



励志改变人生  
编程改变命运

# 零基础学 C语言

第3版

11.5小时多媒体教学视频

康莉 李宽 等编著

## 本书特色

- ◎ 由浅入深，循序渐进，从零开始学C语言，一点都不难
- ◎ 编程基础、编程进阶、编程应用、项目实战、上机练习、面试指南
- ◎ 230个实例、114个练习题、24个面试题

## 超值、大容量DVD

- ◎ 本书教学视频
- ◎ 本书源代码
- ◎ 本书教学PPT
- ◎ 本书习题答案

## 本书技术支持

- ◎ 论坛：<http://www.rzchina.net>

# 零基础学 C语言



第3版

11.5小时多媒体教学视频

康莉 李宽 等编著



机械工业出版社  
China Machine Press



DVD-ROM

## 图书在版编目 ( CIP ) 数据

零基础学 C 语言 / 康莉, 李宽等编著. —3 版. —北京: 机械工业出版社, 2014.5  
(零基础学编程)

ISBN 978-7-111-46108-1

I. 零… II. ①康… ②李… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2014) 第 047287 号

本书站在零基础学习的角度讲授 C 语言, 使初学者能尽快掌握 C 语言程序设计的精髓, 避免走弯路。在讲解知识点时, 笔者采用由浅入深、逐级递进的学习方式进行内容设置安排。本书一共分为 4 篇, 循序渐进地讲述了 C 语言的语法规则和编程思想, 从基本概念到具体实践、从入门知识到高阶主题、从语法语义到数据结构和算法都进行了详细的阐述。主要内容包括数据的存储和获取、屏幕的输入与输出、运算符、表达式、分支语句、循环语句、函数、数组、指针、字符串处理、结构体、共用体、枚举、位运算、文件处理、作用域、预处理、数据结构等。最后一章通过对一些常见的 C 语言面试题的解析, 为读者参加求职考试提供参考资料。

本书非常适合无 C 语言基础或基础薄弱的程序员阅读, 并可作为开发人员的参考手册。

## 零基础学C语言

康莉 等编著

出版发行: 机械工业出版社 (北京市西城区百万庄大街22号 邮政编码: 100037)

责任编辑: 陈佳媛

印刷: 北京瑞德印刷有限公司

开本: 185mm×260mm 1/16

书号: ISBN 978-7-111-46108-1

ISBN 978-7-89405-409-8 (光盘)

责任校对: 董纪丽

版次: 2014年7月第3版第1次印刷

印张: 28

定价: 69.00元 (附光盘)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东



# 前 言

---

C 语言自 1972 年于贝尔实验室诞生以来，一直以其灵活和实用的特性得到了广大用户的喜爱，迅速发展成一种应用广泛的高级语言。不论是网站后台还是底层操作系统，也不论是多媒体应用还是大型网络游戏，均可使用 C 语言来开发。在工业领域，C 语言也是首选的系统语言。各种操作系统，如 UNIX、Linux 和 Windows 等的内核都是采用 C 语言和汇编语言来编写的。

创新推动着软件开发不断进步，在 C 语言之后，各种新的语言相继诞生，如 C++、Java、C# 等，但 C 语言的基础地位依然不可撼动。学好了 C 语言再去看上面几种语言，会发现其中的机理是相通的，所谓万变不离其宗，改变的只是语法的形式，编程思想却没有变化。而且，很多语言的编译器或者解释器就是用 C 语言编写出来的，比如风靡全球的 PHP、Ruby 等。

所以，C 语言是程序开发的基石。希望本书能像一盏明灯，照亮读者学习 C 语言之路。

## 本书特色

本书系统全面地介绍了 C 语言各个方面的知识，从最简单的“Hello World”程序写起，逐步深化、细化。书中对每个知识和技术要点都给出了翔实的示例及代码分析。和其他书籍中罗列代码的做法不同，本书中的代码力求短小精悍，直击要点，避免了细枝末节对读者思维的干扰。在讲解知识点的同时辅以笔者多年的 C 语言编程经验解析，可加深读者的理解。

