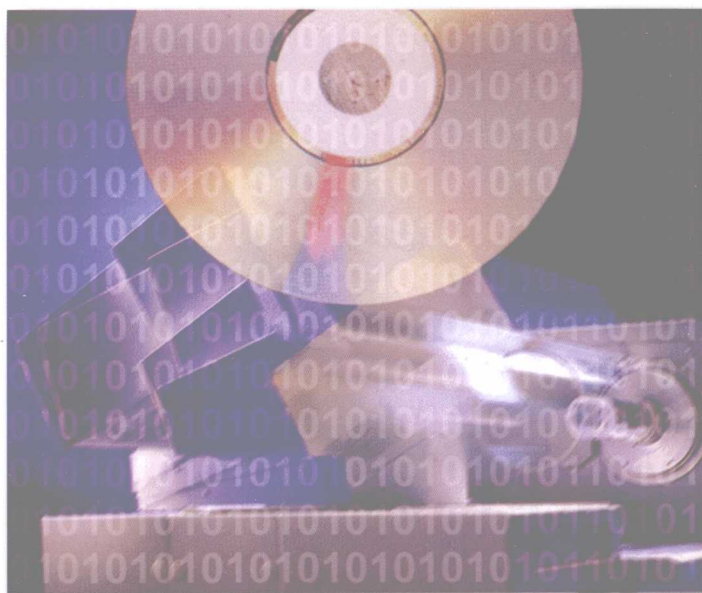


C语言

程序设计教程 (第二版)

秦友淑 曹化工 编著



高等学校教材

C语言程序设计教程

(第二版)

秦友淑 曹化工 编著

华中科技大学出版社

中国·武汉

图书在版编目(CIP)数据

C 语言程序设计教程(第二版)/秦友淑 曹化工 编著. —武汉:华中科技大学出版社,
2002 年 12 月

ISBN 978-7-5609-1185-4

I. C… II. ①秦… ②曹… III. C 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 065041 号

C 语言程序设计教程(第二版)

秦友淑 曹化工 编著

责任编辑:唐元瑜

封面设计:潘 群

责任校对:封春英

责任监印:熊庆玉

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉皇荣文化发展有限责任公司

印 刷:湖北恒泰印务有限公司

开本:787mm×1092mm 1/16

印张:22.75

字数:550 000

版次:2002 年 12 月第 2 版

印次:2009 年 1 月第 14 次印刷

定价:26.80 元

ISBN 978-7-5609-1185-4/TP·158

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书严格遵循 C 语言标准,全面、系统、深入浅出地阐述了 C 语言的基本概念、语法和语义,以及用 C 语言进行程序设计的方法和技术。内容包括数据类型、运算符和表达式、流程控制、函数与程序结构、输入输出及其低层接口。其中,对 C 语言的表达式、类型转换、变量的存储类型和指针的用法等重点和难点均作了明确而详细的介绍。

本书概念清楚、叙述详实、内容新颖实用,实例典型丰富、要点清晰明了。实例程序均经过上机调试运行,并给出了对算法思路的分析及算法步骤。每章末均附有重点小结及精选的习题,便于教学使用。

本书是一本准确而又较全面、深入地反映标准 C 的程序设计教材,既适于高等院校计算机、通信、电子、自动化等信息学科各专业及其他有关专业(含网络远程教育、成人教育)本科及专科教学使用,也适合广大科技人员和研究生自学参考。

第二版前言

本书第一版自1996年出版以来受到了广大读者的好评,感觉该书是一本准确而又较全面反映标准C的程序设计教材,其内容新颖、丰富、实用、适于教学,因而国内很多高校相继在教学中使用。为适应当前高校计算机教育的发展和教学改革的需要,第二版在第一版基础上考虑了IEEE/ACM Computing Curricula 2001对语言部分的知识点及核心内容的要求,从注重理论和实际相结合、加强对学生思维方式的训练的原则出发,对难度较大的章节(主要是第2章和第7章)和一些实用性强的知识点(例如输入输出、动态存储分配、文件)的阐述,从规则、方法、分析、要点归纳及举例说明等方面作了适当的细化和补充。本书第二版中具体的修改和补充归纳如下:

