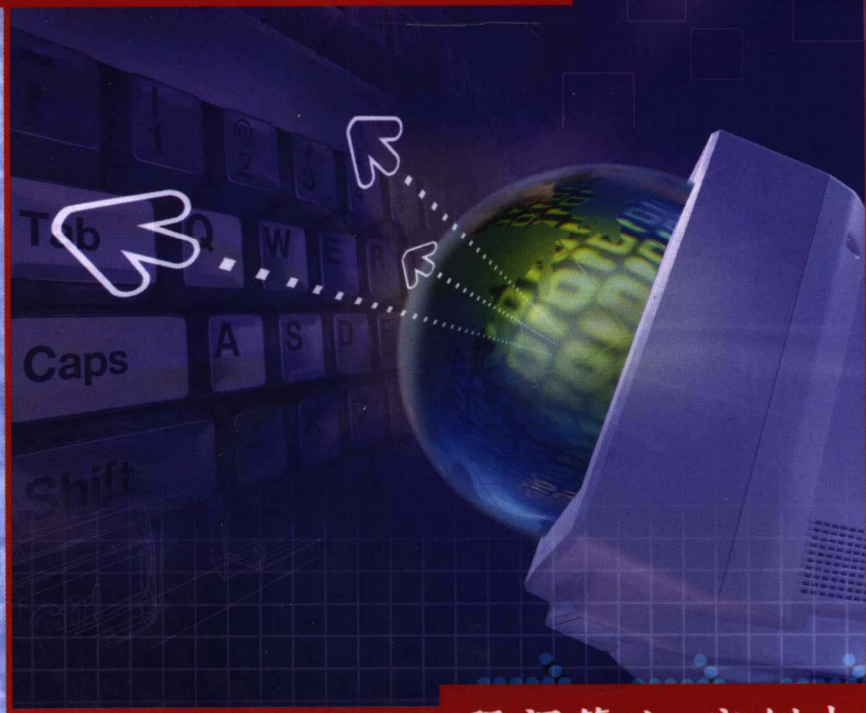


21世纪高等院校计算机教材

C语言程序设计

夏宽理 赵子正 编著

以读者学习程序设计方法为主导
以算法为依据介绍程序设计过程



强调算法 实例丰富

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪高等院校计算机教材

C 语言程序设计

夏宽理 赵子正 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书主要讲解C语言的基本概念及常用设施,包括基本数据类型、结构化控制结构、数组、函数、指针、结构及数据文件等;同时还讲解了计算机程序设计的基本原理、程序设计技巧与方法、常用算法设计方法,以及C++语言面向对象程序设计的基本概念和机制(包括对象、类、继承、多态性及数据流等)。

本书的特点是以读者学习程序设计方法为主导,实例丰富,以算法为依据介绍程序的设计过程。

本书可作为高等院校各类专业学习计算机程序设计的教材,也可作为参加“计算机软件专业技术资格和水平考试”人员学习程序设计的教材,同时也可作为各类进修班、培训班讲授程序设计课程的教程和参考书。

图书在版编目(CIP)数据

C语言程序设计/夏宽理,赵子正编著. —北京:中国铁道出版社,2006.1

21世纪高等院校计算机教材

ISBN 7-113-06924-X

I. C... II. ①夏...②赵... III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2006)第006313号

书 名: C语言程序设计

作 者: 夏宽理 赵子正

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 秦绪好

责任编辑: 严 力 崔晓静 王慧亮

封面设计: 薛 为

封面制作: 白 雪

印 刷: 北京新魏印刷厂

开 本: 787×1092 1/16 印张: 21 字数: 506千

版 本: 2006年2月第1版 2006年2月第1次印刷

印 数: 1~5 000册

书 号: ISBN 7-113-06924-X/TP·1735

定 价: 27.00元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前言

FORDWORD >>>

几乎全社会都在学习计算机，但学习计算机应用软件的使用方法与学习程序设计有重要区别。

学习某种应用软件使用方法目的是学会已有计算机工具软件的使用方法，用工具软件协助完成人们的工作。例如作家学习某种字处理软件的使用方法，以便用计算机作为写作工具；工人学习计算机控制系统的操作方法，以便能正确使用计算机控制生产过程等。上述情况是人们在利用预先开发的应用软件完成他们的工作。

学习程序设计是学会用计算机解决实际问题的本领，当遇到暂时还不能得到能解决问题的软件时，自己动手编写一个能解决现有问题的程序是学习程序设计最基本的目的。学习程序设计就是学习独立开发程序本领的第一步。

对于没有学过计算机的人来说，计算机几乎是什么都会做的智能机器，当读者学了计算机后，就会发现，计算机是什么都不知道怎么做的机器，要让计算机去完成一项新的任务，就必须为它编写一个能让计算机正确完成该项任务的程序。计算机运行的过程就是计算机执行程序的过程。

