21世纪高等学校计算机规划教材

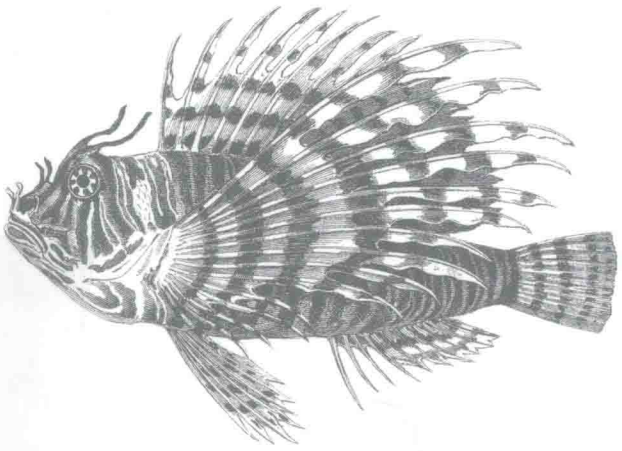
21st Century University Planned Textbooks of Computer Science

案例式C语言程序设计

Case-based C Programming Language

王富强 张春玲 刘明华 主编
孔锐睿 孙劲飞 李朝玲 副主编

- 内容由浅入深、通俗易懂
- 教学深入浅出、循序渐进
- 案例详细丰富、实践性强



高校系列

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

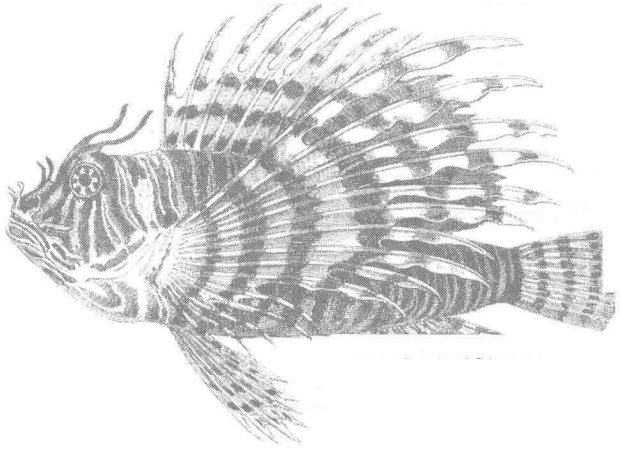
21st Century University Planned Textbooks of Computer Science

案例式C语言程序设计

Case-based C Programming Language

王富强 张春玲 刘明华 主编

孔锐睿 孙劲飞 李朝玲 副主编



高校系列

人民邮电出版社

北京

图书在版编目（CIP）数据

案例式C语言程序设计 / 王富强，张春玲，刘明华主编. -- 北京 : 人民邮电出版社，2016.8
21世纪高等学校计算机规划教材
ISBN 978-7-115-42963-6

Ⅰ. ①案… Ⅱ. ①王… ②张… ③刘… 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(2016)第177692号

内 容 提 要

本书在编撰过程中，遵循注重基础、理论联系实际、案例典型、增强实践开发能力的原则，以社会和企业需求为导向，以互联网+为助力，以C语言的发展为切入点，以基本语法、语句为基础，以结构为主线，以学懂学会学精会用为目的，以案例驱动的编写方式深入浅出地详细阐述了C语言的程序设计思想和流程。本书注重对读者设计开发能力的培养，以训练读者自我思考和解决问题的能力为目标，希望通过本书的学习，读者具备程序开发设计能力和自动化和专业化的数据信息处理能力。

本书共13章，分为4部分。第1部分为基础知识，包括第1章C语言简介，第2章程序设计与算法，第3章数据类型、运算符与表达式；第2部分为程序设计基本结构，包括第4章顺序结构程序设计、第5章选择结构程序设计和第6章循环结构程序设计；第3部分为程序设计方法和具体应用，包括第7章数组、第8章函数、第9章预处理命令、第10章指针、第11章结构体与共同体和第12章文件；第4部分为调试，包括第13章常见错误和程序调试。

