普通高等院校计算机类精品规划教材

# C程序设计

## C Program Design

黄建灯 ◎ 主编


华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

# C 程序设计

主　编 ◎ 黄建灯
副主编 ◎ 吕元长　刘　欣
参　编 ◎ 朱　震　王小琼　沈岚岚
　　李　丹　邱勋拥

华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS
·广州·

**图书在版编目(CIP)数据**

C 程序设计/黄建灯主编. —广州:华南理工大学出版社,2015.2
ISBN 978-7-5623-4522-0

Ⅰ.①C… Ⅱ.①黄… Ⅲ.①C 语言-程序设计-教材 Ⅳ.①TP312

中国版本图书馆 CIP 数据核字(2015)第 020715 号

**C 程序设计**
黄建灯 主编

**出 版 人**:韩中伟
**出版发行**:华南理工大学出版社
(广州五山华南理工大学 17 号楼,邮编 510640)
http://www.scutpress.com.cn E-mail:scutc13@scut.edu.cn
营销部电话:020-87113487 87111048(传真)
**责任编辑**:刘 锋 吴翠微
**印 刷 者**:北京市平谷县早立印刷厂
**开 本**:787mm×1092mm 1/16 **印张**:15.75 **字数**:384 千
**版 次**:2015 年 2 月第 1 版 2015 年 2 月第 1 次印刷
**定 价**:35.00 元

# 前 言

C语言是目前国内外使用最广泛的程序设计语言之一，也是计算机课程体系中的第一门重要的基础课程。它具有简洁紧凑、使用灵活方便、表达能力强、运算符丰富、执行效率高、可移植性好等优点。由于C语言涉及的概念和规则比较多，学习者需要实现思维方式的转换，所以初学者总感觉C语言很难学。

本书从初学者的需求出发，配合高校应用型人才培养目标，真正培养学生使用计算机解决专业领域实际问题的能力，提高学生应用和创新的能力。以C语言为工具，介绍程序设计的基本思想和方法，培养学生无论以后在学习、工作中使用什么语言编程，都能灵活应用这些思想和方法的能力。

本书注重可读性和实用性，以“理论讲解—案例分析—编程实践”的方式组织内容，循序渐进，由浅入深，符合学生的认知过程，把语言和语法的讲解完全融会贯通在程序设计以及案例中。另外，本书每章精选了大量有普遍性、代表性的案例，并详细介绍了每个案例程序的分析和设计过程，通过这些案例的讲解，使读者能够综合应用所学知识解决实际问题，不断提高分析问题、解决问题的能力。

全书共11章，主要内容包括：程序设计基础知识、C语言概述、数据类型与运算规则、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、指针、结构体和共用体、文件等。

本书第1、2、3章由刘欣编写，第4章由沈岚岚编写，第5、6章由朱震编写，第7章由李丹编写，第8章由王小琼编写，第9章由黄建灯编写，第10章由邱勋拥编写，第11章由吕元长编写。全书由黄建灯统稿。

本书在编写过程中得到了许多同事的关心和支持，在此表示深深的谢意。

由于时间仓促、作者水平有限，书中不足和疏漏之处，欢迎读者提出宝贵意见和建议以便再版时进行改进。

编者

2014年11月

# 目 录
# CONTENTS

## 第1章 程序设计基础知识



## 第2章 C语言概述

## 第 3 章 数据类型与运算规则

## 第4章 顺序结构程序设计



## 第5章 选择结构程序设计

## 第6章 循环结构程序设计



## 第7章 数 组



## 第8章 函 数

## 第9章 指 针

## 第10章 结构体和共用体



## 第11章 文 件

# 第1章 程序设计基础知识

随着科学技术的迅猛发展，计算机技术日新月异，计算机程序设计语言也层出不穷。那么，什么是程序语言，什么是程序设计？应该学哪一种程序语言，如何进行程序设计？这些都是程序设计初学者首先遇到的问题，也是程序设计的基本问题和共性问题。

## 1.1 程序与程序语言

### 1.1.1 程序语言

#### 1. 计算机语言

什么是计算机语言(Computer Language)？为什么要使用计算机语言？过去，一提到语言这个词，人们自然想到的是像英语、汉语等这样的自然语言，因为它是人和人相互交流信息不可缺少的工具。而今天，计算机遍布我们生活的每一个角落，除了人和人之间的相互交流之外，我们必须和计算机交流。用什么样的方式和计算机做最直接的交流呢？人们自然想到的是最古老也最方便的方式——语言。人和人交流用的是双方都能听懂和读懂的自然语言，同样，人和计算机交流也要用人和计算机都容易接受和理解的语言，这就是计算机语言。人们用自然语言讲述和书写，目的是给另外的人传播信息。同样，我们使用计算机语言把我们的意图表达给计算机，目的是使用计算机。

