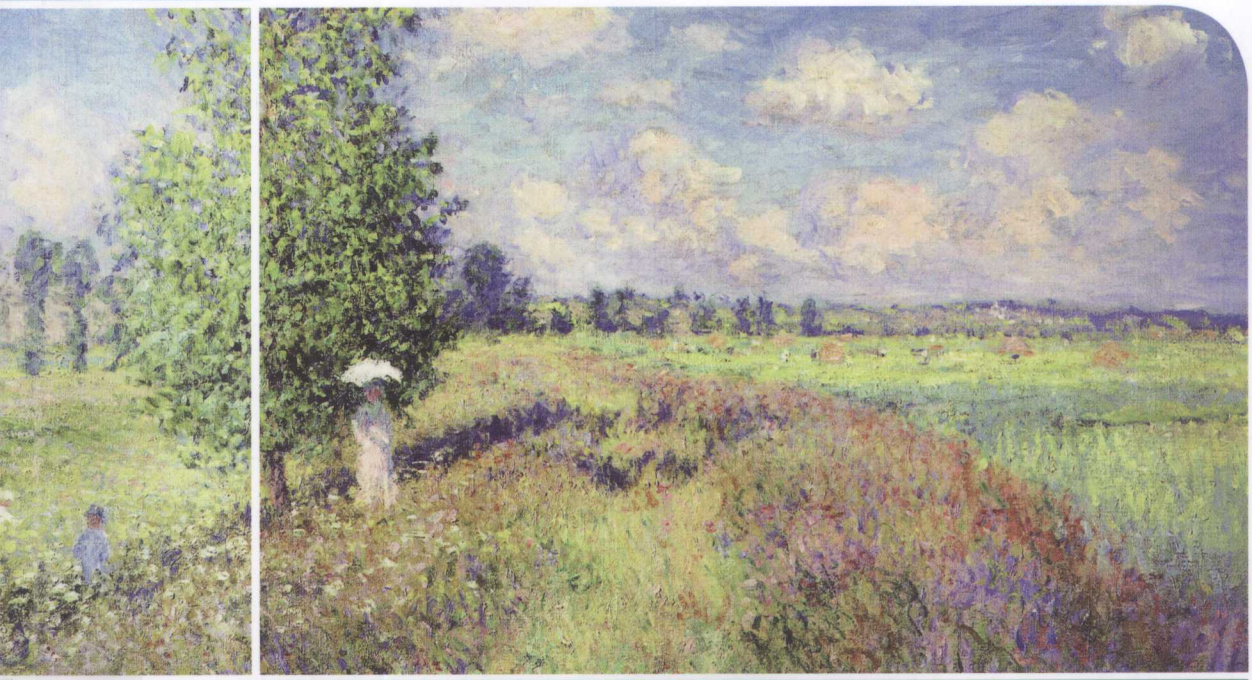
21世纪高等学校计算机
专业实用规划教材

程序设计基础

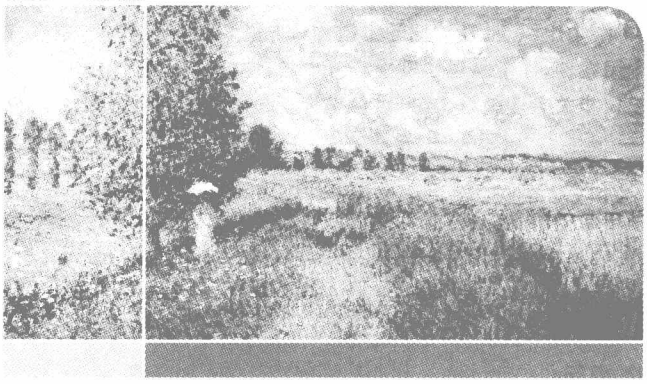
（C语言）（第2版）

◎ 张先伟 马新娟 张立红 王云 田爱奎 编著



清華大學出版社

21世纪高等学校计算机
专业实用规划教材



程序设计基础
（C语言）（第2版）

◎ 张先伟 马新娟 张立红 王云 田爱奎 编著

清华大学出版社
北京

内 容 简 介

本书重点介绍在C语言环境下编写程序的思路与方法。全书以程序设计的基本思想与方法作为主要结构，介绍了程序的基本结构组织、批量数据的组织方式与处理技巧，引入了递推、递归、动态规划、贪心等常用的算法设计方法应用案例，注重强调了程序设计中设计方法与动手实践。

