

“十一五”高等院校规划教材



C语言程序设计

(计算机二级教程)

马俊 夏美云 主编



北京航空航天大学出版社

“十一五”高等院校规划教材

C 语言程序设计

(计算机二级教程)

马俊 夏美云 主编

北京航空航天大学出版社

内 容 简 介

依据高等院校“C 语言程序设计”课程教学内容的基本要求而编写,充分考虑到理论与实践的结合,在讲解 C 语言程序设计基本知识的同时,更注重讲解相应的程序设计技巧、常用算法以及具有实用价值的程序实例,并设有专门章节介绍上机步骤、调试技巧。本书既有严密完整的理论体系,又具有较强的实用性。

本书主要内容包括二级考试基础知识、C 语言程序设计概述、基本数据类型、运算符及表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数、指针、数组、用户标识符的作用域和存储类别、编译预处理和动态存储分配、结构体与共用体、位运算、文件、面向对象程序设计基础、上机考试指导共 16 章。书中给出了大量的例题和习题,书后给出了附录,便于学生自学。

本书适合普通高等院校本、专科计算机与非计算机专业作为“C 语言程序设计”课程教材使用,也适合 C 语言初学者用作计算机二级考试的学习与参考用书。

图书在版编目(CIP)数据

C 语言程序设计(计算机二级教程)/马俊,夏美云主编. —北京:北京航空航天大学出版社,2009.9

ISBN 978-7-81124-904-0

I. C… II. ①马…②夏… III. C 语言—程序设计—水平
考试—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 156440 号

© 2009,北京航空航天大学出版社,版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制或传播本书内容。
侵权必究。

C 语言程序设计(计算机二级教程)

马 俊 夏美云 主编

责任编辑 冯 颖

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:emsbook@gmail.com

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:24 字数:614 千字

2009 年 9 月第 1 版 2009 年 9 月第 1 次印刷 印数:4 000 册

ISBN 978-7-81124-904-0 定价:38.00 元

前 言

目前我国高等教育已进入普及时代,如何培养满足市场需求的应用型人才,是一个重要课题。掌握计算机知识和应用,无疑是培养新型人才的一个重要环节。计算机技术已与其它学科相互交融,成为推动社会发展的动力。无论什么专业的学生,都必须具备计算机的基础知识和应用能力。因为计算机技术已经成为高等院校全面素质教育中极为重要的一部分。

近年来,由于C语言具有功能丰富、表达力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等特点,所以被计算机专业和非计算机专业应用人员所使用。许多高等院校不仅在计算机专业开设了C语言课程,而且也在非计算机专业开设了C语言课程。全国计算机等级考试、全国计算机应用技术证书考试和全国各地组织的大学生计算机统一考试都将C语言列入了考试范围。因此,学习C语言已经成为广大计算机应用人员的基本要求。

由于C语言涉及的概念比较复杂,规则繁多,使用灵活,容易出错,不少初学者感到困难,所以作者在北京航空航天大学出版社的支持下,根据长期从事第一线教学的经验,编写了本书。本教材根据读者对象的性质,力图体现以下特色:

① 起点较低,不需具备程序设计语言基础知识。很多C语言的教材都要求读者先前学过一门程序设计语言。但本教材从程序设计的基础知识讲起,把一些经典算法的来龙去脉交代清楚,读者不需要有其它程序设计语言的基础即可学懂。

② 概念准确,编排合理。在内容编排上,注意分散难点,便于读者循序渐进地学习。

③ 详略得当,重点突出。本书主要讲解C语言最基本、最常用的内容,控制C语言中出现频率很低或与语言的实践版本相关内容的篇幅,把重点放在语言本身的难点(如指针)和程序设计技巧方面。

④ 深入浅出,讲解通俗。根据应用型人才的培养特点,采用基础知识加例题的方法,使读者能够尽快掌握相关知识。

⑤ 强化实践,重视应用。本教材力求使读者学完之后,不仅能学会C语言的语法、语义,更重要的是掌握C语言程序设计的技巧,具备编程解决实际问题的能力。所以本书结合全国计算机等级考试,在各章后提供了较多的习题,使读者能够得到有效的训练。



