C语言程序设计

吴定雪　主编

科学出版社
www.sciencep.com

C语言程序设计

[redacted]

C语言程序设计

吴定雪　主编

科学出版社

北京

内 容 简 介

全书共分为10章，全面、系统地介绍C语言程序设计基础及应用知识，包括C语言程序设计基础，C语言程序设计基本结构，函数与编译预处理，数组，指针，结构体、共用体与枚举类型，位运算，文件，实用程序设计技巧以及C++面向对象程序设计。附录A～C中提供了相应的常用资料，以方便师生查阅相关内容。

本书可作为本科院校计算机专业及理工科类非计算机专业的教材，也可作为计算机等级考试(二级)的自学教材或参考用书，还可作为有关工程技术人员和计算机爱好者学习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/吴定雪主编.—北京：科学出版社，2007
ISBN 978-7-03-020786-9

Ⅰ.C… Ⅱ.①吴… Ⅲ.C语言－程序设计－水平考试－自学参考资料
Ⅳ.TP312

中国版本图书馆CIP数据核字（2007）第204795号

责任编辑：张颖兵 吉正霞／责任校对：梅 莹
责任印制：董 丽／封面设计：苏 波

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
武汉市新华印刷有限责任公司印刷
科学出版社发行 各地新华书店经销
*
2007年12月第 一 版 开本：787×1092 1/16
2007年12月第一次印刷 印张：19 1/2
印数：1—3 000 字数：441 000
定价：29.50元
（如有印装质量问题，我社负责调换）

《C语言程序设计》编者名单

主　编　吴定雪

副主编　郭峰林　汤恒耀

编　委　（按姓氏笔画为序）

朱泽民　汤恒耀　肖小红　吴定雪　何中林

张瑞红　范文萍　周　静　郭峰林　涂焱楚

前　言

C语言是目前世界上最流行、使用最广泛的高级程序设计语言。它不仅具有丰富灵活的控制和数据结构、简洁而高效的表达式语句、清晰的程序结构和良好的可移植性等优点，还具有直接操纵计算机硬件的强大功能。因此，C语言既适合开发系统程序，又适合开发应用程序，深受广大计算机应用人员青睐。目前，高校的计算机专业、绝大多数理工科院校非计算机专业都将“C语言程序设计”作为计算机程序设计语言的首选语言。

我们认为学习程序设计既要熟练地掌握和使用编程语言，又要学会掌握程序设计方法，后者是程序设计的难点，也是重点，因为程序设计是一项严密逻辑思维活动，可以完全独立于具体的编程语言，不受它所依托的具体语言的限制。程序设计涉及很多认识上的技巧，例如，对操作环境和相关开发工具的熟悉，对数据结构、算法的合理运用和测试，对机器内部工作的了解等。因此，我们在介绍程序设计语言时，如果只侧重语法知识介绍就会给读者一个误导，使读者误以为学习程序设计就是记住那些语法规范而已，而忽略了对程序设计方法的训练、对良好程序设计习惯的培养，导致学习完C程序设计课后，仅仅学会了一些C语言的语法，编写不出简单的应用程序，读懂别人编写的简单程序也存在困难。

针对实用型、应用型人才培养特点，我们根据多年在高校从事计算机程序设计语言教学的经验，在分析课程特点和难点的基础上，对教学内容进行了合理的重组，既强调基本知识点，又注重各知识点之间的关联，编写上充分考虑C语言的特点、程序设计课程教学规律、读者群等，注重陈述的生动与通俗，内容紧凑，难度适宜，结构循序渐进，秉承“学得会、用得上”的宗旨，精心编写了这本《C语言程序设计》教材。其与现有的C语言教程相比有以下几个特点：

1. 该书吸取了其他众多同类C语言教材的优点，章节安排由浅入深、循序渐进。重视编程能力培养，围绕结构化与模块化程序设计这个中心，以程序设计为主线，在深入浅出地介绍C语言的基本语法规则的同时，通过精心设计的例题与习题，着重介绍C语言程序设计的基本方法，加强了结构化程序设计和常用算法的训练。这样既使读者掌握C语言基础知识，又能掌握程序设计的基本方法。

