

应用型本科计算机科学与技术规划教材

夏 涛 主编

语言程序设计



北京邮电大学出版社
www.buptpress.com

应用型本科计算机科学与技术规划教材

TP312

2271

2007

C 语言程序设计

夏涛 主 编

刘丹 孙丽云 副主编

北京邮电大学出版社

· 北 京 ·

内 容 简 介

C 语言到目前为止仍然是国内外广泛使用的程序设计语言之一,同时也是国内外大学开设的程序设计基础课程。

本书作为第一门程序设计课程的教材或自学读物,讲述了 C 语言程序设计中涉及到的最基本的语法规则和程序设计的基本方法。全书的各章节从分析程序示例开始,讲述 C 语言程序设计中的概念、方法并分析它们的应用特点。书中的例程都给出了详细注释,并给出了程序的流程图。

由于篇幅有限,本书对 C 语言的内容不可能面面俱到,更多的内容通过“我爱 C”(www.5ic.net.cn)教学辅助系统提供给教师和学生。教师可以利用该系统提供的考试系统辅助教学,学生可以利用该系统提供的练习和讨论系统辅助学习。

本书涵盖了 C 语言的基本概念,其中部分内容有一定的深度,这样设置可以满足不同专业、不同学时教学需要。对信息类专业可以讲授本教材的全部内容,非信息类专业可以讲授本教材的部分内容。

本书和“我爱 C”教学辅助系统也适合计算机水平考试培训及各类成人教育等教学使用,还可以供计算机爱好者自学。

图书在版编目(CIP)数据

C 语言程序设计/夏涛主编. —北京:北京邮电大学出版社,2007

ISBN 978-7-5635-1384-0

I. C... II. 夏... III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 030557 号

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号(100876)

北方营销中心:电话:010-62282185 传真:010-62283578

南方营销中心:电话:010-62282902 传真:010-62282735

E-mail: publish@bupt.edu.cn

经 销:各地新华书店

印 刷:北京忠信诚胶印厂

开 本:185 mm×260 mm 1/16

印 张:14.25

字 数:351 千字

印 数:1—5 000 册

版 次:2007 年 3 月第 1 版 2007 年 3 月第 1 次印刷

ISBN 978-7-5635-1384-0/TP·65

定价:23.00 元

· 如有印装质量问题,请与北京邮电大学出版社营销中心联系 ·

应用型本科计算机科学与技术规划教材

编委会

主 任：乐光新

副主任：（按姓氏笔划排列）

冯 林 吕 翊 胡建萍

唐志宏 曹雪虹 雷章富

委 员：（按姓氏笔划排列）

付 瑜 刘发久 李守明

许学东 李秉智 倪玉华

夏 涛 夏素霞

前 言

欢迎进入 C 语言世界!

C 语言到目前为止仍然是国内外广泛使用的程序设计语言之一,同时也是国内外大学开设的程序设计基础课程。C 语言自身具有功能丰富、表达力强、使用方便灵活、程序代码效率高、可移植性好等特点。此外 C 语言不但具有高级语言的特点,又有汇编语言的特点,具有较强的系统处理能力。C 语言程序采用“自顶向下,逐步细化”的程序设计思路,符合结构化程序设计的要求,函数就是程序的最小模块,通过函数调用实现了模块化程序设计。

本书的目标是作为第一门程序设计课程的教材或自学读物,讲述 C 语言程序设计中涉及到的最基本的语法规则和程序设计的基本方法。当然,仅仅阅读本书是不可能写出高水平的程序的,通过学习本书可以为今后在程序设计领域的进一步学习和工作做好准备。

作为程序设计的入门书,不可能去讨论程序设计的理论成果,也不能过度论述程序设计的细枝末节。

俗话说:读书破万卷,下笔如有神。要想读书破万卷,不但要有决心和毅力,更需要有“读书”的能力,一个不识字的人是永远不可能“破万卷”的。有编程经验的人都知道,学习编程一定是从读程序开始的,所以让读者尽快开始阅读程序是学好程序设计的基础。本书的特点是从介绍三种基本程序开始,然后再介绍其他细节问题,以便让读者能尽快进入阅读程序的状态。此外书中所有的概念都尽可能从程序中引出,这样可以使读者把 C 语言的各种规则和实例紧密联系在一起。书中的程序都给出了详细注释,并配有相关程序的流程图,方便读者学习。

