



普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

应用型教材系列

C语言 程序设计 (第2版)

谭浩强 著

根据“中国高等院校计算机基础教育课程体系”组织编写



清华大学出版社



普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

应用型教材系列

C语言程序设计

(第2版)

谭浩强 著

清华大学出版社
北京

内 容 简 介

C 语言是国内外广泛使用的计算机语言,我国大多数理工类学校都开设了 C 语言程序设计课程。

谭浩强教授所著的《C 程序设计》一书受到国内专家和读者的一致好评,被公认为学习 C 语言程序设计的经典教材,被全国大多数高校选用,是学习 C 语言的主流用书,已累计发行了 1000 多万册,创同类书的全国发行最高纪录。

本书针对我国应用型大学的实际情况,在《C 程序设计》一书的基础上改写而成,在编写中注意了降低起点,精简内容,突出重点,紧扣基本要求,使读者更加容易学习。

本书定位准确,体系合理,内容适当,概念清晰,例题丰富,深入浅出,文字流畅,通俗易懂。针对初学者的特点,采用了“提出问题—解决问题—归纳分析”的三部曲,符合初学者的认知规律,使初学者容易入门,是初学者学习 C 语言的理想教材。

凡具有计算机初步知识的读者都能读懂本书。本书可作为应用型高等学校各专业学习 C 语言程序设计的教材,基础较好的高职高专也可选用本书。本书也是读者不可多得的用于自学的好教材。

另外,本书还配有辅助教材《C 语言程序设计学习辅导》。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/谭浩强著. —2 版. —北京:清华大学出版社,2008.11

(高等院校计算机应用技术规划教材)

ISBN 978-7-302-18260-3

I. C… II. 谭… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 114004 号

责任编辑:谢 琛 赵晓宁

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:21

版 次:2008 年 11 月第 2 版

印 数:1~8000

定 价:35.00 元

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

字 数:486 千字

印 次:2008 年 11 月第 1 次印刷

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:028708-01

教授计算技术的大师
普及现代科技之巨擘

敬颂谭浩强教授创杰业成就

宋健

一九九五年一月

▲ 原国务委员、国家科委主任、全国政协副主席、
中国工程院院长宋健同志给谭浩强教授的题词

编辑委员会

《高等院校计算机应用技术规划教材》

主 任 谭浩强
副 主 任 焦金生 陈 明 丁桂芝

委 员 王智广 孔令德 刘 星 刘荫铭
安志远 安淑芝 李文英 李叶紫
李 琳 李雁翎 宋 红 陈 强
邵丽萍 尚晓航 张 玲 侯冬梅
郝 玲 赵丰年 秦建中 袁 玫
訾秀玲 薛淑斌 谢树煜 谢 琛

序

《高等院校计算机应用技术规划教材》

进入 21 世纪,计算机成为人类常用的现代工具,每一个有文化的人都应当了解计算机,学会使用计算机来处理各种的事务。

学习计算机知识有两种不同的方法:一种是侧重理论知识的学习,从原理入手,注重理论和概念;另一种是侧重于应用的学习,从实际入手,注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对大多数人来说,计算机是作为一种工具来使用的,应当以应用为目的、以应用为出发点。对于应用性人才来说,显然应当采用后一种学习方法,根据当前和今后的需要,选择学习的内容,围绕应用进行学习。

学习计算机应用知识,并不排斥学习必要的基础理论知识,要处理好这二者的关系。在学习过程中,有两种不同的学习模式:一种是金字塔模型,亦称为建筑模型,强调基础宽厚,先系统学习理论知识,打好基础以后再联系实际应用;另一种是生物模型,植物中的树并不是先长好树根再长树干,长好树干才长树冠,而是树根、树干和树冠同步生长的。对计算机应用性人才教育来说,应该采用生物模型,随着应用的发展,不断学习和扩展有关的理论知识,而不是孤立地、无目的地学习理论知识。

传统的理论课程采用以下的三部曲:提出概念—解释概念—举例说明,这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者,我们提倡新的三部曲:提出问题—解决问题—归纳分析。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

