的实施在技术上比较成熟，成本也较低
- C.EDI通过计算机网络，在贸易伙伴之间进行数据交换和自动处理
- D.EDI主要应用于企业与企业、企业与批发商之间的批发业务

试题分析

简单地说，EDI (Electronic Data Interchange, 电子数据交换) 就是按照商定的协议，将商业文件标准化和格式化，并通过计算机网络，在贸易伙伴的计算机网络系统之间进行数据交换和自动处理。

EDI使用电子方法传递信息和处理数据的。EDI一方面用电子传输的方式取代了以往纸单证的邮寄和递送，从而提高了传输效率，另一方面通过计算机处理数据取代人工处理数据，从而减少了差错和延误。

EDI是采用统一标准编制数据信息的。这是EDI与电传、传真等其它传递方式的重要区别，电传、传真等并没有统一格式标准，而EDI必须有统一的标准方能运作。

EDI是计算机应用程序之间的连接。一般的电子通信手段是人与人之间的信息传递，传输的内容即使不完整、格式即使不规范，也能被人所理解。这些通信手段仅仅是人与人之间的信息传递工具，不能处理和返回信息。EDI实现的是计算机应用程序与计算机应用程序之间的信息传递与交换。由于计算机只能按照给定的程序识别和接受信息，所以电子单证必须符合标准格式并且内容完整准确。在电子单证符合标准且内容完整的情况下，EDI系统不但能识别、接受、存储信息，还能对单证数据信息进行处理，自动制作新的电子单据并传输到有关部门。在有关部门就自己发出的电子单证进行查询时，计算机还可以反馈有关信息的处理结果和进展状况。在收到一些重要电子邮件时，计算机还可以按程序自动产生电子收据并传回对方。

EDI系统采用加密防伪手段。EDI系统有相应的保密措施，EDI传输信息的保密通常是采用密码系统，各用户掌握自己的密码，可打开自己的"邮箱"取出信息，外人却不能打开这个"邮箱"，有关部门和企业发给自己的电子信息均自动进入自己的"邮箱"。一些重要信息在传递时还要加密，即把信息转换成他人无法识别的代码，接收方计算机按特定程序译码后还原成可识别信息。为防止有些信息在传递过程中被篡改，或防止有人传递假信息，还可以使用证实手段，即将普通信息与转变成代码的

信息同时传递给接收方，接收方把代码翻译成普通信息进行比较，如二者完全一致，可知信息未被篡改，也不是伪造的信息。

目前，EDI的实施在技术上比较成熟，但是实施EDI需要统一数据格式，成本与代价较大。

试题答案

(20) B

第 6 章：信息化基础

试题4 (2010年11月试题18)

试题4 (2010年11月试题18)

希赛公司欲开发一个门户系统，该系统以商业流程和企业应用为核心，将商业流程中不同的功能模块通过门户集成在一起，以提高公司的集中贸易能力、协同能力和信息管理能力。根据这种需求，采用企业 (18) 门户解决方案最为合适。

(18) A.信息 B.知识 C.应用 D.垂直

试题分析

按照实际应用领域，企业门户可以划分为四类，分别是企业网站、企业信息门户、企业知识门户和企业应用门户。

① 企业网站。随着互联网的兴起，企业纷纷建立自己的网站，供用户或企业员工浏览。这些网站往往功能简单，注重信息的单向传送，忽视用户与企业间、用户相互之间的信息互动。这些网站面向特定的使用人群，为企业服务，因此，可以被看作是EP发展的雏形。

② 企业信息门户。企业信息门户 (Enterprise Information Portal,EIP) 是指在Internet环境下，把各种应用系统、数据资源和互联网资源统一集成到EP之下，根据每个用户使用特点和角色的不同，形成个性化应用界面，并通过对事件和消息的处理传输把用户有机地联系在一起。EIP不仅局限于建立一个企业网站，提供一些企业和产品/服务的信息，更重要的是要求企业能实现多业务系统的集成，能对客户的各种要求做出快速响应，并且能对整个供应链进行统一管理。企业员工、合作伙伴、客户、供应商都可以通过EIP非常方便地获取自己所需的信息。对访问者来说，EIP提供了一个单一的访问入口，所有访问者都可以通过这个入口获得个性化的信息和服务，可以快速了解企业的相关信息；对企业来说，EIP既是一个展示企业的窗口，也可以无缝地集成企业的业务内容、商务活动、社区等，动态地发布存储在企业内部和外部的各种信息，同时还可以支持网上的虚拟社区，访问者可以相互讨论和交换信息。

③ 企业知识门户。企业知识门户 (Enterprise Knowledge Portal,EKP) 是企业员工日常工作所涉及相关主题内容的“总店”。企业员工可以通过EKP方便地了解当天的最新消息、工作内容、完成这些工作所需的知识等。通过EKP，任何员工都可以实时地与工作团队中的其他成员取得联系，寻找能够提供帮助的专家或者快速地连接到相关的门户。不难看出，EKP的使用对象是企业员工，它的建立和使用可以大大提高企业范围内的知识共享，并由此提高企业员工的工作效率。当然，EKP还应该具有信息搜集、整理、提炼的功能，可以对已有的知识进行分类，建立企业知识库并随时更新知识库的内容。目前，一些咨询和服务型企业已经开始建立企业知识门户。

④企业应用门户。企业应用门户 (Enterprise Application Portal,EAP) 实际上是对企业业务流程的集成。它以业务流程和企业应用为核心，把业务流程中功能不同的应用模块通过门户技术集成在一起。从某种意义上说，可以把EAP看成是企业信息系统的集成界面。企业员工和合作伙伴可以通过EAP访问相应的应用系统，实现移动办公、进行网上交易等。

试题答案

(18) C

第 6 章：信息化基础

试题5 (2010年11月试题19)

试题5 (2010年11月试题19)

客户关系管理 (CRM) 系统将市场营销的科学管理理念通过信息技术的手段集成在软件上，能够帮助企业构建良好的客户关系。以下关于CRM系统的叙述中，错误的是 (19)。

- (19) A. 销售自动化是CRM系统中最基本的模块
- B. 营销自动化作为销售自动化的补充，包括营销计划的编制和执行、计划结果分析等
- C. CRM系统能够与ERP系统在财务、制造、库存等环节进行连接，但两者关系相对松散，一般不会形成闭环结构
- D. 客户服务与支持是CRM系统的重要功能。目前，客户服务与支持的主要手段是通过呼叫中心和互联网来实现

试题分析

客户关系管理 (Customer Relationship Management,CRM) 是一种旨在改善企业与客户之间关系的新型管理机制。它通过提供更快速、更周到的优质服务来吸引或保持更多的客户。CRM集成了信息系统和办公系统等的一整套应用系统，从而确保了客户满意度的提高，以及通过对业务流程的全面管理来降低企业的成本。

CRM在坚持以客户为中心的理念的基础上，重构包括市场营销和客户服务等业务流程。CRM的目标不仅要使这些业务流程自动化，而且要确保前台应用系统能够改进客户满意度、增加客户忠诚度，以达到使企业获利的最终目标。

CRM实际上是一个概念，也是一种理念；同时，它又不仅是一个概念，也不仅是一种理念，它是企业参与市场竞争的新的管理模式，它是一种以客户为中心的业务模型，并由集成了前台和后台业务流程的一系列应用程序来支撑。这些整合的应用系统保证了更令人满意的客户体验，因而会使企业直接受益。

CRM的根本要求就是与客户建立起一种互相学习的关系，即从与客户的接触中了解他们在使用产品中遇到的问题，以及对产品的意见和建议，并帮助他们加以解决。在与客户互动的过程中，了解他们的姓名、通讯地址、个人喜好以及购买习惯，并在此基础上进行“一对一”的个性化服务，甚至拓展新的市场需求。例如，你在订票中心预订了机票之后，CRM就会根据了解的信息向你提供唤醒服务或是出租车登记等增值服务。因此，可以看到，CRM解决方案的核心思想就是通过跟客户的“接触”，搜集客户的意见、建议和要求，并通过数据挖掘和分析，提供完善的个性化服务。

一般说来CRM由两部分构成，即触发中心和挖掘中心，触发中心是指客户和CRM通过电话、传真、Web、E-mail等多种方式“触发”进行沟通；挖掘中心则是指CRM记录交流沟通的信息和进行智能分析。由此可见，一个有效的CRM解决方案应该具备以下要素：

(1) 畅通有效的客户交流渠道（触发中心）。在通信手段极为丰富的今天，能否支持电话、Web、传真、E-mail等各种触发手段进行交流，无疑是十分关键的。

(2) 对所获信息进行有效分析（挖掘中心）。

(3) CRM必须能与ERP很好地集成。作为企业管理的前台，CRM的市场营销和客户服务的信息必须能及时传达到后台的财务、生产等部门，这是企业能否有效运营的关键。

CRM的实现过程具体说来，它包含三方面的工作。一是客户服务与支持，即通过控制服务品质以赢得顾客的忠诚度，例如，对客户快速准确的技术支持、对客户投诉的快速反应、对客户提供产品查询等。二是客户群维系，即通过与顾客的交流实现新的销售，例如，通过交流赢得失去的客户等。三是商机管理，即利用数据库开展销售，例如，利用现有客户数据库做新产品推广测试，通过电话促销调查，确定目标客户群等。

在CRM中，销售自动化是其中最为基本的模块，营销自动化作为销售自动化的补充，包括营销计划的编制和执行、计划结果分析等功能。客户服务与支持是CRM系统的重要功能。目前，客户服务与支持的主要手段有两种，分别是呼叫中心和互联网。CRM系统能够与ERP系统在财务、制造、库存等环节进行连接，两者之间虽然关系比较独立，但由于两者之间具有一定的关系，因此会形成一定的闭环反馈结构。

试题答案

(19) C

第6章：信息化基础

试题6（2010年11月试题20）

试题6（2010年11月试题20）

共享数据库是一种重要的企业应用集成方式。以下关于共享数据库集成方式的叙述中，错误的是（20）。

(20) A. 共享数据库集成方式通常将应用程序的数据存储在一个共享数据库中，通过制定统一的数据库模式来处理不同应用的集成需求

B. 共享数据库为不同的应用程序提供了统一的数据存储与格式定义，能够解决不同应用程序中数据语义不一致的问题

C. 多个应用程序可能通过共享数据库频繁地读取和修改相同的数据，这会使共享数据库成为一个性能瓶颈

D. 共享数据库集成方式的一个重要限制来自外部的已封装应用，这些封装好的应用程序只能采用自己定义的数据库模式，调整和集成余地较小

试题分析

共享数据库是一种重要的企业应用集成方式，它通常将应用程序的数据存储在一个共享数据库中，通过制定统一的数据库模式来处理不同应用的集成需求。共享数据库为不同的应用程序提供了统一的数据存储与格式定义，能够在一定程度上缓解数据语义不一致的问题，但无法完全解决该问题。在共享数据库集成中，多个应用程序可能通过共享数据库频繁地读取和修改相同的数据，这会使数据库成为一个性能瓶颈。共享数据库集成方式的一个重要限制来自外部的已封装应用，这些封装好的应用程序只能采用自己定义的数据库模式，调整和集成余地较小。

试题答案

(20) B

第 6 章：信息化基础

试题7 (2010年11月试题21)

试题7 (2010年11月试题21)

某公司欲对其内部的信息系统进行集成，需要实现在系统之间快速传递可定制格式的数据包，并且当有新的数据包到达时，接收系统会自动得到通知。另外还要求支持数据重传，以确保传输的成功。针对这些集成需求，应该采用 (21) 的集成方式。

(21) A.远程过程调用 B.共享数据库 C.文件传输 D.消息传递

试题分析

根据题干描述，该公司需要在应用集成后实现采用可定制的格式频繁地、立即地、可靠地、异步地传输数据包。远程过程调用一般是基于同步的方式，效率较低，而且容易失败；共享数据库和文件传输的集成方式在性能方面较差，系统不能保持即时数据同步，而且容易造成应用与数据紧耦合；消息传递的集成方式能够保证数据的异步、立即、可靠传输，恰好能够满足该公司的集成需求。

试题答案

(21) D

第 6 章：信息化基础

试题8 (2011年11月试题18~19)

试题8 (2011年11月试题18~19)

某企业欲对内部的数据库进行数据集成。如果集成系统的业务逻辑较为简单，仅使用数据库中的单表数据即可实现业务功能，这时采用 (18) 方式进行数据交换与处理较为合适；如果集成系统

的业务逻辑较为复杂，并需要通过数据库中不同表的连接操作获取数据才能实现业务功能，这时采用（19）方式进行数据交换与处理较为合适。

- (18) A.数据网关 B.主动记录 C.包装器 D.数据映射
(19) A.数据网关 B.主动记录 C.包装器 D.数据映射

试题分析

关键要判断在进行集成时，需要数据库的单表还是多表进行数据整合。如果是单表即可完成整合，则可以将该表包装为记录，采用主动记录的方式进行集成；如果需要多张表进行数据整合，则需要采用数据映射的方式完成数据集成与处理。

试题答案

- (18) B (19) D

第6章：信息化基础

试题9（2011年11月试题20~21）

试题9（2011年11月试题20~21）

某大型商业公司欲集成其内部的多个业务系统，这些业务系统的运行平台和开发语言差异较大，而且系统所使用的通信协议和数据格式各不相同，针对这种情况，采用基于（20）的集成框架较为合适。除此以外，集成系统还需要根据公司的新业务需要，灵活、动态地定制系统之间的功能协作关系，针对这一需求，应该选择基于（21）技术的实现方式更为合适。

- (20) A.数据库 B.文件系统 C.总线 D.点对点
(21) A.分布式对象 B.远程过程调用 C.进程间通信 D.工作流

试题分析

针对题干描述，该企业进行系统集成时，“业务系统的运行平台和开发语言差异较大，而且系统所使用的通信协议和数据格式各不相同”。在这种情况下，需要采用总线技术对传输协议和数据格式进行转换与适配。当需要集成并灵活定义系统功能之间的协作关系时，应该采用基于工作流的功能关系定义方式。

试题答案

- (20) C (21) D

第6章：信息化基础

试题10（2011年11月试题37）

试题10（2011年11月试题37）

运用信息技术进行知识的挖掘和 (37) 的管理是企业信息化建设的重要活动。

- (37) A.业务流程 B.IT基础设施 C.数据架构 D.规章制度

试题分析

企业信息化建设是通过IT技术的部署来提高企业的生产运维效率，从而降低经营成本。这个过程中业务流程的管理与知识的挖掘是重要的活动。因为在进行信息化过程中，由于计算机技术的引入，使得企业原本手工化的业务流程需要优化，从而适应计算机化的快速处理。同时从企业已积累的资源库中，挖掘有价值的信息，也是信息化建设的重点，这些知识的挖掘，能给企业带来丰厚的利润。

试题答案

- (37) A

第6章：信息化基础

试题11（2011年11月试题38）

试题11（2011年11月试题38）

以下关于企业信息化方法的叙述中，正确的是 (38)。

- (38) A.业务流程重构是对企业的组织结构和工作方法进行重新设计，SCM（供应链管理）是一种重要的实现手段
B.在业务数量浩繁且流程错综复杂的大型企业里，主题数据库方法往往形成许多“信息孤岛”，造成大量的无效或低效投资
C.人力资源管理把企业的部分优秀员工看作是一种资本，能够取得投资收益
D.围绕核心业务应用计算机和网络技术是企业信息化建设的有效途径

试题分析

选项A描述错误，因为业务流程重构是对业务流程的优化，而非针对组织结构和工作方法。选项B描述错误，因为事务型数据库容易形成信息孤岛，而主题数据库不容易形成“信息孤岛”。C选项描述错误，因为人力资源是把所有员工看作是一种资本，而非部分员工。

试题答案

- (38) D

第6章：信息化基础

试题12（2012年11月试题18）

试题12（2012年11月试题18）

ERP中的企业资源包括 (18) .

- (18) A.物流、资金流和信息流 B.物流、工作流和信息流
- C.物流、资金流和工作流 D.资金流、工作流和信息流

试题分析

企业的所有资源包括三大流：物流、资金流和信息流。企业资源计划 (ERP,enterprise resource planning) 也就是对这三种资源进行全面集成管理的管理信息系统。概括地说，ERP是建立在信息技术基础上，利用现代企业的先进管理思想，全面地集成了企业的所有资源信息，并为企业提供决策、计划、控制与经营业绩评估的全方位和系统化的管理平台。ERP系统是一种管理理论和管理思想，不仅仅是信息系统。它利用企业的所有资源，包括内部资源与外部市场资源，为企业制造产品或提供服务创造最优的解决方案，最终达到企业的经营目标。

ERP理论与系统是从MRP-II发展而来的。MRP-II的核心是物流，主线是计划，但ERP已将管理的重心转移到财务上，在企业整个经营运作过程中贯穿了财务成本控制的概念。ERP极大地扩展了业务管理的范围及深度，包括质量、设备、分销、运输、多工厂管理、数据采集接口等。ERP的管理范围涉及企业的所有供需过程，是对供应链的全面管理。

典型的ERP系统包含以下模块：

(1) 生产预测

市场需求是企业生存的基础，在ERP中首先需要对市场进行较准确的预测。预测主要用于计划，在ERP的5个层次的计划中，前3个层次计划，即经营计划、生产计划大纲和主生产计划的编制都离不开预测。常用的预测方法有德尔菲 (Delphi) 方法、移动平均法、指数平滑法和非线性最小二乘曲线拟合法。

(2) 销售管理 (计划)

销售管理主要是对企业的销售部门的相关业务进行管理。企业销售部门是企业与市场连接的桥梁，其主要职能是为客户提供服务，从而使企业获得利润，实现其经济和社会价值。销售管理从其计划角度来看，属于最高层计划的范畴，是企业最重要的决策层计划之一。

(3) 经营计划 (生产计划大纲)

生产计划大纲 (Production Planning,PP) 是根据经营计划的生产目标制定的，是对企业经营计划的细化，用以描述企业在可用资源的条件下，在一定时期的产量计划。生产计划大纲在企业决策层的三个计划中有承上启下的作用，一方面它是企业经营计划和战略规划的细化，另一方面它又用于指导企业编制主生产计划，指导企业有计划地进行生产。

(4) 主生产计划

主生产计划 (Master Production Schedule,MPS) 是对企业生产计划大纲的细化，说明在一定时期内的如下计划：生产什么，生产多少和什么时候交货。主生产计划的编制以生产大纲为准，其汇总结果应当等同于生产计划大纲，同时，主生产计划又是其下一层计划--物料需求计划的编制依据。

主生产计划的编制是ERP的主要工作内容。主生产计划的质量将大大影响企业的生产组织工作和资源的利用。

(5) 物料需求计划

物料需求计划 (Material Requirement Planning,MRP) 是对主生产计划的各个项目所需的全部制造件和全部采购件的网络支持计划和时间进度计划。它根据主生产计划对最终产品的需求量和交货期，推导出构成产品的零部件及材料的需求量和需求时期，再导出自制零部件的制作订单下达日期和采购件的采购订单发送日期，并进行需求资源和可用能力之间的进一步平衡。物料需求

计划是生产管理的核心，它将主生产计划安排生产的产品分解成各自制零部件的生产计划和采购件的采购计划。物料需求计划属于ERP管理层计划。

(6) 能力需求计划

能力需求计划 (Capacity Requirements Planning,CRP) 是对物料需求计划所需能力进行核算的一种计划管理方法。旨在通过分析比较MRP的需求和企业现有生产能力，及早发现能力的瓶颈所在，为实现企业的生产任务而提供能力方面的保障。

(7) 车间作业计划

车间作业计划 (Production Activity Control,PAC) 是在MRP所产生的加工制造订单 (即自制零部件生产计划) 的基础上，按照交货期的前后和生产优先级选择原则以及车间的生产资源情况 (如设备、人员、物料的可用性、加工能力的大小等)，将零部件的生产计划以订单的形式下达给适当的车间。车间作业计划属于ERP执行层计划。当前主流的车间作业计划模式是JIT (Just In Time) 模式。

(8) 采购与库存管理

采购与库存管理是ERP的基本模块，其中采购管理模块是对采购工作--从采购订单产生至货物收到的全过程进行组织、实施与控制，库存管理 (Inventory Management,IM) 模块则是对企业物料的进、出、存进行管理。

(9) 质量与设备管理

质量管理贯穿于企业管理的始终。企业经营活动中的各环节、各项工作以及各种产品都离不开质量，都要讲究质量。全面质量管理 (Total Quality Management,TQM) 是质量管理的主要实施模式，它要求对企业的全过程进行质量管理，而且明确指出执行质量职能是企业全体人员的责任。

设备管理是指依据企业的生产经营目标，通过一系列的技术、经济和组织措施，对设备寿命周期内的所有设备物资运动形态和价值运动形态进行的综合管理。

(10) 财务管理

会计工作是企业管理的重要组成部分，是以货币的形式反映和监督企业的日常经济活动，并对这些经济业务的数据进行分类、汇总，以便为企业管理和决策提供必要的信息支持。企业财务管理是企业会计工作和活动的统称，财务管理是一种综合性的管理，它渗透在企业全面的经济活动之中，哪里有经济活动，哪里就有资金运动，哪里就有财务管理。

(11) ERP有关扩展应用模块

如客户关系管理、分销资源管理、供应链管理和电子商务等。这几个扩展模块本身也是一个独立的系统，在市场上它们常作为独立的软件产品进行出售和实施。

试题答案

(18) A

CRM是一套先进的管理思想及技术手段，它通过将（19）进行有效的整合，最终为企业涉及的各个领域提供了集成环境。CRM系统的四个主要模块包括（20）。

（19）A.员工资源、客户资源与管理技术 B.销售资源、信息资源与商业智能

C.销售管理、市场管理与服务管理 D.人力资源、业务流程与专业技术

（20）A.电子商务支持、呼叫中心、移动设备支持、数据分析

B.信息分析、网络应用支持、客户信息仓库、工作流集成

C.销售自动化、营销自动化、客户服务与支持、商业智能

D.销售管理、市场管理、服务管理、现场服务管理

试题分析

CRM是一套先进的管理思想及技术手段，它通过将人力资源、业务流程与专业技术进行有效的整合，最终为企业涉及客户或消费者的各个领域提供了完美的集成，使得企业可以更低成本、更高效率地满足客户的需求，并与客户建立起基于学习型关系基础上的一对一营销模式，从而让企业可以最大程度的提高客户满意度及忠诚度，挽回失去的客户，保留现有的客户，不断发展新的客户，发掘并牢牢地把握住能给企业带来最大价值的客户群。CRM系统的主要模块包括销售自动化、营销自动化、客户服务与支持、商业智能。

试题答案

（19）D（20）C

第6章：信息化基础

试题14（2012年11月试题21）

试题14（2012年11月试题21）

企业信息化程度是国家信息化建设的基础和关键，企业信息化方法不包括（21）。

（21）A.业务流程重组 B.组织机构变革

C.供应链管理 D.人力资本投资

试题分析

通过二三十年的发展，人们已经总结出了许多非常实用的企业信息化方法，并且还在探索新的方法。这里只简单介绍几种常用的企业信息化方法。

（1）业务流程重构方法

企业业务流程重构的中心思想是，在信息技术和网络技术迅猛发展的时代，企业必须重新审视企业的生产经营过程，利用信息技术和网络技术，对企业的组织结构和工作方法进行“彻底的、根本性的”重新设计，以适应当今市场发展和信息社会的需求。

（2）核心业务应用方法

任何一家企业，要想在市场竞争的环境中生存发展，都必须有自己的核心业务，否则，必然会被市场淘汰。当然，不同的企业，其核心业务是不同的。比如，一个石油生产企业，原油的勘探开发生产就是它的核心业务。围绕核心业务应用计算机技术和网络技术是很多企业信息化成功的秘诀。

(3) 信息系统建设方法

对大多数企业来说，建设信息系统是企业信息化的重点和关键。因此，信息系统建设成了最具普遍意义的企业信息化方法。

(4) 主题数据库方法

主题数据库就是面向企业业务主题的数据库，也就是面向企业的核心业务的数据库。有些企业，特别是在业务数量浩繁，流程错综复杂的大型企业里，建设覆盖整个企业的信息系统往往很难成功，但是，各个部门的局部开发和应用又有很大弊端，会造成系统分割严重，形成许多“信息孤岛”，造成大量的无效或低效投资。在这样的企业里，应用主题数据库方法推进企业信息化无疑是一个投入少、效益好的方法。

(5) 资源管理方法

计算机技术和网络技术的应用为企业资源管理提供了强大的能力。目前，流行的企业信息化的资源管理方法有很多，最常见的有ERP（企业资源计划）、SCM（供应链管理）等。

(6) 人力资本投资方法

人力资本的概念是经济学理论发展的产物。人力资本与人力资源的主要区别是人力资本理论把一部分企业的优秀员工看做是一种资本，能够取得投资收益。

人力资本投资方法特别适用于那些依靠智力和知识而生存的企业，例如，各种咨询服务、软件开发等企业。

试题答案

(21) B

第6章：信息化基础

试题15（2012年11月试题22~23）

试题15（2012年11月试题22~23）

企业信息资源集成管理的前提是对企业（22）的集成，其核心是对企业（23）的集成。

(22) A.信息功能 B.信息设施 C.信息活动 D.信息处理

(23) A.业务流 B.内部信息流
C.外部信息流 D.内部和外部信息流

试题分析

企业信息资源管理属于微观层次的信息资源管理的范畴，指企业为达到预定的目标运用现代的管理方法和手段对与企业相关的信息资源和信息活动进行组织、规划、协调和控制，以实现对企业信息资源的合理开发和有效利用。

企业信息资源是企业在信息活动中积累起来的以信息为核心的各类信息活动要素（信息技术、设备、信息生产者等）的集合。企业信息资源管理的任务是有效地搜集、获取和处理企业内外信息，最大限度地提高企业信息资源的质量、可用性和价值，并使企业各部分能够共享这些信息资源。由于企业是以利润最大化为目标的经济组织，其信息资源管理的主要目的在于发挥信息的社会效益和潜在的增值功能，为完成企业的生产、经营、销售工作，提高企业的经济效益，同时也为提

高社会效益。一般而言，企业信息资源管理工作的内容主要包括：对信息资源的管理、对人的管理、对相关信息工作的管理。

而企业信息资源管理中的集成管理是一种全新的管理理念和方法。集成管理作为高科技时代的管理创新，正在逐渐渗透和应用到社会经济的各个领域。集成管理是企业信息资源管理的主要内容之一。实行企业信息资源集成的前提是对企业历史上形成的企业信息功能的集成，其核心是对企业内部和外部信息流的集成，其实施的基础是各种信息手段的集成。通过集成管理实现企业信息系统各要素的优化组合，使信息系统各要素之间形成强大的协同作用，从而最大限度地放大企业信息的功能，实现企业可持续发展的目的。

试题答案

(22) A (23) D

第 6 章：信息化基础

试题16 (2012年11月试题24)

试题16 (2012年11月试题24)

为了加强对企业信息资源的管理，企业应按照信息化和现代化企业管理要求设置信息管理机构，建立信息中心。信息中心的主要职能不包括（ 24 ）。

- (24) A. 处理信息，确定信息处理的方法
- B. 用先进的信息技术提高业务管理水平
- C. 组织招聘信息资源管理员
- D. 建立业务部门期望的信息系统和网络

试题分析

为了加强对企业信息资源的管理，企业应按照信息化和现代化企业管理要求设置信息管理机构，建立信息中心，确定信息主管，统一管理和协调企业信息资源的开发、收集和使用。信息中心是企业的独立机构，直接由最高层领导并为企业最高管理者提供服务。其主要职能是处理信息，确定信息处理的方法，用先进的信息技术提高业务管理水平，建立业务部门期望的信息系统和网络并预测未来的信息系统和网络，培养信息资源的管理人员等。

试题答案

(24) C

第 7 章：系统开发基础

考点分析

第7章系统开发基础

7.1 考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

(1) 开发管理

项目的范围、时间、成本

文档管理工作、配置管理

软件开发的质量与风险

软件的运行与评价

(2) 需求管理

需求变更

需求跟踪

需求变更风险管理

(3) 软件开发方法

软件开发生命周期

软件开发模型（瀑布模型、演化模型、增量模型、螺旋模型、原型，构件组装模型、RUP,敏捷方法）

构件与软件重用

逆向工程

形式化方法

(4) 软件开发环境与工具

集成开发环境

开发工具（建模工具、分析设计工具、编程工具、测试工具、项目管理工具等）

(5) 设计方法

分析设计图示（DFD、ERD、UML、流程图、NS图、PAD）

结构化分析与设计

模块设计

面向对象的分析与设计

I/O设计、人机界面设计

设计模式

(6) 基于构件的开发

构件的概念与分类

中间件技术

典型应用架构（J2EE、。NET）

(7) 应用系统构建

应用系统设计与开发（分析与设计方法的使用、外部设计、内部设计、程序设计、测试）

软件包的使用（开发工具、运行管理工具、业务处理工具、ERP、群件、OA工具）

(8) 测试与评审

测试评审方法

验证与确认（V&V）

测试自动化

测试设计和管理方法

从历年的考试情况来看，本章的考点主要集中在以下方面：开发模型、开发环境、配置管理、需求管理、时间管理、面向对象基础、软件测试。具体考查知识点分布情况如表7-1所示。

表7-1 历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题 21~29, 32~34, 42~44	用户文档、配置项、需求变更、需求管理、敏捷方法、项目管理工具、逆向工程、用例包含关系、UML图、界面设计、系统测试、黑盒测试
2010年11月试题 22~32, 41~43	范围定义、时间管理、变更管理、CMM、“4+1”视图模型、螺旋模型、软件开发环境、软件重用、开发模型、有效性验证、系统测试
2011年11月试题 22~32, 39~43	软件产品配置、软件质量保证、需求跟踪能力链、需求定义方法、需求管理或需求开发、RUP、实体类、边界类、控制类、最少知识原则、软件开发方法、系统设计、快速迭代原型开发、静态分析、确认测试、软件维护
2012年11月试题 25~37	软件生存周期模型、螺旋模型、UML的需求分析过程、RAD、软件开发方法、软件开发环境、里氏替换原则、黑盒测试、软件测试工具

第7章：系统开发基础

试题1（2009年11月试题21）

7.2 试题精解

试题1（2009年11月试题21）

用户文档主要描述所交付系统的功能和使用方法。下列文档中，（21）属于用户文档。

- (21) A.需求说明书 B.系统设计文档 C.安装文档 D.系统测试计划

试题分析

用户文档主要描述所交付系统的功能和使用方法，并不关心这些功能是怎样实现的。用户文档是了解系统的第一步，它可以让用户获得对系统准确的初步印象。

用户文档一般包括以下内容。

- ① 功能描述：说明系统能做什么。
- ② 安装文档：说明怎样安装这个系统，以及怎样使系统适应特定的硬件配置。
- ③ 使用手册：简要说明如何着手使用这个系统（通过丰富的例子说明怎样使用常用的系统功能，并说明用户操作错误是怎样恢复和重新启动的）。
- ④ 参考手册：详尽描述用户可以使用的所有系统设施以及它们的使用方法，并解释系统可能产生的各种出错信息的含义（对参考手册最主要的要求是完整，因此通常使用形式化的描述技术）。
- ⑤ 操作员指南（如果有系统操作员的话）：说明操作员应如何处理使用中出现的各种情况。

试题中只有安装文档属于用户文档。其他的：需求说明书、系统设计文档、系统测试计划均属于开发文档。

试题答案

- (21) C

试题2 (2009年11月试题22)

试题2 (2009年11月试题22)

配置项是构成产品配置的主要元素，其中 (22) 不属于配置项。

- (22) A.设备清单 B.项目质量报告 C.源代码 D.测试用例

试题分析

信息系统在其开发、运行、维护的过程中会得到许多阶段性的成果，在开发和运行过程中还需要用到多种工具软件，所有这些信息项都需要得到妥善的管理，决不能出现混乱，以便在提出某些特定的要求时，将它们进行约定的组合来满足使用的目的。这些信息项是配置管理的对象，称为配置项。IEEE对配置项的定义为：硬件、软件或二者兼有的集合，为配置管理指定的，在配置管理过程中作为一个单独的实体对待。

1. 配置项的类型

以下内容可以作为配置项进行管理：外部交付的软件产品和数据、指定的内部软件工作产品和数据、指定的用于创建或支持软件产品的支持工具、供方/供应商提供的软件和客户提供的设备/软件。配置项通常可以分成下列6种类型：

- (1) 环境类。软件开发、运行和维护的环境，例如，编译器、操作系统、编辑软件、管理系统、开发工具、测试工具、项目管理工具和文档编制工具等。
- (2) 定义类。需求分析与系统定义阶段结束后得到的成果，例如，需求规格说明书、项目管理计划、设计标准和验收测试计划等。
- (3) 设计类。设计阶段得到的成果，例如，系统设计说明书、程序规格说明、数据库设计、编码标准、用户界面设计、测试标准、系统测试计划和用户手册等。
- (4) 编码类。编码及单元测试结束后得到的成果，例如，源代码、目标码、单元测试用例、数据和测试结果等。
- (5) 测试类。系统测试完成后的成果，例如，系统测试用例、测试结果、操作手册和安装手册。
- (6) 维护类。维护阶段产品的成果，以上任何需要变更的配置项。

2. 配置项的描述

确定了配置项后，还需要对配置项进行合理、科学的命名。配置项的命名绝不能随意为之，必须满足惟一性和可追溯性。一个典型的实例是采用层次式的命名规则来反映树状结构，树状结构上结点之间存在着层次的继承关系。

由于配置项除了名称外还有一些其他属性和与其他配置项的关系，因此，它可以采用描述对象的方式来进行描述。每个配置项用一组特征信息（名字、描述、一组资源、实现）惟一地标识。配置项之间的关系有整体和部分的关系及层次关系，也有关联关系。配置项间的关系可以用MIL (Module Interconnection Language, 模块连接语言) 表示，MIL描述的是配置项间的相互依赖关系，可自动构造系统的任何版本。

3. 识别配置项的步骤

识别配置项的主要步骤如下：

- (1) 识别配置项。
- (2) 为每个配置项指定惟一性的标识代号。

- (3) 确定每个配置项的重要特征。配置项的特征主要包括作者、日期、类型等。
- (4) 确定配置项进入配置管理的时间。
- (5) 确定每个配置项的拥有者及责任。
- (6) 填写配置管理表。
- (7) 审批配置管理表。CCB审查配置管理表是否符合配置管理计划的规定，审批配置管理表。

试题答案

- (22) A

第 7 章：系统开发基础

试题3 (2009年11月试题23)

试题3 (2009年11月试题23)

一个大型软件系统的需求通常会发生变化的。以下关于需求变更策略的叙述中，错误的是

(23) .

- (23) A.所有需求变更必须遵循变更控制过程
- B.对于未获得核准的变更，不应该做变更实现工作
- C.完成了对某个需求的变更之后，就可以删除或者修改变更请求的原始文档
- D.每一个集成的需求变更必须能追溯到一个经核准的变更请求

试题分析

在软件项目中，需求的变化是不可避免的。需求变更可能来自解决方案提供商、用户或产品供应商等外部因素，也可能来源于项目团队内部。对于项目团队而言，无法阻止需求发生变更，他们只能正确地对待变更，按照既定流程管理变更，尽量降低变更对项目成本、进度和质量的负面影响。

1. 需求基线

需求开发的结果应该有项目视图和范围文档、用例文档和SRS,以及相关的分析模型。经评审批准，这些文档就定义了开发工作的需求基线。这个基线在用户和开发人员之间就构成了软件需求的一个约定，它是需求开发和需求管理之间的桥梁。

基线是一个软件配置管理的概念，它帮助开发人员在不严重阻碍合理变化的情况下控制变化。根据IEEE的定义，基线是指已经通过正式评审和批准的规约或产品，它可以作为进一步开发的基础，并且只能通过正式的变更控制系统进行变化。在软件工程范围内，基线是软件开发中的里程碑，其标志是有一个或多个软件配置项的交付，且已经通过正式技术评审而获得认可。例如，SRS文档通过评审，其中的错误已经被发现并纠正，则就变成了一个基线。根据国家标准《计算机软件配置管理计划规范》(GB/T 12505-1990)的规定，基线可以分为功能基线、指派基线和产品基线三种，通过评审后的SRS属于指派基线。

开发团队可以根据已知的需求基线来区分"旧需求"和"新需求".一旦建立了需求基线，就很容易对新需求进行识别和管理，可以把新需求和已有的基线加以比较，确定适合它的位置以及它是否会与其他需求产生冲突。如果接受新需求，就可以管理它的变更过程。

2. 需求的状态

从需求的整个生命周期来看，其状态的变化如图7-1所示。

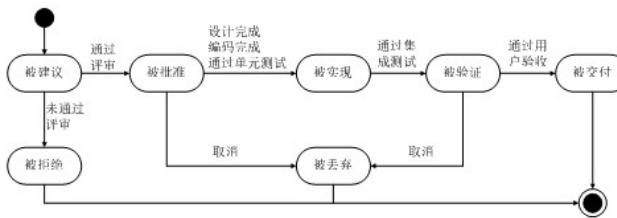


图7-1 需求状态的变化

在需求状态的变化中，项目管理人员首先需要关注的是那些被拒绝和被丢弃的需求。因为这些需求有可能是应该被接受和被实现的需求，如果不是通过有管理的处理过程，就有可能因为疏忽而被遗漏。同时，也应关注被交付的需求，因为可交付物是项目的成果体现，而可交付物的主要内容就是对需求的实现。

3. 需求变更

在各种理论书籍中，都会介绍一些如何减少需求变更的方法和技术。在项目实践中，项目管理人员也会花大量的精力去实践这些方法和技术，以避免需求变更。遗憾的是，“是祸躲不过”，需求变更因各种因素而依然发生，不可避免。当然，这并不是说不应该做避免变更的工作，恰恰相反，在需求变更之前尽量减少变更，以将需求变更带来的风险降到最低，这是对项目进展十分有利的。

需求变更通常意味着新需求的增加和对已有需求的修改，一般不会减少需求，而且减少需求的问题也比较容易处理。需求变更是需要代价的，包括时间、人力、资源等方面。既然需求变更是不可避免的，那么，项目管理人员就应该采取规范的流程去管理变更，而不是一味地避免变更和拒绝变更。在进行需求变更时，可以参考以下的需求变更策略：

- ① 所有需求变更必须遵循变更控制过程；
- ② 对于未获得批准的变更，不应该做设计和实现工作；
- ③ 变更应该由项目变更控制委员会决定实现哪些变更；
- ④ 项目风险承担者应该能够了解变更数据库的内容；
- ⑤ 决不能从数据库中删除或者修改变更请求的原始文档；
- ⑥ 每一个集成的需求变更必须能跟踪到一个经核准的变更请求。

试题答案

(23) C

试题4 (2009年11月试题24)

试题4 (2009年11月试题24)

以下关于需求管理的叙述中，正确的是（24）。

- (24) A. 需求管理是一个对系统需求及其变更进行了解和控制的过程
B. 为了获得项目，开发人员可以先向客户做出某些承诺
C. 需求管理的重点在于收集和分析项目需求

D.软件开发过程是独立于需求管理的活动

试题分析

软件需求开发的最终文档经过评审批准后，则定义了开发工作的需求基线。这个基线在客户和开发者之间构筑了计划产品功能需求和非功能需求的一个约定 (Agreement)。需求约定是需求开发和需求管理之间的桥梁。

需求管理是一个对系统需求变更、了解和控制的过程。需求管理过程与需求开发过程相互关联，当初始需求导出的同时就启动了需求管理规划，一旦形成了需求文档的初稿，需求管理活动就开始了。需求管理强调：

- (1) 控制对需求基线的变动；
- (2) 保持项目计划与需求一致；
- (3) 控制单个需求和需求文档的版本情况；
- (4) 管理需求和联系链，或者管理单个需求和其他项目可交付产品之间的依赖关系；
- (5) 跟踪基线中的需求状态。

CMMI描述了软件处理能力的 5 个成熟级别。为了达到过程能力成熟度模型的第二级，组织机构必须具有 6 个关键过程域 KPA (Key Process Areas)。需求管理是其中之一，它的目标是：

- (1) 为软件需求建立一个基线，提供给软件工程和管理使用；
- (2) 软件计划、产品和活动与软件需求保持一致。

关于需求管理过程域内的原则和策略，可以参考：

(1) 需求管理的关键过程领域不涉及收集和分析项目需求，而是假定已收集了软件需求，或者已由更高一级的系统给定了需求。一旦需求获得并且文档化了，软件开发组和有关的团队 (例如质量保证和测试组) 需要评审文档。发现问题应与客户或者其他需求源协商解决。软件开发计划是基于已确认的需求。

(2) 开发人员在向客户以及有关部门承诺 (Commitment) 某些需求之前，应该确认需求和约束条件、风险、偶然因素、假定条件等。也许不得不面对由于技术因素或者进度等原因，承诺一些不现实的需求。但是，决不要承诺任何无法实现的事。

(3) 关键处理领域同样建议通过版本控制和变更控制来管理需求文档。版本控制确保随时能知道在项目开发和计划中正在使用的需求的版本情况。变更控制提供了支配下的规范的方式来统一需求变更，并且基于业务和技术的因素来同意或者反对建议的变更。当在开发中修改、增加、减少需求时，软件开发计划应该随时更新，确保与新的需求保持一致。

试题答案

- (24) A

(25) 方法以原型开发思想为基础，采用迭代增量式开发，发行版本小型化，比较适合需求变化较大或者开发前期对需求不是很清晰的项目。

(25) A.信息工程 B.结构化 C.面向对象 D.敏捷

试题分析

敏捷方法是从20世纪90年代开始逐渐引起广泛关注的一些新型软件开发方法，以应对快速变化的需求。虽然它们的具体名称、理念、过程、术语都不尽相同，但相对于“非敏捷”而言，它们更强调开发团队与用户之间的紧密协作、面对面的沟通、频繁交付新的软件版本、紧凑而自我组织型的团队等，也更注重人的作用。

1. 敏捷宣言

2001年，Kent Beck等人组织了敏捷联盟，阐述了敏捷开发的原则，试图强调灵活性在快速且有效地开发软件中所发挥的作用，他们共同签署了敏捷软件开发宣言，该宣言认为，个体和交互胜过过程和工具；可工作的软件胜过大量的文档；客户合作胜过合同谈判；响应变化胜过遵循计划。

敏捷方法强调，让客户满意和软件尽早增量发布；小而高度自主的项目团队；非正式的方法；最小化软件工程工作产品以及整体精简开发。产生这种情况的原因是，在绝大多数软件开发过程中，提前预测哪些需求是稳定的和哪些需求会变化非常困难；对于软件项目构建来说，设计和实现是交错的；从指定计划的角度来看，分析、设计、实现和测试不容易预测；可执行原型和部分实现的可运行系统是了解用户需求和反馈的有效媒介。

目前，主要的敏捷方法有极限编程（eXtreme Programming,XP）、自适应软件开发（Adaptive Software Development,ASD）、水晶方法（Crystal）、特性驱动开发（Feature Driven Development,FDD）、动态系统开发方法（Dynamic Systems Development Method,DSDM）测试驱动开发（Test-Driven Development,TDD）、敏捷数据库技术（Agile Database Techniques,AD）和精益软件开发（Lean Software Development）等。虽然这些过程模型在实践上有差异，但都是遵循了敏捷宣言或者是敏捷联盟所定义的基本原则。这些原则包括客户参与、增量式移交、简单性、接受变更、强调开发人员的作用和及时反馈等。

2. 敏捷方法的特点

敏捷方法是一种以人为核心、迭代、循序渐进的开发方法。在敏捷方法中，软件项目的构建被切分成多个子项目，各个子项目成果都经过测试，具备集成和可运行的特征。在敏捷方法中，从开发者的角度来看，主要的关注点有短平快的会议、小版本发布、较少的文档、合作为重、客户直接参与、自动化测试、适应性计划调整和结对编程；从管理者的角度来看，主要的关注点有测试驱动开发、持续集成和重构。

近年来，虽然敏捷方法发展得较快，但在实施的过程中，也暴露出来很多问题，一些敏捷方法的基本原则很难实施，主要体现在以下四个方面：

- (1) 客户参与往往依赖于客户参与的意愿和客户自身的代表性。
- (2) 团队成员的性格可能不适合激烈的投入，可能无法做到与其他成员之间的良好沟通。
- (3) 对系统的变更作出优先级排序可能是极端困难的。
- (4) 维护系统的简洁性往往需要额外的工作，但迫于时间表的压力，可能没有时间执行系统简化过程。

与RUP相比，敏捷方法的周期可能更短。敏捷方法在几周或几个月的时间内完成相对较小的功能，强调的是能尽早将尽量小的可用的功能交付使用，并在整个项目周期中持续改善和增强，并且更加强调团队中的高度协作。相对而言，敏捷方法主要适用于以下场合：

- (1) 项目团队的人数不能太多，适合于规模较小的项目。

(2) 项目经常发生变更。敏捷方法适用于需求萌动并且快速改变的情况，如果系统有比较高的关键性、可靠性、安全性方面的要求，则可能不完全适合。

(3) 高风险项目的实施。

(4) 从组织结构的角度看，组织结构的文化、人员、沟通性决定了敏捷方法是否适用。与这些相关联的关键成功因素有组织文化必须支持谈判、人员彼此信任、人少但是精干、开发人员所作的决定得到认可、环境设施满足团队成员之间快速沟通的需要。

试题答案

(25) D

第7章：系统开发基础

试题6 (2009年11月试题26~27)

试题6 (2009年11月试题26~27)

项目管理工具用来辅助项目经理实施软件开发过程中的项目管理活动，它不能 (26) . (27) 就是一种典型的项目管理工具。

- (26) A. 覆盖整个软件生存周期
- B. 确定关键路径、松弛时间、超前时间和滞后时间
- C. 生成固定格式的报表和裁剪项目报告
- D. 指导软件设计人员按软件生存周期各个阶段的适用技术进行设计工作

- (27) A. 需求分析工具 B. 成本估算工具
- C. 软件评价工具 D. 文档分析工具

试题分析

软件管理过程和软件支持过程往往要涉及到软件生存周期中的多个活动，软件管理和软件支持工具用来辅助管理人员和软件支持人员的管理活动和支持活动，以确保软件高质高效地完成。

辅助软件管理和软件支持的工具有很多，其中常用的工具有项目管理工具、配置管理工具、软件评价工具等。

1、项目管理工具

项目管理工具用来辅助软件的项目管理活动。通常项目管理活动包括项目的计划、调度、通信、成本估算、资源分配及质量控制等。一个项目管理工具通常把重点放在某一个或某几个特定的管理环节上，而不提供对管理活动包罗万象的支持。

例如成本估算工具，采用某种成本估算模型（如 COCOMO模型）对项目的成本进行估算。它可以通过间接的测量（如对代码行和功能点的测量）来估算项目的规模大小，并描述总的项目特征，如问题的复杂度、开发组经验和过程成熟度等。然后按一定的估算模型估算出项目的工作量、工期和开发人员数等。当项目截止期限变更时，可检测它对整个开发成本的影响。

2、配置管理工具

配置管理工具用以辅助完成软件配置项的标识、版本控制、变化控制、审计和状态统计等基本任务，使各配置项的存取、修改和系统生成易于实现，从而简化审计过程，改进状态统计，减少错误，提高系统的质量。

3、软件评价工具

软件评价工具用以辅助管理人员进行软件质量保证的有关活动。它通常可按某个软件质量模型（如 McCall 软件质量模型，ISO 软件质量度量模型等）对被评价的软件进行度量，然后得到相关的软件评价报告。目前许多度量指标还不能定量化，需要通过专家评分，再将得分送给软件评价工具。对一些已经定量化的度量指标则可利用评价工具自动获取。有的评价工具还可分析被评价程序的程序结构，根据某种软件复杂性模型（如 Mc-Cabe 的环路复杂度等）对被评价的程序进行复杂性度量。软件评价工具有助于软件的质量控制，对确保软件的质量有重要的作用。

4、软件开发工具的评价和选择

现在各类软件开发工具十分丰富，有免费的，有价格便宜的，也有昂贵的。评价和选择适合本人、本单位、本项目的软件开发工具，可以根据以下标准来衡量软件开发工具的优劣。

（1）功能

软件开发工具不仅要实现所遵循的功能需求，支持用户所选定的开发方法，还应能检查与之相关的方法学能否正确执行，并保证产生与方法学一致的输出结果。

（2）易用性

软件开发工具应有十分友好的用户界面，用户乐于使用；工具应能剪裁和定制，以适应特定用户的需要；工具应能提示用户的交互操作，提供简单有效的执行方式；工具还应能检查用户的操作错误，尽可能自动改正错误。

（3）稳健性

一个好的软件开发工具应能长期可靠地使用，并能适应环境或其他条件变化的要求；即使在非法操作或故障情况下，也不应导致严重后果。

（4）硬件要求和性能

软件开发工具的性能（如响应速度、占用存储空间的大小等），将直接影响工具的使用效果。合理的性能和对硬件的要求可以使机器的资源能被有效地加以利用，使用户的投资发挥最大的作用。

（5）服务和支持

软件开发工具的生产厂商应能为该工具提供有效的技术服务（如培训、咨询、版本更新等），工具的文档应该齐全、通俗易懂。

试题答案

（26）D（27）B

逆向工程导出的信息可以分为4个抽象层次，其中（28）可以抽象出程序的抽象语法树、符号表等信息；（29）可以抽象出反映程序段功能及程序段之间关系的信息。

（28）A.实现级 B.结构级 C.功能级 D.领域级

（29）A.实现级 B.结构级 C.功能级 D.领域级

试题分析

逆向工程与重构工程是目前预防性维护采用的主要技术。逆向工程术语源于硬件制造业，相互竞争的公司为了了解对方设计和制造工艺的机密，在得不到设计和制造说明书的情况下，通过拆卸实物获取信息，软件的逆向工程也基本类似，不过通常“解剖”的不仅是竞争对手的程序，而且还包括本公司多年前的产品，此时得不到设计“机密”的主要障碍是缺乏文档。因此，所谓软件的逆向工程就是分析已有的程序，寻求比源代码更高级的抽象表现形式。一般认为，凡是在软件生命周期内将软件某种形式的描述转换成更为抽象形式的活动都可称为逆向工程。与之相关的概念是：重构（restructuring），指在同一抽象级别上转换系统描述形式；设计恢复（design recovery），指借助工具从已有程序中抽象出有关数据设计、总体结构设计和过程设计的信息（不一定是原设计）；重构工程（re-engineering），也称修复和改造工程，它是在逆向工程所获信息的基础上修改或重构已有的系统，产生系统的一个新版本。

1、恢复信息的级别

逆向工程导出的信息可分为如下4个抽象层次：

（1）实现级：包括程序的抽象语法树、符号表等信息；

（2）结构级：包括反映程序分量之间相互依赖关系的信息，例如调用图、结构图等；

（3）功能级：包括反映程序段功能及程序段之间关系的信息；

（4）领域级：包括反映程序分量或程序诸实体与应用领域概念之间对应关系的信息。

显然，上述信息的抽象级别越高，它与代码的距离就越远，通过逆向工程恢复的难度亦越大，而自动工具支持的可能性相对变小，要求人参与判断和推理的工作增多。

2、恢复信息的方法

在逆向工程中用于恢复信息的方法有四类。第一类为用户指导下的搜索与变换（user-directed search and transformation）。此类方法用于导出实现级和结构级信息。它要求维护人员在数据库系统的支持下，运用询问语言，针对源代码或与之相近的表示形式，指定待查找的句型（pattern），根据搜索结果析出所需信息或进行特殊变换。

第二类方法为变换式方法（transformational approaches），除领域级外所有抽象级别上的信息都可用此类方法推导。变换式方法又细分为不需要维护人员过多干涉的自动分析法（如静态分析和调用图、控制流图生成等）和基于特定库的用户指导变换法两类。变换方法自动化程度越高，得到的设计信息越粗略，因为任何深层次的分析不可避免地要借助人的智力。一般借助变换法得到程序的某种中间表示形式，通过进一步使用其他工具将已获粗略的设计信息精化为完整、一致的软件设计。

第三类方法是基于领域知识的（domain knowledge-based），主要用于恢复功能级和领域级信息。领域知识一般用规则库表示，用已确定或假定的领域概念与代码之间的对应关系推导进一步的假设，最后导出程序的功能。显然该类方法的不确定性最大，因此目前成熟的工具和原型系统还很少见。

最后一类方法称为铅板恢复（cliche recognition）法，这类方法仅适用于推导实现级和结构级信息。这些方法用于识别程序设计“铅板”或公共结构，“铅板”既可为一个简单算法（如两变量互换值），亦可为相对复杂的成分（如冒泡分类）。因铅板与程序之间可能存在多种匹配形式，所以此

类方法还包含大量的推理与决策。各类方法采用的输入形式、搜索策略和推理策略都不尽相同。后两类方法又称为基于知识的方法。

尽管每个软件组织都可能有数百万行代码可供重构，但由于缺乏时机和支持工具或者因为经济上得不偿失，往往只有那些决定或移植、或重新设计、或为重用而需验证正确性的程序才被选择实施逆向工程。

试题答案

(28) A (29) C

试题8 (2009年11月试题32)

用例 (use case) 用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。

在一个"订单输入子系统"中，创建新订单和更新订单都需要核查用户账号是否正确。用例"创建新订单"、"更新订单"与用例"核查客户账号"之间是 (32) 关系。

- (32) A.包含 (include) B.扩展 (extend)
C.分类 (classification) D.聚集 (aggregation)

试题分析

用例是在系统中执行的一系列动作，这些动作将生成特定参与者可见的价值结果。它确定了一个和系统参与者进行交互，并可由系统执行的动作序列。用例模型描述的是外部执行者 (Actor) 所理解的系统功能。用例模型用于需求分析阶段，它的建立是系统开发者和用户反复讨论的结果，表明了开发者和用户对需求规格达成的共识。

在建立了初步的用例模型后，还可以利用用例之间的关系来调整用例模型。用例之间的关系主要有包含、扩展和泛化，利用这些关系，把一些公共的信息抽取出来，以便于复用，使得用例模型更易于维护。

1、包含关系

当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。其中这个提取出来的公用用例称为抽象用例，而把原始用例称为基本用例或基础用例。例如，图7-2中的"学习课程"和"课程测试"两个用例都需要检查学员的权限，为此，可以定义一个抽象用例"检查权限"。用例"学习课程"和"课程测试"与用例"检查权限"之间的关系就是包含关系。其中"《include》"是包含关系的构造型，箭头指向抽象用例。

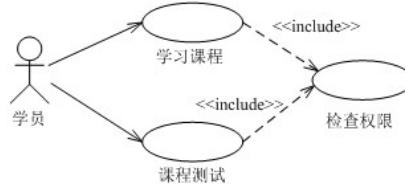


图7-2 包含关系的例子

当多个用例需要使用同一段事件流时，抽象成为公用用例，可以避免在多个用例中重复地描述这段事件流，也可以防止这段事件流在不同用例中的描述出现不一致。当需要修改这段公共的需求时，也只要修改一个用例，避免同时修改多个用例而产生的不一致性和重复性工作。另外，当某个用例的事件流过于复杂时，为了简化用例的描述，也可以将某一段事件流抽象成为一个被包含的用例。

2、扩展关系

如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。例如，图7-2中的学员进行"课程测试"时，其测试的次数可能已超出系统规定的限额，这时就需要学员"充入学习

币".用例"课程测试"和"充入学习币"之间的关系就是扩展关系，如图7-3所示。其中"《extend》"是扩展关系的构造型，箭头指向基本用例。

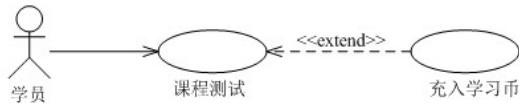


图7-3 扩展关系的例子

3、泛化关系

当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，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。例如，图7-4中学员进行课程注册时，假设既可以通过电话注册，也可以通过网上注册，则"注册课程"用例就是"电话注册"用例和"网上注册"用例的泛化。其中三角箭头指向父用例。

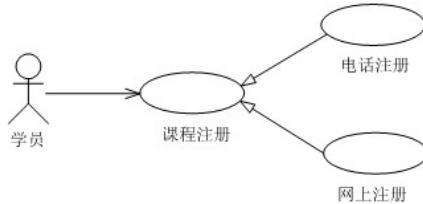


图7-4 泛化关系的例子

从UML事物关系的本质上来看，包含关系和扩展关系都属于依赖关系。对包含关系而言，抽象用例中的事件流是一定插入到基本用例中去的，并且插入点只有一个。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。在实际应用中，很少使用泛化关系，子用例的特殊行为都可以作为父用例中的备选事件流而存在。

试题答案

(32) A

第 7 章：系统开发基础

试题9 (2009年11月试题33~34)

试题9 (2009年11月试题33~34)

面向对象的设计模型包含以 (33) 表示的软件体系结构图，以 (34) 表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

(33) A.部署图B.包图C.协同图D.交互图

(34) A.部署图B.包图C.协同图D.交互图

试题分析

UML 2.0包括14种图，分别列举如下：

(1) 类图 (class diagram)。类图描述一组类、接口、协作和它们之间的关系。在OO系统的建模中，最常见的图就是类图。类图给出了系统的静态设计视图，活动类的类图给出了系统的静态进程视图。

(2) 对象图 (object diagram)。对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图一样，这些图给出系统的静态设计视图或静态进程视图，但它们是从真实案例或原型案例的角度建立的。

(3) 构件图 (component diagram)。构件图描述一个封装的类和它的接口、端口，以及由内嵌的构件和连接件构成的内部结构。构件图用于表示系统的静态设计实现视图。对于由小的部件构建大的系统来说，构件图是很重要的。构件图是类图的变体。

(4) 组合结构图 (composite structure diagram)。组合结构图描述结构化类（例如，构件或类）的内部结构，包括结构化类与系统其余部分的交互点。组合结构图用于画出结构化类的内部内容。

(5) 用例图 (use case diagram)。用例图描述一组用例、参与者及它们之间的关系。用例图给出系统的静态用例视图。这些图在对系统的行为进行组织和建模时是非常重要的。

(6) 顺序图 (sequence diagram, 序列图)。顺序图是一种交互图 (interaction diagram)，交互图展现了一种交互，它由一组对象或参与者以及它们之间可能发送的消息构成。交互图专注于系统的动态视图。顺序图是强调消息的时间次序的交互图。

(7) 通信图 (communication diagram)。通信图也是一种交互图，它强调收发消息的对象或参与者的结构组织。顺序图和通信图表达了类似的基本概念，但它们所强调的概念不同，顺序图强调的是时序，通信图强调的是对象之间的组织结构（关系）。在UML 1.X版本中，通信图称为协作图 (collaboration diagram)。

(8) 定时图 (timing diagram, 计时图)。定时图也是一种交互图，它强调消息跨越不同对象或参与者的实际时间，而不仅仅只是关心消息的相对顺序。

(9) 状态图 (state diagram)。状态图描述一个状态机，它由状态、转移、事件和活动组组成。状态图给出了对象的动态视图。它对于接口、类或协作的行为建模尤为重要，而且它强调事件导致的对象行为，这非常有助于对反应式系统建模。

(10) 活动图 (activity diagram)。活动图将进程或其他计算结构展示为计算内部一步步的控制流和数据流。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要，并强调对象间的控制流程。

(11) 部署图 (deployment diagram)。部署图描述对运行时的处理节点及在其中生存的构件的配置。部署图给出了架构的静态部署视图，通常一个节点包含一个或多个部署图。

(12) 制品图 (artifact diagram)。制品图描述计算机中一个系统的物理结构。制品包括文件、数据库和类似的物理比特集合。制品图通常与部署图一起使用。制品也给出了它们实现的类和构件。

(13) 包图 (package diagram)。包图描述由模型本身分解而成的组织单元，以及它们之间的依赖关系。

(14) 交互概览图 (interaction overview diagram)。交互概览图是活动图和顺序图的混合物。

面向对象的设计模型包含以包图表示的软件体系结构图，以交互图表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用来描述流程化处理的活动图等。

试题答案

试题10 (2009年11月试题42)

试题10 (2009年11月试题42)

系统输入设计中应尽可能地考虑人的因素，以下关于输入设计的一般原理中，错误的是 (42) .

- A.只让用户输入变化的数据
- B.使用创新的模式吸引用户的眼球
- C.表格中各个数据项应有提示信息
- D.尽可能使用选择而不是键盘输入的方式获取数据

试题分析

界面是系统与用户交互的最直接的层面，界面的好坏决定用户对系统的第一印象，而设计优良的界面能够引导用户自己完成相应的操作，起到向导的作用。同时，界面如同人的面孔，具有吸引用户的直接优势，设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉。相反，由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。通常情况下，良好的用户界面设计需要遵循如下一些基本原则：

(1) 置于用户控制之下。在定义人机交互方式时，不强迫用户采用不是必须的或者不情愿的方式来进行操作，允许交互的中断和撤销。当用户操作技能等级提高时，可以实现流水化的交互方式，允许用户定制交互方式，以便使用户界面与内部技术细节隔离，允许用户和出现在屏幕上的对象直接进行交互。

(2) 减轻用户的记忆负担。尽量减轻对用户记忆的要求，创建有意义的缺省设置，定义一些符合用户直觉的访问途径，适当定义一些快捷方式，界面的视觉布局应该尽量与真实世界保持一致，并能够以不断扩展的方式呈现信息。用户可以快速学习并使用系统，提供尽量“傻瓜式”的操作界面，方便用户使用。界面中各个元素的名称应该易懂，用词准确，避免模棱两可的字眼，能够做到“望文知意”，理想的情况是用户不用查阅帮助，就能知道该界面元素的功能，并正确地进行相关操作。

(3) 保持界面一致性。用户应以一致的方式提供或获取信息，所有可视信息的组织需要按照统一的设计标准，在系列化的应用软件中需要保持一致性，用户已经很熟悉的一些界面交互模型不到万不得已时，不要随意进行修改。需要确保用户界面操作和使用的一致性，例如，所有窗口按钮的位置要一致、提示信息和界面元素的命名要一致、界面颜色和风格要一致等。用户界面的一致性可以使用户能够统一地对待系统的各个不同的功能界面，以及系列化的系统，从而降低培训和支持成本。

以上三条原则由着名用户界面设计专家Theo Mandel博士所创造，通常称之为人机交互的“黄金三原则”。另外，在设计用户界面时，还需要保证界面的合理性和独特性，有效进行组合，注重美观与协调；恰到好处地提供快捷方式，注意资源协调等。

试题答案

试题11 (2009年11月试题43)

试题11 (2009年11月试题43)

系统测试将软件、硬件、网络等其他因素结合，对整个软件进行测试。（43）不是系统测试的内容。

- (43) A.路径测试 B.可靠性测试 C.安装测试 D.安全测试

试题分析

系统测试是将已经确认的软件、计算机硬件、外设和网络等其他因素结合在一起，进行信息系统的各种集成测试和确认测试，其目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方。系统测试是根据系统方案说明书来设计测试用例，常见的系统测试主要有恢复测试、安全性测试、压力测试、性能测试、可靠性测试、可用性测试、可维护性测试和安装测试。

试题答案

- (43) A

试题12 (2009年11月试题44)

试题12 (2009年11月试题44)

软件测试是为了发现错误而执行程序的过程。黑盒测试法主要根据（44）来设计测试用例。

- (44) A.程序内部逻辑 B.程序内部功能
C.程序数据结构 D.程序流程图

试题分析

黑盒测试也称为功能测试，主要用于集成测试、确认测试和系统测试阶段。黑盒测试将软件看作是一个不透明的黑盒，完全不考虑（或不了解）程序的内部结构和处理算法，而只检查软件功能是否能按照SRS的要求正常使用，软件是否能适当地接收输入数据并产生正确的输出信息，软件运行过程中能否保持外部信息（例如，文件和数据库等）的完整性等。

黑盒测试根据SRS所规定的功能来设计测试用例，一般包括功能分解、等价类划分、边界值分析、判定表、因果图、状态图、随机测试、猜错法和正交试验法等。

白盒测试也称为结构测试，主要用于软件单元测试阶段。它的主要思想是，将程序看作是一个透明的白盒，测试人员完全清楚程序的结构和处理算法，按照程序内部逻辑结构设计测试用例，检测程序中的主要执行通路是否都能按预定要求正确工作。白盒测试方法主要有控制流测试、数据流

测试和程序变异测试等。另外，使用静态测试的方法也可以实现白盒测试。例如，使用人工检查代码的方法来检查代码的逻辑问题，也属于白盒测试的范畴。

试题答案

(44) B

第 7 章：系统开发基础

试题13 (2010年11月试题22)

试题13 (2010年11月试题22)

详细的项目范围说明书是项目成功的关键。 (22) 不应该属于范围定义的输入。

- (22) A.项目章程 B.项目范围管理计划
- C.需求文件 D.项目文档管理方案

试题分析

定义范围 (范围定义) 是制定项目和产品详细描述的过程。详细项目范围说明书的编制，对项目成功至关重要。项目管理团队应该根据项目启动过程中记载的主要可交付成果、假设条件和制约因素，来编制项目范围说明书。在规划过程中，由于对项目有了更多的了解，所以应该更具体地定义与描述项目范围。项目管理团队应该分析现有风险、假设条件和制约因素的完整性，并在必要时补充其他的风险、假设条件和制约因素。

定义范围最重要的任务就是详细定义项目的范围边界，范围边界是应该做的工作和不需要进行的工作的分界线。定义范围可以增加项目时间、费用和资源估算的准确度，定义实施项目控制的依据，明确相关责任人在项目中的责任，明确项目的范围、合理性和目标，以及主要可交付成果。

根据PMBOK 2012版，定义范围的输入包括范围管理计划、项目章程、需求文件、组织过程资产，输出包括项目范围说明书、项目文件更新，使用的工具与技术有专家判断、产品分析、备选方案生成、引导式研讨会。

试题答案

(22) D

第 7 章：系统开发基础

试题14 (2010年11月试题23)

试题14 (2010年11月试题23)

项目时间管理包括使项目按时完成所必需的管理过程，活动定义是其中的一个重要过程。通常可以使用 (23) 来进行活动定义。

(23) A.鱼骨图 B.工作分解结构 (WBS)

C.层次分解结构 D.功能分解图

试题分析

在给定的时间完成项目是项目的重要约束性目标，能否按进度交付是衡量项目是否成功的重要标志。因此，控制进度是项目控制的首要内容，是项目的灵魂。同时，由于项目管理是一个带有创造性地过程，项目不确定性很大，控制进度是项目管理中的最大难点。

简单地理解，项目时间管理的目标就是使项目按时完成。项目时间管理的过程包过规划进度管理、定义活动、排列活动顺序、估算活动资源、估算活动持续时间、制定进度计划、控制进度。

定义活动 (Define Activity) 是识别和记录为完成项目可交付成果而需采取的具体行动的过程，其主要作用是将工作包分解为活动，作为对项目工作进行估算、进度规划、执行、监督和控制的基础。创建WBS过程已经识别出WBS中底层的可交付成果，即工作包。工作包通常还应进一步细分为更小的组成部分，即活动，代表着为完成工作包所需的工作投入。活动是为完成工作包而必须开展的工作，是开展估算、编制进度计划以及执行和监控项目工作的基础。

WBS中的每个工作包都需要分解成活动，以便通过这些活动来完成相应的可交付成果。定义活动的主要工具和技术有分解、滚动式规划和专家判断等。随着项目的进展，WBS和WBS词典反映了定义活动是一个越来越详细的演变过程。滚动式规划是规划逐步完善的一种表现形式，近期要完成的工作在WBS最下层详细规划，而计划在远期完成的工作则表现在WBS的较高层次上。

定义活动过程的结果主要是活动清单、活动属性和里程碑清单。

(1) 活动清单。活动清单 (Activity List) 是一份包含项目所需的全部进度活动的清单。活动清单中应该包括每个活动的标志和足够详细的工作描述，使项目团队成员知道应当完成哪些工作。

(2) 活动属性。活动属性 (Activity Attribute) 是指每项活动所具有的多种属性，用来扩展对该活动的描述。活动属性随时间演进。在项目初始阶段，活动属性包括活动标识、WBS标识和活动名称；当活动完成时，活动属性则可能还包括活动编码 (Activity Code) 、活动描述、紧前活动 (Predecessor Activity) 、紧后活动 (Successor Activity) 、逻辑关系 (Logical Relationship) 、时间提前与滞后量、资源需求、强制日期 (Imposed Date) 、制约因素和假设条件。活动属性还可用于识别工作执行负责人、实施工作的地区或地点，编制开展活动的项目日历，以及明确活动的类型，例如，支持型活动、独立型活动和依附型活动。

活动属性可用于编制进度计划。还可基于活动属性，在项目报告中以各种方式对进度活动进行选择、排序和分类。活动属性的数量因应用领域而异。

(3) 里程碑清单。里程碑是项目中的重要时点或事件。里程碑清单列出了项目的所有里程碑，并指明每个里程碑是强制性的 (例如，合同要求的) 还是选择性的 (例如，根据历史信息确定的) 。要注意的是，里程碑与常规的进度活动类似，有相同的结构和属性，但是里程碑的持续时间为零，因为里程碑代表的只是一个时间点。

试题答案

(23) B

试题15 (2010年11月试题24)

试题15 (2010年11月试题24)

在实际的项目开发中，人们总是希望使用自动工具来执行需求变更控制过程。下列描述中，

(24) 不是这类工具所具有的功能。

- (24) A.可以定义变更请求的数据项以及变更请求生存期的状态转换图
- B.记录每一种状态变更的数据，确认做出变更的人员
- C.可以加强状态转换图使经授权的用户仅能做出所允许的状态变更
- D.定义变更控制计划，并指导设计人员按照所制定的计划实施变更

试题分析

对许多项目来说，系统软件总要不断完善，一些需求的改进是合理的而且不可避免，要使得软件需求完全不变更，也许是不可能的，但毫无控制地变更是项目陷入混乱、不能按进度完成或者软件质量无法保证的主要原因之一。

一个好的变更控制过程，给项目风险承担者提供了正式的建议需求变更机制。可以通过需求变更控制过程来跟踪已建议变更的状态，使已建议的变更确保不会丢失或疏忽。在实际中，人们总是希望使用自动工具来执行变更控制过程。有许多人使用商业问题跟踪工具来收集、存储、管理需求变更；可以使用工具对一系列最近提交的变更建议产生一个列表给变更控制委员会开会时做议程用。问题跟踪工具也可以随时按变更状态分类统计变更请求的数目。

挑选工具时可以考虑以下几个方面：

- ① 可以定义变更请求的数据项。
- ② 可以定义变更请求生存期的状态转换图。
- ③ 可以加强状态转换图，使经授权的用户仅能做出所允许的状态变更。
- ④ 记录每一种状态变更的数据，确认做出变更的人员。
- ⑤ 可以定义在提交新请求或请求状态被更新后应该自动通知的设计人员。
- ⑥ 可以根据需要生成标准的或定制的报告和图表。

试题答案

(24) D

第 7 章：系统开发基础

试题16 (2010年11月试题25)

试题16 (2010年11月试题25)

需求管理是CMM可重复级中的6个关键过程域之一，其主要目标是 (25) .

- (25) A.对于软件需求，必须建立基线以进行控制，软件计划、产品和活动必须与软件需求保持一致
- B.客观地验证需求管理活动符合规定的标准、程序和要求
- C.策划软件需求管理的活动，识别和控制已获取的软件需求

D.跟踪软件需求管理的过程、实际结果和执行情况

试题分析

过程能力成熟度模型 (Capability Maturity Model,CMM) 在软件开发机构中被广泛用来指导软件过程改进。该模型描述了软件能力的5个成熟级别，每一级都包含若干关键过程域 (Key Process Areas,KPA) 。

CMM的第二级为可重复级，它包括了6个关键过程域，分别是：需求管理、软件项目计划、软件项目跟踪和监督、软件分包合同管理、软件质量保证和软件配置管理。

需求管理的目标是为软件需求建立一个基线，提供给软件工程和管理使用；软件计划、产品和活动与软件需求保持一致。

试题答案

(25) A

第 7 章：系统开发基础

试题17 (2010年11月试题26~27)

试题17 (2010年11月试题26~27)

在RUP中采用"4+1"视图模型来描述软件系统的体系结构。在该模型中，最终用户侧重于

(26) ,系统工程师侧重于 (27) .

