http://www.phei.com.cn

C语言程序设计

张世禄　潘大志　冯天敏　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

C语言程序设计

张世禄　潘大志　冯天敏　编著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书是讲述程序设计而不讲授书中程序的教材。书中使用了程序设计新方法：首先，将问题抽象和归纳成带计算过程和计算条件的计算公式；然后，找出算式所对应C语言中的语句或语句组，编写主干程序段；最后，加上说明和输入输出形成整个程序。书中首次将语言课中的程序设计所涉及的算法分成尝试算法、递归算法和迭代算法三类，并给出了各类算法所对应的基本程序模块，编写程序可像套用数学公式一样方便，从而提高了程序的重用率，降低了程序设计难度。

程序中选用了近百个例题，所有程序都给出了设计过程、带计算过程和计算条件的数学公式。由于数学语言的精炼性，因而较同类教材篇幅短、规律性强。

书中对降低程序复杂度及程序编写难度也做了介绍。教材中所有程序都具有同一风格，语句括号的配对关系、函数段以及复合语句、循环语句、条件语句中的子句书写都规范统一、一目了然。

本书适用于非计算学科专业大学本科及专科学生，也适用于计算学科本科学生入门专业基础课。

图书在版编目（CIP）数据

C语言程序设计／张世禄等编著．—北京：电子工业出版社，2005.8
ISBN 7-121-01233-2

I．C… II．张… III．C语言—程序设计 IV．TP312

中国版本图书馆CIP数据核字（2005）第065523号

责任编辑：张燕虹
审　　校：于秀山　李乐强
印　　刷：北京市海淀区四季青印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：15.5　字数：400千字
印　　次：2005年8月第1次印刷
印　　数：5000册　定价：22.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

前　言

20 世纪 50 年代中后期，各种高级语言相继问世，程序设计大门被打开。一大批科技工作者加入了程序设计行列，同时也就形成软件产业，并涌现出一大批专业软件公司或软件生产厂商，为人类研制出一大批功能奇异的软件并使各国相当多的部门受益。正当软件产业刚显火红之时，20 世纪 60 年代初爆发了软件危机。虽然北大西洋公约的软件学者们于 1968 年提出了“软件工程”，用以解决软件危机，但软件危机依旧存在，软件产品的成功率依然低下（约为 10%）。半个世纪以来，软件学者们绞尽脑汁地在解决软件危机的课题里做了不少工作：开发出第四代计算机语言，生产出不胜枚举的软件平台。

第四代计算机语言出现以后，原来用第三代计算机语言编写程序时所用的编程方法——结构化程序设计方法已显得笨拙和过时，因此应有新的程序设计方法替代。两代计算机语言形式上的最大差异是面向过程和面向对象，实质上的差异是程序、知识的重用率以及相应的程序编写难度的差异。所有第四代计算机语言都有类库结构，类库结构正是第四代计算机语言能提高程序重用率和知识重用率的关键。所有软件平台都是程序设计和软件开发的工具，都能降低程序设计难度和软件开发难度，从这个角度讲，计算机语言也属软件平台。平台中的业务架构平台能改变软件生产关系，由软件公司为用户开发软件变为公司为用户提供软件平台（生产工具），用户自己开发软件。用户自己生产软件，使得程序设计技术显得更为重要，并且使程序设计真正得以普及。实际上，即使是软件专业的硕士生、博士生，只要不在专业软件公司工作，也只能使用软件，只有专业软件公司人员才有资格承担开发、设计任务。

目前功能最强、最受用户青睐的语言是 Java 语言和 C++语言。刚一进门就学 Java 语言或 C++语言不恰当，因为 C++语言是在 C 语言的基础上开发的，且 Java 与 C++ 语言也有千丝万缕的联系，在 Java 环境下就可以直接用 C 程序，因而用 C 语言作为入门语言是较佳选择。

