

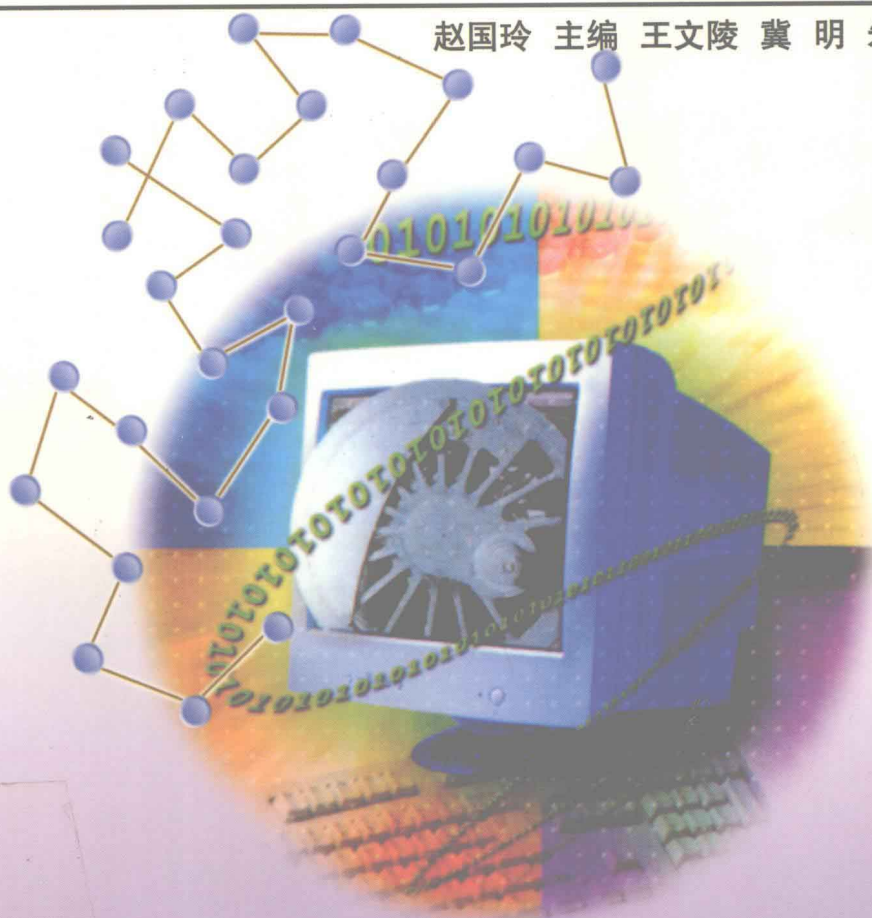
中国电子教育学会中专教育委员会  
全国中专电子类教材协会

推荐教材

中等专业学校教材

# C语言与数据结构

赵国玲 主编 王文陵 冀明 朱宪花 编



电子工业出版社

Publishing House of Electronics Industry

URL: <http://www.phei.com.cn>

中等专业学校教材

# C 语言与数据结构

赵国玲 主编

王文陵 冀 明 朱宪花 编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书共 12 章,主要包括:C 语言与数据结构的基本概念,C 语言的数据类型,C 语言基本操作和语法规则,利用 C 语言进行结构化和模块化程序设计的基本方法,线性表,树和图的数据结构及应用、排序及查找的常用算法,C 语言对图形、窗口和菜单的处理等。

本书在选材上力求精练,问题描述深入浅出,并通过大量实例进行说明。每章后都配有习题和实验内容,供课后练习及上机使用。本书既可作为中等专业学校计算机及相关专业的教材使用,也可作为广大计算机工作者的参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

C 语言与数据结构/赵国玲主编;王文陵等编. - 北京:电子工业出版社,1999.10  
(中等专业学校教材)

ISBN 7-5053-5288-1

I .C… II .①赵… ②王… III .①C 语言-程序设计-专业学校-教材  
②数据结构-专业学校-教材 IV .TP31

中国版本图书馆 CIP 数据核字(1999)第 36776 号

责任编辑:赵家鹏

印 刷 者:北京天宇星印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:19.75 字数:497.6 千字

印 次:2004 年 6 月第 5 次印刷

印 数:3000 册 定价:25.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 [zlhs@phei.com.cn](mailto:zlhs@phei.com.cn),盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

## 出版说明

随着中等专业学校电子类专业教学改革的不深入,尽快组织出版一批适应中专学校教学实际、体现职业技术教育特点的教材,已成为各中专校的迫切要求。有鉴于此,中国电子教育学会中专教育专业委员会、全国中专电子类教材协会决定联合成立全国中专电子类教材工作领导小组,组织出版一套中专电子类教材,以满足中专学校的教学需要。经过一段时期的准备,领导小组会同全国二十余所电子类中等专业学校,成立了“计算机及应用”、“电子技术应用”、“机电技术应用”3个专业教材编委会,共同组织协调这套教材的编审出版工作。