计算机语言指用于人与计算机之间通讯的语言。计算机语言是人与计算机之间传递信息的媒介。计算机系统最大特征是指令通过一种语言传达给机器。为了使电子计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规则，由这些字符和语法规则组成计算机各种指令(或各种语句)。这些就是计算机能接受的语言。

#### 2. 程序

计算机是一种具有内部存储能力的自动、高效的电子设备，它最本质的使命就是执行指令所规定的操作。电脑做的每一次动作、每一个步骤，都是按照已经用计算机语言编好的程序来执行，程序是计算机要执行的指令的集合，而程序全部都是用我们所掌握的语言

来编写的。

计算机语言的种类非常多，总的来说可以分成机器语言、汇编语言、高级语言三大类。但是，我们知道，机器语言是二进制编码，用它编制程序既难记忆，又难掌握，所以，计算机工作者就研制出了各种计算机能够识别、人们又方便使用的高级语言。C 语言就是高级语言中的一种，用 C 语言编写的语句集合就是 C 语言程序。

3. 程序语言的发展

程序语言的产生和发展，直接推动了计算机的普及和应用。计算机语言按使用方式和功能可分为低级语言和高级语言。低级语言包括机器语言和汇编语言。机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合，它与计算机同时诞生，是第一代的计算机语言；后来，人们用与代码指令实际含义相近的英文缩写词、字母和数字等符号来取代指令代码(如用 ADD 表示运算符号“ + ”的机器代码)，于是就产生了汇编语言，被称为第二代语言。不论是机器语言还是汇编语言都是面向硬件的具体操作的，语言对机器的过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对非计算机专业人员是难以做到的，对于计算机的推广应用是不利的。计算机事业的发展，促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的语意确定、规则明确、自然直观和通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称高级语言。高级语言是面向用户的语言。无论何种机型的计算机，只要配备上相应的高级语言的编译或解释程序，则用该高级语言编写的程序就可以通用。可见，高级语言只是要求人们向计算机描述问题的求解过程，而不关心计算机的内部结构，所以把高级语言称为“面向过程语言”，它易于被人们理解和接受。典型的面向过程语言有 C、BASIC、FORTRAN、COBOL、Pascal 等等。

随着计算机技术的迅猛发展，自从 20 世纪 80 年代以来，众多的第四代非过程化语言、第五代智能化语言也竞相推出。如果说第三代语言要求人们告诉计算机怎么做，那么第四代语言只要求人们告诉计算机做什么。因此，人们称第四代语言是“面向对象语言”。面向对象概念的提出是相对于面向过程的一次革命，面向对象技术在系统程序设计、多媒体应用、数据库等诸多领域得到广泛应用。但是，“面向过程”是程序设计的基础，尤其是对于程序设计的初学者。所以，我们将以面向过程的 C 程序设计语言为背景，主要介绍程序设计的基本概念和方法。

### 1.1.2 程序设计

什么是程序设计呢？在日常生活中我们可以看到，同一台计算机，有时可以画图，有时可以制表、有时可以玩游戏，诸如此类，不一而举。也就是说，尽管计算机本身只是一种现代化方式批量生产出来的通用机器，但是，使用不同的程序，计算机就可以处理不同

的问题。今天，计算机之所以能够产生如此之大的影响，其原因不仅在于人们发明了机器本身，更重要的是人们为计算机开发出了不计其数的能够指挥计算机完成各种各样工作的程序。正是这些功能丰富的程序给了计算机无尽的生命力。而程序设计就是用某种程序语言编写这些程序的过程。

更确切地说，所谓程序，是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述，这个描述的过程就称为程序设计。对数据的描述就是指明数据结构形式；对处理方法和步骤的描述也就是下一节我们要讨论的算法问题。因而，数据结构与算法是程序设计过程中密切相关的两个方面。

对于程序设计的初学者来说，首先要学会设计一个正确的程序。一个正确的程序，通常包括两个含义：一是书写正确，二是结果正确。书写正确是指程序在语法上正确，符合程序语言的规则；而结果正确通常是指对应于正确的输入，程序能产生所期望的输出，符合使用者对程序功能的要求。程序设计的基本目标是编制出正确的程序，但这仅仅是程序设计的最低要求。

