

非计算机专业计算机系列教材

PASCAL 程序设计 基础教程

彭启琮 雷维礼 编著



科学出版社

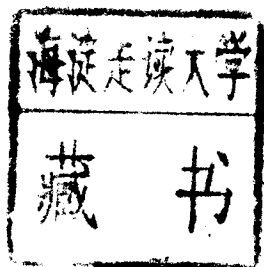
2
1

7312
P08/1

非计算机专业计算机系列教材

PASCAL 程序设计基础教程

彭启琮 雷维礼 编著



科学出版社

1993

1024403

(京) 新登字 092 号

内 容 简 介

本教程是为大学非计算机类专业编写的第一门程序设计课教材。教程选择国内外广泛使用的优秀教学语言 PASCAL 作为讲授线索,深入浅出地介绍结构化程序设计的基本思想与方法,使学生一开始就养成良好的程序设计风格的习惯。考虑到初学者以及从未接触过程序设计的其它读者的知识结构 with 能力,全书十分注意内容的深入浅出,循序渐进,并辅之以适当的例题、习题,力图使学完本课程后,其程序设计能力有较大的提高。全书共分十三章。

本书也适用于电大、夜大、职大、中专、职高以及计算机培训班作为程序设计课程教材,亦可供有关读者自学之用。

JS307/11

非计算机专业计算机系列教材

PASCAL 程序设计基础教程

彭启琮 雷维礼 编著

责任编辑 王淑兰

科学出版社出版

北京东黄城根北街 16 号

邮购 0 年

中国人民解放军第一二二工厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1993 年 8 月第 一 版 开本: 787×1092 1/16

1993 年 8 月第一次印刷 印张: 12 1/2

印数: 1—7200 字数: 287600

ISBN 7-03-003645-X/TP·282

定价: 8.50 元

前 言

本教程是为大学非计算机专业而编写的第一门计算机类教材——PASCAL 程序设计。

在我国, PASCAL 作为一种优秀的教学语言的地位, 反对的意见似乎已不多见, 然而, 要使其广泛地成为非计算机类专业的入门语言, 还得作出很大努力。我们认为, 程序设计课程的教学, 必须改变长期广泛存在的单纯语句教学模式, 以培养学生良好的程序设计风格与习惯为主要目的, 这正是贯串本教程 始终的指导思想。诚然, 对刚刚开始学习程序设计的学生, 深入讨论结构化程序设计方法及至软件工程的观念是不切实际的, 但使学生建立起基本的结构化程序设计概念是必要和可能的, 我们将其列为本教程的基本任务之一。

关于算法问题, 也是程序设计教学中长期有所争论的一个问题。由于这是非计算机专业第一门计算机类课程, 较深入讨论算法有困难, 但应该而且也有可能让学生建立起算法的最基本概念, 懂得算法的取舍会直接影响程序设计的风格和效率, 以及结果的精确度等。

本教程是在多年从事程序设计课程教学的基础上编写而成。打星号部分, 可根据学时与学生状况取舍。

本书第一章至第七章由彭启琮编写, 其余各章由雷维礼编写。与本教程相配套还有惠林同志编写的 PASCAL 程序设计实验教程, 供指导上机实践之用。

本教程由电子科技大学计算机系龚天富教授审阅, 提出了许多中肯的意见和建议, 电子科技大学非计算机类专业计算机课程指导小组和教材科同志对本教程编写成稿给予了指导帮助, 在此一并表示衷心感谢。

由于计算机的迅速发展和普及, 对我们提出了更高的要求, 由于编者水平有限, 不当和错误之处在所难免, 恳请读者批评指正。

编 者

1993 年 5 月于电子科技大学

目 录

第一章 绪论.....	1
§ 1.1 电子计算机在现代社会中的地位与作用	1
§ 1.2 计算机的发展	3
§ 1.3 计算机的硬件系统	4
§ 1.4 计算机的软件系统	6
§ 1.5 程序设计过程	8
§ 1.6 PASCAL 语言	10
第二章 PASCAL 程序的基本结构	12
§ 2.1 PASCAL 程序结构	12
§ 2.2 标准数据类型简介.....	16
§ 2.3 简单表达式与赋值语句.....	18
§ 2.4 简单的输入/输出语句	21
第三章 控制语句——选择结构	25
§ 3.1 IF 语句	25
§ 3.2 CASE 语句	30
§ 3.3 程序举例.....	34
第四章 控制语句——循环结构	38
§ 4.1 条件循环语句 REPEAT	38
§ 4.2 条件循环语句 WHILE	41
§ 4.3 计数循环语句 FOR	43
§ 4.4 循环的嵌套.....	45
§ 4.5 无条件转移语句 GOTO	46
第五章 标准数据类型及输入/输出语句.....	50
§ 5.1 数据类型的概念.....	50
§ 5.2 标准数据类型.....	51
§ 5.3 表达式与运算优先级.....	59
§ 5.4 输入语句.....	61
§ 5.5 输出语句.....	63
第六章 过程与函数	69
§ 6.1 函数.....	69

