普通高等教育“十一五”国家级规划教材　计算机系列教材

# C语言程序设计基础

袁仲雄　王剑云　张超　魏为民　编著

清华大学出版社

普通高等教育"十一五"国家级规划教材　计算机系列教材

袁仲雄　王剑云　张超　魏为民　编著

# C语言程序设计基础

清华大学出版社
北京

## 内 容 简 介

本书是一本理论和实践相结合的C语言程序设计基础教材。

书中全面介绍了C语言程序设计的基本概念和基本方法，包括C语言数据类型、输入输出、结构化程序的三种结构、数组、函数、指针、结构体和共用体、文件操作。

本书可作为C语言程序设计课程的教材，也可供从事程序设计的工程技术人员参考。



**图书在版编目(CIP)数据**

C语言程序设计基础/袁仲雄等编著. —北京：清华大学出版社，2012.8
(计算机系列教材)
ISBN 978-7-302-29385-9

Ⅰ. ①C…　Ⅱ. ①袁…　Ⅲ. ①C语言－程序设计　Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第158590号

**责任编辑**：魏江江　薛　阳
**封面设计**：常雪影
**责任校对**：梁　毅
**责任印制**：张雪娇

**出版发行**：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954
**印 装 者**：北京市密东印刷有限公司
**经　　销**：全国新华书店
**开　　本**：185mm×260mm　　**印　　张**：18.5　　**字　　数**：464千字
**版　　次**：2012年8月第1版　　**印　　次**：2012年8月第1次印刷
**印　　数**：1～3000
**定　　价**：29.50元

产品编号：044467-01

# 前言

FOREWORD

高级语言程序设计是软件设计的重要理论和实践基础，其所介绍的基本技术是软件开发使用的基本方法。高级语言程序设计课程是理论和实践并重的课程，要求学生既要掌握程序设计的基础理论知识，又要掌握编写和调试程序的基本技能，因此高级语言程序设计课程是培养学生掌握软件设计能力的重要环节。

本书选用C语言作为高级语言程序设计的程序设计工具，全面、详细地介绍了C语言的语法、语义和各种成分的用法。通过对C语言程序设计方法的了解和掌握，领会高级语言程序设计的基本思想和基本方法。

本书第1章是程序设计基础，介绍了程序设计的基础概念，还介绍了算法、数据结构的基本概念；第2章介绍了C语言的数据类型、键盘输入和屏幕输出以及C语言的上机过程；第3章介绍了C程序的三种基本结构；第4章介绍了常用的数据结构——数组；第5章介绍了C程序的重点——函数；第6章介绍了C程序的重点和难点——指针；第7章介绍了结构体和共用体；第8章介绍了文件以及文件操作。

本书第1章由袁仲雄、魏为民编写，第2章和第3章由魏为民编写，第4章、第6章和附录由王剑云编写，第5章由袁仲雄、张超编写；第7章和第8章由张超编写。全书由袁仲雄统稿。

本书在编写过程中参考了不少国内外文献和资料，在此谨向这些文献和资料的作者表示衷心的感谢。

由于编者水平有限，书中难免有不妥之处，恳请读者指正。作者邮箱地址：yuanzhongxiong@shiep.edu.cn。

编　者

2012年5月31日

# 目 录

CONTENTS

## 第 1 部 分

## 第 2 部 分

# 第1部分

# 第 1 章　程序设计基础

## 1.1　程序设计基本概念

计算机作为一种智能工具正广泛地应用于人类社会的各行各业，它由硬件和软件组成。本节将介绍软件的一些基本概念。

### 1. 程序

要利用计算机来处理问题，就必须要事先编写出使计算机按照人的意愿工作的应用程序。所谓程序，就是一系列遵循一定规则和思想并能正确完成指定工作的代码（也称为指令序列）。通常，一个计算机程序主要描述两部分内容，第一部分是描述问题的每个对象及它们之间的关系，第二部分是描述对这些对象进行处理的规则。其中关于对象及它们之间的关系涉及数据结构的内容，而处理规则是指求解某个问题的算法。因此，对程序的描述有如下等式：