检查学习好坏的标准,不是“知道不知道”,而是“会用不会用”,学习的目的主要在于应用。因此希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力,启发学生的求知欲望。

全国高等院校计算机基础教育研究会历来倡导计算机基础教育必须坚持

面向应用的正确方向,要求构建以应用为中心的课程体系,大力推广新的教学三部曲,这是十分重要的指导思想,这些思想在《中国高等院校计算机基础课程》中作了充分的说明。本丛书完全符合并积极贯彻全国高等院校计算机基础教育研究会的指导思想,按照《中国高等院校计算机基础教育课程体系》组织编写。

这套《高等院校计算机应用技术规划教材》是根据广大应用型本科和高职高专院校的迫切需要而精心组织的,其中包括4个系列:

(1) 基础教材系列。该系列主要涵盖了计算机公共基础课程的教材。

(2) 应用型教材系列。适合作为培养应用性人才的本科院校和基础较好、要求较高的高职高专学校的主干教材。

(3) 实用技术教材系列。针对应用型院校和高职高专院校所需掌握的技能技术编写的教材。

(4) 实训教材系列。应用型本科院校和高职高专院校都可以选用这类实训教材。其特点是侧重实践环节,通过实践(而不是通过理论讲授)去获取知识,掌握应用。这是教学改革的一个重要方面。

本套教材是从1999年开始出版的,根据教学的需要和读者的意见,几年来多次修改完善,选题不断扩展,内容日益丰富,先后出版了60多种教材和参考书,范围包括计算机专业和非计算机专业的教材和参考书;必修课教材、选修课教材和自学参考的教材。不同专业可以从中选择所需要的部分。

为了保证教材的质量,我们遴选了有丰富教学经验的高校优秀教师分别作为本丛书各教材的作者,这些老师长期从事计算机的教学工作,对应用型的教学特点有较多的研究和实践经验。由于指导思想明确、作者水平较高,教材针对性强,质量较高,本丛书问世9年来,愈来愈得到各校师生的欢迎和好评,至今已发行了240多万册,是国内应用型高校的主流教材之一。2006年被教育部评为普通高等教育“十一五”国家级规划教材,向全国推荐。

由于我国的计算机应用技术教育正在蓬勃发展,许多问题有待深入讨论,新的经验也会层出不穷,我们会根据需要不断丰富本丛书的内容,扩充丛书的选题,以满足各校教学的需要。

本丛书肯定会有不足之处,请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长
《高等院校计算机应用技术规划教材》主编

谭浩强

2008年5月1日于北京清华园

前言

程序设计是计算机工作者的一项基本功。C语言是被国内外计算机用户广泛学习和使用的一种计算机语言,受到广大计算机用户的喜爱。我国几乎所有的理工科大学都开设了C语言程序设计课程。

许多学校积累了多年的教学经验,全国各出版社出版的C语言教材琳琅满目,令人目不暇接。当前的任务是研究怎样推进C语言程序设计课程的改革,提高教学质量,使教学取得更好的效果。

为了使C语言程序设计课程教学取得更好的效果,需要解决以下几个问题:

(1) 为什么要学习程序设计? 计算机的本质是“程序的机器”,程序和指令的思想是计算机系统中最基本的概念。学习程序设计能使我们了解计算机是怎样工作的,更好地理解和应用计算机;能使我们懂得怎样让计算机按照人们的意图进行工作;学习使用计算机处理问题的方法,能培养我们分析问题和解决问题的能力以及具有编写程序的初步能力。非计算机专业的学生学习程序设计,可以了解软件的特点和生产过程,能与程序开发人员更好地沟通,有利于共同合作进行应用软件的开发工作。因此,大学生(尤其是理工类学生)学习程序设计是很必要的。

归纳起来,学习程序设计有两个目的:①学习用计算机解决问题的思路和方法,即算法;②了解怎样用计算机去实现算法,即怎样才能实现用计算机解题。

(2) 要从实际出发,准确定位。人们学习C语言程序设计有着不同的目的和作用:有的是计算机专业的,有的是非计算机专业的;有的是本科,有的是高职;有的以后是计算机专业人员,有的是计算机应用人员;有的要求精通,有的要求粗通,有的则只求一般了解。不同的人有不同的要求。因此必须准确定位,从具体实际出发,对不同对象提出不同的要求,采取不同的方案,编写出不同的教材。切不可不问对象一刀切,统一要求,这样反而达不到应有效果。