• II •

§ 6.2	过程	74
§ 6.3	子程序的嵌套	76
§ 6.4	标识符的作用域	77
* § 6.5	几个深入的问题	80
第七章	算法初步与结构化程序设计方法	85
§ 7.1	算法的基本概念	85
§ 7.2	算法的描述	86
§ 7.3	结构化程序设计方法	87
§ 7.4	程序设计举例	90
第八章	枚举类型与子界类型	99
§ 8.1	PASCAL 中数据类型的进一步说明	99
§ 8.2	枚举类型	101
§ 8.3	子界类型	105
§ 8.4	类型之间的相容关系	108
第九章	数组类型	111
§ 9.1	一维数组	111
§ 9.2	二维数组	125
* § 9.3	多维数组	131
§ 9.4	字符数组和字符串	132
第十章	集合类型	135
§ 10.1	有关集合的一些基本知识	135
§ 10.2	PASCAL 语言中的集合	136
§ 10.3	程序举例	139
第十一章	记录类型	144
§ 11.1	记录的概念	144
§ 11.2	开域语句	146
§ 11.3	程序举例	148
* § 11.4	带变体的记录	153
第十二章	文件	157
§ 12.1	文件的概念与顺序文件	157
§ 12.2	文件的建立和基本操作	158
§ 12.3	文本文件	163
§ 12.4	程序举例	167

第十三章 指针与动态数据结构	171
§ 13.1 静态变量与动态变量.....	171
§ 13.2 指针与动态存储分配.....	172
§ 13.3 链表.....	180
附录	188
附录 1 PASCAL 语法图	188
附录 2 ASCII 码表	192
参考文献.....	192

第一章 绪 论

在科学技术发展史上，象计算机这样发展迅速，对社会造成如些广泛而深远影响的事件还是不多见的。

人们广泛谈论的第二次工业革命、信息时代等等，都是以计算机技术与信息科学为标志的。电子计算机在本世纪中叶出现、发展，以及迅速地普及使用，给我们的产业、社会、乃至人们的生活方式都带来了革命性的变革。

掌握并熟练使用计算机，成了对从事各种类型工作的人们的一项基本要求。因此，很自然地，有关计算机技术的课程，也就成了大学中最重要与最基本的课程之一。不仅对计算机专业是如此，对非计算机专业也是如此；不仅对工科是如此，对理科、医科、农科、乃至文科也是如此。当然，其要求的深度与广度大不相同。

这门课是计算机类课程中的第一门。通过本课程的学习，同学们要了解计算机的基本构成与最简单的工作原理，懂得什么是程序，在遇到一个实际的问题时，如何去分析问题、设计算法、用 PASCAL 语言编制程序，然后在计算机上输入与编辑源程序、编译、调试、修改和运行，直至得到满意的结果。对程序设计而言，要了解什么是结构化程序设计方法，养成良好的编程习惯，从而建立起扎实、良好的软件设计基础。

程序设计是实践性很强的课程。这里所指的实践包括两方面的含义：一方面是在掌握最基本的程序设计方法、算法与程序设计语言的基础上编制程序的实践；另一方面则是在计算机上操作的实践。因此，重视并切实抓紧实践环节，对学好程序设计课程是至关重要的。

§ 1.1 电子计算机在现代社会中的地位与作用

事物的发展，总是以它的社会需要为前提的。计算机的发展也是如此。

要完整地叙述计算机的各种社会应用是相当困难的，因为它涉及的面实在是太广。以至于有人说，如果所有的计算机停转，现代社会就会崩溃。这或许说得过分一点，但它从一个侧面说明了计算机的重要地位。近年来，计算机病毒所造成的危害，以及人们对它的重视程度，或许也可以作为一个旁证。

一、科学计算

最初发明计算机的直接目的，是用于科学计算。因而，科学计算理所当然成为计算机的第一个应用领域，至今仍然是最重要的应用领域之一。

自然科学的理论往往用数学语言（公式、方程式等）来表述。自然界的极其复杂性带来了数学方程式及其求解的复杂性。例如，中长期天气预报中，考虑到各种因素的影响，要求解含有很多未知数的高阶方程组。如果采用人工计算，则需要很多人工作很长时间才能得到结果。然而，迟到的“预报”是没有什么实用意义的。如果采用高速计算

机来进行,就能在预定的时间内作出预报。又如大型水坝与大型建筑物的结构设计,要根据地质条件、水文条件、气象条件、承重情况、所采用的材料等来加以计算,其计算量是很大的。用人工计算时,由于工作量太大,只能忽略一些条件、假设一些条件,来使计算简化。这种比较粗略的计算的结果,既可能造成隐患,又会造成材料的浪费。采用高速计算机来计算,设计周期就短得多,其结果也要准确得多。