(26) A.实现视图 B.进程视图 C.逻辑视图 D.部署视图

(27) A.实现视图 B.进程视图 C.逻辑视图 D.部署视图

试题分析

在RUP中采用"4+1"视图模型来描述软件系统的体系结构。"4+1"视图包括逻辑视图、实现视图、进程视图、部署视图和用例视图。

分析人员和测试人员关心的是系统的行为，因此会侧重于用例视图；最终用户关心的是系统的功能，因此会侧重于逻辑视图；程序员关心的是系统的配置、装配等问题，因此会侧重于实现视图；系统集成人员关心的是系统的性能、可伸缩性、吞吐率等问题，因此会侧重于进程视图；系统工程师关心的是系统的发布、安装、拓扑结构等问题，因此会侧重于部署视图。

试题答案

(26) C (27) D

第 7 章：系统开发基础

试题18 (2010年11月试题28)

试题18 (2010年11月试题28)

(28) 把整个软件开发流程分成多个阶段，每一个阶段都由目标设定、风险分析、开发和有效性验证以及评审构成。

- (28) A.原型模型 B.瀑布模型 C.螺旋模型 D.V模型

试题分析

原型模型又称快速原型。原型模型主要有两个阶段：① 原型开发阶段。软件开发人员根据用户提出的软件系统的定义，快速地开发一个原型。该原型应该包含目标系统的关键问题和反映目标系统的大致面貌，展示目标系统的全部或部分功能、性能等。② 目标软件开发阶段。在征求用户对原型的意见后对原型进行修改完善，确认软件系统的需求并达到一致的理解，进一步开发实际系统。

瀑布模型可以说是最早使用的软件生存周期模型之一。由于这个模型描述了软件生存的一些基本过程活动，所以它被称为软件生存周期模型。这些活动从一个阶段到另一个阶段逐次下降，形式上很像瀑布。瀑布模型的特点是因果关系紧密相连，前一个阶段工作的结果是后一个阶段工作的输入。

螺旋模型是在快速原型的基础上扩展而成的。这个模型把整个软件开发流程分成多个阶段，每个阶段都由4部分组成，它们是：①目标设定。为该项目进行需求分析，定义和确定这一个阶段的专门目标，指定对过程和产品的约束，并且制定详细的管理计划。②风险分析。对可选方案进行风险识别和详细分析，制定解决办法，采取有效的措施避免这些风险。③开发和有效性验证。风险评估后，可以为系统选择开发模型，并且进行原型开发，即开发软件产品。④评审。对项目进行评审，以确定是否需要进入螺旋线的下一次回路，如果决定继续，就要制定下一阶段计划。

V模型是一种典型的测试模型。在V模型中测试过程被加在开发过程的后半部分，分别包括单元测试、集成测试、系统测试和验收测试。

试题答案

- (28) C

第 7 章：系统开发基础

试题19 (2010年11月试题29~30)

试题19 (2010年11月试题29~30)

软件开发环境是支持软件产品开发的软件系统，它由软件工具集和环境集成机制构成。环境集成机制包括：提供统一的数据模式和数据接口规范的数据集成机制；支持各开发活动之间通信、切换、调度和协同的（29）；为统一操作方式提供支持的（30）。

- (29) A.操作集成机制 B.控制集成机制 C.平台集成机制 D.界面集成机制

- (30) A.操作集成机制 B.控制集成机制 C.平台集成机制 D.界面集成机制

试题分析

软件开发环境应支持多种集成机制，例如，平台集成、数据集成、界面集成、控制集成和过程集成等。软件开发环境应支持小组工作方式，并为其提供配置管理，环境的服务可用于支持各种软件开发活动，包括分析、设计、编程、调试和文档等。

较完善的软件开发环境通常具有多种功能，例如，软件开发的一致性与完整性维护，配置管理及版本控制，数据的多种表示形式及其在不同形式之间的自动转换，信息的自动检索与更新，项目控制和管理，以及对开发方法学的支持。软件开发环境具有集成性、开放性、可裁减性、数据格式一致性、风格统一的用户界面等特性，因而能大幅度提高软件生产率。

1. 软件开发环境的分类

软件开发环境可按以下几种角度进行分类：

(1) 按软件开发模型与开发方法分类，有支持瀑布模型、演化模型、螺旋模型和喷泉模型等不同模型，以及结构化方法、面向对象方法等不同方法的软件开发环境。

(2) 按功能与结构特点分类，有单体型、协同型、分散型和并发型等多种类型的软件开发环境。

(3) 按应用范围分类，有通用型和专用型软件开发环境。其中专用型软件开发环境与应用领域有关。

(4) 按开发阶段分类，有前端开发环境（支持系统规划、分析、设计等阶段的活动）、后端开发环境（支持编程、测试等阶段的活动）、软件维护环境和逆向工程环境等。

2. 集成机制

集成机制根据功能的不同，可划分为环境信息库、过程控制与消息服务器、环境用户界面三个部分。

(1) 环境信息库。环境信息库是软件开发环境的核心，用以存储与系统开发有关的信息，并支持信息的交流与共享。环境信息库中主要存储两类信息，一类是开发过程中产生的有关被开发系统的信息，例如，分析文档、设计文档和测试报告等；另一类是环境提供的支持信息，例如，文档模板、系统配置、过程模型和可复用构件等。

(2) 过程控制与消息服务器。过程控制与消息服务器是实现过程集成和控制集成的基础。过程集成是按照具体软件开发过程的要求进行工具的选择与组合，控制集成使各工具之间进行并行通信和协同工作。

(3) 环境用户界面。环境用户界面包括环境总界面和由它实行统一控制的各环境部件及工具的界面。统一的、具有一致性的用户界面是软件开发环境的重要特征，是充分发挥环境的优越性、高效地使用工具并减轻用户的学习负担的保证。

3. 集成计算机辅助软件工程

目前，随着软件开发工具的积累与自动化工具的增多，软件开发环境已经进入了第三代，即ICASE (Integrated Computer-Aided Software Engineering, 集成计算机辅助软件工程) 阶段。集成方式经历了从点到点的数据转换（早期CASE采用的集成方式），到公共用户界面（第二代CASE，在一致的界面下调用众多不同的工具），再到目前的信息库方式。这是ICASE的主要集成方式。ICASE不仅提供数据集成和控制集成，还提供了一组用户界面管理设施和一大批工具，包括垂直工具集（支持软件生命周期各阶段，保证生成信息的完备性和一致性）、水平工具集（用于不同的软件开发方法）和开放工具槽（用于连接新的工具）。

ICASE的信息库不仅定义了面向对象的数据库管理系统，提供了数据-数据集成机制，还建立了可以被环境中所有工具访问的数据模型，提供了数据-工具集成机制，实现了配置管理功能。ICASE的进一步发展则是与软件开发方法的结合，以及智能化的ICASE。

ICASE的最终目标是实现应用软件的全自动开发，即开发人员只要写好软件的需求规格说明书，ICASE就能自动完成软件开发工作，即自动生成供用户直接使用的软件和有关文档。

试题答案

(29) B (30) D

第 7 章：系统开发基础

试题20 (2010年11月试题31)

试题20 (2010年11月试题31)

软件的横向重用是指重用不同应用领域中的软件元素。 (31) 是一种典型的、原始的横向重用机制。

- (31) A.对象 B.构件 C.标准函数库 D.设计模式

试题分析

软件重用是指在两次或多次不同的软件开发过程中重复使用相同或相似软件元素的过程。按照重用活动是否跨越相似性较少的多个应用领域，软件重用可以区别为横向重用和纵向重用。横向重用是指重用不同应用领域中的软件元素，例如数据结构、分类算法和人机界面构建等。标准函数是一种典型的、原始的横向重用机制。纵向重用是指在一类具有较多公共性的应用领域之间进行软部件重用。纵向重用活动的主要关键点是域分析：根据应用领域的特征及相似性预测软部件的可重用性。

试题答案

- (31) C

第 7 章：系统开发基础

试题21 (2010年11月试题32)

试题21 (2010年11月试题32)

下列关于不同软件开发方法所使用的模型的描述中，正确的是 (32) .

- (32) A.在进行结构化分析时，必须使用数据流图和软件结构图这两种模型
B.采用面向对象开发方法时，可以使用状态图和活动图对系统的动态行为进行建模
C.实体联系图 (E-R图) 是在数据库逻辑结构设计时才开始创建的模型
D.UML的活动图与程序流程图的表达能力等价

试题分析

结构化分析方法是一种面向数据流的需求分析方法，其基本思想是自顶向下逐层分解。数据流图是进行结构化分析时所使用的模型，其基本成分包括数据流、加工、数据存储和外部实体。在进行结构化设计时，通过对数据流图进行变换分析和事务分析可以导出程序结构图。

数据库设计可以分为以下4个主要阶段。

① 用户需求分析。数据库设计人员采用一定的辅助工具对应用对象的功能、性能、限制等要求所进行的科学分析。

② 概念设计。概念结构设计是对信息分析和定义，如视图模型化、视图分析和汇总。对应用对象精确地抽象、概括而形成的独立于计算机系统的企业信息模型。描述概念模型的较理想的工具是E-R图。

③ 逻辑设计。将抽象的概念模型转化为与选用的DBMS产品所支持的数据模型相符合的逻辑模型，它是物理设计的基础。包括模式初始设计、子模式设计、应用程序设计、模式评价以及模式求精。

④ 物理设计。逻辑模型在计算机中的具体实现方案。

UML是面向对象软件的标准化建模语言，其中状态图、活动图、顺序图和通信图可以用来对系统的动态行为进行建模。活动图展现了在系统内从一个活动到另一个活动的流程。活动图强调对象之间的控制流程。在活动图上可以表示分支和汇合。活动图与传统的程序流程图是不等价的。

试题答案

(32) B

第 7 章：系统开发基础

试题22 (2010年11月试题41)

试题22 (2010年11月试题41)

系统输入设计中，采用内部控制方式以确保输入系统数据的有效性，(41) 用于验证数据是否位于合法的取值范围。

(41) A.数据类型检查 B.自检位 C.域检查 D.格式检查

试题分析

系统输入设计中，通常通过内部控制的方式验证输入数据的有效性。数据类型检查确保输入了正确的数据类型；自检位用于对主关键字进行基于校验位的检查；域检查用于验证数据是否位于合法的取值范围；格式检查按照已知的数据格式对照检查输入数据的格式。

试题答案

(41) C

第 7 章：系统开发基础

试题23 (2010年11月试题42~43)

试题23 (2010年11月试题42~43)

系统测试由若干个不同的测试类型组成，其中（42）检查系统能力的最高实际限度，即软件在一些超负荷情况下的运行情况；（43）主要是检查系统的容错能力。

（42）A.强度测试 B.性能测试 C.恢复测试 D.可靠性测试

（43）A.强度测试 B.性能测试 C.恢复测试 D.可靠性测试

试题分析

系统测试是将已经确认的软件、计算机硬件、外设和网络等其他因素结合在一起，进行信息系统的各种组装测试和确认测试，其目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方。系统测试是根据系统方案说明书来设计测试例子的，常见的系统测试主要有以下内容。

恢复测试：恢复测试监测系统的容错能力。检测方法是采用各种方法让系统出现故障，检验系统是否按照要求能从故障中恢复过来，并在约定的时间内开始事务处理，而且不对系统造成任何伤害。如果系统的恢复是自动的（由系统自动完成），需要验证重新初始化、检查点、数据恢复等是否正确。如果恢复需要人工干预，就要对恢复的平均时间进行评估并判断它是否在允许的范围内。

安全性测试：系统的安全性测试是检测系统的安全机制、保密措施是否完善，主要是为了检验系统的防范能力。测试的方法是测试人员模拟非法入侵者，采用各种方法冲破防线。系统安全性设计准则是使非法入侵者所花费的代价比进入系统后所得到的好处要大，此时非法入侵已无利可图。

强度测试：是对系统在异常情况下的承受能力的测试，是检查系统在极限状态下运行时，性能下降的幅度是否在允许的范围内。因此，强度测试要求系统在非正常数量、频率或容量的情况下运行。强度测试主要是为了发现在有效的输入数据中可能引起不稳定或不正确的数据组合。例如，运行使系统处理超过设计能力的最大允许值的测试例子；使系统传输超过设计最大能力的数据，包括内存的写入和读出等。

性能测试：检查系统是否满足系统设计方案说明书对性能的要求。性能测试覆盖了软件测试的各阶段，而不是等到系统的各部分所有都组装之后，才确定系统的真正性能。通常与强度测试结合起来进行，并同时对软件、硬件进行测试。软件方面主要从响应时间、处理速度、吞吐量、处理精度等方面来检测。

可靠性测试：通常使用以下两个指标来衡量系统的可靠性：平均失效间隔时间MTBF（mean time between failures）是否超过了规定的时限，因故障而停机时间MTTR（mean time to repairs）在一年中不应超过多少时间。

安装测试：在安装软件系统时，会有多种选择。安装测试就是为了检测在安装过程中是否有误、是否容易操作等。主要监测系统的每一个部分是否齐全，硬件的配置是否合理，安装中需要产生的文件和数据库是否已经产生，其内容是否正确等。

试题答案

（42）A（43）C

试题24 (2011年11月试题22)

软件产品配置是指一个软件产品在生存周期各个阶段所产生的各种形式和各种版本的文档、计算机程序、部件及数据的集合。该集合的每一个元素称为该产品配置中的一个配置项。下列不应该属于配置项的是 (22) 。

- (22) A.源代码清单B.设计规格说明书
- C.软件项目实施计划D.CASE工具操作手册

试题分析

源代码清单、设计规格说明书、软件项目实施计划均可以成为配置项。而CASE工具操作手册是指导开发人员使用CASE工具来做开发的一个说明文档，它与软件产品并无直接关联，不宜作为配置项。

试题答案

- (22) D

第 7 章：系统开发基础

试题25 (2011年11月试题23)

试题25 (2011年11月试题23)

软件质量保证是软件项目控制的重要手段，(23) 是软件质量保证的主要活动之一。

- (23) A.风险评估B.软件评审C.需求分析D.架构设计

试题分析

软件质量管理过程由许多活动组成，一些活动可直接发现缺陷，其他活动则指出深入的检查是否有价值，前者也称为直接缺陷发现活动，许多活动都可以达到这两个目的。这些活动包括质量保证过程、验证过程、确认过程、评审过程、审计过程等。

1. 软件质量保证

软件质量保证过程通过计划制订、实施和完成一组活动提供保证，这些活动保证项目生命周期中的软件产品和过程符合其规定的需求。

软件质量保证计划定义了用于保证为特定产品开发的软件满足用户需求并在项目的约束内具有最高的质量的手段。

2. 验证与确认

验证和确认过程使用能够定位缺陷并便于以后改正的测试技术直接处理软件产品质量问题。验证是指在软件开发周期中的一个给定阶段的产品是否达到在上一阶段确立的需求的过程；确认是指在软件开发过程结束时对软件进行评价以确定它是否和软件需求相一致的过程。

3. 评审与审计

评审与审计过程包括管理评审、技术评审、检查、走查、审计等。

管理评审的目的是监控进展，决定计划和进度的状态，确认需求及其系统分配，或评价用于达到目标适应性的管理方法的有效性。它们支持有关软件项目期间需求的变更和其他变更活动；技术评审的目的是评价软件产品，以确定其对使用意图的适合性，目标是识别规范说明和标准的差异，

并向管理提供证据，以表明产品是否满足规范说明并遵从标准，而且可以控制变更。有关管理评审与技术评审的详细知识，将在13.7节介绍。

检查的目的是检测和识别软件产品异常。一次检查通常针对产品的一个相对小的部分。发现的任何异常都要记录到文档中并提交；走查的目的是评价软件产品，走查也可以用于培训软件产品的听众，主要目标是发现异常、改进软件产品、考虑其他实现、评价是否遵从标准和规范说明。走查类似于检查，但通常不那么正式。走查通常主要由同事评审其工作，以作为一种保障技术。

软件审计的目的是提供软件产品和过程对于可应用的规则、标准、指南、计划和流程的遵从性的独立评价。审计是正式组织的活动，识别违例情况，并产生一个报告，采取更正性行动。

试题答案

(23) B

第 7 章：系统开发基础

试题26 (2011年11月试题24)

试题26 (2011年11月试题24)

利用需求跟踪能力链 (traceability link) 可以跟踪一个需求使用的全过程，也就是从初始需求到实现的前后生存期。需求跟踪能力链有4类，如图7-5所示。

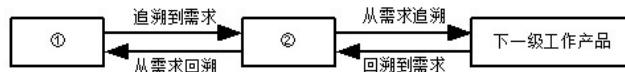


图7-5 需求跟踪能力链示意图

其中的①和②分别是 (24) .

- (24) A.客户需求、软件需求B.软件需求、客户需求
C.客户需求、当前工作产品D.软件需求、当前工作产品

试题分析

需求跟踪是重要的项目需求管理方法，这种方法为项目组织提供了在投资方需求、需求规格说明书、项目产品之间保持一致性的能力。需求跟踪可以改善产品的质量，降低维护成本，而且能提高需求定义、项目产品构件的可重用性。跟踪需求的过程主要包括以下内容：

- (1) 从需求到业务需要、机会、目的和目标；
- (2) 从需求到项目目标；
- (3) 从需求到项目范围 (WBS) 中的可交付成果；
- (4) 从需求到产品设计；
- (5) 从需求到产品开发；
- (6) 从需求到测试策略和测试脚本；
- (7) 从宏观需求到详细需求。

2. 需求跟踪能力链

根据国家标准GB/T 8567-2006,需求规格说明书中的每个配置项的需求到其涉及的系统 (或子系统) 需求都要具有双向可追踪性。所谓双向跟踪，包括正向跟踪和反向跟踪，正向跟踪是指检查

需求规格说明书中的每个需求是否都能在后继工作成果中找到对应点；反向跟踪也称为逆向跟踪，是指检查设计文档、代码、测试用例等工作成果是否都能在需求规格说明书中找到出处。具体来说，需求跟踪涉及五种类型，如图7-6所示。

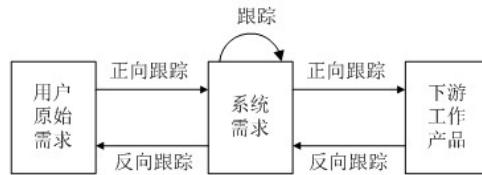


图7-6 五类需求可跟踪

图7-6中的箭头表示需求跟踪能力联系链，它能跟踪需求使用的整个周期，即从需求建议到交付的全过程。

图7-6的左半部分表明，从用户原始需求可向前追溯到系统需求，这样就能区分出开发过程中或开发结束后由于变更受到影响的需求，也确保了需求规格说明书中包括所有用户需求。同样，可以从系统需求回溯到相应的用户原始需求，确认每个系统需求的出处。如果以用例的形式来描述用户需求，图7-6的左半部分就是用例和功能性需求之间的跟踪情况。

图7-6的右半部分表明，由于在开发过程中，系统需求转变为设计和编码等实现元素，所以通过定义单个系统需求和特定的产品元素之间的联系链，可以从系统需求追溯到产品元素。这种联系链使开发人员知道每个需求对应的产品元素，从而确保产品元素满足每个需求。第四类联系链是从产品元素回溯到系统需求，使开发人员知道每个产品元素存在的原因。绝大多数项目不包括与用户需求直接相关的代码，但开发人员应该知道为什么要写这一行代码。如果不能将设计元素、代码段或测试用例回溯到一个系统需求，就可能出现画蛇添足的现象。当然，如果某个孤立的产品元素表明了一个正当的功能，则说明需求规格说明书漏掉了一项需求。

第五类联系链是系统需求之间的跟踪，这种跟踪便于更好地处理系统需求之间的逻辑相关性，检查需求分解中可能出现的错误或遗漏。

3. 需求跟踪的目的

需求跟踪是一项劳动强度很大的任务，在整个系统开发、运行和维护的过程中，要始终保持联系链信息与实际相符。在项目实践中，使用需求跟踪能力，可以获得如下好处：

- (1) 审核。跟踪能力信息可以帮助开发人员审核和确保所有需求都被正确应用。
- (2) 变更影响分析。在增、删、改需求时，跟踪能力信息可以确保不忽略每个受到影响的系统元素。
- (3) 维护。可靠的跟踪能力信息使得维护时能够正确而完整地实施变更，从而提高生产率。
- (4) 项目跟踪。认真记录跟踪能力数据，就可以获得计划功能当前实现状态的记录。
- (5) 再工程。可以列出遗留系统中将要替换的功能，记录它们在新系统中的需求和在构件中的位置。
- (6) 重复利用。跟踪能力信息可以帮助开发人员在新系统中对相同的功能利用现有系统的相关资源。例如，功能设计、相关需求、代码和测试等。
- (7) 减小风险。需求联系文档化可减少由于项目团队关键成员离职带来的风险。
- (8) 测试。测试模块、需求和代码段之间的联系链可以在测试出错时指出最可能有问题的代码段。

4. 需求跟踪矩阵

表示需求和其他系统元素之间的联系链的最普遍方式是使用需求跟踪（能力）矩阵。不论采用何种跟踪方式，都要建立与维护需求跟踪矩阵，它保存了需求与后继工作成果的对应关系。例如，从用户原始需求到系统需求之间的跟踪，可以采用如表7-2所示的矩阵。

表7-2 用户原始需求到系统需求的跟踪矩阵示例

用例 原始需求	UC-1	UC-2	UC-3	UC-n
FR-1					
FR-2					
.....					
FR-m					

对于从系统需求到下游工作产品之间的跟踪，可以采用如表7-3所示的矩阵。

表7-3 系统需求到下游工作产品的跟踪矩阵示例

元素 用例	功能点	设计元素	代码模块	测试用例
UC-1				
UC-2				
.....				
UC-n				

表7-3明确展示了每个用例是如何连接到一个或多个设计、编码和测试元素的。其中设计元素可以是模型中的对象，例如，数据流图、实体联系图或类图等；代码模块可以是类中的方法、源代码文件名、过程或函数。需求跟踪矩阵中可以定义各种系统元素类型间的一对一、一对多和多对多关系，也就是说，允许在表7-3的一个单元格中填入多个元素来实现这些特征。例如，一个代码模块对应一个设计元素，多个测试用例验证一个功能点，每个用例导致多个功能点等。

试题答案

(24) A

第 7 章：系统开发基础

试题27 (2011年11月试题25)

试题27 (2011年11月试题25)

通常有两种常用的需求定义方法：严格定义方法和原型方法。下述的各种假设条件中，"

(25) "不适合使用严格定义方法进行需求定义。

- (25) A.所有需求都能够被预先定义
- B.开发人员与用户之间能够准确而清晰地交流
- C.需求不能在系统开发前被完全准确地说明
- D.采用图形（或文字）充分体现最终系统

试题分析

需求定义的过程也就是形成需求规格说明书的过程，通常有两种需求定义的方法，分别是严格定义方法和原型方法。

1. 严格定义方法

严格定义也称为预先定义，需求的严格定义建立在以下的基本假设之上：

(1) 所有需求都能够被预先定义。假设意味着，在没有实际系统运行经验的情况下，全部的系统需求均可通过逻辑推断得到。这对某些规模较小、功能简单的系统是可能的，但对那些功能庞大、复杂且较大的系统显然是困难的。即使事先做了深入细致的调查和分析，当用户见到新系统的实际效果时，也往往会改变原先的看法，会提出修改或更进一步增加系统功能的要求，因此，再好的预先定义技术也会经常反复。这是因为人们对新事物的认识与理解将随着直观、实践的过程进一步加深，这是与人类认识世界的客观规律相一致的。所以，能够预先定义出所有需求的假设在许多场合是不能成立的。

(2) 开发人员与用户之间能够准确而清晰地交流。假设认为，用户与开发人员之间，虽然每人都有自己的专业、观点和行话，但在系统开发过程中可以使用图形（或文本）等通信工具进行交流，进行清晰且有效的沟通，这种沟通是必不可少的。遗憾的是，在实际开发中，往往对一些共同的约定，每个人可能都会产生自己的理解和解释。即使采用结构化语言、判定树和判定表等工具，仍然存在不精确和技术上的不严密。这将导致人们有意无意地带有个人的不同理解而各行其事，因此，在多学科、多行业人员之间进行有效的通信和交流是有一定困难的。

(3) 采用图形（或文字）可以充分体现最终系统。在使用严格定义需求的开发过程中，开发人员与用户之间交流与通信的主要工具是定义报告，包括叙述文字、图形、逻辑规则和数据字典等技术工具。它们都是静止的、被动的，不能实际演示，很难在用户头脑中形成一个具体的形象。因此，要用静止的图形（或文字）描述来体现一个动态的系统是比较困难的。

由此可见，严格定义法的基本假设在许多情况下并不成立，传统的结构化方法面临着一些难以跨越的障碍。为此，需要探求一种变通的方法。

2. 原型方法

原型方法以一种与严格定义法截然不同的观点看待需求定义问题。原型化的需求定义过程是一个开发人员与用户通力合作的反复过程。从一个能满足用户基本需求的原型系统开始，允许用户在开发过程中提出更好的要求，根据用户的要求不断地对系统进行完善，它实质上是一种迭代的循环型的开发方式。采用原型方法时需要注意以下几个问题：

(1) 并非所有的需求都能在系统开发前被准确地说明。事实上，要想严密、准确地定义任何事情都是有一定难度的，更不用说是定义一个庞大系统的全部需求。用户虽然可以叙述他们所需最终系统的目标和大致功能，但是对某些细节问题却往往不可能十分清楚。一个系统的开发过程，无论对于开发人员还是用户来说，都是一个学习和实践的过程，为了帮助他们在这个过程中提出更完善的需求，最好的方法就是提供现实世界的实例--原型，对原型进行研究和实践，并进行评价。

(2) 项目干系人之间通常都存在交流上的困难，原型提供了克服该困难的一个手段。用户和开发人员通过屏幕和键盘进行对话、讨论和交流，从他们自身的理解出发来测试原型。原型系统由于直观性和动态性，而使得项目干系人之间的交流上的困难得到较好的克服。

(3) 需要实际的、可供用户参与的系统模型。虽然图形和文字描述是一种较好的通信交流工具，但是，其最大缺陷是缺乏直观的和感性的特征，因此不易理解对象的全部含义。交互式的系统原型能够提供生动的需求规格说明，用户见到的是一个“活”的和实际运行着的系统。实际使用在计算机上运行的系统，显然比理解纸面上的系统要深刻得多。

(4) 有合适的系统开发环境。随着计算机硬件、软件技术和软件工具的迅速发展，软件的设计与实现工作越来越方便，对系统进行局部性修改甚至重新开发的代价大大降低。因此，对大系统的原型化已经成为可能。

(5) 反复是完全需要和值得提倡的，需求一旦确定，就应遵从严格的方法。系统分析师应该鼓励用户改进他们的系统，只有做必要的改变后，才能使用户和系统间获得更加良好的匹配。所以，

从某种意义上说，严格定义需求的方法实际上抑制了用户在需求定义以后再改进的要求，这对提高最终系统的质量是有害的。另一方面，原型方法的使用并不排除严格定义方法的运用，当通过原型在演示中得到明确的需求定义后，应采用行之有效的严格方法来完成最终系统的开发。

试题答案

(25) C

第 7 章：系统开发基础

试题28 (2011年11月试题26)

试题28 (2011年11月试题26)

下列关于软件需求管理或需求开发的叙述中，正确的是 (26) .

- (26) A. 所谓需求管理是指对需求开发的管理
- B. 需求管理包括：需求获取、需求分析、需求定义和需求验证
- C. 需求开发是将用户需求转化为应用系统成果的过程
- D. 在需求管理中，要求维持对用户原始需求和所有产品构件需求的双向跟踪

试题分析

需求工程是包括创建和维护系统需求文档所必需的一切活动的过程，可分为需求开发和需求管理两大工作。

(1) 需求开发：包括需求获取、需求分析、编写规格说明书（需求定义）和需求验证4个阶段。在需求开发阶段需要确定产品所期望的用户类型、获取每种用户类型的需求、了解实际的用户任务和目标，以及这些任务所支持的业务需求。同时还包括分析源于用户的信息、对需求进行优先级分类、将所收集的需求编写成为软件规格说明书和需求分析模型，以及对需求进行评审等工作。

(2) 需求管理：通常包括定义需求基线、处理需求变更及需求跟踪等方面的工作。

这两个方面是相辅相成的，需求开发是主线，是目标；需求管理是支持，是保障。换句话来说，需求开发是努力更清晰、更明确地掌握客户对系统的需求；而需求管理则是对需求的变化进行管理的过程。

试题答案

(26) D

第 7 章：系统开发基础

试题29 (2011年11月试题27~28)

试题29 (2011年11月试题27~28)

RUP是一个二维的软件开发模型，其核心特点之一是（27）。RUP将软件开发生命周期划分为多个循环（cycle），每个循环由4个连续的阶段组成，每个阶段完成确定的任务。设计及确定系统的体系结构，制定工作计划及资源要求是在（28）阶段完成的。

（27）A.数据驱动 B.模型驱动 C.用例驱动 D.状态驱动

（28）A.初始（inception） B.细化（elaboration）

C.构造（construction） D.移交（transition）

试题分析

RUP（Rational Unified Process）是Rational公司开发和维护的过程产品，是由Objectory过程演化而来。RUP将项目管理、业务建模、分析与设计等统一起来，贯穿整个开发过程。RUP采用Internet技术，可以增强团队的开发效率，并为所有成员提供最佳的软件实现方案，它使团队中每个开发人员的见解和思想得到统一，使开发小组成员的沟通更为容易，而这正是任何项目要取得成功的关键因素。RUP可以增强开发人员对软件的预见性，最终的好处就是提高了软件质量，并有效缩短了软件从开发到投放市场的时间。RUP过程为软件开发提供了规范性的指南、模板和范例，可用来开发所有类型的应用。

RUP中的软件过程在时间上被分解为四个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和移交阶段。每个阶段结束时都要安排一次技术评审，以确定这个阶段的目标是否已经满足。如果评审结果令人满意，就可以允许项目进入下一个阶段。基于RUP的软件过程模型如图7-7所示。

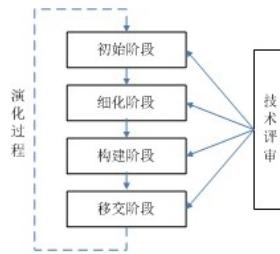


图7-7 基于RUP的软件过程

从图7-7中可以看出，基于RUP的软件过程是一个迭代过程。通过初始、细化、构建和移交四个阶段就是一个开发周期，每次经过这四个阶段就会产生一代软件。除非产品退役，否则通过重复同样的四个阶段，产品将演化为下一代产品，但每一次的侧重点都放在不同的阶段上。

用户需求的变化、运行环境的变更、基础技术方面的变更等都会引发演化过程。通常情况下，演化过程的初始阶段和细化阶段都比较简单，因为基本产品定义和架构在前面的开发过程中就已经决定。但也有例外情况，例如，对软件架构进行重新定义的演化过程。

2. 初始阶段

初始阶段的任务是为系统建立业务模型并确定项目的边界。在初始阶段，必须识别所有与系统交互的外部实体，定义系统与外部实体交互的特性。在这个阶段中，所关注的是整个项目的业务和需求方面的主要风险。对于建立在原有系统基础上的开发项目来说，初始阶段可能很短。初始阶段的实现过程如下：

（1）明确项目规模。建立项目的软件规模和边界条件，包括验收标准；了解环境及重要的需求和约束，识别系统的关键用例。

（2）评估项目风险。软件过程主要关心的是软件开发的已知方面，只能准确描述、计划、分配和评审那些已经知道将要完成的事情。风险管理则主要关心未知方面。在基于RUP的迭代式软件过程中，很多决策要受风险决定。要达到这个目的，开发人员需要详细了解项目所面临的风险，并对如何降低或处理风险有明确的策略。

(3) 制订项目计划。估计整个项目的总体成本、进度和人员配备。综合考虑备选架构，评估设计和自制/外购/复用方面的方案，从而估算出成本、进度和资源。在这个过程中，要通过对一些概念的证实来证明可行性，可以采用可模拟需求的模型形式或用于探索高风险区的初始原型。初始阶段的原型设计工作应该限制在确信解决方案可行就可以了，具体实现留到细化阶段和构建阶段。

(4) 阶段技术评审。初始阶段结束时要进行一次技术评审，检查初始阶段的目标是否完成，并决定继续进行项目还是取消项目。在评审过程中，需要考虑项目的规模定义、成本和进度估算是否适中、估算根据是否可靠、需求是否正确、开发方和用户方对软件需求的理解是否达成一致、是否已经确定所有风险且有针对每个风险的规避策略等问题。

3. 细化阶段

细化阶段的任务是分析问题领域，建立完善的架构，淘汰项目中最高风险的元素。在细化阶段，必须在理解整个系统的基础上，对架构做出决策，包括其范围、主要功能和诸如性能等非功能需求，同时为项目建立支持环境。细化阶段的实现过程如下：

(1) 确定架构。确保架构、需求和计划足够稳定，充分减少风险，从而能够有预见性地确定开发所需的成本和开发进度。通过处理架构方面重要的场景，建立一个已确定基线的架构，并验证其将在适当时间、以合理的成本支持系统需求。

(2) 制订构建阶段计划。为构建阶段制订详细的过程计划并为其建立基线。

(3) 建立支持环境。包括开发环境、开发流程、支持构建团队所需的工具和自动化/半自动化支持。

(4) 选择构件。评估现有的构件库和潜在构件，充分了解自制/外购/复用决策，以便有把握地确定构建阶段的成本和进度。集成所选构件，并按主要场景进行评估。

(5) 阶段技术评审。评审时，需要检验详细的系统目标和范围、架构的选择，以及主要风险的解决方案。

在细化阶段，可执行的原型依赖于项目的范围、规模、风险和先进程度。必须至少处理初始阶段中识别的关键用例，因为关键用例通常揭示了项目的主要技术风险。

4. 构建阶段

在构建阶段，要开发所有剩余的构件和应用程序功能，把这些构件集成为产品，并进行详细测试。从某种意义上说，构建阶段是一个制造过程，其重点放在管理资源及控制操作，以优化成本、进度和质量。构建阶段的主要任务是通过优化资源和避免不必要的报废和返工，使开发成本降到最低；完成所有所需功能的分析、开发和测试，快速完成可用的版本；确定软件、场地和用户是否已经为部署软件作好准备。

在构建阶段，开发团队的工作可以实现某种程度的并行。一些项目的规模大得足够产生许多并行的增量构建过程，即使是较小的项目，也通常包括可以相互独立开发的构件，从而使各团队之间实现并行开发。这些并行活动在加速版本发布的有效性的同时，也增加了资源管理和工作流同步的复杂性。

构建阶段结束时也要进行技术评审，评审产品是否可以在β测试环境中进行安装和运行。

5. 移交阶段

当基线已经足够完善，可以安装到最终用户实际环境中时，则进入交付阶段。交付阶段的重点是确保软件对最终用户是可用的。交付阶段的主要任务是进行β测试，制作产品发布版本；对最终用户支持文档定稿；按用户的需求确认新系统；培训用户和维护人员；获得用户对当前版本的反馈，基于反馈调整产品，例如，进行调试、性能或可用性的增强等。

交付阶段结束时也要进行技术评审，评审目标是否实现，是否应该开始演化过程，用户对交付的产品是否满意等。

RUP由于太过于庞大和复杂，相对于轻量级的敏捷方法来说，显得死板和难以实施。RUP不但不能快速适应需求的变化，而且变更一个需求要经历复杂的过程和很多额外的工作。对于较小的组织和项目来说，使用敏捷方法可能比较合适，而使用RUP似乎有些费力不讨好。

试题答案

(27) C (28) B

第 7 章：系统开发基础

试题30 (2011年11月试题29~30)

试题30 (2011年11月试题29~30)

在面向对象设计中，用于描述目标软件与外部环境之间交互的类被称为 (29)，它可以 (30)

(29) A.实体类 B.边界类 C.模型类 D.控制类

- (30) A.表示目标软件系统中具有持久意义的信息项及其操作
B.协调、控制其他类完成用例规定的功能或行为
C.实现目标软件系统与外部系统或外部设备之间的信息交流和互操作
D.分解任务并把子任务分派给适当的辅助类

试题分析

面向对象技术中的类分为三种：实体类、边界类、控制类。

实体类是用于对必须存储的信息和相关行为建模的类。实体对象（实体类的实例）用于保存和更新一些现象的有关信息，例如：事件、人员或者一些现实生活中的对象。实体类通常都是永久性的，它们所具有的属性和关系是长期需要的，有时甚至在系统的整个生存期都需要。

边界类是一种用于对系统外部环境与其内部运作之间的交互进行建模的类。这种交互包括转换事件，并记录系统表示方式（例如接口）中的变更。

常见的边界类有窗口、通信协议、打印机接口、传感器和终端。如果你在使用 GUI 生成器，你就不必将按钮之类的常规接口部件作为单独的边界类来建模。通常，整个窗口就是最精制的边界类对象。边界类还有助于获取那些可能不面向任何对象的 API（例如遗留代码）的接口。

控制类用于对一个或几个用例所特有的控制行为进行建模。控制对象（控制类的实例）通常控制其他对象，因此它们的行为具有协调性质。控制类将用例的特有行为进行封装。

试题答案

(29) B (30) C

试题31 (2011年11月试题31)

试题31 (2011年11月试题31)

最少知识原则（也称为迪米特法则）是面向对象设计原则之一，是指一个软件实体应当尽可能少地与其他实体发生相互作用。这样，当一个实体被修改时，就会尽可能少地影响其他的实体。下列叙述中，“（31）”不符合最少知识原则。

- (31) A.在类的划分上，应当尽量创建松耦合的类
- B.在类的设计上，只要有可能，一个类型应当设计成不变类
- C.在类的结构设计上，每个类都应当尽可能提高对其属性和方法的访问权限
- D.在对其他类的引用上，一个对象对其他对象的引用应当降到最低

试题分析

面向对象设计原则包括如下。

单一职责原则：设计目的单一的类。

开放-封闭原则：对扩展开放，对修改封闭。

李氏（Liskov）替换原则：子类可以替换父类。

依赖倒置原则：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程。

接口隔离原则：使用多个专门的接口比使用单一的总接口要好。

组合重用原则：要尽量使用组合，而不是继承关系达到重用目的。

迪米特（Demeter）原则（最少知识法则）：一个对象应当对其他对象有尽可能少的了解，其中迪米特原则的主要理念是让一个对象尽可能少地了解其他对象，这样，就能尽可能少地产生违规操作，让设计出来的系统更稳定。在本题中，C选项提到“尽可能提高对其属性和方法的访问权限”违背了迪米特原则。

试题答案

- (31) C

试题32 (2011年11月试题32)

试题32 (2011年11月试题32)

下列关于各种软件开发方法的叙述中，错误的是（32）。

- (32) A.结构化开发方法的缺点是开发周期较长，难以适应需求变化
- B.可以把结构化方法和面向对象方法结合起来进行系统开发，使用面向对象方法进行自顶向下的划分，自底向上地使用结构化方法开发系统
- C.与传统方法相比，敏捷开发方法比较适合需求变化较大或者开发前期需求不是很清晰的项目，以它的灵活性来适应需求的变化

D.面向服务的方法以粗粒度、松散耦合和基于标准的服务为基础，增强了系统的灵活性、可复用性和可演化性

试题分析

B选项中"自底向上地使用结构化方法开发系统"显然是错误的，因为结构化方法的一个核心特色为："自顶向下，逐步求精",而非自底向上。

试题答案

(32) B

第 7 章：系统开发基础

试题33 (2011年11月试题39)

试题33 (2011年11月试题39)

系统设计是软件开发的重要阶段，(39)主要是按系统需求说明来确定此系统的软件结构，并设计出各个部分的功能和接口。

- (39) A.外部设计 B.内部设计
- C.程序设计 D.输入/输出设计

试题分析

在软件开发中，概要设计又称为外部设计，是在软件需求规格说明的基础上，建立系统的总体结构（含子系统的划分）和模块间的关系，定义功能模块及各功能模块之间的关系。

详细设计也称为内部设计，是对概要设计产生的功能模块逐步细化，把模块内部细节转化为可编程的程序过程性描述。详细设计包括算法与数据结构、数据分布、数据组织、模块间接口信息、用户界面等的设计，并写出详细设计报告。

试题答案

(39) A

第 7 章：系统开发基础

试题34 (2011年11月试题40)

试题34 (2011年11月试题40)

快速迭代式的原型开发能够有效控制成本，(40)是指在开发过程中逐步改进和细化原型直至产生出目标系统。

- (40) A.可视化原型开发 B.抛弃式原型开发
- C.演化式原型开发 D.增量式原型开发

试题分析

通常，原型是指模拟某种产品的原始模型。在系统开发中，原型是系统的一个早期可运行的版本，它反映最终系统的部分重要特性。如果在获得一组基本需求说明后，通过快速分析构造出一个小型的系统，满足用户的基本要求，使得用户可在试用原型系统的过程中得到亲身感受和受到启发，做出反应和评价，然后开发者根据用户的意見对原型加以改进。随着不断试验、纠错、使用、评价和修改，获得新的原型版本，如此周而复始，逐步减少分析和通信中的误解，弥补不足之处，进一步确定各种需求细节，适应需求的变更，从而提高了最终产品的质量。

从原型是否实现功能来分，可分为水平原型和垂直原型两种。水平原型也称为行为原型，用来探索预期系统的一些特定行为，并达到细化需求的目的。水平原型通常只是功能的导航，但并未真实现功能。水平原型主要用在界面上；垂直原型也称为结构化原型，实现了一部分功能。垂直原型主要用在复杂的算法实现上。

从原型的最终结果来分，可分为抛弃式原型和演化式原型。抛弃式原型也称为探索式原型，是指达到预期目的后，原型本身被抛弃。抛弃式原型主要用在解决需求不确定性、二义性、不完整性、含糊性等；演化式原型为开发增量式产品提供基础，逐步将原型演化成最终系统。主要用在必须易于升级和优化的场合，适用于Web项目。

试题答案

(40) C

第 7 章：系统开发基础

试题35 (2011年11月试题41)

试题35 (2011年11月试题41)

静态分析通过解析程序文本，从而识别出程序语句中可能存在的缺陷和异常之处；静态分析所包含的阶段中，(41) 的主要工作是找出输入变量和输出变量之间的依赖关系。

(41) A.控制流分析 B.数据使用分析 C.接口分析 D.信息流分析

试题分析

静态分析通过解析程序文本从而识别出程序语句的各个部分，审查可能的缺陷和异常之处，静态分析包括五个阶段：控制流分析阶段找出并突出显示那些带有多重出口或入口的循环以及不可达到的代码段；数据使用分析阶段突出程序中变量的使用情况；接口分析阶段检查子程序和过程说明及它们使用的一致性；信息流分析阶段找出输入变量和输出变量之间的依赖关系；路径分析阶段找出程序中所有可能的路径并画出在此路径中执行的语句。

试题答案

(41) D

试题36 (2011年11月试题42)

试题36 (2011年11月试题42)

确认测试主要用于验证软件的功能、性能和其他特性是否与用户需求一致。下述各种测试中，(42) 为确认测试。

- (42) A.负载测试和压力测试
- B.α测试和β测试
- C.随机测试和功能测试
- D.可靠性测试和性能测试

试题分析

确认测试中，需要"确认"的是用户需求。所以这种测试有一个显着的特点，就是测试必须要有用户的参与。所有选项中，只有B选项涉及的测试都有用户参与。

Alpha测试 (α测试) 是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试，Alpha测试不能由程序员或测试员 (有的地方又说可以让测试人员进行) 完成。

Beta测试 (β测试) 是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场，Beta测试不能由程序员或测试员完成。因而，Beta测试是在开发者无法控制的环境下进行的软件现场应用。

试题答案

- (42) B

试题37 (2011年11月试题43)

试题37 (2011年11月试题43)

软件 (43) 是指改正产生于系统开发阶段而在系统测试阶段尚未发现的错误。

- (43) A.完善性维护
- B.适应性维护
- C.正确性维护
- D.预防性维护

试题分析

软件维护的类型包括：改正性维护（正确性维护）、适应性维护、完善性维护、预防性维护。

改正性维护：在软件交付使用后，必然会有一部分隐藏的错误被带到运行阶段来。这些隐藏下来的错误在某些特定的使用环境下就会暴露出来。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程，就叫做改正性维护。

适应性维护：随着计算机的飞速发展，外部环境（新的硬、软件配置）或数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化，为了使软件适应这种变化，而去修改软件的过程就叫做适应性维护。

完善性维护：在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动叫做完善性维护。

预防性维护：为了提高软件的可维护性、可靠性等而提出的一种维护类型，它为以后进一步改进软件打下良好基础。通常，预防性维护定义为：“把今天的方法学用于昨天的系统以满足明天的需要”。也就是说，采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编制和测试。

试题答案

(43) C

第 7 章：系统开发基础

试题38 (2012年11月试题25)

试题38 (2012年11月试题25)

以下关于软件生存周期模型的叙述，正确的是（25）。

- (25) A.在瀑布模型中，前一个阶段的错误和疏漏会隐蔽地带到后一个阶段
- B.在任何情况下使用演化模型，都能在一定周期内由原型演化到最终产品
- C.软件生存周期模型的主要目标是为了加快软件开发的速度
- D.当一个软件系统的生存周期结束之后，它就进入到一个新的生存周期模型

试题分析

软件产品从形成概念开始，经过开发、使用和维护，直到最后退役的全过程成为软件生存周期。一个完整的软件生存周期是以需求为出发点，从提出软件开发计划的那一刻开始，直到软件在实际应用中完全报废为止。软件生存周期的提出是为了更好地管理、维护和升级软件，其中更大的意义在于管理软件开发的步骤和方法。

软件生存周期模型又称软件开发模型（software develop model）或软件过程模型（software process model），它是从某个特定角度提出的软件过程的简化描述。软件生存周期模型主要有瀑布模型、演化模型、原型模型、螺旋模型、喷泉模型和基于可重用构件的模型等。软件生存周期模型的主要目标是降低开发风险，提高软件品质，而非一味地提高开发速度。

瀑布模型是一个有着严格阶段划分的模型，这种模型要求完成一个阶段，才能开始下一个阶段的工作，先是需求分析，再是设计、编码、测试。该模型最大的缺陷在于，一个阶段的错误若没有被发现，将隐蔽地带到下一个阶段，使下一个阶段接着出错，所以A选项的描述是正确的。

演化模型是根据用户的基本需求，通过快速分析构造出该软件的一个初始可运行版本，这个初始的软件通常称之为原型，然后根据用户在使用原型的过程中提出的意见和建议对原型进行改进，获得原型的新版本。重复这一过程，最终可得到令用户满意的软件产品。采用演化模型的开发过程，实际上就是从初始的原型逐步演化成最终软件产品的过程。演化模型特别适用于对软件需求缺乏准确认识的情况。

试题答案

(25) A

试题39 (2012年11月试题26)

试题39 (2012年11月试题26)

螺旋模型将整个软件开发过程分为多个阶段，每个阶段都由目标设定、（26）、开发和有效性验证以及评审4个部分组成。

- (26) A.需求分析B.风险分析C.系统设计D.架构设计

试题分析

螺旋模型是一个综合了多种模型的特点形成的一种模型，其示意图如图7-8所示。

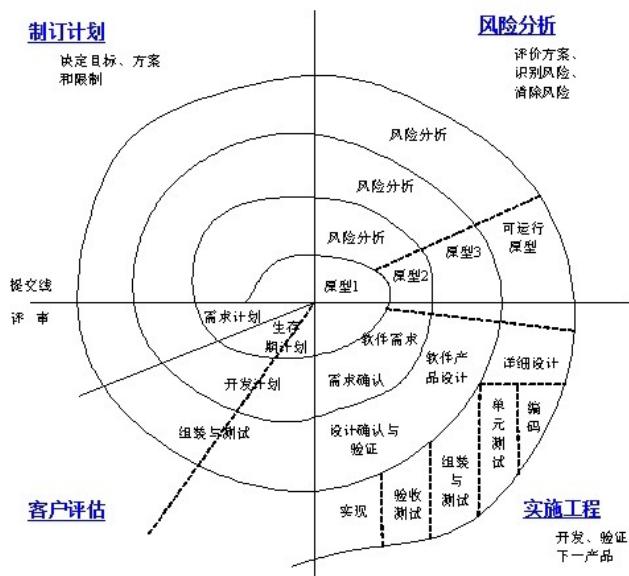


图7-8 螺旋模型示意图

螺旋模型是瀑布模型与演化模型相结合，并加入两者所忽略的风险分析所建立的一种软件开发模型。螺旋模型是一种演化软件过程模型，它将原型实现的迭代特征与线性顺序模型中控制的和系统化的方面结合起来，使软件的增量版本的快速开发成为可能。在螺旋模型中，软件开发是一系列的增量发布。

螺旋模型沿着螺线进行若干次迭代，每次迭代都包括制订计划、风险分析、实施工程和客户评估四个方面的工作。螺旋模型强调风险分析，使得开发人员和用户对每个演化层出现的风险有所了解，继而做出应有的反应。因此，特别适用于庞大、复杂并具有高风险的系统。

与瀑布模型相比，螺旋模型支持用户需求的动态变化，为用户参与软件开发的所有关键决策提供了方便，有助于提高软件的适应能力，并且为项目管理人员及时调整管理决策提供了便利，从而降低了软件开发的风险。在使用螺旋模型进行软件开发时，需要开发人员具有相当丰富的风险评估经验和专门知识。另外，过多的迭代次数会增加开发成本，延迟提交时间。

试题答案

- (26) B

试题40 (2012年11月试题27~28)

试题40 (2012年11月试题27~28)

基于UML的需求分析过程的基本步骤为：利用（27）表示需求；利用（28）表示目标软件系统的总体架构。

- (27) A.用例及用例图 B.包图及类图
- C.剧情及序列图 D.组件图及部署图
- (28) A.用例及用例图 B.包图及类图
- C.剧情及序列图 D.组件图及部署图

试题分析

在初步的业务需求描述已经形成的前提下，基于UML的需求分析过程大致可分为以下两个步骤。

- ① 利用用例及用例图表示需求。从业务需求描述出发获取执行者和场景；对场景进行汇总、分类、抽象，形成用例；确定执行者与用例、用例与用例图之间的关系，生成用例图。
- ② 利用包图和类图表示目标软件系统的总体框架结构。根据领域知识、业务需求描述和既往经验设计目标软件系统的顶层架构；从业务需求描述中提取“关键概念”，形成领域概念模型；从概念模型和用例出发，研究系统中主要的类之间的关系，生成类图。

试题答案

- (27) A (28) B

试题41 (2012年11月试题29~30)

试题41 (2012年11月试题29~30)

快速应用开发（Rapid Application Development, RAD）通过使用基于（29）的开发方法获得快速开发。当（30）时，最适合于采用RAD方法。

- (29) A.用例 B.数据结构 C.剧情 D.构件
- (30) A.一个新系统要采用很多新技术
- B.新系统与现有系统有较高的互操作性
- C.系统模块化程度较高
- D.用户不能很好地参与到需求分析中

试题分析

快速应用开发（Rapid Application Development, RAD）是一种比传统生命周期法快得多的开发方法，它强调极短的开发周期。RAD 模型是瀑布模型的一个高速变种，通过使用基于构件的开发

方法获得快速开发。如果需求理解得很好，且约束了项目范围，利用这种模型可以很快开发出功能完善的信息系统。

1. RAD的基本思想

RAD的基本思想体现在以下四个方面：

(1) 让用户更主动地参与到系统分析、设计和构造活动中来。

(2) 将项目开发组织成一系列重点突出的研讨会，研讨会要让项目投资方、用户、系统分析师、设计人员和开发人员一起参与。

(3) 通过一种迭代的构造方法，加速需求分析和设计阶段。

(4) 让用户提前看到一个可工作的系统。

2. RAD的开发阶段

RAD的流程从业务建模开始，随后是数据建模、过程建模、应用生成、测试与交付。

(1) 业务建模。确定驱动业务过程运作的信息、要生成的信息、如何生成、信息流的去向及其处理等，可以使用数据流图来帮助建立业务模型。

(2) 数据建模。为支持业务过程的数据流查找数据对象集合、定义数据对象属性，并与其他数据对象的关系构成数据模型，可以使用E-R图来帮助建立数据模型。

(3) 处理建模。将数据对象变换为要完成一个业务功能所需的信息流，创建处理以描述增加、修改、删除或获取某个数据对象，即细化数据流图中的加工。

(4) 应用生成。利用第四代语言(4GL)写出处理程序，复用已有构件或创建新的可复用构件，利用环境提供的工具自动生成并构造出整个应用系统。

(5) 测试与交付。因为RAD强调复用，许多构件已经是测试过的，这就减少了测试的时间。由于大量复用，所以一般只做总体测试，但新创建的构件还是要测试的。

3. RAD的特点

RAD采用基于构件的开发方法，复用已有的程序结构（如果可能的话）或使用构件，或者创建可复用的构件（如果需要的话）。在所有情况下，均可以使用CASE工具辅助进行软件构建。如果一个业务能够被模块化使得其中每一个主要功能均可以在不到三个月的时间内完成，那么，它就是RAD的一个候选者。每个主要功能可由一个单独的RAD组来实现，最后再集成起来，形成一个整体。

RAD通过大量使用可复用构件，加快了开发速度。但是，RAD也具有以下局限性：

(1) 并非所有应用都适合RAD。RAD对模块化要求比较高，如果有哪一项功能不能被模块化，那么RAD所需要的构件就会有问题；如果高性能是一个指标，且该指标必须通过调整接口使其适应系统构件才能获得，则RAD也有可能不能奏效。

(2) 开发者和客户必须在很短的时间完成一系列的需求分析，任何一方配合不当，都会导致RAD项目失败。

(3) RAD只能用于管理信息系统的开发，不适合技术风险很高的情况。例如，当一个新系统要采用很多新技术，或当新系统要与现有系统有较高的互操作性时，就不适合使用RAD。

试题答案

(29) D (30) C

试题42 (2012年11月试题31)

试题42 (2012年11月试题31)

以下关于软件开发方法的叙述，错误的是（31）。

- (31) A.对于较为复杂的应用问题，适合采用形式化方法进行需求分析
- B.形式化方法的优势在于能够精确地表述和研究应用问题及其软件实现
- C.净室软件工程将正确性验证作为发现和排除错误的主要机制
- D.净室软件工程强调统计质量控制技术，包括对客户软件使用预期的测试

试题分析

软件开发方法是指软件开发过程所遵循的办法和步骤，从不同的角度可以对软件开发方法进行不同的分类。

形式化方法是一种具有坚实数学基础的方法，从而允许对系统和开发过程做严格处理和论证，适用于那些系统安全级别要求极高的软件的开发。形式化方法的主要优越性在于它能够数学地表述和研究应用问题及软件实现。但是它要求开发人员具备良好的数学基础。用形式化语言书写的大型应用问题的软件规格说明往往过于细节化，并且难于为用户和软件设计人员所理解。由于这些缺陷，形式化方法在目前的软件开发实践中并未得到普遍应用。

净室软件工程（Cleanroom Software Engineering, CSE）是软件开发的一种形式化方法，可以开发较高质量的软件。它使用盒结构规约进行分析和建模，并且将正确性验证作为发现和排除错误的主要机制，使用统计测试来获取认证软件可靠性所需要的信息。CSE强调在规约和设计上的严格性，还强调统计质量控制技术，包括基于客户对软件的预期使用测试。

试题答案

(31) A

试题43 (2012年11月试题32~34)

试题43 (2012年11月试题32~34)

软件开发环境应支持多种集成机制。根据功能不同，可以将集成机制分为三个部分：

(32)，用于存储与系统开发有关的信息，并支持信息的交流与共享；(33)，是实现过程集成和控制集成的基础；(34)，它的统一性和一致性是软件开发环境的重要特征。

- (32) A.算法模型库 B.环境信息库 C.信息模型库 D.用户界面库

- (33) A.工作流与日志服务器 B.进程通信与数据共享服务器
- C.过程控制与消息服务器 D.同步控制与恢复服务器

- (34) A.底层数据结构 B.数据处理方法
- C.业务过程模型 D.环境用户界面

试题分析

软件开发环境 (Software Development Environment, SDE) 是指支持软件的工程化开发和维护而使用的一组软件，由软件工具集和环境集成机制构成。

软件开发环境应支持多种集成机制，根据功能的不同，集成机制可以划分为环境信息库、过程控制与消息服务器、环境用户界面三个部分。

① 环境信息库。环境信息库是软件开发环境的核心，用于存储与系统开发有关的信息，并支持信息的交流与共享。环境信息库中主要存储两类信息，一类是开发过程中产生的有关被开发系统的信息，例如分析文档、设计文档和测试报告等；另一类是环境提供的支持信息，如文档模板、系统配置、过程模型和可复用构件等。

② 过程控制与消息服务器。过程控制与消息服务器是实现过程集成和控制集成的基础。过程集成是按照具体软件开发过程的要求进行工具的选择与组合，控制集成是各工具之间进行并行通信和协同工作。

③ 环境用户界面。环境用户界面包括环境总界面和由它实行统一控制的各环境部件及工具的界面。统一的、具有一致性的用户界面是软件开发环境的重要特征，是充分发挥环境的优越性、高效地使用工具并减轻用户学习负担的保证。

试题答案

(32) B (33) C (34) D

第 7 章：系统开发基础

试题44 (2012年11月试题35)

试题44 (2012年11月试题35)

对于违反里氏替换原则的两个类A和B,可以采用的候选解决方案中，正确的是 (35) .

- (35) A.尽量将一些需要扩展的类或者存在变化的类设计为抽象类或者接口，并将其作为基类，在程序中尽量使用基类对象进行编程
B.创建一个新的抽象类C,作为两个具体类的超类，将A和B共同的行为移动到C中，从而解决A和B行为不完全一致的问题
C.将B到A的继承关系改成组合关系
D.区分是"Is-a"还是"Has-a".如果是Is-a,可以使用继承关系，如果是Has-a,应该改成组合或聚合关系

试题分析

里氏替换原则是面向对象设计原则之一，由Barbara Liskov提出，其基本思想是，一个软件实体如果使用的是一个基类对象，那么一定适用于其子类对象，而且觉察不出基类对象和子类对象的区别，即把基类都替换成它的子类，程序的行为没有变化。反过来则不一定成立，如果一个软件实体使用的是一个子类对象，那么它不一定适用于基类对象。

在运用里氏替换原则时，尽量将一些需要扩展的类或者存在变化的类设计为抽象类或者接口，并将其作为基类，在程序中尽量使用基类对象进行编程。由于子类继承基类并实现其中的方法，程

序运行时，子类对象可以替换基类对象，如果需要对类的行为进行修改，可以扩展基类，增加新的子类，而无需修改调用该基类对象的代码。

试题答案

(35) A

第 7 章：系统开发基础

试题45 (2012年11月试题36)

试题45 (2012年11月试题36)

以下关于黑盒测试用例设计方法的叙述，错误的是 (36) .

- (36) A. 边界值分析通过选择等价类边界作为测试用例，不仅重视输入条件边界，而且也必须考虑输出域边界
- B. 因果图方法是从用自然语言书写的程序规格说明的描述中找出因（输入条件）和果（输出或程序状态的改变），可以通过因果图转换为判定表
- C. 正交试验设计法，就是使用已经造好了的正交表格来安排试验并进行数据分析的一种方法，目的是用最少的测试用例达到最高的测试覆盖率
- D. 等价类划分法根据软件的功能说明，对每一个输入条件确定若干个有效等价类和无效等价类，但只能为有效等价类设计测试用例

试题分析

黑盒测试也称为功能测试，主要用于集成测试，确认测试和系统测试阶段。黑盒测试根据软件需求规格说明所规定的功能来设计测试用例，一般包括功能分解、等价类划分、边界值分析、判定表、因果图、状态图、随机测试、错误推测和正交试验法等。

在设计测试用例时，等价类划分是用得最多的一种黑盒测试方法。所谓等价类就是某个输入域的集合，对每一个输入条件确定若干个有效等价类和若干个无效等价类，分别设计覆盖有效等价类和无效等价类的测试用例。无效等价类是用来测试非正常的输入数据的，所以要为每个无效等价类设计一个测试用例。

边界值分析通过选择等价类边界作为测试用例，不仅重视输入条件边界，而且也必须考虑输出域边界。在实际测试工作中，将等价类划分法和边界值分析结合使用，能更有效地发现软件中的错误。

因果图方法是从用自然语言书写的程序规格说明的描述中找出因（输入条件）和果（输出或程序状态的改变），可以通过因果图转换为判定表。

正交试验设计法，就是使用已经造好了的正交表格来安排试验并进行数据分析的一种方法，目的是用最少的测试用例达到最高的测试覆盖率。

试题答案

(36) D

试题46 (2012年11月试题37)

试题46 (2012年11月试题37)

以下关于软件测试工具的叙述，错误的是（37）。

- (37) A.静态测试工具可用于对软件需求、结构设计、详细设计和代码进行评审、走查和审查
- B.静态测试工具可对软件的复杂度分析、数据流分析、控制流分析和接口分析提供支持
- C.动态测试工具可用于软件的覆盖分析和性能分析
- D.动态测试工具不支持软件的仿真测试和变异测试

试题分析

测试工具根据工作原理不同可分为静态测试工具和动态测试工具。其中静态测试工具是对代码进行语法扫描，找到不符合编码规范的地方，根据某种质量模型评价代码的质量，生成系统的调用关系图等。它直接对代码进行分析，不需要运行代码，也不需要对代码编译链接和生成可执行文件，静态测试工具可用于对软件需求、结构设计、详细设计和代码进行评审、走审和审查，也可用于对软件的复杂度分析、数据流分析、控制流分析和接口分析提供支持；动态测试工具与静态测试工具不同，它需要运行被测试系统，并设置探针，向代码生成的可执行文件中插入检测代码，可用于软件的覆盖分析和性能分析，也可用于软件的模拟、建模、仿真测试和变异测试等。

试题答案

(37) D

考点分析

第8章软件架构设计

8.1 考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 软件架构的概念。
- (2) 软件架构的风格。
- (3) 特定领域软件架构。
- (4) 基于架构的软件开发方法。
- (5) 软件架构评估。
- (6) 软件产品线。
- (7) 设计模式。

从历年的考试情况来看，本章的考点主要集中以下方面：基于构件的开发模型、架构风格、架构评估、架构与质量属性、设计模式。具体考查知识点分布情况如表8-1所示。

表8-1 历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题 30~31, 35~39, 41、45~63	设计模式 (Command 模式)、基于构件的开发模型、基于构件的开发模型、基于构件的开发模型、构件技术、MVC 模式、网络架构数据流图、架构设计所处阶段、架构设计基本概念、软件架构需求、ABSD、软件架构文档、架构复审、软件架构风格选择、架构模式、架构评估方法 (ATAM)
2010年11月试题 33~40, 44~63	工厂方法、构件技术、CORBA、分层架构、客户机/服务器、架构概念、ABSD、架构风格、架构评审、架构风格、DSSA、架构与质量属性、设计模式、架构评估
2011年11月试题 33~34, 44~63	Composite 模式、架构风格、设计模式、“4+1”视图、ABSD、黑板风格、解释器风格、隐式调用、中介者模式、命令模式、责任链模式、架构与质量属性、ATAM、敏感点与风险点
2012年11月试题 33~34, 44~63	ADL、软件架构相关概念、以架构为核心的开发方法、软件架构风格描述、中间件、设计模式、DSSA、架构与质量属性、SAAM

第 8 章：软件架构设计

试题1 (2009年11月试题30~31)

8.2 试题精解

试题1 (2009年11月试题30~31)

某软件公司欲开发一个Windows平台上的公告板系统。在明确用户需求后，该公司的架构师决定采用Command模式实现该系统的界面显示部分，并设计UML类图如图8-1所示。图中与Command模式中的"Invoker"角色相对应的类是 (30)，与"ConcreteCommand"角色相对应的类是 (31)。

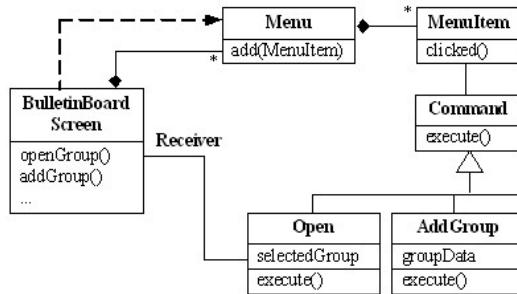


图8-1 类图

(30) A.Command B.MenuItem

C.Open D.BulletinBoardScreen

(31) A.Command B.MenuItem

C.Open D.BulletinBoardScreen

试题分析

Command (命令) 模式是设计模式中行为模式的一种，它将“请求”封装成对象，以便使用不同的请求、队列或者日志来参数化其他对象。Command模式也支持可撤销的操作。Command模式的类图如图8-2所示。

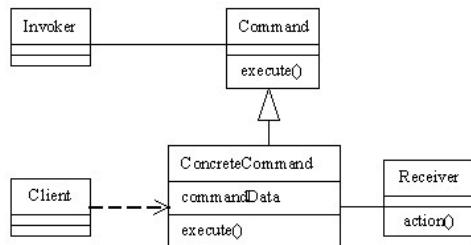


图8-2 Command类图

对于题目所给出的图，与“Invoker”角色相对应的类是MenuItem,与“Concrete Command”角色相对应的类是Open.

Command模式具有以下优点：

(1) 将调用操作的对象与知道如何完成该操作的对象相分离。

(2) 更容易添加新命令，因为不用修改已有类。

在以下情况中，应该使用Command模式：

(1) 想要通过要执行的动作来参数化对象。

(2) 要在不同的时间指定、排序以及执行请求。

(3) 必须支持Undo、日志记录或事务。

试题答案

(30) B (31) C

第8章：软件架构设计

试题2 (2009年11月试题35~37)

试题2 (2009年11月试题35~37)

基于构件的开发模型包括软件的需求分析定义、(35)、(36)、(37)以及测试和发布5个顺序执行的阶段。

(35) A.构件接口设计 B.体系结构设计 C.元数据设计 D.集成环境设计

(36) A.数据库建模 B.业务过程建模 C.对象建模 D.构件库建立

(37) A.应用软件构建 B.构件配置管理 C.构件单元测试 D.构件编码实现

试题分析

基于构件的软件开发 (Component Based Software Development,CBSD) 模型是利用模块化方法，将整个系统模块化，并在一定构件模型的支持下，复用构件库中的一个或多个软件构件，通过组合手段高效率、高质量地构造应用软件系统的过程。CBSD模型融合了螺旋模型的许多特征，本质上是演化型的，开发过程是迭代的。

CBSD模型由软件的需求分析和定义、架构设计、构件库的建立、应用软件构建、测试和发布5个阶段组成。CBSD方法使得软件开发不再一切从头开发，开发的过程就是构件组装的过程，维护的过程就是构件升级、替换和扩充的过程，其优点是提高了软件开发的效率；构件可由一方定义其规格说明，被另一方实现，然后供给第三方使用，CBSD模型允许多个项目同时开发，降低了费用，提高了可维护性，可实现分步提交软件产品。该方法的缺点是：由于采用自定义的组装结构标准，缺

乏通用的组装结构标准，引入具有较大的风险；可复用性和软件高效性不易协调，需要有经验的分析人员和开发人员，一般的开发人员插不上手，客户的满意度低；过分依赖于构件，构件库的质量影响着产品质量。

试题答案

(35) B (36) D (37) A

第 8 章：软件架构设计

试题3 (2009年11月试题38)

试题3 (2009年11月试题38)

以下关于软件构件及其接口的叙述，错误的是 (38) .