$$\text{程序} = \text{数据结构} + \text{算法}$$

一个设计合理的数据结构往往可以简化算法，而一个好的程序有可靠性、易读性、可维护性等良好特性。

### 2. 程序设计

所谓程序设计，就是根据计算机要完成的任务提出相应的需求，并在此基础上设计数据结构和算法，然后再编写相应的程序代码并测试该代码运行的正确性，直到能够得到正确的运行结果为止。通常，程序设计是很讲究方法的，一个良好的设计思想能够大大提高程序的高效性和合理性。因此有人提出如下关系：

$$\text{程序设计} = \text{数据结构} + \text{算法} + \text{程序设计方法学}$$

任何一个程序必须包含这三方面的内容。

### 3. 算法

所谓算法，就是问题的求解方法。通常，一个算法由一系列求解步骤组成。正确的算法要求组成算法的规则和步骤的意义是唯一确定的，不能存在二义性，而且这些规则指定的操作是有序的，按算法指定的操作顺序执行能够在有限的执行步骤后给出正确的结果。

上面提到，算法不允许存在二义性，这是非常重要的，如果算法存在二义性，那么程序的编码工作将无法进行，同时，算法也不允许存在模糊的概念，因为二义性和模糊性都是不可

操作的。如“加一吨的水”有明确的概念，这在程序的计算或控制中是可以操作的，但是如果说“加一些水”，那就无法操作了，因为日常生活中，多少才认为是“一些”呢？这就没有明确的意义，毕竟程序是根据人的具体要求来完成相应工作的。

通常算法的建立过程是逐步求精的，一般是先给出粗略的计算步骤框架，然后再对框架中的具体内容进行逐步细化，添加必要的细节，使之成为较为详细的描述，当然，细化可能做不到一步到位，因此还要进行更进一步的细化，直到能够把需求通过编程语言完全描述为止。

描述算法的常用工具是流程图，也称为程序框图，流程图是算法的图形描述，它往往比程序更直观，更容易阅读和理解。不过它只是一种表现工具，计算机并不能直接识别和运行流程图。

**4. 数据结构**

数据结构是指数据对象及其相互关系和构造方法，程序中的数据结构描述了程序中的数据间的组织形式和结构关系。

数据结构与算法密不可分，一个良好的数据结构将使算法简单化；只有明确了问题的算法，才能较好地设计出数据结构，因此两者是相辅相成的。对于计算机程序而言，其构成与数据结构关系密切，程序在实现算法的同时，还必须完整地体现作为算法操作对象的数据结构。对于复杂问题的求解，常常会发现由于对数据的表示方式和结构的差异，对该问题的抽象求解算法也会完全不同。前面已经提到，对同一个问题求解，当然允许有不同的算法，也允许存在不同的数据结构，而依不同算法编写的操作代码，其执行效率也就不一样了。

**5. 语言**

算法不仅要通过具体的语言来表述，还要采用合适的方法来表达，才能够形成程序。可以这样说，算法是程序的灵魂，是解决“做什么”和“怎样做”的问题。一个好的程序必须要有一个合理、高效的算法，数据结构是程序要处理的具体对象，语言是描述算法过程的工具。

**6. 程序设计的方法**

程序设计方法分为两大类：面向过程的程序设计方法和面向对象的程序设计方法。

面向过程的程序设计方法是将完成某项工作的每一个步骤和具体要求都考虑在内来设计程序，程序主要用于描述完成这项工作所涉及的数据对象和具体操作规则，如先做什么、后做什么、怎样做、如何做。C 语言是一种面向过程的程序设计语言。

面向对象的程序设计方法是将任何事物都看成一个对象，它们之间通过一定的渠道相互联系，对象是活动的、相互对立的，是可以激发的，每个对象都是由数据和操作规则构成的。在进行程序设计时，主要针对一个个对象，所有数据分别属于不同的对象，并被封装在对象内，只要激发每个对象完成相对独立的操作功能，整个程序就会自然完成全部操作。