2. 突出应用与适用。本书以简短的篇幅介绍C语言中最基本、最常用的内容，同时精心设计了一些C语言的编程实例，对所讲述的原理、概念加以辅助说明，学生可以通过这些实例加深对C语言编程的基本原理、方法的掌握与理解。学生通过实例分析，并加以编程练习，既掌握了C语言的内容，获得了开发实用软件的训练，更激发了探索C语言奥妙的兴趣，能达到事半功倍的效果。

3. 注重改革实践教学。C语言程序设计是一门实践性很强的程序课，上机实验是该课程的重要教学内容，学习程序设计是听不会、看不会的，只有通过大量的编程练习，才能在实践中掌握语言知识，培养程序设计实践能力。因此本教材的配套实验教程专门针对这一特点，精心设计了实验项目与实验内容，针对性强。学生可在实验任务的驱动下，进

行操作，并能将完成的结果与系统要求达到的结果进行比较，从而提高实际编程能力。

4. 多元化“立体化”教材建设。本书为了方便教学与学习，还配套编写了相应的实践教程。

5. 为提高学生的程序设计技巧，本书精心编写了实用程序设计技巧（第 9 章）这一章，引入软件工程的思想，介绍程序的模块化设计、模块的设计风格和书写风格，然后讨论了大型 C 语言程序的执行方法，这部分内容为学生应用 C 语言编写一些适用程序做了很好的铺垫。

6. 为开阔学生的视野，本书还介绍了 C++面向对象程序设计基础（第 10 章），这部分内容为学生进一步学习 C++或 VC++打下基础。这些内容为选学内容，教师可根据实际情况选讲或指导学生自学。

另外，每一章后附有精心挑选和设计的多种类型的思考题与练习题，有助于读者复习、巩固所学知识。

本书可作为应用型本科院校计算机专业及理工科类非计算机专业学生学习 C 语言程序设计的教材，还可作为有关工程技术人员和计算机爱好者学习 C 语言程序设计的参考书。

本书由在教学第一线并具有丰富计算机程序设计教学经验的多位教师共同编写。全书由吴定雪任主编，郭峰林、汤恒耀任副主编，其中第 2、10 章及附录 A～C 由汤恒耀编写；第 4 章由何中林编写；第 7、8 章由范文萍编写；第 1、9 章由肖小红编写；第 5 章由张瑞红编写；第 3、6 章由周静编写，最后由吴定雪统定全稿。

本书在编写过程中得到了学校各级领导和教务处以及计算机科学与技术学院全体教师的大力支持，在此表示衷心感谢。还要感谢科学出版社的各级领导和编辑们对我们编写的教材的精心策划、组织和编辑！同时，也对编写过程中所参阅的大量文献的作者致以谢意。限于作者学识水平，研究工作还不够深入，本书难免有疏漏和不妥之处，诚请读者和同行批评指正。

我们的电子邮件地址：computer@hgnc.net；网站地址：http://www.hgnc.net/yuanxi/jsjx/。

目　录

第1章　C语言程序设计基础

电子计算机自从20世纪40年代诞生以来，无论在硬件还是在软件方面，都有了极大的发展，在计算机应用的各个领域也都取得了丰硕的成果。

计算机本身是无生命的机器，要使计算机能够运行起来，为用户解决问题，就必须利用某种软件。例如，编辑文件时使用Word，处理电子表格时使用Excel，浏览网络资源时使用IE，上网聊天时使用QQ等。实际上这些软件都是按照一定算法用某种程序设计语言编制的计算机程序，其中算法是为解决一个问题而采取的方法和步骤，当它用计算机能够识别的某种计算机语言实现出来后，就是所谓的程序。C语言就是一种计算机语言，用C语言编制的程序就称为C程序。

程序设计语言有很多，在众多的程序设计语言中，C语言有其独特之处。一方面，它作为一种高级程序设计语言，具有方便性、灵活性和通用性等特点；同时它还具备低级语言的特征，向程序员提供了直接操作计算机硬件的功能，可以用来编制系统软件，如操作系统，像UNIX操作系统就是用C语言来开发的。C语言的这两大特征使得它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。

下面将要学习的就是用C语言编写计算机程序的一些基础知识。

1.1　程序设计基本概念

1.1.1　程序

程序一词也来自生活，通常指完成某些事务的一种既定方式和过程。从表述方面看，可以将程序看成是对一系列动作的执行过程的描述。日常生活中也可以找到许多“程序”实例。例如，大会议程、按程序办事。又如，一个学生早上起床后的行为可以描述为：起床→刷牙→洗脸→吃饭→早自习。

