

21世纪高等教育计算机规划教材



C 语言 程序设计

C Programming Language

■ 肖晓霞 罗铁清 彭荧荧 主编

—— 揭开 C 程序设计之神秘面纱，夯实实践操作
—— 辅以简单成绩管理系统设计，激发学习兴趣
—— 开启现实世界抽象建模之门，培养计算思维



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

C 语言 程序设计

C Programming Language

肖晓霞 罗铁清 彭荧荧 主编



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 / 肖晓霞, 罗铁清, 彭荧荧主编. —
北京: 人民邮电出版社, 2014. 7
21世纪高等教育计算机规划教材
ISBN 978-7-115-35053-4

I. ①C… II. ①肖… ②罗… ③彭… 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(2014)第076000号

内 容 提 要

本书以学生成绩管理系统为主线, 以简单实例和生活实例为导向, 深入浅出地讲解了C语言程序设计的各个基本知识点和程序设计的基本方法, 旨在强调对实际问题应用计算机处理能力的培养。

本书立足于对计算思维的培养, 通过生活实例和中医药实例的程序设计引导学生对信息化进行思考, 以此强化学生信息化意识, 培养其信息思维。本书对C语言中的概念和要点阐述透彻, 对容易混淆的知识点有特别提示。全书分为两个部分, 共10章。内容包括算法简介、数据类型、结构化编程的三个基本结构语句、函数、数组、指针、构造数据类型和文件等, 涵盖了C语言的基本内容。

本书内容全面, 阐述精简、深入浅出, 文字流畅、通俗易懂, 概念清晰, 难易有度, 既有适合一般性学习的简单实例, 又有适合深入实践和学习的软件系统实例。本书是C语言初学者的理想教材, 可作为高等学校各专业的正式教材, 也是一本适合自学的教材。本书还配套有辅助教材《C语言程序设计实验指导与习题解答》。

◆ 主 编 肖晓霞 罗铁清 彭荧荧

责任编辑 邹文波

执行编辑 吴 婷

责任印制 彭志环 杨林杰

◆ 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>

北京鑫正大印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 17.75

2014年7月第1版

字数: 462千字

2014年7月北京第1次印刷

定价: 39.00元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

本书编委会

主 编：肖晓霞 罗铁清 彭荧荧

副主编：杨连初 杨 平 陈兴华

编 委：（按姓氏拼音排列）

陈兴华 刘东波 罗铁清 穆 珺

彭荧荧 瞿昊宇 任学刚 肖晓霞

杨连初 杨 平

“C 语言程序设计”是计算机相关专业和非计算机专业的基础课程之一，也是全国计算机等级考试科目之一，同时更是培养学生计算思维的课程。本书正是基于培养计算思维的目的，结合编者多年教学改革和前期编写过 C 语言程序设计教材的经验，汲取国内外同类教材的特色，以学生成绩管理小系统为主线，以丰富的程序设计实例、部分生活实例以及中医药实例为辅助，阐述了 C 语言程序设计基本方法和技巧，同时简要地讲述了应用计算机解决实际问题的方法，以此加深读者对信息化的理解。本书主要特点如下。

◆ 以软件系统为主线，激发学习兴趣

以一个学生非常熟悉的“学生成绩管理小系统”为主线，逐渐深入阐述各相关知识点，培养学生的计算思维。从第 2 章开始，最后都有一节应用提高篇围绕成绩管理系统根据该章难度进行实例讲解。独立成节有利于教师根据课时和授课对象水平进行选择，也有利于学生自学过程的循序渐进，学完所有知识点后可以自主开发一个小系统，增强学生学习成就感。

◆ 以简单实例导入，培养探究式学习习惯

每一章节都采用含有 1~2 个新知识点的简单实例引入，这类例题给学生的第一印象是自己会做，但又有小部分问题无法求解，以此引导学生对未知知识点的探索，培养学生探索式学习能力。

◆ 精心设计中医药实例，引导中医药信息化思考

本书精心设计了通俗易懂的中医药相关问题的例题和练习题，目的是通过这类例题和练习题培养学生的计算思维能力，同时也是引导学生对中医药信息化的思考，无论是计算机相关专业的学生还是非计算机专业的学生都可以通过对这类问题的求解引发更深的思考，本书所做的工作也只是抛砖引玉的作用，期待更多的同仁关注和改进。