各种书籍(包括本书)给出的程序并不一定是最优的设计,就本书而言,一些程序示例是为了说明某个基本概念,并不一定是最优的代码实现方法。这就要求读者能思考书中提出的一些问题,给出更好的答案,这将使您受益无穷。

无论是教师的教学还是学生的学习,课本只是其中的一个组成部分,并且由于教材的篇幅有限,不可能面面俱到。为了弥补这个不足,请各位读者登录“我爱 C”(www.5ic.net.cn)教学辅助系统,该系统为教师提供了辅助考试系统,为学生提供了练习系统,并设置了讨论区用于讨论与 C 语言程序设计有关的问题。“我爱 C”教学辅助系统是对本书的一个有益的补充。

考虑到不同学校、不同专业对 C 语言程序设计课程的教学要求不同,各位教师可以根据课程的课时设置选择授课内容。

本书由多年从事 C 语言程序设计课程教学的、具有丰富教学经验和实践经验的教师集体编写。徐晓明和孙丽云合写了第 1 章,孙丽云编写了第 2 章、第 8 章、第 10 章和第 11 章,史益芳编写了第 3 章和第 9 章,刘丹编写了第 4 章、第 5 章和第 7 章,夏涛和史迎春合写了第 6 章,徐晓明编写了第 8 章,李德亮编写了附录。

在这里要感谢北京化工大学莫德举教授、北京化工大学北方学院张春先副校长、北京化工大学北方学院教务处白艳苹处长等对本书编写过程中给予的指导和帮助,感谢北京邮电大学出版社为编写本书给予的大力支持。

由于作者水平有限,再加上时间仓促,书中的缺点和错误在所难免,恳请各位读者不吝赐教,不胜感激! 欢迎登录“我爱 C”(www.5ic.net.cn)辅助教学平台的相关页面给我们留言。

作者电子邮箱:xiatao@mail.buct.edu.cn

夏涛

2007 年 1 月于北京

目 录

第 1 章 引言

1.1 C 语言的特点	1
1.2 简单的 C 程序	2
1.3 C 程序的完整结构及程序执行	4
1.4 算法和 C 程序的三种基本结构	6
1.5 小结	8
习题	8

第 2 章 选择结构及其应用

2.1 关系运算符和关系表达式	11
2.2 逻辑运算符和逻辑表达式	12
2.3 if 语句	14
2.4 switch 语句	22
2.5 小结	26
习题	27

第 3 章 循环结构

3.1 while 循环	29
3.2 do-while 循环语句	31
3.3 for 循环语句	32
3.4 三种循环语句的比较	35
3.5 循环嵌套	36
3.6 break、continue 语句	38
3.7 goto 语句	41
3.8 小结	42
习题	42

第 4 章 数据类型及其运算

4.1 基本字符、标识符	43
4.2 C 语言的数据类型	44
4.3 常量与变量	44

4.4 整型数据	46
4.5 实型数据	47
4.6 字符型数据	48
4.7 变量初始化	49
4.8 数据类型转换	50
4.9 运算符和表达式	51
4.10 简单输入输出函数	56
4.11 小结	60
习题	60
第5章 数组	
5.1 一维数组	62
5.2 字符数组与字符串	68
5.3 多维数组	74
5.4 小结	77
习题	77
第6章 函数	
6.1 函数概述	79
6.2 库函数	81
6.3 用户自定义函数	82
6.4 函数的说明	87
6.5 函数的调用	90
6.6 函数的参数传递	93
6.7 函数的嵌套调用	95
6.8 递归	97
6.9 数组作为函数的参数	99
6.10 局部变量和全局变量	101
6.12 变量的作用域和生存期	109
6.13 内部函数和外部函数	110
6.14 掷骰子游戏	110
6.15 小结	115
习题	115
第7章 预处理命令	
7.1 宏定义与宏定义的撤销	119
7.2 文件包含命令	122
7.3 条件编译命令	124
7.4 小结	126

