

深入
浅出 系列规划教材
云南省精品教材

C语言程序设计

(第2版)

李俊莉 张晴晖 强振平 主编
Li Junli Zhang Qinghui Qiang Zhenping

清华大学出版社

 **深入浅出** 系列规划教材

深入浅出

C语言程序设计 (第2版)

李俊萩 张晴晖 强振平 主 编
徐伟恒 陈 旭 钟丽辉 副主编

清华大学出版社
北 京

内 容 简 介

作者在多年从事计算机程序设计课程教学的基础上,根据实际教学经验,精心组织编写了本教材。书中通过大量实例,深入浅出地介绍了C语言的基础知识,以及用C语言解决实际问题的程序设计与技巧;对于初学者常见错误进行重点剖析,引入计算思维教学方法,例题解析体现提出问题、分析问题、解决问题的思维模式;大量采用比较式教学法,对初学者易混知识点及重点、难点进行分析,帮助初学者快速掌握C语言的语法知识及编程技巧。书中所有实例都在Code::Blocks环境下验证通过并有运行结果的截图。

本教材以提高编程能力为主线,循序渐进,知识结构合理,具有一定的深度,针对大学教学要求进行编写,涵盖了全国计算机等级考试二级C语言的全部知识点。本教材十分适合高等院校本科、专科、成人教育、函授、高职高专计算机及相关专业教学使用,也可作为C语言各类认证考试的参考书,还可供计算机工程技术人员参考。

本教材为教师配有电子课件以及全部例题的源代码,可从清华大学出版社网站 <http://www.tup.com.cn> 下载。

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

深入浅出C语言程序设计/李俊萩,张晴晖,强振平主编. —2版. —北京:清华大学出版社,2015
深入浅出系列规划教材

ISBN 978-7-302-39179-1

I. ①深… II. ①李… ②张… ③强… 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(2015)第017699号

责任编辑:白立军

封面设计:傅瑞学

责任校对:时翠兰

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:24.5

字 数:565千字

版 次:2010年8月第1版 2015年3月第2版

印 次:2015年3月第1次印刷

印 数:1~2000

定 价:45.00元

产品编号:060029-01



为什么开发深入浅出系列丛书？

目的是从读者角度写书，开发出高质量的、适合阅读的图书。

“不积跬步，无以至千里；不积小流，无以成江海。”知识的学习是一个逐渐积累的过程，只有坚持系统地学习知识，深入浅出，坚持不懈，持之以恒，才能把一类技术学习好。坚持的动力源于所学内容的趣味性和讲法的新颖性。

计算机课程的学习也有一条隐含的主线，那就是“提出问题→分析问题→建立数学模型→建立计算模型→通过各种平台和工具得到最终正确的结果”，培养计算机专业学生的核心能力是“面向问题求解的能力”。由于目前大学计算机本科生培养计划的特点，以及受教学计划和课程设置的原因，计算机科学与技术专业的本科生很难精通掌握一门程序设计语言或者相关课程。各门课程设置比较孤立，培养的学生综合运用各方面的知识能力方面有欠缺。传统的教学模式以传授知识为主要目的，能力培养没有得到充分的重视。很多教材受教学模式的影响，在编写过程中，偏重概念讲解比较多，而忽略了能力培养。为了突出内容的案例性、解惑性、可读性、自学性，本套书努力在以下方面做好工作。

1. 案例性

所举案例突出与本课程的关系，并且能恰当反映当前知识点。例如，在计算机专业中，很多高校都开设了高等数学、线性代数、概率论，不言而喻，这些课程对于计算机专业的学生来说是非常重要的，但就目前对不少高校而言，这些课程都是由数学系的老师讲授，教材也是由数学系的老师编写，由于学科背景不同和看待问题的角度不同，在这些教材中基本都是纯数学方面的案例，作为计算机系的学生来说，学习这样的教材缺少源动力并且比较乏味，究其原因，很多学生不清楚这些课程与计算机专业的关系是什么。基于此，在编写这方面的教材时，可以把计算机上的案例加入其中，例如，可以把计算机图形学中的三维空间物体图像在屏幕上的伸缩变换、平移变换和旋转变换在矩阵运算中进行举例；可以把双机热备份的案例融入到马尔科夫链的讲解；把密码学的案例融入到大数分解中等。