◆ 增加变量跟踪图，深入理解程序运行规律

大多数章节有 1~2 个例题会有变量跟踪图，展现程序运行过程，方便学生自学，以此引导学生理解程序运行的过程和规律。

◆ 强调程序后期测试，培养严谨科研习惯

本书非常强调程序编写的后期测试，以此培养学生自主检查程序的能力，旨在培养学生严谨的科研习惯。

◆ 紧扣学习心理，简化知识表达

新知识点的阐述紧扣初学者心理，对容易出错和误解的知识点采用多个简单实例逐一展示，并以容易理解的方式阐述，知识点阐述到位而不啰嗦，让学生迅速理解新知识。

本书整体结构和内容编排由肖晓霞老师设计，“学生成绩管理系统”由软件工程专业老师罗铁清副教授分章完成，每一章节都是由教授 C 语言多年的一线老师

完成,每一章节都融入了各位老师多年的教学经验。本书共有两个部分,第一部分的第 1、9 章由肖晓霞编写,第 2、6 章由穆珺编写,第 3 章由杨平编写,第 4 章由彭炎炎编写,第 5 章由罗铁清编写,第 7 章由杨连初编写,第 8 章由陈兴华,第 10 章由任学刚编写,瞿昊宇、刘东波参与教材的修订;第二部分为附录,由肖晓霞整理。

本书可作为高等院校各学科非计算机专业或计算机应用专业的教材,也可作为高职院校或专科学校的教材,全国计算机等级考试(NCRE)或其他培训机构的培训教材。

全书内容深入浅出,知识点覆盖面广,力求使初学者全面理解 C 语言程序设计基础知识。程序设计是一门实践性很强的课程,既要掌握概念,又要多动手编程,还要上机调试运行。各章后附有大量的习题,以帮助读者理解基本概念,希望读者一定要重视实践环节,完成各章后的习题,进一步熟练掌握 C 语言的语法结构和应用,提高程序设计能力。与本书配套的《C 语言程序设计实验指导与习题解答》给出了本书中习题的全部参考答案和学生上机实验的内容,两本书结合使用效果更好。

本书力求为大家提供一个有启发、有帮助的、清晰的 C 语言入门指导,使初学者能愉快地学习 C 语言。若有更好建议,恳请广大读者不吝赐教。邮箱地址为 amily_x@126.com,在此我们表示真诚的谢意。

本书及《C 语言程序设计实验指导与习题解答》中所有的源文件均在 Microsoft visual C++ 6.0 环境下调试通过。本书的源代码及其他相关资料可从人民邮电出版社教学服务与资源网(www.ptpedu.com.cn)下载,也可直接联系本书作者,联系方式同上。

编 者

2013 年 12 月

目 录

第 1 章 概述..... 1	第 3 章 基本程序语句..... 20
1.1 计算机程序..... 1	3.1 基本数据类型..... 20
1.2 C 语言的发展过程..... 1	3.1.1 C 语言数据类型..... 20
1.3 C 语言的特点..... 2	3.1.2 常量..... 21
1.4 编写程序..... 3	3.1.3 变量..... 21
1.4.1 问题说明和分析..... 3	3.2 整型数据..... 23
1.4.2 编写和编译程序..... 4	3.2.1 整型常量..... 23
1.4.3 执行和测试程序..... 4	3.2.2 整型变量..... 24
1.5 简单的 C 语言程序及其结构..... 4	3.3 实型数据..... 26
1.6 VC6.0 简介..... 6	3.3.1 实型常量..... 26
1.6.1 C 程序上机过程..... 6	3.3.2 实型变量..... 27
1.6.2 在 VC6.0 上运行 C 程序的步骤..... 7	3.4 字符型数据..... 28
1.6.3 在 VC6.0 上调试 C 程序的简介..... 8	3.4.1 字符型常量..... 28
1.7 本章小结..... 9	3.4.2 字符串常量..... 31
习题一..... 10	3.4.3 字符型变量..... 31
第 2 章 简单算法引导..... 11	3.5 运算符与表达式..... 32
2.1 算法设计简介..... 11	3.5.1 算术运算符与算术表达式..... 33
2.1.1 问题分析、抽象和建模..... 11	3.5.2 赋值运算符与赋值表达式..... 37
2.1.2 确定解决问题的方法和步骤..... 12	3.5.3 关系运算符与关系表达式..... 40
2.1.3 算法正确性证明和分析..... 12	3.5.4 逻辑运算符和逻辑表达式..... 41
2.1.4 算法实现..... 12	3.5.5 条件运算符与条件表达式..... 43
2.2 算法的特征..... 13	3.5.6 逗号运算符和逗号表达式..... 44
2.3 结构化程序的算法描述..... 13	3.6 标准输入输出函数..... 44
2.3.1 自然语言..... 14	3.6.1 字符输出函数..... 45
2.3.2 流程图..... 14	3.6.2 格式化输出函数..... 46
2.3.3 伪代码..... 15	3.6.3 字符输入函数..... 49
2.4 应用与提高..... 16	3.6.4 格式化输入函数..... 50
2.4.1 学生成绩管理系统模块化设计..... 16	3.7 应用与提高..... 52
2.4.2 学生成绩管理系统算法设计..... 17	3.8 本章小结..... 53
2.5 本章小结..... 18	习题三..... 53
习题二..... 18	第 4 章 选择结构..... 60
	4.1 if 语句..... 60
	4.1.1 示例问题：根据某同学的