第 8 章 指针

8.1 指针概述	127
8.2 指针的概念	128
8.3 指针与变量	129
8.4 指针运算	131
8.5 指针与数组	133
8.6 指针与字符数组	136
8.7 指针与字符串	137
8.8 指针与函数参数	138
8.9 指针与多维数组	141
8.10 指针与指针数组	145
8.11 指针与函数	147
8.12 指针与返回指针值的函数	147
8.13 小结	149
习题	149

第 9 章 结构体与共用体

9.1 结构体的说明和定义	151
9.2 结构体成员的引用与结构体变量的初始化	155
9.3 结构体数组	158
9.4 结构体指针	161
9.5 结构体在函数间的传递	166
9.6 结构体型函数和结构体指针型函数	170
9.7 共用体简介	173
9.8 枚举类型	175
9.9 用 typedef 定义类型	177
9.10 小结	177
习题	177

第 10 章 文件

10.1 文件概述	178
10.2 文件类型指针	180
10.3 文件的打开与关闭	181
10.4 文件的读写	183
10.5 文件的定位	192
10.6 小结	194
习题	194

第 11 章 链表

11.1 链表的类型及定义·····	195
11.2 处理动态链表的函数·····	197
11.3 动态单链表的基本操作·····	198
11.4 栈和队列·····	204
11.5 小结·····	205
习题·····	205

附录 A “我爱 C”计算机辅助教学系统简介

A.1 系统简介·····	206
A.2 教师端辅助教学系统功能简介·····	206
A.3 学生端练习及答疑系统·····	210
A.4 与传统考试方式的对比·····	211

附录 B 运算符的优先级和结合性表····· 212

附录 C ASCII 码表····· 214

附录 D C 语言的关键字····· 215

参考文献····· 216

第 1 章 引 言

了解 C 程序的结构, C 程序语句的表示方法, C 程序的工作过程; 了解 C 程序是由函数组成的特点。

本书的作者在此向读者致谢并向读者介绍本书的结构。

本书的每一章分为四个部分, 第一部分是导读部分, 其次是讲述本章内容知识的部分, 再其次是本章的小结部分, 最后是练习作业的部分。

导读部分里简要介绍了本章的重点、难点、知识点及必须掌握的基本知识。第二部分讲述本章的知识内容, 是本章的重要部分。第三部分总结本章的内容, 对章节中的重点、难点、知识点及必须掌握的基本知识作一个较全面的归纳总结。最后是练习部分, 在这部分中给出本章的练习题。

本书是 C 语言的基础及应用教材, 它的重点放在对 C 语言语法的理解和阅读程序上, 通过阅读从简单到复杂的 C 程序来理解 C 语言的语法表示, 了解简单的编程方法和编程过程中用到的算法结构。从本书的内容和结构上讲, 它不是一本讲述编程的教材, 如果想学习编写 C 语言的程序, 可以参考 C 语言编程的教材、书籍或专著。

1.1 C 语言的特点

C 语言是国际上广泛流行的、很有发展前途的计算机高级语言。它既可以用来编写系统软件, 也可以用来编写应用软件。

C 语言与 UNIX 操作系统有着密切的联系, 开发 C 语言的主要目的是为了较好地描述 UNIX 操作系统。C 语言的出现, 促进了 UNIX 操作系统的开发, 同时随着 UNIX 使用的日益广泛, C 语言也迅速得到推广, C 语言和 UNIX 可以说是一对孪生兄弟, 在发展过程中相辅相成。

一种语言之所以能够存在和发展, 并具有生命力, 总是有其不同于其他语言的特点。C 语言的主要特点如下:

1. 中级语言

C 语言具有高级语言的先进思想又能直接对存储器进行操作, 能进行位运算, 能实现汇编语言的大部分功能, 生成目标代码质量高, 程序执行效率高。

2. 结构化语言

C 语言用函数作为程序模块,以实现程序的模块化,语言简洁、紧凑,具有结构化的特点。

3. 可移植性好

C 语言不包含依赖硬件的输入输出机制,使 C 语言本身不依赖于硬件系统,可移植性好。

1.2 简单的 C 程序

C 语言是人与计算机交流的一种语言,如果你需要计算机来帮助完成某些工作,则可以用 C 语言来表述你的思想并将它输入到计算机中,让计算机来“运行”它。

C 语言是一种通用性很强的结构化程序设计语言,它具有丰富的运算符和数据类型,语言简单灵活,表达能力强。

我们用三个由简单到复杂的 C 程序来说明用 C 语言编写的程序是什么样的结构,怎样去阅读这些 C 程序。

第一个程序的功能是在屏幕上显示“This is a C Program”。

```
#include "stdio.h"          /* stdio.h 是标准输入/输出头文件 */
main()                     /* main 是 C 语言的主函数 */
{
    printf("This is a C Program");    /* printf 是格式输出函数 */
}
```

第二个程序的功能是从键盘上输入一个表示圆柱高度的数据 h 和一个表示半径的数据 r ,然后根据已给出的数学公式 $v=h * r * r * 3.14159$ 计算体积。

```
/* 这是计算体积的 C 程序 */
#include "stdio.h"          /* stdio.h 是标准输入/输出头文件 */
main()                     /* main 是 C 语言的主函数 */
{
    int h,r;               /* 定义整型变量 h 和 r */
    float v;               /* 定义实型变量 v */
    scanf("%d%d",&h,&r);   /* 格式输入函数 */
    v=h*r*r*3.14159;       /* 计算体积 */
    printf("%f",v);        /* 格式输出函数 */
}
```

第三个程序的功能是从键盘上输入 3 个数据 a, b, c , 这 3 个数据分别表示三角形的 3 条边,用 a, b, c 计算三角形的面积。

```
/* 这是计算三角形面积的 C 程序 */
#include "stdio.h"
#include "math.h"
```

```

float fun_area(int x,int y,int z)
{
    float s,temp;                /* 定义实型变量 s,temp */
    s=(x+y+z)/2.;
    temp=sqrt(s*(s-x)*(s-y)*(s-z));
    return(temp);                /* 返回函数 fun_area 的运算结果 */
}

main()
{
    int a,b,c;                    /* 定义整型变量 a,b,c */
    float area;                   /* 定义实型变量 area */
    scanf("%d%d%d",&a,&b,&c);
    area=fun_area(a,b,c);        /* 调用函数 fun_area */
    printf("%f\n",area);
}

```

这3个程序都是完整的C程序,经过编译和链接后可以产生可执行文件,运行可执行文件,其执行的结果显示在屏幕上。

下面对上述的程序作一些说明,以便了解用C语言编写的程序是怎样的结构以及书写结构,在程序中虽然有一些内容没有讲述过但不影响我们去阅读这3个C程序。

在第一个程序中遇到的`#include "stdio.h"`是一条编译预处理命令,它的作用是通知C语言编译系统在对C程序进行正式编译之前应该做一些预处理工作,详细论述放在后面的章节中。`stdio.h`是C语言系统文件,`stdio`是标准输入输出的缩写,.`h`是文件的扩展名,它说明该文件是一个头文件(head file)。

程序中的`main()`是主函数名,它表明这是一个主函数,是程序的“入口”,是程序执行的第一条语句,阅读C程序就是从这里开始。注意:在C程序中,`main()`函数必须有且只能有一个。主函数的位置可以放在整个C程序中的任何位置。在`main()`函数后面出现了一对花括号,这对花括号括起来的内容称作主函数体,在这个C程序中只有一条语句——`printf("This is a C Program");`。

