

2014 年上半年

全国计算机技术与软件专业技术资格考试 2014 年上半年 软件设计师 上午试卷与解析

- 在 CPU 中, 常用来为 ALU 执行算术逻辑运算提供数据并暂存运算结果的寄存器是 (1)。

(1) A. 程序计数器 B. 状态寄存器 C. 通用寄存器 D. 累加寄存器

☞ 试题分析 在运算器中, 累加寄存器是专门存放算术或逻辑运算的一个操作数和运算结果的寄存器, 能进行加、减、读出、移位、循环移位和求补等操作, 是运算器的主要部分。

参考答案: D

- 某机器字长为 n , 最高位是符号位, 其定点整数的最大值为 (2)。

(2) A. 2^n-1 B. $2^{n-1}-1$ C. 2^n D. 2^{n-1}

☞ 试题分析 由于最高位是符号位, 因此最大的定点整数是:

$$\underbrace{011111\cdots111}_{n-1\text{个}1}$$

最高位 0 表示正数, 值为 $2^0+2^1+2^2+\cdots+2^{n-2}=2^{n-1}-1$ 。

参考答案: B

- 海明码利用奇偶性检错和纠错, 通过在 n 个数据位之间插入 k 个检验位, 扩大数据编码的码距。若 $n=48$, 则 k 应为 (3)。

(3) A. 4 B. 5 C. 6 D. 7

☞ 试题分析 n 和 k 满足关系:

$$2^k-1 \geq n+k$$

题中 $n=48$, k 为 6 可满足上述关系式。

参考答案: C

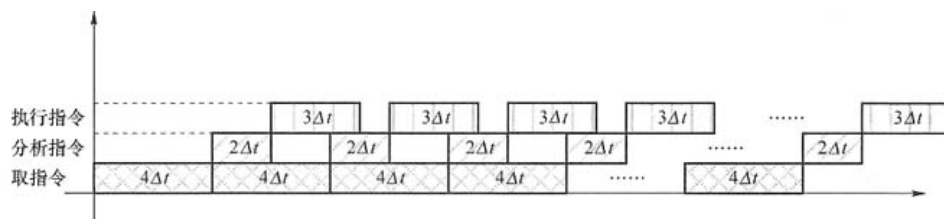
- 通常可以将计算机系统中执行一条指令的过程分为取指令, 分析和执行指令 3 步。若取指令时间为 $4\Delta t$, 分析时间为 $2\Delta t$, 执行时间为 $3\Delta t$, 按顺序方式从头到尾执行完 600 条指令所需时间为 (4) Δt ; 若按照执行第 i 条, 分析第 $i+1$ 条, 读取第 $i+2$ 条重叠的流水线方式执行指令, 则从头到尾执行完 600 条指令所需时间为 (5) Δt 。

(4) A. 2400 B. 3000 C. 3600 D. 5400

(5) A. 2400 B. 2405 C. 3000 D. 3009

☞ 试题分析 按顺序方式需要执行完一条执行之后再执行下一条指令, 执行 1 条执行所需的时间为 $4\Delta t+2\Delta t+3\Delta t=9\Delta t$, 执行 600 条指令所需的时间为 $9\Delta t \times 600=5400\Delta t$ 。

若采用流水线方式, 则处理过程如下图所示, 可见执行完 600 条执行所需要的时间为 $4\Delta t \times 600+2\Delta t+3\Delta t=2405\Delta t$ 。



参考答案: D B

- 若用 256K×8bit 的存储器芯片, 构成地址 40000000H 到 400FFFFFH 且按字节编址的内存区域, 则需 (6) 片芯片。

(6) A. 4 B. 8 C. 16 D. 32

☞ 试题分析 内存区域从 40000000H 到 400FFFFFH, 占用的字节数为

$$400FFFFFH - 40000000H + 1 = 100000H = 1\ 0000\ 0000\ 0000\ 0000B = 2^{20} = 1024K$$

一片 256K×8 bit 的存储器芯片的存储容量为 256KB, 需要的芯片数为 $1024 \div 256 = 4$ 。

参考答案: A

- 以下关于木马程序的叙述中, 正确的是 (7)。

(7) A. 木马程序主要通过移动磁盘传播
B. 木马程序的客户端运行在攻击者的机器上
C. 木马程序的目的是使计算机或网络无法提供正常的服务
D. Sniffer 是典型的木马程序

☞ 试题分析 木马(Trojan), 是指通过特定的程序(木马程序)来控制另一台计算机。木马通常有两个可执行程序: 一个是控制端, 另一个是被控制端。植入对方电脑的是服务端, 而黑客正是利用客户端进入运行了服务端的电脑。运行了木马程序的服务端以后, 会产生一个有着容易迷惑用户的名称的进程, 暗中打开端口, 向指定地点发送数据(如网络游戏的密码, 即时通信软件密码和用户上网密码等), 黑客甚至可以利用这些打开的端口进入电脑系统。

Sniffer, 中文可以翻译为嗅探器, 是一种基于被动侦听原理的网络分析方式。使用这种技术方式, 可以监视网络的状态、数据流动情况以及网络上传输的信息。Sniffer 不是木马程序。

参考答案: B

- 防火墙的工作层次是决定防火墙效率及安全的主要因素, 下面的叙述中正确的是 (8)。

(8) A. 防火墙工作层次越低, 则工作效率越高, 同时安全性越高
B. 防火墙工作层次越低, 则工作效率越低, 同时安全性越低
C. 防火墙工作层次越高, 则工作效率越高, 同时安全性越低
D. 防火墙工作层次越高, 则工作效率越低, 同时安全性越高

☞ 试题分析 防火墙工作层次越高, 实现过程复杂, 对数据包的理解力越好, 对非法包的判断能力越高, 但工作效率越低; 防火墙工作层次越低, 实现过程越简单, 其工作效率越高, 同时安全性越差。

参考答案: D

- 以下关于包过滤防火墙和代理服务防火墙的叙述中, 正确的是 (9)。

(9) A. 包过滤成本技术实现成本较高, 所以安全性能高
B. 包过滤技术对应用和用户是透明的
C. 代理服务技术安全性较高, 可以提高网络整体性能
D. 代理服务技术只能配置成用户认证后才建立连接

☞ 试题分析 包过滤技术是一种基于网络层、传输层的安全技术, 优点是简单实用, 实现成本较低同时, 包过滤操作对于应用层来说是透明的, 它不要求客户与服务器程序做任何修改。但包过滤技术无法识别基于应用层的恶意入侵, 如恶意的 Java 小程序以及电子邮件中附带的病毒。

代理服务技术基于应用层, 需要检查数据包的内容, 能够对基于高层协议的攻击进行拦截, 安全性较包过滤技术要好。缺点是处理速度比较慢, 不适用于高速网之间的应用。

另外, 代理使用一个客户程序与特定的中间节点连接, 然后中间节点与代理服务器进行实际连接。因此, 使用这类防火墙时外部网络与内部网络之间不存在直接连接, 即使防火墙发生了问题, 外部网络也无法与被保护的网络连接。

参考答案: B

- 王某买了一幅美术作品原件, 则他享有该美术作品的 (10)。

(10) A. 著作权 B. 所有权 C. 展览权 D. 所有权与其展览权

● **试题分析** 显然，作品的买卖导致了所有权的转移，著作权法第十八条规定：“美术等作品原件所有权的转移，不视为作品著作权的转移，但美术作品原件的展览权由原件所有人享有。”因此，作品交易后，著作权仍归原作者，王某享有购买的美术作品的所有权和其展览权。

参考答案：D

- 甲、乙两软件公司于 2012 年 7 月 12 日就其财务软件产品分别申请“用友”和“用有”商标注册。两财务软件相似，甲第一次使用时间为 2009 年 7 月，乙第一次使用时间为 2009 年 5 月。此情形下，(11) 能获准注册。

(11) A. “用友”

B. “用友”与“用有”都

C. “用有”

D. 由甲、乙抽签结果确定

● **试题分析** 《中华人民共和国商标法实施条例》第十九条规定“两个或者两个以上的申请人，在同一种商品或者类似商品上，分别以相同或者近似的商标在同一天申请注册的，各申请人应当自收到商标局通知之日起 30 日内提交其申请注册前在先使用该商标的证据。同日使用或者均未使用的，各申请人可以自收到商标局通知之日起 30 日内自行协商，并将书面协议报送商标局；不愿协商或者协商不成的，商标局通知各申请人以抽签的方式确定一个申请人，驳回其他人的注册申请。商标局已经通知但申请人未参加抽签的，视为放弃申请，商标局应当书面通知未参加抽签的申请人。”

参考答案：C

- 以下媒体中，(12) 是表示媒体，(13) 是表现媒体。

(12) A. 图像

B. 图像编码

C. 电磁波

D. 鼠标

(13) A. 图像

B. 图像编码

C. 电磁波

D. 鼠标

● **试题分析** 表示媒体指的是为了传输感觉媒体而人为研究出来的媒体，借助于此种媒体，能有效地存储感觉媒体或将感觉媒体从一个地方传送到另一个地方。如语言编码、电报码、条形码等。

表现媒体指的是用于通信中使电信号和感觉媒体之间产生转换用的媒体。如输入、输出设备，包括键盘、鼠标器、显示器、打印机等。

参考答案：B D

- (14) 表示显示器在横向（行）上具有的像素点数目。

(14) A. 显示分辨率

B. 水平分辨率

C. 垂直分辨率

D. 显示深度

● **试题分析** 显示分辨率一般用显示设备水平方向和垂直方向上的最大像素数目来表示。水平分辨率表示显示器在横向（行）上具有的像素点数目；垂直分辨率则表示显示器在纵向（列）上具有的像素点数目。

参考答案：B

- 以下关于结构化开发方法的叙述中，不正确的是(15)。

(15) A. 将数据流映射为软件系统的模块结构

B. 一般情况下，数据流类型包括变换流型和事务流型

C. 不同类型的数据流有不同的映射方法

D. 一个软件系统只有一种数据流类型

● **试题分析** 结构化分析与设计方法是一种面向数据流的开发方法，它以数据流为中心构建软件的分析模型和设计模型。结构化设计是将结构化分析的结构（数据流图）映射成软件的体系结构（结构图）。根据信息流的特点，可将数据流图分为变换型数据流图和事务型数据流图，其对应的映射分别称为变换分析和事务分析。

