



高职高专“十一五”规划教材·电子信息系列

C 语言程序设计

主编 连卫民 许磊

 大家出版社
全国优秀出版社



第 1 章 绪论 1

C 语言程序设计

第 1 章 绪论

1.1 C 语言的发展及其特点 1

高职高专“十一五”规划教材·电子信息系列

998

C 语言程序设计

105

主 编 连卫民 许 磊

2006

 人民邮电出版社

327

图书在版编目(CIP)数据

C 语言程序设计/连卫民,许磊主编. —郑州:大象出版社,2007.9

高职高专“十一五”规划教材·电子信息系列

ISBN 978 - 7 - 5347 - 4593 - 5

I. C… II. ①连…②许… 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 120953 号

本书编委会名单

主 编	连卫民	许 磊	
副 主 编	王桂芝	刘振民	郭志强
编 委	雷 燕	崔 琳	郝建丽
	王广亮	张社平	

责任编辑 陈洪东

特约编辑 樊园园

责任校对 钟 骄

封面设计 张 帆

出 版 大象出版社 (郑州市经七路 25 号 邮政编码 450002)

网 址 www.daxiang.cn

发 行 全国新华书店

印 刷 河南欣达印务有限公司

版 次 2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷

开 本 787 × 1092 1/16

印 张 21

字 数 485 千字

印 数 1—3 000 册

定 价 27.80 元

若发现印、装质量问题,影响阅读,请与承印厂联系调换。

印厂地址 郑州市海洋路中段

邮政编码 450002

电话 (0371)63720812



前言

C 语言是目前较好的学习程序设计的语言，C 程序设计课程是程序设计的重要基础课，是培养学生程序设计能力的重要课程之一。因此，学好 C 语言程序设计课程，对掌握基本编程方法、培养基本编程素质具有重要意义。

本书总结了作者多年的教学经验和以往各类 C 语言程序设计教材的优点，针对高职高专层次的学生，采用“以用促学”的编写原则，即通过编写实际应用程序来学习 C 语言抽象的标准和规则。本书不仅在内容上强调逻辑性，更注重介绍学习方法，使学生能根据例题举一反三。本书结构新颖、实例丰富，强调语言的规范和程序设计的方法与技巧，注重培养学生程序设计的思维方式和提高学生程序开发的能力。每章配有实例、小结、实训和练习题，方便学生学习、总结和上机训练，最后通过课程设计来提升学生的编程能力。本书共十三章：第 1 章程序设计概述，第 2 章 C 语言的基本知识，第 3 章顺序结构的程序设计，第 4 章选择结构的程序设计，第 5 章循环结构的程序设计，第 6 章模块化的程序设计，第 7 章指针的程序设计，第 8 章数组的程序设计，第 9 章构造的数据类型，第 10 章编译预处理与位运算，第 11 章文件的程序设计，第 12 章图形程序设计，第 13 章是课程设计。全书重点为第 6 章、第 8 章和第 11 章，难点为第 7 章和第 11 章。建议教学时数为 60~68 学时。另外，最好能安排 1~2 次程序设计知识讲座，介绍程序设计的发展和采用的工具，以及软件开发技术的发展趋势等内容。学生可以分组调查某一个计算机语言的特点、发展过程和应用，并以电子文档的形式上交。这样做可以提高学生的学习兴趣，并锻炼学生主动收集信息和整理信息的能力。