C 语言成绩判断是否需要重修	60	6.6.1 函数的嵌套调用	112
4.1.2 单分支 if 语句	60	6.6.2 递归函数	114
4.1.3 双分支 if 语句	61	6.7 变量的存储方式和生存周期	117
4.1.4 多分支 if 语句	62	6.8 内部函数和外部函数	122
4.2 switch 语句	66	6.8.1 内部函数	122
4.2.1 switch 语句简介	66	6.8.2 外部函数	123
4.2.2 break 语句在 switch 结构 中的运用	67	6.9 编译预处理	123
4.3 应用实例	68	6.9.1 宏定义	124
4.4 应用与提高	71	6.9.2 文件包含	128
4.5 本章小结	75	6.9.3 条件编译	129
习题四	75	6.10 应用与提高	132
第 5 章 循环结构	79	6.11 本章小结	134
5.1 示例问题: 累加求和	79	习题六	135
5.2 goto 语句构成的循环	80	第 7 章 数组	141
5.3 for 循环	81	7.1 一维数组	141
5.4 while 循环	84	7.1.1 一维数组初始化	141
5.5 do...while 循环	85	7.1.2 一维数组元素的引用	145
5.6 循环的嵌套	86	7.1.3 一维数组应用举例	146
5.7 break 语句和 continue 语句	88	7.1.4 一维数组与函数	147
5.7.1 break 语句	88	7.2 多维数组	150
5.7.2 continue 语句	90	7.2.1 多维数组的定义	150
5.8 应用与提高	90	7.2.2 二维数组的定义和初始化	150
5.9 本章小结	92	7.2.3 二维数组应用举例	152
习题五	92	7.2.4 二维数组与函数	153
第 6 章 函数	97	7.3 字符数组	155
6.1 函数概述	97	7.3.1 字符数组的初始化	156
6.2 函数的定义	100	7.3.2 字符数组元素的引用	157
6.3 函数的声明和调用	101	7.3.3 字符串的输入和输出	158
6.3.1 函数的声明	102	7.3.4 字符串处理函数	159
6.3.2 函数的调用	103	7.3.5 字符串数组应用举例	161
6.4 函数参数传值方式	105	7.4 应用与提高	162
6.4.1 值传递	106	7.5 本章小结	164
6.4.2 地址传递	107	习题七	165
6.5 局部变量和全局变量	108	第 8 章 指针	171
6.5.1 局部变量	108	8.1 指针的定义	171
6.5.2 全局变量	109	8.2 指针变量	172
6.6 函数的嵌套调用和递归函数	112	8.2.1 指针变量的定义	172
		8.2.2 指针变量与指针变量的引用	172