2. 解惑性

很多教材中的知识讲解注重定义的介绍，而忽略因果性、解释性介绍，往往造成知其然而不知其所以然。下面列举两个例子。

(1) 读者可能对 OSI 参考模型与 TCP/IP 参考模型的概念产生混淆，因为两种模型之

间有很多相似之处。其实,OSI 参考模型是在其协议开发之前设计出来的,也就是说,它不是针对某个协议族设计的,因而更具有通用性。而 TCP/IP 模型是在 TCP/IP 协议栈出现后出现的,也就是说,TCP/IP 模型是针对 TCP/IP 协议栈的,并且与 TCP/IP 协议栈非常吻合。但是必须注意,TCP/IP 模型描述其他协议栈并不合适,因为它具有很强的针对性。说到这里读者可能更迷惑了,既然 OSI 参考模型没有在数据通信中占有主导地位,那为什么还花费这么大的篇幅来描述它呢?其实,虽然 OSI 参考模型在协议实现方面存在很多不足,但是,OSI 参考模型在计算机网络的发展过程中起到了非常重要的作用,并且,它对未来计算机网络的标准化、规范化的发展有很重要的指导意义。

(2) 再例如,在介绍原码、反码和补码时,往往只给出其定义和举例表示,而对最后为什么在计算机中采取补码表示数值?浮点数在计算机中是如何表示的?字节类型、短整型、整型、长整型、浮点数的范围是如何确定的?下面我们来回答这些问题(以 8 位数为例),原码不能直接运算,并且 0 的原码有+0 和-0 两种形式,即 00000000 和 10000000,这样肯定是不行的,如果根据原码计算设计相应的门电路,由于要判断符号位,设计的复杂度会大大增加,不合算;为了解决原码不能直接运算的缺点,人们提出了反码的概念,但是 0 的反码还是有+0 和-0 两种形式,即 00000000 和 11111111,这样是不行的,因为计算机在计算过程中,不能判断遇到 0 是+0 还是-0;而补码解决了 0 表示的唯一性问题,即不会存在+0 和-0,因为+0 是 00000000,它的补码是 00000000,-0 是 10000000,它的反码是 11111111,再加 1 就得到其补码是 10000000,舍去溢出量就是 00000000。知道了计算机中数用补码表示和 0 的唯一性问题后,就可以确定数据类型表示的取值范围了,仍以字节类型为例,一个字节共 8 位,有 00000000~11111111 共 256 种结果,由于 1 位表示符号位,7 位表示数据位,正数的补码好说,其范围从 00000000~01111111,即 0~127;负数的补码为 10000000~11111111,其中,11111111 为-1 的补码,10000001 为-127 的补码,那么到底 10000000 表示什么最合适呢?8 位二进制数中,最小数的补码形式为 10000000;它的数值绝对值应该是各位取反再加 1,即为 $01111111+1=10000000=128$,又因为为负数,所以是-128,即其取值范围是-128~127。

3. 可读性

图书的内容要深入浅出,使人爱看、易懂。一本书要做到可读性好,必须做到“善用比喻,实例为王”。什么是深入浅出?就是把复杂的事物简单地描述明白。把简单事情复杂化的是哲学家,而把复杂的问题简单化的是科学家。编写教材时要以科学家的眼光去编写,把难懂的定义,要通过图形或者举例进行解释,这样能达到事半功倍的效果。例如,在数据库中,第一范式、第二范式、第三范式、BC 范式的概念非常抽象,很难理解,但是,如果以一个教务系统中的学生表、课程表、教师表之间的关系为例进行讲解,从而引出范式的概念,学生会比较容易接受。再例如,在生物学中,如果纯粹地讲解各个器官的功能会比较乏味,但是如果提出一个问题,如人的体温为什么是 37°C ?以此为引子引出各个器官的功能效果要好得多。再例如,在讲解数据结构课程时,由于定义多,表示抽象,这样达不到很好的教学效果,可以考虑在讲解数据结构及其操作时用程序给予实现,让学生看到直接的操作结果,如压栈和出栈操作,可以把 PUSH()和 POP()操作实现,这样效果会好

