软考设计师模拟试题 3（上午题答案）

●一个 3.5 英寸的磁盘,最小磁道的直径为 4 厘米,最大磁道直径为 8 厘米,每分钟 10000 转,共有 30 记录面,每个记录面有 8000 个磁道,每条磁道上有 511 个扇区,每个扇区实际记录有 600 个字节,其中有效数据为 512 个字节.则这个磁盘存储器的有效存储容量是 (1) GB,磁道密度是每毫米 (2) 跳磁道。

(1) A．60
B．58
C．63
D．30

(2) A．350
B．400
C．800
D．200

【解析】1.这个磁盘存储器的存储容量为 512B*30*8000*511=63GB;2.磁道方向的有效记录宽度为((80-40)/2)mm=20mm,磁道密度为 8000/20mm=400 条/m。

●

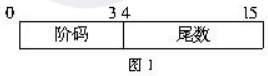

图 1

如图 1 所示为计算机中 16 位浮点数的表示格式。

某机器码为 1110001010000000。

若阶码为移码且尾数为反码，其十进制真值为 (3) ；

若阶码为移码且尾数为原码，其十进制真值为 (4) ；

若阶码为补码且尾数为反码，其十进制真值为 (5) ；

若阶码为补码且尾数为原码，其十进制真值为 (6) ，将其规格化后的机器码为 (7) 。

(3) ～ (6) A．0.078125
B．20
C．1.25
D．20.969375

(7) A．1110001010000000
B．11110101000000
C．1101010100000000
D．11110001010000

【解析】本题考查计算机数据的编码，涉及原码、补码、反码，移码以及浮点数规格化处理。

同一个数可以有不同的浮点表示形式，阶码的大小可以用来调节数值中小数点的位置。将数值数据表示成 N=M*RE,M 被称为 N 的尾数，E 是 N 的指数或称阶码，而 R 是该阶码的基数。

题中阶码用 4 位二进制整数 1110 表示，尾数用 12 位二进制小数 001010000000 表示，尾数中含有符号位，其最高位，即符号位为 0。下面具体分析题目中的各个问题。

如果阶码为移码，由于阶码是 4 位二进制整数，设真值为 X，根据整数移码定义：[X]移码=23+X(1110)2=(14)10，可求得阶码真值为 6。如果尾数为反码，从符号位可判断尾数是正数，根据小数反码定义，正小数的反码就是其自身，可求得尾数的真值为：(0.01010000000)2=(2-2+2-4)=(0.3125)10，根据浮点数定义，该机器码真值为 0.3125*26=20。

如果阶码为移码，同上，真值为 6。如果尾数是原码，从符号位可判断尾数是正数，根据小数原码定义，正小数的原码就是其本身，可求得尾数的真值为 0.3125。由此可知该机器码真值也是 20。

如果阶码为补码，由于阶码是 4 位二进制整数，从符号位判断为负数，设真值为 X，根据负整数定义[X]补码=24+X=(1110)2=(14)10，求得阶码的真值为-2。如果尾数为反码，同问题 A-样求出尾数的真值为 0.3125。这样，该机器码真值为 0.3125*2-2=0.078125。

如果阶码是补码，尾数是原码，可分别参照以上解析求出阶码和尾数的真值分别为-2 和 0.3125，这样该机器码的真值也是 0.078125。

对浮点数进行规格化处理，规定浮点数的尾数部分用纯小数形式表示，当尾数的值不为 0 时，其绝对值应大于或等于 0.5，用二进制表示为 0.1xxx…xx(x 为 0 或 1)。对于不符合这一规定的浮点数，可改变阶码的大小并同时用左右移尾数的方法来满足这一规定。显然尾数 0.01010000000 不合要求，应左移 1 位，而阶码则应相应地减 1，因此规格化处理后的阶码为 1101，尾数为 010100000000。

●UML 称为统一的建模语言，它把 Booch、Rumbaugh 和 Jacobson 等各自独立的 OOA 和 OOD 方法中最优秀的特色组合成一个统一的方法。UML 允许软件工程师使用由一组语法的语义的实用规则所支配的符号来表示分析模型。

在 UML 中用 5 种不同的视图来表示一个系统，这些视图从不同的侧面描述系统。每一个视图由一组图形来定义。这些视图概述如下：

(8) 用使用实例(use case)来建立模型，并用它来描述来自终端用户方面的可用的场景。

(9) 对静态结构(类、对象和关系)模型化。

(10) 描述了在用户模型视图和结构模型视图中所描述的各种结构元素之间的交互和协作。

(11) 将系统的结构和行为表达成为易于转换为实现的方式。

(12) 表示系统实现环境的结构和行为。

(8) ～(10) A．环境模型视图

B．行为模型视图

C．用户模型视图

D．结构模型视图

(11) ，(12) A．环境模型视图

B．实现模型视图

C．结构模型视图

D．行为模型视图

【解析】①用户模型视图：从用户(在 UML 中叫做参与者)角度来表示系统。它用使用实例(use case)来建立模型，并用它来描述来自终端用户方面的可用的场景。②结构模型视图：从系统内部来看数据和功能性，即对静态结构(类、对象和关系)模型化。③行为模型视图：这种视图表示了系统动态和行为。它还描述了在用户模型视图和结构模型视图中所描述的各种结构元素之间的交互和协作。④实现模型视图：将系统的结构和行为表达成为易于转换为实现的方式。⑤环境模型视图：表示系统实现环境的结构和行为。

●商品条码是在流通领域中用于标识商品的 (13) 通用的条码。条码中的 (14) 供人们直接识读，或通过键盘向计算机输入数据。

(13) A．行业

B．国际

C．国内

D．企业

(14) A．商品代码

B．条码符号

C．条码代码

D．商品条码

【解析】商品条码是在流通领域中用于标识商品的国际通用的条码。目前国际上广泛使用的条码是国际物品编码协会的标准化条码 EAN。我国于 1991 年 4 月正式加入国际物品编码协会，我国通用商品条码国家标准的结构与 EAN 条码结构相同，由 13 位数字码以及对应的条码组成：前缀码(3 位)、制造厂商代码(4 位)、商品代码(5 位)和检验码(1 位)。其中 3 位前缀码是标识国家或地区的代码，由 EAN 统一分配给各国家(地区)的编码组织，我国的国家代码为"690"；制造厂商代码由中国物品编码中心统一分配给各个申请厂商，每一个制造厂商的制造厂商代码都不同，在世界范围内惟一。商品条码还有北美地区通用的商品条码 UPC，其结构与国际通用的商品条码 EAN 有所不同。我国推广应用 EAN 条码，UPC 条码主要用于美国、加拿大等国家。我国出口到美国、加拿大的

