21世纪 高等院校计算机系列教材

C程序设计简明教程

王晓东 主 编
杨亚会 张吴波 副主编

中国水利水电出版社
www.waterpub.com.cn

21世纪高等院校计算机系列教材

C程序设计简明教程

王晓东　主　编

杨亚会　张吴波　副主编

中国水利水电出版社

内 容 提 要

本书是学习 C 语言程序设计的适用教材，全书共 12 章。前 11 章系统讲述 C 语言的基本语法、数组、函数、指针等重要知识以及常用算法和编程方法。在此基础上，第 12 章综合前面所学的知识对 C 语言在工程实践中的一些应用进行介绍。

本书注重基础、强调实践，在内容讲解上采用循序渐进、逐步深入的方法，重点突出，案例取舍得当。本书配有《C 程序设计简明教程实验指导与实训》，以方便教学。

本书适合高等学校本专科学生使用，也可作为广大软件开发人员以及工程技术人员的参考用书。

本书配有电子教案，可从中国水利水电出版社网站（http://www.waterpub.com.cn/softdown/）下载。

图书在版编目（CIP）数据

C 程序设计简明教程 / 王晓东主编. —北京：中国水利水电出版社，2006
（21 世纪高等院校计算机系列教材）
ISBN 7-5084-3872-8

Ⅰ. C… Ⅱ. 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（2006）第 075099 号

书　　名	C 程序设计简明教程
作　　者	王晓东　主　编　杨亚会　张吴波　副主编
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京蓝空印刷厂
规　　格	787mm×1092mm　16 开本　16.25 印张　392 千字
版　　次	2006 年 8 月第 1 版　2006 年 8 月第 1 次印刷
印　　数	0001—4000 册
定　　价	25.00 元

前　言

众所周知，电子计算机自诞生以来，在各个领域的应用越来越广泛，与人们的联系越来越紧密。进入21世纪，这种联系程度有进一步加剧的趋势，计算机正加速融入人们的生活。计算机作为一种强有力的工具，使人们越来越离不开它。

计算机通过执行程序来按照要求完成各种各样的工作，程序是用程序设计语言编写的。在众多的程序设计语言中，C语言无疑是其中的佼佼者，它在工程实践中得到了广泛应用。C语言既具有高级语言的全部功能，又具有与计算机硬件操作密切相关的功能。C语言具有丰富灵活的控制和数据结构，语句表达简洁而高效、程序结构清晰，具有良好的可扩充性和可移植性。由于这些特点，C语言得到了众多程序设计者的青睐。

C语言是当今世界上应用最广泛、影响最深远的高级程序设计语言之一。尽管现在是Windows平台和互联网的时代，用纯C语言编程的案例有所减少，但是学好C语言不仅有助于学习C++、Visual C++、C#等C系列后续语言，而且可以深刻理解和掌握结构化编程的思想和方法。从这个角度来看，C语言仍然没有过时，继续在向人们展示它的巨大魅力。

本书依据国家教育部本科《高级语言程序设计课程教学基本要求》编写而成，较为系统地讲解了C语言的基本概念、常用算法和结构化程序设计思想，共12章。第1章介绍程序设计概念及C语言的基本特点；第2章介绍C语言的基本数据类型、变量、运算符以及表达式；第3章介绍C语句、输入输出函数以及顺序程序设计；第4章讲解选择程序设计；第5章讲解循环程序设计；第6章讲解数组；第7章讲解函数；第8章讲解指针；第9章讲解结构体和共用体；第10章介绍位运算；第11章介绍文件；第12章介绍C语言的实际应用。前11章全面介绍了C语言的基本语法、结构化程序的基本控制结构以及数组、函数等重要知识，第12章则由几个技术专题组成，综合应用前面所学的知识，是C语言在工程实践中的高级应用。

本书采用案例教学方式，每章均以问题开始，引入语法和算法等相关知识，在解决问题的过程中将相关知识融会贯通，使学生能够迅速把握C语言编程的要领。本书结构合理、内容实用、行文顺畅、言简意赅，着重于帮助学生形成规范化的编程风格，突出程序设计中算法设计的重要地位；体现启发式教学的风格，培养学生建立C程序设计的思路和方法。