(1) 数据类型是程序设计语言的重要组成部分,理解数据类型和它的作用对于学好程序设计语言和掌握程序设计方法是非常重要的。为此,第2章加强了对运算过程中类型转换规则的举例和说明,以及对在理解表达式计算结果的值时,数据类型的重要作用的说明。

(2) 输入输出的有关内容分布在本书的第3和第9两章。第3章补充了对常用的和几个虽不常用但很实用的输入输出转换字符用法的举例说明,以及对字符和字符串输入输出方法的举例说明。

(3) 原第一版6.3.1节中对多维数组处理方法的说明移到了现第二版7.4.4节讲述二维数组的指针表示之前。由于C语言中数组与指针有着密不可分的联系,因而作这种调整之后有助于理解数组与指针的联系,进而有助于理解二维数组的指针表示。

(4) 第7章在讲述数组的指针表示时,紧紧抓住数组与指针的本质联系和“数组名是数组元素类型的指针”这个关键,只要首先正确地写出数组元素的地址表达式,则通过间访运算符(单目*)与地址表达式的关系可立即写出元素的指针表示形式;此外,由于二维数组存在两种类型的指针(n维数组存在n种类型的指针),因而增加了数组元素的指针表示的复杂性。本书在讲述二维数组的指针表示时,从分析C语言对多维数组的处理方法出发,紧紧抓住二维数组的数组名及二维数组第一级分解的数组元素这两种指针的类型,采用对同一个数组元素的各种等价的表示进行逐步推导、归纳和比较的方法,从而使读者更容易理解和掌握。

(5) 第8章在8.5节中将“C的动态存储分配函数”单独列为一小节,以malloc函数为例讲述动态存储分配函数的用法,还介绍了其他几个常用的动态存储分配函数的原型和功能;在8.5.2节中补充了链表排序的例子。

(6) 第9章补充了一个完整的示范性应用程序,该程序综合运用了几乎全书所有的知识(包括基本类型数据、数组、结构、指针、链表、文件以及各种流程控制语句),是一个由多个函数、多个源文件组成的典型的C程序,可供学生进行课程设计或完成大型作业时参考。

(7) 为便于读者容易理解或自学,第二版对于全书的文字叙述更为详细和通俗。

(8) 为满足初学者上机练习的需要,第二版增加了“附录D在Turbo C 2.0集成环境下运行C程序的方法”。内容包括Turbo C系统文件的组成及环境配置,上机运行C程序的步骤和命令,对常见编译出错及连接出错原因的分析及解决办法,运行出错时程序的动态调试方法以及运行由多个函数、多个源文件组成的C程序的方法。

此外,为便于读者练习解题时参考,我们已出版了一本《C 语言程序设计典型题解析及自测试题》(曹化工,秦友淑编著,西安:西北工业大学出版社,2001),本教材各章习题中的全部编程题均可在其中找到解答。此书对习题的解答及解题思路作了详细分析与说明,是学习 C 语言程序设计的一本很有帮助的辅助教材。

本书第一版的撰写和本次修订,既参考了 **The C Programming Language Second Edition** 和《C 程序设计语言》(王金汉等编著)等国内外多本著作,也融入了作者在华中科技大学计算机学院多年从事教学(C 语言程序设计、PASCAL 语言程序设计、面向对象的程序设计、数据库系统原理、计算机图形学等课程)和科研(C 编译程序和 PASCAL 编译程序等项目)的实践经验和体会,同时还吸收了同行专家学者们的意见和建议,因而使得该书具有概念准确、结构合理、层次清楚、叙述清晰、内容系统全面、适于教学和自学等显著特点。

本书适于各类高等院校计算机、通信、电子、自动化等信息学科各专业及其他有关专业用作本科及专科教材(包括远程教育、成人教育),也适合广大科技人员和研究生自学参考。本书用于不同对象时,建议任课教师对具体讲授内容剪裁如下:

- 对于少学时的一些高校或非 IT 类专业的本科学员,书中加“*”号的章节可以不讲授。
- 对于专科或培训班的学员,书中加“*”和“**”号的章节均可不讲授。
- 本书正文中配有大量实例,所有的实例程序均在 Turbo C2.0 上调试通过。有的实例侧重于说明语法和用法,有的则为了讲述算法或说明应用。任课教师根据需要可只选讲其中的一部分,其余留给学员自己阅读或上机练习。

学习和掌握 C 语言最有效的方法是实践。最初可阅读别人写好的程序(或函数),通过理解程序所要完成的任务(即程序功能),从中学习编程的方法和技巧;进而模仿编写功能类似的程序;最后逐步做到自己独立设计和编写完成指定任务的程序。此外,要真正掌握 C 语言和用 C 语言进行程序设计的方法和技术,必须上机练习,调试运行自己编写的程序。

本书初版的撰写和本次修订均得到了华中科技大学计算机学院领导和同事们的关心和支持,华中科技大学出版社的有关同志为其出版也付出了许多辛勤的劳动,原华中理工大学计算机专业学生曹鹏凌和机制专业学生曹鹏彬在本书初版的誊写整理工作中做了大量的工作,在此一并表示感谢!

由于作者水平所限,书中疏漏或错误之处恳请广大读者批评指正。

编 者

2002 年 5 月于华中科技大学(武汉)

目 录

第 1 章 引论	(1)
1.1 基础知识	(1)
1.1.1 计算机系统	(1)
1.1.2 算法及其表示	(3)
1.1.3 程序设计及程序设计语言	(5)
1.2 C 语言简介	(7)
1.2.1 C 语言的发展过程	(7)
1.2.2 C 语言的特点	(8)
1.3 C 程序的基本结构	(9)
1.4 C 语言的基本语法单位	(12)
1.4.1 标识符	(12)
1.4.2 关键字	(13)
1.4.3 分隔符	(14)
1.5 运行 C 程序的一般步骤	(14)
本章小结	(15)
习题一	(15)
第 2 章 基本数据类型和运算	(17)
2.1 基本数据类型	(17)
2.1.1 C 的数据类型分类	(17)
2.1.2 基本类型的名字及长度	(18)
2.2 常量和变量	(20)
2.2.1 常量的表示方法	(20)
2.2.2 符号常量	(27)
2.2.3 变量说明	(30)
2.3 运算符和表达式	(31)
2.3.1 概述	(31)
2.3.2 算术运算	(33)
2.3.3 关系运算	(35)
2.3.4 逻辑运算	(36)
2.3.5 自增和自减运算	(37)
2.3.6 位运算	(39)
2.3.7 赋值运算	(44)
2.3.8 条件运算	(46)
2.3.9 顺序求值运算	(48)

2.4	类型转换	(49)
2.4.1	类型转换的规则	(49)
2.4.2	类型转换的方法	(51)
2.5	枚举类型	(53)
	本章小结	(55)
	习题二	(56)
第3章	简单程序设计	(58)
3.1	流程结构和语句	(58)
3.2	基本的标准文件输入与输出函数	(59)
3.2.1	字符输入和输出函数 (getchar, putchar)	(60)
3.2.2	格式输出函数 (printf)	(61)
3.2.3	格式输入函数 (scanf)	(66)
3.3	编写简单的 C 程序	(74)
	本章小结	(77)
	习题三	(77)
第4章	流程控制	(79)
4.1	复合语句	(79)
4.2	if 语句	(81)
4.3	switch 语句	(84)
4.4	while 语句	(89)
4.5	for 语句	(96)
4.6	do-while 语句	(102)
4.7	多重循环	(109)
4.8	转移语句和标号语句	(113)
	本章小结	(120)
	习题四	(120)
第5章	函数与程序结构	(122)
5.1	C 程序的一般结构	(122)
5.2	函数定义与函数说明	(123)
5.2.1	函数定义	(123)
5.2.2	函数说明	(126)
5.3	函数调用与参数传递	(128)
5.3.1	函数调用	(129)
5.3.2	参数传递	(131)
5.3.3	参数数目可变的函数	(132)
5.4	变量的存储类型	(133)
5.4.1	存储类型区分符	(133)

