



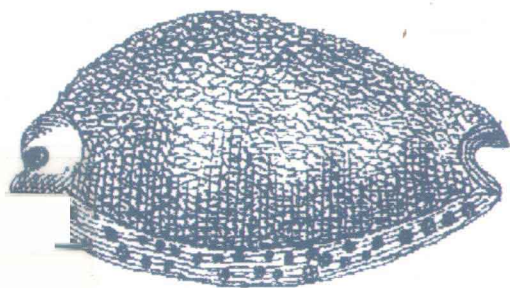
工业和信息化普通高等教育“十二五”规划教材立项项目
21世纪高等学校计算机规划教材
21st Century University Planned Textbooks of Computer Science

C语言 程序设计教程

C Programming Language Tutorials

杨有安 陈维 曹惠雅 鲁丽 编

- 理论与实践相结合的教学思维
- 奠定计算机实践应用的基础
- 提供学生自主学习资源平台



高校系列

 人民邮电出版社
POSTS & TELECOM PRESS



普通高等教育“十二五”规划教材立项项目

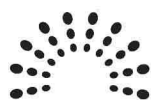
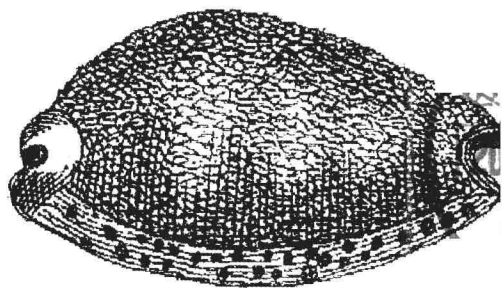
计算机规划教材

books of Computer Science

C语言 程序设计教程

C Programming Language Tutorials

杨有安 陈维 曹惠雅 鲁丽 编



高校系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

C语言程序设计教程 / 杨有安等编. -- 北京: 人民邮电出版社, 2012. 2
21世纪高等学校计算机规划教材
ISBN 978-7-115-27024-5

I. ①C… II. ①杨… 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258024号

内 容 提 要

本书根据全国高等学校计算机基础教育研究会发布的“中国高等院校计算机基础教育课程体系2008”的计算机基础教育的纲领性文件中有关“程序设计”课程的教学要求及人才培养的新要求编写而成。全书共11章, 主要内容包括C语言的基本概念、变量、运算符、表达式、顺序结构、分支结构、循环结构、数组、函数、指针、结构体、联合体和枚举类型、预处理和标准函数、文件、数据结构和数据抽象。同时, 还将介绍程序设计的基本方法和常用算法。

本书内容全面, 由浅入深, 详略得当, 注重实践, 实例丰富, 面向应用。各章附有适量的习题, 便于自学。另外, 针对书中各章内容和上机实验, 本书还配有辅导教材《C语言程序设计实践教程》, 引导读者学习和掌握各章节的知识。全书贯彻传授知识、培养能力、提高素质的教学理念。

本书为高等学校非计算机专业“C语言程序设计”课程的教材, 也可以作为初次学习C语言程序设计的读者、准备计算机二级考试者和计算机工程技术人员的参考书。

工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

C语言程序设计教程

-
- ◆ 编 杨有安 陈 维 曹惠雅 鲁 丽
责任编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.75 2012年2月第1版
字数: 467千字 2012年2月河北第1次印刷

ISBN 978-7-115-27024-5

定价: 35.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

程序设计课程是高等院校计算机基础教育中的重要课程之一。该门课程可以让学生了解程序设计的思想和方法,掌握高级语言程序设计的知识,提高问题求解和程序语言的应用能力。因此,本书旨在实现“以人为本,传授知识,培养能力,提高素质,协调发展”的教育理念,使学生的计算机知识、技能、能力和素质得以协调发展。

C语言是程序设计语言的一种,它不但具有丰富的数据类型与运算符、灵活的控制结构、简洁而高效的表达式、清晰的程序结构、良好的可移植性等优点,而且还具有直接对计算机硬件编程的强大功能;既具有高级语言的优点,又具有低级语言的许多特点;具有完善的结构化、模块化程序结构,是目前世界上流行并广泛使用的高级程序设计语言。C语言既适合于开发系统软件,又适合于开发应用软件,深受程序员的欢迎。

本书在编写过程中以夯实学生程序设计的理论基础,培养学生程序设计的能力和养成良好的程序设计风格为宗旨,并结合编者多年从事程序设计教学和研究的经验,参考了大量同类教材,吸收其优点。本书的特点是:内容的编排符合高等院校学生的特点和认知规律,由浅入深,循序渐进,通俗易懂;通过例题介绍C程序设计的基本方法与基本技巧;通过习题训练学生程序设计的技能,是一本适合初次学习程序设计的读者学习C语言的书籍。

全书分为11章,其中第1章、第2章和第4章由曹惠雅编写,第3章、第8章和第10章由鲁丽编写,第7章、第9章和第11章由陈维编写,第5章和第6章由杨有安编写。杨有安负责全书的统稿工作。本书在编写的过程中得到华中科技大学文华学院各级领导的大力支持,在此表示衷心的感谢。