要学习用计算机开发应用系统的本领，需要学习计算机领域方方面面的知识。在所有必须学习的课程中，首先要学习程序设计，并在以后进一步学习其他课程中继续提高程序设计能力，直到达到具有很强的程序设计动手能力，能达到非常熟练的编写程度。

学习程序设计是一件非常辛苦的事情，要读者有非常强的耐心和实践精神。学好程序设计的主要困难是要学习和熟练掌握一门与人们习惯使用的自然语言非常不一致的程序设计语言。要学会能熟练运用程序设计语言描述计算机求解问题的算法，要学习程序设计的许多常规算法、要求不断学习计算机的新知识。由于程序设计是一门实践性非常强的课程，还要求在学习程序设计基本技能的同时，不断地上机实践，通过实践积累经验。

实践证明，学习程序设计最困难的可能还是人们很难适应计算机算法的思维习惯，人们几乎无法承受计算机算法必须描述得几乎绝对的精细和精确。但对计算机来说，这又是非常必要的。

学习程序设计是达到掌握开发软件系统技术这个大目标的第一步。开发一个程序，特别是一个软件系统，是一件非常复杂的工作，要经历许多阶段。对于一个功能相对简单的计算任务来说，编写一个相对简单的程序，则是程序设计和程序编码阶段的任务。培养开发程序系统的能力，这要从设计和编写简单的程序开始。

编写程序，就是将解决问题的算法用某种程序语言描述后告诉计算机，为此人们为编制计算机程序研制了许许多多的计算机程序语言。本书采用 C 语言作为计算机程序设计的描述语言，这是因为 C 语言是一种具有功能丰富、表达能力强、使用灵活方便、可移植性好等优点的程序设计语言。另外，C 语言在许多方面反映计算机的计算过程，这对于希望深入了解计算机的读者来说，对于他们以后的学习是很有帮助的。特别是后来它引入的面向对象机制，从它发展而来的 C++ 语言更是现在广泛应用的程序语言。它的许多概念和描述方法被现行的

更好的程序语言所采纳，学好了 C 语言就能方便地学习并使用 C++语言和 Java 语言。由于 C++语言是一个更好的 C 语言，一些对编写程序更方便的设施也在介绍 C 语言时一并介绍，建议读者尽量采用 C++语言更简便的设施编写程序。

开发 C 语言的最初目的是研制一种编写系统软件的程序设计语言，为此引入了许多为提高程序执行效率和编制大型程序系统为目的的概念和机制。虽然 C 语言的描述多样性和灵活性，会给初学程序设计的读者对某些方面的正确理解带来一些困难，但这些概念对于读者更进一步理解计算机程序设计的内容会有很大的帮助。本书力求概念叙述准确、内容介绍循序渐进，设法让读者准确了解和完整掌握程序语言的概念和编程方法，通过实践，达到能熟练使用程序语言编写程序的目的。

要能熟练地进行程序设计，除需要掌握一种程序设计语言外，还需要掌握算法、数据结构以及程序设计技巧和方法等多方面的知识。本书特别注重介绍如何正确编写程序，详细地介绍从算法开发到程序编写的全过程。本书在介绍许多实例程序时先给出求解算法的设计过程，最后才给出程序。这能充分体现本书介绍程序设计方法的目标，让读者真正学到程序设计方法、学会如何编写程序，而不只是一些程序设计语言的知识。本书介绍的有关常用算法的设计方法更是进一步迎合了这个目标。让读者学习本书后，不仅能正确了解 C 语言、掌握初步的程序设计方法和技巧，并对最经常使用的算法设计方法也有一定的了解。

本书共分 10 章，各章内容安排如下：

第 1 章介绍程序设计基本概念和 C 语言的基础知识。

第 2 章介绍基本数据类型、各运算符的意义和表达式的书写规则，同时介绍了数据输入输出的基本方法。

第 3 章介绍控制结构，编写简单程序的方法。从这章开始，读者已开始学习编写程序的方法和技巧。

第 4 章介绍数组，学习处理成组数据的处理技术。内容包括数组的基本概念，定义数组和使用数组的方法，还介绍了字符串的处理技术。掌握这些内容，读者已具有组织和处理成批数据所必需的知识 and 能力。

第 5 章介绍函数，学习函数的使用方法和函数编写方法，递归函数的基本知识，局部变量、外部变量、变量的作用域、存储类等概念。还简单介绍了编译预处理命令。学习函数的编写方法，关键是掌握函数形参的设置方法。本章通过实例详细说明基本数据类型、指针类型和数组类型形参的设定方法和使用方法。为了帮助读者掌握编写函数的技术，本章还列举了大量的实例。至此，读者已具有将具有独立功能的程序段编写成函数的能力，为编写更复杂的程序打下一定的基础。

