

21世纪

高等院校计算机科学与技术规划教材



# C语言

# 程序设计教程

朱承学 主 编  
蒋外文 主 审



中国水利水电出版社  
www.waterpub.com.cn

21 世纪高等院校计算机科学与技术规划教材

# C 语言程序设计教程

朱承学 主编

刘莉平 严晖 曹岳辉 贺超 副主编

蒋外文 主审

中国水利水电出版社

## 内 容 提 要

C 语言作为一种结构化的计算机程序设计语言,既具有高级语言的特点,又具有低级语言的功能,在当今软件开发领域有着十分广泛的应用。

本书通过循序渐进地解析程序代码,阐明 C 语言结构的语法和正确使用,系统地讲述了 C 语言程序设计的基本方法和技巧,并揭示出其应用的基础逻辑。

全书选材经典,内容丰富,阐述清晰,层次分明。讲述力求理论联系实际、循序渐进,注重培养读者分析问题和程序设计的能力,注重培养良好的程序设计风格和习惯。

本书的主要内容包括:C 语言的基本语法和概念、数据类型及应用技巧、C 语言模块化程序设计的方法、文件的基本概念和应用以及综合程序设计基础。

为了配合本书的学习及适应程序设计课程教学的三个环节,作者还编写了与本书配套的《C 语言程序设计实验指导与习题解析》及《C 语言程序设计实践训练教程》,可供读者参考使用。

本书可作为高等院校计算机程序设计教学用书,也可作为从事计算机应用的科技人员的参考书和培训教材。本书配有 CAI 教学课件,可到中国水利水电出版社网站(<http://www.waterpub.com.cn>)下载。

## 图书在版编目(CIP)数据

C 语言程序设计教程 / 朱承学主编. —北京:中国水利水电出版社,2004  
(21 世纪高等院校计算机科学与技术规划教材)

ISBN 7-5084-2129-9

I. C… II. 朱… 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(2004)第 076498 号

书 名	C 语言程序设计教程
主 编	朱承学
副 主 编	刘莉平 严晖 曹岳辉 贺超
主 审	蒋外文
出版 发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: <a href="http://www.waterpub.com.cn">www.waterpub.com.cn</a> E-mail: <a href="mailto:mchannel@263.net">mchannel@263.net</a> (万水) <a href="mailto:sales@waterpub.com.cn">sales@waterpub.com.cn</a> 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 20 印张 450 千字
版 次	2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷
印 数	0001—5000 册
定 价	26.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

# 21 世纪高等院校计算机科学与技术规划教材

## 编委会名单

**主任** 陈火旺 中国工程院院士 国防科技大学教授

**委员** 周立柱 清华大学计算机系主任

杨放春 北京邮电大学计算机科学与技术学院院长

杨学军 国防科技大学计算机学院院长

徐晓飞 哈尔滨工业大学计算机科学与技术学院院长

李仁发 湖南大学计算机与通信学院院长

卢正鼎 华中科技大学计算机学院院长

李志蜀 四川大学计算机学院院长

戴居丰 天津大学信息学院、软件学院院长

蒋昌俊 同济大学计算机科学与工程系主任

何炎祥 武汉大学计算机学院院长

周兴社 西北工业大学计算机系主任

陈志刚 中南大学信息科学与工程学院副院长

姜云飞 中山大学软件学院院长

周昌乐 厦门大学软件学院院长

齐勇 西安交通大学计算机科学与技术系主任

赵书城 兰州大学计算机学院院长

孟祥旭 山东大学计算机学院院长

# 前 言

C 语言是国际上广泛流行且有着广泛应用的计算机程序设计高级语言。它既可用于开发系统软件，也可用于开发应用软件。

C 语言简洁、紧凑，数据类型丰富，表达能力强，运算丰富而灵活，流程控制方便，程序结构性和可读性好，体现了结构化程序设计的思想，适合于培养良好的编程风格和优秀的程序设计技术的训练。是继 PASCAL 语言之后的又一门优秀的课程教学语言，并且是教学需要与实际应用相结合的一门语言。