本书是大学计算机和相关专业的“C程序设计”课程的教材,可作为高等学校、各类职业技术学院、各类培训学校的计算机应用教科书及计算机等级考试的参考书,也可以作为企事业单位员工、国家公务员计算机技能培训用书,还可以作为渴望掌握计算机基础知识和基本操作的各类初学者的自学用书。

由于编者水平有限,加之时间仓促,书中难免有不足之处,敬请读者批评指正。

编者

2011年10月

第 1 章 C 语言概述 1	2.2.8 运算符的优先级与结合性 38
1.1 C 语言的发展和特点..... 1	2.3 数据类型的转换..... 39
1.1.1 C 语言的发展 1	2.3.1 自动转换..... 39
1.1.2 C 语言的主要特点 1	2.3.2 赋值转换..... 40
1.2 C 语言的词汇..... 2	2.3.3 强制类型转换 40
1.2.1 C 语言字符集..... 3	小结..... 41
1.2.2 关键字 3	习题..... 41
1.2.3 标识符 3	第 3 章 顺序和选择结构
1.2.4 保留标识符..... 4	程序设计 45
1.2.5 注释..... 4	3.1 程序设计概述..... 45
1.3 C 程序的结构..... 4	3.1.1 程序设计基本步骤..... 45
1.3.1 简单 C 程序举例..... 4	3.1.2 C 语言编写风格 46
1.3.2 C 语言程序的结构特点 6	3.1.3 语句..... 46
1.3.3 书写程序时应遵循的规则 7	3.2 scanf() 函数和字符输入/输出函数调用..... 47
1.4 Visual C++ 6.0 上机简介..... 7	3.2.1 数据输入的概念 47
1.4.1 Visual C++ 6.0 集成开发环境简介 7	3.2.2 scanf() 函数的调用..... 47
1.4.2 Visual C++ 6.0 集成环境上机步骤..... 9	3.2.3 getchar() 函数..... 49
小结..... 13	3.2.4 putchar() 函数..... 50
习题..... 13	3.3 程序的 3 种基本结构..... 51
第 2 章 基本数据类型和运算符 15	3.4 if 选择结构语句..... 52
2.1 基本数据类型..... 15	3.4.1 if 语句的 3 种形式..... 52
2.1.1 常量和变量的概念..... 16	3.4.2 if 语句的嵌套..... 56
2.1.2 常量类型 17	3.5 switch 选择结构语句..... 58
2.1.3 变量类型 20	3.6 程序设计举例..... 61
2.2 运算符和表达式..... 25	小结..... 65
2.2.1 算术运算符与算术表达式 27	习题..... 65
2.2.2 赋值运算符与赋值表达式 28	第 4 章 循环结构程序设计 70
2.2.3 增量运算符与增量表达式 32	4.1 for 循环..... 70
2.2.4 逻辑运算符与逻辑表达式 33	4.2 while 循环 72
2.2.5 关系运算符与关系表达式 35	4.3 do-while 循环..... 74
2.2.6 条件运算符与条件表达式 36	4.4 3 种循环语句的比较..... 76
2.2.7 逗号运算符与逗号表达式 38	4.5 跳转语句 76

4.5.1 break 语句	76	6.2.4 函数参数及函数间的数据传递	131
4.5.2 continue 语句	77	6.3 函数的嵌套调用和递归调用	136
4.5.3 goto 语句	78	6.3.1 函数的嵌套调用	136
4.6 循环语句的嵌套	79	6.3.2 函数的递归调用	139
4.7 程序设计举例	82	6.4 作用域和存储类型	141
小结	88	6.5 内部函数和外部函数	148
习题	89	6.5.1 内部函数	148
第 5 章 数组	92	6.5.2 外部函数	148
5.1 一维数组	92	6.6 模块化程序设计	150
5.1.1 一维数组的定义	92	6.6.1 模块化程序设计方法的指导 思想	150
5.1.2 一维数组元素的引用	93	6.6.2 模块分解的原则	150
5.1.3 一维数组元素的初始化	95	6.7 程序设计举例	151
5.2 二维数组	96	小结	156
5.2.1 二维数组的定义	96	习题	156
5.2.2 二维数组元素的引用	97	第 7 章 指针	159
5.2.3 二维数组元素的初始化	98	7.1 指针的概念	159
5.3 字符型数组	100	7.1.1 地址与指针	159
5.3.1 字符数组的定义	101	7.1.2 指针变量的定义和引用	160
5.3.2 字符数组的引用	101	7.1.3 指针变量的运算	162
5.3.3 字符数组的初始化	102	7.2 指针变量作为函数参数	164
5.3.4 字符串及其结束标志	103	7.3 指针与一维数组	167
5.3.5 字符数组的输入/输出	104	7.3.1 一维数组的指针表示	167
5.3.6 常用的字符串处理函数	106	7.3.2 数组作函数参数时的指针表示	169
5.4 使用数组的程序设计方法	109	7.3.3 字符串的指针表示	171
5.4.1 排序	110	7.4 指针与多维数组	174
5.4.2 查找	112	7.4.1 多维数组的处理	174
5.5 程序设计举例	114	7.4.2 指向多维数组的指针	175
小结	117	7.5 指针数组和多级指针	177
习题	118	7.5.1 指针数组的概念	177
第 6 章 函数和模块设计	122	7.5.2 指针数组的应用	178
6.1 结构化程序设计	122	7.5.3 多级指针(指向指针的指针)	181
6.1.1 结构化程序设计的基本概念	122	7.6 指针与函数	182
6.1.2 结构化程序设计的基本特征	123	7.6.1 指向函数的指针	183
6.2 函数的定义和调用	124	7.6.2 函数指针的应用	184
6.2.1 函数的定义	124	7.6.3 返回指针的函数	187
6.2.2 函数的调用	125	7.7 命令行参数	190
6.2.3 函数的返回值	128	小结	191