`printf()`是格式输出函数,它是系统的库函数(又称为标准函数)。这些库函数由系统开发商事先编写好并存放在系统文件中。该函数的作用是在屏幕光标的位置上输出数据,在这个C程序中,该函数输出的是字符串“This is a C Program”。

第二个C程序都做了什么事情呢?它是从键盘输入两个数据`h`和`r`,求体积`v`。那么,在程序中看到的`int`和`float`表明了什么呢?它们是C语言系统专用的符号,我们称它们为C程序的关键字。这两个关键字的作用是向计算机系统提出请求,申请在内存中占用内存空间。如果计算机操作系统能够满足C程序申请的占用的内存空间请求,则会在内存中分配内存空间给C程序使用。同样是申请内存空间,为何两者关键字是用不同的符号表示呢?这是因为`int`请求使用的内存空间和`float`请求使用的内存空间不同。具体的内容在后面的章节中会详细论述。

`scanf()`是格式输入函数,它也是系统的库函数。该函数是用来输入数据的函数,计算

机执行到该函数时会停下来等待键盘上输入的数据。

$v = h * r * r * 3.14159$; 是一条算术运算语句, 它根据输入的数据 h 和 r 计算出体积 v 。

第三个 C 程序比较复杂, 它由两个函数组成, 即主函数 `main` 和自定义函数 `fun_area`。自定义函数完成三角形面积的计算并把计算完成的结果返回给 `main` 函数。这种程序结构就是函数结构, 关于函数结构的特点, 在下面会作一些说明。

仔细观察这 3 个 C 程序会发现在 C 程序中某些行的后面有分号, 分号的作用是表明语句到此结束。每个 C 程序的语句后面必须有分号, 如果在编写 C 程序时忘记输入分号的话, 那就犯了个大错误。注意: 不是每一行后面都有分号, 只有语句后面才能有分号。

1.3 C 程序的完整结构及程序执行

1.3.1 C 程序的完整结构

通过三个 C 程序就知道了它们的基本结构, 它们的主体结构由头文件、主函数、系统的库函数和自定义函数组成。最简单的 C 程序结构如下:

```
#include "stdio.h"
main() {; }
```

当然, 这是一个什么都不做的 C 程序, 它给我们展示了一个完整的 C 程序结构。

在第二个 C 程序中我们应该学习到哪些知识? 应该知道在主函数体中因语句的作用不同, 它们所处的位置也不同。主函数体分为两大部分, 前面一部分是变量的定义和自定义函数的声明部分, 后面一部分则是程序的可执行部分。变量定义的语句或函数声明的语句不能放在可执行语句之后, 同时注意: C 程序中需要的变量一定要在使用之前进行定义, 这就是“先定义, 后使用”的原则。

程序是如何运行的? 不妨从第三个 C 程序入手看看 C 程序是怎样运行的, 为了讲解的方便把 C 程序的前面加上行号。具体的程序如下:

```
10  #include "stdio.h"
20  #include "math.h"
30  float fun_area(int x,int y,int z)
40  {
50      float s,temp;                /* 定义实型变量 s,temp */
60      s=(x+y+z)/2.;
70      temp=sqrt(s*(s-x)*(s-y)*(s-z));
80      return(temp);                /* 返回函数 fun_area 的运算结果 */
90  }
100 main()
```

```

110 {
120     int a,b,c;                      /* 定义整型变量 a,b,c */
130     float area;                    /* 定义实型变量 area */
140     scanf("%d%d%d",&a,&b,&c);
150     area = fun_area(a,b,c);        /* 调用函数 fun_area */
160     printf("%f\n",area);
170 }

```

math.h 头文件的作用在于:因为在这个 C 程序中用到了开方函数 sqrt,这个标准的库函数放在 math.h 中,所以在程序的头部要有 #include "math.h" 文件包含的描述。