第 6 章介绍指针和引用的概念和使用方法，指针变量和数组的关系、多级指针和指针变量应用实例等。引用是 C++语言的设施，由于在某些特定场合，函数定义引用形参比指针形参更方便。从实用意义出发，建议读者能掌握引用的概念和使用方法，能在函数中正确使用引用形参。

第 7 章介绍结构、结构数组、结构指针、结构形参、结构指针形参、返回结构函数等内容。本章还特别重点介绍了链表的基本处理技术与应用；同时还介绍联合、位域、枚举和类型定义方面的一些基本知识。掌握这些内容，读者已具有编写处理复杂数据结构程序的能力。

第8章介绍数据文件处理技术,文件的基本概念、文件的使用方法和常用文件操作库函数的用法,以及基于文件的应用程序结构和程序实例。这方面知识是编写数据文件处理程序所必需的。

第9章介绍常用算法设计方法,内容包括迭代法、穷举法、递推法、回溯法、贪婪法、分治法、动态规划法。这些内容帮助读者了解计算机程序经常采用的算法设计方法。

第10章介绍C++面向对象程序语言的基本概念和机制、类和抽象数据类型、运算符重载、继承和C++输入输出流等基本知识。为进一步学习和使用C++语言提供必要的帮助。特别是用C++语言的开发环境作为学习程序设计的实习环境的读者,在学习C程序设计时,同时了解C++语言是一种非常好的学习程序设计的方法。

本书主要是为学生学习程序设计而编写的,与其他介绍C语言程序设计的教材相比,主要有两个特色,一个特色是有大量的程序设计实例。在实例程序设计中,强调介绍程序的开发过程,即通过分析问题,先用逐步求精方法寻找问题的求解算法,最后给出问题的程序解。笔者认为,这正是一般小程序或程序模块设计的完整过程。另一个特色是进一步介绍了算法设计的方法,这是因为程序设计是与设计算法紧密相连的,其目的是让读者对程序设计的有关内容有更全面的了解,并使读者学习了本书以后,不仅了解程序语言,并确实能用程序语言编写程序。通过进一步上机实践,逐步达到能独立编写应用程序。

本书也非常适宜于用做自学程序设计的教材,也可作为计算机软件专业技术和水平考试及计算机培训班的教材和参考书。

在本书编写过程中,除夏宽理、赵子正老师直接参与编写外,王春森、杭必政、金惠芳、陈海建、薛万奉等老师对本书的内容、实例的选择等作了很大的贡献,本书在编写过程中还得到多名老师关心和支持,他们给本书的内容提出了许多宝贵的建议,在此深表谢意。

有许多老师从事过和正从事着程序设计教学和编写程序设计教材,他们为程序设计教学积累了非常宝贵的教学经验。本教材中难免有不足之处,我们诚恳期待使用本书的老师和读者的批评指正和建议,以供再版时参考,使本书日臻完善。

笔者电子邮件地址: klxia@fudan.edu.cn

通信地址: 上海复旦大学计算机科学与工程系(邮编 200433)

编 者

2006年1月

目 录

CONTENTS >>>

第 1 章 程序设计基础	1
1.1 计算机和程序设计基础知识	1
1.2 C 语言的历史和特点	10
1.3 几个简单的 C 程序	11
1.4 C 语言的词汇、数据类型、常量和变量	15
1.5 程序开发环境基础知识	17
习题一	17
第 2 章 基本数据及其运算	19
2.1 基本类型数据	19
2.1.1 整型数据	19
2.1.2 浮点型数据	21
2.1.3 字符型数据	22
2.2 输入和输出基础	24
2.2.1 单个字符输出和输入	25
2.2.2 格式输出和输入	26
2.2.3 输入输出流 cin 和 cout	28
2.3 数据运算	31
2.3.1 赋值运算	31
2.3.2 算术运算	32
2.3.3 关系运算和逻辑运算	34
2.3.4 条件运算	35
2.3.5 其他运算	36
2.4 表达式	39
2.4.1 表达式分类	39
2.4.2 表达式的类型转换	40
习题二	42
第 3 章 结构化程序设计	45
3.1 基本语句	45
3.2 顺序结构	47
3.3 选择结构	48
3.3.1 两路条件选择结构	48
3.3.2 多路按值选择结构	53
3.4 循环结构	56
3.4.1 while 循环结构	56