在规模较大的实际系统中，其数据流图往往是变换型和事务型的混合结构，此时可以把变换分析是事务分析应用在同一数据流图的不同部分。

参考答案：D

- 模块 A 提供某个班级某门课程的成绩给模块 B，模块 B 计算平均成绩、最高分和最低分，将计算结果返回给模块 A，则模块 B 在软件结构图中属于(16) 模块。

(16) A. 传入

B. 传出

C. 变换

D. 协调

● **试题分析** 模块结构图中的模块类型如下：

- 1) 传入模块：从下属模块取得数据，经过某些处理，再将其结果传给上级模块。
- 2) 传出模块：从上级模块取得数据，经过某些处理，再将其结果传给下属模块。
- 3) 变换模块：也叫加工模块。它从上级模块取得数据，进行特定的处理，转换成其他形式，再传回上级模块。它加工的数据流称作变换数据流。
- 4) 协调模块：对所有下属模块进行协调和管理的模块。在系统的输入输出部分或数据加工部分可以找到这样的模块。在一个好的模块结构图中，协调模块应在较高层出现。

参考答案：C

- (17) 软件成本估算模型是一种静态单变量模型，用于对整个软件系统进行估算。

- (17) A. Putnam B. 基本 COCOMO
C. 中级 COCOMO D. 详细 COCOMO

● **试题分析** COCOMO 用 3 个不同层次的模型来反映不同程度的复杂性，它们分别为：

基本模型(Basic Model)：是一个静态单变量模型，它用一个以已估算出来的源代码行数(LOC)为自变量的函数来计算软件开发工作量。

中级模型(Intermediate Model)：则在用 LOC 为自变量的函数计算软件开发工作量的基础上，再用涉及产品、硬件、人员、项目等方面属性的影响因素来调整工作量的估算。

详细模型(Detailed Model)：包括中级 COCOMO 型的所有特性，但用上述各种影响因素调整工作量估算时，还要考虑对软件工程过程中分析、设计等各步骤的影响。

参考答案：B

- 以下关于进度管理工具 Gantt 图的叙述中，不正确的是 (18)。

- (18) A. 能清晰地表达每个任务的开始时间、结束时间和持续时间
B. 能清晰地表达任务之间的并行关系
C. 不能清晰地确定任务之间的依赖关系
D. 能清晰地确定影响进度的关键任务

● **试题分析** 甘特图内在思想简单，即以图示的方式通过活动列表和时间刻度形象地表示出任何特定项目的活动顺序与持续时间。基本是一条线条图，横轴表示时间，纵轴表示活动（项目），线条表示在整个期间上计划和实际的活动完成情况。它直观地表明任务计划在什么时候进行，及实际进展与计划要求的对比。管理者由此可便利地弄清一项任务（项目）还剩下哪些工作要做，并可评估工作进度。

优点：能清晰地描述每个任务从何时开始，到何时结束以及各个任务之间的并行性。

缺点：不能清晰地反映出个任务之间的依赖关系，难以确定整个项目的关键所在，也不能反映计划中有潜力的部分。

参考答案：D

- 项目复杂性、规模和结构的不确定性属于 (19) 风险。

- (19) A. 项目 B. 技术 C. 经济 D. 商业

● **试题分析** 项目风险是指可能导致项目损失的不确定性，美国项目管理大师马克思怀德曼将其定义为某一事件发生给项目目标带来不利影响的可能性。

参考答案：A

- 以下程序设计语言中， (20) 更适合用来进行动态网页处理。

- (20) A. HTML B. LISP C. PHP D. JAVA/C++

● **试题分析** HTML 用于处理静态网页；LISP 一种基于 λ 演算的函数式编程语言。

PHP 是一种通用开源脚本语言。语法吸收了 C 语言、Java 和 Perl 的特点，利于学习，使用广泛，主要适用于 Web 开发领域。它可以比 CGI 或者 Perl 更快速地执行动态网页。用 PHP 做出的动态页面与其他的编程语言相比，PHP 是将程序嵌入到 HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成 HTML 标记的 CGI 要高许多；PHP 还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网。

C++ 是一个接近系统底层的综合的，支持面向对象和范编程的程序设计语言，适用于开发要求很高效率的程序，例如大型游戏、大型企业应用、系统应用等。

参考答案：C

- 在引用调用方式下进行函数调用是将 (21)。

(21) A. 实参的值传递给形参 B. 实参的地址传递给形参
C. 形参的值传递给实参 D. 形参的地址传递给实参

☞ 试题分析 引用调用是把实参（如 `int a`）的地址（`&a`）赋给形参（指针变量，比如 `*b`，这时 `b=&a`，即 `b` 指向变量 `a`），如果 `*b`（也即 `a` 对应的内存空间）发生变化，也就是变量 `a` 的值发生了变化。

参考答案：B

- 编译程序对高级语言源程序进行编译的过程中，要不断收集、记录和使用源程序中一些相关符号的类型和特征等信息，并将其存入 (22) 中。

(22) A. 符号表 B. 哈希表
C. 动态查找表 D. 栈和队列

☞ 试题分析 编译过程中编译程序不断汇集和反复查证出现在源程序中各种名字的属性和特征信息等有关信息。这些信息通常记录在一张或几张符号表中。符号表的每一项有两部分：一部分是名字（标识符）；一部分是名字属性（标识符的有关信息）。编译过程中，每当扫描器（词法分析器）识别出一个名字后，编译程序就查阅符号表，看其是否在符号表中。符号表在编译全过程的地位和作用非常重要，是进行上下文合法性检查和语义处理及代码生成的依据。符号表总体结构的设计和实现是与源语言的复杂性（包括词法结构、语法结构的复杂性）有关，还与对于编译系统在时间效率和空间效率方面的要求有关。

参考答案：A

- 设计操作系统时不需要考虑的问题是 (23)。

(23) A. 计算机系统中硬件资源的管理 B. 计算机系统中软件资源的管理
C. 用户与计算机之间的接口 D. 语言编译器的设计实现

☞ 试题分析 操作系统有两个重要作用：①通过资源管理提高计算机系统的效率，操作系统是计算机系统的资源管理者，它含有对系统软/硬件资源实施管理的一组程序。②改善人机界面，向用户提供友好的工作环境。明白操作系统的作用，就不难选出结果。

参考答案：D

- 假设某计算机系统中资源 `R` 的可用数为 6，系统中有 3 个进程竞争 `R`，且每个进程都需要 `i` 个 `R`，该系统可能会发生死锁的最小 `i` 值是 (24)。若信号量 `S` 的当前值为 -2，则 `R` 的可用数和等待 `R` 的进程数分别为 (25)。

(24) A. 1 B. 2 C. 3 D. 4
(25) A. 0、0 B. 0、1 C. 1、0 D. 0、2

☞ 试题分析 如果 `i=1`，即每个进程都需要 1 个 `R`，3 个进程同时运行需要 3 个 `R`，还剩 3 个 `R`，不会发生死锁。如果 `i=2`，即每个进程都需要 2 个 `R`，3 个进程同时运行需要 6 个 `R`，而 `R` 的可用数正好为 6，不会发生死锁。如果 `i=3`，即每个进程都需要 3 个 `R`，当 3 个进程分别占有 2 个 `R` 时，都需要再申请一个 `R` 资源才能正常运行，但此时已经没有 `R` 资源了，进程之间便出现了相互等待的状况，发生死锁。信号量的值小于 0，表示没有可用的资源，其绝对值表示阻塞队列中等待该资源的进程数。

参考答案：C D

- 某计算机系统页面大小为 4K，若进程的页面变换表如下所示，逻辑地址为十六进制 1D16H。该地址经过变换后，其物理地址应为十六进制 (26)。

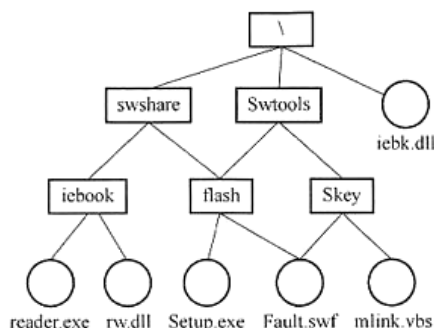
页号	物理块号
0	1
1	3
2	4
3	6

(26) A. 1024H B. 3D16H C. 4D16H D. 6D16H

● **试题分析** 页面大小为 4K，而 $4K=2^{12}$ ，因此逻辑地址的低 12 位对应页内地址，高位对应页号。题目中逻辑地址为十六进制 1D16H，一位十六进制数对应 4 位二进制数，3 位十六进制数则对应 12 位二进制数，因此 D16H 为页内地址，页号为 1。查页面变换表，页号 1 对应的物理块号为 3，将物理块号与页内地址 D16H 拼接起来即可得到物理地址 3D16H。

参考答案：B

- 若某文件系统的目录结构如下图所示，假设用户要访问文件 fault.swf，且当前工作目录为 swshare，则该文件的全文件名为 (27)，相对路径和绝对路径分别为 (28)。



- (27) A. fault.swf B. flash\fault.swf
C. swshare\flash\fault.swf D. \swshare\flash\fault.swf
(28) A. swshare\flash\和\flash\ B. flash\和\swshare\flash\
C. \swshare\flash\和 flash\ D. \flash\和\swshare\flash\

● **试题分析** 全文件名应该从根目录开始，因此为\swshare\flash\fault.swf。相对路径是从当前路径开始的路径，fault.swf 在当前工作目录 swshare 下的 flash 文件夹中，因此相对路径为 flash\。绝对路径是指从根目录开始的路径，即\swshare\flash\。

参考答案：D B

- 以下关于统一过程 UP 的叙述中，不正确的是 (29)。
(29) A. UP 是以用例和风险为驱动，以架构为中心，迭代并且增量的开发过程
B. UP 定义了四个阶段，即起始、精化、构建和确认阶段
C. 每次迭代都包含计划、分析、设计、构造、集成、测试以及内部和外部发布
D. 每个迭代有五个核心 workflow

● **试题分析** 统一过程(UP)的基本特征是用例和风险驱动，以架构为中心，受控的迭代式的增量开发。UP 定义了四个阶段：

起始阶段(inception)：该阶段的主要目的是建立项目的范围和版本，确定项目目标的可行性和稳定性，结束结果包括需求和用例。

精化阶段(elaboration)：该阶段的目的是对问题域进行分析，建立系统需求和架构，确定实现的可行性和稳定性，提交结果包括系统架构，问题领域、修改后的需求及项目开发计划等相关文档。