领导小组和各编委会确立了“根据中专生的培养目标,贯彻中专教育适应社会经济发展的需要,强化应用为教学重点的思想,反映现代职业教育思想、教育方法和教学手段以及综合化、直接化、形象化特点,突出工程实践能力培养”的编写原则,以“新、简、实”作为这套教材的编写特色。所谓“新”,是根据电子技术日新月异、发展迅速的特点,在教材中尽可能反映当前电子信息产业的新技术、新知识、新工艺,缩短教材编审出版周期;所谓“简”,是针对现行教学内容与中专学生的文化基础不相适应,以及中专毕业生越来越直接面向生产第一线这一现实,适当降低教学内容的深度和难度,简化理论知识的讲授;所谓“实”,就是突出教学内容的实用性,强调对学生实践能力和技术应用能力的培养。

各编委会的编审程序大致是,针对中专计算机及应用、电子技术应用、机电技术应用(机电一体化)的教学现状和现行教材存在的问题,尤其是针对目前中专教学改革的新情况,拟定各专业方向的课程设置计划和教材选题计划。在充分酝酿、广泛征集的基础上,由编委会确定每个选题的编写大纲和编审人员。编委会通过责任编委联系制度对编写实行质量控制。

这套教材的编写,都是来自各中专学校教学第一线的经验丰富的教师。由于他们辛勤的工作,这套教材基本反映了近年来各中专学校教学与教材改革的成果。相信这套教材会受到中等专业学校和其他中等职业学校电子类专业广大师生的欢迎。

特别应该感谢电子工业出版社高质量、高效率的工作,为这套教材的出版提供了极大的便利,使之能及早与读者见面。

电子技术发展迅速,中专学校的教学内容也日新月异。我们衷心地希望广大师生对本套教材提出意见和建议,以便再版时予以修正。

全国中专电子类教材工作领导小组  
电 子 工 业 出 版 社

# 前 言

C 语言是当今世界最有影响的程序设计语言之一,它既适合系统程序的设计,也适合应用程序的设计。近年来,C 语言在系统软件、工具软件、软件平台、图像处理、数值分析、人工智能和数据处理等方面都得到了广泛的应用。因此,C 语言已成为广大计算机应用人员必不可少的语言工具。

计算机处理的对象是数据,数据是有结构的,程序设计不仅要有扎实的语言基础,还必须熟悉数据的结构,这样在进行数据处理时,才能选择适当的数据结构和存储结构及相应的算法,使程序设计更加合理、规范,以提高程序的运行效率。

本书以 C 语言为主线,将各种类型的数据结构揉和其中,使 C 语言与数据结构有机地结合在一起,这不但解决了数据结构中的算法描述问题,而且在算法的实现过程中用 C 语言进行程序设计的能力也得到了加强,两者相辅相成,互相促进。

全书共分为 12 章。第 1 章主要介绍 C 语言与数据结构的基本概念。第 2 章介绍 C 语言的基本数据类型及其操作。第 3 章引入结构化程序设计方法,介绍三种基本结构的实现。第 4 章引入模块化程序设计的概念,介绍 C 语言函数的定义、调用及函数间的数据传递。第 5 章和第 6 章是本书的重点部分,介绍 C 语言的精髓部分数组和指针,同时将数据结构中线性表的内容贯穿其中,并结合实例介绍排序及查找的常用算法。第 7 章介绍 C 语言的结构体和共用体数据类型,并介绍了线性表的链式存储及操作。第 8 章和第 9 章介绍数据结构中两种典型的非线性结构树和图。第 10 章和第 11 章分别介绍 C 语言的文件操作及位运算。第 12 章介绍 C 语言对图形、窗口和菜单的处理。

本书第 1 章至第 5 章和第 12 章由山东省电子工业学校赵国玲老师编写,第 6 章、第 9 章及第 7 章的部分内容(7.5~7.7)由福建省电子工业学校王文陵老师编写,第 8 章由山东省电子工业学校朱宪花老师编写,第 10 章、第 11 章和第 7 章(除 7.5~7.7 外的部分)由辽宁电子工业学校冀明老师编写,并由赵国玲老师统编全书。山东师范大学刘弘教授审阅了全书。

在本书的编写过程中,首先得到了山东省电子工业学校校长郑三高级讲师、副校长王协瑞高级讲师、戚琦高级讲师和梁军高级讲师等的大力支持和帮助,软件教研室的王宜贵、张磊、任文娟等老师也对本书的编写提出了许多宝贵意见,谨此表示由衷的感谢。