## 1.2 算法和结构化程序设计

### 1.2.1 算法的概念

当设计一个程序时，通常先分析程序中需要的数据并对数据进行描述，即数据结构设计，然后再根据功能要求设计解决问题的方法，即算法设计，最后用某种计算机语言将其描述出来；同时，为了保证程序有很高的正确性、可靠性、可读性、可理解性、可修改性和可维护性，在整个设计过程中，还必须采用科学的程序设计方法。因此，可以把程序表示为：

程序 = 数据结构 + 算法 + 程序设计方法 + 语言工具和环境

数据结构指的是数据的组织形式，并且必须是允许定义和使用的数据结构。根据数据的性质和结构，可以把数据定义为某种数据类型；同时利用规定的数据类型，还可以构建更复杂的数据结构，如链表、堆栈、树等。算法指的是解决问题的方法和步骤；程序设计语言是全部计算机指令(语句)的集合；按照程序设计语言的词法、语法和语义规则设计的计算机指令(语句)序列就是一个程序，而程序也是计算机算法的体现。当设计和编写程序时，并不是简单地写一个程序，而是像一个工程，为了保证其质量，就必须采用科学的程序设计方法，程序设计方法的种类有很多，主要有结构化程序设计方法、面向对象程序设计方法等。

算法设计是程序设计的主要步骤，没有高质量的算法就没有高质量的程序。

算法(Algorithm)指的是解决问题的方法和步骤。一个算术问题的解题过程、一首乐谱、一份菜谱、一个工作计划等都是一个算法。人们在工作和生活中做每件事情都有其特定的步骤和方法。在现代，特别是计算机诞生之后，人们把计算机解题步骤称为计算机算法，本书谈到的算法，没有特别说明，指的就是计算机算法。

### 1.2.2 算法的特性

当设计和使用某个算法时，必然要考虑其是否可行。使用一个没有价值的算法是没有任何意义的。因而 算法具有若干约束特性，具有约束特性的算法才称其为算法。著名计算机科学家 Knuth 在其《计算机程序设计艺术》一书中详细描述了算法的 5 个特性：

(1) 有穷性：算法是一组有穷步骤序列，即一个算法必须在执行有穷步后结束。一个算法如果永远不能结束或需要运行相当长的时间才能结束，那么这样的算法是没有使用价值的。

(2) 确定性：算法中的每一个步骤都必须要有明确的定义，不能含糊、不能有歧义。如“请面向前方”就有歧义，因为“前方”在无任何参照物的参照下，可能是东、南、西、北等的任何一方。

(3) 大于等于 0 个输入：在算法执行过程中可以有 0 个或若干个输入数据，即算法处理的数据既可以不从外部输入(内部生成)，也可以从外部输入。少量数据适合内部生成，而大量数据一般需要从外部输入，所以多数算法中要有输入数据的步骤。

(4) 大于等于 1 个输出：算法在执行过程中必须要有 1 个以上的输出操作，即算法中必须要有输出数据的步骤。一个没有输出步骤的算法是毫无意义的。

(5) 可行性：算法中的每一步骤都是可实现的，即在现有计算机上是可执行的。如：当 B 是一个很小的实数时，A/B 在代数中是正确的，但在算法中却是不正确的，因为它在计算机上无法执行，而要使 A/B 能正确执行，就必须在算法中使 B 满足条件：$|B|>\delta$，其中 $\delta$ 是一个计算机允许的小的实数。

一个问题可有若干个不同的可行算法。在不同的算法中有好算法，也有差算法，如：针对同一问题，执行 10 分钟的算法要比执行 1 小时的算法好得多。目前，评价算法质量主要有 4 个基本标准：正确性、可读性、通用性和高效性。一个好的算法应满足运行结果正确、可读性好、可适用一类问题的解决并执行速度快、运用时间短、占用内存少标准。当然，高效性和可读性往往是矛盾的，可读性要优先于高效性。目前，在计算机速度比较快、内存比较大的情况下，高效性已处于次要地位。

