面向21世纪

高等职业技术教育计算机类规划教材

C语言程序设计教程

主编　朱接文　　主审　肖伟东

西安电子科技大学出版社

http://www.xduph.com

面向21世纪高等职业技术教育计算机类规划教材

C语言程序设计教程

朱接文　主编

肖伟东　主审

西安电子科技大学出版社

2007

内 容 简 介

C 语言是目前广泛使用的通用程序设计语言之一，也是许多计算机专业人员和计算机爱好者学习程序设计的首选语言。

本书以程序设计为主线，系统介绍了 C 语言程序设计的基本知识、C 语言的基本数据类型和数据运算、程序控制结构、数组、指针、函数、结构和联合、文件、编译预处理等，并通过丰富的程序设计实例，来使读者获得程序设计的一般思路和实际编程的能力。每章都安排了大量的习题，以帮助读者检测 C 语言知识的掌握程度，提高程序设计能力。

本书可作为大专院校“C 语言程序设计”课程的教材及全国计算机等级考试(C 语言)培训班的教材，也可供从事程序开发的工程技术人员参考。

★ 本书配有电子教案，有需要的老师可与出版社联系，免费提供。

图书在版编目(CIP)数据

C 语言程序设计教程 / 朱接文主编. —西安：西安电子科技大学出版社，2007.5

面向 21 世纪高等职业技术教育计算机类规划教材

ISBN 978-7-5606-1823-4

Ⅰ.C… Ⅱ.朱… Ⅲ.C 语言—程序设计—高等学校：技术学校—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2007)第 051887 号

策　　划　毛红兵

责任编辑　雷鸿俊　毛红兵

出版发行　西安电子科技大学出版社(西安市太白南路 2 号)

电　　话　(029)88242885　88201467　邮　编　710071

http://www.xduph.com　E-mail: xdupfxb@pub.xaonline.com

经　　销　新华书店

印刷单位　西安文化彩印厂

版　　次　2007 年 5 月第 1 版　2007 年 5 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　印　张　15.875

字　　数　370 千字

印　　数　1～4000 册

定　　价　20.00 元

ISBN 978-7-5606-1823-4/TP・0950

XDUP 2115001-1

前　言

C 语言是目前国内外广泛使用的程序设计语言之一。C 语言功能丰富、表达能力强、使用灵活方便、程序执行效率高、可移植性好，它的高级语言形式、低级语言功能具有独特的魅力，使用其进行程序设计已成为软件开发的一个主流。如今，大多数高职院校都将 C 语言作为典型的计算机教学语言。

为适应当前高职院校注重培养应用型人才的需求，本书依据高职院校教学大纲以及作者的实际开发经验组织内容，注重理论的严谨性和完整性，书中所选案例丰富、实用性强，力求使学生在掌握 C 语言的同时获得程序开发的基本思路并得到一定程度的项目开发实际训练，以培养学生独立开发较为复杂系统的能力。

本书作为 C 语言程序设计的入门教材，共分 11 章，主要内容包括：C 语言概述、C 语言程序设计的初步知识、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、构造数据类型及文件。全书在内容安排上遵循深入浅出、由简到繁、循序渐进的原则，重点是让学生了解程序设计语言的基本知识、掌握程序设计的基础方法和常用算法，使学生具有使用 C 语言程序设计解决实际问题的初步能力。

本书语言表达严谨，逻辑性强，示例丰富。书中例题均在 Turbo C 2.0 开发环境下调试通过，对于其算法都有具体的分析，对于其关键性语句都有详细的注释。各章后的习题中有很多历年的等级考试题目，可方便各类人员自学及备考。

本书由江西工业工程职业技术学院朱接文主编。江西工业工程职业技术学院肖伟东老师在百忙之中主审了全书，在此深表谢意。

由于作者水平有限，书中疏漏与不足之处在所难免，恳请广大读者批评指正。

编　者

2007 年 1 月

目　录

C语言概述

电子计算机自 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，为人类完成各种各样的工作，就必须让它执行相应的程序。这些程序都是由程序设计语言编制而成的。