习题	193	10.2 文件的基本操作	235
第 8 章 结构体与联合体	195	10.2.1 文件的打开与关闭	235
8.1 结构体	195	10.2.2 文件的读写	237
8.1.1 结构体类型的定义	195	10.2.3 文件检测函数	246
8.1.2 结构体类型变量的定义与使用	196	10.3 程序设计举例	246
8.1.3 结构体类型变量的赋值与初始化	197	小结	249
8.1.4 结构体类型数组的定义与引用	198	习题	249
8.1.5 结构体类型指针的定义和引用	201	第 11 章 数据结构和数据抽象	251
8.1.6 结构体类型数据的动态存储分配	204	11.1 数据抽象	251
8.1.7 链表及其基本操作	206	11.1.1 数据结构和数据类型	251
8.2 联合体	211	11.1.2 抽象数据类型	252
8.3 其他自定义数据类型	213	11.2 线性表	252
8.3.1 枚举类型	213	11.2.1 线性表的定义	252
8.3.2 类型定义符 typedef	215	11.2.2 线性表的基本操作	253
小结	217	11.2.3 线性表的顺序存储	254
习题	217	11.2.4 顺序表上基本运算的实现	255
第 9 章 预处理和输入/输出		11.3 堆栈	256
标准函数	219	11.3.1 抽象栈的定义及基本操作	257
9.1 预处理命令	219	11.3.2 抽象栈的定义	257
9.1.1 宏定义	219	11.3.3 顺序栈的基本运算的实现	257
9.1.2 文件包含	222	11.4 队列	259
9.1.3 条件编译	222	11.4.1 队列的定义	259
9.2 输入/输出标准函数	224	11.4.2 队列的存储结构及其相关算法	259
9.2.1 格式输出函数 (printf)	225	小结	262
9.2.2 格式输入函数 (scanf)	227	习题	263
9.3 自定义头文件设计的原则	229	附录 1 ASCII 代码对照表	264
小结	232	附录 2 C 库函数	265
习题	232	附录 3 Debugger 调试器	
第 10 章 文件	234	使用简介	271
10.1 文件的基本概念	234		

第 1 章

C 语言概述

C 语言是 Combined Language (组合语言) 的简称, 是一种被广泛重视且应用普遍的计算机程序设计语言, 属于 C++ 语言的一部分。它适于作系统描述语言, 既可以用来编写系统软件, 也可以用来编写应用软件。本章主要介绍 C 语言的发展、特点、词汇以及 C 语言程序的结构与 C 语言程序的运行环境。

1.1 C 语言的发展和特点

C 语言作为国际上公认的最重要的几种优秀程序设计语言之一, 有必要追述一下其发展史。

1.1.1 C 语言的发展

C 语言是在 B 语言的基础上, 由美国贝尔实验室的 Dennis.M.Ritchie 于 1972 年设计实现的。当时的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言, 但 C 语言并没有被束缚在任何特定的硬件或操作系统上, 它具有良好的可移植性。1977 年出现了不依赖于具体机器的 C 语言编译文本, 使向各种机器移植 C 语言变得更加简单, 这也推动了 UNIX 系统的广泛实现。随着 UNIX 的日益普及, 又反过来带动了 C 语言的迅速推广, 使它先后被移植到各种大、中、小及微型计算机上, 同时, C 语言的突出优点才引起人们的普遍注意。

1978 年, 贝尔实验室的 Brian.W.Kernighan 和 Dennis.M.Ritchie (合称 K&R) 合著了著名的《The C Programming Language》一书, 并在附录中提供了 C 语言参考手册, 这本书成为广泛使用的 C 语言的基础, 被人们称作非官方的 C 语言标准。1983 年, 美国国家标准研究所 (American National Standards Institute, ANSI) 为 C 语言制定了第一个 ANSI 标准, 称为 ANSI C 标准。1990 年, C 语言成为国际标准化组织 (ISO) 通过的标准语言。

C 语言是作为描述系统的语言而设计的, 但随着其日益广泛的应用, 特别是 20 世纪 80 年代以后各种微机 C 语言的普及, 它已经成为众多程序员最喜爱的语言, 它的使用覆盖了几乎计算机的所有领域, 包括操作系统、编译系统、数据库管理程序、CAD、过程控制、图形图像处理等。

C 语言是 C++ 的基础, C++ 语言和 C 语言在很多方面是兼容的。因此, 掌握了 C 语言, 可为将来学习 C++ 打下坚实的基础。本书使用 Visual C++ 6.0 作为 C 语言程序的运行环境。

