

一本帮助你轻松掌握C语言的指南

C语言 程序设计

(新教材)

白冰 陈青华 袁锋 王宁 主编
姚艳玲 李雅林 编写



山东城市出版传媒集团·济南出版社

图书在版编目 (C I P) 数据

C 语言程序设计/袁锋, 王宁主编. —济南: 济南出版社, 2018. 8

ISBN 978 - 7 - 5488 - 2421 - 3

I. ①C… II. ①袁… ②王… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字 (2018) 第 191473 号

C 语言程序设计

出版人 崔 刚
丛书策划 郭 锐
责任编辑 丁洪玉 侯建辉
装帧设计 刘 畅

出版发行 济南出版社
地 址 山东省济南市二环南路 1 号(250002)
电 话 0531 - 86131729
网 址 www. jnpub. com
经 销 各地新华书店
印 刷 济南龙玺印刷有限公司
版 次 2018 年 8 月第 1 版
印 次 2018 年 8 月第 1 次印刷
开 本 185 毫米 × 260 毫米 16 开
印 张 23.5
字 数 503 千
印 数 1—2200
定 价 59.80 元

法律维权 0531 - 82600329
(济南版图书, 如有印装错误, 请与出版社联系调换)

前 言

C 语言是一门通用计算机编程语言,自 20 世纪 90 年代以来,C 语言迅速在全世界普及推广。作为一种结构化设计语言,C 语言语法清晰、结构简单、运算符丰富,可以在其它高级语言中难以实现的运算。同时,由于数据结构丰富、可移植性好、执行效率高等优点,为其自身的推广使用都产生了积极的影响。

C 语言的设计目标是提供一种能以简易的方式编译、处理低级存储器、产生少量的机器码,以及不需要任何运行环境支持便能运行的编程语言。在高等学校的课程教育环节中,“C 语言程序设计”始终是其中一门基础计算机课程。

本教材针对初学者的特点和认知规律,在利用理论知识进行学术引导的同时,添加经典实践编程案例强化学习者的编程经验,全书共分为 11 章节。第 1 章为概述,对 C 语言的发展和 Visual C++ 2010(学习版)编程环境进行简要介绍,让初学者尽快熟悉 C 语言编程环境;第 2 章是程序中数据的表现形式及运算的基础知识,对 C 语言的常量、变量、运算符及表达式、数据类型及转换进行简要介绍;第 3~4 章为顺序、选择及循环结构学习章节,由潜入深的将结构化程序的设计方法进行了介绍;第 5 章对数值数组和字符数组进行分析和讲解;第 6 章为函数章节,使读者深入学习模块化程序设计思想;第 7 章主要学习指针的用法,分析计算机内存中数据的存入与读取方式;第 8 章为结构体、共用体章节;第 9 章主要为以后新开设的《数据结构》课程做“预习”,并对全国计算机等级考试中二级 C 语言考试部分中的数据结构问题进行理论学习;第 10 章对存储于外存储器上的一组相关数据的有序集合——文件进行了讲解;第 11 章介绍 C 语言中几种常用的预处理功能。

教材各章节末均附有章节小结和习题。书后还有七个附录,附录一是各章节内容的经典训练例题,附录二是各章节内容的 ACM 国际大学生程序设计竞赛经典训练例题,附录三到附录七为常用参考附录。

本教材的编写者均为来自教学一线的 C 语言程序设计课程的任课教师,教学经验丰富,在教材的编写过程中针对不同层次的学生学习基础和学习特点进行了设计,因此本教材适合作为普通本科院校计算机、非计算机专业学生以及普通高校专科学生的学习教材及参考用书。

在本教材的内容组织、整理以及写作过程中,参阅了大量的相关文献,包括同行学者的有关论著和讲稿,同时也得到多位教师、同学的参与和帮助,在此向他们表示衷心感谢!

由于 C 语言不断发展,加之作者的水平有限,书中难免会有不当或疏漏之处,敬请各位读者、专家及同仁批评指正。

教材编写组

2018 年 6 月

目 录

第 1 章 C 语言概述

1.1 C 语言的发展和特点	1
1.2 C 语言常用术语介绍	4
1.3 C 程序的结构	7
1.4 Visual C++ 2010(学习版)上机指导说明	12
章节总结	19
习题	19

第 2 章 数据的表现形式及其运算

2.1 常量和变量	21
2.2 运算符与表达式	24
2.3 数据类型及其转换	32
章节总结	43
习题	43

第 3 章 顺序结构与选择结构程序设计

3.1 C 程序设计概述	45
3.2 输入/输出标准函数	49
3.3 if 选择结构语句	59
3.4 switch 选择结构语句	65
3.5 综合应用举例	67
章节总结	70
习题	71

第 4 章 循环结构程序设计

4.1 for 循环	75
4.2 while 与 do-while 循环	80
4.3 for、while 与 do-while 循环语句的比较	87
4.4 跳转语句 break、continue	88
4.5 循环语句的嵌套	90
4.6 综合应用举例	91
章节总结	94
习题	94

第 5 章 数组

5.1 一维数组	99
5.2 二维数组	111
5.3 字符型数组	116
5.4 综合应用举例	129
章节总结	133
习题	136

第 6 章 函数

6.1 函数概述	139
6.2 函数的调用	149
6.3 函数的嵌套调用、递归调用	153
6.4 数组作函数参数	158
6.5 变量的作用域及存储类别	163
6.6 内部函数和外部函数	172
6.7 综合应用举例	174
章节总结	176
习题	177

第 7 章 指针

7.1 指针变量	179
----------------	-----

7.2 指针与一维数组	187
7.3 指针与多维数组	196
7.4 指针与字符串	203
7.5 指针与函数	212
7.6 指针数组和指向指针的指针	217
7.7 利用指针进行内存的动态分配	221
7.8 综合应用举例	225
章节总结	234
习题	235
第 8 章 结构体与共用体	
8.1 结构体	240
8.2 结构体数组	245
8.3 结构体指针	249
8.4 链表	255
8.5 共用体	259
8.6 其他自定义数据类型	264
8.7 综合应用举例	268
章节总结	271
习题	272
第 9 章 数据结构和数据抽象	
9.1 概述	275
9.2 线性表	277
9.3 综合应用举例	284
章节总结	289
习题	289
第 10 章 文件	
10.1 C 文件概述	291
10.2 文件的打开与关闭	294

10.3 文件的读写	296
章节总结	304
习题	304

第 11 章 预处理命令

11.1 宏定义	306
11.2 “文件包含”处理	310
11.3 条件编译	312
章节总结	316
习题	316

附 录	319
-----------	-----

第 1 章

C 语言概述

章节导读

本章节对 C 语言的发展和 Visual C++2010(学习版)编程环境进行简要介绍。其目的是让读者尽快熟悉 C 语言编程环境,学会简单 C 语言程序编写,掌握 C 语言程序的运行步骤。

作为世界上最流行、使用最广泛的高级程序设计语言之一,C 语言既具有高级语言的特点,又具有汇编语言的特点。作为一门优秀的编程语言,C 语言的优点是限制少、通用性强、与平台无关、简洁高效,具有很强的可移植性,可以实现“一次编写,到处运行”。正因为 C 语言应用范围非常广泛,具备很强的数据处理能力,所以不仅在软件开发方面,在其他各类科研中也都需要用到 C 语言。

1.1 C 语言的发展和特点

1.1.1 C 语言的发展历程

C 语言诞生于 1972 年,它由美国电话电报公司(AT&T)贝尔实验室设计。AT&T 公司主要致力于通信领域,它组建于 1877 年,距今已经有 100 多年历史,目前是主营电信业务,是美国第二大移动运营商。图 1.1 为 AT&T 公司 logo。



图 1.1 AT&T 公司 logo

C 语言作为一种经典底层开发语言,不仅适合开发系统软件,也大量适用于应用软件、平台界面的开发。目前,C 语言主要应用在嵌入式系统开发、底层操作系统上应用程序的开发,以及编写一些简单的应用软件上。常见的 Linux 系统和 Unix 系统都是用 C 语言开发的。

C 语言从开始发展到成熟应用共经历了八个阶段,如图 1.2 所示。

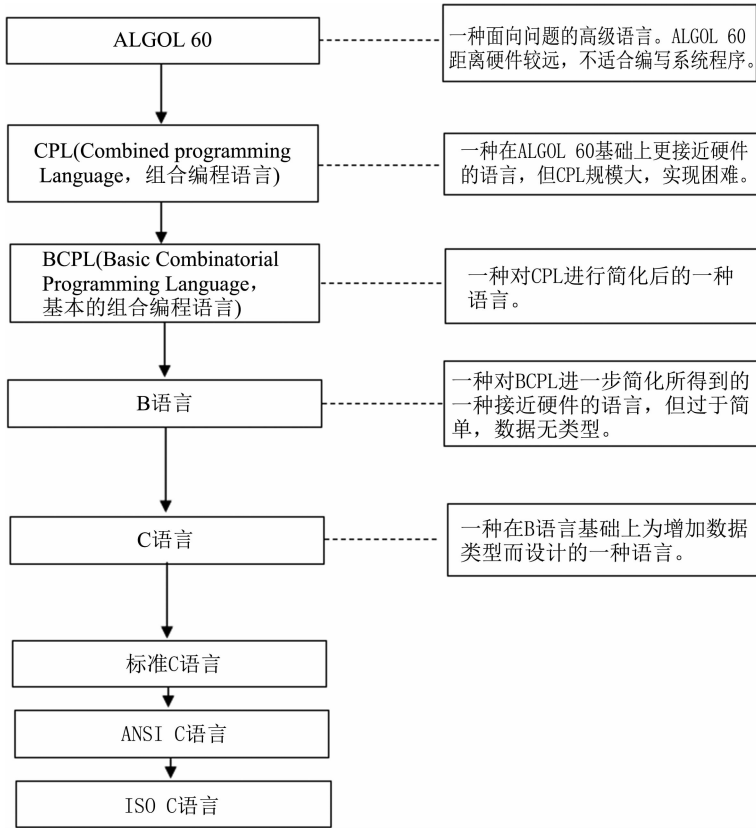


图 1.2 C 语言发展过程

从 C 语言的发展过程可以看出,C 语言既有一种由 ALGOL 60 带来的高级语言特性,又有 BCPL 带来的接近硬件的低级语言特性。因此,C 语言可以称为“高级语言中的低级语言”或“中级语言”,即 C 语言兼有高级语言和低级语言的特点。

而且,之所以在前面称 C 语言为一种经典底层开发语言,是因为很多如 C++、C#、Java、PHP、Perl、LPC 和 Unix 的 C Shell 等编程语言的结构过程都深受 C 语言的影响。一旦掌握 C 语言,再学其他编程语言,入门就较为容易,这也是很多高等学校将《C 语言程序设计》作为工科计算机编程教学的入门语言的原因之一。

1.1.2 C 语言的主要特点

C 语言作为一种兼有高级语言和低级语言特点的经典底层开发语言,在实际进行编程操作的时候,常被用于底层系统的开发,在编写对编程操作效率要求较高或者对可移植性要求较高的系统软件时表现出了非凡的作用和不可替代性。在现代计算机编程技术发展, C 语言的应用更加广泛、深入,编程人员能通过应用这种计算机语言而达成自己的预设目标。C 语言之所以在高级语言中的地位日趋上升,是因为它具有一些优于其他语言的特点,主要表现在:

1. 编程简便、灵活、高效

在运用 C 语言进行计算机程序编写时,其最突出的特点之一就是操作简便灵活。

C 语言共有 37 个关键字(其中有 32 个是常用关键字),9 种控制语句(详见附录)。在实际的程序编写当中,这种计算机语言的使用也较为自由,而编程人员在进行程序书写的时候常会使用小写字母进行程序编写。

注意:C 语言之所以能获得如此突出的效果,主要是由于这种计算机语言能有机地将现代计算机编程中的高级计算机语言应用中的特点和低级计算语言应用中的实用性结合到一起。因此编写程序时完全可以像汇编语言一样直接对字节、位、地址等进行编程,从根本上简化程序设计、提高效率。

2. 运算符丰富

和其他编程语言相比,C 语言编程时可使用的运算符数量较多,可供选择使用的运算符共计 34 种。这些运算符能满足正常编程操作的各种需要。而且,丰富的运算符使编程方便、灵活,不但提高了编程的效率,也提高了程序运行的效率和质量,这是其他高级编程语言难以达到的。

3. 数据结构类型丰富

在应对各种复杂数据运算处理的时候,C 语言处理系统也有较大的优势,能满足各种数据处理的需要。C 语言允许使用的数据类型有基本类型、枚举类型、派生类型以及空类型。

注意:在使用 C 语言进行数据编程处理的时候,引入指针类型的处理技术能使程序运行效率更高。和其他计算机编程语言相比,C 语言的另一个特点就是其强大的图形功能,能支持多种类型的显示器设备以及驱动器设备。

4. 结构化特点突出

在现代计算机语言的发展当中,逐渐出现了特点较为鲜明的结构化语言,这种结构化的计算机语言在使用的时候能做到对编程代码以及不同数据进行分隔化处理。在实际的信息化处理当中,编程程序当中的各个部分除了在运行的时候需要进行关键信息的处理之外,程序各部分之间是相互独立的。这样在对计算机程序进行管理的时候,由于各个部分之间是相互独立的,能使计算机程序之间的层次以及结构更加清晰,从而为计算机程序管理人员以及程序维护人员的工作创造便利条件。

5. 操作自由度较大

在当前的计算机编程领域当中,一般的编程语言在规整性方面要求较高,这样虽然使编程时不易出错,但同时也降低了编程人员的自由度。当使用 C 语言进行程序编写的时候,这种计算机语言给编程人员的自由度比较大,但同时也会对编程人员提出更高的要求。

6. 适用范围较大

在众多计算机编程语言当中,C 语言之所以能异军突起,还和其强大的兼容性有关系。C 语言能完美地适用于多种类型的计算机系统,比如最早的 DOS 操作系统、UNIX

操作系统,同时也适用于多种计算机机型。

总体看来,C语言是一种实用性极强的计算机结构化编程语言,其在实际的操作中不仅具有丰富的运算符以及灵活的数据结构类型,同时在代码编程自由度以及可移植性方面也有突出的表现。可以预见,C语言在计算机编程领域后续的发展进程中仍会有更加广泛的运用。

1.2 C语言常用术语介绍

C语言功能强大,设计灵活,要熟练掌握C语言,就必须从C语言程序基础开始学习。在C语言中,C语言程序的内部部件如字符集(标准字符共计128个,扩展字符共计128个)、关键字、标识符及注释等,为C语言中常用的术语。下面将对常用字符集、关键字、标识符及注释进行介绍。(详细内容及非常用范围内容请参照附录)

1.2.1 字符集

C语言基本字符集分为源字符集和执行字符集。其中,源字符集可理解为书写C语言源文件所用的字符集,执行字符集可理解为C语言程序执行期间解释的字符集。

源字符集包括字母(共计52个)、数字(共计10个)、格式符(共计4个)、特殊字符(共计29个)。执行字符集在源字符集的基础上还包括空字符、行末标志符(换行符)、警报符、退格符(BS)和回车符(CR)。

1. 字母:包含52个大、小写拉丁字母

A、B、C、D、E、F、G、H、I、J、K、L、M、N、O、P、Q、R、S、T、U、V、W、X、Y、Z,

a、b、c、d、e、f、g、h、i、j、k、l、m、n、o、p、q、r、s、t、u、v、w、x、y、z。

2. 数字:10个阿拉伯数字

0、1、2、3、4、5、6、7、8、9。

3. 格式符:空格、水平制表符(HT)、垂直制表符(VT)、换页符(FF)

4. 特殊字符(共计29个):如表1-1所示

另外,需要强调的是,除上述字符外的其他字符,都只能放在注释语句、字符型常量、字符串型常量和文件名中。

表 1-1 特殊字符

字符	名称	字符	名称	字符	名称
!	感叹号	+	加号	"	引号
#	井号(数字号)	=	等号	{	左花括号
%	百分号	~	波浪号	}	右花括号
^	折音符	[左方括号	,	逗号
&	和号]	右方括号	.	句号
*	星号	'	撇号	<	小于号

(续表 1-1)

字符	名称	字符	名称	字符	名称
(左括号		竖线	>	大于号
_	下划线	\	反斜杠	/	除号
)	右括号	;	分号	?	问号
-	连字符	:	冒号		

1.2.2 关键字

C语言中的关键字,又可称之为保留字,是编译器能识别的特殊字符串符号,每种计算机语言都会有其特定的关键字。关键字的数量是由编译器来决定的,关键词大小写敏感性也和编译器有关。

1999年初及其以前,标准C语言程序的标识符中有32个标识符被保留作为关键字,这些关键字不能用于其他用途。关键字是程序设计中代码必须包含的部分,编译器在编译C程序代码的时候,如果关键字写错,编译器就会报错。表1-2中列出了C语言中常用的关键字。

表 1-2 C语言中常用的关键字

C语言中常用的关键字			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

1999年12月16日,ISO推出的C99标准中,新增了5个关键字:

inline、restrict、_bool、_Complex、_Imaginary。

2011年12月8日,ISO发布C语言的新标准C11中,新增了7个C语言关键字:

_Alignas、_Alignof、_Atomic、_Static_assert、_Noreturn、_Thread_local、_Generic。

1.2.3 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。与其他高级语言一样,在C语言程序中,标识符是指用来标识对象(包括变量、函数、数组、类型或其他用户自定义项目的名称等)名字的有效字符序列。

C 语言规定,标识符只能是字母(A~Z,a~z)、数字(0~9)、下划线(_)组成的字符串。

需要注意:

(1)这些名称的标识符不能以数字开头,第一个字符必须是字母或下划线,其后的字符必须是字母、数字、下划线。

(2)C 标识符内不允许出现标点字符,比如@、\$和%。

(3)在标识符中,大小写是有区别的。例如,PROGRAM、program 以及 Program 是不同的标识符。

(4)标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。

另外,在 C 语言执行过程中,标准版 C 语言不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某版本 C 语言中规定标识符前八位有效,当两个标识符前八位相同时,则被认为是同一个标识符。

1.2.4 注释

C 语言中,为理解程序的含义,有时需要使用注释功能加以解释。注释的内容是给程序员或程序使用者观看,而不是计算机,因此注释内容在 C 语言程序进行编译过程中会被自动忽略。

C 语言注释方法有两种:

(1)单行注释。用符号“//”并且后跟注释内容表示,即“//注释内容”。

(2)多行注释。用符号“/* */”并且中间夹杂注释内容表示,即“/*注释内容*/”。

【例 1.1】请编写一个 C 语言程序,输出显示以下内容:

This is an example of a comment.

程序内容如下:

```

1  /* 这是一个有关于注释知识讲解的事例程序 */
2  #include <stdio.h>
3  int main()                                //函数说明
4  {                                          //函数体开始
5      printf("This is an example of a comment.\n"); //函数体内容
6      return 0;
7  }                                          //函数体结束

```

程序结果如图 1.3 所示:

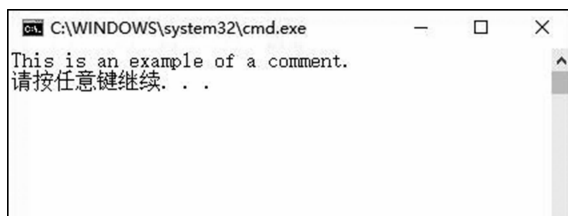


图 1.3 例 1.1 程序结果图

【例题中关键问题说明】

(1)通过图 1.3 可以看出,该例子的运行结果中并没有看到注释内容,只有最终结果“This is an example of a comment.”。

(2)程序头文件之前的“/*……*/”表示对整个程序的解释,可以根据需要写在程序中的任何一行中,以“/*”开始,以“*/”结束。

(3)行注释的表示为“//”,表示在一行中“//”以后的所有内容都是注释,这种表示方式不可跨行。

(4)C 语言编译器在编译时会跳过注释,目标代码中不会包含注释。

1.3 C 程序的结构

1.3.1 C 程序的概念

学习 C 语言的目的就是根据实际问题设计 C 语言程序,既然如此,就应当了解:

(1)C 语言程序是什么样子的?

(2)它是怎样构成的?

(3)它有什么特点呢?

下面通过一个简单的 C 语言程序范例来认识程序的框架、了解 C 语言程序的组成、分析 C 语言程序的特点,以培养学习者良好的编程风格和正确的程序编写习惯,为以后学习编写 C 语言程序打好基础。

【例 1.2】请编写一个 C 语言程序,输出显示以下内容:

Welcome, I hope you enjoy learning C programming!

程序内容如下:

```

1  #include <stdio. h>                                //头文件
2  int main()                                          //函数说明
3  {                                                  //函数体开始
4      printf("Welcome, I hope you enjoy learning C programming! \n"); //函数体内容
5      return 0;
6  }
```

程序结果如图 1.4 所示:

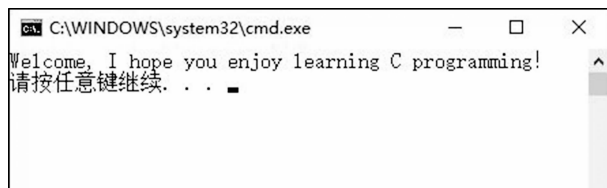


图 1.4 例 1.2 程序结果图

【例题中关键问题说明】

(1) 头文件：“stdio. h”是 C 编译系统提供的一个头文件名，stdio 为“standard input & output”的缩写。添加头文件有两种形式：“#include <stdio. h>”和“#include "stdio. h"”。

①用“ ”表示的是包含的文件首先在当前目录寻找，如果失败就返回系统目录寻找，适合程序人员使用。

②用“< >”表示的是在系统指定的子目录下寻找被包含的文件，适合一般用户人员使用。

(2) 主函数：main() 函数为程序“主函数”，是程序执行的入口。每一个 C 语言程序都必须有且只有一个主函数，并由此执行。

(3) 函数体：函数体为用“{ }”括起来的部分。main() 函数中的所有操作（或语句）都在这一对“{ }”之间，即位于 main() 函数体内。

(4) 语句：程序中，main() 函数有两条语句，分别为“printf("Welcome, I hope you enjoy learning C programming! \n");”和“return 0;”。

①printf() 函数是 C 语言的库函数，其功能是输出程序，使该程序应用时于屏幕上显示：Welcome, I hope you enjoy learning C programming!

其中，“\n”字符是换行符，其作用是使光标移到下行首列。

②return 语句是程序结束后返回到调用的位置。

其中，视不同语句内容，return 语句可省略或不可省略。

(5) 语句结束：每条语句必须以“;”(分号)结束。

(6) 主函数返回值类型：main() 函数的返回值可以是 int 类型，也可以定义为 void 类型，这和具体的 C 程序编译平台或编译器相关。如为 int main() 的形式，返回值为 0，即 return 语句不可省略。

著名计算机科学家尼古拉斯·沃斯(Niklaus Wirth)提出一个公式：

$$\boxed{\text{程序}} = \boxed{\text{数据结构}} + \boxed{\text{算法}}$$

这里说的数据结构是对数据的描述，主要是指数据的类型和数据的组织形式。而算法就是对操作的描述，指的是求解某问题的具体操作步骤。实际上，一个程序除了数据结构和算法外，还必须使用一种计算机语言来描述算法。对于初学者来说，根据实际问题确定一个科学的算法是设计程序的第一步，也是比较重要的一步。

如何理解算法呢？

先看一下算法的定义：做任何事情都要有一定的步骤，解决一个实际问题所采用的方法和步骤就称为算法。

算法的程序或步骤特征：

(1) 确定性：算法内容不应模棱两可。

(2)有限性:算法的步骤应在有限的操作序列之后停止。

(3)可执行性:算法中的每一步必须是可以执行的。

(4)有零个或多个输入:所谓输入是指在执行算法时需要从外界取得必要的信息,可以没有输入,也可以有一个或多个输入。

(5)有一个或多个输出:算法的目的是为了求解,“解”就是输出。但算法的输出不一定是计算机的打印输出或屏幕输出,一个算法得到的结果就是算法的输出。没有输出的算法是没有意义的。

1.3.2 怎样表示一个算法

在最初接触 C 语言程序时,会普遍习惯使用自然语言描述一个算法,其优点是简单、容易理解,但语句往往比较冗杂、长度过长,而且在不同程序员进行描述或不同使用者进行解读时容易出现歧义。

例如,要求得 1 至 5 内的数字和,可采用的自然语言描述,过程如下:

Step 1:开始。

Step 2:先求 $1+2$,得到结果 3。

Step 3:将步骤 2 得到的结果 3 加上 3,得到结果 6。

Step 4:将 6 再加上 4,得到结果 10。

Step 5:将 10 再加上 5,得到结果 15。

Step 6:结束。

这样的算法虽然正确,但缺乏严谨性,而且如果求和的数字数目过多,如 100 个、1000 个数字的累加和,会导致描述步骤过长。

因此,为避免算法描述冗杂、过长或存在歧义,除常用自然语言描述外还可以使用三种基本结构及流程图描述算法、使用伪代码描述算法或使用计算机语言表示算法。

1. 用三种基本结构和传统流程图表示算法

描述算法常使用流程图有传统流程图、N-S 流程图、PAD 流程图等。本部分主要结合 C 语言的三种基本结构介绍传统流程图和 N-S 流程图。

(1)传统流程图

传统流程图又称框图,就是用各种图形及文字说明表示算法的图式。用框图表示算法,直观形象,易于理解,不会产生“二义性”。

美国国家标准化协会 ANSI(American National Standard Institute)规定了一些常用的流程图符号,如图 1.5 所示。

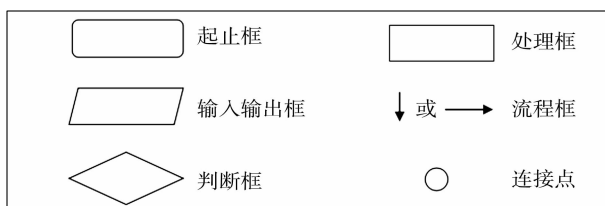


图 1.5 常用流程图符号