本书最突出的特色是内容紧凑、概念描述准确、举例精当、实践性强。其中实例的选取不仅数量大，而且相互呼应，自成体系，集知识性、趣味性和典型性于一身。参与编写本书的教师均具有深厚的工程背景，有着丰富的C系列语言程序设计及软件开发实践经验，书中经验之谈和心得体会比比皆是，显示了作者对C语言的深刻理解和熟练把握运用。此外本书还配有用PowerPoint和《课件大师》编写的电子教案以及配套的实验实训教材，便于教师备课和学生自学。第12章的例题在Turbo C 2.0环境下全部调试通过，其余各章所有例题均已在Visual C++ 6.0环境下调试通过，本书全部代码均可直接使用。

本书由王晓东任主编，杨亚会、张吴波任副主编。全书主要编写人员分工如下：王晓东编写第1、2、6、7、8和12章，并负责全书的统稿及定稿，王晓东、张吴波共同编写第3、4、5章，杨亚会、王晓东共同编写第9、10、11章。参加本书编写的还有：郑克忠、陈艳海、黄

连丽、邱志光、刘林、林海、熊波、卢晓、余立菊、李小波、程世平、张文生、郭宏、张宏彬、王厚林、吕进峰、张俊、吴桂生、张鹏等。

在本书的写作过程中，得到了钱新恩教授和毛高波副教授的大力支持；在修改过程中，得到了阎菲副教授的悉心指导；在书稿的校对过程中，得到了杨毅和高西林的热情帮助，在此一并表示衷心的感谢。

在本书编写过程中，参考了国内外大量的文献资料，在此特向这些文献资料的作者表示深深的谢意。由于作者水平所限，加之时间仓促，书中难免有错误之处，敬请各位专家以及广大读者不吝指教。作者 E-mail 地址是 wangxd_qy@163.com。

王晓东

2006 年 5 月

目　　录

第 1 章　概述

计算机是人类 20 世纪最伟大、最重要的发明之一，它以惊人的速度帮助人们改变了世界。进入 21 世纪，计算机在各个领域的应用越来越广泛，与人们的联系也越来越紧密。由于计算机无法识别人类的语言，人们只能通过程序与之交流，要求其完成各种各样的工作。

在众多的软件开发语言中，C 语言无疑是其中的佼佼者，在工程实践中得到了广泛应用。它既具有高级语言的全部功能，又具有与计算机硬件操作密切相关的特点。它具有丰富灵活的数据结构、简洁而高效的语句表达、清晰的程序结构、良好的可移植性等特点，从而得到了众多程序设计者的青睐。

本章主要介绍程序设计语言发展的历史、程序设计算法、C 语言的特点和基本结构，以及 C 程序的开发环境等内容，使读者对 C 语言有一个初步的感性认识。

1.1　程序设计语言

计算机是用于快速自动处理大批量数据的工具，人们借助于程序告诉计算机要处理哪些数据以及如何处理，这就是程序设计。程序设计语言就是用户用来编写程序的语言，它是人与计算机之间进行交流的工具。了解程序设计语言的发展过程，有助于加深对它的认识，把握其发展趋势。现在程序设计语言的种类繁多，发展迅速，但从其发展历史及特点来看，大致可分成以下几个阶段：

（1）机器语言。计算机是不懂人类语言的，它只能存放和识别由 0 和 1 组成的二进制数字序列，这些序列表示数据和指令。计算机和人就依靠这些二进制的数据和指令进行交流，我们称之为机器语言。在机器语言中，每条指令都用 0 和 1 组成的序列串表示，例如可以用 00000011 表示“加法”指令，用 10111111 表示“移动”指令等。

计算机可以直接执行用机器语言编写的程序，且速度很快。但机器语言的指令不直观，难以记忆而且容易写错，编写机器语言程序时，要求程序员必须非常熟悉计算机硬件结构，程序的通用性和可移植性较差。

（2）汇编语言。20 世纪 50 年代中期，为了便于理解和记忆，人们采用一些指令助记符来代替机器语言指令，例如用 add 表示“加法”指令，用 mov 表示“移动”指令等。这种指令称为汇编指令，采用汇编指令的语言称为汇编语言，用它编写的程序称为汇编程序。

