普通高等教育信息技术类系列规划教材

C语言程序设计

孔垂柳　宋维平　周雅翠　主编
李刚健　主审

科学出版社
www.sciencep.com

普通高等教育信息技术系列规划教材

C 语言程序设计

孔垂柳　宋维平　周雅翠　主编

李刚健　主审

科学出版社

北京

内 容 简 介

本书全面介绍了C语言的体系、概念、语法和语义、特点及结构化程序设计方法。全书共13章，第1章介绍了C语言程序设计的基础知识；第2、3章介绍了C语言的基本数据类型、常量和变量以及表达式；第4～6章介绍了用C语言进行结构化程序设计的基本方法，包括结构化程序的顺序结构、选择结构、循环结构及其设计方法；第7、10章分别介绍了函数及编译预处理相关知识；第8、9章对C语言的数组、指针作了详尽的阐述；第11～13章分别介绍结构体与共用体、文件、位运算相关内容。全书以编程应用为驱动，通过案例和问题引入内容。每一章都有由浅入深的程序范例，以尽可能详尽地解释相关语法的概念、作用、含义和使用方法，强调对C语言基础知识的理解和掌握，注重培养学生的程序设计综合素质和解决实际问题的能力。

本书可作为工科院校相关专业本科生、专科生的程序设计课程教材，也可供程序设计爱好者自学和参考。

图书在版编目（CIP）数据

C语言程序设计/孔垂柳，宋维平，周雅翠主编. —北京：科学出版社，2009

（普通高等教育信息技术系列规划教材）

ISBN 978-7-03-025957-8

Ⅰ.C… Ⅱ.①孔…②宋…③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（2009）第200993号

责任编辑：王纯刚 陈晓萍 / 责任校对：柏连海

责任印制：吕春珉 / 封面设计：东方人华

科学出版社出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

双青印刷厂印刷

科学出版社发行 各地新华书店经销

*

2009年12月第 一 版 开本：787×1092 1/16

2009年12月第一次印刷 印张：17 3/4

印数：1—4 500 字数：426 000

定价：30.00元

（如有印装质量问题，我社负责调换〈双青〉）

销售部电话 010-62134988 编辑部电话 010-62135120-8003

本书编写人员

主　编　孔垂柳　宋维平　周雅翠

副主编　李　杰　李丽新　赵　越

参　编　战冬梅　穆泉伶　张伟杰　张沛露

崔立波　刘立辉　刘舒婷　郑　琦

陈　静　袁一平　岳俊华

主　审　李刚健

前　言

自20世纪80年代以来，高等学校计算机教育发展迅速，计算机教育的内容不断扩展、程度不断加深。特别是近十余年来，计算机向高度集成化、网络化和多媒体化发展的速度一日千里。社会信息化不断向纵深发展，各行各业的信息化进程不断加速。计算机应用技术与其他专业的教学、科研工作的结合更加紧密。各学科与以计算机技术为核心的信息技术的融合，促进了计算机学科的发展，各专业对学生的计算机应用能力也有更高和更加深入的要求。

基于近年来计算机科学的发展以及教育部关于计算机基础教学改革的指导思路，我们确立了本书的编写思想。本书的所有编者均为一线教师，根据长期在C语言程序设计教学中所遇到的问题有针对性地编写了此书。另外，我们在国内高校做了系统、详细的调研。对教育部制订的教学计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教学计划、传播先进教学理念、为培养符合社会需要的高素质创新型、应用型人才服务的创作宗旨。

在本书的策划过程中，我们组织了多次研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

与目前出版的C语言程序设计教材相比，本书的主要特点是：充分考虑到工科院校学生知识、能力、素质的特点和实际教学情况，增强了实用性。采用案例式教学法，以编程应用为驱动，教材内容经过精心组织，体系合理、结构严谨，由浅入深、循序渐进地讲解C语言程序设计的思想和方法。每章均精心设计了由浅入深的例题，能够让学生充分地理解相关知识点，并通过每章的课后习题巩固、加深对基本概念的理解和掌握，提高C语言程序设计的水平。由于函数是C语言中统领全局的重要概念，我们将函数的内容提到数组和指针之前来讲解，并在后边的相关章节中反复强化函数的概念和使用。经验证明，这将更有利于学生对C语言程序设计完整的理解和把握。