构建阶段(construction)：增量式开发可以交付给用户的软件产品。

移交阶段(transition)：目的是将软件产品交付给用户。

参考答案：B

- 某公司要开发一个软件产品，产品的某些需求是明确的，而某些需求则需要进一步细化。由于市场

竞争的压力，产品需要尽快上市，则开发该软件产品最不适合采用 (30) 模型。

(30) A. 瀑布 B. 原型 C. 增量 D. 螺旋

● **试题分析** 瀑布模型将软件生命周期划分为需求分析、软件设计、程序编写、软件测试和运行维护等基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。瀑布模型要求待开发的系统需求是完整的、简明的、一致的，其突出的缺点是不适应用户需求的变化，开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果。对于需求不确定的软件产品不适合采用瀑布模型。

参考答案：A

● 在屏蔽软件错误的容错系统中，冗余附加技术的构成不包括 (31)。

(31) A. 关键程序和数据的冗余存储及调用 B. 冗余备份程序的存储及调用
C. 实现错误检测和错误恢复的程序 D. 实现容错软件所需的固化程序

● **试题分析** 冗余技术又称储备技术，它是利用系统的并联模型来提高系统可靠性的一种手段。冗余附加件包括：独立设计的相同功能冗余备份程序的存储及调用；实现纠错检测及恢复的程序；为实现容错软件所需的固化程序。

参考答案：A

● 采用 McCabe 度量法计算下列程序图的环路复杂性为 (32)。

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

● **试题分析** 在一个强连通的有向图 G 中，环的个数为 $m-n+2p$ 。其中 m 为图中弧的个数， n 为图中的结点数， p 是图中强连通分量的个数。题图为一强连通图，故 $p=1$ ，而图中弧的个数为 8，结点数为 6，故环的个数为 $8-6+2=4$ 。

参考答案：C

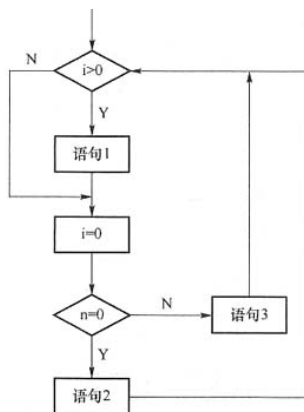
● 以下关于文档的叙述中，不正确的是 (33)。

(33) A. 文档仅仅描述和规定了软件的使用范围及相关的操作命令
B. 文档也是软件产品的一部分，没有文档的软件就不能称之为软件
C. 软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量
D. 高质量文档对于发挥软件产品的效益有着重要的意义

● **试题分析** 文档是软件开发使用和维护中的必备资料。文档能提高软件开发的效率，保证软件的质量，而且在软件的使用过程中有指导、帮助、解惑的作用，尤其在维护工作中，文档是不可或缺的资料。文档不仅包括软件开发过程中产生的文档，还包括硬件采购和网络设计中形成的文档；不仅包括有一定格式要求的规范文档，也包括系统建设过程中的各种往来文件、会议纪要、会计单据等资料形成的不规范文档。

参考答案：A

● 某搜索引擎在使用过程中，若要增加接受语音输入的功能，使得用户可以通过语音输入来进行搜索，此时应对系统进行 (34) 维护。

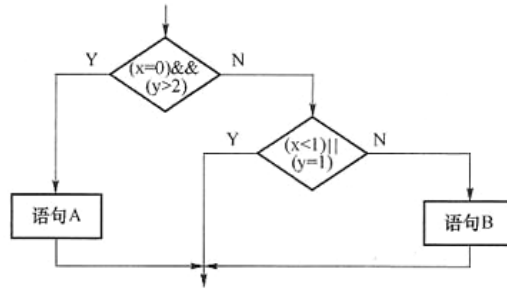


- (34) A. 正确性 B. 适应性 C. 完善性 D. 预防性

● **试题分析** 在搜索引擎中增加新的功能属于完善性维护。完善性维护是在系统运行过程中，根据用户要求，增加一些在系统分析和设计阶段没有规定的功能与性能特征，以及对处理效率和编写程序的改进。

参考答案：C

- 采用白盒测试方法对下图进行测试，设计了4个测试用例：①(x=0, y=3)，②(x=1, y=2)，③(x=-1, y=2)，④(x=3, y=1)。至少需要测试用例①②才能完成 (35) 覆盖，至少需要测试用例①②③或①②④才能完成 (36) 覆盖。



- (35) A. 语句 B. 条件 C. 判定/条件 D. 路径
(36) A. 语 B. 条件 C. 判定/条件 D. 路径

● **试题分析** 语句覆盖要求被测程序中的每一条语句至少执行一次，这种覆盖对程序执行逻辑的覆盖很低。

条件覆盖要求每一判定语句中每个逻辑条件的各种可能的值至少满足一次。判定/条件覆盖要求判定中每个条件的所有可能取值（真/假）至少出现一次，并使得每个判定本身的判定结果（真/假）也至少出现一次。路径覆盖则要求覆盖被测程序中所有可能的路径。

通过测试用例①(x=0, y=3)，能执行到语句 A，同时覆盖左侧路径；通过测试用例②(x=1, y=2)，能执行到语句 B，同时覆盖右侧路径；通过测试用例③(x=-1, y=2)或④(x=3, y=1)，什么也不执行，覆盖中间路径。

参考答案：A D

- (37) 是一个类与它的一个或多个细化类之间的关系，即一般与特殊的关系。

- (37) A. 泛化 B. 关联 C. 聚集 D. 组合

● **试题分析** 泛化表示类与类之间的继承关系，接口与接口之间的继承关系，或类对接口的实现关系。一般泛化关系是从子类指向父类的。

对于两个相对独立的对象，当一个对象的实例与另一个对象的一些特定实例存在同定的对应关系时，这两个对象之间为关联关系。关联体现的是两个类，或者类与接口之间语义级别的一种强依赖关系，这种关系一般是长期性的，而且双方的关系一般是平等的。关联可以是单向、双向的。

聚合是关联关系的一种特例，体现的是整体与部分、拥有的关系，即 has-a 的关系，此时整体与部分之间是可分离的，它们可以具有各自的生命周期，部分可以属于多个整体对象，也可以为多个整体对象共享。

组合也是关联关系的一种特例，体现的是一种 contains-a 的关系，这种关系比聚合更强，也称为强聚合；它同样体现整体与部分间的关系，但此时整体与部分是不可分的，整体的生命周期结束也就意味着部分的生命周期结束。

参考答案：A

- 某些程序设计语言中，在运行过程中当一个对象发送消息请求服务时，根据接收对象的具体情况将请求的操作与实现的方法进行连接，称为 (38)。

- (38) A. 静态绑定 B. 通用绑定 C. 动态绑定 D. 过载绑定

● **试题分析** 所谓静态绑定是指在程序编译过程中,把函数(方法或者过程)调用与响应调用所需的代码结合的过程。

动态绑定是指在执行期间判断所引用对象的实际类型,根据其实际的类型调用其相应的方法。

参考答案: C

- 在面向对象技术中,不同的对象在收到同一消息时可以产生完全不同的结果,这一现象称为 (39),它由 (40) 机制来支持。利用类的层次关系,把具有通用功能的消息存放在高层次,而不同的实现这一功能的行为放在较低层次,在这些低层次上生成的对象能够给通用消息以不同的响应。

(39) A. 绑定 B. 继承 C. 消息 D. 多态
(40) A. 绑定 B. 继承 C. 消息 D. 多态

● **试题分析** 多态性是同一操作作用于不同的类的实例,将产生不同的执行结果,即不同类的对象收到相同的消息时,得到不同的结果。在运行时,可以通过指向基类的指针,来调用实现派生类中的方法。多态是面向对象程序设计的重要特征之一,是扩展性在“继承”之后的又一重大表现。如果一个语言只支持类而不支持多态,只能说明它是基于对象的,而不是面向对象的。

参考答案: D B

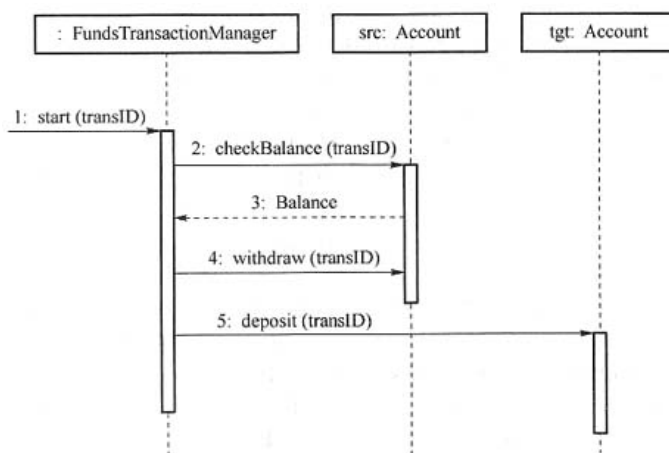
- 对一个复杂用例中的业务处理流程进行进一步建模的最佳工具是 UML 的 (41)。

(41) A. 状态图 B. 顺序图 C. 类图 D. 活动图

● **试题分析** 活动图是 UML 用于对系统的动态行为建模的一种常用工具,它描述活动的顺序,展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流,是内部处理驱动的流程。

参考答案: D

- 如下所示的 UML 序列图中, (42) 表示返回消息, Account 类必须实现的方法有 (43)。

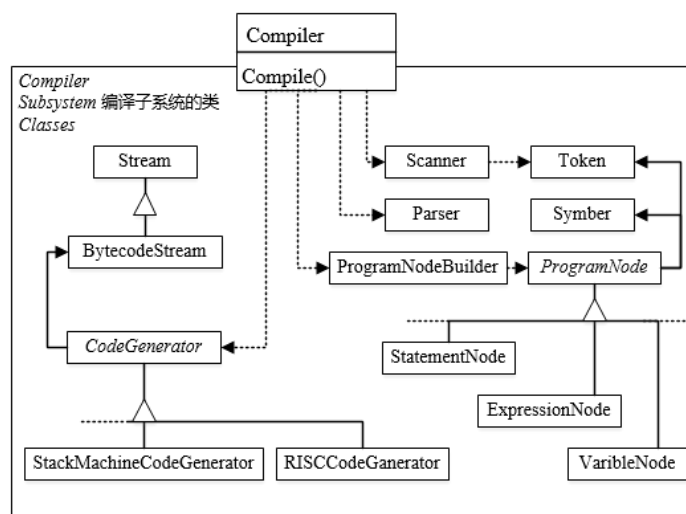


(42) A. tansID B. balance
C. withdraw D. deposit
(43) A. start0 B. checkBalance()和 withdraw()
C. deposit0 D. checkBalance()、withdraw()和 deposit()

