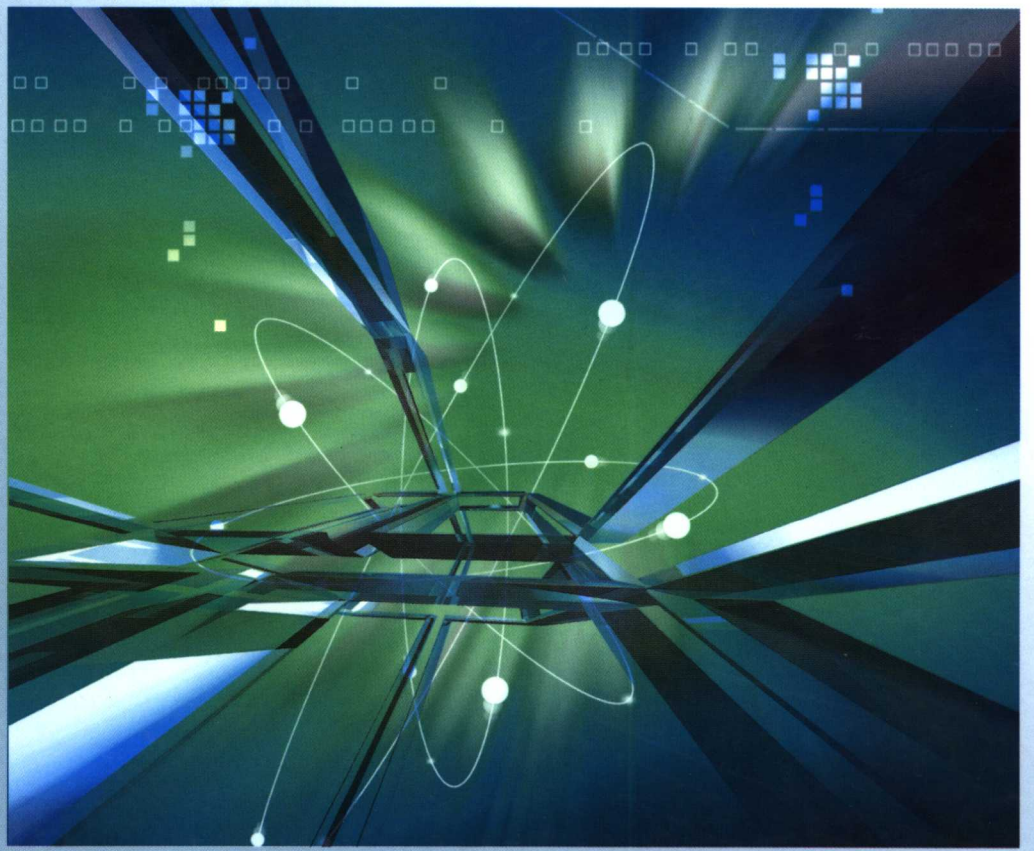
21世纪高职高专计算机系列规划教材

C语言程序设计

柏万里 李红霞 主编　侯梦雅 吴铬 吴昂 饶泉发 副主编　王阳辉 主审



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪高职高专计算机系列规划教材

C 语言程序设计

柏万里　李红霞　主　编

侯梦雅　吴　铬
吴　昂　饶泉发　副主编

王阳辉　主　审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书由浅入深、循序渐进地介绍了C语言的语法结构和使用，系统地讲述了C语言程序设计的基本方法和技巧。本书共分16章，主要内容包括：C语言概述，C语言的编程元素，C语言程序提供的运算，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，预处理命令，指针，结构体、共用体和枚举数据类型，位运算，文件，C语言图形功能，常见错误与程序调试，C++简介。

本书结构清晰、内容丰富、实例恰当，方便教师教学和学生学习，并有配套的习题解答及上机指导教程，可作为高职高专学生学习C语言程序设计的教材，也可供报考计算机等级考试者和其他自学者参考。

图书在版编目（CIP）数据

C语言程序设计/柏万里，李红霞主编. －北京：中国铁道出版社，2006.8
（21世纪高职高专计算机系列规划教材）
ISBN 7-113-07193-7

Ⅰ.C... Ⅱ.①柏...②李... 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（2006）第099888号