3.4.2 do...while 循环结构.....	58
3.4.3 for 循环结构.....	60
3.4.4 三种循环结构比较.....	63
3.4.5 嵌套的循环结构.....	64
3.5 简单程序设计实例.....	66
习题三.....	72
第4章 数组	75
4.1 数组的基本概念.....	75
4.2 一维数组.....	76
4.3 多维数组.....	85
4.4 字符串处理技术基础.....	91
习题四.....	100
第5章 函数	103
5.1 函数的基本概念.....	103
5.2 库函数的使用方法.....	105
5.3 函数定义.....	106
5.4 函数调用.....	108
5.5 函数说明.....	113
5.6 递归函数基础.....	114
5.7 函数程序设计实例.....	117
5.8 存储类别和作用域.....	121
5.9 编译预处理命令简介.....	125
习题五.....	133
第6章 指针和引用	136
6.1 指针基本概念.....	136
6.2 指向数组元素的指针.....	140
6.3 指针形参.....	143
6.4 数组形参.....	147
6.5 指向二维数组一整行的指针.....	150
6.6 指针数组.....	152
6.7 多级指针.....	155
6.8 函数指针.....	156
6.9 返回指针值的函数.....	160
6.10 引用.....	162
习题六.....	166
第7章 结构和链表	170
7.1 结构类型和结构变量.....	170
7.2 结构数组.....	176

7.3 结构与函数	178
7.4 链表	181
7.5 联合	199
7.6 位域	201
7.7 枚举	202
7.8 类型定义	204
7.9 变量定义	206
习题七	208
第 8 章 数据文件处理技术	210
8.1 文件类型和文件类型指针变量	210
8.2 几个常用的数据文件库函数	211
8.3 文件处理程序结构	217
8.4 文件处理程序实例	220
习题八	230
第 9 章 算法设计技术基础	232
9.1 迭代法	232
9.2 穷举法	234
9.3 递推法	236
9.4 回溯法	238
9.5 贪婪法	244
9.6 分治法	248
9.7 动态规划法	250
习题九	254
第 10 章 C++语言简介	258
10.1 面向对象程序语言的基本概念和机制	258
10.2 C++语言对 C 语言更多的改进	260
10.3 类和抽象数据类型	264
10.4 运算符重载	280
10.5 继承	282
10.6 虚函数和多态性	288
10.7 C++输入输出流	292
习题十	301
附录 A 运算符的优先级与结合性	305
附录 B ASCII 字符集	306
附录 C Visual C++使用方法简介	307
附录 D C 语言常用语法提要	316
附录 E 常用库函数	320
参考文献	324

第 1 章 | 程序设计基础

欢迎大家学习计算机程序设计，正如本书前言所说，学习程序设计是一件非常辛苦的事情。笔者也非常负责地告诉你，只要有非常强的耐心和实践精神，一定能学好程序设计。

学习程序设计是达到掌握开发软件系统技术这个大目标的第一步。

本章学习目标

- 了解计算机的组成和工作原理
- 了解什么叫程序、程序设计、程序设计语言
- 了解什么叫算法、数据结构
- 了解什么叫结构化程序设计、结构化控制结构
- 了解 C 程序设计语言的特点，初步认识 C 程序设计语言，C 程序的上机过程

1.1 计算机和程序设计基础知识

在介绍程序设计技术之前，如果能对计算机的基本知识有所了解，对设计和理解程序的算法是很有帮助的。为此，先介绍与计算机程序设计密切相关的概念，并以提问和解答的形式作介绍。如果对计算机基础知识已有一定的了解，可以跳过这一节。

问题 1，什么是计算机？

现在计算机应用日益普及，如同普通机器一样，计算机在各行各业都得到广泛应用。那么，计算机与一般机器有什么不同呢？

这里所说的计算机是电子数字计算机，它是能对用离散符号表示的数据或信息可编程自动进行处理的电子设备。参见图 1-1，计算机主要由中央处理器（CPU）、主存储器（MM）、输入/输出设备（I/O）三大部分组成。CPU 又主要由控制器和运算器两部分组成，对数据进行运算并对运算过程进行控制。控制器的主要功能是自动从主存储器读取指令，予以解释、执行，并控制与输入输出设备联系等。运算器的功能是按照指令的指示完成相应的算术运算或逻辑运算，并与存储器或 I/O 设备交换数据。主存储器简称主存或内存。它是可随机存取的存储器，也称随机存储器 RAM，用来存储计算机运行时随时需要的程序和数据。计算机的输入设备能将待输入的各种形式的信息转换成适宜于计算机处理的信息，并送入计算机。

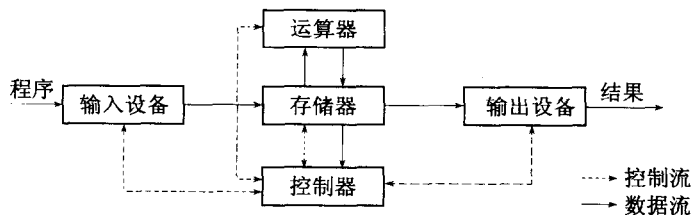


图 1-1 以存储器为中心的计算机结构框图

例如，磁盘机、键盘、鼠标器等。输出设备接受计算机处理过的信息，并能将信息转变成其他机器能识别的形式或人能理解的形式存储、打印或显示。例如，磁盘机、显示屏幕、打印机等。

