



普通高等教育“十三五”规划教材

# 计算机公共基础与 MS Office高级应用

何鸱 孙明玉 吴登峰 主编



科学出版社

普通高等教育“十三五”规划教材

# 计算机公共基础与 MS Office 高级应用

何 鹏 孙明玉 吴登峰 主编

科学出版社

北 京

## 内 容 简 介

本书参照教育部提出的非计算机专业多层次大学计算机基础教学的要求和全国计算机等级考试大纲编写。

本书分为两个部分,共14章,第1部分为计算机公共基础知识,包括算法与数据结构、程序设计基础、软件工程基础、数据库设计基础;第2部分为MS Office高级应用,包括Word 2010高级应用、Excel 2010高级应用、PowerPoint 2010高级应用。

本书适合作为全国计算机等级考试二级MS Office高级应用的应试教材,也可作为高等院校非计算机专业计算机通识课程的教材,还可作为办公自动化课程的培训教材及自学MS Office高级应用的参考用书。

### 图书在版编目(CIP)数据

计算机公共基础与MS Office高级应用/何鸱,孙明玉,吴登峰主编. —北京:科学出版社,2018.12

普通高等教育“十三五”规划教材

ISBN 978-7-03-059984-1

I. ①计… II. ①何… ②孙… ③吴… III. ①电子计算机-高等学校-教材 ②办公自动化-应用软件-高等学校-教材 IV. ①TP3

中国版本图书馆CIP数据核字(2018)第278886号

责任编辑:戴薇 王惠 / 责任校对:赵丽杰

责任印制:吕春珉 / 封面设计:东方人华平面设计部

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

铭浩彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

\*

2018年12月第一版 开本:787×1092 1/16

2018年12月第一次印刷 印张:13 3/4

字数:324 000

定价:38.00元

(如有印装质量问题,我社负责调换〈骏杰〉)

销售部电话 010-62136230 编辑部电话 010-62135397-2052

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

# 前 言

全国计算机等级考试二级公共基础知识主要考查 4 部分内容，分别是基本数据结构与算法、程序设计基础、软件工程基础、数据库设计基础；全国计算机等级考试二级 MS Office 高级应用主要考查 4 部分的内容，分别是计算机基础知识、Word 的功能和使用、Excel 的功能和使用、PowerPoint 的功能和使用。

本书基于以上两个考试大纲编写而成，内容包括计算机公共基础知识和 MS Office 高级应用两部分。使用本书的前提是读者已掌握计算机基础知识和 MS Office 基本操作。本书侧重于对 Word 2010、Excel 2010、PowerPoint 2010 的高级功能进行详细、深入的解析和应用，旨在提高读者完成文案工作的效率和水平。

本书具体内容如下：

第 1 章，主要介绍算法与数据结构，内容包括算法与程序、数据结构、线性表、栈与队列、树与二叉树、查找与排序。

第 2 章，主要介绍程序设计基础，内容包括程序设计方法与风格、结构化程序设计、面向对象的程序设计。

第 3 章，主要介绍软件工程基础，内容包括软件工程的基本知识、软件结构化分析方法、软件设计方法、软件测试方法、程序的调试。

第 4 章，主要介绍数据库设计基础，内容包括数据库系统的基本概念、数据模型、关系数据库、数据库设计与实施。

第 5~7 章，主要介绍 Word 2010 高级应用，内容包括长文档的编辑、文档的修订与共享、通过邮件合并批量处理文档。

第 8~11 章，主要介绍 Excel 2010 高级应用，内容包括 Excel 公式和函数、使用 Excel 创建图表、分析与处理 Excel 数据、Excel 与其他程序的协同与共享。