● **试题分析** 在 URL 序列图中,调用消息用带实心箭头的实线表示,返回消息用带箭头的虚线表示,图中的编号为 1、2、3、5 的消息均为调用消息,只有编号为 3 的消息为返回消息。由题目可知, Fund Transaction Manager 要调用类 Ac-count 的 checkBalance()、withdraw()和 deposit()方法,类 Account 需要实现这 3 个方法。

参考答案: D B

- 下图所示为 (44) 设计模式,适用于 (45)。



- (44) A. 适配器(Adapter) B. 责任链(Chain of Responsibility)
 C. 外观(Facade) D. 桥接(Bridge)
- (45) A. 有多个对象可以处理一个请求，在运行时刻自动确定由哪个对象处理
 B. 想使用一个已经存在的类，而其接口不符合要求
 C. 类的抽象和其实现之间不希望有一个固定的绑定关系
 D. 需要为一个复杂子系统提供一个简单接口

试题分析 外观设计模式要求一个子系统的外部与其内部的通信必须通过一个统一的对象进行交互。外观设计模式提供一个高层次的接口使得子系统更易于使用。

外观模式适用情况如下：

- 1) 为复杂的子系统提供一个简单的接口；
- 2) 客户程序与抽象类的实现部分之间存在着很大的依赖性；
- 3) 构建一个层次结构的子系统时，适用外观模式定义子系统中每层的入口点。

参考答案： C D

● 下列设计模式中，(46) 模式既是类结构型模式，又是对象结构型模式。此模式与 (47) 模式类似的特征是，都给另一个对象提供了一定程度上的间接性，都涉及到从自身以外的一个接口向这个对象转发请求。

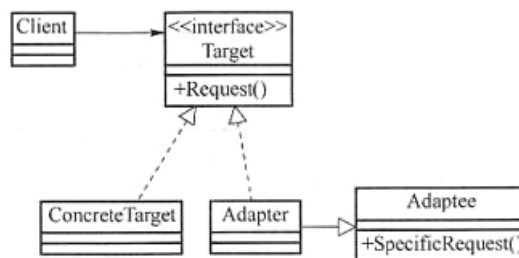
- (46) A. 桥接(Bridge) B. 适配器(Adapter)
 C. 组成(Composite) D. 装饰(Decorator)
- (47) A. 桥接(Bridge) B. 适配器(Adapter)
 C. 组成(Composite) D. 装饰器(Decorator)

试题分析 适配器模式(Adapter)用于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。

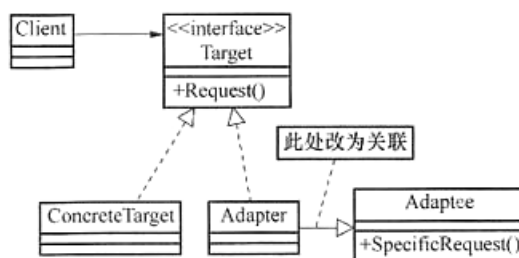
适配器有类结构和对象结构两种模式，如下图所示。在类适配器中，因为 Adapter 类既继承了 Adaptee（被适配类），也实现了 Target 接口，在 Client 类中我们可以根据需求选择并创建任一种符合需求的子类，来实现具体功能。在对象适配器中，Adapter 不是使用多继承或继承再实现的方式，而是使用直接关联，或者称为委托的方式。

装饰器模式用于动态地给一个对象添加一些额外的职责或者行为。装饰器模式提供了改变子类的灵活方案。装饰器模式在不必改变原类文件和使用继承的情况下，动态的扩展一个对象的功能。它是通过创建一个包装对象，也就是装饰来包裹真实的对象。当用于一组子类时，装饰器模式更加有用。如果你拥有一组子类（从一个父类派生而来），你需要在与子类独立使用情况下添加额外的特性，你可以使用装饰器模式，以避免代码重复和具体子类数量的增加。

参考答案: B D



类适配器



对象适配器

- 以下关于实现高级程序设计语言的编译和解释方式的叙述中，正确的是 (48)。

- (48) A. 在编译方式下产生源程序的目标程序，在解释方式下不产生
B. 在解释方式下产生源程序的目标程序，在编译方式下不产生
C. 编译和解释方式都产生源程序的目标程序，差别是优化效率不同
D. 编译和解释方式都不产生源程序的目标程序，差别在是否优化

☞**试题分析** 在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序要参与到程序的运行过程中，运行程序的控制权在解释程序。解释器翻译源程序时不产生独立的目标程序，而编译器则需要将源程序翻译成独立的目标程序。

参考答案: A

- 大多数程序设计语言的语法规则用 (49) 描述即可。

- (49) A. 正规文法 B. 上下文无关文法
C. 上下文有关文法 D. 短语结构文法

☞**试题分析** 形式语言理论中一种重要的变换文法，用来描述上下文无关语言，在乔姆斯基分层中称为 2 型文法。由于程序设计语言的语法基本上都是上下文无关文法，因此应用十分广泛。上下文无关文法拥有足够强的表达力来表示大多数程序设计语言的语法。另一方面，上下文无关文法又足够简单，使得我们可以构造有效的分析算法来检验一个给定字串是否是由某个上下文无关文法产生的。

参考答案: B

- 在某 C/C++ 程序中，整型变量 a 的值为 0 且应用在表达式“c=b/a”中，则最可能发生的情形是 (50)。

- (50) A. 编译时报告有语法错误 B. 编译时报告有逻辑错误
C. 运行时报告有语法错误 D. 运行时产生异常


☞**试题分析** 编译时 a 的值无法确定，表达式“c=b/a”符合 C/C++ 语言的语法逻辑，编译时不会报错。运行时，代入 a 的值，发生错误。

参考答案: D

- 为了保证数据库中数据的安全可靠和正确有效，系统在进行事务处理时，对数据的插入、删除或修

改的全部有关内容先写入__(51)__;当系统正常运行时,按一定的时间间隔,把数据库缓冲区内容写入__(52)__;当发生故障时,根据现场数据内容及相关文件来恢复系统的状态。


- (51) A. 索引文件 B. 数据文件 C. 日志文件 D. 数据字典
(52) A. 索引文件 B. 数据文件 C. 日志文件 D. 数据字典

 **试题分析** 数据库管理系统利用日志文件来进行事务故障恢复和系统故障恢复。在事务处理过程中,DBMS把事务开始、事务结束以及对数据库的插入、删除和修改的每一次操作写入日志文件。当系统正常运行时,按一定的时间间隔,把数据库缓冲区内容写入数据文件;一旦发生故障,DBMS的恢复系统利用日志文件撤销事务对数据库的改变,回退到事务的初始状态。

参考答案: C B

- “当多个事务并发执行时,任一事务的更新操作直到其成功提交的整个过程对其他事务都是不可见的”,这一性质通常被称为事务的__(53)__。

- (53) A. 原子性 B. 一致性 C. 隔离性 D. 持久性

 **试题分析** 事务具有原子性、一致性、隔离性和持久性。

事务的原子性表示事务执行过程中的任何失败都将导致事务所做的任何修改失效。

一致性表示当事务执行失败时,所有被该事务影响的数据都应该恢复到事务执行前的状态。

隔离性表示在事务执行过程中对数据的修改,在事务提交之前对其他事务不可见。

持久性表示已提交的数据在事务执行失败时,数据的状态都应该正确。

参考答案: C

- 假定某企业2014年5月的员工工资如下表所示:


2014年5月员工工资表

员工号	姓名	部门	基本工资	岗位工资	全勤奖	应发工资	扣款	实发工资
1001	王小龙	办公室	680.00	1200.00	100.00	1980.00	20.00	1960.00
1002	孙晓红	办公室	1200.00	1000.00	0.00	2200.00	50.00	2150.00
2001	赵晗珊	企划部	680.00	1200.00	100.00	1980.00	10.00	1970.00
2002	李丽敏	企划部	950.00	2000.00	100.00	3050.00	15.00	3035.00
3002	傅学君	设计部	800.00	1800.00	0.00	2600.00	50.00	2550.00
3003	曹海军	设计部	950.00	1600.00	100.00	2650.00	20.00	2630.00
3004	赵晓勇	设计部	1200.00	2500.00	0.00	3700.00	50.00	3650.00
4001	杨一凡	销售部	680.00	1000.00	100.00	1780.00	10.00	1770.00
4003	景昊星	销售部	1200.00	2200.00	100.00	3500.00	20.00	3480.00
4005	李建军	销售部	850.00	1800.00	100.00	2750.00	98.00	2652.00

查询人数大于2的部门和部门员工应发工资的平均工资的SQL语句如下:

SELECT __ (54) __
FROM 工资表
__ (55) __
__ (56) __;

- (54) A. 部门, AVG(应发工资)AS 平均工资 B. 姓名, AVG(应发工资)AS 平均工资
C. 部门, 平均工资 AS AVG(应发工资) D. 姓名, 平均工资 AS AVG(应发工资)
(55) A. ORDER BY 姓名 B. ORDER BY 部门
C. GROUP BY 姓名 D. GROUP BY 部门
(56) A. WHERE COUNT(姓名)>2 B. WHERE COUNT(DISTINCT(部门))>2
C. HAVING COUNT(姓名)>2 D. HAVING COUNT(DISTINCT(部门))>2

 **试题分析** 本题要查询的是“部门”和“部门员工应发工资的平均工资”,查询条件是“部门人数大于2”。选项中,将AVG(应发工资)命名为“平均工资”,重命名用AS子句实现:Old-name As New-name,可见(54)应选A。本题在查询中应按部门分组,并加上条件“部门人数大于2”。分组应采用

用 GROUP BY, 分组的依据是“部门”, 对分组加上限制使用 HAVING 子句, COUNT(姓名)可统计各个部门中的人数, 而 COUNT(部门)统计的则是部门数。因此, 完整的 SQL 语句应是:

```
SELECT 部门, AVG(应发工资)AS 平均工资
FROM 工资表
GROUP BY 部门
HAVING COUNT(姓名)>2;
```

参考答案: A D C

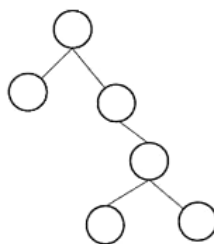
- 若对线性表的最常用操作是访问任意指定序号的元素, 并在表尾加入和删除元素, 则适宜采用 (57) 存储。