本书内容细致、实例丰富、通俗易懂，适合作为普通高等院校理工类本/专科专业的C程序设计语言教材，也可供计算机应用工作者阅读参考。

◆ 主　　编　王富强　张春玲　刘明华
副 主 编　孔锐睿　孙劲飞　李朝玲
责任编辑　吴　婷
责任印制　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京昌平百善印刷厂印刷

◆ 开本：787×1092　1/16
印张：16.5　　2016年8月第1版
字数：430千字　　2016年8月北京第1次印刷

定价：39.80元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

前言

在信息时代及电子信息和“互联网+”作为国家的新兴发展产业的背景下，高等院校的信息教学迎来新的发展机遇。重视教育信息化、注重信息技术的基础体系和程序设计操作能力成为高等院校计算机教育的主题。为此，各高等院校都在制订一系列符合自身定位的教学大纲，引入新的教学理念、提出新的课程教学目标、改革教学内容、丰富教学手段等，力求在21世纪的教学改革中走在前列，培养符合社会与企业需求、理论与技术兼备的新型人才。

“C语言程序设计”是高等院校程序设计语言类的基础课程。通过该课程的学习，学生可逐步认识以程序设计为核心的信息技术在信息化社会的重要作用，全面了解程序设计语言的基本结构、解题算法，从而提高信息素养和编程能力，也为进一步熟练应用计算机，充分利用程序设计语言和成熟算法设计开发大型应用程序，实现数据处理自动化和专业化，打下良好基础。

本书按照教育部高等学校计算机基础课程教学指导委员会提出的“大学计算机教学基本要求”编写而成。在结构设计、内容选择及编写过程中充分考虑了读者需求，并结合全国计算机等级考试——C语言考试大纲，设计了本书的相关章节内容，力求内容全面化，方式新颖化、应用实用化。本书知识体系贯穿了校内教育和校外需求，也适合不同院校、不同专业的教学与社会各层次人士的自学。

本书采用案例驱动模式编写，主要介绍C语言程序设计的基本结构及应用C语言解决学习生活中的某些问题。全书内容包括：C语言简介、程序设计与算法、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体与共用体、文件以及程序常见错误与调试等。本书内容翔实、图文并茂，章节安排各有特色，以浅显易懂、实用的形式详细讲解知识要点，并联系实际，指导读者强化基础知识，应用程序设计基本思想开发设计程序，使读者达到举一反三的目的。

本套书具有以下特色。

（1）编者具有丰富的教学与指导经验。全书编撰思路以企业、社会需求为导向，紧跟当前C语言程序设计的发展和应用水平，贴近“互联网+”的基础技术，注重实际开发设计能力，全面培养学生的程序设计能力和应用能力。

（2）实例丰富、典型性强。编者对本书实例的选取以解决实际问题为导向，选择生活中常碰到的问题作为实例，如迭代法求解、线性代数的矩阵运算、商场的打折促销、学生成绩的数据处理及文件的存储操作等。

（3）分析透彻，可移植性高。编者将程序设计思想贯穿全书。书中涉及的知识点基本配有或简或难的实例讲解，实例基本都配有解题思路的分析和延伸指导。这对读者的思维开拓都具有很好的启发和带动作用。本书所有案例不但通过了Turbo C 3.0之前的版本的测试，在最新的编译环境（如Microsoft Visual Studio 2010/2012）中也能通过，当然Microsoft Visual C++ 6.0编译环境更适合案例的实现。

为了更好地调动初学者和自学者的学习积极性，巩固所学，本书在每章末都附有典型习题。习题的分析解答过程和答案在与之配套的《案例式C语言程序设计实验指导》中。读者可以仔细阅读解题过程，自行在编译环境下进行调试，从而更好地掌握程序设计的编程思想，熟悉不同问题的解题算法，提高自身的程序设计能力和思维分析能力。