8.2.3 指针变量作为参数	175	9.3.2 指向结构体数组的指针	224
8.2.4 void 指针类型	178	9.4 链表	225
8.3 指针与数组	179	9.4.1 链表的概念	225
8.3.1 指向数组元素的指针变量	179	9.4.2 链表的创建和使用	226
8.3.2 指向数组的指针变量	180	9.5 共用体数据类型简介	230
8.3.3 指针的算术运算和 关系运算	181	9.5.1 共用体的定义和共用体 变量的声明	231
8.3.4 数组名作为函数参数	182	9.5.2 引用共用体变量	231
8.4 通过指针引用多维数组	188	9.6 枚举类型	232
8.5 通过指针引用字符数组	191	9.6.1 枚举类型定义	232
8.5.1 字符串的引用方式	191	9.6.2 引用枚举类型变量	232
8.5.2 字符指针作为函数参数	193	9.6.3 枚举类型举例	233
8.5.3 指针在字符串处理中的 综合应用	194	9.7 typedef 简介	233
8.6 指针数组	195	9.8 应用与提高	235
8.6.1 指针数组的定义	195	9.9 本章小结	239
8.6.2 指针数组作为函数参数	196	习题九	239
8.7 返回指针的函数	198	第 10 章 文件	242
8.8 指向函数的指针	199	10.1 文件的概念	242
8.9 多级指针	201	10.1.1 示例问题: 将某同学的 C 语言成绩保存到文本 文件中 (永久保存)	242
8.10 应用与提高	202	10.1.2 文件是什么	243
8.11 本章小结	207	10.1.3 文件的分类	243
习题八	209	10.1.4 C 语言对文件的处理方法	244
第 9 章 构造数据类型	216	10.1.5 文件指针	244
9.1 结构体数据类型	216	10.2 文件的打开与关闭	245
9.1.1 示例问题: 打印学生 基本信息	216	10.2.1 fopen 函数	245
9.1.2 结构体类型的定义	216	10.2.2 fclose 函数	247
9.1.3 结构体变量的定义和 结构体变量的声明	218	10.3 文件的顺序读写	248
9.1.4 结构体变量的引用	219	10.3.1 字符读写函数	248
9.1.5 结构体变量的初始化和 应用举例	220	10.3.2 字符串读写函数	251
9.2 结构体数组	220	10.3.3 格式化读写函数	252
9.2.1 结构体数组的定义	220	10.3.4 数据块读写函数	254
9.2.2 结构体数组应用举例	222	10.4 文件的随机读写	257
9.3 结构体指针	223	10.5 文件操作的错误检测	259
9.3.1 指向结构体变量的指针	223	10.6 应用与提高	260
		10.7 本章小结	263
		习题十	263

附录 A 常用字符与 ASCII

代码对照表 266

附录 B 基本库函数 268

B.1 数学函数 268

B.2 字符串函数 268

B.3 时间日期函数 269

B.4 类型转换函数 270

B.5 存储分配函数 270

B.6 输入输出子程序 271

B.7 文件函数 271

第 1 章

概述

C 语言是一种强大的专业化编程语言，应用广泛且深受业余和专业编程人员的喜爱。本章就是为学习和使用 C 语言编写程序做准备，同时还介绍了使用 Microsoft Visual C++ 6.0 编写程序的过程。

1.1 计算机程序

程序是一系列计算机能识别和执行的指令。计算机技术的发展使很多人认为计算机非常强大，以至于有些初学程序设计者会想当然地认为计算机具有像人一样的思维。例如，初学者容易犯的一个错误就是没有输出指令但认为会有输出结果。而实际上，计算机都是按事先编写好的程序，逐条执行指令，最后得到最终结果的。

计算机语言是用于人与计算机之间通信的语言，能使计算机理解人的指令。C 语言是一种被广泛使用的高级程序设计语言，使用 C 语言编写的程序不能被计算机识别，必须通过 C 编译程序编译成机器代码才可以被计算机识别。一个用 C 语言或其他语言编写的程序都是利用计算机解决现实问题的抽象描述，是有头有尾有顺序的；并且程序必须严格按照计算机语言规定的拼写、语法、标点符号规则和字母大小写要求来编写；程序作为一个整体是一个能正确表达实际意义的指令集合。

总之，利用计算机解决实际问题，需要将实际问题抽象为一系列有序的指令集合，由程序控制计算机解决问题的每一个步骤，以达到让计算机自动按程序解决实际问题的目的。

1.2 C 语言的发展过程

C 语言的发展要追溯到 ALGOL 60 语言。1963 年，为了使 ALGOL 60 语言更接近硬件，英国剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language) 语言。1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL (Basic Combined Programming Language) 语言。1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了简化，推出了 B 语言，并且他用 B 语言写了第一个 UNIX 操作系统。1973 年，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。1973 年，K.Thompson 和 D.M.Ritchie 合作将用汇编语言编写的