书　　名：C语言程序设计
作　　者：柏万里　李红霞　等
出版发行：中国铁道出版社（100054，北京市宣武区右安门西街8号）
策划编辑：严晓舟　胡娟利
责任编辑：苏　茜　谢立和　高婧雅
封面设计：薛　为
封面制作：白　雪
责任校对：李　旸
印　　刷：化学工业出版社印刷厂
开　　本：787×1092　1/16　　**印张**：17.25　**字数**：400千
版　　本：2006年8月第1版　　2006年8月第1次印刷
印　　数：1～4 000册
书　　号：ISBN 7-113-07193-7/TP・1907
定　　价：28.00元

前　言

C 语言是一种具有极强生命力的计算机高级程序设计语言，它是根据结构化程序设计原则设计并实现的。C 语言同时具有高级语言和低级语言的特点，所以，它不仅适合于应用程序设计，而且适合于系统程序设计。

C 语言是大学计算机专业数据结构、操作系统课程的前导课程，也可作为学习其他计算机语言的基础，是计算机等级考试（二级）的内容之一。因此，选择 C 语言作为计算机基础课程的教学内容，适合当前形势发展的需要。

本书根据高等院校计算机专业 C 语言程序设计教学大纲，并参照全国计算机等级考试二级 C 语言考试大纲，在编者十几年讲授 C 语言程序设计的基础上编写而成。在编写过程中，我们遵循了知识讲授和能力训练并重的原则。在讲清基本知识的基础上，注意例题的选择，每个知识点基本上做到先用基本例题讲解，然后再用实际应用例题讲解。

本书共分 16 章，前 13 章为 C 语言的基本知识和程序设计方法；第 14 章为 C 语言图形功能，作为 C 语言课程设计基础知识，为计算机专业学生的选修内容；第 15 章是常见错误与程序调试，提出了初学者在学习过程中常见的错误及程序的调试方法；第 16 章是 C++简介，使读者对 C++有初步了解，为今后学习 C++打下基础。

本书的所有例题，均在 Turbo C++ 3.0 环境下调试通过。前 15 章例题既可在 Turbo C 2.0 环境下调试，也可在 Turbo C++ 3.0 环境下调试。C 语言程序设计实践性要求很强，希望读者在学习过程一定要重视上机调试，不要满足于掌握理论知识。

为帮助读者学习本书，我们同时编写了一本《C 语言程序设计习题解答与上机指导》，提供本书中各章的习题参考答案及上机参考程序，由中国铁道出版社与本书同期出版。

本书由柏万里、李红霞担任主编，负责全书的统稿、审稿和定稿；侯梦雅、吴铬、吴昂、饶泉发担任副主编；王阳辉担任主审。具体分工如下：柏万里编写第 3 章、第 4 章、第 9 章、第 13 章及第 15 章；侯梦雅编写第 1 章、第 2 章；吴铬编写第 5 章、第 6 章；李红霞编写第 7 章、第 8 章；吴昂编写第 10 章、第 14 章；饶泉发编写第 11 章、第 12 章、第 16 章。封园鹏、熊志文参加了编写工作，王阳辉审阅了全稿，在此谨致谢忱。

由于编者水平有限，加之时间仓促，书中疏漏和不足之处在所难免，敬请专家和读者不吝指正。

编　者

2006 年 5 月

目　录

第 1 章　C 语言概述

1.1　结构化程序设计语言简介

1.1.1　结构化程序设计思想的产生

学过计算机的人大都知道“算法+数据结构=程序”这一著名公式。提出该公式的正是 1984 年的图灵奖获得者，瑞士计算机科学家尼克劳斯·威茨（Niklaus Wirth）。到目前为止，他是获得图灵奖殊荣的唯一瑞士学者。

威茨生于 1934 年 2 月 15 日，1958 年从苏黎世工学院取得学士学位后，到加拿大的莱维大学深造，之后进入美国加州大学伯克利分校并获得博士学位。期间，他开发出了 Algol W 及 PL360 两种语言，成功奠定了威茨程序设计语言专家的地位。