参与本书编写的教师都是多年来从事 C 语言程序设计课程教学的一线教师，他们积累了丰富的知识和教学经验，书中的许多知识

就是他们教学经验的总结。本书由连卫民、许磊任主编，王桂芝、刘振民、郭志强任副主编，参加编写的人员还有雷燕、崔琳、王广亮。

由于编写时间紧张，加之编者水平有限，书中不当之处敬请专家、读者批评指正。

编 者
2007年5月

目 录

1	第 1 章 程序设计概述	
1.1	学习 C 语言的目的	(1)
1.2	程序设计的基本知识	(3)
1.3	算法的描述方法	(8)
1.4	C 语言的发展与特点	(13)
1.5	Turbo C 2.0 集成环境简介	(14)
1.6	上机实训	(19)
	本章小结	(21)
	练习题	(22)
23	第 2 章 C 语言的基本知识	
2.1	数据类型概述	(23)
2.2	数据的表示	(24)
2.3	数据的运算	(33)
2.4	语句的表示	(40)
2.5	程序的组成	(42)
2.6	上机实训	(44)
	本章小结	(47)
	练习题	(47)
50	第 3 章 顺序结构的程序设计	
3.1	顺序结构程序的概念	(50)
3.2	数据的输入和输出	(51)
3.3	顺序结构程序设计实例	(58)
3.4	上机实训	(60)
	本章小结	(61)
	练习题	(62)
65	第 4 章 选择结构的程序设计	
4.1	选择结构程序的概念	(65)

	4.2 选择结构控制语句	(66)
	4.3 选择结构程序设计实例	(73)
	4.4 上机实训	(78)
	本章小结	(80)
	练习题	(80)
84	第 5 章 循环结构程序设计	
	5.1 循环结构程序的概念	(84)
	5.2 循环结构控制语句	(85)
	5.3 循环的嵌套	(89)
	5.4 循环结构程序设计实例	(93)
	5.5 上机实训	(98)
	本章小结	(103)
	练习题	(104)
109	第 6 章 模块化的程序设计	
	6.1 模块化结构的概念	(109)
	6.2 函数的定义及调用	(112)
	6.3 存储类型	(119)
	6.4 模块化程序设计实例	(131)
	6.5 上机实训	(138)
	本章小结	(142)
	练习题	(143)
149	第 7 章 指针的程序设计	
	7.1 指针的基本概念	(149)
	7.2 指针变量	(151)
	7.3 函数间用指针传递数据	(155)
	7.4 指向函数的指针变量	(159)
	7.5 指针程序设计实例	(160)
	7.6 上机实训	(162)
	本章小结	(164)
	练习题	(165)
169	第 8 章 数组的程序设计	
	8.1 数组的基本概念	(169)
	8.2 一维数组	(170)
	8.3 二维数组	(184)
	8.4 字符数组	(192)
	8.5 指针数组和指向指针的指针	(201)

	8.6 上机实训	(204)
	本章小结	(207)
	练习题	(208)
215	第9章 构造的数据类型	
	9.1 构造数据类型概述	(215)
	9.2 结构体类型的定义与使用	(217)
	9.3 共用体和枚举类型的定义与使用	(225)
	9.4 构造数据类型的程序设计实例	(229)
	9.5 上机实训	(236)
	本章小结	(239)
	练习题	(239)
243	第10章 编译预处理与位运算	
	10.1 编译预处理	(243)
	10.2 位运算	(249)
	10.3 上机实训	(259)
	本章小结	(260)
	练习题	(261)
265	第11章 文件的程序设计	
	11.1 文件的基本概念	(265)
	11.2 文件处理	(269)
	11.3 文件程序设计实例	(271)
	11.4 上机实训	(280)
	本章小结	(282)
	练习题	(282)
286	第12章 图形程序设计	
	12.1 图形程序设计的基本概念	(286)
	12.2 文本方式下的字符图形程序设计	(288)
	12.3 图形方式下的图形程序设计	(293)
	12.4 上机实训	(303)
	本章小结	(304)
	练习题	(305)
307	第13章 C语言课程设计	
310	附录	
	附录 I 字符的 ASCII 码表	(310)
	附录 II C 运算符的优先级与结合性	(311)

附录III	C 的关键字.....	(312)
附录IV	C 的库函数.....	(313)
附录V	常见的编译错误信息	(317)

