高等院校新课程体系计算机基础教育规划教材

C语言程序设计与应用教程

(第二版)

周 虹 闫瑞峰 王永利 主编

李殿奎 主审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等院校新课程体系计算机基础教育规划教材

C 语言程序设计与应用教程

（第二版）

周　虹　闫瑞峰　王永利　主编
李殿奎　主审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书共 12 章，分别为：程序设计基础，简单的数据类型、运算符和表达式，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，编译预处理，指针，结构体与共用体，位运算，文件。书中程序都经过上机调试通过。

本书文字严谨、流畅，例题丰富，文档规范，注重程序设计技能训练。本书在第一版的基础上增加了小结和习题，同时对部分章节进行了修改。

本书适合作为高等院校非计算机专业学生学习 C 语言程序设计的教材，也可作为程序设计爱好者学习 C 语言程序设计的参考书。

图书在版编目（CIP）数据

C 语言程序设计与应用教程 / 周虹，闫瑞峰，王永利主编. —2 版. —北京：中国铁道出版社，2009

高等院校新课程体系计算机基础教育规划教材

ISBN 978-7-113-10394-1

Ⅰ. C… Ⅱ. ①周…②闫…③王… 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（2009）第 168681 号

书　　名：C 语言程序设计与应用教程（第二版）
作　　者：周　虹　闫瑞峰　王永利　主编

策划编辑：严晓舟　秦绪好
责任编辑：秦绪好　　**编辑部电话**：（010）63583215
编辑助理：孟　欣　贾　星　张　丹
封面设计：路　瑶　　**封面制作**：李　路
版式设计：于　洋　　**责任印制**：李　佳

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号　　邮政编码：100054）
印　　刷：三河市华业印装厂
版　　次：2007 年 5 月第 1 版　　2009 年 10 月第 2 版　　2009 年 10 月第 5 次印刷
开　　本：787mm×1092mm　1/16　**印张**：19.75　**字数**：483 千
印　　数：4 000 册
书　　号：ISBN 978-7-113-10394-1/TP・3506
定　　价：29.00 元

FOREWORD

第二版前言

随着计算机的普及和社会信息化程度的提高，掌握一门计算机语言已经成为计算机用户必备的技能之一。目前，无论是从事计算机专业工作的人员，还是非计算机专业的人员，都将C语言作为程序设计的入门语言。因为C语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。全书在编写过程中，力求做到概念准确、内容简洁、由浅入深、循序渐进、繁简适当、题型丰富，注重常用算法的介绍，有助于培养学生程序设计和分析问题的能力。书中全部实例都经过上机调试通过。

本书是在《C语言程序设计与应用教程》的基础上修订而成，在内容上根据实际需要进行了若干调整，每一章增加了小结和习题。

本书由周虹、闫瑞峰、王永利担任主编。其中，第1章和第8章由胡佳山编写，第2章由刘景春编写，第3章由王永利编写，第4章由李春洁编写，第5章由王晓娟编写，第6章由周虹编写，第7章由于占龙编写，第9章由王超编写，第10章由赵佳彬编写，第11章和第12章由闫瑞峰编写。最后由周虹教授统稿，李殿奎主审。

程序设计是一门实践性很强的课程，不可能只靠听课和看书就能掌握C语言程序设计，应当重视动手编写程序和上机运行程序，上机的时间越多越好。为了帮助读者学习本书，我们还编写了《C语言程序设计与应用实践教程（第二版）》，提供本书中各章学习指导、实验、习题及参考答案。

本书在编写过程中得到了中国铁道出版社和佳木斯大学很多老师的帮助，哈尔滨学院的贾宗福教授审阅了此书，并提出了许多宝贵意见，在此对他们表示衷心的感谢，同时对编写的过程中参考的大量文献资料的作者一并表示感谢。由于时间紧迫，加之编者水平有限，书中难免有不足之处，恳请广大读者提出宝贵意见和建议。

编　者

2009年7月

第一版前言

FOREWORD

随着计算机的普及和社会信息化程度的提高，掌握一门计算机语言已经成为计算机用户必备的技能之一。目前，无论是从事计算机专业工作的人员，还是非计算机专业的人员，都将C语言作为学习程序设计的入门语言。C语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。本书在编写过程中，力求做到概念准确、内容简洁、由浅入深、循序渐进、繁简适当、题型丰富，注重常用算法的介绍，有助于培养学生设计程序和分析问题的能力。书中全部实例都已上机调试通过。

