

第1章 数据结构与算法

经过对部分考生的调查以及对近年真题的总结分析，笔试部分经常考查的是算法复杂度、数据结构的概念、栈、二叉树的遍历、二分法查找，读者应对此部分进行重点学习。

详细重点学习知识点：

1. 算法的概念、算法时间复杂度及空间复杂度的概念
2. 数据结构的定义、数据逻辑结构及物理结构的定义
3. 栈的定义及其运算、线性链表的存储方式
4. 树与二叉树的概念、二叉树的基本性质、完全二叉树的概念、二叉树的遍历
5. 二分查找法
6. 冒泡排序法

1.1 算法

考点1 算法的基本概念

计算机解题的过程实际上是在实施某种算法，这种算法称为计算机算法。

1. 算法的基本特征：可行性、确定性、有穷性、拥有足够的情报。

2. 算法的基本要素：

(1) 算法中对数据的运算和操作

一个算法由两种基本要素组成：一是对数据对象的运算和操作；二是算法的控制结构。

在一般的计算机系统中，基本的运算和操作有以下4类：算术运算、逻辑运算、关系运算和数据传输。

(2) 算法的控制结构：算法中各操作之间的执行顺序称为算法的控制结构。

描述算法的工具通常有传统流程图、N-S结构化流程图、算法描述语言等。一个算法一般都可以用顺序、选择、循环3种基本控制结构组合而成。

考点2 算法复杂度

1. 算法的时间复杂度

算法的时间复杂度是指执行算法所需要的计算工作量。

同一个算法用不同的语言实现，或者用不同的编译程序进行编译，或者在不同的计算机上运行，效率均不同。这表明使用绝对的时间单位衡量算法的效率是不合适的。撇开这些与计算机硬件、软件有关的因素，可以认为一个特定算法“运行工作量”的大小，只依赖于问题的规模（通常用整数 n 表示），它是问题规模的函数。即

$$\text{算法的工作量} = f(n)$$

2. 算法的空间复杂度

算法的空间复杂度是指执行这个算法所需要的内存空间。

一个算法所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间以及算法执行过程中所需要的额外空间。其中额外空间包括算法程序执行过程中的工作单元以及某种数据结构所需要的附加存储空间。如果额外空间量相对于问题规模来说是常数，则称该算法是原地工作的。在许多实际问题中，为了减少算法所占的存储空间，通常采用压缩存储技术，以便尽量减少不必要的额外空间。



疑难解答：算法的工作量用什么来计算？

算法的工作量用算法所执行的基本运算次数来计算，而算法所执行的基本运算次数是问题规模的函数，即算法的工作量 $=f$

(n)，其中n是问题的规模。

1.2 数据结构的基本概念

考点3 数据结构的定义

数据结构作为计算机的一门学科，主要研究和讨论以下三个方面：

- (1) 数据集合中个数据元素之间所固有的逻辑关系，即数据的**逻辑结构**；
- (2) 在对数据元素进行处理时，各数据元素在计算机中的存储关系，即数据的**存储结构**；
- (3) 对各种数据结构进行的运算。

数据：是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素：是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。

数据对象：是性质相同的数据元素的集合，是数据的一个子集。

数据的逻辑结构是对数据元素之间的逻辑关系的描述，它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。数据的逻辑结构有两个要素：一是**数据元素的集合，通常记为D**；二是**D上的关系**，它反映了**数据元素之间的前后件关系，通常记为R**。一个数据结构可以表示成

$$B = (D, R)$$

其中**B表示数据结构**。为了反映D中各数据元素之间的前后件关系，一般用二元组来表示。

数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构（也称数据的物理结构）。

由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同，因此，为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系（即前后件关系），在数据的存储结构中，不仅要存放各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的存储结构有顺序、链接、索引等存储结构。而采用不同的存储结构，其数据处理的效率是不同的。因此，在进行数据处理时，选择合适的存储结构是很重要的。

考点4 线性结构与非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度，一般将数据结构分为两大类型：线性结构与非线性结构。如果一个非空的数据结构满足下列两个条件：

- (1) 有且只有一个根结点；
- (2) 每一个结点最多有一个前件，也最多有一个后件。

则称该数据结构为线性结构。**线性结构又称线性表**。在一个线性结构中插入或删除任何一个结点后还应该是线性结构。如果一个数据结构不是线性结构，则称之为非线性结构。

☹ 疑难解答：空的数据结构是线性结构还是非线性结构？

一个空的数据结构究竟是属于线性结构还是属于非线性结构，这要根据具体情况来确定。如果对该数据结构的算法是按线性结构的规则来处理的，则属于线性结构；否则属于非线性结构。

1.3 栈及线性链表

考点5 栈及其基本运算

1. 栈的基本概念

栈是限定只在一端进行插入与删除的线性表，通常称插入、删除的这一端为栈顶，另一端为栈底。当表中没有元素时称为空栈。栈顶元素总是后被插入的元素，从而也是最先被删除的元素；栈底元素总是最先被插入的元素，从而也是最后才能被删除的元素。栈是按照“先进后出”或“后进先出”的原则组织数据的。

2. 栈的顺序存储及其运算

用一维数组S（1：m）作为栈的顺序存储空间，其中**m为最大容量**。

在栈的顺序存储空间 $S(1:m)$ 中， $S(\text{bottom})$ 为栈底元素， $S(\text{top})$ 为栈顶元素。 $\text{top}=0$ 表示栈空； $\text{top}=m$ 表示栈满。

栈的基本运算有三种：入栈、退栈与读栈顶元素。

(1) 入栈运算：入栈运算是指在栈顶位置插入一个新元素。首先将栈顶指针加一（即 $\text{top}+1$ ），然后将新元素插入到栈顶指针指向的位置。当栈顶指针已经指向存储空间的最后一个位置时，说明栈空间已满，不可能再进行入栈操作。这种情况称为“栈上溢”错误。

(2) 退栈运算：退栈是指取出栈顶元素并赋给一个指定的变量。首先将栈顶元素（栈顶指针指向的元素）赋给一个指定的变量，然后将栈顶指针减一（即 $\text{top}-1$ ）。当栈顶指针为0时，说明栈空，不可进行退栈操作。这种情况称为“栈下溢”错误。

(3) 读栈顶元素：读栈顶元素是指将栈顶元素赋给一个指定的变量。这个运算不删除栈顶元素，只是将它赋给一个变量，因此栈顶指针不会改变。当栈顶指针为0时，说明栈空，读不到栈顶元素。

☺小技巧：栈是按照“先进后出”或“后进先出”的原则组织数据，但是出栈方式有多种选择，在考题中经常考查各种不同的出栈方式。

考点6 线性链表的基本概念

在链式存储方式中，要求每个结点由两部分组成：一部分用于存放数据元素值，称为“数据域”，另一部分用于存放指针，称为“指针域”。其中指针用于指向该结点的前一个或后一个结点（即前件或后件）。

链式存储方式既可用于表示线性结构，也可用于表示非线性结构。

(1) 线性链表

线性表的链式存储结构称为线性链表。

在某些应用中，对线性链表中的每个结点设置两个指针，一个称为左指针，用以指向其前件结点；另一个称为右指针，用以指向其后件结点。这样的表称为“双向链表”。

(2) 带链的栈

栈也是线性表，也可以采用链式存储结构。带链的栈可以用来收集计算机存储空间中所有空闲的存储结点，这种带链的栈称为“可利用栈”。

☹疑难解答：在链式结构中，存储空间位置关系与逻辑关系是什么？

在链式存储结构中，存储数据结构的存储空间可以不连续，各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致，而数据元素之间的逻辑关系是由指针域来确定的。