由于编者水平有限,书中难免还存在一些错误和疏漏,恳请广大读者批评指正。

编 者

1999 年 6 月

# 目 录

|                                |        |
|--------------------------------|--------|
| <b>第 1 章 概述</b> .....          | ( 1 )  |
| 1.1 C 语言的发展和特点 .....           | ( 1 )  |
| 1.1.1 程序设计语言 .....             | ( 1 )  |
| 1.1.2 C 语言简介 .....             | ( 2 )  |
| 1.1.3 C 语言的特点 .....            | ( 3 )  |
| 1.2 数据结构概述 .....               | ( 3 )  |
| 1.2.1 数据结构的基本概念和术语 .....       | ( 3 )  |
| 1.2.2 关于算法分析 .....             | ( 4 )  |
| 1.3 C 语言程序结构及特点 .....          | ( 5 )  |
| 1.3.1 简单程序实例 .....             | ( 5 )  |
| 1.3.2 C 语言程序的结构特点 .....        | ( 6 )  |
| 1.4 C 语言程序上机步骤 .....           | ( 6 )  |
| 1.4.1 TurboC 中运行 C 程序的步骤 ..... | ( 7 )  |
| 1.4.2 TurboC 的编辑功能键 .....      | ( 9 )  |
| 本章小结 .....                     | ( 11 ) |
| 实验一 C 语言程序的运行 .....            | ( 11 ) |
| 习题一 .....                      | ( 12 ) |
| <b>第 2 章 数据类型与基本操作</b> .....   | ( 13 ) |
| 2.1 C 语言的数据类型 .....            | ( 13 ) |
| 2.2 常量和变量 .....                | ( 14 ) |
| 2.2.1 常量 .....                 | ( 14 ) |
| 2.2.2 变量 .....                 | ( 16 ) |
| 2.2.3 各种类型数据的存储和取值范围 .....     | ( 18 ) |
| 2.2.4 变量赋值 .....               | ( 19 ) |
| 2.3 运算符和表达式 .....              | ( 20 ) |
| 2.3.1 C 语言所提供的运算符 .....        | ( 20 ) |
| 2.3.2 算术运算 .....               | ( 21 ) |
| 2.3.3 赋值运算 .....               | ( 22 ) |
| 2.3.4 复合赋值运算 .....             | ( 23 ) |
| 2.3.5 逗号运算 .....               | ( 23 ) |
| 2.4 不同类型数据间的转换 .....           | ( 24 ) |
| 2.4.1 不同类型数据的隐式转换 .....        | ( 24 ) |
| 2.4.2 不同类型数据的显式转换 .....        | ( 25 ) |
| 2.5 基本输入与输出 .....              | ( 25 ) |
| 2.5.1 数据输出 .....               | ( 26 ) |
| 2.5.2 数据输入 .....               | ( 29 ) |

|                              |      |
|------------------------------|------|
| 2.5.3 数据输入、输出应用举例 .....      | (33) |
| 本章小结 .....                   | (34) |
| 实验二 数据基本运算及基本输入、输出 .....     | (34) |
| 习题二 .....                    | (36) |
| <b>第3章 C语言程序的流程控制</b> .....  | (39) |
| 3.1 程序设计基础知识 .....           | (39) |
| 3.1.1 程序流程图 .....            | (39) |
| 3.1.2 程序设计方法 .....           | (39) |
| 3.2 基本C语句 .....              | (40) |
| 3.2.1 表达式语句 .....            | (41) |
| 3.2.2 复合语句 .....             | (41) |
| 3.2.3 流程控制语句 .....           | (42) |
| 3.3 关系运算及逻辑运算 .....          | (42) |
| 3.3.1 关系运算 .....             | (42) |
| 3.3.2 逻辑运算 .....             | (43) |
| 3.3.3 条件运算 .....             | (44) |
| 3.4 选择结构程序设计 .....           | (44) |
| 3.4.1 条件语句 .....             | (45) |
| 3.4.2 switch语句 .....         | (48) |
| 3.5 循环程序结构设计 .....           | (51) |
| 3.5.1 while语句 .....          | (52) |
| 3.5.2 do...while循环语句 .....   | (53) |
| 3.5.3 for循环语句 .....          | (54) |
| 3.5.4 循环嵌套 .....             | (55) |
| 3.5.5 continue和break语句 ..... | (56) |
| 3.6 循环程序举例 .....             | (57) |
| 3.6.1 三种循环的比较 .....          | (57) |
| 3.6.2 应用举例 .....             | (58) |
| 本章小结 .....                   | (61) |
| 实验三 分支与循环程序设计 .....          | (61) |
| 习题三 .....                    | (62) |
| <b>第4章 函数</b> .....          | (65) |
| 4.1 函数的定义及调用 .....           | (65) |
| 4.1.1 函数的定义 .....            | (66) |
| 4.1.2 函数声明 .....             | (68) |
| 4.1.3 函数调用 .....             | (69) |
| 4.2 函数的递归调用 .....            | (71) |
| 4.3 变量的存储属性 .....            | (72) |
| 4.3.1 局部变量与全局变量 .....        | (73) |
| 4.3.2 动态变量与静态变量 .....        | (76) |
| 4.3.3 寄存器变量 .....            | (78) |
| 4.4 C语言的编译预处理 .....          | (78) |
| 4.4.1 宏定义#define命令 .....     | (79) |
| 4.4.2 文件包含#include命令 .....   | (80) |