- (38) A.构件是软件系统中相对独立且具有一定意义的构成成分
- B.构件在容器中进行管理并获取其属性或者服务
- C.构件不允许外部对所支持的接口进行动态发现或调用
- D.构件可以基于对象实现，也可以不基于对象实现

试题分析

一般认为，构件是指语义完整、语法正确和有可重用价值的单位软件，是软件重用过程中可以明确辨识的系统；结构上，它是语义描述、通讯接口和实现代码的复合体。简单地说，构件是具有一定的功能，能够独立工作或能同其它构件装配起来协调工作的程序体，构件的使用同他的开发、生产无关。从抽象程度来看，面向对象 (Object Orientation,OO) 技术已达到了类级重用 (代码重用)，它以类为封装的单位。这样的重用粒度还太小，不足以解决异构互操作和效率更高的重用。构件将抽象的程度提到一个更高的层次，它是对一组类的组合进行封装，并代表完成一个或多个功能的特定服务，也为用户提供了多个接口。整个构件隐藏了具体的实现，只用接口对外提供服务。

与对象相比，构件可以基于对象实现，也可以不作为对象实现。构件需要在容器中管理并获取容器提供的服务；客户程序可以在运行状态下利用接口动态确定构件所支持的功能并调用。

试题答案

(38) C

第 8 章：软件架构设计

试题4 (2009年11月试题39)

试题4 (2009年11月试题39)

在一个典型的基于MVC (Model-View-Controller) 的J2EE应用中，分发客户请求、有效组织其他构件为客户端提供服务的控制器由 (39) 实现。

- (39) A.Entity Bean B.Session Bean C.Servlet D.JSP

试题分析

MVC是一种目前广泛流行的软件设计模式。近年来，随着J2EE (Java 2Enterprise Edition) 的成熟，MVC成为了J2EE平台上推荐的一种设计模式。MVC强制性地把一个应用的输入、处理、输出流程按照视图、控制、模型的方式进行分离，形成了三个核心模块：控制器、模型、视图。

(1) 控制器 (Controller)：控制器接受用户的输入并调用模型和视图去完成用户的需求。该部分是用户界面与Model的接口。一方面它解释来自于视图的输入，将其解释成为系统能够理解的对象，同时它也识别用户动作，并将其解释为对模型特定方法的调用；另一方面，它处理来自于模型的事件和模型逻辑执行的结果，调用适当的视图为用户提供反馈。

(2) 模型 (Model)：模型是应用程序的主体部分。模型表示业务数据和业务逻辑。一个模型能为多个视图提供数据。由于同一个模型可以被多个视图重用，所以提高了应用的可重用性。

(3) 视图 (View)：视图是用户看到并与之交互的界面。视图向用户显示相关的数据，并能接收用户的输入数据，但是它并不进行任何实际的业务处理。视图可以向模型查询业务状态，但不能改变模型。视图还能接受模型发出的数据更新事件，从而对用户界面进行同步更新。

MVC是构建应用框架的一个较好的设计模式，可以将业务处理与显示分离，将应用分为控制器、模型和视图，增加了应用的可拓展性、强壮性及灵活性。基于MVC的优点，目前比较先进的Web应用框架都是基于MVC设计模式的。

J2EE平台下实现MVC,系统的界面通常由JSP构件实现，分发客户请求、有效组织其他构件为客户端提供服务的控件器由Servlet构件实现，数据库相关操作由Entity Bean构件实现，系统核心业务逻辑由Session Bean构件实现。

试题答案

- (39) C

第 8 章：软件架构设计

试题5 (2009年11月试题41)

试题5 (2009年11月试题41)

系统应用架构设计中，网络架构数据流图的主要作用是将处理器和设备分配到网络中。

- (41) 不属于网络架构数据流图的内容。

- (41) A.服务器、客户端及其物理位置B.处理器说明信息
C.单位时间的数据流大小D.传输协议

试题分析

应用架构构建模中要绘制的第一个物理数据流图 (PDFD) 是网络架构DFD,它们不显示单位时间的数据流量，需要显示的信息包括服务器及其物理位置；客户端及其物理位置；处理器说明；传输协议。

试题答案

(41) C

第 8 章：软件架构设计

试题6 (2009年11月试题45)

试题6 (2009年11月试题45)

软件架构贯穿于软件的整个生命周期，但在不同阶段对软件架构的关注力度并不相同，(45)阶段，对软件架构的关注最多。

- (45) A.需求分析与设计B.设计与实现
- C.实现与测试D.部署与变更

试题分析

软件架构贯穿于软件的整个生命周期，但在不同的阶段对软件架构的关注力度并不相同。其中需求分析阶段主要关注问题域；设计阶段主要将需求转换为软件架构模型；软件实现阶段主要关注将架构设计转换为实际的代码；软件部署阶段主要通过组装软件组件提高系统的实现效率。其中设计与实现阶段在软件架构上的工作最多，也最重要，因此关注力度最大。

试题答案

(45) B

第 8 章：软件架构设计

试题7 (2009年11月试题46)

试题7 (2009年11月试题46)

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键活动。以下关于软件架构重要性的叙述中，错误的是 (46)。

- (46) A.架构设计能够满足系统的性能、一可维护性等品质
- B.良好的架构设计能够更好地捕获并了解用户需求
- C.架构设计能够使得不同的利益相关人 (Stakeholders) 达成一致的目标
- D.架构设计能够支持项目计划和项目管理等活动

试题分析

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键因素。

- 1、架构设计能够满足系统的品质

系统的功能性是软件构架师通过组成体系架构的多种元素之间的交互作用来支持的。架构设计用于实现系统的品质，如性能、安全性和可维护性等。通过架构设计文档化，可以尽早的评估项目的这些品质。

2、架构设计使受益人达成一致的目标

架构设计的过程使得不同的受益人达成一致的目标，体系架构的过程需要确保架构设计被清楚地传达与理解。一个被有效传达的体系架构使得涉众们可以辩论决议和权衡，反复讨论，最终达成共识。文档化体系架构是非常重要的，这是软件构架师的主要职责。

3、架构设计能够支持计划编制过程

架构设计将确定组件之间的依赖关系，直接支持项目计划和项目管理的活动，例如：细节化分，日程安排，工作分配，成本分析，风险管理技能开发等；构架师还能协助估算项目成本，例如：体系架构决定使用第三方组件的成本，以及支持开发的所有工具的成本；构架师支持技术风险的管理，包括制定每一个风险的优先次序，以及确定一个恰当的风险缓解策略。

4、架构设计对系统开发的指导性

架构设计主要目标就是确保体系架构能够为设计人员和实现人员所承担的工作提供可靠的框架。很明显，这比简单的传送一个体系架构视图要复杂的多。为了确保最终体系架构的完整性，构架师必须明确的定义体系架构，因为它确定了体系架构的重要元素，例如系统的组件，组件之间的接口以及组件之间的通信。

构架师同时还必须定义恰当的标准和指导方针，它们将会引导设计人员和实现人员的工作。对开发过程活动采取恰当的架构回顾和评估，能够确保体系架构的完整性。这些QA活动的任务是确定体系架构的标准和指导方针的有效性。

5、架构设计能够有效地管理复杂性

如今的系统越来越复杂，这种复杂性需要我们去管理。体系架构通过构件及构件之间关系，描述了一个抽象的系统，因而提供了高层次的复杂管理的方法。同样，架构设计过程考虑组件的递归分解。这是处理一个大的问题的很好的一个方法，它可以把这个大问题分解成很多的小问题，再逐个的解决。

6、架构设计为复用奠定了基础

架构设计过程可以同时支持使用和建立复用资源。复用资源可以降低一个系统的成本，并且可以改进系统的质量，这些好处已经被证明。一个体系架构的建立，能够支持大粒度的资源复用。例如，体系架构的重要组件和它们之间的接口和质量，能够支持现货供应的组件，存在的系统和封装的应用程序等等的选择，从而可以用来实现这些组件。

7、架构设计能够降低维护费用

架构设计过程可以在很多方面帮助我们降低维护费用。首先最重要的是架构设计过程要确保系统的维护人员是一个主要的涉众，并且他们的需求被作为首要的任务满足。一个被恰当文档化的体系架构不应该仅仅为了减轻系统的可维护性；构架师还应该确保结合了恰当的系统维护机制，并且在建立体系架构的时候还要考虑系统的适应性和可扩充性。

8、架构设计能够支持冲突分析

架构设计的一个重要的好处是它可以允许我们在采取改变之前推断它所产生的影响。一个软件构架确定了主要的组件和它们之间的交互作用，两个组件之间的依赖性以及这些组件对于需求的可追溯性。有了这个信息，例如需求的改变等可以通过组件的影响来分析。同样的，改变一个组件的影响可以在依靠它的其它组件上分析出来。

试题答案

试题8 (2009年11月试题47)

试题8 (2009年11月试题47)

软件架构需求是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。以下活动中，不属于软件架构需求过程范畴的是 (47) 。

- (47) A.设计构件B.需求获取C.标识构件D.架构需求评审

试题分析

架构需求受技术环境和设计师的影响。需求过程主要是获取用户需求，标识系统中所要用到的构件。如果以前有类似的系统架构的需求，可以从需求库中取出，加以利用和修改，以节省需求获取的时间，减少重复劳动，提高开发效率。

需求过程主要是获取用户需求，标识系统中所要用到的构件，并进行架构需求评审。其中标识构件又详细分为生成类图、对类图进行分组和将类打包成构件三步。

1、需求获取

体系结构需求一般来自三个方面，分别是系统的质量目标、系统的业务目标和系统开发人员的业务目标。软件体系结构需求获取过程主要是定义开发人员必须实现的软件功能，使得用户能完成他们的任务，从而满足业务上的功能需求。与此同时，还要获得软件质量属性，满足一些非功能需求。

2、标识构件

该过程为系统生成初始逻辑结构，包含大致的构件。这一过程又可分为三步来实现。

第一步：生成类图。生成类图的CASE工具有很多，例如 Rational Rose 2000就能自动生成类图。

第二步：对类进行分组。在生成的类图基础上，使用一些标准对类进行分组可以大大简化类图结构，使之更清晰。一般地，与其他类隔离的类形成一个组，由泛化关联的类组成一个附加组，由聚合或组合关联的类也形成一个附加组。

第三步：把类打包成构件。把在第二步得到的类簇打包成构件，这些构件可以分组合并成更大的构件。

3、需求评审

组织一个由不同代表（如分析人员、客户、设计人员、测试人员）组成的小组，对体系结构需求及相关构件进行仔细的审查。审查的主要内容包括所获取的需求是否真实反映了用户的要求，类的分组是否合理，构件合并是否合理等。

必要时，可以在“需求获取-标识构件-需求评审”之间进行迭代。

试题答案

- (47) A

试题9 (2009年11月试题48)

试题9 (2009年11月试题48)

基于架构的软件设计 (ABSD) 强调由商业、质量和功能需求的组合驱动软件架构设计。以下关于ABSD的叙述中，错误的是 (48) .

- (48) A.使用ABSD方法，设计活动可以从项目总体功能框架明确就开始
- B.ABSD方法是一个自顶向下，递归细化的过程
- C.ABSD方法有三个基础：功能分解、选择架构风格实现质量和商业需求以及软件模板的使用
- D.使用ABSD方法，设计活动的开始意味着需求抽取和分析活动可以终止

试题分析

ABSD方法为产生软件系统的概念架构提供基础，概念架构是由Hofmeister、Nord和Soni提出的四种不同的架构中的一种，它描述了系统的主要设计元素及其关系。概念架构代表了在开发过程中作出的第一个选择，相应地，它是达到系统质量和业务目标的关键，为达到预定功能提供了一个基础。

ABSD方法取决于决定系统的架构驱动。所谓架构驱动，是指构成架构的业务、质量和功能需求的组合。使用ABSD方法，设计活动可以在架构驱动一决定就开始，这意味着需求获取和分析还没有完成（甚至远远没有完成），就开始了软件设计。设计活动的开始并不意味着需求获取和分析活动就可以终止，而是应该与设计活动并行。特别是在不可能预先决定所有需求时，例如产品线系统或长期运行的系统，快速开始设计是至关重要的。

ABSD方法有3个基础。第一个基础是功能的分解。在功能分解中，ABSD方法使用已有的基于模块的内聚和耦合技术。第二个基础是通过选择架构风格来实现质量和业务需求。第三个基础是软件模板的使用。软件模板利用了一些软件系统的结构。然而，对于设计方法来说，软件模板的使用是一个新概念，下面，进行简单的介绍。

软件模板是一个特殊类型的软件元素，包括描述所有这种类型的元素在共享服务和底层构造的基础上如何进行交互。软件模板还包括属于这种类型的所有元素的功能，这些功能的例子有：每个元素必须记录某些重大事件，每个元素必须为运行期间的外部诊断提供测试点等。在软件产品线系统中，软件模板显得格外重要，因为新元素的引入是一个通用的技术，这种技术用来使产品线体系结构适应一个特定的产品。

ABSD方法是递归的，且迭代的每一个步骤都是清晰地定义的。因此，不管设计是否完成，体系结构总是清晰的，这有助于降低体系结构设计的随意性。

试题答案

- (48) D

试题10 (2009年11月试题49)

试题10 (2009年11月试题49)

软件架构文档是对软件架构的正式描述，能够帮助与系统有关的开发人员更好地理解软件架构。软件架构文档的写作应该遵循一定的原则。以下关于软件架构文档写作原则的叙述中，错误的是 (49) 。

- (49) A. 架构文档应该从架构设计者的角度进行编写
- B. 应该保持架构文档的即时更新，但更新不要过于频繁
- C. 架构文档中的描述应该尽量避免不必要的重复
- D. 每次架构文档修改，都应该记录修改的原则

试题分析

软件架构文档是对软件架构的一种描述，帮助程序员使用特定的程序设计语言实现软件架构。

软件架构文档的写作应该遵循一定的原则，这些原则包括：文档要从使用者的角度进行编写；必须分发给所有与系统有关的开发人员；应该保持架构文档的即时更新，但更新不要过于频繁；架构文档中描述应该尽量避免不必要的重复：每次架构文档修改都应该记录进行修改的原则。

试题答案

- (49) A

试题11 (2009年11月试题50)

试题11 (2009年11月试题50)

架构复审是基于架构开发中一个重要的环节。以下关于架构复审的叙述中，错误的是_____。

- (50) A. 架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误
- B. 架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试
- C. 架构复审人员由系统设计与开发人员组成
- D. 架构设计、文档化和复审是一个迭代的过程

试题分析

架构复审是基于架构开发中一个重要的环节。架构设计、文档化和复审是一个迭代的过程。从这个方面来说，在一个主版本的软件架构分析之后，要安排一次由外部人员（用户代表和领域专家）参加的复审。架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试。架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误。

试题答案

- (50) C

试题12 (2009年11月试题51~52)

试题12 (2009年11月试题51~52)

Windows操作系统在图形用户界面处理方面采用的核心架构风格是 (51) 风格。Java语言宣传的"一次编写，到处运行"的特性，从架构风格上看符合 (52) 风格的特点。

- (51) A.虚拟机 B.管道-过滤器 C.事件驱动 D.微内核-扩展
(52) A.虚拟机 B.管道-过滤器 C.事件驱动 D.微内核-扩展

试题分析

Garlan和Shaw对通用软件架构风格进行了分类，他们将软件架构分为数据流风格、调用/返回风格、独立构件风格、虚拟机风格和仓库风格。

1. 数据流风格

数据流风格包括批处理序列和管道/过滤器两种风格。

(1) 批处理序列。构件为一系列固定顺序的计算单元，构件之间只通过数据传递交互。每个处理步骤是一个独立的程序，每一步必须在其前一步结束后才能开始，数据必须是完整的，以整体的方式传递。

(2) 管道/过滤器。每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常是通过对输入数据流的变换或计算来完成的，包括通过计算和增加信息以丰富数据、通过浓缩和删除以精简数据、通过改变记录方式以转化数据和递增地转化数据等。这里的构件称为过滤器，连接件就是数据流传输的管道，将一个过滤器的输出传到另一个过滤器的输入。

2. 调用/返回风格

调用/返回风格包括主程序/子程序、数据抽象和面向对象，以及层次结构。

(1) 主程序/子程序。单线程控制，把问题划分为若干个处理步骤，构件即为主程序和子程序，子程序通常可合成为模块。过程调用作为交互机制，即充当连接件的角色。调用关系具有层次性，其语义逻辑表现为主程序的正确性取决于它调用的子程序的正确性。

(2) 数据抽象和面向对象。这种风格的构件是对象，对象是抽象数据类型的实例。在抽象数据类型中，数据的表示和它们的相应操作被封装起来，对象的行为体现在其接受和请求的动作。连接件即是对象间交互的方式，对象是通过函数和过程的调用进行交互的。对象具有封装性，一个对象的改变不会影响其他对象。

(3) 层次结构。层次系统的构件组织成一个层次结构，连接件通过决定层间如何交互的协议来定义。该风格的特点是每层为上一层提供服务，使用下一层的服务，只能见到与自己邻接的层。通过层次结构，可以将大的问题分解为若干个渐进的小问题逐步解决，可以隐藏问题的复杂度。在层次结构中，修改某一层，最多影响其相邻的两层（通常只能影响上层）。上层必须知道下层的身份，不能调整层次之间的顺序。例如，网络通信协议和操作系统就属于层次结构。

3. 独立构件风格

独立构件风格包括进程通信和事件驱动的系统。

(1) 进程通信。构件是独立的过程，连接件是消息传递。这种风格的特点是，构件通常是命名过程，消息传递的方式可以是点对点、异步或同步方式，以及远程过程（方法）调用等。

(2) 事件驱动的系统。构件不直接调用一个过程，而是触发或广播一个或多个事件。构件中的过程在一个或多个事件中注册，当某个事件被触发时，系统自动调用在这个事件中注册的所有过程。一个事件的触发就导致了另一个模块中的过程调用。这种风格中的构件是匿名的过程，它们之间交互的连接件往往是以过程之间的隐式调用（implicit invocation）来实现的。基于事件的隐式调用风格的主要优点是为软件复用提供了强大的支持，为构件的维护和演化带来了方便，其缺点是构件放弃了对系统计算的控制。

4. 虚拟机风格

虚拟机风格包括解释器和基于规则的系统。

(1) 解释器。解释器通常包括一个完成解释工作的解释引擎、一个包含将被解释的代码的存储区、一个记录解释引擎当前工作状态的数据结构，以及一个记录源代码被解释执行的进度的数据结构。具有解释器风格的软件中含有一个虚拟机，可以仿真硬件的执行过程和一些关键应用，其缺点是执行效率比较低。

(2) 基于规则的系统。基于规则的系统包括规则集、规则解释器、规则/数据选择器和工作内存，一般用在人工智能领域和DSS中。

5. 仓库风格

仓库风格包括数据库系统、黑板系统和超文本系统。

(1) 数据库系统。数据库系统是仓库风格最常见的形式。在数据库系统中，构件主要有两大类，一类是中央共享数据源，保存当前系统的数据状态；另一类是多个独立处理单元，处理单元对数据元素进行操作。

(2) 黑板系统。黑板系统包括知识源、黑板和控制三个部分。知识源包括若干独立计算的不同单元，提供解决问题的知识。知识源响应黑板的变化，也只修改黑板；黑板是一个全局数据库，包含问题域解空间的全部状态，是知识源相互作用的唯一媒介；知识源响应是通过黑板状态的变化来控制的。黑板系统通常应用在对于解决问题没有确定性算法的软件中，例如，信号处理、问题规划和编译器优化等。

(3) 超文本系统。超文本系统中出现的构件以网状链接方式相互连接，用户可以在构件之间进行按照人类的联想思维方式任意跳转到相关构件。超文本是一种非线性的网状信息组织方法，它以结点为基本单位，链作为结点之间的联想式关联。超文本系统通常应用在互联网领域。

Windows操作系统在图形用户界面处理方面采用的是典型的“事件驱动”的架构风格。首先注册事件处理的是回调函数，当某个界面事件发生时（例如键盘敲击、鼠标移动等），系统会查找并选择合适的回调函数处理该事件。Java语言是一种解释型语言，在Java虚拟机上运行，这从架构风格上看是典型的“虚拟机”风格，即通过虚拟机架构屏蔽不同的硬件环境。

试题答案

(51) C (52) A

试题13 (2009年11月试题53)

试题13 (2009年11月试题53)

某软件开发公司负责开发一个Web服务器服务端处理软件，其核心部分是对客户端请求消息的解析与处理，包括HTTP报头分离、SOAP报文解析等功能。该公司的架构师决定采用成熟的架构风格指导整个软件的设计，以下 (53) 架构风格，最适合该服务端处理软件。

- (53) A.虚拟机 B.管道-过滤器 C.黑板结构 D.分层结构

试题分析

根据题干描述，Web服务器服务端的核心功能是数据处理，由于Web服务在数据传输方面具有协议分层的特征，即底层协议会包装上层协议（HTTP协议体中包含整个SOAP消息内容），因此需要数据内容的逐步分解与分阶段处理。比较选项中的架构风格，由于管道-过滤器的架构风格支持分阶段数据处理，因此特别适合该服务端处理软件的要求。

试题答案

- (53) B

第 8 章：软件架构设计

试题14 (2009年11月试题54)

试题14 (2009年11月试题54)

希赛公司欲开发一个基于图形用户界面的集成调试器。该调试器的编辑器和变量监视器可以设置调试断点。当调试器在断点处暂停运行时，编辑程序可以自动卷屏到断点，变量监视器刷新变量数值。针对这样的功能描述，采用 (54) 的架构风格最为合适。

- (54) A.数据共享 B.虚拟机 C.隐式调用 D.显式调用

试题分析

根据题干描述，调试器在设置断点时，其本质是在断点处设置一个事件监听函数，当程序执行到断点位置时，会触发并调用该事件监听函数，监听函数负责进行自动卷屏、刷新变量数值等工作。这是一个典型的回调机制，属于隐式调用的架构风格。

试题答案

- (54) C

第 8 章：软件架构设计

试题15 (2009年11月试题55)

试题15 (2009年11月试题55)

某公司欲开发一种工业机器人，来进行汽车零件的装配。公司的架构师经过分析与讨论，给出了该机器人控制软件的两种候选架构方案：闭环控制和分层结构。以下对于这两种候选架构的选择理由，错误的是（55）。

- (55) A. 应该采用闭环控制架构，因为闭环结构给出了将软件分解成几个协作构件的方法，这对于复杂任务特别适合
- B. 应该采用闭环控制结构，因为闭环控制架构中机器人的主要构件（监控器、传感器、发动机等）是彼此分开的，并能够独立替换
- C. 应该采用分层结构，因为分层结构很好地组织了用来协调机器人操作的构件，系统结构更加清晰
- D. 应该采用分层结构，因为抽象层的存在，满足了处理不确定性的需要：在较低层次不确定的实现细节在较高层次会变得确定

试题分析

采用闭环结构的软件通常由几个协作构件共同构成，且其中的主要构件彼此分开，能够进行替换与重用，但闭环结构通常适用于处理简单任务（如机器装配等），并不适用于复杂任务。分层结构的特点是通过引入抽象层，在较低层次不确定的实现细节在较高层次会变得确定，并能够组织层间构件的协作，系统结构更加清晰。

试题答案

(55) A

第8章：软件架构设计

试题16（2009年11月试题56）

试题16（2009年11月试题56）

一个软件的架构设计是随着技术的不断进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。以下关于编译器架构的叙述中，错误的是（56）。

- (56) A. 早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码
- B. 早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表并不是程序数据的一部分
- C. 现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示
- D. 现代的编译器采用以数据共享为中心的架构风格，但由于分析树是在语法分析阶段结束后才产生作为语义分析的输入，因此分析树不是数据中心的共享数据

试题分析

一个软件的架构设计是随着技术的不断进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码。早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表

并不是程序数据的一部分。现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示。现代的编译器采用以数据共享为中心的架构风格，分析树是在语法分析阶段结束后才产生作为语义分析的输入，分析树是数据中心中重要的共享数据，为后续的语义分析提供了帮助。

试题答案

(56) D

第 8 章：软件架构设计

试题17 (2009年11月试题57~59)

试题17 (2009年11月试题57~59)

(57) 的选择是开发一个软件系统时的基本设计决策；(58) 是最低层的模式，关注软件系统的设计与实现，描述了如何实现构件及构件之间的关系。引用-计数是C++管理动态资源时常用的一种 (59) 。

(57) A.架构模式 B.惯用法 C.设计模式 D.分析模式

(58) A.架构模式 B.惯用法 C.设计模式 D.分析模式

(59) A.架构模式 B.惯用法 C.设计模式 D.分析模式

试题分析

模式是给定上下文中普遍问题的普遍解决方案，在软件开发方面，分高层的模式与低层的模式，具体包括体系结构模式、设计模式 (design pattern) 和惯用法等。

架构模式是指软件架构的风格，这是软件设计中的高层决策。例如，C/S结构就属于架构模式，架构模式反映了开发软件系统过程中所作的基本设计决策。

设计模式主要关注软件系统的设计，与具体的实现语言无关。设计模式是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结，使用设计模式的目的是为了提高代码的可重用性，让代码更容易被他人理解，并保证代码可靠性。毫无疑问，这些设计模式已经在前人的系统中得以证实并广泛使用，它使代码编写真正实现工程化，将已证实的技术表述成设计模式，也会使新系统开发者更加容易理解其设计思路。

惯用法则是实现时通过某种特定的程序设计语言来描述构件与构件之间的关系，例如，引用-计数就是C++语言中的一种惯用法。

试题答案

(57) A (58) B (59) B

第 8 章：软件架构设计

试题18 (2009年11月试题60)

试题18 (2009年11月试题60)

某软件公司基于面向对象技术开发了一套图形界面显示构件库VisualComponent.在使用该库构建某图形界面时，用户要求为界面定制一些特效显示效果，如带滚动条、能够显示艺术字体的透明窗体等。针对这种需求，公司采用（60）最为灵活。

- (60) A.桥接模式 B.命令模式 C.组合模式 D.装饰模式

试题分析

首先，简单介绍题目中涉及到的4种设计模式。

(1) 桥接 (bridge) 模式。桥接模式将抽象部分与它的实现部分分离，使它们都可以独立地变化。它是一种对象结构型模式，又称为柄体 (handle and body) 模式或接口 (interface) 模式。桥接模式类似于多重继承方案，但是多重继承方案往往违背了类的单一职责原则，其复用性比较差，桥接模式是比多重继承方案更好的解决方法。

(2) 命令 (command) 模式。命令模式是一种对象的行为型模式，类似于传统程序设计方法中的回调机制，它将一个请求封装为一个对象，从而使得可用不同的请求对客户进行参数化；对请求排队或者记录请求日志，以及支持可撤销的操作。命令模式是对命令的封装，将发出命令的责任和执行命令的责任分割开，委派给不同的对象，以实现发送者和接收者完全解耦，提供更大的灵活性和可扩展性。

(3) 组合 (composite) 模式。组合模式又称为整体-部分 (part-whole) 模式，属于对象的结构模式。在组合模式中，通过组合多个对象形成树形结构以表示整体-部分的结构层次。组合模式对单个对象（即叶子对象）和组合对象（即容器对象）的使用具有一致性。

(4) 装饰 (decorator) 模式。装饰模式是一种对象结构型模式，可动态地给一个对象增加一些额外的职责，就增加对象功能来说，装饰模式比生成子类实现更为灵活。通过装饰模式，可以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；当需要动态地给一个对象增加功能，这些功能可以再动态地被撤销时可使用装饰模式；当不能采用生成子类的方法进行扩充时也可使用装饰模式。

每一种设计模式，都有自身的一些优势，以应对某些特定的应用场合。根据题目的描述，需要开发的是图形界面构件库，并要求为图形界面提供一些定制的特效，例如，带滚动条的图形界面，能够显示艺术字体且透明的图形界面等。这要求能够动态地对一个对象进行功能上的扩展，也可以对其子类进行功能上的扩展。对照选项中的4种设计模式，装饰模式最符合这一要求。

试题答案

- (60) D

试题19 (2009年11月试题61)

试题19 (2009年11月试题61)

某软件公司承接了为某工作流语言开发解释器的工作。该工作流语言由多种活动节点构成，具有类XML的语法结构。用户要求解释器工作时，对每个活动节点进行一系列的处理，包括执行活动、日志记录、调用外部应用程序等，并且要求处理过程具有可扩展能力。针对这种需求，公司采用（61）最为恰当。

（61）A.适配器模式 B.迭代器模式

 C.访问者模式 D.观察者模式

试题分析

首先，简单介绍题目中涉及到的4种设计模式。

（1）适配器（adapter）模式。适配器模式将一个接口转换成客户希望的另一个接口，从而使接口不兼容的那些类可以一起工作。适配器模式既可以作为类结构型模式，也可以作为对象结构型模式。在类适配器模式中，通过使用一个具体类将适配者适配到目标接口中；在对象适配器模式中，一个适配器可以将多个不同的适配者适配到同一个目标。

（2）迭代器（iterator）模式。迭代器模式是一种对象的行为型模式，提供了一种方法来访问聚合对象，而不用暴露这个对象的内部表示。迭代器模式支持以不同的方式遍历一个聚合对象，复杂的聚合可用多种方法来进行遍历；允许在同一个聚合上可以有多个遍历，每个迭代器保持它自己的遍历状态，因此，可以同时进行多个遍历操作。

（3）访问者（visitor）模式。访问者模式是一种对象的行为型模式，用于表示一个作用于某对象结构中的各元素的操作，它使得用户可以在不改变各元素的类的前提下定义作用于这些元素的新操作。访问者模式使得增加新的操作变得很容易，但在一定程度上破坏了封装性。

（4）观察者（observer）模式。观察者模式又称为发布-订阅模式、模型-视图模式、源-监听器模式或从属者（dependents）模式，是一种对象的行为型模式。它定义了对象之间的一种一对多的依赖关系，使得每当一个对象状态发生改变时，其相关依赖对象都得到通知并被自动更新。观察者模式的优点在于实现了表示层和数据层的分离，并定义了稳定的更新消息传递机制，类别清晰，抽象了更新接口，使得相同的数据层可以有各种不同的表示层。

根据题目的描述，可以看出本题的核心在于对某个具有固定结构的活动节点需要多种处理能力，且处理能力可扩展，也就是说要求在不改变原来类结构（活动节点）的基础上增加新功能。对照4个选项，发现访问者模式最符合要求。

试题答案

（61）C

第8章：软件架构设计

试题20（2009年11月试题62）

试题20（2009年11月试题62）

Architecture Tradeoff Analysis Method (ATAM) 是一种软件架构的评估方法，以下关于该方法的叙述中，正确的是（62）。

（62）A. ATAM是一种代码评估方法

- B. ATAM需要评估软件的需求是否准确
- C. ATAM需要对软件系统进行测试
- D. ATAM不是一种精确的评估工具

试题分析

使用ATAM方法对软件架构进行评估的目的，是依据系统质量属性和业务需求评估设计决策的结果。ATAM希望揭示出架构满足特定质量目标的情况，使架构设计师更清楚地认识到质量目标之间的联系，即如何权衡多个质量目标。这些设计决策很重要，一直会影响到整个软件生命周期，并且在软件实现后很难修改这些决策。

1.评估参与者

在ATAM方法中，参加评估的人员主要有评估小组、项目决策者和其他项目干系人。

(1) 评估小组。该小组是所评估架构项目外部的小组，通常由3~5人组成，他们可能是开发组织内部的，也可能是外部的。评估小组的每个成员都要扮演大量的特定角色。

(2) 项目决策者。项目决策者对开发项目具有发言权，并有权要求进行某些改变，他们包括项目管理人员、重要的客户代表和架构设计师等。

(3) 项目干系人。包括关键模块开发人员、测试人员和用户等。

2.评估活动

整个ATAM评估过程包括九个步骤，如图8-3所示。

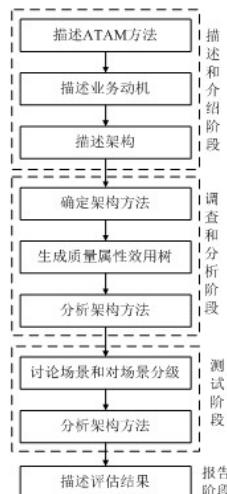


图8-3 ATAM方法的步骤

(1) 描述ATAM方法。评估小组负责人向参加会议的项目干系人介绍ATAM评估方法。在这一步中，要解释每个人将要参与的过程，并预留出解答疑问的时间，设置好其他活动的环境和预期结果。关键是要使每个人都知道要收集哪些信息，如何描述这些信息，将要向谁报告等。

(2) 描述业务动机。项目决策者从业务的角度介绍系统的概况。该描述应该包括系统最重要的功能、技术/管理/经济和政治方面的任何相关限制、与该项目相关的业务目标和上下文、主要的项目干系人，以及架构的驱动因素等。参加评估的所有人员必须理解待评估的系统。

(3) 描述架构。首席设计师或设计小组要对架构进行详略适当的介绍，至少应该包括技术约束（例如，操作系统、硬件和中间件等）、将与本系统进行交互的其他系统、用以满足质量属性要求的架构方法等。这一步很重要，将直接影响到可能要做的分析及分析的质量。

(4) 确定架构方法。ATAM评估方法主要通过理解架构方法来分析架构，在这一步，由架构设计师确定架构方法，由分析小组捕获，但不进行分析。

(5) 生成质量属性效用树。评估小组、设计小组、管理人员和客户代表一起确定系统最重要的质量属性目标，并对这些质量目标设置优先级和细化。这一步很关键，它对以后的分析工作起指导作用。即使是架构级的分析，也不一定是全局的，所以，需要集中所有相关人员的精力，注意架构的各个方面，这通常是通过构建效用树的方式来实现的。效用树的输出结果是对具体质量属性需求的优先级的确定，这种优先级列表为ATAM评估方法的后面几步提供了指导，它告诉评估小组应该把有限的时间花在哪里，特别是应该到哪里去考察架构的方法与相应的风险、敏感点和权衡点。

(6) 分析架构方法。一旦有了效用树的结果，评估小组可以对实现重要质量属性的架构方法进行考察。这是通过文档化这些架构决策和确定它们的风险、敏感点和权衡点等来实现的。在这一步中，评估小组要对每一种架构方法都考察足够的信息，完成与该方法有关的质量属性的初步分析。这一步的主要结果是一个架构方法或风格的列表，与之相关的一些问题，以及设计师对这些问题的回答。通常产生一个风险列表、敏感点和权衡点列表。在这一步结束时，评估小组应该对整个架构的绝大多数重要方面所做出的关键设计决策、风险列表、敏感点、权衡点有一个清楚的认识。

(7) 讨论场景和对场景分级。场景在驱动ATAM测试阶段起主导作用。项目干系人进行两项相关的活动，分别是集体讨论用例场景和改变场景。用例场景是场景的一种，在用例场景中，项目干系人是一个终端用户，使用系统执行的一些功能。一旦收集了若干个场景后，必须设置优先级。评估人员通过投票表决的方式来完成，每个项目干系人分配相当于总场景数的30%的选择，且此数值只入不舍。例如，如果共有17个场景，则每个风险承担者将拿到6张选票，这6张选票的具体使用则取决于项目干系人，他可以把这6张票全部投给一个场景，或者每个场景2~3张票，还可以一个场景一张票等。

(8) 分析架构方法。在收集并分析了场景之后，设计师就可把最高级别的场景映射到所描述的架构中，并对相关的架构如何有助于该场景的实现做出解释。在这一步中，评估小组要重复第6步中的工作，把新得到的最高优先级场景与尚未得到的架构工作产品对应起来。在第7步中，如果未产生任何在以前的分析步骤中都没有发现的高优先级场景，则在第8步就是测试步骤。

(9) 描述评估结果。最后，要把ATAM分析中所得到的各种信息进行归纳，并反馈给项目干系人。这种描述一般要采用辅以幻灯片的形式，但也可以在ATAM评估结束之后，提交更完整的书面报告。在描述过程中，评估负责人要介绍ATAM评估的各个步骤，以及各步骤中得到的各种信息，包括业务环境、驱动需求、约束条件和架构等。最重要的是要介绍ATAM评估的结果。ATAM的评估结果包括一个简洁的架构描述、表达清楚的业务目标、用场景集合捕获的质量属性、所确定的敏感点和权衡点的集合、有风险决策和无风险决策、风险主题的集合。

在具体的软件架构评估过程中，可以修改这九个步骤的顺序，以满足架构信息的特殊需求。也就是说，虽然这九个步骤按编号排列，但并不总是一个瀑布过程，评估人员可在这九个步骤中跳转或进行迭代。

评估是软件系统详细设计、实现和测试之前的阶段工作，因此评估不涉及系统的实现代码和测试，因为评估是考查软件体系结构是否能够合适地解决软件系统的需求，并不对软件需求自身是否准确进行核实，而软件需求是否准确是需求评审阶段的工作。ATAM并不是一种精确的评估方法，该方法表现的主要形式是评审会议。

试题答案

试题21 (2009年11月试题63)

试题21 (2009年11月试题63)

识别风险点、非风险点、敏感点和权衡点是ATAM方法中的关键步骤。已知针对某系统所做的架构设计中，提高其加密子系统的加密级别将对系统的安全性和性能都产生非常大的影响，则该子系统一定属于 (63) 。

- (63) A.风险点和敏感点B.权衡点和风险点
- C.权衡点和敏感点D.风险点和非风险点

试题分析

敏感点是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点是指影响多个质量属性的特性，是多个质量属性的敏感点。

风险点是指潜在的、存在隐患的架构决策所带来的隐患。

加密子系统的加密级别会对安全性和性能产生影响，一般而言，加密程度越高，安全性越好，但是其性能会降低；而加密程度越低，安全性越差，但性能一般会提高。因此该子系统将在安全性和性能两个方面产生冲突，所以该子系统一定属于权衡点和敏感点。

试题答案

- (63) C

试题22 (2010年11月试题33~34)

试题22 (2010年11月试题33~34)

某银行系统采用Factory Method方法描述其不同账户之间的关系，设计出的类图如图8-4所示。其中与Factory Method中的"Creator"角色相对应的类是 (33) ；与"Product"角色相对应的类是 (34) 。

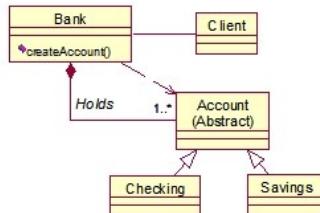


图8-4 类图

- (33) A.BankB.AccountC.Checking D.Savings
- (34) A.BankB.AccountC.Checking D.Savings

试题分析

Factory Method (工厂方法) 模式又称为虚拟构造器 (virtual constructor) 模式或多态模式，属于类的创建型模式。在工厂方法模式中，父类负责定义创建对象的公共接口，而子类则负责生成具体的对象，这样做的目的是将类的实例化操作延迟到子类中完成，即由子类来决定究竟应该实例化 (创建) 哪一个类。工厂方法模式的意图是，定义一个用于创建对象的接口，让子类决定实例化哪一个类。Factory Method是一个类的实例化延迟到其子类。工厂方法模式的类图如图8-5所示。

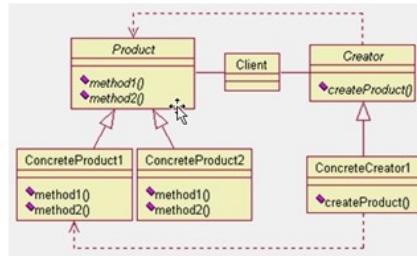


图8-5 类图

其中：

- (1) 类Product定义了Factory Method所创建的对象的接口；
- (2) 类ConcreteProduct用于实现Product接口；
- (3) 类Creator声明了工厂方法，该方法返回一个Product类型的对象。Creator也可以定义一个工厂方法的缺省实现，它返回一个缺省的ConcreteProduct对象。
- (4) 类ConcreteCreator重定义了工厂方法，以返回一个ConcreteProduct实例。

对照两张类图可以看出，与"Creator"角色相对应的类是Bank;与"Product"角色相对应的类是Account.

试题答案

- (33) A (34) B

第 8 章：软件架构设计

试题23 (2010年11月试题35)

试题23 (2010年11月试题35)

- (35) 是一个独立可交付的功能单元，外界通过接口访问其提供的服务。
- (35) A.面向对象系统中的对象 (Object)
 - B.模块化程序设计中的子程序 (Subroutine)
 - C.基于构件开发中的构件 (Component)
 - D.系统模型中的包 (Package)

试题分析

在基于构件的开发中，构件包含并扩展了模块化程序设计中子程序、面向对象系统中对象或类和系统模型中包的思想，它是系统设计、实现和维护的基础。构件定义为通过接口访问服务的一个独立可交付的功能单元。

试题答案

(35) C

第 8 章：软件架构设计

试题24 (2010年11月试题36~37)

试题24 (2010年11月试题36~37)

在基于构件的软件开发中，(36) 描述系统设计蓝图以保证系统提供适当的功能；
(37) 用来了解系统的性能、吞吐率等非功能性属性。

- (36) A.逻辑构件模型B.物理构件模型C.组件接口模型D.系统交互模型
(37) A.逻辑构件模型B.物理构件模型C.组件接口模型D.系统交互模型

试题分析

在基于构件的软件开发中，逻辑构件模型用功能包描述系统的抽象设计，用接口描述每个服务集合，以及功能之间如何交互以满足用户需求，它作为系统的设计蓝图以保证系统提供适当的功能。物理构件模型用技术设施产品、硬件分布和拓扑结构、以及用于绑定的网络和通信协议描述系统的物理设计，这种架构用于了解系统的性能、吞吐率等许多非功能性属性。

试题答案

(36) A (37) B

第 8 章：软件架构设计

试题25 (2010年11月试题38)

试题25 (2010年11月试题38)

对象管理组织 (OMG) 基于CORBA基础设施定义了四种构件标准。其中，(38) 的状态信息是由构件自身而不是由容器维护。

- (38) A.实体构件B.加工构件C.服务构件D.会话构件

试题分析

对象管理组织 (OMG) 基于CORBA基础设施定义了四种构件标准。实体 (Entity) 构件需要长期持久化并主要用于事务性行为，由容器管理其持久化。加工 (Process) 构件同样需要容器管理其持久化，但没有客户端可访问的主键。会话 (Session) 构件不需要容器管理其持久化，其状态信息必须由构件自己管理。服务 (Service) 构件是无状态的。

试题答案

(38) D

试题26 (2010年11月试题39)

试题26 (2010年11月试题39)

分布式系统开发中，通常需要将任务分配到不同的逻辑计算层。业务数据的综合计算分析任务属于 (39) .

- (39) A.表示逻辑层B.应用逻辑层C.数据处理层D.数据层

试题分析

分布式系统开发分为五个逻辑计算层，分别是表示层、表示逻辑层、应用逻辑层、数据处理层、数据层：

表示层实现用户界面。

表示逻辑层为了生成数据表示而必须进行的处理任务，如输入数据编辑等。

应用逻辑层包括为支持实际业务应用和规则所需的应用逻辑和处理过程，如信用检查、数据计算和分析等。

数据处理层包括存储和访问数据库中的数据所需的应用逻辑和命令，如查询语句和存储过程等。

数据层是数据库中实际存储的业务数据。

试题答案

- (39) B

试题27 (2010年11月试题40)

试题27 (2010年11月试题40)

在客户机/服务器系统开发中，采用 (40) 时，应将数据层和数据处理层放置于服务器，应用逻辑层、表示逻辑层和表示层放置于客户机。

- (40) A.分布式表示架构B.分布式应用架构
C.分布式数据和应用架构D.分布式数据架构

试题分析

客户机/服务器系统开发时可以采用不同的分布式计算架构：分布式表示架构是将表示层和表示逻辑层迁移到客户机，应用逻辑层、数据处理层和数据层仍保留在服务器上；分布式数据架构是将数据层和数据处理层放置于服务器，应用逻辑层、表示逻辑层和表示层放置于客户机；分布式数据

和应用架构数据层和数据处理层放置在数据服务器上，应用逻辑层放置在应用服务器上，表示逻辑层和表示层放置在客户机。

试题答案

(40) D

第 8 章：软件架构设计

试题28 (2010年11月试题44)

试题28 (2010年11月试题44)

软件架构是降低成本、改进质量、按时和按需交付产品的关键因素。以下关于软件架构的描述，错误的是 (44) 。

- (44) A.根据用户需求，能够确定一个最佳的软件架构，指导整个软件的开发过程
- B.软件架构设计需要满足系统的质量属性，如性能、安全性和可修改性等
- C.软件架构设计需要确定组件之间的依赖关系，支持项目计划和管理活动
- D.软件架构能够指导设计人员和实现人员的工作

试题分析

软件架构是降低成本、改进质量、按时和按需交付产品的关键因素，软件架构设计需要满足系统的质量属性，如性能、安全性和可修改性等，软件架构设计需要确定组件之间的依赖关系，支持项目计划和管理活动，软件架构能够指导设计人员和实现人员的工作。一般在设计软件架构之初，会根据用户需求，确定多个候选架构，并从中选择一个较优的架构，并随着软件的开发，对这个架构进行微调，以达到最佳效果。

试题答案

(44) A

第 8 章：软件架构设计

试题29 (2010年11月试题45)

试题29 (2010年11月试题45)

软件架构设计包括提出架构模型、产生架构设计和进行设计评审等活动，是一个迭代的过程。

以下关于软件架构设计活动的描述，错误的是 (45) 。

- (45) A.在建立软件架构的初期，一般需要选择一个合适的架构风格
- B.将架构分析阶段已标识的构件映射到架构中，并分析这些构件之间的关系
- C.软件架构设计活动将已标识的构件集成到软件架构中，设计并实现这些构件

D.一旦得到了详细的软件架构设计，需要邀请独立于系统开发的外部人员对系统进行评审

试题分析

软件架构设计包括提出架构模型、产生架构设计和进行设计评审等活动，是一个迭代的过程，在建立软件架构的初期，一般需要选择一个合适的架构风格，将架构分析阶段已标识的构件映射到架构中，并分析这些构件之间的关系，一旦得到了详细的软件架构设计，需要邀请独立于系统开发的外部人员对系统进行评审。一般来说，软件架构设计活动将已标识的构件集成到软件架构中，设计这些构件，但不予以实现。

试题答案

(45) C

第 8 章：软件架构设计

试题30 (2010年11月试题46~47)

试题30 (2010年11月试题46~47)

基于软件架构的设计 (Architecture Based Software Development,ABSD) 强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用 (46) 来描述软件架构，采用 (47) 来描述需求。

(46) A.类图和序列图 B.视角与视图

C.构件和类图 D.构件与功能

(47) A.用例与类图 B.用例与视角

C.用例与质量场景 D.视角与质量场景

试题分析

当考虑架构时，重要的是从不同的视角 (perspective) 来检查，这促使设计师考虑体架构的不同属性。例如：展示功能组织的静态视角能判断质量特性，展示并发行为的动态视角能判断系统行为特性。在ABSD方法中，使用不同的视角来观察设计元素，一个子系统并不总是一个静态的架构元素，而是可以从动态和静态视角观察的架构元素。

将选择的特定视角或视图与Kruchten提出的类似，也就是逻辑视图、进程视图、实现视图和配置视图。使用逻辑视图来记录设计元素的功能和概念接口，设计元素的功能定义了它本身在系统中的角色，这些角色包括功能性能等。进程视图也称为并发视图，使用并发视图来检查系统多用户的并发行为。使用"并发"来代替"进程"，是为了强调没有对进程或线程进行任何操作，一旦这些执行操作，则并发视图就演化为进程视图。使用的最后一个视图是配置视图，配置视图代表了计算机网络中的节点，也就是系统的物理结构。配置视图只能用在多处理器的系统中。

用例已经成为推测系统在一个具体设置中的行为的重要技术，用例被用在很多不同的场合，在本节中，用例是系统的一个给予用户一个结果值的功能点，用例用来捕获功能需求。

正如用例使功能需求具体化一样，用例还必须使质量需求具体化。所谓的"系统必须易于修改"之类的需求是没有多大意义的，因为相对于某些修改而言，所有的系统都是易于修改的。而相对于另

一些修改而言，所有系统又都是难以修改的。所以，上述需求应该按如下格式具体化：“系统应该易于增加下列类型的新功能……”。

在使用用例捕获功能需求的同时，通过定义特定场景来捕获质量需求，并称这些场景为质量场景。这样一来，在一般的软件开发过程中，使用质量场景捕获变更、性能、可靠性和交互性，分别称之为变更场景、性能场景、可靠性场景和交互性场景。质量场景必须包括预期的和非预期的刺激（stimuli）。例如，一个预期的性能场景是估计每年用户数量增加10%的影响，一个非预期的场景是估计每年用户数量增加100%的影响。非预期场景可能不能真正实现，但它们在决定设计的边界条件时很有用。

试题答案

(46) B (47) C

第 8 章：软件架构设计

试题31 (2010年11月试题48)

试题31 (2010年11月试题48)

某游戏公司欲开发一个大型多人即时战略游戏，游戏设计的目标之一是能够支持玩家自行创建战役地图，定义游戏对象的行为和之间的关系。针对该目标，公司应该采用（48）架构风格最为合适。

(48) A.管道-过滤器 B.隐式调用 C.主程序-子程序 D.解释器

试题分析

根据题干描述，该软件系统特别强调用户定义系统中对象的关系和行为这一特性，这需要在软件架构层面提供一种运行时的系统行为定义与改变的能力，根据常见架构风格的特点和适用环境，可以知道最合适的架构设计风格应该是解释器风格。

试题答案

(48) D

第 8 章：软件架构设计

试题32 (2010年11月试题49)

试题32 (2010年11月试题49)

某公司欲为某种型号的示波器开发内置软件。该公司的架构师设计了如图8-6所示的软件架构。在软件架构评审时，专家认为该架构存在的问题是（49）。

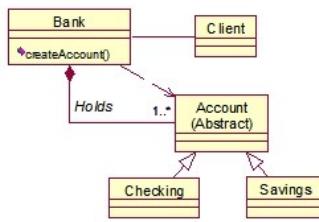


图8-6 软件架构设计图

- (49) A.在功能划分上将各个模块独立起来
 B.在硬件构件的混合和替换方面不是很灵活
 C.没有清晰地说明用户怎样与其交互
 D.没有明确的层次关系，没有强调功能之间的交互

试题分析

根据图中示波器的功能描述，结合示波器常见的功能和使用方式，可以看出图中系统设计最大的缺陷在于没有建模系统与外界，特别是用户之间的交互方式。而与用户的交互无疑是示波器的一个十分重要的功能。

试题答案

- (49) C

第 8 章：软件架构设计

试题33 (2010年11月试题50)

试题33 (2010年11月试题50)

某公司承接了一个开发家用空调自动调温器的任务，调温器测量外部空气温度，根据设定的期望温度控制空调的开关。根据该需求，公司应采用 (50) 架构风格最为合适。

- (50) A.解释器 B.过程控制 C.分层 D.管道-过滤器

试题分析

根据题目描述，调温器需要实时获取外界的温度信息，并与用户定义的温度进行比较并做出动作。根据该系统的应用领域和实际需求，可以看出这是一个典型的过程控制架构风格的应用场景。

试题答案

- (50) B

第 8 章：软件架构设计

试题34 (2010年11月试题51)

试题34 (2010年11月试题51)

某公司欲开发一个漫步者机器人，用来完成火星探测任务。机器人的控制者首先定义探测任务和任务之间的时序依赖性，机器人接受任务后，需要根据自身状态和外界环境进行动态调整，最终自动完成任务。针对这些需求，该机器人应该采用（51）架构风格最为合适。

- (51) A.解释器 B.主程序-子程序 C.隐式调用 D.管道-过滤器

试题分析

根据题目描述，漫步者机器人需要根据自身状态和外界环境进行动态调整，这是一个典型的根据外部事件进行响应的场景。比较4个候选项，隐式调用比较适合根据外部事件进行处理和动作的情景。

试题答案

- (51) C

第8章：软件架构设计

试题35（2010年11月试题52）

试题35（2010年11月试题52）

某公司欲开发一个语音识别系统，语音识别的主要过程包括分割原始语音信号、识别音素、产生候选词、判定语法片断、提供语义解释等。每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用（52）架构风格最为合适。

- (52) A.解释器 B.面向对象 C.黑板 D.隐式调用

试题分析

根据题目描述，语音识别系统是一个十分典型的专家系统，其特点是求解的正确结果不止一个，求解过程比较复杂，需要通过专家知识和反馈逐步得到正确结果。因此对比4个候选项，黑板结构特别适合求解这类问题。

试题答案

- (52) C

第8章：软件架构设计

试题36（2010年11月试题53~54）

试题36（2010年11月试题53~54）

特定领域软件架构（DSSA）是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施DSSA的过程中包括一系列基本的活动，其中（53）活动的主要目的是为了获得DSSA.该活动参加人员中，（54）的主要任务是提供关于领域中系统的需求规约和实现的知识。

(53) A.领域需求 B.领域分析 C.领域设计 D.领域实现

(54) A.领域专家 B.领域分析者 C.领域设计者 D.领域实现者

试题分析

简单地说，DSSA就是在一个特定应用领域中为一组应用提供组织结构参考的标准软件架构。对DSSA研究的角度、关心的问题不同导致了对DSSA的不同定义。

1、DSSA的定义

Hayes Roth对DSSA的定义如下：“DSSA就是专用于一类特定类型的任务（领域）的、在整个领域中能有效地使用的、为成功构造应用系统限定了标准的组合结构的软件构件的集合”。

Tracz的定义为：“DSSA就是一个特定的问题领域中支持一组应用的领域模型、参考需求、参考架构等组成的开发基础，其目标就是支持在一个特定领域中多个应用的生成”。

通过对众多的DSSA的定义和描述的分析，可知DSSA的必备特征为：

- ①一个严格定义的问题域和问题解域。
- ②具有普遍性。使其可以用于领域中某个特定应用的开发。
- ③对整个领域的构件组织模型的恰当抽象。
- ④具备该领域固定的、典型的在开发过程中可重用元素。

一般的DSSA的定义并没有对领域的确定和划分给出明确说明。从功能覆盖的范围角度有两种理解DSSA中领域的含义的方式。

①垂直域：定义了一个特定的系统族，包含整个系统族内的多个系统，结果是在该领域中可作为系统的可行解决方案的一个通用软件架构。

②水平域：定义了在多个系统和多个系统族中功能区域的共有部分。在子系统级上涵盖多个系统族的特定部分功能。

在垂直域上定义的DSSA只能应用于一个成熟的、稳定的领域，但这个条件比较难以满足：若将领域分割成较小的范围，则更相对容易，也容易得到一个一致的解决方案。

2、DSSA的基本活动

实施DSSA的过程中包含了一些基本的活动。虽然具体的DSSA方法可能定义不同的概念、步骤和产品等，但这些基本活动大体上是一致的。以下将分三个阶段介绍这些活动。

(1) 领域分析

这个阶段的主要目标是获得领域模型。领域模型描述领域中系统之间的共同的需求，即领域模型所描述的需求为领域需求。在这个阶段中首先要进行一些准备性的活动，包括定义领域的边界。从而明确分析的对象；识别信息源，整个领域工程过程中信息的来源，可能的信息源包括现存系统、技术文献、问题域和系统开发的专家、用户调查和市场分析、领域演化的历史记录等，在此基础上就可以分析领域中系统的需求，确定哪些需求是领域中的系统广泛共享的，从而建立领域模型。当领域中存在大量系统时，需要选择它们的一个子集作为样本系统。对样本系统需求的考察将显示领域需求的一个变化范围。一些需求对所有被考察的系统是共同的，一些需求是单个系统所独有的。很多需求位于这两个极端之间，即被部分系统共享。

(2) 领域设计

这个阶段的目标是获得DSSA。DSSA描述在领域模型中表示的需求的解决方案，它不是单个系统的表示，而是能够适应领域中多个系统的需求的一个高层次的设计。建立了领域模型之后，就可以派生出满足这些被建模的领域需求的DSSA，由于领域模型中的领域需求具有一定的变化性，DSSA也要相应地具有变化性。它可以通过表示多选一的（alternative）、可选的（optional）解决方案等

来做到这一点。模型和DSSA来组织的，因此在这个阶段通过获得DSSA,也就同时形成了重用基础设施的规约。

(3) 领域实现

这个阶段的主要目标是依据领域模型和DSSA开发和组织可重用信息。这些可重用信息可能是从现有系统中提取得到，也可能需要通过新的开发得到。它们依据领域模型和DSSA进行组织，也就是领域模型和DSSA定义了这些可重用信息的重用时机，从而支持了系统化的软件重用。这个阶段也可以看作重用基础设施的实现阶段。

值得注意的是，以上过程是一个反复的、逐渐求精的过程。在实施领域工程的每个阶段中，都可能返回到以前的步骤，对以前的步骤得到的结果进行修改和完善，再回到当前步骤，在新的基础上进行本阶段的活动。

3、参与DSSA的人员

参与DSSA的人员可以划分为四种角色：领域专家、领域分析师、领域设计人员和领域实现人员。

(1) 领域专家

领域专家可能包括该领域中系统的有经验的用户、从事该领域中系统的需求分析、设计、实现以及项目管理的有经验的软件工程师等。领域专家的主要任务包括提供关于领域中系统的需求规约和实现的知识，帮助组织规范的、一致的领域字典，帮助选择样本系统作为领域工程的依据，复审领域模型、DSSA等领域工程产品等。

领域专家应该熟悉该领域中系统的软件设计和实现、硬件限制、未来的用户需求及技术走向等。

(2) 领域分析人员

领域分析人员应由具有知识工程背景的有经验的系统分析员来担任。领域分析人员的主要任务包括控制整个领域分析过程，进行知识获取，将获取的知识组织到领域模型中，根据现有系统、标准规范等验证领域模型的准确性和一致性，维护领域模型。

领域分析人员应熟悉软件重用和领域分析方法；熟悉进行知识获取和知识表示所需的技术、语言和工具；应具有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互；应具有较高的进行抽象、关联和类比的能力；应具有较高的与他人交互和合作的能力。

(3) 领域设计人员

领域设计人员应由有经验的软件设计人员来担任。领域设计人员的主要任务包括控制整个软件设计过程，根据领域模型和现有的系统开发出DSSA,对DSSA的准确性和一致性进行验证，建立领域模型和DSSA之间的联系。

领域设计人员应熟悉软件重用和领域设计方法；熟悉软件设计方法；应有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互。

(4) 领域实现人员

领域实现人员应由有经验的程序设计人员来担任。领域实现人员的主要任务包括根据领域模型和DSSA,或者从头开发可重用构件，或者利用再工程的技术从现有系统中提取可重用构件，对可重用构件进行验证，建立DSSA与可重用构件间的联系。

领域实现人员应熟悉软件重用、领域实现及软件再工程技术；熟悉程序设计；具有一定的该领域的经验。

4、DSSA的建立过程

因所在的领域不同，DSSA的创建和使用过程也各有差异，Tract曾提出了一个通用的DSSA应用过程，这些过程也需要根据所应用到的领域来进行调整。一般情况下，需要用所应用领域的应用开发者习惯使用的工具和方法来建立DSSA模型。同时Tract强调了DSSA参考架构文档工作的重要性。因为新应用的开发和对现有应用的维护都要以此为基础。

DSSA的建立过程分为五个阶段，每个阶段可以进一步划分为一些步骤或子阶段。每个阶段包括一组需要回答的问题，一组需要的输入，一组将产生的输出和验证标准。本过程是并发的（concurrent）、递归的（recursive）、反复的（iterative）。或者说，它是螺旋模型（spiral）。完成本过程可能需要对每个阶段经历几遍，每次增加更多的细节。

（1）定义领域范围。本阶段的重点是确定什么在感兴趣的领域中以及本过程到何时结束。这个阶段的一个主要输出是领域中的应用需要满足一系列用户的需求。

（2）定义领域特定的元素：本阶段的目标是编译领域字典和领域术语的同义词词典。在领域工程过程的前一个阶段产生的高层块图将被增加更多的细节，特别是识别领域中应用间的共同性和差异性。

（3）定义领域特定的设计和实现需求约束：本阶段的目标是描述解空间中有差别的特性。不仅要识别出约束，并且要记录约束对设计和实现决定造成的后果，还要记录对处理这些问题时产生的所有问题的讨论。

（4）定义领域模型和架构：本阶段的目标是产生一般的架构，并说明构成它们的模块或构件的语法和语义。

（5）产生，搜集可重用的产品单元：本阶段的目标是为DSSA增加构件，使它可以被用来产生问题域中的新应用。

DSSA的建立过程是并发的、递归的和反复进行的。该过程的目的是将用户的需要映射为基于实现限制集合的软件需求，这些需求定义了DSSA。在此之前的领域工程和领域分析过程并没有对系统的功能性需求和实现限制进行区分，而是统称为“需求”。

试题答案

（53）C（54）A

第8章：软件架构设计

试题37（2010年11月试题55~57）

试题37（2010年11月试题55~57）

某服务器软件系统对可用性（Availability）、性能（Performance）和可修改性（Modification）的要求较高，（55）设计策略能提高该系统的可用性，（56）设计策略能够提高该系统的性能，（57）设计策略能够提高该系统的可修改性。

（55）A.Ping/Echo B.限制访问
C.运行时注册 D.接口-实现分离

（56）A.分层结构 B.事务机制 C.主动冗余 D.队列调度

（57）A.信息隐藏 B.记录/回放 C.任务模型 D.回滚

试题分析

同策略主要针对一个或多个软件质量属性，其中Ping/Echo主要提高系统的可用性；限制访问主要提高系统的安全性；运行时注册主要提高系统的可修改性；接口-实现分离主要提高系统的可修改性；主动冗余提高系统的可靠性；队列调度主要提高系统的性能；信息隐藏主要提高系统的可修改性；记录-回放主要提高系统的可测试性，等等。

试题答案

(55) A (56) D (57) A

第 8 章：软件架构设计

试题38 (2010年11月试题58)

试题38 (2010年11月试题58)

某服务器软件系统能够正确运行并得出计算结果，但存在"系统出错后不能在要求的时间内恢复到正常状态"和"对系统进行二次开发时总要超过半年的时间"两个问题，上述问题依次与质量属性中的 (58) 相关。

- (58) A.可用性和性能 B.性能和可修改性
- C.性能和可测试性 D.可用性和可修改性

试题分析

"系统出错后不能在要求的时间内恢复到正常状态",这是对系统错误恢复能力的描述，属于系统可用性的范畴。"对系统进行二次开发时总要超过半年的时间",这是对系统进行调整和维护方面能力的描述，属于系统可修改性的范畴。

试题答案

(58) D

第 8 章：软件架构设计

试题39 (2010年11月试题59)

试题39 (2010年11月试题59)

某公司欲开发一个软件系统的在线文档帮助系统，用户可以在任何一个查询上下文中输入查询关键字，如果当前查询环境下没有相关内容，则系统会将查询按照一定的顺序转发给其他查询环境。基于上述需求，采用 (59) 最为合适。

- (59) A.责任链模式B.桥接模式C.装饰模式D.适配器模式

试题分析

本题所涉及的四种模式中，桥接模式、装饰模式、适配器模式均已经在前面进行了介绍。

责任链 (chain of responsibility) 模式是一种对象的行为型模式，避免请求发送者与接收者耦合在一起，让多个对象都有可能接收请求，将这些对象连接成一条链，并且沿着这条链传递请求，直到有对象处理它为止。职责链模式不保证每个请求都被接受，由于一个请求没有明确的接收者，那么就不能保证它一定会被处理。

根据题干描述，在线文档系统需要根据用户的查询需求逐步将查询请求依次传递，对比4个候选项，其中在责任链模式里，很多对象由每一个对象对其下家的引用而连接起来形成一条链。请求在这个链上传递，直到链上的某一个对象决定处理此请求。因此责任链模式是能够满足该要求的最好模式。

试题答案

(59) A

第 8 章：软件架构设计

试题40 (2010年11月试题60)

试题40 (2010年11月试题60)

某公司欲开发一套窗体图形界面类库。该类库需要包含若干预定义的窗格 (Pane) 对象，例如 TextPane、ListPane 等，窗格之间不允许直接引用。基于该类库的应用由一个包含一组窗格的窗口组成，并需要协调窗格之间的行为。基于该类库，在不引用窗格的前提下实现窗格之间的协作，应用开发者应采用 (60) 最为合适。

(60) A.备忘录模式 B.中介者模式 C.访问者模式 D.迭代器模式

试题分析

本题所涉及的四种模式中，中介者模式、访问者模式、迭代器模式均已在前面进行了介绍。备忘录 (memento) 模式是确保在不破坏封装的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态，这样可以在以后将对象恢复到原先保存的状态。备忘录模式提供了一种状态恢复的实现机制，使得用户可以方便地回到一个特定的历史步骤。

根据题干描述，应用系统需要使用某公司开发的类库，该应用系统由一组窗格组成，应用需要协调窗格之间的行为，并且不能引用窗格自身，在这种要求下，对比4个候选项，其中中介者模式用一个中介对象封装一系列的对象交互。中介者使用各对象不需要显式地相互调用，从而使其耦合松散。可以看出该模式最符合需求。

试题答案

(60) B

试题41 (2010年11月试题61)

试题41 (2010年11月试题61)

某公司开发一个文档编辑器，该编辑器允许在文档中直接嵌入图形对象，但开销很大。用户在系统设计之初提出编辑器在打开文档时必须十分迅速，可以暂时不显示当前页面以外的图形。针对这种需求，公司可以采用 (61) 避免同时创建这些图形对象。

- (61) A.代理模式B.外观模式C.桥接模式D.组合模式

试题分析

本题所涉及的四种模式中，桥接模式和组合模式在前面已经进行了介绍。

(1) 代理 (proxy) 模式。代理模式是一种对象结构型模式，可为某个对象提供一个代理，并由代理对象控制对原对象的引用。代理模式能够协调调用者和被调用者，能够在一定程度上降低系统的耦合度，其缺点是请求的处理速度会变慢，并且实现代理模式需要额外的工作。

(2) 外观 (facade) 模式。外观模式是对象的结构模式，要求外部与一个子系统的通信必须通过一个统一的外观对象进行，为子系统中的一组接口提供一个一致的界面，外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

根据题干描述，该编辑器需要在文档中嵌入显示开销很大的图形对象，为了能够提高系统效率，需要避免同时创建这些图像。对这些要求，对比候选项，可以发现代理模式可以解决直接访问对象时带来的问题，例如：要访问的对象在远程的机器上；对象创建开销很大，或者某些操作需要安全控制，或者需要进程外的访问等。因此代理模式是最为合适的设计模式。

试题答案

- (61) A

试题42 (2010年11月试题62~63)

试题42 (2010年11月试题62~63)

正确识别风险点、非风险点、敏感点和权衡点是进行软件架构评价的关键步骤。其中 (62) 是实现一个特定质量属性的关键特征，该特征为一个或多个软件构件所共有。“改变加密的级别可能会对安全性和性能都产生显着的影响”，这是一个对系统 (63) 的描述。

- (62) A.风险点B.非风险点C.敏感点D.权衡点

- (63) A.风险点B.非风险点C.敏感点D.权衡点

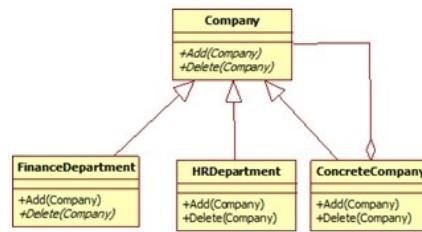
试题分析

正确识别风险点、非风险点、敏感点和权衡点是进行软件架构评价的关键步骤。其中敏感点是实现一个特定质量属性的关键特征，该特征为一个或多个软件构件所共有。系统权衡点会影响一个或多个属性，并对于多个属性来说都是敏感点。基于该定义，可以看出“改变加密的级别可能会对安全性和性能都产生显着的影响”正是一个对系统权衡点的描述。

试题答案

试题43 (2011年11月试题33~34)**试题43 (2011年11月试题33~34)**

某公司欲开发一个门户网站，将公司的各个分公司及办事处信息进行整合。现决定采用 Composite 设计模式来实现公司的组织结构关系，并设计了如图8-7所示的UML类图。图中与 Composite 模式中的“Component”角色相对应的类是 (33)，与“Composite”角色相对应的类是 (34)。

**图8-7 UML类图**

(33) A. Company B. FinanceDepartment

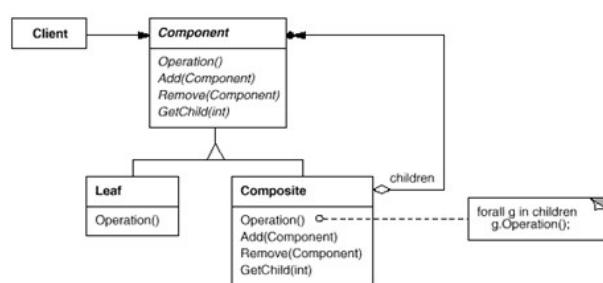
C. HRDepartment D. ConcreteCompany

(34) A. Company B. FinanceDepartment

C. HRDepartment D. ConcreteCompany

试题分析

图8-8为组合模式的UML图例。与题目给出的图例进行匹配可得出答案。

**图8-8 组合模式UML类图****试题答案**

(33) A (34) D

试题44 (2011年11月试题44~45)

试题44 (2011年11月试题44~45)

(44) 描述了一类软件架构的特征，它独立于实际问题，强调软件系统中通用的组织结构选择。垃圾回收机制是Java语言管理内存资源时常用的一种 (45) .