1.4 树与二叉树

考点7 树与二叉树及其基本性质

误区警示：

满二叉树也是完全二叉树，而完全二叉树一般不是满二叉树。应该注意二者的区别。

1、树的基本概念

树(tree)是一种简单的非线性结构。在树结构中，每一个结点只有一个前件，称为父结点，没有前件的结点只有一个，称为树的根结点。每一个结点可以有多个后件，它们称为该结点的子结点。没有后件的结点称为“叶子”结点。

在树结构中，一个结点所拥有的后件个数称为该结点的度。叶子结点的度为0。在树中，所有结点中的最大的度称为树的度。

2、二叉树及其基本性质

(1) 二叉树的定义

二叉树是一种很有用的非线性结构，具有以下两个特点：

①非空二叉树只有一个根结点；

②每一个结点最多有两棵子树，且分别称为该结点的左子树和右子树。

由以上特点可以看出，在二叉树中，每一个结点的度最大为2，即所有子树（左子树或右子树）也均为二叉树，而树结构中的每一个结点的度可以是任意的。另外，二叉树中的每个结点的子树被明显地分为左子树和右子树。在二叉树中，一个结点可以只有左子树而没有右子树，也可以只有右子树而没有左子树。当一个结点既没有左子树也没有右子树时，该结点即为叶子结点。

（2）二叉树的基本性质

二叉树具有以下几个性质：

性质1：在二叉树的第 k 层上，最多有 2^{k-1} ($k \geq 1$) 个结点；

性质2：深度为 m 的二叉树最多有 $2^m - 1$ 个结点；

性质3：在任意一棵二叉树中，度为0的结点（即叶子结点）总是比度为2的结点多一个。

性质4：具有 n 个结点的二叉树，其深度至少为 $\lceil \log_2 n \rceil + 1$ ，其中 $\lceil \log_2 n \rceil$ 表示取 $\log_2 n$ 的整数部分。

☺小技巧：在二叉树的遍历中，无论是前序遍历，中序遍历还是后序遍历，二叉树的叶子结点的先后顺序都是不变的。

3、满二叉树与完全二叉树

满二叉树是指这样的一种二叉树：除最后一层外，每一层上的所有结点都有两个子结点。在满二叉树中，每一层上的结点数都达到最大值，即在满二叉树的第 k 层上有 2^{k-1} 个结点，且深度为 m 的满二叉树有 $2^m - 1$ 个结点。

完全二叉树是指这样的二叉树：除最后一层外，每一层上的结点数均达到最大值；在最后一层上只缺少右边的若干结点。

对于完全二叉树来说，叶子结点只可能在层次最大的两层上出现：对于任何一个结点，若其右分支下的子孙结点的最大层次为 p ，则其左分支下的子孙结点的最大层次为 p ，或为 $p+1$ 。

完全二叉树具有以下两个性质：

性质5：具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。

性质6：设完全二叉树共有 n 个结点。如果从根结点开始，按层次（每一层从左到右）用自然数 $1, 2, \dots, n$ 给结点进行编号，则对于编号为 k ($k=1, 2, \dots, n$) 的结点有以下结论：

①若 $k=1$ ，则该结点为根结点，它没有父结点；若 $k>1$ ，则该结点的父结点编号为 $\text{INT}(k/2)$ 。

②若 $2k \leq n$ ，则编号为 k 的结点的左子结点编号为 $2k$ ；否则该结点无左子结点（显然也没有右子结点）。

③若 $2k+1 \leq n$ ，则编号为 k 的结点的右子结点编号为 $2k+1$ ；否则该结点无右子结点。

考点8 二叉树的遍历

在遍历二叉树的过程中，一般先遍历左子树，再遍历右子树。在先左后右的原则下，根据访问根结点的次序，二叉树的遍历分为三类：前序遍历、中序遍历和后序遍历。

（1）前序遍历：先访问根结点、然后遍历左子树，最后遍历右子树；并且，在遍历左、右子树时，仍然先访问根结点，然后遍历左子树，最后遍历右子树。

（2）中序遍历：先遍历左子树、然后访问根结点，最后遍历右子树；并且，在遍历左、右子树时，仍然先遍历左子树，然后访问根结点，最后遍历右子树。

（3）后序遍历：先遍历左子树、然后遍历右子树，最后访问根结点；并且，在遍历左、右子树时，仍然先遍历左子树，然后遍历右子树，最后访问根结点。

☺疑难解答：树与二叉树的不同之处是什么？

在二叉树中，每一个结点的度最大为2，即所有子树（左子树或右子树）也均为二叉树，而树结构中的每一个结点的度可以是任意的。

1.5 查找技术

考点9 顺序查找

查找是指在一个给定的数据结构中查找某个指定的元素。从线性表的第一个元素开始，依次将线性表中的元素与被查找的元素相比较，若相等则表示查找成功；若线性表中所有的元素都与被查找元素进行了比较

但都不相等，则表示查找失败。

在下列两种情况下也只能采用顺序查找：

- (1) 如果线性表为无序表，则不管是顺序存储结构还是链式存储结构，只能用顺序查找。
- (2) 即使是有序线性表，如果采用链式存储结构，也只能用顺序查找。

考点10 二分法查找

二分法只适用于顺序存储的，按非递减排列的有序表，其方法如下：

设有序线性表的长度为 n ，被查找的元素为 i ，

- (1) 将 i 与线性表的中间项进行比较；
- (2) 若 i 与中间项的值相等，则查找成功；
- (3) 若 i 小于中间项，则在线性表的前半部分以相同的方法查找；
- (4) 若 i 大于中间项，则在线性表的后半部分以相同的方法查找。

☹ 疑难解答：二分查找法适用于哪种情况？

二分查找法只适用于顺序存储的有序表。在此所说的有序表是指线性表中的元素按值非递减排列（即从小到大，但允许相邻元素值相等）。

这个过程一直进行到查找成功或子表长度为0为止。

对于长度为 n 的有序线性表，在最坏情况下，二分查找只需要比较 $\log_2 n$ 次。

1.6 排序技术

考点11 交换类排序法

冒泡排序法和快速排序法都属于交换类排序法。

(1) 冒泡排序法

首先，从表头开始往后扫描线性表，逐次比较相邻两个元素的大小，若前面的元素大于后面的元素，则将它们互换，不断地将两个相邻元素中的大者往后移动，最后最大者到了线性表的最后。

然后，从后到前扫描剩下的线性表，逐次比较相邻两个元素的大小，若后面的元素小于前面的元素，则将它们互换，不断地将两个相邻元素中的小者往前移动，最后最小者到了线性表的最前面。

对剩下的线性表重复上述过程，直到剩下的线性表变空为止，此时已经排好序。

在最坏的情况下，冒泡排序需要比较次数为 $n(n-1)/2$ 。

(2) 快速排序法

它的基本思想是：任取待排序序列中的某个元素作为基准（一般取第一个元素），通过一趟排序，将待排序元素分为左右两个子序列，左子序列元素的排序码均小于或等于基准元素的排序码，右子序列的排序码则大于基准元素的排序码，然后分别对两个子序列继续进行排序，直至整个序列有序。

