

普通高等学校“十一五”省级规划教材

# C语言

## 程序设计

(第4版)

吴国凤 主编



合肥工业大学出版社  
HEFEI UNIVERSITY OF TECHNOLOGY PRESS

高等学校“十一五”省级规划教材

# C 语言程序设计

(第4版)

主 编 吴国凤

副主编 黄存东 梅灿华 黄 兵

编 委 (按姓氏笔画排序)

王 胜 李中永 李家兵

吴国凤 范文广 孟林树

宣善立 贺继东 黄存东

黄 兵 梅灿华

主 审 王 浩

合肥工业大学出版社

## 内 容 简 介

本书是面向高职高专学生学习C语言程序设计的理想教材。全书共分12章,主要内容包括:C语言概述、数据类型与基本运算、顺序结构、选择结构、循环结构、数组与字符数据处理、函数、指针、结构体与联合、位运算、文件等。

本书内容丰富,重点突出,针对初学者的特点,讲解简明扼要,深入浅出,通俗易懂。本书对C语言的语法规则进行了提炼,注重讲解程序设计的概念和方法,培养学生编写程序的能力。各章配备丰富的例题、习题,并对典型例题进行精解,另配《C语言程序设计实训教程》一书,特别适合作为工科高等院校各专业程序设计语言和高职高专各专业的课程教材,也可作为自学教材。

378196

### 图书在版编目(CIP)数据

C语言程序设计/吴国凤主编. —4版. —合肥:合肥工业大学出版社,2012.8  
ISBN 978-7-5650-0840-5

I. ①C… II. ①吴… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第180582号

### C语言程序设计(第4版)

主编 吴国凤

责任编辑 陆向军

出版	合肥工业大学出版社	版次	2012年8月第4版
地址	合肥市屯溪路193号	印次	2012年8月第7次印刷
邮编	230009	开本	787毫米×1092毫米 1/16
电话	综合编辑室:0551-2903028 发行部:0551-2903198	印张	16
网址	www.hfutpress.com.cn	字数	360千字
E-mail	hfutpress@163.com	发行	全国新华书店
		印刷	安徽江淮印务有限责任公司

ISBN 978-7-5650-0840-5

定价:26.00元

如果有影响阅读的印装质量问题,请与出版社发行部联系调换

# 计算机系列教材编委会

主任 胡学钢（合肥工业大学）

王 浩（合肥工业大学）

委员（以姓氏笔画为序）

丁亚明（安徽水利水电职业技术学院）

王志宏（安徽国防科技职业学院）

付建民（安徽工业经济职业技术学院）

刘 力（安徽财贸职业技术学院）

刘竞杰（安徽工贸职业技术学院）

杨克玉（安徽商贸职业技术学院）

李 燕（安徽广播影视职业学院）

张兴元（六安职业技术学院）

黄存东（安徽国防科技职业学院）

梅灿华（淮南职业技术学院）

## 第4版前言

《C语言程序设计》自2005年8月初版以来,承蒙学术界同行和广大读者的厚爱,纷纷采用本书作为高职高专计算机类专业教材,使本书的发行量迅速增加。2007年10月,本书被列为安徽省“十一五”规划教材。虽然如此,实践表明本书仍存在不足之处。为了保证本书的先进性和实用性,我们进行了多次修订。

本次修订主要是调整部分章节内容和图表,增加了课后练习,力求使本书趋于完美。

C语言是目前国内外广泛使用的程序语言之一。C语言功能丰富、表达能力强、使用灵活方便、程序执行效率高、可移植性好。C语言的模块化、结构化特性使其可以满足现代程序设计的需要,既可以用来编写系统软件,又可以用来编写应用软件。因而,C语言是软件工作者必须掌握的一个工具。

为了满足高职高专教学改革的需要,我们组织了省内有关高校长期在教学第一线的、有经验的教师编写了《C语言程序设计》一书,供高职高专的师生们使用。为适合教学方式的变革,我们在编写教材的同时,一并编写了配套的实验教材、课件、电子教案及相应的程序设计素材库。

本书共分12章,全面、系统地讲述了C语言的基本概念、原理和方法。教材组织精练,讲解深入浅出,循序渐进。特别强调理论联系实际,通过大量的案例解析,说明了语法结构、编写程序的方法以及程序设计的技巧。本书既注重培养学习者程序设计的能力,又提倡良好的程序设计风格。另外,每章还配备了大量的习题,便于学生练习掌握。本书由吴国凤担任主编,黄存东、梅灿华、黄兵担任副主编。全书由吴国凤统编定稿。