本书由王富强、张春玲、刘明华担任主编，孔锐睿、孙劲飞、李朝玲担任副主编。王富强、孔锐睿负责书稿的设计、修改和统稿，其中，第1章、第3章、第6章和第7章由王富强编写，第2章、第11章由李朝玲编写，第4章和第12章由刘明华编写，第8章和第9章由孙劲飞编写，第10章、第13章由张春玲编写，第5章由王富强、孔锐睿共同编写，附录等由孔锐睿编写。

在本书编写过程中，得到了青岛科技大学相关职能部门、信息科学技术学院很多教师的支持与帮助，在此表示感谢。由于时间仓促，编者的水平有限，书稿中难免出现错误和不妥之处，恳请各位读者指正，以便再版时能及时修正。

编　者

2016年5月2日

目 录

第一部分 基础知识

第 2 部分　程序设计基本结构

第 3 部分　程序设计方法和具体应用

第 1 部分　基础知识

第 1 章 C 语言简介

C 语言是一门高级程序设计语言，也是现在国际上比较流行的计算机程序设计语言之一。C 语言自 1973 年在美国贝尔实验室成为一种标准语言之后，便受到广大开发者欢迎，如今已经成为世界上使用最广泛、最流行的高级程序设计语言之一。那么，C 语言到底什么样子呢？我们看下面简单的案例。

【例 1-1】 第一个程序“Hello, World!”。

```
// example1-1
#include "stdio.h"
void main()
{
    printf("Hello,World! \n ");
}
```

我们从【例 1-1】入手，分析 C 语言的格式和特点和介绍 C 语言。

1.1 计算机语言的发展

介绍 C 语言之前，先了解计算机语言，因为 C 语言是计算机语言的一种。

计算机语言（Computer Language）是人与计算机之间传递信息的媒介，用于人与计算机之间的通信。为了使计算机按照人类的指令进行各种工作，计算机系统就需要有一套人能够编写并能翻译后计算机能读懂的程序，用来表示生活中的数字、字符和语法规则，以通过指令把命令传达给机器。由这些字符和语法规则组成的计算机的各种指令（或各种语句）就是计算机语言。

计算机语言的发展经历了机器语言、汇编语言、高级语言 3 个阶段。

1.1.1 机器语言

机器语言是指计算机能够完全识别的指令集合，是最低、最早的程序语言，是由“0”和“1”组成的二进制数（代码），而二进制是计算机的语言基础。计算机发明之初，人们将一串串由“0”和“1”组成的指令序列交由计算机执行。这就是计算机唯一能够真正识别的机器语言。使用机器语言编写程序是十分痛苦的，特别是程序有错需要修改的时候。

1.1.2 汇编语言

为了减轻使用机器语言编程的痛苦，人们进行了一些改进。用一些简洁的英文字母、符号串来替代一个（串）特定的已编写的指令的二进制串，比如用“ADD”代表加法，“MOV”代表数据传递等。这样一来，人们很容易读懂并理解程序在干什么，纠错及维护就变得方便了。这种程序设计语言即第二代计算机语言，称为汇编语言。然而，计算机是不认识这些符号的。这就需要一个专门的程序，专门负责将这些符号翻译成二进制代码的机器语言。这种翻译程序被称为汇编程序。

汇编语言十分依赖于机器硬件，移植性不好，但效率十分高，尤其在结合计算机硬件方向上更能发挥特长，所以至今仍是一种强有力的软件开发工具。

1.1.3 高级语言

从最初与计算机交流的痛苦经历中，人们意识到应该设计一种语言。这种语言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，编出的程序能在所有机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了。60 多年来，共有几百种高级语言出现，有重要意义的、影响较大、使用较普遍的有 FORTRAN、BASIC、Pascal、C、PROLOG、C++、VC、VB、Java 等。从另一个角度分类，高级语言中的 VC、Java 等也被定义为面向对象语言，所以也有把面向对象语言划分为第四类语言。