UNIX 的 90%的代码用 C 语言改写，也就是 UNIX 第 5 版。

为了使 UNIX 操作系统得到推广，1977 年，Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年，Brian W.Kernighan 和 Dennis M.Ritchie 合著了非常有影响力的名著《The C Programming Language》(通常简称为《K&R》)，这本书中介绍的 C 语言被称为标准 C。随着 C 的发展和更加广泛地应用于更多种类的系统上，程序员们意识到它需要一个更加全面、新颖和严格的标准。为了满足这一要求，美国国家标准化组织 (ANSI) 于 1983 年为 C 语言制定了 ANSI C 标准，此标准于 1989 年正式采用。ANSI C 标准定义了语言和一个标准 C 库。国际标准化组织于 1990 年采用了一个 C 标准——ISO C。ISO C 和 ANSI C 都可以称为标准 C。

1.3 C 语言的特点

1. 简洁紧凑、运算符丰富

C 语言一共有 32 个关键字 (见表 1-1)，9 种控制语句，程序书写自由。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 的运算符包含的范围很广泛，共有 34 个运算符。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。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

表 1-1

C 语言的保留关键字

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

2. 数据结构丰富

C 具有丰富的数据类型：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。C 语言能用来实现各种复杂的数据类型的运算，并引入了指针概念，使程序效率更高。另外 C 语言具有强大的图形处理功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

3. C 是结构化程序设计语言

结构化程序设计语言的显著特点是代码及数据的分离化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使用户采用自顶向下的规划、结构的编程以及模块化的设计，它使程序层次清晰，便于使用、维护以及调试。C 的结构化编程特点使编写出的程序更可靠、更易懂，且容易维护。

4. 可直接访问物理地址、直接对硬件进行操作

C 语言通常被称为中级语言，它既具有高级语言的全部功能，又具有低级语言的许多功能。它能够像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元，可以用来写系统软件。UNIX 操作系统 90%的代码就是用 C 语言编写，而且很多语言的编译器和解释器都是用 C 语言编写的，如 Logo、PASCAL 和 BASIC 等。

5. 生成代码质量高, 执行效率高

C 程序紧凑、运行速度快, 而且, C 可以表现出只有汇编语言才具有的精细控制能力, 它可以被细调以获得最大速度或占用最小内存。一般只比汇编程序生成的目标代码效率低 10%~20%。

6. 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。C 是一种可移植性语言, 即在一个系统上编写的 C 程序经过很少的改动或不经修改就可以在其他系统上运行, 而且通常需要修改的也只是伴随主程序的一个头文件的几项内容。但 C 程序中为访问特定硬件设备或操作系统的特殊功能而专门编写的部分, 通常是不可移植的。

7. 灵活、设计自由度高

C 编译器对语法的检查并不严格, 如, C 语言不对数组元素的下标做语法检测。对变量类型的使用非常灵活, 如, 整型数据、字符型数据和逻辑型数据可以通用, 因此程序员编写程序时具有较大的自由度。C 语言可以使用自由度非常高的跳转语句 `goto`, 此语句几乎可以使程序跳转到任意地方。C 在表达方面的自由可以凸显程序员的编程技巧, 同时也增加了程序设计中不必要的麻烦和风险。如, `goto` 语言的过多的使用会使程序难以理解, 指针不恰当地使用很可能造成难以追踪的编程错误, 这些都要求程序员提高警惕, 养成良好的编程习惯, 避免出错。

1.4 编写程序

用 C 语言编写程序就必须了解计算机的工作原理, 计算机在运行时, 先从内存中取出第一条指令, 通过控制器的译码, 按指令的要求, 从存储器中取出数据进行指定的运算和逻辑操作等加工, 然后再按地址把结果送到内存中去。接下来, 再取出第二条指令, 在控制器的指挥下完成规定操作, 依次进行直至遇到停止指令。编写程序实际上就是将具体的现实问题抽象为数学模型后利用计算机进行处理, 所以编写程序需要如下所示步骤。

1.4.1 问题说明和分析

对于实际问题的求解, 必须完全了解问题是什么, 例如求 $1!+2!+3!+\cdots+n!$ 的值, 其中 $n<21$ 。从这个问题可以看出是求阶乘之和, 那么什么是阶乘? 如何求解呢? 这就需对实际问题进行分析, 得到解决问题的算法。算法的具体内容会在第 2 章进行简述, 若对算法感兴趣可以参考更专业的资料。下面就求阶乘之和的问题, 简单介绍问题说明和分析包含的重要几点。