虽然经我们多次认真的修订、补充和校正,但由于我们的理论水平、研究能力和知识深广度的限制,书中难免还存在缺点和错误。真诚希望同行专家和广大读者批评指正。

编者

2012年8月

## 目 录

第 1 章 C 语言概述 .....	(1)
1.1 程序设计与 C 语言 .....	(1)
1.2 C 程序初识 .....	(3)
1.3 程序设计方法与算法 .....	(7)
1.4 C 程序运行环境与学习方法 .....	(11)
1.5 例题精解 .....	(15)
1.6 本章小结 .....	(16)
习题 .....	(16)
第 2 章 数据类型、运算符与表达式 .....	(18)
2.1 C 语言基础 .....	(18)
2.2 常量 .....	(21)
2.3 变量 .....	(22)
2.4 基本运算符与表达式 .....	(23)
2.5 不同数据类型间的转换和运算 .....	(30)
2.6 例题精解 .....	(32)
2.7 本章小结 .....	(33)
习题 .....	(34)
第 3 章 顺序结构程序设计 .....	(36)
3.1 C 语言中的语句 .....	(36)
3.2 数据的输入输出 .....	(38)
3.3 例题精解 .....	(45)
3.4 本章小结 .....	(48)
习题 .....	(48)
第 4 章 选择结构程序设计 .....	(53)
4.1 关系运算和逻辑运算 .....	(53)
4.2 if 语句 .....	(56)
4.3 switch 语句 .....	(61)

---

---

4.4 例题精解 .....	(63)
4.5 本章小结 .....	(66)
习题 .....	(66)
<b>第5章 循环结构程序设计 .....</b>	<b>(71)</b>
5.1 概述 .....	(71)
5.2 while 语句 .....	(71)
5.3 do while 语句 .....	(73)
5.4 for 语句 .....	(75)
5.5 循环嵌套 .....	(79)
5.6 break 语句、continue 语句和 goto 语句与标号 .....	(81)
5.7 例题精解 .....	(83)
5.8 本章小结 .....	(88)
习题 .....	(88)
<b>第6章 数 组 .....</b>	<b>(96)</b>
6.1 一维数组 .....	(96)
6.2 二维数组 .....	(99)
6.3 字符数组和字符串 .....	(103)
6.4 例题精解 .....	(108)
6.5 本章小结 .....	(112)
习题 .....	(113)
<b>第7章 指 针 .....</b>	<b>(117)</b>
7.1 指针的基本概念 .....	(117)
7.2 指针与数组 .....	(124)
7.3 指针数组和指向指针的指针 .....	(133)
7.4 例题精解 .....	(138)
7.5 本章小结 .....	(140)
习题 .....	(141)
<b>第8章 函 数 .....</b>	<b>(145)</b>
8.1 函数的基本概念 .....	(145)
8.2 函数参数和函数的值 .....	(146)
8.3 函数的调用 .....	(148)
8.4 数组作为函数参数 .....	(149)



---

---

8.5	指针作为函数参数 .....	(152)
8.6	嵌套调用与递归调用 .....	(154)
8.7	存储类型 .....	(157)
8.8	命令行参数 .....	(161)
8.9	例题精解 .....	(162)
8.10	本章小结 .....	(166)
	习题 .....	(166)
<b>第9章</b>	<b>编译预处理 .....</b>	<b>(172)</b>
9.1	宏定义 .....	(172)
9.2	文件包含 .....	(175)
9.3	条件编译 .....	(176)
9.4	例题精解 .....	(177)
9.5	本章小结 .....	(179)
	习题 .....	(180)
<b>第10章</b>	<b>结构体与共用体 .....</b>	<b>(183)</b>
10.1	结构体 .....	(183)
10.2	共用体 .....	(199)
10.3	枚举 .....	(202)
10.4	用户定义类型 .....	(203)
10.5	例题精解 .....	(204)
10.6	本章小结 .....	(205)
	习题 .....	(206)
<b>第11章</b>	<b>位运算 .....</b>	<b>(210)</b>
11.1	位运算的概念 .....	(210)
11.2	位运算 .....	(210)
11.3	位域(位段) .....	(214)
11.4	本章小结 .....	(217)
	习题 .....	(217)
<b>第12章</b>	<b>文 件 .....</b>	<b>(219)</b>
12.1	文件概念 .....	(219)
12.2	文件指针 .....	(220)
12.3	文件的打开与关闭 .....	(221)



