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

周虹　等编著

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

周虹　等编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

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。书中程序都在计算机上调试通过。

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，本书可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，也可作为其他人员学习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/周虹等编著. —北京：中国铁道出版社，2007.4

高等院校新课程体系计算机基础教育规划教材

ISBN 978-7-113-07754-9

Ⅰ.C… Ⅱ.周… 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（2007）第050204号

书　　名：C语言程序设计与应用教程

作　　者：周　虹　等

出版发行：中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑：严晓舟　秦绪好

责任编辑：祁　云　贾　星

封面设计：路　瑶

封面制作：白　雪

责任校对：包　宁

印　　刷：北京市彩桥印刷有限责任公司

开　　本：787×1092　1/16　印张：18.25　字数：426千

版　　本：2007年5月第1版　　2007年5月第1次印刷

印　　数：1～5 000册

书　　号：ISBN 978-7-113-07754-9/TP·2119

定　　价：24.00元

FOREWORD

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 章由阎瑞华编写，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 章由张竞达编写，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，哈尔滨大学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

- 了解程序设计的一些基础知识
- 了解算法的概念和特性，掌握一种流程图的画法
- 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，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。

本章主要介绍计算机语言、算法、程序设计及C语言程序设计基础。

1.1　程序设计初步

1．计算机语言

要使计算机按人的指令工作，必须将人的想法以其能理解的方式告诉计算机，与计算机交流。正如人和人进行信息交流时需要用语言一样，人与计算机交流也要用某种特定的语言，这就是计算机语言。用计算机语言编写的指令序列称为程序。

计算机语言是人与计算机之间进行交流的工具，计算机语言是人们根据描述问题的需要设计出来的。当人需要计算机工作时，就用计算机语言将工作步骤和方法告诉计算机，计算机按照人的指令工作，得到工作结果。

计算机语言的种类很多，但一般来说都包含以下4种成分。

- 数据成分：描述程序中涉及的数据。
- 运算成分：描述程序中包含的运算。
- 控制成分：描述程序中的控制结构。
- 传输成分：表示程序中数据的传输。

随着计算机技术的不断发展，不同风格的计算机语言不断出现，逐步形成了计算机语言体系，经历了由低级到高级的发展过程。

（1）机器语言

计算机只能识别二进制数0和1，每一种类型的计算机都规定了若干个二进制数0和1组成的序列，用来表示计算机的某种操作。0和1组成的序列称为机器指令。这种计算机能直接识别和执行的机器指令的集合称为机器语言。

由于不同计算机系统的硬件构成不同，其机器指令也不同，所以不同类型计算机的机器语言也不同。机器语言不是通用的计算机语言，故称为面向机器的语言。

（2）汇编语言

虽然机器语言的指令能被计算机直接识别，但机器指令烦琐、难以记忆，为克服机器语言的缺点，用指令助记符来代替 0、1 序列。这种用助记符代替二进制指令的语言称为汇编语言。

汇编语言的指令不能直接被计算机识别，必须经过翻译，计算机才能识别。同机器语言一样，汇编语言也是面向机器的语言，不同计算机系统有不同的汇编语言。

（3）高级语言

汇编语言虽然较机器语言有所改善，但仍然有它的局限性，而且与自然语言相差太远，不符合人类的习惯。人们在长期实践基础上，于 20 世纪 50 年代末创造了独立于计算机、表达方式接近人类自然语言、容易学习和使用的高级语言。高级语言又称算法语言，是一种独立于机器，面向于应用，实现算法的语言，其有易学、易懂，通用性强，兼容性好，便于移植等优点。经过几十年的发展，已经出现了多种高级语言，但是计算机不能直接识别高级语言的指令。

（4）面向对象程序设计语言

面向对象程序设计是软件系统设计与实现的新方法。面向对象程序设计语言能提供特定的语法成分来保证和支持面向对象程序设计，如 Visual C 和 Visual Basic 等。面向对象程序设计语言可分为两大类：纯粹的面向对象语言和混合型的面向对象语言。在纯粹的面向对象语言中，几乎所有的语言成分都是“对象”，而混合型的面向对象语言是在原传统的算法语言中加入各种面向对象的语言机制。