(57) A. 顺序表 B. 单链表 C. 双向链表 D. 哈希表

试题分析 采用顺序表 (即数组), 可以任意访问指定序号的元素, 便于在表尾加入和删除元素, 但不便于在表头插入和删除元素, 在表头操作时需要移动大量元素。要注意的是, 题目中要求在表尾加入和删除元素, 而不是在表头操作, 因此适宜采用顺序表。采用链表插入、删除元素较为方便, 但是访问指定序号的元素较为麻烦, 需要从头指针开始遍历。

参考答案: A

- 某二叉树如图所示, 若进行顺序存储 (即用一维数组元素存储该二叉树中的结点且通过下标反映结点间的关系, 例如, 对于下标为 i 的结点, 其左孩子的下标为 $2i$ 、右孩子的下标为 $2i+1$), 则该数组的大小至少为 (58); 若采用三叉链表存储该二叉树 (各个结点包括结点的数据、父结点指针、左孩子指针、右孩子指针), 则该链表的所有结点中空指针的数目为 (59)。

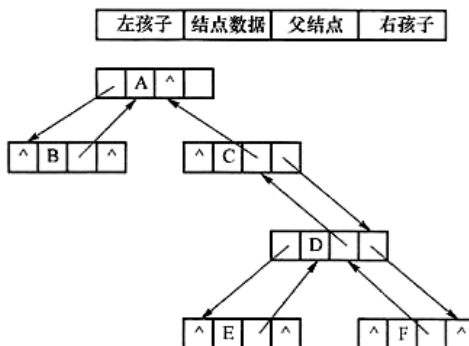


(58) A. 6 B. 10 C. 12 D. 15
(59) A. 6 B. 8 C. 12 D. 14

试题分析 采用顺序存储结构存储二叉树时, 一般的二叉树也必须按照完全二叉树的形式存储, 需要填上一些不存在的“虚结点”。题中二叉树的高度为 4, 需要的存储空间为 $2^4-1=15$, 如下:

A B C ^ ^ ^ D ^ ^ ^ ^ ^ E F

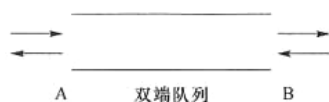
采用三叉链表存储二叉树, 如下:



可见, 空指针的数目为 8。

参考答案: D B

- 某双端队列如下图所示，要求元素进出队列必须在同一端口，即从 A 端进入的元素必须从 A 端出、从 B 端进入的元素必须从 B 端出，则对于 4 个元素的序列 e1、e2、e3、e4，若要求前 2 个元素(e1、e2)从 A 端口按次序全部进入队列，后两个元素(e3、e4)从 B 端口按次序全部进入队列，则可能得到的出队序列是(60)。



- (60) A. e1、e2、e3、e4 B. e2、e3、e4、e1
C. e3、e4、e1、e2 D. e4、e3、e2、e1

☞**试题分析** e1、e2 从 A 端口按次序进入队列，由于从 A 端进入的元素必须从 A 端出，则 e2 要先于 e1 出队；e3、e4 从 B 端口按次序进入队列，而从 B 端进入的元素必须从 B 端出，则 e4 要先于 e3 出队，只有选项 D 满足要求。

参考答案：D

- 实现二分查找（折半查找）时，要求查找表(61)。

- (61) A. 顺序存储，关键码无序排列 B. 顺序存储，关键码有序排列
C. 双向链表存储，关键码无序排列 D. 双向链表存储，关键码有序排列

☞**试题分析** 二分查找又称折半查找，优点是比较次数少，查找速度快，平均性能好；其缺点是要求待查表为有序表，且插入删除困难。因此，折半查找方法适用于不经常变动而查找频繁的有序列表。

二分查找算法要求：①必须采用顺序存储结构；②必须按关键字大小有序排列。

参考答案：B

- 某个算法的时间复杂度递归式 $T(n)=T(n-1)+n$ ，其中 n 为问题的规模，则该算法的渐进时间复杂度为(62)，若问题的规模增加了 16 倍，则运行时间增加(63)倍。

- (62) A. $O(n)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n^2 \lg n)$
(63) A. 16 B. 64 C. 256 D. 1024

☞**试题分析** 对于递归式，假设 $T(1)=1$ ，则：

$$T(n)=T(n-1)+n=T(n-2)+n-1+n=T(n-3)+n-2+n-1+n=1+2+\dots+n-1+n=n(n+1)/2$$

可见，时间复杂度为 $O(n^2)$ 。若问题的规模增加了 16 倍，则运行时间增加了 $16^2=256$ 倍。

参考答案：C C

- Prim 算法和 Kruskal 算法都是无向连通网的最小生成树的算法，Prim 算法从一个顶点开始，每次从剩余的顶点中加入一个顶点，该顶点与当前的生成树中的顶点的连边权重最小，直到得到一颗最小生成树；Kruskal 算法从权重最小的边开始，每次从不在当前的生成树顶点中选择权重最小的边加入，直到得到一颗最小生成树，这两个算法都采用了(64)设计策略，且(65)。

- (64) A. 分治 B. 贪心 C. 动态规划 D. 回溯
(65) A. 若网较稠密，则 Prim 算法更好 B. 两个算法得到的最小生成树是一样的
C. Prim 算法比 Kruskal 算法效率更高 D. Kruskal 算法比 Prim 算法效率更高

☞**试题分析** Prim 算法和 Kruskal 算法都是基于贪心算法的应用。Prim 算法的时间复杂度为 $O(n^2)$ ，与图中边数无关，该算法适合于稠密图。Kruskal 算法的时间复杂度只和边有关系，为 $O(e \log_2 e)$ ，由于 Kruskal 算法只与边有关，因此适合求稀疏图的最小生成树。

参考答案：B A

- IP 地址块 155.32.80.192/26 包含了(66)个主机地址，以下 IP 地址中，不属于这个网络的地址是(67)。

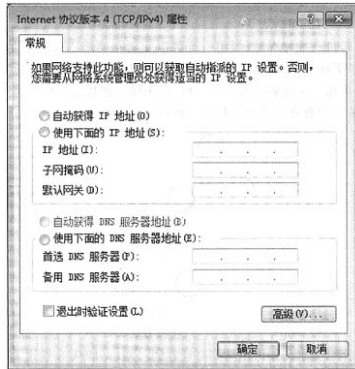
- (66) A. 15 B. 32 C. 62 D. 64
(67) A. 155.32.80.202 B. 155.32.80.195 C. 155.32.80.253 D. 155.32.80.191

☞**试题分析** 155.32.80.192/26 表示 32 位长度的 IP 地址中，前 26 位是网络前缀，后 6 位是主机号，

因此包含的主机地址个数为 $2^6-2=62$ ，主机地址范围为 155.32.80.193~155.32.80.254，显然 155.32.80.191 不属于这个网络。

参考答案：C D

- 校园网连接运营商的 IP 地址为 202.117.113.3/30，本地网关的地址为 192.168.1.254/24，如果本地计算机采用动态地址分配，在下图中应该如何配置 (68)。



- (68) A. 选取“自动获得 IP 地址”
B. 配置本地计算机 IP 地址为 192.168.1.×
C. 配置本地计算机 IP 地址为 202.115.113.×
D. 在网络 169.254.×.×中选取一个不冲突的 IP 地址

试题分析 由于本地计算机采用动态地址分配，因此选用“自动获得 IP 地址”，DHCP 服务器会为本地计算机动态分配 IP 地址。

参考答案：A

- 某用户在使用校园网中的一台计算机访问某网站时，发现使用域名不能访问该网站，但是使用该网站的 IP 地址可以访问该网站，造成该故障产生的原因有很多，其中不包括 (69)。

- (69) A. 该计算机设置的本地 DNS 服务器工作不正常
B. 该计算机的 DNS 服务器设置错误
C. 该计算机与 DNS 服务器不在同一子网
D. 本地 DNS 服务器网络连接中断

试题分析 本题主要考查网络故障判断的相关知识。如果本地的 DNS 服务器工作不正常或者本地 DNS 服务器网络连接中断都有可能该计算机的 DNS 无法解析域名，而如果直接将该计算机的 DNS 服务器设置错误也会导致 DNS 无法解析域名，从而出现使用域名不能访问该网站，但是使用该网站的 IP 地址可以访问该网站。但是该计算机与 DNS 服务器不在同一子网不会导致 DNS 无法解析域名的现象发生，通常情况下大型网络里面的上网计算机与 DNS 服务器本身就不在一个子网，只要路由可达 DNS 都可以正常工作。

参考答案：C

- 中国自主研发的 3G 通信标准是 (70)。

- (70) A. CDMA2000 B. TD-SCDMA C. WCDMA D. WiMAX

试题分析 国际电信联盟(ITU)在 2000 年 5 月确定 W-CDMA、CDMA2000 和 TDS-CDMA 三大主流无线接口标准，写入 3G 技术指导性文件《2000 年国际移动通讯计划》（简称 IMT-2000）。

W-CDMA：英文名称是 Wideband Code Division Multi-ple Access，中文译名为宽带码分多址，它可支持 384kbps 到 2Mbps 不等的数据传输速率，支持者主要以 GSM 系统为主的欧洲厂商。

CDMA2000：亦称 CDMA Multi-Carrier，由美国高通北美公司为主导提出，摩托罗拉、Lucent 和后来加入的韩国三星都有参与，韩国现在成为该标准的主导者。

TD-SCDMA：该标准是由中国独自制定的 3G 标准，由于中国的庞大的市场，该标准受到各大主要


电信设备厂商的重视，全球一半以上的设备厂商都宣布可以支持 TD-SC-DMA 标准。

参考答案：B

- Cloud computing is a phrase used to describe a variety of computing concepts that involve a large number of computers (71) through a real-time communication network such as the Internet. In science, cloud computing is a (72) for distributed computing over a network, and means the (73) to run a program or application on many connected computers at the same time.

The architecture of a cloud is developed at three layers: infrastructure, platform, and application. The infrastructure layer is built with virtualized compute storage and network resources. The platform layer is for general-purpose and repeated usage of the collection of software resources. The application layer is formed with a collection of all needed software modules for SaaS applications. The infrastructure layer serves as the (74) for building the platform layer of the cloud. In turn, the platform layer is foundation for implementing the (75) layer for SaaS application.