|                               |              |
|-------------------------------|--------------|
| 本章小结 .....                    | (81)         |
| 实验四 函数的定义和使用 .....            | (81)         |
| 习题四 .....                     | (82)         |
| <b>第 5 章 数组数据类型及其应用 .....</b> | <b>(85)</b>  |
| 5.1 一维数组 .....                | (85)         |
| 5.1.1 一维数组的定义 .....           | (85)         |
| 5.1.2 数组的初始化 .....            | (86)         |
| 5.1.3 数组元素的访问 .....           | (86)         |
| 5.1.4 数组作函数参数 .....           | (87)         |
| 5.2 二维数组 .....                | (89)         |
| 5.2.1 二维数组和多维数组的定义 .....      | (89)         |
| 5.2.2 二维和多维数组的初始化 .....       | (90)         |
| 5.2.3 二维数组的引用及举例 .....        | (91)         |
| 5.3 字符数组与字符串 .....            | (94)         |
| 5.3.1 字符串的处理 .....            | (94)         |
| 5.3.2 字符串的输入\输出 .....         | (95)         |
| 5.3.3 字符串的运算 .....            | (96)         |
| 5.3.4 二维字符数组及字符串的应用 .....     | (98)         |
| 5.3.5 字符串应用举例 .....           | (99)         |
| 5.4 线性表 .....                 | (100)        |
| 5.4.1 线性表的定义和基本运算 .....       | (100)        |
| 5.4.2 线性表的存储结构 .....          | (101)        |
| 5.4.3 线性表的运算 .....            | (101)        |
| 5.5 栈和队列 .....                | (103)        |
| 5.5.1 栈 .....                 | (104)        |
| 5.5.2 队列 .....                | (104)        |
| 5.6 稀疏矩阵 .....                | (106)        |
| 5.7 常用数据查找算法 .....            | (108)        |
| 5.7.1 顺序查找 .....              | (108)        |
| 5.7.2 折半查找 .....              | (109)        |
| 5.8 常用数据排序算法 .....            | (111)        |
| 5.8.1 基本概念 .....              | (111)        |
| 5.8.2 插入排序 .....              | (112)        |
| 5.8.3 交换排序 .....              | (114)        |
| 5.8.4 选择排序 .....              | (117)        |
| 本章小结 .....                    | (119)        |
| 实验五 数组及线性表 .....              | (120)        |
| 实验六 排序及查询 .....               | (121)        |
| 习题五 .....                     | (121)        |
| <b>第 6 章 指针 .....</b>         | <b>(123)</b> |
| 6.1 指针的概念 .....               | (123)        |
| 6.2 指针变量的定义和引用 .....          | (124)        |
| 6.2.1 指针变量的定义 .....           | (124)        |
| 6.2.2 指针变量的引用 .....           | (125)        |



|            |                          |       |
|------------|--------------------------|-------|
| 6.2.3      | 有关指针的两个运算符·····          | (126) |
| 6.2.4      | 指针变量应用举例·····            | (127) |
| 6.3        | 指针与数组·····               | (130) |
| 6.3.1      | 一维数组与指针·····             | (130) |
| 6.3.2      | 二维数组的指针表示法·····          | (134) |
| 6.4        | 指针与字符串·····              | (135) |
| 6.5        | 指针与函数·····               | (138) |
| 6.5.1      | 指针变量作为函数的参数·····         | (138) |
| 6.5.2      | 返回指针值的函数·····            | (143) |
| 6.5.3      | 指向函数的指针变量·····           | (144) |
| 本章小结       | ·····                    | (146) |
| 实验七        | 指针的使用·····               | (147) |
| 习题六        | ·····                    | (148) |
| <b>第7章</b> | <b>结构体与共用体数据类型</b> ····· | (149) |
| 7.1        | 结构体数据类型的定义和引用·····       | (149) |
| 7.1.1      | 结构体类型的定义·····            | (149) |
| 7.1.2      | 结构体类型变量的定义·····          | (150) |
| 7.1.3      | 结构体类型变量的引用·····          | (151) |
| 7.1.4      | 结构体类型变量的初始化·····         | (152) |
| 7.2        | 结构体数组及初始化·····           | (153) |
| 7.2.1      | 结构体数组的定义·····            | (153) |
| 7.2.2      | 结构体数组的初始化·····           | (154) |
| 7.3        | 结构体与指针·····              | (155) |
| 7.3.1      | 指向结构体变量的指针·····          | (155) |
| 7.3.2      | 指向结构体数组的指针·····          | (156) |
| 7.4        | 结构体与函数·····              | (158) |
| 7.5        | 线性链表·····                | (160) |
| 7.5.1      | 线性链表的概念·····             | (160) |
| 7.5.2      | 内存单元的动态分配·····           | (161) |
| 7.5.3      | 线性链表的初始化、建立和输出·····      | (163) |
| 7.5.4      | 线性链表的基本操作·····           | (166) |
| 7.6        | 循环链表·····                | (170) |
| 7.6.1      | 循环链表·····                | (170) |
| 7.6.2      | 双向循环链表·····              | (170) |
| 7.7        | 链栈和链队·····               | (173) |
| 7.7.1      | 链栈·····                  | (173) |
| 7.7.2      | 链队·····                  | (174) |
| 7.8        | 共用体数据类型·····             | (175) |
| 7.8.1      | 共用体类型的定义·····            | (175) |
| 7.8.2      | 共用体类型变量的引用·····          | (176) |
| 7.9        | 枚举类型·····                | (178) |
| 本章小结       | ·····                    | (180) |
| 实验八        | 链表的使用·····               | (181) |
| 习题七        | ·····                    | (181) |