很多,并且会激发学生的学习兴趣。

4. 自学性

一本书如果适合自学学习,对其语言要求比较高。写作风格不能枯燥无味,让人看一眼就拒人千里之外,而应该是风趣、幽默,重要知识点多举实际应用的案例,说明它们在实际生活中的应用,应该有画龙点睛的说明和知识背景介绍,对其应用需要注意哪些问题等都要有提示等。

一书在手,从第一页开始的起点到最后一页的终点,如何使读者能快乐地阅读下去并获得知识?这是非常重要的问题。在数学上,两点之间的最短距离是直线。但在知识的传播中,使读者感到“阻力最小”的书才是好书。如同自然界中没有直流的河流一样,河水在重力的作用下一定沿着阻力最小的路径向前进。知识的传播与此相同,最有效的传播方式是传播起来损耗最小,阅读起来没有阻力。

是为序。

欢迎老师投稿: bailj@tup.tsinghua.edu.cn。

2014年12月15日



“C 语言程序设计”是计算机专业及各理工科专业一门重要的基础课程。C 语言既有高级语言的强大功能,又有很多直接操作计算机硬件的功能,因此,C 语言通常称为中级语言。学习和掌握 C 语言,既可增进对于计算机底层工作机制的了解,又可为进一步学习其他高级语言打下坚实基础。

本课程的教学目的是引导初学者掌握 C 语言的基本语法,锻炼实际动手能力和逻辑思维能力,并能将理论知识应用于解决各种实际问题。作者根据多年的教学经验,在分析国内外多种同类教材的基础上,编写了本书。

本书内容深入浅出、循序渐进,采用大量实例帮助初学者快速掌握 C 语言的语法知识,书中所有实例都在 Code::Blocks 环境下验证通过并有运行结果的截图。例题的选择基于以下几点考虑。

(1) 例题分为基础型和应用型两种。基础型例题主要演示 C 语言基本语法的使用格式及应用场合;应用型例题融会贯通了前后知识点,帮助初学者理解并掌握解决各类实际问题的编程方法。

(2) 为从多角度扩展读者的思路,部分有代表性的例题在解决同一问题时给出了多种编程方法。

(3) 一些例题的讲解采用了“比较式教学法”,既给出正确的程序,又给出错误的程序,引导读者深入分析问题,学会判断错误。

(4) 例题的讲解引入计算思维的教学理念,体现提出问题、分析问题、解决问题的思维方式,首先给出编程思路,然后给出源代码,并对源代码中的重点、难点进行剖析,最后提出思考问题,激发学生的创新思考。

本书与国内同类教材最大的一个不同点是,对初学者容易出现的各类常见错误进行重点分析。很多同类教材只讲对的,不讲错的,对于一个编程者来说,如果不能了解和认识错误,就不可能真正懂得编程。因此,本书采用各种对比手法及大量表格为初学者剖析容易混淆的各种基本概念。实践表明,人们从错误中学到的东西往往比从正确中学到的东西多得多,懂得错误并能识别错误才能真正掌握程序设计的方法。

本书共由 3 部分组成,分为 14 章,内容编排由浅入深。第一部分介绍基础知识,包括第 1~3 章;第二部分介绍程序设计的 3 种基本结构,包括第 4~6 章;第三部分是程序设计的提高与进阶,包括第 7~14 章。

第一部分介绍基础知识,其中第 1 章介绍 C 语言的发展、程序设计的步骤及原理、C 语言开发环境等内容;第 2 章主要介绍结构化程序设计的方法,通过流程图对初学者建立程