参加本书编写的人员均为长期从事 C 语言教学的一线教师,具有丰富的教学经验。本书由马俊、夏美云担任主编,负责制定编写要求和详细的内容编写目录,并对全书进行统稿和定稿。郭永利、卢珂、李爽任副主编,负责协助主编工作。第 1、2 章由马俊编写,第 3、6、11 章由郭永利、卢珂共同编写,第 5、7、16 章由李爽、张彦峰共同编写,第 4、9 章由周晓燕、王文志共同编写,第 13、14 章和附录 D 由刘庆华编写,第 8、10、15 章和附录 A~C 由马军涛和赵锋共同编写,第 12 章由夏美云编写。本书由河南理工大学李长有副教授负责审阅,李老师在百忙中认真细致地审阅了全部书稿,并提出了宝贵建议。北京航空航天大学出版社的工作人员为本书的成功出版付出了艰辛的劳动。编者在此对为本书成功出版作出贡献的所有工作人员表示衷心的感谢。

由于编者水平有限,对于书中存在的错误与不当之处,敬请读者指正,以便不断改进。有兴趣的读者,可以发送电子邮件到 jzzf@live.cn,与作者进一步交流;也可以发送电子邮件到 buaafy@sina.com,与本书策划编辑进行交流。

编者
2009 年 7 月

目 录

第 1 章 二级考试基础知识

1.1 程序设计基础	1
1.1.1 面向结构的程序设计	1
1.1.2 面向对象的程序设计	2
1.2 数据结构	4
1.2.1 算 法	5
1.2.2 链表、队列、栈的基本概念	5
1.2.3 二叉树的遍历	7
1.3 数据库	9
1.3.1 数据、信息和数据处理	9
1.3.2 数据库系统概述	10
1.3.3 数据库描述	11
1.3.4 数据库管理系统	13
1.4 软件工程	14
1.4.1 软件工程的基本概念	14
1.4.2 结构化分析方法	15
1.4.3 结构化设计方法	15
1.4.4 软件测试	16
1.4.5 程序的调试	17
本章小结	17
历年试题汇集	17

第 2 章 C 语言程序设计概述

2.1 C 语言概述	26
2.1.1 C 语言的发展	26
2.1.2 C 语言的特点	27
2.2 简单的 C 程序构成及格式	27



2.3 C 语言开发工具	29
2.3.1 C 语言的执行过程	29
2.3.2 Visual C++ 开发环境介绍	30
2.4 良好的程序设计风格	31
本章小结	33
历年试题汇集	33
课后练习	34

第 3 章 基本数据类型、运算符及表达式

3.1 C 语言的数据类型	36
3.2 常量、变量和标识符	37
3.2.1 标识符	37
3.2.2 常量	38
3.2.3 变量	39
3.3 整型数据	40
3.3.1 整型常量	41
3.3.2 整型变量	42
3.4 实型数据	44
3.4.1 实型常量	44
3.4.2 实型变量	45
3.5 字符型数据	46
3.5.1 字符型常量	46
3.5.2 字符型变量	47
3.5.3 字符串常量	47
3.6 C 语言的运算符与表达式	48
3.6.1 C 语言运算符的种类	48
3.6.2 算术运算符及表达式	48
3.6.3 关系运算符及表达式	50
3.6.4 逻辑运算符及表达式	51
3.6.5 条件运算符与表达式	52
3.6.6 赋值运算符及表达式	53
3.6.7 逗号运算符及表达式	53
3.7 数据类型转换	54
3.7.1 自动类型转换	54
3.7.2 强制类型转换	55
本章小结	55
历年试题汇集	55
课后练习	60

第 4 章 顺序结构程序设计

4.1 C 语言的 3 种基本结构	63
4.1.1 流程图	63
4.1.2 3 种基本结构	64
4.2 C 语言的语句	65
4.3 格式输入/输出函数	66
4.3.1 格式输出函数——printf 函数	67
4.3.2 格式输入函数——scanf 函数	74
4.4 字符数据的输入/输出函数	76
4.4.1 字符输出函数——putchar 函数	76
4.4.2 字符输入函数——getchar 函数	77
4.5 顺序结构程序举例	77
本章小结	79
历年试题汇集	80
课后练习	85

第 5 章 选择结构程序设计

5.1 选择结构概述	91
5.2 if 语句的 3 种基本形式	91
5.2.1 if 语句的 3 种形式	92
5.2.2 if 语句的嵌套	96
5.3 switch 语句	97
5.4 选择结构程序举例	100
本章小结	102
历年试题汇集	103
课后练习	107