本书可作为高等院校相关专业教材，亦可供从事计算机相关领域的科研人员参考自学。

图书在版编目(CIP)数据

程序设计基础：C语言/张先伟等编著. —2版. --北京：清华大学出版社，2016
21世纪高等学校计算机专业实用规划教材
ISBN 978-7-302-44086-4

Ⅰ. ①程…　Ⅱ. ①张…　Ⅲ. ①C语言－程序设计－高等学校－教材　Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2016)第132437号

责任编辑：刘　星
封面设计：刘　键
责任校对：李建庄
责任印制：王静怡

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：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：23.75　　**字　　数**：576千字
版　　次：2011年6月第1版　2016年9月第2版　　**印　　次**：2016年9月第1次印刷
印　　数：1～2000
定　　价：45.00元

产品编号：067530-01

第2版前言

程序设计基础是高等学校计算机专业学生的入门基础课程，本课程以C程序设计语言作为基本工具，以程序设计的思想与方法作为核心内容，以动手编程解决实际问题能力的培养为最终目标。通过课程的学习，不仅使学生掌握C程序设计语言本身的语法与结构，更重要的是逐步培养学生用计算机解决问题的思维、习惯与方法。

与传统的C语言教材相比较，本书虽然以C语言作为工具，但是不再以琐碎的C语言语法知识作为内容的核心，而是以程序设计的基本思想与方法作为主要结构，在理论体系中突出重点，淡化不常用、非必须且难理解的语法内容，引入常用的算法设计(如递推、递归、动态规划、贪心等)方法的应用案例与练习，使得教材内容的程序设计知识体系比较完整且重点突出。

本书内容共分为三部分，分别介绍程序的基本结构、批量数据的组织方式与处理技巧、常用算法的思路与应用典型案例。第一部分主要介绍程序结构与函数，力求结合案例以简练的内容介绍最常用的知识与离散数据的处理方法，是程序设计能力的基础；第二部分主要以经典案例与实践训练相结合的方法介绍批量数据的组织方法与处理技巧，是对程序设计能力的拔高；第三部分主要以基本算法理论与经典案例结合的方式介绍常用算法，是程序设计能力的升华。

本书第1版由田爱奎、张先伟策划、统稿。其中第1、2章由田爱奎编写，第7、10、11、12章由张先伟编写，第5、6、8、9章由张立红编写，第3、4章由王云编写。刘晓红、马新娟等在教材编写过程中也提出了许多有益的意见与建议。

本书第2版由张先伟、马新娟负责策划、修订，其中张先伟完成第1、2、8、9、13、14、15章的修订编写，马新娟完成第3、4、5、6、7、10、11、12章的修订编写。第2版对教材内容进行了较大的修订，重新调整了全书的章节结构，补充完善了更多有代表性的题目，尽量保证教材结构更趋合理、内容更加丰富、表述更加清晰。由于时间仓促，作者水平有限，书中难免有纰漏之处，敬请读者批评指正。

编　者

2016年5月

目　录

第1章 程序设计引论

1.1 计算机程序与计算机语言

1.1.1 计算机程序

随着信息技术的飞速发展，计算机的应用已经深入到人们日常生产生活的方方面面，大到类似载人太空宇宙飞船等高精尖技术的应用，小到日常生活中智能手机功能的开发，计算机日益成为人们不可或缺的助手。从表面来看，似乎计算机是无所不能的，可以自动完成许多工作，但实际上计算机所做的每一项工作都是由程序员预先设计的。只有程序员事先设计完成针对具体任务的计算机程序，并把这些程序代码输入到计算机中，计算机才能按照程序员所设计的程序代码依次执行每一条数据处理指令，从而完成任务要求的相关处理流程。