(44) A.架构风格B.开发方法C.设计模式D.分析模式

(45) A.架构风格B.开发方法C.设计模式D.分析模式

试题分析

架构风格往往是从全局的角度来考虑问题，它是一种独立于实际问题的通用组织结构。例如，常用的B/S架构，在很多不同的系统中，都有应用。

而设计模式着眼于解决某一特定的局部问题，是一种局部解决方案的应用。例如，在很多的软件系统中，创建对象时，希望有统一的机制对这些对象的创建进行管理，所以出现了工厂模式，创建者模式等设计模式。而内存垃圾的回收机制也做成了一种设计模式。

试题答案

(44) A (45) C

第 8 章：软件架构设计

试题45 (2011年11月试题46~48)

试题45 (2011年11月试题46~48)

1995年Kruchten提出了著名的"4+1"视图，用来描述软件系统的架构。在"4+1"视图中，

(46) 用来描述设计的对象模型和对象之间的关系； (47) 描述了软件模块的组织与管理；

(48) 描述设计的并发和同步特征。

(46) A.逻辑视图B.用例视图C.过程视图D.开发视图

(47) A.逻辑视图B.用例视图C.过程视图D.开发视图

(48) A.逻辑视图B.用例视图C.过程视图D.开发视图

试题分析

"4+1"视图中的"4"，指的是：逻辑视图、开发视图、进程视图、物理视图，"1"指的是场景视图。

场景视图又称为用例视图，显示外部参与者观察到的系统功能。

逻辑视图从系统的静态结构和动态行为角度显示系统内部如何实现系统的功能。

开发视图又称为实现视图，显示的是源代码以及实际执行代码的组织结构。

处理视图又称为过程视图，显示程序执行时并发的状态。

物理视图展示软件到硬件的映射。

试题答案

(46) A (47) D (48) C

试题46 (2011年11月试题49)

试题46 (2011年11月试题49)

基于架构的软件设计 (ABSD) 强调由商业、质量和功能需求的组合驱动软件架构设计。ABSD 方法有三个基础：功能分解、（49）和软件模板的使用。

- (49) A.对需求进行优先级排列 B.根据需求自行设计系统的总体架构
C.选择架构风格实现质量及商业需求 D.开发系统原型用于测试

试题分析

请参考试题9的分析。

试题答案

- (49) C

试题47 (2011年11月试题50)

试题47 (2011年11月试题50)

某公司研发一种语音识别软件系统，需要对用户的语音指令进行音节分割、重音判断、语法分析和语义分析，最终对用户的意图进行推断。针对上述功能需求，该语音识别软件应该采用（50）架构风格最为合适。

- (50) A.隐式调用 B.管道-过滤器 C.解释器 D.黑板

试题分析

其实从应用的角度来看，这些经典的架构风格提得越来越少了，但这些架构风格有一些经典的应用是要求掌握的。例如，管道-过滤器风格常常用于实现编译器。以规则为中心的虚拟机系统适合于实现专家系统。黑板风格适合于自然语言处理、语音处理、模式识别、图像处理。

试题答案

- (50) D

试题48 (2011年11月试题51)

试题48 (2011年11月试题51)

某企业内部现有的主要业务功能已经封装为Web服务。为了拓展业务范围，需要将现有的业务功能进行多种组合，形成新的业务功能。针对业务灵活组合这一要求，采用 (51) 架构风格最为合适。

- (51) A.管道-过滤器 B.解释器 C.显式调用 D.黑板

试题分析

解释器是指在程序语言定义的计算和有效硬件操作确定的计算之间建立对应的联系。完成信息识别和转换工作。题目中的场景需要用到信息的识别和转换，所以可以用解释器风格。

试题答案

- (51) B

第 8 章：软件架构设计

试题49 (2011年11月试题52)

试题49 (2011年11月试题52)

编译器的主要工作过程是以文本形式输入的代码逐步转化为多种形式，最终生成可执行代码。现代编译器主要关注编译过程和程序的中间表示，围绕程序的各种形态进行转化与处理。针对这种特征，现代编译器应该采用 (52) 架构风格最为合适。

- (52) A.数据共享 B.虚拟机 C.隐式调用 D.管道-过滤器

试题分析

根据题干描述，现代编译器主要关注编译过程和程序的中间表示，围绕程序的各种形态进行转化与处理。这种情况下，可以针对程序的各种形态构建数据库，通过中心数据库进行转换与处理。根据上述分析，选项中列举的架构风格中，数据共享风格最符合要求。

试题答案

- (52) A

第 8 章：软件架构设计

试题50 (2011年11月试题53)

试题50 (2011年11月试题53)

某软件公司正在设计一个通用的嵌入式数据处理平台，需要支持多种数据处理芯片之间的数据传递与交换。该平台的核心功能之一要求能够屏蔽芯片之间的数据交互，使其耦合松散，并且可以独立改变芯片之间的交互过程。针对上述需求，采用（53）最为合适。

- （53）A.抽象工厂模式 B.策略模式 C.中介者模式 D.状态模式

试题分析

在本题的四种设计模式中，中介者模式已经在前面进行了介绍。

（1）抽象工厂（abstract factory）模式。抽象工厂模式又称为Kit模式，属于对象创建型模式。抽象工厂模式是所有形式的工厂模式中最为抽象和最具一般性的一种形态，它提供了一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。在抽象工厂模式中，引入了产品等级结构和产品族的概念，产品等级结构是指抽象产品与具体产品所构成的继承层次关系，产品族是同一个工厂所生产的一系列产品，即位于不同产品等级结构且功能相关联的产品组成的家族。当抽象工厂模式退化到只有一个产品等级结构时，即变成了工厂方法模式。

（2）策略（strategy）模式。策略模式是一种对象的行为型模式，定义一系列算法，并将每一个算法封装起来，并让它们可以相互替换。策略模式让算法独立于使用它的客户而变化，其目的是将行为和环境分隔，当出现新的行为时，只需要实现新的策略类。

（3）状态（state）模式。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，需要事先确定状态种类，这也导致在状态模式中增加新的状态类时将违反开闭原则，新的状态类的引入将需要修改与之能够进行转换的其他状态类的代码。状态模式的使用必然会增加系统类和对象的个数。

根据题干描述，该系统需要能够支持不同芯片之间的数据交互，并能够独立改变芯片之间的数据交互过程。这种情况下，可以引入一个中介层，通过中介层屏蔽不同芯片之间的两两交互。根据上述分析，选项中列举的设计模式中，中介者模式最符合要求。

试题答案

- （53）C

第8章：软件架构设计

试题51（2011年11月试题54）

试题51（2011年11月试题54）

某软件公司正在设计一个图像处理软件，该软件需要支持用户在图像处理过程中的撤销和重做等动作，为了实现该功能，采用（54）最为合适。

- （54）A.单例模式 B.命令模式 C.访问者模式 D.适配器模式

试题分析

在本题的四种设计模式中，命令模式、访问者模式、适配器模式已经在前面进行了介绍。

单例（singleton）模式确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例，这个类称为单例类，它提供全局访问的方法。

根据题干描述，系统需要支持用户在图像处理过程中的撤销和重做的动作，因此可以将用户动作封装成对象，通过对对象之间的传递和转换实现撤销和重做等动作。根据上述分析，选项中列举的设计模式中，命令模式最符合要求。

试题答案

(54) B

第 8 章：软件架构设计

试题52 (2011年11月试题55)

试题52 (2011年11月试题55)

某互联网公司正在设计一套网络聊天系统，为了限制用户在使用该系统时发表不恰当言论，需要对聊天内容进行特定敏感词的过滤。针对上述功能需求，采用 (55) 能够灵活配置敏感词的过滤过程。

(55) A.责任链模式 B.工厂模式 C.组合模式 D.装饰模式

试题分析

本题考查常见设计模式的特点，下面对23种设计模式做个总结。

Abstract Factory (抽象工厂模式)：提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

Adapter (适配器模式)：将一个类的接口转换成客户希望的另外一个接口。Adapter模式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)：将抽象部分与它的实现部分分离，使它们都可以独立地变化。

Builder (建造者模式)：将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

Chain of Responsibility (责任链模式)：为解除请求的发送者和接收者之间耦合，而使多个对象都有机会处理这个请求。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它。

Command (命令模式)：将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可取消的操作。

Composite (组合模式)：将对象组合成树形结构以表示“部分-整体”的层次结构。它使得客户对单个对象和复合对象的使用具有一致性。

Decorator (装饰模式)：动态地给一个对象添加一些额外的职责。就扩展功能而言，它比生成子类方式更为灵活。

Facade (外观模式)：为子系统中的一组接口提供一个一致的界面，Facade模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

Factory Method (工厂模式)：定义一个用于创建对象的接口，让子类决定将哪一个类实例化。Factory Method使一个类的实例化延迟到其子类。

Flyweight (享元模式)：运用共享技术有效地支持大量细粒度的对象。

Interpreter (解释器模式) : 给定一个语言 , 定义其文法的一种表示 , 并定义一个解释器 , 该解释器使用该表示来解释语言中的句子。

Iterator (重述模式) : 提供一种方法顺序访问一个聚合对象中的各个元素 , 而又不需暴露该对象的内部表示。

Mediator (中介模式) : 用一个中介对象来封装一系列的对象交互。中介者使各对象不需要显式地相互引用 , 从而使其耦合松散 , 而且可以独立地改变它们之间的交互。

Memento (备忘录模式) : 在不破坏封装性的前提下 , 捕获一个对象的内部状态 , 并在该对象之外保存这个状态。这样以后就可将该对象恢复到保存的状态。

Observer (观察者模式) : 定义对象间的一种一对多的依赖关系 , 以便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自动刷新。

Prototype (原型模式) : 用原型实例指定创建对象的种类 , 并且通过拷贝这个原型来创建新的对象。

Proxy (代理模式) : 为其他对象提供一个代理以控制对这个对象的访问。

Singleton (单例模式) : 保证一个类仅有一个实例 , 并提供一个访问它的全局访问点。

State (状态模式) : 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所属的类。

Strategy (策略模式) : 定义一系列的算法 , 把它们一个个封装起来 , 并且使它们可相互替换。本模式使得算法的变化可独立于使用它的客户。

Template Method (骨架模式) : 定义一个操作中的算法的骨架 , 而将一些步骤延迟到子类中。Template Method使得子类可以不改变一个算法的结构 , 即可重定义该算法的某些特定步骤。

Visitor (访问者模式) : 表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作。

依据题意 , 需要限制用户在使用聊天系统时发表不恰当言论 , 需要对聊天内容进行特定敏感词的过滤 , 最为关键的一点是需要灵活配置过滤关键字。如果本系统采用责任链模式 , 即可达到这一点。

试题答案

(55) A

第 8 章 : 软件架构设计

试题53 (2011年11月试题56~57)

试题53 (2011年11月试题56~57)

某公司在对一家用车库门嵌入式软件系统进行架构设计时 , 识别出两个关键的质量属性场景 , 其中 " 当车库门正常下降时 , 如果发现下面有障碍物 , 则系统停止下降的时间需要控制在 0.1 秒内 " 与 (56) 质量属性相关 ; " 系统需要为部署在远程 PC 上的智能家居系统留有控制接口 , 并支持在智能家居系统中对该系统进行远程错误诊断与调试 " 与 (57) 质量属性相关。

(56) A. 可用性 B. 性能 C. 可修改性 D. 可测试性

(57) A.可用性 B.性能 C.可修改性 D.可测试性

试题分析

在解题之前，考生应了解基本质量属性的含义。

① 性能：性能 (performance) 是指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段事件内系统所能处理的事件的个数。经常用单位事件内所处理事务的数量或系统完成某个事务处理所需的时间来对性能进行定量的表示。性能测试经常要使用基准测试程序。

② 可靠性：可靠性 (reliability) 是软件系统在应用或系统错误面前，在意外或错误使用的情况下维持软件系统的功能特性的基本能力。可靠性是最重要的软件特性，通常用它衡量在规定的条件和时间内，软件完成规定功能的能力。可靠性通常用平均失效等待时间 (mean time to failure,MTTF) 和平均失效间隔时间 (mean time between failure,MTBF) 来衡量。在失效率为常数和修复时间很短的情况下，MTTF和MTBF几乎相等。

③ 可用性：可用性 (availability) 是系统能够正常运行的时间比例。经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。

④ 安全性：安全性 (security) 是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。安全性是根据系统可能受到的安全威胁的类型来分类的。安全性又可划分为机密性、完整性、不可否认性及可控性等特性。其中，机密性保证信息不泄露给未授权的用户、实体或过程；完整性保证信息的完整和准确，防止信息被非法修改；可控性保证对信息的传播及内容具有控制的能力，防止为非法者所用。

⑤ 可修改性：可修改性 (modifiability) 是指能够快速地以较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准，通过考察这些变更的代价衡量可修改性。可修改性包含四个方面。

可维护性 (maintainability)。这主要体现在问题的修复上：在错误发生后“修复”软件系统。为可维护性做好准备的软件体系结构往往能做局部性的修改，并能使对其他构件的负面影响最小化。

可扩展性 (extendibility)。这一点关注的是使用新特性来扩展软件系统，以及使用改进版本来替换构件并删除不需要或不必要的特性和构件。为了实现可扩展性，软件系统需要松散耦合的构件。其目标是实现一种体系结构，它能使开发人员在不影响构件客户的情况下替换构件。支持把新构件集成到现有的体系结构中也是必要的。

结构重组 (reassemble)。结构重组处理的是重新组织软件系统的构件及构件间的关系，例如通过将构件移动到一个不同的子系统而改变它的位置。为了支持结构重组，软件系统需要精心设计构件之间的关系。理想情况下，它们允许开发人员在不影响实现的主体部分的情况下灵活地配置构件。

可移植性 (portability)。可移植性使软件系统适用于多种硬件平台、用户界面、操作系统、编程语言或编译器。为了实现可移植，需要按照硬件无关的方式组织软件系统，其他软件系统和环境被提取出。可移植性是系统能够在不同计算环境下运行的能力。这些环境可能是硬件、软件，也可能是两者的结合。在关于某个特定计算环境的所有假设都集中在一个构件中时，系统是可移植的。如果移植到新的系统需要做些更改，则可移植性就是一种特殊的可修改性。

⑥ 功能性：功能性 (functionality) 是系统能够完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。

⑦ 可变性

可变性 (changeability) 是指体系结构经扩充或变更而成为新体系结构的能力。这种新体系结构应该符合预先定义的规则，在某些具体方面不同于原有的体系结构。当要将某个体系结构作为一个系列相关产品（例如，软件产品线）的基础时，可变性是很重要的。

⑧ 互操作性

作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。为了支持互操作性 (inter-operation)，软件体系结构必须为外部可视的功能特性和数据结构提供精心设计的软件入口。程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题，这种互操作性也影响应用的软件体系结构。

试题答案

(56) B (57) D

第 8 章：软件架构设计

试题54 (2011年11月试题58~60)

试题54 (2011年11月试题58~60)

软件质量属性通常需要采用特定的设计策略实现。例如，(58) 设计策略能提高该系统的可用性，(59) 设计策略能够提高该系统的性能，(60) 设计策略能够提高该系统的安全性。

- (58) A.心跳机制 B.数据驱动 C.关注点分离 D.信息隐藏
- (59) A.引入中间层 B.事务机制 C.主动冗余 D.优先级队列
- (60) A.信息隐藏 B.内置监控器 C.限制访问 D.检查点

试题分析

提高可用性的手段包括：命令/响应机制、心跳机制、异常处理机制、冗余机制等。

提高性能的手段包括：引入并发、维持数据或计算的多个副本、增加可用资源、控制采样频率、限制执行时间、固定优先级调度等。

提高安全性的手段包括：身份认证、限制访问、检测攻击、维护完整性等。

试题答案

(58) A (59) D (60) C

第 8 章：软件架构设计

试题55 (2011年11月试题61)

试题55 (2011年11月试题61)

架构权衡分析方法 (ATAM) 是一种常用的软件架构评估方法 , 下列关于该方法的叙述中 , 正确的是 (61) .

- (61) A.ATAM需要对代码的质量进行评估
- B.ATAM需要对软件系统需求的正确性进行评价
- C.ATAM需要对软件系统进行集成测试
- D.ATAM需要对软件质量属性进行优先级排序

试题分析

ATAM是评价软件构架的一种综合全面的方法。这种方法不仅可以揭示出构架满足特定质量目标的情况 , 而且 (因为它认识到了构架决策会影响多个质量属性) 可以使我们更清楚地认识到质量目标之间的联系 -- 即如何权衡诸多质量目标。

ATAM是针对软件架构的评估方法 , 其层次较高 , 不会涉及具体代码质量的评估 , 所以 A 选项不正确。而对于软件系统需求的正确性评价 , 应是需求验证的主要工作 , 也非 ATAM 所关注的内容。集成测试是在软件开发的测试阶段需要完成的任务 , 此时 , 架构设计、架构评审 (即用 ATAM,SAAM 进行软件架构评审) 、软件详细设计、编码、单元测试工作都已完成 , 所以该工作 , 也非 ATAM 所关注的内容。只有 D 选项的属性优先级排序是 ATAM 所要做的。

试题答案

(61) D

第 8 章 : 软件架构设计

试题 56 (2011 年 11 月试题 62~63)

试题 56 (2011 年 11 月试题 62~63)

识别风险点、非风险点、敏感点和权衡点是软件架构评估过程中的关键步骤。针对某系统所做的架构设计中 , " 系统需要支持的最大并发用户数量直接影响传输协议和数据格式 " 描述了系统架构设计中的一个 (62) ; " 由于系统的业务逻辑目前尚不清楚 , 因此现有系统三层架构中的第二层可能会出现功能重复 , 这会影响系统的可修改性 " 描述了系统架构设计中的一个 (63) .

- (62) A. 敏感点 B. 风险点 C. 非风险点 D. 权衡点
- (63) A. 敏感点 B. 风险点 C. 非风险点 D. 权衡点

试题分析

风险点与非风险点不是以标准专业术语形式出现的 , 只是一个常规概念 , 即可能引起风险的因素 , 可称为风险点。

敏感点是一个或多个构件 (和 / 或构件之间的关系) 的特性。研究敏感点可使设计人员或分析员明确在搞清楚如何实现质量目标时应注意什么。

权衡点是影响多个质量属性的特性 , 是多个质量属性的敏感点。例如 , 改变加密级别可能会对安全性和性能产生非常重要的影响。提高加密级别可以提高安全性 , 但可能要耗费更多的处理时间 , 影响系统性能。如果某个机密消息的处理有严格的时间延迟要求 , 则加密级别可能就会成为一个权衡点。

试题答案

(62) A (63) B

第 8 章：软件架构设计

试题57 (2012年11月试题9)

试题57 (2012年11月试题9)

以下关于软件中间件的叙述，错误的是（9）。

- (9) A.中间件通过标准接口实现与应用程序的关联，提供特定功能的服务
- B.使用中间件可以提高应用软件可移植性
- C.使用中间件将增加应用软件设计的复杂度
- D.使用中间件有助于提高开发效率

试题分析

中间件是一类较为特殊的构件。中间件工作于操作系统与应用程序之间，分布式应用软件借助这种软件在不同的技术之间共享资源。中间件有以下几种类型。

远程过程调用：它是一种广泛使用的分布式应用程序处理方法。应用程序使用RPC来远程执行一个位于不同地址空间里的过程，并且从效果上看和执行本地调用相同。要注意的是，这里的“远程”既可以指不同的计算机，也可以指同一台计算机上的不同进程。一个RPC应用可分为两个部分，分别是服务器和客户。这里的“服务器”和“客户”是指逻辑上的进程，而不是指物理计算机。

面向消息的中间件：利用高效可靠的消息传递机制进行平台无关的数据交换，并基于数据通信来进行分布式系统的集成。通过提供消息传递和消息排队模型，它可在分布式环境下扩展进程间的通信，并支持多种通信协议、语言、应用程序、硬件和软件平台。例如，IBM的MQSeries、BEA的MessageQ等都属于面向消息的中间件产品。

事务处理监控器：也称为交易中间件，是当前应用最广泛的中间件之一。它能支持数以万计的客户进程对服务器的并发访问，使系统具有极强的扩展性，因此，适于电信、金融、证券等拥有大量客户的领域。在对效率、可靠性要求严格的关键任务系统中具有明显优势。TPM一般支持负载均衡，支持分布式两阶段提交，保证事务完整性和数据完整性，并具有安全认证和故障恢复等功能，能很好地满足应用开发的要求。

数据库访问中间件：通过一个抽象层访问数据库的技术，从而允许使用相同或相似的代码访问不同的数据库资源。例如常见的ODBC与JDBC就属于数据库访问中间件。

通过引入中间件技术，可以使应用软件可移植性提高、开发效率提高，同时由于一些复杂的应用程序之间的通信可由中间件完成，所以还降低了应用软件设计的复杂性。所以C选项的描述不正确。

试题答案

(9) C

试题58 (2012年11月试题16)

试题58 (2012年11月试题16)

以下关于软件架构风格与系统性能关系的叙述，错误的是 (16) .

- (16) A.对于采用层次化架构风格的系统，划分的层次越多，系统的性能越差
- B.对于采用管道-过滤器架构风格的系统，可以通过引入过滤器的数据并发处理提高系统性能
- C.对于采用面向对象架构风格的系统，可以通过减少功能调用层次提高系统性能
- D.对于采用过程调用架构风格的系统，可以通过将显式调用策略替换为隐式调用策略提高系统性能。

试题分析

对于采用层次化架构风格的系统，划分的层次越多，系统完成某项功能需要的中间调用操作越多，其性能越差。对于采用管道-过滤器架构风格的系统，可以通过引入过滤器的数据并发处理可以有效地提高系统性能。对于采用面向对象架构风格的系统，可以通过减少功能调用层次提高系统性能。对于采用过程调用架构风格的系统，将显式调用策略替换为隐式调用策略能够提高系统的灵活性，但会降低系统的性能。

试题答案

- (16) D

试题59 (2012年11月试题38)

试题59 (2012年11月试题38)

架构描述语言 (Architecture Description Language,ADL) 是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言。ADL主要包括以下组成部分：组件、组件接口、

(38) 和架构配置。

- (38) A.架构风格 B.架构实现 C.连接件 D.组件实现

试题分析

架构描述语言 (Architecture Description Language,ADL) 是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言，基于底层语义的工具为架构的表示、分析、演化、细化、设计过程等提供支持。ADL的三个基本元素是：

- (1) 构件及接口：计算或数据存储单元。
- (2) 连接件：用于构件之间交互建模的架构构造块及其支配这些交互的规则。
- (3) 架构配置：描述架构的构件与连接件的连接图。

主要的架构描述语言有Aesop、MetaH、C2、Rapide、SADL、Unicon和Wright等，尽管它们都描述软件架构，却有不同的特点。Aesop支持架构风格的应用，MetaH为设计师提供了关于实时电子控制软件系统的设计指导，C2支持基于消息传递风格的用户界面系统的描述，Rapide支持架构设计的模拟并提供了分析模拟结果的工具，SADL提供了关于架构的形式化基础，Unicon支持异构的构件和连接类型并提供了关于架构的高层编译器，Wright支持架构构件之间交互的说明和分析。

各种ADL强调了架构不同的侧面，对架构的研究和应用起到了重要的作用，但也有负面影响。每一种ADL都以独立的形式存在，描述语法不同且互不兼容，同时又有许多共同的特征，这使设计人员很难选择一种合适的ADL，若设计特定领域的架构又需要从头开始描述。

试题答案

(38) C

第 8 章：软件架构设计

试题60 (2012年11月试题39~41)

试题60 (2012年11月试题39~41)

ANSI IEEE 1471-2000是对软件密集型系统的架构进行描述的标准。在该标准中，(39)这一概念主要用于描述软件架构模型。在此基础上，通常采用 (40) 描述某个利益相关人 (Stakeholder) 所关注架构模型的某一方面。 (41) 则是对所有利益相关人关注点的响应和回答。

(39) A.上下文 B.架构风格 C.组件 D.视图

(40) A.环境 B.资源 C.视角 D.场景

(41) A.架构 B.系统 C.模型 D.使命

试题分析

在ANSI/IEEE 1471-2000标准中，系统是为了达成利益相关人 (Stakeholder) 的某些使命 (Mission)，在特定环境 (Environment) 中构建的。每一个系统都有一个架构 (Architecture)。架构 (Architecture) 是对所有利益相关人的关注点 (Concern) 的响应和回答，通过架构描述 (Architecture Description) 来说明。每一个利益相关人都有各自的关注点。这些关注点是指对其重要的，与系统的开发、运营或其他方面相关的利益。架构描述 (Architecture Description) 本质上是多视图的。每一个视图 (View) 是从一个特定的视角 (Viewpoint) 来表述架构的某一个独立的方面。试图用一个单一的视图来覆盖所有的关注点当然是最好的，但实际上这种表达方式将很难理解。视角 (Viewpoint) 的选择，基于要解决哪些利益相关人的哪些关注点。它决定了用来创建视图的语言、符号和模型等，以及任何与创建视图相关的建模方法或者分析技术。一个视图 (View) 包括一个或者多个架构模型 (Model)，一个模型也可能参与多个视图。模型较文本表述的好处在于，可以更容易的可视化、检查、分析、管理和集成。

试题答案

(39) D (40) C (41) A

试题61 (2012年11月试题42~43)

试题61 (2012年11月试题42~43)

采用以架构为核心的软件开发方法，在建立软件架构的初期，首要任务是选择一个合适的（42），在此基础上，开发人员通过架构模型，可以获得关于（43）的理解，为将来的架构实现与演化过程建立了目标。

(42) A.分析模式 B.设计模式 C.架构风格 D.架构标准

(43) A.架构需求 B.架构属性 C.架构优先级 D.架构约束

试题分析

在以架构为核心的软件开发方法中，架构用来激发和调整设计策略，不同的视图用来表达与质量目标有关的信息。架构设计是一个迭代过程，在建立软件架构的初期，选择一个合适的架构风格是首要的，在此基础上，开发人员通过架构模型，可以获得关于软件架构属性的理解，为将来的架构实现与演化过程建立了目标。

试题答案

(42) C (43) B

试题62 (2012年11月试题49~50)

试题62 (2012年11月试题49~50)

若系统中的某子模块需要为其他模块提供访问不同数据库系统的功能，这些数据库系统提供的访问接口有一定的差异，但访问过程却都是相同的，例如，先连接数据库，再打开数据库，最后对数据进行查询。针对上述需求，可以采用（49）设计模式抽象出相同的数据库访问过程，该设计模式（50）。

(49) A.外观 B.装饰 C.桥接 D.享元

(50) A.可以动态、透明地给单个对象添加职责
B.为子系统定义了一个高层接口，这个接口使得这一子系统更加容易使用
C.通过运用共享技术，有效支持大量细粒度的对象
D.将抽象部分与它的实现部分分离，使它们都可以独立地变化

试题分析

外观 (Facade) 模式是对象的结构模式 , 要求外部与一个子系统的通信必须通过一个统一的外观对象进行 , 为子系统中的一组接口提供一个一致的界面 , 外观模式定义了一个高层接口 , 这个接口使得这一子系统更加容易使用。

试题答案

(49) A (50) B

第 8 章 : 软件架构设计

试题63 (2012年11月试题51~53)

试题63 (2012年11月试题51~53)

某软件公司欲设计一款图像处理软件 , 帮助用户对拍摄的照片进行后期处理。在软件需求分析阶段 , 公司的系统分析师识别出了如下3个关键需求。

图像处理软件需要记录用户在处理照片时所有动作 , 并能够支持用户动作的撤销与重做等行为。

图像处理软件需要根据当前正在处理的照片的不同特征选择合适的处理操作 , 处理操作与照片特征之间具有较为复杂的逻辑关系。

图像处理软件需要封装各种图像处理算法 , 用户能够根据需要灵活选择合适的处理算法 ; 软件还要支持高级用户根据一定的规则添加自定义处理算法。

在系统设计阶段 , 公司的架构师决定采用设计模式满足上述关键需求中对系统灵活性与扩展性的要求。具体来说 , 为了支持灵活的撤销与重做等行为 , 采用 (51) 最为合适 ; 为了封装图像操作与照片特征之间的复杂逻辑关系 , 采用 (52) 最为合适 ; 为了实现图像处理算法的灵活选择与替换 , 采用 (53) 最为合适。

(51) A. 工厂模式 B. 责任链模式 C. 中介者模式 D. 命令模式

(52) A. 状态模式 B. 适配器模式 C. 组合模式 D. 单例模式

(53) A. 模板方法模式 B. 访问者模式 C. 策略模式 D. 观察者模式

试题分析

题干描述了某软件公司一款图像处理软件的需求分析与设计过程 , 并明确指出采用设计模式实现关键需求对系统灵活性与扩展性的要求。

针对需求1,为了支持灵活的撤销与重做等行为 , 采用命令模式最为合适 , 因为命令模式可以将一个请求封装为一个对象 , 从而使你可用不同的请求对客户进行参数化 , 还可以对请求排队 , 或记录请求日志 , 以及支持可撤销的操作。

针对需求2,为了封装图像操作与照片特征之间的复杂逻辑关系 , 采用状态模式最为合适 , 因为状态模式将每一个条件分支放入一个独立的类中 , 这样就可以根据对象自身的情况将对象的状态作为一个对象 , 这一对象可以不依赖于其他对象而独立变化。

针对需求3,为了实现图像处理算法的灵活选择与替换 , 采用策略模式最为合适 , 因为策略模式定义一系列的算法 , 把它们封装起来 , 并且使它们可相互替换 , 使得算法可独立于使用它的客户而变化。

试题答案

(51) D (52) A (53) C

第 8 章：软件架构设计

试题64 (2012年11月试题54~55)

试题64 (2012年11月试题54~55)

特定领域软件架构 (Domain Specific Software Architecture.DSSA) 是在一个特定应用领域中 , 为一组应用提供组织结构参考的标准软件体系结构。DSSA的基本活动包括领域分析、领域设计和领域实现。其中领域分析的主要目的是获得 (54) , 从而描述领域中系统之间共同的需求 , 即领域需求 ; 领域设计的主要目标是获得 (55) , 从而描述领域模型中表示需求的解决方案 ; 领域实现的主要目标是开发和组织可重用信息 , 并对基础软件架构进行实现。

- (54) A. 领域边界 B. 领域信息 C. 领域对象 D. 领域模型
- (55) A. 特定领域软件需求 B. 特定领域软件架构
- C. 特定领域软件设计模型 D. 特定领域软件重用模型

试题分析

DSSA以一个特定问题领域为对象 , 形成由领域参考模型、参考需求、参考架构等组成的开发基础架构 , 其目标是支持一个特定领域中多个应用的生成。DSSA的基本活动包括领域分析、领域设计和领域实现。其中领域分析的主要目的是获得领域模型 , 领域模型描述领域中系统之间共同的需求 , 即领域需求 ; 领域设计的主要目标是获得DSSA,DSSA描述领域模型中表示需求的解决方案 ; 领域实现的主要目标是依据领域模型和DSSA开发和组织可重用信息 , 并对基础软件架构进行实现。

试题答案

(54) D (55) B

第 8 章：软件架构设计

试题65 (2012年11月试题56~61)

试题65 (2012年11月试题56~61)

某公司欲开发一个在线交易系统 , 在架构设计阶段 , 公司的架构师识别出3个核心质量属性场景。其中 " 在并发用户数量为 1000 人时 , 用户的交易请求需要在 0.5 秒内得到响应 " 主要与 (56) 质量属性相关 , 通常可采用 (57) 架构策略实现该属性 ; " 当系统由于软件故障意外崩溃后 , 需要在 0.5 小时内恢复正常运行 " 主要与 (58) 质量属性相关 , 通常可采用 (59) 架构策略实现该属

性；"系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录"主要与（60）质量属性相关，通常可采用（61）架构策略实现该属性。

- （56）A.性能 B.吞吐量 C.可靠性 D.可修改性
- （57）A.操作串行化 B.资源调度 C.心跳 D.内置监控器
- （58）A.可测试性 B.易用性 C.可用性 D.互操作性
- （59）A.主动冗余 B.信息隐藏 C.抽象接口 D.记录/回放
- （60）A.可用性 B.安全性 C.可测试性 D.可修改性
- （61）A.内置监控器 B.记录/回放 C.追踪审计 D.维护现有接口

试题分析

对于题干描述："在并发用户数量为1000人时，用户的交易请求需要在0.5秒内得到响应"，主要与性能这一质量属性相关，实现该属性的常见架构策略包括：增加计算资源、减少计算开销、引入并发机制、采用资源调度等。

"当系统由于软件故障意外崩溃后，需要在0.5小时内恢复正常运行"主要与可用性质量属性相关，通常可采用心跳、Ping/Echo、主动冗余、被动冗余、选举等架构策略实现该属性。

"系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录"主要与安全性质量属性相关，通常可采用入侵检测、用户认证、用户授权、追踪审计等架构策略实现该属性。

试题答案

- （56）A
- （57）B
- （58）C
- （59）A
- （60）B
- （61）C

第8章：软件架构设计

试题66（2012年11月试题62~63）

试题66（2012年11月试题62~63）

基于场景的架构分析方法（Scenarios-based Architecture Analysis Method, SAAM）是美国卡耐基梅隆大学软件工程研究所的Kazman等人于1983年提出的一种非功能质量属性的架构分析方法，是最早形成文档并得到广泛应用的软件架构分析方法。SAAM的主要输入是问题描述、（62）和架构描述文档，其分析过程主要包括场景开发、（63）、单个场景评估、场景交互和总体评估。

- （62）A.问题说明 B.问题建模 C.需求说明 D.需求建模
- （63）A.架构需求 B.架构描述 C.架构设计 D.架构实现

试题分析

SAAM方法是最早形成文档并得到广泛应用的软件架构分析方法，最初是用来分析架构的可修改性的，但实践证明，SAAM方法也可用于对许多其他质量属性及系统功能进行快速评估。SAAM方法的目的是验证基本的架构假设和原则，评估架构固有的风险。SAAM指导对架构的检查，使其主要关注潜在的问题点，例如，需求冲突等。SAAM不仅能够评估架构对于特定系统需求的使用能力，也能用来比较不同的架构。

1. 评估活动

与ATAM方法相比，SAAM比较简单，这种方法易学易用，进行培训和准备的工作量都比较少。SAAM评估可以分六个步骤进行，如图8-9所示。

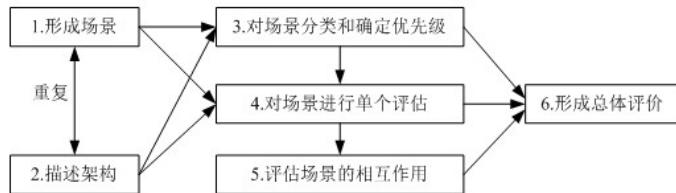


图8-9 SAAM方法的步骤

在这些步骤进行之前，通常有必要对系统做简要的介绍，包括对架构的业务目标的说明等。

(1) 形成场景。在形成场景的过程中，要注意全面捕捉系统的主要用途、系统用户类型、系统将来可能的变更、系统在当前及可预见的未来必须满足的质量属性等信息。只有这样，形成的场景才能代表与各种项目干系人相关的任务。形成场景是通过集中讨论来实现的，使项目干系人在一个友好的氛围中提出一些场景，这些场景反映了他们的需求，也体现了他们对架构将如何实现需求的认识。

(2) 描述架构。在这一步，架构设计师应该采用参加评估的所有人员都能够充分理解的形式，对待评估的架构进行适当的描述。这种描述必须要说明系统中的运算和数据构件，以及它们之间的联系。除了要描述这些静态特性外，还要对系统在某段时间内的动态特征做出说明。描述既可采用自然语言，也可采用形式化的手段。

(3) 场景的分类和优先级确定。场景可分为直接场景和间接场景（潜在场景）。直接场景是按照现有架构开发出来的系统能够直接实现的场景。与在设计时已经考虑过的需求相对应的直接场景能增进对架构的理解，促进对诸如性能和可靠性等其他质量属性的研究；间接场景就是需要对现有架构做某些修改才能支持的场景。间接场景对衡量架构对系统在演化过程中将出现的变更的适用情况十分关键。通过各种间接场景对架构的影响，可以确定架构在相关系统的生命周期内对不断演化的使用的适应情况。直接场景类似于用例，而间接场景有时也叫变更案例。评估人员通过对场景设置优先级，可以保证在评估的有限时间内考虑最重要的场景。这里的“重要”完全是由项目干系人及其所关心的问题确定的。项目干系人可以通过投票表达所关心的问题。

(4) 对场景进行单个评估。一旦确定了要考虑的一组场景，就要把这些场景与架构的描述对应起来。对于直接场景而言，架构设计师需要讲清所评估的架构将如何执行这些场景；对于间接场景而言，设计师应说明需要对架构做哪些修改才能适应间接场景的要求。

(5) 评估场景的相互作用。场景的相互作用暴露了设计方案中的功能分配。场景相互作用的多少与结构复杂性、耦合度、内聚性有关。同时，场景的相互作用能够暴露出架构设计文档未能充分说明的结构分解。

(6) 形成总体评估。最后，评估人员要对场景和场景之间的交互作一个总体的权衡和评价，这一权衡反映组织对表现在不同场景中的目标的考虑优先级。根据对系统成功的相对重要性来为每个场景设置一个权值，权值的确定通常要与每个场景所支持的业务目标联系起来。

如果是要比较多个架构，或者针对同一架构提出了多个不同的方案，则可通过权值的确定来得出总体评价。权值的设置具有很强的主观性，所以，应该让所有项目干系人共同参与，但也应合理组织，要允许对权值及其基本思想进行公开讨论。

2. 评估结果

SAAM评估的主要有形输出包括以下两个方面：

(1) 把代表了未来可能做的更改的场景与架构对应起来，显现出架构中未来可能会表现出较高复杂性的地方，并对每个这样的更改的预期工作量做出评估。

(2) 理解系统的功能，对多个架构所支持的功能和数量进行比较。

如果所评估的是一个架构，SAAM评估将指明架构中未能满足其修改性需求的地方，有时还会指出一种效果更好的设计。SAAM评估也能对多个备选架构进行比较，明确其中哪一个架构能够较好地满足质量属性，而且做的更改较少、不会在未来导致太多的问题。

试题答案

(62) C (63) B

考点分析



第9章应用数学

9.1考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 概率统计应用。
- (2) 图论应用。
- (3) 组合分析。
- (4) 算法（数值算法与非数值算法）的选择与应用。
- (5) 运筹方法（网络计划技术、线性规划、预测、决策、库存管理、模拟）。
- (6) 数学建模。

从历年的考试情况来看，本章的考点主要集中于数学建模、决策论、线性规划、图论。具体考查知识点分布情况如表9-1所示。

表9-1历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题 69~70	数学建模、转移矩阵
2010年11月试题 69~70	数学建模、盈亏平衡点
2011年11月试题 69~70	线性规划
2012年11月试题 69~70	函数曲线、项目成本分析

试题1 (2009年11月试题69)**9.2 试题精解****试题1 (2009年11月试题69)**

对实际应用问题建立了数学模型后，一般还需要对该模型进行检验。通过检验尽可能找出模型中的问题，以利于改进模型，有时还可能会否定该模型。检验模型的做法有多种，但一般不会 (69)。

- (69) A. 利用实际案例数据对模型进行检验
- B. 进行逻辑检验，分析该模型是否会出现矛盾
- C. 用计算机模拟实际问题来检验模型
- D. 检验该模型所采用的技术能否被企业负责人理解

试题分析

针对实际问题建立的数学模型往往是近似的，往往忽略了许多复杂因素。这种模型能否解决实际问题还需要检验。检验的方法有多种。

利用实际案例数据对模型进行检验是很常见的。将模型作为一个黑盒，通过案例数据的输入，检查其输出是否合理。这是应用人员常用的方法。

有时可以请专家来分析模型是否合理。经验丰富的专家一般会根据模型自身的逻辑，再结合实际情况，分析是否会出现矛盾或问题。

有时很难用实际案例或聘请专家来检验模型，例如，试验或实验的代价太大，难以取得实际案例，有的项目技术比较新，缺乏有经验的专家。这时，如果能利用计算机来模拟实际问题，再在计算机上检验该数学模型，这往往是一种有效的办法。例如，对某种核辐射防护建立的数学模型，采用计算机模拟方法来检验就十分有效。

企业负责人需要提供一切必要的支持来解决实际问题。至于解决过程中采用的技术问题，则需要由技术人员研究决定。企业负责人只需要听取汇报，从宏观上认可就可以，不需要理解其中的技术细节。

试题答案

(69) D

试题2 (2009年11月试题70)**试题2 (2009年11月试题70)**

某类产品n种品牌在某地区的市场占有率常用概率向量 $u = (u_1, u_2, \dots, u_n)$ 表示（各分量分别表示各品牌的市场占有率，值非负，且总和为1）。市场占有率每隔一定时间的变化常用转移矩阵

$P^n \cdot n$ 表示。设初始时刻的市场占有率为向量 u , 则下一时刻的市场占有率就是 uP , 再下一时刻的市场占有率为 uP^2 , ... 如果在相当长时期内, 该转移矩阵的元素均是常数, 则市场占有率会逐步稳定到某个概率向量 z , 即出现 $ZP = Z$. 这种稳定的市场占有率体现了转移矩阵的特征, 与初始时刻的市场占有率无关。

假设占领某地区市场的冰箱品牌A与B, 每月市场占有率的变化可用如下常数转移矩阵来描述:

$$P = \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix}$$

则冰箱品牌A与B在该地区最终将逐步稳定到市场占有率 (70) .

- (70) A. (1/4, 3/4) B. (1/3, 2/3) C. (1/2, 1/2) D. (2/3, 1/3)

试题分析

根据题意, 该地区冰箱品牌A与B每月占有率的变化描述为常数转移矩阵 P . 不管初始时刻这两种品牌的市场占有率 (以概率向量来描述) 如何, 最终将稳定到概率向量 Z , 而且有关系式 $ZP = Z$. 这表明, Z 的下一时刻仍然是 Z .

设 $Z = (Z_1, Z_2)$, 其中 $Z_1 \geq 0$, $Z_2 \geq 0$, $Z_1 + Z_2 = 1$, 从 $ZP = Z$ 可以列出方程:

$$\begin{aligned} 0.8Z_1 + 0.4Z_2 &= Z_1 \\ 0.2Z_1 + 0.6Z_2 &= Z_2 \end{aligned}$$

根据上述条件, 求解该方程, 得到 $Z_1 = 2/3$, $Z_2 = 1/3$.

因此, 冰箱品牌A与B在该地区最终将逐步稳定到市场占有率 (2/3, 1/3)。品牌A将占有2/3的市场, 品牌B将占有1/3的市场。

试题答案

- (70) D

第9章: 应用数学

试题3 (2010年11月试题69)

试题3 (2010年11月试题69)

对实际应用问题建立数学模型并求得结果后, 还需要根据建模的目的和要求, 利用相关知识, 结合研究对象的特点, 进行模型分析。模型分析工作一般不包括 (69) .

- (69) A. 模型的合理性分析 B. 模型的误差分析
C. 模型的先进性分析 D. 参数的灵敏性分析

试题分析

本题主要考查数学建模的基本过程, 在对实际应用问题建立数学模型并求得结果后, 还需要根据建模的目的和要求, 利用相关知识, 结合研究对象的特点, 进行模型分析。模型分析工作主要包括模型的合理性分析、模型的误差分析和参数的灵敏性分析等, 一般不包括模型的先进性分析。

试题答案

- (69) C

试题4 (2010年11月试题70)

试题4 (2010年11月试题70)

某软件公司开发某种软件产品时花费的固定成本为16万元，每套产品的可变成本为2元，设销售单价为10元，则需要销售 (70) 套才能达到盈亏平衡点。

- (70) A.13000 B.16000 C.18000 D.20000

试题分析

设共销售了n套，则总成本=固定成本+每套产品的可变成本×n=160000+2n.销售总收入=单价×n=10n.盈亏平衡时，总成本=销售总收入，所以， $160000+2n=10n$ ，从而， $n=20000$ (套)。当销售量低于2万套时，会有亏损；当销售量超过2万套时就会有盈利。

试题答案

- (70) D

试题5 (2011年11月试题69)

试题5 (2011年11月试题69)

在军事演习中，张司令希望将部队尽快从A地通过公路网 (如图9-1所示) 运送到F地：

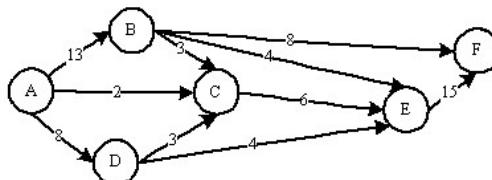


图9-1 公路网示意图

图中标出了各路段上的最大运量 (单位：千人/小时)。根据该图可以算出，从A地到F地的最大运量是 (69) 千人/小时。

- (69) A.20 B.21 C.22 D.23

试题分析

本题是架构考试中常见的一类计算题。该题解题关键是需要将图中节点的输入/输出流量调整平衡，因为只有输入/输出流量平衡才能表现出真实的运量。

例如，对于结点E，它的输出运力为15，而所有输入运力之和为14，则E的最大真实运力，只能达到14，所以将E的输出运力修改为14。对于D结点，其输出运力和为7，而输入运力为8，则需要平衡为7。结点B也需要调，但情况比较复杂，我们需要综合分析B的输出运力与C的输出运力，分析可知，当B到C的运力调整为1时，既能达到结点运力的平衡，又能使运力最大，所以应调整为1。当完成这些调整之后，可轻易得出结论，最大运力为22。

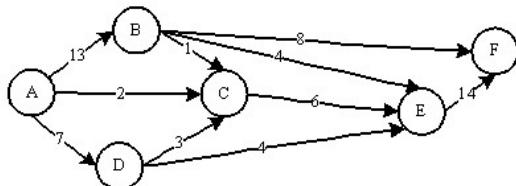


图9-2 公路网运力调整示意图

试题答案

(69) C

第 9 章 : 应用数学

试题6 (2011年11月试题70)

试题6 (2011年11月试题70)

某公司需要将4吨贵金属材料分配给下属的甲、乙、丙三个子公司 (单位 : 吨) 。据测算 , 各子公司得到这些材料后所能获得的利润 (单位 : 万元) 如表9-2所示。

表9-2 材料分配利润情况表

子 公 司 \ 材 料	1吨	2吨	3吨	4吨
甲	4	7	10	13
乙	5	9	11	13
丙	4	6	11	14

根据此表 , 只要材料分配适当 , 该公司最多可以获得利润 (70) 万元。

(70) A.14 B.16 C.17 D.18

试题分析

三个子公司分4吨金属材料。分法包括如下四种。

一、1+1+2方案 , 即 : 1家公司分2吨 , 另外2家公司分1吨。

该方案下的子方案包括如下。

(1) 甲2吨 + 乙1吨 + 丙1吨 : $7+5+4=16$

(2) 甲1吨 + 乙2吨 + 丙1吨 : $4+9+4=17$

(3) 甲1吨 + 乙1吨 + 丙2吨 : $4+5+6=15$

二、2+2方案 , 即 : 2家公司每家分2吨 , 另外1家公司不分。

该方案下的子方案包括如下。

(1) 甲2吨 + 乙2吨 : $7+9=16$

(2) 甲2吨 + 丙2吨 : $7+6=13$

(3) 乙2吨 + 丙2吨 : $9+6=15$

三、3+1方案 , 即 : 1家公司分3吨 , 1家公司分1吨 , 另外1家公司不分。

该方案有多种子方案组合 , 但此处是选择题 , 只需要做一些分析即可得到结论。3+1的方案 , 无论如何组合 , 都是将题目表格中的1吨列中与3吨列中各取1个数相加得来。而此处能得到的最佳方案也就是 $5+11=16$, 而之前我们已算出17的方案 , 所以3+1产生不了最佳方案。

四、4+0方案 , 即 : 1家公司分4吨 , 其余两家公司不分。

该方案也就对应着题目表格中的4吨这一列，最大值为14吨，也非最佳方案，所以最佳方案为甲1吨 + 乙2吨 + 丙1吨=17.

试题答案

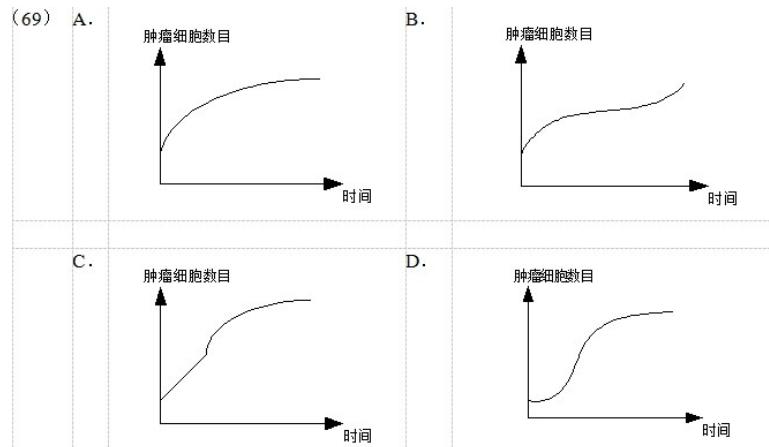
(70) C

第 9 章：应用数学

试题7 (2012年11月试题69)

试题7 (2012年11月试题69)

研究表明，肿瘤细胞的生长有以下规律：当肿瘤细胞数目超过 10^{11} 时才是临床可观察的；在肿瘤细胞生长初期，几乎每隔一定时间就会观测到肿瘤细胞数量翻一番；在肿瘤细胞生长后期，肿瘤细胞的数目趋向某个稳定值。为此，图 (69) 反映了肿瘤细胞的生长趋势。



试题分析

本题考查应用数学基础知识。

用函数曲线来表示事物随时间变化的规律十分常见。我们可以用函数 $f(t)$ 表示肿瘤细胞数量随时间变化的函数。那么，当肿瘤细胞数目超过 10^{11} 时才是临床可观察的，可以表示为 $f(0) = 10^{11}$ 。在肿瘤生长初期，几乎每隔一定时间就会观测到肿瘤细胞数量翻一番，可以表示为 $t < t_0$ 时， $f(t+c) = 2f(t)$ 。符合这种规律的函数是指数函数： $f(t) = at$ ，其曲线段呈凹形上升态。在肿瘤生长后期，肿瘤细胞的数目趋向某个稳定值，表示当 $t > T$ 时， $f(t)$ 逐渐逼近某个常数，即函数曲线从下往上逐渐靠近直线 $y = L$ 。

试题答案

(69) D

第 9 章：应用数学

试题8 (2012年11月试题70)

试题8 (2012年11月试题70)

九个项目A11、A12、A13、A21、A22、A23、A31、A32、A33的成本从1百万、2百万、…，到9百万各不相同，但并不顺序对应。已知A11与A21、A12与A22的成本都有一倍关系，A11与A12、A21与A31、A22与A23、A23与A33的成本都相差1百万。由此可以推断，项目A22的成本是(70)百万。

(70) A.2 B.4 C.6 D.8

试题分析

本题考查应用数学基础知识。

为便于直观分析，题中的叙述可以用下图来表示：

A11	○	A12	A13
●	●	●	
A21	A22	○	A23

九个项目 A_{ij} ($i=1,2,3; j=1,2,3$) 的成本值 (单位为百万 , 从1到9各不相同) 将分别填入 i 行 j 列对应的格中。格间的黑点表示相邻格有一倍关系 , 白点表示相邻格相差1。

已知 A_{22} 与 A_{12} 的值有一倍关系 , 那就只可能是1-2,2-4,3-6或4-8,因此 A_{22} 的值只可能是1,2,3,4,6,8。

如果 $A_{22}=1$,则 $A_{23}=A_{12}=2$,出现相同值 , 不符合题意。

如果 $A_{22}=2$,则 A_{12} 只能是4 ($A_{12}=1$ 将导致 $A_{11}=A_{22}=2$ 矛盾) , A_{23} 只能为3 ($A_{23}=1$ 将导致 $A_{33}=A_{22}=2$ 矛盾) , A_{33} 出现矛盾。

如果 $A_{22}=3$,则 $A_{12}=6$, $A_{11}=5$ 或7,不可能与 A_{21} 有一倍关系。

如果 $A_{22}=4$,则 $A_{12}=2$ 或8. $A_{12}=8$ 将导致 $A_{11}=7$ 或9,不可能与 A_{21} 有成倍关系。因此 $A_{12}=2$, A_{23} 只能是5 ($A_{23}=3$ 将导致 A_{33} 矛盾) , $A_{33}=6$,而 $A_{11}=1$ 或3都将导致 A_{21} 矛盾。

如果 $A_{22}=8$,则 $A_{12}=4$, A_{23} 只能是7 ($A_{23}=9$ 将导致 $A_{33}=8$ 矛盾) , A_{33} 只能是6, A_{11} 只能是3 ($A_{11}=5$ 将导致 A_{21} 矛盾) , $A_{21}=6$ 矛盾。

因此 , A_{22} 只可能为6。

实际上 , 当 $A_{22}=6$ 时 , $A_{12}=3$, A_{23} 只能为7 ($A_{23}=5$ 将最终导致矛盾) , $A_{33}=8$.此时 , A_{11} 、 A_{21} 、 A_{31} 可能分别是2、4、5,也可能是4、2、1.

参考答案

(70) C

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 标准化意识，标准化的发展，标准的生命周期。
- (2) 国际标准、美国标准、国家标准、行业标准、地方标准、企业标准。
- (3) 代码标准、文件格式标准、安全标准、软件开发规范和文档标准。
- (4) 标准化机构。
- (5) 知识产权。

从历年的考试情况来看，本章所涉及的法律法规包括：著作权法、计算机软件保护条例、商标法、地理标志权、专利法、不正当竞争法、标准化法。这些法律法规从条款上来看，非常的多，想要把他们全部弄明白，需要花费大量的时间，但从另外一个维度来看，考试当中，主要考查的问题是保护期限、知识产权人的确定、侵权判定以及标准的类型，所以本章将从该角度来进行分析讲解。具体考查知识点分布情况如表10-1所示。

表10-1历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题 66~68	著作权保护期限、著作权法、侵权判定
2010年11月试题 66~68	标准类型、商标法、侵权判定
2011年11月试题 66~68	著作权侵权问题、商业秘密权、职务作品
2012年11月试题 66~68	注册商标、著作权法、行业标准

第10章：知识产权与标准化

试题1 (2009年11月试题66)

10.2 试题精解

试题1 (2009年11月试题66)

我国的《著作权法》对一般文字作品的保护期是作者有生之年和去世后50年，德国的《版权法》对一般文字作品的保护期是作者有生之年和去世后70年。假如某德国作者已去世60年，以下说法中正确的是 (66)。

- (66) A. 我国M出版社拟在我国翻译出版该作品，需要征得德国作者继承人的许可方可在我国出版发行
- B. 我国M出版社拟在我国翻译出版该作品，不需要征得德国作者继承人的许可，就可在我国出版发行
- C. 我国M出版社未征得德国作者继承人的许可，将该翻译作品销售到德国，不构成侵权
- D. 我国M出版社未征得德国作者继承人的许可，将该翻译作品在我国销售，构成侵权

试题分析

本题考查知识产权方面的基础知识。按照《伯尔尼公约》的规定，一个成员国给予其他成员国作品的版权保护期，应按照该成员国版权法的规定。依据我国著作权法的规定，该德国作者的作品已经超过法定版权保护期，不再受到版权保护。因此，出版社不需要征得德国作者继承人的许可，即可在我国出版发行该德国作者的作品。如果将该翻译出版作品未征得德国作者继承人的许可销售到德国，已构成侵权。这是因为德国的《版权法》规定作品的版权保护期是作者有生之年和去世后

70年，作者去世60年，作品的保护期尚未超过，所以我国出版社若将该翻译出版作品未征得德国作者继承人的许可销售到德国，则构成侵权。

我国的《着作权法》对一般文字作品的保护期是作者有生之年和去世后50年，该作者已去世60年，超过了我国《着作权法》对一般文字作品的保护期，在我国也不再受着作权保护。所以我国出版社不需要征得德国作者继承人的许可，即可在我国出版发行该德国作者的作品。

试题答案

(66) B

第 10 章：知识产权与标准化

试题2 (2009年11月试题67)

试题2 (2009年11月试题67)

(67) 不属于我国着作权法所保护的内容。

- (67) A.为保护其软件着作权而采取的技术措施
- B.软件权利电子信息
- C.通过信息网络传播的软件
- D.采用反编译技术获得的软件

试题分析

本题考查知识产权方面的基础知识。我国着作权法采取列举方法，规定了侵权行为的表现形式。其中包括未经着作权人许可，复制、发行、表演、放映、广播、汇编、通过信息网络向公众传播其作品的行为；未经着作权人或者与着作权有关的权利人许可，故意避开或者破坏权利人为其作品、录音录像制品等采取的保护着作权或者与着作权有关的权利的技术措施的行为；未经着作权人或者与着作权有关的权利人许可，故意删除或者改变作品、录音录像制品等的权利管理电子信息的行为。这三种表现形式分别涵盖了试题中A、B、C选项。虽然利用反向编译技术、净室技术和反向工程技术等获得他人软件技术构思、技术方案并直接用于其软件产品中的行为是一种“复制”软件技术构思、技术方案的行为，但是对于这些行为在法律上不会受到制止。目前，我国对软件实施反编译是否合法还没有相应的法律规定。

由于着作权不保护思想，软件开发设计人员对体现在软件中的创造性的构思和技术方案不能得到保护。任何人都可以利用反向编译技术、净室技术和反向工程技术等获得他人软件所使用的思路、原理、结构、算法、处理过程和运行方法等设计要素，直接用于自己的软件产品中，这在着作权法上并不构成侵权。

试题答案

(67) D

试题3 (2009年11月试题68)

试题3 (2009年11月试题68)

王某原是X公司的项目经理，在X公司任职期间主持开发了某软件，但未与X公司签定劳动合同及相应的保密协议。X公司对该软件进行了软件着作权登记并获准。王某随后离职并将其在X公司任职期间掌握的该软件技术信息、客户需求及部分源程序等秘密信息提供给另一家软件公司。王某的行为（68）。

- (68) A.既侵犯了科技公司的商业秘密权，又侵犯了科技公司的软件着作权
- B.既未侵犯科技公司的商业秘密权，又未侵犯科技公司的软件着作权
- C.侵犯了科技公司的商业秘密权
- D.侵犯了科技公司的软件着作权

试题分析

本题考查知识产权方面的基础知识，涉及软件着作权和商业秘密权的相关概念。王某作为公司的职员，在任职期间主持开发的软件为职务软件，公司对该软件享有软件着作权。王某将该软件源程序擅自提供给其他公司的行为已构成对公司软件着作权的侵犯。王某的行为将使得另一家软件公司很快就会开发出类似的产品，在市场上与科技公司竞争，这样无疑会损害科技公司的利益。软件商业秘密包括软件技术秘密，如源程序、设计方法、技术方案、功能规划、开发情况和测试结果等；软件经营秘密，如经营方法、产销策略、客户情报（客户名单、客户需求）和软件市场分析等。商业秘密受到法律保护的依据是必须具备构成商业秘密的三个条件，即不为公众所知悉、具有实用性、采取了保密措施，缺少三个条件之一都会造成商业秘密丧失法律保护。公司未与王某签定劳动合同及相应的保密协议，可以认为科技公司主观上没有保守商业秘密的意愿，客观上没有采取相应的保密措施，那么公司的软件技术秘密和软件经营秘密就不具有保密性。所以，不认为王某侵犯了公司的商业秘密权。

试题答案

- (68) D

试题4 (2010年11月试题66)

试题4 (2010年11月试题66)

《GB 8567-88计算机软件产品开发文件编制指南》是（66）标准，违反该标准而造成不良后果时，将依法根据情节轻重受到行政处罚或追究刑事责任。

- (66) A.强制性国家 B.推荐性国家
- C.强制性软件行业 D.推荐性软件行业

试题分析

我国国家标准的代号由大写汉字拼音字母构成，强制性国家标准代号为GB,推荐性国家标准的代号为GB/T.

强制性标准是国家技术法规，具有法律约束性。其范围限制在国家安全、防止欺诈行为、保护人身健康与安全等方面。根据《标准化法》的规定，企业和有关部门对涉及其经营、生产、服务、管理有关的强制性标准都必须严格执行，任何单位和个人不得擅自更改或降低标准。对违反强制性标准而造成不良后果以至重大事故者，由法律、行政法规规定的行政主管部门依法根据情节轻重给予行政处罚，直至由司法机关追究刑事责任。

推荐性标准是自愿采用的标准。这类标准是指导性标准，不具有强制性，一般是为了通用或反复使用的目的，为产品或相关生产方法提供规则、指南或特性的文件。任何单位均有权决定是否采用，违犯这类标准，不构成经济或法律方面的责任。由于推荐性标准是协调一致的文件，不受政府和社会团体的利益干预，能更科学地规定特性或指导生产，我国《标准化法》鼓励企业积极采用推荐性标准。应当指出的是，推荐性标准一经接受并采用，或由各方商定后同意纳入经济合同中，就成为各方必须共同遵守的技术依据，具有法律上的约束性。

由行业机构、学术团体或国防机构制定，并适用于某个业务领域的标准。行业标准代号由国务院各有关行政主管部门提出其所管理的行业标准范围的申请报告，国务院标准化行政主管部门审查确定并正式公布该行业标准代号。已正式公布的行业代号为：QJ（航天）、SJ（电子）、JB（机械）、JR（金融）等等，暂无软件行业。行业标准代号由汉字拼音大写字母组成，再加上斜线T组成推荐性行业标准（如SJ/T）。

试题答案

(66) A

第 10 章：知识产权与标准化

试题5（2010年11月试题67）

试题5（2010年11月试题67）

某软件企业开发了一套能够同硬件结合以提高设备性能的软件产品，向国家专利局申请方法发明专利，获得了专利权，并为该软件产品冠以“昆仑”商品专用标识，但未进行商标注册上市销售。此情况下，该软件产品不可能得到我国（67）的保护。

(67) A.着作权法 B.专利法 C.商标法 D.刑法

试题分析

该企业在软件开发完成后就取得了着作权，该企业向国家专利局申请方法发明专利并获得了专利权，此时该软件就可以同时受到着作权法、专利法的保护。

在《刑法》中明确规定了严重侵犯知识产权的行为应当承担刑事责任。在《计算机软件保护条例》第24条中，规定对侵权行为触犯刑律的，依照刑法关于侵犯着作权罪、销售侵权复制品罪的规定，依法追究刑事责任。例如《刑法》中第213条规定，未经注册商标所有人许可，在同一种商品上使用与其注册商标相同的商标，情节严重的，处三年以下有期徒刑或者拘役，并处或者单处罚金；情节特别严重的，处三年以上七年以下有期徒刑，并处罚金；216条规定对假冒他人专利，情节严重

的，处三年以下有期徒刑或者拘役，并处或者单处罚金；217条规定对以营利为目的，未经着作权人许可，复制发行其计算机软件的，违法所得数额较大或者有其他严重情节的，处三年以下有期徒刑或者拘役，并处或者单处罚金；违法所得数额巨大或者有其他特别严重情节的，处三年以上七年以下有期徒刑，并处罚金；218条规定对以营利为目的，销售明知是本法第二百一十七条规定的侵权复制品，违法所得数额巨大的，处三年以下有期徒刑或者拘役，并处或者单处罚金。所以，该软件产品能够得到刑法的保护。

商标权是商标所有人依法对其商标所享有的专有使用权。在我国，商标权的取得实行的是注册原则，即商标所有人只有依法将自己的商标注册后，商标注册人才能取得商标权，其商标才能得到法律的保护。该企业虽然对其软件产品已经冠以商品专用标识，但未进行商标注册，没有取得商标专用权，此时该软件产品就不能得到商标法的保护。

商标法虽然不能提供针对计算机软件的实质内容与表达的直接保护，却可以为软件提供商业化的保护。计算机软件的权利人可以通过商标法来实现对其所属软件的保护。首先，软件的权利人可以针对软件产品申请注册商标，并通过标注在其产品包装上的商标来表明其身份以及商誉；其次，软件的权利人还可以通过技术措施在软件中设置其特有的商业标记，例如，通过技术手段将与注册商标相同的文字商标、图形商标或者其两者的结合出现在软件的界面上。这些措施不仅可以防止他人对商品化的计算机软件实施侵权，也可以在遭受侵权后，利用商标法有效地打击侵权行为。尤其是当软件被盗版的时候，这些盗版软件经常是不标明商标、产地等，甚至是假冒商标，软件权利人可以依据商标法请求工商行政管理部门进行查处，在实践中，行政手段通常要比诉讼程序更简便、快捷，达到保护计算机软件的目的。当然，商标法还对侵犯软件商标权的行为应承担的民事以及刑事责任进行了规定。

试题答案

(67) C

第 10 章：知识产权与标准化

试题6 (2010年11月试题68)

试题6 (2010年11月试题68)

张某是M国际运输有限公司计算机系统管理员。任职期间，根据公司的业务要求开发了"空运出口业务系统",并由公司使用。随后，张某向国家版权局申请了计算机软件着作权登记，并取得了《计算机软件着作权登记证书》，证书明确软件名称是"空运出口业务系统V1.0",着作权人为张某。以下说法中，正确的是 (68) .

- (68) A.空运出口业务系统V1.0的着作权属于张某
- B.空运出口业务系统V1.0的着作权属于M公司
- C.空运出口业务系统V1.0的着作权属于张某和M公司
- D.张某获取的软件着作权登记证是不可以撤销的

试题分析

张某开发的软件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“空运出口业务系统V1.0”,即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《着作权法》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职务作品:一般职务作品的着作权由作者享有,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;特殊职务作品,除署名权以外,着作权的其他权利由单位享有。所谓特殊职务作品是指《着作权法》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着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。张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,其工作成果应由公司享有。因此,该软件的着作权应属于国际运输有限公司,但根据法律规定,张某享有署名权。

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着作权以及订立许可合同、转让合同的重要的书面证据,并不是软件着作权产生的依据。该软件是张某针对本职工作中明确指定的开发目标所开发的,该软件的着作权应属于公司。明确真正的着作权人之后,软件着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。)。

为促进我国软件产业发展,增强我国软件产业的创新能力和竞争能力,1992年4月6日机械电子部发布了《计算机软件着作权登记办法》,鼓励软件登记并对登记的软件予以重点保护,而不是强制软件登记。软件登记可以分为软件着作权登记、软件着作权专有许可合同和转让合同的登记。软件着作权登记的申请人应当是该软件的着作权人,而软件着作权合同登记的申请人,应当是软件着作权专有许可合同和转让合同的当事人。如果未经软件着作权人许可登记其软件,或是将他人软件作为自己的软件登记的,或未经合作者许可、将与他人合作开发的软件作为自己单独完成的软件登记,这些行为都属于侵权行为,侵权人要承担法律责任。

试题答案

(68) B

第 10 章 : 知识产权与标准化

试题7 (2011年11月试题66)

试题7 (2011年11月试题66)

甲公司的某个注册商标是乙画家创作的绘画作品,甲申请该商标注册时未经乙的许可,乙认为其着作权受到侵害。在乙可采取的以下做法中,错误的是(66)。

- (66) A.向甲公司所在地人民法院提起着作权侵权诉讼
B.请求商标评审委员会裁定撤销甲的注册商标
C.首先提起诉讼，如对法院判决不服再请求商标评审委员会进行裁定
D.与甲交涉，采取许可方式让甲继续使用该注册商标

试题分析

本题看似是考着作权与商标权相关内容。但实际上是在考查一般争议处理的流程。对于任何争议基本上都是采取先找主管行政管理部门进行仲裁，仲裁不成功再进行诉讼，而C选项的说法，刚好弄反了。

试题答案

- (66) C

第 10 章：知识产权与标准化

试题8 (2011年11月试题67)

试题8 (2011年11月试题67)

利用 (67) 可以对软件的技术信息、经营信息提供保护。

- (67) A.着作权 B.专利权 C.商业秘密权 D.商标权

试题分析

本题考查商业秘密相关概念。商业秘密是《反不正当竞争法》中提出的，商业秘密 (Business Secret)，按照我国《反不正当竞争法》的规定，是指不为公众所知悉、能为权利人带来经济利益，具有实用性并经权利人采取保密措施的技术信息和经营信息。

试题答案

- (67) C

第 10 章：知识产权与标准化

试题9 (2011年11月试题68)

试题9 (2011年11月试题68)

M公司的程序员在不影响本职工作的情况下，在L公司兼职并根据公司项目开发出一项与M公司业务无关的应用软件。该应用软件的着作权应由 (68) 享有。

- (68) A.M公司 B.L公司
C.L公司与M公司共同 D.L公司与程序员共同

试题分析

依据题意，该应用软件是程序员在L公司兼职，并按L公司的工作要求开发出的软件，应属于L公司的职务作品，所以著作权归L公司所有。

试题答案

(68) B

第 10 章：知识产权与标准化

试题10 (2012年11月试题66)

试题10 (2012年11月试题66)

中国M公司与美国L公司分别在各自生产的平板电脑产品上使用iPad商标，且分别享有各自国家批准的商标专用权。中国Y手电筒经销商，在其经销的手电筒高端产品上也使用iPad商标，并取得了注册商标。以下说法正确的是 (66) .

- (66) A. L公司未经M公司许可在中国市场销售其产品不属于侵权行为
- B. L公司在中国市场销售其产品需要取得M公司和Y经销商的许可
- C. L公司在中国市场销售其产品需要向M公司支付注册商标许可使用费
- D. Y经销商在其经销的手电筒高端产品上使用iPad商标属于侵权行为

试题分析

本题考查知识产权知识，涉及商标权的相关概念。知识产权具有地域性的特征，按照一国法律获得承认和保护的知识产权，只能在该国发生法律效力，即知识产权受地域限制，只有在一定地域内知识产权才具有独占性（专用性）。或者说，各国依照其本国法律授予的知识产权，只能在其本国领域内受其国家的法律保护，而其他国家对这种权利没有保护的义务，任何人均可在自己的国家内自由使用外国人的知识产品，既无须取得权利人的许可，也不必向权利人支付报酬。

通过缔结有关知识产权的国际公约的形式，某一国家的国民（自然人或法人）的知识产权在其他国家也能取得权益。参加知识产权国际公约的国家，会相互给予成员国国民的知识产权保护。虽然众多知识产权国际条约等的订立，使地域性有时会变得模糊，但地域性的特征不但是知识产权最“古老”的特征，也是最基本的特征之一。目前知识产权的地域性仍然存在，如是否授予权利、如何保护权利，仍须由各成员国按照其国内法律来决定。依据我国商标法52条规定，未注册商标不得与他人在同一类或类似商品上已经注册的商标相同或近似。若未经商标注册人的许可，在同一种商品或者类似商品上使用与他人注册商标相同或者近似的商标的，属于侵犯专用权的行为，应当承担相应的法律责任。

知识产权的利用（行使）有多种方式，许可使用是其之一，它是指知识产权人将自己的权利以一定的方式，在一定的地域和期限内许可他人利用，并由此获得报酬（即向被许可人收取一定数额的使用费）的法律行为。对于注册商标许可而言是指注册商标所有人通过订立许可使用合同，许可他人使用其注册商标的法律行为。

依据我国商标法规定，不同类别商品（产品）是可以使用相同或类似商标的，如在水泥产品和化肥产品都可以使用“秦岭”商标，因为水泥产品和化肥产品是不同类别的产品。但对于驰名商标来说，不能在任何商品（产品）使用与驰名商标相同或类似的标识。

试题答案

(66) C

第 10 章：知识产权与标准化

试题11 (2012年11月试题67)

试题11 (2012年11月试题67)

M画家将自己创作的一幅美术作品原件赠予了L公司。L公司未经该画家的许可，擅自将这幅美术作品作为商标注册，且取得商标权，并大量复制用于该公司的产品上。L公司的行为侵犯了M画家的 (67) .

(67) A.著作权 B.发表权 C.商标权 D.展览权

试题分析

本题考查知识产权基础知识，涉及侵权的相关概念。绘画、书法、雕塑等美术作品的原件可以买卖、赠与。但是，获得一件美术作品并不意味着获得该作品的着作权。我国着作权法第18条规定：“美术等作品原件所有权的转移。不视为作品着作权的转移，但美术作品原件的展览权由原件所有人享有”。这就是说，艺术类作品物转移的事实并不引起美术作品着作权的转移，受让人只是取得物的所有权和作品原件的展览权，作品的着作权仍然由作者等着作权人享有。除了艺术类作品之外，对任何原件所有权可能转移的作品，如书籍、软件等都要注意区分作品物质载体的财产权和作品的着作权这两种不同的权利。

发表权是指作者决定作品是否公之于众和以何种方式公之于众的权利。发表权作为着作权人的一种权利，其包含两种含义：一是权利人有权决定是否发表，或许可他人发表；二是权利人有权以某种方式发表，如出版、发行、展览、销售等，以及确定在什么时间、地点发表。在一些情况下，作者虽未将作品公之于众，但可推定作者同意发表其作品。例如，作者许可他人使用其未发表的作品意味着作者同意发表其作品，认为作者已经行使发表权。又如，作者将其未发表的作品原件所有权转让给他人，意味着发表权与着作财产权的一起行使，即作者的发表权也已行使完毕，已随着财产权转移。再如，展览尚未发表的作品即为作品的发表，展览行为必然附带着发表，所以推定已经行使了发表权。发表权是一次性权利，即发表权行使一次后，不再享有发表权。例如，第一次出版、第一次表演、上网公布等都属于行使发表权。以后再次使用作品与发表权无关，而是行使作品的使用权。发表权须借助一定的作品使用方式行使，即作品的公之于众要以某种确定的方式实现。也就是说，发表权难以孤立地行使，要以某种确定的方式实现，如书籍的出版、剧本的上演、绘画的展出等，既是作品的发表，同时也是作品的使用。

在我国商标权的取得实行的是注册原则，即商标所有人只有依法将自己的商标注册后，商标注册人才能取得商标权，其商标才能得到法律的保护。M画家并未将其美术作品实施商标注册，不享有其美术作品的商标权，因此L公司的行为未侵犯M画家的商标权，而是侵犯了M画家的在先权利。在先权利包括着作权、外观设计专利权、商号权、地理标志权、姓名权等。

展览权是将作品原件或复印件公开陈列的权利。公开陈列的作品既可以是已经发表的作品，也可以是尚未发表的作品。画展、书法展、摄影展等都是公开陈列。

试题答案

(67) A

第 10 章：知识产权与标准化

试题12 (2012年11月试题68)

试题12 (2012年11月试题68)

以下我国的标准代号中， (68) 表示行业标准。

- (68) A.GB B.SJ C.DB11 D.Q

试题分析

本题考查标准与标准化基本知识。我国标准分为国家标准、行业标准、地方标准和企业标准四类。

国家标准是由国务院标准化行政主管部门制定的标准。国家标准的编号由标准代号、标准发布顺序号和标准发布年代号构成。国家标准的代号由大写汉字拼音字母构成，强制性国家标准代号为 GB，推荐性国家标准的代号为 GB/T。例如，“GB 8567 - 1988 软件产品开发文件编制指南”为强制性国家标准。

行业标准代号由国务院各有关行政主管部门提出其所管理的行业标准范围的申请报告，国务院标准化行政主管部门审查确定并正式公布该行业标准代号。已正式公布的行业代号：QJ（航天）、SJ（电子）、JB（机械）、JR（金融）、HB（航空）等等。行业标准代号由汉字拼音大写字母组成，再加上斜线T组成推荐性行业标准。行业标准的编号由行业标准代号、标准发布顺序及标准发布年代号组成。例如，“HB 6698-1993 软件工具评价与选择的分类特性体系”为强制性行业标准编号。GJB为中华人民共和国国家军用标准代号。

地方标准是由省、自治区、直辖市标准化行政主管部门制定并报国务院标准化行政主管部门和国务院有关行业行政主管部门备案的标准。地方标准代号由大写汉字拼音DB加上省、自治区、直辖市行政区划代码的前两位数子（北京市11、天津市12），再加上斜线T组成推荐性地方标准（DBXX/T），不加斜线T为强制性地方标准（DBXX）。地方标准的编号由地方标准代号、发布顺序号、发布年三部分组成，即DBXX XXX - XXXX。

企业标准是由企业自行组织制定、作为组织生产依据的相应标准。企业标准一经制定颁布，即对整个企业具有约束性，是企业法规性文件，没有强制性企业标准和推荐企业标准之分。企业代号可用大写拼音字母或阿拉数字或两者兼用所组成（Q/XXX），按中央所属企业和地方企业分别由国务院有关行政主管部门或省、自治区、直辖市人民政府标准化行政主管部门会同同级有关行政主管部门加以规定。企业标准的编号由企业标准代号，发布顺序号和发布年代号组成，即Q/XXX XXXX - XXXX。

试题答案

(68) B

考点分析

第11章系统配置与性能评价

11.1考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 多层结构、分布式系统。
- (2) 系统配置方法（双份、双重、热备份、容错、集群）。
- (3) 性能计算（响应时间、吞吐量、TAT）。
- (4) 性能设计（系统调整、Amdahl解决方案、响应特性、负载均衡）。
- (5) 性能指标（SPEC-Int、SPEC-Fp、TPC、Gibsonmix、响应时间）。
- (6) 性能评估。
- (7) 系统可靠性设计（容错技术、避错技术）。
- (8) 系统可靠性指标与评估。

从历年的考试情况来看，本章的考点主要集中于性能评估和系统可靠性设计，该方面的内容考查不仅限于综合知识考试，也有可能以案例分析及论文方式考查。具体考查知识点分布情况如表 11-1 所示。

表11-1历年考查知识点分布情况表

试 题	考查知识点
2009 年 11 月试题 16~17	基准程序测试、计算机性能优化
2010 年 11 月试题 16~17	多处理器性能计算、系统性能评价
2011 年 11 月试题 16~17	负载均衡、数据备份
2012 年 11 月试题 17	MIPS

试题1 (2009年11月试题16)

11.2 试题精解

试题1 (2009年11月试题16)

以下关于基准测试的叙述中，正确的是（16）。

- (16) A.运行某些诊断程序，加大负载，检查哪个设备会发生故障
- B.验证程序模块之间的接口是否正常起作用
- C.运行一个标准程序对多种计算机系统进行检查，以比较和评价它们的性能
- D.根据程序的内部结构和内部逻辑，评价程序是否正确

试题分析

把应用程序中用得最多、最频繁的那部分核心程序作为评估计算机系统性能的标准程序，称为基准测试程序（benchmark）。基准程序法是目前一致承认的测试系统性能的较好方法。

1. Dhrystone基准程序

Dhrystone是一个综合性的整数基准测试程序，它是为了测试编译器和CPU处理整数指令和控制功能的有效性，人为地选择一些典型指令综合起来形成的测试程序。Dhrystone基准程序用100条C语言语句（包括各种赋值语句、各种数据类型和数据区、各种控制语句、过程调用和参数传送、整数运算和逻辑操作）编写而成，这种基准程序当今很少使用。

2. Linpack基准程序

Linpack是国际上最流行的用于测试高性能计算机系统浮点性能的测试。Linpack基准程序是一个用Fortran语言写成的子程序软件包，称为基本线性代数子程序包，此程序完成的主要操作是浮点加法和浮点乘法操作。测量计算机系统的Linpack性能时，让机器运行Linpack程序，测量运行时间，将结果用MFLOPS表示。

Linpack通过对高性能计算机采用高斯消元法求解一元n次稠密线性代数方程组的测试，评价高性能计算机的浮点性能。Linpack测试包括三类，分别是Linpack100、Linpack1000和HPL（High Performance Linpack,高性能Linpack）。Linpack100求解规模为100阶的稠密线性代数方程组，它只允许采用编译优化选项进行优化，不得更改代码，甚至代码中的注释也不得修改；Linpack1000要求求解1000阶的线性代数方程组，达到指定的精度要求，可以在不改变计算量的前提下做算法和代码上的优化；HPL也称为高度并行计算基准测试，它对线性代数方程组的阶数n没有限制，即求解问题的规模可以改变，除基本算法（计算量）不可改变外，可以采用其他任何优化方法。前两种测试运行规模较小，已不是很适合现代计算机的发展。

3. Whetstone基准程序

Whetstone是用Fortran语言编写的综合性测试程序，主要由执行浮点运算、功能调用、数组地址、条件转移和超越函数的程序组成。Whetstone的测试结果用Kwips表示，1 Kwips表示机器每秒钟能执行1000条Whetstone指令。这种基准程序当今已很少使用。

4. SPEC

SPEC（System Performance Evaluation Cooperative,系统性能评估机构）基准程序对计算机系统性能的测试有两种方法，一种是测试计算机完成单项任务有多快，称为速度测试；另一种是测试计算机在一定时间内能完成多少项任务，称为吞吐率测试。SPEC的两种测试方法又分为基本的和非基本的两类，基本的是指在编译程序的过程中严格限制所用的优化选项；非基本的是可以使用不同的编译器和编译选项以得到最好的性能，这就使得测试结果的可比性降低。

SPEC基准程序测试结果一般以SPECmark（SPEC分数）、SPECint（SPEC整数）和SPECfp（SPEC浮点数）等形式来表示，测定指标越高，则代表性能越好。SPEC还有针对多CPU系统的两组测试程序，称为SPECrate，用于衡量多处理器系统在整数处理（SPECint_rate）和浮点处理（SPECfp_rate）方面的总体吞吐能力。

SPEC原来主要是测试CPU性能的，现在强调开发能反映真实应用（例如，实际负载等）的基准测试程序，并已推广至多层结构计算、商业应用、I/O子系统等。例如，SPECjbb是一套Java基准测试程序，用于测试Java服务器性能；SPECweb用于评测Web服务器能够支持的最大连接数的基准测试。

要注意的是，SPEC的指标是随着计算机硬件水平的提高和应用要求的改变而不断更新的，每隔几年都会公布新的基准测试程序。

5. TPC

TPC（Transaction Processing Council,事务处理委员会）基准程序用以评测计算机在事务处理、数据库处理、企业管理与决策支持系统等方面的性能。该基准程序的评测结果用每秒完成的事

务处理数TPC来表示。TPC-A基准程序规范用于评价在OLTP环境下的数据库和硬件的性能，不同系统之间用性能/价格比进行比较；TPC-B测试的是不包括网络的纯事务处理量，用于模拟企业计算环境；TPC-C测试的是联机订货系统；TPC-D、TPC-H和TPC-R测试的都是决策支持系统，其中TPC-R允许有附加的优化选项；TPC-E测试的是大型企业信息服务系统。TPC-W是基于Web应用的基准程序，用来测试一些通过Internet进行市场服务和销售的商业行为，所以TPC-W可以看作是一个服务器的测试标准。

试题答案

(16) C

第 11 章：系统配置与性能评价

试题2 (2009年11月试题17)

试题2 (2009年11月试题17)

以下关于计算机性能改进的叙述中，正确的是（17）。

- (17) A.如果某计算机系统的CPU利用率已经达到100%，则该系统不可能再进行性能改进
- B.使用虚存的计算机系统如果主存太小，则页面交换的频率将增加，CPU的使用效率就会降低，因此应当增加更多的内存
- C.如果磁盘存取速度低，引起排队，此时应安装更快的CPU，以提高性能
- D.多处理机的性能正比于CPU的数目，增加CPU是改进性能的主要途径

试题分析

计算机运行一段时间后，经常由于应用业务的扩展，发现计算机的性能需要改进。

计算机性能改进应针对出现的问题，找出问题的瓶颈，再寻求适当的解决方法。

计算机的性能包括的面很广，不单是CPU的利用率。即使CPU的利用率已经接近100%，这只能说明目前计算机正在运行大型计算任务。其他方面的任务可能被外设阻塞着，而改进外设成为当前必须解决的瓶颈问题。

如果磁盘存取速度低，则应增加新的磁盘或更换使用更先进的磁盘。安装更快的CPU不能解决磁盘存取速度问题。

多处理机的性能并不能正比于CPU的数目，因为各个CPU之间需要协调，需要花费一定的开销。

使用虚存的计算机系统如果主存太小，则主存与磁盘之间交换页面的频率将增加，业务处理效率就会降低，此时应当增加更多的内存。这就是说，除CPU主频外，内存大小对计算机实际运行的处理速度也密切相关。

试题答案

(17) B

试题3 (2010年11月试题16)

试题3 (2010年11月试题16)

假设单个CPU的性能为1,则由n个这种CPU组成的多处理机系统的性能P为：

$$P = \frac{n}{1 + (n-1)a}$$

其中, a是一个表示开销的常数。例如, a=0.1,n=4时, P约为3.也就是说, 由4个这种CPU组成的多机系统的性能约为3.该公式表明, 多机系统的性能有一个上限, 不管n如何增加, P都不会超过某个值。当a=0.1时, 这个上限是 (16) .