第 6 章 循环结构程序设计

6.1 循环结构	114
6.2 while 语句	115
6.3 do-while 语句	116
6.4 for 语句	118
6.5 break 语句和 continue 语句	121
6.5.1 break 语句	121
6.5.2 continue 语句	122
6.6 goto 语句	123
6.7 循环的嵌套	125
6.8 循环结构程序举例	126



本章小结	127
历年试题汇集	128
课后练习	136
第 7 章 函 数	
7.1 函数的分类	142
7.2 函数的定义	143
7.2.1 无参函数	143
7.2.2 有参函数	144
7.2.3 空函数	145
7.3 函数的参数及其返回值	145
7.3.1 形式参数和实际参数	145
7.3.2 函数的返回值	146
7.4 函数的调用	148
7.4.1 函数调用的一般形式	148
7.4.2 函数调用的方式	148
7.4.3 被调函数的声明	149
7.4.4 函数的嵌套调用	149
7.4.5 函数的递归调用	151
7.5 函数应用举例	154
本章小结	155
历年试题汇集	156
课后练习	164
第 8 章 指 针	
8.1 地址和指针的概念	168
8.2 指针变量	169
8.2.1 指针变量的定义	169
8.2.2 指针变量的初始化	169
8.2.3 指针变量的基本运算	170
8.2.4 指向指针的指针变量	171
8.3 指针与函数	172
8.3.1 函数的形参为指针类型	172
8.3.2 函数返回值为指针类型	173
8.3.3 指向函数的指针	174
本章小结	176
历年试题汇集	176
课后练习	182

第 9 章 数 组

9.1 数组的引出	186
9.2 一维数组	187
9.2.1 一维数组的定义	187
9.2.2 一维数组的初始化	189
9.2.3 一维数组的引用	189
9.2.4 一维数组和指针	190
9.2.5 一维数组的应用举例	192
9.3 二维数组	195
9.3.1 二维数组的定义	195
9.3.2 二维数组的初始化	195
9.3.3 二维数组的引用	196
9.3.4 二维数组和指针	198
9.3.5 二维数组的应用举例	200
9.4 字符数组与字符串	201
9.4.1 字符数组	201
9.4.2 字符串	202
9.4.3 字符串处理函数	202
9.4.4 字符串和指针	205
9.4.5 字符数组应用举例	205
本章小结	209
历年试题汇集	210
课后练习	218

第 10 章 用户标识符的作用域和存储类型

10.1 用户标识符的作用域	224
10.2 用户标识符的存储类型	224
10.3 用户标识符的生存期	225
10.3.1 静态变量的存储类型和作用域	225
10.3.2 动态变量的存储类型和作用域	225
10.3.3 局部变量的作用域和生存期	226
10.4 全局变量的作用域和生存期	228
10.4.1 全局变量的作用域和生存期	228
10.4.2 在同一编译单位内用 extern 说明全局变量的作用域	229
10.4.3 在不同编译单位内用 extern 说明全局变量的作用域	229
10.4.4 static 全局变量	229
10.5 函数的存储类型	230
本章小结	230



历年试题汇集..... 230
课后练习..... 238

第 11 章 编译预处理和动态存储分配

11.1 编译预处理..... 244
 11.1.1 宏定义..... 244
 11.1.2 文件包含..... 247
 11.1.3 条件编译..... 248
11.2 动态存储分配..... 249
 11.2.1 malloc 函数和 free 函数 249
 11.2.2 calloc 函数 250
本章小结..... 250
历年试题汇集..... 251
课后练习..... 254

第 12 章 结构体与共用体

12.1 结构体的引出..... 258
12.2 结构体类型..... 259
 12.2.1 结构体类型的定义和结构体变量的定义..... 259
 12.2.2 结构体变量的引用..... 261
 12.2.3 结构体变量的初始化..... 261
12.3 结构体数组..... 262
 12.3.1 结构体数组的定义..... 262
 12.3.2 结构体数组的初始化..... 263
 12.3.3 结构体数组的应用..... 263
 12.3.4 结构体指针..... 264
 12.3.5 结构体与函数..... 266
 12.3.6 链 表..... 268
12.4 共用体..... 270
 12.4.1 共用体的概念及特点..... 270
 12.4.2 共用体类型的定义..... 271
 12.4.3 共用体变量的引用..... 272
12.5 枚 举..... 273
12.6 用 typedef 定义类型 275
本章小结..... 276
历年试题汇集..... 276
课后练习..... 291