2．程序

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。

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，然后才能执行。

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，翻译成机器语言的程序称为目标程序。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 1-1 所示。

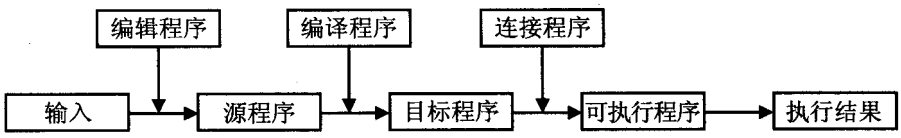

图 1-1　高级语言程序执行过程

1.2　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。

- 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。
- 对操作的描述。即操作步骤，也就是算法。

1.2.1　算法的特性

算法必须具备如下 5 个特性。

（1）有穷性

一个算法必须总是执行有限个操作步骤并且在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。

1.2.2　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，用自然语言表示算法。

第 1 步：计算 1+2，得到结果 3。

第 2 步：将第 1 步的计算结果 3 加上 3 得到结果 6。

第 3 步：将第 2 步的计算结果 6 加上 4 得到结果 10。

第 4 步：将第 3 步的计算结果 10 加上 5 得到结果 15。

这种算法在累加项较少的情况下使用比较直观，但累加项较多时该算法就比较麻烦。例

如计算 1+2+3+…+100 时，就需要写 99 个步骤。

如果对这个算法进行分析可以发现，从第 2 步开始每次计算都是在上一步计算结果的基础上再累加一个数，每次累加的数都比上一次的加数大 1。根据分析，若设一个变量 sum 存放每一步的一个加数，设变量 k 存放另一个加数，每步的累加和重新存放在变量 sum 中，为下一次累加作准备，那么可以使用以下算法完成该运算。

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。

这个算法不仅可以计算 $\sum_{k=1}^{5} k$，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。

对于同一个问题可以有不同的解题方法和步骤，这个例题还可以有其他的算法。算法有优劣之分，有些算法只需很少的步骤，而有些算法则需要较多的步骤。一般来说应采用方法简单、步骤少的算法。因此为了有效地解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适的算法。

用自然语言表示的算法简单、通俗易懂，但文字冗长，不易准确表达，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，用户可以随心所欲，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的顺序输出的算法，如图 1-2 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，流程图只包括 3 种基本程序结构。

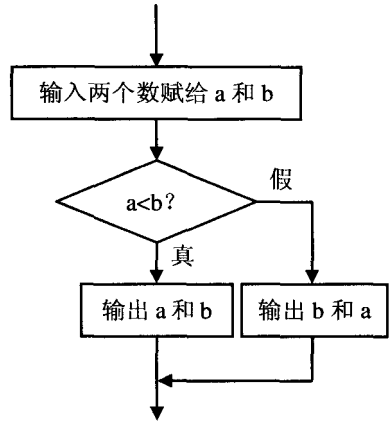

图 1-2　两个数由小到大输出

（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，中间可以包含若干操作。顺序结构的流程图如图 1-3 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

分支结构又称选择结构，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。选择结构的流程图如图 1-4 所示，当判断条件成立时，执行处理 A，否则执行处理 B。

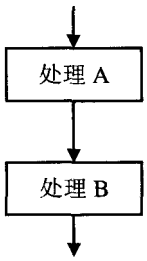

图 1-3　顺序结构

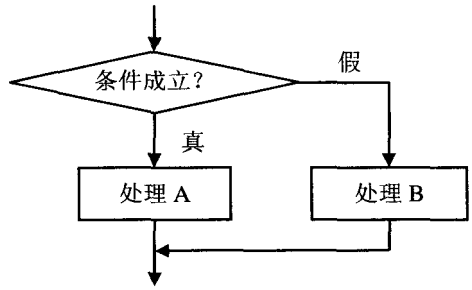

图 1-4　选择结构

（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 1-5 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 1-6 所示。

