

高等学校公共课计算机教材系列

C语言 程序设计 (第2版)

张继生 白秋颖 主编 张静 唐笑非 副主编



清华大学出版社

内 容 简 介

本书详实地讲解了 C 语言的基本概念、原理和使用方法,力求给读者打下一个扎实的程序设计基础,培养读者程序设计的能力。本书主要内容包括 C 语言程序设计基础知识、基本数据类型及运算符、C 语言的控制结构、数组、函数、指针、构造数据类型、编译预处理、文件等。本书采用循序渐进、深入浅出、通俗易懂的讲解方法,本着理论与实际相结合的原则,通过大量经典实例对 C 语言知识进行了重点讲解,使程序设计语言的初学者能够掌握利用 C 语言进行结构化程序设计的技术和方法。

本书以 C 语言编程基本技能训练为主线,突出基本技能的培养,内容完整,阐述准确,层次清楚。通过本书的学习,读者可以牢固掌握程序设计的基本技能,以适应信息时代对读者的科学素质的要求。

本书适用于高等学校各专业的程序设计基础教学,尤其适合应用型本科、高职院校的计算机及非计算机相关专业的学生使用,同时也是一本实用的计算机等级考试备考辅导书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/张继生,白秋颖主编. —2 版. —北京:清华大学出版社,2011.9
(高等学校公共课计算机教材系列)

ISBN 978-7-302-26250-3

I. ①C… II. ①张… ②白… III. ①C 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 137743 号

责任编辑:索 梅 薛 阳

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:14.5 字 数:362 千字

版 次:2011 年 9 月第 2 版 印 次:2011 年 9 月第 1 次印刷

印 数:1~4000

定 价:25.00 元

产品编号:043121-01

出版说明

随着计算机技术的普及及其向其他学科的快速渗透,非计算机专业的学生的计算机知识已普遍不能适应当今的形势,这在就业及进入新的工作方面,就更加突出。而非计算机专业的学生选修计算机专业的课程,并不符合其以应用为主、培养复合型创新人才的教学目标。目前在本科教育中有不少高校建立了以素质教育为取向的跨学科公共课体系,开设了本科生公(通)选课程,以拓宽学生的知识基础,培养不断学习和创造知识的能力和素质,以便在就业与新的工作岗位上取得更大的优势。许多高校在教学体系建设中已将计算机教学纳入基础课的范畴,在非计算机专业教学和教材改革方面也做了大量工作,积累了许多宝贵经验,起到了教学示范作用。将他们的教研成果转化为教材的形式,向全国其他学校推广,对于深化我国高等学校的教学改革具有十分重要的意义。

2005年1月,在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出:“要科学制订人才培养目标和规格标准,把加强基础与强调适应性有机结合,着力培养基础扎实、知识面宽、能力强、素质高的人才,更加注重学生能力培养。要继续推进课程体系、教学内容、教学方法和手段的改革,构建新的课程结构,加大选修课程开设比例,积极推进弹性学习制度建设。”然而,目前明确定位于非计算机专业以应用为主这一教学目标的教材十分缺乏,使得一些公共课不得不选用计算机专业教材或非教材的店销图书及讲义来替代,在这种背景下,出版一套符合目前非计算机专业学习、拓宽计算机及相关领域知识的适用教材以填补这一空白,推进、配合高校新的教改需求,十分必要。清华大学出版社在对计算机基础教学现状进行了广泛的调查研究的基础上,决定组织编写一套《高等学校公共课计算机教材系列》。

本系列教材将延续并反映清华版教材二十年来形成的技术准确、内容严谨的风格,并具有以下特点:

1. 目的明确

本系列教材针对当前高等教育改革的新形势,以社会对人才的需求为导向,以重点院校已开设的公共课程为基础,同时也吸收一般院校的优秀公共课教材,广泛吸纳全国各高等学校的优秀教师参与编写,从中精选出版确实反映非计算机专业计算机教学方向的特色教材,以配套各高校开设公选课程。

2. 面向就业,突出应用

本系列教材力求突出各学科对计算机知识应用的特征,在知识结构上强调应用能

力和创新能力,以使学生能比较熟练地应用计算机知识解决实际问题,满足就业单位的需求。

3. 结合教育与学科发展的需求,动态更新

本系列教材将根据计算机学科的发展和各专业的需要进行更新,同时教材的出版载体形式也会随计算机、网络 and 多媒体技术的发展而变化,以体现教学方法和教学手段的更新。