问题 2，计算机是如何工作的？

CPU 是计算机的控制中心和计算机执行程序时的工作平台。内存是计算机存储程序和数据存储的场所，内存被分隔成许多个单元或字节，为区别内存不同单元的存储位置，各内存单元按顺序对应一个二进制编号，内存单元的编号称为内存单元的地址（Address）。

CPU 又包含有几个关键部件，它们是指令译码部件、算术逻辑运算部件和若干寄存器。其中指令译码部件分析当前要执行的指令，并将它转换成能让算术逻辑部件理解的电子信号；算术逻辑部件完成算术或逻辑运算；寄存器如同手工计算时所用的便笺，用来暂存计算时所需要的操作数和计算过程中所产生的临时结果。

能让计算机直接执行的程序是由计算机的机器指令组成的。一台计算机的全部机器指令构成了该计算机的机器语言。机器指令是由二进制信息的二进位组成的代码，其中一部分代码指定指令的操作功能，另一部分代码指出指令的操作对象和其他特征信息。在 CPU 中还有一个称为指令计数器的寄存器，由它指出程序中下一条将要执行的指令在内存中的地址。计算机执行一条指令大致包括 5 个阶段。

（1）取指令阶段：根据指令计数器取出程序中要执行的指令，并使指令计数器改为下一条指令的地址。译码单元将取得的指令中的操作码转换成各指令执行单元能理解的电子信号。

（2）取操作数阶段：根据指令给定的操作数地址取出操作数。

（3）执行阶段：执行单元完成译码单元传送来的命令，完成指令所规定的计算功能。如果是算术逻辑运算指令，则由算术逻辑单元完成计算。

（4）结果处理阶段：存储计算结果，并建立结果特征信息。

（5）推进阶段：回到第一阶段继续工作。

一个新的程序开始执行时，首先要将程序从外存储器调入内存，并将程序开始执行的第一条指令的地址放入指令计数器中。在取指令阶段，中央处理机根据指令计数器中的地址，从该内存地址中取出正要执行的指令，指令取出后指令计数器的内容立即被修改为下一条指令的地址。在取操作数阶段，计算机根据指令给定的寻址方式和地址寄存器的内容等有关计算操作数地址的信息，计算出操作数地址，并从该内存地址中取出操作数。在执行阶段，CPU 按指令规定的操作功能对操作数进行加工。结果处理阶段是根据目的操作数地址保存计算结果，并设置结果状态。推进阶段实现连续执行程序中的指令。上述是指令的大致执行过程，有些指令的执行过程不一定要经历 5 个阶段。每个阶段要完成的工作，不同指令也不全相同。如控制转移指令，该指令将转移地址存入指令计数器，使计算机从新的程序位置开始取指令。如此周而复始，直到停机。

由以上解释的计算机执行程序的过程可知，当计算机要执行某道程序时，只有先把当前正要执行的程序段和正要操作的数据从辅助存储器调入内存后，该程序段才能被计算机执行。如当前要执行的程序段或操作数据暂时还不在内存时，就得先把它们调入内存后才能继续执

行。对于大型程序或处理大量数据的程序来说,程序或数据可以分段调入内存,即把正要执行的那部分程序和数据先调入内存,而把暂不执行的那部分程序或暂不使用的数据暂时保留在外存,当需要它们时,再把它们调入内存。

问题 3, 什么是程序?

要使计算机能完成人们预定的工作,就必须把要完成工作的具体步骤编写成计算机能执行的一条条指令。计算机执行这个指令序列后,就能完成指定的功能,这样的指令序列就是程序。所以,程序就是供计算机执行后,能完成特定功能的指令序列。

计算机程序主要包含两方面的内容:数据结构和算法。数据结构描述数据对象及数据对象之间的关系;算法描述数据对象的处理过程。

计算机程序有以下性质:

- 目的性——程序有明确的目的,程序运行时能完成赋予它的功能。
- 分步性——程序由计算机可执行的一系列基本步骤组成。
- 有序性——程序的执行步骤是有序的,不可随意改变程序步骤的执行顺序。
- 有限性——程序所包含的指令序列是有限的。
- 操作性——有意义的程序总是对某些对象进行操作,完成程序预定的功能。

问题 4, 什么叫程序设计?

程序设计就是根据问题的需求,设计数据结构和算法,编制程序和调试程序,使计算机程序能正确完成需求所设定的任务。简单地说,程序设计是设计和编制程序的过程。

程序首先应能正确完成任务,是可靠的。同时,由于程序在使用过程中,因使用环境改变或需要修改程序功能等原因,可能会经常修改。因此,除了为程序编写详尽正确的文档外,编写容易阅读的结构化程序也是对一个好程序的要求。总体来说,好程序有可靠性、易读性、可维护性等良好特性。为达到这些目标,应采用正确的程序设计方法,以便从程序设计方法上能保证设计出具有上述良好特性的程序。

问题 5, 什么是程序设计语言?