汇编程序必须翻译成机器语言程序后才能执行，前者为源程序，后者为目标程序。不同的计算机其汇编指令也不尽相同，并且编写程序时同样需要对计算机硬件结构比较熟悉。

（3）高级语言。机器语言与汇编语言都是面向机器的语言，它们同属于低级语言的范畴。由于低级语言对计算机硬件的依赖性太强，要求程序员对机器结构比较熟悉，普通用户是很难胜任这一工作的。为了克服低级语言这一弱点，使人们将程序设计的精力集中在求解问题上，便出现了面向过程的程序设计语言，又称之为高级语言，例如 C 语言、BASIC 语言、Pascal 语言和 FORTRAN 语言等。这类语言接近于人的自然语言，因而设计出的程序更容易理解，

难度大大降低。例如，用 C 语言编写“5+3”的程序为：

```
main()
{
    int i,j,k;
    i=5;
    j=3;
    k=i + j;
    printf("k=%d\n",k);
}
```

计算机不能直接执行高级语言程序，必须先经过编译程序翻译成为机器语言程序后才能执行。高级语言不依赖于具体硬件，其程序设计的重心在于算法和数据结构。算法一旦确定，采用各种高级语言均可实现对同一问题的求解，因而编程的效率和质量得到了较大提高。高级语言采用结构化程序设计思想，将任务分解为一个个功能模块，确定模块之间的调用关系，并分别实现这些模块。这些特点使得人们可以较为容易地编写程序，完成各种复杂的功能。

（4）面向对象语言。随着程序的规模越来越大，用高级语言编写的程序逐渐暴露出数据与算法分离、代码重用困难等弱点，于是又出现了面向对象语言，例如 C++语言和 Java 语言等。面向对象语言采用面向对象的编程思想，程序由一个个对象组成，对象具有属性和行为，对象之间通过消息相互联系。具有相同属性和行为的对象抽象成类，可以通过派生的方式产生新的类。面向对象语言适于开发较大规模的软件，程序可读性强、安全性高、较易维护和扩充，目前是软件开发的主流语言之一。

1.2 程序设计与算法

1.2.1 算法的概念

著名的计算机科学家沃思（N • Wirth）曾提出过一个经典的公式：程序=数据结构+算法。这个公式的重要性在于说明了程序与数据结构以及算法的关系。

所谓算法，就是一个有穷规则的集合，它确定了一个解决某个特定类型问题的运算序列。简单地说，就是为解决一个具体问题而采取的确定有限步骤。编写一个程序的关键就在于合理地组织数据和设计算法。解决一个问题可能会有多种算法，例如关于排序算法就有冒泡排序法、选择排序法和插入排序法等。可以用不同的算法以及不同的程序设计语言来解决同一个问题，只是速度和效率上有所不同而已。

一般地说，算法具有如下特性：

（1）有穷性。算法包含的操作步骤应是有限的，每一步都应在合理的时间内完成。

（2）确定性。算法的每个步骤都应是确定的，不允许有歧义。

（3）有效性。算法的每个步骤都应能有效地执行，且能得到确定的结果。

（4）有输出。算法的实现是以得到计算结果为目的，没有输出结果的算法是没有意义的。

1.2.2 结构化程序设计

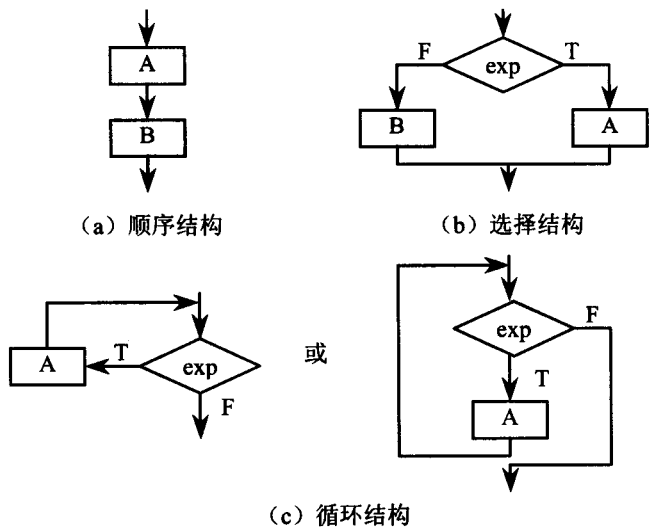
结构化程序设计的概念最早由 Dijkstra 于 20 世纪 60 年代提出。1966 年有人证明，只用三