本书将介绍 C 语言的语法句法，但更重要的是介绍如何在 C 环境下进行程序设计，如何与第四代计算机语言的程序设计挂钩。

程序设计就是将用非计算机语言表述的算法翻译成用计算机语言表述的算法，程序本身就是算法。程序结构是一动态结构，程序运行过程是一动态过程。每一组带计算过程和计算条件的计算公式能准确地描述任何动态过程且都能找到对应的一组语句。本书介绍的程序设计方法和过程是，将算法先用带计算过程和计算条件的计算公式表述（这就是一般书中抽象应得的结果）；再找出公式对应的语句组，编写主体程序段；最后加上说明和输入输出形成完整程序。本书将算法和程序设计基本结构挂钩，对算法和程序基本结构进行了分类，从而将算法类和类库类相结合。

软件属工科，工科学生都要实习。上机实习不是练习程序录入和编辑，而是程序设计和程序调试技术，因而书中所有实习中的程序都需学生自己编写，必须如实记录程序调试过程，

而不是抄写书中提供的程序。

C 程序设计不是 C 大全，C 语言的不少功能对一般程序设计用处不大，甚至令人迷惑，因此本书有选择地介绍了 C 语言中的主要功能和内容。

本书第 1 章绪论部分、第 5 章、第 6 章以及上机实习部分由张世禄先生编写，其余章节由潘大志先生编写，书中所有程序都由潘大志先生和冯天敏女士调试并通过。全书将制作成网络课件，冯天敏女士完成最后整理工作。全书经张世禄先生多次审阅修改、增删。

作　者

目　录

第1章　绪论及C语言简介

开发业务架构平台以及各类软件平台（也称为中间件）是当前国内外软件业的动向。几乎所有的平台软件都用C++语言或用Java语言编写，而用C语言编写的程序可直接用于C++和Java环境。因此，C语言完全可以作为C++语言和Java语言的入门课程。

1.1　绪　　论

1946年，第一台计算机在美国宾夕法尼亚州问世。虽然第一台计算机庞大、笨拙、功能极差，但它孕育了一门新型学科——计算学科，孕育了两个新型产业：计算机硬件产业和软件产业。

最近60年是人类社会发展变化最快的时期，这些变化无不与计算机和软件相关。计算机硬件产业一直受到人类关注，硬件产业一直兴旺、红火；软件更受到人们青睐，人类社会各产业，尤其是高科技产业早已离不开软件了。与硬件产业的一帆风顺不同，软件产业虽然发展得也很快，但却饱含辛酸。

1.1.1　软件产业的兴旺和辛酸

计算机刚出现的头几年，硬件和软件是不分家的。当时的程序是用机器语言设计的，程序操作数是数字，运算符也是数字，可读性极差，设计难度极大，再加上当时计算机几乎只用于与国防有关的科技计算，因而软件人员极少，自然也就没有软件产业。软件只是硬件的附属，这也是本门学科的学科名一直称为计算机（Computer），直到20世纪90年代才更名为计算的原因。20世纪50年代中后期，随着多种高级语言问世，程序设计大门终于打开，大批科技工作者加入了程序设计大军，随之而来才有了软件产业。20世纪50年代后期和60年代初期，发达国家的软件人员为众多领域开发了不少软件并使众多部门受益。不过，正当软件产业刚红火、兴旺之时，60年代初期爆发了软件危机。为了解决软件危机，北大西洋公约组织的软件人员于1968年在前联邦德国慕尼黑市召开了国际学术会议，会后推出了软件工程。基于以前生产软件周期的理念，软件工程制定了软件生产各时期的任务并给出了一些辅助工具，其目的是希望程序设计规范化，文档书写规范化。软件危机现象虽好描述，但软件危机产生的根本原因却难找到。当时，人们认为作坊式生产是产生软件危机原因，实际上这一说法欠科学，因为不能说今天的软件生产仍然是作坊式生产。当今软件产品成功率低下又是什么原因呢？