第 12~14 章，主要介绍 PowerPoint 2010 高级应用，内容包括幻灯片中对象的编辑、幻灯片的交互效果设置、幻灯片的播放与共享。

书中实例所用素材下载地址：<http://www.abook.cn>。

本书由何鹄、孙明玉、吴登峰担任主编。感谢陈然、晏愈光、孙英娟、姜艳、刘妍、杨鑫、贾学婷、吴爽等在本书编写过程中给予的支持和帮助，使本书得以顺利成稿。

由于时间紧迫及编者水平有限，书中难免有不足之处，恳请广大读者批评指正。

编 者

2018 年 10 月

# 目 录

## 第 1 部分 计算机公共基础知识

第 1 章 算法与数据结构	3
1.1 算法与程序	3
1.1.1 算法的概念	3
1.1.2 程序的概念	7
1.2 数据结构	8
1.2.1 数据结构的基本概念	8
1.2.2 数据的基本结构	8
1.3 线性表	10
1.3.1 线性表的基本概念	10
1.3.2 线性表的存储结构	10
1.4 栈与队列	13
1.4.1 栈	13
1.4.2 队列	14
1.5 树与二叉树	16
1.5.1 树的相关概念	16
1.5.2 二叉树及其基本性质	17
1.5.3 二叉树的遍历	17
1.6 查找与排序	18
1.6.1 查找	18
1.6.2 排序	19
第 2 章 程序设计基础	21
2.1 程序设计方法与风格	21
2.2 结构化程序设计	22
2.2.1 结构化程序设计的原则	22
2.2.2 结构化程序设计的基本结构	22
2.2.3 结构化程序设计的应用	23
2.3 面向对象的程序设计	23
2.3.1 面向对象方法的优点	23
2.3.2 面向对象方法的基本概念	24

第 3 章 软件工程基础 .....	27
3.1 软件工程的基础知识 .....	27
3.1.1 软件工程的基本概念 .....	27
3.1.2 软件工程周期 .....	28
3.1.3 软件工具与软件开发环境 .....	34
3.2 软件结构化分析方法 .....	35
3.3 软件设计方法 .....	36
3.4 软件测试方法 .....	37
3.5 程序的调试 .....	39
第 4 章 数据库设计基础 .....	42
4.1 数据库系统的基本概念 .....	42
4.1.1 数据库和 DBMS .....	42
4.1.2 数据库技术的发展 .....	43
4.1.3 典型数据库系统 .....	44
4.1.4 数据库系统的基本特点 .....	45
4.2 数据模型 .....	48
4.3 关系数据库 .....	50
4.3.1 关系相关术语 .....	50
4.3.2 关系运算 .....	51
4.3.3 关系的完整性 .....	52
4.4 数据库设计与实施 .....	54

## 第 2 部分 MS Office 高级应用

第 5 章 长文档的编辑 .....	57
5.1 定义并使用样式 .....	57
5.1.1 新建样式 .....	57
5.1.2 修改样式 .....	59
5.1.3 导入/导出样式 .....	60
5.1.4 应用样式 .....	62
5.1.5 重命名样式 .....	62
5.1.6 删除样式 .....	62
5.1.7 更新样式 .....	63
5.1.8 使用样式集 .....	63

5.2	文档的分栏处理	65
5.3	文档的分页处理	67
5.4	文档的分节处理	67
5.5	设置页眉和页脚	68
5.5.1	建立页眉和页脚	68
5.5.2	插入页码	71
5.6	项目符号、编号和多级列表	72
5.6.1	添加和更改项目符号和编号	72
5.6.2	定义和使用多级列表	73
5.7	编辑文档目录	77
5.7.1	创建目录	77
5.7.2	更新目录	79
5.8	插入文档封面	80
5.9	插入脚注和尾注	80
<b>第 6 章</b>	<b>文档的修订与共享</b>	<b>83</b>
6.1	修订文档	83
6.2	管理文档	87
6.3	共享文档	92
<b>第 7 章</b>	<b>通过邮件合并批量处理文档</b>	<b>94</b>
7.1	邮件合并基础	94
7.2	制作信封	94
7.3	制作邀请函	98
<b>第 8 章</b>	<b>Excel 公式和函数</b>	<b>103</b>
8.1	使用公式的基本方法	103
8.2	定义与引用名称	105
8.2.1	定义名称	105
8.2.2	引用名称	108
8.3	函数的基本用法	109
8.3.1	插入和编辑函数	109
8.3.2	常用函数	111
8.4	常见问题及解决方法	121