作为程序设计课程，其根本目的是让读者掌握基本的程序设计的概念、方法和技巧，而程序设计语言本身只是一个编制程序的工具。本书在详细介绍、解析 C 语言语法知识的同时，由浅入深、循序渐进，通过大量涉及科学计算、数据处理等方面的例题，充分展示了计算机解决问题的思想和方法，突出了程序设计的基本方法的阐述。

本书是作者在多年从事程序设计语言及计算机相关课程的教学实践，在多次编写教案、教材、教辅资料的基础上经过精心整理、组织而成。同时，作者多年一直参与全国计算机等级考试的组织、辅导与评卷工作，对全国计算机等级考试的大纲有透彻的理解，并将其精髓贯穿于本书之中。因此，本书对于通过全国计算机等级考试也是一本不可多得的教程。

为了配合程序设计课程的教学和读者的学习，我们还编写了《C 语言程序设计实验指导与习题解析》及《C 语言程序设计实践训练教程》，作为本书的配套参考书，满足了程序设计课程三个教学环节的需要。

本书由中南大学朱承学主编，刘莉平、严晖、曹岳辉、贺超任副主编，由中南大学蒋外文教授主审。朱承学编写第 1、2、9 章，并负责全书的统稿；刘莉平编写第 3、4、5、8 章；严晖编写第 6、7、10 章；曹岳辉、贺超编写第 11、12、13 章；裘嵘参与了第 2 章部分内容的编写；李小兰参与了第 6 章部分内容的编写；王新辉参与了第 8、9 章部分内容的编写；徐彤参与了第 13 章部分内容的编写。

在本书的写作过程中，得到了蒋外文教授和刘卫国教授的热情支持和指导，在此表示衷心感谢。

本书不当之处敬请广大读者不吝赐教。

编 者

2004 年 6 月

# 目 录

## 前言

<b>第 1 章 程序设计语言概述</b> .....	1
1.1 程序设计概述 .....	1
1.1.1 软件、程序与程序设计 .....	1
1.1.2 算法与算法描述 .....	2
1.1.3 程序设计风格 .....	5
1.2 C 语言简介 .....	6
1.2.1 C 语言出现的历史背景 .....	6
1.2.2 C 语言的特点及应用 .....	7
1.2.3 简单的 C 语言程序介绍 .....	7
1.3 C 语言的上机操作步骤 .....	8
1.3.1 Turbo C 2.0 集成开发环境简介 .....	9
1.3.2 Turbo C++ 3.0 集成开发环境简介 .....	11
<b>第 2 章 C 语言的基本数据类型与表达式</b> .....	12
2.1 C 语言的语法基础 .....	12
2.1.1 C 语言的字符集 .....	12
2.1.2 C 语言的标识符 .....	12
2.1.3 C 语言的关键字 .....	13
2.2 C 语言的基本数据类型 .....	13
2.2.1 数据类型概述 .....	13
2.2.2 整型 .....	14
2.2.3 实型 .....	15
2.2.4 字符型 .....	16
2.3 常量与变量 .....	16
2.3.1 常量与符号常量 .....	16
2.3.2 变量与变量说明 .....	17
2.3.3 变量的初始化与赋值 .....	17
2.4 运算符与表达式 .....	18
2.4.1 算术运算符与算术表达式 .....	18
2.4.2 关系运算符与关系表达式 .....	20
2.4.3 逻辑运算符与逻辑表达式 .....	21
2.4.4 赋值运算符与赋值表达式 .....	23