程序具有如下的两个直观特征：①在一个程序描述中，总有一批预先假定的“基本动作”，这些基本动作是执行程序者能够理解和直接完成的；②一个程序总有开始与结束，在执行此程序的过程中，执行者（无论是不是人）需要按照程序的描述执行一系列的动作，直到程序结束。

为了让计算机能够按人的意图处理事务，人们必须事先设计好完成各种任务的程序，并预先将它们存放在存储器中。在计算机中的程序，指的是用计算机语言描述的某一问题的解决步骤，是符合一定语法规则的符号序列。这里计算机能够处理的语言，即是程序设计语言，处理的对象即是数据结构，解决步骤即是算法。因此，关于程序还有一种公式化的描述：

程序＝算法＋数据结构＋程序设计语言

日常生活中的程序性工作中有更多变数，许多事情并不要求完全按程序做，可以有许多“灵活性”；但计算机对程序的执行则完全是严格的，一步步按程序中的指令办事，一点

“商量”的余地也没有。

1.1.2 程序设计语言

从交流的角度看“语言”一词，通常指人在生活、工作中使用的自然语言，如汉语、英语等。这些语言随着人类发展进步而自然形成，是人相互间交流信息的工具和媒介。在前面所描述的现实生活中的程序实例中，就是用汉语作为描述程序的语言，所描述的程序是为了给人看，要人去做的。为了实现与计算机的交流，也需要有一种计算机能够识别的语言，这就是程序设计语言或计算机语言。通过用程序语言写程序，人能指挥计算机完成各种特定工作，完成各种计算。程序设计语言是人与计算机交流信息的媒介。

从编程的角度看，程序设计语言是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并能精确地定义在不同情况下所应当采取的行动。

当今程序设计语言的发展非常迅速，新的程序设计语言层出不穷，其功能越来越强大；但不管现代程序设计语言的功能如何增强，程序设计语言的种类怎样增多，从其发展历史以及功能情况看，大致可分为如下几个阶段。

1. 机器语言

一组由 0 和 1 序列构成的指令码。下面是某 CPU 指令系统中的两条机器语言指令：

10000000　加

10010000　减

用机器语言编程序，就是从所使用的 CPU 的指令系统中挑选合适的指令，组成一个指令序列。机器语言是一种面向机器的语言，其指令集依 CPU 的不同而异。用机器语言编写的程序，虽然可以被机器直接理解和执行，并能控制计算机的硬件，却由于它们不直观，难记、难认、难理解、不易查错，只能被少数专业人员掌握，同时程序的生产效率很低，质量难以保证，又是面向具体机器的，不具备可移植性。

2. 汇编语言

20 世纪 50 年代中期，为了减轻人们使用机器语言编程的负担，开始采用一些助记符号来表示机器语言中的机器指令，这样便形成了汇编语言。助记符一般都是采用代表某种操作的英文字母的缩写，与机器语言相比，便于识别和记忆。例如，上例中的两条指令用汇编语言描述如下：

```
ADD A, B
SUB A, B
```

不过计算机不能直接执行用汇编语言编写的程序，它必须经过一个叫汇编程序的系统软件翻译成机器语言程序后才能执行。通常将用汇编语言编写的程序称为源程序，而翻译后的称为目标程序。

汇编语言指令与机器语言指令基本上具有一一对应的关系，因此不同的计算机其汇编语言指令也不尽相同，所以汇编语言与机器语言一样，也是面向机器的语言。用面向机器的语言编程，可以编出效率极高的程序，但是程序员用它们编程时，不仅要考虑解题思

路，还要熟悉机器的内部结构，并且要“手工”地进行存储器分配。这种编程的劳动强度仍然很大，给计算机的普及推广造成很大的障碍。

3. 高级语言

1954 年出现的 FORTRAN 语言以及随后出现的其他高级语言，使人们开始摆脱进行程序设计必须先熟悉机器的桎梏，把精力集中于解题思路和方法上，使程序设计语言开始与解题方法相结合。