4. 注重服务

本系列教材都将力求配套能用于网上下载的教学课件及辅助教学资源。

由于各个学校办学特色有所不同,对教材的要求也会呈现自己的特点,我们希望大家在使用教材的过程中,及时给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

清华大学出版社

联系人: 郑寅堃 zhengyk@tup.tsinghua.edu.cn

前言

C 语言是一种在国际上广泛流行的计算机程序设计语言,它具有表达能力强、概念和功能丰富、目标程序质量高、可移植性好、使用灵活方便等特点,既具有高级语言的优点,又具有低级语言的某些特点,能够有效地用来编制各种系统软件和应用软件。同时,C 语言的控制结构简明清晰,是非常适合进行结构化程序设计的一种计算机语言。因此,目前国内大部分高等院校都把 C 语言作为计算机和非计算机相关专业的一门程序设计语言课程。

C 语言涉及的概念多、规则复杂,容易出错,初学者学起来往往觉得困难。本书在详细阐述程序设计语言的基本概念、原理和方法的基础上,采用循序渐进、深入浅出、通俗易懂的讲解方法,本着理论与实际相结合的原则,通过大量经典实例重点讲解了 C 语言的概念、规则和使用方法,使程序设计语言的初学者能够在建立正确程序设计理念的前提下,掌握利用 C 语言进行结构化程序设计的技术和方法。全书共 9 章,主要内容包括:第 1 章 C 语言概述、第 2 章数据描述与基本操作、第 3 章 C 语言的控制结构、第 4 章数组、第 5 章函数、第 6 章指针、第 7 章构造数据类型、第 8 章编译预处理、第 9 章文件。书中对数组、函数、指针、变量的存储类型、结构体和共用体、文件等重点和难点的内容进行了深入讲解和分析。“C 语言程序设计”课程作为程序设计的入门课程,重视对程序设计和 C 语言基本概念、原理和规则的讲解,力求给读者打下一个扎实的基础,培养读者良好的编程风格,提高读者进一步学习新程序设计语言的能力。

本书可作为高等学校各专业程序设计基础教学的教材,尤其适合应用型本科、高职院校的计算机及非计算机专业的学生使用,同时也可作为编程人员和 C 语言自学者的参考用书。

本书第 1~3 章由白秋颖编写;第 4、5、9 章由张继生编写;第 6 章由唐笑非编写;第 7、8 章由张静编写。在本书的编写过程中,刘尚懿、艾青也做了许多工作,在此一并表示感谢!最后还要感谢为本书付出心血的编辑、审稿人员等各位朋友!

为了帮助读者学习,每章设有小结和习题,同时本书有配套的实验用书《C 语言程序设计(第 2 版)上机指导与习题解答》(清华大学出版社,2011 年版),其中重点介绍了 Visual C++ 6.0 编译系统的使用方法,使读者在实践学习过程中能迅速掌握 C 语言程序的编辑、编译、调试和运行方法。

由于编者水平有限,书中难免存在一些缺点和错误,殷切希望广大读者批评指正。

编 者

2011 年 7 月

目录

CONTENTS

第 1 章 C 语言概述	1
1.1 C 语言的发展与特点	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	2
1.2 程序设计基础	3
1.2.1 程序	3
1.2.2 程序设计	3
1.2.3 算法	4
1.2.4 数据结构	9
1.3 C 语言程序的结构	9
1.4 C 语言程序的开发与环境	11
1.4.1 C 语言程序的开发	11
1.4.2 C 语言程序的开发环境	12
本章小结	14
习题 1	15
第 2 章 数据描述与基本操作	16
2.1 C 语言的数据类型	16
2.2 常量与变量	17
2.2.1 常量和变量的概念	17
2.2.2 整型数据	18
2.2.3 浮点型数据	22
2.2.4 字符型数据	24
2.2.5 变量的初始化	26
2.3 运算符与表达式	27
2.3.1 算术运算符和算术表达式	27
2.3.2 赋值运算符和赋值表达式	29
2.3.3 关系运算符与关系表达式	31
2.3.4 逻辑运算符与逻辑表达式	32

2.3.5 逗号运算符和逗号表达式	34
2.4 位运算	34
2.4.1 按位与运算	34
2.4.2 按位或运算	35
2.4.3 按位异或运算	35
2.4.4 求反运算	36
2.4.5 左移运算	36
2.4.6 右移运算	36
2.4.7 位域(位段)	37
2.5 输入和输出函数	39
2.5.1 字符型数据的输入和输出	40
2.5.2 格式输入与输出函数	41
本章小结	47
习题2	47
第3章 C语言的控制结构	49
3.1 结构化程序设计	49
3.1.1 程序的基本结构	49
3.1.2 结构化程序设计的特点	50
3.1.3 结构化程序设计的方法	50
3.1.4 结构化程序设计的步骤	51
3.2 顺序结构程序设计	52
3.3 选择结构程序设计	56
3.3.1 if 语句	56
3.3.2 多分支(多情况)选择结构	61
3.3.3 条件运算符和条件表达式	63
3.4 循环结构程序设计	64
3.4.1 while 语句	65
3.4.2 do-while 语句	67
3.4.3 for 语句	68
3.4.4 多重循环	71
3.4.5 几种循环的比较	71
3.4.6 break 语句	72
3.4.7 continue 语句	72
3.4.8 程序举例	73
本章小结	76
习题3	76

第4章 数组	78
4.1 一维数组的定义和引用	78
4.1.1 一维数组的定义	78
4.1.2 一维数组的初始化	79
4.1.3 一维数组元素的引用	80
4.1.4 一维数组的应用举例	81
4.2 二维数组的定义和引用	86
4.2.1 二维数组的定义	86
4.2.2 二维数组的初始化	87
4.2.3 二维数组元素的引用	88
4.2.4 二维数组的应用举例	89
4.3 字符数组的定义和引用	91
4.3.1 字符数组的定义	91
4.3.2 字符数组的初始化	92
4.3.3 字符数组元素的引用	92
4.3.4 字符串与字符数组	94
4.3.5 字符数组的输入与输出	95
4.3.6 字符串处理函数	98
4.3.7 字符数组的应用举例	101
本章小结	107
习题4	108
第5章 函数	109
5.1 模块化程序设计与函数	109
5.1.1 模块化程序设计原则	109
5.1.2 模块与函数	110
5.1.3 C源程序的结构	110
5.2 函数的定义	111
5.2.1 标准库函数	111
5.2.2 函数的定义形式	112
5.2.3 函数的参数	113
5.3 函数的调用与返回值	114
5.3.1 函数的调用	115
5.3.2 函数的返回值	118
5.3.3 函数的声明	119
5.4 函数的递归调用	120
5.5 数组作为函数参数	122
5.5.1 数组元素作实参	122

5.5.2 一维数组名作函数参数	123
5.5.3 用多维数组名作函数参数	125
5.6 变量的作用域与存储属性	126
5.6.1 局部变量与全局变量	127
5.6.2 变量的存储属性	130
5.6.3 局部变量的存储类型	130
5.6.4 全局变量的存储类型	133
5.7 内部函数和外部函数	134
5.7.1 内部函数	134
5.7.2 外部函数	135
本章小结	135
习题 5	135
第 6 章 指针	137
6.1 指针的概念	137
6.2 指针变量的定义	138
6.3 指针的运算	139
6.3.1 指针变量的赋值运算	139
6.3.2 指针变量的算术运算	141
6.4 指针与数组	144
6.4.1 指向一维数组的指针	144
6.4.2 指向二维数组的指针	147
6.4.3 指向字符串的指针	150
6.5 指针与函数	152
6.5.1 指针变量作为函数参数	152
6.5.2 数组指针作为函数参数	154
6.5.3 指针作为函数的返回值	158
6.5.4 指向函数的指针变量	159
6.6 指针数组与指向指针的指针	161
6.6.1 指向指针的指针	163
6.6.2 main 函数的参数	163
本章小结	164
习题 6	165
第 7 章 构造数据类型	166
7.1 结构体数据类型	166
7.1.1 结构体类型的定义	166
7.1.2 结构体类型变量的定义	167
7.1.3 结构体变量的初始化	169

7.1.4 结构体变量成员的引用	170
7.2 结构体数组	172
7.2.1 结构体数组的定义	172
7.2.2 结构体数组的初始化	172
7.2.3 结构体数组的引用	173
7.3 结构体指针	174
7.3.1 指向结构体变量的指针	174
7.3.2 指向结构体数组的指针	176
7.3.3 结构体指针变量作为函数参数	177
7.3.4 结构体指针变量作为函数返回值	178
7.4 链表	179
7.4.1 动态存储分配	179
7.4.2 链表的操作	180
7.5 共用体数据类型	186
7.6 枚举类型	188
7.7 类型定义符 typedef	189
本章小结	190
习题 7	191
第 8 章 编译预处理	192
8.1 宏定义	192
8.1.1 不带参数的宏定义	192
8.1.2 带参数的宏定义	195
8.2 文件包含	198
8.3 条件编译	199
本章小结	200
习题 8	201
第 9 章 文件	202
9.1 C 文件概述	202
9.1.1 C 文件的分类	202
9.1.2 文件指针	204
9.2 文件的打开与关闭	204
9.2.1 文件的打开	205
9.2.2 文件的关闭	206
9.3 文件的读写	207
9.3.1 字符读写函数	207
9.3.2 字符串读写函数	210
9.3.3 数据块读写函数	211

9.3.4 格式化读写函数.....	214
9.4 文件定位函数	214
9.4.1 重置文件指针函数.....	214
9.4.2 设置指针位置函数.....	215
9.4.3 取指针位置函数.....	215
9.5 文件出错检测函数	216
9.5.1 读写出错检测函数.....	216
9.5.2 清除文件出错标志函数.....	216
9.5.3 清除文件函数.....	216
本章小结.....	217
习题 9	218
参考文献.....	219

第 1 章

C语言概述

C语言是国际上广泛流行的高级程序设计语言之一,它具有语言简洁、使用方便灵活、移植性好、能直接对系统硬件和外围接口进行控制等特点。作为系统描述语言,C语言既可以用来编写系统软件,也可以用来编写应用软件,集汇编语言和高级语言的优点于一身。本章将简要介绍C语言的发展和特点、程序设计基础知识、C语言程序结构及C语言程序的运行环境。

本章要点

- 了解C语言的发展和特点。
- 了解程序设计基础知识。
- 掌握C语言程序的结构。
- 熟悉C语言的开发环境。

1.1 C语言的发展与特点

1.1.1 C语言的发展

C语言的原型是Algol 60语言。1963年,剑桥大学将Algol 60语言发展成为CPL(Combined Programming Language)语言。1967年,剑桥大学的Martin Richards对CPL语言进行了简化,于是产生了BCPL语言。1970年,美国贝尔实验室的Ken Thompson对BCPL语言进行了修改,并为它起了一个有趣的名字B语言,意思是将CPL语言中的精华提炼出来。他用B语言编写了第一个UNIX操作系统。1973年,美国贝尔实验室的Dennis M. Ritchie在B语言的基础上设计出了一种新的语言,他用BCPL的第二个字母作为这种语言的名字,即C语言。

为了推广UNIX操作系统,1977年Dennis M. Ritchie发表了不依赖于具体机器系统的C语言编译文本《可移植C语言编译程序》。

1978年Brian W. Kernighan和Dennis M. Ritchie出版了著名的*The C Programming Language*,从而使C语言成为目前世界上流行最广泛的高级程序设计语言。

随着微型计算机的日益普及,社会上出现了许多C语言版本。由于没有统一的标准,这些C语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)于1983年成立了专门的定义C语言标准的委员会,该委员会花了6年时间使C语

言迈向标准化。随着C语言被广泛关注与应用,ANSI C标准于1989年被采用,该标准一般被称为ANSI/ISO Standard C,成为现行的C语言标准,而且成为最受欢迎的语言之一。

到了1995年,在ANSI C的基础上增加了一些库函数,出现了初步的C++语言。C++语言进一步扩充和完善了C语言,成为一种面向对象的程序设计语言。C++目前比较流行的版本是Microsoft Visual C++ 6.0。Microsoft Visual C++ 6.0提出了一些更为深入的概念,它所支持的面向对象的概念容易将问题空间直接映射到程序空间,为程序员提供了一种与传统结构化程序设计语言不同的思维方式和编程方法,因而也增加了整个语言的复杂性。

C语言是C++语言的基础,C++语言兼容了C语言的主体功能。本书是以Microsoft Visual C++ 6.0系统为平台来介绍C语言程序设计的,所有的C语言程序都基于Microsoft Visual C++ 6.0环境下运行。

1.1.2 C语言的特点

C语言是一种极具生命力的语言,它具有很多方面的特点。一般可归纳如下:

(1) 语言简洁、紧凑,使用方便、灵活。共有32个关键字、9种控制语句,程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分。

(2) 具有结构化控制语句(如if-else语句、while语句、do-while语句);用函数作为程序的模块单位,便于实现程序的模块化,且函数之间调用灵活、方便。C语言是完全模块化和结构化的语言。

(3) 运算符丰富。有34种运算符和15个等级的运算优先顺序,使表达式类型多样化;灵活使用运算符,可以实现在其他高级语言中难以实现的运算。

(4) 数据类型丰富。提供的数据类型有整型、浮点型、字符型、数组类型、指针类型、结构体类型及共用体类型等,能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。

(5) 比较接近硬件。允许直接访问物理地址,能进行位操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C语言即具有高级语言的功能,又具有低级语言的许多功能,可以用来编写系统软件。C语言的这种双重性,使它既是成功的系统描述语言,又是通用的程序设计语言。因为C语言程序也要通过编译、连接才可以执行,所以仍习惯将C语言称为高级语言。

(6) 语法限制少,程序设计自由度大。允许程序编写者有较大的自由度,放宽了以往高级语言严格的语法检查,较好地处理了“限制”与“灵活”这一对矛盾。

(7) 生成目标代码质量高,程序执行效率高。一般只比汇编程序生成的目标代码效率低10%~20%。

(8) 可移植性好。基本上不做修改就能应用于各种型号的计算机和各种操作系统。

综上所述,可以看出C语言是非常重要的程序设计语言,在软件行业有着广泛的应用。虽然,目前面向对象的各类高级语言盛行,但因为C语言是面向对象程序设计语言Visual C++的基础,所以C语言仍然是非常重要的一门课程,学习C语言之后,进一步学习Visual C++语言就会非常轻松。

1.2 程序设计基础

计算机技术的发展,为科学计算与数据处理提供了高速度、高精度的计算工具。但计算机在本质上只能机械地执行人输入的命令,它本身不会主动地进行思维,也不可能发挥任何创造性。因此,在利用计算机解决问题时,要进行程序设计——用计算机语言描述出解决问题的方法。

1.2.1 程序

计算机是一种具有内部存储能力的自动、高效的电子设备,它最本质的使命就是执行指令所规定的操作。如果我们需要计算机完成什么工作,只要将其步骤用若干条指令的形式描述出来,并把这些指令存放在计算机的内部存储器中,需要结果时就向计算机发出一个简单的命令,计算机就会自动地逐条顺序执行指令,全部指令执行完就得到了预期的结果。这种可以被连续执行的一条条指令的集合称为计算机的程序。也就是说,程序是计算机的指令序列,编制程序的工作就是为计算机安排指令序列。

但是,计算机指令是二进制编码,用它编制程序既难记忆,又难掌握,所以,计算机工作者就研制出了各种计算机能够懂得、人们又方便使用的计算机语言,程序就是用计算机语言来编写的。因此,计算机语言通常被称为程序语言,一个计算机程序总是用某种程序语言书写的。

1.2.2 程序设计

什么是程序设计呢?在日常生活中可以看到,同一台计算机,有时可以画图,有时可以制表,有时可以玩游戏,诸如此类,不一而足。也就是说,尽管计算机本身只是以一种现代化方式批量生产出来的通用机器,但是,使用不同的程序,计算机就可以处理不同的问题。计算机之所以能够产生如此大的影响,其原因不仅在于人们发明了机器本身,更重要的是人们为计算机开发出了不计其数的能够指挥计算机完成各种各样工作的程序。这些功能丰富的程序给了计算机无尽的生命力,而它们正是程序设计工作的结晶。程序设计是用某种程序语言编写程序的过程。

更确切地说,所谓程序,是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述,而这个描述的过程就称为程序设计。对数据的描述就是数据的结构形式;对处理方法和步骤的描述就是算法问题。因而,数据结构与算法是程序设计过程中密切相关的两个方面。发明 Pascal 语言的著名计算机科学家 Niklaus Wirth 教授提出了关于程序的著名公式:程序+数据结构=算法。这个公式指明了程序设计的主要任务。本书是介绍程序设计语言之一——C 语言的,有关数据结构部分不做介绍,关于算法的问题在下一节给出初步的介绍。

对于程序设计的初学者来说,首先要学会设计一个正确的程序。一个正确的程序,通常

包括两个含义：一是书写正确，二是结果正确。书写正确是指程序在语法上正确，符合程序语言的规则；而结果正确通常是指对应于正确的输入，程序能产生所期望的输出，符合使用者对程序功能的要求。程序设计的基本目标是编制出正确的程序和高质量的程序。所谓高质量是指程序具有良好的结构、可读性好、可靠性高、便于维护等一系列特点。

那么，如何进行程序设计呢？一个设计简单的程序一般包含以下4个步骤：

(1) 分析问题，建立数学模型。使用计算机解决具体问题时，首先要对问题进行充分的分析，确定问题是什么，解决问题的步骤又是什么。针对所要解决的问题，找出已知的数据和条件，确定所需的输入、处理及输出对象。将解题过程归纳为一系列的数学表达式，建立起各种量之间的关系，即建立起解决问题的数学模型。有许多问题的数学模型很显然或者很简单，以至于设计者感觉到不需要模型。但是有更多复杂的问题需要靠分析问题来构造数学模型，模型的好坏、对与错，在很大程度上决定了程序的正确性和复杂程度。

(2) 确定数据结构和算法。根据建立的数学模型，对指定的输入数据和预期的输出结果，确定存放数据的数据结构。针对所建立的数学模型和确定的数据结构，选择合适的算法加以实现。注意，这里所说的“算法”泛指解决某一问题的方法和步骤，而不仅仅是指“计算”。

(3) 编制程序。根据确定的数据结构和算法，用程序设计语言把这个解决方案严格地描述出来，也就是编写出程序代码。

(4) 调试程序。在计算机上用实际的输入数据对编好的程序进行调试，分析所得到的运行结果，进行程序的测试和调整，直至获得预期的结果。

由此可见，一个完整的程序要涉及4方面的问题：数据结构、算法、编程语言和程序设计方法。这4个方面的知识都是程序设计人员所必须具备的，其中算法是至关重要的一个方面。关于数据结构和算法问题有专门的著作，本书的重点是介绍编程语言和程序设计方法。但是，如果程序设计人员对算法一无所知，就无法进行基本的程序设计。因此，下面对算法的基本概念、基本设计和表示方法作初步介绍，目的是使初学者了解程序设计是如何开始的。

1.2.3 算法

1. 算法的基本概念

什么是算法？当代著名计算机科学家 D. E. Knuth 在他撰写的 *THE ART OF COMPUTER PROGRAMMING* 一书中写到：“一个算法，就是一个有穷规则的集合，其中之规则规定了一个解决某一特定类型的问题的运算序列”。简单地说，任何解决问题的过程都是由一定的步骤组成的，解决问题的确定的方法和有限的步骤被称为算法。

不是只有计算问题才有算法，做任何事情都有一定的步骤。例如，你想去北京开会，首先要买火车票，然后按照车票标示的时间地点坐火车到达目的地，参加会议；新生开学报到，要凭录取通知书到指定学校报到注册。我们从事的任何活动，都必须按一定的步骤进行，才能顺利完成。

通常计算机算法分为两大类：数值运算算法和非数值运算算法。数值运算是指对问题

求数值解,例如对微分方程求解、对函数的定积分求解等,都属于数值运算范围。非数值运算包括的领域非常广泛,例如资料检索、事务管理、数据处理等。数值运算有确定的数学模型,一般都有比较成熟的算法。许多常用算法通常还会被编写成通用程序并汇编成各种程序库的形式,供用户在不同需要时直接调用,例如数学程序库、数学软件包等。而非数值运算的种类繁多,要求不一,很难提供统一的算法。

下面通过3个简单的问题说明设计算法的思维方法。

【例 1-1】 有黑和蓝两个墨水瓶,但却错把黑墨水装在了蓝墨水瓶子里,而把蓝墨水装在了黑墨水瓶子里,要求将其互换。

算法分析:这是一个非数值运算问题。因为两个瓶子的墨水不能直接交换,所以,解决这一问题的关键是需要引入第3个墨水瓶。

设第3个墨水瓶为白色,其交换步骤如下:

- (1) 将黑瓶中的蓝墨水装入白瓶中;
- (2) 将蓝瓶中的黑墨水装入黑瓶中;
- (3) 将白瓶中的蓝墨水装入蓝瓶中;
- (4) 交换结束。

【例 1-2】 计算函数 $M(x)$ 的值。函数 $M(x)$ 为: $M(x) = \begin{cases} x^2 + 1 (x \geq 0) \\ x^2 - 1 (x < 0) \end{cases}$ 。

算法分析:本题是一个数值运算问题。其中 M 代表要计算的函数值,有两个不同的表达式,根据 x 的取值决定采用哪一个算式。

根据计算机具有进行逻辑判断的基本功能,用计算机解题的算法如下:

- (1) 将 x 的值输入到计算机;
- (2) 判断 $x \geq 0$ 是否成立,如果条件成立,执行第(3)步,否则执行第(4)步;
- (3) 按表达式 $x^2 + 1$ 计算出结果存放到 M 中,然后执行第(5)步;
- (4) 按表达式 $x^2 - 1$ 计算出结果存放到 M 中,然后执行第(5)步;
- (5) 输出 M 的值;
- (6) 算法结束。

【例 1-3】 给定两个正整数 m 和 $n (m \geq n)$,求它们的最大公约数。

算法分析:这也是一个数值运算问题,它有成熟的算法,我国数学家秦九韶在《算术九章》一书中曾记载了这个算法。求最大公约数的问题一般用辗转相除法(也称欧几里得算法)求解。

例如:设 m 为 35, n 为 15,余数用 r 表示。它们的最大公约数的求法如下:

35 除以 15 商 2 余数为 5,以 n 作 m ,以 r 作 n ,继续相除;15 除以 5 商 3 余数为 0。当余数为 0 时,所得 n 即为两数的最大公约数。所以 35 和 15 的最大公约数为 5。

用这种方法求两数的最大公约数,其算法可以描述如下:

- (1) 将两个正整数存放到变量 m 和 n 中;
- (2) 求余数:计算 m 除以 n ,将所得余数存放到变量 r 中;
- (3) 判断余数是否为 0:若余数为 0 则执行第(5)步,否则执行第(4)步;
- (4) 更新被除数和余数:将 n 的值存放到 m 中,将 r 的值存放到 n 中,并转向第(2)步继续循环执行;

(5) 输出 n 的当前值,算法结束。

由上述 3 个简单的例子可以看出,一个算法由若干操作步骤构成,并且这些操作是按一定的控制结构所规定的次序执行的。例 1-1 是顺序执行的,称为顺序结构。而在例 1-2 中,则不是所有步骤都执行。如第(3)步和第(4)步的两个操作就不能同时被执行,需要根据条件判断决定执行哪个操作,这种结构称为分支结构。例 1-3 不仅包含了判断,而且需要重复执行。如第(2)步到第(4)步之间的操作就需要根据条件判断是否重复执行,并且一直延续到条件“余数为 0”为止,这种具有重复执行功能的结构称为循环结构。

2. 算法的两要素

由上述 3 个例子可以看出,任何简单或复杂的算法都是由基本功能操作和控制结构这两个要素组成。任何类型计算机的算法最基本的功能操作都是一致的,包括以下 4 个方面:

- (1) 逻辑运算:与、或、非;
- (2) 算术运算:加、减、乘、除;
- (3) 数据比较:大于、小于、等于、不等于、大于等于、小于等于;
- (4) 数据传送:输入、输出、赋值。

算法的控制结构决定了算法的执行顺序。如以上例题所示,算法的基本控制结构通常包括顺序结构、分支结构和循环结构。不论是简单的还是复杂的算法,都是由这三种基本控制结构组合而成的。

算法是对程序控制结构的描述,而数据结构是对程序中数据的描述。因为算法的处理对象必然是问题中所涉及的相关数据,所以不能离开数据结构去抽象地分析程序的算法,也不能脱离算法去孤立地研究程序的数据结构,而只能将算法和数据结构统一起来去认识程序。但是,在计算机的高级语言中,数据结构是通过数据类型表现的,本书在后面章节中,将通过对 C 语言数据类型的详细描述说明数据结构在程序设计中的作用。这里我们只讨论算法的问题。

3. 算法的特征

一个算法应该具有以下几个重要的特征。

- (1) 有穷性:一个算法必须保证在执行有限步骤之后结束,并且每一步都在合理的时间范围内完成。
- (2) 确切性:算法的每一步骤必须有确切的定义,不允许有模棱两可的解释,也不允许有多义性。
- (3) 可行性:算法在原则上能够精确地运行,而且人们用笔和纸做有限次运算后也可完成。
- (4) 输入:一个算法有 0 个或多个输入,以刻画运算对象的初始情况。所谓 0 个输入是指算法本身定出了初始条件。
- (5) 输出:一个算法有一个或多个输出,以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

4. 算法设计的要求

设计一个算法需要注意以下几方面。