☹ 疑难解答：冒泡排序和快速排序的平均执行时间分别是多少？

冒泡排序法的平均执行时间是 $O(n^2)$ ，而快速排序法的平均执行时间是 $O(n \log_2 n)$ 。

1.7 例题详解

一、选择题

【例1】算法的时间复杂度取决于_____。（考点2）

- | | |
|----------|--------------|
| A) 问题的规模 | B) 待处理的数据的初态 |
| C) 问题的难度 | D) A) 和B) |

解析：算法的时间复杂度不仅与问题的规模有关，在同一个问题规模下，而且与输入数据有关。即与输入数据所有的可能取值范围、输入各种数据或数据集的概率有关。

答案：D)

【例2】在数据结构中，从逻辑上可以把数据结构分成_____。（考点3）

- A) 内部结构和外部结构
C) 紧凑结构和非紧凑结构

- B) 线性结构和非线性结构
D) 动态结构和静态结构

解析：逻辑结构反映数据元素之间的逻辑关系，线性结构表示数据元素之间为一对一的关系，非线性结构表示数据元素之间为一对多或者多对一的关系，所以答案为B)。

答案：B)

【例3】以下_____不是栈的基本运算。(考点5)

- A) 判断栈是否为素空
C) 删除栈顶元素

- B) 将栈置为空栈
D) 删除栈底元素

解析：栈的基本运算有：入栈，出栈（删除栈顶元素），初始化、置空、判断栈是否为空或满、提取栈顶元素等，对栈的操作都是在栈顶进行的。

答案：D)

【例4】链表不具备的特点是_____。(考点6)

- A) 可随机访问任意一个结点
C) 不必事先估计存储空间

- B) 插入和删除不需要移动任何元素
D) 所需空间与其长度成正比

解析：顺序表可以随机访问任意一个结点，而链表必须从第一个数据结点出发，逐一查找每个结点。所以答案为A)。

答案：A)

【例5】已知某二叉树的后序遍历序列是DACBE，中序遍历序列是DEBAC，则它的前序遍历序列是_____。(考点8)

- A) ACBED
C) DECAB

- B) DEABC
D) EDBAC

解析：后序遍历的顺序是“左子树—右子树—根结点”；中序遍历顺序是“左子树—根结点—右子树”；前序遍历顺序是“根结点—左子树—右子树”。根据各种遍历算法，不难得出前序遍历序列是EDBAC。所以答案为D)。

答案：D)

【例6】设有一个已按各元素的值排好序的线性表（长度大于2），对给定的值k，分别用顺序查找法和二分查找法查找一个与k相等的元素，比较的次数分别是s和b，在查找不成功的情况下，s和b的关系是_____。(考点9)

A) $s=b$

B) $s>b$

C) $s<b$

D) $s\geq b$

解析：对于顺序查找，查找不成功时和给定关键字比较的次数为 $n+1$ 。二分查找查找不成功的关键字比较次数为 $\lceil \log_2 n \rceil + 1$ 。当 $n \geq 2$ 时，显然 $n+1 > \lceil \log_2 n \rceil + 1$ 。

答案：B)

【例7】在快速排序过程中，每次划分，将被划分的表（或子表）分成左、右两个子表，考虑这两个子表，下列结论一定正确的是_____。(考点11)

- A) 左、右两个子表都已各自排好序
B) 左边子表中的元素都不大于右边子表中的元素
C) 左边子表的长度小于右边子表的长度
D) 左、右两个子表中元素的平均值相等

解析：快速排序基本思想是：任取待排序表中的某个元素作为基准（一般取第一个元素），通过一趟排序，将待排元素分为左右两个子表，左子表元素的排序码均小于或等于基准元素的排序码，右子表的排序码则大于基准元素的排序码，然后分别对两个子表继续进行排序，直至整个表有序。

答案：B)

二、填空题

【例1】问题处理方案的正确而完整的描述称为_____。(考点1)

解析：计算机解题的过程实际上是在实施某种算法，这种算法称为计算机算法。

答案：算法

【例2】一个空的数据结构是按线性结构处理的，则属于_____。(考点4)

解析：一个空的数据结构是线性结构或是非线性结构，要根据具体情况而定。如果对数据结构的运算是按线性结构来处理的，则属于线性结构，否则属于非线性结构。

答案：线性结构

【例3】设树T的度为4，其中度为1、2、3和4的结点的个数分别为4、2、1、1，则T中叶子结点的个数为_____。（考点7）

解析：根据树的性质：树的结点数等于所有结点的度与对应的结点个数乘积之和加1。

因此树的结点数为 $1 \times 4 + 2 \times 2 + 3 \times 1 + 4 \times 1 + 1 = 16$ 。叶子结点数等于树结点总数减去度不为0的结点数之和，即 $16 - (4 + 2 + 1 + 1) = 8$ 。

答案：8

【例4】二分法查找的存储结构仅限于_____且是有序的。（考点10）

解析：二分查找，也称折半查找，它是一种高效率的查找方法。但二分查找有条件限制：要求表必须用顺序存储结构，且表中元素必须按关键字有序(升序或降序均可)。

答案：顺序存储结构

第2章 程序设计基础

经过对部分考生的调查以及对近年真题的总结分析，笔试部分经常考查的是结构化程序设计的原则、面向对象方法的基本概念，读者应对此部分进行重点学习。

详细重点学习知识点：

1. 结构化程序设计方法的四个原则
2. 对象、类、消息、继承的概念、类与实例的区别

2.1 结构化程序设计

考点1 结构化程序设计的原则

20世纪70年代提出了“结构化程序设计”的思想和方法。结构化程序设计方法引入了**工程化思想**和**结构化思想**，使大型软件的开发和编程得到了极大的改善。结构化程序设计方法的主要原则为：自顶向下、逐步求精、模块化和限制使用goto语句。

☹ 疑难解答：如何进行自顶向下设计方法？

程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标；不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。

2.2 面向对象的程序设计

考点2 面向对象方法的基本概念

误区警示：

当使用“对象”这个术语时，既可以指一个具体的对象，也可以泛指一般的对象，但是当使用“实例”这个术语时，必须是指一个具体的对象。

面向对象方法涵盖对象及对象属性与方法、类、继承、多态性几个基本要素。

(1) 对象

通常把对对象的操作也称为方法或服务。

属性即对象所包含的信息，它在设计对象时确定，一般只能通过执行对象的操作来改变。属性值应该指的是纯粹的数据值，而不能指对象。

操作描述了对象执行的功能，若通过信息的传递，还可以为其他对象使用。

对象具有如下特征：标识惟一性、分类性、多态性、封装性、模块独立性。

(2) 类和实例

类是具有共同属性、共同方法的对象的集合。它描述了属于该对象类型的所有对象的性质，而一个对象则是其对应类的一个实例。

类是关于对象性质的描述，它同对象一样，包括一组数据属性和在数据上的一组合法操作。

(3) 消息

消息是实例之间传递的信息，它请求对象执行某一处理或回答某一要求的信息，它统一了数据流和控制流。

一个消息由三部分组成：接收消息的对象的名称、消息标识符（消息名）和零个或多个参数。

(4) 继承

广义地说，继承是指能够直接获得已有的性质和特征，而不必重复定义它们。

继承分为单继承与多重继承。单继承是指，一个类只允许有一个父类，即类等级为树形结构。多重继承是指，一个类允许有多个父类。

(5) 多态性

对象根据所接收的消息而做出动作，同样的消息被不同的对象接收时可导致完全不同的行动，该现象称为多态性。

🤖 疑难解答：能举一下现实中的对象及其属性和操作吗？

一辆汽车是一个对象，它包含了汽车的属性（如颜色、型号等）及其操作（如启动、刹车等）。一个窗口是对象，它包含了窗口的属性（如大小、颜色等）及其操作（如打开、关闭等）。