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。

本章主要从程序设计的角度出发，结合 C 语言的特点和发展，介绍有关程序设计的基本概念，以及 C 语言程序的基本结构、上机调试、算法等内容。

1.1 C语言的发展及特点

1.1.1 C语言的发展

C 语言是当今社会应用广泛，并受到众多用户欢迎的一种计算机算法语言。它既可作为系统软件的描述语言，也可用来开发应用软件。

C 语言的出现是与 UNIX 操作系统紧密联系在一起的，C 语言本身也有一个发展过程，且目前仍然处于发展和完善之中。

从历史发展来看，C 语言起源于 1968 年发表的 CPL 语言(Combined Programming -Language)，它的许多重要思想来自于 Martin Richards 在 1969 年研制的 BCPL 语言，以及以 BCPL 语言为基础的、由 Ken Thompson 在 1970 年研制的 B 语言。Ken Thompson 用 B 语言编写了第一个 UNIX 操作系统，应用在 PDP-7 计算机上。D. M. Ritchie 1972 年在 B 语言的基础上研制出了 C 语言，并用 C 语言编写了第一个在 PDP II 计算机上实现的 UNIX 操作系统。1977 年出现了独立于机器的 C 语言编译文本——可移植 C 语言编译程序，从而大大简化了把 C 语言编译程序移植到新环境所需做的工作，也使得 UNIX 操作系统迅速地在众多机器上得以实现。例如 VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的广泛使用，C 语言也迅速得到推广。

1983 年，美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本，对 C 语言的发展和扩充制定了新的标准，称为 ANSI C。1987 年，ANSI 又公布了新标准——87ANSI C。

目前在微型计算机上使用的有 Microsoft C、Quick C、Turbo C 等多种版本。这些不同的 C 语言版本的基本部分是相同的，只在有关规定上略有差异。本书在 Turbo C 2.0 环境

下对 C 语言进行介绍。Turbo C 是一种快速、高效的编译程序。它不仅提供了一个集成开发的环境，同时也按传统方式提供了一个命令行编译程序版本，以满足不同用户的需要。

1.1.2 C 语言的特点

事实证明，C 语言是一种极具生命力的语言，它的特点是多方面的，一般可归纳如下：

(1) 语言简练、紧凑，使用方便、灵活。C 语言总共只有 34 个关键字，9 种控制语句，程序编写形式自由，对大小写敏感。C 语言的这些特点使得编程者的个性易于发挥，从而写出更高效的程序。

(2) 运算符丰富。C 语言共有 34 种运算符，括号、赋值、强制类型转换都以运算符形式出现，从而使 C 语言的表现能力和处理能力极强，使 C 比其他语言更容易实现算法。

(3) 数据结构丰富。C 的数据类型有基本数据类型和构造数据类型，能够方便地实现各种复杂的数据结构。

(4) 是一种结构化语言。C 语言采用函数作为程序的基本单位，易于做到层次清晰，便于按照模块化方式组织程序的控制流语句(if-else 语句、switch 语句、while 语句、for 语句)。

(5) 语法灵活，限制不十分严格。C 语言中的整数类型数据、字符型数据和逻辑型数据可以通用，对数组下标不做越界检查，这使程序员有很大的发挥空间。但另一方面，放松语法检查也容易引起语法错误。

(6) 可以直接访问内存的物理地址，进行位(bit)一级的操作。由于 C 语言实现了对硬件的编程操作，它集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适用于应用软件的开发，因此 C 被广泛地移植到了各类单片机上。

(7) 效率高，可移植性强。

1.2 C 语言程序的基本结构

下面通过一个简单的程序例子，介绍 C 程序的一些基本结构，使读者对 C 语言程序有一个初步的了解。

【例 1.1】 输入矩形的两条边长，求矩形的面积。

程序如下：

```
#include <stdio.h>
main()
{
  float a, b, s;          /*定义 a, b, s 三个变量*/
  a=1.2;                  /*给矩形的两条边赋值*/
  b=3.6;
  s=a*b;                  /*计算矩形的面积，赋值给 s*/
  printf("s=%f\n", s);    /*输出矩形的面积*/
}
```