327 参考文献

第 1 章 程序设计概述

程序设计就是利用某种计算机语言，按照一定的算法，对特定的数据进行处理。本章主要介绍学习 C 语言的目的、程序设计的基本知识、算法的描述方法、C 语言程序简介、Turbo C 2.0 集成环境简介。通过本章的学习，使读者了解 C 语言的基本特点，熟悉 C 程序的基本结构和程序设计的方法步骤，掌握算法描述的方法。

1.1 学习 C 语言的目的

C 语言是国际上广泛流行的计算机高级程序设计语言，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，也可用于应用软件的开发。本节主要介绍计算机系统的工作原理和学习 C 语言的重要性。

1.1.1 计算机系统的工作原理

1. 计算机系统组成

根据冯·诺依曼设计计算机的思想，计算机系统由计算机硬件和计算机软件两部分组成。计算机硬件是指组成计算机系统的设备或机器，是“看得见，摸得着”的物理部件，它是组成计算机系统的基础。计算机硬件一般包括中央处理器（CPU 即运算器和控制器）、内存储器、外存储器、输入设备和输出设备。计算机硬件之间的关系如图 1-1 所示。

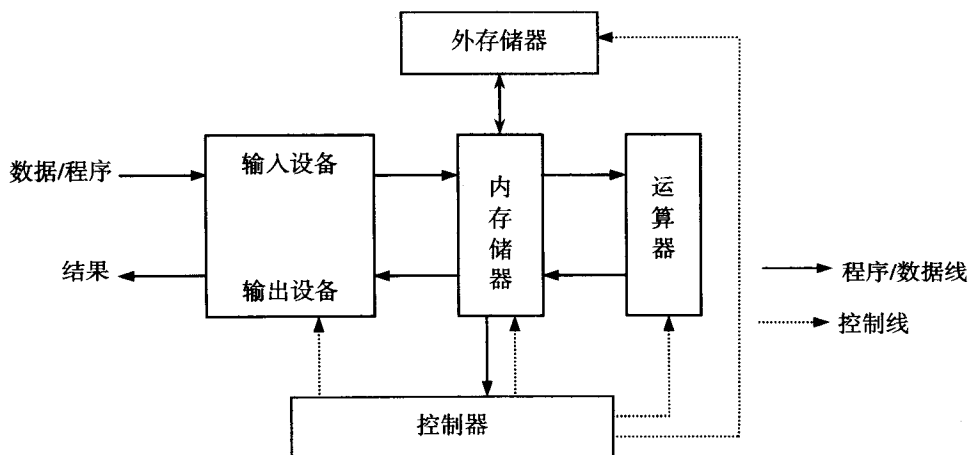


图 1-1 计算机硬件之间的关系



计算机软件是指组成计算机系统的程序、数据和文档。程序是指令的有序集合；数据是信息在计算机中的表示，是计算机处理的对象；文档是各种说明文本，是软件操作的辅助性资源。计算机的所有工作都必须在软件的控制下才能进行，没有软件的计算机称为“裸机”，是任何工作都不能做的。根据软件的作用可以把软件分为系统软件和应用软件。系统软件是支持和管理计算机硬件的软件，是服务于硬件的，它创立的是一个平台；应用软件是完成用户某项要求的软件，是服务于特定用户的，它满足某一个应用领域。软件的作用如图 1-2 所示。

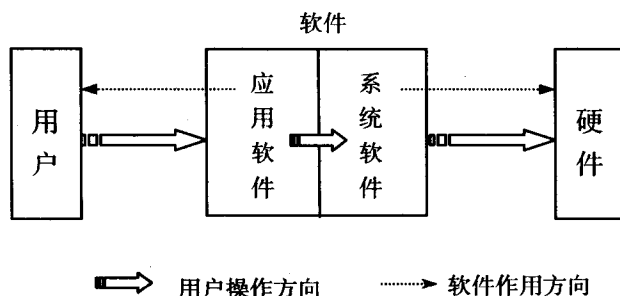


图 1-2 软件的作用

2. 计算机工作过程

开启计算机后，CPU 首先执行固化在只读存储器 (ROM) 中的一小部分操作系统程序，这部分程序称为基本输入输出系统 (BIOS)，它启动操作系统的装载过程，先把一部分操作系统从磁盘读入内存，然后再由读入的这部分操作系统装载其他的操作系统程序。装载操作系统的过程称为自举或引导。操作系统被装载到内存后，计算机才能接收用户的命令，执行其他的程序，直到用户关机。程序是由一系列指令所组成的有序集合，计算机执行程序就是执行这一系列指令。

(1) 指令和程序的概念。指令就是让计算机完成某个操作所发出的命令，即计算机完成某个操作的依据。一条指令通常有操作码和操作数两部分组成，操作码指明该指令要完成的操作，操作数指明要完成操作的对象。一台计算机的所有指令的集合，称为该计算机的指令系统。

使用者根据解决某一问题的步骤，选用一条条指令进行有序的排列。计算机执行了一个指令序列，便可完成预定的任务。这一个指令序列就称为程序。显然，程序中的每一条指令必须使用计算机的指令系统中的指令。因此指令系统是提供给使用者编制程序的基本依据。指令系统反映了计算机的基本功能，不同的计算机其指令系统也不同。

(2) 计算机执行指令的过程。计算机执行指令一般分为三个阶段。首先将要执行的指令从内存中取出送入 CPU；然后由 CPU 对指令进行分析译码，判断该条指令要完成的操作；最后向各个部件发出完成该操作的控制信号，完成该指令的功能。当一条指令执行完后就处理下一条指令。一般将第一阶段称为取指令阶段，第二阶段称为分析指令阶段，第三个阶段称为执行指令阶段。

(3) 程序的执行过程。计算机在运行时，从内存读出一条指令到 CPU 内执行，指令执



行完后再从内存读出下一条执行。CPU 不断的取指令、分析指令、执行指令。

总之,计算机的工作就是执行程序,即自动连续地执行一系列指令,而程序开发人员的工作就是编制程序。一条指令的功能虽然有限,但是在人们精心编制下的一系列指令组成的程序可完成的任务却是无限的。

1.1.2 学习 C 语言的重要性

在各种计算机语言尽显魅力的今天,选用哪种计算机语言学习程序设计最为合适呢?就技术性、基础性和实用性而言,仍当属 C 语言最具有代表性,这也是国内外高校计算机教育经久不衰的原因之一。

第一,C 语言语法结构简洁,写出的程序高效。它便于描述算法,大多数的程序员愿意使用 C 语言去描述算法。所以,如果想在程序设计方面有所建树,就必须首先学好它。

第二,C 语言能够深入系统底层,编程流畅,效率高,应用广泛。我国高校 1993 年前后逐步引入了 C 语言。C 语言最早产生于 Unix 操作系统平台,后来普及到微型计算机上,由于其编码效率较高,能进行底层操作,如机器的中断地址、设备端口、寄存器、字位操作等,还可以处理字符和图形、编写界面;它使用方便,是开发系统软件的良好选择;基本上很多操作系统,都是 C 语言写的,所有的 Windows、Unix、Linux、Mac、os/2,没有一个例外,如果不懂 C 语言,怎么可能深入到这些操作系统当中去呢?更不要说去写它们的内核程序了。

第三,许多新型的语言都是衍生自 C 语言,它也是现代网络编程如 Java、面向对象可视化编程 Visual C++ 等程序设计的基础。掌握了 C 语言,就掌握了很多门语言,经过简单的学习,可以用这些新型的语言去开发了。这再一次验证了 C 语言是程序设计的重要基础。还有,即使现在招聘程序员,都是考 C 语言,所以一定要掌握好 C 语言。

如今,学好计算机技术已经成为人们完善自我、提高工作技能的重要需求,是人们现代生活的一部分。因为相当多的职业都离不开计算机,而计算机的发展与运用正是现代科学技术发展的重要标志。对于学生来说,充分利用和掌握计算机技能,才能达到所期望的职业技能应用水平,如果能掌握一些硬件的基本知识,再掌握 C 语言程序设计方法,将成为就业的优势之一。

1.2 程序设计的基本知识

程序设计就是利用某种计算机语言,按一定的算法,对特定的数据进行处理。程序设计的最终结果是用某种计算机语言描述的计算机程序。本节主要介绍程序设计概念、程序设计步骤、程序设计语言、程序设计方法、程序设计评判标准。

1.2.1 程序设计概念

程序设计就是设计程序,而程序=算法+数据结构,算法必须与计算机处理问题的步骤一致。所以,程序设计的主要考虑两个因素,就是算法和数据结构。



1. 算法

算法是为了解决某个特定问题而采取的确定且有限的步骤。所以，算法的要素主要是操作和控制结构。即一个算法由一些操作组成，这些操作又是按一定的控制结构所规定的次序执行的。计算机可以执行的基本操作是以指令的形式向人们提供的，指令是计算机所完成的最基本的功能。

计算机的基本操作有：算术运算、关系运算、逻辑运算、数据传送即赋值运算。算法的功能不仅取决于所选用的操作，而且还取决于控制结构。算法的基本控制结构有顺序、选择和循环三种。算法的控制结构可以用图示的方法形象地表示出来。这一点将在下一节讲解。

一个算法具有以下三个特性：

(1) 有穷性。一个算法通过若干步骤即可实现预定目标，算法的每一步都可以在合理的时间内完成。

(2) 确定性。算法中的每一条指令都必须有确切的含义，同样的步骤、同样的条件，结果也是相同的。

(3) 输入和输出。用计算机解决问题可以分为三步，如图 1-3 所示。

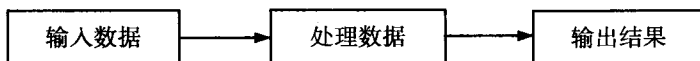


图 1-3 计算机的工作步骤

首先，把要处理的数据输入到计算机中；然后，根据处理要求对输入的数据进行处理，得到处理结果；最后，把结果显示或打印出来。

计算机工作的步骤具有不可逆的方向性。再简单或再复杂的计算机概莫能外。也就是说，一个算法再简单，也要有输入、处理、输出三个步骤。

2. 数据结构

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。定义中的“关系”描述的是数据元素之间的逻辑关系，称为逻辑结构；数据结构在计算机中的表示（映象）称为数据的物理结构，有顺序结构（如数组）和随机结构（如链表）两种；数据结构要解决的三个问题是：数据的表示、数据元素间的关系以及在这种关系上的运算。

从计算机语言上看，数据结构体现为数据类型。数据类型是一个值的集合和定义在这个值集上的一组操作的总称。它是信息特性的抽象。

从硬件上看，引入数据类型的目的是为了了解释计算机内存中信息含义的一种手段；从用户的角度看，引入数据类型实现了信息的隐蔽，即将一切用户不必了解的细节都封装在类型中。

3. 算法与数据结构的关系

算法与数据结构是人们用计算机解题时所作的两种抽象。算法是从计算机操作的角度对解题过程的抽象；数据结构是从如何组织被处理对象的角度进行抽象。这两种抽象相互依赖，互为补充，有效地降低了问题的复杂性。



1.2.2 程序设计步骤

在程序设计的过程中,注意按照以下步骤进行:数据在计算机内如何存储,数据如何输入,数据如何处理,数据如何输出。其处理过程如图1-4所示。

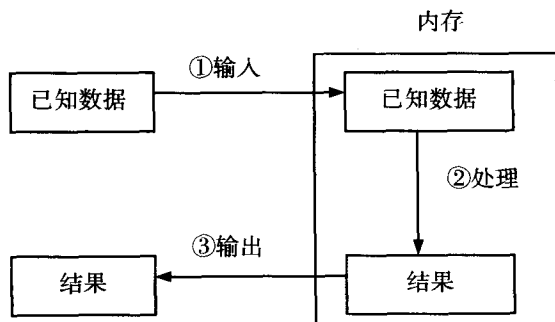


图1-4 计算机处理问题的过程

首先要将计算机之外的信息输入到计算机内,然后由已知数据根据一定的方法进行处理,得到结果数据,最后把计算机结果输出到显示器或打印机上。

1. 数据存储

数据在计算机内存中的存储是通过变量来表示的,这就要重点考虑变量的数据类型。在计算机语言中,数据类型确定了这种数据的存储长度、表示范围和所允许的运算。如在C语言中,整型数据的存储长度是2个字节,数据的表示范围是-32768到32767,可以进行算术运算、关系运算和逻辑运算。

数据类型一般有基本的数据类型和构造的数据类型之分。基本的数据类型就是计算机语言已经定义的数据类型,即标准的数据类型,用户可以直接使用的数据类型。构造的数据类型是需要用户在使用之前先定义的数据类型。

2. 数据输入

数据输入就是指变量如何获得数据,即获得数据的途径。变量获得数据的途径有三种:从程序中获得,从键盘上获得,从文件里获得。

(1) 从程序中获得。即把要处理的数据通过指令传给变量。在C语言中,是通过给变量赋初值的形式来实现的。例如:

```
int x=12;      /* 在定义变量x为整型数据的时候,给x赋值12 */
```

(2) 从键盘上获得。即在编程时不考虑变量的具体值,在程序运行时,再从键盘上给变量赋值。用于临时数据或即时数据的输入。

(3) 从文件中获得。即把处理的数据事先存放在文件里,在编程时通过对文件的操作把数据传给变量。文件是指存放在外存上的信息集合,用于存储需要重复处理和长久存放的数据。

3. 数据处理

数据处理是指由已知数据求未知数据的过程。重点是考虑算法,即处理语句和控制结构。常用的算法有:数据的组织、筛选与排序,递推与递归,链表与文件的处理。



4. 数据输出

数据输出是指把处理的结果显示、打印或存储到文件里。

(1) 输出到显示器。输出到显示器是指把处理的结果按照一定的格式输出到显示器上。重点在于控制输出的格式，如每个数据的输出宽度，一行输出几个数据，共输出多少行等。这也是初学编程的主要输出方式，需要重点掌握。

(2) 输出到打印机。输出到打印机是指把处理的结果按照一定的格式输出到打印机上。重点也在于控制输出的格式：每个数据的输出宽度，一行输出几个数据，输出多少行。

(3) 输出到文件。输出到文件是指把处理的结果按照一定的格式输出到外存的文件里。重点在于文件的类型，即是文本文件，还是二进制文件。

1.2.3 程序设计语言

程序设计语言，也称为编程语言，是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。程序设计语言分为机器语言、汇编语言和高级语言。

1. 机器语言

机器语言属于低级语言，与特定的机器有关，功效高但使用复杂、繁琐、费时，易出差错。机器语言是表示成数码形式的机器基本指令集，或者是操作码经过符号化的基本指令集。机器语言是由 0 和 1 组成的二进制代码指令的集合，计算机 CPU（中央处理器）能够直接识别并执行，不需要任何翻译程序。

2. 汇编语言

汇编语言是用助记符表达的计算机语言，是一种符号语言。20 世纪 50 年代初出现了汇编语言。汇编语言和机器语言相比，表示方法有一定的改进，比二进制代码容易书写，也容易记忆。汇编语言的编译效率很高，最接近机器指令。汇编语言在表达算法方面，还不能尽如人意，随即就有了更近似人们日常生活中使用的高级语言。