第 9 章 使用 Excel 创建图表	123
9.1 创建图表	123
9.2 编辑图表	126
9.3 创建和编辑迷你图表	131
第 10 章 分析与处理 Excel 数据	138
10.1 数据排序	138
10.1.1 简单排序	138
10.1.2 高级排序	139
10.2 数据筛选	143
10.3 分类汇总与分级显示	147
10.4 数据透视表和透视图	152
10.5 合并计算	156
第 11 章 Excel 与其他程序的协同与共享	159
11.1 共享工作簿	159
11.2 修订工作簿	163
11.3 插入批注	166
11.4 获取外部数据	167
11.4.1 导入文本文件	167
11.4.2 插入超链接	171
11.5 与其他程序共享数据	172
11.6 宏的简单应用	173
第 12 章 幻灯片中对象的编辑	178
12.1 使用图形	178
12.2 使用图片	181
12.3 使用表格	182
12.4 使用图表	184
12.5 使用视频和音频	186
12.6 使用艺术字	188
12.7 使用自动版式插入对象	189

第 13 章 幻灯片的交互效果设置 .....	191
13.1 动画效果 .....	191
13.2 设置切换效果 .....	196
13.3 幻灯片的超链接 .....	197
第 14 章 幻灯片的播放与共享 .....	200
14.1 播放幻灯片 .....	200
14.2 播放设置 .....	201
14.3 共享幻灯片 .....	204
14.4 幻灯片的输出 .....	205
参考文献 .....	208



# 第 1 部分 计算机公共基础知识

本部分内容主要介绍全国计算机等级考试二级公共基础知识考试大纲所涉及的内容，包括 4 章：算法与数据结构、程序设计基础、软件工程基础、数据库设计基础。



# 第 1 章 算法与数据结构

算法是程序设计的核心，是编程的基础，也最能体现利用计算机求解实际问题的思维方法。数据结构是计算机语言所提供的各种数据类型和数据的组织形式。

## 1.1 算法与程序

用计算机语言解决问题的过程包括建立数学模型、设计算法及编程调试 3 个步骤。著名计算机科学家沃思曾提出：数据结构+算法=程序。

### 1.1.1 算法的概念

#### 1. 算法的定义与特性

算法是对特定问题求解步骤的一种描述，是指令的有限序列。算法可以理解为由基本运算及规定的运算顺序所构成的完整的解题步骤。一个算法应该具有以下 5 个特性：

- 1) 有穷性。一个算法必须在有穷步骤之后结束，即必须在有限时间内完成。
- 2) 确定性。算法的每一步骤必须有确切的定义，无二义性。
- 3) 可行性。算法中的每一步都可以通过已经实现的基本运算的有限次执行来实现。
- 4) 输入。一个算法有 0 个或多个输入。
- 5) 输出。一个算法有一个或多个输出。

算法的含义与程序十分相似，但又有区别。一方面，一个程序不一定满足有穷性，如操作系统，只要整个系统不被破坏，它将永远不会停止，即使没有作业处理，它仍然处于动态等待中，因此操作系统不是一个算法。另一方面，程序中的指令必须是机器可执行的，而算法中的指令则无此限制。算法代表了对问题的求解步骤，而程序则是算法设计在计算机上的实现。一个算法若用程序设计语言来描述，则它就是一个程序。

要设计一个好的算法通常要考虑以下要求：

- 1) 正确性。算法的执行结果应当满足预定的功能与性能要求。
- 2) 可读性。算法应当思路清晰、层次分明、易读易懂。
- 3) 健壮性。当输入非法数据时，不会引起严重后果。
- 4) 高效性。有效使用存储空间和有较高的时间效率。

#### 2. 算法的评价

同一问题可用不同的算法来解决，而一个算法的质量将影响算法乃至程序的效率。那么如何来评价算法的优劣呢？首先，算法必须是正确的。此外，主要考虑以下几个方面：

- 1) 执行算法所耗费的时间，通常用算法的时间复杂度来衡量。
- 2) 执行算法所耗费的存储空间，通常用算法的空间复杂度来衡量。
- 3) 算法应该易于理解、易于编码、易于调试。

从理论上来看，我们希望寻找一个执行时间短、占用存储空间小、其他性能也好的算法。但是，现实中很难做到十全十美，原因是上述要求常常相互制约。节约时间的算法往往以牺牲空间为代价，而节约空间的算法往往以牺牲时间为代价。因此，只能根据具体情况有所侧重。