(3) 大学生学习C语言程序设计,带有基础学习的性质。非计算机专业的学生,其中大多数人将来并不一定会成为专业的软件开发人员,不一定有许多机会用C语言编写实用程序。但是在大学期间学习程序设计仍然是必要

的,这就需要制定恰当的标准。在有限的学时内,只能学习基本的内容,为今后进一步的学习和使用打下必要的基础。如果有少数人将来需要熟练地编程,甚至成为专业的软件开发人员,则应当在学习本课程的基础上继续进行更深入的学习,例如至少应当进行一个综合作业,完成一个具有一定规模的应用软件的实践项目。

(4) 正确处理好语言和算法的关系。如前所述,学习程序设计课程的目的,一是掌握设计程序的思路,二是学会用计算机语言编写程序。应当指出:算法是程序的核心和灵魂,语言是工具。接受一个任务后,首先要考虑解题的思路,即设计算法,然后考虑怎样编写成程序。因此,算法和语法二者都很重要,都要重视。不掌握算法就不可能解决任务,不掌握语法也就谈不上编程,即使编出来,也会错误百出。不要孤立地学习算法,更不能孤立地学习语法。正确的做法是把语言和算法紧密结合起来,从最简单的程序入手,这时的算法和语法都是简单的,引导学生很自然地接触 C 程序设计,使他们学会初步编写简单的 C 程序。然后随着算法的逐步深入,语法也逐步深入,二者很自然地结合在一起。始终把算法与语法紧密捆绑在一起,同步地走向深入。读者可以通过大量的例题,学习到怎样设计一个算法、构造一个程序,并将它付诸实现。

要进行程序设计需要学习以下几方面的知识:①算法,即解题思路;②程序设计语言及其使用环境;③程序设计方法(如结构化程序设计方法、面向对象程序设计方法)。

作者在 1991 年编著了《C 程序设计》一书,由清华大学出版社出版。各校师生普遍认为该书内容系统,讲解详尽,包含了许多其他教材中没有的内容,尤其是针对编程实践中容易出现的问题作了提醒和分析,是学习 C 语言程序设计的理想教材。2001 年,针对应用型大学 and 高职院校的特点,作者在《C 程序设计》一书的基础上编写出版了《C 语言程序设计》一书,适当压缩了内容,降低了起点。以上两本书出版后受到了广大读者的热烈欢迎,至今已累计发行了 1000 多万册。

经过几年的教学实践,作者决定对《C 语言程序设计》一书进行修订,在修订时考虑了以下一些因素:①当前大多数学校是把“C 语言程序设计”作为第一语言来学习的(而在多年前,许多学生是先学了 BASIC 语言后再学 C 语言的),由于学生没有任何程序设计知识的基础,因此起点必须再降低些;②各校情况差别很大,如专业方向、学生基础、教学条件、师资状况都很不相同,在全国范围内不能统一要求,不能用一本教材去满足所有学校的要求,必须有不同内容、不同程度、不同写法的教材,以满足不同类型学校的需求;③由于教学总学时压缩,许多学校分配给“C 语言程序设计”的学时往往并不充分,难以学习更多的内容。

在本次修订中按照以下原则进行修订:

(1) 由于作者所著的《C 程序设计》和《C 语言程序设计》内容全面,叙述

清晰,通俗易懂,在教学实践中取得了很好的效果,因此在修订时尽量保持原有的优点,基本上保留原书的体系,注意概念准确,讲解透彻,使读者对 C 语言和程序设计有全面、完整的理解。

(2) 考虑到不同对象的差别,把教学内容分为基本要求和提高要求两个部分。在每一章中把一部分较深入的内容作为“提高部分”,单独列出,放在各章的最后,供选学。如果学时不够,可以只学基本部分。通过学习基本部分,能初步掌握 C 程序设计的基本内容,写出比较简单的程序。如果希望学习更深入一些,掌握更多的编程思路 and 技巧,可以选学各章的“提高部分”。也可以由教师讲授基本部分,学生自学“提高部分”,以培养学生的自学能力。