以上程序的运行结果如下：

s=4.320000

例 1.1 的程序中，main 是主函数名，C 语言规定必须用 main 作为主函数名。其后的一对圆括号中间可以是空的，但这一对圆括号不能省略。程序中的第 2 行是主函数的起始行。一个 C 程序可以包含任意多个不同名的函数，但必须有一个而且只能有一个主函数。一个 C 程序总是从主函数开始执行。

在函数的起始行后面是函数体。函数体用左花括号“{”开始，用右花括号“}”结束。其间可以有定义(说明)部分和执行部分。例 1.1 程序中的第 4 行就是程序的定义部分；第 5～8 行是程序的执行部分，执行部分的语句称为可执行语句，必须放在说明部分之后，语句的数量不限。程序中由可执行语句向计算机系统发出操作指令。

定义语句用分号结束，在例 1.1 程序中只有一个定义语句，对程序中所用到的变量 a、b、s 进行定义并且说明它们为 float 类型。

程序的第 5、6 行用两条语句分别给两条边赋值，第 7 行计算出矩形面积并赋给变量 s，第 8 行按设计的格式把 s 的值输出到终端屏幕。C 程序中的每一条执行语句都必须用分号结束，分号是 C 语句的一部分，不是语句之间的分隔符。

在程序中可以对程序进行注释，注释部分必须用符号“/*”和“*/”括起来。“/*”和“*/”必须成对出现，“*”和“/”之间不可以有空格。注释可以用英文，也可以用中文。注释可以出现在程序中任意合适的地方。注释部分对程序的运行不起作用。在注释中可以说明变量的含义或程序段的功能，以便帮助人们阅读程序。因此，一个好的程序应该有详细的注释。

C 语言程序有比较自由的书写格式，但是过于“自由”的程序书写格式，往往使人们很难读懂程序，初学者应该从一开始就养成良好的书写习惯。在书写 C 语言程序时应注意以下几点：

(1) C 程序中，虽然一行可以写多条语句，一条语句可以分写在多行，但为清晰起见，一般一条语句单独占一行。

(2) 用{}括起来的部分通常表示程序的某一层次结构，{}一般与该结构语句的第一个字母对齐，并单独占一行。

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写(通常层层右缩两格)，以便使层次更加清晰，增加程序的可读性。

(4) 标识符与关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，可以加一个空格来增加清晰度。

1.3 C 语言程序上机调试的步骤和方法

C 语言是一种编译型的高级语言，描述解决问题算法的 C 语言源程序文件(*.c)必须先用 C 语言编译程序(Compiler)编译，形成中间目标程序文件(*.obj)，然后再用连接程序(Linker)将该中间目标程序文件与有关的库文件(*.lib)和其他有关的中间目标程序文件连接起来，形成最终可以在操作系统平台上运行的二进制形式的可执行程序文件(*.exe)。所以从纸上写好的一个 C 语言源程序文件到可以在计算机操作系统平台上执行的可执行程序文件需要经过以下几个步骤(如图 1.1 所示)。

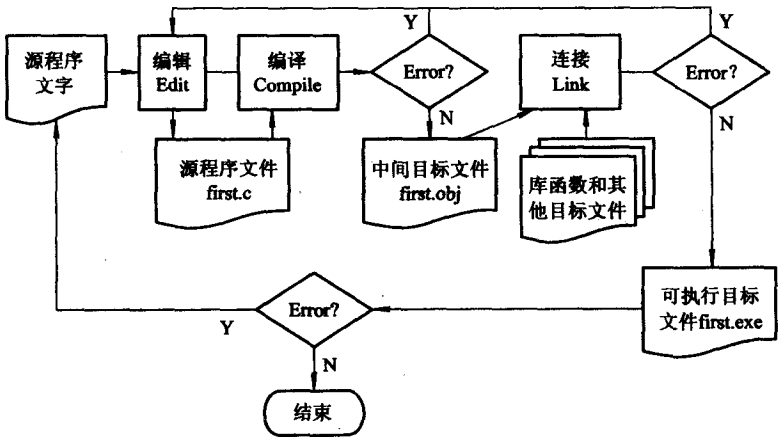

图 1.1　C 语言程序上机调试流程图

(1) 编辑(Edit)：用任何一种编辑软件将源程序代码输入计算机，形成源程序文件。这期间必须注意严格按照 C 语言的语法规则输入，特别注意不要添加格式字符，更不能输入不允许的特殊字符，例如全角的关键字。所以建议不要使用 Word 之类的编辑软件编辑源程序。