程序设计的基本框架;第 3 章介绍 C 语言的数据类型、运算符及表达式,为后面的程序设计奠定基础。通过这一部分的学习,读者应认识程序设计的一般步骤,能够书写简单表达式语句,能够用流程图对简单算法的结构化程序进行描述。

第二部分介绍程序设计的 3 种基本结构(顺序结构、选择结构、循环结构)。其中第 4 章是顺序结构程序设计,重点介绍程序设计中使用的最多的输入输出函数;第 5 章是选择结构程序设计,介绍构成选择结构的语句及运算符;第 6 章是循环结构程序设计,介绍构成循环结构的 3 种语句,以及用在循环结构中使用的非结构化语句。通过这部分的学习,读者应掌握结构化程序设计的方法,学会设计简单的算法,并依据算法编写程序解决问题。

第三部分介绍多种程序设计的方法。其中第 7 章介绍一维数组、二维数组的知识;第 8 章和第 10 章介绍指针相关的知识,由于指针是 C 语言的精华及难点,因此将指针的基础与提高分两章进行介绍;第 9 章是函数,函数是 C 程序的基本组成单位,体现了模块化设计的思想;第 11 章是字符串,主要介绍数组与指针操作字符串的方法,以及常用字符串处理函数;第 12 章是结构体与共用体,介绍构造类型用于处理大量复杂类型数据的方法;第 13 章是文件,介绍 C 语言对文件的操作,重点是文件常用函数的使用方法;第 14 章是位运算,本章体现了 C 语言不同于其他高级语言,能够对计算机硬件直接操作的特点。通过这些章节的学习,读者应掌握程序的模块化设计方法及更多的编程手段,采用计算思维的方法分析问题,灵活运用数组、指针、结构体、链表等手段解决问题,培养创新思维和解决问题的能力。

本书特色如下。

(1) 对初学者各类常见错误进行重点剖析,帮助初学者快速掌握 C 语言语法知识。

对初学者来说,学习编程的第一步是掌握程序设计语言的语法知识,但是学习过程中往往会出现各种各样的语法错误,使得程序编写难以达到预期结果,很多初学者常常纠结于各种语法错误,从而丧失对程序设计语言学习的兴趣与信心。作者根据多年的实际教学经验,总结了初学者常见的各类语法错误和逻辑错误,在教材中采用大量表格及各种比较手段剖析这些错误,以帮助初学者能够识别并避免各类错误,尽快迈过语法关。

(2) 引入计算思维的教学方式,例题讲解遵循提出问题、分析问题、解决问题的思维模式。

本课程的教学目的是培养学生分析问题和解决问题的能力,最终能够将 C 语言应用于解决各类实际问题。为了达到这一目的,例题的分析与讲解是关键。书中很多例题的代码配有流程图,以帮助初学者更好地建立编程思维。例题的讲解通常是首先给出编程思路,然后给出源代码,并对源代码中的重点和难点问题进行分析,最后进行总结,并提出思考问题,引导读者深入理解用 C 语言编写程序解决实际问题的方法和步骤,提升读者的学习兴趣。

(3) 本书主要知识点的讲解贯穿了“比较教学模式”,主要体现在以下几个方面。

① 对容易混淆的基本概念进行比较,例如用表格对初学者最易混淆的运算符 `==` 和 `=` 进行比较,以帮助初学者正确掌握这两个运算符的使用方法。

② 对同一问题的解决给出多种方法,帮助读者融会贯通所学知识、扩展思路、从中体会哪种方法解决问题更为便捷。例如,用指针引用数组元素既可以用下标法,也可以用指

针法,在一个程序中同时采用多种方法,以帮助读者深刻理解各种方法的特点。

③ 对一个复杂问题的讲解采用由易到难多步骤进行,通过对每个步骤源代码的比较,读者便在潜移默化中掌握了复杂问题的编程方法。例如,在讲解循环嵌套的知识点时,按照由易到难的步骤,依次讲解输出矩形、直角三角形、正三角形的方法,通过比较输出这几种图形的源代码,读者可以逐步掌握循环嵌套的编程方法。