|                        |       |
|------------------------|-------|
| <b>第 8 章 树</b>         | (184) |
| 8.1 树的基本概念             | (184) |
| 8.1.1 树的定义             | (184) |
| 8.1.2 树的基本术语           | (185) |
| 8.2 树的存储结构             | (186) |
| 8.2.1 双亲表示法            | (186) |
| 8.2.2 孩子表示法            | (187) |
| 8.2.3 孩子兄弟表示法          | (187) |
| 8.3 二叉树                | (188) |
| 8.3.1 二叉树的定义和性质        | (188) |
| 8.3.2 二叉树的存储结构         | (190) |
| 8.4 二叉树的遍历             | (192) |
| 8.4.1 二叉树的中根遍历         | (192) |
| 8.4.2 二叉树的先根遍历         | (193) |
| 8.4.3 二叉树的后根遍历         | (194) |
| 8.5 线索树                | (197) |
| 8.5.1 线索树的结构           | (197) |
| 8.5.2 中根线索树的建立         | (198) |
| 8.5.3 结点的检索            | (200) |
| 8.5.4 结点的插入            | (201) |
| 8.6 二叉查找树              | (203) |
| 8.6.1 二叉查找树的定义及其结构     | (204) |
| 8.6.2 二叉查找树的建立         | (204) |
| 8.6.3 二叉查找树的查找过程       | (206) |
| 8.6.4 二叉查找树的删除         | (207) |
| 8.6.5 二叉查找树的查找分析及评价    | (209) |
| 8.7 堆排序                | (210) |
| 8.7.1 堆定义              | (210) |
| 8.7.2 堆的建立             | (210) |
| 8.7.3 堆排序的实现           | (211) |
| 本章小结                   | (213) |
| 实验九 二叉树的遍历             | (214) |
| 习题八                    | (214) |
| <b>第 9 章 图</b>         | (216) |
| 9.1 基本概念               | (216) |
| 9.2 图的存储结构             | (218) |
| 9.2.1 邻接矩阵             | (219) |
| 9.2.2 邻接表              | (220) |
| 9.3 图的遍历               | (222) |
| 9.3.1 深度优先搜索(dfs)      | (223) |
| 9.3.2 广度优先搜索(bfs)      | (225) |
| 9.4 拓扑排序               | (228) |
| 9.5 最短路径               | (230) |
| 9.5.1 从某一源点到其他各顶点的最短路径 | (230) |

|                               |       |
|-------------------------------|-------|
| 9.5.2 每对顶点间的最短路径 .....        | (233) |
| 本章小结 .....                    | (235) |
| 习题九 .....                     | (235) |
| <b>第 10 章 文件</b> .....        | (237) |
| 10.1 C 文件概述 .....             | (237) |
| 10.1.1 流和文件 .....             | (237) |
| 10.1.2 缓冲区文件与非缓冲区文件 .....     | (237) |
| 10.2 文件的打开与关闭 .....           | (238) |
| 10.2.1 缓冲区文件定义 .....          | (238) |
| 10.2.2 文件的打开与关闭 .....         | (238) |
| 10.3 文件的输入与输出 .....           | (240) |
| 10.3.1 读写一个字符 .....           | (241) |
| 10.3.2 读写一个字符串 .....          | (243) |
| 10.3.3 读写数据字段 .....           | (244) |
| 10.3.4 格式化读写 .....            | (247) |
| 10.4 文件的定位与随机读写 .....         | (248) |
| 10.4.1 rewind()函数 .....       | (249) |
| 10.4.2 fseek()函数 .....        | (249) |
| 10.4.3 ftell()函数 .....        | (250) |
| 本章小结 .....                    | (250) |
| 实验十 文件操作 .....                | (251) |
| 习题十 .....                     | (251) |
| <b>第 11 章 C 语言的位运算</b> .....  | (253) |
| 11.1 位运算符 .....               | (253) |
| 11.1.1 按位与运算符(&) .....        | (253) |
| 11.1.2 按位或运算符( ) .....        | (254) |
| 11.1.3 按位异或运算符(^) .....       | (254) |
| 11.1.4 取反运算符(~) .....         | (254) |
| 11.1.5 左移运算符(<<) .....        | (254) |
| 11.1.6 右移运算符(>>) .....        | (255) |
| 11.2 位段 .....                 | (255) |
| 本章小结 .....                    | (258) |
| 习题十一 .....                    | (258) |
| <b>第 12 章 图形及用户界面技术</b> ..... | (259) |
| 12.1 设置图形模式 .....             | (259) |
| 12.1.1 几种工作模式及特点 .....        | (259) |
| 12.1.2 图形模式的初始化 .....         | (260) |
| 12.1.3 颜色的设置 .....            | (262) |
| 12.2 图形函数及图形设计 .....          | (264) |
| 12.2.1 画点 .....               | (264) |
| 12.2.2 画线 .....               | (265) |
| 12.2.3 画圆 .....               | (266) |
| 12.2.4 画组合图 .....             | (267) |
| 12.3 窗口设计 .....               | (268) |