本书可作为工科院校相关专业本、专科学生程序设计课程教材，也可供程序设计爱好者自学和参考。

教育的改革不会停止，教材也将不断地推陈出新。本书将接受广大教学第一线教师的检验。由于我们的水平和经验有限，本书在编审、出版工作中还存在不少缺点和不足，欢迎使用本书的广大读者提出批评与建议，以便改进我们的工作，使教材质量不断提高。

目　录

第 1 章　C 语言概述

计算机本身是无生命的机器，要使计算机能够运行起来，实现各种功能，完成各种任务，就必须让它执行相应的程序。这些程序都是依靠程序设计语言编制出来的。

在众多的程序设计语言中，C 语言有其独特之处。它作为一种高级的程序设计语言，具备方便性、灵活性和通用性等特点；同时，它还向程序员提供了直接操作计算机硬件的功能，具备低级语言的特点，适合各种类型的软件开发。因此，C 语言是深受软件工作者欢迎的程序设计语言。

本章首先简单地介绍了 C 语言的发展和特点，然后从程序设计的角度介绍有关基本概念，最后通过几个简单的 C 程序实例描述 C 语言程序的基本结构。

1.1　C 语言的发展与特点

1.1.1　C 语言的发展

C 语言是由美国 Bell 实验室的 Dennis Ritchie 设计，在 BCPL 语言和 B 语言的基础上，于 1972 年在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上实现的，它取 BCPL 的第二个字母而得名。

BCPL（Basic Combined Programming Language）语言是由英国剑桥大学的 Martin Richards 于 1967 年提出的一种系统程序语言。1970 年美国 Bell 实验室的 Ken Thompson 以 BCPL 语言为基础，作了进一步简化，设计出了很简单的而且很接近硬件的 B 语言，并且用 B 语言在 PDP-7 计算机上写出了第一个 UNIX 操作系统，1971 年又用 B 语言在 PDP-11/12 计算机上写出了 UNIX 操作系统。

C 语言既保持了 BCPL 语言和 B 语言的优点（精炼，接近硬件），又克服了它们的缺点（过于简单，功能有限，数据无类型等）。最初的 C 语言只是为描述和实现 UNIX 操作系统而设计的一种系统工作语言。C 语言与 UNIX 操作系统密切相关，UNIX 操作系统、编译程序和应用程序大多由 C 语言写成。1973 年，Ken Thompson 和 Dennis Ritchie 两人合作，将原来由他们于 1969 年用汇编语言写的 UNIX 操作系统的 90%以上用 C 语言改写，形成了 UNIX 第 5 版。

最初的 C 语言主要是在美国 Bell 实验室内部使用，直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的普遍关注。1977 年出现了不依赖于机器的 C 语言编译文本《可移植 C 语言编译程序》，使得用 C 语言开发的软件在移植到其他机器时所做的工作大大简化，这也推动了 UNIX 操作系统迅速地在各种机器上实现。随着 UNIX 操作系统被日益广泛使用，C 语言也迅速得到推广。在 C 语言与 UNIX 操作系统的发展过程中，它们相辅相成，相互促进。1978 年以后，C 语言已先后移植到大、中、小、微型机上，已独立于 UNIX 操作系统和 PDP 计算机了。

1978年，Brian Kernighan 和 Dennis Ritchie 两人以 UNIX 第7版中的C语言编译程序为基础，合著了影响深远的名著《The C Programming Language》，这本书中所介绍的C语言在以后一直是事实上的C语言标准版本。

1983年初夏，美国国家标准化协会（ANSI）专门成立了一个委员会，为C语言制定了ANSI标准，称为ANSI C，比原来的C有了很大的发展。1988年，Brian Kernighan 和 Dennis Ritchie 按照 ANSI C 的标准修改了他们的经典著作《The C Programming Language》。Turbo C、Visual C++等完全是按照ANSI的C语言标准实施的，是一种快速、高效的编译程序，并提供了集成开发环境。

目前，C语言已风靡全世界，成为应用最广泛的计算机高级语言之一。C语言不仅可用来开发系统软件，也可用来开发应用软件。C语言具有一般高级语言的特性（可读性、可移植性好，结构化程序设计），同时具有低级语言（如汇编语言）的特性（可直接对硬件进行操作，如对位、字节和内存地址的操作），它是一种集高级语言和低级语言优点于一身的语言。因此，C语言常被称为计算机“中级语言”。

1.1.2 C语言的特点

C语言是世界上自20世纪70年代以来最有影响的计算机语言之一，也是最常用的语言之一。C语言能得到越来越广泛的使用，归功于它有不同于（或优于）其他语言的特点，特别是优良的“可移植性”。它不需要做大的修改就能在每一台支持C语言的机器上运行。概括起来，C语言的主要特点如下。