---

---

12.4	文件的读写 .....	(222)
12.5	文件的定位操作 .....	(229)
12.6	文件的错误检测 .....	(231)
12.7	本章小结 .....	(232)
	习题 .....	(232)
附 录	.....	(234)
1.	常用字符与 ASCII 代码对照表 .....	(234)
2.	C 语言运算符的优先级与结合性 .....	(235)
3.	C 库函数 .....	(236)
4.	常见错误信息表 .....	(243)
参考文献	.....	(246)

# 第1章 C语言概述

## 【教学提示】

本章主要介绍C语言及其程序设计的基本知识。通过案例分析,了解C语言的简史和特点;掌握C语言程序基本结构;掌握计算机算法的基本概念;掌握C程序的运行环境与学习方法。

## 【核心概念】

C程序的构成 main函数和其他函数 C程序的书写风格 计算机算法 C程序的运行环境

## 1.1 程序设计与C语言

### 1.1.1 程序与程序设计语言

#### 1. 程序

计算机最能吸引人之处是它能自动执行指定的程序。所谓程序,就是用语言、文字、图表等方式表达解决某个问题的方法和步骤,是计算机解决某些特定问题所需的代码化指令序列。程序设计者根据预先制定的功能和规则,编写一系列完整指令,由计算机执行,实现预定的功能和任务,这就是计算机程序。

#### 2. 程序设计语言

程序设计就是根据给定的任务,设计、编写、调试程序的过程。在程序设计过程中要用一定的工具以及环境,C语言就是一种常用的计算机高级程序设计语言工具。程序设计语言的种类很多,大体上分为3大类。

##### (1) 机器语言

机器语言是用0和1表示的二进制代码的指令序列,是计算机能直接识别和执行的语言。机器语言中的每一条语句(称为指令)由操作码和操作数组成,反映了操作的对象和性质。但由于机器语言是用二进制表示,难于学习和记忆,编写工作量大,程序可读性差,非计算机专业人员难于掌握。

##### (2) 汇编语言

汇编语言采用助记符号表示机器语言中的指令和数据,即用助记符号代替了二进制表示的机器指令,例如用ADD(Addition)表示做加法的助记符,每条汇编语言的指令对应一条机器语言的指令。

计算机硬件只能识别机器指令。要执行汇编语言编写的源程序,必须用汇编处理程序将源程序翻译成机器语言程序才能执行。

##### (3) 高级语言

高级语言表达方式接近被描述的问题,接近于自然语言和数学表达式,易于人们接受和掌握。随着计算机技术的发展,形成了为数众多的计算机高级语言并得到广泛应用,每种高

级语言都有其适合的应用领域。

如:FORTRAN 语言适用于数值计算;BASIC 语言简单易学,适合于初学者学习;Pascal 语言适用于教学;C 语言适用于系统软件和应用软件的开发;Java 语言既是通用的程序设计语言,又是可以用于 Internet 开发的面向对象的程序设计语言。

由于计算机只能识别机器语言,因此,用高级语言编写的程序(或称源程序)要通过专用的程序将高级语言翻译成机器语言程序(或称目标程序)后才能被计算机执行。将高级语言翻译成机器语言的方式有两种:一种是将高级语言程序一次性翻译成机器语言,再通过连接程序将系统库连接到程序中形成可执行程序,然后运行,这种语言称为编译型语言,如图 1-1 所示;另一种是翻译一句执行一句,称为解释型语言。



图 1-1 编译型语言的执行过程

C 语言是典型的编译型语言。面向过程的计算机的语言的核心是将输入数据通过算法转换为输出数据。因此,程序主要描述的是:数据的表示,对数据的处理算法以及数据输入输出。