二、信息处理与管理

现代社会发展的结果,使得分工越来越细,而国民经济各部门之间的关系与相互依赖性越来越强,对于组织管理,以及各种信息的采集、存贮、传输、处理与管理的要求也越来越高。

大到国民经济计划的制订,需要多方的考虑,进行浩繁的计算;小到一个库房的管理,成千上万种货物,其规格、型号、生产厂家、生产时间;单价、数量、存放位置、入库与出库的动态等等,靠人力来管理是繁琐与费力的,有的统计分析根本无法进行,而用计算机来作就得得心应手。

计算机的高速处理能力,大容量信息存贮能力,加上现代化的通讯手段构成网络,为信息的存贮、传输、处理与管理提供了强有力的手段。全球飞机订票系统、银行管理系统、文献检索系统等都是很好的例子。

以文献检索来说,随着科技事业的发展,科技文献量急剧增大,有人称为“信息爆炸”。每年出版几十万种图书、几万种期刊、几百万篇论文,而且其数量还在快速地增长。查找文献成了科技工作者必须进行而又费时费力的工作。通过计算机文献检索网,我们就可以从存放在世界各地的千百万篇文献中,很方便地查阅所需要的资料。

有统计表明,用于信息处理与管理的计算机,已占到计算机总拥有量的80%,可见其使用之广泛。

三、计算机辅助设计

CAD(计算机辅助设计)已广泛地使用于航空航天飞行器、汽车、轮船、机械、电子、化工、建筑、人工合成大分子等的设计中。它可以提高程序质量、加快设计速度。对一些工作来说,甚至是决定性的、不可替代的。例如,计算机本身所需要的超大规模集成电路,要求在一块小小的硅片上,设计与制作几万,几十万,乃至几百万个晶体管及其它元件。元件之间的连线已细至微米以下。这样的芯片不用计算机来设计是不可能的。因为,人们已不可能再用几百万个器件来作模拟实验,即便作了,实验也是失真的,因为引线太粗、太长,分布参数、散热等条件已完全不同了。

除开计算机辅助设计之外,用计算机来进行的辅助测量、辅助制造、辅助教学等,更是层出不穷。

四、过程控制

对于各种生产、运行过程实施控制,是计算机的又一大应用领域。在高速运转、高层空间、高危险性作业等场合,计算机起着无可替代的作用。例如火箭发射、同步卫星转轨、卫星姿态控制,飞机盲着陆、核反应堆的控制等。现代化的钢铁、化工、水泥等

大型生产线的管理、运转、连续生产与品位控制，也都离不开计算机。

五、各种消费产品

计算机技术的应用领域，早已渗透入人们的日常生活之中。人们所熟知的全自动照相机、全自动洗衣机、录音、录象机等许多智能型设备中，都可以找到计算机的踪影。

从以上简单而很不全面的介绍中，我们已不难领略到计算机及计算机技术应用领域的广泛，以及对社会生活渗透之深入的程度。

§ 1.2 计算机的发展

人类进行计数、计算，存贮与处理信息的历史，与人类自身的历史一样悠久。手指、石头、绳结、筹码，都曾经是，有的目前还是人们用以计算与存贮信息的工具。我们的祖先发明的算盘，是中华民族对人类文明的一大贡献，至今，既作为计算工具，也作为训练与开发智力的有力工具，在全世界范围内广泛地使用着。

我们现在所说的计算机，则是指数字式电子计算机 (Electronic Digital Computer)。

第一台数字式电子计算机，是本世纪 40 年代后期在美国研制成功的 ENIAC (Electronic Numerical Integrator And Calculator)，速度为 5000 次/秒。一般将这种用电子管作成的计算机称为第一代计算机。

随着 50 年代晶体管的发明，立即应用在计算机上取代电子管，使其体积与耗电量大大减小，速度加快，可靠性提高，这就是第二代计算机。

60 年代后期到 70 年代初，集成电路 IC (Integrated Circuit) 的使用，使计算机步入第三代。

随着集成电路集成度的不断提高，计算机又发展成为使用大规模集成电路 LSI (Large scale Integrated Circuit) 与超大规模集成电路 VLSI (Very Large Scale Integrated Circuit) 的第四代。

计算机的变革与发展正在不断地、迅速地进行着。一些新发展的高新技术，如激光技术、超导技术、生物电子技术等，迅速地应用在计算机上，使其性能与可靠性不断提高，成本与价格却大幅度下降。软件技术也在不断发展，使计算机的功能越来越强，但使用却越来越方便，不再是只有少数专家才能问津的神秘的东西。这些，都为计算机的普及使用开辟了道路。而使用越普及、市场需求量越大，又反过来极大地刺激了技术与生产的发展。计算机的发展与普及正是进入了这样一种良性循环。