5.4.2	自动变量	(134)
5.4.3	外部变量	(136)
5.4.4	静态变量	(139)
5.4.5	寄存器变量	(141)
5.4.6	变量的存储类型小结	(142)
* 5.5	递归函数与递归调用	(143)
* 5.6	编译预处理	(147)
5.6.1	宏替换	(147)
5.6.2	文件嵌入	(150)
5.6.3	条件编译	(150)
** 5.6.4	行控制	(153)
本章小结		(153)
习题五		(154)

第 6 章	数组	(155)
6.1	一维数组	(155)
6.1.1	数组的说明、引用和初始化	(155)
6.1.2	数组的运算和数组作为函数参数	(157)
6.2	字符数组	(163)
6.2.1	字符数组的说明和引用	(163)
6.2.2	字符数组的初始化	(164)
6.2.3	字符串处理函数	(165)
6.3	多维数组	(172)
6.3.1	多维数组的说明、引用和存储结构	(172)
6.3.2	多维数组的初始化	(174)
6.3.3	多维数组的运算	(175)
6.3.4	字符串数组	(178)
** 6.4	应用程序举例	(179)
本章小结		(184)
习题六		(184)

第 7 章	指针	(186)
7.1	指针的概念与指针的使用	(186)
7.1.1	指针的概念	(186)
7.1.2	指针的使用	(188)
7.2	指针运算	(191)
7.3	指针参数	(193)
7.4	数组的指针表示	(196)
7.4.1	一维数组的指针表示	(197)
7.4.2	数组参数的指针表示	(200)

7.4.3	字符数组的指针表示	(201)
7.4.4	多维数组的指针表示与数组的指针	(205)
7.5	指针数组	(213)
7.5.1	指针数组的说明及使用	(213)
7.5.2	指针变量的指针	(218)
*7.5.3	带参数的 main 函数	(222)
*7.6	函数的指针	(224)
7.7	指针函数	(230)
*7.8	复杂说明	(235)
	本章小结	(237)
	习题七	(238)
第 8 章	结构与联合	(240)
8.1	结构的说明和引用	(240)
8.1.1	结构的说明	(240)
8.1.2	结构的引用	(242)
8.2	结构的指针	(246)
8.3	结构和函数	(248)
8.4	结构数组	(252)
*8.5	结构和指针的应用	(261)
8.5.1	动态存储分配	(261)
8.5.2	链表	(263)
8.5.3	二叉树	(276)
*8.6	字段结构	(284)
8.7	联合	(287)
*8.8	用 typedef 定义类型名	(291)
	本章小结	(294)
	习题八	(294)
第 9 章	输入输出与低层接口	(297)
9.1	概述	(297)
9.2	流式文件输入输出	(298)
9.2.1	文件的打开与关闭	(298)
9.2.2	文件的读写操作	(302)
9.2.3	文件的随机存取	(314)
9.2.4	其他有关函数	(318)
9.2.5	文件处理程序实例	(319)
**9.3	输入输出的低层接口	(329)
9.3.1	文件的创建、打开、关闭和删除	(329)
9.3.2	文件的读、写	(330)

9.3.3 文件的随机存取	(332)
9.3.4 标准输入输出函数的实现方法	(333)
本章小结	(337)
习题九	(337)
附录 A ASCII 字符集	(339)
附录 B C 语言语法	(340)
附录 C 常用标准库函数	(343)
附录 D 在 turbo C 2.0 集成环境下运行 C 程序的方法	(346)
主要参考文献	(351)

第 1 章 引 论

1.1 基础知识

1.1.1 计算机系统

自 1946 年世界上第一台电子计算机诞生以来,计算机系统的结构有了很大的变化,其功能有了惊人的提高。但无论是大型机、中型机、小型机、微型机、个人计算机或计算机网络,一个完整的计算机系统均由硬件和软件两部分组成。计算机硬件是构成计算机系统的物理设备,是整个计算机系统的物质基础;计算机软件(包括使用和管理计算机所需的各种程序、有关的数据和文档)是计算机系统的灵魂。