成名后的他拒绝了斯坦福大学的挽留，于 1967 年回到祖国，先在苏黎世大学任职，第二年转到母校苏黎世工学院。在这里，他在 CDC6000 上成功设计了 PASCAL 语言。

1971 年，基于自己的开发程序设计语言和编程的实践经验，威茨首次提出了“结构化程序设计”（Structured Programming）的概念。威茨提出的这种结构化程序设计方法又称为“自顶向下”或“逐步求精”法，采用了模块分解与功能抽象和自顶向下、分而治之的方法，从而有效地将一个较复杂的程序设计任务分解成许多易于控制和处理的子程序，便于开发和维护。因此，结构化程序设计方法迅速走红，在程序设计领域引发了一场革命，并在整个 20 世纪 70 年代的软件开发中占绝对统治地位。

今天，结构化程序设计和设计技术已经是无处不在了，几乎每一种程序设计语言都具有支持结构化程序设计所需的手段，甚至像 BASIC 那样传统的非结构化语言也已开始利用结构化程序设计结构。原因是很简单的，结构化程序已被证明比非结构化程序容易编写和维护。

1.1.2　结构化程序设计方法

一个程序必须包括以下两个部分：

（1）对数据的描述

在程序中要指定数据的类型和数据的组织形式，即数据结构（Data Structure）。

（2）对数据操作的描述

即操作步骤，也就是算法。算法是为解决一个问题而采取的方法和步骤。

它有以下几个特性：

① 有穷性：一个算法应包含有限的操作步骤，而不能是无限的，否则就失去了实际意义。

② 确定性：算法中每一个步骤都应当是确定的，不能含糊、模棱两可，否则会导致结果不确定。

③ 有零个或多个输入。

④ 有一个或多个输出。

⑤ 有效性/算法的每一步都应该能有效地执行。

数据是操作的对象，操作的目的是对数据进行加工处理，以得到期望的结果。

（3）著名计算机科学家威茨（Niklaus Wirth）提出的公式：

程序=算法+数据结构

（4）实际上一个程序除了上述两大元素之外还涉及到所用到的具体语言和设计思想，因此还可以这样表示：

程序=算法+数据结构+程序设计方法+语言

学习计算机语言的目的就是用该语言工具设计出可供计算机运行的程序。那么拿到一个实际问题之后，怎样动手编写程序呢？一般按图 1-1 所示的步骤进行。

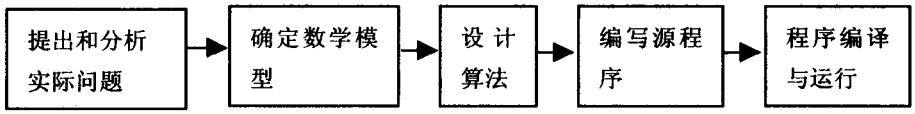


图 1-1　程序设计的步骤

结构化程序按它们所执行的操作来组织。从本质上讲，程序由执行较大、较复杂的过程离散出较小、较简单的执行单独的任务的过程（也称做函数）。这些过程之间尽可能地保持相互独立，每一个都有其自己的数据和逻辑。通过使用参数在过程之间传递信息，过程可以有不能在过程范围以外存取的局部数据。从某一个角度来讲，函数可以被融和在一起构成一个应用程序，目标是使软件开发相对简单，同时提高程序的可靠性和可维护性。

1.1.3　结构化程序设计的步骤

结构化程序设计可采用自顶向下、逐步求精的方法。

1．自顶向下的模块化设计

（1）第 1 步先把这个程序高度抽象，看作是一个最简单的控制结构，而实际上是一个庞大而复杂的功能模块。

（2）第 2 步分析这个功能的完成可以由几部分组成，或可以划分为几个步骤，可以进一步分解成若干个较低一层的模块，每个模块都表示了一个较上层功能较小的功能。然后，对分解出来的每一个下层模块，反复运用第 2 步的方法，逐层分解到非常简单、功能很小、能够容易地用程序语句实现的最低一层的模块。