2.4.5	条件运算符与条件表达式 .....	23
2.4.6	逗号运算符与逗号表达式 .....	24
2.4.7	位运算符 .....	25
2.4.8	其他运算符 .....	27
2.5	数据类型转换与各种类型数据之间的混合运算 .....	28
2.5.1	表达式中的类型转换 .....	28
2.5.2	强制转换 .....	30
<b>第 3 章</b>	<b>顺序结构程序设计 .....</b>	<b>31</b>
3.1	C 语句概述 .....	31
3.1.1	简单语句 .....	31
3.1.2	复合语句 .....	33
3.2	程序的三种基本结构 .....	34
3.2.1	顺序结构 .....	34
3.2.2	选择结构 .....	35
3.2.3	循环结构 .....	35
3.3	输入与输出语句 .....	36
3.3.1	字符输入与输出函数 .....	36
3.3.2	格式输出函数 .....	38
3.3.3	格式输入函数 .....	42
3.4	顺序结构程序设计举例 .....	46
<b>第 4 章</b>	<b>选择结构程序设计 .....</b>	<b>48</b>
4.1	if 语句 .....	48
4.1.1	if 语句的三种形式 .....	48
4.1.2	if 语句的嵌套 .....	53
4.2	switch 语句 .....	55
4.3	选择结构程序设计举例 .....	59
<b>第 5 章</b>	<b>循环结构程序设计 .....</b>	<b>62</b>
5.1	while 语句 .....	62
5.2	do-while 语句 .....	66
5.3	for 语句 .....	68
5.4	goto 语句以及用 goto 语句构成的循环 .....	74
5.5	break 语句和 continue 语句 .....	75
5.6	循环的嵌套 .....	78
5.7	循环结构程序设计举例 .....	80
5.8	循环结构程序设计注意事项 .....	83
<b>第 6 章</b>	<b>函数与模块化程序设计基础 .....</b>	<b>86</b>
6.1	概述 .....	86

6.1.1	模块与函数 .....	86
6.1.2	模块设计原则 .....	89
6.2	函数的定义 .....	89
6.2.1	标准库函数与头文件 .....	90
6.2.2	用户自定义函数 .....	91
6.3	函数的调用 .....	93
6.3.1	函数的一般调用与声明 .....	93
6.3.2	形参与实参 .....	96
6.3.3	函数的嵌套调用 .....	98
6.3.4	函数的递归调用 .....	99
6.4	变量的作用域及存储特性 .....	103
6.4.1	变量的作用域 .....	103
6.4.2	变量的存储特性 .....	106
<b>第7章</b>	<b>编译预处理命令 .....</b>	<b>111</b>
7.1	宏定义 .....	111
7.1.1	简单宏定义 .....	111
7.1.2	带参数的宏定义 .....	113
7.2	文件包含处理 .....	116
7.2.1	#include 命令格式 .....	117
7.2.2	#include 命令的嵌套使用 .....	118
7.3	条件编译 .....	118
7.3.1	控制条件为常量表达式的条件编译 .....	119
7.3.2	控制条件为定义标识符的条件编译 .....	120
7.3.3	#undef 命令行 .....	122
<b>第8章</b>	<b>数组 .....</b>	<b>123</b>
8.1	一维数组 .....	123
8.1.1	一维数组的定义 .....	124
8.1.2	一维数组元素的引用 .....	124
8.1.3	一维数组的初始化 .....	126
8.2	二维数组 .....	129
8.2.1	二维数组的定义 .....	130
8.2.2	二维数组元素的引用 .....	131
8.2.3	二维数组的初始化 .....	131
8.2.4	二维数组元素值的输入与输出 .....	132
8.3	字符数组与字符串 .....	133
8.3.1	字符数组的定义 .....	133
8.3.2	字符数组的存储结构 .....	134



8.3.3	字符数组的初始化 .....	134
8.3.4	字符数组与字符串的输入与输出 .....	135
8.3.5	字符串处理函数 .....	138
8.4	数组与函数 .....	143
8.4.1	数组元素作为函数参数 .....	143
8.4.2	数组名作为函数参数 .....	144
8.5	数组程序设计举例 .....	146
<b>第9章</b>	<b>指针 .....</b>	<b>154</b>
9.1	指针的概念 .....	154
9.2	指针变量 .....	156
9.2.1	指针变量的定义 .....	156
9.2.2	指针变量的运算 .....	158
9.2.3	指向指针的指针 .....	161
9.3	指针与函数 .....	163
9.3.1	指针作为函数参数 .....	163
9.3.2	指针函数 .....	168
9.3.3	指向函数的指针 .....	170
9.4	指针与数组 .....	175
9.4.1	指针与一维数组 .....	176
9.4.2	指针与二维数组 .....	181
9.4.3	指针与字符串 .....	183
9.4.4	指针数组与命令行参数 .....	185
<b>第10章</b>	<b>结构体与链表 .....</b>	<b>191</b>
10.1	结构体类型的定义与变量说明 .....	192
10.1.1	结构体类型的定义 .....	192
10.1.2	结构体类型变量的定义 .....	192
10.1.3	结构体类型的嵌套 .....	194
10.2	结构体类型变量的引用与初始化 .....	195
10.2.1	结构体类型变量的引用 .....	195
10.2.2	结构体类型变量的初始化 .....	197
10.3	结构体类型与数组 .....	198
10.3.1	结构体数组的定义 .....	198
10.3.2	结构体数组的初始化与结构体数组元素的引用 .....	198
10.4	结构体类型与指针 .....	202
10.4.1	指向结构体变量的指针 .....	202
10.4.2	指向结构体数组的指针 .....	203
10.5	结构体与函数 .....	205