某些类商品需要申请使用 UPC 条码。条码是一组规则排列的条、空及其对应字符组成的标记，用以表示一定的信息。条码中的条、空组合部分称为条码符号，条、空分别由两种不同深浅的颜色(通常为黑、白色)表示，并满足一定的光学对比度要求，其目的是便于光电扫描设备识读后将数据输入计算机。条码中对应条码符号的一组阿拉伯数字称为条码代码，条码代码供人们直接识读，或通过键盘向计算机输入数据。条码符号和条码代码相对应，表示的信息一致。

●在 CORBA 体系结构中，负责屏蔽底层网络通信细节的协议是 (15) 。

(15) A．IDL

B．RPC

C．ORB

D．GIOP

【解析】在 CORBA 应用中，通过一定的通信协议来屏蔽网络通信的细节，这个协议就是 GIOP 协议，IDL 是接口定义语言的简称，是用来定义对应的服务方接口，RPC 则是远程的过程调用，ORB 则是对象请求代理，负责控制客户方与服务方的交互，也就是我们俗称的 CORBA 平台的内核。

●电子商务具有 (16) 的运作模式。

(16) A．B2C

B．C2B

C．C2C

D．A2B

【解析】B2C(Business to Client)是电子商务的初始层面，它注重网络技术所带来的信息无限性和超越时空性。有效减少买卖的中间环节、降低经营成本、直接面对客户，从而有利于企业制定经营策略。B2C 商务模式的本质是一种强调整合物流的商务模式，是一种零售业，在相当程度上是低利润的代表，是直接面对消费者的商务形式。B2C 模式消除中间批发商，明显加快物流和资金流的运转，既能减少中间成本，也能减少欺诈风险。同时经营企业直接面对消费者，将市场调查、市场运作、产品更新、客户跟踪集于一身，自主性明显加强，从而容易建立价格优势。经营企业有能力针对市场情况，快速决策，制定有利价格，快速占领市场。

B2B(Business to Business)是供应链上的联合经营，其本质是追求信息增值。B2B 模式的行业特征为：信息化程度较高，商品标准化程度高；主观判断较少；产品结构复杂、产品市场足够大；需要与物质经济资源网全面整合，同步成长、相得益彰；跨地域、跨行业、低成本和大联盟；"关注顾客价值"和"追求信息增值"。大合作是 B2B 商务经济大发展的特点，它的信息技术包括对商务应用与企业资源规划 ERP（Enterprise Resource Planning)，客户关系管理 CRM(Customer Relationship Management)，供应链管理 SCM（Supply Chain Management)，人力资源管理 HRM(Human Resource Management)。B2B 是由多个买方和卖方(供应链的上游、下游甚至同级的企业)共同构成的企业联盟，共同提倡全球标准化，提倡开放式 EC 解决方案，并实现了信息服务、交易服务、支付服务、物流服务等各类要素高度结合而形成了新的价值链（Value Chain)经济。

●人们对软件存在着许多错误的观点，这些观点表面上看起来很有道理，符合人们的直觉，但实际上给管理者和开发人员带来了严重的问题。下述关于软件开发的观点中正确的是 (17) 。

(17) A．我们拥有一套讲述如何开发软件的书籍，书中充满了标准与示例，可以帮助我们解决软件开发中遇到的任何问题

B．如果我们已经落后于计划，可以增加更多的程序员和使用更多的 CASE 工具来赶上进度

C．项目需求总是在不断变化，我们可以采用瀑布模型来解决此类问题

D．需要得多是软件项目失败的主要原因

【解析】1）好的参考书无疑能指导我们的工作，充分利用书籍中的方法、技术和技巧，可以有效地解决软件开发中大量常见的问题。但实践者并不能依赖于书籍，因为在现实工作中，由于条件千差万别，即使是相当成熟的软件工程规范，常常也无法套用。另外，软件技术日新月异，没有哪一种软件标准能长盛不衰。2）软件开发不同于传统的机械制造，人多不见得力量大。如果给落后于计划的项目增添新人，可能会更加延误项目。因为新人会产生很多新的错误，使项目混乱，并且原有的开发人员向新人解释工作和交流思想都要花费时间，使实际的开发时间更少，所以制定恰如其分的项目计划是很重要的。3）软件需求确实是经常变化的，但这些变化产生的影响会随着其引入时间的不同而不同。对需求把握得越准确，软件的修修补补就越少。有些需求在一开始时很难确

定，在开发过程中要不断地加以改正。软件修改越早代价越少，修改越晚代价越大。4）不完善的系统定义是软件项目失败的主要原因。关于待开发软件的应用领域、功能、性能、接口、设计约束和标准等需要详细的描述，而这些只有通过用户和开发人员之间的通信交流才能确定。越早开始写程序，就要花越长时间才能完成它。

●下面是关于树和线性结构的描述：

线性结构存在惟一的没有前驱的 (18) ，树存在惟一的没有前驱的 (19) ：线性结构存在惟一的没有后继的 (20) ，树存在多个没有后继的 (21) ；线性结构其余元素均存在 (22) ，树其余结点均存在惟一的前驱(双亲)结点和多个后继(孩子)结点。

由此可见，由于线性结构是一个顺序结构，元素之间存在的是一对一的关系，而树是一个层次结构，元素之间存在的是一对多的关系。

(18) ～(21) A．根结点

B．首元素

C．尾元素

D．叶子

(22) A．惟一的前驱元素和后继元素

B．惟一的前驱(双亲)结点和多个后继(孩子)结点

C．叶子

D．一对一

【解析】线性结构是一个数据元素的有序(次序)集合。这里的"有序"仅指在数据元素之间存在一个"领先"或"落后"的次序关系，而非指数据元素"值"的大小可比性。它有 4 个基本特征：

①集合中必存在惟一的一个"第一元素"。②集合中必存在惟一的一个"最后元素"。③除最后元素外，其他数据元素均有惟一的"后继"。④除第一元素外，其他数据元素均有惟一的"前驱"。

树是以分支关系定义的层次结构，结构中的数据元素之间存在着"一对多"的关系，因此它为计算机应用中出现的具有层次关系或分支关系的数据，提供了一种自然的表示方法。如用树描述人类社会的族谱和各种社会组织机构。在计算机学科和应用领域中树也得到广泛应用。例如，在编译程序中，用树来表示源程序的语法结构等。