本书的特点主要体现在以下几个方面：

- 编排采用密切结合、循序渐进的方式，每章主题鲜明，要点突出，适合初中级读者逐步掌握 C 语言的语法规则和编程思想。
- 示例丰富，关键知识点都辅以示例帮助读者理解。示例程序简洁，但并不是简单的代码罗列，而是采用短小精炼的代码紧扣所讲的技术细节，并配以详细的代码解释和说明，使读者印象深刻，对所学知识理解得更加透彻。
- 示例可移植性强，与编译环境和平台无关，读者可轻易地将代码复制到自己的机器上进行实验，自行实践和演练，直观体会所讲要点，感受 C 语言的无限魅力。本书的所有示例、源代码都附在随书光盘中，方便读者使用。
- 结构清晰，内容全面，几乎涉及了 C 语言的所有特性。
- 图文并茂，帮助读者对知识点建立直观印象。
- 结合笔者多年的 C 语言编程和系统开发经验，特别标注出易出错的技术点或初学者易误解的细节，使读者在学习中少走弯路，加快学习进度。

- 很多 C 语言书籍只讲语法规则，不讲数据结构，读者即便掌握了语法理论，也无法写出大型的 C 语言程序。而本书介绍了数据结构和算法的知识，阐述了结构化程序设计的思想，探讨了高质量编程的问题，为读者以后深入学习软件开发打下基础。
- 注重加强读者对技术点本质的理解，对诸如“编译器如何为程序实体分配内存”、“函数调用细节”等技术问题做了很多独创性的介绍。

## 本书内容

本书共分为 4 篇，23 章，第一篇从 C 语言的基础知识讲起，使读者初步了解 C 语言语法和编程机制。如果将编写 C 语言程序比作盖房子，那么基础知识就相当于砖瓦水泥。第二篇讲述如何将这些知识组织起来以构成完整的 C 语言程序。第三篇介绍了进阶内容，讨论一些深层次的技术细节，理解困难、易出错的要点。第四篇介绍了案例实践和面试技巧。

第一篇（第 1 章~第 9 章）C 语言基础。讲述了 C 语言的基础知识，包括 C 语言介绍、C 语言程序开发步骤、不同的开发环境、C 语言程序的组成、变量及数据类型、输入与输出、运算符和表达式、语句、分支、循环等。通过阅读本篇，读者可对 C 语言程序有个初步而全面的认识，了解 C 语言的由来及强大功能，明确开发环境如何通过文本形式的代码生成二进制形式的代码，熟悉 C 语言程序的结构，知道如何声明变量，如何组织语句。学完本篇，读者便可自行书写简单的 C 语言程序。这 9 章的知识是进一步学习的基础。

第二篇（第 10 章~第 15 章）一窥 C 语言门庭。C 语言博大精深，掌握了第一篇中的基础知识可以说只到了大门口。本篇从 C 语言的核心——函数讲起，介绍了与数组、指针、字符串和结构体相关的内容。指针是 C 语言的难点，也是 C 语言灵活性和实用性的直接体现。数组、字符串和结构体也是 C 语言初学者容易感觉头疼的地方。所以说，学完本篇才算迈进了 C 语言的大门。

第三篇（第 16 章~第 21 章）C 语言进阶主题。第二篇从较为独立的角度讲述了函数、数组、指针和结构体的知识，在实际应用中，这些要素彼此交叉，应用组合方式千变万化，这也是 C 语言灵活性的具体体现。本篇用两章的篇幅，分别介绍了指针和函数的技术细节，对初学者来说，理解起来可能略有难度，但这是通往高层次 C 语言学习的必经之路。此外，本篇还介绍了文件处理、编译及预处理、变量的生存期、作用域、可见域及数据结构方面的内容。本篇将使读者对 C 语言有更深入的体会和理解。

第四篇（第 22 章~第 23 章）C 语言程序设计实例与面试题解析。本篇旨在让读者掌握如何用 C 语言开发案例和实践项目。本篇提供了几种常见游戏的开发，帮助读者进一步掌握 C 语言的语法和一些经典算法。最后一章通过一些常见的 C 语言面试题，为读者踏入职场、参加求职考试提供参考资料。

本书由浅入深，由理论到实践，尤其适合初级、中级读者逐步学习和完善自己的知识结构。

## 本书读者对象

本书作为 C 语言的基础教程，适合于以下人士：

- C 语言的初、中级读者
- 了解 C 语言，但所学不全面的人员

- 高等院校学习 C 语言课程的学生
- 使用 C 语言进行毕业设计的学生
- 使用 C 语言进行项目开发的人员
- 其他相关技术人员

## 本书作者

本书主要由康莉、李宽编写，其他参与编写和资料整理的人员有：冯华君、刘博、刘燕、叶青、张军、张立娟、张艺、彭涛、徐磊、戎伟、朱毅、李佳、李玉涵、杨利润、杨春娇、武鹏、潘中强、王丹、王宁、王西莉、石淑珍、程彩虹、邵毅、郑丹丹、郑海平、顾旭光。