第 13 章 位运算

13.1 位运算符	295
13.2 位运算符的运算功能	296
13.2.1 按位与运算符(&)	296
13.2.2 按位或运算符()	298
13.2.3 按位异或运算符(^)	298
13.2.4 按位取反运算符(~)	299
13.2.5 左移运算符(<<)	299
13.2.6 右移运算符(>>)	300
13.3 位复合赋值运算符	301
本章小结	301
历年试题汇集	301
课后练习	303

第 14 章 文 件

14.1 概 述	305
14.1.1 文件的概念	305
14.1.2 文件的分类	305
14.1.3 文件类型指针	306
14.2 文件的操作	306
14.2.1 文件的打开函数	306
14.2.2 文件的关闭函数	308
14.2.3 单个字符读/写函数	308
14.2.4 字符串读/写函数	310
14.2.5 数据块读/写函数	310
14.2.6 格式化读/写函数	311
14.2.7 文件的定位函数	312
14.3 文件程序举例	313
本章小结	315
历年试题汇集	316
课后练习	321

第 15 章 面向对象程序设计基础

15.1 C++语言概述	324
15.2 类和对象	325
15.3 数据的抽象和封装	325
15.3.1 类的封装	325
15.3.2 类的定义	325



15.3.3 类的成员函数	326
15.3.4 构造函数和析构函数	326
15.3.5 创建对象	327
15.3.6 友元	329
15.4 继承性	331
15.5 多态性	333
15.6 程序举例	335
本章小结	336
历年试题汇集	336
课后练习	339
第 16 章 上级考试指导	
16.1 上机考试系统说明	343
16.1.1 上机考试时间	343
16.1.2 上机考试题型及分值	343
16.1.3 上机考试登录	343
16.1.4 试题内容查阅工具的使用	345
16.1.5 编辑、连接和运行	346
16.1.6 考生文件夹和文件的恢复	347
16.1.7 文件名的说明	347
16.2 上机考试内容	347
16.2.1 程序填空题	347
16.2.2 程序修改题	349
16.2.3 程序编写题	350
本章小结	351
历年试题汇集	351
课后练习	352
附录 A C 语言常用关键字及说明	359
附录 B ASCII 码表	360
附录 C C 语言运算符及优先级	361
附录 D 常用库函数	362
参考文献	370

第 1 章

二级考试基础知识

【本章考点和学习目标】

1. 结构化程序设计与面向对象程序设计的特点和区别。
2. 各种数据结构的概念和特点;实现各种数据结构的算法。
3. 数据库的基本概念和原理。
4. 软件工程的基本概念和主要特点。

【本章重难点】

重点:数据结构的概念和特点。

难点:数据结构的算法。

本章主要从全国计算机等级考试二级考试的角度出发,介绍了程序设计、数据结构、数据库和软件工程 4 个方面的基本概念和理论。

1.1 程序设计基础

1.1.1 面向结构的程序设计

结构化程序的概念首先是从以往编程过程中无限制地使用转移语句而提出的。转移语句可以使程序的控制流程强制性地转向程序的任一处,在传统流程图中就是用“很随意”的流程线来描述这种转移功能的。如果一个程序中多处出现这种转移情况,将会导致整个程序流程无序可寻,程序结构杂乱无章。这样的程序是令人难以理解和接受的,并且容易出错。在实际软件产品开发过程中,为了增强软件的可读性,往往限制转移语句的使用,而改用循环语句或选择语句。很多理论和实践已经证明,选择语句和循环语句完全可以替代转移语句,而不增加软件的实现难度。结构化程序由迪克斯特拉(E. W. Dijkstra)在 1969 年提出,它是以模块化设计为中心,将待开发的软件系统划分为若干个相互独立的模块,这样使完成每一个模块的工作变得单纯而明确,为设计一些较大的软件打下了良好的基础。

由于模块相互独立,因此在设计其中一个模块时,不会受到其它模块的牵连,因而可将原来较为复杂的问题化简为一系列简单模块的设计。模块的独立性还为扩充已有的系统、建立新系统带来了不少的方便,因为我们可以充分利用现有的模块作积木式的扩展。