一个优秀的程序员，除了程序的正确性以外，更要注重程序的高质量。所谓高质量是指程序具有结构化程度高、可读性好、可靠性高、便于调试维护等一系列特点。毫无疑问，无论是一个正确的程序，还是一个高质量的程序，都需要设计才能使之达到预期的目标。

那么，如何进行程序设计呢？一个简单的程序设计一般包含以下四个步骤：

(1)分析问题。对于给定的任务要进行认真的分析。研究所给定的条件，数据以怎样的形式组织，分析最后应达到的目标，找出解决问题的规律，选择解题的方法。

(2)设计算法。设计出解题的方法和具体步骤。

(3)编写程序。将算法翻译成计算机程序设计语言，对源程序进行编辑、编译和链接。

(4)运行程序，分析结果。运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，要对结果进行分析，看它是否合理。不合理要对程序进行调试，即通过上机发现和排除程序中的故障的过程，直至获得预期的结果。

由此可见，一个完整的程序要涉及四个方面的问题：数据结构、算法、编程语言和程序设计方法。这四个方面的知识都是程序设计人员所必须具备的，其中算法是至关重要的一个方面。如果我们对算法一无所知，就无法进行基本的程序设计。因此，下面一节我们对算法的基本概念、基本设计和表示方法作初步介绍。

## 1.2 算法和算法的表示

### 1.2.1 算法的概念

#### 1. 算法的基本概念

什么是算法？当代著名计算机科学家 D. E. Knuth 在他撰写的 *The Art of Computer Programming*一书中写道："一个算法，就是一个有穷规则的集合，其中之规则规定了一个解决某一特定类型的问题的运算序列。"简单地说，任何解决问题的过程都是由一定的步骤组成的，把解决问题确定的方法和有限的步骤称作为算法。

需要说明的是，不是只有计算问题才有算法。例如，加工一张写字台，其加工顺序是：桌腿→桌面→抽屉→组装，这就是加工这张写字台的算法。当然，如果是按"抽屉→桌面→桌腿→组装"这样的顺序加工，那就是加工这张写字台的另一种算法，这其中是没有计算问题的。

通常计算机算法分为两大类：数值运算算法和非数值运算算法。数值运算是指对问题求数值解，例如对微分方程求解、对函数的定积分求解、对高次方程求解等，都属于数值运算范围。非数值运算包括非常广泛的领域，例如资料检索、事务管理、数据处理等。数值运算有确定的数学模型，一般都有比较成熟的算法。许多常用算法通常还会被编写成通用程序并汇编成各种程序库的形式，用户需要时可直接调用，例如数学程序库、数学软件包等。而非数值运算的种类繁多，要求不一，很难提供统一规范的算法。在一些关于算法分析的著作中，一般也只是对典型算法作详细讨论，其他更多的非数值运算是需要用户设计其算法的。

