新世纪应用型高等教育
计算机类课程规划教材

C语言程序设计教程

新世纪应用型高等教育教材编审委员会　组编

主编　李　莉　林　庆

大连理工大学出版社

新世纪
计算

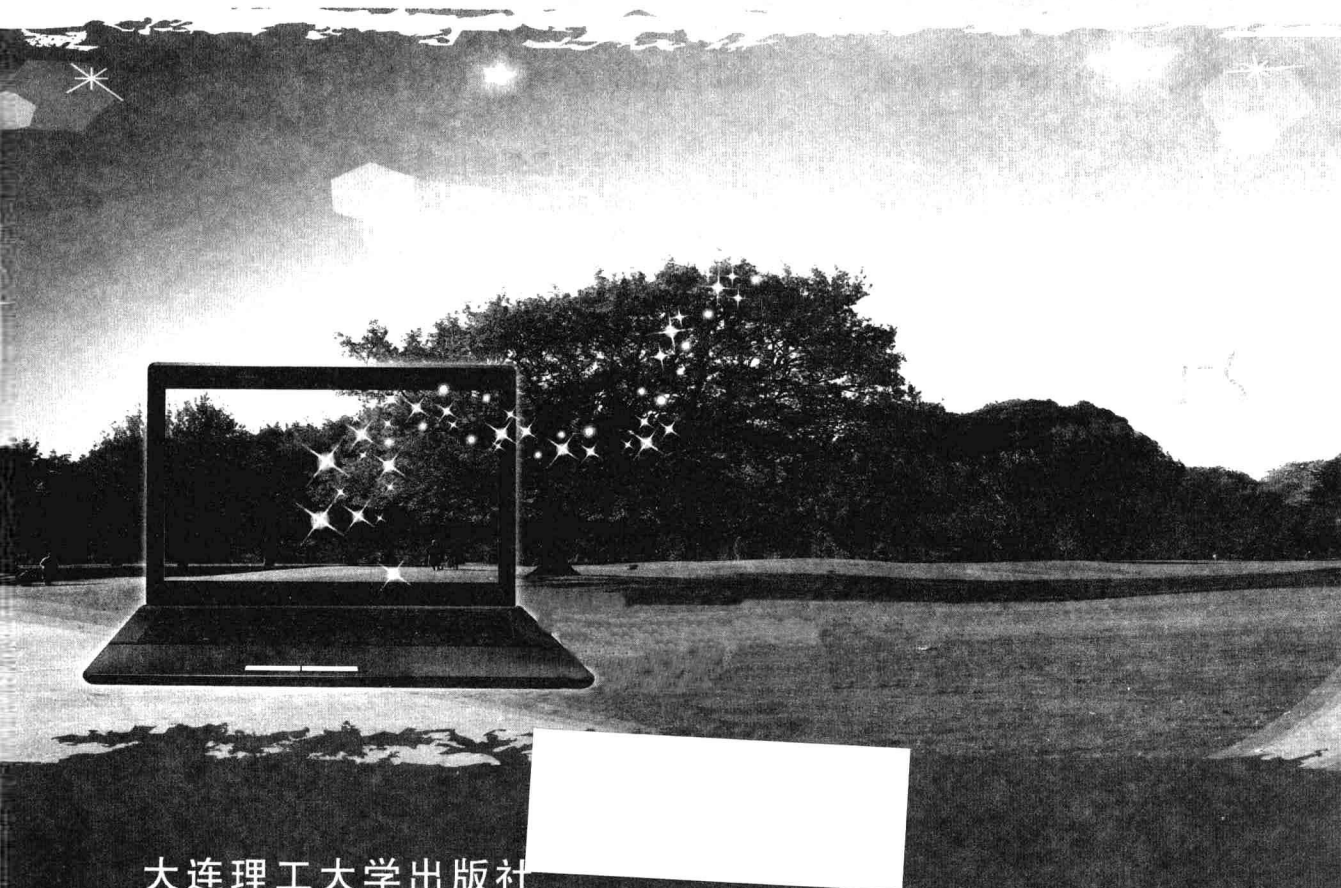
C语言程序设计教程

C YUYAN CHENGXU SHEJI JIAOCHENG

新世纪应用型高等教育教材编审委员会 组编

主编 李 莉 林 庆

参编 梁 军 才智杰



大连理工大学出版社

图书在版编目(CIP)数据

C语言程序设计教程 / 李莉，林庆主编. — 大连：大连理工大学出版社，2012.3
新世纪应用型高等教育计算机类课程规划教材
ISBN 978-7-5611-6648-2