- (16) A.5 B.10 C.15 D.20

试题分析

本题实际上就是求a=0.1时, n趋向于无穷大, P的值。

下面不涉及复杂的数学理论, 简单一点来分析:

首先可以将 $P = \frac{n}{1 + (n-1)a}$ 转化为: $P = n / (1 + an - a) = n / (0.9 + 0.1n)$ 。

由于当n趋向于无穷大时, 常数0.9可以忽略不计, 即: $P = n / 0.1n = 10$.所以上限应为: 10.

试题答案

- (16) B

试题4 (2010年11月试题17)

试题4 (2010年11月试题17)

以下关于系统性能的叙述中, 不正确的是 (17) .

- (17) A.常见的Web服务器性能评估方法有基准测试、压力测试和可靠性测试
B.评价Web服务器的主要性能指标有最大并发连接数、响应延迟和吞吐量
C.对运行系统进行性能评估的主要目的是以更好的性能/价格比更新系统
D.当系统性能降到基本水平时, 需要查找影响性能的瓶颈并消除该瓶颈

试题分析

题目中的描述"对运行系统进行性能评估的主要目的是以更好的性能/价格比更新系统"是错误的, 对系统进行性能评估通常是为了找出系统瓶颈, 改善系统性能, 而非以更好的性能/价格比更新系统。

试题答案

- (17) C

试题5 (2011年11月试题16)

试题5 (2011年11月试题16)

随着业务的增长，信息系统的访问量和数据流量快速增加，采用负载均衡 (Load Balance) 方法可避免由此导致的系统性能下降甚至崩溃。以下关于负载均衡的叙述中，错误的是 (16) 。

- (16) A.负载均衡通常由服务器端安装的附加软件来实现
- B.负载均衡并不会增加系统的吞吐量
- C.负载均衡可在不同地理位置、不同网络结构的服务器群之间进行
- D.负载均衡可使用户只通过一个IP地址或域名就能访问相应的服务器

试题分析

负载均衡是集群系统中的一项重要技术，可以提高集群系统的整体处理能力，也提高了系统的可靠性，最终目的是加快集群系统的响应速度，提高客户端访问的成功概率。集群的最大特征是多个节点的并行和共同工作，如何让所有节点承受的负荷平均，不出现局部过大负载或过轻负载的情况，是负载均衡的重要目的。如果出现局部过大负载，必然导致硬件压力比较大，老化和损坏的可能性比较大；如果出现局部过轻负载，设备资源被搁置浪费，不符合成本最低原则。

负载均衡有两方面的含义。首先，大量的并发访问或数据流量分担到多个节点上分别处理，减少用户等待响应的时间；其次，单个重负载的运算分担到多个节点上做并行处理，每个节点处理结束后，将结果汇总，返回给用户，系统处理能力得到大幅度提高。

1. 调度算法

目前，人们提出了许多负载均衡的调度算法，用于实现计算机集群系统的负载均衡。这些算法大致可分为静态调度算法和动态调度算法两类。静态调度算法是指调度算法在调度时无需考虑节点当前的负载状态，而是依据不同原则在调度前选择一种均匀调度规则来完成服务请求的调度。该类算法数量众多，典型的有轮转算法、加权轮转算法、最小连接数算法、加权最小连接数算法、源地址哈希散列算法、目标地址哈希散列算法和随机算法等。静态调度算法由于其调度策略事先确定，无法根据当前节点的负载状况进行自适应调整；动态调度算法是指在进行服务请求的调度前，需考虑节点当前的一些动态指标，根据这些动态指标来决定服务请求的调度，典型的有加权百分比算法等。

(1) 轮转算法。轮转算法是一种经典的分配算法，该算法每次轮流将服务请求 (任务) 调度给不同的节点。该算法的优点是简单，它无需记录当前所有请求的状态，所以是一种无状态调度。轮转算法假定所有节点的处理性能均相同，而且不管节点的当前负载、请求个数和响应速度的差异，不适用于节点处理性能不一样的情况。另外，当各请求响应时间变化比较大时，轮转算法极易导致节点之间的负载不平衡。轮转算法的粒度是基于每个请求连接的，同一用户的不同请求会被调度到不同的节点上。

(2) 加权轮转算法。加权轮转算法是轮转算法的一个改进，其思想是首先按照各节点的性能分别指定不同权值，然后按权值来分配给节点相应的请求数量。加权轮转算法是按权值的高低和轮转方式把请求分配到各节点的，权值高的节点比权值低的节点处理更多的请求，相同权值的节点处理相同数目的请求。

(3) 最小连接数算法。最小连接数算法是一种根据各节点的负载状况来分配请求的算法，其基本思想是调度程序把每个新请求分配给当前活动请求数量最少的节点。最小连接数算法的优点是，当所有节点具有相同的处理能力时，算法把负载变化大的请求调度到多个节点上，所有处理时间比较长的请求不可能被调度到同一个节点上。但是，当各节点处理能力不同时，该算法并不理想。

(4) 加权最小连接数算法。加权最小连接数算法与加权轮转算法类似，只不过它是基于最小连接数来加权计算。各个节点用相应的权值表示其处理性能，在调度新请求时，尽可能使节点处理的请求数量与其权值成比例。

(5) 基于局部性的最小连接数算法 (Locality-Based Least Connections, LBLC)。LBLC算法针对请求报文的目标IP地址进行负载均衡调度，算法的设计目标是在节点负载基本平衡的情况下，将相同目标IP地址的请求调度到同一个节点上，以提高各个节点的访问局部性和主存命中率，从而提高整个集群系统的处理能力。LBLC算法先根据请求的目标IP地址，找出该目标IP地址最近使用的节点，若该节点是可用的且没有超载，就将请求调度到该节点；若该节点不存在或该节点超载，且有节点处于其一半的工作负载，则用最小连接的原则选出一台可用的节点，将请求发送到该节点。

(6) 带复制的基于局部性的最小连接数 (Locality-Based Least Connections with Replication, LBLCR) 算法。LBLCR算法也是针对目标IP地址进行负载均衡调度，它与LBLC算法的不同之处在于，它要维护从一个目标IP地址到一组节点的映射，而LBLC维护从一个目标IP地址到一个节点的映射。LBLCR算法先根据请求的目标IP地址，找出该目标IP地址对应的节点组，按最小连接的原则从该节点组中选出一个节点，若该节点没有超载，将请求发送到节点；若该节点超载，则按最小连接的原则从整个集群中选出一个节点，将该节点加入到节点组中，将请求发送到该节点。同时，当该节点组有一段时间没有被修改，将最忙的节点从节点组中删除，以降低复制的程度。

(7) 目标地址哈希散列算法。目标地址哈希散列算法是一种静态映射算法，通过一个散列函数将目的IP地址映射到一个节点。该算法先以请求的目的IP地址作为散列键，从静态分配的散列表中找出对应的节点，若该节点是可用的且未超载，就将请求发送到该节点；否则，返回空值。

(8) 源地址哈希散列算法。源地址哈希散列算法与目标地址哈希散列算法相似，只不过它以请求的源IP地址作为散列键。

(9) 随机分配算法。对于每个服务请求，通过随机选择的方式选择一个节点为其提供服务。随机分配算法也是一种实现简单、无状态的调度算法。

(10) 加权百分比算法。加权百分比算法考虑了节点的利用率、内存利用率、硬盘速率、进程个数、分配的任务数等，使用利用率来表现剩余处理能力，并通过对每个因素选择一个影响系数来表现对节点整体工作性能产生的作用。该算法实现难度在于选择各个因素的系数。系数选择得好，集群系统可以达到较好的负载均衡效果；否则，可能还不如其他算法。

2. 技术实现

在实际应用中，比较常用的负载均衡实现技术主要有以下几种：

(1) 基于特定软件的负载均衡。很多网络协议都支持重定向功能，例如，在HTTP协议中支持Location指令，接收到这个指令的浏览器将自动重定向到Location指明的另一个URL上。由于发送Location指令比起执行服务请求，对节点的负载要小得多，因此，可以根据这个功能来设计一种负载均衡的节点。当节点认为自己负载较大的时候，就不再直接给浏览器发送所请求的网页，而是发送一个Location指令，让浏览器在计算机集群中的其他节点上获得所需要的网页。

这种方式的具体实现有很多困难，例如，一个节点如何能保证它重定向的节点是比较空闲的，并且不会再次发送Location指令等。Location指令和浏览器都没有这方面的支持能力，这样，很容

易形成一种死循环。因此，在实际应用中，这种方式并不多见，使用这种方式实现的计算机集群软件也较少。

(2) 基于DNS的负载均衡。基于DNS的负载均衡是在DNS服务器中为同一个主机名配置多个IP地址，在应答DNS查询时，DNS服务器对每个查询将以DNS文件中主机记录的IP地址按顺序返回不同的解析结果，将客户端的访问引导到不同的节点上去，使得不同的客户端访问不同的节点，从而达到负载均衡的目的。

DNS负载均衡的优点是经济、简单易行，并且节点可以位于Internet上任意的位置。但它也存在不少缺点，例如，为了保证DNS数据及时更新，一般都要将DNS的刷新时间设置得较小，但太小就会造成太大的额外网络流量，并且更改了DNS数据之后也不能立即生效；DNS负载均衡采用的是简单的轮转算法，不能区分节点之间的差异，不能反映节点的当前运行状态，不能做到为性能较好的节点多分配请求，甚至会出现客户请求集中在某一个节点上的情况。另外，要给每个节点分配一个Internet上的IP地址，这势必会占用过多的IP地址。

(3) 基于NAT (Network Address Translation, 网络地址转换) 的负载均衡。基于NAT的负载均衡将一个外部IP地址映射为多个内部IP地址，对每次连接请求动态地转换为一个内部节点的地址，将外部连接请求引到转换得到地址的那个节点上，从而达到负载均衡的目的。基于NAT的负载均衡是一种比较完善的负载均衡技术，起着NAT负载均衡功能的设备一般处于内部节点到外部网之间的网关位置，例如，路由器、防火墙、四层交换机、专用负载均衡器等，均衡算法也较灵活，例如，使用随机选择、最小连接数等来分配负载。

基于NAT的负载均衡可以通过软硬件方式来实现。通过软件方式来实现NAT负载均衡的设备往往受到带宽和系统本身处理能力的限制，由于NAT比较接近网络的低层，因此，可以将它集成在硬件设备中，例如，四层交换机和专用负载均衡器等，四层交换机的一项重要功能就是基于NAT的负载均衡。

(4) 反向代理负载均衡。反向代理负载均衡是将来自Internet上的连接请求以反向代理的方式动态地转发给内部网络上的多个节点进行处理，从而达到负载均衡的目的。反向代理负载均衡既能以软件方式实现，也能在高速缓存器和负载均衡器等硬件设备上实现。反向代理负载均衡可以将优化的负载均衡策略和代理服务器的高速缓存技术结合在一起，提升静态网页的访问速度，提高系统性能。另外，由于网络外部用户不能直接访问真实的节点计算机，反向代理负载均衡还具备额外的安全性（同理，基于NAT的负载均衡也有此优点）。

反向代理负载均衡的缺点主要表现在两个方面。首先，反向代理处于OSI参考模型应用层，因此，必须为每种应用服务专门开发一个反向代理服务器，这样，就限制了反向代理负载均衡技术的应用范围，现在一般都用于对Web服务器的负载均衡；其次，针对每一次代理，代理服务器都必须打开两个连接，一个对外，一个对内。在并发连接请求数量非常大的时候，代理服务器的负载也就非常大，代理服务器本身会成为服务的瓶颈。

(5) 混合型负载均衡。在有些大型网络中，由于多个计算机集群内硬件设备、各自的规模、提供的服务等的差异，可以考虑给每个集群采用最合适的负载均衡方式，然后又在这多个集群之间再一次进行负载均衡（即将每个集群系统当做新的集群中的一个节点），从而达到最佳的性能。

试题答案

(16) B

试题6 (2011年11月试题17)

试题6 (2011年11月试题17)

数据备份是信息系统运行管理时保护数据的重要措施。 (17) 可针对上次任何一种备份进行，将上次备份后所有发生变化的数据进行备份，并将备份后的数据进行标记。

- (17) A.增量备份 B.差异备份 C.完全备份 D.按需备份

试题分析

数据备份从备份量来分，可以分为完全备份、增量备份、差异备份。

完全备份：备份所有数据。即使两个备份时间点之间数据没有任何变动，所有数据还是会备份下来。

增量备份：跟完全备份不同，增量备份在做数据备份前会先判断数据的最后修改时间是否比上次备份的时间晚。如果不是，则表示该数据并没有被修改过，这次不需要备份。所以该备份方式，只记录上次备份之后的变动情况，而非完全备份。

差异备份：差异备份与增量备份一样，都只备份变动过的数据。但前者的备份是针对上次完整备份后，曾被更新过的。

从以上对备份方式的分析可以得知：增量备份可针对上次任何一种备份进行。

试题答案

- (17) A

试题7 (2012年11月试题17)

试题7 (2012年11月试题17)

峰值MIPS (每秒百万次指令数) 用来描述计算机的定点运算速度，通过对计算机指令集中基本指令的执行速度计算得到。假设某计算机中基本指令的执行需要5个机器周期，每个机器周期为3微秒，则该计算机的定点运算速度为 (17) MIPS.

- (17) A.8 B.15 C.0.125 D.0.067

试题分析

在计算机技术的发展过程中，性能评估的常用方法有时钟频率法、指令执行速度法、等效指令速度法、数据处理速率法、综合理论性能法和基准程序法等。

1.时钟频率法

计算机的时钟频率在一定程度上反映了机器速度。显然，对同一种机型的计算机，时钟频率越高，计算机的工作速度就越快。但是，由于不同的计算机硬件电路和器件的不完全相同，所以其所需要的时钟频率范围也不一定相同。相同频率、不同体系结构的机器，其速度和性能可能会相差很多倍。

在计算机中，为了便于管理，常把一条指令的执行过程划分为若干个阶段，每一个阶段完成一项工作。例如，取指令、存储器读、存储器写等，这每一项工作称为一个基本操作。完成一个基本操作所需要的时间称为机器周期。一般情况下，一个机器周期由若干个时钟周期组成。

指令周期是执行一条指令所需要的时间，一般由若干个机器周期组成。指令不同，所需的机器周期数也不同。对于一些简单的单字节指令，在取指令周期中，指令取出到指令寄存器后，立即译码执行，不再需要其他的机器周期。对于一些比较复杂的指令，例如，转移指令、乘法指令等，则需要两个或者两个以上的机器周期。

为了帮助读者搞清楚这些概念之间的关系，下面，通过一个例子来说明。

假设微机A和微机B采用同样的CPU，微机A的主频为20MHz，微机B的主频为60MHz。如果两个时钟周期组成一个机器周期，平均三个机器周期可完成一条指令，则微机A的时钟周期为 $1/(20M) = 50\text{ns}$ ，机器周期为 $2 \times 50\text{ns} = 100\text{ns}$ ，平均指令周期为 $3 \times 100\text{ns} = 300\text{ns}$ 。也就是说，指令平均执行速度为 $1/(300\text{ns}) \approx 3.33\text{MIPS}$ 。因为微机B的主频为60MHz，是微机A主频的 $60/20 = 3$ 倍，所以，微机B的平均指令执行速度应该比微机A的快3倍，即微机B的指令平均执行速度为 $3.33 \times 3 \approx 10\text{MIPS}$ 。

2. 指令执行速度法

在计算机发展的初期，曾用加法指令的运算速度来衡量计算机的速度。因为加法指令的运算速度大体上可反映出乘法、除法等其他算术运算的速度，而且逻辑运算、转移指令等简单指令的执行时间往往设计成与加法指令相同，因此，加法指令的运算速度有一定代表性。

表示机器运算速度的单位是MIPS。常用的有峰值MIPS、基准程序MIPS和以特定系统为基准的MIPS。MIPS依赖于指令集，所以用MIPS比较指令集不同的系统性能是很不准确的。在同一台机器上，MIPS因程序不同而变化，这种变化有时是很大的。用MIPS进行测试，得到的性能结果可能与事实相反。例如，因为浮点运算远慢于整数运算，所以很多计算机提供了可选的硬件浮点运算部件，但是软件实现浮点运算的MIPS高，而硬件实现浮点运算的时间少。这时，MIPS与计算机性能恰好相反。

MFLOPS用于衡量计算机的科学计算速度，常用的有峰值MFLOPS和以基准程序测得的MFLOPS。MFLOPS可用于比较和评价在同一系统上求解同一问题的不同算法的性能，还可用于在同一源程序、同一编译器、相同的优化措施、同样的运行环境下，测试不同系统的浮点运算速度。由于实际程序中各种操作所占比例不同，所以测得的MFLOPS也不相同。MFLOPS值没有考虑运算部件与存储器、I/O系统等速度之间相互协调等因素，因此，只能说明在特定条件下的浮点运算速度，而不能体现计算机的整体性能。

3. 等效指令速度法

等效指令速度法也称为吉普森混合法 (Gibson mix) 或混合比例计算法，是通过各类指令在程序中所占的比例 (W_i) 进行计算得到的。若各类指令的执行时间为 t_i ，则等效指令的执行时间为：

$$T = \sum_{i=1}^n W_i t_i$$

其中， n 为指令类型数。

对某些程序来说，采用等效指令速度法可能严重偏离实际，尤其是对CISC系统，因为某些指令的执行时间是不固定的，数据的长度、cache的命中率、流水线的效率等都会影响计算机的运算速度。

4. 数据处理速率法

因为在不同程序中，各类指令使用频率是不同的，所以，固定比例方法存在着很大的局限性，而且数据长度与指令功能的强弱对计算的速度影响极大。同时，这种方法也不能反映现代计算机中

cache、流水线、交叉存储等结构的影响。具有这种结构的计算机的性能不仅与指令的执行频率有关，而且也与指令的执行顺序与地址的分布有关。

数据处理速率法 (Processing Data Rate, PDR) 采用计算PDR值的方法来衡量机器性能，PDR值越大，机器性能越好。PDR与每条指令和每个操作数的平均位数以及每条指令的平均运算速度有关，其计算方法如下：

$$PDR = L/R$$

其中， $L=0.85G+0.15H+0.4J+0.15K$, $R=0.85M+0.09N+0.06P$. 式中G是每条定点指令的位数，M是平均定点加法时间，H是每条浮点指令的位数，N是平均浮点加法时间，J是定点操作数的位数，P是平均浮点乘法时间，K是浮点操作数的位数。此外，还作了一系列的规定。

PDR值主要对CPU和主存储器的速度进行度量，但不适合衡量机器的整体速度，不能全面反映计算机的性能，因为它没有涉及cache、多功能部件等技术对性能的影响。PDR曾是美国及巴黎统筹委员会用来限制计算机出口的系统性能指标估算方法，1991年9月停止使用，取而代之的是CTP (Composite Theoretical Performance, 综合理论性能) 。

5. 综合理论性能法

CTP是美国政府为限制高性能计算机出口所设置的运算部件综合性能估算方法。CTP用MTOPS (Million Theoretical Operations Per Second, 每秒百万次理论运算) 表示。CTP的估算方法是，首先算出处理部件每个计算单元 (例如，定点加法单元、定点乘法单元、浮点加法单元、浮点乘法单元等) 的有效计算率，再按不同字长加以调整，得出该计算单元的理论性能，所有组成该处理部件的计算单元的理论性能之和即为CTP.

在本题中，由于题目中提到"计算机中基本指令的执行需要5个机器周期"且"每个机器周期为3微秒", 所以1条基本指令的执行时间为15微秒。所以1秒钟能完成的指令数为：
 $1*1000*1000/15=66666$. 即0.067MIPS.

试题答案

(17) D

第 12 章：专业英语

考点分析

第12章专业英语

12.1考点分析

根据考试大纲，专业英语对考生的要求非常空泛：

- (1) 具有高级工程师所要求的英文阅读水平；
- (2) 掌握本领域的英语术语。

从目前的考试情况来看架构设计师专业英语主要考查软件架构设计相关基础知识。具体考查知识点分布情况如表12-1所示。

表12-1历年考查知识点分布情况表

试 题	考查知识点
2009年11月试题 71~75	架构风格
2010年11月试题 71~75	软件架构
2011年11月试题 71~75	信息系统设计
2012年11月试题 71~75	分层数据流图(DFD)

第12章：专业英语

试题1 (2009年11月试题71~75)

12.2 试题精解

试题1 (2009年11月试题71~75)

An architectural style defines as a family of such systems in terms of a (71) of structural organization. More specifically, an architectural style defines a vocabulary of (72) and connector types, and a set of (73) on how they can be combined. For many styles there may also exist one or more (74) that specify how to determine a system's overall properties from the properties of its parts. Many of architectural styles have been developed over the years.

The best-known examples of (75) architectures are programs written in the Unix shell.

- (71) A.pattern B.data flow C.business process D.position level
- (72) A.metadata B.components C.models D.entities
- (73) A.functions B.code segments C.interfaces D.constraints
- (74) A.semantic models B.weak entities
C.data schemas D.business models
- (75) A.event-based B.object-oriented C.pipe-and-filter D.layered

参考译文

一种架构风格以一种结构化组织模式定义一组这样的系统。具体来说，一种架构风格定义了一个构件及连接器类型的词汇表，以及一组关于它们如何能够被关联的约束对于许多风格来说，可能也存在一个或多个语义模型，从系统部件的特性来确定系统的整体特性。许多架构风格已经发展了很多年，众所周知的管道-过滤器架构的例子就是用UNIX shell编写的程序。

试题答案

- (71) A (72) B (73) D (74) A (75) C

第12章：专业英语

试题2 (2010年11月试题71~75)

试题2 (2010年11月试题71~75)

The software architecture is a set of software components, subsystems, relationships, interactions, the properties of each of these elements, and the set of guiding principles that together constitute the fundamental properties and constraints of a software system or set of systems.

(71) defines a general set of element types and their interactions. The examples include Pipes and Filters, Model-View-Controller, and Reflection. A (72) in software architecture is a representation used to understand or document one or more aspects of a problem or solution. Architecture is usually used in conjunction with many adjunct terms. The (73) defines the key strategies, organization, goals and related processes of the enterprise. At the enterprise level, the (74) may be more of a set of guidelines on how the various software architectures should be constructed consistently across the enterprise. The (75), which describes the high-level set of elements involved in application from a particular domain along with their interactions, is often used to focus on subsystem definition rather than application process level definition.

- (71) A.Architectural pattern B.Architectural description
- C.Architectural view D.Architectural viewpoint
- (72) A.model B.domain C.component D.subsystem
- (73) A.enterprise architecture B.technical architecture
- C.infrastructure architecture D.business architecture
- (74) A.enterprise architecture B.data architecture
- C.application architecture D.information architecture
- (75) A.product-line architecture B.reference architecture
- C.technology architecture D.infrastructure architecture

参考译文

软件架构是一组软件构件、子系统、关联关系、交互关系以及其中每个元素的特性和一组指导原则，这些共同构成一个软件系统或一组系统的基本特性和约束。架构模式定义了一组通用的元素类型及其交互关系。其中的例子包括Pipes and Filters、MVC和Reflection。软件架构中的模型是一种表现形式，用于从一个或多个方面理解或记录一个问题或解决方案。架构通常会与一些附属词结合起来使用。业务架构定义了企业的关键策略、组织、目标和相关过程。在企业层面，应用架构可能更多的是一组关于应该如何在整个企业一致地构建各种软件架构的指导原则。参考架构描述了源自特定领域的应用涉及的高层元素集合及其交互关系，通常用于关注子系统的定义而不是应用过程级别定义。

试题答案

- (71) A (72) A (73) D (74) C (75) B

试题3 (2011年11月试题71~75)

试题3 (2011年11月试题71~75)

Information systems design is defined as those tasks that focus on the specification of a detailed computer-based solution. Typically, there are four systems design tasks for in-house development.

1) The first task is to specify (71) , which defines the technologies to be used by one, more, or all information systems in terms of their data, processes, interfaces, and network components. This task is accomplished by analyzing the data models and process models that are initially created during requirements analysis.

2) The next systems design task is to develop the (72) . The purpose of this task is to prepare technical design specifications for a database that will be adaptable to future requirements and expansion.

3) Once the database prototype has been built, the systems designer can work closely with system users to develop input, output and dialogue specifications. The (73) must be specified to ensure that the outputs are not lost, misrouted, misused, or incomplete.

4) The fourth design task involves packaging all the specifications from the previous design tasks into a set of specifications that will guide the (74) activities during the following phases of the systems development methodology.

Finally, we should (75) and update the project plan accordingly. The key deliverable should include a detailed plan for the construction phase that should follow.

(71) A.an application architecture B.a distributed system

C.a system scope D.a system physical model

(72) A.database design specifications B.database organization decisions

C.data structure specifications D.data distribution decisions

(73) A.format and layout B.transaction details

C.additional instructions D.internal controls

(74) A.system administrator's B.system analyst's

C.computer programmer's D.system designer's

(75) A.adjust the project schedule B.reevaluate project feasibility

C.evaluate vendor proposals D.select the best vendor proposal

参考译文

信息系统设计被定义为一些任务，它们主要关注一个详细的计算机解决方案的规格说明。通常来说，内部开发有四种系统设计任务。

1) 第一项任务是确定一个应用程序架构，它以数据、过程、接口和网络组件的方式定义一个、多个或所有信息系统要使用的技术。完成这项任务需要分析最初创建于需求分析期间的数据模型和过程模型。

2) 下一项系统设计任务是开发数据库设计的规格说明。该任务的目的是准备一个数据库技术设计规格说明，以适应将来的需求和扩展。

3) 一旦建成了数据库原型，系统设计人员能够和系统用户密切合作开发输入、输出和对话框规格说明。必须指定内部控件来确保输出不会丢失、误传、滥用或不完整。

4) 第四项设计任务包括把之前所有设计任务的规格说明打包为一套规格说明 , 将在系统开发方法的后续阶段中指导计算机程序员的活动。

最后 , 我们应该重新评估项目的可行性并相应地更新项目计划。主要交付成果将包括构建阶段应该遵循的一个详细计划。

试题答案

(71) A (72) A (73) D (74) C (75) B

第 12 章 : 专业英语

试题4 (2012年11月试题71~75)

试题4 (2012年11月试题71~75)

An application architecture specifies the technologies to be used to implement one or more information systems. It serves as an outline for detailed design, construction, and implementation. Given the models and details, include (71) , we can distribute data and processes to create a general design of application architecture. The design will normally be constrained by architecture standards, project objectives, and (72) . The first physical DFD to be drawn is the (73) . The next step is to distribute data stores to different processors. Data (74)

are two types of distributed data which most RDBMSs support. There are many distribution options used in data distribution. In the case of (75) we should record each table as a data store on the physical DFD and connect each to the appropriate server.

- (71) A.logical DFDs and ERD
- B.ideal object model and analysis class model
- C.use case models and interface prototypes
- D.physical DFDs and database schema

- (72) A.the database management system
- B.the feasibility of techniques used
- C.the network topology and technology
- D.the user interface and process methods

- (73) A.context DFD B.system DFD
- C.network architecture DFD D.event-response DFD

- (74) A.vertical partitioning and horizontal replication
- B.vertical replication and horizontal partitioning
- C.integration and distribution
- D.partitioning and replication

- (75) A.storing all data on a single server
- B.storing specific tables on different servers

- C.storing subsets of specific tables on different servers
- D.duplicating specific tables or subsets on different servers

参考译文

应用架构说明了实现一个或多个信息系统所使用的技术，它作为详细设计、构造和实现的一个大纲。给定了包括逻辑数据流图和实体联系图在内的模型和详细资料，我们可以分配数据和过程以创建应用架构的一个概要设计。概要设计通常会受到架构标准、项目目标和所使用技术的可行性的制约。需要绘制的第一个物理数据流图是网络架构数据流图。接下来是分配数据存储到不同的处理器。数据分区和复制是大多数关系型数据库支持的两种分布式数据形式。有许多分配方法用于数据分布。在不同服务器上存储特定表的情况下，我们应该将每个表记录为物理数据流图中的一个数据存储，并将其连接到相应的服务器。

参考答案

- (71) A (72) B (73) C (74) D (75) B

第13章：案例分析

考点分析

第13章案例分析

13.1考点分析

根据考试大纲，系统架构设计师考试中的案例分析试题涉及以下内容。

(1) 系统规划：包括系统项目的提出与可行性分析，系统方案的制定，评价和改进，新旧系统的分析和比较，现有软件、硬件和数据资源的有效利用。

(2) 软件架构设计：软件架构设计、XML技术、基于架构的软件开发过程、软件质量属性、架构模型（风格）、特定领域软件架构、基于架构的软件开发方法、架构评估、软件产品线、系统演化。

(3) 设计模式：设计模式的概念、设计模式的组成、模式和软件架构、设计模式分类、设计模式的实现。

(4) 系统设计：处理流程设计、人机界面设计、文件设计、存储设计、数据库设计、网络应用系统的设计、系统运行环境的集成与设计、中间件、应用服务器、性能设计与性能评估、系统转换计划。

(5) 软件系统建模：系统需求、建模的作用和意义、定义问题（目标、功能、性能等）与归结模型（静态结构模型、动态行为模型、物理模型）、结构化系统建模、数据流图、面向对象系统建模、统一建模语言（UML）、数据库建模、E-R图、逆向工程。

(6) 分布式系统设计：分布式通信协议的设计、基于对象的分布式系统设计、基于Web的分布式系统设计、基于消息和协同的分布式系统设计、异构分布式系统的互操作性设计。

(7) 嵌入式系统设计：实时系统和嵌入式系统特征、实时任务调度和多任务设计、中断处理和异常处理、嵌入式系统开发设计。

(8) 系统的可靠性分析与设计：系统的故障模型和可靠性模型、系统的可靠性分析和可靠度计算、提高系统可靠性的措施、系统的故障对策和系统的备份与恢复。

(9) 系统的安全性和保密性设计：系统的访问控制技术、数据的完整性、数据与文件的加密、通信的安全性、系统的安全性设计。

在历年的考试试题中，系统架构设计师案例分析试题一共5道题，试题一为必答题，试题二到试题五选答（四选二），每道试题25分，满分为75分。每道试题的结构基本相同，一般首先是一段说明性文字（即案例背景介绍），然后跟随3至4个问题。历年考试的试题如表13-1所示。

表13-1系统架构设计师案例分析试题分布表

考试时间	主要知识点	考查内容说明
2009. 11	软件质量特性	软件质量特性中子特性的归类，每个质量特性在开发中体现于哪些场景，或哪些需求对应着哪些特性
	系统流程图与数据流图	系统流程图与数据流图的区别、以及数据流图的基本应用
2009. 11	嵌入式系统架构	架构风格
	系统软件架构	系统软件架构的选择
	系统安全	安全风险与身份认证方式
2010. 11	软件架构设计	常见架构风格的选用
	数据架构建模	分布式数据库的设计
	嵌入式系统	层次化架构
	ESB	ESB 及设计模式
	系统可靠性	可靠性的计算、检错技术和容错技术
2011. 11	软件架构评估	软件质量属性场景、架构风险、敏感点、权衡点
	软件系统数据建模	反规范化技术的优点、缺点、实施技术
	嵌入式系统设计	嵌入式系统架构及系统设计
	Web 应用系统架构设计	数据采集与统计方案选择、REST
	信息系统安全性	对称加密策略与公钥加密策略、机密性与完整性机制、授权决策机制
2012. 11	软件架构风格	不同架构风格的优势、劣势、应用场合的比较分析
	分布式系统设计	ODP 的五个视点、分布式中间件的应用、分布式系统通信问题
	嵌入式软件架构设计	GOA 的相关概念与特点、GOA 中的一系列接口
	设计模式	创建型、结构型、行为型模型的作用，设计模式的选择与应用
	数据架构设计	NoSQL 的特点及优缺点

第 13 章：案例分析

试题1（2009年11月试题1）

13.2 试题精解

试题1（2009年11月试题1）

阅读以下软件架构设计的问题，在答题纸上回答问题1和问题2。

某软件开发公司欲为某电子商务企业开发一个在线交易平台，支持客户完成网上购物活动中的在线交易。在系统开发之初，企业对该平台提出了如下要求。

- (1) 在线交易平台必须在1秒内完成客户的交易请求。
- (2) 该平台必须保证客户个人信息和交易信息的安全。
- (3) 当发生故障时，该平台的平均故障恢复时间必须小于10秒。
- (4) 由于企业业务发展较快，需要经常为该平台添加新功能或进行硬件升级。添加新功能或进行硬件升级必须在6小时内完成。

针对这些要求，该软件开发公司决定采用基于架构的软件开发方法，以架构为核心进行在线交易平台的设计与实现。

【问题1】

软件质量属性是影响软件架构设计的重要因素。请用200字以内的文字列举六种不同的软件质量属性名称，并解释其含义。

【问题2】

请对该在线交易平台的4个要求进行分析，用300字以内的文字指出每个要求对应何种软件质量属性；并针对每种软件质量属性，各给出2种实现该质量属性的架构设计策略。

试题分析

这是一道软件质量特性的试题，软件质量特性是软件架构以及软件架构设计师的一个重要关注点。因为如果在软件架构的设计阶段不考虑软件质量特性，则产生的软件质量隐患是在后期的设计与开发中无法弥补的。

软件质量特性主要包括以下几个方面。

- ① 功能性：系统所能完成期望工作的能力。
- ② 性能：系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理事件的个数。
- ③ 可用性：系统能够正常运行的时间比例。
- ④ 可靠性：软件系统在应用或错误面前，在意外或错误使用的情况下维持软件系统功能特性的基本能力。
- ⑤ 健壮性：在处理或环境中，系统能够承受压力或变更的能力。
- ⑥ 安全性：系统在向合法用户提供服务的同时，能够阻止非授权用户使用的企图或拒绝服务的能力。
- ⑦ 可修改性：能够快速地以较高的性能价格比对系统进行变更的能力。
- ⑧ 可变性：体系结构经扩充或变更成为新体系结构的能力。
- ⑨ 易用性：衡量用户使用一个软件产品完成指定任务的难易程度。
- ⑩ 可测试性：软件发现故障并隔离、定位其故障的能力特性，以及在一定的时间和成本前提下，进行测试设计、测试执行的能力。

互操作性：系统与外界或系统与系统之间的相互作用能力。

问题1是纯概念题，从以上的属性中任选6个作答即可。

问题2难度稍大，需要结合题目给出的案例，来分析系统有哪些质量属性的需求，同时需要给出实现该质量属性的策略。下面逐一分析题目给出的场景。

(1) 在线交易平台必须在1秒内完成客户的交易请求。该要求主要对应性能，可以采用的架构设计策略有增加计算资源、改善资源需求（减少计算复杂度等）、资源管理（并发、数据复制等）和资源调度（先进先出队列、优先级队列等）。

(2) 该平台必须严格保证客户个人信息和交易信息的保密性和安全性。该要求主要对应安全性，可以采用的架构设计策略有抵御攻击（授权、认证和限制访问等）、攻击检测（入侵检测等）、从攻击中恢复（部分可用性策略）和信息审计等。

(3) 当发生故障时，该平台的平均故障恢复时间必须小于10秒。该要求主要对应可用性，可以采用的架构设计策略有Ping/Echo、心跳、异常和主动冗余等。

(4) 由于企业业务发展较快，需要经常为该平台添加新功能或进行硬件升级。添加新功能或进行平台升级必须在6小时内完成。该要求主要对应可修改性，可以采用的架构设计策略有软件模块泛化、限制模块之间通信、使用中介和延迟绑定等。

试题答案

【问题1】

常见的软件质量属性有多种，例如性能（Performance）、可用性（Availability）、可靠性（Reliability）、健壮性（Robustness）、安全性（Security）、可修改性（Modification）、可变性（Changeability）、易用性（Usability）、可测试性（Testability）、功能性（Functionality）和互操作性（Inter-operation）等。

这些质量属性的具体含义如下。

① 性能是指系统的响应能力，即要过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理事件的个数。

② 可用性是系统能够正常运行的时间比例。

③ 可靠性是指软件系统在应用或错误面前，在意外或错误使用的情况下维持软件系统功能特性基本能力。

④ 健壮性是指在处理或环境中，系统能够承受压力或变更的能力。

⑤ 安全性是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务能力。

⑥ 可修改性是指能够快速地以较高的性能价格比对系统进行变更的能力。

⑦ 可变性是指体系结构经扩充或变更成为新体系结构的能力。

⑧ 易用性是衡量用户使用一个软件产品完成指定任务的难易程度。

⑨ 可测试性是指软件发现故障并隔离、定位其故障的能力特性，以及在一定的时间和成本前提下，进行测试设计、测试执行的能力。

⑩ 功能性是系统所能完成所期望工作的能力。

互操作性是指系统与外界或系统与系统之间的相互作用能力。

【问题2】

(1) 在线交易平台必须在1秒内完成客户的交易请求。该要求主要对应性能，可以采用的架构设计策略有增加计算资源、改善资源需求（减少计算复杂度等）、资源管理（并发、数据复制等）和资源调度（先进先出队列、优先级队列等）。

(2) 该平台必须严格保证客户个人信息和交易信息的保密性和安全性。该要求主要对应安全性，可以采用的架构设计策略有抵御攻击（授权、认证和限制访问等）、攻击检测（入侵检测等）、从攻击中恢复（部分可用性策略）和信息审计等。

(3) 当发生故障时，该平台的平均故障恢复时间必须小于10秒。该要求主要对应可用性，可以采用的架构设计策略有Ping/Echo、心跳、异常和主动冗余等。

(4) 由于企业业务发展较快，需要经常为该平台添加新功能或进行硬件升级。添加新功能或进行平台升级必须在6小时内完成。该要求主要对应可修改性，可以采用的架构设计策略有软件模块泛化、限制模块之间通信、使用中介和延迟绑定等。

试题2（2009年11月试题2）

试题2（2009年11月试题2）

阅读以下关于结构化软件系统建模的叙述，在答题纸上回答问题1至问题3。

希赛公司拟开发一个商业情报处理系统，使公司能够针对市场环境的变化及时调整发展战略，以获取最大的商业利益。项目组经过讨论，决定采用结构化分析和设计方法。在系统分析阶段，为了更好地对情报数据处理流程及其与外部角色的关联进行建模，项目组成员分别给出了自己的设计思路：

（1）小张提出先构建系统流程图（System Flowcharts），以便更精确地反映系统的业务处理过程及数据的输入和输出；

（2）小李提出先构建系统数据流图（Data Flow Diagrams），来展现系统的处理过程和定义业务功能边界，并给出了情报分类子系统的0层和1层数据流图，后者如图13-1所示。

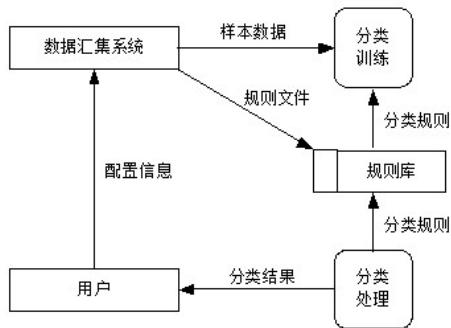


图13-1 情报分类子系统的1层数据流图

项目组经讨论确定以数据流图作为本阶段的建模手段。工程师老王详细说明了流程图和数据流图之间的区别与联系，并指出了图13-1的数据流图中存在的错误。

【问题1】

流程图和数据流图是软件系统分析设计中常用的两种手段，请用300字以内文字简要说明流程图与数据流图的含义及其区别，并说明项目组为何确定采用数据流图作为建模手段。

【问题2】

请分析指出图13-1所示的数据流图中存在的错误及其原因，并针对图13-1的1层数据流图绘制出情报分类子系统的0层数据流图。

【问题3】

高质量的数据流图是可读的、内部一致的并能够准确表示系统需求。请用300字以内文字说明在设计高质量的数据流图时应考虑的三个原则。

试题分析

这是一道流程图与数据流图的试题，主要考查系统流程图与数据流图的区别、以及数据流图的基本应用。

【问题1】

问题1主要考查系统流程图与数据流图的区别。系统流程图是描绘物理系统的传统工具。它的基本思想是用图形符号以黑盒子形式描绘系统里面的每个部件（程序、文件、数据库、表格、人工过程等）。系统流程图表达的是部件的信息流程，而不是表示对信息进行加工处理的控制过程。

数据流图就是采用图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

1. 数据流图的主要作用

数据流图从数据传递和加工的角度，利用图形符号通过逐层细分描述系统内各个部件的功能和数据在它们之间传递的情况，来说明系统所完成的功能。具体来说，数据流图的主要作用如下：

(1) 数据流图是理解和表达用户需求的工具，是需求分析的手段。由于数据流图简明易懂，不需要任何计算机专业知识就可以理解它，因此，系统分析师可以通过数据流图与用户进行交流。

(2) 数据流图概括地描述了系统的内部逻辑过程，是需求分析结果的表达工具，也是系统设计的重要参考资料，是系统设计的起点。

(3) 数据流图作为一个存档的文字材料，是进一步修改和充实开发计划的依据。

2. 数据流图的基本符号

在数据流图中，通常会出现4种基本符号，分别是数据流、加工、数据存储和外部实体（数据源及数据终点）。数据流是具有名字和流向的数据，在数据流图中用标有名字的箭头表示。加工是对数据流的变换，一般用圆圈表示。数据存储是可访问的存储信息，一般用直线段表示。外部实体是位于被建模的系统之外的信息生产者或消费者，是不能由计算机处理的成分，它们分别表明数据处理过程的数据来源及数据去向，用标有名字的方框表示。

3. 数据流图的层次

结构化方法的思路是依赖于数据流图进行自顶而下的分析。这也是因为系统通常比较复杂，很难在一张图上就将所有的数据流和加工描述清楚。因此，数据流图提供一种表现系统高层和低层概念的机制。也就是先绘制一张较高层次的数据流图，然后在此基础上，对其中的加工进行分解，分解成为若干个独立的、低层次的、详细的数据流图，而且可以这样逐一的分解下去，直至系统被清晰地描述出来。

(1) 顶层图。顶层图是描述系统最高层结构的数据流图，它的特点是将整个待开发的系统表示为一个加工，将所有的外部实体和进出系统的数据流都画在一张图中。例如，图13-2就是一个顶层图的实例，只不过在绘制时做了一些处理，使得它看上去更加直观易懂。

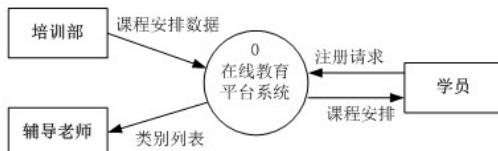


图13-2 顶层图示例

顶层图用来描述系统有什么输入和输出数据流，与哪些外部实体直接相关，可以把整个系统的范围勾画出来。

(2) 逐层分解。当完成了顶层图的建模之后，就可以在此基础上进行进一步的分解。对图13-2进行分解，在对原有流程了解的基础上，可以得到图13-3。

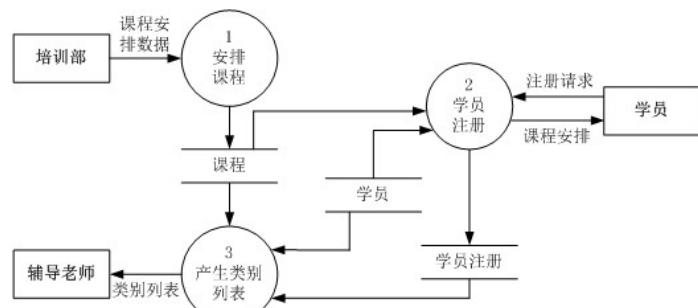


图13-3 0层图示例

图13-3是在顶层图13-2的基础上做的第一次分解，而在图13-2中只有一个加工，那就是系统本身，可以将其编号为0.因此，对顶层图进行的分解，其实就是对这个编号为0的加工进行更细化的描述，在这里引入了新的加工和数据存储，为了能够区分其位于的级别，在这个层次上的加工将以1、2、3为序列进行编号。

正是由于这是对加工0的分解，因此也称为0层图。可以根据需要对0层图上的加工进行类似的再分解，称之为1层图，在1层图中引入的新加工，其编号规则就是1.1、1.2、…，以及2.1、2.2、…，依次类推，直到完成分析工作。

4. 如何画数据流图

数据流图的绘制是一个自顶向下、由外到里的过程，通常按照以下几个步骤进行：

(1) 画系统的输入和输出：在图的边缘标出系统的输入数据流和输出数据流。这一步其实是决定研究的内容和系统的范围。在画的时候，可以先将尽可能多的数据流画出来，然后再删除多余的，增加遗漏的。

(2) 画数据流图的内部：将系统的输入、输出用一系列的处理连接起来，可以从输入数据流画向输出数据流，也可以从中间画出去。

(3) 为每一个数据流命名：命名的好坏与数据流图的可理解性密切相关，应避免使用空洞的名字。

(4) 为加工命名：使用动宾短语为每个加工命名。

每画好一张数据流图，就需要进行检查和修改，检查和修改的原则如下：

(1) 数据流图中的所有图形符号只限于前述4种基本图形元素，图上每个元素都必须有名字。

(2) 每个加工至少有一个输入数据流和一个输出数据流，而且要保持数据守恒。也就是说，一个加工的所有输出数据流中的数据必须能从该加工的输入流中直接获得，或者通过该加工能产生的数据。一个加工的输出数据流不应与输入数据流同名，即使它们的组成完全相同。

(3) 在数据流图中，需按层给加工编号。编号表明了该加工处在哪一层，以及上下层的父图与子图的对应关系。

(4) 规定任何一个数据流图子图必须与它上层的一个加工对应，两者的输入数据流和输出数据流必须一致。此即父图与子图的平衡。也就是说，父图中的某加工的输入输出流必须与它的所有子图的输入输出数据流在数量上和名字上相同。值得注意的是，如果父图中的一个输入（输出）数据流对应于子图中的几个输入（输出）数据流，而子图中组成这些数据流的数据项的全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

(5) 在整套数据流图中，每个数据存储必须既有读的数据流，又有写的数据流。但是在某张子图中，可能只有读没有写，或者只有写没有读。

(6) 可以在数据流图中加入物质流，帮助用户理解数据流图，但不可夹带控制流。

数据流图和流程图的区别主要表现在：数据流图用于系统的逻辑建模，而流程图用于物理建模。数据流图展现的是全局处理过程，而流程图往往是局部的处理过程。流程图是顺序执行的处理过程，即一个时间点，通常是处理一项事务，而数据流图的处理过程是并行的，多个处理流程同时进行。数据流图展现的是系统数据流，而流程图展现的是系统控制流。

【问题2】

问题2是找错误数据流以及根据低层数据流图来画高层数据流图。

在找错误数据流时，首先需要了解错误数据流的类型：一类是语法错误，包括外部实体之间、数据存储之间或外部实体与数据存储之间不经过加工而存在直接数据流；另一类是逻辑错误，包括数据黑洞（只有输入没有产生输出）、灰洞（输入不足以产生输出）和无输入。

"分类训练"加工属于数据黑洞错误；
"分类处理"加工属于无输入错误；
"规则文件"和"配置信息"数据流属于语法错误。

对于复杂系统，一般采用分层数据流图逐步细化系统的内部处理逻辑。0层数据流图是1层数据流图更高级别的抽象。"分类训练"和"分类处理"加工属于内部加工，"分类规则"数据流属于内部数据流，抽象为"情报分类子系统"加工。其中，"样本数据"、"规则文件"和"配置信息"为输入数据流；"分类结果"为输出数据流。

【问题3】

问题3属于纯概念题，高质量的数据流图首先要求满足数据流一致性要求，即数据流的输入/输出应该匹配，输入数据的信息量足以产生输出数据，所有的输出数据都能够从输入数据中找到数据的来源。对于复杂性系统，可以采用数据流图分层结构将系统的复杂性逐步分解降低，以提高数据流图的有效性。高质量的数据流图应该简化复杂的接口，最小化接口或连接的数目。

试题答案

【试题1】

数据流图作为一种图形化工具，用来说明业务处理过程、系统边界内所包含的功能和系统中的数据流。

流程图以图形化的方式展示应用程序从数据输入开始到获得输出为止的逻辑过程，描述处理过程的控制流。

两者的区别如下。

- ① 数据流图中的处理过程可并行；流程图在某个时间点只能处于一个处理过程。
- ② 数据流图展现系统的数据流；流程图展现系统的控制流。
- ③ 数据流图展现全局的处理过程，过程之间遵循不同的计时标准；流程图中处理过程遵循一致的计时标准。
- ④ 数据流图适用于系统分析中的逻辑建模阶段；流程图适用于系统设计中的物理模阶段。

【问题2】

如图13-1所示的数据流图中存在的错误有以下4种。

- (1) "分类训练"加工：只有输入没有输出，产生数据黑洞。
- (2) "分类处理"加工：有输出没有输入，无中生有。
- (3) "规则文件"数据流：外部实体没有经过加工处理，直接到数据存储。
- (4) "配置信息"数据流：外部实体之间没有加工处理，存在直接数据流。

针对图13-1的1层数据流图绘制出情报分类子系统的0层数据流图如图13-4所示。



图13-4 情报分类子系统的0层数据流图

【问题3】

高质量数据流图设计时应考虑的三个原则如下。

- (1) 复杂性最小化原则。数据流图分层结构就是把信息划分为小的且相对独立的一大堆子集例子，这样就可以单独考查每一个数据流图。如果要了解某个过程更加详的信息，可以跳转到该过程的下一层；如果要知道一个数据流图如何与其他数据流图相关联，可以跳转到上一层的数据流图进行考查。

(2) 接口最小化原则。接口最小化是复杂性最小化的一种具体规则。在设计模式时，应使得模型中各个元素之间的接口数或连接数最小化。

(3) 数据流一致性原则。一个过程和它的过程分解在数据流内容中是否有差别？是否存在有数据流出但没有相应的数据流入的加工？是否存在有数据流入但没有相应的数据流出的加工？

第13章：案例分析

试题3（2009年11月试题3）

试题3（2009年11月试题3）

阅读以下关于嵌入式软件体系架构的叙述，在答题纸上回答问题1至问题3。

希赛公司承担了一项宇航嵌入式设备的研制任务。本项目除对硬件设备环境有很高的要求外，还要求支持以下功能：

- ① 设备由多个处理机模块组成，需要时外场可快速更换（即LRM结构）；
- ② 应用软件应与硬件无关，便于软硬件的升级；
- ③ 由于宇航嵌入式设备中要支持不同功能，系统应支持完成不同功能任务间的数据隔离；
- ④ 宇航设备可靠性要求高，系统要有故障处理能力。

公司在接到此项任务后，进行了反复论证，提出三层栈（TLS）软件总体架构，如图13-5所示，并将软件设计工作交给了李工，要求其在三周内完成软件总体设计工作，给出总体设计方案。



图13-5 宇航嵌入式设备软件架构

【问题1】

用150字以内的文字，说明公司制定的TLS软件架构的层次特点，并针对上述功能需求①~④，说明架构中各层内涵。

【问题2】

在TLS软件架构的基础上，关于选择哪种类型的嵌入式操作系统问题，李工与总工程师发生了严重分歧。李工认为，宇航系统是实时系统，操作系统的处理时间越快越好，隔离意味着以时间作代价，没有必要，建议选择类似于VxWorks5.5的操作系统；总工程师认为，应用软件间隔离是宇航系统安全性要求，宇航系统在选择操作系统时必须考虑这一点，建议选择类似于Linux的操作系统。

请说明两种操作系统的主要差异，完成表13-2中的空白部分，并针对本任务要求，用200字以内的文字说明你选择操作系统的类型和理由。

表13-2 两种操作系统的主要差异

比较类型	VxWorks5.5	Linux
工作方式	操作系统与应用程序处于同一存储空间	①
多任务支持	支持多任务（线程）操作	②
实时性	③	实时系统
安全性	④	⑤
标准 API	支持	支持

【问题3】

故障处理是宇航系统软件设计中极为重要的组成部分。故障处理主要包括故障监视、故障定位、故障隔离和系统容错（重组）。用150字以内的文字说明嵌入式系统中故障主要分哪几类？并分别给出两种常用的故障滤波算法和容错算法。

试题分析

这是一道嵌入式系统架构试题，题目以宇航嵌入式设备的研制为背景。考查软件需求分析、软件架构、框架设计、可靠性设计以及嵌入式操作系统相关的一些知识。题目所述的项目中提出了一种新型的架构TLS,由于是项目中提出的架构，大家必然对这种架构感觉比较陌生，但从图中，我们可以很明显地看到，架构是分层的结构。这也就回到了我们熟悉的技术框架之内了。

【问题1】

问题1要求说明“公司制定的TLS软件架构的层次特点”，这需要从层次型架构的特点，以及题目已给出的说明两方面着手进行说明。层次型架构总的来说，特点有：下层为上层提供服务，层次与层次之间有着良好的独立性，结合题目的说明，可以总结出以下的特点。

- ① 应用软件仅与操作系统服务相关，不直接操作硬件。
- ② 操作系统通过模块支持原软件访问硬件，可与具体硬件无关。
- ③ 模块支持层将硬件抽象成标准操作。
- ④ 通过三层栈的划分可实现硬件的快速更改与升级，应用软件的升级不会引起硬件的变更。

基于这种思想，TLS结构框架实现了宇航设备中软件功能服务，这样考生就可以完整地叙述各层软件的设计内涵如下。

- ① 应用层主要完成宇航设备的具体工作，由多个功能任务组成，各功能任务间的隔离由操作系统层实现。
- ② 操作系统层实现应用软件与硬件的隔离，为应用软件提供更加丰富的计算机资源服务。操作系统为应用软件提供标准的API接口（如POSIX），确保了应用软件的可升级性。
- ③ 模块支持层为操作系统管理硬件资源提供统一管理方法，用一种抽象的标准接口实现软件与硬件的无关性，达到硬件的升级要求，便于硬件的外场快速更换。

【问题2】

问题2考查嵌入式操作系统的相关知识。简单地说，实时系统可以看成对外部事件能够及时响应的系统。这种系统最重要的特征是时间性，也就是实时性，实时系统的正确性不仅依赖于系统计算的逻辑结果，还依赖于产生这些结果的时间。在现实世界中，并非所有的嵌入式系统都具有实时特性，所有的实时系统也不一定都是嵌入式的。但这两种系统并不互相排斥，兼有这两种系统特性的系统称为实时嵌入式系统。

嵌入式操作系统（Embedded Operating System, EOS）是指运行在嵌入式系统上，支持嵌入式应用程序的操作系统，是用于控制和管理嵌入式系统中的硬件和软件资源、提供系统服务的软件集合。EOS是嵌入式软件的一个重要组成部分，它的出现提高了嵌入式软件开发的效率和应用软件的可移植性，有力地推动了嵌入式系统的发展。

1. 嵌入式操作系统的特点

与通用操作系统相比，嵌入式操作系统主要有以下特点：

(1) 微型化：嵌入式操作系统的运行平台不是通用计算机，而是嵌入式计算机系统。这类系统一般没有大容量的内存，几乎没有外存，因此，嵌入式操作系统必须做得小巧，以尽量少占用系统资源。为了提高系统地执行速度和系统的可靠性，嵌入式系统中的软件一般都固化在存储器芯片中，而不是存放在磁盘等载体中。

(2) 代码质量高：在大多数应用中，存储空间依然是宝贵的资源，这就要求程序代码的质量要高，代码要尽量精简。

(3) 专业化：嵌入式系统的硬件平台多种多样，处理器的更新速度快，每种都是针对不同的应用领域而专门设计。因此，嵌入式操作系统要有很好适应性和移植性，还要支持多种开发平台。

(4) 实时性强：嵌入式系统广泛应用于过程控制、数据采集、通信、多媒体信息处理等要求实时响应的场合，因此实时性成为嵌入式操作系统的又一特点。

(5) 可裁减、可配置：应用的多样性要求嵌入式操作系统具有较强的适应能力，能够根据应用的特点和具体要求进行灵活配置和合理裁减，以适应微型化和专业化的要求。

2. 嵌入式操作系统的分类

嵌入式操作系统的种类繁多，可以从不同角度对其进行分类。

从嵌入式操作系统的获得形式上，可以分为商业型和免费型两类：

(1) 商业型。商业型嵌入式操作系统一般功能稳定、可靠，有完善的技术支持、齐全的开发工具和售后服务。如WindRiver公司的VxWorks、pSOS和Palm公司的Palm OS等。但是，价格昂贵，用户通常得不到系统的源代码。

(2) 免费型。免费型嵌入式操作系统的优点在于价格方面，另外，应用系统开发者可以获得系统源代码，给开发带来了方便。但免费型的操作系统功能简单、技术支持差、系统的稳定性也不够好。典型代表系统有嵌入式Linux、uC/OS等。

从嵌入式操作系统的实时性上，可以分为实时嵌入式操作系统和非实时嵌入式操作系统两类。

(1) 实时嵌入式操作系统 (Real-Time Embedded OS, RTEOS)。实时嵌入式操作系统支持实时系统工作，其首要任务是调度一切可利用资源，以满足对外部事件响应的实时时限，其次着眼于提高系统的使用效率。实时嵌入式操作系统主要用在控制、通信等领域。目前，大多数商业嵌入式操作系统都是实时操作系统。

(2) 非实时嵌入式操作系统。这类操作系统不特别关注单个任务响应时限，其平均性能、系统效率和资源利用率一般较高，适合于实时性要求不严格的消费类电子产品，如个人数字助理、机顶盒等。

3. 如何选择嵌入式操作系统

在选择嵌入式操作系统时，也需要做多方面的考虑：

(1) 嵌入式操作系统的功能。根据项目需要的嵌入式操作系统功能来选择嵌入式操作系统产品，要考虑系统支持嵌入式操作系统的全部功能还是部分功能，是否支持文件系统和人机界面，是实时系统还是分时系统，以及系统是否可裁减等因素。

(2) 配套开发工具。有些实时操作系统只支持该系统供应商的开发工具。也就是说，还必须向嵌入式操作系统供应商获取编译器和调试器等；有些嵌入式操作系统使用广泛，且有第三方工具可用，因此，选择的余地比较大。

(3) 嵌入式操作系统的可移植性。嵌入式操作系统到硬件的移植是一个重要的问题，是整个系统能否按期完工的关键因素，因此，要选择那些可移植性程度高的嵌入式操作系统，从而避免嵌入式操作系统难以向硬件移植而带来的种种困难，加速系统的开发进度。

(4) 嵌入式操作系统的内存需求。均衡考虑是否需要额外RAM或EEPROM来迎合嵌入式操作系统对内存的较大要求。有些嵌入式操作系统对内存的要求是目标相关的，例如，Tornado/VxWorks等，开发人员能按照应用需求分配所需的资源，而不是为嵌入式操作系统分配资源。

(5) 嵌入式操作系统附加软件包。嵌入式操作系统是否包含所需的软件部件，例如，网络协议栈、文件系统和各种常用外设的驱动等。

(6) 嵌入式操作系统的实时性如何。有些嵌入式操作系统只能提供软实时性能，对于需要达到硬实时性能要求的系统就不适用；有些嵌入式操作系统即可满足软实时要求，也能满足硬实时要求，例如，MS Windows CE 2.0等。

(7) 嵌入式操作系统的灵活性。嵌入式操作系统是否具有可剪裁性，即能否根据实际需要进行系统功能的剪裁。有些嵌入式操作系统具有较强的可剪裁性，例如，嵌入式Linux和ECos等。

4. VxWorks和Linux的比较

VxWorks和Linux两个常见嵌入式操作系统，其特点如表13-3所示。

表13-3 两种操作系统的的主要差异

比较类型	VxWorks5.5	Linux
工作方式	操作系统与应用程序处于同一存储空间	①操作系统与应用程序处于不同存储空间
多任务支持	支持多任务（线程）操作	②支持多进程、多线程操作
实时性	③硬实时系统	实时系统
安全性	④任务间无隔离保护	⑤支持进程间隔离保护
标准 API	支持	支持

通过比较，显然选用类似于Linux的嵌入式操作系统适用本题要求。理由包括以下三点。

- ① Linux操作系统是一种安全性较强的操作系统。内核工作在系统态，应用软件工作在用户态（这点是系统安全性要求），可以有效防止应用软件对操作系统的破坏。
- ② Linux操作系统调度的最小单位是线程，线程归属于进程，进程具有自己独立的资源。进程通过存储器管理部件（MMU）实现多功能应用间隔离。
- ③ 嵌入式Linux操作系统支持硬件抽象，可有效地实现TLS结构，并将硬件抽象与操作系统分离，可方便实现硬件的外场快速更换。

【问题3】

作为宇航系统的嵌入式设备，硬/软件故障是宇航系统最为关注的内容，根据宇航系统的特点和系统组成，故障一般分为三类，即硬件故障、应用软件故障和操作系统故障，在设计中，应考虑这三类故障的处理方法。硬件故障一般包括CPU运算错误、存储器访问/越界错误、MMU配置错误、定时器计数错误和内总线错误等；应用软件故障一般包括计算越界、除0、溢出和超时等各种异常情况；操作系统故障一般包括越权访问、死锁、资源枯竭、调度超时、配置越界和操作系统异常等。

一般情况下，宇航系统的故障分为瞬态故障和永久故障，瞬态故障是指偶然发生的错误，而永久故障是指发生后不可消失的错误。在容错系统中，故障一般是由瞬态故障向永久故障转变，将鉴别瞬态故障和永久故障的算法称为滤波算法。在嵌入式实时系统中，常用的滤波算法包括门限算法、递减算法、递增算法和周期滤波算法等。

故障的鉴别目的是实现宇航系统的容错与重构。常用的容错算法是N+1备份，冷备、温备和热备。N+1备份是指N个通用模块之一的任何一个模块发生故障后，将故障模块的任务迁移到备份模块运行；冷备、温备和热备是三种备份方式，可根据宇航系统的总体备份时间或重要程度安排不同的备份算法。

试题答案

【问题1】

TLS结构框架的主要特点如下。

- ① 应用软件仅与操作系统服务相关，不直接操作硬件。
- ② 操作系统通过模块支持原软件访问硬件，可与具体硬件无关。
- ③ 模块支持层将硬件抽象成标准操作。
- ④ 通过三层栈的划分可实现硬件的快速更改与升级，应用软件的升级不会引起硬件的变更。

TLS结构框架的各层内涵如下。

- ① 应用层主要完成宇航设备的具体工作，由多个功能任务组成，各功能任务间的隔离由操作系统层实现。
- ② 操作系统层实现应用软件与硬件的隔离，为应用软件提供更加丰富的计算机资源服务。操作系统为应用软件提供标准的API接口（如POSIX），确保了应用软件的可升级性。
- ③ 模块支持层为操作系统管理硬件资源提供统一管理方法，用一种抽象的标准接口实现软件与硬件的无关性，达到硬件的升级要求，便于硬件的外场快速更换。

【问题2】

两种操作系统的差异如表13-4所示。

表13-4 两种操作系统的主要差异

比较类型	VxWorks5.5	Linux
工作方式		①操作系统与应用程序处于不同存储空间
多任务支持		②支持多进程、多线程操作
实时性	③硬实时系统	
安全性	④任务间无隔离保护	⑤支持进程间隔离保护
标准 API		

选择类似于Linux的嵌入式操作系统，理由如下。

- ① Linux操作系统是一种安全性较强的操作系统。内核工作在系统态，应用软件工作在用户态，可以有效地防止应用软件对操作系统的破坏。
- ② Linux操作系统调度的最小单位是线程，线程归属于进程，进程具有自己独立的资源。进程通过存储器管理部件（MMU）实现多功能应用间隔离。
- ③ 嵌入式Linux操作系统支持硬件抽象，可有效地实现TLS结构，并将硬件抽象与操作系统分离，可方便实现硬件的外场快速更换。

【问题3】

- ① 嵌入式系统中故障分为如下几种。

硬件故障：如CPU、存储器和定时器等。

应用软件故障：如数值越界、异常和超时等。

操作系统故障：如越权访问、死锁和资源枯竭等。

- ② 滤波算法有如下几种。

门限算法。

递减算法。

递增算法。

周期滤波算法。

- ③ 容错算法有如下几种。

N+1备份。

冷备。

温备。

热备。

试题4（2009年11月试题4）

试题4（2009年11月试题4）

阅读以下软件系统架构选择的问题，在答题纸上回答问题1至问题3。

希赛公司欲开发一个车辆定速巡航控制系统，以确保车辆在不断变化的地形中以固定的速度行驶。图13-6给出了该系统的简化示意图。表13-5描述了各种系统输入的说明。

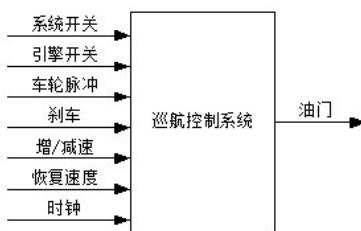


图13-6 定速巡航控制系统简化示意图

表13-5 定速巡航控制系统输入说明

输入名称	作用
系统开关	开启/关闭巡航控制系统
引擎开关	开启/关闭汽车引擎（引擎开启时，巡航控制系统处于就绪状态）
车轮脉冲	车轮每转一次，相应地发出一次脉冲
刹车	当刹车被踩下时，定速巡航控制系统会临时恢复到人工控制
增/减速	增加或减慢当前车速（仅在定速巡航控制系统处于开启的状态下可用）
恢复速度	恢复原来保持的车速（仅在定速巡航控制系统处于开启的状态下可用）
时钟	每毫秒定时脉冲

公司的领域专家对需求进行深入分析后，将系统需求认定为：任何时刻，只要定速巡航控制系统处于工作状态，就要有确定的期望速度，并通过调整引擎油门的设定值来维持期望速度。

在对车辆定速巡航控制系统的架构进行设计时，公司的架构师王工提出采用面向对象的架构风格，而李工则主张采用控制环路的架构风格。在架构评估会议上，专家对这两种方案进行综合评价，最终采用了面向对象和控制环路相结合的混合架构风格。

【问题1】

在实际的软件项目开发中，采用成熟的架构风格是项目成功的保证。请用200字以内的文字说明：什么是软件架构风格；面向对象和控制环路两种架构风格各自的特点。

【问题2】

用户需求没有明确给出该系统如何根据输入集合计算输出。请用300字以内的文字针对该系统的增减速功能，分别给出两种架构风格中的主要构件，并详细描述计算过程。

【问题3】

实际的软件系统架构通常是多种架构风格的混合，不同的架构风格都有其适合的应用场景。以该系统为例，针对面向对象架构风格和控制环路架构风格，各给出两个适合的应用场景，并简要说明理由。

试题分析

软件系统架构风格多种多样，如何根据系统的要求选择合适的架构风格，是系统架构设计师需要掌握的重要技能。

【问题1】

问题1考查概念层面的问题，根据美国卡耐基·梅隆大学软件工程研究所的定义，软件架构风格是描述某一类特定应用领域中软件系统组织方式和惯用方式。

面向对象架构风格的特征是将数据表示和基本操作封装在对象中。这种模式的构件是对象，对象维护自身表示的完整性，对象之间通过消息机制进行通信，对象交互时需要知道彼此的标识，通过对象之间的协作完成计算过程。

控制环路架构风格是将过程输出的指定属性维护在一个特定的参考值（设定点）。控制环路风格包括过程变量、被控变量、输入变量、操纵变量和设定点等构件，通过收集实际和理想的过程状态信息，并能调整过程变量使得实际状态趋于理想状态。

【问题2】

问题2难度比问题1要大一些。需要根据面向对象的风格特点与控制环路的风格特点，来分析系统功能应如何拆分成构件。

采用面向对象风格的巡航控制系统首先会定义司机、油门、时钟、速度计和车轮等对象。整个计算过程通过对对象之间的合理交互完成期望的功能。

控制环路的架构风格将以控制器为核心，期望速度、车轮脉冲、时钟和油门等作为构件。通过不断地反馈机制实现对系统的控制。

【问题3】

问题3主要从应用场景的角度比较两种架构风格。面向对象系统比较适合事件驱动的场景，特别是离散突发事件的处理；而控制环路则适合连续事件的处理，比如维持恒定车速等。

试题答案

【问题1】

软件架构风格是描述某一类特定应用领域中软件系统的组织方式和惯用方式。

面向对象架构风格的特征是将数据表示和基本操作封装在对象中。这种模式的构件是对象，对象维护自身表示的完整性，对象之间通过消息机制进行通信，对象交互时需要知道彼此的标识，通过对象之间的协作完成计算过程。

控制环路架构风格是将过程输出的指定属性维护在一个特定的参考值（设定点）。控制环路风格包括过程变量、被控变量、输入变量、操纵变量和设定点等构件，通过收集实际和理想的过程状态信息，并能调整过程变量使得实际状态趋于理想状态。

【问题2】

对于系统的增减速功能，采用面向对象风格的巡航控制系统首先会定义司机、油门、时钟、速度计和车轮等构件。

整个计算的主要过程如下。

- ① 司机进行增/减速操作设置期望速度，该期望速度以消息的形式传递给速度计。
- ② 速度计通过向车轮和时钟发送消息获取车轮转速和时钟值，得到当前速度。
- ③ 速度计计算当前速度和期望速度的速度差值。
- ④ 该差值以消息的形式发送给油门，油门通过速度差值调节自身状态。
- ⑤ 整个过程在时钟的控制下定期向速度计发送消息，重复执行②~④。

控制环路的架构风格以控制器为核心，期望速度、车轮脉冲、时钟和油门等作为构件。

具体的计算过程如下。

- ① 司机进行增/减速操作设置期望速度值。
- ② 将设定值置为期望速度值。
- ③ 控制器采集车轮脉冲和时钟值，计算出当前速度。
- ④ 比较期望速度和当前速度，计算速度差值，控制油门动作。
- ⑤ 反复执行③和④。

【问题3】

适合面向对象架构风格的应用场景如下。

① 用户刹车，立即退出巡航控制系统。理由：这是一个典型的事件驱动的场景，适合于面向对象风格。

② 系统对突发事件的处理，如某些部件失灵等。理由：当发生突发事件时，系统会同时产生数据和事件，这种情况用对象建模较为恰当。

适合面向控制环路架构风格的应用场景如下。

① 在达到期望速度后，系统维持恒定速度行驶。理由：这是一个典型的闭环控制的情景，系统需要在外界情况不断发生变化的情况下进行调整，使得系统状态尽可能接近期望状态。

② 用户改变期望速度后，系统不断地进行调整，直至到达恒定速度。理由：这是一个闭环控制情景，当用户设定期望速度值后，系统需要在不断获取当前速度和外界条件的情况下对系统状态持续调整，使得系统状态尽可能地接近这个新的期望状态。

第 13 章：案例分析

试题5（2009年11月试题5）

试题5（2009年11月试题5）

阅读以下关于信息系统安全性的叙述，在答题纸上回答问题1至问题3。

某企业根据业务扩张的要求，需要将原有的业务系统扩展到互联网上，建立自己的B2C业务系统，此时系统的安全性成为一个非常重要的设计需求。为此，该企业向软件开发商提出如下要求。

- ① 合法用户可以安全地使用该系统完成业务。
- ② 灵活的用户权限管理。
- ③ 保护系统数据的安全，不会发生信息泄露和数据损坏。
- ④ 防止来自于互联网上各种恶意攻击。
- ⑤ 业务系统涉及到各种订单和资金的管理，需要防止授权侵犯。
- ⑥ 业务系统直接面向最终用户，需要在系统中保留用户使用痕迹，以应对可能的商业诉讼。

该软件开发商接受任务后，成立方案设计小组，提出的设计方案是：在原有业务系统的基础上，保留了原业务系统中的认证和访问控制模块；为了防止来自互联网的威胁，增加了防火墙和入侵检测系统。

企业和软件开发商共同组成方案评审会，对该方案进行了评审，各位专家对该方案提出了多点不同意见。李工认为，原业务系统只针对企业内部员工，采用了用户名/密码方式是可以的，但扩展为基于互联网的B2C业务系统后，认证方式过于简单，很可能造成用户身份被盗取：王工认为，防

止授权侵犯和保留用户痕迹的要求在方案中没有体现。而刘工则认为，即使是在原有业务系统上的扩展与改造，也必须全面考虑信息系统面临的各种威胁，设计完整的系统安全架构，而不是修修补补。

【问题1】

信息系统面临的安全威胁多种多样，来自多个方面。请指出信息系统面临哪些方面的安全威胁并分别予以简要描述。

【问题2】

认证是安全系统中不可缺少的环节，请简要描述主要的认证方式，并说明该企业应采用哪种认证方式。

【问题3】

请解释授权侵犯的具体含义；针对王工的意见给出相应的解决方案，说明该解决方案的名称、内容和目标。

试题分析

这是一道信息系统安全的试题。对于任何一个系统而言，安全性都是至关重要的，可以说一个系统的安全性如果得不到保障，那么该系统的功能越强大，造成的危害也就越大。

【问题1】

网络威胁是指对网络安全缺陷的潜在利用，这些缺陷可能导致非授权访问、信息泄露、资源耗尽、资源被盗或者被破坏等。网络安全威胁的种类有：窃听、假冒、重放、流量分析、数据完整性破坏、拒绝服务、资源的非授权使用、陷门和特洛伊木马、病毒、诽谤等。

对于信息系统来说，威胁可以是针对物理环境、通信链路、网络系统、操作系统、应用系统以及管理系统等方面。物理安全威胁，是指对系统所用设备的威胁，自然灾害、电源故障、操作系统引导失败或数据库信息丢失、设备被盗/被毁造成数据丢失或信息泄露。通信链路安全威胁，是指在传输线路上安装窃听装置或对通信链路进行干扰。网络安全威胁，互联网的开放性、国际性与无安全管理性，对内部网络形成严重的安全威胁。操作系统安全威胁，对系统平台最危险的威胁是在系统软件或硬件芯片中植入威胁，如“木马”和“陷阱门”、BIOS有万能密码。应用系统安全威胁，是指对于网络服务或用户业务系统安全的威胁，也受到“木马”和“陷阱门”的威胁。管理系统安全威胁，必须从人员管理上杜绝安全漏洞。

具体来讲，常见的安全威胁有：

- (1) 信息泄露：信息被泄露或透露给某个非授权的实体。
- (2) 破坏信息的完整性：数据被非授权地进行增删、修改或破坏而受到损失。
- (3) 拒绝服务：对信息或其它资源的合法访问被无条件地阻止。
- (4) 非法使用（非授权访问）：某一资源被某个非授权的人、或以非授权的方式使用。
- (5) 窃听：用各种可能的合法或非法的手段窃取系统中的信息资源和敏感信息。例如对通信线路中传输的信号进行搭线监听，或者利用通信设备在工作过程中产生的电磁泄露截取有用信息等。
- (6) 业务流分析：通过对系统进行长期监听，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等参数进行研究，从而发现有价值的信息和规律。
- (7) 假冒：通过欺骗通信系统（或用户）达到非法用户冒充成为合法用户，或者特权小的用户冒充成为特权大的用户的目的。黑客大多是采用假冒进行攻击。
- (8) 旁路控制：攻击者利用系统的安全缺陷或安全性上的脆弱之处获得非授权的权利或特权。例如，攻击者通过各种攻击手段发现原本应保密，但是却又暴露出来的一些系统“特性”。利用这些“特性”，攻击者可以绕过防线守卫者侵入系统的内部。

(9) 授权侵犯 : 被授权以某一目的使用某一系统或资源的某个人 , 却将此权限用于其它非授权的目的 , 也称作 " 内部攻击 ".