现在开始运行这个程序,最初是从行号为 100 的位置开始执行,因为 main() 是程序的入口,接着执行行号为 120 的定义变量的语句,它定义了三个整型变量 a,b,c。再执行行号为 130 的定义变量的语句,它定义了一个实型变量 area。然后执行行号为 140 的数据输入语句,从键盘读入三个整型数据。再执行行号为 150 的语句,这是一个函数调用语句,执行这条语句时程序的执行转向到行号为 30 的位置,这是自定义函数的定义语句。在执行这条语句时系统定义了三个整型变量 x,y,z。执行行号为 50 的语句,该语句定义实型变量 s,temp。行号为 60 的语句计算出 $(x+y+z)/2$ 的值并将它赋给变量 s。行号为 70 的语句计算出 $\sqrt{s*(s-x)*(s-y)*(s-z)}$ 的值并将它赋给变量 temp。执行行号为 80 的语句是把变量 temp 的值作为自定义函数的运算结果返还给主函数 main(),执行该语句时程序的执行转向到行号为 150 的位置上,将自定义函数的运算结果赋给变量 area。注意:C 程序是从行号 150 处转出去的,此时又从行号 80 处转回到行号 150 处。最后执行行号 160 的语句,它输出变量 area 的值。

这三个在 C 程序中都可以看到 /* */ 这样的描述,它们出现在程序的开始或在语句的后面,这是为了对 C 程序或某些语句进行说明而采取的描述。我们把要说明的文字放在 /* */ 之间,这样给阅读 C 程序的读者提供方便,使读者可以很好地理解语句的作用。放在 /* */ 之间的文字不会被当成程序的一部分,编译系统在对 C 程序进行语法检查时把它们当作文字的注释部分。

对 C 程序的书写格式没有严格的规定,它不同于 C 程序的组成结构,C 程序的书写格式比较自由,但它仍有不成文的规范约定,从第二个 C 程序和第三个 C 程序的书写格式中可以看出。不成文的约定就是书写 C 程序按控制关键字从属关系缩进的方法,这种书写方法在写判断结构程序代码和循环结构程序代码时尤为重要,它可以清楚地描述 C 程序中的各种功能流程控制结构。一般缩进八个字符的位置较为合适,也就是八个制表位,按一下 TAB 键可获得八个制表位。

1.3.2 C 程序的编译及执行

1.2 节中的 C 语言程序代码编写完成之后,就可以在机器上运行它了。我们知道 C 语言是一种程序设计语言,它很容易被人们看懂和接受。但对于计算机来说,它只能接受机器语言。为此必须首先把 C 语言程序翻译成相应的机器语言程序,这个工作叫编译。C 程序