1.1.4 计算机语言的概念

了解了计算机语言的发展，下面我们再了解几个概念。

指令：一条机器语言称为一条指令。指令是不可分割的最小功能单元。

程序：早期的程序就是一个个的二进制文件，如今程序可以定义为“计算机要执行的指令的集合”。

机器语言是第一代计算机语言。早期人们通过机器语言向计算机发出指令，无需借助翻译程序就能运行机器语言编好的程序来执行。

汇编语言是第二代语言，其实质和机器语言是相同的，都是直接对硬件操作，只不过指令采用了英文缩写的标识符，更容易识别和记忆。

高级语言是目前绝大多数编程者的选择。它们虽然需要借助翻译程序才能被计算机识别，但其简化了程序中的指令，并且去掉了与具体操作有关但与完成工作无关的细节。

高级语言的发展经历了从早期语言到结构化程序设计语言及面向过程到非过程化程序语言的过程。20 世纪 60 年代中后期，软件各自为战，后期出现的“软件危机”就是因为兼容性错误和困难造成的。1970 年面向过程的结构化程序语言——Pascal 的出现，标志着结构化程序设计时期的开始。20 世纪 80 年代初开始，面向对象的程序设计语言如 C++、Visual Basic、Delphi 出现。高级语言的下一个发展目标是面向应用，也就是说只需要告诉程序要干什么，程序就能自动生成算法进行处理。这是非过程化的程序语言。

1.2 C 语言的发展及其特点

1.2.1 C 语言的发展

C 语言是目前世界上最流行、使用最广泛的面向过程的高级程序设计语言之一。C 语言在操

作系统、编译程序及硬件模块需求等方面的操作优势，明显优于其他高级语言，许多大型应用软件都是用 C 语言编写的。

C 语言的原型是 ALGOL 60（ALGOrithmic Language 60）语言（也称 A 语言）。1963 年剑桥大学将 ALGOL 60 语言发展成为 CPL（Combined Programming Language）语言。1967 年马丁·理查兹（Matin Richards）简化了 CPL 语言产生了 BCPL（Basic Combined Programming Language）语言。1970 年美国贝尔实验室的肯·汤普森（Ken Thompson）将 BCPL 进行了修改，起了一个有趣的名字“B 语言”，并编写了第一个 UNIX 操作系统。

1973 年美国贝尔实验室的 D.M.RITCHIE 最终设计出了一种新的语言——C 语言，名字取自 BCPL 的第二个字母。1978 年布莱恩·科尔尼干（Brian W.Kernighian）和丹尼斯·里奇（Dennis M.Ritchie）出版了名著《The C Programming Language》（中文译名为《C 程序设计语言》），称之为《K&R》标准，但是《K&R》中并没有定义一个完整的标准 C 语言。1983 年，美国国家标准化协会（American National Standards Institute，ANSI）制定了一个 C 语言标准，通常称之为 ANSI C。1987 年，C 语言有了 ANSI 标准，立刻成为最受欢迎的语言之一。

1990 年，国际化标准组织（International Standard Organization，ISO）接受了 87 ANSI C 为 ISO C 的标准（ISO9899-1990），简称为 C90。1999 年，ISO 对 C 语言标准进行修订，主要是增加了一些功能，尤其是 C++中的一些功能，简称为 C99。2011 年又发布了新的标准，简称为 C11。目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的，但不同版本的 C 编译系统实现的语言功能和语法规则略有差别。

C 语言在发展的过程中，逐步完善，拥有绘图能力强、可移植性好及很强的数据处理能力等优点。因此，系统软件的编写及二维、三维图形的绘制和动画制作与处理等都是它的强项之一。

1.2.2　C 语言的特点

C 语言的特点主要包括以下几个方面。

1. 简洁紧凑，灵活方便

C 语言一共有 32 个关键字、9 种控制语句，程序书写自由，主要用小写字母表示。C 语言把高级语言的基本结构和语句与低级语言的实用性结合了起来，简洁紧凑，灵活方便。