### 1.1.2 C 语言的发展与特点

#### 1. C 语言的发展

C 语言是在 B 语言的基础上发展起来的。它的前身是 ALGOL 60(1960 年,面向问题的高级语言,离硬件比较远,不宜用来编写系统程序)。1963 剑桥大学推出了 CPL(复合程序设计语言),CPL 语言接近硬件一些,但规模较大,难以实现。1967 年剑桥大学的马丁·理查德对 CPL 进行了简化,推出了 BCPL(基本复合程序设计语言)。1970 年美国贝尔实验室的肯·汤普逊对 BCPL 进行了进一步的简化,突出敢硬件处理能力,并取了“BCPL”的第一个字母“B”作为新语言的名称,并且用 B 语言编写了第一个 UNIX 操作系统。但“B”语言过于简单,功能有限,在 1972 年贝尔实验室的布朗·W. 卡尼汉和丹尼斯·M. 利奇对 B 语言进行了完善和扩充,在 B 语言的基础上设计出了 C 语言。C 语言保持了 BCPL 语言和 B 语言精练、接近硬件的优点,又克服了它们过于简单、数据无类型、功能有限的缺点。后来多次改进,在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布突出优点才引起人们普遍注意。

1977 年,为了让 C 语言脱离 UNIX 操作系统,成为在任何计算机上都能运行的通用计算机语言,布朗·W. 卡尼汉和丹尼斯·M. 利奇(K&R)合著了著名的《C 程序设计语言》一书,对 C 语言的语法进行了规范化的描述,成为当时的标准。通常简称为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准 C 语言,1983 年美国国家标准委员会对 C 语言进行了标准化,颁布了第一个 C 语言标准草案为 83 ANSI C;1987 年又颁布了新标准为 87 ANSI C;1990 年国际标准化组织 ISO(International Standard Organization)接受 87 ANSI C 为 ISO 的标准。目前流行的 C 编译系统都以此为基础。

#### 2. C 语言的特点

C 语言同时具有汇编语言和高级语言的优势,它把高级语言面向过程和低级语言与硬

件关系密切的优点有机地结合起来,具有许多显著的特点:

(1)语言简洁、紧凑,使用方便、灵活。ANSI C 为标准只有 32 个关键字(详见第 2 章 2.1.1);9 种控制语句;数据构造能力强;程序书写格式自由。与 Pascal 相比,源程序短,压缩了一切不必要的成分,见表 1-1 所列。

表 1-1 C 语言与 Pascal 语言比较

C 语言	Pascal 语言	含义
{ }	BEGIN... END	复合语句
if(e) s;	IF(e) THEN s	条件语句
int i;	VAR i:INTEGER	定义 i 为整型变量
int a[10];	VAR a:ARRAY[1..10] OF INTEGER	定义整型数组 a
int f();	FUNCTION f():INTEGER	定义 f 为返回值整型的函数
int *p;	VAR p:↑INTEGER	定义 p 为指向整型变量的指针
i+=2;	i=i+2	赋值 $i+2 \Rightarrow i$
i++,++i;	i=i+1	i 自增值

(2)运算符和数据结构类型丰富。C 语言有 34 种运算符(见附录 2)。并把括号、赋值、逗号等都作为运算符处理。从而使 C 的运算类型极为丰富,可以实现其他高级语言难以实现的运算。

C 语言的数据类型有:基本类型、构造类型(数组、结构体、联合体)、指针类型和空类型,为处理各种复杂数据类型提供了实用的手段。

(3)具有结构化的控制语句。C 语言具有多种结构化的控制语句(if ~ else、for、while、do ~ while 等),可以很容易地实现结构化的各种基本结构,用来设计结构化程序。

(4)功能强大。C 语言既可用于书写系统软件,也可用于书写应用程序。

(5)生成的目标代码质量高,程序执行效率高。

(6)C 语言允许直接访问物理地址,具有一定的低级语言的特性。

(7)移植性好。在某种计算机系统上的 C 程序,只要少许改动,甚至不作改动就可用于各种型号的计算机和各种操作系统。

## 1.2 C 程序初识

### 1.2.1 简单 C 程序案例

为了让初学者对 C 语言程序有初步的认识,下面通过几个简单的案例,介绍 C 程序设计语言的特点以及 C 源程序的基本构成。

【例 1.1】最简单的 C 语言程序,在屏幕上输出“This is a C Program.”。

```
#include <stdio. h>
void main(void)
```

```
printf("This is a C Program. \n");
```

程序运行结果,如图 1-2 所示。

**【解析】**

其中,第 1 行 #include <stdio. h>为文件包含,基

本输入/输出函数原型及有关信息包含在“stdio. h”库文件中(或称头文件),有关文件包含命令的使用,请参阅第 9 章;