（1）结构简洁、紧凑

C语言一共只有32个关键字（即保留字），压缩了一切不必要的成分，因此C语言编写的源程序短，但C语言却被公认为是一门比较强有力的语言。

另一方面，C语言的运算符和表达式的表示方法也力求简练，如增量运算符“++”及赋值表达式和逗号表达式概念的引入使表达式变得非常紧凑，函数和复合语句使用{}，而不用文字表示等，以至有时可以把多条语句浓缩成一条语句或者一个表达式。正如C语言设计者指出的那样，把C的规模压缩到较小会带来一些好处，语言的简洁特性自然地提高了程序员的潜在的生产力。对于语言本身的描述也简单，所以易于学习，便于理解和使用。

（2）数据类型丰富，控制结构完善

C语言数据类型丰富，它不仅有不同长度的整型、字符型和单双精度实型（即浮点型）等基本数据类型，还有指针类型以及数组、结构体和联合体等构造数据类型，能用这些类型来实现各种复杂的数据结构（如链表、树、栈等）的运算。尤其是指针类型，使用起来更为灵活、多样。

C语言提供了9种控制语句来实现3种结构（顺序、分支和循环结构）的程序设计，以支持良好的程序结构。同时，用函数作为程序模块以实现程序的模块化，每一个C程序由若干个函数组成，用户不仅可以调用丰富的库函数，还可以自定义函数。C语言函数的定义是平行、独立的，但函数可以嵌套调用和递归调用。构成一个C程序的若干个函数可以存在于几个不同的源文件中，它们可以单独编译。因此，C语言是结构化的理想语言，符合现代编程风格要求。

（3）运算符丰富，表达能力强

C 语言的运算符非常丰富，共有 34 种运算符（详见第 3 章）。它不仅具有算术运算符、关系运算符、逻辑运算符等，还把括号、赋值、逗号、变量值增（减）1、强制类型转换等都作为运算符处理，同时还将字符、逻辑值等数值化。C 语言的运算符不仅具有优先级的概念，还具有结合性的概念。因此，C 语言的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符和表达式，不仅可以使程序简洁，还可以实现在其他语言中难以实现的运算。

例如，整型变量与字符型变量可以互用，整型数据、字符型数据以及逻辑型数据可以混合运算等，这样编辑程序对表达式的语法检查都很容易通过，程序设计自由度大。如表达式 2<=x<5、x=3+(y=5)、z=8+5、5&&100>'a'+4||(x=3,y=3)等都是合法的表达式。再如，分支语句：

```
if(x==0)语句_1 else 语句_2
```

的作用是：如果变量 x 的值等于 0，则执行语句_1，否则执行语句_2。如果将它误写为

```
if(x=0)语句_1 else 语句_2
```

则该分支语句编译时也能通过，但无论变量 x 是什么值，它都执行语句_2。这是因为 C 语言将“x=0”看成是赋值语句，它的值为 0，作为条件时看成是逻辑“假”。该分支语句执行时，不仅不能正确地选择语句_1 或语句_2 进行执行，而且还将变量 x 重新赋值为 0。

（4）语法限制不太严格，程序设计自由度大

C 程序的书写形式自由，主要用小写字母，一个语句可以写在几行，一行也可以写几个语句。

C 语言对数组下标的越界也不作检查，由程序编写者自己保证程序的正确。如果对数组越界操作，很可能导致系统的崩溃。

一般的高级语言语法检查比较严格，能检查出几乎所有的语法错误，而 C 语言允许编写者有较大的自由度，因此放宽了语法检查。程序员应仔细检查程序，保证其正确，而不要过分依赖 C 编译程序去查错。“限制”与“灵活”是一对矛盾。严格限制，就失去灵活性；而强调灵活，就必然放松限制。对于一个不熟练的人员，特别是初学者，编写一个正确的 C 程序可能会比编写一个其他高级语言程序难一些。

（5）同时具有高级语言和低级语言的特点

C 语言允许直接访问物理地址，能进行位操作，可以直接对硬件进行操作，能实现汇编语言的大部分功能。因此，C 语言既具有高级语言的功能，又具有低级语言的许多功能。C 语言的这个特点使得它既是成功的系统描述语言，又是通用的程序设计语言；既可用来写系统软件，又可用来编写应用程序。