(3) 本书较前一版本更加通俗易懂,易于学习,按照作者提出的教学三部曲组织教学,即:“提出问题—解决问题—归纳分析”。实践证明这对于计算机基础教育是十分有效的。在介绍每一个程序时,一般包括以下几个部分:提出问题、解题思路、编写程序、运行结果、程序说明、深入思考等。这可以使读者思路清晰、容易理解。

(4) 在每一章最后增加一节“本章小结”,归纳本章的要点,提纲挈领,画龙点睛,以加深印象,增强条理性。

经教育部批准,该书已被作为“普通高等教育‘十一五’国家级规划教材”,向全国各高校推荐。

为了适应不同读者的需求,作者已编写了 3 种不同程度的 C 语言教材,组成了一个 C 语言的系列教材供选用:

(1) 《C 程序设计(第 3 版)》,内容详尽,讲解透彻,程度较深,适合基础较好、学时较多、要求较高的学校使用。

(2) 《C 程序设计教程》,是《C 程序设计(第 3 版)》的简缩本,在《C 程序设计(第 3 版)》基础上删去一些不常用的部分和与高等数学有关的内容,使读者学习起来更加容易。适合于学时较少的教学。

(3) 本书,内容更加简练,写法上更加通俗易懂,适合于应用型大学非计算机专业使用,基础较好的高职高专也可以使用。

以上 3 种教材已由清华大学出版社正式出版。每种教材都配有学习辅导书。

应当怎样学习 C 语言程序设计呢? 作者给出以下建议:

(1) 在学习开始时不要在语法细节上死记死抠。请记住,重要的是学会编程序,而不是背语法。一开始就要学会看懂程序,编写简单的程序,然后逐步深入。有一些语法细节是需要通过较长期的实践才能熟练地掌握的。初学时,切忌过早地滥用 C 语言的某些容易引起错误的细节(如不适当地使用 ++ 和 -- 的副作用)。

(2) 不能设想今后一辈子只使用在学校里学过的某一种语言。但是无论用哪一种语言进行程序设计,其基本思路和方法都是一样的。从这个意义上说,在大学里学哪一种语言,并不是一个很重要的原则问题。学会了一种计算

机语言,可以很快地学会另一种语言。因此,在学习时一定要学活用活,举一反三,掌握规律,在以后需要时能很快地掌握其他的语言。

(3) 在学校学习阶段,主要是学习程序设计的方法,进行程序设计的基本训练,打下进一步学习的基础。对多数学生来说,不可能通过几十个小时的学习,由一个门外汉变成编程高手,编写出大型而实用的程序。学习程序设计课程时,应该把精力放在最基本、最常用的内容上,学好基本功。如果对学生有较高的程序设计要求,应当在学习本课程后,安排一次集中的课程设计环节,按照实际工作的要求,完成有一定规模的程序设计。

(4) 程序设计是一门实践性很强的课程,既要掌握概念,又要动手编程,还要上机调试运行,希望读者一定要重视实践环节,包括编程和上机。既会编写程序,又会调试程序。衡量这门课学习的好坏,不是看你“知不知道”,而是“会不会干”。考核的方法不能主要用是非题和选择题,而应当把重点放在编写程序和调试程序上。

(5) 使用哪一种编译系统并不是原则问题,重要的是编程能力的培养。程序编好以后,用哪一种编译系统进行编译都可以。读者不应该只会用某一种编译环境,应当了解、接触和使用不同的编译环境。不同的编译系统,其功能和使用方法有些不同,编译时给出的信息也不完全相同,要注意参阅使用说明书,特别要在使用中积累经验,举一反三。