2.3 例题详解

一、选择题

【例1】结构化程序设计方法提出于_____。（考点1）

- A) 20世纪50年代
- B) 20世纪60年代
- C) 20世纪70年代
- D) 20世纪80年代

解析：20世纪70年代提出了“结构化程序设计（structured programming）”的思想和方法。结构化程序设计方法引入了工程化思想和结构化思想，使大型软件的开发和编程得到了极大的改善。

答案：C)

【例2】结构化程序设计方法的主要原则有下列4项，不正确的是_____。（考点1）

- A) 自下向上
- B) 逐步求精
- C) 模块化
- D) 限制使用goto语句

解析：结构化程序设计方法的主要原则为：

(1) 自顶向下：即先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。

(2) 逐步求精：对复杂问题，应设计一些子目标作过渡，逐步细化。

(3) 模块化：把程序要解决的总目标分解为分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。

(4) 限制使用goto语句。

答案：A)

【例3】面向对象的开发方法中，类与对象的关系是_____。（考点2）

- A) 抽象与具体
- B) 具体与抽象
- C) 部分与整体
- D) 整体与部分

解析：现实世界中的很多事物都具有相似的性质，把具有相似的属性和操作的对象归为类，也就是说类

是具有共同属性、共同方法的对象的集合，是对对象的抽象。它描述了该对象类型的所有对象的性质，而一个对象则是对应类的一个具体实例。所以本题正确答案为A)项。

答案：A)

二、填空题

【例1】在面向对象方法中，使用已经存在的类定义作为基础建立新的类定义，这样的技术叫做_____。(考点2)

解析：继承是面向对象方法的一个主要特征。继承是使用已有的类定义作为基础建立新类的定义技术。已有的类可当作基类来引用，则新类相应地可当作派生类来引用。

答案：继承

【例2】对象的基本特点包括_____、分类性、多态性、封装性和模块独立性好等5个特点。(考点2)

解析：对象具有如下的基本特点：

- (1) 标识惟一性。对象是可区分的，并且由对象的内在本质来区分；
- (2) 分类性。可以将具有相同属性和操作的对象抽象成类；
- (3) 多态性。同一个操作可以是不同对象的行为；
- (4) 封装性。只能看到对象的外部特征，无需知道数据的具体结构以及实现操作的算法；
- (5) 模块独立性。面向对象是由数据及可以对这些数据施加的操作所组成的统一体。

答案：标识惟一性

【例3】对象根据所接收的消息而做出动作，同样的消息被不同的对象所接收时可能导致完全不同的行为，这种现象称为_____。(考点2)

解析：对象根据所接收的消息而做出动作，同样的消息被不同的对象接收时可导致完全不同的行为，该现象称为多态性。

答案：多态性

第3章 软件工程基础

经过对部分考生的调查以及对近年真题的总结分析，笔试部分经常考查的是软件生命周期、软件设计的基本原理，软件测试的目的、软件调试的基本概念，读者应对此部分进行重点学习。

详细重点学习知识点：

1. 软件的概念、软件生命周期的概念及各阶段所包含的活动
2. 概要设计与详细设计的概念、模块独立性及其度量的标准、详细设计常用的工具
3. 软件测试的目的、软件测试的4个步骤、
4. 软件调试的任务

3.1 软件工程基本概念

考点1 软件定义与软件特点

软件指的是计算机系统中与硬件相互依存的另一部分，包括程序、数据和相关文档的完整集合。程序是软件开发人员根据用户需求开发的、用程序设计语言描述的、适合计算机执行的指令序列。数据是使程序能正常操纵信息的数据结构。文档是与程序的开发、维护和使用有关的图文资料。可见，软件由两部分组成：

- (1) 机器可执行的程序和数据；
- (2) 机器不可执行的，与软件开发、运行、维护、使用等有关的文档。

软件的特点：

- (1) 软件是逻辑实体，而不是物理实体，具有抽象性；
- (2) 没有明显的制作过程，可进行大量的复制；
- (3) 使用期间不存在磨损、老化问题；
- (4) 软件的开发、运行对计算机系统具有依赖性；
- (5) 软件复杂性高，成本昂贵；
- (6) 软件开发涉及诸多社会因素。

根据应用目标的不同，软件可分应用软件、系统软件和支撑软件（或工具软件）。

☺ 小提示：应用软件是为解决特定领域的应用而开发的软件；系统软件是计算机管理自身资源，提高计算机使用效率并为计算机用户提供各种服务的软件；支撑软件是介于两者之间，协助用户开发软件的工具性软件。

考点2 软件工程过程与软件生命周期

软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期。一般包括可行性分析研究与需求分析、设计、实现、测试、交付使用以及维护等活动，如图3—1所示。

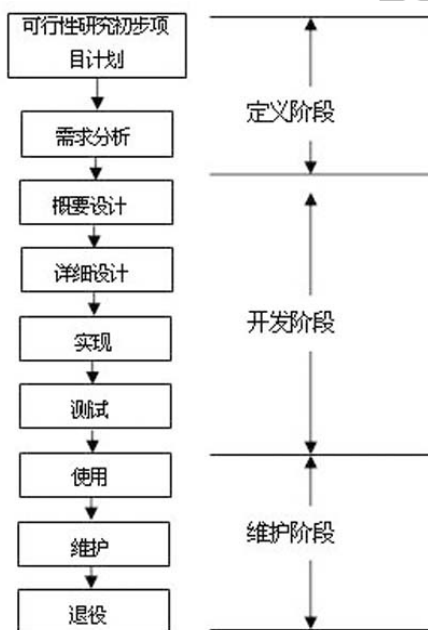


图3—1 软件生命周期

还可以将软件生命周期分为如上图所示的软件定义、软件开发和软件运行维护3个阶段。

生命周期的主要活动阶段是：可行性研究与计划制定、需求分析、软件设计、软件实施、软件测试及运行与维护。

3.2 结构化设计方法

考点3 软件设计的基本概念

误区警示：

在程序结构中，各模块的内聚性越强，则耦合性越弱。软件设计应尽量做到高内聚，低耦合，即减弱模块之间的耦合性和提高模块内的内聚性，有利于提高模块的独立性。

1. 软件设计的基础

从技术观点上看，软件设计包括软件结构设计、数据设计、接口设计、过程设计。

- (1) 结构设计定义软件系统各主要部件之间的关系；
- (2) 数据设计将分析时创建的模型转化为数据结构的定义；
- (3) 接口设计是描述软件内部、软件和协作系统之间以及软件与人之间如何通信；

(4) 过程设计则是把系统结构部件转换为软件的过程性描述。

从工程管理角度来看, 软件设计分两步完成: 概要设计和详细设计。

(1) 概要设计将软件需求转化为软件体系结构、确定系统级接口、全局数据结构或数据库模式;

(2) 详细设计确立每个模块的实现算法和局部数据结构, 用适当方法表示算法和数据结构的细节。

2. 软件设计的基本原理

(1) 抽象: 软件设计中考虑模块化解决方案时, 可以定出多个抽象级别。抽象的层次从概要设计到详细设计逐步降低。

(2) 模块化: 模块是指把一个待开发的软件分解成若干小的简单的部分。模块化是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。