(1) 定义程序实现的功能

开始编写程序时, 总是需要清晰地知道程序可以做什么, 在这一步无需计算机语言, 只需明确程序的功能就可以。例如上例中要实现的功能就是求 20 以内阶乘之和。

(2) 定义算法适用范围

从问题中我们发现, 求和的每一项都是阶乘, 求 n 阶乘的算法很简单, 就是 $1\times 2\times 3\times\cdots\times n$ 即可。在问题求解中可以发现, $n!$ 的值随着 n 的增长而迅速增长。那么程序求解过程中是否能表示这一整数呢? 在实际应用中恰当地处理极大的数字或避免出现极大的数字是相当复杂的。

(3) 定义输入和输出

针对实际问题, 需要明确是否有输入。输入的数据的类型、数值范围是什么, 由此来确定数据的具体类型。根据例中求解目标——求 20 以内的阶乘之和, 可以判断输入项是一个小整

数，而大整数的阶乘之和是无法简单采用步骤（2）中的方法来直接求解的。

此外还得考虑输出内容，首先得考虑能正确输出结果的类型、范围和精度，如从例中可以看到 $20!$ 将是一个非常大的数，要选择一个可以表示的类型来存储输出结果，对精度的考虑要涉及到 C 语言数据表示的范围以及多次反复使用一个非精确值计算所带来的精度问题；其次还得考虑输出内容能明确表达该结果属于哪一个输入集等，例如上例输出内容可以为“ $1!+2!+\cdots+6!=873$ ”。

（4）定义常量和变量以及相关公式

很多应用程序是需要使用一个或多个公式和一些物理常量的，这些信息也是问题说明的一部分。

1.4.2 编写和编译程序

在对问题进行分析后有了清晰的说明，得到一个确定的算法后，就可以开始编写程序了。编写程序时可以根据需要和程序规模选择合适的计算机语言，在此不做赘述。

程序编写好后，就可以进行编译，如果没有任何错误就可以生成目标文件。编译程序往往可以检查出拼写、语法和标点符号等错误，用户可以根据编译结果提示修改程序。在 C 语言中，有的错误消息可能会误导程序员，程序员可以根据错误提示对应程序行上下几行检查是否有误。并且编译程序不能检查出逻辑错误，逻辑错误往往需要通过运行和测试程序来检测。

1.4.3 执行和测试程序

程序通过编译后就可以链接，编译和链接都可以发现很多错误。编译链接成功后生成可执行文件，程序便可执行，此时就是查找逻辑错误的开始。监测逻辑错误的主要方法是使用不同的输入数据测试程序，然后观察输出结果是否正确。输入集的不同可以导致不同的结果，有时程序在一些数据集上得到正确结果，而另一些却得到错误结果，有时一个异常的条件可能导致程序崩溃或死循环，所以在任何情况下都必须验证答案的正确性。

通常测试和验证程序之前，要考虑程序可能的输入集，对所有输入集进行监测。在错误修改中需要调试，定位错误并修改，有时甚至需要重新编写程序。

总而言之，编程是一个“编辑—编译—链接—检测”不断反复的过程，直到没有任何错误为止。

1.5 简单的 C 语言程序及其结构

为了说明 C 程序的结构，在此介绍一个简单程序。

【例 1.1】求两个数的和。

```
/* p1_1.c */
/*以下两个语句称为函数头*/
#include<stdio.h>
void main()
/*以下{}中的语句称为函数体*/
{
    int x,y,s;
    /*函数体以"{"开头，以"}"结尾*/
    /*说明部分，定义了三个整型变量 x,y,z*/
```

```

printf("\n请输入两个整数: ");
scanf("%d%d", &x, &y);           //输入两个整数
s=sum(x,y);                       //调用函数 sum, 并将函数返回值赋给 s
printf("两个整数之和为: %d\n", s); //输出和 s
}

int sum(int x,int y)               //定义 sum 函数, 函数值为整型, x、y 为两整型参数
{
    return x+y;                   //将 x+y 的值返回, 由函数 sum 带回调用处
}

```

运行结果如下。

```

请输入两个整数: 5 6
两个整数之和为: 11

```

程序说明如下。

(1) main 表示“主函数”，C 程序都有一个主函数，并且程序的执行从 main 开始，在 main 中结束。

(2) 适当使用注释可增强 C 程序的可读性。在 VC6.0 中，有两种注释，一种是将注释内容用/*和*/括起，另一种是用“//”引导一行注释，如例 1.1 中所示。注释不参与程序的编译，只是对程序进行解释说明，以便用户更好地理解程序。