为了帮助读者学习本书,作者还编写了《C语言程序设计(第2版)学习辅导》,提供本书中各章习题的参考答案以及上机实习指导。该书也将随后出版。

本书肯定会有不少缺点和不足,热切期望得到专家和读者的批评指正。

谭浩强

2008年5月1日于清华园

目录

► 第1章 C语言概述	1
1.1 计算机与程序、程序设计语言	1
1.2 C语言的出现和发展过程	2
1.3 简单的C语言程序	2
1.4 运行C程序的步骤与方法	6
1.4.1 运行C程序的步骤	6
1.4.2 上机运行C程序的方法	7
本章小结	14
习题	15
► 第2章 数据的存储与运算	16
2.1 数据在计算机中是怎样存储的	16
2.1.1 数据在计算机中是以二进制形式存储的	16
2.1.2 位、字节和地址	17
2.1.3 不同类型数据的存储方式	18
2.2 整型数据的运算与分析	19
2.2.1 整型数据运算程序举例和分析	19
2.2.2 整型常量与整型变量	22
2.3 实型数据的运算与分析	24
2.3.1 实型数据的运算举例	24
2.3.2 实型常量的表示形式	26
2.3.3 实型变量	27
2.4 字符型数据的运算	29
2.4.1 字符数据运算的简单例子	29
2.4.2 字符常量和字符变量	29
2.4.3 字符串常量	33

2.5	符号常量	34
2.5.1	符号常量的作用	35
2.5.2	符号常量的性质和使用方法	36
2.6	算术运算符和算术表达式	38
2.6.1	算术运算符	38
2.6.2	算术表达式	39
2.7	C 运算符和 C 表达式	41
2.7.1	C 运算符	41
2.7.2	C 表达式	42
2.8	提高部分	42
2.8.1	求补码的方法	42
2.8.2	整型常量的表示形式	43
2.8.3	整型变量的类型	44
2.8.4	整型常量的类型	45
2.8.5	运算符的优先级与结合性	45
	本章小结	46
	习题	46

第 3 章 最简单的 C 程序设计——顺序程序设计 48

3.1	算法是程序的灵魂	48
3.1.1	算法的概念	48
3.1.2	怎样表示算法	50
3.2	程序的三种基本结构	53
3.3	C 语句综述	54
3.4	赋值表达式和赋值语句	56
3.4.1	赋值表达式	56
3.4.2	赋值过程中的类型转换	59
3.4.3	赋值语句	60
3.4.4	变量赋初值	60
3.5	数据输入输出的概念	61
3.6	字符数据的输入输出	63
3.6.1	用 putchar 函数输出一个字符	63
3.6.2	用 getchar 函数输入一个字符	64
3.7	简单的格式输入与输出	66
3.7.1	用简单的 printf 函数输出数据	66
3.7.2	用简单的 scanf 函数输入数据	70
3.8	顺序结构程序设计举例	73

3.9	提高部分	76
3.9.1	无符号数据与有符号数据之间的赋值	76
3.9.2	较复杂的输入输出格式控制	78
3.9.3	简单易用的 C++ 的输入输出	85
3.9.4	C++ 用“//”作为注释的标识	86
	本章小结	86
	习题	87

► 第 4 章 选择结构程序设计 89

4.1	条件判断	89
4.1.1	条件判断的含义	89
4.1.2	关系运算符和关系表达式	90
4.1.3	逻辑运算符和逻辑表达式	91
4.2	用 if 语句实现选择结构	95
4.2.1	用 if 语句实现选择结构举例	95
4.2.2	if 语句的一般形式	97
4.2.3	if 语句的使用说明	97
4.2.4	使用嵌套的 if 语句实现多层判断	99
4.3	利用 switch 语句实现多分支选择结构	103
4.4	程序综合举例	105
4.5	提高部分	109
4.5.1	用条件表达式实现简单的选择结构	109
4.5.2	在程序中使用条件表达式	110
	本章小结	111
	习题	112

► 第 5 章 循环结构程序设计 115

5.1	程序中需要用循环结构	115
5.2	用 while 语句和 do...while 语句实现循环	115
5.2.1	用 while 语句实现循环	115
5.2.2	用 do...while 语句实现循环	117
5.3	用 for 语句实现循环	119
5.3.1	for 语句的一般形式和执行过程	120
5.3.2	for 循环程序举例	120
5.4	循环的嵌套	124
5.5	提前结束循环	124
5.5.1	用 break 语句提前退出循环	124