●下面是关于树和线性结构的描述：

线性结构存在惟一的没有前驱的首元素，树存在惟一的没有前驱的根结点；线性结构存在惟一的没有后继的尾元素，树存在多个没有后继的叶子；线性结构其余元素均存在惟一的前驱元素和后继元素，树其余结点均存在 (23) 。

由此可见，由于线性结构是一个 (24) 结构，元素之间存在的是 (25) 的关系，而树是一个 (26) 结构，元素之间存在的是 (27) 的关系。

(23) A．惟一的前驱元素和后继元素

B．惟一的前驱(双亲)结点和多个后继(孩子)结点

C．叶子

D．一对一

(24) ~(27) A．一对一

B．一对多

C．顺序

D．层次

【解析】线性结构是一个数据元素的有序(次序)集合。这里的"有序"仅指在数据元素之间存在一个"领先"或"落后"的次序关系，而非指数据元素"值"的大小可比性。它有 4 个基本特征：

①集合中必存在惟一的一个"第一元素"。②集合中必存在惟一的一个"最后元素"。③除最后元素外，其他数据元素均有惟一的"后继"。④除第一元素外，其他数据元素均有惟一的"前驱"。

树是以分支关系定义的层次结构，结构中的数据元素之间存在着"一对多"的关系，因此它为计算机应用中出现的具有层次关系或分支关系的数据，提供了一种自然的表示方法。如用树描述人类社会的族谱和各种社会组织机构。在计算机学科和应用领域中树也得到广泛应用。例如，在编译程序中，用树来表示源程序的语法结构等。

●软件开发模型用于指导软件开发。演化模型是在快速开发一个 (28) 的基础上，逐步演化成最终的软件。

螺旋模型综合了 (29) 的优点，并增加了 (30) 。

喷泉模型描述的是面向 (31) 的开发过程，反映了该开发过程的 (32) 特征。

(28) A．模块
B．运行平台
C．原型
D．主程序

(29) A．瀑布模型和演化模型
B．瀑布模型和喷泉模型
C．演化模型和喷泉模型
D．原型和喷泉模型

(30) A．质量评价
B．进度控制
C．版本控制
D．风险分析

(31) A．数据流
B．数据结构
C．对象
D．构件(Component)

(32) A．迭代和有间隙
B．迭代和无间隙
C．无迭代和有间隙
D．无迭代和无间隙

【解析】软件开发模型是指软件开发全部过程、活动和任务的结构框架。常用的软件开发模型有瀑布模型、演化模型、螺旋模型、喷泉模型等。

瀑布模型给出了软件生存周期各阶段的固定顺序，上一阶段完成后才能进入下一阶段。演化模型是在快速开发一个原型的基础上，根据用户在试用原型的过程中提出的反馈意见和建议，对原型进行改进，获得原型的新版本。重复这一过程，直到演化成最终的软件产品。螺旋模型将瀑布模型和演化模型相结合，它综合了两者的优点，并增加了风险分析。它以原型为基础，沿着螺线自内向外旋转，每旋转一圈都要经过制订计划、风险分析、实施工程、客户评价等活动，并开发原型的一个新版本。经过若干次螺旋上升的过程，得到最终的软件。喷泉模型主要用来描述面向对象的开发过程。它体现了面向对象开发过程的迭代和无间隙特征。迭代意味着模型中的开发活动常常需要多次重复;无间隙是指开发活动(如分析、设计)之间不存在明显的边界，各项开发活动往往交叉迭代地进行。

●ISO 为运输层定义了 4 种类型的服务原语，由运输层服务用户产生的原语是 (33) 。

(33) A．请求原语指示原语
B．请求原语响应原语
C．指示原语确认原语
D．相应原语确认原语

【解析】运输服务原语见表 4：

【解析】运输服务原语见表4：

表 4 运输服务原语

阶 段	服 务	原 语	参 数
TC 建立	TC 建立	T－CONNECT. request T－CONNECT. indication	Called address 被叫地址 Calling address 主叫地址 Expedited data option 加速数据选择 Quality of Service 服务质量 Ts－User data 用户数据
		T－CONNECT. response T－CONNECT. confirm	Responding address 响应地址 Expedited data option 加速数据选择 Quality of Service 服务质量 TS－User data 用户数据
数据传送	正常数据传送	T－DATA. request T－DATA. indication	Ts－User data 用户数据
	加速数据传送	T－EXPEDITED－DATA. request. T－EXPEDITED－DATA. indication	TS－User data 用户数据
TC 释放	TC 释放	T－DISCONNECT. request	TS－User data 用户数据
		T－DISCONNECT. indication	Disconnect reason 释放原因 TS－User data 用户数据
无连接数据传送	普通数据传送	T－UNITDATA. request. T－UNITDATA. indication	Called address 被叫地址 Calling address 主叫地址 Quality of Service 服务质量 TS－User data 用户数据

【解析】运输服务原语见表4：

表 4 运输服务原语

阶 段	服 务	原 语	参 数
TC 建立	TC 建立	T－CONNECT. request T－CONNECT. indication	Called address 被叫地址 Calling address 主叫地址 Expedited data option 加速数据选择 Quality of Service 服务质量 Ts－User data 用户数据
		T－CONNECT. response T－CONNECT. confirm	Responding address 响应地址 Expedited data option 加速数据选择 Quality of Service 服务质量 TS－User data 用户数据
数据传送	正常数据传送	T－DATA. request T－DATA. indication	Ts－User data 用户数据
	加速数据传送	T－EXPEDITED－DATA. request. T－EXPEDITED－DATA. indication	TS－User data 用户数据
TC 释放	TC 释放	T－DISCONNECT. request	TS－User data 用户数据
		T－DISCONNECT. indication	Disconnect reason 释放原因 TS－User data 用户数据
无连接数据传送	普通数据传送	T－UNITDATA. request. T－UNITDATA. indication	Called address 被叫地址 Calling address 主叫地址 Quality of Service 服务质量 TS－User data 用户数据

【解析】运输服务原语见表 4：

●IEEE 802 规范主要与 OSI 模型的 (34) 有关。

(34)A.较低的 4 层 B.转输层和网络层

●试题答案：(34)C

【解析】局域网的体系结构以 IEEE 802 委员会定义的标准为主，对应的 ISO 标准是 ISO802，局域网标准只定义了相当于 ISO 模型中的低两层，即物理层和数据链路层的规范。