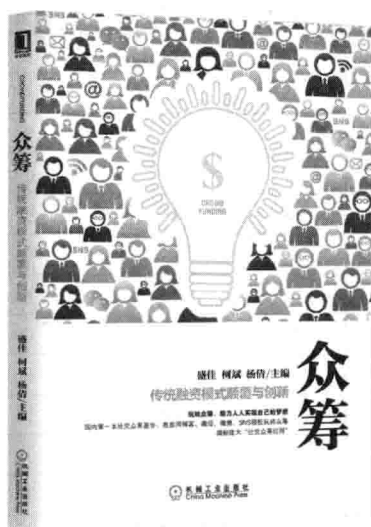
编者

2014 年 2 月

---

## 推荐阅读

---



### 众筹

---

**国内第一本社交众筹著作：教你用博客、微信、微博、SNS轻松玩转众筹，揭秘庞大“社交众筹红利”**

未来属于众筹。十年内，众筹在全球将有3000亿美元的市场规模。

本书站在市场最前沿，回眸众筹历史，描述众筹的当下图景，理性分析众筹模式的革命性，勾勒出在社交网站上玩转众筹的模式，并深入解读中美众筹业不同的发展机遇与监管规则，解密推动众筹成为主流筹资方式的动力所在。

本书适合希望在互联网金融新浪潮中所斩获的读者，是低收入群体、初始创业者、梦想家及中小微企业通过互联网融资方式找到机遇、迅速成长的必备金融服务读本。

---

# 目 录

---

## 前言

## 第一篇 C语言基础

第 1 章 踏上征程前的思想动员 .....	1
1.1 为什么选择 C 语言 .....	1
1.2 如何学好 C 语言 .....	3
1.3 语言概述 .....	3
1.3.1 什么是语言 .....	3
1.3.2 什么是机器语言 .....	3
1.3.3 什么是汇编语言 .....	4
1.3.4 面向过程的语言 .....	5
1.3.5 什么是 C 语言 .....	5
1.4 程序的开发周期 .....	5
1.4.1 编辑 C 源代码 .....	6
1.4.2 编译 C 源代码 .....	6
1.4.3 连接目标文件 .....	7
1.4.4 编译连接过程示例 .....	7
1.4.5 运行程序 .....	9
1.5 VC++、C++、C 和 TC 的区别 .....	9
1.6 小结 .....	11
1.7 习题 .....	12
第 2 章 跟我写 Hello World .....	13
2.1 了解需求才能创建程序 .....	13
2.2 认识 LCC-Win32 开发环境 .....	13
2.2.1 为什么选择 LCC-Win32 .....	14
2.2.2 启动 LCC-Win32 .....	14
2.2.3 新建 Hello World 工程 .....	14



2.2.4	定义新工程	15
2.2.5	添加源代码到工程	16
2.2.6	编译器设置	16
2.2.7	连接器设置	16
2.2.8	调试器设置	17
2.2.9	开始编辑代码	17
2.3	编译运行	18
2.4	调试排错 (debug)	19
2.4.1	debug 的由来	19
2.4.2	设置断点	20
2.4.3	调试运行	20
2.5	第一个程序容易出现的错误	21
2.5.1	包含命令 include	21
2.5.2	关键字 main	21
2.5.3	表示代码行结束的分号	22
2.6	小结	22
2.7	习题	22
<b>第 3 章</b>	<b>分解 Hello World——最简单 C 程序的组成</b>	<b>24</b>
3.1	C 程序的构成	24
3.2	C 程序的注释	25
3.3	C 程序必须有的 main 函数	26
3.4	调用函数在屏幕上显示文字	27
3.5	#include 预处理器指示符	28
3.5.1	函数声明及其作用	28
3.5.2	试验寻找#include 的作用	29
3.5.3	#include 的作用	30
3.6	计算 1+1 的程序实例	31
3.7	数据从哪里来, 又到哪里去——变量	32
3.7.1	变量在内存中的表现形式	32
3.7.2	编译器使用变量符号表	32
3.7.3	变量及其使用	33
3.8	自己设计 C 函数	35
3.8.1	在 main 函数中计算 3 个整数的平均数	35
3.8.2	在 main 函数中分 3 次计算 3 个整数的平均数	36
3.8.3	自编函数实现计算 3 个整数的平均数	36
3.8.4	如何自编写函数	37