5.5.2	用 continue 语句提前结束本次循环	126
5.6	几种循环的比较	128
5.7	程序举例	128
5.8	提高部分	132
5.8.1	while 和 do...while 循环的比较	132
5.8.2	for 语句的各种形式	134
	本章小结	137
	习题	137
第 6 章 利用数组处理批量数据		139
6.1	为什么要用数组	139
6.2	定义和引用一维数组	139
6.2.1	定义一维数组	140
6.2.2	引用一维数组的元素	140
6.2.3	一维数组的初始化	141
6.2.4	一维数组程序举例	142
6.3	二维数组的定义和引用	145
6.3.1	定义二维数组	145
6.3.2	引用二维数组的元素	146
6.3.3	二维数组的初始化	147
6.3.4	二维数组程序举例	148
6.4	字符数组	151
6.4.1	定义字符数组及对其初始化	151
6.4.2	引用字符数组	152
6.4.3	字符串和字符串结束标志	153
6.4.4	字符数组的输入输出	155
6.4.5	字符串处理函数	157
6.4.6	字符数组应用举例	157
6.5	提高部分	161
6.5.1	定义二维数组时为何采用两对双括号形式	161
6.5.2	C 语言字符串函数的详细介绍	162
	本章小结	166
	习题	167
第 7 章 用函数实现模块化程序设计		168
7.1	函数是什么	168
7.2	函数的定义和调用	170

7.2.1	为什么要定义函数	170
7.2.2	定义函数	171
7.2.3	函数的调用	172
7.2.4	对被调用函数的声明和函数原型	175
7.3	函数的嵌套调用和递归调用	177
7.3.1	函数的嵌套调用	177
7.3.2	函数的递归调用	179
7.4	数组作为函数参数	184
7.4.1	数组元素作函数实参	185
7.4.2	数组名作函数参数	186
7.5	变量的作用域和生存期	192
7.5.1	变量的作用域——局部变量和全局变量	192
7.5.2	变量的存储方式和生存期	195
7.5.3	作用域和生存期的小结	199
7.6	内部函数和外部函数	200
7.6.1	内部函数	200
7.6.2	外部函数	201
7.7	提高部分	203
7.7.1	实参求值的顺序	203
7.7.2	递归的典型例子——Hanoi(汉诺)塔问题	203
	本章小结	206
	习题	208

► 第8章 善于使用指针 210

8.1	什么是指针	210
8.2	指针变量	211
8.2.1	使用指针变量访问变量的例子	211
8.2.2	怎样定义指针变量	212
8.2.3	怎样引用指针变量	214
8.2.4	指针变量作为函数参数	215
8.3	通过指针引用数组	220
8.3.1	数组元素的指针	220
8.3.2	通过指针引用数组元素	221
8.3.3	用数组名作函数参数	224
8.4	通过指针引用字符串	229
8.4.1	字符串的表示形式	229
8.4.2	字符指针作函数参数	233

8.4.3	对使用字符指针变量和字符数组的讨论	236
8.5	提高部分	239
8.5.1	指针使用的技巧	239
8.5.2	多维数组的指针	240
8.5.3	指向函数的指针	241
8.5.4	返回指针值的函数	242
8.5.5	指针数组	242
8.5.6	多重指针——指向指针的指针	243
	本章小结	244
	习题	247

► 第9章 使用结构体类型处理组合数据——用户自定义

	数据类型	248
9.1	定义和使用结构体变量	248
9.1.1	自己建立结构体类型	248
9.1.2	定义结构体类型变量	250
9.1.3	结构体变量的初始化和引用	252
9.2	结构体数组	255
9.3	结构体指针	258
9.4	用结构体变量和结构体变量的指针作函数参数	262
9.5	用指针处理链表	265
9.5.1	什么是线性链表	265
9.5.2	建立简单的静态链表	266
9.5.3	建立动态链表	267
9.6	提高部分	269
9.6.1	共用体类型	269
9.6.2	枚举类型	270
	本章小结	272
	习题	273

► 第10章 利用文件保存数据

10.1	C文件的有关概念	275
10.1.1	什么是文件	275
10.1.2	文件名	276
10.1.3	文件的分类	276