1. 计算机硬件

计算机单机系统的硬件包括主机和外部设备。主机由中央处理单元(简称 CPU)和主存储器(简称内存)组成;外部设备包括输入设备、输出设备和外部存储设备。输入设备和输出设备通常简称为 I/O 设备;外部存储设备通常简称为外存,外存是内存的辅助存储设备。计算机系统的硬件组成如图 1.1 所示。

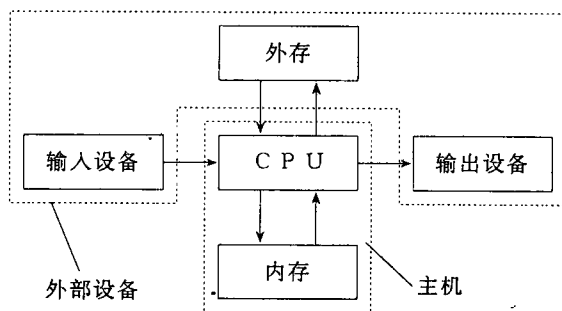


图 1.1 计算机系统的硬件组成

1) CPU

CPU 是计算机系统的核心,主要包括运算器和控制器两部分。运算器完成所有的算术运算和逻辑运算;控制器负责从内存取出指令和数据,并执行指令规定的动作以及将执行结果存入内存。

2) 内存

内存用于存储正在执行的程序和被程序处理的数据。内存由若干连续的存储单元组成,存储单元是一些仅具有 0 和 1 两种状态的电子元件。因此,每个这样的电子元件能够表示二进制数的一位,称为一个 bit。多个相邻的 bit(二进制位)就能够表示由 0 和 1 数字串组成的代

码，称为二进制代码，内存中的程序和数据就是以这种二进制代码形式存储的。

存储单元的编号称为存储单元的地址或内存地址。8个相邻的二进制位称为一个字节(Byte)，存储单元从0开始依次按字节编址。内存所包含的存储单元的总字节数称为内存的容量。CPU数据总线的bit数称为计算机的机器字长，例如：若机器字长为2字节则称为16位机，若机器字长为4字节则称为32位机。内存中存储的程序和数据称为存储单元的内容，从存储单元取出内容称为读内存，将程序或数据存入内存称为写内存。存取或读写内存统称为访问内存。访问内存必须由CPU通过存储单元的地址进行。

在高级语言程序中，存取内存中的数据一般是通过程序中的变量进行的，每个变量对应于一定数目的存储单元（以字节为单位），存储单元的内容就是变量的值。在C程序中，还可以通过存储单元的地址（即指针）访问内存中的数据。

作为一个程序员，必须了解内存的以下重要性质：

- 任何时刻计算机一经加电则存储单元中必有内容，但在未通过程序存入内容之前，这些内容仅是随机的，是无意义的信息。
- 任何时刻计算机一旦切断电源则存储单元中的一切内容立即消失，将不复存在。
- 读内存操作永远不改变内存中的内容，即如果仅仅执行读内存的操作，无论读多少次，存储单元中的内容保持不变。
- 写内存操作要破坏存储单元中原来的内容。任何时候一旦执行了写内存操作，则存储单元中原来的内容立即被新写入的内容所代替，原来的内容不复存在（称为“被冲掉”或被覆盖）。

3) 输入输出 (I/O) 设备

I/O设备用于人和计算机通信。输入设备用于将程序、各种形态的数据或文档送入计算机内存；输出设备用于将存储在计算机内存中的各种信息送到外部设备上显示、打印或保存。

输入设备分为字符设备和图形设备两类。最普通的字符输入设备是终端键盘，最普通的图形输入设备是鼠标。输出设备主要包括终端显示器、打印机和绘图仪，打印机和终端显示器可以兼作字符输出设备和图形输出设备，绘图仪只能用于输出图形。其中，终端键盘和终端显示器分别被系统约定为标准输入设备和标准输出设备。