种基本的控制结构就能实现任意结构的算法，这三种基本结构就是顺序结构、选择结构和循环结构，它们的流程图如图 1-1 所示。

（1）顺序结构。各个操作是按书写的顺序执行的，从操作序列的第一个操作开始，顺序执行序列的所有操作，直至序列的最后一个操作。如图 1-1（a）所示，先执行 A，然后执行 B。

（2）选择结构。对指定的条件进行判断，根据判断的结果在两条分支路径中选取其中一条执行。如图 1-1（b）所示，表达式（exp）的取值为“真”（即 T）时执行 A，为“假”（即 F）时执行 B。

（3）循环结构。根据给定的条件是否满足，来决定是否继续执行循环中的操作，如图 1-1（c）所示。



（a）顺序结构　（b）选择结构

（c）循环结构

图 1-1　三种基本的控制结构

三种控制结构为结构化程序设计技术奠定了理论基础，结构化程序设计技术作为一种新的程序设计思想和方法逐渐得到人们的普遍认可，并成为一种规范。它采用自顶向下逐步求精的设计方法和模块化的程序结构，并且每个模块只包含顺序、选择和循环三种控制结构。

逐步求精方法是由 Wirth 提出的一种自顶向下的设计策略，面对现实的复杂问题，首先不触及问题解法的细节，而是先从全局出发，用较自然的抽象语句来表示问题，从而得到抽象算法。接下来对抽象算法进行细化，在这一阶段设计的算法中，已经开始含有程序设计语言的成分。随着算法的不断细化，越来越多地开始实现“如何做”，算法中程序设计语言的成分也越来越多，当最后把算法全部细化为程序设计语言描述时，程序设计也就随之完成了。

实践证明，采用自顶向下逐步求精的方法，符合人类解决复杂问题的普遍规律，可以大大提高软件开发的效率。由于采取了先全局后局部、先整体后细节、先抽象后具体的逐步求精方法，使得开发出来的程序层次结构清晰，因此容易阅读和理解，方便测试与维护。

1.2.3 算法的描述方法

在进行算法设计时，可以使用一些算法描述工具表示算法，常用的描述工具有自然语言、传统流程图、N-S 图和伪代码等。

1. 自然语言

用自然语言描述算法时，可使用汉语、英语和数学符号，这比较符合人们日常的思维习惯，且通俗易懂。但是使用自然语言描述的算法在表达上容易出现疏漏，可能引起理解上的歧义，不易直接转化为程序，因此自然描述语言只适用于算法比较简单的情况。