由于分解出来的每一个模块都属于基本控制结构的集合，因此这个模块化的程序就是一个结构化程序。

2．逐步求精

自顶向下模块化设计，把一个程序分解为若干个层次模块，但它虽然表达了程序中各功能之间的关系，却不能表达每个模块的内部逻辑。采用逐步细化的方法，把每一个模块功能进一步分解成程序的内部逻辑。对每一个模块的细化应包括功能细化、数据细化和逻辑细化 3 个方面。

（1）功能细化应对本模块的功能进行分析，力图分解为若干个更为简单的子功能。

（2）数据细化应列出本模块涉及到的数据项和各数据类型。

（3）逻辑细化确定所构成的子模块之间的结构关系，用基本控制结构来描述这些关系，从而形成各个模块的内部处理逻辑，一般用程序流程图或其他工具来表示。

这样对每个模块都进行上述 3 个方面的细化，就可以将整个程序的逻辑过程描述清楚，为编程做好准备。

1.1.4　结构化程序设计的风格

良好的程序设计风格包括以下几个方面。

1．标识符的命名

标识符命名应注意以下几点：

（1）命名规则在整个程序中前后一致，不要中途变化，给阅读理解带来困难。

（2）命名时一定要避开程序设计语言的保留字，否则程序在运行中会产生莫名其妙的错误。

（3）尽量避免使用意义容易混淆的标识名。

2．程序中的注释

进行程序注释时应注意以下几点：

（1）注释一定要在编程时书写，不要在程序完成之后进行补写。

（2）解释性程序不是简单直译程序语句，而是要说明程序段的动机和原因，提供的是从程序本身难以得到的信息，用来说明“做什么”。

（3）一定要保持注释与程序的一致性，程序修改后，注释也要及时做相应修改。

3．程序的布局格式

一个程序可以充分利用空格、空行和缩进等改善程序的布局，以获得较好的视觉效果。

1.2　C 语言发展概况和主要特点

1.2.1　C 语言出现的历史背景

C 语言于上个世纪 70 年代初诞生于美国的贝尔实验室。在此之前，人们编写系统软件主要是使用汇编语言。由于汇编语言编写的程序依赖于计算机硬件，其可读性和可移植性都比较差。而高级语言的可读性和可移植性虽然较汇编语言好，但一般高级语言又不具备低级语言能够直观地对硬件实现控制和操作，程序执行速度相对较快的优点。在这种情况下，人们迫切需要一种既具有一般高级语言特性，又具有低级语言特性的语言。于是 C 语言就应运而生了。

由于 C 语言既具有高级语言的特点又具有低级语言的特点，因此迅速普及，成为当今最有发展前途的计算机高级语言之一。C 语言既可以用来编写系统软件，也可以用来编写应用软件。现在，C 语言广泛地应用在机械、建筑和电子等行业，用来编写各类应用软件。

C 语言的发展历程如下。

（1）ALGOL60：一种面向问题的高级语言。ALGOL60 离硬件较远，不适合编写系统程序。

（2）CPL（Combined Programming language，组合编程语言）：CPL 是一种在 ALGOL60 基础上更接近硬件的一种语言。CPL 规模大，实现困难。

（3）BCPL（Basic Combined Programming language，基本的组合编程语言）：BCPL 是对 CPL 进行简化后的一种语言。

（4）B 语言：是对 BCPL 进一步简化所得到的一种很简单接近硬件的语言。 B 语言取 BCPL 语言的第一个字母。B 语言精练、接近硬件，但过于简单，数据无类型。B 语言诞生后，Unix 开始用 B 语言改写。

（5）C 语言：是在 B 语言基础上增加数据类型而设计出的一种语言。C 语言取 BCPL 的第二个字母。C 语言诞生后，UNIX 很快用 C 语言改写，并被移植到其他计算机系统。

（6）标准 C 、ANSI C、ISO C：C 语言的标准化。

最初 UNIX 操作系统是采用汇编语言编写的，B 语言版本的 UNIX 是第一个用高级语言编写的 UNIX。在 C 语言诞生后，UNIX 很快用 C 语言改写，C 语言良好的可移植性很快使 UNIX 从 PDP 计算机移植到其他计算机平台，随着 UNIX 的广泛应用，C 语言也得到推广。从此 C 语言和 UNIX 像一对孪生兄弟，在发展中相辅相成，UNIX 和 C 语言很快风靡全球。