本书既可作为高等院校本、专科学生的教材，也可作为其他计算机应用人员学习高级程序设计语言的参考书。

本书内容共分 12 章，分别为程序设计基础、简单的数据类型和运算符及表达式、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、编译预处理、指针、结构体和共用体及枚举类型、位运算以及文件。其中第 1 章和第 6 章由周虹和宋旭明编写，第 2 章由薛佳楣编写，第 3 章和第 4 章由闫瑞华编写，第 5 章由周虹编写，第 7 章由支援编写，第 8 章由曲思龙编写，第 9 章和第 10 章由富春岩编写，第 11 章由张竟达编写，第 12 章由刁树民编写。最后由周虹统稿，葛茂松担任主审。

程序设计是一门实践性很强的课程，不可能只凭借听课和看书就能掌握，应当十分重视自己动手编写程序和上机运行程序，上机实践多多益善。为了帮助同学们学习本书，编者还编写了一本配套的学习指导书《C语言程序设计与应用实践教程》，提供本书中各章的学习指导、实验、习题及参考答案。

本书在编写过程中得到了中国铁道出版社和佳木斯大学很多老师的帮助，哈尔滨学院的贾宗福教授审阅了此书，并提出了许多宝贵意见，在此对他们表示衷心的感谢。同时对在编写过程中参考的大量文献资料的作者一并表示感谢。由于时间紧迫，加之编者水平有限，书中难免有疏漏和不足之处，恳请读者提出宝贵意见和建议。

编　者

2007年2月

CONTENTS

目 录

第1章　程序设计基础

学习目标

- 了解算法的概念和特性，掌握至少一种流程图的画法。
- 了解程序设计及结构化程序设计方法。
- 掌握C语言程序的构成及书写风格，对C语言程序有一个初步了解。

计算机是20世纪最伟大的发明之一，它的出现和飞速发展对社会的各个领域都产生了深远的影响，已被广泛地应用到各行各业。使用计算机语言开发应用程序，解决实际问题是科学技术人员应具备的能力。

为了有效地进行程序设计，编写质量高、易读性好的程序，至少应掌握以下3个方面的知识：

（1）掌握一门高级语言。

（2）掌握解题的方法和步骤，即算法设计，它是程序设计的核心。

（3）掌握结构化程序的设计方法。

1.1　算法及表示

为了解决一个问题而采取的方法和步骤称为算法。

一个程序应包括以下两方面的内容：

（1）数据的描述：在程序中要指定数据的类型和数据的组织形式，即数据结构。

（2）对数据操作的描述：即操作步骤，也就是算法。

1.1.1　算法的特性

算法须具备如下5个特性：

（1）有穷性：一个算法必须总是在执行有限个操作步骤和可以接受的时间内完成其执行过程。

（2）确定性：算法的每一步操作都必须有确切的含义，不允许有二义性；对于相同的输入数据则应有相同的输出结果。

（3）输入：有零个或多个输入，即执行算法时需要从外界取得要处理的信息。

（4）输出：有一个或多个输出，输出结果。

（5）可行性：算法中的操作都是可以通过已经实现的基本运算执行有限次来完成的。

1.1.2　算法的表示

算法可以使用各种不同的方法来描述。常见的算法表示方法有：自然语言、伪码、传统流程图、N-S结构流程图等。

1．用自然语言表示算法

自然语言就是人们日常使用的语言，可以是中文、英文等。

【例 1.1】求 $\sum_{k=1}^{5}k$，即 1+2+3+4+5 的值。用自然语言表示算法。

第 1 步：将 1 存放到变量 sum 中，即 sum=1。

第 2 步：将 2 存放到变量 k 中，即 k=2。

第 3 步：计算 sum+k，并将结果存放到 sum 中，即 sum= sum+k。

第 4 步：将 k 的值增加 1，即 k=k+1。

第 5 步：判断 k 是否大于 5，若 $k\leqslant 5$，再执行第 3～5 步；若 $k>5$，算法结束，此时变量 sum 的值就是最后的结果。

此算法不仅可以计算 $\sum_{k=1}^{5}k$，还可以计算 $\sum_{k=1}^{100}k$，只不过要将第 5 步改为判断 k 是否大于 100。

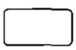
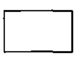
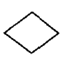
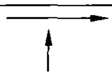
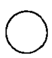
用自然语言表示的算法简单、通俗易懂，但文字冗长，表达上不易准确，易有二义性，所以一般不用自然语言描述算法。