(2) 编译(Compile)：将上一步形成的源程序文件作为编译程序的输入，进行编译。编译程序会自动分析、检查源程序的语法错误，并按两类错误类型(Warning 和 Error)报告出错行和原因。用户根据报告信息修改源程序，再编译，直到程序正确后，输出中间目标程序文件。

(3) 连接(Link)：使用连接程序，将上一步形成的中间目标文件与所指定的库文件和其他中间目标文件连接，这期间可能出现缺少库函数等连接错误，同样连接程序会报告错误信息。用户根据错误报告信息再修改源程序，再编译，再连接，直到程序正确无误后输出可执行文件。

(4) 运行(Run)：连接完成后，就可以运行可执行文件，得到运行结果。当然也可能由于算法问题而使源程序具有逻辑错误，得到错误的运行结果；或者由于语义上的错误，例如用 0 做除数，出现运行时错误(Division by zero)。这就需要检查算法问题，重新编写源程序并执行以上各步，直到运行结果正确为止。要保证结果的正确性，就需要设计出程序测试计划，进行全面、细致而艰苦的测试工作。

1.4　算　　法

在程序设计中，不可避免地需要涉及算法。有人这样说过："计算机科学就是研究算法的科学"，足见算法在程序设计中的重要性。

1.4.1　算法的概念

著名的瑞士计算机科学家、PASCAL 语言的发明者 Niklaus Wirth 教授提出了程序定义的著名公式：

程序＝算法＋数据结构

这个公式的重要性在于它说明了程序与算法的关系；同时，说明了对于数据结构的选择也是十分重要的。对于程序而言，算法与数据结构是统一的关系。

通常认为，算法是对特定问题求解步骤的一种描述。对于同一问题，可以有不同的解题方法和步骤。当然，方法有优劣，有的方法只需很少的步骤，而有些方法则需要较多的步骤。我们希望采用简单的、运算步骤少的方法。

1.4.2 算法的特性

当我们拿到一个需要求解的问题之后，怎样才能编写出程序呢？除了选定合理的数据结构外，一般来说，十分关键的一步是设计算法，有了一个好的算法，就可以用任何一种计算机高级语言把算法转换为程序(即编写程序)。

一个算法应当具有以下五个特性：

(1) 有穷性。一个算法应包含有限个操作步骤。也就是说，在执行若干个操作步骤之后，算法将结束，而且每一步都在合理的时间内完成。

(2) 确定性。算法中每一条指令必须有确切的含义，不能有二义性，对于相同的输入必能得出相同的执行结果。

(3) 可行性。算法中指定的操作，都可以通过已经实现的基本运算执行有限次后实现。

(4) 有零个或多个输入。在计算机上实现的算法是用来处理数据对象的，在大多数情况下，这些数据对象需要通过输入来得到。

(5) 有一个或多个输出。算法的目的是为了求“解”，这些“解”只有通过输出才能得到。

算法必须能在有限时间内完成，且对相同的输入有相同的输出。在程序设计语言中，与算法密切相关的便是语句，包括与程序执行处理有关的“功能语句”(如输入语句、输出语句、赋值语句、调用语句等)和与程序执行流程有关的语句(如条件语句、循环语句等)。对程序设计而言，算法的确定也就是如何合理安排这些语句以完成人们所要求的特定功能。