- | | | | |
|---------------------|----------------|----------------|------------------|
| (71) A. connected | B. implemented | C. optimized | D. virtualized |
| (72) A. replacement | B. switch | C. substitute | D. synonym (同义词) |
| (73) A. ability | B. approach | C. function | D. method |
| (74) A. network | B. foundation | C. software | D. hardware |
| (75) A. resource | B. service | C. application | D. software |

 **试题分析** 云计算是一个用来描述各种计算概念的一个短语，计算概念涉及大量计算机通过实时通信的网络，如 Internet，连接在一起。在科学研究中，云计算是分布式网络计算的同义词，意味着有能力同时在多台互连的计算机上运行一个程序或应用。

云的结构分为 3 层：基础设施、平台和应用。基础实施层由虚拟计算机存储和网络资源构成；平台层是具有通用性和复用性的软件资源的集合；应用层是云上针对 SaaS 应用的所有应用程序的集合。基础设施层时建立平台层的硬件基础；相应地，平台层是执行应用层 SaaS 应用的基础。

参考答案：A D A D C

全国计算机技术与软件专业技术资格考试

2014 年上半年 软件设计师 下午试卷与解析

试题一（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某巴士维修连锁公司欲开发巴士维修系统，以维护与维修相关的信息，该系统的主要能如下：

（1）记录巴士 ID 和维修问题。巴士到车库进行维修，系统将巴士基本信息和 ID 记录在巴士列表文件中，将待维修机械问题记录在维修记录文件中，并生成维修订单。

（2）确定所需部件，根据维修订单确定维修所需部件，并在部件清单中进行标记。

（3）完成维修，机械师根据维修记录文件中的待维修机械问题，完成对巴士的维修，登记维修情况；将机械问题维修情况记录在维修记录文件中，将所用部件记录在部件清单中，并将所用部件清单发送给库存管理系统以对部件使用情况进行监控，巴士司机可查看已维修机械问题。

（4）记录维修工时，将机械师提供的维修工时记录在人事档案中，将维修总结发送给主管进行绩效考核。

（5）计算维修总成本。计算部件清单中实际所用部件、人事档案中所用维修工时的总成本；将维修工时和所用部件成本详细信息给会计进行计费。

现采用结构化方法对巴士维修系统进行分析与设计，获得如图 1 所示的上下文数据流图和图 2 所示的 0 层数据流图。

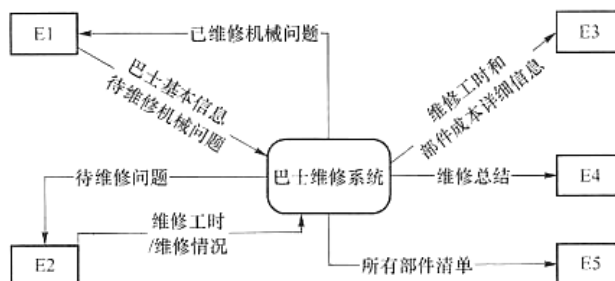


图 1 上下文数据流图

【问题 1】（5 分）

使用说明中的词语，给出图 1 中的实体 E1~E5 的名称。

答案：E1：巴士司机 E2：机械师 E3：会计 E4：主管 E5：库存管理系统

【解析】

本题考查数据流图(DFD)的应用，采用结构化方法进行系统分析与设计，是一道传统题目，要求考生细心分析题目中所描述的内容。DFD 是一种便于用户理解、分析系统数据流的图形化建模工具，是系统逻辑模型的重要组成部分。

本问题考查顶层 DFD。顶层 DFD 一般用来确定系统边界。结合图 1 上下文数据流图以及根据系统功能的描述(3)“巴士司机可查看已维修问题”可知，E1 为“巴士司机”；根据系统功能描述(4)“机械师提供的维修工时记录在人事档案中”和“维修总结发送给主管进行考核”可知，E2 为“机械师”，E4 为“主管”；根据系统功能描述(5)“维修工时和所用成本详细信息给会计进行计费”可知，E3 为“会计师”，由“所用部件清单发送给库存管理系统”可推断，E5 为库存管理系统。

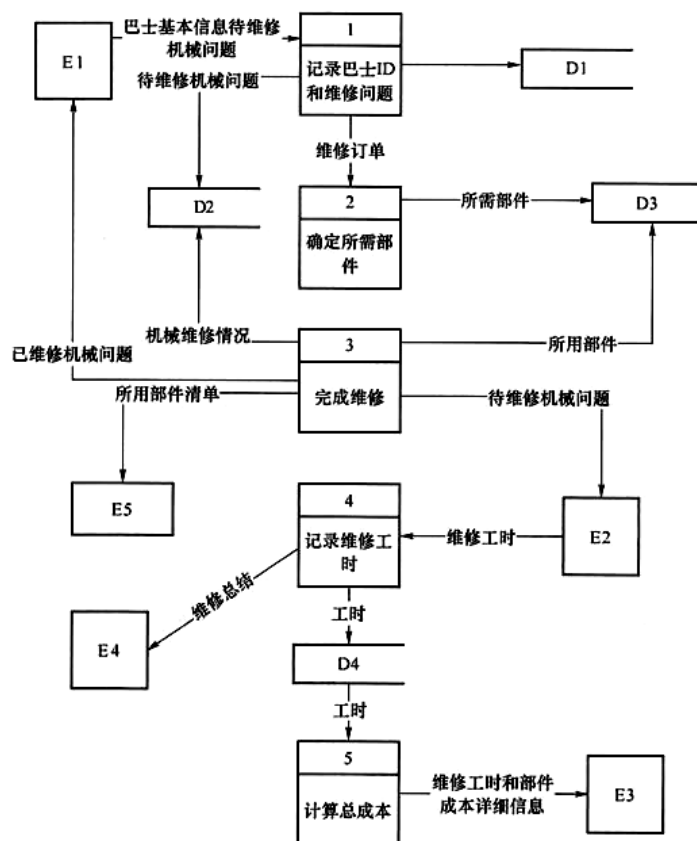


图2 层数据流图

【问题2】（4分）

使用说明中的词语，给出图2中的数据存储D1~D4的名称

答案：D1：巴士列表文件 D2：维修记录文件 D3：部件清单 D4：人事档案

【解析】

本问题考查DFD中数据存储的确定。本题涉及的数据存储有5个，即巴士列表文件、维修记录文件、部件清单和人事档案。接下来需要结合图2所示的0层数据流进行确定。D1与“记录巴士ID和维修问题”有关，根据题意，D1是“巴士列表文件”；待维修机械问题记录在D2中，可知，D2为“维修记录文件”；所需部件记录在D3中，由题意可知D3为“部件清单”；D4记录的是工时，由题意，D4是“人事档案”。

【问题3】（4分）

说明图2中所存在的问题。

答案：缺失以下数据流：

(1)名称：待维修机械问题；起点：D2；终点：3完成维修。

(2)名称：实际所用部件；起点：D3；终点：5计算总成本。

【解析】

本问题考查DFD中的数据流。在分查DFD中，需要保持父图和子图的平衡，即父图中某加工的输入/输出数据流必须与其子图的输入/输出数据流在数量和名字上相同，或者父图的一个输入(或输出)数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是父图中的一个数据流。题中(5)的描述“计算部件清单中实际所用部件、人事档案中所用维修工时的总成本”可知，缺少关于实际所用部件数据流，因此增加一个数据流实际所用部件。起点：D3，终点：5计算总成本。根据“完成维修”功能中的描述，机械师可根据维修记录文件中的待维修机械问题，完成巴士维修，因此

缺少数据流“待维修机械问题”，起点：D2，终点：3 完成维修。

【问题 4】（2 分）

根据说明和图中术语，采用补充数据流的方式，改正图 2 中的问题。要求给出所补充数据流的名称、起点和终点。

答案：补充以下数据流：

(1)名称：待维修机械问题；起点：D2；终点：3 完成维修。

(2)名称：实际所用部件；起点：D3；终点：5 计算总成本。

【解析】

根据问题 3 的解析，补充以下数据流：实际所用部件，起点：D3，终点：5 计算总成本；待“维修机械问题”，起点：D2，终点：3 完成维修。

试题二（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某家电销售电子商务公司拟开发一套信息管理系统，以方便对公司的员工、家电销售、家电厂商和客户等进行管理。

[需求分析]

系统需要维护电子商务公司的员工信息、客户信息、家电信息和家电厂商信息等，员工信息主要包括：工号、姓名、性别、岗位、身份证号、电话、住址，其中岗位包括部门经理和客服等，客户信息主要包括：客户 ID、姓名、身份证号、电话、住址、账户余额。家电信息主要包括：家电条码、家电名称、价格、出厂日期、所属商，家电厂商信息包括：厂商 ID、厂商名称、电话、法人代表信息、厂址。

电子商务公司根据销售情况，由部门经理向家电厂商订购各类家电，每个家电厂商只能由一名部门经理负责。

客户通过浏览电子商务公司网站查询家电信息，与客服沟通获得优惠后，在线购买。

[概念模型设计]

根据需求阶段收集的信息，设计的实体联系图(不完整)如下图所示。

[逻辑结构设计]

根据概念模型设计阶段完成的实体联系图，得出如下关系模式[不完整]

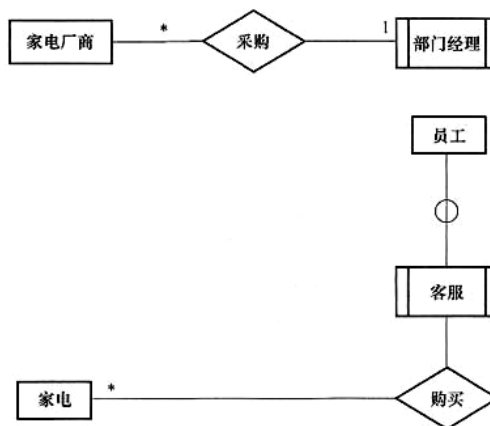
客户(客户 ID、姓名、身份证号、电话、住址、账户余额)

员工(工号、姓名、性别、岗位、身份证号、电话、住址)

家电[家电条码、家电名称、价格、出厂日期、(4)]

家电厂商[厂商 ID、厂商名称、电话、法人代表信息、厂址、(5)]

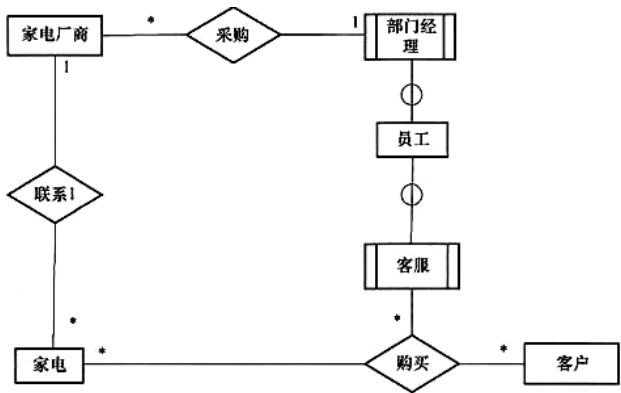
购买[订购单号、(6)、金额]