3.8.5 试验观察总结函数声明和函数定义的意义 .....	38
3.9 语句构成程序 .....	39
3.10 优秀程序员的代码风格 .....	40
3.11 小结 .....	40
3.12 习题 .....	41
<b>第4章 常量、变量及数据类型 .....</b>	<b>42</b>
4.1 计算机是如何表示数据的 .....	42
4.1.1 二进制、八进制、十六进制 .....	42
4.1.2 表示数据的字节和位 .....	44
4.1.3 内存是存储数据的房间 .....	45
4.2 数据类型 .....	46
4.2.1 整数类型 .....	46
4.2.2 整数的有符号和无符号 .....	48
4.2.3 实数类型 .....	50
4.2.4 字符类型 .....	50
4.2.5 数据类型总结 .....	51
4.3 常量 .....	51
4.3.1 直接常量和符号常量 .....	52
4.3.2 符号常量的优点 .....	53
4.3.3 直接常量的书写格式 .....	54
4.3.4 转义字符 .....	55
4.3.5 字符串 .....	57
4.4 变量 .....	58
4.4.1 变量的声明 .....	58
4.4.2 变量声明的意义 .....	59
4.4.3 标识符和关键字 .....	60
4.4.4 变量在内存中占据的空间和变量的值 .....	62
4.4.5 为变量赋初值 .....	65
4.4.6 变量使用时常见的错误 .....	66
4.5 几个与变量相关的经典算法 .....	67
4.5.1 累加和累乘 .....	67
4.5.2 交换两个变量的值 .....	68
4.6 小结 .....	70
4.7 习题 .....	70
<b>第5章 用屏幕和键盘交互——简单的输出和输入 .....</b>	<b>71</b>
5.1 输入—处理—输出：这就是程序 .....	71

5.2 向屏幕输出——printf()函数详解	72
5.2.1 printf()函数的一般形式	72
5.2.2 printf()函数的输出原理	73
5.2.3 格式控制字符串的一般形式	74
5.2.4 输出带符号的整数%d	75
5.2.5 设置最小宽度的输出	75
5.2.6 长整型输出%ld	76
5.2.7 输出八进制形式	76
5.2.8 输出十六进制的形式	77
5.2.9 输出十进制的 unsigned 型数据	77
5.2.10 输出字符	78
5.2.11 输出字符串%s	79
5.2.12 输出实型数据%f	81
5.2.13 输出指数形式的实数	82
5.2.14 自动选择%f 或者%e 形式输出%g	83
5.2.15 printf()函数的几点说明	83
5.3 字符输出 putchar()和字符串输出 puts()	84
5.3.1 字符输出函数	84
5.3.2 字符串输出函数	85
5.4 接收键盘输入——scanf()函数详解	86
5.4.1 scanf()函数的一般形式	86
5.4.2 scanf()函数的输入原理	87
5.4.3 多数据输入分隔规则	89
5.4.4 控制输入的格式字符	90
5.4.5 scanf()函数的使用说明	91
5.4.6 使用 scanf()函数的注意事项	93
5.5 字符输入函数 getchar()	95
5.6 输入和输出程序举例	96
5.7 小结	98
5.8 习题	99
<b>第6章 程序的基本构成——运算符和表达式</b>	<b>100</b>
6.1 认识 C 中的运算符和表达式	100
6.1.1 运算符和表达式	100
6.1.2 操作数、运算符和表达式	101
6.1.3 C 运算符简介	102
6.2 算术运算符和算术表达式	102