当一个算法转换成程序并在计算机上运行时，其执行的时间取决于下列因素：

- 1) 计算机硬件的运算速度。
- 2) 编写程序所用的语言。
- 3) 编译程序所生成的目标代码的质量。
- 4) 问题的规模。

显然，在各种因素都不能确定的情况下，很难确定算法的执行时间，即使用算法执行的绝对时间来衡量算法的效率是难以实现的。因此，在评价算法的效率时，通常不考虑计算机的软硬件因素。这样，一个算法的效率就只依赖于问题的规模（通常用正整数  $n$  表示），或者说，它是问题规模的函数。

### 3. 算法的时间复杂度

一个算法的时间复杂度是指执行算法所需要的计算工作量。

一个算法由控制结构和原操作构成，其执行时间取决于两者的综合效率。为了便于比较同一问题的不同算法，通常的做法是，从算法中选取对于所研究的问题来说是基本运算的原操作，以原操作重复执行的次数作为算法的时间度量。一般来说，算法中原操作的重复执行次数是问题规模  $n$  的函数，用  $f(n)$  表示，算法的时间复杂度记为

$$T(n) = O(f(n))$$

因为算法执行时间的增长率和  $f(n)$  的增长率成正比，所以  $f(n)$  越小，算法的时间复杂度越低，算法的效率越高。

许多时候，要精确计算  $T(n)$  是很困难的，可引入“渐进时间复杂度”在数量上估计一个算法的执行时间，达到分析算法的目的。在计算的时候，先找出算法的基本操作，然后根据相应的各语句确定它的执行次数，再找出  $T(n)$  的同数量级（它的同数量级通常有  $1$ 、 $\log_2 n$ 、 $n$ 、 $n \log_2 n$ 、 $n^2$ 、 $n^3$ 、 $2^n$ 、 $n!$ 、 $n^n$  等），找出后，令  $f(n) =$  该数量级，若  $T(n)/f(n)$  求极限可得到一常数  $c$ ，则时间复杂度  $T(n) = O(f(n))$ 。

### 4. 算法的空间复杂度

一个算法的空间复杂度是指运行完一个算法所需的存储空间，利用算法的空间复杂度可对程序运行所需要的内存进行估计。一个程序执行时除了需要存储本身所使用的指令、常数、变量和输入数据外，还需要一些对数据进行操作的工作单元和存储现实计算所需信息的辅助空间。程序执行时所需的存储空间包括以下两部分：

- 1) 固定部分。这部分空间的大小与所处理数据的大小和个数无关，主要包括程序

代码、常量、简单变量等所占的空间。

2) 可变部分。这部分空间与程序执行中所处理的特定数据的大小和规模有关, 如 100 个数据元素和 100 个数据元素的排序所需的存储空间是不相同的。

类似于算法的时间复杂度, 算法的空间复杂度记为

$$S(n) = O(f(n))$$

## 5. 算法的描述

从上面的分析可知, 算法是描述某一问题求解的有限步骤, 而且必须有结果输出。设计一个算法或描述一个算法, 最终是由程序设计语言来实现的。但是, 算法与程序有区别, 算法是考虑实现某一个问题的方法与步骤, 是解决问题的框架流程, 而程序是根据这一求解的框架进行语言细化, 是实现这一问题的具体过程。

算法可以使用流程图、自然语言、伪代码等多种不同的方法来描述。

### (1) 流程图

流程图是指用一些图框来表示各种类型的操作, 用流线表示这些操作的执行顺序。用流程图描述算法直观、易懂, 但对于比较复杂的算法显得不够方便。

### (2) 自然语言

自然语言即人们日常进行交流的语言, 如英语、汉语。用自然语言来描述算法通俗易懂, 便于用户之间的交流。但是, 自然语言表示的含义不够准确, 容易产生二义性。

### (3) 伪代码

伪代码介于自然语言与程序设计语言之间, 它以程序设计语言的书写形式来描述算法。伪代码虽不能直接在计算机上运行, 但便于算法实现。

## 6. 常用算法介绍

### (1) 枚举法

枚举法的基本思想是根据提出的问题, 列举所有可能的情况, 并用问题中提出的条件检验哪些是需要的, 哪些是不需要的。枚举法的特点是算法比较简单, 但当列举的可能情况较多时, 执行列举算法的工作量会很大。因此, 在用枚举法设计算法时, 应该重点注意方案优化, 尽量减少运算工作量。