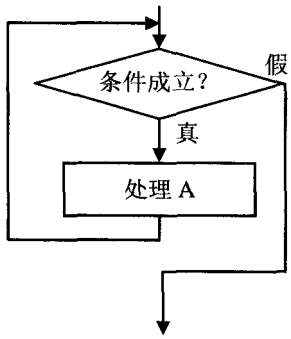

图 1-5　当型循环结构

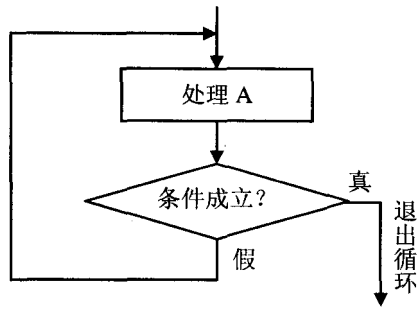

图 1-6　直到型循环结构

3. 用 N-S 流程图表示算法

针对传统流程图存在的问题，提出一种新的结构化流程图形式，简称为 N-S 流程图。

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，该矩形框由 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 1-7 所示。

（2）选择结构。选择结构的 N-S 流程图如图 1-8 所示。

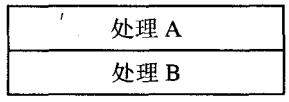

图 1-7　顺序结构

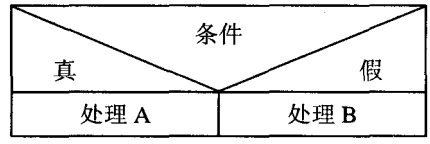

图 1-8　选择结构

（3）循环结构。当型循环结构的 N-S 流程图如图 1-9 所示，直到型循环结构的 N-S 流程图如图 1-10 所示。

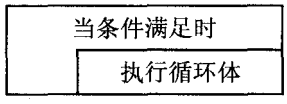

图 1-9　当型循环结构

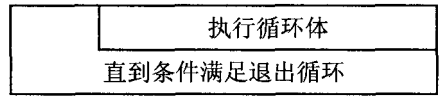

图 1-10　直到型循环结构

【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 1-11 所示。

【例 1-4】用 N-S 流程图表示求 10 个数之和的算法，如图 1-12 所示。

4. 用伪码表示算法

伪码用一种介于自然语言和计算机语言之间的文字和符号来描述算法。

例如，用伪码描述上述算法：

```
input x,y,z
x=max
if y>max then y=max
if z>max then z=max
output max
```

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。

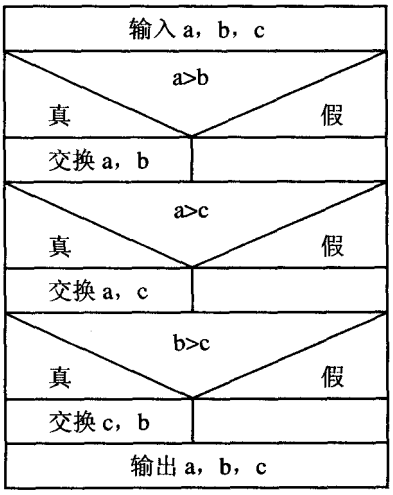

图 1-11　对 3 个数排序的算法

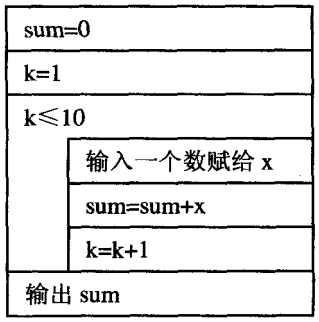

图 1-12　求 10 个数之和的算法

1.3　程序设计及结构化程序设计方法

1.3.1　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，并作抽象的数学描述。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。例如，计算数值积分，可以用矩形法、梯形法和辛普生法等计算。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。