高职高专规划教材

C YUYAN CHENGXU SHEJI YU SHIXUN

C语言程序设计与实训

闻红军 王 鹏 主编

冶金工业出版社
http://www.cnmip.com.cn

高职高专规划教材

C语言程序设计与实训

闻红军　王　鹏　主　编
罗大伟　齐　宁　副主编

北　京
冶　金　工　业　出　版　社
2008

内 容 提 要

本书详细阐述了C语言程序设计的基本概念及技术基础，内容主要包括：C语言基础、数据类型、运算符、C语言的输入与输出、C程序流程设计、模块化程序设计等。书中收录了大量的经典实例，旨在提高学生的程序设计分析及操作能力。附录中收录了大量的实例及常用函数等供读者参考使用。

本书以技能训练为主，以基本理论学习为辅，内容编排由浅入深，循序渐进，便于学习掌握，是学习计算机编程的基础教材。本书可作为高职高专院校教学用书，也可供计算机编程爱好者学习使用，同时还可供有一定编程基础的技术人员参考。

图书在版编目(CIP)数据

C语言程序设计与实训/闻红军，王鹏主编. —北京：冶金工业出版社，2008.3

高职高专规划教材

ISBN 978-7-5024-4493-8

Ⅰ.C… Ⅱ.①闻… ②王… Ⅲ.语言—程序设计—高等学校：技术学校—教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2008)第018730号

出 版 人 曹胜利

地 址 北京北河沿大街嵩祝院北巷39号，邮编100009

电 话 (010)64027926 电子信箱 postmaster@cnmip.com.cn

责任编辑 杨 敏 宋 良 美术编辑 李 心 版式设计 张 青

责任校对 卿文春 责任印制 丁小晶

ISBN 978-7-5024-4493-8

北京兴华印刷厂印刷；冶金工业出版社发行；各地新华书店经销

2008年3月第1版，2008年3月第1次印刷

787mm×1092mm 1/16；15印张；400千字；229页；1-4000册

30.00元

冶金工业出版社发行部 电话：(010)64044283 传真：(010)64027893

冶金书店 地址：北京东四西大街46号(100711) 电话：(010)65289081

前　言

计算机语言也称程序设计语言（Program Language），即编写计算机程序所用的语言。计算机语言是人和计算机交流信息的工具，它是软件的重要组成部分。

C 语言现在是当今软件工程师最喜爱的语言之一，数以百万计的程序员用它来编写各种数据处理、实时控制、系统仿真和网络通讯等软件。

C 语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，它还可以直接访问内存的物理地址，进行位（bit）一级的操作。由于 C 语言实现了对硬件的编程操作，因此 C 语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发，也可用于应用软件的开发。

本书是作者在总结多年教学经验及应用实践经验的基础上编写而成的。书中系统介绍了 C 语言的大部分常用功能，本书以实践性操作为主，以精练的理论为辅，课程自成体系。全书共分 7 章，其内容主要包括：C 语言基础、数据类型、运算符、C 语言的输入与输出、C 程序流程设计、模块化程序设计和典型实例。同时书中收录了大量的经典实例及常用函数等供读者参考使用。

本书内容翔实、通俗易懂、以实例为主，试图通过实例的操作过程，使读者掌握 C 语言编程的基本方法和技巧。本书第 1 章～第 6 章都有习题和实训，可以使读者在实践中学习，并可巩固、检验所学的内容。通过对本书的认真学习，读者可以基本做到精通 C 语言编程。本书作为高职高专教材，可满足大约 60 学时教学使用，其中实验和实训大约占 20 个学时，典型实例部分可作为选修内容，根据各学校的具体情况安排。本书可供计算机编程爱好者学习使

用，同时也可供计算机程序设计工作者参考。

本书由吉林电子信息职业技术学院闻红军负责统稿和定稿，其中闻红军编写了第1章、第7章和附录，潘谈和王婷婷参与编写其中的部分内容；王鹏编写第2章和第3章，其中的部分内容由朱岩编写；罗大伟编写第5章和第6章，其中的部分内容由郑茵编写；齐宁编写第4章。

由于作者水平有限，加之计算机技术的发展十分迅速，书中难免存在疏漏及不妥之处，请广大读者批评指正。作者的电子邮箱为：whj0525@163.com。

编 者

2008年1月

目　录

1 C语言基础

1.1 程序设计语言

计算机语言也称程序设计语言（Program Language），即编写计算机程序所用的语言。计算机语言是人和计算机交流信息的工具，它是软件的重要组成部分。计算机语言粗略地分为机器语言、汇编语言和高级语言。高级语言是接近人类习惯使用的自然语言和数学语言的计算机程序设计语言。它独立于计算机，用户可以不了解机器指令，也可以不必了解机器的内部结构和工作原理，就能用高级语言编程序。高级语言通用性好、易学习、易使用、不受机器型号的限制，而且易于交流和推广。早期电脑都直接采用机器语言，即用“0”和“1”为指令代码来编写程序，读写困难，编程效率极低。为了方便编程，随即出现了汇编语言，虽然提高了效率，但仍然不够直观简便。从1954年起，电脑界逐步开发了一批“高级语言”，采用英文词汇、符号和数字，遵照一定的规则来编写程序。高级语言诞生后，软件业得到突飞猛进的发展。

1.1.1 程序设计语言的发展

1953年12月，IBM公司程序师约翰·巴科斯（J. Backus）写了一份备忘录，建议为IBM 704设计一种全新的程序设计语言。巴科斯曾在“选择顺序控制计算机”（SSEC）上工作过3年，深深体会到编写程序的困难性。他说：“每个人都看到程序设计有多昂贵，租借机器要花去好几百万，而程序设计的费用却只会多不会少。”巴科斯的目标是设计一种用于科学计算的“公式翻译语言”（FORmula TRANslator）。他带领一个13人小组，包括有经验的程序员和刚从学校毕业的青年人，在IBM 704电脑上设计出编译器软件，于1954年完成了第一个电脑高级语言——FORTRAN语言。1957年，西屋电气公司幸运地成为FORTRAN的第一个商业用户，巴科斯给了他们一套存储着语言编译器的穿孔卡片。以后，不同版本的FORTRAN纷纷面世，1966年，美国统一了它的标准，称为FORTRAN 66语言。40多年过去，FORTRAN仍然是科学计算选用的语言之一，巴科斯因此摘取了1977年度图林奖。FORTRAN广泛运用的时候，还没有一种可以用于商业计算的语言。美国国防部注意到这种情况，1959年5月，五角大楼委托格雷斯·霍波博士领导一个委员会，开始设计面向商业的通用语言（COmmon Business Oriented Language），即COBOL语言。COBOL最重要的特征是语法与英文很接近，可以让不懂电脑的人也能看懂程序；编译器只需做少许修改，就能运行于任何类型的电脑。委员会一个成员害怕这种语言的命运不会太长久，特地为它制作了一个小小的墓碑。然而，COBOL语言却幸存下来。1963年，美国国家标准局将它进行了标准化。用COBOL写作的软件，要比其他语言多得多。1958年，一个国际商业和学术计算机科学家组成的委员会在瑞士苏黎世开会，探讨如何改进FORTRAN语言，并且设计一种标准化的电脑语言，巴科斯也参加了这个委员会。1960年，该委员会在1958年设计的基础上，定义了一种新的语言版本——国际代数语言ALGOL 60，首次引进了局部变量和递归的概念。ALGOL语言没有被广泛运用，但它演变为其他程序语言的概念基础。20世纪60年代中期，美国达特默斯学院约翰·凯梅尼（J. Kemeny）和托马斯·卡茨

(T. Kurtz) 认为，像 FORTRAN 那样的语言都是为专业人员设计，而他们希望能为无经验的人提供一种简单的语言，特别希望那些非计算机专业的学生也能通过这种语言学会使用电脑。于是，他们在简化 FORTRAN 的基础上，研制出一种“初学者通用符号指令代码”(Beginner's All Purpose Symbolic Intruction Code)，简称 BASIC。由于 BASIC 语言易学易用，它很快就成为最流行的电脑语言之一，几乎所有小型电脑和个人电脑都在使用它。经过不断改进后，它一直沿用至今，出现了像 QBASIC、VB 等新一代 BASIC 版本。1967 年，麻省理工学院人工智能实验室希摩尔·帕伯特 (S. Papert)，为孩子设计出一种叫 LOGO 的电脑语言。帕伯特曾与著名的瑞士心理学家皮亚杰一起学习。他发明的 LOGO 最初是个绘图程序，能控制一个“海龟”图标，在屏幕上描绘爬行路径的轨迹，从而完成各种图形的绘制。帕伯特希望孩子不要机械地记忆事实，强调创造性的探索。他说：“人们总喜欢讲学习，但是，你可以看到，学校的多数课程是记忆一些数据和科学事实，却很少着眼于真正意义上的学习与思考。”他用 LOGO 语言启发孩子们学会学习，在马萨诸塞州列克星敦，一些孩子用 LOGO 语言设计出了真正的程序，使 LOGO成为一种热门的电脑教学语言。1971 年，瑞士联邦技术学院尼克劳斯·沃尔斯 (N. Wirth) 教授发明了另一种简单明晰的电脑语言，这就是以帕斯卡的名字命名的 PASCAL 语言。PASCAL语言语法严谨，层次分明，程序易写，具有很强的可读性，是第一个结构化的编程语言。它一出世就受到广泛欢迎，迅速地从欧洲传到美国。沃尔斯一生还写作了大量有关程序设计、算法和数据结构的著作，因此，他获得了 1984 年度图林奖。PASCAL 语言不仅用作教学语言，而且也用作系统程序设计语言和某些应用。所谓系统程序设计语言，就是用这种语言可以编写系统软件，如操作系统、编译程序等。PASCAL 语言是一种安全可靠的语言。不过它的后继者 Delphi 已经成为最有生命力的编程语言之一，同时具有 VB 和 C 语言的优点，成为聪明的编程者的必然选择。1983 年度的图林奖则授予了 AT&T 贝尔实验室的两位科学家邓尼斯·里奇 (D. Ritchie) 和他的协作者肯·汤姆森 (K. Thompson)，以表彰他们共同发明著名的电脑语言 C。C 语言的设计哲学是“Keep It Simple, Stupid”，因而程序员可以轻易掌握整个 C 语言的逻辑结构而不用一天到晚翻手册写代码。于是，众多的程序员倒向了 C 语言的怀抱，C 语言迅速并广泛地传播开来。C 语言现在是当今软件工程师最宠爱的语言之一。里奇最初的贡献是开发了 Unix 操作系统软件。他说，这里有一个小故事：他们答应为贝尔实验室开发一个字处理软件，要求购买一台小型电脑 PDP-11/20，从而争取到 10 万美元经费。可是当机器购回来后，他俩却把它用来编写 Unix 系统软件。Unix 很快有了大量追随者，特别是在工程师和科学家中间引起巨大反响，推动了工作站电脑和网络的成长。1970 年，作为 Unix 的一项“副产品”，里奇和汤姆森合作完成了 C 语言的开发，这是因为研制 C 语言的初衷是为了用它编写 Unix。这种语言结合了汇编语言和高级语言的优点，大受程序设计师的青睐。1983 年，贝尔实验室另一研究人员比加尼·斯楚士舒普 (B. Stroustrup)，把 C 语言扩展成一种面向对象的程序设计语言 C++。如今，数以百万计的程序员用它来编写各种数据处理、实时控制、系统仿真和网络通讯等软件。斯楚士舒普说：“过去所有的编程语言对网络编程实在太慢，所以我开发 C++，以便快速实现自己的想法，也容易写出更好的软件。”1995 年，《BYTE》杂志将他列入计算机工业 20 个最有影响力的人的行列。

1.1.2　程序设计语言的支持环境

每种语言都有自己的支持环境，而 C 语言的支持环境最简单也最多，而现在最普遍用的就是 Visual C++6.0 环境。Visual C++6.0 是 Microsoft 公司的产品，是一个使用广泛的应用项目开发环境。它既可用于管理基于 Windows 的应用项目，也可用于管理基于 DOS 的应用项目。

基于 DOS 的应用系统也称为控制台应用系统，这里我们主要结合控制台应用系统的开发过程使用 Visual C++6.0 开发环境，实现我们的目的。

在此之前，使用较多的是在 DOS 环境下的 Turbo C 环境，我们在本书的附录里面对 Turbo C2.0 环境做了介绍，以方便大家的使用。Visual C++6.0 环境在本章后面的实训里有详细的操作方法，本书以此方法为主。

1.2 C 程序的基本结构

C 语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构。它还可以直接访问内存的物理地址，进行位（bit）一级的操作。由于 C 语言实现了对硬件的编程操作，因此 C 语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发，也适合于应用软件的开发。此外，C 语言还具有效率高，可移植性强等特点。因此广泛地移植到了各类型计算机上，从而形成了多种版本的 C 语言。

1.2.1 C 程序的结构特点

（1）一个 C 语言源程序可以由一个或多个源文件组成。

（2）每个源文件可由一个或多个函数组成。

（3）一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。

（4）源程序中可以有预处理命令（include 命令仅为其中的一种），预处理命令通常应放在源文件或源程序的最前面。

（5）每一个说明，每一个语句都必须以分号结尾。但预处理命令，函数头和花括号“｝”之后不能加分号。

（6）标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

1.2.2 C 程序的书写格式

对于编程，应该养成良好的习惯，书写程序时应遵循一定的规则，使程序清晰易读，是书写程序的基本原则。从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

（1）一个说明或一个语句占一行。

（2）用 ｛｝ 括起来的部分，通常表示了程序的某一层次结构。｛｝ 一般与该结构语句的第一个字母对齐并单独占一行。

（3）低一层次的语句或说明，应比高一层次的语句或说明缩进若干格后书写，形成退格。以便看起来更加清晰，增加程序的可读性。在编程时应力求遵循这些规则，以养成良好的编程风格。

但要注意，C 语言本身对格式并没有严格的要求，只要语法正确，无论怎么写，都可以正常执行。

1.2.3 C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空白符、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

（1）字母。小写字母a～z共26个，大写字母A～Z共26个。

（2）数字。0～9共10个。

（3）空白符。空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时，只起间隔作用，编译程序忽略它们的存在。因此在程序中使用空白符与否，对程序的编译不发生影响，但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

（4）标点和特殊字符。

1.2.4　C语言的词法

在C语言中使用的词汇分为六类：标识符、关键字、运算符、分隔符、常量、注释符等。

1.2.4.1　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。除库函数的函数名由系统定义外，其余都由用户自定义。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，并且其第一个字符必须是字母或下划线。

以下标识符是合法的：

a,x,_3x。BOOK_1,sum5

以下标识符是非法的：

3s（以数字开头）

s*T（出现非法字符*）

-3x（以减号开头）

bowy-1（出现非法字符-（减号））

在使用标识符时还必须注意以下几点：

（1）标准C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。

（2）在标识符中，大小写是有区别的。例如BOOK和book是两个不同的标识符。

（3）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。因此，命名应尽量有相应的意义，以便阅读理解，做到“顾名思义”。

1.2.4.2　关键字

关键字是由C语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C语言的关键字分为以下几类：

（1）类型说明符。用于定义、说明变量、函数或其他数据结构的类型。如int、double等。

（2）语句定义符。用于表示一个语句的功能。如if else就是条件语句的语句定义符。

（3）预处理命令字。用于表示一个预处理命令。如include。

1.2.4.3　运算符

C语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

1.2.4.4　分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如把int a写成inta，C编译器会把inta当成

一个标识符处理，其结果必然出错。

1.2.4.5 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第2章中将专门给予介绍。

1.2.4.6 注释符

C语言的注释符是以“/*”开头并以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译跳过不作处理，待调试结束后再去掉注释符，这称为保留字。用户定义的标识符不应与关键字相同。

1.2.5 常用基本结构及语法

从程序流程的角度来看，程序可以分为三种基本结构，即顺序结构、分支结构、循环结构。这三种基本结构可以组成所有的各种复杂程序。C语言提供了多种语句来实现这些程序结构。这里简单介绍这些基本语句及其应用，使读者对C程序有一个初步的认识，为后面各章的学习打下基础。

（1）顺序结构。C程序的程序是按由前至后的顺序执行的。

（2）选择结构。按条件不同，决定某些语句是否执行，典型的是if语句。格式如下：

```
if（表达式）
  语句1；
else
  语句2；
```

其语义是：如果表达式的值为真，则执行语句1，否则执行语句2。

（3）循环结构。在一定条件下，反复执行某些语句，以达到简化程序的目的。典型的是while语句，格式如下：

while语句的一般形式为：while（表达式）语句；其中表达式是循环条件，语句为循环体。

while语句的语义是：计算表达式的值，当值为真（非0）时，执行循环体语句。

1.3 用函数组装C程序

C源程序是由函数组成的，函数是C源程序的基本模块，通过对函数模块的调用实现特定的功能。在一个程序中必须有且只有一个主函数main（），主函数是程序执行的开始。实用程序往往由多个函数组成。C语言中的函数相当于其他高级语言的子程序。C语言不仅提供了极为丰富的库函数（如Turbo C，MS C都提供了三百多个库函数），还允许用户建立自己定义的函数。用户可把自己的算法编成一个个相对独立的函数模块，然后用调用的方法来使用函数。

可以说C程序的全部工作都是由各式各样的函数完成的，所以也把C语言称为函数式语言。由于采用了函数模块式的结构，C语言易于实现结构化程序设计。使程序的层次结构清晰，便于程序的编写、阅读、调试。

1.3.1 使用库函数

库函数由C系统提供，用户无需定义，也不必在程序中作类型说明，只需在程序前有包含了该函数原型的头文件即可在程序中直接调用。如：printf、scanf、getchar、putchar、gets、puts、strcat等函数均属此类，具体使用方法请见附录。

1.3.2　使用自定义函数

用户自定义函数为用户按需要而编写的函数。对于用户自定义函数，不仅要在程序中定义函数本身，而且在主调函数模块中还必须对该被调函数进行类型说明，然后才能使用，在以后的章节中会有详细的介绍。

1.3.3　C程序的组成形式

为了说明C语言源程序结构的特点，先看以下的程序。虽然有关内容还未介绍，但可从这些例子中了解到组成一个C源程序的基本部分和书写格式。

```
main( )
{
  printf( "世界,您好! \n" );
}
```

main是主函数的函数名，表示这是一个主函数。每一个C源程序都必须有，且只能有一个主函数（main函数）。函数调用语句中printf函数的功能是把要输出的内容送到显示器去显示。printf函数是一个由系统定义的标准函数，可在程序中直接调用。

```
#include "stdio. h"
#include "math. h"   /* include 为文件包含命令*/
main( )
{
  double x,s;  /*定义两个实数变量,以备后面程序使用*/
  printf( "input number:\n" );  /*显示提示信息*/
  scanf( "%lf" ,&x);  /*从键盘获得一个实数 x*/
  s = sin(x);  /*求 x 的正弦,并把它赋给变量 s*/
  printf( "sine of %lf is %lf\n" ,x,s);  /*显示程序运算结果*/
}  /* main 函数结束*/
```

程序的功能是从键盘输入一个数x，求x的正弦值，然后输出结果。在main（）之前的两行称为预处理命令（详见后面）。预处理命令还有其他几种，这里的include称为文件包含命令，其意义是把尖括号< >或引号"" 内指定的文件包含到本程序来，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h。因此也称为头文件或首部文件。C语言的头文件中包括了各个标准库函数的函数原型。因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。在本例中，使用了三个库函数：输入函数scanf、正弦函数sin、输出函数printf。sin函数是数学函数，其头文件为math. h文件，因此在程序的主函数前用include命令包含了math. h。scanf和printf是标准输入输出函数，其头文件为stdio. h，在主函数前也用include命令包含了stdio. h文件。

在例题中的主函数体中又分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明。例题中未使用任何变量，因此无说明部分。C语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，这与解释型的BASIC语言是不同的。说明部分是C源程序结构中很重要的组成部分。本

例中使用了两个变量 x，s，用来表示输入的自变量和 sin 函数值。由于 sin 函数要求这两个量必须是双精度浮点型，故用类型说明符 double 来说明这两个变量。说明部分后的四行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 printf 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第二行为输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 x 中。第三行是调用 sin 函数并把函数值送到变量 s 中。第四行是用 printf 函数输出变量 s 的值，即 x 的正弦值。最后程序结束。

```
printf("input number:\n");
scanf("%lf",&x);
s = sin(x);
printf("sine of %lf is %lf\n",x,s);
```

运行本程序时，首先在显示器屏幕上给出提示串 input number，这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数，如 5，按下回车键，接着在屏幕上给出计算结果。

1.4 实训

1.4.1 应用项目的建立

一个应用项目（Project）是由若干编译单元（简称单元）组成的，而每个编译单元由一个程序文件（扩展名是 .cpp）及与之相关的头文件（扩展名是 .h）组成。在组成项目的所有单元中，必须有一个（也只能有一个）单元包含主函数 main（）的定义，这个单元称为主单元，相应的程序文件称为主程序文件。一个简单的控制台应用系统可以只有一个单元，即主单元。通过编译，每个单元生成一个目标程序文件（扩展名是 .obj）。通过连接这些目标程序文件，整个系统生成一个唯一的可执行文件（扩展名是 .exe），而主名与项目名称相同。

在建立应用项目的同时系统会自动建立一个工作区，工作区的名称与所建项目同名。工作区在建立时，自动生成扩展名为 .dsw 的工作区文件，以及扩展名为 .dsp，.ncb，.opt 等其他文件，用来保存工作区信息和项目信息，这些文件都由编译系统自动维护。在工作区建立的同时系统会自动生成一个与项目名同名的文件目录，所有与该工作区相关的文件都将保存在该目录中，其中包括与工作区名称相同的、扩展名为 .dsw 的工作区文件。

现在介绍一个简单应用系统项目的构成。假定程序清单保存在文件 add.cpp 中。作为一个简单的控制台应用系统，它只需一个编译单元，即主单元，项目中的文件包括：

add.cpp：主单元的程序文件；

add.obj：主单元的目标文件；

add.exe：项目的可执行文件；

这里没有列出主单元所用到的头文件 iostream.h、stdio.h 等，因为它是系统提供的头文件，是不能修改的，因此一般不把它们看成是项目的组成部分。

Visual C++ 能够识别项目中的各种文件之间的依赖关系，自动维持这些文件的一致性。例如，若某个头文件被修改了，则所有用#include 命令插入该头文件的程序文件都将重新编译，更新原来的 OBJ 文件，并且重新进行链接，生成新的 EXE 文件。

下面以求两数之和程序为例，说明建立一个控制台应用项目的过程，项目名称为 add。