其中一种方法是把解题过程看作是数据被加工的过程。基于这种方法的程序设计语言称为面向过程的程序设计语言。C 语言就是一种面向过程的程序设计语言。面向过程的程序设计认为，每个程序都是完成一些规定的功能。每个功能的实现是通过对数据进行一系列的加工的过程而实现的，因而程序设计包括组织数据——设计数据结构，以及设计对数据结构进行加工的过程——设计算法两个部分。这类程序设计语言接近自然语言和数学表达式，面向问题，移植性强，易阅读和修改。

还有一种方法是面向对象的方法，面向对象方法的本质，就是主张从客观世界固有的事物出发来构造系统，提倡用人类在现实生活中常用的思维方法来认识、理解和描述客观事物，强调最终建立的系统能够映射问题域，也就是说，系统中的对象及对象之间的关系能够如实地反映问题域中固有事物及其关系。它是一种结构模拟方法，它把现实世界看成是由许多对象所组成，对象之间通过互相发送和接收消息进行联系；消息激发对象本身的运动，形成对象状态的变化。从程序结构的角度，每个对象都是一个数据和方法的封装体——抽象数据类型。面向对象的程序比面向过程的程序更清晰、易懂，更适宜编写更大规模的程序，正在成为当代程序设计的主流。面向对象的程序设计语言有 Simmla-67、Smalltalk80、Java 等。

1.1.3 程序设计

简单地说，人们借助计算机能够处理的语言，告诉计算机要处理什么（即处理哪些数据）以及如何处理（即按什么步骤来处理），这便是程序设计。程序设计更为全面的理解是指设计、编制、调试程序的方法和过程。它是目标明确的智力活动。由于程序是软件的主体，软件的质量主要是通过程序的质量来体现的，在软件研究中，程序设计的工作非常重要，内容涉及有关的基本概念、工具、方法以及方法学等。

注意：上面提到了程序设计语言、程序设计以及程序三个概念。它们的关系是：程序设计的结果用程序设计语言描述出来即是程序。可以这样打个比方：程序设计好比是写作，程序设计语言好比是写作的语言（可以用英语、汉语或日语进行写作），而程序则好比是最终写出来的文章。此外，还有一个概念程序员，即专门从事编程的人，这就好比是专门从事写作的作家。

1.1.4 算法

1. 算法的概念

前面在讲程序时，提到了这样的一个公式：程序＝算法＋数据结构＋程序设计语言。这里的算法，简单地说就是进行操作的方法和步骤。从广义的角度说，算法早就融入现实

生活中。比如，早上上学，就开始了上学的算法：走哪条路，坐什么车，如果堵车怎么办等，一直到到达学校，这个算法就完成了。用计算机解决问题也是按照相应的步骤（算法）一步一步完成的。这些步骤处理的对象是数据，实现用的是计算机语言。下面给出的是一简单算法示例：

例 1-1 求 5！的值。

算法如下：①先求 1×2，得到结果 2；②将步骤 1 的结果×3；③将步骤 2 的结果×4；④将步骤 3 的结果×5。

一般来说，不同的工作，采用的方法和步骤也不同。对于同一个问题可以有不同的解题方法和步骤，也就是有不同的算法。算法有优劣，通常情况下应当选择简单的，运算步骤少的，运算快、内存开销小的算法。

2. 算法的特性

（1）有穷性：一个算法应当“在合理范围之内”的有限步骤内结束，而不能是无限的；同时应当在执行一定数量的步骤后，算法结束。

（2）确定性：算法中的每一个步骤都不应当产生歧义，其含义应当是确定的。例如：“将成绩优秀的同学名单打印输出”就是有歧义的，“成绩优秀”的含义不明确。

（3）有 0 个或多个输入：所谓输入是指算法执行时从外界获取必要的信息（可以是从键盘输入的数据，也可以是其他部分传递给算法的数据）。一个算法可以没有输入，也可以有输入。