程序设计语言是人与计算机进行信息通信的工具,是一种用来书写计算机程序的语言。计算机发展到今天,程序设计语言有几千种,它们大致可分为三类:机器语言、汇编语言和高级语言。

1. 机器语言

计算机的指令系统称为机器语言,所有的计算机都只能直接执行由其自身机器语言编写的程序。机器语言与计算机的硬件密切相关,机器语言中的计算机指令用二进制形式的代码表示,由若干位 1 和 0 组成。通常,一条计算机指令只能指示计算机完成一个最基本的操作。例如,将某个地址中的内容读入某个寄存器、某寄存器的内容加上另一寄存器的内容,将某寄存器的内容存入某地址等。

2. 汇编语言

由于计算机的机器语言很难被人理解和阅读,人们用类似英语单词缩写的符号指令代替机器语言的二进制代码指令,汇编语言就是用有助于记忆的符号表示计算机机器指令的程序

设计语言。例如，取数指令“LD GR0, X”表示从对应变量的内存中取数到寄存器 GR0。加指令“ADD GR0, GR1”将寄存器 GR1 中的内容与寄存器 GR0 中的内容相加，并把结果存于寄存器 GR0。存数指令“ST GR0, X”将寄存器 GR0 中的内容存入与变量 X 对应的内存中。

用汇编语言编写的程序要在计算机上执行，先将用汇编语言编写的源程序转换成机器语言程序，完成这个转换功能的程序称为“汇编程序”。

3. 高级语言

高级语言主要由语句构成，有一定书写规则，程序员用语句表达要计算机完成的操作。与汇编语言比较，高级语言有统一的语法，独立于具体机器，便于人们编码、阅读和理解。

用高级语言编写的源程序要在计算机上执行，也先将源程序转换成机器语言程序。把用高级语言编写的源程序转换成机器语言程序的翻译程序称为“编译器”或“编译程序”。

高级语言是一种既能方便地描述客观对象，又能借助于编译器为计算机所接受、理解和执行的人工语言。例如，用于科学计算的 FORTRAN 语言，早期非常普及的 BASIC 语言，第一个用严格的文法描述的 ALGOL60 语言，便于编写结构化程序的 PASCAL 语言，以及本书讲述的 C 语言和 C++ 语言都是高级语言。

问题 6，什么叫面向过程语言？

高级语言又可分为面向过程语言和面向问题语言两类。目前最流行最经常使用的高级语言属于面向过程型的语言。面向过程语言虽然可以独立于计算机编写程序，但用这类语言编写程序时，程序要非常详细地告诉计算机如何做，程序需要详细描述解题的过程和细节。C 语言就是一种面向过程的语言。例如，在某个职工数据文件中查找工号为 22650 的职工，面向过程语言需要详细描述查找过程。以下算法是适应面向过程语言的一个查找过程的描述。

- (1) 打开职工文件。
 - (2) 当文件未结束时重复执行以下工作。
 - ① 读取文件的当前记录。
 - ② 如果当前记录中的工号是 22650，则结束步骤 (2)。
 - (3) 关闭职工文件。
 - (4) 如果找到，则返回找到的职工信息，否则返回找不到的标志信息。
- 其中，步骤 (2) 是一个循环控制结构，控制记录一个一个地读取和比较。

问题 7，什么叫面向问题的语言？

面向问题的语言通常是在特定应用领域中使用的高级语言。人们使用面向问题语言时，不要详细给出问题的求解算法和求解过程，只需指出问题做什么，数据输入和输出形式，就能得到所需的计算结果。实际上，计算机是根据预先的规定，执行早先准备好的程序，回答问题的结果。面向问题语言又称为非过程化语言或陈述性语言，如报表语言、SQL (Structured Query Language) 语言等。SQL 语言是数据库查询语言，在数据库管理系统的支持下，用 SQL 语言提出的查询或操纵要求，就能由数据库管理系统完成。由于使用面向问题语言解题，只

要告诉计算机做什么,不必告诉计算机如何做,能方便用户的使用和提高程序的开发速度。但实现面向问题语言的系统从最一般的意义下实现问题如何做,通常实现的效率较低。另外,面向问题语言要求问题已有确定的求解方法,目前其应用范围还比较狭窄。

如果用面向问题的语言 SQL 描述问题 6 中所述的查找要求,只需要用一条能表达以下含义的简单命令:

从职工数据文件选取信息,条件是工号等于 22650。

数据文件打开、用循环控制结构描述记录一个一个地读取和比较,以及数据文件使用结束的关闭等细节都不再详述。

问题 8, 什么叫面向对象语言?