2. 运算符和数据结构丰富

C 语言的运算符较多，共有 40 多个。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。程序开发者灵活使用 C 语言的各种运算符可以实现在其他高级语言中难以实现的运算。

C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。这使得 C 语言能实现各种复杂的数据类型的运算。另外，C 语言引入了指针概念，使程序效率更高。

3. C 是结构式语言

结构式语言是 C 语言的显著特点。结构化方式使程序层次清晰，便于使用、维护及调试。C 语言是以函数形式提供给用户的，多种循环、条件语句控制和函数调用使程序完全结构化。

4. C 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 语言允许程序编写者有较大的自由度，限制并不严格，尤其在越界检查方面，几乎没有限制。

5. 允许直接访问物理地址，直接操作硬件

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，而这 3 者是计算机最基本的工作单元，因此，C 语言可以用来编写系统软件。

6. 程序执行效率高

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%。

7. 可移植性好

C 语言有一个突出的优点就是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

当然，C 语言也有自身的不足，比如 C 语言的语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不做检查等。从应用的角度，C 语言相比其他高级语言较难掌握。

1.3 C 语言的程序格式和结构

1.3.1 最简单的 C 语言程序举例

要了解 C 语言的程序格式与结构，可以从引入案例的程序中分析 C 语言程序的特点，总结其格式和基本结构。

以【例 1-1】第一个程序“Hello, World!”为例。

```
// example1-1  The first C Program 行 1 —— 注释
#include "stdio.h"    //行 2——编译预处理
void main()          //行 3 ——函数
{
    printf("Hello,World! \n ");  //行 5 ——语句
}
```

本程序运行后输出如图 1-1 所示信息。

图 1-1 Hello, World!

本程序与引例程序相比，增加了注释说明，解析如下。

第一行是注释，对程序的编译和执行不起任何作用，目的是使读者无需看后续很长的程序代码也能知晓本程序的功能。注释语句常用“//” 开头，其后的任何信息都是注释信息。除了“//”之外，还可以使用对称结构“/*…*/”作为注释语句，详细的注释信息在“/*”和“*/”之间，二者顺序不能更换，个数不能多。注释信息可以放在程序的任何位置，可以为任何文字或字符，可以单独成行，也可以与其前被注释的信息为一行。

第二行是编译预处理，对 C 语言程序中的输入/输出等系统函数调用声明，printf 输出函数和 scanf 输入函数保存在“stdio.h”头文件中，所以编译预处理“#include "stdio.h"”不可少。如果有其他系统函数被调用，也必须有对应的编译预处理文件声明，详细的介绍见第 9 章。

第三行中的 main 是函数的名字，main 前面的 void 表示此函数类型为空类型，即执行此函数不产生一个函数值。有些函数会返回一个值，如数学函数 *sin*(*x*)、*cos*(*x*)等。C 语言中规定：任何一个 C 语言程序都必须有一个 main 函数，并且只能有一个 main 函数。

第四行和第六行：C 语言程序中，在函数后面的函数体是以“{”和“}”一对大花括号括起来的，在函数体后面的第一个“{”和最后一个“}”可分别对应函数体的开始与结束。

第五行为函数语句，以半角分号（;）作为语句结束符。在 C 语言中，没有行的概念，只是以分号（;）作为语句的结束符，也就是 C 语言程序以语句作为最基本的函数体单位。第五行的“printf("Hello,World!\n ");”把双引号内的内容“Hello,World!”原样输出，并换行，其中“\n”是换行符。

【例 1-2】 第二个程序，计算两个数 *x* 和 *y* 的和。

```
#include "stdio.h"      //行 1 编译预处理
void main()             //行 2   main 主函数
{
   int x,y,sum;         //行 4   定义整数型变量 x、y 和 sum
   x=123;               //行 5   对变量 x 赋值
   y=456;               //行 6   对变量 y 赋值
   sum=x+y;             //行 7   求和赋值给变量 sum
   printf("sum is =%d\n",sum);  //行 8   输出 sum
}
```