(10) 特洛伊木马 : 软件中含有一个察觉不出的或者无害的程序段 , 当它被执行时 , 会破坏用户的安全。这种应用程序称为特洛伊木马 (Trojan Horse).

(11) 陷阱门 : 在某个系统或某个部件中设置了 " 机关 " , 使得当提供特定的输入数据时 , 允许违反安全策略。

(12) 抵赖 : 这是一种来自用户的攻击 , 比如 : 否认自己曾经发布过的某条消息、伪造一份对方来信等。

(13) 重放 : 所截获的某次合法的通信数据拷贝 , 出于非法的目的而被重新发送。

(14) 计算机病毒 : 所谓计算机病毒 , 是一种在计算机系统运行过程中能够实现传染和侵害的功能程序。一种病毒通常含有两个功能 : 一种功能是对其它程序产生 " 感染 "; 另外一种或者是引发损坏功能、或者是一种植入攻击的能力。

(15) 人员不慎 : 一个授权的人为了钱或利益、或由于粗心 , 将信息泄露给一个非授权的人。

(16) 媒体废弃 : 信息被从废弃的磁的或打印过的存储介质中获得。

(17) 物理侵入 : 侵入者通过绕过物理控制而获得对系统的访问 ;

(18) 窃取 : 重要的安全物品 , 如令牌或身份卡被盗 ;

(19) 业务欺骗 : 某一伪系统或系统部件欺骗合法的用户或系统自愿地放弃敏感信息。

通过对网络面临的安全风险威胁和实施相应控制措施的支出进行合理的评价 , 提出有效合理的安全技术 , 形成提升网络信息的安全性质的安全方案 , 是安全架构设计的根本目标。在实际应用中 , 可以从安全技术的角度提取出五个方面的内容 : 认证鉴别、访问控制、内容安全、冗余恢复和审计响应。

【问题2】

用户的身份认证是许多应用系统的第一道防线 , 其目的在于识别用户的合法性 , 从而阻止非法用户访问系统。身份识别对确保系统和数据的安全保密是极其重要的 , 目前 , 计算机网络系统中常用的身份认证方式主要有以下几种 :

1、口令认证

用户名/密码是最简单也是最常用的身份认证方法 , 密码是由用户自己设定的 , 只有用户自己才知道。只要能够正确输入密码 , 计算机就认为操作者就是合法用户。实际上 , 由于许多用户为了防止忘记密码 , 经常采用诸如生日和电话号码等容易被猜测的字符串作为密码 , 或者将密码抄在纸上 , 放在一个自认为安全的地方。这样 , 很容易造成密码泄漏 , 即使能保证用户密码不被泄漏 , 由于密码是静态的数据 , 在验证过程中需要在网络中传输 , 而每次验证使用的信息都是相同的 , 很容易被驻留在内存中的木马程序或网络中的监听设备截获。因此 , 从安全性上讲 , 用户名/密码方式是一种极不安全的身份认证方式。

2、令牌认证

在使用令牌进行认证的系统中 , 进行验证的密钥存储于令牌中。对密钥的访问用口令进行控制。令牌是一个像IC卡一样可以加密存储并运行相应加密算法的设备 , 这种简单认证可以快速、方便地实现用户身份认证 , 但是认证的安全强度不高。通过令牌可以完成对用户必须拥有某物的验证。令牌的实现分为质询响应令牌和时间戳令牌。其中使用较多的是时间戳令牌。

质询响应令牌的工作原理是 : 在进行身份认证时 , 认证服务器首先发送一个随机数到客户机的登录程序。用户将这个随机数读出 , 输入令牌 , 并输入令牌的PIN码 (实际就是口令) , 得以访问令牌。令牌对输入的随机数用存储的私钥进行签名 , 并把结果用Base64编码输出。用户把令牌的输出

填入客户机的验证程序中，数据传输到认证的服务器端，在服务器端将使用用户的公钥对签名进行验证，以确定是否允许客户通过登录认证。在该方案中，由于使用数字签名进行登录认证，系统的安全强度大大增加：私钥采用令牌存储的方式解决了私钥自身的安全问题。令牌是一个可移动的设备，可以随身携带，而且令牌有PIN码保护，对令牌的非法访问超过一定的次数后，令牌会死锁。

时间戳令牌解决了质询响应令牌中随机数的问题，时间戳令牌利用时间代替上面的随机数。时间戳令牌每时每刻都在工作，一般每分钟产生一个登录数据，用户只需输入PIN码。登录数据被传送到认证的服务器端，服务器利用当前时间对登录数据进行验证，完成用户的登录过程。使用时间戳令牌需要重点考虑时间同步问题，由于令牌的时钟和认证服务器的时钟不同步，产生的验证码并不会通过验证。解决方法是在验证服务器上进行多次试探验证，在一个时间范围内试探，如果成功则在服务器上存储令牌时钟与服务器时钟的偏移量，以便下次登录时使用。目前，在安全性要求较高的认证系统中，多是采用这种方案。

采用PIN码与令牌实现了双因素验证，根据用户知道什么、拥有什么进行认证，也提供了一个保密认证密钥的方法。但是实现双因素验证需要用户输入数据，给用户的操作增加了麻烦。

3、生物识别认证

生物特征识别是通过可测量的身体或行为等生物特征进行身份认证的一种技术。生物特征分为身体特征和行为特征两类，身体特征包括指纹、掌型、视网膜、虹膜、人体气味、脸型、手的血管和DNA (NeoxyriboNucleic Acid, 脱氧核糖核酸) 等；行为特征包括签名、语音和行走步态等。目前，部分学者将视网膜识别、虹膜识别和指纹识别等归为高级生物识别技术，将掌型识别、脸型识别、语音识别和签名识别等归为次级生物识别技术，将血管纹理识别、人体气味识别、DNA识别等归为深奥的生物识别技术。

4、三因素认证

基于用户知道什么（口令）、拥有什么（私钥和令牌）、是什么（生物特征）的3因素认证是目前强认证中使用最多的手段。在安全性要求较高的系统中，认证必须能对用户进行身份鉴定。要将用户知道什么、拥有什么、是什么结合起来，同时对认证用的密钥进行保护

【问题3】

问题3属于技术应用型的题。给出了一些现象，让考生分析原因，分析缺陷。从描述来看，我们可以很明显得知系统缺乏安全审计的策略。而在安全体系中，审计占有非常重要的地位，安全审计系统可以帮助发现系统入侵和漏洞、帮助发现系统性能上的不足、为一些安全案件提供有效的追击证据。同时缺乏抗抵赖的机制。

试题答案

【问题1】

信息系统面临的安全威胁来自于物理环境、通信链路、网络系统、操作系统、应用系统以及管理等多个方面。

物理安全威胁是指对系统所用设备的威胁，如自然灾害、电源故障、数据库故障和设备被盗等造成数据丢失或信息泄露。

通信链路安全威胁是指在传输线路上安装窃听装置或对通信链路进行干扰。

网络安全威胁当前主要是指由于因特网的开放性、国际性与无安全管理性，对内部网络形成的严重安全威胁。

操作系统安全威胁指的是操作系统本身的后门或安全缺陷，如"木马"和"陷阱门"等。

应用系统安全威胁是指对于网络服务或用户业务系统安全的威胁，包括应用系统自身漏洞，也受到"木马"的威胁。

管理系统安全威胁指的是人员管理和各种安全管理制度。

【问题2】

目前主要的认证方式有以下三类。

① 用户名和口令认证：主要是通过一个客户端与服务器共知的口令（或与口令相关的数据）进行验证。根据处理形式的不同，分为验证数据的明文传送、利用单向散列函数处理验证数据、利用单向散列函数和随机数处理验证数据。

② 使用令牌认证：该方式中，进行验证的密钥存储于令牌中，目前的令牌包括安全证书和智能卡等方式。

③ 生物识别认证：主要是根据认证者的图像、指纹、气味和声音等作为认证数据。根据该企业的业务特征，采用令牌认证较为合适。

【问题3】

授权侵犯指的是被授权以某一目的使用某一系统或资源的某个人，将此权限用于其他非授权的目的，也称作“内部攻击”。

针对王工的建议，从系统安全架构设计的角度需要提供抗抵赖框架。

抗抵赖服务包括证据的生成、验证和记录，以及在解决纠纷时随即进行的证据恢复和再次验证。

框架中抗抵赖服务的目的是提供有关特定事件或行为的证据。例如，必须确认数据原发者和接收者的身份和数据完整性，在某些情况下，可能需要涉及上下文关系（如日期、时间、原发者/接收者的地点等）的证据，等等。

第 13 章：案例分析

试题6（2010年11月试题1）

试题6（2010年11月试题1）

阅读以下关于软件系统架构选择的说明，在答题纸上回答问题1至问题3。

希赛公司欲针对Linux操作系统开发一个KWIC（Key Word in Context）检索系统。该系统接收用户输入的查询关键字，依据字母顺序给出相关帮助文档并根据帮助内容进行循环滚动阅读。在对KWIC系统进行需求分析时，公司的业务专家发现用户后续还有可能采用其他方式展示帮助内容。根据目前需求，公司的技术人员决定通过重复剪切帮助文档中的第一个单词并将其插入到行尾的方式实现帮助文档内容的循环滚动，后续还将采用其他的方法实现这一功能。

在对KWIC系统的架构进行设计时，公司的架构师王工提出采用共享数据的主程序-子程序的架构风格，而李工则主张采用管道-过滤器的架构风格。在架构评估会议上，大家从系统的算法变更、功能变更、数据表示变更和性能等方面对这两种方案进行评价，最终采用了李工的方案。

【问题1】

在实际的软件项目开发中，采用恰当的架构风格是项目成功的保证。请用200字以内的文字说明什么是软件架构风格，并对主程序-子程序和管道-过滤器这两种架构风格的特点进行描述。

【问题2】

请完成表13-6中的空白部分（用+表示优、-表示差），对王工和李工提出的架构风格进行评价，并指出来采用李工方案的原因。

表13-6 王工与李工的架构风格评价

架构风格 评价要素	共享数据的主程序-子程序	管道-过滤器
算法变更	-	(1)
功能变更	(2)	+
数据表示变更	(3)	(4)
性能	(5)	(6)

【问题3】

图13-7是李工给出的架构设计示意图，请将恰当的功能描述填入图中的(1)~(4)。

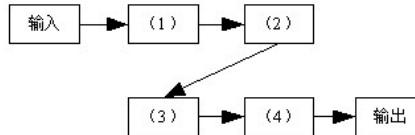


图13-7 李工给出的架构示意图

试题分析

本题是一道架构设计方面的试题，考查的内容是常见架构风格的选用。这就涉及不同架构风格的优势、劣势、应用场合的比较分析。

【问题1】

问题1考查架构风格的基本概念与主程序-子程序、管道-过滤器的特点。这一空属于送分题，难度较低。

1、软件架构风格的概念

软件架构设计的一个核心问题是能否使用重复的架构模式，即能否达到架构级的软件重用。也就是说，能否在不同的软件系统中，使用同一架构。基于这个目的，学者们开始研究和实践软件架构的风格和类型问题。

软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式 (idiomatic paradigm)。架构风格定义了一个系统“家族”，即一个架构定义、一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而约束指出系统是如何将这些构件和连接件组合起来的。架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统。按这种方式理解，软件架构风格定义了用于描述系统的术语表和一组指导构建系统的规则。

对软件架构风格的研究和实践促进了对设计的重用，一些经过实践证实的解决方案也可以可靠地用于解决新的问题。架构风格的不变部分使不同的系统可以共享同一个实现代码。只要系统是使用常用的方法来组织，就可使别的设计师很容易地理解系统的架构。例如，如果某人把系统描述为客户/服务器模式，则不必给出设计细节，相关人员立刻就会明白系统是如何组织和工作的。

2、主程序与子程序

Garlan和Shaw对通用软件架构风格进行了分类，他们将软件架构分为数据流风格、调用/返回风格、独立构件风格、虚拟机风格和仓库风格。题目中的主程序-子程序架构风格属于调用/返回风格，管道-过滤器架构风格属于数据流风格。

主程序-子程序是单线程控制，把问题划分为若干个处理步骤，构件即为主程序和子程序，子程序通常可合成为模块。过程调用作为交互机制，即充当连接件的角色。调用关系具有层次性，其语义逻辑表现为主程序的正确性取决于它调用的子程序的正确性。

3、管道-过滤器

在管道-过滤器风格的软件架构中，每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常通过对输入流的变换及增量计算来完成，所以在输入被完全消费之前，输出便产生了。因此，这里的构件被称为过滤器，这种风格的连接件就象是数据流传输的管道，将一个过滤器的输出传到另一过滤器的输入。此风格特别重要的过滤器必须是独立的实体，它不能与其它的过滤器共享数据，而且一个过滤器不知道它上游和下游的标识。一个管道/过滤器网络输出的正确性并不依赖于过滤器进行增量计算过程的顺序。

图13-8是管道与过滤器风格的示意图。一个典型的管道-过滤器架构的例子是以UNIX Shell编写程序。Unix既提供一种符号，以连接各组成部分（UNIX的进程），又提供某种进程运行时机制以实现管道。另一个著名的例子是传统的编译器。传统的编译器一直被认为是一种管道系统，在该系统中，一个阶段（包括词法分析、语法分析、语义分析和代码生成）的输出是另一个阶段的输入。

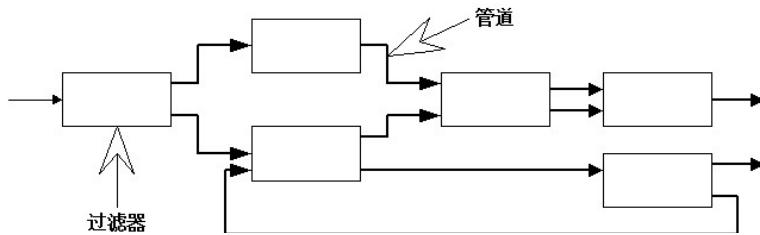


图13-8 管道与过滤器风格

管道-过滤器风格的软件架构具有许多很好的特点：

- (1) 使得软构件具有良好的隐蔽性和高内聚、低耦合的特点。
- (2) 允许设计师将整个系统的输入/输出行为看成是多个过滤器的行为的简单合成。
- (3) 支持软件重用。只要提供适合在两个过滤器之间传送的数据，任何两个过滤器都可被连接起来。
- (4) 系统维护和增强系统性能简单。新的过滤器可以添加到现有系统中来；旧的可以被改进的过滤器替换掉。
- (5) 允许对一些如吞吐量、死锁等属性的分析。
- (6) 支持并行执行。每个过滤器是作为一个单独的任务完成，因此可与其它任务并行执行。

但是，这样的系统也存在着若干不利因素。

- (1) 通常导致进程成为批处理的结构。这是因为虽然过滤器可增量式地处理数据，但它们是独立的，所以设计师必须将每个过滤器看成一个完整的从输入到输出的转换。
- (2) 不适合处理交互的应用。当需要增量地显示改变时，这个问题尤为严重。
- (3) 因为在数据传输上没有通用的标准，每个过滤器都增加了解析和合成数据的工作，这样就导致了系统性能下降，并增加了编写过滤器的复杂性。

【问题2】

问题2考查主程序-子程序和管道-过滤器优缺点对比。这两种风格的优缺点包括多个方向的很多内容，但要应对该题，并不需要我们面面俱到地把每一个细节记清楚。只要了解两者的核心思想即可。

本问题的核心是对两种架构风格在算法变更、功能变更、数据表示变更和性能等方面的特点进行比较。其中共享数据的主程序-子程序在算法变更方面灵活性较差，算法变更时一般需要重新编译整个系统；在功能变更方面也比较差；在数据表示方面，当需要变更时，也意味着程序传递参数的变化以及整个程序的调整，表现比较差；在性能方面，由于整个程序处在一个紧耦合的状态，因此性能较高。管道-过滤器架构风格在算法变更方面实现比较简单，只需要修改过滤器的实现即可；在

功能变更方面也比较简单；在数据表示方面，需要同时改变数据格式和过滤器的结构，相对比较复杂；在性能方面，由于整个系统是松耦合连接在一起的，因此性能不高。

根据题干描述：“用户后续还有可能采用其它方式展示帮助内容”，因此KWIC系统对功能变更要求较高；根据题干描述：“…，后续还将采用其它的方法实现这一功能”，因此KWIC系统对实现某一个功能的算法变更要求较高；KWIC是一个支持用户交互的窗口界面程序，因此对性能要求并不高；KWIC系统的显示帮助内容为文本，数据的表示基本不变，因此对数据表示变更要求不高。针对这些考虑，可以看出应该采用管道-过滤器的架构风格。

【问题3】

问题3是补充架构设计示意图。其实这个图要表现出来的，无非就是利用管道-过滤器架构，需要处理的信息的操作有哪些，按什么顺序排列。根据题干中的关键描述“…接收用户输入的查询关键字，依据字母顺序给出相关帮助文档并根据帮助内容进行循环滚动阅读…”和“…公司的技术人员决定通过重复剪切帮助文档中的第一个单词并将其插入到行尾的方式实现帮助文档内容的循环滚动…”，可以看出整个系统的流程是：输入→文档检索→查询结果按字母排序→剪切行首单词→单词插入行尾→输出。

试题答案

【问题1】

软件架构风格是描述特定软件系统组织方式的惯用模式。组织方式描述了系统的组成构件和这些构件的组织方式，惯用模式则反映众多系统共有的结构和语义。

主程序-子程序架构风格中，所有的计算构件作为子程序协作工作，并由一个主程序顺序地调用这些子程序，构件通过共享存储区交换数据。

管道-过滤器架构风格中，每个构件都有一组输入和输出，构件接受数据输入，经过内部处理，然后产生数据输出。这里的构件称为过滤器，构件之间的连接件称为数据流传输的管道。

【问题2】

表13-7 王工与李工的架构风格评价（完整表）

评价要素	架构风格	共享数据的主程序-子程序	管道-过滤器
算法变更	-	(1) +	
功能变更	(2) -	+	
数据表示变更	(3) -	(4) -	
性能	(5) +	(6) -	

根据题干描述：“用户后续还有可能采用其他方式展示帮助内容”，因此KWIC系统对功能变更要求较高。

根据题干描述：“…，后续还将采用其他的方法实现这一功能”，因此KWIC系统对实现某一个功能的算法变更要求较高。

KWIC是一个支持用户交互的窗口界面程序，因此对性能要求并不高。

KWIC系统显示的帮助内容为文本，数据的表示基本不变，因此对数据表示变更要求不高。

综合上述分析，可以看出应该采用李工提出的管道-过滤器架构风格。

【问题3】

完整的架构示意图如图13-9所示。

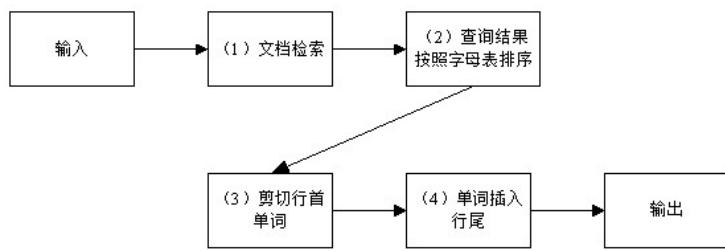


图13-9 李工给出的架构示意图(完整图)

第13章：案例分析

试题7 (2010年11月试题2)

试题7 (2010年11月试题2)

阅读以下关于软件系统数据架构的说明，在答题纸上回答问题1至问题3。

RMO是一家运动服装制造销售公司，计划在五年时间内将销售区域从华南地区扩展至全国范围。为了扩大信息技术对于未来业务发展的价值，公司邀请咨询顾问帮助他们制订战略信息系统规划。经过评审，咨询顾问给出的战略规划要点之一是建立客户关系支持系统CRSS。RMO公司决定由其技术部成立专门的项目组负责CRSS的开发和维护工作。

项目组在仔细调研和分析系统需求的基础上，确定了基于互联网的CRSS系统架构。但在确定系统数据架构时，张工认为应该采用集中式的数据架构，给出的理由是结构简单、易维护且开发及运行成本低；而刘工建议采用分布式的数据架构，并提出在开发中通过“局部数据库+缓存”的读/写分离结构实现，具有较好的运行性能和可扩展性。

项目组经过集体讨论，考虑到公司的未来发展规划，最终采用了刘工的建议。

【问题1】

请用300字以内的文字，说明张工和刘工提出的数据架构的基本思想。

【问题2】

在刘工建议的基础上，为了避免CRSS系统的单点故障，请用200字以内文字简要说明如何建立CRSS的数据库系统；对于数据的读取、添加、更改和删除操作分别如何实现。

【问题3】

RMO公司销售区域将在未来五年大面积扩展，其潜在客户数量也会因此大幅度增加，所以良好的可扩展性是CRSS系统所必需的质量属性。请分别说明在集中式和分布式数据架构下，可以采用哪些方法提升系统的可扩展性。

试题分析

本题主要考查考生对于软件系统数据架构的掌握情况。数据架构定义了信息系统中文件和数据库的分布结构。数据架构是以数据为中心，建模业务数据类型和结构，以及设计满足应用需求的数据库系统。传统以主机为中心的信息系统开发中，利用单个的数据库系统实现数据的集中式存储，物理上所有的数据位于同一个位置，构成的是一种集中式的数据架构；现代基于网络的分布式系统开发中，很少有组织会将其全部的数据存储在单个的数据库中，通常需要多个数据库系统

组成，数据在这些数据库系统之间可以传送，由多个不同的数据库管理系统控制，构成的是一种分布式的数据架构。

【问题1】

集中式数据架构中，一个或多个局域网中的客户共享一个单独计算机系统中的单个数据库。系统提供数据处理能力，用户可以在同样的站点上操作，也可以在地理位置隔开的其他站点上通过远程终端来操作。系统及其数据管理被某个或中心站点集中控制。单个数据库服务器结构的主要优点就是简单、易维护开发及运行成本低；但由于所有的客户直接请求服务器，容易发生性能瓶颈，如果服务失败，单个服务器不能提供备份和恢复，所有依赖的应用程序都将不能工作。

分布式数据架构中，使用多个计算机系统以及用户能够访问远程系统的数据，数据可以在多个不同的数据库中进行传送，由不同的数据库管理系统软件进行管理，运行在多种不同的计算机上，支持多种不同的操作系统。这些机器位于（或分布在）不同的地理位置并通过多种通信网络连接在一起。企业数据可以分布在不同的计算机上，一个应用程序可以操作位于不同地理位置的机器上的数据。多个数据库服务器结构的主要优点就是系统的容错能力和对广域网容量的需求有所降低，可以采用多种策略提升整个系统的服务质量；由于多个数据库系统分布在不同的网络节点上，位于不同位置的数据之间需要同步和协作，系统结构复杂、运行成本高并且维护困难。

在实际应用系统的数据架构构建中，应根据不同的应用需求选择集中式或分布式的数据架构。本题中由于RMO要扩展其销售区域，其潜在客户数量也会因此大幅度增加，所以良好的可扩展性是CRSS系统所必需的质量属性；并且由于其销售区域扩大后，系统中的数据会存储于不同的地理位置，所以采用分布式数据架构最为合理。

【问题2】

读写分离架构应用非常广泛，很多网站采用数据库+缓存的方式来实现。通过缓存层来承载大量的读访问，如广泛采用的Memcached，其自身往往不具备持久层存储的功能，通常和数据库一起组成分布式的数据架构，由数据库负责数据持久化存储和写入功能，缓存负责承载大量的并发访问，从而提高了系统的数据处理效率。要避免数据访问的单点故障，通常采用主数据库热备份的方式来实现。所以，要实现题目要求的分布式数据架构，需要多个局部数据库系统、多个热备份数据库系统和多个数据缓存组成。

读写分离结构中，应用读取数据时访问缓存，如果没有命中所需数据，则从主数据库中读取数据并写入缓存；对于新增、修改和删除操作，需要采用延迟加载的策略，新增时只修改主数据库，修改和删除时除了修改主数据库中的内容，还需要将缓存中的数据标记为失效。

【问题3】

传统的集中式数据架构中由于只有单个的数据库系统，所以要满足可扩展性的要求，更多的只能通过硬件的方式来实现。具体的实现方式包括硬件扩容（增加CPU/内存容量/磁盘数量）和硬件升级（更换高端主机或高速磁盘等）。

基于网络的分布式数据架构中由多个数据库系统共同组成，可以通过更改和优化数据分布来满足系统可扩展性的要求。具体的实现方式包括数据复制、数据垂直切分（或/和）水平切分、缓存和全文搜索。

试题答案

【问题1】

（1）张工提出的集中式数据架构是由一个处理器、与它相关联的数据存储设备以及其他外围设备组成，它被物理地定义到单个位置。系统提供数据处理能力，用户可以在同样的站点上操作，也

可以在地理位置隔开的其他站点上通过远程终端来操作。系统及其数据管理被某个中心站点集中控制。

(2) 刘工提出的分布式数据架构使用多个计算机系统上的多个局部数据库系统构成，数据可以在多个不同的局部数据库中进行传送，由不同的数据库管理系统软件进行管理，运行在多种不同的计算机上，支持多种不同的操作系统。这些机器位于（或分布在）不同的地理位置并通过多种通信网络连接在一起。企业数据可以分布在不同的计算机上，一个应用程序可以操作位于不同地理位置的机器上的数据。

【问题2】

读写分离架构利用了数据库的复制技术，将数据的读和写分布在不同的处理节点上，从而达到提高可用性和扩展性的目的。

通过采用局部数据库的双机热备可以解决单点故障的问题，CRSS的分布式数据库系统可由多个局部数据库系统、相应的热备份数据库系统和数据缓存组成。

局部数据库负责数据的写入，其热备份系统准备随时为局部数据库提供故障切换，数据缓存为应用提供所读取的数据。

(1) 读取数据：应用访问缓存，如果命中则返回，否则从局部数据库系统中读取数据并将数据加载到缓存后返回。

(2) 添加数据：采用延迟加载策略，应用将数据直接写入局部数据库。

(3) 更改数据：应用更改局部数据库中的数据，将缓存中的数据标记为失效。

(4) 删除数据：应用删除局部数据库中的数据，将缓存中的数据标记为失效。

【问题3】

张工提出的集中式数据架构通过向上扩展（Scale Up）提升系统的可扩展性。具体的实现方式包括硬件扩容（增加CPU数量、内存容量、磁盘数量）和硬件升级（更换为高端主机或高速磁盘等）。

刘工提出的分布式数据架构通过向外扩展（Scale Out）提升系统的可扩展性。具体的实现方式包括数据复制、数据垂直切分（或/和）水平切分、缓存和全文搜索。

第 13 章：案例分析

试题8（2010年11月试题3）

试题8（2010年11月试题3）

阅读以下关于汽车电子基础软件架构的说明，在答题纸上回答问题1至问题3。

某软件公司承担了某项国家重点项目的研制工作，任务重点是参考国外汽车电子发展趋势，开发某汽车企业的企业汽车电子基础软件的架构，逐步实现汽车企业未来的技术发展规划。

该软件公司接收此任务后，调动全体技术人员深入收集国外相关技术资料，经过多方调研和分析，公司提出遵照国际组织最新推出的AUTOSAR规范，按统筹规划、分步骤实施的原则，实现汽车企业的基础软件架构设计。图13-10给出了AUTOSAR规范所定义的工作包，图中灰色部分代表本项目工作所包含的内容，即软件架构和基础软件。

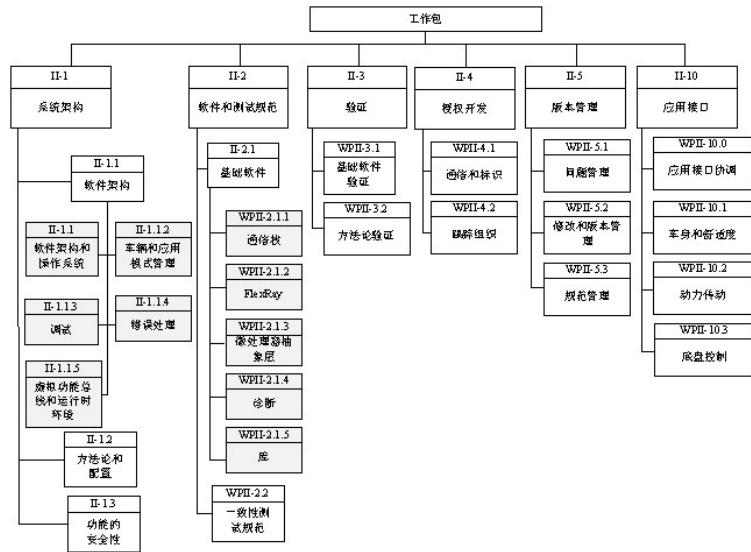


图13-10 AUTOSAR规范WBS分解图

【问题1】

AUTOSAR规范中要求，汽车电子软件开发流程应尽量满足并发、可多次迭代的特性。为了定义汽车电子的软件开发过程，公司李工和王工分别提出了两种软件开发流程，这两种开发流程见图13-11和图13-12（图中ECU是指汽车电子中的电子控制单元）。请说明李工和王工谁定义的流程更符合AUTOSAR的规定，并说明理由。

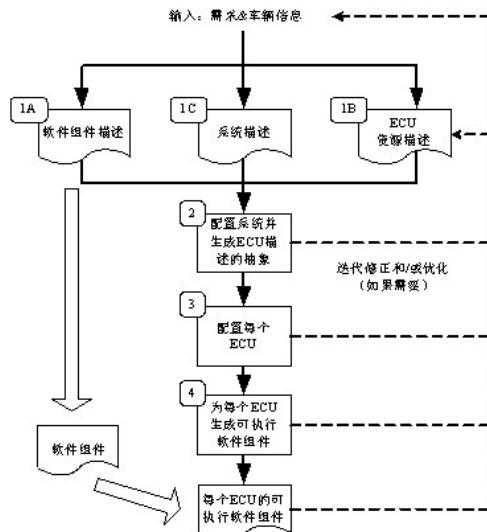


图13-11 李工设计的流程

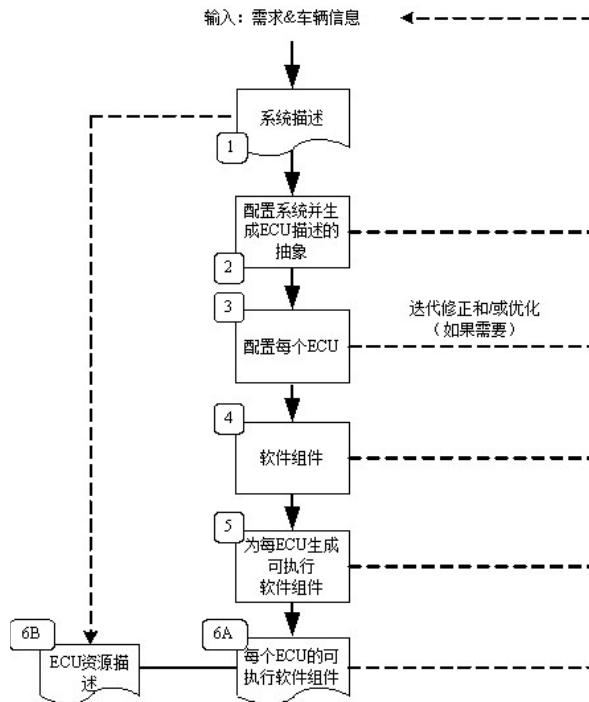


图13-12 王工设计的流程

【问题2】

图13-10中的II-1.1.1项中定义了软件架构和操作系统的要求，图13-13是满足AUTOSAR定义的操作系统各功能模块的层次结构，请说明(1)~(5)箭头所标的具体操作含义。

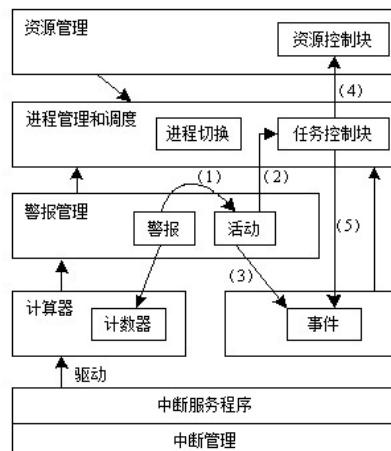


图13-13 AUTOSAR定义的操作系统结构

【问题3】

AUTOSAR是一种开放式架构，用150字以内的文字，说明采用AUTOSAR架构的主要优点，并说明汽车电子ECU覆盖汽车的哪三个领域。

试题分析

本题主要考查汽车电子基础软件架构的分析与设计，特别是系统的开发和架构设计方面。

【问题1】

本问题主要考查在一定规范的约束下，系统设计流程的设计与定义。AUTOSAR规范中要求，汽车电子软件开发流程应尽量满足并发、可多次迭代的特性。因此应该紧扣该规范对系统设计流程方面的要求，对李工和王工定义的流程进行评价。具体来说：

(1) 李工定义的流程是将软件组件描述、系统描述和ECU资源描述同时定义，而王工定义的流程仅仅只做系统描述；

(2) 王工定义的流程没有考虑软件组件的描述，只是简单将软件组件作为第4步被集成；

(3) 李工定义的ECU软件开发流程的优势是明确了顶层定义阶段，并行度高，迭代清晰。

综上，应该采用李工的设计方案。

【问题2】

本问题是一道读图题，要求考生在分析AUTOSAR定义的操作系统结构进行分析与理解的基础上进行填写。根据图示，系统从下至上依次可以分为中断管理、事件管理、警报管理、进程管理和调度以及资源管理四个部分。根据图中模块之间的关系，可以看出，(1)处主要表示当操作系统的警报管理发现ECU系统出错时，启动错误处理程序；(2)表示错误处理程序将具体动作交由进程管理完成对发生错误的任务进行处理；(3)表示错误处理程序产生一个错误事件；(4)表示任务控制块处理程序调用资源管理功能，实现硬件资源重分配；(5)表示任务控制块处理程序通知事件管理，对错误事件进行应答。

【问题3】

本题主要考查考生对AUTOSAR架构的分析与总结能力。根据题干和上述两个题目的回答，可以看出，采用AUTOSAR开放式架构的优点主要包括：

(1) 具有厂商独立性，可以有效支持多厂家汽车电子基础软件的研制；

(2) 软件层次上的重用性，可根据不同的ECU结构，通过数据配置，自动生成各种ECU软件组件；

(3) 支持汽车电子软件的全生存周期，包括构架、开发、测试、验证、授权、版本和接口；

另外，该规范覆盖整个汽车电子的三大领域为动力、底盘和车身。

试题答案

【问题1】

李工设计的流程符合AUTOSAR要求，理由是：

李工定义的流程是将软件组件描述、系统描述和ECU资源描述同时定义，而王工定义的流程仅代表只做系统描述；

王工定义的流程没有考虑软件组件的描述，只是简单将软件组件作为第4步被集成；

李工定义的ECU软件开发流程的优势是明确了顶层定义阶段，并行度高，迭代清晰。

【问题2】

(1) 操作系统的警报管理发现ECU系统出错时，启动错误处理程序；

(2) 错误处理程序将具体动作交由进程管理，完成对发生错误的任务进行处理；

(3) 错误处理程序产生一个错误事件；

(4) 任务控制块处理程序调用资源管理功能，实现硬件资源重分配；

(5) 任务控制块处理程序通知事件管理，对错误事件进行应答。

【问题3】

采用AUTOSAR开放式架构的优点是：

(1) 可以有效支持多厂家汽车电子基础软件的研制；

(2) 有利于软件的重用，可根据不同的ECU结构，通过数据配置，自动生成各种ECU软件组件；

(3) AUTOSAR定义的软件框架，支持了汽车电子软件的全生存周期，包括构架、开发、测试、验证、授权、版本和接口；

AUTOSAR规范覆盖整个汽车电子的三大领域：动力、底盘、车身。

试题9（2010年11月试题4）

试题9（2010年11月试题4）

阅读以下关于系统设计与开发工具集成的说明，在答题纸上回答问题1至问题3。

TeleDev是一个大型的电信软件开发公司，公司内部采用多种商业/开源的工具进行软件系统设计与开发工作。为了提高系统开发效率，公司管理层决定开发一个分布式的系统设计与开发工具集成框架，将现有的系统设计与开发工具有效集成在一起。集成框架开发小组经过广泛调研，得到了如下核心需求：

- (1) 目前使用的系统设计与开发工具的运行平台和开发语言差异较大，集成框架应无缝集成各个工具的功能；
- (2) 目前使用的系统设计与开发工具所支持的通信协议和数据格式各不相同，集成框架应实现工具之间的灵活通信和数据格式转换；
- (3) 集成框架需要根据实际的开发流程灵活、动态地定义系统工具之间的协作关系；
- (4) 集成框架应能集成一些常用第三方实用工具，如即时通信，邮件系统等。

集成框架开发小组经过分析与讨论，最终决定采用企业服务总线（ESB）作为集成框架的基础架构。

【问题1】

ESB是目前企业级应用集成常用的基础架构。请列举出ESB的4个主要功能，并从集成系统的部署方式、待集成系统之间的耦合程度、集成系统的可扩展性3个方面说明为何采用ESB作为集成框架的基础架构。【问题2】

在ESB基础架构的基础上，请根据题干描述中的4个需求，说明每个需求应该采用何种具体的集成方式或架构风格最为合适。

【问题3】

请指出在实现工具之间数据格式的灵活转换时，通常采用的设计模式是什么，并对实现过程进行简要描述。

试题分析

本题主要考查系统集成的相关知识及应用，需要考生结合题干描述和自己的实际经验进行回答。

【问题1】

本题主要考查企业服务总线（ESB）的基本概念，需要考生列举出企业服务总线七个核心功能中的任意四个。

ESB的概念是从SOA发展而来的，它是一种为进行连接服务提供的标准化的通信基础结构，基于开放的标准，为应用提供了一个可靠的、可度量的和高度安全的环境，并可帮助企业对业务流程进行设计和模拟，对每个业务流程实施控制和跟踪、分析并改进流程和性能。

在一个复杂的企业计算环境中，如果服务提供者和服务请求者之间采用直接的端到端的交互，那么随着企业信息系统的增加和复杂度的提高，系统之间的关联会逐渐变得非常复杂，形成一个网

状结构，这将带来昂贵的系统维护费用，同时也使得IT基础设施的复用变得困难重重。ESB提供了一种基础设施，消除了服务请求者与服务提供者之间的直接连接，使得服务请求者与服务提供者之间进一步解耦。

ESB是由中间件技术实现并支持SOA的一组基础架构，是传统中间件技术与XML、Web Service等技术结合的产物，是在整个企业集成架构下的面向服务的企业应用集成机制。具体来说，ESB具有以下功能：

- (1) 支持异构环境中的服务、消息和基于事件的交互，并且具有适当的服务级别和可管理性。
- (2) 通过使用ESB，可以在几乎不更改代码的情况下，以一种无缝的非侵入方式使现有系统具有全新的服务接口，并能够在部署环境中支持任何标准。
- (3) 充当缓冲器的ESB（负责在诸多服务之间转换业务逻辑和数据格式）与服务逻辑相分离，从而使不同的系统可以同时使用同一个服务，用不着在系统或数据发生变化时，改动服务代码。
- (4) 在更高的层次，ESB还提供诸如服务代理和协议转换等功能。允许在多种形式下通过像HTTP、SOAP和JMS总线的多种传输方式，主要是以网络服务的形式，为发表、注册、发现和使用企业服务或界面提供基础设施。
- (5) 提供可配置的消息转换翻译机制和基于消息内容的消息路由服务，传输消息到不同的目的地。
- (6) 提供安全和拥有者机制，以保证消息和服务使用的认证、授权和完整性。

ESB的以上功能中，可以总结出其核心功能包括应用程序的位置透明性、传输协议转换、消息格式转换、消息路由、消息增强、安全支持、监控和管理。

在企业应用集成方面，与现存的、专有的集成解决方案相比，ESB具有以下优势：

- (1) 扩展的、基于标准的连接。ESB形成一个基于标准的信息骨架，使得在系统内部和整个价值链中可以容易地进行异步或同步数据交换。ESB通过使用XML、SOAP和其他标准，提供了更强大的系统连接性。
- (2) 灵活的、服务导向的应用组合。基于SOA，ESB使复杂的分布式系统（包括跨多个应用、系统和防火墙的集成方案）能够由以前开发测试过的服务组合而成，使系统具有高度可扩展性。
- (3) 提高复用率，降低成本。按照SOA方法构建应用，提高了复用率，简化了维护工作，进而减少了系统总体成本。
- (4) 减少市场反应时间，提高生产率。ESB通过构件和服务复用，按照SOA的思想简化应用组合，基于标准的通信、转换和连接来实现这些优点。

根据集成系统的部署方式，可以看出采用ESB作为集成框架，能够实现灵活的部署结构，包括CS结构、P2P结构等。

从待集成系统之间的耦合程度，可以看出采用ESB作为集成框架，待集成系统只需要和总线进行联系，彼此之间不需要互相通信，这样就大大降低了系统的耦合程度。

从集成系统的可扩展性，可以看出采用ESB作为集成框架，在加入新的待集成系统时，只需要采用插件的方式实现传输协议和数据格式的适配即可，系统的可扩展性较强。

【问题2】

对于需求(1)“目前使用的系统设计与开发工具的运行平台和开发语言差异较大，集成框架应无缝集成各个工具的功能”来说，由于需要共享系统的功能，并且系统的运行平台与语言差异较大，应该采用面向服务的方式进行功能集成，可以将工具的功能包装为服务，实现跨语言与跨平台访问。

对于需求(2)"目前使用的系统设计与开发工具所支持的通信协议和数据格式各不相同,集成框架应实现工具之间的灵活通信和数据格式转换"来说,工具所支持的通信协议和数据格式各不相同,并需要实现工具之间的灵活通信协议和数据格式交换,因此应该基于消息总线,以协议及数据适配器的方式实现灵活的通信协议和数据格式转换。

对于需求(3)"集成框架需要根据实际的开发流程灵活、动态地定义系统工具之间的协作关系"来说,集成框架需要根据实际的软件系统开发流程,灵活、动态地定义系统设计与开发工具之间的协作关系,因此应该采用解释器架构风格,引入工作流定义语言及其引擎来动态描述工具之间的协作关系。

对于需求(4)"集成框架应能集成一些常用第三方实用工具,如即时通信,邮件系统等"来说,应该采用界面集成的方法对第三方工具进行集成,绕过工具内部的复杂处理逻辑,实现功能集成。

【问题3】

本题主要考查数据转换在实现层面上的常用方法。在实现工具之间数据格式的灵活转换时,通常采用适配器设计模式。即应首先定义一个统一的数据转换接口类,然后针对不同的数据格式转换需求定义对应的实际转换类,实际转换类需要继承数据转换接口类,并实现接口转换类定义的接口。

试题答案

本题主要考查考生应用集成的掌握情况。

【问题1】

ESB的主要功能包括:

- (1) 应用程序的位置透明性
- (2) 传输协议转换
- (3) 消息格式转换
- (4) 消息路由
- (5) 消息增强
- (6) 安全支持
- (7) 监控和管理

采用ESB作为集成框架,能够实现灵活的部署结构,包括CS结构、P2P结构等。

采用ESB作为集成框架,待集成系统只需要和总线进行联系,彼此之间不需要互相通信,这样就大大降低了系统的耦合程度。

采用ESB作为集成框架,在加入新的待集成系统时,只需要采用插件的方式实现传输协议和数据格式的适配即可,系统的可扩展性较强。

【问题2】

对于需求(1)来说,由于需要共享系统的功能,并且系统的运行平台与语言差异较大,应该采用面向服务的方式进行功能集成,可以将工具的功能包装为服务,实现跨语言与跨平台访问。

对于需求(2)来说,工具所支持的通信协议和数据格式各不相同,并需要实现工具之间的灵活通信协议和数据格式交换,因此应该基于消息总线,以协议及数据适配器的方式实现灵活的通信协议和数据格式转换。

对于需求(3)来说,集成框架需要根据实际的软件系统开发流程,灵活、动态地定义系统设计与开发工具之间的协作关系,因此应该引入工作流定义语言及其引擎来动态描述工具之间的协作关系。

对于需求(4)来说，应该采用界面集成的方法对第三方工具进行集成，绕过工具内部的复杂处理逻辑。

【问题3】

在实现工具之间数据格式的灵活转换时，通常采用适配器设计模式。即应首先定义一个统一的数据转换接口类，然后针对不同的数据格式转换需求定义对应的实际转换类，实际转换类需要继承数据转换接口类，并实现接口转换类定义的接口。

第13章：案例分析

试题10 (2010年11月试题5)

试题10 (2010年11月试题5)

阅读以下信息系统可靠性问题的说明，在答题纸上回答问题1至问题3。

某软件公司开发一项基于数据流的软件，其系统的主要功能是对输入数据进行多次分析、处理和加工，生成需要的输出数据。需求方对该系统的软件可靠性要求很高，要求系统能够长时间无故障运行。该公司将该系统设计交给王工负责。王工给出该系统的模块示意图如图13-14所示。王工解释：只要各个模块的可靠度足够高，失效率足够低，则整个软件系统的可靠性是有保证的。

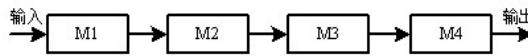


图13-14 王工建议的软件系统模块示意图

李工对王工的方案提出了异议。李工认为王工的说法有两个问题：第一，即使每个模块的可靠度足够高，但是整个软件系统模块之间全部采用串联，则整个软件系统的可靠度明显下降。假设各个模块的可靠度均为0.99，则整个软件系统的可靠度为 $0.99^4 \approx 0.96$ ；第二，软件系统模块全部采用串联结构时，一旦某个模块失效，则意味着整个软件系统失效。

李工认为，应该在软件系统中采用冗余技术中的动态冗余或者软件容错的N版本程序设计技术，对容易失效或者非常重要的模块进行冗余设计，将模块之间的串联结构部分变为并联结构，来提高整个软件系统的可靠性。同时，李工给出了采用动态冗余技术后的软件系统模块示意图，如图13-15所示。

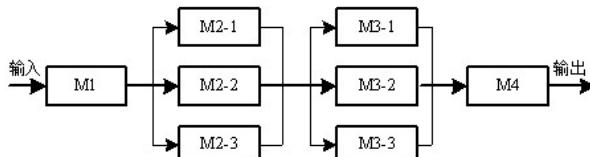


图13-15 李工建议的系统模块示意图

刘工建议，李工方案中M1和M4模块没有采用容错设计，但M1和M4发生故障有可能导致严重后果。因此，可以在M1和M4模块设计上采用检错技术，在软件出现故障后能及时发现并报警，提醒维护人员进行处理。

注：假设各个模块的可靠度均为0.99。

【问题1】

在系统可靠性中，可靠度和失效率是两个非常关键的指标，请分别解释其含义。

【问题2】

请解释李工提出的动态冗余和N版本程序设计技术，给出图13-14中模块M2采用图13-15动态冗余技术后的可靠度。

请给出采用李工设计方案后整个系统可靠度的计算方法，并计算结果。

【问题3】

请给出检错技术的优缺点，并说明检测技术常见的实现方式和处理方式。

试题分析

本题考查信息系统中可靠性的设计，是比较传统的题目，要求考生细心分析题目中所描述的内容。

【问题1】

本问题考查信息系统可靠性的两个基本概念：可靠度和失效率。可靠性是指产品在规定的条件下和规定的时间内完成规定功能的能力。考虑到软件本身的复杂性，软件可靠性的定义是：在规定的条件下，在规定的时间内，软件不引起系统失效的概率。

在软件可靠性的定量描述中，软件可靠性可以基于使用条件、规定时间、系统输入、系统使用和软件缺陷等变量构建数学表达式，来对软件可靠性进行定量描述。相关概念有规定时间、失效率、可靠度、失效强度、失效率、平均无失效时间等。其中可靠度是表示可靠性最直接的方式，是软件系统在规定的条件下、规定的实践内不发生失效的概率。而失效率又称风险函数，也可以称为条件失效强度，是指运行至此刻系统未出现失效的情况下，单位时间系统出现失效的概率。

【问题2】

本问题考查在常规的软件设计中，应用各种方法和技术，使程序设计在兼顾用户功能和性能需求的同时，全面满足软件的可靠性要求。常见的软件可靠性技术主要有容错设计、检错设计和降低复杂度设计等技术。

其中，容错设计技术主要有恢复快设计、N版本程序设计和冗余设计三种方法。

1、冗余设计

冗余是指在正常系统运行基础上加上一定数量的资源，包括信息、时间、硬件、和软件。冗余是容错技术的基础，通过冗余资源的加入，可以使系统的可靠性得到较大的提高。主要的冗余技术有结构冗余（硬件冗余和软件冗余）、信息冗余、时间冗余和冗余附加四种。结构冗余是常用的冗余技术，按其工作方式，可分为静态冗余、动态冗余和混合冗余三种。

（1）静态冗余。静态冗余又称为屏蔽冗余或被动冗余，常用的有三模冗余和多模冗余。静态冗余通过表决和比较来屏蔽系统中出现的错误。例如，三模冗余是对三个功能相同，但由不同的人采用不同的方法开发出的模块的运行结果进行表决，以多数结果作为系统的最终结果。即如果模块中有一个出错，这个错误能够被其他模块的正确结果“屏蔽”。由于无需对错误进行特别的测试，也不必进行模块的切换就能实现容错，故称为静态容错。

（2）动态冗余。动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来顶替它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可不工作。前者叫做热备份系统（双重系统），后者叫做冷备份系统（双工系统、双份系统）。在热备份系统中，两套系统同时、同步运行，当联机子系统检测到错误时，退出服务进行检修，而由热备份子系统接替工作，备用模块在待机过程中其失效率为0；处于冷备份的子系统平时停机或者运行与联机系统无关的运算，当联机子系统产生故障时，人工或自动进行切换，使冷备份系统成为联机系统。在运

行冷备份时，不能保证从系统断点处精确地连续工作，因为备份机不能取得原来的机器上当前运行的全部数据。

(3) 混合冗余。混合冗余技术是将静态冗余和动态冗余结合起来，且取二者之长处。它先使用静态冗余中的故障屏蔽技术，使系统免受某些可以被屏蔽的故障的影响。而对那些无法屏蔽的故障则采用主动冗余中的故障检测、故障定位和故障恢复等技术，并且对系统可以作重新配置。因此，混合冗余的效果要大大优于静态冗余和动态冗余。然而，由于混合冗余既要有静态冗余的屏蔽功能，又要有动态冗余的各种检测和定位等功能，它的附加硬件的开销是相当大的，所以混合冗余的成本很高，仅在对可靠性要求极高的场合中采用。

(4) 信息冗余。信息冗余是在实现正常功能所需要的信息外，再添加一些信息，以保证运行结果正确性的方法。例如，检错码和纠错码就是信息冗余的例子。这种冗余信息的添加方法是按照一组预定的规则进行的。符合添加规则而形成的带有冗余信息的字称为码字，而那些虽带有冗余信息但不符合添加规则的字则称为非码字。当系统出现故障时，可能会将码字变成非码字，于是在译码过程中会将引起非码字的故障检测出来，这就是检错码的基本思想。纠错码则不仅可以将错误检测出来，还能将由故障引起的非码字纠正成正确的码字。

由此可见，信息冗余的主要任务在于研究出一套理想的编码和译码技术来提高信息冗余的效率。编码技术中应用最广泛的是奇偶校验码、海明校验码和循环冗余校验码。

(5) 时间冗余。时间冗余是以时间（即降低系统运行速度）为代价以减少硬件冗余和信息冗余的开销来达到提高可靠性的目的。在某些实际应用中，硬件冗余和信息冗余的成本、体积、功耗、重量等开销可能过高，而时间并不是太重要的因素时，可以使用时间冗余。时间冗余的基本概念是重复多次进行相同的计算，或称为重复执行（复执），以达到故障检测的目的。

实现时间冗余的方法很多，但是其基本思想不外乎是对相同的计算任务重复执行多次，然后将每次的运行结果存放起来再进行比较。若每次的结果相同，则认为无故障；若存在不同的结果，则说明检测到了故障。不过，这种方法往往只能检测到瞬时性故障而不宜检测永久性的故障。

(6) 冗余附加。冗余附加是指为实现上述冗余技术所需的资源和技术，包括程序、指令、数据，以及存放和调用它们的空间等。动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来顶替它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可不工作。前者叫做热备份系统（双重系统），后者叫做冷备份系统（双工系统、双份系统）。在热备份系统中，两套系统同时、同步运行，当联机子系统检测到错误时，退出服务进行检修，而由热备份子系统接替工作，备用模块在待机过程中其失效率为0；处于冷备份的子系统平时停机或者运行与联机系统无关的运算，当联机子系统产生故障时，人工或自动进行切换，使冷备份系统成为联机系统。在运行冷备份时，不能保证从系统断点处精确地连续工作，因为备份机不能取得原来的机器上当前运行的全部数据。

2、恢复块方法

恢复块方法是一种动态的故障屏蔽技术，采用后向恢复策略。恢复块方法提供具有相同功能的主块和几个后备块，一个块就是一个执行完整的程序段，主块首先投入运行，结束后进行验证测试，如果没有通过验证测试，系统经现场恢复后由后备块1运行。后备块1运行结束后也进行验证测试，如果没有通过验证测试，系统经现场恢复后由后备块2运行。重复这一过程，可以重复到耗尽所有的后备块，或者某个程序故障行为超出了预料，从而导致不可恢复的后果。在程序设计时，应保证实现主块和后备块之间的独立性，避免相关错误的产生，使主块和后备块之间的共性错误降到最低程度。

恢复块方法依赖于一个裁决者，那就是验证测试（可接受测试），由它来决定同一算法不同实现的计算结果是否正确。带有恢复块的系统被分成故障可恢复的块。整个系统就由这些容错块组成。每一块包含至少一个一级模块、一个二级模块和一个例外处理模块，以及一个验证测试模块。验证测试模块完成故障检测功能，它本身的故障对恢复块方法而言是共性的，因此，必须确保它的正确性。同时，验证测试模块是为了确定模块计算结果的正确性，它必须尽可能的简单。

3、N版本程序设计

N版本程序设计是一种静态的故障屏蔽技术，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中N个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少N个版本的程序在表决点上相关错误的概率。

4、可靠性计算

计算机系统是一个复杂系统，影响其可靠性的因素很多，很难直接进行可靠性分析，往往需要建立对应的数学模型。组合模型是分析系统可靠性的一种常用方法。

组合模型下可靠度的计算方法为：

串联系统： $R = R_1 \times R_2 \times \dots \times R_n$ ；

并联系统： $R = 1 - (1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)$ ；

串联和并联混合系统则根据实际情况，灵活运用上述两个计算公式。

M2采用动态冗余后，成为并联系统，则其可靠度为： $R = 1 - (1 - 0.99)^3 = 0.999999$ 。

李工给出的方案同时采用了串联和并联方式，其计算方法为首先计算出中间M2和M3两个并联系统的可靠度，再按照串联系统的计算方法计算出整个系统的可靠度。

$$R = 0.99 \times 0.999999 \times 0.999999 \times 0.99 = 0.98$$

【问题3】

本问题考查软件可靠性设计中的检错技术。

检错技术常见的实现方式有多种，最直接的一种实现方式是判断返回结果，如果返回结果超出正常范围，则进行异常处理；计算运行时间也是一种常用技术，如果某个模块或函数运行时间超过预期时间，可以判断出现故障；还有置状态标志位等多种方法，自检的实现方式需要根据实际情况来选用。

检错技术的处理方式也有多种，大多数都采用“查处故障-停止软件运行-报警”的处理方式。但根据故障的不同情况，也有采用不停止或部分停止软件系统运行的情况，这一般由故障是否需要实时处理来决定。

检错技术实现的代价一般低于容错技术和冗余技术，但有一个明显的缺点，就是不能自动解决故障，出现故障后如果不进行人工干预，将最终导致软件系统不能正常运行。

试题答案

【问题1】

可靠度就是系统在规定的条件下、规定的时间内不发生失效的概率。

失效率又称风险函数，也可以称为条件失效强度，是指运行至此刻系统未出现失效的情况下，单位时间系统出现失效的概率。

【问题2】

动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来替

代它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可以不工作。前者叫热备份系统（双重系统），后者叫冷备份系统（双工系统、双份系统）。

N版本程序设计是一种静态的故障屏蔽技术，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中N个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少N个版本的程序在表决点上相关错误的概率。

M2采用动态冗余后的可靠度为：

$$R = 1 - (1 - 0.99)^3 \approx 0.999999$$

李工的方案同时采用了串联和并联方式，其计算方法为首先计算出中间M2和M3两个并联系统的可靠度，再按照串联系统的计算方法计算出整个系统的可靠度。

$$R = 0.99 \times 0.999999 \times 0.999999 \times 0.99 \approx 0.98$$

【问题3】

检错技术实现的代价一般低于容错技术和冗余技术，但有一个明显的缺点，就是不能自动解决故障，出现故障后如果不进行人工干预，将最终导致软件系统不能正常运行。

检错技术常见的实现方式：最直接的一种实现方式是判断返回结果，如果返回结果超出正常范围，则进行异常处理；计算运行时间也是一种常用技术，如果某个模块或函数运行时间超过预期时间，可以判断出现故障；还有置状态标志位等多种方法，自检的实现方式需要根据实际情况来选用。

检错技术的处理方式，大多数都采用“查处故障-停止软件运行-报警”的处理方式。但根据故障的不同情况，也有采用不停止或部分停止软件系统运行的情况，这一般由故障是否需要实时处理来决定。

第13章：案例分析

试题11（2011年11月试题1）

试题11（2011年11月试题1）

阅读以下关于软件架构评估的说明，在答题纸上回答问题1和问题2。

某网上购物电子商务公司拟升级正在使用的在线交易系统，以提高用户网上购物在线支付环节的效率和安全性。在系统的需求分析与架构设计阶段，公司提出的需求和关键质量属性场景如下：

- (a) 正常负载情况下，系统必须在0.5秒内对用户的交易请求进行响应；
- (b) 信用卡支付必须保证99.999%的安全性；
- (c) 对交易请求处理时间的要求将影响系统的数据传输协议和处理过程的设计；
- (d) 网络失效后，系统需要在1.5分钟内发现错误并启用备用系统；
- (e) 需要在20人月内为系统添加一个新的CORBA中间件；
- (f) 交易过程中涉及的产品介绍视频传输必须保证画面具有600*480的分辨率，20帧/秒的速度；
- (g) 更改加密的级别将对安全性和性能产生影响；
- (h) 主站点断电后，需要在3秒钟内将访问请求重定向到备用站点；

(i) 假设每秒中用户交易请求的数量是10个，处理请求的时间为30毫秒，则“在1秒内完成用户的交易请求”这一要求是可以实现的；

(j) 用户信息数据库授权必须保证99.999%可用；

(k) 目前对系统信用卡支付业务逻辑的描述尚未达成共识，这可能导致部分业务功能模块的重複，影响系统的可修改性；

(l) 更改Web界面接口必须在4人周内完成；

(m) 系统需要提供远程调试接口，并支持系统的远程调试。

在对系统需求和质量属性场景进行分析的基础上，系统的架构师给出了三个候选的架构设计方案。公司目前正在组织系统开发的相关人员对系统架构进行评估。

【问题1】

在架构评估过程中，质量属性效用树（utility tree）是对系统质量属性进行识别和优先级排序的重要工具。请给出合适的质量属性，填入图13-16中（1）、（2）空白处；并选择题干描述的（a）~（m），填入（3）~（6）空白处，完成该系统的效用树。

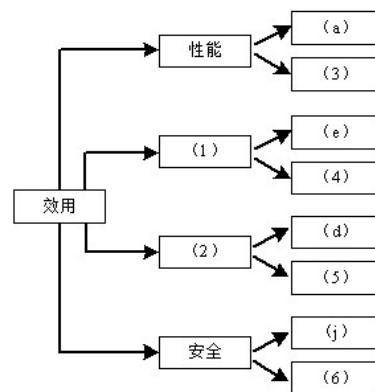


图13-16 在线交易系统效用树

【问题2】

在架构评估过程中，需要正确识别系统的架构风险、敏感点和权衡点，并进行合理的架构决策。请用300字以内的文字给出系统架构风险、敏感点和权衡点的定义，并从题干（a）~（m）中各选出1个对系统架构风险、敏感点和权衡点最为恰当的描述。

试题分析

本题考查软件质量属性的相关内容，以及架构风险、敏感点、权衡点的基本概念。软件质量属性在架构设计中是一个重要关注点，架构设计的过程往往就是对不同质量属性的平衡与取舍。

【问题1】

问题1考查考生对各种质量属性的理解。质量属性种类繁多，如：性能、可用性、可修改性、安全性等。

首先分析（3）应填写的内容，该空的解答较为直接，只需要分析题目给出的质量属性场景中，除（a）还有哪个属于性能。“（f）交易过程中涉及的产品介绍视频传输必须保证画面具有600*480的分辨率，20帧/秒的速率；”描述中，强调了视频必须保证的画面分辨率以及每秒帧数，这是对性能的要求。

第（1）和（2）空的分析较为复杂，需要通过反向推导的方式分析其分支之下的（e）与（d）属于哪个质量属性。“（e）需要在20人月内为系统添加一个新的CORBA中间件；”涉及在原有系统基础之上，增加新的功能，这个时限要求原系统具有良好的可修改性，否则无法按期修改完成，所

以(1)应为可修改性。同理：“(d)网络失效后，系统需要在1.5分钟内发现错误并启用备用系统；”是对系统可用性的要求，所以(2)应填可用性。

当完成前面的几个空以后，接下来的几个空就比较容易解决了。即判断剩余的质量场景：

(b)、(c)、(g)、(h)、(i)、(k)、(l)、(m)，哪个属于可修改性，哪个属于可用性，哪个属于安全性。“(b)信用卡支付必须保证99.999%的安全性；”显然体现的是安全性；“(h)主站点断电后，需要在3秒钟内将访问请求重定向到备用站点；”是一种保障系统在出现问题时，仍能继续使用的机制，即提高可用性的方法；“(l)更改Web界面接口必须在4人周内完成；”体现出系统的可修改性。

【问题2】

问题2属于概念题，系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。敏感点是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。

题干描述中的“(k)目前对系统信用卡支付业务逻辑的描述尚未达成共识，这可能导致部分业务功能模块的重复，影响系统的可修改性”属于架构风险，因为未达成共识的业务逻辑描述存在隐患。

“(c)对交易请求处理时间的要求将影响系统的数据传输协议和处理过程的设计”是敏感点，因为对交易请求处理时间的要求将影响到数据传输协议和处理过程的设计，这也就意味着有多个构件将受其影响。