6.2.1 基本的算术运算符	103
6.2.2 ++自增、--自减运算符	104
6.2.3 算术表达式和运算符的优先级及结合性	106
6.3 逗号运算符和逗号表达式	107
6.3.1 逗号表达式的一般形式	107
6.3.2 逗号表达式的优先级和结合性	108
6.4 关系运算符和关系表达式	109
6.4.1 关系运算符的种类	109
6.4.2 关系表达式的一般形式	109
6.5 逻辑运算符和逻辑表达式	110
6.5.1 逻辑运算符	111
6.5.2 逻辑真值表	111
6.6 赋值运算符和赋值表达式	112
6.6.1 赋值表达式	112
6.6.2 复合运算符	113
6.7 强制类型转换和自动类型转换	113
6.7.1 强制类型转换	113
6.7.2 自动类型转换	115
6.8 运算符的优先级	116
6.8.1 优先级、结合性汇总	116
6.8.2 sizeof 运算	117
6.9 取地址运算符	117
6.10 小结	118
6.11 习题	118
<b>第7章 程序的最小独立单元——语句</b>	<b>120</b>
7.1 5种语句类型	120
7.1.1 表达式语句	120
7.1.2 函数调用语句	122
7.1.3 空语句	122
7.1.4 复合语句(块语句)	122
7.1.5 流程控制语句	123
7.2 结构化程序设计	123
7.2.1 什么是算法	123
7.2.2 算法的表示	123
7.2.3 算法的伪代码表示	124
7.2.4 算法的流程图表示	124

7.2.5	3 种控制结构 .....	124
7.2.6	算法示例 .....	125
7.3	小结 .....	125
7.4	习题 .....	126
<b>第 8 章</b>	<b>条件判断——分支结构 .....</b>	<b>127</b>
8.1	if 语句 .....	127
8.1.1	判断表达式 .....	127
8.1.2	花括号和 if 结构体 .....	128
8.2	if...else 结构 .....	129
8.2.1	关键在 else .....	129
8.2.2	if...else 结构体 .....	130
8.3	多分支语句和分支语句嵌套 .....	131
8.3.1	多分支 .....	131
8.3.2	多分支 if 结构 .....	133
8.3.3	分支语句嵌套 .....	134
8.4	switch 结构 .....	135
8.4.1	一般形式 .....	135
8.4.2	为什么叫开关语句 .....	137
8.4.3	default 语句 .....	138
8.4.4	if 结构和 switch 结构之比较 .....	139
8.4.5	switch 结构的常见错误与解决方法 .....	139
8.5	小结 .....	140
8.6	习题 .....	141
<b>第 9 章</b>	<b>一遍又一遍——循环结构 .....</b>	<b>142</b>
9.1	构造循环 .....	142
9.1.1	循环的条件 .....	142
9.1.2	当型循环和直到型循环 .....	143
9.2	while 和 do...while 循环结构 .....	143
9.2.1	语法规则 .....	144
9.2.2	代码块 .....	145
9.2.3	while 语句的常见错误 .....	145
9.2.4	do...while 语句的常见错误 .....	146
9.3	for 循环结构 .....	147
9.3.1	基本形式 .....	147
9.3.2	表达式省略 .....	149
9.3.3	循环终止和步长 .....	150

9.3.4 for 语句的常见错误	150
9.4 循环嵌套	151
9.4.1 嵌套示例	151
9.4.2 嵌套的效率	152
9.4.3 循环嵌套程序的常见错误	153
9.5 与循环密切相关的流程转向控制语句	154
9.5.1 用 break 跳出循环	154
9.5.2 用 continue 重来一次	155
9.5.3 用 goto 实现跳转	156
9.6 小结	157
9.7 习题	157

## 第二篇 一窥C语言门庭

第 10 章 同一类型多个元素的集合——简单数组	159
10.1 什么是数组	159
10.1.1 数组是一大片连续内存空间	159
10.1.2 数组元素的访问	160
10.2 一维数组	160
10.2.1 一维数组的声明	160
10.2.2 一维数组元素的访问	161
10.2.3 一维数组的初始化	162
10.2.4 不合法的数组操作	163
10.3 二维数组	164
10.3.1 二维数组的声明	164
10.3.2 二维数组的初始化	164
10.3.3 二维数组应用举例	165
10.4 更高维的数组	166
10.4.1 高维数组的声明和元素访问	166
10.4.2 初始化	166
10.4.3 多维数组在内存中是如何排列元素的	167
10.5 小结	168
10.6 习题	168
第 11 章 写程序就是写函数——函数入门	170
11.1 什么是函数	170
11.1.1 函数的由来	170
11.1.2 C 语言中的函数	170