本程序运行后输出如图 1-2 所示信息。

图 1-2　计算两个数 *x* 和 *y* 的和

本程序省略了注释行，没有注释信息。程序的主要作用是求已知的两个整数 *x* 和 *y* 的和，并输出 *sum* 的值。下面着重解析第四行到第八行。

第四行：声明部分，定义变量 *x* 和 *y*，其中定义的 *x* 和 *y* 为整数类型（int）变量，存放整数。

第五行和第六行：赋值语句，对定义的变量 *x*、*y* 分别赋值 123 和 456。

第七行：对已经赋值的整数 *x*、*y* 进行求和计算，计算后的和赋值给变量 sum（存放和，否则求和数据会存放在计算机中的任何变量中，无法寻找和输出）。

第八行：输出 *sum*。输出语句中使用了格式控制符，字符串“sum is =”原样输出，*sum* 以“十进制整数类型”输出。格式控制符的相关知识将在第 4 章详细讲述。

【例 1-3】 第三个程序，求两个数 *x* 和 *y* 最大值。

```
#include "stdio.h"              //行 1   编译预处理
void main()                     //行 2   main 主函数
{
   int max(int x,int y);        //行 4  声明被调用函数
   int a,b,c;                   //定义变量 a、b、c
   scanf("%d%d",&a,&b);         //行 6   输入变量 a、b 的值
   c=max(a,b);                  //行 7   调用 max 函数求最大值，结果赋值给变量 c
   printf("max is %d\n",c);     //输出最大值
}
int max(int x,int y)            //行 10   自定义求最大值函数 max,形式参数有 x 和 y 两个
{
   int z;                       //定义变量 z
   if(x>y) z=x;
   else z=y;                    //使用 if 语句求任意两个变量 x、y 的最大值
   return z;                    //返回最大值 z
}
```

本程序运行后输出如图 1-3 所示信息。

图 1-3　求两个数 *x* 和 *y* 最大值

解析如下。

第四行：“int max(int x,int y);”声明调用函数 max。

max 是自定义函数，其作用就是求出两个数的最大值并返回，详细的函数功能代码见第十行后。

第六行：从键盘上读入两个数分别赋值给变量 *a*、*b*。

第七行：调用自定义函数 max 求变量 *a*、*b* 的最大值，并把最大值赋值给变量 *c* 存储。

第十行：自定义函数 max 的功能是求最大数，其中 max 有两个形式参数 *x* 和 *y*，max 函数的函数体，其作用是求 *x* 和 *y* 的最大值，把最大值赋值给 *z*，最后把求出的最大值 *z* 返回。

本程序一共包括两个函数，其中一个是 main 主函数，另一个是求最大值的自定义函数 max。main 函数调用 max 函数把最大值返回到调用 max 函数的位置。

main 函数中 scanf 函数的作用是输入变量 *a*、*b* 的值，而 printf 函数的作用是输出最大值到屏幕上。

1.3.2 C 语言程序的结构

通过以上几个例子，可以总结出 C 语言程序组成和结构如下。

（1）一个程序由一个或多个源程序文件组成。简单的程序如【例 1-1】和【例 1-2】的源程序文件只由一个 main 函数组成，而【例 1-3】的源程序文件包含两个函数。一般源程序文件包含以下几个部分。

① 预处理指令：主要包括“#include”“#define”等以“#”为开始，在程序运行前实现的预处理。

② 变量声明：函数体“{}”内的声明是局部声明，在函数“{}”之外的声明为全局声明，两者的有效作用域不同，详见第 9 章预处理命令。

③ 函数定义：函数是 C 程序中最重要的部分，可以说整个 C 程序几乎都是由函数组成的。函数就是实现一定功能的程序模块，所以说函数的定义是 C 程序的功能体现。

（2）函数是 C 程序的基本单位，是 C 语言程序的主要组成部分。一个 C 程序是由一个或多个函数组成的，但 main 函数必须有并且只能有一个。C 语言程序总是从 main 函数开始，以 main 函数结束，其他函数只能通过 main 函数直接或间接调用。