●因为 ATM (35) ，即信元沿同一条路径走，所以，信元一般不会失序。

(35) A．是异步的

B．采用了分组交换的技术

C．采用电路交换的技术

D．用虚电路

【解析】在 ATM 中使用了虚电路概念，即每个信元中都含有虚电路标志，带有相同标志的信元属于同一个虚电路，这些信元将得到相同的处理并按先后顺序在 ATM 网络中传送。ATM 最重要的特点是能适用于一般电路交换和分组交换都不能胜任的高速宽带信息业务，它可适应范围宽广的可变速率，终端产生的数据比特流可以是突发式的，也可以是连续的。

●当存储器采用段页式管理时，主存被划分为定长的 (36) ，程序按逻辑模块分成 (37) 。在某机器的多道程序环境下，每道程序还需要一个 (38) 作为有用户标志号，每道程序都有对应 (39) 。一个逻辑地址包括 (38) ，x、段号 s、页号 p 和页内地址 d 等 4 个部分。

设逻辑地址长度分配如下，其中 x、s、p、d 均以二进制数表示。

212019141311100

xSpd

其转换后的地址为 (40) 。

(36) A．段
B．页
C．区域
D．块

(37) A．区域
B．页
C．块
D．段

(38) A．模块号
B．区域号
C．基号
D．区域

(39) A．一个段表和一个页表
B．一个段表和一组页表
C．一组段表和一个页表
D．一组段表和一组页表

(40) A．x*220+s*214+p*211+d
B．(((x)+s)+p+d
C．(((x)+s)+p)*211+(d)
D．(((x)+s)+p*211+d

【解析】段页式存储组织综合了段式组织与页式组织的特点，主存被划分为定长的页，段页式系统中的虚地址形式是(段号、页号、位移)。系统为每个进程建立一个段表，为每个段建立一个页表。也就是说，先将程序按逻辑模块(如主程序、子程序和数据段等)分为若干段，再将每个段分为若干页。对于多道程序环境，每道程序有一个基号与其他程序相区分，每道程序可以有多个段，但只有一个段表，每个程序可以有多个页表。段页式存储体系中逻辑地址与物理地址的转换：首先由基号段号得到段表的地址，再访问段表得到页表的地址，再由页表得到物理块的地址，此时得到的地址是高 11 位的地址，因此需乘以 211 再加上页内地址，才得到真正的物理地址。

●程序设计语言包括 (41) 等几个方面，它的基本成分包括 (42) 。Chomsky(乔姆斯基)提出了形式语言的分层理论，他定义了四类文法：短语结构文法、上下文有关文法、上下文无关文法和正则文法。一个文法可以用一个四元组 G=(Σ,V,S,P)表示，其中，Σ是终结符的有限字符表，V 是非终结符的有限字母表，S(∈V)是开始符号，P 是生成式的有限非空集。在短语文法中，P 中的生成式都是 α→β 的形式，其中 a∈ (43) ，β∈(Σ∪V)*。在上下文有关文法中，P 中的生成式都是 α1Aα2→α1βα2 的形式，其中 A∈ (44) ，β∈(Σ∪V*),β≠ε。在上下文无关文法中，P 中的生成式的左部∈ (45) 。

(41) A．语法、语义
B．语法、语用
C．语义、语用
D．语法、语义、语用

(42) A．数据、传输、运算
B．数据、运算、控制
C．数据、运算、控制、传输
D．顺序、分支、循环

(43)A．V+
B．(Σ∪V)
C．(Σ∪V)*
D．(Σ∪V)*V(Σ∪V)*

(44) A．V
B．V+
C．Σ∪V
D．(Σ∪V)*

(45) A．V
B．V+
C．Σ∪V
D．(Σ∪V)*

【解析】Chomsky 提出了形式语言的分层理论，他定义了 4 类文法：短语结构文法、上下文有关文法、上下文无关文法和正则文法。一个文法 G 可以用一个四元组 G=(VT，VN，S，P)来表示，其中 VT 是终结符的有限字符集，VN 是非终结符的有限字母表，S∈VN 是文法的开始符号，P 是形如α→β的形式，如果 P 中的产生式都是α→β的形式，其中α∈(VT∪VN)*VN(VT∪VN)*，β∈(VT∪VN)*，则称该文法为短语文法；如果 P 中的产生式都是αAδ→αβδ的型式，其中 A∈VN，α和δ∈(VT∪VN)*，β∈(VT∪VN)+，则称该文法是上下文有关文法；如果 P 中的产生式都是α→β的形式，其中 A∈VN，B∈(VT∪VN)*，则称该文法为上下无关文法；如果 P 中的产生式都是 A→α或 A→αB 的形式，其中 A 和 B∈VN，α∈VT*，则称该文法为正则文法。

●设有关系模式 S (Sno，Sname，Pno，Pname，Q，A)表示销售员销售商品情况，其中各属性的含义是：Sno 为销售员员工号，Sname 为销售员姓名，Pno 为商品号，Pname 为商品名称，Q 为销售商品数目，A 为销售商品总金额。根据定义有如下函数依赖集：P={Sno→Sname，Sno→Q，Sno→A，Pno→Pname}。

关系模式 S 的关键字是 (46) ，W 的规范化程度最高达到 (47) 。若将关系模式 S 分解为 3 个关系模式 S1(Sno，Sname，Q，A)，S2(Sno，Pno，Pname)，则 S1 的规范化程度最高达到 (48) ，S2 的规范化程度最高达到 (49) 。SQL 中集合成员资格的比较操作"元组 IN(集合)"中的"IN"与 (50) 操作符等价。

(46) A．(Sno，Q)
B．(Pno，A)
C．(Sno，Pno)
D．(Sno，Pno，Q)