3. 高级语言

高级语言的出现使得计算机程序设计语言不再过度地依赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被编译成不同的机器语言，而不是直接被机器执行。C 语言属于高级语言。

高级语言的表示方法要比低级语言更接近于待解问题的表示方法，其特点是在一定程度上与具体机器无关，易学、易用、易维护。而且不同的计算机系统上所配置的高级语言基本上都是相同的，即高级语言具有很强的通用性，这样描述程序算法显然就得心应手得多。

程序设计语言的基本成分有：① 数据成分，用于描述程序所涉及的数据；② 运算成分，用于描述程序中所包含的运算；③ 控制成分，用于描述程序中所包含的控制；④ 传输成分，用于表达程序中数据的传输。

程序设计语言是软件的重要基础，其发展趋势是模块化、简明化、形式化、并行化和可视化。



1.2.4 程序设计方法

程序设计的方法包括结构化的程序设计方法和面向对象的程序设计方法。

1. 结构化的程序设计方法

结构化程序设计方法可归结为“程序 = 算法 + 数据结构”，其核心是算法设计，是一种面向过程的设计方法，使数据与数据处理分离。基本思想是采用自顶向下、逐步细化的设计方法和单入单出的控制结构。自顶向下、逐步细化指将一个复杂任务按照功能进行分解，形成由若干模块组成的树状层次结构，逐层细化到便于理解和描述的程度，各模块尽可能独立；而单入单出的控制结构，指的是每个模块内部均用顺序、选择、循环三种基本结构来描述，整个模块一个入口，一个出口。结构化的程序设计方法将程序分解成若干个可单独命名和编址的功能模块，如图 1-5 所示。

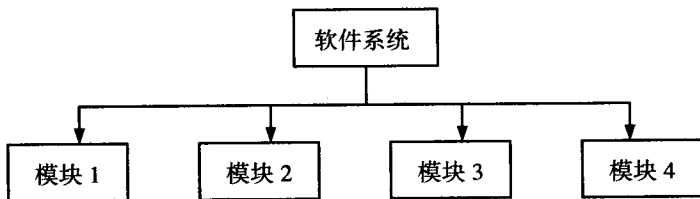


图 1-5 结构化的程序设计方法

结构化程序设计方法为处理复杂问题提供了有利手段，一度成为程序设计的主流方法。C 语言即为目前应用比较广泛的结构化程序设计语言。

然而，随着计算机应用的不断发展，这种设计方法开始逐渐暴露出缺陷。如在大型多文件软件系统中，随着数据量的增大，程序变得越来越难以理解。另外，采用结构化方法所设计的程序，可重用性比较差。于是，又出现了面向对象的程序设计方法。

2. 面向对象的程序设计方法

结构化的程序设计方法的缺点的根源在于数据与数据处理分离。而面向对象的程序设计方法正是克服了这个缺点，同时吸纳了结构化设计思想的合理部分而发展起来的。这两种方法并非对立关系。

面向对象的设计思想模拟人们对自然界认识和处理事物的方法，将数据和对数据的处理方法放在一起，形成一个相对独立的整体——对象，对同类型对象抽象出共性，形成类。任何类中的数据都只是用本类自有的方法进行处理，并通过简单的接口与外部联系。对象通过消息进行通信。面向对象的程序设计方法同样遵循分层抽象、逐步细化的原则。该方法的核心是从问题中抽象出合适的对象，即首先解决“做什么”的问题。至于“怎么做”，则以操作的设计封装在对象内部。对于操作方法的设计，核心仍然是算法设计，完全吸收了结构化程序设计的思想。面向对象的方法把软件系统分解成为相互协作而双彼此独立的对象的集合，如图 1-6 所示。

面向对象方法应该尽量选用支持面向对象技术的语言来实现。目前比较流行的面向对象编程语言有 C++、Java 等，它们都是以 C 语言为基础的。