第 2 行 main 被称作“主函数”,在任何一个 C 程序中都必须有一个且只有一个 main 函数。在 main()函数中有很多 C 语句,用一对大括号括起来,称为函数体。

第 3 行左大括号“{”代表开始,最后一行右大括号“}”代表结束;

在本例的程序中,函数体中只包括一条语句,printf()是 C 语言中的输出函数,作用是将双引号之间的内容原样输出到屏幕上。“\n”是换行符,即在输出“This is a C Program.”后回车换行。

**【例 1.2】** 求两个整数 a、b 累加和。

```
#include <stdio. h>
void main(void)
{
    int a,b,sum;           /* 定义变量 */
    a=10;b=8;             /* 给变量赋值 */
    sum=a+b;              /* 求累加和 */
    printf(" %d+%d=%d\n",a,b,sum); /* 输出结果 */
}
```

程序运行结果,如图 1-3 所示。

**【解析】**程序中,第 1 行、第 2 行、第 3 行含义同上;第 4 行为变量说明语句,说明本程序需用 3 个整型变量;第 5 行是赋值语句,表明 a、b 变量的初值分别为 10、8;第 6 行将 a+b 的累加和送入变量 sum 中;第 7 行是将运算的结果输出到屏幕上。printf()中的“%d”为格式说明符,表示在这个位置上将用一个整型数值代替。

“/\* ... \*/”表示注释,注释部分不参与也不影响程序的运行,这些注释只是用来帮助人们阅读和理解程序,注释内容可以加在程序的任何位置。

C 程序中的语句最重要的一个特点就是每条基本语句的后面都要有一个分号,作为语句的结束标志。

**【例 1.3】** 输入两个整数,输出两个数中较大的数。

```
#include <stdio. h>
void main(void)
{
    int x,y,imax;
```

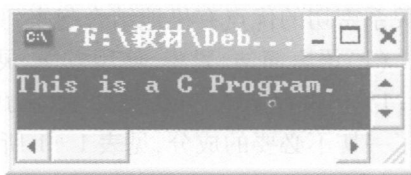


图 1-2 例 1.1 程序运行结果

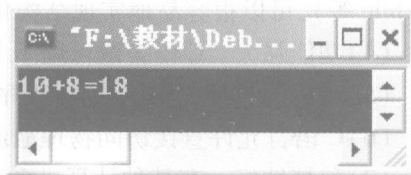


图 1-3 例 1.2 程序运行结果

```
scanf("%d,%d",&x,&y);
if (x>y) imax=x;
else imax=y;
printf("\nThe Max number is %d\n",imax);
}
```

若从键盘输入:5,8 ↵,则程序运行结果,如图 1-4 所示。

【解析】此程序在执行时,先由 scanf() 函数从键盘读取两个数据,这时由用户从键盘上输入 5,8 ↵(↵表示回车键),表示将 2 个整型值,分别存放到 x 和 y 存储单元。if 语句用来比较 x 和 y 值的大小,取其中大数赋给最大值变量 imax。最后通过 printf 函数将大数输出到屏幕上。

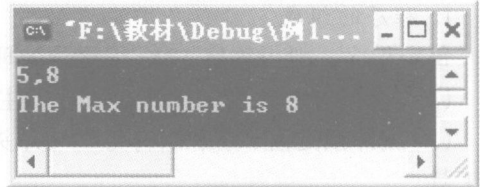


图 1-4 例 1.3 程序运行结果

C 语言实际是一个函数式的语言,可以把一个复杂程序分解成若干个函数(模块)来完成,每个函数来实现特定的功能。对上一题,我们也可以利用主函数和子函数合作的形式来完成。

【例 1.4】 输入两个整数,输出两个数中较大的数(用子函数的方法)。

```
#include <stdio.h>
int Max(int, int); /* 子函数原型声明 */
void main(void)
{
    int x,y,imax;
    scanf("%d,%d",&x,&y); /* 任意输入两个数 */
    imax=Max(x,y);
    printf("\nThe Max number is: %d\n",imax);
}
int Max(int x,int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
    return(z);
}
```