(3) 函数体的内容必须由“{”和“}”括起来。

(4) C 程序一般包括一个主函数 main 和多个自定义函数，在这些函数中通常还会调用库函数。典型 C 程序结构如图 1-1 所示。例如例 1.1 程序中有两个函数：一个主函数 main 和一个被调用函数 sum。sum 函数的作用是求两个整数的和，return 语句将和返回给主函数 main。返回值是通过函数名带回到函数调用处的，本题中返回的函数值被赋给了变量 s。

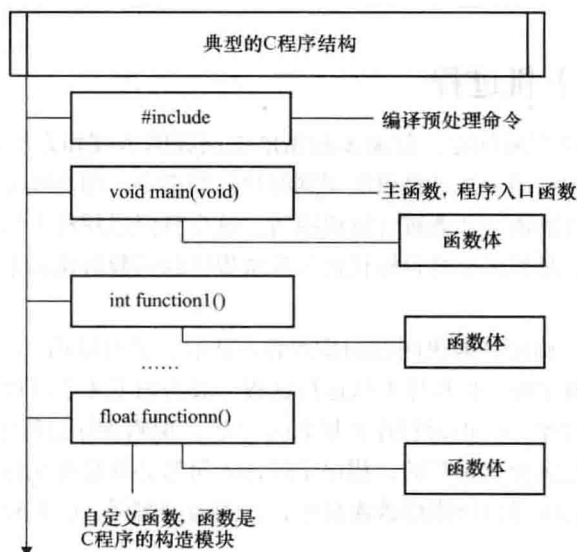


图 1-1 典型的 C 程序结构

(5) main 函数中使用了两个库函数：格式输出函数 printf 和格式输入 scanf。为了程序的简洁性和可移植性，C 不提供输入输出语句，输入输出功能由函数来完成。printf 和 scanf 两个函数

是包含在头文件 `stdio.h` 中的, 所以程序开头部分有一个预处理命令 `"#include<stdio.h>"`。本例中函数 `scanf` 的功能是输入两个整数 `x` 和 `y`, `&x` 和 `&y` 中的 `&` 是取址运算符, `scanf` 的作用就是将两个整数输入到 `&x` 和 `&y` 指定的存储单元中, 即将两个整数输入到变量 `x` 和 `y` 中。

(6) 函数一般包括函数头 (预处理命令和函数名) 和函数体 (如图 1-2 所示), 函数体内开头部分为说明部分, 用于变量和函数的说明, 本例中说明部分只定义了 3 个整型变量。说明部分之后的语句为执行部分, 用于描述函数的功能。

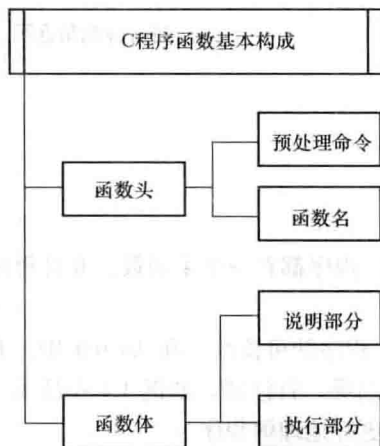


图 1-2 C 程序函数基本构成

1.6 VC6.0 简介

1.6.1 C 程序上机过程

为了利用计算机解决实际问题, 就需要根据解决问题的步骤和方法编写程序。所谓程序就是按照解决问题的要求编写的一组计算机能识别和执行的指令, 用高级语言编写的程序称为源程序。源程序必须翻译成机器语言才能被计算机执行, 这个翻译过程称为编译, 编译的结果就是将源程序翻译成目标代码, 最后还要将目标代码与系统提供的函数链接起来形成可执行文件, 程序才可以被执行。

对于一个实际问题, 确定好解决问题的步骤和方法后, 就可以用 C 语言编写程序, 编写好程序后又如何上机运行程序呢? C 程序上机运行过程一般为如下 4 个过程。

(1) 输入和编辑源程序, C 的源程序扩展名为 `“.c”`。编辑程序过程中, 初学者必须记住: C 语言程序中的英文字母是区分大小写的, 程序中的标点符号必须是英文标点符号。

(2) 编译源程序, 若编译出错则修改源程序, 直到通过编译。C 源程序编译后生成的目标程序的扩展名为 `“.obj”`。

(3) 与库函数链接形成可执行文件, C 程序的可执行文件扩展名为 `“.exe”`。

(4) 运行可执行文件, 分析运行结果。结果正确则完成上机过程, 否则就得检查程序是否有逻辑错误, 分析解决问题的步骤和方法是否正确, 甚至需要重新设计程序。

分析运行结果的过程非常重要,初学者通常认为程序运行有结果,程序就是正确的,其实不然。不同类型的程序结果分析的方法是不同的,此问题有专门的论述,感兴趣的读者可以在学完编程的基本内容后深入学习。