2. 程序流程图

程序流程图是历史最悠久、使用最广泛的描述软件设计的方法，由于它独立于任何一种程序设计语言，比较直观，易于学习掌握，因此至今仍是软件开发者普遍采用的一种工具。

图 1-2 是程序流程图中常用的一些符号。

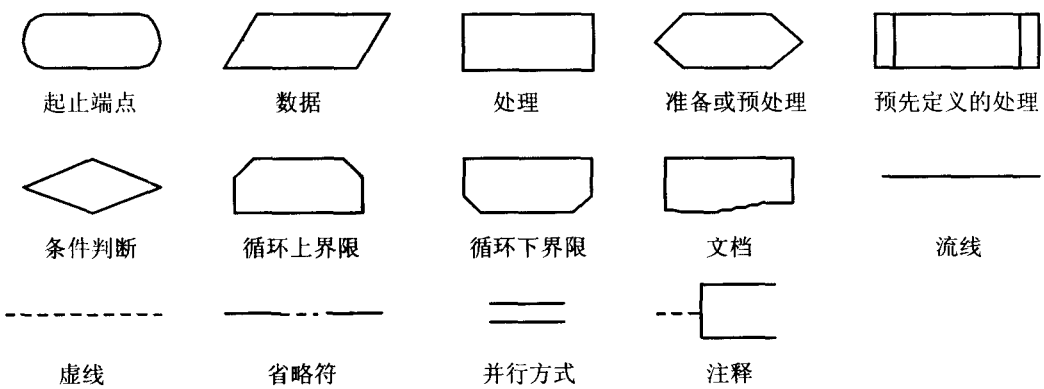


图 1-2 程序流程图中常用的符号

程序流程图用平行四边形表示数据，平行四边形内可注明数据名称、来源、用途或其他的文字说明；用矩形框表示处理，框内可注明处理名称或其主要功能；用一个菱形框表示判断，菱形框内可注明判断的条件，通过对条件进行判断，从而决定执行哪个分支；用带有双竖边线的矩形表示预先定义的处理，例如库函数或其他已定义的函数等，矩形内可注明特定处理名称或其主要功能；用流线表示执行的流程，箭头代表执行的方向。

下面用流程图描述表达式 1+2+3+4+ 5 的算法，如图 1-3 所示。

程序流程图虽然直观灵活且比较容易掌握，但由于其随意性较大，使得不可避免地存在着一些缺点：

（1）容易使程序员过早地考虑程序的具体控制流程，而忽略程序的全局结构。

（2）程序员可以不受任何约束，随意转移控制流程的方向。

（3）在表示数据结构方面存在不足。

3. N-S 图

为了限制程序流程图的随意性，1973 年，Nassi 和 Shneiderman 提出用盒图来代替传统的程序流程图，因此又称为 N-S 图。同程序流程图相比，N-S 图以一种结构化的方式严格限制从一个处理到另一个处理的控制转移。N-S 图有以下一些特点：

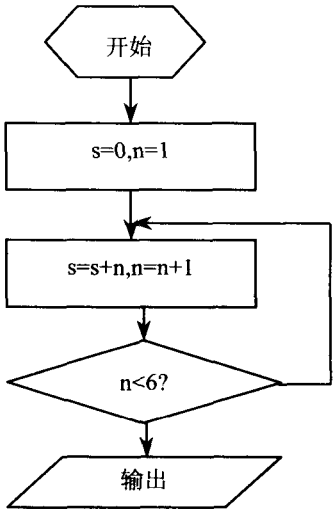


图 1-3　描述 1+2+3+4+5 的算法流程图

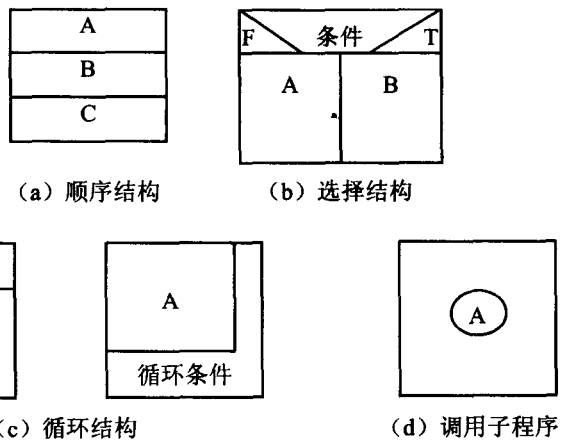
（1）功能域（即某一个特定控制结构的作用域）有明确的规定，并且较为直观。

（2）它的控制转移不能任意规定，必须遵守结构化程序设计的要求。

（3）较容易确定局部数据和全局数据的作用域。

（4）较容易表现嵌套关系，也可以表示模块的层次结构。

N-S 图的基本元素是矩形块，用它表示的结构化控制结构如图 1-4 所示。图 1-4（a）表示顺序结构，图 1-4（b）表示选择结构，图 1-4（c）表示循环结构，图 1-4（d）表示调用子程序。N-S 图保留了程序流程图形象直观的特点，但消除了容易导致程序非结构化的流线。



（a）顺序结构　　（b）选择结构

（c）循环结构　　（d）调用子程序

图 1-4　N-S 图的基本符号

坚持使用 N-S 图作为程序设计的工具，可以使程序员逐步养成用结构化的方式思考问题和解决问题的习惯，但是 N-S 图的修改不太方便。

4. *伪代码*

伪代码是介于自然语言和计算机语言之间的一种代码，是帮助程序员制定算法的智能化语言。它不能在计算机上直接执行，但是使用起来比较灵活，无固定格式和规范，只要写出来自