Ⅰ. ①C… Ⅱ. ①李… ②林… 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(2011)第258025号

大连理工大学出版社出版
地址：大连市软件园路80号　邮政编码：116023
发行：0411-84708842　邮购：0411-84703636　传真：0411-84701466
E-mail：dutp@dutp.cn　URL：http://www.dutp.cn
大连业发印刷有限公司印刷　大连理工大学出版社发行

幅面尺寸：185mm×260mm　印张：13.75　字数：318千字
印数：1～3000
2012年3月第1版　2012年3月第1次印刷

责任编辑：杨慎欣　责任校对：潘素君
封面设计：张　莹

ISBN 978-7-5611-6648-2　定　价：28.00元

前言

C语言是目前世界上最流行、使用最广泛的高级程序设计语言。它不仅具有丰富灵活的控制及数据结构、简洁高效的表达式语句、清晰明了的程序结构以及良好的可移植性等优点，还可以直接操纵计算机硬件。因此，C语言既适合开发系统程序，又适合开发应用程序，深受广大计算机应用人员的青睐。目前，绝大多数高校的计算机专业、理工类非计算机专业都将C语言作为计算机程序设计语言的首选语言。

不过，现有的教材一般围绕C语言体系本身，比较注重语法知识讲解，并辅以一些编程技巧的介绍，而在学生编程方法的讲解和编程能力的训练等方面不够突出。学习程序设计既要熟练地掌握和使用编程语言，又要掌握程序设计方法。后者是程序设计的难点，也是重点。程序设计是一项严密的逻辑思维活动，可以完全独立于具体的编程语言，不受它所依托的具体语言的限制。程序设计涉及很多认识上的技巧，例如，对操作环境和相关开发工具的熟悉，对数据结构、算法的合理运用和测试，对机器内部工作原理的了解等。因此，如果在介绍程序设计语言时只侧重语法知识介绍，就会误导读者，使读者误以为学习程序设计就是记住那些语法规范，而忽略了对程序设计方法的训练和良好程序设计习惯的培养，往往导致学习完C语言后，仅学会了一些C语言的语法，编写不出简单的应用程序，甚至在识读别人编写的简单程序上也存在困难。

作者根据多年在高校从事计算机程序设计语言教学的经验，编写了这本《C语言程序设计教程》。与当前已经出版的其他C语言教材相比，本书具有以下特点：

1. 本书吸取了其他众多同类C语言教材的优点，章节安排由浅入深、循序渐进。重视编程能力的培养，围绕着结构化与模块化程序设计这个中心，以程序设计为主线，在深入浅出介绍C语言的基本语法规则的同时，通过精心设计的例题，着重介绍C语言程序设计的基本方法，加强了结构化程序设计和常用算法的训练。这样可使读者既掌握C语言基础知识，又能掌握程序设计的基本方法。

2. 突出应用与实用。本教材以简短的篇幅介绍C语言中最基本、最常用的内容，同时精心设计一些C语言的编程实例

(如分类统计、排序、哥德巴赫猜想、查找、插入、删除、单词数统计、字符加密解密、文件加密保护等),对所讲述的原理、概念加以辅助说明。学生可以通过这些实例加深对C语言编程的基本原理、方法的掌握与理解。通过实例分析,并加以编程实现,使学生既掌握了C语言的内容,进行了开发实用软件的训练,更激发了他们探索C语言奥妙的兴趣,能达到事半功倍的效果。

3. 注重改革实践教学。C语言程序设计是一门实践性很强的课程,上机实验是该课程的重要教学内容,只有通过大量的编程练习,才能在实践中掌握语言知识,培养程序设计实践能力。本书每个章节遵循从"知识要点"到"具体内容"到"应用与提高"再到"习题"的思路,特别注重改革实践教学。每一章后精心挑选和设计的多种类型的练习题,有助于读者复习、巩固所学知识。

本教材由江苏大学多位工作在教学第一线并具有丰富计算机程序设计教学经验的教师共同编写。其中,李莉、林庆任主编,梁军、才智杰参与了教材的编写,具体编写分工为:第1、11章及附录由才智杰编写,第2、9、10章由梁军编写,第3、7、8章由李莉编写,第4、5、6章由林庆编写。