尽管爆发了软件危机，而且危机也未解除，但由于当时发达国家已离不开软件，因而软件产业仍得到高速发展。据IDC统计，1975年全球软件产业的市场规模仅为2亿美元，1982年上升到100亿美元，1985年上升到250亿美元。IDC预测，2006年全球仅应用软件消费额将增至1310亿美元。由于未能找出软件危机的根本原因，软件工程只能为软件产业治标，软件产业只能带伤运作。因此，软件产业表面上看起来兴旺，但背后饱含辛酸。“在《软件系统模式的失败与成功（Patterns of Software Systems Failure and Success）》一书中，琼斯（Capers Jones）对系统软件、信息系统、通信软件、外包软件、军用软件以及消费类软件等6个领域

几千个项目做了分析，所得结论可概括为：

（1）软件开发仍然具有高度不可预知性，只有约 10%的软件项目在最初估计的预算和进度内能成功地交付。

（2）管理规范是成功和失败的鉴别器。

（3）软件产品的高废品率和高返工率便是不成熟的象征。

美国著名的研究机构 Standish 在其著名的《CHAOS》报告中指出，美国 2001 年软件项目中的成功率仅为 28%，其中有 23%的软件半途而废，49%不成功。

这一切不正是软件产业的辛酸吗？

Jones 的结论中的第 1 项内容和第 3 项内容正是软件危机的写照，2001 年美国软件的成功率仅为 28%，同样说明美国 2001 年仍出现软件危机。

1.1.2 软件危机产生的根本原因

有人认为，20 世纪 60 年代的软件危机产生的原因是作坊生产方式，而当今的软件危机产生的原因按 Jones 的结论应是管理不规范。20 世纪 80 年代之后，世界上已有数以千计的软件公司，像微软公司这类世界级软件公司并非采用作坊式生产，这些公司（有时公司就生产管理软件）管理也并非不规范，那么产生软件危机的根本原因是什么呢？

要想了解产生软件危机的根本原因，应先了解软件分类。

1．软件分类及特点

按软件开发的原因，软件可分为专用应用软件、通用应用软件和系统软件。

1）专用应用软件

用户根据自己生产、经营或管理以及其他方面的需要，向软件商家要求为其生产的软件，均属专用软件。

专用应用软件必须突出用户的个性，只供其专用。就用户而言，除了突出个性之外，不定时希望软件商家所生产的软件能适应当前的变化和需要。

2）通用应用软件

软件开发商对市场进行长期调研后，为同类用户所开发的软件称为通用应用软件。

同类用户都具有自己的个性，这些个性正是这些用户在当今世界能生存的重要原因，为了保证软件的通用性，只得舍去各个用户的个性，只保留共性，这类软件使用起来常常不方便，众多用户常常不满意。当然，这类软件的修改和增删比前一类软件容易，但仍难做到与时俱进。

3）系统软件——软件平台

软件开发商基于降低程序和软件编写难度，由软件专业人员开发的专业软件都是系统软件。由于系统软件都是作为工具使用的，是生产软件的软件，因此当前也称为软件平台。

系统软件的最大特点是跨专业性，即所有用户都能使用，和用户专业无关，例如所有用户都能利用操作系统，方便地使用硬、软件资源。开发系统软件无须了解用户专业知识，但必须熟悉软件的专业知识。现在业界，由于开发系统软件不必了解用户专业，因而增删、修改容易，速度更快。

2．软件危机或软件成功率低下的根本原因

按原软件工程观点，产生软件危机的原因是作坊式生产，Jones 认为是管理不规范。实际上，现有不少软件生产厂家（公司）生产软件时已相当工业化，不少软件公司常常生产管理软件，这些公司以及为其生产软件的用户管理都很严格，都很规范，但软件生产仍然成功率低下，即上述原因至少不是根本原因。