本书可以作为普通高等院校计算机专业及理工类专业的本科教材,也可作为各类C语言认证考试的复习参考书,以及作为计算机工程技术人员的参考书。

本书由李俊菽、张晴晖、强振平任主编,徐伟恒、陈旭、钟丽辉任副主编。

本书第4章、第9章、第10章由李俊菽编写,第7章、第11章、第14章由张晴晖编写,第6章和第13章由徐伟恒编写,第12章由陈旭编写,第3章和第5章由钟丽辉编写。

由于作者水平有限,书中难免有不当之处,恳请广大读者批评指正。

如果需要书中的程序源代码或教学课件,可与作者联系:li_junqiu@sohu.com。

作者

2015年1月

目 录



第 1 章	程序设计入门	1
1.1	计算机与程序设计语言	1
1.1.1	计算机的基本原理简介	1
1.1.2	计算机语言	2
1.2	C 语言的历史及优缺点	4
1.2.1	C 语言的发展历史	4
1.2.2	C 语言的优缺点	5
1.3	C 语言程序设计的工作原理	6
1.3.1	C 语言程序的编译运行过程	6
1.3.2	简单的 C 语言例子	7
1.4	C 语言开发环境介绍	9
1.4.1	Code::Blocks	10
1.4.2	Microsoft Visual C++ 6.0	16
1.4.3	GCC	19
1.4.4	程序调试实例	20
1.5	本章小结	26
1.6	习题	27
1.6.1	选择题	27
1.6.2	填空题	28
1.6.3	编程题	28
第 2 章	程序结构描述	29
2.1	算法与程序	29
2.1.1	算法	29
2.1.2	程序	30
2.1.3	常用开发语言简介	31
2.2	结构化程序设计方法简介	33
2.3	结构化程序的描述	33
2.4	简单程序分析	36

2.5	本章小结	38
2.6	习题	38
第 3 章	C 语言预备知识	39
3.1	C 语言的标识符	39
3.2	C 语言的数据类型	40
3.2.1	数据类型	40
3.2.2	数据类型的作用	40
3.3	常量	43
3.3.1	整型常量	43
3.3.2	实型常量	43
3.3.3	字符常量	44
3.3.4	字符串常量	45
3.3.5	符号常量	45
3.4	变量	46
3.5	运算符和表达式	49
3.5.1	运算符和表达式简介	49
3.5.2	算术运算符和算术表达式	52
3.5.3	赋值运算符和赋值表达式	53
3.5.4	自增、自减运算符和表达式	54
3.5.5	求字节运算符 sizeof	55
3.5.6	逗号运算符和逗号表达式	55
3.6	本章小结	56
3.7	习题	57
3.7.1	选择题	57
3.7.2	填空题	59
3.7.3	程序改错题	60
第 4 章	顺序结构程序设计	61
4.1	C 语言的语句分类	61
4.1.1	表达式语句	61
4.1.2	函数调用语句	62
4.1.3	空语句	62
4.1.4	复合语句	62
4.1.5	流程控制语句	63
4.2	格式化屏幕输出函数 printf()	63
4.2.1	printf()函数的格式	63
4.2.2	printf()函数应用举例	67

4.2.3	printf()函数常见错误举例	69
4.3	格式化键盘输入函数 scanf()	70
4.3.1	scanf()函数的格式	70
4.3.2	scanf()函数应用举例	72
4.3.3	scanf()函数常见错误举例	74
4.4	单个字符输入输出函数	76
4.4.1	单个字符输出函数 putchar()	76
4.4.2	单个字符输入函数 getchar()	76
4.5	getche()函数和 getch()函数	77
4.6	顺序结构应用实例	79
4.7	本章小结	81
4.8	习题	83
4.8.1	选择题	83
4.8.2	填空题	85
4.8.3	编程题	86
第5章	选择结构程序设计	87
5.1	关系运算符及其表达式	87
5.2	逻辑运算符及其表达式	89
5.3	条件运算符及其表达式	91
5.4	if 语句	92
5.4.1	单分支 if 语句	92
5.4.2	双分支 if 语句	95
5.4.3	多分支 if 语句	97
5.4.4	if 语句的嵌套结构	99
5.5	switch 语句	101
5.6	选择结构应用实例	105
5.7	本章小结	111
5.8	习题	113
5.8.1	选择题	113
5.8.2	编程题	116
第6章	循环结构程序设计	117
6.1	while 语句	117
6.2	do...while 语句	120
6.3	for 语句	122
6.3.1	for 语句的一般形式	122
6.3.2	for 语句缺省表达式的形式	124