为克服面向过程语言过分强调求解过程的细节,程序不易复用等缺点,推出了面向对象程序设计方法和面向对象程序设计语言。面向对象语言引入了对象、消息、类、继承、封装、抽象、多态性等概念和机制。用面向对象语言进行程序设计时,以问题域中的对象为基础,将具有类似性质的对象抽象成类,并利用继承机制,仅对差异进行程序设计。对于大型程序,特别是已有早先用面向对象方法曾解决过的类似问题的程序时,面向对象语言能对提高程序开发的效率给予有力的支持。

问题 9, 什么叫算法?

算法就是问题的求解方法。一个算法由一系列求解步骤组成。算法的描述由经明确说明的一组简单指令和规则组成,计算机按规则执行其中的指令能在有限的步骤内解决一个问题或者完成一个函数的计算。问题 6 中给出的就是一个在职工文件中查找某个职工的算法。

正确的算法要求组成算法的规则和步骤的意义应是唯一确定的,是没有二义性的,由这些规则指定的操作是有序的,必须按算法指定的操作顺序执行,能在执行有限步骤后给出问题的结果。

求解同一个问题可能会有多种算法可供选择,选择算法的标准首先是算法的正确性和可靠性,算法简单性和易理解性。其他的标准还有算法执行速度快,所需要的存储空间少等。

描述算法可用流程图,又称框图。流程图是算法的图形描述,由于流程图往往比程序更直观清晰,容易阅读和理解,所以它不仅可以作为编写程序的依据,而且也是交流算法思想的重要工具。例如,图 1-2 描述输入 10 个整数求和计算的算法。

在逐步求精的结构化程序设计方法中,目前多数采用结构化的伪代码来描述算法。本书采用与 C 语言控制结构一致的伪代码描述算法。例如,图 1-2 的算法用伪代码可描述如下:

```
s=0;k=1;
do{
    输入 x;
    s=s+x;
    k=k+1;
} while(k<11);
```

通常算法的开发过程是由粗到细分多个阶段设计完成的。先给出粗略的计算步骤,然后对其中的粗略步骤作详细分析,添加上一些实现细节,变成较为详细的描述。可能其中还包含有实现细节不明确的部分,则还需对其作进一步的细化。直到算法所包含的计算步骤全部

都清晰明了，能用某种程序语言完整描述为止，这种方法称为逐步求精开发方法。

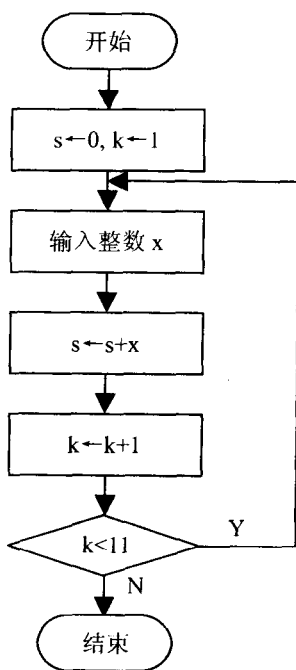


图 1-2 输入 10 个整数求和计算的算法

问题 10，什么叫数据结构？

计算机程序的处理对象是描述客观事物的数据，由于客观事物的多样性，会有不同形式的数据，如整数、实数、字符，以及所有计算机能够接收、存储和处理的各种各样的符号集合。

在计算机程序中，形式不同的数据采用数据类型来标识。变量的数据类型说明变量可能取的值的集合，以及可能对变量进行的操作的集合。例如，对于记录苹果个数的变量，它只能取大于或等于零的整数，对它能进行加减操作，但不能进行乘除操作。所以数据类型不仅定义了一个形式相同的数据集，也定义了对这组数据可进行的一个操作集。

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

数据结构与算法有密切的关系，只有明确了问题的算法，才能较好地设计数据结构；要选择好的算法，又常常依赖于合理的数据结构。数据结构是构造算法的基础。

对计算机而言，程序是一组供计算机执行的指令序列。程序告诉计算机如何对指定的数据进行操作，最终得到希望的结果。从问题求解来看，求解问题的程序是对一个抽象的求解算法的详细描述，这种描述是建立在数据的特定的表示形式和结构基础上的。不了解对数据进行如何操作，也就无法决定如何构造数据。同样，确定算法结构和算法步骤也依赖于数据结构。也就是说，程序的构成是和数据结构不可分割的。程序在描述算法的同时，也必须完整地描述数据结构。对于一些复杂的问题，常因数据的表示形式和结构的差异，问题的求解算法也会完全不同。

问题 11, 什么叫结构化程序设计?