在软件产业中，应用软件无论是在数量上，还是在金额上都远远大于系统软件。

若要开发专用应用软件，为了在软件中突出用户个性，必须做透彻的需求分析。由于软件人员和用户之间的专业差异需求分析很难透彻，软件无法突出用户个性，所以用户不会满意。由于社会环境和自然环境在不断变化，用户自身（包括个性）也在不断变化，当前计算机还不是人工智能计算机，软件不能随需求而变化，就必须定期增删、修改软件。修改软件之前又必须做新的需求分析，软件公司不可能长期为每个用户留下足够多的软件人员不断修改软件，用户也付不起相应经费，因此与时俱进只能是一句空话。对于通用应用软件，软件公司在开发之前，不可能与众多用户分析需求关系，只能考虑通性，不可能考虑各用户个性，因而很难使各用户都满意。20 世纪 70 年代，美国曾投入大量的人力，同时开发红色语言、灰色语言和绿色语言。红色语言尚未研制出来就宣告放弃；灰色语言虽然完成了，但也未投入使用；留下来的仅是绿色语言。实际上，现在最走红的 Java 语言也差点胎死腹中，Java 语言的原来名字也不是 Java 语言。系统软件成功率低下的主要原因是，系统软件涉及到计算学科中众多硬软件知识，设计一个系统软件必须熟悉很多相关系统软件，难度大。在 C 语言、C++语言和 Java 语言问世之前，系统软件是用汇编语言编写的，编程难度太大，因此 20 世纪 80 年代之前系统软件甚少，也没有软件平台。自 C 语言、C++语言和 Java 语言出现以后，情况则发生了根本性变化，近 15 年来，软件平台的数量至少是前 50 年的 100 倍，用户软件平台只有满意程度差而无不满意，但研制难，加上竞争剧烈因而该行业破产，倒闭的也不少。由于研制系统软件只需要本学科知识，无专业鸿沟，系统软件是软件商自己生产的，不少系统软件本身就是软件厂商的拳头产品，因而更新容易，能做到与时俱进。20 世纪 70 年代，甚至 80 年代初，系统软件大多是免费提供的，或者是和硬件配套提供的，现在却不同，不少系统软件也必须花钱买。

综合上述内容可知，应用软件和通用软件成功率低下的根本原因是，软件人员和用户之间的专业鸿沟导致了需求分析不透彻，软件厂商为用户提供软件这种生产关系难以适应用户个性的变化，无法做到与时俱进。

1.1.3 解决应用软件成功率低下的途径

自 20 世纪 60 年代初爆发了软件危机之后，不少人一直在探索、寻找解决软件危机的途径，即一直在寻找解决软件成功率低下的途径，这些途径可归结为下述三种。

1．生产工具性软件\降低程序设计和软件生产难度

这里的软件工具包括计算机语言和各种软件平台。在这方面，不少人做了不少工作，因篇幅关系不再赘述。

2．改变由软件厂商为用户生产软件这一生产关系

软件不由软件厂商生产，就得由用户自己生产。软件生产是难度相当大的工作，软件专业人员尚感吃力，用户自己能生产吗？

答案是能，这当然要涉及到软件工具问题。现在，不少软件厂商生产的软件工具的功能大都是为了降低程序设计难度，降低软件生产难度。这些工具大都是为在软件厂商名下的专业软件人员所用。若能提供为用户自己生产软件的软件工具，便可解决问题。

现在，已有软件厂商生产这类软件且已获得了成功。我国的 Justep（思维加速）公司所生产的业务架构平台就具有这个功能。该平台用于管理软件的生成，用户按照该软件的使用说明，自己编出程序并建立起相应数据库后就能自己集成软件。该公司已有几百个用户，这些用户都已开发出自己的管理软件。

用户自己开发自己的软件可避免专业差异，不存在专业鸿沟，需求分析没问题；用户自己生产的软件，更新容易，自然能做到与时俱进。

3. 研究新的软件生产方法和程序设计方法

由于本书只涉及程序设计，故只介绍程序设计。

第四代计算机语言是面向对象的程序设计语言。这类语言和第三代计算机语言的本质区别是，提供了提高程序重用率的可能性。用第四代计算机语言编写程序时，应注重程序重用率和知识重用率。结构化程序设计方法对于用第三代计算机语言编写程序来说，是一种好方法，但该方法是针对具体问题的。一个问题一个算法已不适应新语言。程序设计实际上就是将不同语言表述的算法翻译成用计算机语言表述的算法。用带计算过程和计算条件的数学公式表述算法，并对算法进行分类且和第四代计算机语言的基类库挂钩，便可提高程序的重用率和知识的重用率。