6.3.3	比较 3 种循环语句·····	126
6.4	循环的嵌套·····	127
6.5	break 语句·····	130
6.6	continue 语句·····	132
6.7	goto 语句·····	134
6.8	循环结构应用实例·····	135
6.9	本章小结·····	140
6.10	习题·····	141
6.10.1	选择题·····	141
6.10.2	填空题·····	142
6.10.3	编程题·····	144
第 7 章	数组·····	145
7.1	一维数组·····	145
7.1.1	一维数组的定义·····	145
7.1.2	一维数组元素的引用·····	146
7.1.3	一维数组的初始化·····	150
7.1.4	一维数组编程举例·····	151
7.2	二维数组·····	154
7.2.1	二维数组的定义·····	154
7.2.2	二维数组元素的引用·····	154
7.2.3	二维数组的初始化·····	155
7.2.4	二维数组编程举例·····	156
7.3	数组应用实例·····	160
7.4	本章小结·····	166
7.5	习题·····	166
7.5.1	选择题·····	166
7.5.2	填空题·····	168
7.5.3	编程题·····	169
第 8 章	指针基础·····	170
8.1	指针的概念·····	170
8.2	指针变量·····	171
8.2.1	指针变量的定义·····	171
8.2.2	指针变量的赋值·····	172
8.2.3	指针变量的间接引用·····	173
8.3	指针变量的基础类型·····	176
8.4	指针的运算·····	178

8.4.1	指针的算术运算	178
8.4.2	指针相减	180
8.4.3	指针比较	180
8.5	指针与一维数组	181
8.6	指针应用实例	182
8.7	本章小结	186
8.8	习题	188
8.8.1	选择题	188
8.8.2	填空题	189
8.8.3	编程题	190
第9章	函数	191
9.1	函数基础	191
9.1.1	函数定义	193
9.1.2	函数调用	196
9.1.3	函数声明	201
9.2	参数的传值与传地址方式	204
9.2.1	参数的传值方式	204
9.2.2	参数的传地址方式	206
9.2.3	参数传值方式与传地址方式的比较	207
9.2.4	一维数组与函数	209
9.3	变量的作用范围与存储类别	214
9.3.1	变量的作用范围	214
9.3.2	变量的存储类别	216
9.4	函数的递归调用	220
9.5	编译预处理命令	222
9.5.1	文件包含	223
9.5.2	宏定义	223
9.5.3	条件编译命令 #ifdef 和 #ifndef	226
9.6	函数应用实例	227
9.7	本章小结	231
9.8	习题	232
9.8.1	选择题	232
9.8.2	填空题	235
9.8.3	编程题	236
第10章	指针提高篇	237
10.1	二级指针	237