(3) 信息隐蔽: 信息隐蔽是指在一个模块内包含的信息(过程或数据), 对于不需要这些信息的其他模块来说是不能访问的。

(4) 模块独立性: 模块独立性是指每个模块只完成系统要求的独立的子功能, 并且与其他模块的联系最少且接口简单。模块的独立程度是评价设计好坏的重要度量标准。衡量软件的模块独立性使用耦合性和内聚性两个定性的度量标准。内聚性是信息隐蔽和局部化概念的自然扩展。一个模块的内聚性越强则该模块的模块独立性越强。一个模块与其他模块的耦合性越强则该模块的模块独立性越弱。

内聚性是度量一个模块功能强度的一个相对指标。内聚是从功能角度来衡量模块的联系, 它描述的是模块内的功能联系。内聚有如下种类, 它们之间的内聚度由弱到强排列: 偶然内聚、逻辑内聚、时间内聚、过程内聚、通信内聚、顺序内聚、功能内聚。

耦合性是模块之间互相连接的紧密程度的度量。耦合性取决于各个模块之间接口的复杂度、调用方式以及哪些信息通过接口。耦合可以分为下列几种, 它们之间的耦合度由高到低排列: 内容耦合、公共耦合、外部耦合、控制耦合、标记耦合、数据耦合、非直接耦合。

在程序结构中, 各模块的内聚性越强, 则耦合性越弱。一般较优秀的软件设计, 应尽量做到高内聚, 低耦合, 即减弱模块之间的耦合性和提高模块内的内聚性, 有利于提高模块的独立性。

☺ 小提示: 上面仅是对耦合机制进行的一个分类。可见一个模块与其他模块的耦合性越强则该模块独立性越弱。原则上讲, 模块化设计总是希望模块之间的耦合表现为非直接耦合方式。但是, 由于问题所固有的复杂性和结构化设计的原则, 非直接耦合是不存在的。

考点4 详细设计

详细设计的任务是为软件结构图中的每个模块确定实现算法和局部数据结构, 用某种选定的表达表示工具算法和数据结构的细节。

详细过程设计的常用工具有:

(1) 图形工具: 程序流程图, N-S, PAD, HIPO。

(2) 表格工具: 判定表。

(3) 语言工具: PDL(伪码)。

程序流程图的5种控制结构: 顺序型、选择型、先判断重复型、后判断重复型和多分支选择型。

方框图中仅含5种基本的控制结构, 即顺序型、选择型、多分支选择型、WHILE重复型和UNTIL重复型。

PAD图表示5种基本控制结构, 即顺序型、选择型、多分支选择型、WHILE重复型和UNTIL重复型。

过程设计语言(PDL)也称为结构化的语言和伪码, 它是一种混合语言, 采用英语的词汇和结构化程序设计语言, 类似编程语言。

PDL可以由编程语言转换得到, 也可以是专门为过程描述而设计的。

☹ 疑难解答: 程序流程图, N-S图, PAD图的控制结构的异同点是什么?

相同点是三种图都有顺序结构, 选择结构和多分支选择, 并且N-S图和PAD图还有相同的WHILE重复型、UNTIL重复型; 不同点是程序流程图没有WHILE重复型、UNTIL重复型而有后判断重复型和先判断重复型。

3.3 软件测试

考点5 软件测试的目的

软件测试是在软件投入运行前对软件需求、设计、编码的最后审核。其工作量、成本占总工作量、总成本的40%以上，而且具有较高的组织管理和技术难度。

- (1) 软件测试是为了发现错误而执行程序的过程；
- (2) 一个好的测试用例是能够发现至今尚未发现的错误的用例；
- (3) 一个成功的测试是发现了至今尚未发现的错误的测试。

考点6 软件测试的实施

软件测试过程分4个步骤，即单元测试、集成测试、验收测试和系统测试。

单元测试是对软件设计的最小单位—模块（程序单元）进行正确性检验测试。单元测试的技术可以采用静态分析和动态测试。

集成测试是测试和组装软件的过程，主要目的是发现与接口有关的错误，主要依据是概要设计说明书。集成测试所设计的内容包括：软件单元的接口测试、全局数据结构测试、边界条件和非法输入的测试等。集成测试时将模块组装成程序，通常采用两种方式：非增量方式组装和增量方式组装。

确认测试的任务是验证软件的功能和性能，以及其他特性是否满足了需求规格说明中确定的各种需求，包括软件配置是否完全、正确。确认测试的实施首先运用黑盒测试方法，对软件进行有效性测试，即验证被测试软件是否满足需求规格说明确认的标准。

系统测试是通过测试确认软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、支撑软件、数据和人员等其他系统元素组合在一起，在实际运行（使用）环境下对计算机系统进行一系列的集成测试和确认测试。

系统测试的具体实施一般包括：功能测试、性能测试、操作测试、配置测试、外部接口测试、安全性测试等。

3.4 软件的调试

考点7 软件调试的基本概念

误区警示：

程序经调试改错后还应进行再测试，因为经调试后有可能产生新的错误，而且测试是贯穿生命周期的整个过程。

在对程序进行了成功的测试之后将进入程序调试（通常称Debug，即排错）。程序的调试任务是诊断和改正程序中的错误。调试主要在开发阶段进行。

程序调试活动由两部分组成，一是根据错误的迹象确定程序中错误的确切性质、原因和位置；二是对程序进行修改，排除这个错误。程序调试的基本步骤：

(1) 错误定位。从错误的外部表现形式入手，研究有关部分的程序，确定程序中出错位置，找出错误的内在原因；

(2) 修改设计和代码，以排除错误；

(3) 进行回归测试，防止引进新的错误。

调试原则可以从以下两个方面考虑：

(1) 确定错误的性质和位置时的注意事项

分析思考与错误征兆有关的信息；避开死胡同；只把调试工具当作辅助手段来使用；避免用试探法，最多只能把它当作最后手段。

(2) 修改错误原则

在出现错误的地方，很可能有别的错误；修改错误的一个常见失误是只修改了这个错误的征兆或这个错误的表现，而没有修改错误本身；注意修正一个错误的同时有可能会引入新的错误；修改错误的过程将迫使人们暂时回到程序设计阶段；修改源代码程序，不要改变目标代码。



疑难解答：软件测试与软件调试有何不同？

软件测试是尽可能多地发现软件中的错误，而软件调试的任务是诊断和改正程序中的错误。软件测试贯穿整个软件生命周

期，调试主要在开发阶段。

3.5 例题详解

一、选择题

【例1】对软件的特点，下面描述正确的是_____。（考点1）

- A) 软件是一种物理实体
- B) 软件在运行使用期间不存在老化问题
- C) 软件开发、运行对计算机没有依赖性，不受计算机系统的限制
- D) 软件的生产有一个明显的制作过程

解析：软件在运行期间不会因为介质的磨损而老化，只可能因为适应硬件环境以及需求变化进行修改而引入错误，导致失效率升高从而软件退化，所以本题正确答案为B)。

答案：B)

【例2】以下哪项是软件生命周期的主要活动阶段？_____（考点2）

- A) 需求分析
- B) 软件开发
- C) 软件确认
- D) 软件演进

解析：B)、C)、D) 项都是软件工程过程的基本活动，还有一个是软件规格说明。

答案：A)