所谓计算机程序，是一组计算机能识别和执行的指令的集合，其中每一条指令使计算机执行特定的操作，这一组指令就是计算机处理某项具体事务过程中的一个操作序列（或称之为处理流程）。计算机执行该程序时，实际上是在“自动地”执行程序中的各条指令所对应的操作，从而有条不紊地完成处理工作。

编写一个计算机程序就像写一篇文章，写文章首先要学会基本的字、词、句等语法知识，因为文章本身就是一些基本字、词、句的组合体，不掌握这些语法知识就无法写出别人可以读懂的文章。但仅仅只是字、词、句的组合体还不是一篇精彩的文章，好的文章应该有精巧的组织结构来把这些字、词、句合理地组织在一起，通过这样的文章架构来表达出深刻的思想内涵。

同写文章类似，要想写出一个有实际用途的计算机程序，也至少需要两个基本要素：程序语言开发工具与程序设计方法。程序语言开发工具是程序实现的基本载体，正如写文章需要以字、词、句语法知识为基础一样，程序语言开发工具提供了计算机能够识别的各种数据表示与数据处理的“字、词、句”基本形式，提供了可以实现人机交互的基本工具；程序设计方法是程序设计的灵魂，它针对不同具体问题提供了有效解决问题的策略，也就是指定了一个计算机程序所需要的操作序列以及序列中这些操作之间的组合方式，正如在一篇文章中指定了构成的字、词、句以及这些字、词、句的组合方式一样。

总之，计算机的一切操作都是由程序控制的，计算机帮助人们解决实际问题的过程就是一个执行具体程序的过程。因此，程序是计算机系统中最基本的概念。只有了解程序设计，才能真正了解计算机是怎样工作的；也只有掌握了程序设计，也才能更好地使用计算机。

1.1.2 计算机内存

学习计算机程序设计,首先应该了解计算机工作的基本原理与计算机程序运行的基本机制。

现代计算机的主要部件有这样的几个部分:中央处理单元(CPU)担负着绝大部分的计算工作;随机访问存储器(RAM)作为一个工作区来保存程序与文件;永久存储器,一般是硬盘,负责永久地存储程序与文件;各种外围设备(如键盘、鼠标、显示器)用来实现人机之间的数据通信,如图 1.1 所示。

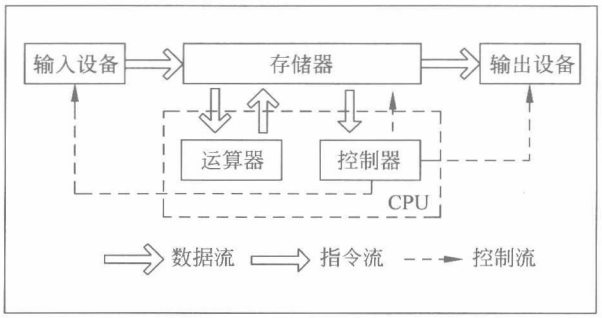


图 1.1 计算机的基本结构

中央处理单元 CPU 是进行数据计算的核心部件,它从内存中获取一个指令并执行该指令,然后从内存中获取下一条指令并执行。一个千兆 CPU 可以在一秒钟内进行大约一亿次这样的操作,所以 CPU 能以惊人的速度来从事枯燥的工作。

计算机程序最终可以分解成若干条 CPU 可以识别的指令,CPU 执行计算机程序,就是依次执行该程序对应的每一条指令的过程,就是一个不断地从内存读取数据、处理数据、往内存回写数据的过程。由此看来,计算机程序是与计算机内存紧密相关的。要了解计算机程序,必须先要了解计算机内存。