在书稿的编写过程中得到了南通大学程显毅教授,江苏大学晏立教授、王昌达副教授的帮助和支持,他们对本书提出了很多有益的建议。江苏大学计算机学院基础部全体教师对本书的编写给予了大力支持。值本书出版之际,向他们表示衷心感谢。

本书可作为高等院校计算机专业及理工类非计算机专业学生学习C语言程序设计的教材,也可作为有关工程技术人员和计算机爱好者学习C语言程序设计的参考书。

尽管我们在本教材的编写过程中付出了很大的努力,但由于水平有限,加之计算机学科教材更新速率较快,书中难免有不妥之处,恳请广大读者批评指正,以便我们在修订中不断改进和完善。

所有意见和建议请发往:dutpbk@163.com
欢迎访问我们的网站:http://www.dutpbook.com
联系电话:0411-84707492 84706104

编 者
2012年3月

目　录

第1章

C语言概述

知识要点

- 算法分类{数值运算算法 / 非数值运算算法}
- 算法的特性以及表示
- C语言程序结构
- C语言的数据类型
- 常量与变量(整型、实型及其相对应的存储形式)

1.1 学生成绩简单处理程序实例

【例 1-1】 输入10位学生本学期三门功课的成绩,输出一张二维表,统计每门功课的平均成绩、及格人数和不及格人数。

源程序如下:

```
#define Student_Num 10          /* 学生数 */
#define Course_Num 3            /* 课程数 */
#include <stdio.h>
/* 输入学生的成绩 */
void In_grade(int grade[Student_Num][Course_Num])
{
    int i,j;
    printf("\n  请输入 %d 名学生 %d 门功课的成绩 \n",Student_Num,Course_Num);
    for(i=0; i<Student_Num; i++)
        for(j=0; j<Course_Num; j++)
          scanf("%d",&grade[i][j]);
}
/* 统计每门功课的平均成绩、及格人数和不及格人数 */
void Stat_grade(int grade[][Course_Num],float average[],int pass[])
{
    int i,j;
    for(j=0; j<Course_Num; j++)
    {
        average[j]=0;
```

```
        for(pass[j]=i=0; i<Student_Num; i++)
        {
            if(grade[i][j]>=60)
              ++pass[j];
            average[j]=average[j]+grade[i][j];
        }
        average[j]=average[j]/Student_Num;
    }
}
/* 输出10名学生的每门功课的平均成绩并统计及格人数和不及格人数 */
void Out_grade(int(*grade)[Course_Num],float average[],int *pass)
{
    int i,j;
    printf("\n%8c学生的成绩表\n",' ');
    for(i=0;i<33;i++)
        printf("=");
    printf("\n");
    printf("%3c序号%5c",' ',' ');
    for(j=1; j<=Course_Num; j++)
        printf("课程%d ",j);
    printf("\n");
    for(i=0;i<33;i++)
        printf("-");
    printf("\n");
    for(i=0; i<Student_Num; i++)
    {
        printf("%6d%4c",i+1,' ');
        for(j=0; j<Course_Num; j++)
            printf("%3c%4d",' ',grade[i][j]);
        printf("\n");
    }
    printf("\n平均成绩%3c",' ');
    for(j=0; j<Course_Num; j++)
        printf("%2c%5.1f",' ',*average++);
    printf("\n及格人数");
    for(j=0; j<Course_Num; j++)
        printf("%3c%4d",' ',pass[j]);
    printf("\n不及格人数");
    for(j=0; j<Course_Num; j++)
        printf("%3c%4d",' ',Student_Num-pass[j]);
    printf("\n");
    for(i=0;i<33;i++)
```

```
        printf("=");
}
void main ()
{
    int grade[Student_Num][Course_Num];   /* 成绩 */
    float average[Course_Num];            /* 平均成绩 */
    int pass[Course_Num];                 /* 及格人数 */
    In_grade(grade);                      /* 输入学生的成绩 */
    /* 统计每门功课的平均成绩、及格人数和不及格人数 */
    Stat_grade(grade,average,pass);
    /* 输出学生的成绩、每门功课的平均成绩、及格人数和不及格人数 */
    Out_grade(grade,average,pass);
}
```