(47) A．1NF
B．2NF
C．3NF
D．BCNF

(48) A．1NF
B．2NF
C．3NF
D．BCNF

(49) A．1NF
B．2NF
C．3NF
D．BCNF

(50) A．<>ANY
B．=ANY
C．<>Like
D．=Like

【解析】①根据给定的函数依赖集和 Armstrong 公理，可以推导出：

Sno，Pno→Sname，Pname，Q，A

并且(Sno，Pno)中任意一个属性都不能用函数决定其他所有属性，所以，对于关系模式 S 的关键字是(Sno，Pno)。

②在关系 S 中，函数依赖 Pno→Pname 和 Sno→Sname，Q，A

可以得出非主属性 Pname、Sname、Q 和 A 均部分依赖于主关键字，违背第二范式的定义，因此关系 S 最高满足

第一范式。

③对于分解后的两个关系，根据原函数依赖集，S1 仅存在函数依赖：

Sno→Sname，Q，A

也就是 Sno 函数决定关系 S1 中所有属性，所以 Sno 是关系 S1 的关键字，因此关系模式 S1 满足 BCNF。

④根据原关系函数依赖集，S2 中存在函数依赖：Pno→Pname

对于关系 S2 来说，Pno 和 Sno 共同才能函数决定关系中所有属性，因此关系 S2 的关键字是(Pno，Sno)。而函数依赖 Pno→Pname，非主属性 Pname 部分依赖于主关键字，违背第二范式的定义，因此关系 S2 最高满足第一范式。

⑤运算符 IN 表示元组在集合中，=ANY 表示元组等于集合中某一个值，两者的含义是相同的。

●为了保证数据库的完整性(正确性)，数据库系统必须维护事务的以下特性 (51) 。

(51) A．原子性、一致性、隔离性、持久性

B．原子性、一致性、隔离性、闭包性

C．一致性、隔离性、持久性、完整性

D．隔离性、闭包性、时间性、适用性

【解析】为了保证数据库的完整性(正确性)，数据库系统必须维护事务的以下特性(简称 ACID)：

①原子性(Atomicity)：事务中的所有操作要么全部执行，要么都不执行。

②一致性(Consistency)：主要强调的是，如果在执行事务之前数据库是一致的，那么在执行事务之后数据库也是一致的。

③隔离性(Isolation)：即使多个事务并发(同时)执行，每个事务都感觉不到系统中有其他的事务在执行，因而也就能保证数据库的一致性。

④持久性(Durability)：事务成功执行后它对数据库的修改是永久的，即使系统出现故障也不受影响。

●在平衡二叉排序树上进行查找时，其时间复杂度为 (52) 。

(52) A．O(log2n+1)

B．O(log2n)

C．O(log2n-1)

D．log22n

【解析】此题是考查二叉树的查找效率问题。这是二叉树的基本查找问题，因为是平衡二叉树，其时间复杂度即为树的高，所以为 log2。

●各种需求方法都有它们共同适用的 (53) 。

(53) A．说明方法

B．描述方式

C．准则

D．基本原则

【解析】虽然各种分析方法都有独特的描述方法，但所有的分析方法还是有它们共同适用的基本原则。这些基本原则包括：

要能够表达和理解问题的信息域和功能域。

要能以层次化的方式对问题进行分解和不断细化。

要分别给出系统的逻辑视图和物理视图。

●对于单链表，如果仅仅知道一个指向链表中某结点的指针 p， (54) 将 p 所指结点的数据元素与其确实存在的直接前驱交换，对于单循环链表来说 (55) ，而对双向链表来说 (56) 。

(54) ~(56) A．可以

B．不可以

C．不确定

D．仅能一次

【解析】单链表和单循环链表的结点结构为：date、next

双向链表的结点结构为：prior、date、next

①单链表。②单循环链表。③双向链表。

(1)从单链表中的 p 结点出发，找不到它的直接前驱，因此不能与其直接前驱交换。

(2)从单循环链表中的 p 结点出发，可以找到其直接前驱，因此，可以与其直接前驱结点交换数据。程序段如下：

```
q=p->next;
while(q->next!=p)
q=q->next;
temp=p->data;
p->data=q->data;
q->data=temp;
```

(3)从双循环链表中的 p 结点出发，可以找到其直接前驱，因此，可以与其直接前驱结点交换数据。程序段如下：

```
temp=p->prior->data;
p->prior->data=p->data;
p->data=temp;
```

●采用邻接表存储的图的深度优先遍历算法类似于二叉树的 (57) 。

(57) A．中序遍历

B．前序遍历

C．后序遍历

D．按层遍历

【解析】图的深度优先遍历即纵向优先遍历，类似于二叉树的前序遍历。

●采用邻接表存储的图的广度优先遍历算法类似于二叉树的 (58) 。

(58) A．中序遍历

B．前序遍历

C．后序遍历

D．按层遍历

【解析】图的广度优先遍历即横向优先遍历，类似于二叉树的按层遍历。

●用顺序存储的方法将完全二叉树中的所有结点逐层存放在一维数组 R[1]到 R[n]中，那么，结点 R[i]若有左子树，则左子树是结点〖ZZ(Z〗 (59) 。

(59) A．R[2i+1]

B．R[2i-1]

C．R[i／2]

D．R[2f]

【解析】根据二叉树的性质 5，对完全二叉树从上到下、从左至右给结点编号，若编号为 2i 的结点存在，则 i 的左子树一定是 2i。

●假定一棵三叉树的结点数为 50，则它的最小高度为 (60) 。

(60) A．3

B．4

C．5

D．6

【解析】结点数相同而高度最小的三叉树是满三叉树或完全三叉树(深度为 h 的三叉树，若前面 h-1 层是满的，只有第 h 层从右边连续缺若干个结点的三叉树称为完全三叉树)。根据完全二叉树的性质 4(即具有 n 个结点的完全二叉树，其深度 h=[log2n]+1)，可推得三叉树的相应性质，即具有 n 个结点的完全三叉树，其深度 h=[log3n]+1。故具有 50 个结点的三叉树，其最小高度为[log350]+1=5。

●任何一棵二叉树的叶结点在前序、中序、后序序列中的相对次序 (61) 。

(61) A．不发生改变

B．发生改变

C．不能确定

D．以上都不对

【解析】如果用符号 D 表示访问根结点，用 L 表示遍历左子树，用 R 表示遍历右子树，那么前序、中序、后序遍历可分别表示为：DLR、LDR、LRD。由此可见，在三种遍历序列中 L 和 R 的相对次序都是 L 在前、R 在后。所以，任何一棵二叉树的叶结点在前序、中序、后序序列中的相对次序都不会发生改变。