“(g)更改加密的级别将对安全性和性能产生影响”描述的是权衡点，因为更改加密级别将影响多个质量属性的特性，这两个方面的影响往往是：安全性提高的同时，性能降低；而安全性降低的同时性能提高。

试题答案

【问题1】

表13-8 效用树答案表

编 号	答 案
(1)	可修改性
(2)	可用性
(3)	(f)
(4)	(l)
(5)	(h)
(6)	(b)

【问题2】

系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。

题干描述中，(k)描述的是系统架构风险；(c)描述的是敏感点；(g)描述的是权衡点。

试题12 (2011年11月试题2)

阅读以下关于软件系统数据建模的说明，在答题纸上回答问题1至问题3。

某软件公司成立项目组为某高校开发一套教职工信息管理系统。与教职工信息相关的需求和处理需求如下。

(1) 数据需求：在教职工信息中能够存储学校所有在职的教工和职工信息，包括姓名、所属部门、出生年月、工资编号、工资额和缴税信息；部门信息中包括部门编号、部门名称、部门人数和办公地点信息。

(2) 处理需求：能够根据编制内或外聘教职工的工资编号分别查询其相关信息；每个月的月底统一核发工资，要求系统能够以最快速度查询出教工或者职工所在部门名称、实发工资金额；由于学校人员相对稳定，所以数据变化及维护工作量很少。

项目组王工和李工针对上述应用需求分别给出了所设计的数据模型（如图13-17和图13-18所示）。王工遵循数据库设计过程，按照第三范式对数据进行优化和调整，所设计的数据模型简单且基本没有数据冗余；而李工设计的数据模型中存在大量数据冗余。

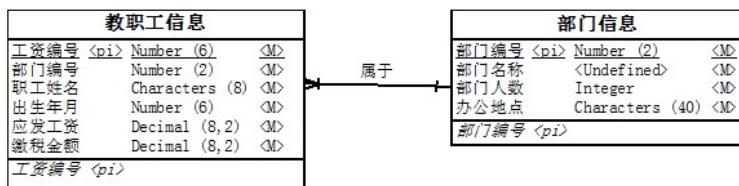


图13-17 王工设计的数据模型

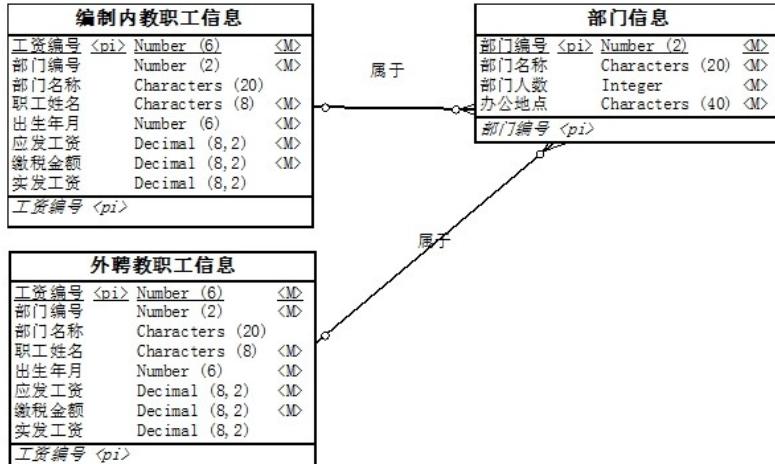


图13-18 李工设计的数据模型

项目组经过分析和讨论，特别是针对数据处理中对数据访问效率的需求，最终选择了李工给出的数据模型设计方案。

【问题1】

请用300字以内的文字，说明什么是数据库建模中的反规范化技术，指出采用反规范化技术能获得哪些益处，可能带来哪些问题。

【问题2】

请简要叙述常见的反规范化技术有哪些。

【问题3】

请分析李工是如何应用反规范化技术来满足教职工信息管理需求的。

试题分析

本题考查数据库相关的知识。题目涉及了反规范化技术。

规范化技术在数据库中的应用十分常见。当数据库对数据模型进行规范化处理后，会发现这些经过规范化处理的模型在进行查询操作时效果并不理想。因为经过规范化处理的数据模型形成了一系列的小表，每个表的数据量较小，进行查询操作时往往需要应用程序对这些表进行动态的连接操作，这就要在不同的表中进行多次I/O操作。表的连接操作对于较少的，小容量表也许不会产生较大的影响，但是对于数据量十分庞大的数据库，这种多表连接操作在时间上是很难让用户接受的。此时，提高效率的最好方法就是使这些小表合并在一起，这就是数据的反规范化处理。

【问题1】

经过以上对于规范化处理的描述可以得知：反规范化技术主要是为了提升查询性能而进行的一系列处理，包括对数据表进行连接，合理规划数据的存放，以及引入合理的数据冗余。其益处在：降低连接操作的需求、降低外码和索引的数目，还可能减少表的数目，能够提高查询效率。带来的问题是：数据的重复存储，浪费了磁盘空间；可能出现数据的完整性问题，为了保障数据的一致性，增加了数据维护的复杂性，会降低修改速度。

【问题2】

问题2为纯概念题，请参看本题参考答案。

【问题3】

问题3需要利用问题2的结论并结合题目要求进行解题。教职工信息管理系统要求"能够根据编制内或外聘教职工的工资编号分别查询其相关信息"，并从题目"特别是针对数据处理中对数据访问效率的需求"可以看出系统对数据访问效率非常重视。而李工正是采用了反规范化技术，使得效率能明显提升。对比王工和李工所设计的数据模型可以发现：在李工设计的数据模型中，增加了"部门名称"列，这属于增加冗余列的技术；增加了"实发工资"列，这属于增加派生列的技术；将教职工信息表分割为"编制内教职工信息"表和"外聘教职工信息"表，这属于水平分割表。

试题答案

【问题1】

规范化设计后，数据库设计者希望牺牲部分规范化来提高性能，这种从规范化设计的回退方法称为反规范化技术。

采用反规范化技术的益处：降低连接操作的需求、降低外码和索引的数目，还可能减少表的数目，能够提高查询效率。

可能带来的问题：数据的重复存储，浪费了磁盘空间；可能出现数据的完整性问题，为了保障数据的一致性，增加了数据维护的复杂性，会降低修改速度。

【问题2】

(1) 增加冗余列：在多个表中保留相同的列，通过增加数据冗余减少或避免查询时的连接操作。

(2) 增加派生列：在表中增加可以由本表或其他表中数据计算生成的列，减少查询时的连接操作并避免计算或使用集合函数。

(3) 重新组表：如果许多用户需要查看两个表连接出来的结果数据，则把这两个表重新组成一个表来减少连接而提高性能。

(4) 水平分割表：根据一列或多列数据的值，把数据放到多个独立的表中，主要用于表数据规模很大、表中数据相对独立或数据需要存放到多个介质上时使用。

(5) 垂直分割表：对表进行分割，将主键与部分列放到一个表中，主键与其他列放到另一个表中，在查询时减少I/O次数。

【问题3】

在教职工信息管理系统的需求中，能够根据编制内或外聘教职工的工资编号分别查询其相关信息，数据查询要求有很高的处理效率。李工所设计的数据模型中采用了三种反规范化技术：

(1) 增加冗余列：增加“部门名称”列，消除了数据查询中“教职工信息”表和“部门信息”表之间的连接；

(2) 增加派生列：增加“实发工资”列，消除了实发工资的计算过程；

(3) 水平分割表：将教职工信息表分割为“编制内教职工信息”表和“外聘教职工信息”表，减少了数据查询的范围。

第13章：案例分析

试题13（2011年11月试题3）

试题13（2011年11月试题3）

阅读以下有关嵌入式系统设计的说明，在答题纸上回答问题1至问题3。

某公司承接了某机载嵌入式系统的研制任务。该机载嵌入式系统由数据处理模块、大容量模块、信号处理模块、数据交换模块和电源模块等组成。数据处理模块有2个，分别完成数据融合和导航通信任务；大容量模块的主要功能是存储系统数据，同时要记录信号处理模块、数据处理模块的自检测、维护数据，向数据处理模块提供地图数据；信号处理模块的处理器为专用的DSP，接收红外、雷达等前端传感器数据并进行处理，将处理后的有效数据（数据带宽较大）发送给数据处理模块；数据交换模块主要负责系统的数据交换；电源模块主要负责给其他模块供电，电源模块上没有软件。

要求该机载嵌入式系统符合综合化、模块化的设计思想，并考虑系统在生命周期中的可靠性和安全性，以及硬件的可扩展性和软件可升级性，还要求系统通信延迟小，支持多模块上的应用任务同步。

【问题1】

在设计系统架构时，李工提出了如图13-19所示的系统架构，即模块间的网络通信采用光纤通信（Fiber Channel, FC）技术，而王工认为应采用VME总线架构，如图13-20所示。王工的理由是公司多年来基于VME总线技术设计了多个产品，技术成熟，且费用较小。但公司经过评审后，决定采用李工的方案。

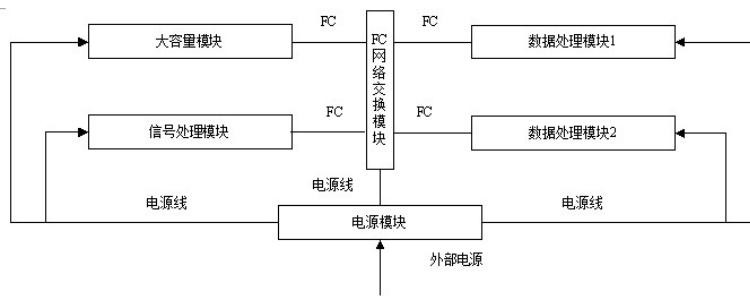


图13-19 基于FC技术的机载嵌入式系统架构

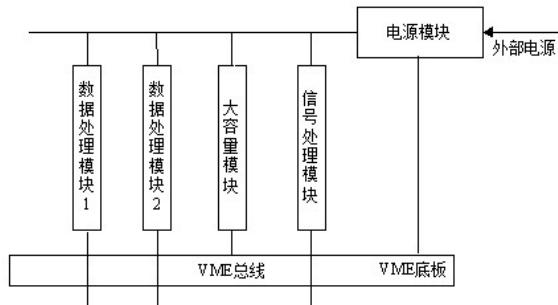


图13-20 基于VME总线的机载嵌入式系统架构

请用500字以内的文字，说明VME和FC各自的特点，并针对机载嵌入式系统的要求，指出公司采用李工方案的理由。

【问题2】

公司依据ARINC653标准，设计了满足ARINC653标准的操作系统，该操作系统对系统中可能发生的模块级、分区级和进程级的错误进行处理，实现了如图13-21所示的系统健康监控机制，请分别将备选答案中的各种错误和健康监控部件填入图13-21中的(1)~(5)。

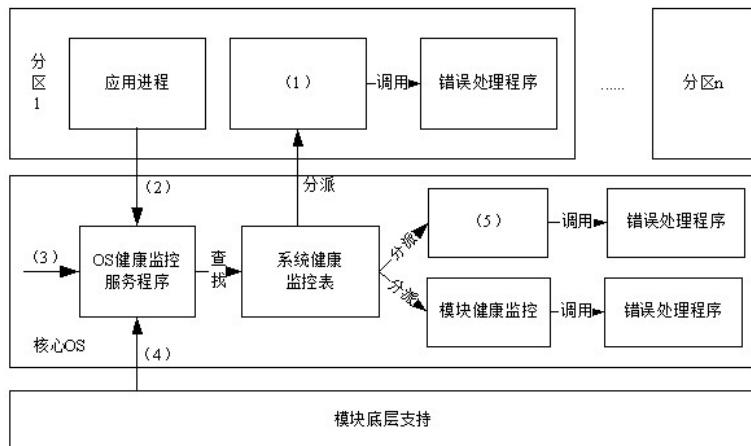


图13-18 系统健康监控机制

备选答案：分区健康监控、分区初始化阶段出现的分区配置错误、分区切换时出现的错误、应用进程错误、进程健康监控。

注：ARINC653标准 (Avionics Application Software Standard Interface) 是美国航空电子工程协会AEC于1997年为航空民用飞机的模块化综合航空电子系统定义的应用程序接口标准，该标准提出了分区 (Partition) 的概念以及健康监控 (health monitoring) 机制。分区是应用的一种功能划分，也是操作系统调度的基本单位，严格按预先分配的时间片调度。分区间具有时空隔离特点。分区内的每一执行单元称为进程。

【问题3】

为了实现满足ARINC653标准的操作系统的时空分区隔离机制，项目组选择了Power PC作为数据处理模块的处理器 (CPU)。这样，当一个分区出现故障时，不会蔓延到模块中同一处理器的其他分区。请用500字以内的文字，说明如何采用Power PC实现应用与内核以及诸应用之间的隔离和保护。

试题分析

本题考查嵌入式架构设计的相关知识。

嵌入式系统的设计和实现是一个较为复杂的工程，涉及软件和硬件基础知识，本题要求按综合化、模块化的设计思想，其中涉及PC、VME总线以及ARINC653操作系统的概念，比较新颖，需要考生仔细阅读题意，结合相关计算机软硬件知识回答问题。

【问题1】

问题1给出了李工的基于FC技术机载嵌入式系统架构和王工的基于VME总线机载嵌入式系统架构，这两种架构都较为常见。若要分析此处使用哪种方案更优，需要从题目已给出的一些信息结合分析。即考虑综合化、模块化的设计思想，以及系统的安全性与可靠性，硬件的扩展与软件的升级。同时要求系统通信延迟小，支持多模块上的应用任务同步。此外系统要求接收红外、雷达等前端传感器数据并进行处理，将处理结果通过网络发送给数据处理模块。

VME (VersaModule Eurocard) 总线是一种通用的计算机总线，结合了Motorola公司Versa总线的电气标准和在欧洲建立的Eurocard标准的机械形状因子，是一种开放式架构。它定义了一个在紧密耦合硬件构架中可进行互连数据处理、数据存储和连接外围控制器件的系统。VME总线采用存储映射方式，多主仲裁机制，同一时刻由单一主机控制，同时仲裁机制为菊花链方式。针对本系统要求，当多主机设备仲裁时，按菊花链的连接次序将一个主机处理完成后，才能将控制权交给另一个主机来控制总线，导致任务执行延时大，不能满足"系统通信延迟小"以及"支持多模块上的应用任务同步"的要求。VME数据传输速率为0~500Mbps,而FC的数据传输率为1Gbps到10Gbps.

FC具有的特点是传输率高，允许在同一接口上传输多种不同的协议，传输距离远，采用单模光纤作为数据传输介质时，传输距离可高达10km;连接简单灵活，误码率低等。

由于FC采用消息包交换机制，支持广播和组播，任务执行并发性好，能满足"系统通信延迟小"以及"支持多模块上的应用任务同步"的要求；与VME比较，FC实时性好，带宽高；允许在同一接口上传输多种不同的协议，对上层应用实现提供了便利；FC采用消息机制，扩展好，如模块较多可采用多个FC网络交换机模块级联；传输距离远，当与外部其他设备相连时比较方便；系统采用统一的FC网络代替了VME底板总线，降低总线驱动的功耗，简化了底板。所以最后采用FC的系统结构。

【问题2】

问题2是一个选择题，在解答本题时，关键在于区分不同的错误归属于哪一个错误级别。

模块级错误一般包括：

- 1.模块初始化时发生的模块配置错误；
- 2.模块初始化时的其他错误；
- 3.系统功能执行期间出现的错误；
- 4.分区切换时发生的错误；
- 5.电源故障。

分区级错误一般包括：

- 1.分区初始化阶段出现的分区配置错误；
- 2.分区初始化阶段出现的其他错误；
- 3.进程管理中的错误；
- 4.错误处理进程的错误。

进程级错误一般包括：

- 1.应用进程产生的应用错误；
- 2.非法操作系统请求；
- 3.进程执行错误（溢出、缓冲区冲突等）。

【问题3】

问题3看似在考技术应用，实质上是考概念。采用Power PC实现系统隔离与保护有两种机制，这两种机制的详细说明，参看本题试题答案。

试题答案

【问题1】

1.VME总线采用存储映射方式，多主机仲裁机制，同一时刻由单一主机控制，同时仲裁机制为菊花链方式。针对本系统要求，采用VME方案存在如下问题。

(1) 当多主机设备仲裁时，按菊花链的连接次序一个主机处理完成后，才能将控制权交给另一主机控制总线，导致任务执行延时大，不能满足“系统通信延迟小”以及“支持多模块上的应用任务同步”的要求。

(2) VME总线方式限制了可扩展性。与FC相比，VME总线实时性差，带宽低。

2.FC采用消息包交换机制，支持广播和组播。针对本系统要求，采用FC方案有以下优点。

(1) 由于采用消息包交换机制，支持广播和组播，任务执行并发性好，能满足“系统通信延迟小”以及“支持多模块上的应用任务同步”的要求。

(2) FC的误码率低，可靠性高。与VME比较，FC实时性好，带宽高。

(3) 允许在同一接口上传输多种不同的协议，对上层应用实现提供了便利。

(4) FC采用消息机制，FC可扩展性好，如模块较多可采用多个FC网络交换模块级联。

(5) FC的传输距离远，当与外部其他设备相连时，比较方便。

(6) 系统采用统一的FC网络代替了VME底板总线，降低总线驱动的功耗，简化了底板。

【问题2】

表13-9 系统健康监控机制答案表

编 号	答 案
(1)	进程健康监控
(2)	应用进程错误
(3)	分区初始化阶段出现的分区配置错误
(4)	分区切换时出现的错误
(5)	分区健康监控

【问题3】

采用Power PC实现系统隔离和保护的两种机制如下。

第一种是内存管理机制（MMU）。MMU能够实现逻辑地址到物理地址的转化，并且对访问权进行控制，既可以保护系统内核不受应用软件有意或无意的破坏，也可有效防止各应用软件之间的互相破坏。

第二种是TRAP系统调用机制。操作系统为实现对内核以及应用之间的保护，提供了用户态和系统态两种运行形态。操作系统内核在系统态运行，因此用户态的应用不能直接调用系统内核提供的功能接口，必须通过TRAP系统调用的方式进行。因此可以实现应用与内核之间的隔离与保护。

阅读以下Web应用系统架构设计的说明，在答题纸上回答问题1至问题3。

某公司拟开发一个市场策略跟踪与分析系统，根据互联网上用户对公司产品信息的访问情况和产品实际销售情况来追踪各种市场策略的效果。其中互联网上用户对公司产品信息的访问情况需要借助两种不同的第三方Web分析软件进行数据采集与统计，并生成不同格式的数据报表；公司产品的实际销售情况则需要通过各个分公司的产品销售电子表格或数据库进行采集与汇总。得到相关数据后，还要对数据进行分析与统计，并通过浏览器以在线的方式向市场策略制定者展示最终的市场策略效果。

在对市场策略跟踪与分析系统的架构进行设计时，公司的架构师王工提出采用面向服务的系统架构，首先将各种待集成的第三方软件和异构数据源统一进行包装，然后将数据访问功能以标准Web服务接口的形式对外暴露，从而支持系统进行数据的分析与处理，前端则采CSS等技术实现浏览器数据的渲染与展示。架构师李工则认为该系统的核心在于数据的定位、汇聚与转换，更适合采用面向资源的架构，即首先为每种数据元素确定地址，然后将各种数据格式统一转换为JSON格式，通过对JSON数据的组合支持数据的分析与处理任务，处理结果经过渲染后在浏览器的环境中进行展示。在架构评估会议上，专家对这两种方案进行综合评价，最终采用了李工的方案。

【问题1】

请根据题干描述，对市场策略跟踪与分析系统的数据源特征与数据操作方式进行分析，完成表13-10中的(1)~(3)，并用200字以内的文字说明李工方案的优点。

表13-10 系统数据源特征与数据操作方式

数据源类型	数据源特征		数据操作方式
	数据形态	数据访问实时性	
互联网用户访问信息	(1)	非实时	(3)
产品销售信息	电子表格与数据库	(2)	只读

【问题2】

请从数据获取方式、数据交互方式和数据访问的上下文无关性三个方面对王工和李工的方案进行比较，并用500字以内的文字说明为什么没有采用王工的方案。

【问题3】(6分)

表现层状态转换(REST)是面向资源架构的核心思想，请用200字以内的文字解释什么是REST，并指出在REST中将哪三种关注点进行分离。

试题分析

本题主要考查Web应用背景下多元数据的集成与处理方式，并要求考生理解数据集成和处理各种方案的优劣，进行比较和选择。

【问题1】

本问题主要考查两种不同Web数据源的数据特点。就题干描述的市场策略跟踪与分析系统特征而言，对于互联网用户访问数据源来说，该数据的数据形态一般为数据报表形态，数据为非实时性访问，数据操作方式一般为只读方式。对于产品销售信息，该数据源的数据形态一般为电子表格和数据库，数据访问方式为非实时访问。数据操作方式一般为只读方式。

【问题2】

本问题主要考查两种方案的比较，考生需要紧扣提问要点，从数据获取方式、数据交互方式和数据访问的上下文无关性三个方面进行比较。

从数据获取方式看，王工的方案需要将现有的多个系统和异构的数据源包装为服务，采用Web服务暴露数据接口，客户端需要通过服务调用获取数据，这种方法工作量大，复杂度较高。李工的

方案则绕开了复杂的功能封装，只需要明确数据的位置与标识，通过特定的网络协议直接使用标识定位并获取数据，与王工的方案相比工作量小，实现简单。

从数据交互方式看，王工的方案采用远程过程调用和异步XML消息等模式实现数据交互，这种方式适合于系统之间功能调用时进行的少量数据传输，而在进行单纯的数据访问时效率不高，稳定性也较差。李工的方案则以数据资源为核心，在对数据资源进行标识的基础上，通过标识符直接对数据资源进行访问与交互，实现简单且效率较高。

从数据访问的上下文无关性看，王工的方案中数据访问是与上下文有关的，具体表现在每次客户端进行数据请求都需要附加唯一的请求标识，并且服务端需要区分不同的客户端请求，效率较低。李工的方案中数据访问是与上下文无关的，客户端通过全局唯一的统一资源标识符（URI）请求对应的数据资源，服务端不需要区分不同的客户端请求。

【问题3】

REST（Representational State Transfer，表述性状态转移）是一种只使用HTTP和XML进行基于Web通信的技术，可以降低开发的复杂性，提高系统的可伸缩性。它的简单性和缺少严格配置文件的特性，使它与SOAP很好地隔离开来，REST从根本上来说只支持几个操作（POST、GET、PUT和DELETE），这些操作适用于所有的消息。

Web应用程序最重要的REST原则是，客户端和服务器之间的交互在请求之间是无状态的。从客户端到服务器的每个请求都必须包含理解请求所必需的信息。如果服务器在请求之间的任何时间点重启，客户端不会得到通知。此外，无状态请求可以由任何可用服务器回答，这十分适合云计算之类的环境。客户端可以缓存数据以改进性能。

在服务器端，应用程序状态和功能可以分为各种资源。资源是一个有趣的概念实体，它向客户端公开。资源的例子有：应用程序对象、数据库记录、算法等等。每个资源都使用URI得到一个唯一的地址。所有资源都共享统一的界面，以便在客户端和服务器之间传输状态。使用的是标准的HTTP方法，比如GET、PUT、POST和DELETE。Hypermedia是应用程序状态的引擎，资源表示通过超链接互联。

另一个重要的REST原则是分层系统，这表示组件无法了解它与之交互的中间层以外的组件。通过将系统知识限制在单个层，可以限制整个系统的复杂性，促进了底层的独立性。

当REST架构的约束条件作为一个整体应用时，将生成一个可以扩展到大量客户端的应用程序。它还降低了客户端和服务器之间的交互延迟。统一界面简化了整个系统架构，改进了子系统之间交互的可见性。REST简化了客户端和服务器的实现。

总结来说，REST提出了如下一些设计概念和准则：

- (1) 网络上的所有事物都被抽象为资源。
- (2) 每个资源对应一个唯一的资源标识。
- (3) 通过通用的连接件接口对资源进行操作。
- (4) 对资源的各种操作不会改变资源标识。
- (5) 所有的操作都是无状态的。

试题答案

【问题1】

表13-11 系统数据源特征与数据操作方式答案表

数据源类型	数据源特征		数据操作方式
	数据形态	数据访问实时性	
互联网用户访问信息	(1) 数据报表	非实时	(3) 只读
产品销售信息	电子表格与数据库	(2) 非实时	只读

通过对系统的数据源特征和数据操作方式进行分析可以看出，待集成的数据均为持久型数据（文件或数据库），系统对数据的访问均为只读非实时性的。针对上述应用特征，李工提出的面向资源的架构方式以对数据资源的只读访问为核心，通过数据唯一标识直接对各种数据进行访问与获取，系统架构清晰、实现简单、效率较高。

【问题2】

从数据获取方式看，王工的方案需要将现有的多个系统和异构的数据源包装为服务，采用Web服务暴露数据接口，客户端需要通过服务调用获取数据，这种方法工作量大，复杂度较高。李工的方案则绕开了复杂的功能封装，只需要明确数据的位置与标识，通过特定的网络协议直接使用标识定位并获取数据，与王工的方案相比工作量小，实现简单。

从数据交互方式看，王工的方案采用远程过程调用和异步XML消息等模式实现数据交互，这种方式适合于系统之间功能调用时进行的少量数据传输，而在进行单纯的数据访问时效率不高，稳定性也较差。李工的方案则以数据资源为核心，在对数据资源进行标识的基础上，通过标识符直接对数据资源进行访问与交互，实现简单且效率较高。

从数据访问的上下文无关性看，王工的方案中数据访问是上下文有关的，具体表现在每次客户端进行数据请求都需要附加唯一的请求标识，并且服务端需要区分不同的客户端请求，效率较低。李工的方案中数据访问是上下文无关的，客户端通过全局唯一的统一资源标识符（URI）请求对应的数据资源，服务端不需要区分不同的客户端请求。

【问题3】

REST从资源的角度来定义整个网络系统结构，分布在各处的资源由统一资源标识符（URI）确定，客户端应用程序通过URI获取资源的表现，并通过获得资源的表现使得其状态发生改变。

REST中将资源、资源的表现和获取资源的动作三者进行分离。

第13章：案例分析

试题15（2011年11月试题5）

试题15（2011年11月试题5）

阅读以下关于信息系统安全性的说明，在答题纸上回答问题1至问题3。

某大型跨国企业的IT部门一年前基于SOA（Service-Oriented Architecture）对企业原有的多个信息系统进行了集成，实现了原有各系统之间的互连互通，搭建了支撑企业完整业务流程运作的统一信息系统平台。随着集成后系统的投入运行，IT部门发现在满足企业正常业务运作要求的同时，系统也暴露出明显的安全性缺陷，并在近期出现了企业敏感业务数据泄露及系统核心业务功能非授权访问等严重安全事件。针对这一情况，企业决定由IT部门成立专门的项目组负责提高现有系统的安全性。

项目组在仔细调研和分析了系统现有安全性问题的基础上，决定首先为在网络中传输的数据提供机密性（Confidentiality）与完整性（Integrity）保障，同时为系统核心业务功能的访问提供访问控制机制，以保证只有授权用户才能使用特定功能。

经过分析和讨论，项目组决定采用加密技术为网络中传输的数据提供机密性与完整性保障。但在确定具体访问控制机制时，张工认为应该采用传统的强制访问控制（Mandatory Access Control）机制，而王工则建议采用基于角色的访问控制（Role-Based Access Control）与可扩展访问控制标记语言（eXtensible Access Control Markup Language, XACML）相结合的机制。项目组经过集体讨论，最终采用了王工的方案。

【问题1】

请用400字以内的文字，分别针对采用对称加密策略与公钥加密策略，说明如何利用加密技术为在网络中传输的数据提供机密性与完整性保障。

【问题2】

请用300字以内的文字，从授权的可管理性、细粒度访问控制的支持和对分布式环境的支持三个方面指出项目组采用王工方案的原因。

【问题3】

图13-22给出了基于XACML的授权决策中心的基本结构以及一次典型授权决策的执行过程，请分别将备选答案填入图中的（1）~（4）。

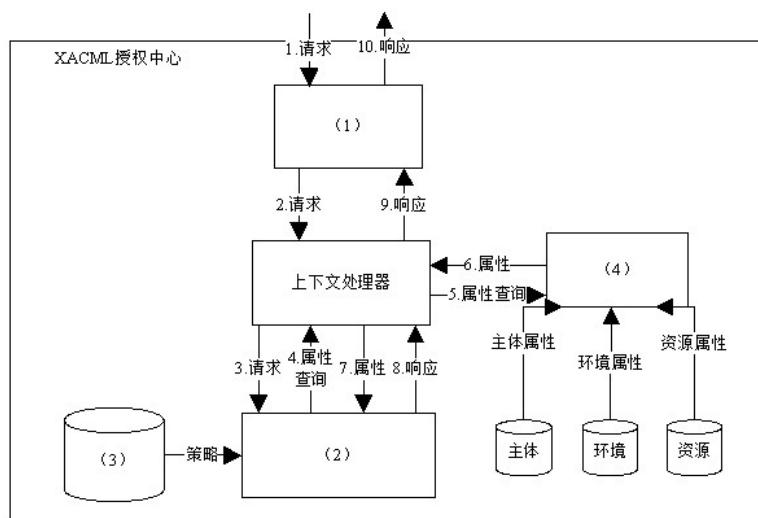


图13-22 基于XACML的授权决策中心的基本结构

备选答案：策略管理点（PAP）、策略执行点（PEP）、策略信息点（PIP）、策略决策点（PDP）

试题分析

本题考查安全保密和访问控制技术。

【问题1】

问题1考查考生对于对称与非对称加密技术的理解，以及如何应用这些技术。

对称加密技术，顾名思义，就是指加密与解密密钥是一样的。这种机制之下，发送者与接收者拥有同样的密钥，需要保密传输数据时，由发送方加密信息内容，将其发给接收方，接收方用相同的密钥解密即可。在利用对称加密技术做完整性保障时，可以将需要发送的信息产生摘要，将摘要使用对称密钥加密，然后将信息与加密后的摘要一起发送给接收方，当接收方收到消息后，将摘要的加密包解开，并对收到的信息重新产生摘要，将两个摘要进行对比，便可知信息在传递过程中完整性是否遭到破坏。

非对称加密技术是指加密与解密使用不同的密钥进行。这种密钥是成对产生的，一对密钥中，公钥加密，私钥解密；私钥加密，则公钥解密。使用该技术进行保密传输时，使用接收方的公钥加密，然后进行传输，当接收方收到信息后，使用自己的私钥解密。而当使用该技术进行完整性保障

时，也需要用信息摘要技术，先将需要发送的信息产生摘要，摘要使用发送方的私钥加密，接收方收到以后，如果能以发送方的公钥解密出相应摘要，则说明信息完整。

【问题2】

本问题考查访问控制机制及XACML标准的相关知识。从“基于SOA对企业原有多个信息系统进行了集成”可以看出系统访问控制的主要对象为集成后的服务，因此对细粒度访问控制的支持和对分布式环境的支持要求较高，而系统是支撑跨企业整体业务流程的，业务功能复杂，因而对授权的可管理性要求较高。针对上述需求，王工提出的方案比张工提出的方案具有以下的优势。

授权的可管理性：RBAC将用户与权限分离，相比MAC减小了授权管理的复杂性，更适合于大型企业级系统的安全管理。

细粒度访问控制的支持：XACML提供了统一的访问控制策略描述语言，策略表达能力强，可用来描述各种复杂的和细粒度的访问控制安全需求，更适合企业复杂业务功能的访问控制要求。

分布式环境的支持：XACML的标准性便于各子系统的协作交互，各子系统或企业业务部门可以分布管理访问控制权限，而MAC则通常需要对访问控制权限集中管理，不太适合企业基于SOA集成后的分布式系统。

【问题3】

问题3考查考生对XACML授权架构的理解。其中PEP是在具体应用环境下执行访问控制的实体，它接收外部的授权请求并生成相应的授权响应，因此（1）处应填PEP；而PDP是系统中授权决策的实体，依据XACML描述的访问控制策略以及其他属性信息进行访问控制决策，因此（2）处应填PDP；PAP系统中产生和维护安全策略的实体，因此（3）处应填PAP；PIP是获取主体、资源和环境的属性信息的实体，因此（4）处应填PIP。

试题答案

【问题1】

1.对称加密策略

（1）机密性：发送者利用对称密钥对要发送的数据进行加密，只有拥有正确相同密钥的接收者才能将数据正确解密，从而提供机密性。

（2）完整性：发送者根据要发送的数据生成消息认证码（或消息摘要），利用对称密钥对消息认证码进行加密并附加到数据上发送；接收者使用相同密钥将对方发送的消息认证码解密，并根据接收到的数据重新生成消息认证码，比较两个认证码是否相同以验证数据的完整性。

2.公钥加密策略

（1）机密性：发送者利用接收者的公钥对要发送的数据进行加密，只有拥有对应私钥的接收者才能将数据正确解密，从而提供机密性。

（2）完整性：发送者根据要发送的数据生成消息认证码（或消息摘要），利用自己的私钥对消息认证码进行加密并附加到数据上发送；接收者利用对方的公钥将对方发送的消息认证码解密，并根据接收到的数据重新生成消息认证码，比较两个认证码是否相同以验证数据的完整性。

【问题2】

（1）授权的可管理性：RBAC将用户与权限分离，与MAC相比，减小了授权管理的复杂性，更适合于大型企业级系统的安全管理。

（2）细粒度访问控制的支持：XACML提供了统一的访问控制策略描述语言，策略表达能力强，可用来描述各种复杂的和细粒度的访问控制安全需求，更适合企业复杂业务功能的访问控制要求。

(3) 分布式环境的支持：XACML的标准性便于各子系统的协作交互，各子系统或企业业务部门可以分布管理访问控制权限，而MAC则通常需要对访问控制权限集中管理，不太适合企业基于SOA集成后的分布式系统。

【问题3】

- (1) 策略执行点 (PEP)。
- (2) 策略决策点 (PDP)。
- (3) 策略管理点 (PAP)。
- (4) 策略信息点 (PIP)。

第13章：案例分析

试题16 (2012年11月试题1)

试题16 (2012年11月试题1)

阅读以下关于软件架构风格的说明，在答题纸上回答问题1和问题2。

某软件公司为其新推出的字处理软件设计了一种脚本语言，专门用于开发该字处理软件的附加功能插件。为了提高该语言的编程效率，公司组织软件工具开发部门为脚本语言研制一套集成开发环境。软件工具开发部门根据字处理软件的特点，对集成开发环境进行了需求分析，总结出以下3项核心需求：

- (1) 集成开发环境需要提供对脚本语言的编辑、语法检查、解释、执行和调试等功能的支持，并要实现各种功能的灵活组合、配置与替换。
- (2) 集成开发环境需要提供一组可视化的编程界面，用户通过对界面元素拖拽和代码填充的方式就可以完成功能插件核心业务流程的编写与组织。
- (3) 在代码调试功能方面，集成开发环境需要实现在脚本语言编辑界面中的代码自动定位功能。具体来说，在调试过程中，编辑界面需要响应调试断点命中事件，并自动跳转到当前断点处所对应的代码。

针对上述需求，软件工具开发部门对集成开发环境的架构进行分析与设计，王工认为该集成开发环境应该采用管道-过滤器的架构风格实现，李工则认为该集成开发环境应该采用以数据存储为中心的架构风格来实现。公司组织专家对王工和李工的方案进行了评审，最终采用了李工的方案。

【问题1】

请用200字以内的文字解释什么是软件架构风格，并从集成开发环境与用户的交互方式、集成开发环境的扩展性、集成开发环境的数据管理三个方面说明为什么最终采用了李工的设计方案。

【问题2】

在对软件系统架构进行设计时，要对架构需求进行分析，针对特定需求选择最为合适的架构风格，因此实际的软件系统通常会混合多种软件架构风格。请对核心需求进行分析，说明为了满足需求(2)和(3)，分别应采用何种架构风格，并概要说明采用相应架构风格后的架构设计过程。

试题分析

本题主要考查考生对软件架构风格的理解与掌握，特别是针对实际应用问题，如何采用基于软件架构风格的系统软件架构设计。软件架构风格是指描述特定软件系统组织方式的惯用模式。组织方式描述了系统的组成构件和这些构件的组织方式，惯用模式则反映众多系统共有的结构和语义。

【问题1】

本问题主要考查设计方案的比较与选型。题干明确指出从集成开发环境与用户的交互方式、集成开发环境的扩展性、集成开发环境的数据管理三个方面对两种方案进行比较，并说明选用李工方案的原因。

从集成开发环境与用户的交互方式看，根据题干描述，用户通常采用交互式的方式对脚本语言进行编辑、解释执行与调试。在这种情况下，采用以数据存储为中心的架构风格能够很好地支持交互式数据处理，而管道-过滤器架构风格则对用户的交互式数据处理支持有限。

从集成开发环境的扩展性来看，根据题干描述，要求实现各种编辑、语法检查、解释执行等多种功能的灵活组织、配置与替换。在这种情况下，采用以数据存储为中心的架构风格，以数据格式解耦各种功能之间的依赖关系，并可以灵活定义功能之间的逻辑顺序。管道-过滤器架构风格同样以数据格式解耦数据处理过程之间的依赖关系，但其在数据处理逻辑关系的灵活定义方面较差。

从集成开发环境的数据管理来看，集成开发环境需要支持脚本语言、语法树（用于检查语法错误）、可视化模型、调试信息等多种数据类型，并需要支持数据格式的转换。以数据存储为中心的架构将数据存储在统一的中心存储器中，中心存储器能够表示多种数据格式，并能够为数据格式转换提供各种支持。管道-过滤器架构风格通常只能支持有限度的数据格式，并且在数据格式转换方面的灵活性较差。

考生在回答上述问题时，不能仅仅列举教科书中对数据存储为中心的架构风格和管道-过滤器架构风格的特点描述，必须紧紧围绕题干对系统要求的描述，将系统要求与架构风格特点结合起来进行回答。

【问题2】

本问题主要考查考生如何根据应用要求选择合适的架构风格。

需求（2）要求“集成开发环境需要提供一组可视化的编程界面，用户通过对界面元素拖拽和代码填充的方式就可以完成功能插件核心业务流程的编写与组织”，这是一个对可视化开发过程的典型描述，而可视化开发的核心是如何定义并解释可视化编程语言，其核心应该是解释器架构风格。因此针对需求（2），应该采用解释器架构风格。具体来说，需要：① 为可视化编程元素及其拖拽关系定义某种语言，并描述其语法与语义；② 编写解释器对该语言进行解释；③ 生成对应的脚本语言程序。

需求（3）要求“在代码调试功能方面，集成开发环境需要实现在脚本语言编辑界面中的代码自动定位功能。具体来说，在调试过程中，编辑界面需要响应调试断点命中事件，并自动跳转到当前断点处所对应的代码。”从描述中可以看出，这是一个具有“事件触发”能力的功能描述，即由“断点命中”事件触发事先定义的“代码自动定位”功能。可以看出，这样的需求采用隐式调用架构风格最为恰当。具体来说，首先需要定义“断点在调试过程中命中”这一事件，并实现当断点命中后的屏幕定位函数。集成开发环境维护一个事件注册表结构，将该事件与屏幕定位函数关联起来形成注册表中的一个记录项。在调试过程中，集成开发环境负责监听各种事件，当“断点在调试过程中命中”这一事件发生时，集成开发环境查找事件注册表，找到并调用屏幕定位函数，从而实现脚本语言编辑界面与调试代码的自动定位。

试题答案

【问题1】

软件架构风格是指描述特定软件系统组织方式的惯用模式。组织方式描述了系统的组成构件和这些构件的组织方式，惯用模式则反映众多系统共有的结构和语义。

从集成开发环境与用户的交互方式看，用户通常采用交互式的方式对脚本语言进行编辑、解释执行与调试。在这种情况下，采用以数据存储为中心的架构风格能够很好地支持交互式数据处理，而管道-过滤器架构风格则对用户的交互式数据处理支持有限。

从集成开发环境的扩展性来看，系统核心需求要求实现各种编辑、语法检查、解释执行等多种功能的灵活组织、配置与替换。在这种情况下，采用以数据存储为中心的架构风格，以数据格式解耦各种功能之间的依赖关系，并可以灵活定义功能之间的逻辑顺序。管道-过滤器架构风格同样以数据格式解耦数据处理过程之间的依赖关系，但其在数据处理逻辑关系的灵活定义方面较差。

从集成开发环境的数据管理来看，集成开发环境需要支持脚本语言、语法树（用于检查语法错误）、可视化模型、调试信息等多种数据类型，并需要支持数据格式的转换。以数据存储为中心的架构将数据存储在统一的中心存储器中，中心存储器能够表示多种数据格式，并能够为数据格式转换提供各种支持。管道-过滤器架构风格通常只能支持有限度的数据格式，并且在数据格式转换方面的灵活性较差。

【问题2】

为了满足需求（2），应该采用解释器架构风格。具体来说，需要：① 为可视化编程元素及其拖拽关系定义某种语言，并描述其语法与语义；② 编写解释器对该语言进行解释；③ 生成对应的脚本语言程序。

为了满足需求（3），应该采用隐式调用架构风格。具体来说，首先需要定义“断点在调试过程中命中”这一事件，并实现当断点命中后的屏幕定位函数。集成开发环境维护一个事件注册表结构，将该事件与屏幕定位函数关联起来形成注册表中的一个记录项。在调试过程中，集成开发环境负责监听各种事件，当“断点在调试过程中命中”这一事件发生时，集成开发环境查找事件注册表，找到并调用屏幕定位函数，从而实现脚本语言编辑界面与调试代码的自动定位。

第 13 章：案例分析

试题17（2012年11月试题2）

试题17（2012年11月试题2）

阅读以下关于分布式系统设计的说明，在答题纸上回答问题1至问题3。

某软件公司拟开发一套电信领域的分布式系统，该系统后台多个功能模块同时运行时的计算负载较大，且需要控制不同的特定电信硬件设备，由于硬件体积和I/O端口冲突等原因，这些设备需要分散安装在多个不同的计算机系统中。该系统上线运行后将为企业最终用户提供7×24小时的不间断服务，而用户的单次接入服务往往需要后台多个模块共同协作完成。基于上述原因，该系统后台软件模块需分布在局域网内的多台计算机上。

项目组决定基于ISO的开放分布进程（ODP）规范来进行系统架构的设计与开发，近期项目组召开了多次会议，对架构设计阶段的关键问题进行了讨论分析。

【问题1】

ODP从5个标准的视点组织分析系统的架构，这些视点描述了同一系统的重要方面，请根据图13-23中不同视点所关注的核心内容，将备选的架构视点填入图中的(1)~(5)。

备选答案：技术选择架构、企业业务架构、分布式工程架构、计算接口架构、逻辑信息架构

【问题2】

在技术选择架构规划时，王工认为系统应基于现有分布式基础设施（分布式中间件）来构建，因为这样可以充分利用现有基础设施提供的各种支撑，在更短时间内构造出质量更高的分布式系统；而李工则认为可基于基本的进程间通信机制自主开发系统的支撑平台，这样可以避免对特定中间件的依赖，项目组经过认真讨论，最终采用了王工的方案。请用400字以内文字，从构件管理支持、互操作支持以及公共服务支持三个方面说明现有分布式基础设施为构建分布式系统所提供的基本支撑。

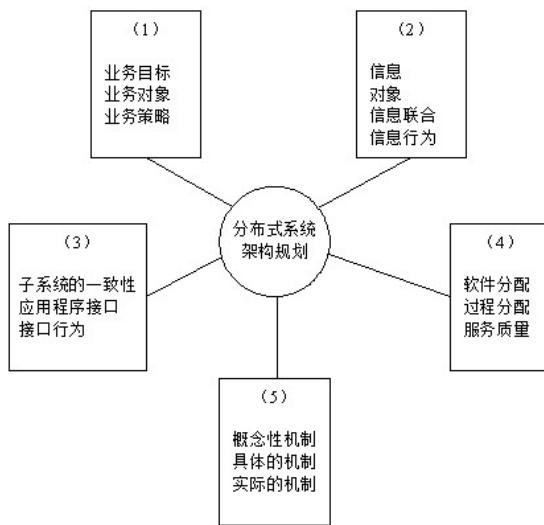


图13-23 ODP架构视点示意图

【问题3】

由于系统后台模块的分布式特性，后台分布式对象之间的互操作机制是需要考虑的核心问题之一。如图13-24所示是当前分布式基础设施中支持分布式对象互操作的基本机制，请将相应部件名称填入图中(1)~(2)；基于图13-24给出的结构，用300字以内文字说明完成一次分布式对象调用的详细步骤。

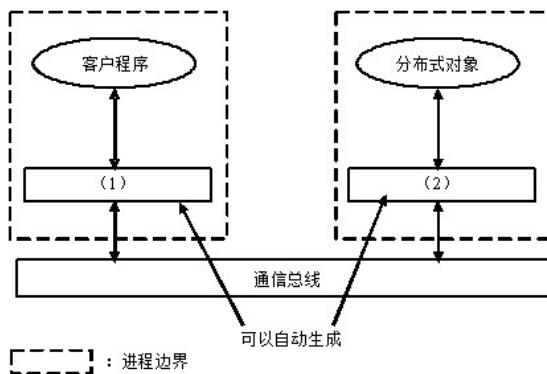


图13-24 分布式对象互操作基本机制

试题分析

本题考查考生对典型分布式系统架构设计相关基础知识的掌握，要求考生了解基于ODP的架构设计过程中不同架构视点所关注的核心工作，充分理解现有分布式基础设施为分布式系统构建提供的各种基本支撑，掌握分布式对象互操作的基本原理与交互过程。

【问题1】

本问题考查考生对ODP架构视点的理解程度。

ISO定义的ODP参考模型为分布式计算进程提供了一个基本框架，它定义了分布式系统中包括开放性、整体性、灵活性、可塑性、联合性、可操作管理性、优质服务、安全性和透明性等在内的重要性质。其体系结构视点把抽象的符号或图表（如UML图表）运用到具体的体系结构开发任务中，每个视点有具体的建模目标和系统相关者。ODP从5个标准的视点组织分析了系统的架构，描述了同一系统不同重要方面，这些视点包括企业业务架构、逻辑信息架构、计算接口架构、分布式工程架构和技术选择架构。

企业业务架构从IT的角度，对企业的业务结构、企业机构与业务的关系、企业内部的关系以及企业与外部机构的关系进行整理定义。该架构视点关注的核心内容包括业务目标、业务对象和业务策略等。

逻辑信息架构标识出系统必须知道什么，这种架构通过一个对象模型来表达，强调定义系统状态的属性。该架构视点关注的核心内容包括信息、对象、信息联合和信息行为等。

计算接口架构定义顶层的应用程序接口，这些是完全工程化的子系统边界的接口，这些接口的架构控制对于一个支持变化和控制复杂性的稳定的系统结构来说，是非常重要的。该架构视点关注的核心内容包括子系统的一致性、应用程序接口和接口行为等。

分布式工程架构定义了底层结构的需求，而独立于所选择的技术，它很好地解决了包括物理位置、系统规模可变性和通信服务质量等在内的复杂系统策略。该架构视点关注的核心内容包括软件分配、过程分配和服务质量等。

技术选择架构确定了实际的技术选择，所有其他视点都独立于这些决定，一个系统的选择过程包括初始概念机制的确认、具体机制的标识及最终实际机制的选择。该架构视点关注的核心内容包括概念性机制、具体的机制和实际的机制等。

基于上述分析，图13-23中的（1）~（5）应分别填写企业业务架构、逻辑信息架构、计算接口架构、分布式工程架构和技术选择架构。

【问题2】

本问题考查考生对现有分布式基础设施的理解。

随着软件系统规模与复杂度的不断提高，软件开发过程中复杂度高、开发周期长、可靠性保证难等突出问题并没有得到根本缓解，而分布式软件面临更大的挑战，分布式软件所运行的网络环境具有明显的分布性、开放性、演化性、异构性、并发性等特征，因此分布式软件必须解决互操作、数据交换、分布性、可行性等一系列更复杂的问题。现有的分布式基础设施典型地为三层/多层结构的分布式软件系统提供各种开发支撑，因为三层结构的分布式软件的核心为中间层，因此支撑主要集中在对中间层开发的支撑上。现有基础设施为分布式软件系统提供的基本支持与分布式软件所运行的网络环境密切相关，具体可归为提供构件运行环境、提供互操作机制与提供公共服务三个方面，基于分布式基础设施提供的支持，可以在更短时间构造出质量更高的分布式系统。

（1）提供构件运行环境：现有分布式基础设施均提供构件化的基本支持，支持方便开发与使用符合特定规范的构件（分布式对象）。中间件一般通过构件容器为构件提供基本的运行环境，具体功能一般包括管理构件的实例及其生命周期、管理构件的元信息等。

（2）提供互操作机制：因为分布式软件跨越了多台计算机，因此跨越网络的交互是每个分布式系统必须解决的首要问题，现有分布式基础设施均集成了早期远程过程/方法调用中间件的功能，提供了很强的高层通信协议以屏蔽节点的物理特性以及各节点在处理器、操作系统等方面的异构性。

基于中间件的互操作支持，开发人员在开发与调用分布式对象时，均不需自己编写处理底层通信的代码。

(3) 提供公共服务：除了互操作的支持外，现有分布式基础设施将早期各种中间件中针对分布式软件的通用支持集成于一身，以公共服务的形式提供给应用程序。公共服务又称为系统级服务，是指由中间件（应用服务器）实现的、应用程序使用的软件系统中共性程度高的功能成分。与应用程序中开发人员开发的构件实现的功能不同，公共服务通常不实现应用系统中具体的业务逻辑，而是为具体业务逻辑的实现提供共性的支持，而开发人员开发的构件则实现具体的业务逻辑。常见的公共服务包括命名服务、事务服务、安全服务、持久性服务等。

【问题3】

本问题考查考生对现有分布式对象互操作机制的理解。

各种互操作通信协议，包括RPC、IIOP、DCOM通信协议、JRMP、RMI/IIOP等，均可以帮助应用程序完成编组与解组等跨越网络通信的底层工作。这些高层通信协议尽管具体的实现细节不尽相同，但是在实现方式上与开发模式上均采用了RPC中相同的通信模型与类似的开发模式，它们采用的通信模型称为存根/框架（也称为桩/框架或代理/存根）结构。

RPC是一种广泛使用的分布式应用程序处理方法。应用程序使用RPC来远程执行一个位于不同地址空间里的过程，并且从效果上看和执行本地调用相同。要注意的是，这里的“远程”既可以指不同的计算机，也可以指同一台计算机上的不同进程。一个RPC应用可分为两个部分，分别是服务器和客户。这里的“服务器”和“客户”是指逻辑上的进程，而不是指物理计算机。支持RPC的软件如图13-25所示。

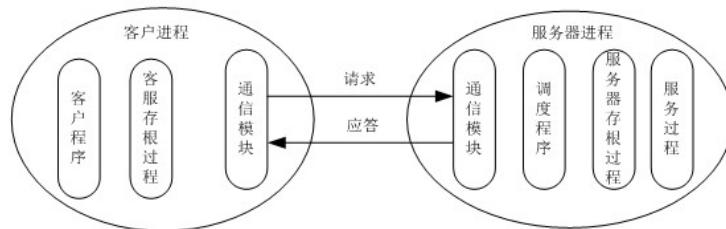


图13-25 RPC模型

在图13-25中，通信模块实现请求-应答协议，在客户和服务器之间传递请求和应答消息。服务器提供一个或多个远程过程，客户向服务器发出远程调用。服务器和客户可以位于同一台计算机，也可以位于不同的计算机，甚至运行在不同的操作系统之上。它们通过网络进行通信，相应的存根（stub）过程和运行支持提供数据转换和通信服务，从而屏蔽不同的操作系统和网络协议。存根过程用来解码请求消息中的参数、调用相应的服务过程和编码应答消息中的返回值。服务过程实现服务接口中的过程。调度程序根据请求消息中的过程标识选择一个服务器存根过程。客户存根过程（也称为客户端桩）、服务器存根过程（Skeleton,也称为服务端框架）和调度程序都可由接口编译器根据服务的接口定义而生成。

在这里，RPC通信是同步的。如果采用线程的方式，则可以进行异步调用。在RPC模型中，客户和服务器只要具备了相应的RPC接口，并且具有RPC运行支持，就可以完成相应的互操作，而不必限制于特定的服务器。因此，RPC为分布式计算提供了强有力的支持。同时，RPC所提供的基于过程的服务访问，客户与服务器进行直接连接，没有中间机构来处理请求，因此，也具有一定的局限性。例如，RPC通常需要一些网络细节以定位服务器；在客户发出请求的同时，要求服务器必须是活动的。

在该结构中，由客户端桩替客户端完成与服务端程序交互的具体底层通信工作，客户程序中的远程对象引用实际上是对本地桩的引用；而服务端框架负责替服务端完成与客户端交互的具体底层

通信工作。由于客户端桩与服务端框架分别位于客户端与服务端程序的进程内，因此开发人员开发客户端与服务端程序时只需分别与本进程内的桩与框架构件交互即可实现与远端的交互，而负责底层通信的客户端桩与服务端框架在开发过程中自动生成而非由开发人员编写，从而为开发人员省去了底层通信相关的开发工作。

在存根/框架机制的支撑下，客户程序与分布式对象之间按照以下8个步骤完成一次远程调用。

- ① 客户程序将调用请求发送给客户端桩，对于客户程序来说，桩就是服务程序在客户端的代理。
- ② 客户端桩负责将远程调用请求进行编组并发送给通信总线。
- ③ 调用请求经通信总线传送到服务端框架。
- ④ 服务端框架将调用请求解组，并分派给真正的远程对象实现（服务程序）。
- ⑤ 服务程序完成客户端的调用请求，将结果返回给服务端框架。
- ⑥ 服务端框架将调用结果编组，并发送给通信总线。
- ⑦ 调用结果经通信总线传送到客户端桩。
- ⑧ 客户端桩将调用结果解组并返回给客户程序，客户程序得到调用结果。

试题答案

【问题1】

- (1) 企业业务架构
- (2) 逻辑信息架构
- (3) 计算接口架构
- (4) 分布式工程架构
- (5) 技术选择架构

【问题2】

- (1) 构件管理支持：现有分布式基础设施一般通过构件容器为构件提供基本的运行环境；具体功能一般包括管理构件的实例及其生命周期、管理构件的元信息等。
- (2) 互操作支持：现有分布式基础设施均提供了高层通信协议，以屏蔽节点的物理特性以及各节点在处理器、操作系统、程序设计语言等方面的异构性；基于互操作支持，开发人员在开发与调用分布式对象时，均不需自己编写处理底层通信的代码。
- (3) 公共服务支持：现有分布式基础设施通常将针对分布式软件的通用支持集成于一身，以公共服务的形式提供给应用程序；其提供的常见公共服务包括命名服务、事务服务、安全服务、持久性服务等。

【问题3】

- (1) 存根/桩

- (2) 框架

或

- (1) 代理

- (2) 存根

一次远程调用的过程如下：

- ① 客户程序将调用请求发送给客户端桩，对于客户程序来说，桩就是服务程序在客户端的代理。
- ② 客户端桩负责将远程调用请求进行编组并发送给通信总线。
- ③ 调用请求经通信总线传送到服务端框架。

- ④ 服务端框架将调用请求解组，并分派给真正的远程对象实现（服务程序）。
- ⑤ 服务程序完成客户端的调用请求，将结果返回给服务端框架。
- ⑥ 服务端框架将调用结果编组，并发送给通信总线。
- ⑦ 调用结果经通信总线传送到客户端桩。
- ⑧ 客户端桩将调用结果解组并返回给客户程序，客户程序得到调用结果。

第 13 章：案例分析

试题18 (2012年11月试题3)

试题18 (2012年11月试题3)

请详细阅读有关嵌入式软件架构设计方面的描述，回答问题1和问题2。

在嵌入式系统中，软件采用开放式架构已成为新的发展趋势。软件架构设计的优劣将直接影响软件的重用和移植能力。

某软件公司主要从事宇航领域的嵌入式软件研发工作。经过二十多年的发展，其软件产品已被广泛应用于各种航天飞行器中。该公司积累了众多成熟软件，但由于当初没有充分考虑软件的架构，原有软件无法被再利用，为适应嵌入式软件技术发展需要，该公司决策层决定成立宇航嵌入式软件开放式架构研究小组，为公司完成开放式架构的定义与设计，确保公司软件资源能得到充分利用。

研究小组查阅了大量的国外资料和标准，最终将研究重点集中在了SAE AS4893《通用开放式架构（GOA）框架》标准，图13-26给出了GOA定义的架构图。

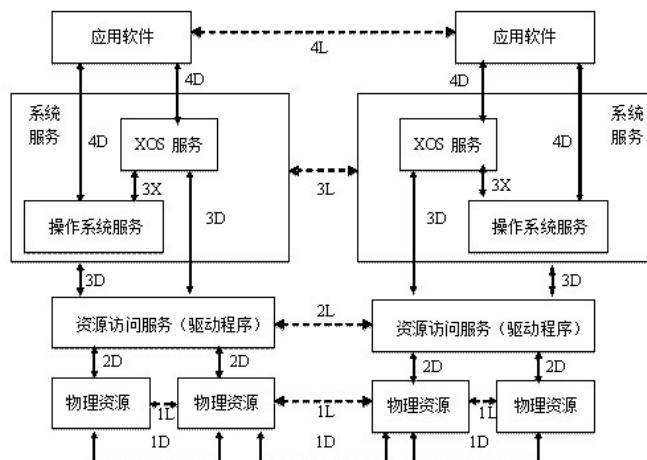


图13-26 GOA开放式架构

【问题1】

请用300字以内的文字简要说明开放式架构的四个基本特点。

【问题2】

如图13-26所示，GOA框架规定了软件、硬件和接口的结构，以在不同应用领域中实现系统功能。GOA框架规定了一组接口，其重要特点是建立了关键组件及组件间接口关系，这些接口的确定可用于支持软件的可移植性和可升级性，以满足功能的增加和技术的更新要求。除操作系统服务与

扩展操作系统之间的接口 (3X) 外, GOA将其他接口分为两类:即直接接口 (iD (i=1,2,3,...)) 和逻辑接口 (iL (i=1,2,3,...)), 直接接口定义了信息传输方式;逻辑接口定义了对等数据交换的要求, 逻辑接口没有定义真正的信息传输方式, 其传输发生在一个或多个直接接口。根据图13-22所标注的接口在框架中的具体位置, 请填写表13-12的 (1) ~ (8) 处空白。

表13-12 GOA中的接口与功能

序号	接口功能描述	接口名称
范例	实现处理器之间有效的通信方式, 操作系统服务和操作系统扩展服务之间的接口	3X
1	(1)	4D
2	一组对等的物理资源之间数据交换接口 协议的要求组成的接口, 它能实现通信链路物理资源访问 (物理资源逻辑接口)	(2)
3	一组软件 (操作系统) 访问硬件资源的服务接口。该组接口为软件与硬件资源之间定义了一个边界 (系统服务到资源访问直接接口)	(3)
4	提供在任何处理机中应用软件与其他应用软件之间的接口。也包括不同系统间的应用软件之间的接口 (应用逻辑接口)	(4)
5	(5)	1D
6	(6)	3L
7	根据对等信息 数据交换要求, 在同一处理机或不同处理机间, 资源访问服务之间的对等操作服务的接口 (资源访问服务逻辑接口)	(7)
8	由服务于硬件指令机制和寄存器使用的资源访问服务组成的接口 (资源服务到物理资源直接接口)	(8)

试题分析

本题主要考查嵌入式软件开放式架构的理解与掌握。

【问题1】

本问题主要考查开放架构的基本特点。开放架构于20世纪80年代初提出, 与开放系统概念的提出和实现密切相关。它的发展是为了适应更大规模地推广计算机的应用和计算机网络化的需求, 现仍处于继续发展和完善之中。开放架构具有应用系统的可移植性和可剪裁性、网络上各结点机间的互操作性和易于从多方获得软件的体系结构。

开放架构有如下四个基本特点。

- ① 可移植性。各种计算机应用系统可在具有开放架构特性的各种计算机系统间进行移植, 不论这些计算机是否为同种型号、同种机型。
- ② 可互操作性。如计算机网络中的各结点机都具有开放架构的特性, 则该网上各结点机间可相互操作和资源共享。
- ③ 可剪裁性。如某个计算机系统是具有开放架构特性的, 则在该系统的低档机上运行的应用系统应能在高档机上运行, 原在高档机上运行的应用系统经剪裁后也可在低档机上运行。
- ④ 易获得性。在具有开放架构特性的机器上所运行的软件环境易于从多方获得, 不受某个来源所控制。

【问题2】

本问题主要考查对SAE AS4893《通用开放式架构 (GOA) 框架》标准的理解与掌握。考生需要在对题干描述以及示意图进行认真解读的基础上填写空白。

根据题干描述, GOA框架规定了软件、硬件和接口的结构, 以在不同应用领域中实现系统功能。GOA框架规定了一组接口, 其重要特点是建立了关键组件及组件间接口关系, 这些接口的确定可用于支持软件的可移植性和可升级性, 以满足功能的增加和技术的更新要求。除操作系统服务与扩展操作系统之间的接口 (3X) 外, GOA将其他接口分为两类:即直接接口 (iD (i=1,2,3,...)) 和逻辑接口 (iL (i=1,2,3,...)), 直接接口定义了信息传输方式;逻辑接口定义了对等数据交换的要求, 逻辑接口没有定义真正的信息传输方式, 其传输发生在一个或多个直接接口。

根据上述提示，可以看出：

4D的功能是为任何处理机中的服务功能提供各应用软件互操作服务的接口（应用到系统服务的直接接口）。

1L的功能是一组对等的物理资源之间数据交换接口/协议的要求组成的接口，它能实现通信链路物理资源访问（物理资源逻辑接口）。

3D的功能是一组软件（操作系统）访问硬件资源的服务接口。该组接口为软件与硬件资源之间定义了一个边界（系统服务到资源访问直接接口）。

4L的功能是提供在任何处理机中应用软件与其他应用软件之间的接口。也包括不同系统间的应用软件之间的接口（应用逻辑接口）。

3L的功能是在同一个或不同的处理机之间，为处理机中的系统服务提供逻辑服务和远程服务的接口（系统服务逻辑接口）。

2L的功能是根据对等信息/数据交换要求。在同一处理机或不同处理机间，资源访问服务之间的对等操作服务的接口（资源访问服务逻辑接口）。

2D的功能是：由服务于硬件指令机制和寄存器使用的资源访问服务组成的接口（资源服务到物理资源直接接口）。

试题答案

【问题1】

开放架构应具有以下4个基本特点。

① 可移植性。各种计算机应用系统可在具有开放架构特性的各种计算机系统间进行移植，不论这些计算机是否为同种型号、同种机型。

② 可互操作性。如计算机网络中的各结点机都具有开放架构的特性，则该网上各结点机间可相互操作和资源共享。

③ 可剪裁性。如某个计算机系统是具有开放架构特性的，则在该系统的低档机上运行的应用系统应能在高档机上运行，原在高档机上运行的应用系统经剪裁后也可在低档机上运行。

④ 易获得性。在具有开放架构特性的机器上所运行的软件环境易于从多方获得，不受某个来源所控制。

【问题2】

表13-13 GOA中的接口与功能

序号	接口功能描述	接口名称
范例	实现处理机之间有效的通信方式，支持提供操作系统服务和操作系统扩展服务之间的接口	3X
1	(1)为任何处理机中的服务功能提供各应用软件互操作服务的接口（应用到系统服务的直接接口）	4D
2	一组对等的物理资源之间数据交换接口/协议的要求组成的接口，它能实现通信链路物理资源访问（物理资源逻辑接口）	(2) 1L
3	一组软件（操作系统）访问硬件资源的服务接口。该组接口为软件与硬件资源之间定义了一个边界（系统服务到资源访问直接接口）	(3) 3D
4	提供在任何处理机中应用软件与其他应用软件之间的接口。也包括不同系统间的应用软件之间的接口（应用逻辑接口）	(4) 4L
5	(5)物理资源与物理资源之间以及物理资源与外部环境之间的接口（物理资源到物理资源直接接口）	1D
6	(6)在同一个或不同的处理机之间，为处理机中的系统服务提供逻辑服务和远程服务的接口（系统服务逻辑接口）	3L
7	根据对等信息/数据交换要求。在同一处理机或不同处理机间，资源访问服务之间的对等操作服务的接口（资源访问服务逻辑接口）	(7) 2L
8	由服务于硬件指令机制和寄存器使用的资源访问服务组成的接口（资源服务到物理资源直接接口）	(8) 2D

试题19 (2012年11月试题4)

试题19 (2012年11月试题4)

阅读以下关于面向对象系统建模的叙述，在答题纸上回答问题1至问题3。

某软件企业为影音产品销售公司W开发一套在线销售系统，以提升服务的质量和效率。项目组经过讨论后决定采用面向对象方法开发该系统。在设计建模阶段需要满足以下设计要求。

(1) W公司经常进行促销活动。根据不同的条件（如订单总额、商品数量、产品种类等），公司可以提供百分比折扣或现金减免等多种促销方式供提交订单的用户选择。实现每种促销活动的代码量很大，且会随促销策略的不同经常修改。系统设计中需要考虑现有的促销和新的促销，而不用经常地重写控制器类代码。

(2) 该在线销售系统需要计算每个订单的税率，不同商品的税率及计算方式会有所区别。所以W公司决定在系统中直接调用不同商品供应商提供的税率计算类，但每个供应商的类提供了不同的调用方法。系统设计中需要考虑如果公司更换了供应商，应该尽可能少地在系统中修改或创建新类。

项目组架构师决定采用设计模式来满足上述设计要求，并确定从当前已经熟练掌握的设计模式中进行选择，这些设计模式包括：适配器模式（Adapter）、构造器模式（Builder）、命令模式（Command）、外观模式（Facade）、中介模式（Mediator）、原型模式（Prototype）、代理模式（Proxy）、状态模式（State）和策略模式（Strategy）等。

【问题1】

设计模式按照其应用模式可以分为三类：创建型、结构型和行为型，请用200字以内文字说明三者的作用。

【问题2】

请将项目组已经掌握的设计模式按照其作用分别归类到创建型、结构型和行为型模式中。

【问题3】

针对题目中所提出的设计要求（1）和（2），项目组应该分别选择何种设计模式？请分别用200字以内文字说明具体的解决方案。

试题分析

本题主要考查系统建模技术中的设计模式。

设计模式是前人经验的总结，它使人们可以方便地复用成功的设计和架构。当人们在特定的环境下遇到特定类型的问题，采用他人已经使用过的一些成功的解决方案，一方面可以降低分析、设计和实现的难度，另一方面可以使系统具有更好的可复用性和灵活性，随着面向对象技术的发展和广泛应用，设计模式已经成为系统架构设计师所需掌握的基本技能之一。

【问题1】

设计模式根据目的和用途不同，可以分为创建型（Creational）模式、结构型（Structural）模式和行为型（Behavioral）模式三种。创建型模式主要用于创建对象，为设计类实例化新对象提供指南；结构型模式主要用于处理类或对象的组合，对类如何设计以形成更大的结构提供指南；行为型模式主要用于描述类或对象的交互以及职责的分配，对类之间交互以及分配责任的方式提供指南。

【问题2】

创建型模式对类的实例化过程进行了抽象，能够使得软件模块做到与对象的创建和组织无关。创建型模式隐藏了对象是如何被创建的和组合在一起的，以达到使整个系统独立的目的。创建型模式包括工厂方法模式、抽象工厂模式、原型模式、单例模式和构造器模式等。

结构型模式描述如何将类或对象组合在一起形成更大的结构。结构型模式描述两种不同的事物，即类与类的实例。结构型模式包括适配器模式、桥接模式、组合模式、装饰模式、外观模式、享元模式和代理模式等。

行为型模式是对在不同的对象之间划分责任和算法的抽象，它不仅是关于类和对象的，而且是关于它们之间的相互作用的。行为型模式包括职责链模式、命令模式、解释器模式、迭代器模式、中介模式、备忘录模式、观察者模式、状态模式、策略模式、模板方法模式和访问者模式等。

【问题3】

设计要求（1）中不同的促销方法有不同的实现代码，在不同促销活动中需要能够替换实现代码，适合采用策略模式。策略模式是一种对象的行为型模式，定义一系列算法，并将每一个算法封装起来，并让它们可以相互替换。策略模式让算法独立于使用它的客户而变化，其目的是将行为和环境分隔，当出现新的行为时，只需要实现新的行为型模式。

设计要求（2）中不同商品供应商提供的税率计算类需要提供统一的方法调用接口，适合采用适配器模式。适配器模式将一个接口转换成客户希望的另一个接口，从而使接口不兼容的那些类可以一起工作。适配器模式既可以作为类结构型模式，也可以作为对象结构型模式。在类结构型模式中，通过使用一个具体类将适配器适配到目标接口中；在对象适配器模式中，一个适配器可以将多个不同的适配器适配到同一个目标。

试题答案

【问题1】

创建型模式主要用于创建对象，为设计类实例化新对象提供指南。

结构型模式主要用于处理类或对象的组合，对类如何设计以形成更大的结构提供指南。

行为型模式主要用于描述类或对象的交互以及职责的分配，对类之间交互以及分配责任的方式提供指南。

【问题2】

创建型模式：构造器模式、原型模式。

结构型模式：适配器模式、外观模式、代理模式。

行为型模式：命令模式、中介模式、状态模式和策略模式。

【问题3】

（1）策略模式

解决方案：在具有公共接口的独立类中定义每个计算。可以利用该模式创建各种促销类，它们从同一个超类继承。每个类都有相同名称的标准接口方法，用于根据订单编号计算将要折扣的金额总数。计算每个促销的内部代码对促销类来说完全不同（3分）。

（2）适配器模式

解决方案：增加一个类作为适配器，转换类的接口到客户端类期望的另一个接口。实现一个适配器类，这个类为系统的其他部分提供了一个不变的方法供调用，为了集成不同商品供应商提供的税率计算类，编写一个适配器类的子类，包含调用购买类所需的代码。该子类将系统的调用映射到某个供应商的税率计算类。如果要更换供应商，那么只需要写一个新的适配器子类，其他保持不变。

试题20 (2012年11月试题5)

试题20 (2012年11月试题5)

请详细阅读有关数据架构方面的描述，回答问题1至问题3。

某软件公司欲开发一个基于Web 2.0的大型社交网络系统。就该系统的数据架构而言，李工决定采用公司熟悉的数据架构，使用通用的商用关系型数据库，系统内部数据采用中央集中方式存储。

该系统投入使用后，初期用户数量少，系统运行平稳。6个月后，用户数出现了爆炸式增长，系统暴露出诸多问题，集中表现在：

- (1) 用户执行读/写操作时，响应时间均变得很慢；
- (2) 随着系统功能的扩充，原有数据格式发生变化，又出现新的数据格式，维护困难；
- (3) 数据容量很快超过系统原有的设计上限，数据库扩容困难；
- (4) 软件系统不断出现宕机，整个系统可用性较差。

经过多次会议讨论，公司的王工建议采用NoSQL数据库来替代关系数据库，以解决上述问题。但李工指出NoSQL数据库出现时间不长，在使用上可能存在风险。公司技术人员对NoSQL数据库产品进行了认真测试，最终决定采用NoSQL数据库来替代现有的数据库系统。

【问题1】

分别解释产生问题 (1) ~ (4) 的原因。

【问题2】

请针对问题 (1) ~ (4)，分别指出NoSQL数据库的哪些特点促使公司最终采用了NoSQL数据库。

【问题3】

请指出该系统采用NoSQL数据库时可能存在的问题。

试题分析

本题主要考查考生对NoSQL数据库的理解与掌握情况。

随着互联网web 2.0网站的兴起，非关系型的数据库成了一个极其热门的新领域，非关系数据库产品的发展非常迅速。而传统的关系数据库在应付web 2.0网站，特别是超大规模和高并发的SNS类型的web 2.0纯动态网站已经显得力不从心，暴露了很多难以克服的问题，例如：