### (2) 归纳法

归纳法的基本思想是通过列举少量的特殊情况, 经过分析, 最后找出一般关系。归纳是一种抽象, 即从特殊现象中找出一般关系。由于在归纳的过程中不可能对所有的可能情况进行列举, 因而最后得到的结论只是一种归纳假设, 对于归纳假设还必须加以严格的证明。

### (3) 递推法

递推法是指从已知的初始条件出发, 逐次推出所要求的各中间结果及最后结果。其中, 初始条件或问题本身已给定, 或可以通过对问题的分析与化简确定。

### (4) 递归法

递归法是指人们在解决复杂问题的时候, 为了降低问题的复杂度, 常常将问题逐层

分解, 把问题转化为规模较小的同类问题的子问题, 直到子问题能够简单求解为止, 再从子问题开始逆序求解原问题。

#### (5) 贪心算法

贪心算法也称贪婪算法, 它在对问题进行求解时, 总是做出在当前看来最好的选择。也就是说, 它不从整体最优上加以考虑, 所得的解一般只是在某种意义上的局部最优解。虽然如此, 但是贪心算法在许多问题上能够产生整体最优解或整体最优解的近似解。贪心算法可以解决图的最小生成树、哈夫曼编码等一些最优化问题。一旦一个问题可以通过贪心算法来解决, 那么贪心算法一般是解决这个问题的最好方法。

贪心算法的基本思路如下:

- 1) 建立数学模型来描述问题。
- 2) 把求解的问题分成若干个子问题。
- 3) 对每一子问题求解, 得到子问题的局部最优解。
- 4) 把子问题的局部最优解合成原问题的一个解。

#### (6) 分治算法

分治算法是把一个复杂的问题分解为若干个与原问题相同或相似的规模较小的子问题, 再把子问题分成更小的子问题, 直到最后可以简单地直接求解, 并通过合并子问题的解得到原问题的解。分治算法的要点是子问题与原问题结构相同, 因此子问题也同样可以利用分治策略进行求解。

分治算法的基本思路如下:

- 1) 分解, 将要解决的问题划分成若干规模较小的同类问题。
- 2) 求解, 当子问题划分得足够小时, 用较简单的方法解决。
- 3) 合并, 按原问题的要求, 将子问题的解逐层合并构成原问题的解。

#### (7) 动态规划算法

动态规划算法与分治算法类似, 其基本思想也是将待求解问题分解成若干个子问题, 先求解子问题, 然后从这些子问题的解得到原问题的解。与分治算法不同的是, 动态规划算法适合于用动态规划方法求解的问题, 经分解得到的子问题往往不是互相独立的。若用分治算法来解这类问题, 则分解得到的子问题数目太多, 有些子问题被重复计算了很多次。动态规划算法正是利用了这种子问题的重叠性质, 每一个子问题只计算一次, 然后将其计算结果保存在一个表格中, 当再次需要计算已经计算过的子问题时, 只需在表格中简单地查看一下结果, 从而获得较高的效率。适用动态规划算法的问题必须满足最优化原理和无后效性。

#### (8) 回溯算法

回溯算法是一种选优搜索法, 按选优条件向前搜索, 以达到目标。但当探索到某一步时, 发现原先的选择并不优或达不到目标, 就退回一步重新选择, 这种走不通就退回再走的方法称为回溯算法, 而满足回溯条件的某个状态的点称为回溯点。

回溯算法解决问题的过程是先选择某一可能的线索进行试探, 每一步试探都有多种方式, 将每一种方式一一试探, 如有问题就返回纠正, 反复进行这种试探再返回纠正, 直到得出全部符合条件的答案或问题无解为止。回溯算法的本质是用深度优先的方法在解的空间树中搜索, 因此算法中需要用堆栈来保存搜索路径。

## 1.1.2 程序的概念

### 1. 程序的定义

程序是用计算机语言描述的某一问题的解决步骤，是符合一定语法规则的指令（语句）序列。通过在计算机上运行程序，向计算机发出一系列指令，便可使计算机按人的要求解决特定的问题。