10.5.1	结构体变量作为函数的参数 .....	205
10.5.2	指向结构体变量的指针作为函数的参数 .....	206
10.5.3	结构体类型的函数 .....	208
10.6	链表 .....	209
10.6.1	链表概述 .....	210
10.6.2	链表的建立 .....	213
10.6.3	链表的访问 .....	217
10.6.4	链表的插入操作 .....	219
10.6.5	链表的删除操作 .....	222
<b>第 11 章</b>	<b>共用体及其他数据类型 .....</b>	<b>226</b>
11.1	共用体 .....	226
11.1.1	共用体的概念 .....	226
11.1.2	共用体变量的引用方式 .....	227
11.1.3	共用体变量的应用 .....	228
11.2	枚举类型 .....	230
11.2.1	枚举类型的定义 .....	230
11.2.2	枚举类型的应用 .....	231
11.3	用 typedef 定义类型 .....	232
11.4	位段结构 .....	233
<b>第 12 章</b>	<b>文件 .....</b>	<b>238</b>
12.1	概述 .....	236
12.1.1	文件的定义与分类 .....	236
12.1.2	数据文件的存储形式 .....	237
12.1.3	标准文件系统与非标准文件系统 .....	237
12.1.4	文件的存取方式 .....	237
12.2	标准文件操作 .....	238
12.2.1	标准文件类型指针 FILE .....	238
12.2.2	标准文件的打开 .....	238
12.2.3	标准文件的关闭 .....	240
12.2.4	标准文件的读与写 .....	240
12.2.5	文件的随机读写 .....	249
12.2.6	文件的检测 .....	250
12.3	非标准文件操作 .....	252
12.3.1	非标准文件的打开和关闭 .....	252
12.3.2	非标准文件的读与写 .....	253
<b>第 13 章</b>	<b>C 语言综合应用程序设计 .....</b>	<b>255</b>
13.1	多模块程序设计基础 .....	255

13.1.1	多模块程序的建立 .....	255
13.1.2	程序模块之间的通信 .....	258
13.2	图形处理基础 .....	259
13.2.1	图形程序运行的条件 .....	259
13.2.2	基本图形处理函数及基本图形处理方法 .....	260
13.3	与操作系统的接口 .....	272
13.3.1	DOS 简介 .....	272
13.3.2	操作系统接口库函数 .....	272
13.3.3	鼠标器及其应用 .....	276
13.4	C 程序中的汉字显示技术 .....	282
13.4.1	汉字显示的基本原理 .....	282
13.4.2	中文 DOS 下显示汉字 .....	286
13.4.3	西文 DOS 下显示汉字 .....	287
13.4.4	汉字的缩放 .....	291
<b>附录 I</b>	<b>ASCII 字符编码一览表 .....</b>	<b>295</b>
<b>附录 II</b>	<b>运算符的优先级别和结合方向一览表 .....</b>	<b>296</b>
<b>附录 III</b>	<b>Turbo C 常用库函数 .....</b>	<b>297</b>
<b>附录 IV</b>	<b>图形模式表 .....</b>	<b>305</b>
<b>附录 V</b>	<b>鼠标器功能及功能编号 .....</b>	<b>306</b>
<b>附录 VI</b>	<b>鼠标器驱动程序常用功能及参数 .....</b>	<b>307</b>
<b>参考文献</b>	<b>.....</b>	<b>308</b>

# 第 1 章 程序设计语言概述

## 知识点

- 软件、程序、程序设计、程序设计语言
- 算法、算法描述
- C 语言的特点及上机操作

## 本章导读

本章主要介绍了与程序设计相关的一些基本概念,如软件、程序、程序设计及程序设计语言,C语言的产生、特点及C语言的上机操作过程等内容。通过本章的学习,读者可以对计算机程序和学习计算机程序设计课程的意义有一个总体认识,并对程序设计的方法和步骤有初步的认识。

## 1.1 程序设计概述

### 1.1.1 软件、程序与程序设计

电子计算机自 1946 年诞生至今的短短几十年,对人类历史进程的贡献是无法估量的。毋庸置疑,计算机在今后的人类生产生活中的应用领域将越来越广,作用将越来越大,它是人们生产生活中必不可缺少的工具。

计算机系统由计算机硬件和软件组成。计算机硬件是计算机系统物理部分,是计算机工作的物质基础。计算机软件是指计算机所使用的各种程序,它包括计算机系统中配置的各种系统软件、各种程序设计语言和使用这些语言编制的各种应用程序等。

软件与程序是两个不同的概念,软件设计与程序设计也是两个不同的概念。软件的一个显著特点是规模庞大,而且软件的设计与开发已不再是几个程序员在一起编写程序了。自 1968 年北大西洋公约组织的计算机科学家国际会议讨论“软件危机”并正式提出“软件工程\*”这个名词以来,软件开发就作为一门新兴的工程学科诞生了。软件工程是指导计算机软件开发和维护的工程学科。它采用工程的概念、原理、技术和方法来开发与维护软件,它把先进的管理技术和计算机技术结合起来。采用软件工程的方法来进行软件开发,其目的是为了系统地、分阶段地、有条不紊地从抽象的逻辑概念逐步发展到具体的物理实现;

---

\*软件工程是软件工程学的简称,是从二十世纪六十年代发展起来的一门新兴学科,是软件开发工作者必修的一门课程。

确保开发的软件与实际需要相符合,减少重复劳动;提高软件的可维护性,减少软件维护代价。因此,软件开发的成果不仅要提交程序,还要提交在完成每个阶段任务时采用的结构分析设计技术和适当的辅助工具,以及各个阶段的结论与建议,这就是所谓的文档资料。

程序设计是作为软件工程的一个步骤存在的。程序是把软件设计的结果用程序设计语言表达出来,而这个过程就是所谓的设计。程序设计的目的是实现人和计算机的交流,指挥计算机按照人的意志正确工作。程序的质量主要取决于软件设计的质量。程序设计语言是人和计算机通信的最基本的工具。同时,程序设计语言的特性和编码的方法也会对程序的可靠性、可读性、可测性和可维护性产生深远的影响。因而,选择合适的程序设计语言是程序设计之前的一项重要工作。

程序设计语言从其发展历史及功能来看,可分为机器语言、汇编语言(也称低级语言)、算法语言(也称高级语言)、面向任务和面向对象的语言。C语言也有人把它称为中级语言,因为它把高级语言的一些特性同低级语言(即汇编语言)的能力结合起来了。

### 1.1.2 算法与算法描述

无论使用计算机解决一个简单问题还是一个复杂问题,都必须事先编制好相应的程序,然后计算机按照相应的程序执行有关操作,得到所需结果。简单地说,求解一个实际问题,一般要经过以下几个步骤:

(1) 分析问题。确定已知数据,输出结果,所需硬件和软件环境,并提出求解问题的数学模型与计算方法。

(2) 算法分析。确定求解问题的数据对象和求解步骤,即给出算法。

(3) 程序设计。使用合适的程序设计语言把已确定的数据和算法描述出来,即编写程序或称为编码。

(4) 程序调试。将编制的程序输入计算机系统中,进行编译和试运行,修正其中可能存在的语法和逻辑错误。

(5) 运行与维护。经调试后的程序基本正确,可投入使用。但程序中的有些错误并不一定在调试中被发现,用户在使用过程中会发现一些隐藏的错误,需要维护人员进行修改。同时,技术的更新,也需要维护人员对该程序进行改进。

#### 1. 算法与算法的基本特性

算法(Algorithm),简单地说,是为解决一个特定问题而采取的确定的、有限的方法和步骤。也就是对特定问题求解步骤的一种描述,是指令的有限序列,其中每一条指令表示一个或多个操作。一般而言,一个算法应具有下列5个基本特性:

(1) 有穷性。一个算法必须在执行有穷步以后结束,且每一步都应在有穷时间内完成。

(2) 确定性。算法中的每一条指令必须有确切的含义,不存在二义性。

(3) 可行性。算法中所描述的操作是可以通过已经实现的基本运算执行有限步实现。

(4) 输入。一个算法有零个或多个原始数据的输入,这些原始数据来源于某个特定的数据集。

(5) 输出。一个算法有零个或多个数据的输出。这些输出数据与输入数据有着某些

特定的关系。

## 2. 算法的描述

一般可以使用以下几类方法描述算法：

(1) 自然语言。自然语言即人们日常交流的语言，例如汉语、英语等。用自然语言来描述算法，对用户来说，可读性及可理解性强，并且方便用户之间的相互交流。但目前还不能将用自然语言描述的算法直接提交给计算机进行处理。

【例 1-1】输入 10 个整数，求出其中的最大值。

算法如下：

- 1) 取最小整数-32768 设为最大值，并把它放在变量 max 中。
- 2) 计数器的值置 0。
- 3) 输入一个整数 x，且计数器加 1。
- 4) 若 x 大于 max，则 max 的值改为 x；否则 max 的值不变。
- 5) 若计数器的值小于 10，则重新执行第 3 步~第 5 步；否则执行第 6 步。
- 6) 输出 max 即最大值。/\*算法结束\*/

显然，这个算法就是一个“打擂台”的过程。第 1 个上擂台的人就是擂主，以后依次有人上来打擂台，每次留在擂台上人的就是新的擂主；当每个人都参与了一次比赛以后，最后留在擂台上的人就是最后的擂主。因此，上述算法第①步中的 max 的初始值也可设置为 10 个整数中的任意一个，为方便起见，一般取第 1 个。请读者自己完成相应的算法。

(2) 框图语言。框图语言是用图形来表示算法，即用一些几何图形框来代表各种不同性质的操作。目前常用的框图语言有传统流程框图语言及 N-S 结构流程图。

传统程序流程图的优点是方便、直观、灵活、流程清晰、无“歧义性”。但占用面积大，且有一个致命的缺点，它允许流程线指向任意一个框。对程序流程不加任何限制，就会导致算法的逻辑难以让人理解。这种描述方法的可读性、可靠性及可维护性差。

ANSI 规定的一些常用传统流程图符号如图 1-1 所示。

提示：ANSI (American National Standards Institute) 即美国国家标准协会。

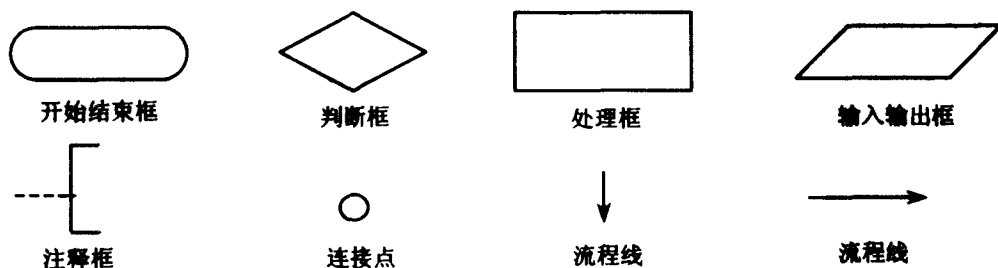


图 1-1 传统程序流程框图的常用符号

【例 1-2】用传统程序流程图描述例 1-1 的算法。流程图如图 1-2 所示。

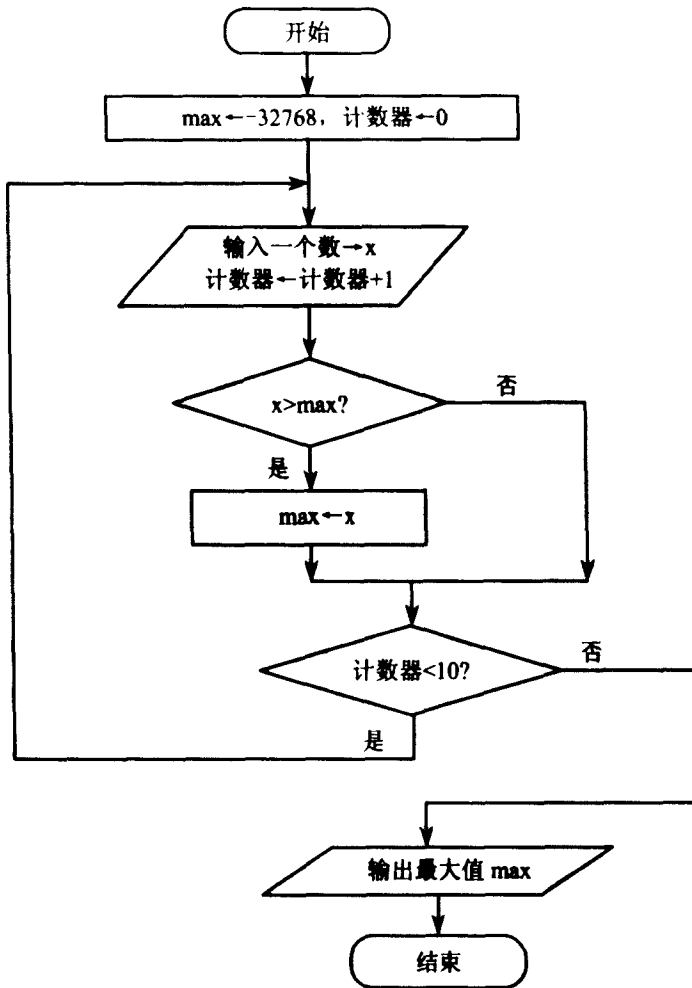


图 1-2 例 1-1 的算法的传统程序流程框图

1973 年美国学者 I.NASS 和 B.SHNEIDERMAN 提出了一种新的流程图表示方法。这种流程图中完全去掉了流程线，全部算法写在同一个矩形框内，一个矩形框中又可包含其他的矩形框。这种流程图即所谓的 N-S 结构流程图。

N-S 结构流程图的几种基本结构如图 1-3 所示。

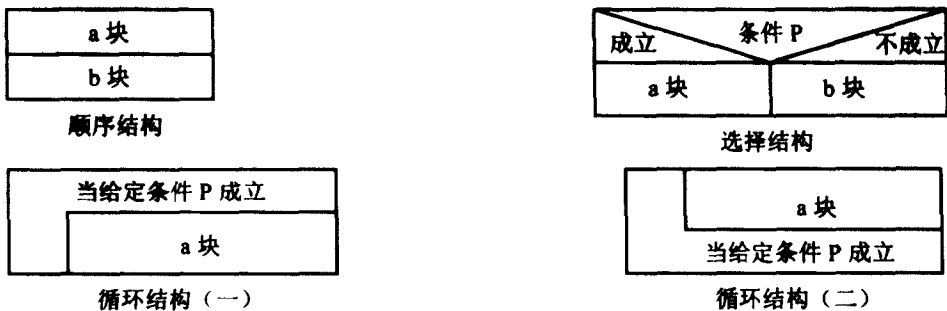


图 1-3 N-S 结构流程图的几种基本结构

【例 1-3】用 N-S 结构流程图描述例 1-1 的算法。算法如图 1-4 所示。

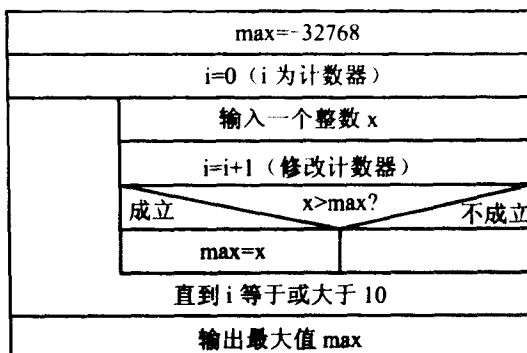


图 1-4 例 1-1 的算法的 N-S 结构流程图

(3) 伪代码。伪代码 (Pseudocode), 也称伪码, 它包括高级程序设计语言的三种基本控制结构和自然语言成分, 它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。伪代码描述算法时, 不用图形, 可读性好, 并方便算法转换为程序。对于 C 语言程序设计中的伪代码描述常称为类 C 语言描述。

【例 1-4】用类 C 语言描述例 1-1 的算法。

```

Status function_max( )
{
    /*输入 10 个整数, 求其中的最大值*/
    /*max 存放最大值, i 作为计数器*/
    max=-32768;i=0; /*先将最小整数存放在 max 中, 计数器初始化*/
    while(i<10)
    { 输入 x;
      if (x>max) max=x;
      i++;
    }
    输出 max 的值
} /*function_max */
  
```

### 1.1.3 程序设计风格

程序设计风格指的是编写程序的风格。一个好程序不只是一个正确的程序, 还要有良好的程序设计风格。良好的程序设计风格是程序设计成功的保障。遵循通常的编程原则, 有利于编写出有效的、结构清晰的和易于理解的程序。这里给出编写程序时应注意的几个方面。

#### 1. 选用合适的常量标识符

在程序中将一些值相对确定的数据定义为常量, 其优点有: 合适的名字 (即标识符) 可增加可读性; 集中在宏定义部分便于查找; 便于适当的时候对常量的值进行修改。同时, 常量最好用大写字母, 便于与变量区分, 并可避免逻辑错误。



## 2. 选用有实际意义的标识符作为变量名

例如，在程序中用 `radius`，`diameter`，`area` 分别表示圆半径、直径及面积，则便于对程序的理解，增加了程序的可读性。又如，用 `student_name`，`worker_number`，`gren_gz` 分别表示学生姓名、工人数量及工人工资；同时，这种用下划线来划分英文词汇或汉语拼音的形式表示标识符也很实用。在 C 语言中，变量名一般用小写字母。

## 3. 程序书写的缩进规则

根据语句的并列关系及包含关系，将包含关系中的被包含语句缩进行书写，一般通过按“Tab”键来完成。按缩进规则书写的程序中语句间的逻辑结构清晰，便于阅读程序。

## 4. 适当的注释

注释是一种便于阅读和理解程序的信息，它为程序员本人及他人提供了帮助。C 语言的注释语句是用“/\*”和“\*/”中间加注释内容表示。

## 5. 适当的交互性

在程序的适当时候加入一些输出提示语句，提示用户当前的状态或告诉用户响应方法。例如：

```
printf("Please enter radius of circle: ");  
scanf("%f",&radius);
```

如果没有前面的 `printf` 语句，则用户就有可能不知如何使程序继续执行。

# 1.2 C 语言简介

## 1.2.1 C 语言出现的历史背景

C 语言是国际上广泛流行且有着广泛应用的计算机程序设计高级语言。它既可用于开发系统软件，也可用于开发应用软件。

早期的系统软件如操作系统等主要是用汇编语言编写的。汇编语言作为一种低级语言，依赖于计算机硬件，程序的可读性和可移植性较差。要提高程序的可读性和可移植性，最好采用高级语言，但一般的高级语言又难以实现汇编语言的某些功能。因此，人们就试图找到一种既具有高级语言特性，又具有低级语言特性的语言，并集它们的优点于一身。于是，C 语言就这样逢时而生了。

C 语言是在 B 语言的基础上发展起来的。二十世纪七十年代初，美国贝尔实验室<sup>\*</sup>的 D.M.Ritchie 在 B 语言的基础上设计出 C 语言。1978 年，D.M.Ritchie 和 B.W.Kernighan 合著 *The C Programming Language*，该名著成为世界广泛使用的 C 语言版本的基础，并被称为标准 C。1983 年，美国国家标准化协会（ANSI）制定了一个 C 语言标准，称为 83 ANSI C。1987 年，ANSI 又公布了新标准即 87 ANSI C。1990 年，国际标准化组织（ISO）确认

---

<sup>\*</sup> 贝尔实验室（Bell Labs），主要从事通信与控制器件系统的研究开发，创建于 1925 年，美国电气公司和电报电话公司（AT&T）共同所有。它以电话发明家贝尔命名，至今已培养了 11 名诺贝尔物理奖获得者。