1.1.4 程序设计的重要性

第二代计算机语言问世后，程序设计大门虽然打开，但自软件产业形成之前，程序设计实际上仍只是专业软件厂家的专业软件人员的专职工作。即使是软件专业的硕士、博士，只要不在专业软件产业中工作，也都只能使用软件，而与开发软件（编写程序）无关。

业务架构平台的生产已是当前软件产业的新动向。业务架构平台兴盛之后，程序设计的大门才真正被打开。这时，程序设计将是各专业业务人员的本分工作，因此程序设计知识比以往显得更实际、更重要。

1.2 C 语言简介

下面主要介绍 C 语言的沿革、C 语言特点以及本教材的特点。

1.2.1 C 语言的沿革

C 语言是国际上广泛流行且很有发展前途的计算机高级语言，不仅用来编写应用软件，也用来编写系统软件。

在 C 语言诞生以前，操作系统及其他系统软件主要是用汇编语言实现的。由于汇编语言程序设计依赖于计算机硬件，其可读性和可移植性都很差，而一般的高级语言又难以实现对计算机硬件的直接操作，因此人们需要一种兼有汇编语言和高级语言特性的语言。C 语言就是在这种环境下产生的。

C 语言最早是由 Dennis Richie 于 1973 年设计并实现。它的产生与 UNIX 系统之间具有非常密切的联系——C 语言是在 UNIX 系统上开发的。无论 UNIX 系统本身，还是其上运行的

大部分程序，都是用 C 语言编写而实现的。同时，它同样适合于编写不同领域中的大多数程序。C 语言已经成为全球程序员的公共语言，并且由此产生了当前两个主流的语言 C++和 Java——它们都建立在 C 语言的语法和基本结构的基础上，而且现在世界上的许多软件都是在 C 语言及其衍生的各种语言的基础上开发而成。

目前，在微机上广泛使用的 C 语言编译系统有 Turbo C、Borland C++、Microsoft Visual C++等。虽然它们的基本部分都是相同的，但还是有一些差异，本书采用 Turbo C 20 作为上机编程调试环境。

1.2.2 C 语言的特点

C 语言是一种由 ALGOL 语言派生和发展起来通用流行的程序设计语言，许多大型软件均采用 C 语言编写，它同时具有汇编语言和高级语言的特性。具体地说，它具有如下特点：

（1）语言简洁，结构紧凑，使用方便、灵活。C 语言共有 32 个关键字和 9 条控制语句，且源程序的书写格式自由。

（2）运算符极其丰富，数据处理能力强。C 语言共有 45 种运算符，它把括号、赋值符号、强制类型转换符号等都作为运算符处理，使得 C 语言的运算符类型极为丰富，表达式类型多样化。灵活使用可以实现其他高级语言难以实现的运算和操作。

（3）数据结构丰富。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、公用体类型等，用它们可以实现各种复杂的数据结构（如链表、树等）。特别是指针类型，使用起来灵活多变。

（4）具有结构化的控制语句，是一种模块化的程序设计语言。例如，if...else 语句、while 语句、for 语句等，可以在程序中使用所有的控制语句。另外，函数是 C 语言的基本单位，用函数作为程序模块的基本单位，可以实现程序的模块化。

（5）可移植性好。对 C 程序基本上可以不做任何修改，就能运行在各种不同型号的计算机和各种操作系统环境上。

（6）C 语言提供了某些接近汇编语言的功能，能直接访问物理地址，直接对硬件进行操作，从而有利于编写系统软件。

这些是 C 语言的一般特点。正是由于这些特点，使得它的应用非常广泛。

在 C 语言的基础上，贝尔实验室于 1983 年又推出了 C++语言。C++语言进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C 语言是 C++语言的基础，因此在掌握了 C 语言后，再进一步学习 C++语言就更容易，并能达到事半功倍的效果。