若从键盘输入 20,12 ↵,程序运行结果,如图 1-5 所示。

【解析】此程序包括两个函数,一个是主函数 main(),另一个子函数 Max()用于求两个数 x、y 中较大数。在执行时,先由 scanf() 函数从键盘读取两个数据,这时由用户从键盘上输入 20,12 ↵(↵表示回车键)。此时 x 被赋值 20,y 被

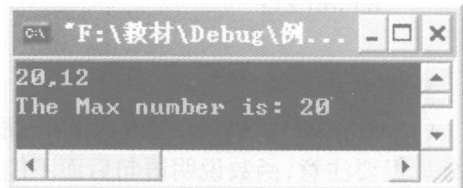


图 1-5 例 1.4 程序运行结果

赋值 12。然后执行函数调用语句:imax = Max(x,y),将 x、y 的值传入 Max 函数中。在 Max 函数中经过判断后,z 中的值就是两个数的较大值。用 return 语句将 z 的值返回函数调用处,然后继续执行主函数,直到程序结束。

### 1.2.2 C 程序基本构成

通过上述例子可以看出,C 语言程序结构大体上由下面几个部分组成。

#### 1. 程序的组成

每个程序一般由程序主体、编译预处理和注释三部分组成。

(1) 程序主体:函数定义是 C 程序的主体部分,程序的功能由函数来完成。

(2) 编译预处理:每个以符号“#”开头的行,称为编译预处理行,是 C 提供的一种模块工具。

(3) 注释:是用“/\* ... \*/”括起的内容,其作用给程序设计者一种提示或记号。注释内容不参加程序的执行,主要是为了提高程序的可读性。

#### 2. 函数的组成

(1) 每个函数(包括主函数)的定义分为两个部分:函数首部和函数体。

(2) 函数首部包括:返回值的类型,函数名,形式参数表。

函数首部的格式:

返回值的类型 函数名(type 形参 1,type 形参 2,...)

例如:int Max(int x,int y)

(3) 函数体是由一对{}括起的语句序列,其中包括:变量说明部分和实现函数功能的语句两部分组成。

函数体的格式:

{

    变量说明部分

    实现函数功能的语句

}

例如:

```
int Max(int x,int y)    /* 函数的首部 */
{                       /* 代表函数体开始 */
    int z;              /* 变量说明 */
    if (x>y) z=x;      /* 具体实现函数功能的语句 */
    else z=y;
    return(z);
}
```

#### 3. 语句的组成

(1) 语句是组成程序的基本单元。每个语句最后要用分号(;)结尾,分号是语句的组成部分。但要注意,函数说明语句后面不能有分号。如上例中 void main(void) 和 int Max(int x,int y) 后面不能加分号。

(2) C 程序中的语句含有各种关键字、运算符、表达式等。例如:int 为关键字,用来说明



变量的类型; $x>y$  中的“ $>$ ”为关系运算符; $z=x$  为赋值表达式。

(3) C 程序的书写格式比较自由,可以在一行上写若干语句,也可以将一条语句写在多行上。

注意:在 C 语言程序中英文字母是区分大、小写的。在定义变量时,同样的字母的大、小写代表不同的变量,而 C 语言的关键字和基本语句都是用小写字母表示。

#### 4. 程序结构特点

(1) C 程序是由一个或多个函数构成。每个 C 程序有且仅有一个主函数,函数名规定为 `main()` 函数,它由系统指定的。除主函数外,可以有一个或若干个子函数(如 `Max` 函数),并且在主函数中调用它。

(2) 主函数是整个程序的主控模块,是程序的入口,C 程序总是从 `main()` 函数开始执行,最终在 `main()` 函数中结束。`main()` 函数可以放在程序的任意位置。

(3) 函数在执行过程中可以根据程序的需要调用系统提供的库函数,例如在程序用到的 `scanf()`/`printf()` 函数,它们的原型包含在 `stdio.h` 文件中。因而在今后的应用案例中,若要调用系统提供的库函数,则在调用前必须将被包含文件嵌入到源文件中。

(4) C 语言的变量在使用之前必须先定义其数据类型,未经定义的变量不能使用。定义变量类型的语句应在可执行语句的前面,如上例中 `main()` 函数中的第一个语句就是变量定义语句,它必须放在第一个执行语句 `scanf()` 的前面。