|        |                         |       |
|--------|-------------------------|-------|
| 12.3.1 | 什么是窗口 .....             | (268) |
| 12.3.2 | 如何定义窗口 .....            | (268) |
| 12.3.3 | 文本窗口的颜色设置 .....         | (269) |
| 12.3.4 | 窗口的操作函数 .....           | (270) |
| 12.3.5 | 窗口设计示例 .....            | (271) |
| 12.4   | 光带选择菜单设计 .....          | (273) |
| 12.4.1 | 功能键识别 .....             | (273) |
| 12.4.2 | 光标控制 .....              | (274) |
| 12.4.3 | 利用标准输出函数设计菜单 .....      | (275) |
| 12.4.4 | 利用窗口操作函数设计菜单 .....      | (277) |
| 12.5   | 应用举例 .....              | (279) |
|        | 本章小结 .....              | (282) |
|        | 实验十一 综合应用程序设计 .....     | (283) |
|        | 习题十二 .....              | (283) |
| 附录 1   | TurboC 的安装及各选项的作用 ..... | (284) |
| 附录 2   | 错误信息英中文对照 .....         | (289) |
| 附录 3   | ASC II 代码与字符对照表 .....   | (291) |
| 附录 4   | 关键字和运算符 .....           | (293) |
| 附录 5   | 常用库函数 .....             | (294) |
|        | 参考文献 .....              | (299) |

# 第 1 章 概 述

自 1946 年第一台电子计算机问世以来,计算机技术发展之迅猛是令人难以想象的。计算机的应用也日益普及和深入,目前,计算机已广泛应用于社会的各个领域。而实际应用中,要使计算机发挥作用就离不开程序设计和程序设计语言,程序设计语言是人与计算机进行信息交流的工具,程序设计必须在一定的语言环境下进行。

C 语言是当今世界极为流行的高级语言之一,它既适合系统程序的设计,也适合应用程序的设计。

## 1.1 C 语言的发展和特点

在具体介绍 C 语言之前,我们先介绍一下有关程序设计语言的一些基本知识。

### 1.1.1 程序设计语言

若要使用计算机解决某个具体问题,必须先将该问题分解为一系列解题步骤,然后将这些解题步骤用计算机语言表示为一组指令,计算机执行这组指令使问题得到解决。这组解决实际问题的指令,就是通常所说的程序,描述这组指令的语言就是程序设计语言。

从计算机问世以来,计算机程序设计语言的发展经历了以下三个阶段:

#### 1. 机器语言

对于任何一台计算机,都有一套指令系统,如 8088 指令系统,80286 和 80386 指令系统等。每一套指令系统都由一组指令组成,每一条指令均由 0 或 1 的代码组成。例如某指令系统有如下两条指令:

10000000      表示两数相加

10010000      表示两数相减

某机器的指令系统就构成了该机器的机器语言,不同的机器系统有不同的机器语言。

对于用机器语言编写的程序,计算机可以直接理解直接执行。但是这种用 0、1 代码序列组成的程序,难记、难认、难理解,且不易查错,只有少数专业人员才能掌握,编写程序相当困难,程序效率低,质量难以保证。

#### 2. 汇编语言

随着计算机应用范围的逐渐扩大,在 50 年代中期,人们开始用一些“助记符”来代替 0、1 代码序列,例如上面两条指令可表示为:

10000000      ADD   A,B

10010000      SUB   A,B

这种助记符构成的指令系统,就称为符号语言或汇编语言。

用汇编语言编程,程序的效率和质量都有所提高,但是用汇编语言编写的程序,计算机无法直接执行,因此必须有一种工具,能将汇编语言程序翻译成前后意义等价的机器语言程序,这种工具就称为汇编程序。

汇编语言程序→汇编程序→机器语言程序

因此,用汇编语言编写程序必须有汇编程序支持。汇编语言和机器语言都依赖于计算机本身,因而它们都称为面向机器的语言。使用这两种语言都需要对计算机的内部结构有较深刻的了解,因而用它们编写程序不仅劳动强度很大而且程序的通用性差,推广使用这些语言存在很大的障碍。