在算法设计时，常用流程图表示算法。

2．用传统流程图表示算法

传统流程图是用规定的一组图形符号、流程线和文字说明来表示各种操作的算法，如表 1–1 所示。

表 1-1　传统流程图常用符号

符　　号	符 号 名 称	含　　　　义
	起止框	表示算法的开始和结束
	输入/输出框	表示输入/输出操作
	处理框	表示对框内的内容进行处理
	判断框	表示对框内的条件进行判断
	流程线	表示流程的方向
	连接点	表示两个具有同一标记的“连接点”应连接成一个点

用传统流程图表示算法直观形象，算法的逻辑流程一目了然，便于理解，但画起来比较麻烦，且由于允许使用流程线，使用者可以随心所欲，使流程可以任意转移，从而造成阅读和修改上的困难。

【例 1.2】用传统流程图表示对两个数按从小到大顺序输出的算法，如图 1–1 所示。

为克服上述弊病，提出了结构化的程序设计方法。在结构化的程序设计方法中，流程图包括 3 种基本程序结构：

（1）顺序结构：顺序结构是结构化程序设计中最简单的结构，它由若干条简单语句组成，完全

按照语句排列顺序执行。顺序结构有一个入口和一个出口，中间可以包含若干个操作。顺序结构的流程图如图 1-2 所示，该图表示先执行处理 A，然后再顺序执行处理 B。

（2）选择结构：选择结构又称分支结构，它由一个条件和两组语句组成，执行时根据条件的真假来选择执行的分支。它有一个入口和两个出口。选择结构的流程图如图 1-3 所示。当判断条件成立时，执行处理 A，否则执行处理 B。

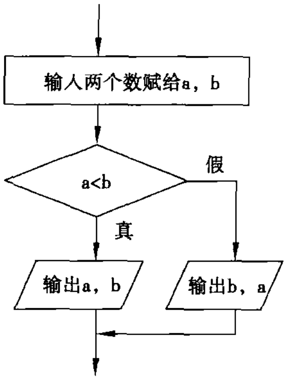


图 1-1　两个数由小到大输出

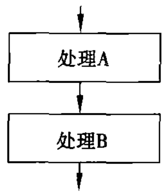


图 1-2　顺序结构

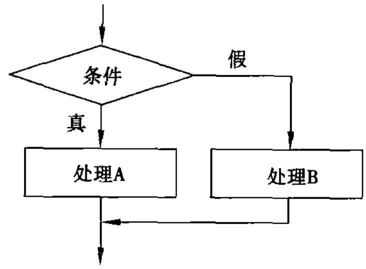


图 1-3　选择结构

（3）循环结构：循环结构是根据一定的条件，对某些语句重复执行的结构，被重复执行的部分称为循环体。循环结构由两部分组成，一是循环条件，二是循环体。是否执行循环体由循环条件决定。根据对循环条件判断位置的不同，循环结构又分为当型循环和直到型循环两种。

① 当型循环：当型循环是先判断循环条件。如果条件满足，执行一次循环体；如果条件不满足，退出循环结构。在当型循环结构中，当判断条件成立时，就反复执行处理 A（循环体），直到条件不成立时结束。当型循环结构的流程图如图 1-4 所示。

② 直到型循环：直到型循环是先执行一次循环体，再判断条件。如果条件不满足，再执行一次循环体，直到条件满足，退出循环体。在直到型循环结构中，反复执行处理 A，直到判断条件成立时结束（即判断条件不成立时继续执行）。直到型循环结构的流程图如图 1-5 所示。