(5) C 语言中提供了丰富的函数,被称作库函数。标准 C 中提供了 100 多条库函数, Turbo C 和 MS C 中提供了 300 多条库函数。利用这些系统提供的函数可以非常轻松地编写一些功能强大的程序。

(6) 主函数可以调用任何其他函数;任何非主函数都可以相互调用,但不能调用主函数。

重点:函数是程序的基本单位;语句要以分号作为结束标志。

## 1.3 程序设计方法与算法

### 1.3.1 程序设计方法

从 1.2 节中,我们已经初步了解了 C 语言程序设计的过程以及程序的基本组成。但在程序设计中需要用一定的方法来指导,以便提高程序的可读性、维护性、稳定性及编程的效率。目前有两种重要的程序设计方法:结构化程序设计方法和面向对象程序设计方法。

#### 1. 结构化程序设计方法

结构化程序设计(Structured Programming)方法是由 E. Dijkstra 等人于 1972 年提出来的,它建立在 Bohm、Jacopini 证明的结构定理的基础上。结构定理指出:任何程序逻辑都可以用顺序、选择和循环三种基本结构来表示。实验证明:结构化程序设计方法确实使程序执行效率提高,并且由于减少程序的出错率,从而大大减少了维护的费用。

结构化程序设计有两个主要特点:

(1) 程序总是由三种基本结构组成,即顺序结构、选择结构和循环结构。三种控制结构共有的特点:单入口、单出口;无死语句;无死循环。其流程图如图 1-6 所示。

(2) 自顶向下,逐步求精和模块化是结构化程序设计方法中最典型、最具有代表性的方法。

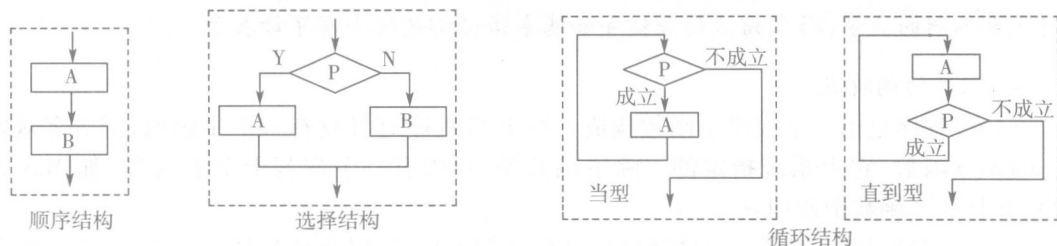


图 1-6 三种控制结构

## 2. 面向对象程序设计方法

面向对象的程序设计是另一种重要的程序设计方法。在面向对象的程序设计方法中,一个程序由若干个对象组成,对象之间通过消息相互作用。

面向对象程序设计语言主要有三个特征:封装性、继承性和多态性。

(1) 封装性:对象是一个封装体,在其中封装了该对象所具有的属性 and 操作,对象作为操作的基本单元,实现了数据和数据处理相结合。一般用类来实现封装。

(2) 继承性:一种支持重用的思想,用现有的类型派生出新的子类。

(3) 多态性:是指在类中定义的属性或行为被特殊继承后,可以具有不同的数据类型或表现出不同的行为。

C 语言是一个面向过程的语言,它所采用的是用结构化程序设计方法;C++是一个面向对象的语言,因而采用的面向对象程序设计方法。C 是 C++的基础,C++是 C 的超集。当我们掌握了 C 语言,再去学习 C++,达到事半功倍的效果。

**重点:**自顶向下,逐步求精和模块化是结构化程序设计中最典型的方法。

### 1.3.2 算法

算法是解决问题的逻辑步骤,是对特定问题求解步骤的一种描述。只有通过算法能够表示的问题,才能通过计算机求解。算法采用形式有多种,通常有自然语言、流程图、伪代码、N-S 图、PAD 图、程序语言等多种方法。能够用算法描述的问题称为可形式化的问题。由于算法仅仅是程序开发过程中的一个逻辑步骤,因而采用何种方法应考虑自己掌握的熟悉程度,但最终都要转换成计算机所能识别的程序。

正确的算法具备三个条件:

- (1) 每个逻辑步骤有可以实现的语句来完成;
- (2) 每个步骤间的关系是惟一的;
- (3) 算法要能终止(防止死循环)。

#### 1. 算法的基本特征

算法是一个有穷规则的集合,这些规则确定了解决某类问题的一个运算序列。对于该