10.2	指针与二维数组	238
10.2.1	二维数组中指针的概念	238
10.2.2	行指针与二维数组	239
10.2.3	指针数组与二维数组	240
10.3	指针的动态存储分配	242
10.3.1	动态存储分配与释放	242
10.3.2	一维动态数组	245
10.3.3	二维动态数组	246
10.4	函数指针	247
10.5	main 函数的参数	250
10.6	本章小结	251
10.7	习题	253
10.7.1	选择题	253
10.7.2	编程题	256
第 11 章	字符串	257
11.1	字符串的概念	257
11.2	字符数组与字符串	258
11.2.1	为字符数组初始化字符串	258
11.2.2	利用字符数组输入、输出字符串	259
11.2.3	基于字符数组的字符串编程举例	261
11.3	字符指针与字符串	262
11.3.1	字符指针指向字符串的方法	262
11.3.2	比较字符指针与字符数组	263
11.3.3	基于字符指针的字符串编程举例	264
11.4	字符串处理函数	266
11.4.1	求字符串长度函数 strlen()	266
11.4.2	字符串复制函数 strcpy()	268
11.4.3	字符串连接函数 strcat()	270
11.4.4	字符串比较函数 strcmp()	271
11.5	字符串数组	272
11.5.1	利用字符型二维数组构造字符串数组	272
11.5.2	利用字符型指针数组构造字符串数组	273
11.5.3	比较二维数组和指针数组构造字符串数组	274
11.6	字符串应用实例	275
11.7	本章小结	279
11.8	习题	279
11.8.1	选择题	279

11.8.2	填空题	281
11.8.3	编程题	283
第 12 章	结构体和共用体	284
12.1	声明结构体类型	284
12.1.1	声明结构体类型的一般方法	284
12.1.2	使用 typedef 命名结构体类型	285
12.2	定义结构体变量和结构体指针	286
12.2.1	定义结构体变量	286
12.2.2	定义结构体指针	288
12.3	引用结构体成员	289
12.3.1	结构体成员的引用	289
12.3.2	结构体成员的初始化	291
12.4	结构体数组	292
12.4.1	定义结构体数组	292
12.4.2	结构体数组的初始化	292
12.4.3	结构体数组与指针	293
12.4.4	结构体数组应用举例	294
12.5	结构体与函数	295
12.6	链表	300
12.6.1	链表概述	300
12.6.2	链表的建立和输出	301
12.6.3	链表的插入	305
12.6.4	链表的删除	306
12.7	共用体	307
12.7.1	声明共用体类型	307
12.7.2	定义共用体变量	308
12.7.3	引用共用体成员	310
12.7.4	共用体类型数据的特点	311
12.8	枚举类型	312
12.9	本章小结	313
12.10	习题	314
12.10.1	选择题	314
12.10.2	填空题	315
12.10.3	编程题	317
第 13 章	文件	318
13.1	C 语言文件系统概述	318

13.1.1	C 文件概述	318
13.1.2	缓冲文件系统	319
13.1.3	非缓冲文件系统	319
13.2	文件类型指针	320
13.3	文件的打开与关闭	321
13.3.1	文件打开函数 <code>fopen()</code>	321
13.3.2	文件关闭函数 <code>fclose()</code>	323
13.4	文件的读写	324
13.4.1	格式化读写函数 <code>fscanf()</code> 和 <code>fprintf()</code>	324
13.4.2	字符读写函数 <code>fgetc()</code> 和 <code>fputc()</code>	327
13.4.3	字符串读写函数 <code>fgets()</code> 和 <code>fputs()</code>	329
13.4.4	数据块读写函数 <code>fread()</code> 和 <code>fwrite()</code>	331
13.4.5	判断文件结束函数 <code>feof()</code>	333
13.5	文件的定位	333
13.5.1	定位函数 <code>fseek()</code>	334
13.5.2	获取位置函数 <code>ftell()</code>	334
13.5.3	反绕函数 <code>rewind()</code>	335
13.6	出错检测	336
13.6.1	<code>ferror()</code> 函数	336
13.6.2	<code>clearerr()</code> 函数	337
13.7	文件应用实例	337
13.8	本章小结	342
13.9	习题	343
13.9.1	选择题	343
13.9.2	编程题	345
第 14 章	位运算	346
14.1	位运算符	346
14.1.1	按位与运算符	347
14.1.2	按位或运算符	348
14.1.3	按位异或运算符	349
14.1.4	按位取反运算符	351
14.1.5	按位左移运算符	351
14.1.6	按位右移运算符	352
14.2	位运算应用实例	353
14.3	本章小结	354
14.4	习题	355
14.4.1	选择题	355