### 1.2.3 算法的描述

对于算法，需要选择一种合适的描述工具进行描述。常用的描述工具有自然语言、流程图、伪代码等。

#### 1. 用自然语言描述算法

用自然语言描述算法就是选择某种日常使用的语言(如汉语、英语)来描述算法。使用自然语言描述算法的优点是描述自然、通俗易懂而且灵活多样，但缺点是容易产生歧义；因此，在算法设计中应少用或不用自然语言描述算法。通常是在设计初步算法时适当采用自然语言描述，然后还需用其他描述工具细化算法描述。

**【例 1-1】** 输入两个数 a、b 的值，然后交换两数的值。

算法描述如下。

S1：输入两个数 a、b 的值；

S2：使 a 的值赋给 c；

S3：使 b 的值赋给 a；

S4：使 c 的值赋给 b；

S5：输出 a、b 的值。

**【例 1-2】** 输入两个正整数 m 和 n，求两数的最大公约数。

算法描述如下。

S1：输入正整数 m、n 的值；

S2：若 m 小于 n，则交换两个数的值；

S3：求解 m 整除 n 的值，并赋给 r；

S4：使 n 的值赋给 m；

S5：使 r 的值赋给 n；

S6：当 r 的值不等于 0，转 S3 继续执行；否则执行 S7；

S7：输出两数的最大公约数的值 m。

【例 1-3】 输入一个年份，判定其是否为闰年。

闰年判定的条件为：能被 4 整除但不能被 100 整除的年份是闰年；或者能被 100 整除又能被 400 整除的年份是闰年。

算法描述如下。

S1：输入一个年份 y；

S2：若 y 不能被 4 整除，则输出“y 不是闰年”。

S3：若 y 能被 4 整除，但不能被 100 整除，则输出“y 是闰年”。

S4：若 y 既能被 100 整除，又能被 400 整除，则输出“y 是闰年”；否则输出“y 不是闰年”。

**2. 用流程图描述算法**

流程图是采用一些框图来描述算法的一种工具。其优点是描述简洁、清晰和直观，缺点是由于转移箭头可以无约束使用，所以会影响算法的可靠性。流程图由以下几部分组成。

开始结束框：表示流程图的起点或终点，即开始或结束，框中给出开始或结束说明。

处理框：表示各种处理功能，框中给出处理说明或一组操作。

输入/输出框：表示数据的输入或输出，框中给出输入或输出数据说明。

判断框：表示一个逻辑判断，框中给出判断条件说明或条件。一般情况有两个出口，分别表示条件的成立或不成立（真或假、是或否），但在执行过程中只有一个出口被激活，如图 1-1 所示。

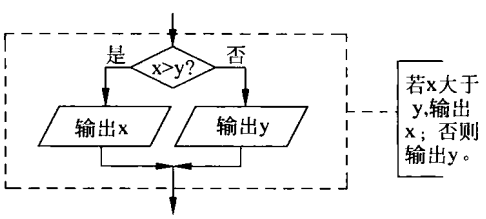


图 1-1 判断框

流程线：表示算法的执行方向，且一定为单向线。

注释框：表示对流程图某一部分的解释、说明。一般绘制在侧面，如图 1-1 所示。

虚线：表示被注释的范围，如图 1-1 所示。

连接：表示流程线的断点（去向或来源），在图中给出断点编号，如图 1-2 所示。

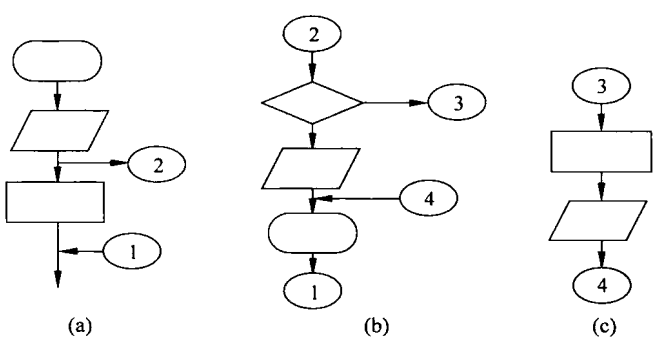


图 1-2 连接示意图