11.2 自定义函数	171
11.2.1 定义的语法	171
11.2.2 函数定义范例	172
11.2.3 不要重复定义	173
11.3 函数调用与返回	173
11.3.1 形参和实参	174
11.3.2 传址调用	175
11.3.3 函数返回	176
11.4 告诉编译器有这么一个函数	177
11.4.1 函数声明的语法	177
11.4.2 声明不同于定义	179
11.4.3 标准库函数的声明	180
11.5 函数的调用过程	180
11.6 小结	180
11.7 习题	180
<b>第 12 章 C 语言难点——指针初探</b>	<b>182</b>
12.1 计算机中的内存	182
12.1.1 内存地址	182
12.1.2 内存中保存的内容	183
12.1.3 地址就是指针	183
12.2 指针的定义	183
12.2.1 指针变量的声明	183
12.2.2 指针变量的初始化	184
12.2.3 指针变量的值	185
12.2.4 取地址操作符&	185
12.2.5 指针变量占据一定的内存空间	185
12.2.6 指向指针的指针	186
12.2.7 指针变量常见的错误分析与解决	186
12.3 使用指针	187
12.3.1 运算符*	188
12.3.2 指针的类型和指针所指向的类型	189
12.3.3 同类型指针的赋值	189
12.3.4 void 指针	190
12.3.5 指针的类型和指针所指向的类型不同	190
12.4 指针的运算	192
12.4.1 算术运算之“指针+整数”或“指针-整数”	192

12.4.2 指针-指针	193
12.4.3 指针的大小比较	194
12.5 指针表达式与左值	194
12.5.1 指针与整型	194
12.5.2 指针与左值	195
12.5.3 指针与 const	195
12.6 动态内存分配	196
12.6.1 动态分配的好处	196
12.6.2 malloc 与 free 函数	197
12.6.3 calloc 与 free	198
12.6.4 free 函数与指针	199
12.6.5 内存泄漏	199
12.6.6 释放已经释放了的内存会出问题	200
12.7 小结	200
12.8 习题	200
<b>第 13 章 字符串及字符串操作</b>	<b>202</b>
13.1 C 风格字符串	202
13.1.1 C 风格字符串的声明	202
13.1.2 C 风格字符串在内存中的表示	202
13.2 字符数组的输入输出	203
13.2.1 字符数组的声明	203
13.2.2 字符数组可以进行整体输入输出	204
13.2.3 使用 gets() 函数读取整行	205
13.2.4 访问字符数组中某个元素	205
13.2.5 使用 puts() 函数实现字符串的输出	206
13.2.6 使用字符数组的常见问题	206
13.3 字符串处理函数	207
13.3.1 理解：数组名是常指针	208
13.3.2 strlen() 函数与 size_t	210
13.3.3 字符串复制函数 strcpy()	211
13.3.4 字符串比较函数 strcmp()	212
13.3.5 字符串连接函数 strcat()	213
13.3.6 全转换为大写形式	213
13.3.7 链式操作	214
13.4 小结	214
13.5 习题	215



第 14 章 结构体、共用体、枚举和 typedef	216
14.1 结构体	216
14.1.1 结构体的定义	216
14.1.2 声明结构体变量	217
14.1.3 初始化结构变量	218
14.1.4 访问结构体成员	218
14.1.5 结构体定义的位置	219
14.1.6 结构体变量赋值	219
14.2 特殊结构体	220
14.2.1 结构体嵌套	220
14.2.2 匿名结构体	223
14.3 共用体	224
14.3.1 什么是共用体	224
14.3.2 共用体的定义	224
14.3.3 声明共用体变量	225
14.3.4 共用体变量的初始化	225
14.3.5 共用体成员访问	225
14.3.6 共用体赋值	226
14.4 结构体和共用体的内存差异	226
14.4.1 结构体变量和共用体变量内存形式的不同	227
14.4.2 结构体变量的内存大小	227
14.4.3 字节对齐	228
14.4.4 最宽基本类型	228
14.4.5 共用体的大小	229
14.5 枚举类型	230
14.5.1 什么是枚举类型	230
14.5.2 枚举类型的定义	230
14.5.3 声明枚举变量	231
14.5.4 枚举常量是什么	231
14.5.5 使用枚举时常见的错误	232
14.6 给类型取个别名——typedef	233
14.6.1 typedef 基本用法	233
14.6.2 #define 用法	233
14.7 小结	234
14.8 习题	234