（6）目标代码质量高，可移植性好

C 语言程序不仅编译的目标代码质量高，程序执行效率高，而且可移植性好。可移植性是指从一个硬件环境不加修改或稍加修改移到另一种环境上运行的性能。虽然 C 适合用在许多计算机机型上，它却独立于具体的计算机系统。灵活地运用预处理程序可以提高程序的可移植性。

1.2 程序设计的基本概念

1.2.1 程序

“程序”这个词我们并不陌生，在日常生活中，我们会碰到各种各样的程序。如开学典礼的安排就是一个程序，每一项干什么都事先安排好了，典礼完毕后，程序也就结束了；在法律上，进行任何法律活动都有一个明确的规定，我们常说要按照法定程序办事就是这个意思。另外，有时候尽管我们没有用到“程序”这个词，但是却隐含了类似的意思。举一个常见的例子，如“用煤气炉烧一壶水”，要完成这个任务，就要对烧开水的过程进行分析，将该过程分解为一系列步骤。

- 把水装到壶里。
- 把壶放到煤气炉上。
- 打火。
- 把水烧开。
- 熄火。

按照这个顺序依次执行这些步骤就可以把这个任务完成。在这里，虽然没有用到程序这个词，但是这些有先后次序的步骤序列就相当于一个程序。从这些例子中，我们可以看出日常生活程序的大概意思，就是按一定顺序安排好的工作（操作）序列，它包括两部分：一是完成这组操作序列所设计的对象；二是对这些对象所作用的动作规则。像上面这个例子中，水、壶、煤气炉等都是对象，而装、放、打、烧、熄等则形成了动作规则。

计算机中的程序与日常生活中的程序的概念是类似的，只不过执行日常生活程序的主体是人，而执行计算机程序的主体是计算机。计算机程序就是要由计算机进行解释和执行的程序，它表示的是计算机处理事务的时间顺序和处理问题的步骤。这样的程序只能由计算机可以解释和执行的基本操作组成，也只能作用于计算机可以描述的对象——数据。组成计算机程序的基本单位一般称为指令，因此简单地说，程序就是事先编制好的具有特定功能的指令序列。

1.2.2 程序设计

既然程序是按一定次序编排的指令序列，那么编写指令序列的过程就是程序设计。用什么来编写指令序列？由于指令序列是给计算机执行的，因此这些指令应该是以计算机能够理解的语言表示的，这种语言就是程序设计语言。正如人们之间交流思想、相互沟通要以自然语言为媒介一样，任何计算机之间交流也需要一种语言。在理想的情况下，人们应该可以使用人的自然语言和计算机交流，但是目前计算机还不能理解自然语言，至少从目前来看这个目标还是遥不可及的。因此，现在能够充当人和计算机之间交流工具的就是计算机语言，包括各种命令语言和程序设计语言，主要是程序设计语言。

程序设计语言的定义是：一种适合于计算机和人的阅读方式的描述计算的记号系统。程序设计语言从问世到现在已经有半个世纪的历史了，经历了从机器语言、汇编语言到

高级语言的发展阶段。机器语言由二进制的序列组成，是计算机硬件能够直接识别的、不用机器翻译就能执行的程序设计语言。机器语言是计算机真正“理解”并识别的唯一语言。而汇编语言是符号化的机器语言，它是用符号来表示每一条指令和地址，和机器语言相比，汇编语言指令的含义比较直观，也易于阅读和理解。机器语言和汇编语言都是面向机器的，都与具体机器的硬件系统相关，因此又称为“低级语言”。低级语言编写的程序可移植性差，抽象水平低，较难编写和理解，于是后来又出现了高级语言。高级语言是面向问题的语言，独立于具体的机器，比较接近于人类的语言习惯和数学表达形式，如目前绝大多数高级语言都是用简单的英语表达。高级语言与计算机结构无关，便于学习和使用，具有更强大的表达能力，高级语言写成的程序可移植性强，便于推广。目前绝大多数程序设计语言如 C 语言、Pascal 语言都是高级语言，绝大多数程序员也是使用高级语言。