输入如下数据：

83　76　82　64　65　78
53　78　53　89　85　79
93　81　87　81　67　72
88　89　69　77　95　48
82　78　71　65　76　76

运行结果如图 1-1 所示。

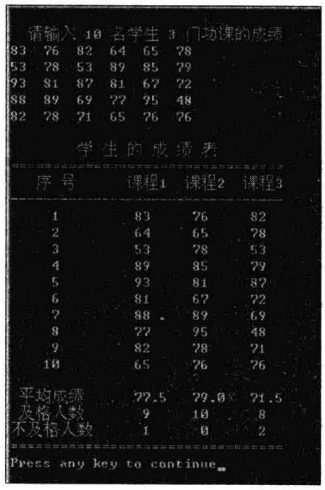


图 1-1　例 1-1 运行结果

1.2　C 语言程序设计

1.2.1　程序与程序设计

上述实例需要用计算机完成，按步骤用诸条指令的形式描述出来，并把这些指令存

放在计算机的内部存储器中，需要结果时向计算机发出一个运行命令，计算机就会自动逐条地执行操作，直至全部指令执行完。被连续执行的一条条指令的集合称为计算机的程序。因此，程序设计就是针对某一要解决的问题，设计出解决该问题的计算机指令序列，并编译和调试程序的方法和过程。所有指令都是二进制编码，用它编写程序时既难记忆，又难掌握，因此研制了一种接近于数学语言或人的自然语言，同时又不依赖于计算机硬件的计算机高级语言来编写程序，也就是程序设计语言。

程序设计语言分为低级语言和高级语言两大类。低级语言依赖于硬件体系，计算机能识别的机器语言和用助记符代替机器语言操作码的汇编语言等都属于低级语言；高级语言独立于机器，面向人类语言，可移植性好，重用率高，如 FORTRAN、BASIC、C 语言等。用高级语言编写的源程序都不能直接在计算机上执行，需要程序设计语言处理系统对其进行适当的转换，转换为可在计算机上执行的程序。C 语言兼具高级语言和低级语言的优点。

程序设计就是用某种编程语言编写程序的过程。所谓程序，是指用计算机语言对所要解决的问题中的数据以及解决问题的方法和步骤所做的完整而准确的描述。对数据的描述就是指明数据结构形式，对解决问题的方法和步骤的描述即为算法。

计算机科学家沃思(Nikiklaus Wirth)提出了关于程序的著名公式：

程序＝数据结构＋算法

一个程序应包括：

(1)数据结构：对数据的描述，在程序中要指定数据的类型和数据的组织形式。

(2)算法：对操作的描述，即操作步骤。

1.2.2 算 法

1. 算法简介

算法是解决问题的方法和步骤。实际上，在日常生活中，我们做任何事情都有一定的方法和步骤，如学生每天上课，理发师做每一种发型，工厂生产一种型号的产品等。

利用计算机解决问题，首先要设计出适合计算机执行的算法，此算法包含的步骤必须是有限的，且每一步都必须是明确的，最终能被计算机执行而得到结果。

算法可分为两类：

(1)数值运算算法：对问题求数值解，通过运算得出一个具体值，如求方程的根等，此类算法一般有现成的模型，较为成熟。

(2)非数值运算算法：用于事务管理，如图书检索、人事管理等。

根据实际问题设计算法时，还要尽量考虑用重复的步骤去实现，使算法简明扼要，通用性强，这不仅能减少编写和调试程序的时间，还能减少程序本身所占用的内存空间。

算法应该具有以下特性：

(1)有穷性：一个算法应包含有限的操作步骤而不能是无限的。

(2)确定性：算法中每一个步骤应当是确定的，不能具有二义性。

(3)有零个或多个输入：通常，处理的数据对象需要通过外接设备输入来获得数据。

(4)有一个或多个输出：算法的目的就是得到结果并将结果输出。没有输出的算法是无意义的。

(5)有效性:算法中每一个步骤应当能有效地执行,并得到确定的结果。

【例 1-2】 现有标有 A 和 B 标签的两个杯子,分别盛放酒和醋,求将它们互换的算法。

解析:该算法为非数值运算算法。实际解决这类问题时需借助于第三个 C 杯,先将 A 杯的酒倒入 C 杯,然后将 B 杯的醋倒入 A 杯,最后将 C 杯的酒倒入 B 杯,实现两杯中酒与醋互换。其交换步骤如下:

S1:C ← A

S2:A ← B

S3:B ← C

【例 1-3】 已有 10 个整数,求这 10 个数中的最大数。

解析:该算法为非数值运算算法。通常采用数与数间进行比较的方法求出最大数。引入两个变量 a 和 max,变量 max 存放最大数,变量 a 存放待比较的数。其算法如下:

S1:将第 1 个数同时赋给变量 a 与 max,第 1 个数视为最大数;

S2:输入下一个数存放在变量 a 中;

S3:比较 max 中的数与变量 a 中的数的大小。若 max 小于 a,执行 S4,否则执行 S5;

S4:将变量 a 中的数赋给变量 max,之后执行 S5;

S5:判断 10 个数是否都已比较完,若没有则返回 S2,否则执行 S6;

S6:输出 max,结束程序。

【例 1-4】 计算函数 f(x)的值。函数 f(x)为:

$$f(x)=\begin{cases}x^2+1 & x\geqslant 0\\ 4x-2 & x<0\end{cases}$$

解析:该算法为数值运算算法,求函数 f(x)的值。给定自变量 x 值,通过判断其值是大于等于 0 还是小于 0 来决定用哪个算式求函数 f(x)的值。求值步骤如下:

S1:输入 x 的值;

S2:判断 x 的值是否小于 0,若成立,执行 S3,否则执行 S4;

S3:以算式 $4x-2$ 求值,之后执行 S5;

S4:以算式 x^2+1 求值,之后执行 S5;

S5:输出函数值,算法结束。

2. 算法的表示

表示一个算法,可以用不同的方法,常用的方法有:自然语言、传统流程图、N-S 流程图、伪代码、计算机语言等。用流程图表示算法直观形象,易于理解。传统流程图由图1-2中所示的几种基本框图组成。

图 1-2　传统流程图的图标

传统流程图对流程线的使用没有严格限制，因此难以实现结构化程序设计。为了限制流程线的滥用，流程图设计时提出三种基本结构：顺序结构、选择结构和循环结构。

程序设计语言中最典型的三种基本结构的流程图如图 1-3、图 1-4 和图 1-5 所示。

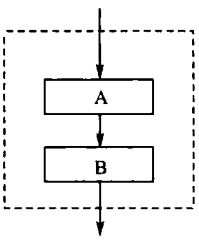
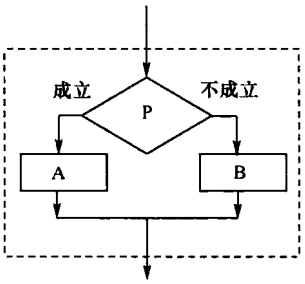


图 1-3　顺序结构



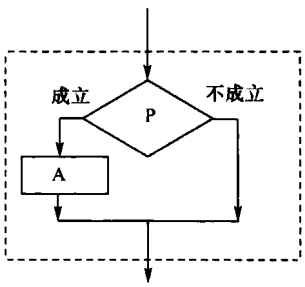
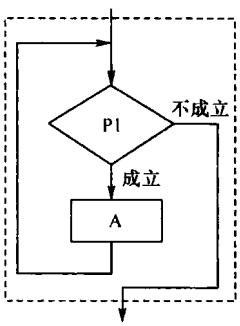


图 1-4　选择结构



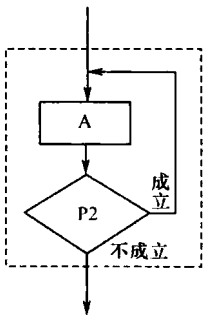


图 1-5　循环结构

1.2.3　C 语言程序结构

1. C 语言程序的组成

C 语言程序由函数组成，函数是 C 语言程序的基本组成单位。一个 C 源程序至少包含一个主函数，记作 main 函数。换句话说，C 源程序至少包含一个 main 函数，或者有且仅有一个 main 函数和若干个子函数。C 语言源程序文件的扩展名为. c。

函数可以由用户定义，也可以是系统提供的库函数。main 函数和若干其他函数在结构和功能上相对独立，但又存在一定的联系，C 源程序的执行总是从 main 函数开始，再通过 main 函数对其他函数的调用以及其他函数相互之间的调用实现各函数功能，然后层层返回各级调用点，最后返回调用的起点(main 函数)。即 C 语言程序总是从 main 函数开始执行，并以 main 函数结束。