从 C 语言的发展历史可以看出，C 语言是一种既具有一般高级语言特性（ALGOL60 带来的高级语言特性），又具有低级语言特性（BCPL 带来的接近硬件的低级语言特性）的程序设计语言。C 语言从一开始就用于编写大型、复杂的系统软件，当然 C 语言也可以用来编写一般的应用程序。

IBM 微机 DOS、Windows 平台上常见的 C 语言版本有：

（1）Borland 公司

Turbo C、Turbo C++、Borland C++及 C++ Builder（Windows 版本）。

（2）Microsoft 公司

Microsoft C 及 Visual C++（Windows 版本）。

1.2.2 C 语言的特点

C 语言归纳起来具有下列特点。

（1）C 语言是结构化的语言。

C 语言程序有 3 种基本结构：

① 顺序结构

② 选择结构

③ 循环结构

而由这 3 种基本结构组成的程序可以解决许多复杂的问题。C 语言还具有结构化的控制语句，如 if...else 语句、while 语句、switc 语句以及 for 语句等，使用这些语句可以方便地控制程序的流程。因此，C 语言是理想的结构化语言，符合现代编程风格的要求。

（2）C 语言是模块化的语言

C 语言主要用于编写系统软件和应用软件。一般来说，一个较大的系统程序往往被分为若干个模块，每一个模块用来实现特定的功能。

在 C 语言中，用函数作为程序的模块单位，便于实现程序的模块化。在程序设计时，将一些常用的功能模块编写成函数，放在函数库中供其他函数调用。模块化的特点是可以大大减少重复编程。程序设计时，只要善于利用函数，就可减少劳动量、提高编程效率。

（3）语言简洁紧凑，方便灵活

C 语言一共只有 32 个关键字和 9 种控制语句，程序书写形式自由，主要用小写字母表示。

（4）程序可移植性好

C 语言程序便于移植，适用于各种型号的计算机和各种操作系统。

（5）数据结构丰富，具有现代化语言的各种数据结构

C 语言的基本数据类型有整型、实型以及字符型等。在此基础上还可创建各种构造数据类型，如数组、指针、结构体和共用体等。使用 C 语言还能用来实现复杂的数据结构，如链表、树等。这样丰富的数据结构无疑极大地增强了 C 语言的功能。

（6）C 语言运算符丰富、代码效率高

C 语言共有 44 种运算符，使用各种运算符可以实现在其他高级语言中难以实现的运算。在代码质量上，C 语言程序的代码效率仅比用汇编语言编写的程序低 10%～20%。

1.3　熟悉 C 语言程序结构和书写格式

本书列举以下几个例子来说明 C 语言程序结构和书写格式。

【例 1.1】在屏幕上输出一行信息：“Thank you.”

```
main()
{ printf ("Thank you.\n");}
```

运行结果：

```
Thank you.
```

说明：

（1）main 表示主函数，每一个 C 语言程序都必须有一个 main 函数。

（2）函数体由大括弧{ }括起来。

（3）本例中主函数内只有一个输出语句，printf 函数是 C 编译系统提供的标准函数库中的输出函数，将双引号内的字符串原样输出；“\n”是换行符，即在输出“Thank you.”后自动换行。语句最后有一个分号。

【例 1.2】计算两个整数 a 和 b 之和，并把结果放在变量 x 中。

```
main()
{ int a,b,x;/*定义变量*/
  a=2;b=3;
  x=a+b;
  printf("x=%d",x);}
```

运行结果：

```
x=5
```

说明：

（1）/*...*/表示注释部分，为了便于理解，可以加在程序中的任何位置。

（2）第 2 行是变量定义部分，说明 a、b、c 为整型（int）变量。

（3）第 3 行是两个赋值语句，使 a 和 b 的值分别为 2 和 3。第 4 行使 x 的值为 a + b。

（4）第 5 行中“%d”是输入输出的“格式字符”，用来指定输入输出时的数据类型和格式，“%d”表示“十进制整数类型”。在执行输出时，此位置上代以一个十进制整数值。printf 函数中括弧内最右端 x 是要输出的变量，现在它的值为 5。