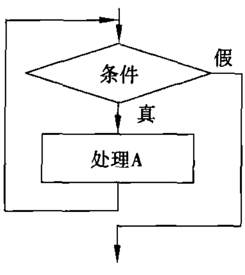


图 1-4　当型循环结构

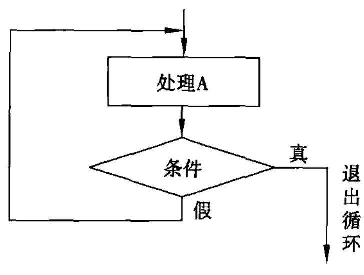


图 1-5　直到型循环结构

3. 用 N-S 流程图表示算法

N–S 流程图的主要特点是取消了流程线，不允许有随意的控制流，整个算法的流程写在一个矩形框内，该矩形框以 3 种基本结构复合而成。

N–S 流程图表示的 3 种基本结构如下：

（1）顺序结构。顺序结构的 N–S 流程图如图 1-6 所示。

（2）选择结构。选择结构的 N-S 流程图如图 1-7 所示。

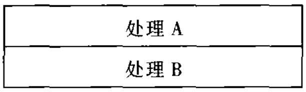


图 1-6　顺序结构

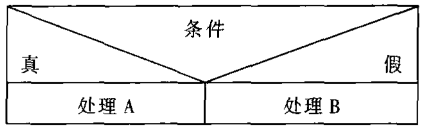


图 1-7　选择结构

（3）循环结构。当型循环结构的 N-S 流程图如图 1-8 所示，直到型循环结构的 N-S 流程图如图 1-9 所示。

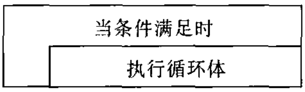


图 1-8　当型循环结构

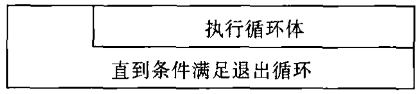


图 1-9　直到型循环结构

【例 1.3】 用 N-S 流程图表示对 3 个数进行从小到大排序的算法，如图 1-10 所示。

【例 1.4】 用 N-S 流程图表示求 10 个数之和的算法，如图 1-11 所示。

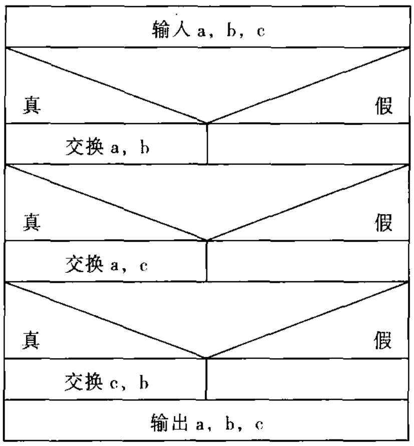


图 1-10　对 3 个数排序的算法

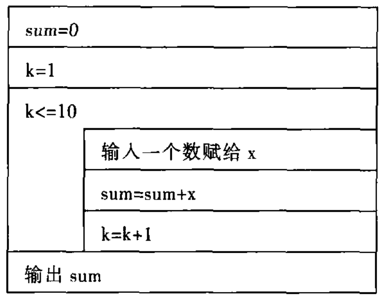


图 1-11　求 10 个数之和的算法

4．用伪码表示算法

伪码是用一种介于自然语言和计算机语言之间的用来描述算法的文字和符号。

伪码不能在计算机上实际执行，但是用伪码表示算法方便、友好，便于向计算机程序过渡。伪码的表现形式灵活自由、格式紧凑，没有严谨的语法格式。

1.2　程序设计及结构化程序设计方法

人与计算机交流的工具是计算机语言，交流的方法是使用程序。程序是为解决某一个特定问题而用某一种计算机语言编写的指令序列。

1.2.1　高级语言源程序的执行

用高级语言编写的程序称为高级语言源程序，它不能在计算机上直接运行。高级语言程序必须经过编译、连接，形成一个完整的机器语言程序，然后再执行。

编译是将高级语言源程序翻译成机器语言程序的过程，完成这个操作的程序称为编译程序，翻译成的机器语言程序称为目标程序。每种高级语言程序都有各自的编译程序，编译程序的主要功能如下：

- 对源程序进行词法分析和检查。
- 对源程序进行语法检查和语法分析。
- 为变量分配存储空间。
- 生成目标程序。

经过编译得到的目标程序是不能直接运行的，因为目标程序可能要调用内部函数、外部函数或系统提供的库函数等。因此，在执行之前还需要将所有的目标程序和系统提供的库函数等连接在一起成为一个完整的机器语言程序，这个机器语言程序称为可执行程序。完成这个过程的程序称为连接程序。

计算机执行高级语言程序的过程如图 1-12 所示。

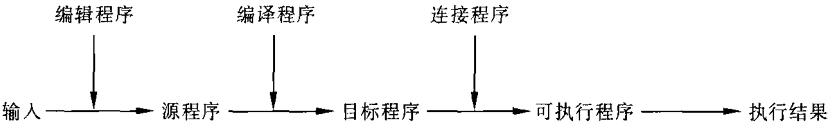


图 1-12　计算机执行高级语言程序的过程

1.2.2　程序设计

程序设计是指借助计算机，使用计算机语言准确地描述问题的算法，并正确进行计算的过程。程序设计的核心是“清晰”，程序的结构要清晰，算法的思路要清晰。

程序设计的过程可以分为若干个相互关联的阶段。针对问题的要求，从分析问题的需求出发，逐步深入，到最后编制能计算出正确结果的程序。

（1）分析问题，确定问题的需求：接受任务后，首先要对所要解决问题的处理对象进行深入地了解，深刻掌握题意，分析问题要求。只有准确定义了问题的要求，才能找到正确答案。