按照结构化程序设计的观点,任何算法功能都可以通过由程序模块组成的 3 种基本程序结构的组合——顺序结构、选择结构和循环结构来实现。

顺序结构:顺序结构是一种线性、有序的结构,它依次执行各语句模块。

选择结构:选择结构是根据条件成立与否选择程序执行的通路。



循环结构:循环结构是重复执行一个或几个模块,直到满足某一条件为止。

采用结构化程序设计方法,可使程序结构清晰,易于阅读、测试、排错和修改。由于每个模块执行单一功能,模块间联系较少,使程序的编制比过去更简单,程序运行更可靠,而且增加了可维护性,因为每个模块都可以独立编制、测试。结构化程序设计是为了解决早期计算机程序难以阅读、理解和调试,难以维护和扩充,以及开发周期长,不易控制程序的质量等问题而提出来的,它的产生和发展奠定了软件工程的基础。

结构化程序设计的基本思想是:自顶向下,逐步求精,将整个程序结构划分成若干个功能相对独立的模块,模块之间的联系尽可能简单;每个模块用顺序、选择和循环 3 种基本结构来实现;每个模块只有一个入口和一个出口。

结构化程序设计有很多优点:各模块可以分别编程,使程序易于阅读、理解、调试和修改;方便新功能模块的扩充;功能独立的模块可以组成子程序库,有利于实现软件复用等。因此,结构化程序设计方法出现以后,很快被人们接受并得到广泛应用。

结构化程序设计方法以解决问题的过程作为出发点,其方法是面向过程的。它把程序定义为“数据结构+算法”,程序中数据与处理这些数据的算法(过程)是分离的。这样,对不同的数据结构作相同的处理,或对相同的数据结构作不同的处理,都要使用不同的模块,从而降低了程序的可维护性和可复用性。同时,由于这种分离,导致了数据可能被多个模块使用和修改,难以保证数据的安全性和一致性。因此,对于小型程序和中等复杂程度的程序来说,它是一种较为有效的技术,但对于复杂的、大规模的软件来说,使用它就难以维护和复用。

1.1.2 面向对象的程序设计

面向对象的程序设计(Object - Oriented Programming, 简记为 OOP)立意于创建软件重用代码,具备更好地模拟现实世界环境的能力,这使它被公认为是自上而下编程的优胜者。它通过给程序中加入扩展语句,把函数“封装”进编程所必需的“对象”中。面向对象的编程语言使得复杂的工作条理清晰、编写容易。说它是一场革命,不是对对象本身而言,而是对它们处理工作的能力而言。

面向对象程序设计是在结构化程序设计的基础上发展起来的,它吸取了结构化程序设计中最为精华的部分,有人称它是“被结构化了的结构化程序设计”。

1. 类

许多对象具有相同的结构和特性,例如不管是数学书还是化学书,它们都具有大小、定价、编者等特性。在现实生活中,我们通常将具有相同性质的事物归纳、划分成一类,例如数学书和化学书都属于“书”这一类。同样在面向对象程序设计中也会采用这种方法。在面向对象程序设计中,引入了一个全新的数据类型——类。

类是对相同类型的数据结构进行抽象,形成各种属性和行为。属性是数据结构的表现形式,行为则对数据结构进行处理。类将数据结构及其处理代码隐蔽起来,仅通过一个可控的接口与外界交互。

2. 对象

从一般意义上讲,客观世界中的任何一个事物都可以看成是一个对象,例如一本书,一名

学生等。对象具有自己的静态特征和动态特征。静态特征可以用某种数据来描述,如一名学生的身高、年龄、性别等;动态特征是对象所表现的行为或具有的功能,如学生学习、运动、休息等。

面向对象方法中的“对象”是系统中用来描述客观事物的一个实体,它是用来构成系统的一个基本单位,由一组属性和一组行为构成。其中,属性是用来描述对象静态特征的数据项,行为是用来描述对象动态特征的操作序列。

类代表了一组对象的共性和特征,类是对象的抽象,而对象是类的具体实例。例如,家具设计师按照家具的设计图做成一把椅子,那么设计图就好比是一个类,而做出来的椅子则是该类的一个对象,一个具体实例。

3. 类的封装

类的封装就是将对象的属性和行为结合成一个独立的实体,并尽可能隐蔽对象的内部细节,对外形成一道屏障,只保留有限的对外接口使之与外界发生联系。类的成员包括数据成员和成员函数,分别描述类所表达问题的属性和行为。对类成员的访问加以控制就形成了类的封装,这种控制是通过设置成员的访问权限来实现的。

在面向对象程序设计中,通过封装将一部分行为作为外部接口,而将数据和其它行为进行有效的隐蔽,就可以达到对数据访问权限的合理控制。把整个程序中不同部分间的相互影响降到最低。