（3）一个函数包含两个部分：函数首部和函数体。

① 函数首部：函数的第一行，包括函数名、函数类型、函数参数（形式参数）和参数类型。例如，【例 1-3】中的 int　max (int　x, int　y)。

② 函数体：函数首部下面的大括号开始的部分，当然如果一个函数体内存在若干个大括号，则最外层（或第一个“{”和对应的最后一个“}”）的一对大括号才是函数体语句范围。

函数体一般包括以下部分。

- 声明部分：包括变量定义和调用函数的声明，如【例 1-2】中的 int x,y,sum;和【例 1-3】中的 int max(int x,int y);。
- 执行部分：由若干个语句组成，在函数中执行一定操作，如【例 1-1】中的 printf("hello world!\n");是为了执行输入实现打印功能的。

（4）C 程序的函数由语句组成。C 程序语句以半角分号（;）作为分隔符，其也是语句唯一的终止标志。

（5）C 程序中没有程序行的概念，习惯使用小写字母。

（6）程序可以包含注释。注释在程序的执行中不起任何作用，也不会产生任何代码。

1.4　C 语言程序的运行与调试

1.4.1　C 语言程序的运行环境

一个 C 语言程序的运行离不开它的翻译程序，其称之为编译环境。目前使用最多的集成开发环境（Integrated Development Environment，IDE），就是把 C 语言程序需要连接的步骤——编辑、编译、链接和运行集成在一个界面上，通过不同的操作步骤实现。集成环境的优点：简单实用，功能丰富，直观易学。

不同的编译环境对 C 程序的操作是不同的。常用的编译程序有 Turbo C 2.0、Turbo C++ 3.0、Borland C++、Visual C++6.0、Microsoft Visual Studio 2010 等。在 20 世纪 90 年代，Turbo C 2.0 编译环境应用最为普遍，但其缺点是进入 DOS 环境后不能使用鼠标操作，几乎只能通过键盘完成。随后的 Turbo C++ 3.0 编译环境虽然已经完成启动文件快捷方式“tc.exe”，但鼠标只能执行部分操作，如文件保存、基本菜单的选择等。如今编译环境有了长足发展，尤其是全国计算机等级考试（C 语言模块）的编译环境——Microsoft Visual C++6.0 的推广，使 Visual C++6.0 编译环境的应用占据了很大的部分。随着 CPU 处理能力的进一步增强，64 位机逐渐成为主流，Microsoft Visual Studio 2010 支持 64 位的优势逐渐体现，在 Windows8 以上操作系统下安装使用 Microsoft Visual Studio 2010 越来越多。

Visual C++6.0 为用户开发 C 程序提供的集成环境包括源程序的输入和编辑、源程序的编译和链接、程序运行时的调试和跟踪、项目的自动管理、为程序的开发提供各种工具并具有窗口管理和联机帮助等功能。尤其可贵的是，在 Visual C++6.0 编译环境中，鼠标、键盘非常方便，复制、粘贴、剪切等基本操作与 Windows 环境下的操作几乎没有区别。

1.4.2　C 语言程序的几个概念

在了解 C 语言程序运行之前，先了解 C 语言中的几个概念。

程序：可以连续执行的指令集合。

源程序：使用高级语言编写的程序，如 Visual Basic、C、C++、Java 等编写的程序，其中 C 语言编写的程序可称为源程序，C 源程序文件后缀为.c。

目标程序：由二进制代码表示的程序，C 源程序生成的目标程序文件的后缀为.obj。

可执行程序：可移植可执行的文件格式，可加载到内存中由操作系统加载程序执行，如 C 源程序经过编译和链接后生成的后缀为.exe 的可直接运行的文件。

编译程序：具有翻译功能的软件如 Visual C++ 6.0、Microsoft Visual Studio 2010 等称为编译程序。

以上几个概念在 C 语言的调试运行的不同阶段出现。在编译程序 Microsoft Visual C++ 6.0 中输入 C 源程序，保存后的源程序经过编译命令生成目标程序，目标程序链接库函数后进一步生成可执行程序，最后运行可执行程序查看程序运行结果。