计算机程序执行时,组成程序的指令以及指令要处理的对象(数据)都必须存储到某个特定的位置,这个计算机内特定的存放位置就是内存,也称为主存(Main Memory)或者随机访问存储器(Random Access Memory,RAM)。RAM 是易失性存储器,关闭 PC 后,RAM 的内容就会丢失。因此在关闭 PC 之前,应该把 RAM 中需要保存的信息写入外存储器(如硬盘或者 U 盘等),这样再次需要时可以从外存中找到保存的程序或者数据信息。

计算机的 RAM 可以看做是井然有序的一排盒子,其中每个盒子都只有两种状态:满或者空,分别用二进制数的 1 和 0 来表示。每个盒子称为一位(bit),及二进制数(Binary Digit)的缩写。

为表示数据更加方便,内存中的位以 8 个为一组,每组的 8 位称为一字节(byte)。为了识别内存中的不同字节,根据这些字节在内存中的先后位置,依次对每个字节进行标记:第一个字节用 0 表示,第二个字节用 1 表示,直到内存中的最后一个字节。字节的这种内存位置的标记称为字节的地址(address)。每个字节的位置都是唯一的,不同的字节地址表示了内存中不同的存储位置。

计算机内存存储容量的常用单位是千字节(KB)、兆字节(MB)、吉字节(GB)、太字节

(TB)。这些字节的含义如下：

1KB 是 1024(2^{10})字节；

1MB 是 1024KB，也就是 1 048 576(2^{20})字节；

1GB 是 1024MB，也就是 1 073 741 824(2^{30})字节；

1TB 是 1024GB，也就是 1 099 511 627 776(2^{40})字节。

如果 PC 有 1GB 的 RAM，字节地址就是 0～1 073 741 823。为什么不使用人们日常生活中更习惯的十进制整数，如千、百万或者亿来表示？从 0 到 1023 共 1024 个数字，恰好是十位的二级制数所表示的最小的十个 0(00 0000 0000)到最大的十个 1(11 1111 1111)。与十进制数相比较，这种二进制的表示形式更加方便，0、1 的数据状态表示稳定且数值运算更容易实现。

1.1.3 计算机语言

语言是一种交流的工具。“人有人言，兽有兽语”，不管是人与人之间，还是动物与动物之间，不同个体之间要相互交流思想、传递感情，最直接的工具就是语言。人们要利用计算机解决实际问题，就要把自己解决问题的思想以特定的方式传递给计算机，让计算机能够按照所接受的信息去执行相应的操作，再把执行的结果反馈给用户，这是人与计算机之间交流的过程。人与计算机之间的交流也需要一种“语言”，借助这种语言，使得计算机和人都能读懂对方传递给自己的信息，从而实现人与计算机之间有效的信息交流。这种能够实现人与计算机交流的“语言”载体被称为计算机语言。

随着计算机技术的不断发展，计算机语言也经历了一个从低级到高级、从简单到复杂的发展过程。迄今为止，计算机语言发展主要经历了以下三个阶段。

1. 机器语言

用二进制位串形式表示的指令是计算机可以直接识别和执行的指令，人们把这样的指令称为机器指令(Machine Instruction)。机器指令是用 0 和 1 组成的一串代码，它们有一定的位数，并分成若干段，各段的编码表示不同的含义，例如某台计算机字长为 16 位，即有 16 个二进制数组成一条指令或其他信息。16 个 0 和 1 可组成各种排列组合，通过线路变成电信号，让计算机执行各种不同的操作。例如，某种计算机的指令为 1011011000000000，它表示让计算机进行一次加法操作；而指令 1011010100000000 则表示进行一次减法操作。它们的前八位表示操作码，而后八位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第 0 位算起的第 7 和第 8 位不同。这种机型可包含 256(2^8)个不同的指令，由于在一个任务中需要计算机执行的操作有很多，要使计算机识别和执行不同的操作，就要给出许多条由 0、1 位串组成的不同指令，计算机按照指令的要求执行各种操作。