有了程序设计语言后，程序设计就有了工具，程序设计才成为可能，但是程序设计语言并不能保证高质量的程序或者软件，程序设计需要方法学和理论上的指导。在程序设计的早期，对一个可解的问题（当时是较简单的问题），常常由一两个人包揽，因此程序设计技术不免被看成是一种与个人思想、经验和技术相联系的技巧。然而，随着软件的规模越来越大，越来越复杂，这种手工式的程序设计方法越来越不能满足要求，20 世纪 60 年代末的软件危机是这种矛盾的集中爆发。所谓“软件危机”，是指当时一方面需要大量的软件系统，如操作系统、数据库管理系统；另一方面，软件研制周期长，可靠性差，维护困难。在这种背景下，1968 年，北大西洋公约组织（NATO）在前联邦德国召开了第一次软件工程会议，分析了危机的局面，研究了问题的根源，第一次提出了用工程学的办法解决软件研制和生产的问题，本次会议可以算作是软件发展史上的一个重要的里程碑。1969 年，国际信息处理协会（IFIP）成立了“程序设计方法学工作组”，专门研究程序设计方法学，程序设计从手工艺式向工程化的方法迈进。从那时开始，人们开始意识到程序设计是一门技术科学，从那以后，程序设计方法与程序设计技术取得了很大的进展，开始出现了结构化的方法、面向对象的方法等程序设计方法。

1.2.3 算法

做任何事情都有一定的步骤。例如，去看一场电影，要先去售票处买票，到了电影放映时间凭电影票入场，看完电影后再出场。这些步骤都是按一定的顺序进行的，缺一不可，而且次序错了也不行。也就是说，从事各种工作和活动，都必须事先想好进行的步骤，然后按部就班地进行，才能避免错误。

因此，广义地讲，“算法”指的是为解决一个特定问题而采取的确定的、有限的步骤。

为解决同一个问题，有多种不同的方法和步骤，即多个算法。例如，求 1+2+…+100 的结果，有人是先求 1+2，再把和加上 3，再加 4，…，一直加到 100；而高斯采取的办法是，先将 100 个数分为若干个组：（100），（1，99），（2，98），…，（49，51），（50），前面 50 个组每个组的和都是 100，因此结果为 50×100+50。当然还有其他的方法。

一般而言，方法有优劣之分，算法的优劣可以有多种评价标准，可以从时间上来评价，可以从空间上来评价，也可以从其他的角度来评价。人们当然愿意选择较优的算法。因此，为了有效解题，不仅需要保证算法的正确性，还要考虑算法的质量，选择合适的

算法。

1.2.4 数据结构

一个程序包括以下两个方面的内容。

1）对操作的描述：即操作步骤，也就是算法。

2）对数据的描述：程序中要用到的数据的类型和组织形式，即数据结构。

数据是操作的对象，操作是对数据进行加工处理，而程序的目的就是按照既定的步骤（算法）对数据进行加工，也就是说，程序是算法和数据的结合体。因此，著名计算机科学家沃斯（Niklaus Wirth）教授提出了著名的公式：

程序 = 数据结构 + 算法

那么到底什么是数据结构？我们举一个例子说明。在一个由 n 个城市组成的交通网中，现要将一些原有的公路修建成高速公路，使所有城市之间都有高速通道存在。这时有多种方案可供选择，但若要考虑投入和效益，最佳的方案是确定 n–1 条线路连接这 n 个城市，并且代价最小，这类道路、交通问题的数学模型就是一种被称为“图”的数据结构，这个问题就是其中的最小生成树问题。

简单地说，数据结构是相互之间存在一种或多种特定关系的数据元素的集合。在任何问题中，数据元素都不是孤立存在的，而是存在着某种关系，数据元素相互之间的这种关系称之为结构。在使用计算机解决问题时，需要确定所要解决问题的数学模型，还要将问题所涉及的数据和数据元素之间的关系用合适的数据结构表示出来，然后才能加以解决。

通常有下列 4 种基本结构。

1）集合。结构中的元素之间除了存在“属于同一集合”的关系外，不存在其他的关系。

2）线性结构。结构中的元素之间存在一个一对一的关系。

3）树形结构。结构中的元素之间存在一个一对多的关系。

4）图（网）状结构。结构中的元素之间存在多个多对多的关系。

每种数据结构都有自己的特点，也都有一些常用的算法。选择一个合适的数据结构，对解决问题是非常有帮助的。程序设计的实质就是针对确定的问题选择一种合适的数据结构并设计出一个好的算法。

1.3 C 语言的字符集与标识符

1. 字符集

满足 C 语言词法要求的字符包括所有大小写的英文字母、阿拉伯数字以及部分特殊符号，这些特殊符号如图 1.1 所示。

+	–	*	/	%	_(下划线)	=	<	>	&
~	(	)	[	]	.	{	}	:	?
;	"	!	#	'	^	空格			

图 1.1 C 语言字符集中的特殊符号