下面通过三个简单的问题说明设计算法的思维方法。

【例 1-1】 有黑和蓝两个墨水瓶，有人错把黑墨水装在了蓝墨水瓶子里，而把蓝墨水错装在了黑墨水瓶子里，要求将其互换。

算法分析：这是一个非数值运算问题。因为两个瓶子的墨水不能直接交换，所以，解决这一问题的关键是需要引入第三个墨水瓶。设第三个墨水瓶为白色，其交换步骤如下：

(1)将黑瓶中的蓝墨水装入白瓶中；

(2)将蓝瓶中的黑墨水装入黑瓶中；

(3)将白瓶中的蓝墨水装入蓝瓶中；

(4)输出交换以后的结果；

(5)算法结束。

【例 1-2】 计算半径为 $r$ 的圆面积。

算法分析：本题是一个数值运算问题。计算圆面积的公式为 $\pi \times r^2$。根据计算机具有计算的基本功能，用计算机解题的算法如下：

(1)将 $r$ 的值输入计算机；

(2)判断 $r$ 是否大于等于 0($r \geq 0$?)，如果条件成立，执行第(3)步，否则执行第(4)步；

(3)按表达式 $\pi \times r^2$ 计算出圆面积，输出圆面积的值，然后执行第(5)步；

(4)输出提示“输入的 $r$ 值不能小于0”，然后执行第(5)步；

(5)算法结束。

【例 1-3】　输出 10 个随机数。

算法分析：这里要使用 C 语言类库中的 rand( )函数自动产生一个随机数，用计算机解题的算法如下：

(1)用变量 count 作为次数统计的计数器，初值为 1；

(2)使用 rand( )函数产生一个随机数并输出；

(3)计数器 count 的值加 1；

(4)返回第(2)步，直到计数器 count 的值大于 10，则算法结束。

由上述三个简单的例子可以看出，一个算法由若干操作步骤构成，并且这些操作是按一定的控制结构所规定的次序执行。

2. 算法的两要素

由上述两个例子可以看出，任何简单或复杂的算法都是由基本功能操作和控制结构这两个要素组成。不论计算机的种类如何之多，它们最基本的功能操作是一致的。计算机的基本功能操作包括以下四个方面：

(1)逻辑运算：与、或、非；

(2)算术运算：加、减、乘、除；

(3)数据比较：大于、小于、等于、不等于、大于等于、小于等于；

(4)数据传送：输入、输出、赋值。

需要强调的是，设计算法与演绎数学有明显区别，演绎数学是以公理系统为基础，通过有限次推演完成对问题的求解。每次推演都是对问题的进一步求解，如此不断推演，直到能将问题的解完全描述出来为止。而设计算法则是充分利用解题环境所提供的基本操作，对输入数据进行逐步加工、变换和处理，从而达到解决问题的目的。

### 1.2.2　算法的基本特征

算法是一个有穷规则的集合，这些规则确定了解决某类问题的运算序列。对于该类问题的任何初始输入值，它都能机械地一步一步地执行计算，经过有限步骤后终止计算并输

出结果。归纳起来，算法具有以下基本特征：

(1)有穷性：一个算法必须在执行有限个操作步骤后终止。

(2)确定性：算法中每一步的含义必须是确切的，不可出现任何二义性。

(3)有效性：算法中的每一步操作都应该能有效执行，一个不可执行的操作是无效的。例如，一个数被 0 除的操作就是无效的，应当避免这种操作。

(4)有零个或多个输入：这里的输入是指在算法开始之前所需要的初始数据。这些输入的多少取决于特定的问题。例如，例 1 - 1 的算法中没有输入，例 1 - 2 的算法中则需要输入一个初始数据 $r$，例 1 - 3 的算法中无需输入值。

(5)有一个或多个输出：所谓输出是指与输入有某种特定关系的量，在一个完整的算法中至少会有一个输出。如上述关于算法的两个例子中，每个都有输出。

通常算法都必须满足以上五个特征。需要说明的是，有穷性的限制是不充分的。一个实用的算法，不仅要求有穷的操作步骤，而且应该是使用尽可能少的操作步骤。例如，对线性方程组求解，理论上可以用行列式的方法。但是我们知道，要对 $n$ 阶方程组求解，需要计算 $n+1$ 个 $n$ 阶行列式的值，要做的乘法运算是 $(n!)(n-1)(n+1)$ 次。假如 $n$ 取值为 20，用每秒千万次的计算机运算，完成这个计算需要上千万年的时间。可见，尽管这种算法是正确的，但它没有实际意义。由此可知，在设计算法时，要对算法的执行效率作一定的分析。

### 1.2.3 算法的表示

原则上说，算法可以用任何形式的语言和符号来描述，通常有自然语言、程序语言、流程图、N - S 图、PAD 图、伪代码等。前面提到的例子是用自然语言来表示算法，而所有的程序是直接用程序设计语言表示算法。流程图、N - S 图和 PAD 图是表示算法的图形工具，其中，流程图是最早提出的用图形表示算法的工具，所以也称为传统流程图。它具有直观性强、便于阅读等特点，具有程序无法取代的作用。N - S 图和 PAD 图符合结构化程序设计要求，是软件工程中强调使用的图形工具。

因为流程图便于交流，又特别适合于初学者使用，对于一个程序设计工作者来说，会看会用传统流程图是必要的。本书只介绍和使用传统流程图表示算法。

#### 1. 流程图符号

所谓流程图，是以特定的图形符号加上说明，表示算法的图。流程图又称为框图，它用规定的一系列图形、流程线及文字说明来表示算法中的基本操作和控制流程，其优点是形象直观、简单易懂、便于修改和交流。美国国家标准化协会(American National Standard Institute，ANSI)规定了一些常用的符号，表 1 - 1 中分别列出了标准的流程图符号的名称和图形化表示形式。