的编译及执行过程如图 1-1 所示。

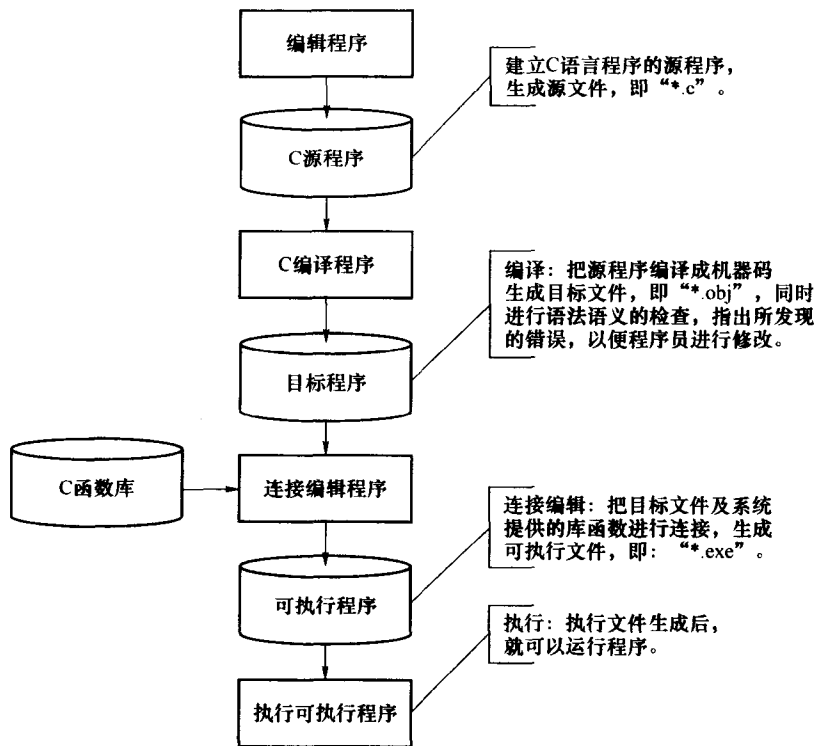


图 1-1 C 程序的编译及执行过程

1.4 算法和 C 程序的三种基本结构

算法是为了解决一个问题而采取的方法和步骤。比如:讲授《C 语言程序设计》的老师临时有事需要调课,这位老师可能会这样安排:领取调课单——填写调课单——找院长签字——将调课单送教务处批复——通知班长调课。他的这一系列步骤和完成每一步采用的方法就可以称之为“算法”。在计算机科学中,算法要用计算机算法语言描述,算法代表用计算机解决一类问题的精确、有效的方法。算法和程序之间存在密切的关系。

1.4.1 算法的特点

算法具有以下特点:

1. 确定性

算法的每一种运算必须有确定的意义,该种运算应执行何种动作应无二义性,目的明确,也就是说,没有异议和歧义是描述一个算法时最基本的要求。

2. 可行性

要求算法中有待实现的运算都是基本的,每种运算至少在原理上能由人用纸和笔在有

限的时间内完成。

3. 输入

一个算法有 0 个或多个输入,在算法运算开始之前给出算法所需数据的初值,这些输入取自特定的对象集合;没有输入的程序是缺乏灵活性的。

4. 输出

作为算法运算的结果,一个算法产生一个或多个输出,输出是同输入有某种特定关系的量;没有输出的算法是没有任何意义的。

5. 有穷性

一个算法总是在执行了有穷步的运算后终止,即该算法是可达的。

1.4.2 算法的表示方法

算法的表示方法很多,通常有以下几种:

1. 用自然语言表示

自然语言表示算法可以用任何语言,比如:汉语、英语、俄语等,当然也可以用数学表达式。用自然语言表示通俗易懂,但可能文字冗长,不太严格,并且复杂的算法表示很不方便。所以除了简单的问题外,一般不用自然语言描述算法。

2. 用传统流程图表示

传统流程图可用一些图框和流程线来表示各种类型的操作。优点是直观形象,易于理解,缺点是传统流程图不易修改。

传统流程图常用符号如图 1-2 所示。

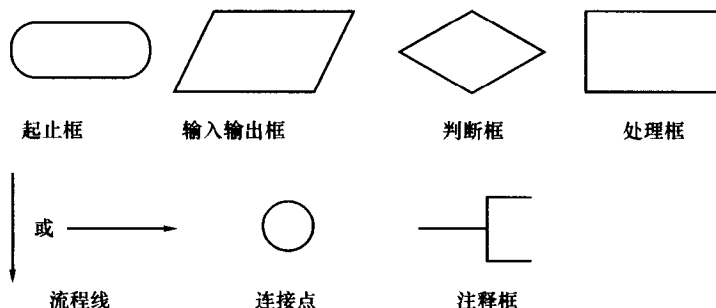


图 1-2 流程图常用符号

3. 用 N-S 流程图表示

N-S 流程图是一种新的流程图形式。在这种流程图中,完全去掉了带箭头的流程线,全部算法写在一个矩形框内,在该框内还可以包含其他从属于它的框。N-S 流程图适用于结构化程序设计。

4. 用伪代码表示

用流程图表示算法直观易懂,但不容易修改。伪代码可以克服流程图的这个弱点。

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法的,它不用图形符号,因此书写方便,格式紧凑,好懂,也便于向计算机程序转换。

一般在写程序之前,先列出算法,会使编程思路清晰,虽然有些简单的程序可以直接写