4. 类的继承

面向对象的程序设计过程中,一个很重要的特点就是代码的可重用性。C++是通过继承这一机制来实现代码重用的。所谓“继承”指的是类的继承,也就是在原有类的基础上建立一个新类,新类将从原有类那里得到已有的特性。例如,现有一个学生类,定义了学号、姓名、性别、年龄4项内容,如果除了用到以上4项内容外,还需要用到电话和地址等信息,那么我们就可以在学生类的基础上再增加相关的内容,而不必再重新定义一个类。

换个角度来说,从已有类产生新类的过程就是类的派生。类的继承与派生允许在原有类的基础上进行更具体、更详细的修改和扩充。新的类由原有的类产生,包含了原有类的关键特征,同时也加入了自己所特有的性质。新类继承了原有类,原有类派生出新类。原有的类称为基类或父类,而派生出的类称为派生类或子类。比如:所有的 Windows 应用程序都有一个窗口,可以说它们都是从一个窗口类中派生出来的,只不过有的应用程序应用于文字处理,有的则应用于图像显示。

类的继承与派生的层次结构是人们对自然界中事物进行分类、分析、认识过程在程序设计中的体现。现实世界中的事物都是相互联系、相互作用的,人们在认识过程中,根据它们的实际特征,抓住其共同特性和细小差别,利用分类的方法进行分析和描述。

5. 类的多态性

面向对象程序设计过程中的另外一个重要特点是多态性。多态性是指同一操作作用于不同的对象,可以有不同的解释,产生不同的执行结果。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势,很好地解决了应用程序函数同名的问题。多态性通过派生类重载基类中的虚函数型方法来实现。



同一操作作用于不同的对象,可以有不同的解释,并产生不同的执行结果,这就是多态性。多态性通过派生类重载基类中的虚函数型方法来实现。

现实世界中的对象之间存在着各种各样的联系。正是这种联系和相互作用,才构成了世界中的不同系统。同样,面向对象程序设计中的对象之间也存在着联系,称为对象的交互;提供对象交互的机制称为消息传递。当某个行为作用于对象时,称该对象执行了一个方法,这个方法定义了该对象要执行的一系列计算步骤,所以方法是对象操作过程的算法。一个对象向另一个对象发出的请求称为消息,它是一个对象要求另一个对象执行某个操作的规格说明,通过消息传递才能完成对象之间的相互请求和协作。类提供了完整地解决特定问题的条件,它描述了数据结构(对象属性)、算法(行为或方法)和外部接口(消息)。对象通过外部接口接收它能识别的消息,按照自己的方式来解释这个消息并调用某个方法来执行对数据的处理,从而完成对特定问题的解决。

1.2 数据结构

在现实社会中存在着许多非数值计算问题,其数学模型难以用数学方程描述。非数值计算问题的数学模型是表、树和图之类的数据结构。

数据结构是一门介于数学、计算机硬件、计算机软件之间的核心课程,是设计和实现编译系统、操作系统、数据库系统及其它系统程序和各種应用程序的基础,是一门研究非数值计算的程序设计问题中计算机操作对象以及它们之间关系与操作的学科。它主要包括 3 个要素:对象、关系及操作(运算)。

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。数据结构是一个二元组,记为 $\text{data_structure} = (D, S)$ 。其中, D 为数据元素的集合, S 是 D 关系的集合。

数据元素相互之间的关系称为结构(Structure)。根据数据元素之间关系的不同特性,通常有下列 4 类基本结构:

集合——数据元素间的关系是同属一个集合。

线性结构——数据元素间存在一对一的关系。

树形结构——数据元素间的关系是一对多的关系。

图(网)状结构——数据元素间的关系是多对多的关系。

1968 年美国克努特教授开创了数据结构的最初体系:数据的逻辑结构、存储结构及其操作。即数据结构一般包括以下 3 方面的内容:

逻辑结构——数据元素之间的逻辑关系。

存储结构——数据元素及其关系在计算机存储器的表示。

数据运算——对数据元素施加的操作。

其中,数据的存储结构一般分成 2 类:顺序存储结构和链式存储结构。

顺序存储结构——把逻辑上相邻的结点存储在物理位置上相邻的存储单元里,结点间的逻辑关系由存储单元的邻接关系来体现。通常顺序存储结构是借助于语言的数组来描述的。

链式存储结构——不要求逻辑上相邻的结点物理上也相邻,结点间的逻辑关系是由附加的指针表示的,通常要借助于语言的指针类型来描述。