2. 函数的组成

一个函数由函数首部和函数体组成。

函数定义的一般格式为：

```
类型说明符　函数名(形参列表)　　/*函数首部的定义*/
{
   说明部分;
   执行部分;                    /*函数体*/
}
```

函数体通常包括说明部分和执行部分，说明部分和执行部分都由若干条语句组成。C 语言的语句由表达式加分号组成，也就是说每条语句都必须以分号结尾。

如，例 1-1 中输入学生成绩的函数定义：

```
类型说明符  函数名        形参列表
void  In_grade(int grade[Student_Num][Course_Num])      /*函数首部的定义*/
 {
     int i, j;                /* 说明部分 */
     printf("\n%d %d\n", Student_Num, Course_Num);
     for(i=0; i<Student_Num; i++)                        /*执行部分*/
       for(j=0; j<Course_Num; j++)
           scanf(" %d", &grade [i][j]);
 }
```

如，实例中 main 函数的定义：

```
类型说明符  函数名      形参列表为空
   void   main()
   {
     int grade [Student _Num] [Course_ Num];
     float average[Course_Num];                          /*说明部分*/
     int pass[Course_Num];
     In_grade (grade);
     Stat_grade (grade, average, pass);                  /*执行部分*/
     Out_grade (grade, average, pass);
   }
```

3. C 语言源程序书写的基本规则

(1)C 语言的书写格式比较自由，一行可以写多条语句，一条语句也可以写在多行上。通常采用一行一条语句的书写方式。

(2)C 语言的语句必须以分号“;”结尾。

(3)C 语言区分大小写。标识符常用小写字母，库函数、C 语言关键字等用小写字母，而符号常量通常用大写字母。同一字母的大小写视为两个不同的字符。

(4)C 语言的注释符常以"/ * "开头并以" * /"结尾。在"/ * "和" * /"之间的即为注释信息。程序编译时,不对注释信息作任何处理。注释可出现在程序中的任何位置。注释用来提示信息或解释程序的意义,可提高程序的可读性。在调试程序时对暂不使用的语句也可用注释符括起来,使编译跳过不作处理,待调试结束后再去掉注释符。

1.3 数据类型

C 语言提供了丰富的数据类型。在程序设计时所有的数据都必须定义数据类型。数据类型描述了某种数据的特性,其表现形式、占据存储空间的多少以及构造特点等。不同的数据类型具有不同的取值范围和存储格式。在 C 语言中,数据类型可分为基本数据类型、构造类型、指针类型和空类型四大类。如图 1-6 所示 。

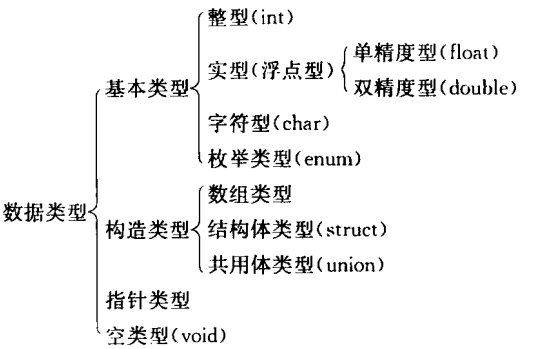


图 1-6 C 语言的数据类型

1.4 常量与变量

1.4.1 常量和符号常量

在程序中,不同类型的数据可以以常量或变量两种形式出现。常量是指在程序运行中其值不能被改变的量。

(1)直接常量或字面常量:这一类常量直接可以从其字面形式来判别。如 12、0、-3 等为整数,即为整型常量;4.6、-1.23 等为实数,即为实型常量;'a'、'b'等为字符常量;"china"、"123@%"等为字符串常量。

(2)符号常量:用一个符号来表示一个常量,该符号称为符号常量。

在 C 语言中,用编译预处理命令来定义符号常量。其定义的一般格式为:

```
#define 标识符 常量
```

其中,define 是一条预处理命令(预处理命令都以"#"开头),称为宏定义命令,功能是把该标识符定义为其后的常量值,一经定义,以后在程序中所有出现该标识符的地方均以该常量值代替。习惯上符号常量的标识符用大写字母,变量标识符用小写字母,以示区别,其目的是方便程序的阅读。

【例 1-5】 已知某科技书籍每本 20.5 元,计算买 6 本所需的钱。