1.2.3 C 语言程序设计教材的特点

根据 C 语言注重实践的特点和实际的教学情况，该教材具有下述特点。

（1）首次真正地在（C 语言）程序设计教材中突出讲授程序设计，而不是程序，明确地给出了程序设计主要过程：将非计算机语言表述的算法改写成带计算过程和计算条件的计算公式的过程。

（2）突出算法在计算学科中的灵魂作用。强调程序设计之前，应将解决问题的办法抽象成带有计算过程和计算条件的数学公式，即使枚举法也应该用带计算过程和计算条件的数学公式来表示枚举过程和方法。

（3）将所有带计算过程和计算条件的数学公式按基本程序模块分成了递推算法、迭代算法和尝试算法三类，使程序规范化、规律化，大大降低了程序设计难度。

（4）所有需要设计的程序都给出了设计过程，都给出了带计算过程、计算条件的数学公式以及与算式有关的重要的数据字典，使程序设计教材和软件工程并轨。

（5）所有上机实习都有实习内容、目的、过程和结果，这一切都由学生书写，上机只给题目、提示，不给程序。程序由学生编写，学生上机时，老师必须在场指导。上机实习更是突出了学生是主体，以学为主的教育思想。

（6）所有程序设计例题都配有 CAI & CAL，既可以在课堂上讲授，又可以由学生自学。

（7）C 语言程序设计不是 C 语言大全，因此教材中的一些和程序设计关系不大的函数，不在教材中介绍而放在附录中。另外，本书也未介绍有关绘图等方面的内容，有兴趣者可查阅有关资料。

（8）C 语言的功能相当强。同一结果可以由不同的语句经不同的途径实现。程序设计属工科，软件项目属工程项目。不应追求华丽，而应以实用为宗旨。因此，本书强调哪些语句、哪些格式、哪些运算符适用，哪些会引起混乱。

此外，书中还给出了几类常见的有实际意义的简化算法（用赋值语句或一 while 型语句代替原来的嵌套的条件语句），而这些算法原来用嵌套的条件语句来实现。

注：为了节省篇幅，本书用 s 表示句子，用 v 表示表达式，用 r 表示变量。

第 2 章　数据类型、运算符和表达式

数据是程序的必要组成部分，也是程序的处理对象。数据类型指明对象应具有何种数值以及可进行何种操作。变量和常量是程序处理的两种基本数据对象。在 C 语言程序中，应遵循先定义后使用的原则，程序中所使用的任何变量和数据都必须先通过说明语句定义其数据类型，然后才能使用。运算符用于指定对象进行的操作，表达式是把变量和常量通过运算符组合成一个式子，形成一个新的值。本章将介绍 C 语言提供的各种数据类型、运算符和表达式。

2.1　标 志 符

为了能够在程序中使用变量和常量，区分不同的数据对象，需要通过为不同的对象命名，通过名字来区分和使用。因此，每种程序设计语言中都规定了如何为对象命名。程序中的这些名字称为标志符，它是由字母、数字及下划线组成的序列，在为对象命名时必须满足标志符的构成规则。

C 语言中允许在标志符中出现的字符如下：

a～z，A～Z（26 个英文字母，包括大小写）

0～9（数字符号）

_（下划线）

构成规则如下：

必须由字母或下划线作为第一个符号；后面跟任意的字母、数字和下划线。

注意

- C 语言中，要区分大小写，如 name、Name、NAME 等表示不同的标志符。
- 命名时，要简洁，最好做到“见名知义”，增加程序的可读性。
- C 语言中，对标志符的长度没有限制，长度可以任意，但长度要适当。
- C 语言中有一些在系统中早已定义好的标志符——关键字，关键字不能作为一般标志符使用，如 int、float、char、double 等，因为编译程序对这些标志符已赋予了特殊的含义，已为语言本身专用，故不能使用。

2.2　C 语言中的基本数据类型

C 语言提供了下列几种基本数据类型。

（1）char：字符型。

（2）int：整型。

（3）float：单精度浮点型。

（4）double：双精度浮点型。

char、int、float、double 均为表示数据类型的关键字。此外，还可以在这些基本数据类型的前面加上一些限定符（short、long、unsigned、signed），用来扩充基本数据类型的意义，从