1.4.3 算法的表示方法

算法可以用各种描述方法来进行描述，常用的是自然语言、伪代码、传统流程图、N-S 流程图等。

1. 自然语言表示法

所谓自然语言，就是人们日常使用的语言，可以是汉语、英语或其他语言。

【例 1.2】 求 1＋2＋3＋4＋5＋6＋7＋8＋9＋10 之和。

先设 s 代表累加之和，以 n 代表计数器，用自然语言表示的算法为：

(1) 使 s=0，n=1。

(2) 使 s+n，得到的和仍放在 s 中。

(3) 使 n 的值加 1。

(4) 如果 n≤10，则返回第(2)步重新执行；否则循环结束，此时 s 中的值就是从 1 加到 10 之和，输出 s。

上述用自然语言表示的算法通俗易懂，但它的缺点是：

(1) 比较繁琐冗长。往往要用一段冗长的文字才能说清楚所要进行的操作。例如例 1.2 中的“使 s+n，得到的和仍放在 s 中“不如写成“s+n=>s”简洁。

(2) 容易出现“歧义性”。自然语言往往要根据上下文才能正确判断出其含义，不太严格。如“王五要张三把他的笔拿来”，究竟指的是谁的笔，就有“歧义性”。

(3) 用自然语言表示顺序执行的步骤比较好懂，但如果算法中包含判断和转移时，用自然语言描述就不够直观清晰。

因此，除了对那些很简单的问题之外，一般不用自然语言表示算法。

2. 伪代码表示法

伪代码是一种近似高级语言但又不受语法约束的一种语言描述方式，用介于自然语言和计算机语言之间的文字和符号来表示算法，即计算机程序设计语言中具有的语句关键字用英文表示，其他的可用汉字也可用英文表示，只要便于书写和阅读即可。用伪代码表示算法，并无固定的、严格的语法规则，只要求把意思表达清楚，并且书写的格式要清晰易懂。

【例 1.3】 求 m!，用伪代码表示的算法如下：

```
开始
    从键盘输入一个正整数给 m
    置 p 的初值为 1
    置 i 的初值为 1
    当 i<=m，重复执行下面的操作：
        使 p=p×i
        使 i=i+1
    (循环体到此结束)
    打印 p 的值
结束
```

也可以写成以下形式：

```
begin(算法开始)
    input m
    1→p
    1→i
    while i<=m
      { p×i→p
        i+1→i }
    print p
end(算法结束)
```

从例 1.3 可以看到，伪代码书写格式比较自由，可以随手写下去，容易表达设计者的思想。

3. 传统流程图表示法

流程图也是描述算法的一种很好的形式。传统的流程图由图 1.2 中所示的几种基本符号组成。

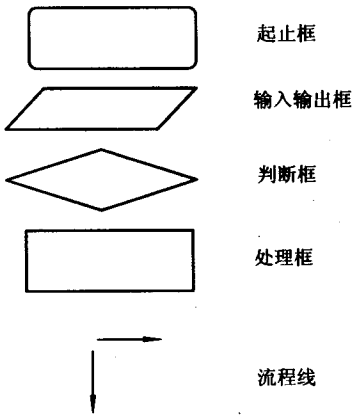

图 1.2 流程图符号示例

结构化程序设计中采用三种基本结构，即顺序结构、选择结构和循环结构。这三种基本结构有以下共同特点：

(1) 只有一个入口；

(2) 只有一个出口；

(3) 结构内的每一部分都有机会被执行到；

(4) 结构内不存在“死循环”(无终止的循环)。

已经证明，由以上三种基本结构组成的算法结构，可以解决任何复杂的问题。用流程图可以表示以下三种基本结构。

1) 顺序结构

顺序结构的程序是按语句的书写顺序执行的，如图 1.3 所示。A 和 B 两个框是顺序执行的，即在执行完 A 框所指定的操作后，必然紧接着执行 B 框所指定的操作。顺序结构是最简单的一种基本结构。

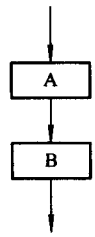

图 1.3 顺序结构图

2) 选择结构

选择结构又称分支结构或条件结构，如图 1.4 所示。此结构中必包含一个判断框，根据给定的条件 P 是否成立来进行选择。若 P 成立，则执行 A 框中的操作；否则，执行 B 框中的操作。

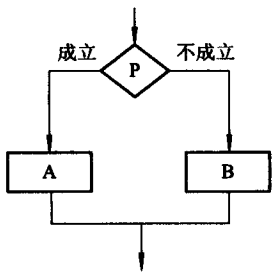

图 1.4 选择结构图