●多媒体电子出版物创作的主要过程可分为 (62) 。基于内容检索的体系结构可分为两个子系统： (63) 。

(62) A．应用目标分析、脚本编写、各种媒体数据准备、设计框架、制作合成、测试

B．应用目标分析、设计框架、脚本编写、各种媒体数据准备、制作合成、测试

C．应用目标分析、脚本编写、设计框架、各种媒体数据准备、制作合成、测试

D．应用目标分析、各种媒体数据准备、脚本编写、设计框架、制作合成、测试

(63) A．用户访问和数据库管理子系统

B．多媒体数据管理和调度子系统

C．特征抽取和查询子系统

D．多媒体数据查询和用户访问子系统

【解析】(62)空中多媒体电子出版物创作的主要过程可分为应用目标分析、脚本编写、设计框架、各种媒体数据准备、制作合成、测试。(63)空是基于内容的检索作为一种信息检索技术，接入或嵌入到其他多媒体系统中，如超媒体(浏览器)系统、会议系统、多媒体信息系统、关系数据库系统等，提供基于多媒体数据内容的信息查询和检索。因此，将基于内容的检索设计为多媒体数据库的检索引擎结构，在体系结构上划分为两个子系统：特征抽取子系统和查询子系统。

●MIDI 是一种数字音乐的国际标准，MIDI 文件存储的 (64) 。它的重要特色是 (65) 。

(64) A．不是乐谱而是波形

B．不是波形而是指令序列

C．不是指令序列而是波形

D．不是指令序列而是乐谱

(65) A．占用的存储空间少

B．乐曲的失真度少

C．读写速度快

D．修改方便

【解析】MIDI 是英文 Musical Instrument Digital Interface 的缩写，英文的直译就是乐器数字接口。它是由世界上主要电子乐器厂商建立起来的一个通信标准，与多媒体结合后，日趋完善。MIDI 音频是多媒体计算机产生声音(特别是音乐)的主要方式之一。MIDI 文件记录的不是声音本身，不是声波的采样值。它是把每个音符记录为数字，记录着定时、音长、音量、力度、通道信息等。其文件是一系列的指令。MIDI 文件既有强大的功能，又节省大量的存储空间。例如半小时的立体声音乐，如用波形文件记录，大约要 300MB，而 MIDI 文件只要 200KB 就够了，这是 MIDI 文件的重要特色。

● (66) is a protocol that a host uses to inform a router when it joins or leaves anInternet multicast group.

(67) is an error detection code that most data conlmunication networks use.

(68) is an interior gateway protocol that uses a distance vector algorithm to propagaterouting information.

(69) is a transfer mode in which all types of information are organized into fixed form cells on all asynchronous or nonperiodic basis over a range of mediA.

(70) is an identifier of a web page.

(66) A．ICMP

B．SMTP

C．IGMP

D．ARP

(67) A．4B／5B

B．CRC

C．Manchester Code

D．Huffman Code

(68) A．OSPF

B．RIP

C．RARP

D．BGP

(69) A．ISDN

B．x.25

C．Frame Relay

D．ATM

(70) A．HTTP

B．URL

C．HTML

D．TAG

【解析】(参考译文)IGMP 协议是接收者发现协议。IGMP 协议运行在主机和路由器之间，用于路由器维护组播组是否有组成员。IGMP 只维护组播组是否有成员，而不维护组播组有哪些成员，因此状态信息不会因组播组成员的增加而增加。

循环冗余校验(cyclic-redundancy check，CRC)是由分组线性码的分解而来，其主要应用是二元码组，主要应用于数据传输过程中的差错控制。

OSPF 是一个内部网关协议(Interior Gateway Protocol，IGP)，用于在单一自治系统(autonomous system，AS)内决策路由。与 RIP 相对，OSPF 是链路状态路由协议，而 RIP 是距离向量路由协议。它是依靠距离，即跳数来衡量一条路径的好坏。通过距离向量路由算法来传播路由信息。

ATM 是异步传输模式，实现 OSI 物理层和链路层功能。ATM 以独有的 ATM 信元进行数据传输，每个 ATM 信元为 53 个字节。ATM 不严格要求信元交替地从不同的源到来，每一列从各个源来的信元，没有特别的模式，信元可以从任意不同的源到来，而且不要求从一台计算机来的信元流是连续的，数据信元可以有间隔，这些间隔由特殊的空闲信元(idle cell)填充。

URL(Uniform Resource Locator)是单一资源定位符，是一种用来鉴别文件与资源在 WWW 中地址的专用表示。

● Network managers have long awaited practical voice-over IP(VOIP)solutions.VOIP promises(71)network management and decreases costs by(72)a company's telephony and data infrastructures into one network.And a VOIP solution implemented at a company's head-quarters with far-reaching branch offices can(73)tremendous amounts of(74)in long distance phone bills，provided that solution delivers POTS-like voice(75)over the Internet.

(71)A.complicated B.useful C.ease D.orderly

(72)A.converging B.dividing C.combine D.bringing

(73)A.get B.put C.save D.waste

(74)A.cash B.money C.space D.time

(75)A.qualtity B.quality C.volune D.speed

【解析】(参考译文)网络管理者很久之前就等待着 VoIP 的出现。VoIP 使网络管理更加简便，并通过将公司语音和数据结合在一个网络内传送达到减小开销的目的。VoIP 可以节省公司总部与遥远的分支机构间的长途话音业务的开销，并可保证同传统话音有着相同的话音质量。

软考设计师模拟试题 3（下午题答案）

●试题一

［问题 1］

【答案】医院收费系统的 0 层图中″处方记录″

［问题 2］

【答案】1.″1.1 检查病人信息″的″不合格病人信息″输出数据流。

2.″1.2 计算费用″的″收据″输出数据流。

［问题 3］

【答案】1.从″病人基本情况″到″3.1 检查处方单″的数据流。

2.从″3.2 记录处方″到″处方记录″的数据流。

3.从″定价表″到″3.3 制作收据″的数据流。

4.从″3.3 制作收据″到″收费记录″的数据流。

【解析】在 0 层图中有 0 层图中″处方记录″是加工 3″处方收费″的局部数据文件，所以不必画出。

找出缺少的数据流的一个关键是父图与子图的平衡，即子图的输入输出数据流与父图相应的加工的输入输出数据必须一致。

从 0 层图中可以看到对于加工 1″病历收费″有输入流″病人信息″，输出流″不合格病人信息″，″病历″和″收据″。而加工 1 子图中却只有″病人信息″和″病历″，所以一定缺少 2 条输出流″不合格病人信息″和″收据″。病人信息是否合格是在加工 1.1″检查病人信息″中处理，因此加工 1.1 出一条输出流″合格病人信息″外，还缺少一条输出流″不合格病人信息″。对合格的病人信息，加工 1.2 计算收费后，理应提供收据给病人，所以另一条缺少的数据流是″1.2 计算费用″的″收据″输出数据流。