程序设计的主要工作包括设计数据结构和算法, 编写程序以及测试程序。对于问题相对简单的情况, 一旦算法和数据结构确定后, 就可选用合适的程序设计语言编制程序。接着在计算机上测试程序, 修正程序中可能存在的错误。

随着计算机的应用日益广泛, 计算机软件的规模和复杂性不断增加, 对软件的测试和维护也越来越困难。人们为了克服软件危机, 提出结构化程序设计的方法和软件工程的概念, 要求软件开发必须遵循一套严格的工程准则, 以得到可靠、结构合理、容易维护的软件产品。

结构化程序设计主要包括程序结构的自顶向下模块化设计方法、算法的逐步求精设计方法以及用结构化控制结构描述算法和编写程序。

1. 自顶向下模块化设计方法

程序结构自顶向下模块化设计方法就是把一个大程序按功能划分成一些较小的部分, 每个较小的部分完成独立的功能, 用一个程序模块(或函数)来实现。

分解模块的原则是简单性、独立性和完整性。按模块化设计方法开发程序, 能使程序具有较高的可靠性和灵活性, 同时便于程序的测试和维护。

用模块化方法划分程序模块时, 应尽量让模块具有如下所述的多项良好性质:

- 让模块具有单一入口和单一出口。
- 模块不宜过大, 应让模块具有单一的功能。
- 模块的执行不能对环境产生副作用。
- 让模块与环境的联系仅限于明确定义的输入参数和输出参数, 模块的内部结构与调用它的程序无关。
- 尽量用模块的名字调用模块。

2. 逐步求精设计方法

在程序设计过程中, 常用的方法有抽象、枚举和归纳。

抽象包括算法抽象和数据抽象。算法抽象是指算法的寻求(或开发)采用逐步求精、逐层分解的方法。数据抽象是指在算法抽象的过程中逐步完善数据结构和引入新的数据及确定关于数据的操作。

算法的设计采用逐步求精设计方法, 即先设计出一个抽象算法, 这是一个在抽象数据上实施一系列抽象操作的算法, 由控制结构和抽象的计算步骤组成。抽象操作只指明“做什么”, 对这些抽象操作的细化就是想方设法回答它“如何做”。采用逐步求精的方法, 由粗到细, 将抽象步骤(大任务)进一步分解成若干个子任务。分而治之, 对仍不具体的子任务再进行分解。如此反复地一步一步细化, 算法越来越具体, 抽象成分越来越少, 直到可以编程为止。

枚举是指设计算法时, 先找出所有可能的情况。程序设计经验告诉人们, 如果设计的算法没有把某个特殊情况考虑在内, 当这种情况一旦发生, 程序必定会发生意想不到的错误。枚举就是把所有可能的情况, 以及每种情况的处理方法都包含在所设计的算法中。

归纳就是将所有可能情况和各种可能的处理方法作归纳, 找出其中相似或相同的部分, 将其中相似或相同的部分统一设计成基本操作, 以便按需要被程序重复使用。这样做可以减少程序代码, 便于程序的调试以及基本操作代码的复用。

3. 结构化的控制结构

在早期的程序设计活动中,因没有统一的现成方法,因人习惯而异,编制的程序结构混乱,很难阅读和修改。经计算机科学工作者多年努力,已归结出顺序、选择和循环三种结构化的控制结构,并在理论上证明了解决可计算问题的程序都可用这三种控制结构来描述。

(1) 顺序结构

把复杂工作分解成多项简单的工作,并逐一完成这些简单工作,最终完成复杂工作。顺序结构就是用于控制若干简单计算的顺序执行。

顺序结构中有一个操作步骤序列。顺序结构执行时,从操作步骤序列的第一个操作步骤开始,顺序执行序列中的每个操作,直到序列的最后一个操作执行后结束。图 1-4 表示只有操作步骤 A 和操作步骤 B 组成的一个顺序结构,该结构控制操作步骤 A 和操作步骤 B 的顺序执行。

图 1-3 的顺序结构也可用以下形式的伪代码表示:

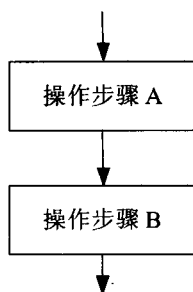


图 1-3 顺序结构示意图

```

{
    操作步骤 A;
    操作步骤 B;
}
  
```

例如,为交换变量 x 和 y 的值,可分解为顺序执行的三个操作,写成顺序结构有:

```

{
    temp=x;      /* 将 x 的值暂存于 temp */
    x=y;          /* 将 x 置成 y 的值 */
    y=temp;      /* 将 y 置成暂存于 temp 的值 */
}
  
```

(2) 条件选择结构

根据不同情况,自动选择操作步骤执行的是自动计算的需要,条件选择结构能实现这样的控制。条件选择结构由一个判断条件和两个供选择的操作步骤 A 和 B 组成,该控制结构实现如图 1-4 所示的选择控制。条件选择结构执行时,先计算条件的值,如果条件的值为真,即条件成立,则执行操作步骤 A;否则,条件的值为假,即条件不成立,则执行操作步骤 B。

注意:无论条件为何值,条件选择结构只能执行操作步骤 A 或操作步骤 B 中的某一个。条件选择结构用伪代码描述可写成以下形式: