

高职高专 计算机专业

GAOZHI GAOZHUAN
JISUANJI ZHUANYE XILIE JIAOCAI

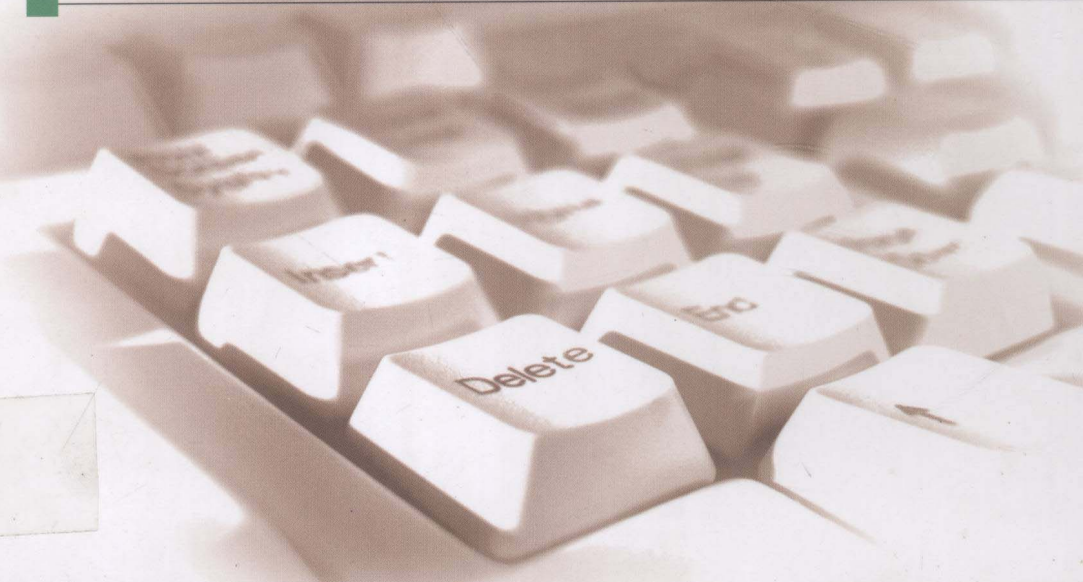
系列教材

赖万钦 主编

严桂兰 主审

C语言

程序设计基础



厦门大学出版社
XIAMEN UNIVERSITY PRESS

C 语言程序设计基础

主 编 赖万钦

副主编 陈 琳

主 审 严桂兰

编写者 (以姓氏笔画为序)

吴小惠 程伍端 雷筱珍

厦门大学出版社

图书在版编目(CIP)数据

C 语言程序设计基础/赖万钦编著. —厦门:厦门大学出版社,2006.7
(高职高专计算机专业系列教材)
ISBN 7-5615-2589-3

I. C… II. 赖… III. C 语言-程序设计-高等学校:技术学校-教材 IV.
TP312

中国版本图书馆 CIP 数据核字(2006)第 079248 号

厦门大学出版社出版发行
(地址:厦门大学 邮编:361005)
<http://www.xmupress.com>
xmup@public.xm.fj.cn
南平市武夷美彩印中心印刷
2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷
开本:787×1092 1/32 印张:20.25
字数:515 千字 印数:0001~3000 册
定价:29.00 元
本书如有印装质量问题请直接寄承印厂调换

高职高专计算机专业系列教材编委会

主 任：宁正元

副主任：薛永生 陈利永

成 员：

鄂大伟	刘传才	卢昌荆	郑汉垣	杨 升	陈志明	黄翠兰
方春春	杨文元	卢 英	余先锋	陈常晖	赖万钦	黄华国
杜旭日	许华荣	程仁贵	吴清寿	兰添才	宋文艳	眭 蔚

总序

计算机的快速发展不过几十年的时间,而它的应用已渗透到各行各业。可以说,计算机的知识与技能已经成为现今人才构成的最重要要素之一。顺应时代要求,计算机教育也在迅猛地发展。

高职高专的计算机教育不同于本科,有着自己的特点。既需要一定的理论基础,同时理论又不能太过深入;学生侧重计算机应用技术与能力的培养,需要熟悉各种应用软件,特别是办公软件、数据库等。市场上各种高职高专计算机教材繁多而庞杂,可真正适合教学的却很难寻觅。为了推进福建省计算机教材的建设与改革,提高福建省高职高专计算机教学水平,我们和厦门大学出版社一起,组织编写了这套高职高专的计算机专业系列教材。本系列教材目前包括《计算机导论》、《电路电子学基础》、《计算机数学基础》、《数据结构》、《微机原理与接口技术》、《操作系统基础教程》、《办公软件使用与操作》、《计算机网络基础》、《SQL 数据库应用基础》、《软件工程与实践》、《Visual FoxPro 程序设计实训教程》、《计算机专业英语》、《C 语言程序设计基础》13 种。其中,既包括基础的“长线”课程,又有高职高专学生必须掌握的“短线”课程。以后随着计算机科学的发展,其他计算机教材也会陆续补充进来,将会逐步形成一个较为完整的计算机类高职高专教材体系。

本套教材是依照高职高专计算机的教学特点而编写的。编写之前,厦门大学出版社做了大量的调研工作,考察了各校的教学实际。在具体编写过程中,厦门大学出版社负责联络各参编学校,组织召开教材编写会议;福建省计算机学会出版与高等教育工作委员会负责审定各教材的编写大纲,并在编写过程中给予具体指导。具体编写过程是:组建了福建省高职高专计算机专业教材编委会,各教材也成立一编写小组,设立一主编及主审,主编由对该领域非常熟悉,有着丰富教学经验者担任,主审由福建省计算机学会出版与高等教育工作委员会负责推荐或指定,亦是相关领域的专家、教授;编写者是从事教学多年的一线骨干教师,他们熟悉高职高专的教学规律和特点,具有丰富的教学实践经验;主编负责联系各参编老师,按各校教学计划,编写符合高职高专教学实际的大纲,大纲经主审审定后再编写各章节,完稿后主编统稿,同时主审审阅,根据主审意见再讨论修改,直至定稿。整个过程有条不紊,周密而审慎。

高职高专教育是今后一个时期高等教育发展的重点,正如火如荼地发展,而加强高职高专教材建设为它的发展提供了保证。所以说,我们编好这套教材可谓意义深远,责任重大。在这里,我谨向为撰写、审阅和编辑出版该教材系列付出辛勤劳动,做出重要贡献的各位专家和厦门大学出版社编辑出版的同志致以衷心的感谢!也愿福建省高职高专的计算机教育更上一层楼!

福建省计算机基础教育研究会理事长、教授 宁正元

2006 年 7 月



前 言

本书是高职高专计算机专业系列教材之一。

高职高专教育是高等教育的一个新类型,它与传统的普通高等教育既有紧密的联系,又有本质的区别,高职高专教育强调面向社会、生产、管理、服务第一线,培养技术应用型人才。本书即为满足高职高专计算机程序设计基础教学的需要而编写,以大量的实例来推动概念的讲解,真正从培养程序设计实际能力出发,由浅入深,循序渐进,避虚就实,理论与实践有机结合,融会贯通,知识传授与能力培养并重,培养学生程序设计的基本思想,掌握其基本方法。

计算机应用能力是新世纪人才不可缺少的基本素质,而程序设计是理工类学生重要的技术基础。C语言是目前广泛使用的一种程序设计语言,是国内外学生学习计算机程序设计方法的首选语言之一。

C语言是应用很广泛的一种语言,它的结构简单,数据类型丰富,表达能力强,使用灵活方便,既有高级语言的优点,又具有低级语言的许多特点。用它编写的程序,具有速度快、效率高、代码紧凑、较好的移植性等优点。利用C语言,可编制各种系统软件(例如著名的UNIX操作系统就是用C语言编写的)和应用软件。

我们组织了多年从事计算机C语言程序设计基础教学,具有丰富教学经验的教师编写了本书。本书内容丰富,注重实践,案例广泛,图文并茂,深入浅出。全书共12章和4个附录,第1章程序设计概述,第2章C语言概述,第3章数据类型与数据的输入输出,第4章运算符和表达式,第5章顺序和选择结构程序设计,第6章循环结构程序设计,第7章数组,第8章函数,第9章指针,第10章结构体和共用体,第11章编译预处理,第12章文件,包括所有高职高专学生应该学习和掌握的内容。在每一章的后面均安排与教学内容相对应的习题及上机实验,以帮助读者更好地掌握C语言程序设计理论知识,提高实践技能。书中“ ”表示空格,“↵”表示回车。

本书由赖万钦主编,陈琳任副主编,资源C语言专家、华侨大学教授严桂兰主审。其中第1、2、3章由福建交通职业技术学院雷筱珍编写,第4、5、6、7章由福建交通职业技术学院赖万钦编写,第8、9章由福建信息职业技术学院陈琳编写,第10、11章由福建交通职业技术学院吴小惠编写,第12章及附录部分由福建交通职业技术学院程伍端编写。



由于作者时间仓促,水平有限,书中难免有错误和不妥之处,诚请读者与专家指正。如有意见和建议,请与作者联系。E-mail:lwq@fjetc.cn.

编者

2006 年 7 月



目 录

总序

前言

第 1 章 程序设计概述	1
1.1 程序设计基础	1
1.1.1 程序设计语言	1
1.1.2 常用编程语言及其适用范围	3
1.2 算法及其表示	4
1.2.1 算法的概念	4
1.2.2 算法的表示	4
1.2.3 算法的特点	9
1.3 程序设计的基本流程	9
1.3.1 程序设计的基本流程	9
1.3.2 结构化程序设计方法	10
1.3.3 程序设计风格	11
小结	11
习题	12
第 2 章 C 语言概述	14
2.1 C 语言的历史背景	14
2.1.1 简述	14
2.1.2 C 语言的发展过程	14
2.2 C 语言的特点	15
2.3 C 程序的基本结构	16
2.3.1 两个简单的 C 程序实例	16
2.3.2 C 程序的基本组成	18
2.4 C 程序的上机调试	19
2.4.1 调试步骤	19
2.4.2 Turbo C 的集成开发环境	20
小结	22
习题与上机实验	23
第 3 章 数据类型与数据的输入输出	25
3.1 C 语言的数据类型	25



3.2 常量和变量	26
3.2.1 标识符	26
3.2.2 常量	27
3.2.3 变量	28
3.3 整型数据	29
3.3.1 整型常量	29
3.3.2 整型变量	30
3.4 实型数据	32
3.4.1 实型常量	32
3.4.2 实型变量	33
3.5 字符型数据	35
3.5.1 字符常量	35
3.5.2 字符变量	37
3.5.3 字符串常量	38
3.6 数据类型转换	39
3.6.1 自动转换	39
3.6.2 强制类型转换	40
3.7 数据的输出	40
3.7.1 格式输出函数 printf	41
3.7.2 字符输出函数 putchar	44
3.8 数据的输入	45
3.8.1 格式化输入函数 scanf	45
3.8.2 字符输入函数 getchar	47
小结	48
习题与上机实验	48
第4章 运算符和表达式	54
4.1 C语言运算符简介	54
4.2 算术运算符和算术表达式	54
4.2.1 算术运算符	54
4.2.2 表达式和算术表达式	55
4.3 自增与自减运算符	56
4.4 赋值运算和赋值表达式	58
4.5 关系运算和关系表达式	59
4.6 逻辑运算符和逻辑表达式	61
4.7 位运算符	63
4.8 逗号运算符和逗号表达式	66
4.9 求字节运算符	66



小结	67
习题与上机实验	67
第 5 章 顺序和选择结构程序设计	73
5.1 C 语句概述	73
5.2 顺序结构程序设计	74
5.3 选择结构程序设计	75
5.3.1 if 语句的三种形式	75
5.3.2 if 语句嵌套	80
5.3.3 条件运算符	82
5.4 switch 语句	83
小结	87
习题与上机实验	87
第 6 章 循环结构程序设计	94
6.1 概述	94
6.2 while 语句	94
6.3 do...while() 语句	97
6.4 for 语句	100
6.5 循环嵌套	103
6.6 循环控制语句和 goto 转移语句	107
6.6.1 break 语句	107
6.6.2 continue 语句	109
6.6.3 goto 语句	111
小结	112
习题与上机实验	112
第 7 章 数组	119
7.1 一维数组	119
7.1.1 一维数组的定义	119
7.1.2 一维数组元素的引用	120
7.1.3 一维数组的初始化	121
7.1.4 一维数组应用举例	121
7.2 二维数组	124
7.2.1 二维数组的定义	125
7.2.2 二维数组元素的引用	125
7.2.3 二维数组的初始化	126
7.2.4 二维数组应用举例	126
7.3 字符数组	128
7.3.1 字符数组的定义、初始化	129



7.3.2 字符数组的输入与输出	130
7.3.3 字符串处理函数	131
7.3.4 字符数组应用举例	133
小结	136
习题与上机实验	137
第 8 章 函数	140
8.1 函数的基本概念	140
8.2 函数的定义	141
8.3 函数的参数和返回值	143
8.3.1 形式参数和实际参数	143
8.3.2 函数的返回值	143
8.4 函数的调用	144
8.4.1 函数调用的方式	144
8.4.2 函数调用的执行过程	145
8.5 函数的声明	148
8.6 函数的嵌套调用	150
8.7 函数的递归调用	151
8.8 数组作函数的参数	155
8.9 局部变量和全局变量	160
8.9.1 局部变量	160
8.9.2 全局变量	162
8.10 变量的存储类型	164
8.10.1 自动变量类型	165
8.10.2 外部变量类型	167
8.10.3 静态变量	168
8.10.4 寄存器变量	170
小结	171
习题与上机实验	172
第 9 章 指针	178
9.1 指针的基本概念	178
9.2 指针变量的定义和使用	179
9.2.1 指针变量的定义和使用	179
9.2.2 指针变量的引用	180
9.2.3 地址运算	181
9.3 指针作为函数参数	182
9.4 指针与数组	183
9.4.1 指向数组元素的指针	183



9.4.2 指向数组的指针变量	186
9.4.3 指向一个由 n 个元素组成的数组指针变量	189
9.5 指针与字符串	190
9.6 指向函数的指针变量	192
9.7 指针数组和指向指针的指针	194
9.7.1 指针数组	194
9.7.2 带有形参的 main 函数	195
9.7.3 指向指针的指针	197
小结	199
习题与上机实验	200
第 10 章 结构体和共用体	209
10.1 结构体	209
10.1.1 结构体的说明	209
10.1.2 结构体变量的定义	210
10.1.3 结构体成员的引用与初始化	213
10.2 定义结构体数组	214
10.3 结构体与指针	216
10.3.1 结构体指针变量	216
10.3.2 指向结构体数组的指针	217
10.3.3 用结构体变量和指向结构体的指针变量作函数参数	218
10.4 链表的建立、插入和删除	219
10.4.1 单链表的建立	219
10.4.2 单链表的插入与删除	222
10.5 共用体	231
10.5.1 共用体的概念	231
10.5.2 共用体变量的引用方式	233
10.5.3 共用体类型数据的特点	233
10.6 枚举类型	235
10.6.1 枚举类型的概念	235
10.6.2 枚举类型的声明	235
10.6.3 枚举类型变量的定义	235
10.6.4 如何正确使用枚举类型变量	236
10.7 用 typedef 定义类型	239
小结	240
习题与上机实验	241
第 11 章 编译预处理	248
11.1 宏定义	248



11.1.1 不带参数宏定义(简单替换).....	248
11.1.2 带参数宏定义.....	250
11.2 文件包含.....	252
11.3 条件编译命令.....	253
小结.....	255
习题与上机实验.....	255
第 12 章 文件	259
12.1 C 语言文件概述.....	259
12.2 文件类型指针.....	260
12.3 文件的打开与关闭.....	261
12.3.1 fopen 函数	261
12.3.2 fclose 函数	263
12.4 文件的读写操作.....	264
12.4.1 fputc 函数	264
12.4.2 fgetc 函数	265
12.4.3 fread 函数	266
12.4.4 fwrite 函数	267
12.4.5 fprintf 函数	268
12.4.6 fscanf 函数	270
12.4.7 putw 函数	271
12.4.8 getw 函数	272
12.4.9 fgets 函数	273
12.4.10 fputs 函数.....	274
12.5 文件的定位.....	275
12.5.1 rewind 函数	276
12.5.2 fseek 函数	276
12.5.3 ftell 函数.....	278
12.6 出错检验.....	279
12.6.1 feof 函数	279
12.6.2 ferror 函数	280
12.6.3 clearerr 函数	282
小结.....	282
习题与上机实验.....	284
附录 A ASCII 字符集	290
附录 B C 语言运算符的优先级和结合性	292
附录 C C 语言常用标准库函数	294
附录 D Turbo C 上机调试常见错误信息	301
参考文献	309



第 1 章

程序设计概述

自计算机诞生以来,特别是近 20 年来,计算机技术得到了飞速发展,计算机的应用也日益普及和深入,并逐步渗透到人类生活的各个领域。要使计算机发挥作用,就必须为它编写种类不同的程序,而为了能使计算机识别这些程序,就需要有特定的编程语言——它是人们与计算机进行信息交流的工具。

本章主要介绍程序设计的基本概念和算法的基础知识。

1.1 程序设计基础

1.1.1 程序设计语言

要让计算机按人们预先安排的步骤进行工作,就要解决人机交流的问题。人们给计算机一系列的命令,计算机按给定的命令一步步地工作,这种命令就是人机交流的语言,称为程序设计语言。自计算机问世以来,程序设计语言的发展大体经历了以下几个过程。

1. 机器语言

机器语言是最底层、最早产生和使用的计算机语言。用机器语言编写的程序,计算机硬件可以直接识别。在用机器语言编写的程序中,每一条机器指令都是二进制形式的指令代码,即由 0 和 1 组合起来的一连串的二进制编码,每一条指令规定了计算机要完成的某个操作。在指令代码中,一般包括操作码和地址码,其中操作码告诉计算机做何种操作,即“干什么”,地址码则指出被操作的对象存放在哪里。

机器语言程序是由二进制 0 和 1 组成的系列,编写起来非常烦琐,可以说是“难学、难记、难写、难检查、难调试”。特别是用机器语言编写的程序完全依赖于机器,所以程序的可移植性差。但由于用机器语言编写的程序直接针对计算机硬件,因此它的执行效率比较高,能充分发挥出计算机的速度性能,这是机器语言的优点。

2. 汇编语言

为了克服机器语言的缺点,人们对机器语言进行了改进,用一些容易记忆和辨别的有意义的符号代替机器指令,如用指令助记符来代替机器语言指令代码中的操作码,用地址符号来代替地址码。用这样一些符号代替机器指令所产生的语言就称为汇编语言,也称为符号语言。

例如,为了计算表达式“9+8”的值,用汇编语言编写的程序与用机器语言(8086CPU 的指



令系统)编写的程序如表 1-1 所示。

表 1-1 汇编语言程序与机器语言程序举例

语句序号	汇编语言指令	机器指令	指令功能
1	MOV AL,9	10110000 00001001	把加数 9 送到累加器 AL 中
2	ADD AL,8	00000100 00001000	把累加器 AL 中的内容与另一数相加,结果存在累加器 AL 中(即完成 9+8 运算)
3	HLT	11110100	停止操作

从表 1-1 中可以看出,在该汇编语言程序中,以 MOV(MOVE 的缩写)代表“数据传送”,ADD 代表“加”,HLT(HALT 的缩写)代表“停止”。这些符号含义明确,容易记忆,所以又称为助记符。用这些助记符编写的程序,可读性好,容易查错,修改方便,但计算机硬件不能直接识别,必须由一种专门的翻译程序将汇编语言程序翻译成机器语言程序后,计算机才能识别并执行。这种翻译的过程称为“汇编”,负责翻译的程序称为汇编程序,翻译出的程序称为目标程序,而翻译前的程序称为源程序,如图 1-1 所示。

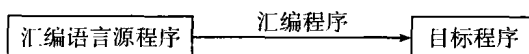


图 1-1 汇编程序的作用

汇编语言依赖于机器硬件,移植性不好,但效率高。针对计算机特定硬件而编制的汇编语言程序能准确发挥计算机硬件的功能和特长,程序精练且质量高,所以至今仍是一种常用且强有力的软件开发工具。

3. 高级语言

在计算机诞生的初期,其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。这样,面向机器的语言编程已不能满足要求,1954 年出现了第一种高级语言——FORTRAN。

40 多年来,有几百种高级语言出现,有重要意义的有几十种。影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、Basic、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、Visual C++、Visual Basic、Delphi、Java 等。

使用高级语言编程时人们不必熟悉计算机内部的具体构造,也不必熟记机器指令,而是把主要精力放在算法描述上面。因此,面向过程的高级语言又称算法语言(或过程语言)。用高级语言编写的程序计算机也不能直接执行,也需要一种工具将它翻译为机器语言程序,这种工具就是高级语言的编译系统。将高级语言程序翻译为意义等价的机器语言程序的方法一般有编译方式和解释方式两类。

(1) 编译方式

编译方式是将高级语言编写的程序(称为源程序)全部翻译成机器语言程序(称为目标程序)后,再将目标程序交给计算机执行,最后得出源程序执行结果。实现翻译的程序为编译程序。它实际上包括翻译和查错两个功能,具体功能包括词法分析、语法分析、语义分析、生成目标程序及优化目标程序等。



(2) 解释方式

解释方式是由解释程序来实现的,它边翻译边执行,即翻译一句,执行一句,不产生整个目标程序。一般高级语言(如 C、FORTRAN 等)都用编译方式,而 Basic 语言及 FoxPro 等数据库语言多用解释方式。解释方式使用灵活方便,占内存少,但执行时间长,效率低;编译方式得到的目标程序执行速度快,但占内存多。

4. 非过程化的高级语言

以上介绍的高级语言属于面向过程的高级语言,用这种语言解决问题时,人们首先要理解“做什么”,然后去构造“怎么做”的解题过程,程序设计者要详细地规定计算机操作的每一个细节。而非过程化的高级语言只需告诉计算机“做什么”,而无须去构造“怎么做”过程,如各种数据库语言 SQL、dBASE、FoxBASE、FoxPro 等均属此类语言。例如,只要告诉机器“找出平均成绩在 60 分以下的学生名单”,机器便会按人们的意图去完成这项工作。

非过程化的高级语言的解题效率更高,使用更为简便,更适合于非计算机专业人员使用。但是,它的运行效率及灵活性都不如过程化语言。因此,学习用过程化的高级语言进行程序设计仍是计算机应用人员的一项基本训练。

5. 面向对象的程序设计语言

随着计算机处理问题复杂程度的增加,程序的规模越来越大。于是,20 世纪 80 年代,人们又提出面向对象的程序设计方法,它使程序设计变得更加简练、自然。

近几年来,面向对象的程序设计语言得到了迅速发展和应用,例如 C++、Visual C++、Visual Basic、Delphi 等。它们比面向过程的程序更清晰易懂,更适合编写大规模的程序。

1.1.2 常用编程语言及其适用范围

目前,用于程序设计的语言已有数百种,每种语言都有各自不同的特点,可适用于不同的场合。下面对几种应用比较广泛的高级语言的特点及适应范围作一简单介绍。

FORTRAN 语言是世界上最早出现的高级语言,广泛应用于科学、工程计算。在数值处理领域,FORTRAN 至今仍是一种可选的语言。

Basic 语言原是为初学者学习程序设计而研制的一种语言,目前,Basic 语言已有多种版本。Microsoft Visual Basic 进一步提供了一个可视的开发环境,具有图形设计工具、结构化的事件驱动编程模式,使用户可以既快又简便地编制出 Windows 下的各种应用程序。

C 语言最初是作为操作系统设计语言而研制的,UNIX 操作系统就是用 C 语言实现的。C 语言具有很强的功能且十分灵活。它不仅用于操作系统的设计,也可用于设计各类应用程序。目前,流行的 C 语言版本是 Borland 公司的 Turbo C。

C++ 已成为当今最受欢迎的面向对象的程序设计语言,因为它既融合了面向对象的能力,又与 C 语言兼容,保留了 C 语言的许多重要特性,使 C 程序员不必放弃自己已经十分熟悉的 C 语言,而只要补充学习 C++ 提供的那些面向对象的概念即可。目前比较流行的 C++ 版本主要有 Borland C++ 和 Microsoft Visual C++。

Java 也是一种面向对象的程序设计语言,基本功能类似于 C++ 语言,但做了许多重大修



改。它适用于 Internet 环境,并具有较强的交互性和实时性。它提供了网络应用的支持和多媒体的存取,推动了 Internet 和企业网络的进步。

除以上介绍的这些通用程序设计语言之外,还有一类是为特殊应用而设计的专用语言,它们通常具有自己特殊的语法形式,面向特定的问题。较有代表性的专用语言有 APL、LISP 和 PROLOG。APL 是一种简单的对数组和向量处理非常有效的语言,主要用来解决一些数据计算问题。LISP 和 PROLOG 均适用于人工智能领域,用于专家系统的分析和构造。

1.2 算法及其表示

1.2.1 算法的概念

从事各种工作和活动,都必须事先想好进行的步骤,然后按部就班地进行,才能避免产生错乱。因此为解决一个问题而采取的方法和步骤,就称为“算法”。

对同一个问题,可以有不同的解题方法和步骤。方法有优劣之分。有的方法只需进行很少的步骤,而有些方法则需要较多的步骤。人们一般希望采用简单的运算步骤少的方法,因此,为了有效地解决问题,不仅要保证算法正确,还要考虑算法的质量,选择合适的算法。

计算机算法可分为两大类:数值运算和非数值运算。数值运算的目的是求数值解。非数值运算应用则十分广泛,最常见的是用于事务管理领域。目前,计算机在非数值运算方面的应用远远超过了其在数值运算方面的应用。由于数值运算有现成的模型,可以运用数值分析方法,因此对数值运算的算法研究比较深入,算法比较成熟。各种数值运算都有比较成熟的算法可供选用。人们常常把这些算法汇编成册(写成程序形式),或者将这些程序存放在磁盘或磁带上,供用户调用。

而非数值运算的种类繁多,要求各异,难以规范化,因此只对一些典型的非数值运算算法(例如排序算法)作比较深入的研究。其他的非数值运算问题,往往需要使用者参考已有的类似算法重新设计解决特定问题的专门算法。

1.2.2 算法的表示

对一个算法的表示,可以用不同的方法。常用的描述算法的方法有自然语言、传统流程图、N-S 结构流程图、伪代码、PAD 图等,这里主要介绍前三种。

1. 用自然语言表示算法

自然语言就是人们日常使用的语言,可以是汉语、英语或其他语言。用自然语言描述算法具有通俗易懂的优点,但缺点也比较多:

(1) 比较烦琐。往往要用一段烦琐的文字才能说清楚所要进行的操作。