1.1.2 C 语言的主要特点

C 语言之所以能存在和发展, 并具有强大的生命力主要因为它具有强大的功能。

① C 语言简洁、紧凑、使用方便、灵活。

C 语言一共只有 32 个关键字，9 种控制语句，程序书写形式自由，语法控制不严格，表达式简练、灵活、实用。

② 运算符丰富。

C 语言中共有 34 个运算符，它们与丰富的数据类型相结合，构成了各种各样的表达式，实现了在其他高级语言中难以实现的各种复杂运算。

③ 数据结构丰富。

C 语言的数据类型有整型、实型、字符型、数组、指针、结构体、联合体等，能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。尤其是指针类型数据的引入，使程序运行效率更高。另外，C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

④ 具有结构化的控制语句。

C 语言是良好的结构化语言，符合现代编程风格的要求。结构化语言的显著特点是程序代码模块化，即程序的各个部分除了必要的信息交流外彼此独立。C 语言是由函数集合构成的，函数之间彼此独立，具有编写结构式程序所必需的基本流程控制语句。C 语言的源文件可以分割成多个源程序，分别进行编译，然后通过连接形成可执行的目标文件，为开发大型软件提供了方便。同时，C 语言还提供了多种存储属性，使数据可以在需要的作用域中起作用，从而提高了程序的可靠性。C 语言的这种结构化方式使程序层次更清晰，使用、维护及调试更方便。

⑤ C 语言语法限制不太严格、程序设计自由度大。

一般的高级语言语法检查比较严，能够检查出几乎所有的语法错误。而 C 语言则放宽了语法检查，允许程序编写者有较大的自由度。例如，整型数据与字符数据可以通用，对数组下标越界不作检查等。因此，在程序设计中，程序员不要过分依赖编译器的语法检查。

⑥ C 语言允许直接访问物理地址。

C 语言既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、字节和地址进行操作，还可以用来编写系统软件。C 语言的这种特性，使它既是成功的系统描述语言，又是通用的程序设计语言。

⑦ 生成目标代码质量高。

C 语言程序执行效率高，一般只比汇编程序生成的目标代码效率低 10%~20%。

⑧ 适用范围大、可移植性好。

C 语言编写的程序中没有依赖于硬件的输入/输出语句，程序的输入/输出功能是通过调用输入/输出函数实现的，而这些函数是由系统提供的独立于 C 语言的程序模块，所以编写好的 C 语言源程序基本上不作修改就可以用于各种型号的计算机和各种操作系统，从而便于硬件结构不同的计算机之间实现程序的移植。

1.2 C 语言的词汇

为了按照一定的语法规则构成 C 语言的各种成分（如常数、变量等），C 语言规定了基本词法单位——单词，而构成单词的最重要的形式是关键字、标识符和保留标识符。下面分别作简单介绍。

1.2.1 C 语言字符集

组成 C 语言源程序代码的基本字符称为 C 语言字符集，它是构成 C 语言的基本元素。C 语言允许使用的基本字符如下：

- ① 大小写英文字符：A~Z, a~z；
- ② 数字字符：0~9；
- ③ 特殊字符：+ = - _ (下画线) () * & ^ % # ! , . ; : ? ' " ~ \ | / < > { } [] ;
- ④ 不可打印的字符：空格、换行符、制表符、响铃符。

一般的 C 语言源程序仅仅包含以上字符集中的字符，但在具体的 C 语言编译系统中可对上述字符集加以扩充。

1.2.2 关键字

关键字又称为保留字，是具有特定含义的、专门用来说明 C 语言特定成分的一类单词。例如，关键字 `int` 用来定义整型变量，而关键字 `float` 则用来定义实型变量。C 语言的所有关键字通常都是用小写字母表示。例如，关键字 `int` 不能写成 `Int`。由于每个关键字都有特定的含义，所以不能作为用户程序中的变量名、函数名等，否则会产生编译错误。在 C89 标准中共有如下 32 个关键字：

<code>auto</code>	<code>break</code>	<code>case</code>	<code>char</code>	<code>const</code>	<code>continue</code>	<code>default</code>
<code>do</code>	<code>double</code>	<code>else</code>	<code>enum</code>	<code>extern</code>	<code>float</code>	<code>for</code>
<code>goto</code>	<code>if</code>	<code>int</code>	<code>long</code>	<code>register</code>	<code>return</code>	<code>short</code>
<code>signed</code>	<code>sizeof</code>	<code>static</code>	<code>struct</code>	<code>switch</code>	<code>typedef</code>	<code>union</code>
<code>unsigned</code>	<code>void</code>	<code>volatile</code>	<code>while</code>			

在新的 C99 标准中，又增加了如下 5 个关键字：

`_Bool` `_Complex` `_imaginary` `inline` `restrict`

1.2.3 标识符

计算机程序用来描述数据处理的过程，处理的对象是数据。在程序中，通过名字建立对象定义与使用的关系。为了满足这种需要，每种程序语言都规定了在程序中名字描述的规则。在 C 语言中用于标识名字的有效字符序列称为标识符，对标识符作了如下规定：

- ① 标识符的第一个字符必须是英文字母或下画线 (`_`)；
- ② 如果第一个字符后面还有字符序列，则它应是英文字母、下画线符或数字组成的序列。

标识符中的英文字母大小写是有区别的，如标识符 `abc` 与标识符 `ABC` 不相同。为了便于读者对标识符有进一步的认识，下面列举若干个正确标识符和不正确标识符：