（2）分析问题，建立数学模型：任何一个生产过程、科学计算或技术设计都可通过一系列分析和实验，找出它们运算操作和活动的规律，然后进行归纳，并做抽象的数学描述。这种用数学方法来描述实际问题的方法称为建立数学模型。只有较准确地明确所解问题的目标，给出问题的约束条件，在一定的输入和输出情况下，才能建立好数学模型。

（3）选择计算方法：对建立的数学模型，选择一种合适的计算方法。对于同一个数学模型，往往存在多种可供使用的计算方法，即可以通过多种不同的途径处理数学模型的计算操作问题。各种不同的算法虽然都能达到计算目的，但在计算速度、求值的精度要求、存储空间的占用上都存在差异。因此要针对选定的数学模型，在多种计算方法中选择一种合理有效的方法。

（4）设计算法，绘制流程图：在编写程序之前，应该先按选取的计算方法，整理好思路，设计好一步一步运算的步骤，即算法。然后用流程图描述出来。形象直观的流程图能清晰地反映算法的基本思想和操作步骤。流程图可以根据需要，把程序设计的具体结构和细节都表示出来。有了详细的流程图，程序的编写工作就显得简单有条理，有利于程序的调试、修改和交流。

（5）编写程序：把流程图描述的算法用计算机程序设计语言恰当地进行描述，成为能交给计算机运行的源程序，这项工作就是编写程序。在编写程序的过程中，编程者要熟悉语言的语义和各种语法规则、规定，以求程序能准确地描述算法。

（6）调试程序：在程序编写中，尤其是对一些大型复杂的计算和处理过程中，由于对语言语法的忽视或书写上的问题，难免会出现一些错误，致使程序不能运行，这类错误称为语法错误。有时程序虽然能运行，但得不到正确的结果，这是由于程序描述上的错误或是对算法的错误理解造成的。有时对特定的运算对象是正确的，而对大量的运算对象进行运算时会产生错误，造成这类错误的主要因素是数学模型上的原因。这类错误属于逻辑错误。为了使程序正确地解决实际问题，在程序投入运行前，必须对程序进行反复调试，仔细分析和修改程序中的每一处错误。对于语法错误，一般根据编译程序提供的语法错误信息逐个修改。逻辑错误的情况比较复杂，必须在调试的试运行中查看计算结果是否达到预期的要求，发现错误后，要认真分析，查出症结所在，然后进行修改。在查找错误时可以采取分段调试、逐层分析等有效的调试手段。调试的目的是获得一个完整的、能正确投入运行的程序。

（7）整理资料和交付使用：程序编写、调试结束后，为了使用户能了解程序的具体功能和掌握程序的运行操作，有利于程序修改、阅读和交流，必须将程序设计各阶段形成的资料和有关说明加以整理，写成程序说明书，内容包括：程序名称、任务的具体要求、给定的原始数据、算法、程序流程图、程序清单、调试及运行结果、程序操作说明、程序的运行环境要求以及其他需要说明的资料。程序说明书作为程序设计的技术报告，在程序正式交付使用时，应随同程序一起交给用户。用户根据程序说明书的要求将程序投入实际运行，并以此对程序的技术性能和质量作出评价。

为了编写程序，必须先设计出算法。有了正确的算法才能正确地编写程序。另一方面，程序处理的对象是数据，每个数据都有一定的特性，而且数据之间还有一定的联系。当处理的对象比较复杂时，编程者必须仔细地分析数据以及它们之间的联系，把它们合理地组织起来，也就是说要选择合适的数据结构。对不同的数据结构，在程序中要采用不同的方法处理。因此，程序不仅要描述算法，还应当描述数据结构。著名的计算机科学家沃思（Wirth）说："程序就是在数据的某些特定的表示方式和结构的基础上，对抽象算法的具体描述。"他提出了一个著名的公式来表达程序的实质：

算法+数据结构=程序

对同一个问题的求解，可以采用不同的数据结构和不同的算法，而不同的数据结构直接影响着算法的复杂度和解题效果。

1.2.3 结构化程序设计

结构化程序设计方法只使用顺序、分支和循环 3 种基本结构来实现算法，编写的程序有结构清晰、可读性强、易查错等特点，使程序设计的效率和质量都得以提高。

模块化设计方法、自顶向下设计方法和逐步求精设计方法是结构化程序设计方法最典型、最具有代表性的方法。

1. 模块化设计方法

模块化设计方法是指将一个复杂的程序或算法分解成若干个功能单一、相对独立的模块，再按层次结构将其联系起来。