C 程序的上机过程看似简单,实际上是个非常艰辛的过程。程序是否能上机通过不仅依赖程序员本身的知识面,还要求程序员养成良好的编程习惯,并且多上机练习。读者朋友们不妨从现在开始加强上机练习,从简单程序开始逐步加深练习。

1.6.2 在 VC6.0 上运行 C 程序的步骤

程序的集成开发环境就是将编辑器、编译器、链接器和其他软件单元集成在一起,在这种环境里,程序员可以方便地编辑、编译、链接和运行程序,无需多个软件间切换。适合开发 C 程序的集成开发环境非常丰富,常用的有 Microsoft Visual C++ 6.0 (VC6.0), Borland C++, Watcom C++, Borland C++, Borland C++ Builder, TurboC2.0 (TC2.0), TurboC3.0, Win-Tc 等,本书所述内容未做特别说明的都是指在 VC6.0 开发环境下。不同的集成开发环境的程序编辑、编译、链接和运行过程是有差异的,数据存储单元的大小也是略有不同,如 TC2.0 下 int 类型的数据占 2 个字节的存储单元,而在 VC6.0 则占 4 个字节的存储单元。为了便于大家熟悉不同开发环境,我们在本书配套实验教材里简单介绍了 TC2.0 开发环境的使用。下面我们简单介绍在 VC6.0 上运行 C 程序的步骤。

(1) 编辑程序

打开 VC6.0 后,单击“文件”—“新建”打开新建对话框,在对话框中单击“文件”选项卡后选择“c++ Source File”,在图 1-3 所示中输入文件名(如 pl_1.c)后,单击“确定”按钮就可以打开如图 1-4 所示编辑窗口,在编辑窗口中编辑程序。

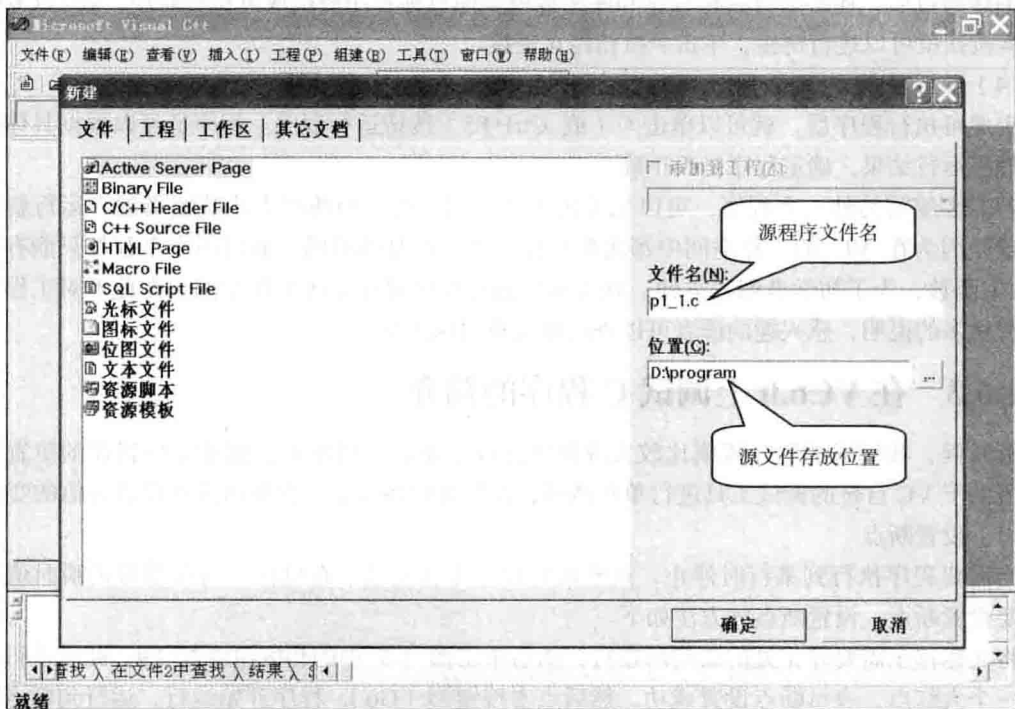


图 1-3 新建一个 C 源程序