【例3】从技术观点看，软件设计包括_____。（考点3）

- A) 结构设计、数据设计、接口设计、程序设计
- B) 结构设计、数据设计、接口设计、过程设计
- C) 结构设计、数据设计、文档设计、过程设计
- D) 结构设计、数据设计、文档设计、程序设计

解析：技术角度，要进行结构、接口、数据、过程的设计。结构设计是定义系统各部件关系，数据设计是根据分析模型转化数据结构，接口设计是描述如何通信，过程设计是把系统结构部件转化为软件的过程性描述。

答案：B)

【例4】以下哪个是软件测试的目的？_____（考点5）

- A) 证明程序没有错误
- B) 演示程序的正确性
- C) 发现程序中的错误
- D) 改正程序中的错误

解析：关于测试目的的基本知识，IEEE的定义是：使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的要求，或是弄清预期结果与实际结果之间的差别，所以正确答案是C)。

答案：C)

【例5】以下哪个测试要对接口测试？_____。（考点6）

- A) 单元测试
- B) 集成测试
- C) 验收测试
- D) 系统测试

解析：检查对测试实施各阶段的了解，集成测试时要进行接口测试、全局数据结构测试、边界条件测试和非法输入的测试等，所以正确答案是B)。

答案：B)

【例6】程序调试的主要任务是_____。（考点7）

- A) 检查错误
- B) 改正错误
- C) 发现错误
- D) 以上都不是

解析：程序的调试任务是诊断和改正程序中的错误。调试主要在开发阶段进行。

答案：B)

【例7】以下哪些不是程序调试的基本步骤？_____（考点7）

- A) 分析错误原因
- B) 错误定位
- C) 修改设计代码以排除错误
- D) 回归测试，防止引入新错误

解析：程序调试的基本步骤：

(1) 错误定位。从错误的外部表现形式入手,研究有关部分的程序,确定程序中出错位置,找出错误的内在原因。

(2) 修改设计和代码,以排除错误。

(3) 进行回归测试,防止引进新的错误。

答案: A)

【例8】在修改错误时应遵循的原则有_____。(考点7)

A) 注意修改错误本身而不仅仅是错误的征兆和表现

B) 修改错误的是源代码而不是目标代码

C) 遵循在程序设计过程中的各种方法和原则

D) 以上3个都是

解析: 修改错误原则:

(1) 在出现错误的地方,很可能有别的错误;

(2) 修改错误的一个常见失误是只修改了这个错误的征兆或这个错误的表现,而没有修改错误本身;

(3) 注意修正一个错误的同时有可能会引入新的错误;

(4) 修改错误的过程将迫使人们暂时回到程序设计阶段;

(5) 修改源代码程序,不要改变目标代码。

答案: D)

二、填空题

【例1】软件设计是软件工程的重要阶段,是一个把软件需求转换为_____的过程。(考点3)

解析: 软件设计是软件工程的重要阶段,是一个把软件需求转换为软件表示的过程。其基本目标是用比较抽象概括的方式确定目标系统如何完成预定的任务,即软件设计是确定系统的物理模型。

答案: 软件表示

【例2】_____是指把一个待开发的软件分解成若干小的简单的部分。(考点3)

解析: 模块化是指把一个待开发的软件分解成若干小的简单的部分。如高级语言中的过程、函数、子程序等。每个模块可以完成一个特定的子功能,各个模块可以按一定的方法组装起来成为一个整体,从而实现整个系统的功能。

答案: 模块化

【例3】数据流图采用4种符号表示_____,数据源点和终点、数据流向和数据加工。(考点4)

解析: 数据流图可以表达软件系统的数据存储、数据源点和终点、数据流向和数据加工。其中,用箭头表示数据流向,用圆或者椭圆表示数据加工,用双杠表示数据存储,用方框来表示数据源点和终点。

答案: 数据存储

第4章 数据库设计基础

经过对部分考生的调查以及对近年真题的总结分析,笔试部分经常考查的是数据库管理系统,数据库基本特点,数据库系统的三级模式及二级映射, E-R模型, 关系模型和关系代数, 读者应对此部分进行重点学习。

详细重点学习知识点:

1. 数据的概念、数据库管理系统提供的数据库语言、数据库管理员的主要工作、数据库系统阶段的特点、数据的物理独立性及逻辑独立性、数据统一管理与控制、三级模式及两级映射的概念
2. 数据模型3个描述内容、E-R模型的概念及其E-R图表示法、关系操纵、关系模型三类数据约束
3. 关系模型的基本操作、关系代数中的扩充运算

4. 数据库设计生命周期法的4个阶段

4.1 数据库系统的基本概念

考点1 数据、数据库、数据库管理系统

数据是数据库中存储的基本对象，描述事物的符号记录。

数据库是长期储存在计算机内、有组织的、可共享的大量数据的集合，它具有统一的结构形式并存放于统一的存储介质内，是多种应用数据的集成，并可被各个应用程序所共享。

数据库管理系统（DBMS，Database Management System）是数据库的机构，它是一种系统软件，负责数据库中的数据组织、数据操作、数据维护、控制及保护和数据服务等。数据库管理系统是数据系统的核心，主要有如下功能：数据模式定义、数据存取的物理构建、数据操纵、数据的完整性、安全性定义和检查、数据库的并发控制与故障恢复、数据的服务。

为完成数据库管理系统的功能，数据库管理系统提供相应的数据语言：数据定义语言、数据操纵语言、数据控制语言。

数据库管理员的主要工作如下：数据库设计、数据库维护、改善系统性能，提高系统效率。

考点2 数据库系统的发展

数据管理技术的发展经历了3个阶段，见表4-1：

表4-1 各阶段特点的详细说明

背景		人工管理阶段	文件系统阶段	数据库系统阶段
背景	应用背景	科学计算	科学计算、管理	大规模管理
	硬件背景	无直接存取存储设备	磁盘、磁鼓	大容量磁盘
	软件背景	没有操作系统	有文件系统	有数据库管理系统
处理方式		批处理	联机实时处理、批处理	联机实时处理、分布处理、批处理
特点	数据的管理者	用户（程序员）	文件系统	数据库管理系统
	数据面向的对象	某一应用程序	某一应用	现实世界
	数据的共享程度	无共享，冗余度大	共享性差，冗余度大	共享性高，冗余度小
	数据的独立性	不独立，完全依赖于程序	独立性差	具有高度的物理独立性和一定的逻辑独立性
	数据结构化	无结构	记录内有结构、整体无结构	整体结构化，用数据模型描述
	数据控制能力	应用程序自己控制	应用程序自己控制	由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力

考点3 数据库系统的基本特点

数据独立性是数据与程序间的互不依赖性，即数据库中的数据独立于应用程序而不依赖于应用程序。数据的独立性一般分为物理独立性与逻辑独立性两种。

（1）物理独立性：指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。当数据的物理结构（包括存储结构、存取方式等）改变时，如存储设备的更换、物理存储的更换、存取方式改变等，应用程序都不用改变。

（2）逻辑独立性：指用户的应用程序与数据库的逻辑结构是相互独立的。数据的逻辑结构改变了，如修改数据模式、增加新的数据类型、改变数据间联系等，用户程序都可以不变。

数据统一管理与控制主要包括以下3个方面：数据的完整性检查、数据的安全性保护和并发控制。

考点4 数据库系统的内部结构体系

误区警示:

一个数据库只有一个概念模式。一个概念模式可以有若干个外模式。三级模式都有几种名称,读者应该熟记每个模式的另一些名称。

1. 数据库系统的3级模式

(1) **概念模式**,也称逻辑模式,是对数据库系统中全局数据逻辑结构的描述,是全体用户(应用)公共数据视图。一个数据库只有一个概念模式。

(2) **外模式**,外模式也称子模式,它是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述,它是由概念模式推导而出来的,是数据库用户的数据视图,是与某一应用有关的数据的逻辑表示。一个概念模式可以有若干个外模式。

(3) **内模式**,内模式又称物理模式,它给出了数据库物理存储结构与物理存取方法。

内模式处于最底层,它反映了数据在计算机物理结构中的实际存储形式,概念模式处于中间层,它反映了设计者的数据全局逻辑要求,而外模式处于最外层,它反映了用户对数据的要求。

☺ 小提示:内模式处于最底层,它反映了数据在计算机物理结构中的实际存储形式,概念模式处于中间层,它反映了设计者的数据全局逻辑要求,而外模式处于最外层,它反映了用户对数据的要求。

2. 数据库系统的两级映射

两级映射保证了数据库系统中数据的独立性。

(1) 概念模式到内模式的映射。该映射给出了概念模式中数据的全局逻辑结构到数据的物理存储结构间的对应关系;

(2) 外模式到概念模式的映射。概念模式是一个全局模式而外模式是用户的局部模式。一个概念模式中可以定义多个外模式,而每个外模式是概念模式的一个基本视图。

☹ 疑难解答:数据库应用系统的结构是什么样的?

数据库应用系统的7个部分以一定的逻辑层次结构方式组成一个有机的整体,它们的结构关系是:应用系统工、应用开发工具软件、数据库管理系统、操作系统、硬件。

4.2 数据模型

考点5 数据模型的基本概念

数据模型用来抽象、表示和处理现实世界中的数据和信息。分为两个阶段:把现实世界中的客观对象抽象为概念模型;把概念模型转换为某一DBMS支持的数据模型。

数据模型所描述的内容有3个部分,它们是**数据结构**、**数据操作**与**数据约束**。

考点6 E-R模型

1. E-R模型的基本概念

(1) **实体**:现实世界中的事物可以抽象成为实体,**实体**是概念世界中的基本单位,它们是客观存在的且又能相互区别的事物。

(2) **属性**:现实世界中事物均有一些特性,这些特性可以用属性来表示。

(3) **码**:**唯一**标识实体的**属性集**称为码。

(4) **域**:属性的取值范围称为该属性的域。

(5) **联系**:在现实世界中事物间的关联称为联系。

两个实体集间的联系实际上是实体集间的函数关系,这种函数关系可以有下面几种:一对一的联系、一对多或多对一联系、多对多。

2. E-R模型的图示法

E-R模型用E-R图来表示。

(1) **实体表示法**:在E-R图中用矩形表示实体集,在矩形内写上该实体集的名字。

(2) **属性表示法**:在E-R图中用椭圆形表示属性,在椭圆形内写上该属性的名称。

(3) **联系表示法**:在E-R图中用菱形表示联系,菱形内写上联系名。

考点7 层次模型

满足下面两个条件的基本层次联系的集合为层次模型。

- (1) 有且只有一个结点没有双亲结点，这个结点称为根结点；
- (2) 除根结点以外的其他结点有且仅有一个双亲结点。

考点8 关系模型

误区警示：

当对关系模型进行查询运算，涉及到多种运算时，应当注意它们之间的先后顺序，因为有可能进行投影运算时，把符合条件的记录过滤，产生错误的结果。

关系模型采用二维表来表示，二维表一般满足下面7个性质：

- (1) 二维表中元组个数是有限的——元组个数有限性；
- (2) 二维表中元组均不相同——元组的唯一性；
- (3) 二维表中元组的次序可以任意交换——元组的次序无关性；
- (4) 二维表中元组的分量是不可分割的基本数据项——元组分量的原子性；
- (5) 二维表中属性名各不相同——属性名唯一性；
- (6) 二维表中属性与次序无关，可任意交换——属性的次序无关性；
- (7) 二维表属性的分量具有与该属性相同的值域——分量值域的统一性。

在二维表中惟一标识元组的最小属性值称为该表的键或码。二维表中可能有若干个键，它们称为表的候选码或候选键。从二维表的所有候选键选取一个作为用户使用的键称为主键或主码。表A中的某属性集是某表B的键，则称该属性值为A的外键或外码。

关系操纵：数据查询、数据删除、数据插入、数据修改。

关系模型允许定义三类数据约束，它们是实体完整性约束、参照完整性约束以及用户定义的完整性约束。

☹ 小提示：关系模式采用二维表来表示，一个关系对应一张二维表。可以这么说，一个关系就是一个二维表，但是一个二维表不一定是一个关系。

☹ 疑难解答：E-R图是如何向关系模式转换的？

从E-R图到关系模式的转换是比较直接的，实体与联系都可以表示成关系，E-R图中属性也可以转换成关系的属性。实体集也可以转换成关系。

4.3 关系代数

考点9 关系代数

误区警示：

当对关系模型进行查询运算，涉及到多种运算时，应当注意它们之间的先后顺序，因为有可能进行投影运算时，把符合条件的记录过滤，产生错误的结果。

1. 关系模型的基本操作

关系模型的基本操作：插入、删除、修改和查询。

其中查询包含如下运算：

①投影运算。从R中选择出若干属性列组成新的关系。

②选择运算。选择运算是一个一元运算，关系R通过选择运算（并由该运算给出所选择的逻辑条件）后仍为一个关系。设关系的逻辑条件为F，则R满足F的选择运算可写成： $\sigma F(R)$

③笛卡尔积运算。设有n元关系R及m元关系S，它们分别有p、q个元组，则关系R与S经笛卡尔积记为R×S，该关系是一个n+m元关系，元组个数是p×q，由R与S的有序组合而成。

☹ 小提示：当关系模式进行笛卡尔积运算时，读者应该注意运算后的结果是n+m元关系，元组个数是p×q，这是经常混淆的。

2. 关系代数中的扩充运算

(1) 交运算：关系R与S经交运算后所得到的关系是由那些既在R内又在S内的有序组所组成，记为 $R \cap S$ 。

(2) 除运算

如果将笛卡尔积运算看作乘运算的话，除运算就是它的逆运算。当关系 $T=R \times S$ 时，则可将除运算写成：

$$T \div R = S \text{ 或 } T/R = S$$

S称为T除以R的商。除法运算不是基本运算，它可以由基本运算推导而出。

(3) 连接与自然连接运算

连接运算又可称为 θ 运算，这是一种二元运算，通过它可以将两个关系合并成一个大关系。设有关系R、S以及比较式 $i \theta j$ ，其中i为R中的域，j为S中的域， θ 含义同前。则可以将R、S在域i、j上的 θ 连接记为：

$$R \mid_{\begin{smallmatrix} \times \\ i \theta j \end{smallmatrix}} S$$

在 θ 连接中如果 θ 为“=”，就称此连接为等值连接，否则称为不等值连接；如 θ 为“<”时称为小于连接；如 θ 为“>”时称为大于连接。

自然连接（natural join）是一种特殊的等值连接，它满足下面的条件：

- ①两关系间有公共域；
- ②通过公共域的等值进行连接。

设有关系R、S，R有域 A_1, A_2, \dots, A_n ，S有域 B_1, B_2, \dots, B_m ，并且 $A_{i1}, A_{i2}, \dots, A_{ij}$ 与 B_1, B_2, \dots, B_j 分别为相同域，此时它们自然连接可记为：

$$R \mid_{\times} S$$

自然连接的含义可用下式表示：

$$R \mid_{\times} S = \pi_{A_1, A_2, \dots, A_n, B_{j+1}, \dots, B_m} \left(\sigma_{A_{i1}=B_1 \wedge A_{i2}=B_2 \wedge \dots \wedge A_{ij}=B_j} (R \times S) \right)$$