（4）有 1 个或多个输出：算法必须得到结果，没有结果的算法是没有意义的（结果可以显示在屏幕上，也可以传递给程序的其他部分或文件）。

（5）有效性：算法的每个步骤都应当能有效执行，并能得到确定的结果。例如：b＝0，则 a/b 是不能有效执行的。

3. 算法的表示方法

描述算法有很多种方法，一般有自然语言、流程图、N-S 图、伪代码、程序语言等，最常见的是流程图和 N-S 图。下面进行简单的介绍。

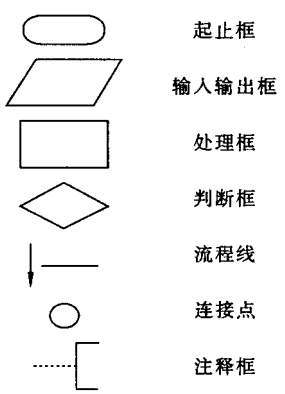

图 1-1 常用流程图符号

1）用自然语言表示算法

如上述的例 1-1，使用人类能理解的自然语言来描述求解问题的步骤，可以是汉语、英语或其他语言。用自然语言表示通俗易懂，但文字冗长，容易出现“歧义性”。自然语言表示的含义往往不太严格，要根据上下文才能判断其正确含义。此外，用自然语言描述包含分支和循环的算法，不很方便。因此，除了很简单的问题以外，一般不用自然语言描述算法。

2）用流程图表示算法

流程图用一些图框表示各种操作，用箭头表示算法流程。这种用图形表示算法的方法，直观形象，易于理解。

美国标准化协会 ANSI 规定了一些常用的流程图符号，

如图 1-1 所示，已为世界各国程序工作者普遍采用。

起止框：表示算法的开始和结束。一般内部只写“开始”或“结束”。

输入输出框：表示算法请求输入需要的数据或算法将某些结果输出。一般内部常常填写“输入…”或“打印/显示…”。

处理框：表示算法的某个处理步骤，一般内部常常填写赋值操作。

菱形框(判断框)：作用是对一个给定条件进行判断，并根据给定的条件是否成立来决定如何执行其后的操作。它有一个入口，两个出口。

连接点：用于将画在不同地方的流程线连接起来。同一个编号的点是相互连接在一起的，实际上同一编号的点是同一个点，只是画不下才分开画。

注释框：注释框不是流程图中必须的部分，不反映流程和操作，它只是对流程图中某些框的操作做必要的补充说明，以帮助阅读流程图的人更好地理解流程图的作用。

C 语言是一种结构化程序设计语言，结构化程序有顺序结构、选择结构和循环结构三种基本结构。

三种结构的流程图表示如图 1-2 所示。

例 1-2　用流程图表示求 5！的算法，流程图如图 1-3 所示。

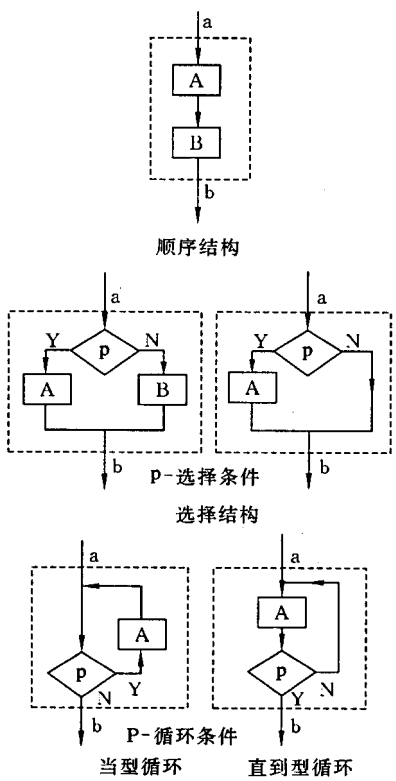

图 1-2　三种基本程序结构的流程图表示

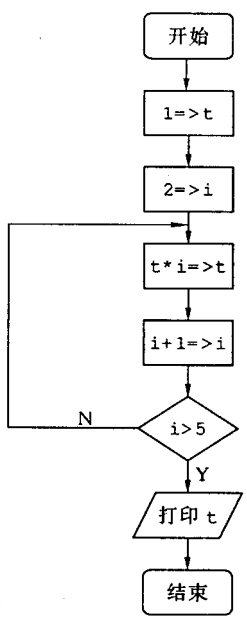

图 1-3　求 5！的算法流程图

用流程图表示的算法直观形象，比较清楚地显示出各个框之间的逻辑关系，因此得到广泛使用；但这种流程图占用篇幅较多，尤其当算法比较复杂时，画流程图既费时又不方