根据说明″系统首先根据病人基本情况检查处方单中病历号是否正确″，因此，在加工 3.1″检查处方单″中，需读入病人基本情况，所以缺少从″病人基本情况″到″3.1 检查处方单″的数据流。然后系统″记录合格的处方单″，所以加工 3.2″记录处方″中需讲处方的内容记录到文件″处方记录″中，因此缺少从″3.2 记录处方″到″处方记录″的数据流。加工 3.3″制作收据″中需根据文件″定价表″的各项目或药品的价格来计算所需收取的费用，因此图中还缺少从″定价表″到″3.3 制作收据″的数据流。最后收费的记录需写入文件″收费记录″中，所以缺少的第 4 条数据流是从″3.3 制作收据″到″收费记录″的数据流。

●试题二

［问题 1］

【答案】0 层图中的″采购清单″多余，应去掉。采购只需有采购请求就可以。

［问题 2］

【答案】加工 1 子图中遗漏了″配件库存″文件到 1.3 加工的数据流。加工 1 子图中 1.4 加工遗漏了″提货单″输出数据流。加工 1 子图中 1.5 加工遗漏了″到货通知″输入数据流。加工 2 子图中 2.3 加工遗漏了″采购请求″输入数据流。

●试题三

【答案】

（1）COUNT(*)(若答 COUNT 或 COUNT*得 2 分)

（2）GAMES.INO=IFEM.INO

（3）GAMES.ANO=’100872’(注：(2)、(3)可互换、无前缀得 1 分)

（4）EXISTS

（5）*或 ANO 或 INO 或 SCORE 或后 3 个列名的任意组合

（6）CREATEVIEW

（7）ATHLETE，ITEM，GAMES(3 项可交换。)

注：（4）、（5）也可为

（4）ANOIN

（5）ANO

【解析】本题是关于系数据库标准语言--SQL(Structured Query Language)语言的题目，由题目中给出的 ER 图可知 3 个表中，ATHLETE 和 ITEM 是基本表，表 ATHETE 的主键是运动员编号 ANO，表 ITEM 的主键是项目编号 INO，表 GAMES 是一个视图，以 ANO、INO 为外键。

程序 1 统计参加比赛的男运动员人数，也就是表 ATHLETE 中，AEX=′ M′ 的记录的个数，所以要用到库函数 COUNT(*)。这里要注意的是 COUNT 与 COUNT(*)区别，COUNT 的功能是对一列中的值计算个数，而 COUNT(*)才是计算数据库中记录的个数。所以填空①的答案为″COUNT(*)″。

程序的 2 统计 100872 号运动员参加的所有项目及比赛时间和地点，所以 SELECT 后面的内容是项目编号 ITEM.INO、项目名称 INAME 时间 ITIME 及地点 IPLACE。统计涉及比赛表 GAMES 和项目表 ITEM，所以 FROM 后面的内容为 GAMES、ITEM。本题考的是连接查询，所谓连接查询指的是涉及两个以上的表的查询。由于是统计 100872 号运动员参加的所有项目及比赛时间和地点，所以查询条件中必然有 GAMES.INO=′ 100872′ (程序中引用到字段时，若字段名在各个表中是惟一的，则可以把字段名前的表名去掉，否则，应当加上表名作为前缀，以免引起混淆)。由于 GAMES 表中只有比赛的成绩，那些关于项目的数据必须从项目表 ITEM 中取得，所以还应该有两个表之间的关联，即 GAMES.INO=TTEM.INO。所以填空②和③可交换，不影响查询结果。

程序 3 要求查参加 100035 项目的所有运动员名单。分析查询表达式，必首先查询 GAMES 表，找出参加 100035 项目的那些运动员的编号 ANO，即 GAMES.ANO=ATHLETE.ANO AND INO=′ 100035′ ，然后再根据查询到的运动员号 ANO 从 ATHLETE 表中抽取运动员的数据。所以填空④的答案为″EXISTS″或″ANOIN″，填空⑤的答案为″ANO″。

程序 4 要求建立运动员成绩视图。建立视图的命令为 CREATEVIEW，所以填空⑥的答案一定是″CREATEVIEW″。建立的是运动员成绩视图，那么一定涉及运动员情况、运动员参加的项情况和该项目的成绩，所以要用到 ATHLETE、ITEM 和 GAMES 这 3 个表，因此 FROM 子句后为 ATHLETE、GAMES、 ITEM，3 个表可以是任意次序，不影响结果。

●试题四

【答案】(1)(*a!=′ ＼0′)＆＆(*b!=′ ＼0′)(2)a++

(3)b++(4)*w==′ ＼0′ (5)*(++w)=t(6)a++

(7)*(++w)=′ ＼0′ (8)s[i]>s[j] (9)s[j]=t(10)strmerge(s1，s2，s3)

【解析】根据题意，对字符串的处理分为三步：第一步是从键盘上输入两个字符串；第二步是将两个字符串分别排序；第三步是将字符串合并；第四步是显示处理结果。

第一步和第四步容易实现，关键是第二步和第三步的处理，下面分别加以说明。

字符串排序是指将一个字符串中各个字符按照 ASCⅡ码值的大小排序。例如，字符串″Beijing″由小到大的排序结果应该是："Bejiign"。排序算法很多，第二个例子，我们就要介绍快速排序算法。在这里使用简单的冒泡排序算法：即将字符串中的每一个字符一个个进行比较，找出最小的字符，然后再在剩下的字符中找最小的字符。例如，字符″Beijing″的排序过程如下：

第一次将字符″Beijing″中的每一个字符：′ B′ 、′ e′ 、′ i′ 、′ j′ 、′ i′ 、′ n′ 、′ g′ 进行比较，找到最小的字符′ B′ 。

第二次在剩下的字符′ e′ 、′ i′ 、′ j′ 、′ i′ 、′ n′ 、′ g′ 中，找到最小的字符′ e′ 。

……

第三次在剩下的字符′ j′ 、′ i′ 、′ n′ 、′ g′ 中，找到最小的字符′ j′ 。

第三步是合并字符串，合并后的字符串仍然由小到大排序。由于待合并的两个字符串已经排好序。假定两个排好序的字符串分别为 A 和 B，合并后的字符串为 C，要使待合并后的字符串仍然由小到大排序，可采取下述步骤：