正确的标识符：

`Abc` `abc` `_Abc` `_4a5`

不正确的标识符：

`a?` (含有不合法字符 “?”)
`2abc` (第一个字符不允许为数字)
`a b` (标识符中不允许有空格)
`yes/no` (含有不合法字符 “/”)
`πr` (“π” 为不合法字符)

能够被编译程序识别，并区别于其他标识符的那一部分所包含的字符数目称为标识符的有效长度。标识符中有效字符的个数视系统不同而不同。例如，Turbo C 规定前 32 个字符有效，超过的部分忽略。比如，对于 8 个字符有效的标识符而言，`identifi` 与 `identifier` 被视为同一标识符，因后者中的 `er` 已被忽略。

以后将会看到，标识符用来为变量、符号常量、数组、函数等取名。使用时，标识符的选择由程序员自定，但不能与关键字相同。另外，为了增加程序的可读性，选择标识符时应遵循“见名知义”的原则，即选择描述性的标识符，标识符应尽量与所要命名的对象之间有一定的联系，以助于识别和记忆。例如：

```
length    (表示长度)
time      (表示时间)
pi        (表示圆周率 $\pi$ )
```

1.2.4 保留标识符

保留标识符是系统保留的一部分标识符，通常用于系统定义和标准库函数的名字。例如，C 标准库定义了很多以下画线开头的标识符，虽然它们也是合法的标识符，但用作一般标识符时可能会出现运行错误，因此要尽量避免使用以下画线开头的标识符来定义自己的变量。

1.2.5 注释

在 C 语言程序中，注释部分的格式是：

```
/*注释内容*/ 或 //注释内容
```

例如：`/*学生成绩管理程序*/、//My c program。`

注释不是程序代码，只是对程序解释说明的一种标注，用来向用户提示或解释程序的意义。程序编译时会把注释部分作为空白符跳过而不予处理。

使用注释需要注意 3 点：

① 注释不允许嵌套使用，例如：

```
/*test 1/*test 2*/test 3*/
```

顶格编译器解释时，只把 `/*test 1/*test 2*/` 看成注释，后面的 `test 3*/` 无法解释错误；

② 用 `/*.....*/` 形式表示注释时可以跨行，也可以穿插在一行代码中；

③ 用 `//.....` 形式注释内容时，表示从 `//` 开始直到该行末尾的所有字符都属于注释，这种注释不能跨行，也不能穿插在一行代码中间，这是从 C++ 借鉴的语法。

1.3 C 程序的结构

用 C 语言编写的程序称为 C 语言源程序，简称为 C 程序。为了说明 C 语言源程序的结构特点，先看以下几个程序。这几个程序由简单到复杂，虽然有关内容还未介绍，但可以从中了解到 C 程序的基本组成结构及其书写风格。

1.3.1 简单 C 程序举例

【例 1-1】编写一个 C 语言程序，输出“good morning!”。

程序如下:

```
/*c1_1.c*/
#include <stdio.h>          /*为文件包含, 扩展名为.h的文件称为头文件*/
void main()
{
    printf("good morning!\n"); /*通过显示器输出 good morning!*/
}
```

说明:

① C 语言程序中可以使用注释, 但注释内容不参与编译。

② #include 称为文件包含命令, #include <stdio.h>是文件包含, 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。被包含的文件通常由系统提供, 其扩展名为.h, 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。详细内容将在后续章节介绍。

③ main 是主函数的函数名, 表示这是一个主函数。每个完整的 C 语言源程序都必须有主函数, 且只能有一个主函数 (main 函数), 程序总是从 main 函数开始执行, 并终止于 main 函数。函数体由一对大括弧“{ }”括起来, 其间一般包括程序的说明部分和执行部分。

④ 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。其功能是将输出的内容送到显示器显示。

该程序正确执行后, 会在显示器上显示输出:

```
good morning!
```

【例 1-2】从键盘输入一个学生两门课 (语文 yw 和数学 sx) 的成绩, 并求其平均成绩。

程序如下:

```
/*c1_2.c*/
#include<stdio.h>
void main()
{
    int yw,sx,sum;          /*定义 3 个整型变量*/
    printf("Input two numbers:"); /*显示提示信息*/
    scanf("%d%d",&yw,&sx);    /*输入 yw, sx 值*/
    sum=yw+sx;             /*求出 yw 与 sx 之和, 并把它赋予变量 sum*/
    printf("average=%d\n",sum/2); /*输出语文和数学的平均成绩*/
}
```

程序分析:

① 该程序中使用了 yw、sx 和 sum 3 个变量, 所有变量在使用之前必须先定义。

② scanf 函数是一个由系统定义的标准函数, 可在程序中直接调用。它的功能是输入变量 x 和 y 的值。&yw 和 &sx 中“&”的含义是“取变量地址”, 表示将从键盘输入的两个值分别存放到地址标志为 yw 和 sx 的存储单元中。