☹ 疑难解答：连接与自然连接的不同之处在什么？

一般的连接操作是从行的角度进行运算，但自然连接还需要取消重复列，所以是同时从行和列的角度进行运算。

4.4 数据库设计与管理

考点10 数据库设计概述

数据库设计中有两种方法，**面向数据**的方法和**面向过程**的方法：

面向数据的方法是以信息需求为主，兼顾处理需求；面向过程的方法是以处理需求为主，兼顾信息需求。由于数据在系统中稳定性高，数据已成为系统的核心，因此面向数据的设计方法已成为主流。

数据库设计目前一般采用生命周期法，即将整个数据库应用系统的开发分解成目标独立的若干阶段。它们是：需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段、编码阶段、测试阶段、运行阶段和进一步修改阶段。在数据库设计中采用前4个阶段。

☹ 疑难解答：数据库设计的前4个阶段的成果分别是什么？

数据库设计中一般采用前4个阶段，它们的成果分别是需求说明书、概念数据模型、逻辑数据模型和数据库内模式。

4.4 例题详解

一、选择题

【例1】对于数据库系统，负责定义数据库内容，决定存储结构和存取策略及安全授权等工作的是_____。（考点1）

A) 应用程序员

B) 用户

C) 数据库管理员

D) 数据库管理系统的软件设计员

解析：数据库管理员（简称DBA），具有如下的职能：设计、定义数据库系统；帮助用户使用数据库系统；监督与控制数据库系统的使用和运行；改进和重组数据库系统；转储和恢复数据库；重构数据库。所以，定义数据库内容，决定存储结构和存取策略及安全授权等是数据库管理员（DBA）的职责。

答案：C)

【例2】在数据库管理技术的发展过程中，经历了人工管理阶段、文件系统阶段和数据库系统阶段。在这几个阶段中，数据独立性最高的是_____。(考点2)

- A) 数据库系统
- B) 文件系统
- C) 人工管理
- D) 数据项管理

解析：在人工管理阶段，数据无法共享，冗余度大，不独立，完全依赖于程序。在文件系统阶段，数据共享性差，冗余度大，独立性也较差。所以B)选项和C)选项均是错误的。

答案：A)

【例3】在数据库系统中，当总体逻辑结构改变时，通过改变_____，使局部逻辑结构不变，从而使建立在局部逻辑结构之上的应用程序也保持不变，称之为数据和程序的逻辑独立性。(考点3)

- A) 应用程序
- B) 逻辑结构和物理结构之间的映射
- C) 存储结构
- D) 局部逻辑结构到总体逻辑结构的映射

解析：模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。当模式改变时，由数据库管理员对外模式/模式映射做相应改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，从而应用程序也不必改变。保证了数据与程序的逻辑独立性，即数据的逻辑独立性。

答案：D)

【例4】数据库系统依靠_____支持数据的独立性。(考点4)

- A) 具有封装机制
- B) 定义完整性约束条件
- C) 模式分级，各级模式之间的映射
- D) DDL语言和DML语言互相独立

解析：。数据库的三级模式结构指数据库系统由外模式、模式和内模式3级构成。数据库管理系统在这3级模式之间提供了两层映射：外模式/模式映射，模式/内模式映射。这两层映射保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

答案：C)

【例5】将E-R图转换到关系模式时，实体与联系都可以表示成_____。(考点6)

- A) 属性
- B) 关系
- C) 键
- D) 域

解析：E-R图由实体、实体的属性和实体之间的联系3个要素组成，关系模型的逻辑结构是一组关系模式的集合，将E-R图转换为关系模型：将实体、实体的属性和实体之间的联系转化为关系模式。

答案：B)

【例6】用树形结构来表示实体之间联系的模型称为_____。(考点7)

- A) 关系模型
- B) 层次模型
- C) 网状模型
- D) 数据模型

解析：满足下面两个条件的基本层次联系的集合为层次模型：

- (1) 有且只有一个结点没有双亲结点，这个结点称为根结点；
- (2) 根以外的其它结点有且仅有一个双亲结点。

层次模型的特点：

- (1) 结点的双亲是惟一的；
- (2) 只能直接处理一对多的实体联系；
- (3) 每个记录类型定义一个排序字段，也称为码字段；
- (4) 任何记录值只有按其路径查看时，才能显出它的全部意义；
- (5) 没有一个子女记录值能够脱离双亲记录值而独立存在。

答案：B)

【例7】对数据库中的数据可以进行查询、插入、删除、修改（更新），这是因为数据库管理系统提供了_____。(考点8)

- A) 数据定义功能
- B) 数据操纵功能
- C) 数据维护功能
- D) 数据控制功能

解析：数据库管理系统包括如下功能：

(1) 数据定义功能：DBMS提供数据定义语言（DDL），用户可以通过它方便地对数据库中的数据对象进行定义；

(2) 数据操纵功能：DBMS还提供数据操作语言（DML），用户可以通过它操纵数据，实现对数据库的基本操作，如查询、插入、删除和修改；

(3) 数据库的运行管理：数据库在建立、运用和维护时由数据库管理系统统一管理，统一控制以保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复；

(4) 数据库的建立和维护功能：它包括数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组、功能和性能监视等等。

答案：B)

【例8】设关系R和关系S的属性元数分别是3和4，关系T是R与S的笛卡儿积，即 $T=R \times S$ ，则关系T的属性元数是_____（考点9）

- A) 7 B) 9
C) 12 D) 16

解析：笛卡儿积的定义是设关系R和S的元数分别是r和s，R和S的笛卡儿积是一个 $(r+s)$ 元属性的集合，每一个元组的前r个分量来自R的一个元组，后s个分量来自S的一个元组。所以关系T的属性元数是 $3+4=7$ 。

答案：A)

【例9】下述_____不属于数据库设计的内容。（考点10）

- A) 数据库管理系统 B) 数据库概念结构
C) 数据库逻辑结构 D) 数据库物理结构

解析：数据库设计是确定系统所需要的数据库结构。数据库设计包括概念设计、逻辑设计和建立数据库（又称物理设计）。

答案：A)

二、填空题

【例1】一个数据库的数据模型至少应该包括以下3个组成部分，_____、数据操作和数据的完整性约束条件。（考点5）

解析：数据模型是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特性、动态特性和完整性约束条件。因此，数据模型通常由数据结构、数据操作和完整性约束3部分组成。其中，数据结构是对系统静态特性的描述，数据操作是对系统动态特性的描述，数据的完整性约束用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确性、有效性和相容性。

答案：数据结构

【例2】在关系数据模型中，二维表的列称为属性，二维表的行称为_____。（考点8）

解析：一个关系是一张二维表。表中的行称为元组，一行对应一个元组，一个元组对应存储在文件中的一个记录值。

答案：元组