1.从前往后取 A 中的字符，并按从前往后的顺序与 B 中的字符比较，若 A 中的字符较小，则将该字符存入 C，并移到 A 的下一个字符，继续与 B 中的字符比较。

2.若 A 中的字符较大，则将 B 中的字符存入 C，并移到 B 的下一个字符，继续与 A 中的字符比较。

3.若 A 与 B 中的字符相等，则将 A 或 B 中的字符存入 C，并将 A 和 B 均移到下一个字符。

4.若 A 或 B 字符串到达末尾，则将 B 或 A 的剩余部分加到字符串 C 中。

需要注意的是：A、B 和 C 三个字符串均可以用字符数组来表示，C 数组的长度不能小于 A、B 两数组的长度之和。另外，判别字符串是否结尾的方法是：从 A 或 B 中取出的字符是否为′ ＼0′ ，所有字符串都是以′ ＼0′

结尾的。

●试题五

【答案】（1）color［*ip］（2）adj［i］[j］ != 0 && color［j］ == c

（3）i,k,adj,color（4）select （i,c+l,adj,color）（5）color［i］=c

【解析】（1）Back（）函数将 color 数组中紧邻*ip 位置的，颜色值为 4 的一个连续区域的元素赋值为－1。（2）colorOK（）判断区域 i 对其之前的所有区域是否可以着色 c。该句是检查 i 的相邻区域是否已有颜色为 c 的。（3）这是 colorOK 的参数列表。Select 为区域 i 选择一种颜色，使用 colorok 函数对各种颜色（值为 c～4 的一种，不一定是所有颜色）分别进行检查。（4）Coloring（）函数寻找各种着色方案。它先从区域 0 开始，检查颜色，并着色（着色的顺序总是从小色值的颜色开始的）。当发现某一区域无法着色时，就使用 back（）函数，将该区域之前的一个连贯区域进行洗色（对应 color 数组中赋值为－1）并回溯，并从回溯后的位置，重新开始进行颜色检查和赋色，但使用的色值比该位置洗色前的颜色值更大。若所有区域均已着色，则输出该着色方案。然后，使用 back 函数，重新进行着色。当所有颜色方案均已找到后，函数结束。（5）该句对区域 i 赋颜色 c。c 为之前 select 函数所选出的可以用的颜色。

●试题六

【答案】（1）quot=_quot；exp=_exp；next=NULL；（2）p!=NULL && exp < p -> exp

（3）new Item （quot，exp）（4）L1.list -> exp + L2.list -> exp

（5）pL1 -> exp +pL2 -> exp < k（6）quot += pL1 -> quot * pL2 -> quot

【解析】程序主要由类 Item 和 List 组成，其中类 Item 定义多项式中的项，由三个私有成员组成分别是：系数 quot、指数 exp 和指向多项式的下一项的指针 next，该类定义了一个构造函数，其作用是创建系数为_quot、指数为_exp 的项，即创建类 Item 的一个对象，因此（1）处应该填″quot=_quot;exp=_exp;next=NULL″。类 List 定义了多项式的操作，数据成员 list 是多项式链表的链头指针，类成员函数 creatlist（）的功能是创建按照指数降序链接的多项式，创建的方法是不断将新的项插入到链表的合适位置上。若链表中存在一项 p，其指数大于待插入项的指数，则待查入项为 p 的前驱。该成员函数在链表上遍历，直到找到满足上述条件的项；若链表中不存在这样的项，那么待插入项称为新的链尾。因此（2）处应填″p!=NULL&&exp<p->exp″；（3）处应填″new Item（quot,exp）″，根据读入的指数和系数创建要插入的项。若链表中存在指数与待插入项指数相等的项则合并同类项。

成员函数 multiplyList（List L1,List L2）计算多项式 L1 和 L2 的乘积。首先计算了 L1 和 L2 的乘积多项式的最高幂次，即多项式 L1 和 L2 的最高此项的指数之和。由于多项式 L1 和 L2 是按照指数的降序排列的，两个多项式的第一项分别是最高幂项，这两项的指数之和就是乘积多项式的最高次幂，因此（4）处填″L1.list->exp+L2.list->exp″。

为了实现系数的乘积求和计算，当多项式 L1 从幂次高至幂次低逐一考虑各项的系数时，多项式 L2 应从幂次低至幂次高的顺序考虑各项的系数，以便将两多项式所有幂次和为 k 的两项系数相乘后累计。由于是单链表，所以成员函数先将其中多项式 L2 的链接顺序颠倒，计算完成之后，再将多项式 L2 的链接顺序颠倒，即恢复原来的链接顺序。乘积多项式从高次幂系数至 0 次幂系数的顺序逐一计算。

为求 k 次幂这一项的系数，对于 L1 的考查顺序是从高次幂项至最低次幂项，而对于 L2 的考查相反。首先跳过多项式 L1 中高于 k 次幂的项，设低于 k 次幂的项最高次幂是 j 次幂；对于多项式 L2，跳过低于 k-j 次幂的项，因此（5）处应该填″pL1->exp+pL2->exp<k″。

考虑多项式 L1 和 L2 剩余各项的循环，若两多项式的当前项幂次和为 k，则累计他们系数的乘积，并分别准备考虑下一项，因此（6）处应该填″quot+=pL1->quot*pL2->quot″；若两多项式的当前幂次和大于 k，则应考虑幂次和更小的项，这样应该准备考虑多项式 L1 的下一项；若两多项式的当前幂次和小于 k，则应考虑幂次和更大的项，这样应该准备考虑多项式 L2 的下一项。若所有幂次和为 k 的项的系数乘积之和不等于 0，则应该在乘积多项式中有这一项，生成这一项的新结点，并将它插在乘积多项式的末尾。

●试题七

【答案】（1）out=h*3600+m*60+s（2）caculateSecond()（3）"合计："+nSum+"秒", 20, 90

（4）"合计："+nSum+"秒"（5）System.out.println

【解析】本题主要考查 Applet 的窗口，文件和文件 I/O，面向对象的基本概念以及基于文本的应用。解题关键是熟悉 Applet 的执行过程，会使用 Graphics 类的基本方法在用户界面中输出字符信息，会将 Applet 面向对象的基

本思想与文件操作相结合，编写有一定综合性的程序。本题中，1 小时等于 3600 秒，这里主要是要熟练掌握运算表达式的写法。程序中不可以直接用 objTime3_3 对象访问类的成员变量，应该调用成员方法，如果不调用方法去计算，得不到正确的结果。