4) 外存

外存既是信息的永久存储设备，同时又起着输入输出设备的作用。主要的外部存储设备有磁盘驱动器、磁带机和光盘驱动器，内存的信息可以通过这些外部存储设备写到磁盘、磁带或光盘等存储媒体上（其中光盘驱动器只能用于从光盘读信息到内存，而将信息从内存写到光盘上则要使用刻录机）。信息一经写入，只要存储媒体未被损坏或未重新写入新的信息，则信息将被永久保存（关掉电源以后信息也不会丢失）。另一方面，保存在外存上的任何程序必须从外存读入到内存才能被执行，任何数据必须从外存读入到内存才能被处理。换言之，当需要执行外存上的程序或需要处理外存上的数据时，必须首先将它们从外存输入到内存。

外存容量比内存容量大得多。外存容量一般以GB ($1\text{GB}=10^9\text{Bytes}$)为度量单位，而内存容量一般以MB ($1\text{MB}=10^6\text{Bytes}$)为度量单位。所以，外存是内存的辅助存储设备，可用于存放大量的程序、数据和文档。

2. 计算机软件

计算机系统的软件包括系统软件和应用软件。

1) 系统软件

系统软件由计算机厂家及软件公司提供, 是用于使用和管理计算机的各种程序及相关数据和文档的总称。系统软件主要包括操作系统、语言处理程序、各种实用程序和数据库管理系统等。

- 操作系统: 操作系统是最基本的核心系统软件, 用于管理计算机系统中各种软、硬件资源并控制各种程序的正常执行。例如 DOS, Windows, UNIX 等。

- 语言处理程序: 语言处理程序是将程序设计语言转变为计算机能够直接识别的机器语言的翻译程序。例如 IBM 汇编程序、BASIC 解释程序、C 编译程序和 PASCAL 编译程序等。

- 服务性程序: 服务性程序是为用户提供某种特定功能的实用程序。常用的服务性程序有文本编辑程序 (例如 DOS 环境下的 EDIT)、图文排版程序 (例如 Windows 环境下的 WORD), 用于内存、磁盘管理的工具程序 (例如 DOS, Windows 环境下的磁盘扫描程序 scandisk), 计算机病毒防治与杀病毒软件 (例如金山毒霸) 等。

- 数据库管理系统 (DBMS): DBMS 是能够集中存储和统一管理特定应用范围内用户的所有相关信息, 并具有最大的数据独立性、完整性、一致性和安全性的数据管理软件。

2) 应用软件

应用软件是为使用计算机处理各种实际问题而开发的具有专门用途的程序, 由于计算机的应用极为广泛, 因而应用软件多种多样。例如各种 MIS 系统、CAD 系统和过程控制系统等。

1.1.2 算法及其表示

1. 算法的基本特性

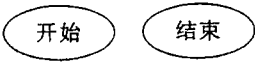
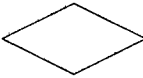
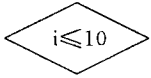
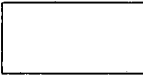
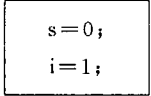
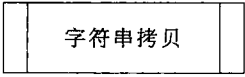
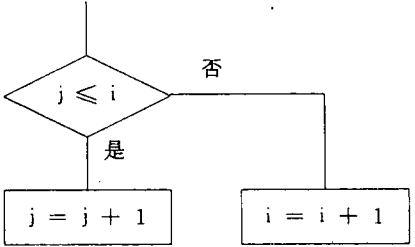
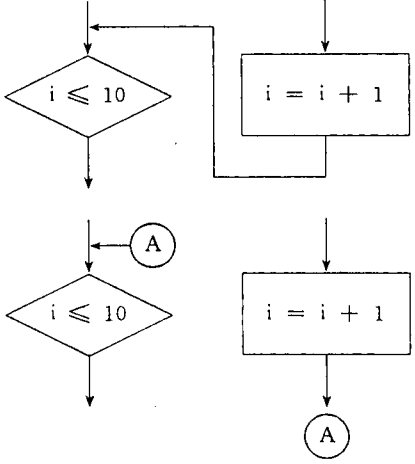
人们通常把计算机为解决某一问题所使用的方法与步骤称为算法。算法可分为两大类: 一类是科学计算领域用于处理数值数据的算法, 例如求定积分、解方程、求极限等; 另一类是数据处理领域用于处理非数值数据的算法, 例如分类排序、情报检索、绘图等。算法具有以下基本特性:

- 有穷性: 一个算法包含有限个步骤, 即算法经过有限步执行后必须终止。
- 确定性: 算法的每一步所规定的动作不能有两种以上的理解, 即算法每一步的动作不能有语义的二义性。
- 有输入: 一个算法有一个或多个输入。输入是执行算法时所需的信息, 包括被算法处理的数据和执行的控制信息。
- 有输出: 一个算法有一个或多个输出。输出是算法执行的结果。
- 有效性: 算法每一步所规定的动作都能够有效地执行。例如: 两个数相除, 当除数为零时则不能执行除运算, 即算法所规定的动作无法执行。

2. 算法的表示

用一种适当的方式来描述算法称为算法的表示。算法有多种表示方式, 本书采用最基本、最常用的两种方式, 即自然语言和传统流程图来表示算法。传统流程图通常简称流程图或框图。流程图用图形符号 (框和线) 表示算法的每一步及各步之间的联系, 流程图常用符号及其含义如表 1.1 所示。

表 1.1 流程图常用符号

符 号	符号名称	意 义	例
	起止框	表示算法的开始或结束	
	判断框	表示判断选择：根据框中条件从两种可选动作中选一执行	
	处理框	表示按顺序执行的处理	
	调用框	表示调用函数	
	流程线	表示两个步骤相邻，且执行顺序由箭尾一方到箭头一方。对于自上而下和自左而右的顺序，箭头可省缺	
	连接点	连接点必须以相同的形式成对出现，用于表示一条流程线被断开后的两个端点	下面两种表示是等价的： 

注：表 1.1 所列符号摘自国际信息处理标准 ISO8631-1986E。

3. 算法表示举例

问题：求 $s = \sum_{n=1}^{10} n^n$

下面分别用自然语言和流程图表示该级数求和的解题算法。

1) 用自然语言表示

- (1) 用变量 s 存放各项的累加和，置初值为 0；用变量 i 作项数计数器，置初值为 1。
- (2) 如果 $i \leq 10$ 则计算 s ，否则转步骤 (3)。

计算 s ：

- (2)-1 用变量 j 作每一项的累乘次数计数器，置初值为 1；
- (2)-2 用变量 ai 存放第 i 项的累乘积，置初值为 1；
- (2)-3 如果 $j \leq i$ 则计算第 i 项，否则转步骤(2)-4；

计算第 i 项：

- (2)-3-1 $ai = ai * i$ ；
- (2)-3-2 $j = j + 1$ ，转步骤(2)-3；
- (2)-4 将第 i 项加到累加和中去：
- (2)-4-1 $s = s + ai$ ；
- (2)-4-2 $i = i + 1$ ，转步骤(2)；

- (3) 输出 s ，结束。

2) 用流程图表示 (见图 1.2)