③ “%d”是输入/输出数据的“格式说明”, 用来指定输入/输出时的数据类型和格式, %d 表示“十进制整数类型”, 在执行输出时, 屏幕上显示一个十进制整数。

④ sum=yw+sx 为赋值表达式, 表示将 x+y 之和赋值给 sum 变量所标识的存储单元。

该程序正确执行后，会在显示器上显示输出：

```
Input two numbers:89 95
average=92
```

【例 1-3】从键盘输入两个整数，进行比较后将较大数输出。

程序如下：

```
/*c1_3.c*/
#include<stdio.h>
void main()
{
    int x,y,z;                /*定义 3 个整型变量*/
    int max(int a,int b);     /*函数类型说明*/
    printf("Input two numbers:"); /*显示提示信息*/
    scanf("%d%d",&x,&y);     /*输入 x, y 值*/
    z=max(x,y);              /*调用 max 函数*/
    printf("max=%d\n",z);    /*将较大数输出*/
}

int max(int a,int b)        /*定义 max 函数*/
{
    int c;                  /*定义一个整型变量*/
    c=a>b?a:b;              /*求出变量 c 的值*/
    return c;               /*将 c 的值返回到主调函数*/
}
```

程序分析：

① 本程序包括两个函数：主函数 main 和自定义函数 max。max 函数的作用是将 a 和 b 中较大者的值赋予变量 c；return 语句将 c 的值返回主调函数。

② 在调用 max 函数时，将实际参数 x 和 y 的值分别对应传给 max 函数中的形式参数 a 和 b。

③ $a>b?a:b$ 是一个条件表达式，当 $a>b$ 成立时， $a>b?a:b$ 表达式的值为 a 的值；反之则为 b 的值。详细内容将在第 2 章介绍。

该程序正确执行后，会在显示器上显示输出：

```
Input two numbers:8 2
max=8
```

本例中涉及函数调用、实际参数、形式参数等概念，详细内容将在第 6 章中介绍。

1.3.2 C 语言程序的结构特点

通过上面 3 个 C 语言源程序，可以看出其基本结构具有以下几个特点。

① C 语言源程序的基本结构单位是函数。所有的 C 语言程序都由一个或多个函数构成，其中 main 函数必须有且只能有一个。

② main() 函数可以出现在 C 源程序的任何位置，程序执行时总是从 main() 函数开始，又在 main() 函数结束。主函数可以调用标准库函数（如 printf()、scanf() 等）和用户自定义函数，但标准库函数和用户自定义函数却不能调用主函数。

③ 源程序中的预处理命令通常放在源文件或源程序的最前面。

④ 分号“;”是 C 语句的必要组成部分。每个语句或每个变量说明都必须以分号结尾。但预处理命令、函数头和花括号“{”、“}”后面不能加分号。

⑤ 标识符、关键字之间必须至少加一个空格以示分隔。

⑥ 可以在程序的任何位置用“/*注释内容*/”或“//注释内容”的形式对程序或语句进行注释，以增加程序的可读性。

1.3.3 书写程序时应遵循的规则

C语言程序的书写格式非常自由，但从书写清晰，便于阅读、理解、维护的角度出发，建议在书写C语言程序时遵循以下几个规则。

① 一个说明或一条语句占一行。

② 用{}括起来的部分，通常表示了程序的某一层结构（如函数体、循环体、复合语句等）。{}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比高一层次的语句或说明向后缩进若干格后书写，同一层次的语句或说明左对齐，以增强程序编写的层次感，增加程序的可读性。

④ 函数块与函数块之间加一空行分隔，以便清楚地分出程序中有几个函数。

编程时应力求遵循上述规则，以养成良好的编程习惯。

1.4 Visual C++ 6.0 上机简介

Visual C++（简称VC++）是美国Microsoft公司开发的Microsoft Visual Studio的一部分，是一个基于Windows操作系统的可视化、面向对象且使用广泛的C/C++集成开发环境（Integrated Development Environment, IDE）。它成功地将面向对象和事件驱动编程概念联系起来，并得到了很好的配合，使得编写Windows应用程序的过程变得简单、方便且代码量小。VC++ 6.0集程序的代码编辑、编译、连接、调试于一体，给编程人员提供了一个完整、方便的开发界面和许多有效的辅助开发工具。

VC++ 6.0的编辑环境包含了许多独立的组件，它们是：文本编辑器、资源编辑器、C/C++编译器、连接器、调试器、AppWizard、ClassWizard、源程序浏览器以及联机帮助。所有这些构件的功能都隐藏在VC++ 6.0的菜单和工具条中。通过该集成环境，程序员可以观察和控制整个开发过程。

1.4.1 Visual C++ 6.0 集成开发环境简介

在已安装Visual C++的计算机上，可以直接从桌面双击Microsoft Visual C++图标，进入Visual C++集成开发环境，或者单击【开始】|【程序】菜单，选择Microsoft Visual Studio 6.0中的Microsoft Visual C++ 6.0菜单项，进入Visual C++ 6.0集成开发环境。

Visual C++集成开发环境主要由标题栏、菜单栏、工具栏、项目工作区、编辑区、输出区等组成，如图1-1所示。

1. 项目工作区

Visual C++集成开发环境以项目工作区来组织应用程序的工程，项目工作区文件扩展名为.dsw，这种类型的文件在Visual C++中级别是最高的。项目工作区含有工作区的定义和工程中所包含文件的所有信息。所以，要打开一个工程，只需打开对应的项目工作区文件（*.dsw）即可。

项目工作区窗格位于屏幕左侧，包含 ClassView（类视图）、ResourceView（资源视图）和 FileView（文件视图）3 种视图。

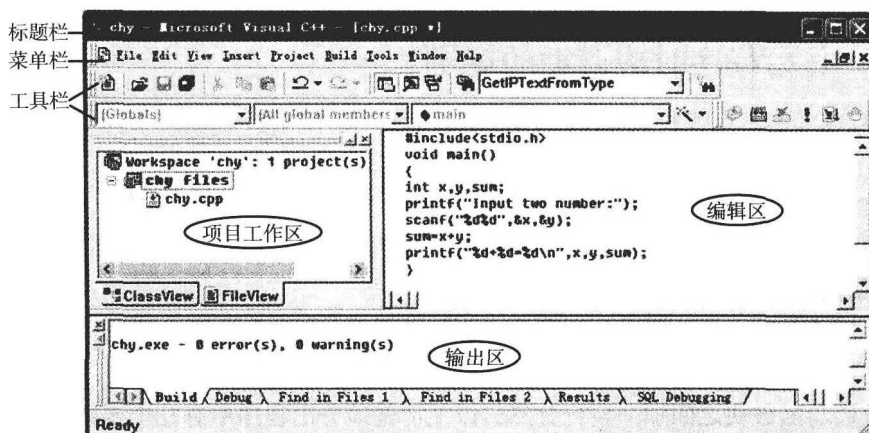


图 1-1 Microsoft Visual C++集成开发环境

(1) ClassView

ClassView 用于显示工程中定义的类。展开文件夹将显示工程中所有的类，包括系统生成的和用户自定义的。单击类名左边的“+”图标，就可以看到类的数据成员和成员函数。

在 ClassView 视图中，双击某个类，可以打开声明该类的头文件 (*.h)，且光标会停留在该类的声明处。双击某个成员变量，光标会停留在所属类的头文件 (*.h) 中该变量的声明处。双击某个成员函数，光标会停留在所属类的实现文件 (*.cpp) 中该成员函数的实现处。在一个类的头文件中，可以依据 Visual C++ 的语法直接修改类的成员函数、数据成员、全局变量、函数、类定义等，并反映到 ClassView 视图中。此外，右击某个类名或成员，选择快捷菜单项，可以进行该类数据成员或成员函数的浏览、添加、删除等操作。

(2) ResourceView

ResourceView 用于显示工程中所包含的资源文件。展开文件夹可显示所有的资源类型。显示的资源类型包括 Accelerator（加速键）、Dialog（对话框）、Icon（图标）、Menu（菜单）、StringTable（串表）、Toolbar（工具条）、Version（版本）等。双击底层某个图标或资源文件名，可以打开相应的资源编辑器。

(3) FileView

FileView 用于显示所创建的工程。展开文件夹后可以看到工程中所包含的文件，除了查看，还可以管理文件，包括增加、删除、移动、重命名、复制文件等。单击文件类型左边的“+”图标，可看到工程中该种类型的所有文件，双击一个文件即可打开该文件。一个应用程序工程主要包含实现源文件 (*.cpp)、头文件 (*.h)、资源文件 (*.rc) 等文件类型。

2. 编辑区

编辑区为开发者提供了编辑文件和资源的手段。通过编辑窗口，开发者可以编辑和修改源程序和各种类型的资源。

资源是以文本的形式存放在资源定义文件中，并由编译器编译为二进制代码。资源包括菜单、对话框、图标、字体、快捷键等。VC++ 提供了一个资源编辑器，开发者可以在图形方式下对各种资源进行编辑，进而定义 Windows 程序的界面部分。

3. 输出区

输出区用于输出一些用户操作后的反馈信息，它由一些页面组成，每个页面输出一种信息，输出的信息种类主要有：

编译信息：在编译时输出，主要是编译时的错误和警告；

调试信息：在对程序进行调试时输出，主要是程序当前的运行状况；

查找结果：在用户从多个文件中查找某个字符串时产生，显示查找结果的位置。

1.4.2 Visual C++ 6.0 集成环境上机步骤

一个应用项目（Project）由若干个编译单元组成，而每个编译单元由一个程序文件（扩展名为.cpp）及与之相关的头文件（扩展名为.h）组成。在组成项目的所有单元中，必须有一个且只能有一个单元包含函数 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。计算机硬件不能直接执行 C 语言源程序，必须将其翻译成二进制目标程序。翻译工作是由一个称为编译程序的系统软件完成的，翻译的过程称为编译。通过编译，每个单元生成一个浮动程序文件（也称为目标程序文件，扩展名为.obj）。通过连接这些浮动程序文件，整个系统生成一个唯一的可执行文件，其扩展名为.exe。

建立一个控制台应用项目的过程分为 3 步：建立工作空间及项目，建立主程序文件，C 源程序的编辑、编译、连接和运行。

1. 建立工作空间及项目

工作空间（Workspace）是一个包含用户的所有相关项目和配置的实体。由若干个关系密切的项目构成一个工作空间，工作空间在建立时会自动生成扩展名为.dsw 的工作空间文件及其他文件。

项目（Project）为一个配置和一组文件，用以生成最终的程序或二进制文件。一个工作空间可以包含多个项目，这些项目既可以是同一类型的项目，也可以是不同类型的项目（如 Visual C++ 和 Visual J++ 项目）。

建立工作空间及项目的操作步骤如下。

① 启动 Visual C++ 后，选择“File”菜单下的“New”命令，屏幕上即出现新建对话框，其中包括 Files、Projects、Workspaces 和 Other Documents 4 个标签。一般当前标签是“Projects”，如果不是，则单击“Projects”标签，使之成为当前标签，如图 1-2 所示。

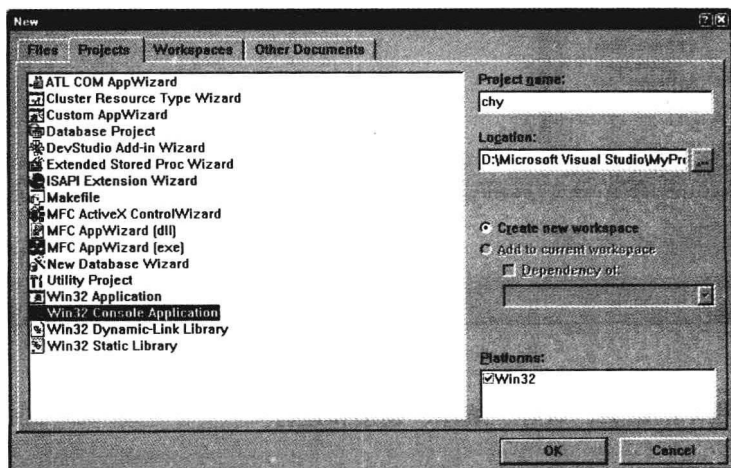


图 1-2 新建工程对话框

② 选择“Win32 Console Application (32 位控制台应用程序)”, 在“Project name”文本框中输入要建立的项目名称“chy”, 在“Location”文本框输入工程所在的路径, 然后单击“OK”按钮, 即弹出如图 1-3 所示的应用程序生成向导界面。

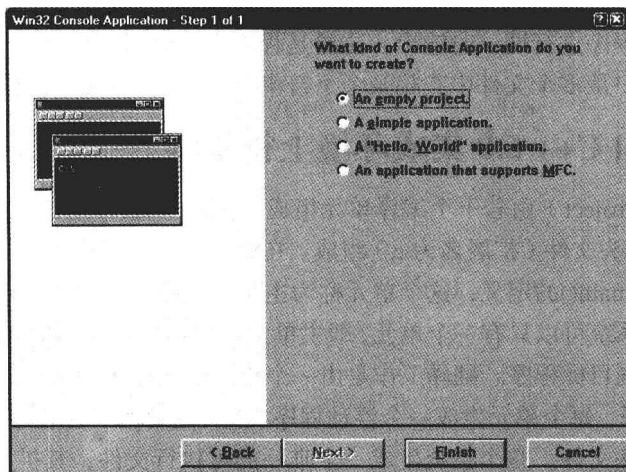


图 1-3 应用程序生成向导界面

③ 应用程序生成向导界面用于确定要生成的项目类型。在此, 选择“An empty project (空项目)”, 然后单击“Finish”按钮, 检查无误后再单击“OK”按钮即可实现项目和工作空间的建立。

2. 建立主程序文件

选择“File”菜单下的“New”命令, 在弹出的新建对话框中选择“Files”标签, 从窗口选择“C++ Source File (C++源程序)”, 在窗口右侧“File”后的文本框中输入文件名为“chy”, 在“Location”文本框中输入该文件存放的路径, 然后单击“OK”按钮, 弹出如图 1-4 所示的窗口。在 VC++ 6.0 窗口右侧出现的一个空文件, 即为源程序的编辑区域。

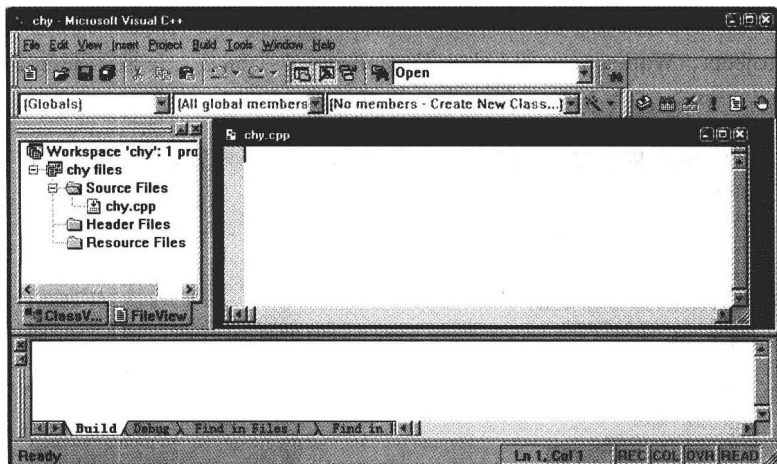


图 1-4 主程序文件建立后的界面

3. C 语言源程序的编辑、保存、编译、连接和运行

(1) 编辑程序

所谓编辑, 就是对文件内容进行输入、修改、删除等操作。在 VC++ 6.0 窗口的文件编辑区