1、对数据库高并发读写的需求

web 2.0网站要根据用户个性化信息来实时生成动态页面和提供动态信息，所以基本上无法使用动态页面静态化技术，因此数据库并发负载非常高，往往要达到每秒上万次读写请求。关系数据库应付上万次SQL查询还勉强顶得住，但是应付上万次SQL写数据请求，硬盘IO就已经无法承受了。

其实对于普通的BBS网站，往往也存在对高并发写请求的需求。

2、对海量数据的高效率存储和访问的需求

对于大型的SNS网站，每天用户产生海量的用户动态，以某SNS网站为例，该网站每个月产生2.5亿条用户动态，对于关系数据库来说，在一张2.5亿条记录的表里面进行SQL查询，效率是极其低下乃至不可忍受的。

3、对数据库的高可扩展性和高可用性的需求

在基于Web的架构当中，数据库是最难进行横向扩展的，当一个应用系统的用户量和访问量与日俱增的时候，数据库却没有办法像Web服务器和应用服务器那样简单的通过添加更多的硬件和服务节点来扩展性能和负载能力。对于很多需要提供24小时不间断服务的网站来说，对数据库系统进行升级和扩展是非常痛苦的事情，往往需要停机维护和数据迁移，为什么数据库不能通过不断的添加服务器节点来实现扩展呢？

在上面提到的“三高”需求面前，关系数据库遇到了难以克服的障碍，而对于Web 2.0网站来说，关系数据库的很多主要特性却往往无用武之地，例如：

1、数据库事务一致性需求

很多Web实时系统并不要求严格的数据库事务，对读一致性的要求很低，有些场合对写一致性要求也不高。因此数据库事务管理成了数据库高负载下一个沉重的负担。

2、数据库的写实时性和读实时性需求

对关系数据库来说，插入一条数据之后立刻查询，是肯定可以读出来这条数据的。并不要求这么高的实时性。

3、对复杂的SQL查询，特别是多表关联查询的需求

任何大数据量的web系统，都非常忌讳多个大表的关联查询，以及复杂的数据分析类型的复杂SQL报表查询，特别是SNS类型的网站，从需求以及产品设计角度，就避免了这种情况的产生。往往更多的只是单表的主键查询，以及单表的简单条件分页查询，SQL的功能被极大的弱化了。

因此，关系数据库在这些越来越多的应用场景下显得不那么合适了，为了解决这类问题的非关系数据库应运而生。

NoSQL (Not Only SQL,不仅仅是SQL) 提倡运用非关系型的数据存储，它打破了长久以来关系型数据库与ACID理论大一统的局面。NoSQL 数据存储不需要固定的表结构，通常也不存在连接操作。在大数据存取上具备关系型数据库无法比拟的性能优势。

关系型数据库中的表都是存储一些格式化的数据结构，每个元组字段的组成都一样，即使不是每个元组都需要所有的字段，但数据库会为每个元组分配所有的字段，这样的结构可以便于表与表之间进行连接等操作，但从另一个角度来说它也是关系型数据库性能瓶颈的一个因素。而非关系型数据库以键值对存储，它的结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，这样就不会局限于固定的结构，可以减少一些时间和空间的开销。

与关系型数据库相比，NoSQL数据库具有以下优点：

1、易扩展

NoSQL数据库种类繁多，但是一个共同的特点都是去掉关系数据库的关系型特性。数据之间无关系，这样就非常容易扩展。也无形之间，在架构的层面上带来了可扩展的能力。

2、大数据量，高性能

NoSQL数据库都具有非常高的读写性能，尤其在大数据量下，同样表现优秀。这得益于它的无关系性，数据库的结构简单。一般MySQL使用 Query Cache,每次表的更新Cache就失效，是一种大粒度的Cache,在针对web2.0的交互频繁的应用，Cache性能不高。而NoSQL的 Cache是记录级别的，是一种细粒度的Cache,所以NoSQL在这个层面上来说就要性能高很多了。

3、灵活的数据模型

NoSQL无需事先为要存储的数据建立字段，随时可以存储自定义的数据格式。而在关系数据库里，增删字段是一件非常麻烦的事情。如果是非常大数据量的表，增加字段简直就是一个噩梦。这点在大数据量的web2.0时代尤其明显。

4、高可用

NoSQL在不太影响性能的情况下，就可以方便的实现高可用的架构。比如Cassandra, HBase模型，通过复制模型也能实现高可用。

当然，NoSQL也存在很多缺点，例如，并未形成一定标准，各种产品层出不穷，内部混乱，各种项目还需时间来检验，缺乏相关专家技术的支持等。

【问题1】

问题1主要考查对题干(1)~(4)出现问题的分析与理解情况，考生一定要紧紧围绕大型社交网络的特点，即海量用户、大量实时操作等特征进行论述。

【问题2】

问题2承接问题1，主要考查如何结合实际场景，描述如何采用NoSQL数据库解决上述问题。注意需要有的放矢，不要发散。

【问题3】

问题3主要考查考生对NoSQL数据库缺点的理解。需要注意必须围绕题干中系统建设的特点和要求进行论述。

试题答案

【问题1】

其原因主要是以下4个方面。

(1) 用户响应时间慢。大型社交网络系统要根据用户个性化信息来实时生成动态页面和提供动态信息，所以基本上无法使用动态页面静态化技术，因此数据库并发负载非常大，往往要达到每秒上万次读/写请求。关系数据库应付上万次SQL查询还勉强可以，但是应付上万次SQL写数据请求，硬盘I/O就已经无法承受了。特别是涉及多表连接操作，会导致响应变慢。

(2) 数据格式变化。大型社交网络系统随着用户的使用，会不断地增加新的功能，导致原有数据格式发生变化，甚至出现新的数据格式。但关系数据库中采用元组方式组织数据，难以使用新型数据格式，难以维护。

(3) 数据容量超过设计上限。对于大型社交网络系统，往往会在很短时间内产生海量数据。关系数据库多采用中央数据存储，使得数据容量受限于前期设计的上限，很难实现数据容量的横向扩展。

(4) 系统可用性差：关系数据库采用中央数据存储，容易成为系统的性能瓶颈，单点故障很容易导致系统崩溃，负载过高往往导致系统出现宕机现象。

【问题2】

针对问题(1)，NoSQL数据库支持高并发数据访问，性能较高。

针对问题(2)，NoSQL数据库的数据存储结构松散，能够灵活支持多种类型的数据格式。

针对问题(3)，NoSQL数据库能够支持海量数据的存储，且易于横向扩展。

针对问题(4)，NoSQL数据库基于分布式数据存储，不存在单点故障和性能瓶颈，系统可用性高。

【问题3】

该系统采用NoSQL数据库时可能存在的问题有：

(1) NoSQL数据库的现有产品不够成熟，大多数产品处于初创期。

(2) NoSQL数据库并未形成一定的标准，产品种类繁多，缺乏官方支持。

(3) NoSQL数据库不提供对SQL的支持，学习和应用迁移成本较高。

(4) NoSQL数据库支持的特性不够丰富，现有产品提供的功能比

第14章：架构设计论文

考点分析

第14章架构设计论文

14.1考点分析

根据考试大纲，信息系统项目管理论文考试的出题范围定位于以下6个方面。

(1) 系统建模：包括定义问题与归结模型；结构化系统建模；面向对象系统建模；数据库建模。

(2) 软件架构设计：包括软件架构设计；特定领域软件架构；基于架构的软件开发方法；软件演化。

(3) 系统设计：包括处理流程设计；系统人机界面设计；文件设计；存储设计；数据库设计；网络应用系统的设计；系统运行环境的集成与设计；系统性能设计；中间件；应用服务器。

(4) 分布式系统设计：包括分布式通信协议的设计；基于对象的分布式系统设计；基于Web的分布式系统设计；基于消息和协同的分布式系统设计；异构分布式系统的互操作性设计。

(5) 系统的可靠性分析与设计：包括系统的故障模型和可靠性模型；提高系统可靠性的措施；系统的故障对策和系统的备份与恢复。

(6) 系统的安全性和保密性设计：包括系统的访问控制技术；数据的完整性；数据与文件的加密；通信的安全性；系统的安全性设计。

在历年考试试题中，系统架构设计师论文试题一共4道，由考生选择其中一道试题做答，满分75分。历年考试的试题如表13-1所示。

表13-1系统架构设计师论文试题分布表

考试时间	考试题目	知识点说明
2009. 11	论基于DSSA的软件架构设计与应用	领域分析、领域设计、领域实现
	论信息系统建模方法	常见的信息系统建模方法及选择依据
	论基于REST服务的Web应用系统设计	REST的特点及其应用
	论软件可靠性设计与应用	可靠性的影响因素及可靠性设计技术
2010. 11	论软件的静态演化和动态演化及其应用	静态演化和动态演化的特点、优缺点及应用场合
	论数据挖掘技术的应用	数据挖掘的主要任务、数据挖掘的方法及实际应用效果
	论大规模分布式系统缓存设计策略	常见的缓存工作模式及应用场景
	论软件可靠性评价	可靠性模型及其选择依据
2011. 11	论模型驱动架构在系统开发中的应用	模型驱动架构（Model Driven Architecture, MDA）的优缺点，及如何利用该方法进行相关开发工作
	论企业集成平台的架构设计	企业集成平台的基本功能和相关的关键技术
	论企业架构管理与应用	企业架构管理所包含的内容及主要活动
	论软件需求获取技术及应用	需求获取技术以及每种技术的基本方法

续表

考试时间	考试题目	知识点说明
2012. 11	论基于架构的软件设计方法及应用	要求了解ABSD的基本阶段
	论企业应用系统的数据持久层架构设计	每种数据访问模式的主要内容
	论决策支持系统的开发与应用	建立决策支持系统时需解决的一般关键问题
	论企业信息化规划的实施与应用	企业信息化规划目标及规划的具体内容

第 14 章：架构设计论文

试题1 (2009年11月试题1)

14.2 试题精解

试题1 (2009年11月试题1)

论基于DSSA的软件架构设计与应用

软件架构设计的一个重要课题是如何解决软件重用问题。特定领域软件架构 (Domain Specific Software Architecture,DSSA) 是一种有效实现特定领域软件重用的手段。按照Tracz的说法，DSSA就是一个特定的问题领域中由领域模型、参考需求、参考架构等组成的开发基础架构，其目标就是支持一个特定领域中多个应用的生成。

DSSA的基本活动包括领域分析、领域设计和领域实现。领域分析的主要目的是获得领域模型，领域模型描述领域中系统之间共同的需求，即领域需求；领域设计的主要目标是获得DSSA,DSSA描述领域模型中表示需求的解决方案：领域实现的主要目标是依据领域模型和DSSA开发和组织可重用信息。

请围绕"基于DSSA的软件架构设计与应用"论题，依次从以下三个方面进行论述。

- (1) 概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。
- (2) 就你所熟悉的领域，请给出针对该特定领域，在基于DSSA的软件设计开发中所涉及的领域模型、参考需求和参考架构以及相应的支持环境或设施。
- (3) 具体阐述你参与管理和开发的项目中使用DSSA的情况，包括领域分析、领域设计和领域实现等活动是如何具体实施的，最终实际效果如何。

写作要点

这是一道关于DSSA的试题。DSSA的概念以及基本活动，题目中都进行了简要的说明。值得注意的是在论文的写作选材方面，作者的项目需要具备一定的行业特色。如电力行业，通信行业，铁路系统等。关于DSSA所具备的特征以及如何用DSSA的体系来具体创建系统的软件架构。

本文写作要点，大致包括以下几个方面。

- (1) 简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。
- (2) 应结合自己所熟悉的领域，定义领域范围，确定领域应用需要满足的用户需求；定义领域特定的元素、领域字典和领域术语；定义领域特定的设计和实现需求约束；在此基础上，定义领域模型，产生该领域的参考架构，并说明构件的语法和语义；最后，产生、搜集可重用的产品单元，为DSSA增加构件，为问题域实现新应用提供支持。这个DSSA的建立过程是并发、递归和反复进行的。

所给出的DSSA应该具备以下4个方面的特征：

一个严格定义的问题域和/或解决域；
具有普遍性，使其可以用于领域中某个特定应用的开发；
对整个领域能有合适程度的抽象；
具备该领域固定的、典型的在开发过程中的可重用元素。
(3) 需要结合项目实际，指出在架构设计时使用DSSA的情况，包括领域分析、领域设计和领域实现等活动是如何具体实施的，要给出实际的效果并进行分析。

第 14 章：架构设计论文

试题2 (2009年11月试题2)

试题2 (2009年11月试题2)

论信息系统建模方法

系统模型在软件开发中扮演着重要的角色。可为已有的系统创建模型，以便更好地理解这些系统；也可以针对待开发的系统创建模型，作为记录业务需求或技术设计的方法。模型是建立信息系统的基础。恰当地运用信息系统建模方法，是成功地进行软件开发的一个关键环节。

请围绕"论信息系统建模方法"论题，依次从以下三个方面进行论述。

- (1) 概要叙述你参与管理和开发的信息系统项目以及你在其中所承担的主要工作。
- (2) 论述常见的信息系统建模方法的主要内容（包括每种建模方法的核心思想以及所创建的模型）。
- (3) 具体阐述你参与管理和开发的项目中选择使用的信息系统建模方法以及选择该方法的原因，给出具体的实施过程和实施效果。

写作要点

这是一道关于信息系统的建模的试题。值得注意的是，有不少考生把该试题的建模方法理解成了开发模型与架构，这都是不正确的。信息系统的建模方法可以分为：面向过程的建模、面向数据的建模、面向信息的建模、面向决策的建模和面向对象的建模五种。

面向过程的建模方法是把过程看作系统模型的基本部分，数据是随着过程而产生的。最有影响的面向过程的设计方法是Yourdon设计法。

面向数据的建模方法把模型的输入/输出看成是最重要的，因此，首先定义的是数据结构，而过程模块是从数据结构中导出的，即功能跟随数据。最有影响的面向数据的设计方法是Jackson设计法。

面向信息的建模方法与面向数据的建模方法的区别就是信息和数据的区别。信息和数据都是信息系统中最基本的术语，数据是指记载下来的事实，是客观实体属性的值，而信息是构成一定含义的一组数据。面向信息建模方法是从整个系统的逻辑数据模型开始的，通过一个全局信息需求视图来说明系统中所有基本数据实体及其相互关系，然后，在此基础上逐步构造整个模型，信息模型记录系统运作所需的信息实体，如：人员、地点、事物、观念等，为分析现行系统提供信息的图形化表示。数据建模的目的是设计和实现满足系统信息需求的数据库结构，即数据建模支持系统设计。

决策支持系统由数据库、模型库和各自的管理系统组成。决策支持系统模型需要反映的问题是系统的决策制定原则和机理、系统的组织机构和人员配置。通过对决策系统的建模，企业的领导可以对企业有一个细致的了解，从而发现其中问题。如组织结构臃肿，职权划分不清，权力范围不合理等，据此进行相应的改革。比较成熟的决策支持系统建模方法有Petri网和GRAI法。

面向对象的分析方法是利用面向对象的信息建模概念，如实体、关系、属性等，同时运用封装、继承、多态等机制来构造模拟现实系统的方法。传统的结构化设计方法的基本点是面向过程，系统被分解成若干个过程。而面向对象的方法是采用构造模型的观点，在系统的开发过程中，各个步骤的共同的目标是建造一个问题域的模型。在面向对象的设计中，初始元素是对象，然后将具有共同特征的对象归纳成类，组织类之间的等级关系，构造类库。在应用时，在类库中选择相应的类。

在完成该论文时，首先应对上面的方法进行介绍，然后说明自己的系统中采用了哪种方法，具体是怎么样来使用这种方法的。

第 14 章：架构设计论文

试题3（2009年11月试题3）

试题3（2009年11月试题3）

论基于REST服务的Web应用系统设计

REST (REpresentational State Transfer) 是指从几种基于网络的架构风格衍生出来的一种混合架构风格，它是目前互联网的核心架构风格。基于REST服务 (RESTful Service) 的Web应用系统设计任务主要包括：识别并设计REST风格的服务，采用面向服务的思想进行REST服务集成。采用这种方法设计的Web应用系统能够结合REST风格和面向服务思想的优点，近年来受到了广泛的关注。

请围绕“基于REST服务的Web应用系统设计”论题，依次从以下三个方面进行论述。

- (1) 概要叙述你参与实施的Web应用系统开发项目以及你所承担的主要工作。
- (2) 简要叙述与传统的Web服务相比，采用REST服务构建的Web应用具有哪些优势和不足。
- (3) 阐述你在设计基于REST服务的Web应用系统时遇到了哪些问题，如何解决。

写作要点

(1) 论文中要具体介绍项目的总体需求（特别是质量属性需求）、Web应用系统的逻辑与物理拓扑结构、采用的技术等内容和承担的实际工作。

(2) REST (Representational State Transfer) 是指从几种基于网络的架构风格衍生出来的一种混合架构风格，目前Web的体系结构正是基于REST风格的。REST风格中的特点是客户端/服务器、无状态、缓存、统一接口、分层系统和按需代码。REST组件通过一种数据格式转移资源的表述进行通信，可以基于接收者的能力和期待的内容，以及资源的性质动态地选择不同的表述。

与传统的Web服务相比，REST服务主要有以下优势。

REST服务基于W3C/IETF的标准与规范（包括HTTP、XML、URI和MIME等），其实现技术简单、成熟。

REST服务基于URI和超链接技术，不需要通过集中式的服务信息仓库即可发现服务资源。

REST服务支持缓存，具有无状态的特性，这些使得REST服务能够支持大量客户端，构建的应用系统具有较强的伸缩性。

REST服务基于轻量级的Web框架，只需要基本的开发工具支持，构建过程简单且成本较低。

REST服务的测试相对简单，采用浏览器即可完成服务功能测试。

与传统的Web服务相比，REST服务主要存在如下不足。

REST服务倡导的REST风格与实际实现尚存在一定差距。例如高层REST服务倡导使用GET、PUT、POST和DELETE所有4个统一接口，在REST实现部分通常只能采用GET和POST接口，因为大多数的代理和防火墙会屏蔽其他接口；并且XHTML表单中只能使用GET和POST接口。

REST服务要求所有的输入参数都必须在URI中传递，这样会产生对参数容量大小的限制（目前的大小是4KB）。如果超出该数量，会导致HTTP协议错误（错误代码414:Request-URI too long）。

在URI中表达复杂类型的参数比较困难，且目前对URI中的参数不存在一种公认的编组（marshalling）和解编（un-marshalling）方法。

（3）进行基于REST服务的Web应用系统的设计时可能遇到的问题包括：如何识别并设计REST风格服务；构建REST服务的运行环境，包括HTTP服务器与应用服务器选型等；富客户端表现方式及编程语言的选择；系统逻辑与物理拓扑结构的分析与设计等。

第14章：架构设计论文

试题4（2009年11月试题4）

试题4（2009年11月试题4）

论软件可靠性设计与应用

目前在企业中，以软件为核心的产品得到了广泛的应用。随着系统中软件部分比例的不断增加，使得系统对软件的依赖性越来越强，对软件的可靠性要求也越来越高。软件可靠性与其他质量属性一样，是衡量软件架构的重要指标。

软件工程中已有很多比较成熟的设计技术，如结构化设计、模块化设计、自顶向下设计等，这些技术为保障软件的整体质量发挥了重要作用。在此基础上，为了进一步提高软件的可靠性，通常会采用一些特殊的设计技术，即软件可靠性设计技术。

在软件可靠性工程体系中，包含有可靠性模型与预测、可靠性设计和可靠性测试方法等。实践证明，保障软件可靠性最有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制。

请围绕“软件可靠性设计与应用”论题，依次从以下三个方面进行论述。

（1）概要叙述你参与实施的软件开发项目以及你所承担的主要工作。

（2）简要叙述影响软件可靠性的因素有哪些。

（3）阐述常用的软件可靠性设计技术以及你如何应用到实际项目中，效果如何。

写作要点

这是一道关于软件可靠性设计的试题，主要考查影响软件可靠性的因素以及软件可靠性设计技术与应用。本题需要特别注意的一点是：文章不能以“系统可靠性设计与应用”为主线组织内容。“系统可靠性设计”与“软件可靠性设计”的区别在于：系统可靠性包括了系统硬件的可靠性和系统软件的可靠性。

软件的可靠性设计技术主要有：恢复块方法、N-版本程序设计、防卫式程序设计。

恢复块设计就是选择一组操作作为容错设计单元，从而把普通的程序块变为恢复块。一个恢复块中包含若干功能相同、设计差异的程序块，每一时刻有一个程序块处于运行状态，一旦某程序块出现故障，则用备份程序块予以替换。

N-版本程序设计的核心是通过设计出多个模块或不同版本，对于相同初始条件和相同输入的操作结果进行多数表决（防止因其中某一软件模块/版本的故障而提供了错误的服务，以实现软件容错）。冗余设计的思路来源于硬件系统，但有所不同。软件冗余设计技术是采用多种不同路径、不同算法或不同实现方法的模块或系统作为备份，在出现故障时进行替换，维持系统的正常运行。

防卫式程序设计是一种不采用任何一种传统的容错技术就能实现软件容错的方法，对于程序中存在的错误和不一致性，防卫式程序设计的基本思想是通过在程序中包含错误检查代码和错误恢复代码，使得一旦错误发生，程序能撤销错误状态，恢复到一个已知的正确状态中去。其实现策略包括错误检测，破坏估计和错误恢复三个方面。

第 14 章：架构设计论文

试题5 (2010年11月试题1)

试题5 (2010年11月试题1)

论软件的静态演化和动态演化及其应用

软件演化 (Software Evolution) 是指软件在其生命周期内的更新行为和过程。演化是一系列贯穿软件生命周期始终的活动，系统需求改变、功能实现增强、新功能加入、软件架构改变、软件缺陷修复、运行环境改变均要求软件系统能够快速适应变化，具有较强的演化能力。软件静态演化 (Static Evolution) 和动态演化 (Dynamic Evolution) 是目前软件演化的两种重要类型。

请围绕“软件的静态演化和动态演化及其应用”论题，依次从以下三个方面进行论述。

(1) 概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

(2) 请分别对软件静态演化和动态演化的特点进行论述，说明两种软件演化类型各自的优缺点及其应用场合，并举例说明各自的常见演化技术手段。

(3) 具体阐述你参与管理和开发的项目中所进行的软件演化活动的特点、演化的类型，以及所采取的对应演化技术手段，说明具体实施过程以及实际应用的效果。

写作要点

一、简要叙述所参与管理和开发的软件项目，并明确指出你在其中承担的主要任务和开展的主要工作。

二、软件演化可分为静态演化和动态演化两种情形。

(1) 静态演化 (Static Evolution)。静态演化是指软件在停机状态下的演化。其优点是不用考虑运行状态的迁移，同时也没有活动的进程需要处理。然而停止一个应用程序就意味着中断它提供的服务，造成软件暂时失效。

软件静态演化是指发生在应用程序停止时的软件修改和更新，即一般意义上的软件维护和升级。静态演化的优点是没有状态迁移或活动线程的问题要解决，缺陷是停止应用程序意味着停止其所提供的服务，也就是使软件系统暂时失效。在软件交付之后，静态演化（类似于一般意义上的软件维护）就成为软件变更的一个常规过程。变更可以是一种更正代码错误的简单变更，也可以是更正设计错误的较大范围的变更，还可以是对描述错误进行修正或提供新需求这样的重大改进。有三种不同的软件维护：改正性维护、适应性维护和完善性维护。维护过程一般包括变更分析、版本规划、系统实现和向客户交付系统等活动。

在面向对象技术中，使用子类型方法来扩展程序，它适合于软件静态演化和代码重用。子类型化一个类意味着保留父类中的参数和方法，并尽可能地增加新的参数和方法。另外，使用重载和多态性作为主要的演化机制。实际上，建立类的新版本，最简单的机制是创建它的子类，然后重载需要变更的方法，最后，使用多态性调用新创建的方法。在基于构件的软件技术中，构件采取接口和实现相分离技术，构件之间只能通过接口进行通信，这使得具有兼容接口的不同构件实现可以相互取代，从而成为软件静态演化的一条途径。

(2) 动态演化 (Dynamic Evolution)。动态演化是指软件在执行期间的软件演化。其优点是软件不会存在暂时的失效，有持续可用性的明显优点。但由于涉及状态迁移等问题，比静态演化从技术上更难处理。

动态演化是最复杂也是最有实际意义的演化形式。动态演化使得软件在运行过程中，可以根据应用需求和环境变化，动态地进行软件的配置、维护和更新，其表现形式包括系统元素数目的可变性、结构关系的可调节性和结构形态的动态可配置性。软件的动态演化特性对于适应未来软件发展的开放性、动态性具有重要意义。

动态演化是指软件在运行期间的演化。在许多重要的应用领域中，例如金融、电力、电信及空中交通管制等，系统的持续可用性是一个关键性的要求，运行时刻的系统演化可减少因关机和重新启动而带来的损失和风险。此外，越来越多的其他类型的应用软件也提出了运行时刻演化的要求，在不必对应用软件进行重新编译和加载的前提下，为最终用户提供系统定制和扩展的能力。

动态演化可分为两种类型：预设的和非预设的。在Web环境中，软件应用常常需要处理多种类型的信息，因此它们常被设计为可以动态下载并安装插件以处理当前所面临的新类型的信息；而分布式Web应用也需要增减内部处理节点的数目，以适应多变的负载。这些动态改变都是软件设计者能够预先没想到的，可实现为系统的固有功能。另有一些必须对系统配置进行修改和调整的情况是直到系统投入运行以后才发现的，这就要求系统能够处理在原始设计中没有完全预料到的新需求。这种情况下一般需要关闭整个系统，重新开发、重新装入并重新启动系统。然而，为了进行局部的修改而关闭整个系统在某些情况下是不允许的（例如，关键运行系统）或者代价太高。精心设计的动态演化技术可以在不关闭整个系统的前提下修改系统的结构配置，并尽量使未受影响的部分继续工作以提高系统的可用度。

为支持软件的动态演化性，已在语言、机制和环境等方面做了大量工作。在程序语言的层次上，引进各种机制以支持软件动态演化，例如动态装载技术允许增加代码到已运行的程序中，延迟绑定是在运行时而不是编译时决定类和对象的绑定。Java hotswap允许在运行时改变方法：当一个方法终止时这个方法的新版本可以有效地替换旧版本，在类层次上代码的二进制兼容被支持。Gilgul

语言也允许更换运行时对象。但程序语言层次上的动态演化机制仅局限于函数、类方法和对象等小粒度的替换，只支持预设的有限变更，变更由事件触发。

通过标准化运行级构件的规约，依靠构件运行平台（中间件平台）提供的基础设施，使软件在构件层次上的动态演化成为可能。中间件中具有的如命名服务、反射技术和动态适配等机制，为运行态构件的动态替换和升级提供支撑，从而推动了软件动态演化的发展。命名服务就是给构件实例提供一个名称，以便客户通过这些名称来获取构件实例。对工业标准构件EJB和CORBA构件的引用都可以通过中间件平台的命名服务进行。同一构件标识可以被映射到多个构件实例，从而根据具体情境对某一名字的构件引用导向到不同的构件实例。反射技术是系统的一种自描述（self-representation）和自推理的技术，它提供了关于自身行为的表示，这种表示可以被检查和调整，且与它所描述的系统行为是因果相联（causally connected）的。因果相联，意味着对自表示的改动将立即反映在系统的实际状态和行为中，反之亦然。将反射性引入中间件能够以可控的方式开放平台内部的实现，从而提高中间件的定制能力和运行时的适应能力。动态适配机制中比较著名的是CORBA提供的动态接口服务：动态调用接口DII和动态骨架接口DSI。前者支持动态客户请求调用，而后者支持将请求动态指派（Dispatch）给构件。因此，软件构件化技术使得软件具有良好的构造性，软件演化的粒度更大。中间件技术则为基于构件的软件动态演化提供了坚实的基础设施和方便的操作界面。

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目中所进行的软件演化活动的特点、演化的类型，以及所采取的对应演化技术手段。要给出实施软件演化活动的具体过程、方法以及对实际应用效果的分析。

第 14 章：架构设计论文

试题6（2010年11月试题2）

试题6（2010年11月试题2）

论数据挖掘技术的应用

随着信息技术的高速发展，各组织机构积累的数据量急剧增长。如何从海量的数据中提取有用的知识成为当务之急。数据挖掘（Data Mining）就是为顺应这种需要应运而生发展起来的数据处理技术，是知识发现的关键步骤。数据挖掘就是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。

请围绕“论数据挖掘技术的应用”论题，依次对以下三个方面进行论述。

- (1) 概要叙述你参与分析和开发的软件项目以及你所担任的主要工作。
- (2) 数据挖掘的主要任务是什么？具体论述你在项目中使用数据挖掘技术所解决的问题。
- (3) 数据挖掘的方法主要有哪些？分析并讨论你所选择的数据挖掘方法，简述其具体实现过程和实际应用效果。

写作要点

一、结合自己所参与的软件项目，概要介绍该项目的背景及主要内容，并明确指出在其中所承担的主要任务和开展的主要工作。

二、数据挖掘的任务主要是关联分析、聚类分析、分类、预测、时序模式和偏差分析等。

1、关联分析。两个或两个以上变量的取值之间存在某种规律性，就称为关联。数据关联是数据库中存在的一类重要的、可被发现的知识。关联分析的目的是找出数据库中隐藏的关联网。一般用支持度和可信度两个阈值来度量关联规则的相关性。

2、聚类分析。聚类是把数据按照相似性归纳成若干类别，同一类中的数据彼此相似，不同类中的数据相异。聚类分析可以建立宏观的概念，发现数据的分布模式，以及可能的数据属性之间的相互关系。

3、分类。分类就是找出一个类别的概念描述，它代表了这类数据的整体信息，即该类的内涵描述，并用这种描述来构造模型，一般用规则或决策树模式表示。分类是利用训练数据集通过一定的算法而求得分类规则。分类可被用于规则描述和预测。

4、预测。预测是利用历史数据找出变化规律，建立模型，并由此模型对未来数据的种类及特征进行预测。预测的精度和不确定性被重点关注，通常用预测方差来度量。

5、时序模式。时序模式是指通过时间序列搜索出的重复发生概率较高的模式。与回归一样，它也是用已知的数据预测未来的值，但这些数据的区别是变量所处时间的不同。

6、偏差分析。在偏差中包括很多有用的知识，数据库中的数据存在很多异常情况，发现数据库中数据存在的异常情况是非常重要的。偏差检验的基本方法就是寻找观察结果与参照之间的差别。

论文中须明确指出自己在该项目应用数据挖掘技术所要解决的具体问题是什么。

三、主要的数据挖掘方法

1.神经网络方法

神经网络由于本身良好的鲁棒性、自组织自适应性、并行处理、分布存储和高度容错等特性非常适合解决数据挖掘的问题，因此近年来越来越受到人们的关注。典型的神经网络模型主要分3大类：以感知机、BP反向传播模型、函数型网络为代表的，用于分类、预测和模式识别的前馈式神经网络模型；以hopfield的离散模型和连续模型为代表的，分别用于联想记忆和优化计算的反馈式神经网络模型；以art模型、koholon模型为代表的，用于聚类的自组织映射方法。神经网络方法的缺点是“黑箱”性，人们难以理解网络的学习和决策过程。

2.遗传算法

遗传算法是一种基于生物自然选择与遗传机理的随机搜索算法，是一种仿生全局优化方法。遗传算法具有的隐含并行性、易于和其他模型结合等性质使得它在数据挖掘中被加以应用。

3.决策树方法

决策树是一种常用于预测模型的算法，它通过将大量数据有目的分类，从中找到一些有价值的信息。它的主要优点是描述简单，分类速度快，特别适合大规模的数据处理。最有影响和最早的决策树方法是由quinlan提出的著名的基于信息熵的id3算法。它的主要问题是：id3是非递增学习算法；id3决策树是单变量决策树，复杂概念的表达困难；同性间的相互关系强调不够；抗噪性差。针对上述问题，出现了许多较好的改进算法，如 schlimmer和fisher设计了id4递增式学习算法等。

4.粗集方法

粗集理论是一种研究不精确、不确定知识的数学工具。粗集方法有几个优点：不需要给出额外信息；简化输入信息的表达空间；算法简单，易于操作。粗集处理的对象是类似二维关系表的信息表。目前成熟的关系数据库管理系统和新发展起来的数据仓库管理系统，为粗集的数据挖掘奠定了坚实的基础。

5.覆盖正例排斥反例方法

它是利用覆盖所有正例、排斥所有反例的思想来寻找规则。首先在正例集合中任选一个种子，到反例集合中逐个比较。与字段取值构成的选择种子相容则舍去，相反则保留。按此思想循环所有正例种子，将得到正例的规则。比较典型的算法有michalski的aq11方法等。

6.统计分析方法

在数据库字段项之间存在两种关系：函数关系（能用函数公式表示的确定性关系）和相关关系（不能用函数公式表示，但仍是相关确定性关系），对它们的分析可采用统计学方法，即利用统计学原理对数据库中的信息进行分析。可进行常用统计（求大量数据中的最大值、最小值、总和、平均值等）、回归分析（用回归方程来表示变量间的数量关系）、相关分析（用相关系数来度量变量间的相关程度）、差异分析（从样本统计量的值得出差异来确定总体参数之间是否存在差异）等。

7.模糊集方法

利用模糊集合理论对实际问题进行模糊评判、模糊决策、模糊模式识别和模糊聚类分析。系统的复杂性越高，模糊性越强，一般模糊集合理论是用隶属度来刻画模糊事物的亦此亦彼性的。

论文中必须明确指出使用了上述七种方法中的哪一种或哪几种数据挖掘方法，并给出该方法的具体实现过程；分析所选择的数据挖掘方法的实现效果。

第 14 章：架构设计论文

试题7 (2010年11月试题3)

试题7 (2010年11月试题3)

论大规模分布式系统缓存设计策略

大规模分布式系统通常需要利用缓存技术减轻服务器负载、降低网络拥塞、增强系统可扩展性。缓存技术的基本思想是将客户最近经常访问的内容在缓存服务器中存放一个副本，当该内容下次被访问时，不必建立新的数据请求，而是直接由缓存提供。良好的缓存设计，是一个大规模分布式系统能够正常、高效运行的必要前提。在进行大规模分布式系统开发时，必须从一开始就针对应用需求和场景对系统的缓存机制进行全面考虑，设计一个可伸缩的系统缓存架构。

请围绕“大规模分布式系统缓存设计策略”论题，依次从以下三个方面进行论述。

- (1) 概要叙述你参与实施的大规模分布式系统开发项目以及你所担任的主要工作。
- (2) 从不同的用途和应用场景考虑，请详细阐述至少两种常见的缓存工作模式，并说明每种工作模式的适应场景。
- (3) 阐述你在设计大规模分布式系统的缓存机制时遇到了哪些问题，如何解决。

写作要点

一、论文中要具体介绍项目的总体需求（特别是应用需求中对缓存机制的要求）、系统的逻辑与物理架构、采用的技术等内容和担任的实际工作。

二、从不同的用途和应用场景来考虑，大体上可以将缓存分为三种工作模式，即单实例缓存模式（Single Instance）、复制模式（Replication Cache）和分区模式（Partition Cache）。每种工作模式都有其适应的场景和优缺点。

1、单实例模式。单实例模式是一种较为简单的缓存模式，多个应用服务器共享一个中央的缓存服务器。通过共享缓存的数据，能够极大地提高系统的性能。该模式的主要限制在于缓存服务器的内存大小和节点增加之后服务器的处理能力和网络带宽。该模式的适应场景是：对缓存的要求比较简单；系统的吞吐量和数据量不大；性能要求不高。

2、复制模式。复制模式将缓存的数据复制到多台机器上，对于单一缓存服务器性能出现问题的情况下，可以通过缓存复制的方式将压力分解到多个缓存服务器。该模式的工作原理是：缓存客户端可以访问自己的缓存服务器，多个缓存服务器之间的数据是彼此同步的，对于性能要求更高的场景，这样的部署架构能够获得更高的吞吐能力。该模式的适应场景是：数据量不是特别大；需要极高的性能；数据改动的频率不是特别大。

3、分区模式。当需要缓存的数据已经超过一台服务器的内存上限时，可以考虑采用分区模式对数据进行线性缩放，也就是通过增加缓存服务器来解决数据增长和压力增加的情况。在分区模式中，其架构是无共享架构（Shared Nothing Architecture, SNA），每个节点之间数据彼此独立，一个节点出现故障后不会影响到其他节点。在出现某个节点宕机或者本地故障的情况下，致使这部分的分区缓存无法使用，并不妨碍其他数据节点数据的正常工作。该模式的适应场景是：总体数据量较大，已经超出了单个缓存服务器的内存上限；系统缓存要求具有很大的可伸缩性；客户端数量庞大，单个客户端对缓存数据的数据量要求不大。

三、进行大规模分布式系统缓存机制设计时可能遇到的问题包括缓存服务器的工作模式如何选择；高可用性的设计考虑；缓存一致性与分布式算法；对象状态同步的考虑；缓存钝化/激活/过期和初始化等。

第 14 章：架构设计论文

试题8 (2010年11月试题4)

试题8 (2010年11月试题4)

论软件可靠性评价

软件可靠性评价是指选用和建立合适的可靠性数学模型，运用统计技术和其他手段，对软件可靠性测试和系统运行期间的软件失效数据（也可能包含软件生命周期内其他可靠性数据）进行处理，并评估和预测软件可靠性的过程。

软件可靠性评价是软件可靠性活动的重要组成部分，既可以在软件开发过程实施，也可以针对最终软件系统实施。软件可靠性评价的难点在于软件可靠性模型的选择和软件可靠性数据的收集与处理。

请围绕“软件可靠性评价”论题，依次从以下三个方面进行论述。

- (1) 简要概述你参与实施的软件开发项目以及你承担的主要工作。
- (2) 说明你在项目实施过程中所选择的软件可靠性模型，并论述在软件可靠性模型选择时应该考虑的主要因素。
- (3) 收集软件可靠性数据时经常遇到的问题有哪些？简述你收集软件可靠性数据时所遇到的具体问题及解决的方法。

写作要点

一、说明软件开发项目的基本情况以及自己承担的主要工作。
二、当前的软件可靠性模型众多，但并没有一个最好的或者可以适用所用软件系统的软件可靠性模型，因此对于不同的软件系统，出于不同的可靠性分析目的，需要选择合适的软件可靠性模型。

常见的10类软件可靠性模型有种子法模型、失效率类模型、曲线拟合类模型、可靠性增长模型、程序结构分析模型、输入域分类模型、执行路径分析方法模型、非齐次泊松过程模型、马尔可夫过程模型和贝叶斯分析模型。

软件可靠性模型的选择主要需要考虑以下4个方面。

1.模型假设的适用性：模型假设是可靠性模型的基础，模型假设需要符合软件系统的现有状况，在软件系统中与假设冲突的因素达到几乎不存在的程度。往往一个模型的假设有很多，需要在选择模型时对每一条假设进行分析，评估现有软件系统中不符合假设的因素对可靠性评价有多大影响，以确定模型是否符合软件系统的可靠性评价工作。

2.模型预测的能力与质量：预测的能力和质量是指模型根据现在和历史的可靠性数据，预测将来的可靠性和失效概率的能力，以及预测结果的准确程度。因此，应尽可能选择比较成熟的、应用较广的模型。

3.模型输出值能否满足可靠性评价需求：根据可靠性测试目的来确定哪些模型的输出值满足可靠性评价需求。重要的可靠性定量指标包括：当前可靠度、平均无失效时间、故障密度、期望达到规定可靠性目标的日期、达到规定可靠性目标的成本要求等。

4.模型使用的简便性：模型使用的数据在软件系统中易于收集；模型应该简单易懂；模型应该便于使用，最好有工具支持。

三、软件可靠性数据的收集是一项艰巨而又烦琐的工作，受到许多潜在因素的影响和制约。常见的问题如下：

- (1) 可靠性数据规范不一致，对软件进行度量的定义混乱；
- (2) 数据收集过程存在于整个软件生命周期，但由于成本等因素，其连续性往往不能保证；
- (3) 缺乏有效的技术和工具支持，难以进行自动分析；
- (4) 数据完整性不能保证，收集到的数据大多数是不完全的；
- (5) 数据质量和准确性不能保证；
- (6) 缺乏可靠性数据的交流与共享。

考生应叙述在项目中遇到了上述中的哪些问题。

可供采用的解决方法主要有：

- (1) 尽早确定可靠性模型，明确需要搜集的可靠性数据，确定涉及的术语、记录方法等；
- (2) 制定可实施的可靠性数据搜集计划，并指定专人负责。保证数据的收集和验证与软件开发过程同步进行；
- (3) 重视软件测试，特别是可靠性测试产生的测试结果的整理和分析；
- (4) 尽可能地利用工具进行收集工作，例如利用数据库进行存储和分析等。

试题9 (2011年11月试题1)

试题9 (2011年11月试题1)

论模型驱动架构在系统开发中的应用

模型驱动架构 (Model Driven Architecture,MDA) 是对对象管理组织提出的软件体系架构方法学 , 它基于UML以及一系列工业标准 , 能够支持基于可视化模型驱动的软件设计、内容存储与交换。MDA的核心思想是抽象出与实现技术无关、完整描述业务功能的核心平台无关模型 (PIM) , 然后针对不同实现技术制定多个映射规则 , 通过映射规则和辅助工具将PIM转换成与具体实现技术有关的平台相关模型 (PSM) , 最后完成PSM到代码的转换。通过PIM和PSM,MDA分离业务建模与底层实现技术 , 降低技术变迁对业务模型带来的影响。

请围绕“模型驱动架构在系统开发中的应用”论题 , 依次从以下三个方面进行论述。

(1) 简要叙述你参与管理和开发的、与MDA相关的软件开发项目 , 以及你所担任的主要工作。

(2) 简要分析模型驱动架构能够为软件开发带来哪些好处 , 详细论述采用模型驱动架构进行开发的过程。

(3) 具体阐述你参与管理和开发的项目中使用模型驱动架构的情况与实际开发效果。

写作要点

一、模型驱动架构能够为软件开发带来的好处

(1) 模型驱动架构将开发人员的注意力转移到了平台无关模型中 , 可以避免陷入到具体的实现细节当中去 , 从而简化了系统开发的工作量 , 提高了软件的开发效率。

(2) 对于多种流行平台 , 很多工具会支持从平台无关模型到平台相关模型的转换 ; 对于将来可能出现的新技术和平台 , 确定了平台表示及公共中间件的概念和功能 , 利用转换规则快速实现从平台无关模型到新技术平台的迁移 , 提高了系统的可移植性。

(3) 利用模型驱动架构中基于平台无关模型的桥接器 , 实现了多个平台相关模型之间跨平台的相互通信 , 加强了互操作性。

(4) 对于系统变更 , 通过修改平台无关模型并重新生成平台相关模型和代码 , 能够降低系统维护的成本。

(5) 平台无关模型帮助团队成员之间提高沟通效率并减少错误 , 自动生成代码能够保证代码的质量和一致性 , 确保了软件的质量。

(6) 使用模型驱动架构时 , 功能和架构独立定义 , 针对新技术 , 能够利用原有的设计产生对应的实现 , 延长了系统的生命周期。

二、模型驱动架构的开发过程

(1) 使用平台无关模型从如何以最好的方式支持商业逻辑的角度对系统进行建模 , 开发人员根据用户需求和其他因素对平台无关模型进行精化 , 以使它能够更加精确地描述系统。

(2) 将平台无关模型转换到一个或多个特定技术相关的平台相关模型 , 对于每种特定的技术都会生成独立的平台相关模型。

(3) 根据技术特性对生成的平台相关模型进行修改以满足程序设计人员的要求 , 这些修改可以反映到平台无关模型中去。

(4) 对平台相关模型不断精化 , 以指导代码生成器生成质量更高的程序代码。

(5) 最后将每个平台相关模型转换到代码 , 进行后续的完善和系统测试。

三、结合项目的实际情况，具体阐述你参与管理和开发的项目中使用模型驱动架构的情况，包括平台无关模型构建、平台相关模型的技术方案选择和实际开发效果及分析。

第 14 章：架构设计论文

试题10 (2011年11月试题2)

试题10 (2011年11月试题2)

论企业集成平台的架构设计

企业集成平台是一个支持复杂信息环境下信息系统开发、集成和协同运行的软件支撑环境，它基于企业各种经营业务的信息特征，在异构分布环境（操作系统、网络、数据库）下为应用提供一致的信息访问和交互手段，对在其上运行的应用进行管理，为应用提供服务，并支持各种特定领域应用系统的集成。

请围绕“企业集成平台的架构设计”论题，依次从以下三个方面进行论述。

- (1) 简要叙述你参与管理和开发的企业集成平台项目，以及你在其中所承担的主要工作。
- (2) 请说明企业集成平台的基本功能，并结合项目实际，详细说明所设计的企业集成平台的架构，以及实现时用到了哪些关键技术。
- (3) 具体说明所设计的企业集成平台的使用情况，最终实施效果如何。

写作要点

一、企业集成平台的基本功能

- (1) 通信服务。提供分布环境下透明的同步/异步通信服务功能，使用户和应用程序无须关心具体的操作系统和应用程序所处的网络物理位置，而以透明的函数调用或对象服务方式完成它们所需的通信服务要求。
- (2) 信息集成服务。为应用提供透明的信息访问服务，通过实现异种数据库系统之间的数据交换、互操作、分布数据管理和共享信息模型定义，使在集成平台上运行的应用、服务或客户端能够以一致的语义和接口实现对数据的访问与控制。
- (3) 应用集成服务。通过高层应用编程接口来实现对相应应用程序的访问。这些接口以函数或对象服务的方式向平台的组件模型提供信息，用户无须对原有系统进行修改，只要在原有系统的基础上加上相应的访问接口就可以将现有的、用不同技术实现的系统互联起来，通过为应用提供数据交换和访问操作，使各种不同的系统能够相互协作。

- (4) 提供对二次开发的支持。集成平台需要提供一组帮助用户开发特定应用程序的支持工具，简化用户在企业集成平台实施过程中的开发工作。

- (5) 平台运行管理。需要提供企业集成平台的运行管理和控制模块，负责企业集成平台系统的静态和动态配置、集成平台应用运行管理和维护、事件管理和出错管理等。通过命名服务、目录服务、平台的动态静态配置，以及其中的关键数据的定期备份等功能来维护整个服务平台的系统配置及稳定运行。

二、结合项目实际说明你所设计的企业集成平台的架构。对架构的说明应包括从架构层面上如何支持业务流程的编写与管理；如何向用户提供功能与信息服务；如何集成业务伙伴的功能；如何与底层数据库、现有系统等进行交互，等等。在实现企业集成平台时所使用的关键技术包括如下。

- (1) 数据交换格式。企业集成中常用的数据交换格式有：EDI、XML、STEP、PDML。
- (2) 分布式集成应用基础框架。主要有CORBA、J2EE、Web Service。
- (3) 实现数据集成的常用模式。数据联邦、数据复制和基于接口的数据集成。
- (4) 实现应用集成的常用模式。适配器集成、信使集成、面板集成、代理集成模式。

三、需要具体说明你所设计的企业应用集成平台的使用情况，包括如何采用集成平台为企业应用提供一致的信息访问和交互手段，如何对在平台上运行的应用进行管理，如何为应用提供服务等。针对每种使用场景，需要详细说明最终的实施效果。

第 14 章：架构设计论文

试题11 (2011年11月试题3)

试题11 (2011年11月试题3)

论企业架构管理与应用

企业架构管理 (Enterprise Architecture Management, EAM) 从功能、应用、信息和技术四个层面定义了企业应用系统的结构，并通过业务需求驱动开发过程，为企业应用系统的开发提供标准和指导。EAM将企业的业务和技术需求联系在一起，以管理业务变更为核心，强调业务与技术对齐，构建一个高内聚、动态的企业应用解决方案。

EAM能够帮助企业识别可以提高运营效率的潜在领域，有助于企业建立从战略到解决方案交付的各种关系，识别技术解决方案中最优的业务成果，能够在业务重组、兼并、收购和其他业务变更计划中为企业最大化地节约成本，降低相关风险。

请围绕“企业架构管理与应用”论题，依次从以下三个方面进行论述。

- (1) 简要叙述你参与实施的企业应用系统的开发背景与总体需求、系统所采用的技术体制、实施企业软件架构管理的动机与期望，以及你所承担的实际工作。
- (2) 结合项目实际，简要阐述企业架构管理包含哪些方面的内容，每个方面包括哪些主要活动。
- (3) 阐述你在实施企业架构管理的过程中都遇到了哪些实际问题，以及解决这些问题的方法和过程。

写作要点

一、企业架构管理 (EAM) 以管理业务变更为核心，根据业务目标确定 IT 投资的优先级；强调业务驱动技术，从管理的角度看待企业架构。企业架构管理主要包含以下几个方面的内容。

(1) 架构管理 (Architecture Administration)。其作用是对企业架构进行管理与配置，主要活动包括存储管理、元模型管理、访问和认证管理、多语方管理和自动化管理。

存储管理：组织并管理企业架构相关的信息与存储，并对其进行生命周期管理。

元模型管理：定义并管理企业架构中的元模型，并实现元模型在不同应用之间的交互与映射。

访问和认证管理：管理企业内部用户、用户群组、用户目录和用户对企业架构信息的访问。

多语言管理：如果企业架构描述存在多种语言，需要对这些不同版本的描述进行存储及一致性管理。

自动化管理：对EAM的整个过程选择合适的自动化工具，并对工具进行适当的配置与管理。

(2) 架构组装与建模 (Architecture Populating and Modeling)。其作用是将架构描述信息进行整合，并将其放入存储结构中。主要活动包括手工组装与建模、自动化组装与建模和与非结构化数据的连接。

手工组装与建模：手工将图表、文档等形式描述的企业架构信息进行整合并录入架构存储结构。

自动化组装与建模：将数据库、XML等结构化形式描述的企业架构信息进行迁移与整合。

与非结构化数据的连接：将企业架构信息通过内容管理系统与非结构化的数据（例如网页、图片、视频等）进行连接与关联。

采用一些通用的框架或标准对架构进行描述、建模并存储。

(3) 架构分析 (Architecture Analysis)。其主要作用是理解并分析企业架构内容，并做出相关判断。主要活动包括浏览和检索、结构分析、定性/定量分析和基于时间的分析。

浏览和检索：支持企业内部用户对架构内容进行有效地浏览与检索。

结构分析：对企业架构进行结构分析，发现其中的不足、冗余和架构制品之间的相互影响情况。

定性/定量分析：对企业架构的代价与优势、利用率等指标进行定性或定量分析。

基于时间的分析：分析随着时间的推移，企业架构的变化及变化带来的影响。

(4) 架构通信 (Architecture Communication)。其主要作用是对企业架构内容进行发布与传播。主要活动包括信息发布、报告、企业实时信息反映与报告和可视化。

信息发布：在企业内容的门户系统或共享文件夹中发布企业架构相关的信息。

报告：在企业相关报告中使用企业架构的内容，并利用企业架构内容为相关活动进行指导。

企业实时信息反映与报告：对企业架构内容进行分析、统计等工作，在企业内部形成能够反映企业运营状况的实时信息报告。

可视化：能够为企业应用系统的关联人员创建可视化的企业架构内容，更好地实现他们之间的交流。

(5) 架构治理 (Architecture Governance)。其主要作用是在企业架构过程中引入解决方案发布、变更管理和质量保证等重要的治理过程与能力。主要活动包括完成与发布管理、变更管理、使用追踪和质量保证。

完成与发布管理：严格定义并执行企业架构内容完成与发布的工作流程。

变更管理：严格定义并执行对企业架构内容的变更控制与追踪。

使用追踪：追踪用户和用户组对企业架构内容的实际使用情况。

质量保证：保证架构内容的完整性、一致性和无二义性。

二、在实施企业架构管理的过程中可能遇到的问题包括：如何选择合适的EAM工具，如何在企业内部有效地共享企业架构信息，如何结合企业实际进行企业架构的变更管理，如何保证架构内容的质量，等等。针对每个问题，说明解决的方法和过程。

试题12 (2011年11月试题4)

试题12 (2011年11月试题4)

论软件需求获取技术及应用

软件需求是指用户对新系统在功能、行为、性能、设计约束等方面的期望。软件需求获取是一个确定和理解不同的项目干系人的需求和约束的过程。需求获取是否科学、准备充分，对获取的结果影响很大，这是因为大部分用户无法完整地描述需求，而且也不可能看到系统的全貌。因此，掌握各种不同的需求获取技术，熟练地在实践中运用它，并与用户有效合作，是十分重要的。

请围绕“需求获取技术及应用”论题，依次从以下三个方面进行论述。

- (1) 简要叙述你参与管理和开发的软件项目，以及你在其中所承担的主要工作。
- (2) 详细说明目前有哪些比较常用的需求获取技术？说明每种需求获取技术的基本方法。
- (3) 详细论述在你参与分析和开发的软件项目中所采取的需求获取技术以及选取这些技术的原因，并说明需求获取的具体实施步骤。

写作要点

一、常用的需求获取技术有：用户访谈、用户调查、现场观摩、阅读历史文档、联合讨论会（联合需求计划）等。

(1) 用户访谈。用户访谈是最基本的一种需求获取手段，其形式包括结构化和非结构化两种，结构化是指事先准备好一系列问题，有针对性地进行；非结构化则是只列出一个粗略的想法，根据访谈的具体情况发挥。最有效的访谈是结合这两种方法进行的。用户访谈具有良好的灵活性，用较宽广的应用范围，但是也存在着许多困难，诸如客户经常较忙，你难以安排到时间；面谈时信息量大，记录较为困难；沟通需要很多技巧，同时需要分析员有足够的领域知识；另外，在访谈时会遇到一些对于组织来说比较机密和敏感的话题。因此，这看似简单的技术，也需要分析人员拥有足够的经验和较强的沟通能力。

(2) 用户调查。用户访谈时最大的难处在于很多关键的人员时间有限，不容易安排过多的时间；而且客户面经常较广，不可能一一访谈。因此，我们就需要借助用户调查，通过精心设计要问的问题，然后下发到相关的人员手里，让他们填写答案。这样就可以有效地克服前面提到的两个问题。但是与用户访谈相比，用户调查最大的不足就是缺乏灵活性；而且双方未见面，分析人员无法从他们的表情等其他动作来获取一些更隐性的信息；还有就是客户有可能在心理上会不重视一张小小的表格，不认真对待从而使得反馈的信息不全面。因此较好的做法是将这两种技术结合使用。具体来说，就是先设计问题，制作成为用户调查表，下发填写完后，进行仔细的分组、整理、分析，以获得基础信息，然后再针对这个结果进行小范围的用户访谈，作为补充。

(3) 现场观摩。对于许多较为复杂的流程和操作而言，是比较难以用言语表达清楚的，而且这样做也会显得很低效。因此，针对这一现象，分析团队可以就一些较复杂、较难理解的流程、操作采用现场观摩的方法来获取需求。具体来说，就是走到客户的工作现场，一边观察，一边听客户的讲解，甚至可以安排人员跟随客户工作一小段时间。这样就可以使得分析人员更加直观地理解需求。

(4) 阅读历史文档。这种方式也称为“文档考古”。对于一些数据流比较复杂的、工作表单较多的项目，有时是很难通过语言，或者通过观察来了解需求细节的。这个时候就可以借助于阅读历史文

档的方法，对历史存在的一些文档进行研究，从中获得所需的信息。这个方法的主要风险是历史的文档可能与新系统的流程、数据有一些不吻合的地方，并且还可以承载一些原有系统的缺陷。要想有效地避免和发现这些问题，就需要分析人员能够运用自己的聪明才智，将其与其他需求捕获技术结合对照。还有一个负面影响就是，这些历史的文档中记载的信息有可能涉及客户的商业秘密，因此对数据信息的保密也是分析人员基本的职业道德。

(5) 联合讨论会。这是一种相对来说成本较高的需求获取方法，但也是十分有效的一种。它通过联合各个关键客户代表、分析人员、开发团队代表一起，通过有组织的会议来讨论需求。通常该会议的参与人数为6~18人，召开时间为1~5小时。在会议之前，应该将与讨论主题相关的材料提前分发给所有将要参加会议的人。在会议开始之后，首先应该花一些时间让所有的与会者互相认识，以使交流在更加轻松的气氛下进行。会议的最初，就是针对所列举的问题进行逐项专题讨论，然后对原有系统、类似系统的不足进行开放性交流，第三步则是大家在此基础上对新的解决方案进行一番设想，在过程中将这些想法、问题、不足记录下来，形成一个要点清单。第四步就是针对这个要点清单进行整理，明确优先级，并进行评审。这种联合讨论会将会起到群策群力的效果，对于一些问题最有歧义的时候、对需求最不清晰的领域都是一种十分有用的技术。而且最大的难度就是会议的组织，要做到言之有物，气氛开放，否则将难以达到预想的效果。

二、结合项目实际工作，举例说明你在获取需求时分别采用了哪些需求获取技术；详细说明你选择这些技术的原因及具体实施步骤。

第 14 章：架构设计论文

试题13 (2012年11月试题1)

试题13 (2012年11月试题1)

基于架构的软件设计 (Architecture-Based Software Design,ABSD) 方法以构成软件架构的商业、质量和功能需求等要素来驱动整个软件开发过程。ABSD是一个自顶向下，递归细化的软件开发方法，它以软件系统功能的分解为基础，通过选择架构风格实现质量和商业需求，并强调在架构设计过程中使用软件架构模板。采用ABSD方法，设计活动可以从项目总体功能框架明确后就开始，因此该方法特别适用于开发一些不能预先决定所有需求的软件系统，如软件产品线系统或长生命周期系统等，也可为需求不能在短时间内明确的软件项目提供指导。

请围绕"基于架构的软件开发方法及应用"论题，依次从以下三个方面进行论述。

- (1) 概要叙述你参与开发的、采用ABSD方法的软件项目，以及你在其中所承担的主要工作。
- (2) 结合项目实际，详细说明采用ABSD方法进行软件开发时，需要经历哪些开发阶段？每个阶段包括哪些主要活动？
- (3) 阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

写作要点

一、论文中要具体介绍项目的背景与总体需求、系统所采用的技术路线，以及你所承担的实际工作。

二、采用ABSD方法进行软件开发时，需要经历架构需求、架构设计、架构文档化、架构复审、架构实现和架构演化六个阶段。

1.架构需求阶段需要明确用户对目标软件系统在功能、行为、性能、设计约束等方面期望。其主要活动包括需求获取、标识构件和架构评审。

1) 需求获取活动需要定义开发人员必须实现的软件功能，使得用户能够完成他们的任务，从而满足功能需求。与此同时，还要获得软件质量属性，满足一些非功能性需求。

2) 标识构件活动首先需要获得系统的基本结构，然后对基本结构进行分组，最后将基本结构进行打包成构件。

3) 架构需求评审活动组织一个由系统涉众（用户、系统分析师、架构师、设计实现人员等）组成的小组，对架构需求及相关构件进行审查。审查的主要内容包括所获取的需求是否真实反映了用户需求，构件合并是否合理等。

2.架构设计阶段是一个迭代过程，利用架构需求生成并调整架构决策。主要活动包括提出架构模型、将已标识的构件映射到架构中、分析构件之间的相互作用、产生系统架构和架构设计评审。

3.架构文档化的主要活动是对架构设计进行分析与整理，生成架构规格说明书和测试架构需求的质量设计说明书。

4.在一个主版本的软件架构分析之后，需要安排一次由外部人员（客户代表和领域专家）参加的架构复审。架构复审需要评价架构是否能够满足需求，质量属性需求是否在架构中得以体现、层次是否清晰、构件划分是否合理等。从而标识潜在的风险，及早发现架构设计中的缺陷和错误。

5.架构实现主要是对架构进行实现的过程，主要活动包括架构分析与设计、构件实现、构件组装和系统测试。

6.架构演化阶段主要解决用户在系统开发过程中发生的需求变更问题。主要活动包括架构演化计划、构件变动、更新构件的相互作用、构件的组装与测试和技术评审。

三、在软件开发的过程中可能遇到的问题包括：在架构需求获取过程中如何对捕获的架构需求进行筛选和优先级排序；在架构复审过程中如何解决评审人员的意见不一致问题；在架构实现过程中如何根据项目组实际情况选择开发语言与开发平台；在架构演化过程中如何筛选并处理用户的需求变更，等等。

第 14 章：架构设计论文

试题14 (2012年11月试题2)

试题14 (2012年11月试题2)

论企业应用系统的数据持久层架构设计

数据持久层（Data Persistence Layer）通常位于企业应用系统的业务逻辑层和数据源层之间，为整个项目提供一个高层、统一、安全、并发的数据持久化机制，完成对各种数据进行持久化的编程工作，并为系统业务逻辑层提供服务。它能够使程序员避免手工编写访问数据源的方法，使其专注于业务逻辑的开发，并且能够在不同项目中重用本框架，这大大简化了数据的增加、删除、修改、查询功能的开发过程，同时又不丧失多层结构的天然优势，继承延续应用系统架构的可伸缩性和可

扩展性。当运用关系型数据库作为数据存储机制时，在业务层与数据源间加入数据持久层，能够解决对象与关系的“阻抗不匹配”问题，将对象的状态持久化存储到关系型数据库中。

请围绕“企业应用系统的数据持久层架构设计”论题，依次从以下三方面进行论述。

(1) 概要叙述你参与分析和设计的企业应用系统开发项目，以及你所担任的主要工作。

(2) 分析在企业应用系统的数据持久层架构设计中有哪些数据访问模式，并详细阐述每种数据访问模式的主要内容。

(3) 数据持久层架构设计的好坏决定着应用程序性能的优劣，请结合实际说明在数据持久层架构设计中需要考虑哪些问题。

写作要点

一、简要描述你所参与分析和设计的企业应用系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、分析在企业应用系统的数据持久层架构设计中有哪些数据访问模式，并详细阐述每种数据访问模式的主要内容。

企业应用系统的数据持久层架构设计中主要有如下五种数据访问模式。

(1) 在线访问 (Online Access)。OA是最基本的数据访问模式，也是在实际开发过程中最常采用的。这种数据访问模式会占用一个数据库连接，读取数据，每个数据库操作都会通过这个连接不断地与后台的数据源进行交互。

(2) 数据访问对象 (Data Access Object)。DAO模式是标准的J2EE设计模式之一，开发人员常常用这种模式将底层数据访问操作与高层业务逻辑分离开。一个典型的DAO实现通常包括：一个DAO工程类；一个DAO接口；一个实现了DAO接口的具体类，包含访问特殊数据源中数据的逻辑；数据传输对象。

(3) 数据传输对象 (Data Transfer Object)。DTO是经典EJB设计模式之一，它本身是一组对象或者数据的容器，需要跨越不同的进程或者网络的边界来传输数据。对象本身应该不包含具体的业务逻辑，并且通常这些对象内部职能要进行一些诸如内部一致性检查和基本验证之类的方法，而且这些方法最好不要再调用其他的对象行为。在具体实现DTO时，可以使用编程语言内置的集合对象，也可以通过创建自定义类来实现DTO对象。

(4) 离线数据模型 (Off-line Data Model)。ODM以数据为中心，数据从数据源获取之后，将按照某种预定义的结构存放在系统中，成为应用的中心。离线方式可以使得对数据的各种操作独立于各种与后台数据源之间的连接或者事务；通过与XML集成数据可以方便地与XML格式的文档之间相互转换；独立于数据源，ODM定义了数据的存储结构和规则。

(5) 对象关系映射 (Object Relational Mapping)。ORM是随着面向对象软件开发方法发展而产生的，面向对象开发方法是主流的开发方法，关系型数据库是企业级应用环境中永久存放数据的主流数据存储系统。对象和关系数据是业务实体的两种表现形式，业务实体在内存中表现为对象，在数据库中表现为关系数据。ORM一般以中间件的形式存在，能够帮助将应用程序中的数据转换成关系型数据库中的记录；或者将关系数据库中的记录转换成应用程序中便于操作的对象。

三、数据持久层架构设计的好坏决定着应用程序性能的优劣，无论在C/S还是在B/S结构中，持久层在处理数据的同时，对服务器锁的类型和持续时间、输入/输出活动量以及处理器负荷等产生主要影响，并由此影响应用程序的总体性能。在持久层设计阶段需要考虑的问题包括：网络流量问题；返回结果集的问题；查询或锁定超时的问题；应用程序开发工具的问题；使用游标的问题；应用层设计的问题等。

试题15 (2012年11月试题3)

试题15 (2012年11月试题3)

论决策支持系统的开发与应用

决策支持系统 (Decision Support Systems, DSS) 是以管理科学、运筹学、控制论和行为科学为基础，以计算机技术、仿真技术和信息技术为手段，以人机交互方式进行半结构化和非结构化决策的信息系统。它调用各种信息资源，并提供各种分析工具，为决策者提供分析问题、建立模型、模拟决策过程和方案的环境，帮助决策者提高决策水平和质量。决策支持系统在许多领域得到了广泛的应用，已成为许多行业经营管理中一个不可缺少的现代化支持工具。

请围绕“决策支持系统的开发与应用”论题，依次从以下三个方面进行论述。

- (1) 概要叙述你参与管理和开发的决策支持系统项目，以及在其中所担任的主要工作。
- (2) 简要叙述决策支持系统包含的典型组成部件及对应的基本功能。说明在建立决策支持系统时需解决的一般关键问题。
- (3) 说明你所参与管理和开发的决策支持系统的应用场合以及对决策结果的要求，具体阐述在开发过程中所采用的关键技术、实施过程和实际应用的效果。

写作要点

一、简要叙述你所参与管理和开发的决策支持系统项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、决策支持系统包括如下典型组件。

- (1) 接口部分，即输入/输出的界面，是人机交互的窗口。
- (2) 模型管理子系统，具有存储、动态建模的功能。目前模型管理的实现是通过模型库系统来完成的。
- (3) 知识管理子系统，集中管理决策问题领域的知识（规则和事实），包括知识的获取、表达、管理等功能。
- (4) 数据管理子系统，DSS的数据库通常包括在数据仓库中。数据仓库是集成的、面向主题的数据库集合。数据仓库通常从内部和外部数据源中抽取。内部数据主要来自组织的交易处理系统。外部数据包括行业数据、市场调查数据等。
- (5) 用户，用户可看作系统的一部分。DSS的用户主要是企业各层次的管理者和商业分析人员。

在建立决策支持系统时，主要的几个关键问题如下。

1. 建立数据仓库系统

数据仓库系统必须为决策支持的分析处理提供以下服务。

- (1) 根据主题需要，从OLTP数据库中抽取分析用的数据。为此在抽取过程中要对原始数据进行分类、求和、统计等处理，抽取的过程实际上是数据的再组织。
- (2) 在抽取过程中，完成数据净化，即去掉不合格的原始数据，必要时还必须对缺损的数据加以补充。

(3) 在改变分析决策的主题时，可以按主题进行数据查询和访问。

(4) 采用多级存储模式，解决数据量巨大及按照主题、粒度划分的数据组织问题。

2. 模型、方法和知识管理系统

采用数据仓库和多维数据库技术的数据管理子系统将数据进行整理（预处理）和净化之后，形成可靠的易于进行决策的“数据源”（即数据仓库或多维数据库），这个“数据源”的结构与形式和决策支持系统所采用的模型与知识有关。决策粗略地分为结构化决策支持、非结构化决策支持、半结构化决策支持。一个较好的决策支持系统必须完成这三方面的决策支持。

模型、方法和知识的管理是决策支持系统的核心，它对依据问题建立的模型库、方法库和知识库进行管理。

(1) 对模型库、方法库和知识库进行维护。模型、方法和知识管理系统必须有对这三库的维护界面；可根据问题的需要对模型、方法和知识库进行增加、删除和修改，并保证三库的一致性：一是系统运行过程调用每个库时不发生矛盾，特别是对知识库的维护更为复杂；二是每种模型、方法和知识都能调用到。

(2) 模型、方法和知识管理系统根据用户的要求和数据仓库提供的数据，能有效地选择模型、方法和知识，经系统运行得到相应结果，并将结果送给交互环境进行输出。

智能决策支持系统一般是在模型、方法和知识管理系统的基本上增加专家系统和数据采掘与知识发现技术。

智能决策支持系统（Intelligence Decision Support System, IDSS）的主要任务包括：

(1) 分析和识别问题；

(2) 描述决策问题和决策知识；

(3) 形成候选的决策方案（目标、规划、方法和途径等）；

(4) 构造决策问题的求解模型（如数学模型、运筹学模型、程序模型、经验模型等）；

(5) 建立评价决策问题的各种准则（如价值准则、科学准则、效益准则等）；

(6) 多方案、多目标、多准则情况下的比较和优化；

(7) 综合分析，包括决策结果或方案对实际问题可能产生的作用和影响的分析，以及各种环境因素、变量对决策方案或结果的影响程序分析等。

3. 用户交互环境

用户交互环境是决策者或决策部门与决策支持系统打交道的界面，它负责接收用户发出的各种命令，根据这些命令调用不同的子系统，并获得处理结果，最后再将这些结果输出给用户。

交互环境的好坏直接影响着用户对系统的使用。一个好的交互环境，其输入应当简单、易学、易用。其输出应当做到内容丰富、形式活泼。

三、考生需结合自身参与项目的实际状况，指出你参与管理和开发的决策支持系统的应用行业或领域，选择一个关键问题说明其设计、实现的具体过程、方法以及对实际应用效果的分析。

试题16 (2012年11月试题4)

论企业信息化规划的实施与应用

企业信息化建设是一项长期而艰巨的任务，不可能在短时间内完成。信息化规划是企业信息化建设的纲领和向导，是信息系统设计和实施的前提和依据。信息化规划以整个企业的发展目标和战略、企业各部门的目标与功能为基础，同时结合行业信息化方面的实践和对信息技术发展趋势的掌握，制定出企业信息化远景、目标和发展战略，从而达到全面、系统地指导企业信息化建设的目的。

请围绕“企业信息化规划的实施与应用”论题，依次从以下三个方面进行论述。

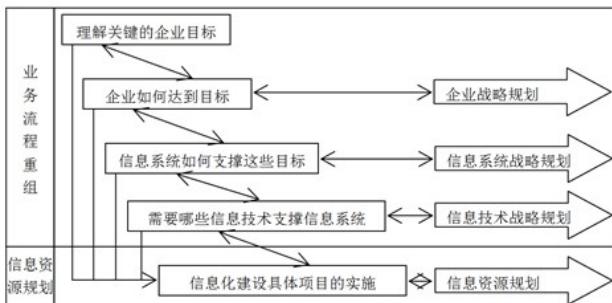
- (1) 概要叙述你参与的企业信息化规划项目，以及你在其中所担任的主要工作。
- (2) 简要叙述企业信息化规划的主要内容。结合你参与的项目的实际情况，详细分析有关企业的信息化规划目标及规划的具体内容。
- (3) 说明你所参与实施的企业信息化规划的步骤及效果，介绍其是否达到了预期的目标并分析原因。

写作要点

一、简要叙述你所参与管理和开发的企业信息化规划项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、企业信息化规划的内容

企业信息化规划不仅涉及信息系统规划，同时与企业规划、业务流程建模等紧密相关，是融合企业战略、管理规划、业务流程重组等内容的“业务+管理+技术”的规划活动，如下图所示。



涉及业务流程重组和信息资源规划、信息技术战略规划、信息系统战略规划和企业战略规划等多个领域。所有的规划都应该围绕企业关键目标的实现而展开，并为企业目标的实现提供支持和必要的服务。

进行信息化规划时，需要做好以下几个方面的工作：

- (1) 明确发展目标和实施重点；
- (2) 成立领导机构；
- (3) 做好企业业务信息化需求分析；
- (4) 确定企业信息化不同发展阶段的投资预算；
- (5) 制定必要的促进企业信息化建设的规章制度。

三、结合实际项目，详细阐述企业信息化规划的目标和实施重点，对于企业业务信息化需求分析应进行重点论述。说明企业信息化规划的实施过程，总结实施效果并进行进一步的分析。