由机器指令及其语法规则所构成的集合就是该计算机的机器语言(Machine Language)。机器语言又称为二进制代码语言，计算机可以直接识别，不需要进行任何翻译。由于每台计算机的指令、指令格式和代码所代表的含义都是硬性规定的，故称为面向机器的语言，简称机器语言。

机器语言是第一代计算机语言，对不同型号的计算机来说一般是不同的。既然是面向机器的，这种用机器语言编写的程序是计算机可以直接识别的符号，但是由于其 01 代码串的形式与人们习惯用的自然语言差别太大，不容易学，也不容易读。因此初期只有极少数的

计算机专业人员会编写计算机程序。

2. 汇编语言

为了使编程语言编写的程序更具备可读性,人们逐渐考虑用一些符号来表示指令中的操作码和地址码,就诞生了符号语言(Symbolic Language)。它是用一些英文字母和数字符号表示一个指令,例如用ADD代表"加法操作",SUB代表"减法操作",LD代表"数据传送"等。如机器指令1011011000000000可以改用符号指令代替:ADD A,B(执行A+B→A,将寄存器A中的数与寄存器B中的数相加,放到寄存器A中)。

由于计算机只能识别二进制位串形式的指令(比如1011011000000000),并不能直接识别和执行符号语言的指令如"ADD A,B",这时计算机就需要用一种称为汇编程序的软件,首先用该软件把用户用符号语言的表示的指令"ADD A,B"转换为机器指令1011011000000000,然后再完成该指令的执行。这种由符号语言的指令转换为对应机器语言指令的过程称为"代真"或"汇编",因此,符号语言又称为汇编语言(Assembler Language)。

汇编语言比机器语言易于读写、调试和修改,同时具有机器语言全部优点。但在编写复杂程序时,代码量仍然较大,而且汇编语言依赖于具体的处理器体系结构,不能通用,因此不能直接在不同处理器体系结构之间移植。人们把机器语言与汇编语言称为计算机低级语言(Low Level Language)。

3. 高级语言

由于汇编语言依赖于硬件体系,且助记符量大难记,于是人们又发明了更加易用的所谓高级语言。这种语言的语法和结构类似普通英文,语句表述形式符合人的自然语言表达的句式与习惯,且由于该语言远离对硬件的直接操作,不依赖于具体机器,用它写出的程序具备可移植性,所以该语言的编程知识更方便用户的学习与使用。

高级语言与人的"距离"更近,而与具体机器"距离"较远,因此计算机就不能直接识别由高级语言所编写的程序。程序运行之前要首先进行"翻译",目的是把机器不能识别的高级语言语句转换为可以直接识别的二进制位串指令形式。具体过程就是用一种称为编译程序的软件把用高级语言编写的程序(称为源程序,Source Program)转换为机器指令的程序(称为目标程序,Object Program),然后让计算机执行机器指令程序,最后得到结果。

从20世纪50年代第一个高级语言FORTRAN诞生之后,先后出现了2000多种不同的高级语言。每种高级语言都有其特定的用途,其中影响较大的有FORTRAN(20世纪50年代推出的第一个计算机高级语言)、ALGOL(适合数值计算)、BASIC(适合初学者的小型会话语言)、COBOL(适合商业管理)、Pascal(适合教学的结构程序设计语言)、PL/1(大型通用语言)、LISP和PROLOG(人工智能语言)、C(系统描述语言)、C++(支持面向对象程序设计的大型语言)、Visual Basic(支持面向对象程序设计的语言)和Java(适于网络的语言)等。

在上述高级语言中,C语言是Combined Language(组合语言)的简称,是一种功能强大且有代表性的语言。它既具有高级语言的特点,又具有汇编语言的特点。它可以作为操作系统设计语言,编写系统应用程序;也可以作为应用程序设计语言,编写不依赖计算机硬件的应用程序。因此,它的应用范围广泛,不仅仅是在软件开发上,而且各类科研(比如单片机以及嵌入式系统开发等)都需要用到C语言。

因此,本书选择以C语言作为工具载体,在C语言基本语法知识的基础上来介绍程序