为解决某一问题所编写的程序并不是唯一的，不同的程序设计人员所设计的程序也不完全相同。不同的程序有不同的效率，这涉及程序设计语言、程序的优化、程序所采用的数据结构及算法等多方面的因素。

一个程序应该包括以下两方面：

- 1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式，即数据结构。
- 2) 对操作的描述。对操作的描述即操作步骤，也就是算法。

### 2. 程序设计语言

用来书写计算机程序的语言称为程序设计语言。程序设计语言可以分为低级语言和高级语言，其中低级语言又分为机器语言和汇编语言。

#### (1) 机器语言

机器语言是指直接用二进制指令表达的计算机语言，指令是用0和1组成的。不同的计算机具有不同的机器语言。用机器语言编写的程序，计算机可以直接执行，其执行效率高。但机器语言的指令不直观，难认、难记、难理解且烦琐，容易出错。编写机器语言程序时，要求程序员必须相当熟悉计算机的结构，因此很少直接用机器语言编写程序。

#### (2) 汇编语言

为了减轻使用机器语言编程的负担，人们采用一些助记符来表示机器语言中的机器指令，这样便形成了汇编语言，汇编语言也称为符号语言。助记符一般是代表某种操作的英文字母的缩写。与机器语言相比，汇编语言便于识别和记忆。汇编语言通常是为特定的计算机或系列计算机而设计的，不同的机器具有不同的汇编语言，因此，它也是面向机器的语言。使用汇编语言编写的程序能较好地发挥机器的特性，但是编写程序时仍需要对计算机的内部结构比较熟悉，汇编语言依然烦琐、复杂。

使用汇编语言编写的源程序，机器不能直接执行，需要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序称为汇编程序，汇编程序是系统软件中的语言处理系统软件。汇编程序把汇编语言翻译成机器语言的过程称为汇编。

#### (3) 高级语言

机器语言和汇编语言都是面向机器的语言，它们同属于低级语言。在使用它们设计程序时，要求对机器比较熟悉。为了克服低级语言的缺点，人们将程序设计的精力集中到解决问题的算法上，便出现了面向算法过程的程序设计语言，称为算法语言，也称为高级语言。高级语言接近自然语言的形式，可以方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，易学易懂。因此，使用高级语言可以大大降低编程的难度，提高编程的效率与质量。

高级语言不依赖于机器，为某种类型的计算机编写的高级语言程序，可以很方便地移植到其他类型的计算机上运行，但其执行的速度相对较慢。使用高级语言编写的源程序，机器同样不能直接执行。如同汇编语言一样，用高级语言编写的程序必须经过翻译，才能由计算机执行。

## 1.2 数据结构

数据结构是相互之间存在一种或多种关系的数据元素的集合。

### 1.2.1 数据结构的基本概念

#### 1. 数据

数据是描述客观事物的符号表示，是能够输入计算机并且能被计算机程序处理的符号的总称，包括数值、字符、图形、图像、声音等内容。

#### 2. 数据元素

数据元素是组成数据的基本单位，是数据集合中的个体。数据元素用于完整地描述一个对象，如一个学生的信息，包括学号、姓名、性别、院系、专业，这些信息合在一起才能描述这个学生，那么这些信息的整体就是一个数据元素。

#### 3. 数据项

数据项是组成数据的、有独立含义的、不可分割的最小单位，如学生信息中的学号就是一个数据项。

#### 4. 数据对象

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。例如，整数对象的集合是  $N = \{0, \pm 1, \pm 2, \dots\}$ ，英文大写字母数据对象的集合是  $C = \{A, B, C, \dots, Z\}$ 。

#### 5. 数据类型

数据类型是和数据结构密切相关的一个概念，在使用高级程序语言编写的程序中，每个变量、常量或表达式都有确定的数据类型。类型明显或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围，以及在这些值上允许进行的操作。

### 1.2.2 数据的基本结构

数据结构就是符合某种结构的数据元素的集合，“结构”就是指数据元素之间的关系。数据结构包括逻辑结构和物理结构，这两种结构互相独立。

#### 1. 逻辑结构

数据的逻辑结构只抽象地反映数据元素之间的逻辑关系，而不考虑其在计算机中的存储方式。数据的逻辑结构可以看作从具体问题抽象出来的数学模型。