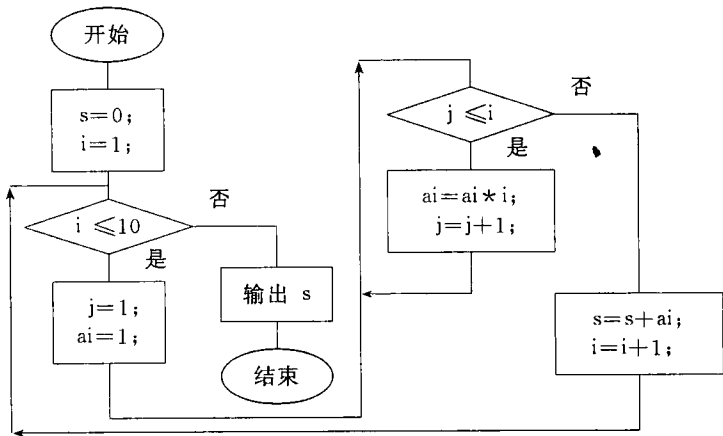


图 1.2 计算 $s = \sum_{n=1}^{10} n^n$ 的算法流程图

显然，用流程图表示算法比用自然语言更简明、直观。

1.1.3 程序设计及程序设计语言

程序是用程序设计语言表示的计算机解题算法或计算机解题任务。程序设计是将解题任务转变成程序的过程，一般包括：分析问题；确定算法（对复杂算法需画出流程图）；用选定的程序设计语言编写源程序；上机调试、运行程序等基本步骤。程序设计语言是计算机能够理解的、用于人和计算机通信的语言，程序设计语言可分为低级语言、高级语言、面向对象

的语言和专用语言四类。

1. 低级语言

低级语言又分为机器语言、符号语言和汇编语言。机器语言用二进制代码表示机器指令和数据，机器语言程序能够直接被机器理解和执行，虽然这种语言程序效率高，但编程繁琐，且不利于记忆和阅读，因而程序维护困难。符号语言用符号代替二进制代码表示机器指令，而汇编语言进而用符号来表示指令和数据的内存地址。现在人们用低级语言编程通常指用汇编语言编程。

用汇编语言编写的程序称为汇编语言源程序。汇编语言程序必须被转换为机器语言程序才能被计算机理解和执行，完成这种转换任务的系统软件称为汇编程序；这种转换过程称为汇编。低级语言是面向机器的，用低级语言写的程序效率高，但没有可移植性，即不能从一个机器系统上移到另一个机器系统上运行。此外，用机器语言编写程序要求程序员必须懂得具体机器系统的硬件结构（指令系统）。

2. 高级语言

高级语言是类似于人类自然语言英语和数学语言的程序设计语言，例如 C 语言、PASCAL 语言。用高级语言编写的程序称为高级语言源程序，简称源程序或程序。例如 C 源程序或 C 程序，PASCAL 源程序或 PASCAL 程序。

与汇编语言源程序类似，高级语言源程序也必须被转换成机器语言程序才能够被机器理解和执行，完成这种转换任务的系统软件称为编译程序。例如 C 编译程序、PASCAL 编译程序。将高级语言源程序转换成机器语言程序的过程称为编译。

高级语言是面向解题过程的，语言本身与具体机器系统无关，因而用高级语言编写的应用程序可移植性好。编译程序是一种语言的具体实现，编译程序与具体机器系统有关，人们常将一个编译程序称为某种语言的一个版本。同一种语言的不同版本不完全相同，在使用一种具体的高级语言及其编译程序开发软件时，必须参考与编译程序配套的有关资料。本书阐述的内容严格遵循 C 语言标准，对于大多数 C 编译程序具有通用性。考虑到上机环境尽量简单以便于学习和练习，书中所有例题均在 Turbo C 2.0 上通过运行。

3. 面向对象的语言

典型的、目前应用最普遍的面向对象的语言是 C++ 语言，C++ 语言是 20 世纪 80 年代为适应中型和大型复杂软件的开发和维护而发展起来的面向对象的程序设计语言，其目标是程序员的程序开发活动提供一个优良的程序设计环境，以产生模块化程度高、可重用性和可维护性好的程序。20 世纪 90 年代推出的 Java 语言是一种适合于分布式计算的新型面向对象的程序设计语言，Java 语言主要用于 Internet 应用程序的开发和编制。

4. 专用语言

专用语言是专门为特定的应用领域而设计的计算机程序设计语言。例如，计算机辅助设计（CAD）系统中使用的绘图语言，数据库管理系统（DBMS）中的数据查询语言 SQL、军事应用领域的 ADA 语言、商业应用领域的 COBOL 语言等。专用语言是面向问题的语言，它比高级语言更抽象，描述能力比高级语言更强。用专用语言编程不需要指出“如何做”，只需