【问题 1】(5 分)

补充图中的联系和联系的类型。

答案：



【解析】

本题考查数据库设计。设计考点有数据库的概念结构设计和逻辑结构设计。

根据题意，每个家电厂商可以购买多种家电，因此家电厂商和家电之间是一对多的联系。一种家电可由多个客户购买，一个客户可购买多件家电，因此家电和客户之间是多对多的联系，由此可画出完整的实体联系图。

【问题 2】(5 分)

根据题干中的图，将逻辑结构设计阶段生成的关系模式中的空(1)~(3)补充完整，用下划线支出“家电”、“家电厂商”和“购买”关系模式的主键。

答案：(1)厂商 ID (2)工号 (3)家电条码，客户 ID，工号 (4)家电关系的主键：家电条码 (5)家电厂商关系的主键：厂商 ID (6)购买关系的主键：订单号

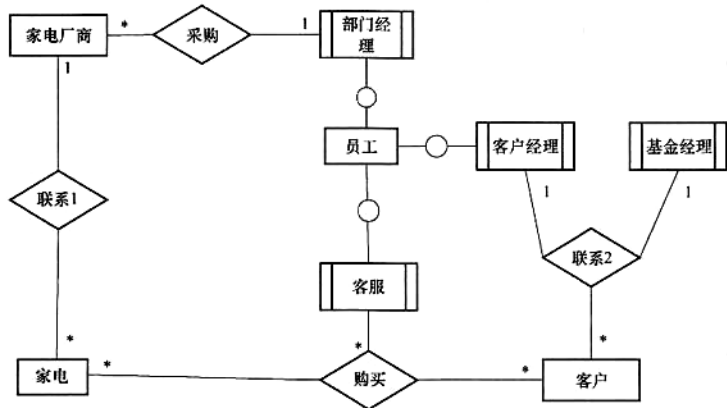
【解析】

电子商务公司根据销售情况，由部门经理向家电厂商订购各类家电，每个家电厂商只能由一名部门经理负责，家电厂商中需要保存是由哪名部门经理负责的，因此这里需要有员工的工号，这样才能在员工中找到唯一对应的员工。

【问题 3】(5 分)

电子商务公司的主营业务是销售各类家电，对账户有余额的客户，还可以联合第三方基金公司提供理财服务，为此设立客户经理岗位。客户通过电子商务公司的客户经理和基金公司的基金经理进行理财，每名客户只有一名客户经理和一名基金经理负责，客户经理和基金经理均可负责多名客户。请根据该要求，对题干图进行修改，画出修改后的实体间联系和联系的类型。

答案：



【解析】

由题意可知，可增设实体客户经理和基金经理，客户和客户经理、基金经理都有联系，且客户经理和客户、基金经理和客户都是一对多的联系，由此可补充题干图。

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某高校图书馆欲建设一个图书馆管理系统，目前已经完成了需求分析阶段的工作，功能需求均使用用例进行描述，其中用例“借书(Check Out 800ks)”的详细描述如下。

参与者：读者<Patron>

典型事件流：

1. 输入读者 ID；
2. 确认该读者能够借阅图书，并记录读者 ID；
3. 输入所要借阅的图书 ID；
4. 根据图书目录中的图书 ID 确认该书可以借阅，计算归还时间，生成借阅记录；
5. 通知读者图书归还时间；

重复步骤 3~5，直到读者结束借阅图书。

备选事件流：

2a. 若读者不能借阅图书，说明读者违反了图书馆的借书制度(例如，没有支付借书费用等)。

①告知读者不能借阅，并说明拒绝借阅的原因；

②本用例结束。

4a. 读者要借阅的书无法外借。

①告知读者本书无法借阅；

②回到步骤 3。

说明：图书的归还时间与读者身份有关。如果读者是教师，图书可以借阅一年；如果是学生，则只能借阅 3 个月，读者 ID 中包含读者身份信息，现采用面向对象方法开发该系统，得到如图 1 所示的系统类模型(部分)；以及如图 2 所示的系统操作。“CheckOut(bookID).(借书)”通信图”(或协作图)。

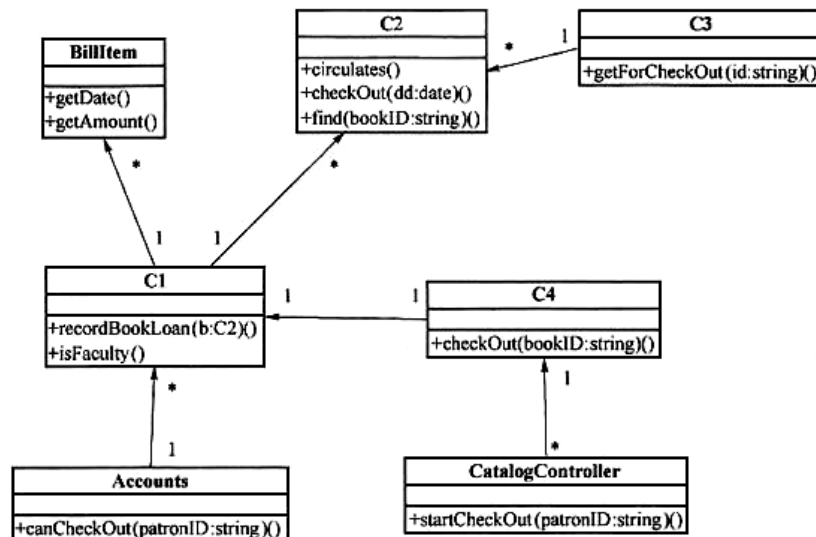


图 1 系统类模型

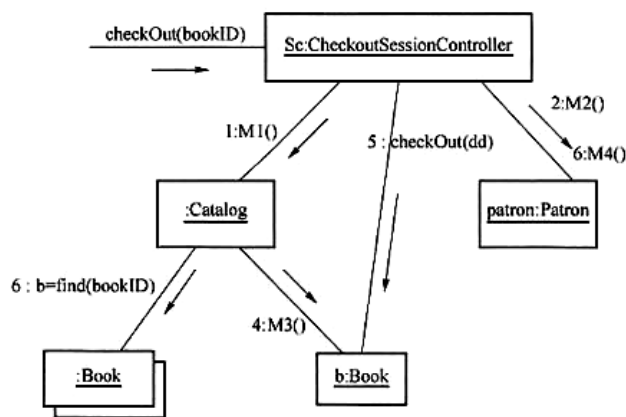


图2 系统操作 checkout 的通信图

【问题 1】（5 分）

根据说明中的描述，以及图 1 和图 2，给出图 1 中 C1-C4 处所对应的类名（类名使用图 1 和图 2 中给出的英文词汇）。

答案：C1:Patron C2:Book C3:Catalog C4:CheckoutSessioncontroller

【解析】

根据系统类模型，我们可以各个类之间的关联关系。

首先从类 Accouts 中的 canCheckOut(patronID:string)方法，可以看出 Accouts 关联 Patron，因此图中 C1 为 Pa-tron。

C1 为 Patron，则 C1 必会与书关联，从 C1 中的 record-BookLoad(b,C2)，可以看出 C1 关联 C2。因此 C2 为 Book。

C2 为 Book，根据系统操作 checkOut 的通信图，可以看出与 Book 关联的是 Catalog，因此 C3 为 Catalog。

结合两图，则可以得出 C4 为 chec-koutSessioncon-troller。

【问题 2】（5 分）

根据说明中的描述，以及图 1 和图 2，给出图 2 中 M1-M4 处所对应的方法名(方法名使用图 1 和图 2 中给出的英文词汇)。

答案：M1():getforcheck(bookID) M2():isFaculty M3():circulateso M4():recordBookLoan

【解析】

结合典型事件流：

1. 输入读者 ID；

2. 确认该读者能够借阅图书，并记录读者 ID；

以上两步实际上就是判断读者是不是老师，也就是 is-Faculty()，因此 M2 为 isFaculty()；

3. 输入所要借阅的图书 ID；对应的操作就是 M1: get- {forcheck(bookID)；

4. 根据图书目录中的图书 ID 确认该书可以借阅，计算归还时间，生成借阅记录；对应的操作就是 M3:circulates()；

5. 通知者图书归还时间，对应的操作就是 M4:record-BookLoan()。

【问题 3】（5 分）

用例“借书”的备选事件流 4a 中，根据借书制度来判定读者能否借阅图书，随着图书馆的借书制度会不断地扩充，并需要根据图书馆的实际运行情况来调整具体使用哪些制度，为满足这一要求，在原有类设计的基础上，可以采用何种设计模式？简要说明原因。

答案：应采用策略模式，策略模式的优势在于可以灵活的添加对同一问题的不同处理方案，这与题目要求非常吻合。

试题四（15 分）

阅读下列说明和 C 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

采用归并排序对 n 个元素进行递增排序时，首先将 n 个元素的数组分成各含 $n/2$ 个元素的两个子数组，然后用归并排序对两个子数组进行递归排序，最后合并两个已经排序的子数组得到排序结果。

下面的 C 代码是对上述归并算法的实现，其中的常量和变量说明如下：

arr:待排序数组

P,q,r:一个子数组的位置从 P 到 q，另一个子数组的位置从 q+1 到 r

begin,end:待排序数组的起止位置

left,right:临时存放待合并的两个子数组

n1,n2:两个子数组的长度

i,j,k:循环变量

mid:临时变量

[C 代码]

```
#include<stdio, h>
```

```
#include<stdlib, h>
```

```
Define MAX 65536
```

```
void merge(int arr [ ],int p,int q,int r) {
```

```
int * left,* right;
```

```
int n1,n2,i,j,k;
```

```
n1=q-p+1;
```

```
n2=r-q;
```

```
If(left=(int *)malloc((n1+1) * sizeof(int)))=NULL) {
```

```
Perror( "malloc error" );
```

```
exit(1)
```

```
}
```

```
If((right = (int *)malloc((n2+1) * sizeof(int)))=NULL)
```

```
Perror("malloc error");
```

```
exit(1);
```

```
}
```

```
for(i=0;i<n1;i++){
```

```
left[i]=arr [p+i];
```

```
}
```

```
left[i]=MAX;
```

```
for(i=0;i<n2;i++){
```

```
right[i]=arr[q+i+1]
```

```
}
```

```
right[i]=MAX;
```

```
i=0;j=0;
```

```
For(k=p;_____;k++){
```

```
If(left[i]>right[j] {
```

```
_____
```

```
j++;
```

```
}else{
```

```
arr[k]=left[i];
```

```
i++;
```

```
}
```

```

    }
    }
    Void merge Sort(int arr[ ], int begin, int end) {
    int mid;
    if(____){
    mid=(begin + end)/2;
    merge Sort(arr,begin,mid);
    ____;
    Merge(arr,begin,mid,end);
    }
    }

```

【问题 1】(5 分)

根据以上说明和 C 代码，把横线处补充完整。

答案: $k \leq r$ $arr[k]=right[j]$ $begin < end$ $mergeSort(arr, mid+1, end)$

【解析】

首先，函数 `void merge(int arr[], int p, int q, int r)` 的意思是：对子数组 `arr[p...q]` 和子数组 `arr[q+1..r]` 进行合并。因此第一空为 $k \leq q$ ；由于是采用归并排序对 n 个元素进行递增排序，所以第二空是将 `left[i]` 和 `right[j]` 的小者存放到 `arr[k]` 中去，即 `arr[k]=right[j]`；当数组长度为 1 时，停止递归，因为此时该数组有序，则第三空为 $begin < end$ ，即数组至少有两个元素才进行递归。合并了 `begin` 到 `mid` 之间的元素，继续合并 `mid+1` 到 `end` 之间的元素，则第四空为 `mergeSort(arr, mid+1, end)`。

【问题 2】(5 分)

根据题干说明和以上 C 代码，算法采用了_____算法设计策略，分析时间复杂度时，列出其递归式位_____，接触渐进时间复杂度_____ (用 O 符号表示)。空间复杂度为_____。(用 O 符号表示)

答案: 分治 $T(n)=2T(n/2)+O(n)$ $O(n \log n)$ $O(n)$

【解析】

归并算法实际上就是将数组一直往下分割，直到分割到由一个元素组成的 n 个子数组，再往上两两归并。

将数组进行分割需要 $\log N$ 步，因为每次都是讲数组分割成两半 ($2x=N$, $x=\log N$)。

合并 N 个元素，需要进行 N 步，也就是 $O(N)$ ，则总的时间复杂度为 $O(N \log N)$ 。

合并过程中，使用了 n 个中间变量存储，`left=(int*)malloc((n+1)*sizeof(int))`。所以空间复杂度为 $O(n)$ 。

推导递归式：

假设 n 个元素进行归并排序需要 $T(n)$ ，可以将其分割成两个分别有 $n/2$ 个元素的数组分别进行归并，也就是 $2T(n/2)$ ，在将这两个合并，需要 $O(n)$ 的时间复杂度，则推导公式为 $T(n)=2T(n/2)+O(n)$ 。

【问题 3】(5 分)

两个长度分别为 n_1 和 n_2 的已经排好序的子数组进行归并，根据上述 C 代码，则元素之间比较次数为_____。

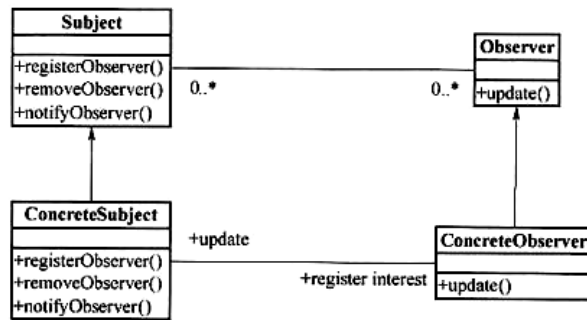
答案: n_1+n_2

试题五 (15 分)

阅读下列说明和 C++ 代码，回答问题，将答案填入相应横线处。

【说明】

某实验室欲建立一个实验室环境监测系统，能够显示实验室的温度、湿度以及洁净度等环境数据。当获取到最新的环境测量数据时，显示的环境数据能够更新现在采用观察者(observer)模式来开发该系统，观察者模式的类图如下图所示。



(c++代码)

```

#include<iostream>
#include<vector>
Using namespace std;
Class Observer {
Public:
Virtual void update(float temp,float humidity,float cleanness)=0;
};
Class Subject {
Public:
Virtual void registerObserver(Observer * o)=0;//注册对主题感兴趣的观察者
Virtual void removeObserver(Observer * o)=0;//删除观察者
Virtual void notifyObservers14=0;//当主题发生变化时通知观察者
Class EnvironmentData:public __ (1) __ {
Private:
Vector<Observer*> observers;
Float temperature,humidity,cleanness;
Public:
Void registerObserver(Observer * o) {observers.push back(o); }
Void removeObserver(Observer * o)< /*代码省略*/
Void notifyObservers15 {
for(vector<Observe*>::const iterator it=observers, begin14;it!=observers, end14;it++)
{ __ (2) __; }
}
Void measurementsChanged14{ __ (3) __; }
Void setMeasurements(float temperature,float humidity,float cleanness) {
this->temperature=temperature;
this->humidity=humidity;
this->cleanness=cleanness;
__ (4) __;
}
};
Class CurrentConditionsDisplay:public __ (5) __ {
private:
float temperature,humidity,cleanness;
Subject * envData;
Public:
CurrentConditionsDisplay(Subject * envData) {
this->envData=envData;
}
}
  
```

____(6)____:

```
Void update(float temperature,float humidity,float cleanness> {this->temperature=temperature;
This->humidity=humidity;
This->cleanness=cleanness;
display15;
}
Void display15{/*代码省略*/}
int main15{
EnvironmentData * envData=new EnvironmentData0;
CurrentConditionsDisplay * currentDisplay=new CurrentConditionsDisplay(envData);
envData->setMeasurements(80,65,30.4f);
return 0;
}
```

答案:

- | | |
|-----------------------|----------------------------------------------------|
| (1) Subject | (2) (*it)->update(temperature,humidity,clean-ness) |
| (3) notifyObservers() | (4) measurementsChanged() |
| (5) Observer() | (6) this->envData->registerObserver(this) |

【解析】

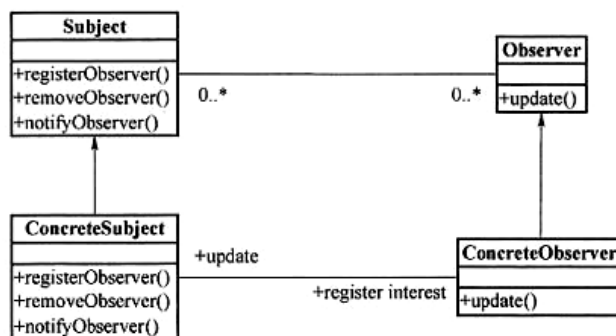
当主题中的环境发生变化，也就是调用 measurementsChanged()，会通知观察者，即调用 notifyObservers()，在 notifyObservers()方法中，观察者会调用自身的 update(float temperature,float humidity,float cleanness)。在主题的 registerObserver(Observer*o)中会注册对主题感兴趣的观察者。

试题六（15 分）

阅读下列说明和 JAVA 代码，回答问题，将答案填入相应横线处。

【说明】

某实验室欲建立一个实验室环境监测系统，能够显示实验室的温度、湿度以及洁净度等环境数据。当获取到最新的环境测量数据时，显示的环境数据能够更新现在采用观察者(observer)模式来开发该系统，观察者模式的类图如下图所示。



[Java 代码]

```
import java.util.
interface Observer {
Public void update(float temp,float humidity,float cleanness);
interface Subject {
Public void registerObserver(Observer o);    //注册对主题感兴趣的观察者
Public void removeObserver(Observer o);    //删除观察者
Public void notifyObservers();    //当主题发生变化时通知观察者
Class EnvironmentDataimplements ____ (1) ____ {
```

```

Private ArrayList observers;
Private float temperature,humidity,cleanness;
Public EnvironmentData0 { observers=new ArrayList0; }
Public void registerObserver(Observer o) { observers.add(o); }
Public void removeObserver(Observer o) { /*代码省略*/ }
public void notifyObservers20 {
For (int i=0; i<observers.size20;i++) {
Observer observer=(Observer)observers.get(i);
(2) ;
}
public void measurementsChanged20 { (3) ; }
public void setMeasurements(float temperature,float humidity,float cleanness) {
this.temperature=temperature;
this.humidity=humidity;
this.cleanness=cleanness;
(4) ;
}
}
Class CurrentConditionsDisplay implements (5)
private float temperature;
private float humidity;
private float cleanness;
private Subject envData;
public CurrentConditionsDisplay(Subject envData) {
this.envData=envData;
(6) ;
}
Public void update(float temperature,float humidity,float cleanness) {
this.temperature=temperature;
this.humidity=humidity;
this.cleanness=cleanness;
Display20;
}
Public void display0( /* 代码省略 */)
}
Class EnvironmentMonitor{
Public static void main(String [ ] args) {
EnvironmentData envData=new EnvironmentData0;
CurrentConditionsDisplay currentDisplay=new CurrentConditionsDisplay(envData);
envData.setMeasurements(80,65,30.4f);
)
Public void display20( /*代码省略*/ )
}
Class EnvironmentMonitor{
Public static void main (String[] args) {
EnvironmentData envData=new EnvironmentData0;
CurrentConditionsDisplay currentDisplay=new CurrentConditionsDisplay(envData);
envData.setMeasurements(80,65,30.4f);
}
}

```

答案:

- (1) Subject
- (2) observer.update(temperature,humidity,cleanness)
- (3) notifyObservers()
- (4) measurementsChanged()
- (5) Observer()
- (6) envData.registerObserver(this)

【解析】

EnvironmentData 是环境数据，也就是我们要监测的对象，即主题(Subject)，因此(1)处为 Subject。(2)处为通知观察者，因此遍历观察者容器，遍历到一个观察者对象，则更新该观察者的数据，即调用观察者的 update() 方法，当环境数据变化时，需要通知观察者，因此(4)处是调用环境变化方法 measurementsChanged()，通过此方法通知观察者更新数据，因此，(3)处为 notifyObservers()。

根据 CurrentConditionsDisplay 类中的 update()方法可知：CurrentConditionsDisplay 是个观察者，因此(5)处为 Observer(6)是将观察者添加到主题中去。