### 3. 高级语言

计算机初期的使用范围仅限于科学计算。随着计算机技术的发展,其应用范围逐渐由数值计算领域拓宽到非数值计算领域,例如行政事务处理、工业控制等。这样,对程序(软件)的需求量越来越大,面向机器的语言编程已不能满足要求,到1954年出现了第一种高级语言——FORTRAN。高级语言不再是面向机器的,而是面向解题的过程,因而又称为算法语言。使用高级语言编程序时,人们不需要熟悉计算机内部的具体结构,因而可以将主要精力放到算法描述上。例如求A、B的最大值可描述为:

```
if a>b
    max=a;
else
    max=b;
```

用高级语言编写的程序,计算机也不能直接执行,必须有一个工具将它翻译为机器语言程序。这种将高级语言程序翻译为意义等价的机器语言程序的工具就是编译程序。

高级语言程序→编译程序→机器语言程序

近几年来,面向对象的程序设计语言(Object Oriented Language 简称OOL)得到了迅速发展和应用,使程序设计变得更加简练、自然。

#### 1.1.2 C语言简介

C语言是目前极为流行的一种高级语言,它既具有高级语言的功能,又具有机器语言的一些特性。它是1972年由美国BELL实验室的Dennis Ritchie和Brian Kernighan等人首先推出的。之后,C语言又经过不断改进,使其逐步完善。直到1978年Brian Kernighan和Dennis Ritchie合著了影响深远的名著《The C Programming Language》,首次向世人系统介绍了C语言。此后,C语言的发展非常迅速,各种版本的C语言相继涌现出来。由于没有统一的标准,使得这些C语言之间出现了一些不统一的地方,为了改变这种情况,美国国家标准化协会(ANSI)于1983年制定了一套标准,称为ANSI C(标准C)成为各种C语言版本的基础。

目前微机中使用的C语言版本有很多,比较精典的有Turbo C、Borland C、Microsoft C等。近年来,又推出了包含面向对象程序设计思想和方法的C++,它们均支持ANSI C,本书主要介绍ANSI C中的基础部分,同时,兼顾各种版本的通用性和一致性。

### 1.1.3 C 语言的特点

C 语言的特点是多方面的,从不同的角度可总结出众多的特点。归纳起来主要有以下几点:

(1) C 语言既具有高级语言的通用性及易写易读的特点,又具有汇编语言的“位处理”、“地址操作”等能力。这使得 C 语言不仅像 PASCAL、FORTRAN、BASIC 等高级语言那样用于应用软件的设计,还能像汇编语言那样用于计算机系统软件和控制软件的开发。

(2) C 语言是一种结构化程序设计语言。具有结构化的控制语句,用函数作为程序模块以实现程序的模块化。

(3) 语句简练、紧凑,语法规则少,数据类型丰富,编译后生成的代码质量高,运行速度快。

(4) 可移植性好。用 C 语言编写的程序可以从一种环境不加或稍加改动就可以转到另一种环境中运行。

(5) C 语言功能丰富。它不仅提供了丰富的运算符,还提供了各种功能强大的系统函数。

尽管 C 语言有很多优点,但也存在一些缺点和不足。比如它的类型检验弱,转换比较随便,优先级太多,不便记忆等。这些都对程序设计者提出了更高的要求,也给初学者增加了困难。

## 1.2 数据结构概述

随着计算机应用的日益深入,计算机加工处理的对象由纯粹的数值发展到字符、表格、图像和声音等各种具有一定结构的数据,这就给程序设计带来一些新的问题。为了编写一个好的程序,必须分析待处理的数据的特性以及各数据之间存在的关系。这就是“数据结构”这门学科形成发展的背景。

在当今的程序设计中,人们对数据结构越来越重视,认为程序设计的本质就是对要进行处理的问题选择一种好的数据结构,同时在此结构上施加一种好的算法。著名的计算机科学家沃思(N. Wirth)提出的

$$\text{算法} + \text{数据结构} = \text{程序}$$

正是体现了这一观点。

其中,算法是对解决给定问题的有穷操作步骤的一种描述,它指明了问题的解题过程和步骤。

数据结构则是对程序中所处理数据的特性以及各数据之间的关系加以分析研究,并将这些数据进行合理的存储。

本节只对数据结构的基本概念和基本内容作一扼要的介绍。

### 1.2.1 数据结构的基本概念和术语

#### 1. 数据(Data)

数据是能输入到计算机中并能被计算机处理的一切对象。它是对客观事物的符号表示。这

里必须强调指出,所谓数据绝不能仅理解为整数或实数这种狭义的“数”,必须作广义的理解。例如,一个用某种程序语言编写的程序、一篇文章、一张地图以及图画、声音等等都可视为“数据”。今后随着计算机的发展,还会不断扩充数据的范围。

## 2. 数据元素(Data Element)

数据元素是数据处理的基本单位,在计算机程序中要作为一个整体进行考虑和处理。由于数据范围非常广泛,因此基本单位可大可小,小到可以是一个字符,大到可以是一个国家的地图或一本书等等。有时,一个数据元素可由若干个“数据项”(Data Item)组成,这种由多个数据项组成的数据元素,通常又称为记录(Record)。如一本书的目录卡片作为一个数据元素,而书目信息中的每一项如:书名、作者名、出版社名、出版日期等为数据项。

## 3. 数据结构(Data Structure)

数据结构是彼此具有一定关系的数据元素的集合。这些关系反映了客观事物之间的联系。由于客观事物之间存在着各种不同的联系形式,所以反映在数据关系上也就各不相同。根据数据之间关系的不同特性,常用的数据结构有以下三种。

(1)线性结构:结构中的数据元素之间存在一个对一个的关系。比如一个班的学生名单是一个接着一个排列的,一个名字唯一地对应一名学生。

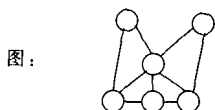
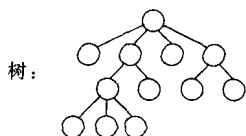
(2)树形结构:结构中的数据元素之间存在一个对多个的关系。如上级单位与多个下级单位的关系、父辈与子辈的关系、全体与部分的关系等,就需要用所谓的“树结构”来表示。

(3)图状或网状结构:结构中的数据元素之间存在多个对多个的关系。如各城市之间的交通联系或电讯联系、一个大的工程项目的施工进度等,就要用复杂的“图结构”来表示。

其中后两种结构又称为非线性数据结构。

线性: ○—○—○—○—○

如图 1.1 所示为上述各结构的关系图,其中一个○表示一个数据元素。



从另一个角度讲,数据结构又分为逻辑结构和物理结构两个方面。逻辑结构是指数据元素之间存在的相互关系,如上面所举的线性结构、树结构、图结构等。物理结构又称存储结构,是数据结构在计算机中的表示(又称映像),它包括数据元素的表示和关系的表示。同一种逻辑结构可以有几种不同的物理结构来实现它。在程序设计中,研究物理结构更为重要,因为对于同一问题,数据的存储结构不同,解决问题的方法就有所不同。

图 1.1 基本结构关系图

至此,我们可以对数据结构的作用简单概括如下:

数据结构研究数据的逻辑结构和物理结构,并在这种结构上定义相关的运算,设计实现这些运算的算法,分析算法的效率。

### 1.2.2 关于算法分析

解决一个问题(例如对  $N$  个数进行排序)可以有多种算法,那么该怎样判断它们的优劣呢?当然判断算法优劣的标准很多,本书不做深入讨论。这里主要讨论的是:一个算法除了正确性必须保证以外(一个有错误的算法根本不能称为算法!),一个主要指标是它的效率。



算法的效率包括时间与空间两个方面,分别称为时间复杂度与空间复杂度。所谓时间复杂度是指一个算法执行时所需运算次数的多少;空间复杂度是指一个算法执行时所需的辅助内存空间的大小。这两种度量都不是表示为绝对的量,而是用量级的概念来表示,并且在对一个算法进行分析时主要考虑其时间效率。这种量级通常用大写字母“O”来表示。如: $O(1)$ 、 $O(n)$ 、 $O(n^2)$ 、 $O(2^n)$ 、 $O(\log n)$  分别表示常量阶、线性阶、平方阶、指数阶、对数阶。显然对时间复杂度为  $O(1)$  的算法执行时间最省,则对指数阶的算法,当  $n$  较大时其增长率达到惊人的程度。因此我们应避免使用那些复杂度达到指数阶的算法。

例如有一算法的程序如下:

```
for (m=0 ; m<= n ; m++)
    for (j=0; j<=n; j++)
        { r[m][n]=0;
          for (k=0 ; k<=n; k++)
              r[m][j]=r[m][j]+a[m][k] * b[k][j] ;
        }
```

根据程序的执行,该算法的时间复杂度量级为  $O(n^3)$ 。

### 1.3 C 语言程序结构及特点

每种程序设计语言都有其特定的语法要求和程序结构风格,本节通过几个 C 语言书写的简单程序,使大家对 C 语言程序的结构有一个总体的认识。

#### 1.3.1 简单程序实例

【例 1-1】简单程序示例

```
main( )
{ printf("I learn C language.");
}
```

本程序执行后将在屏幕上显示一行信息:

I learn C language.

程序中 `main()` 表示“主函数”,是每个 C 程序所必须的。`{ }` 括起的内容称为函数体。`printf` 是 C 语言的输出函数,双引号内的字符串被原样输出。

【例 1-2】以下 C 程序可以从键盘上接收三个整数,然后求出它们的平均值并输出。

```
float average(int x,int y,int z);/* 求三个数的平均 */
{ float aver;
  aver=(x+y+z)/3.0;
  return(aver);
}
```