1.4.3　C 语言程序的运行调试

本书采用 Visual C++ 6.0 作为程序设计调试的环境，常用的 Microsoft Visual Studio 2010 的调试运行步骤在实验指导中详细介绍。

1. 启动 Visual C++ 6.0

通过鼠标双击桌面上的 Visual C++ 6.0 的图标，或通过菜单方式启动 Visual C++ 6.0，即用鼠标单击“开始”菜单，选择“程序”，选择“Microsoft Visual Studio 6.0”，选择“Microsoft Visual C++ 6.0”启动 Visual C++ 6.0。图 1-4 所示为启动后的可视化集成环境，窗口包括标题栏、菜单栏、工具栏和状态栏等。

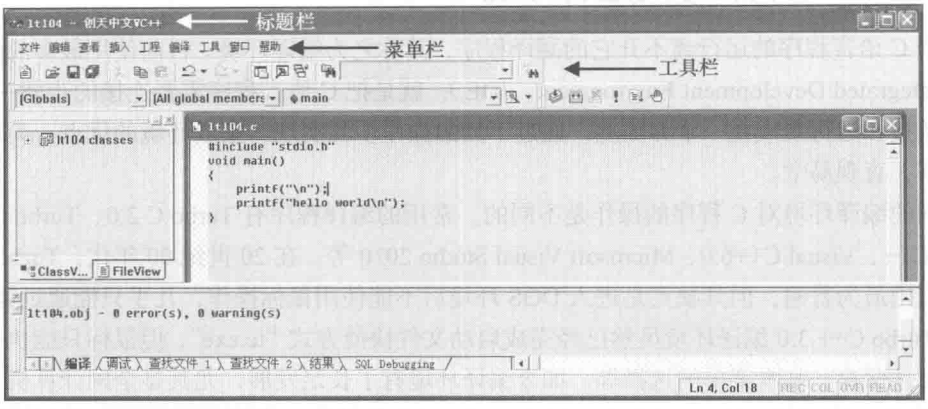


图 1-4　Visual C++ 6.0 集成环境

2. 生成源程序文件

选择“文件（File）”菜单中的“新建（New）”命令，产生“新建（New）”对话框，单击“文件”选项卡，选择 C/C++ Source File 选项，文件命名为*.c 格式如 lt104.c，并设置源文件保存目录，单击“确定”，生成源程序文件，如图 1-5 所示。

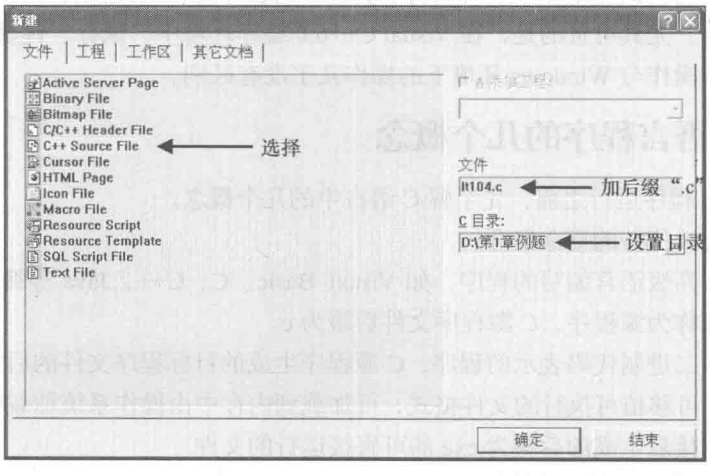


图 1-5　源文件生成

指定的文件名后缀为“.c”，如果输入的文件名缺少后缀“.c”，则系统默认为 C++ 源程序文件，自动加上后缀“.cpp”，因此后缀“.c”不能省略。

3. 编辑源程序

在程序编辑区输入源程序，如图 1-6 所示，选择“文件”菜单下的“保存”。