人们往往以运算速度与信息存贮容量的大小来衡量计算机的性能。

计算机的运算速度已从最初的每秒几千次运算，提高到目前的每秒上千亿次运算，半个世纪提高了上亿倍，其发展速度实在是惊人。我国自行设计与研制的银河系列巨型机，标志着我国跻身于世界上少有的能制造亿次巨型机的国家的行列。

在谈到计算机信息处理与存贮容量的时候，有必要先简单介绍几个有关计算机内信息表示与存贮的最基本的概念与术语。

我们知道，计算机内大量的电子器件只能处于“通”或“断”两种状态之一，往往以“0”，“1”来表示其状态。以此为基础的数字系统称为二进制系统 (Binary)，以与我

们日常使用的 0—9 十个数字为基础的十进制系统 (Decimal) 相区别。

二进制中的每一位, 即一个“0”, 或“1”, 称为一位 (Bit)。

8 位组合起来, 称为一个字节 (Byte), 它是计算机信息存贮与传输中最通行的基本单位。在存贮量很大时, 使用字节作单位显得太小, 所以常用 KB ($1\text{KB} = 2^{10}\text{Byte} = 1024\text{Byte}$), MB ($1\text{MB} = 2^{20}\text{Byte} = 1,048,576\text{Byte}$) 与 GB ($1\text{GB} = 2^{30}\text{Byte} = 1,073,741,824\text{Byte}$) 来表示。

由于计算机的性能不同, 它可以同时处理的字节数也不相同。我们将计算机同时处理的字节数称为一个机器字 (word) 或字长。例如, 同时只能处理一个字节的机器称为 8 位机, 同时处理 2 个字节的机器称为 16 位机, 同时处理 4 个字节的机器称为 32 位机, 等等。性能越高的计算机, 其机器字就越长, 标志着其处理速度越快, 结果的精度也越高。

§ 1.3 计算机的硬件系统

所谓硬件 (hardware) 是指组成计算机系统的各种电子的、机械的元件、器件与部件。图 1.1 是一个计算机系统的最简单的原理框图。我们来简要介绍一下其中的各主要部分。

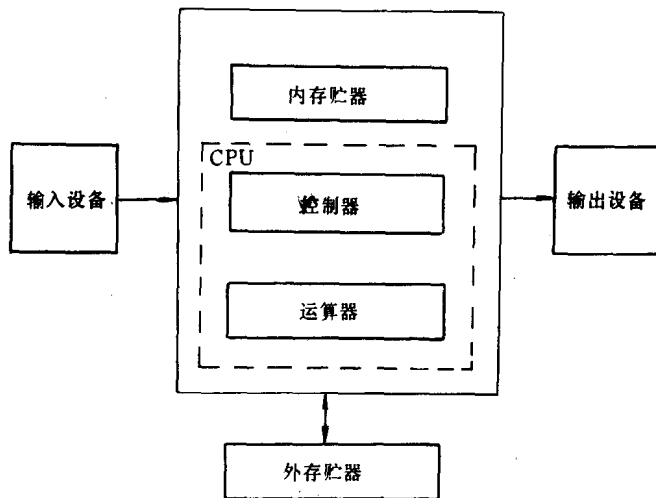


图 1.1 计算机系统原理框图

一、中央处理器 CPU (Central Processing Unit)

CPU 是计算机的心脏, 对计算机所进行的信息输入、输出、传输、处理、存贮等实施灵活、高效的管理。它由两部分组成:

1. 控制器 CU (Control Unit)

它根据事先编制好的程序, 指挥系统的各部分去完成程序中每条指令所规定的动作。

2. 运算器 ALU (Arithmetic and Logical Unit)

实施算术与逻辑运算。例如，程序规定要对两个数 A 与 B 作加法，存结果 C。当控制器得到这条指令后，就从存储器中分别读出 A 与 B 两个数，送入运算器，实施加法运算，控制器再将结果 C 存入存储器。

有关算术与逻辑运算，我们还将在今后的章节中讨论。

二、内存贮器

内存贮器是在计算机内用以存贮信息的重要部件，目前都采用超大规模集成电路形式的半导体存储器。不同类型的计算机的内存容量从几百 KB 到几十 MB 不等。

存储器又分为随机存储器 RAM 与只读存储器 ROM。

RAM (Random Access Memory) 每一个存储器单元既可以读出，又可以写入信息。写入的信息保留到下一次新的信息写入或计算机断电为止。

ROM (Read Only Memory) 存储单元内的信息是事先写好并长期保留的，即使计算机断电也不会丢失。用户可以读出存储的信息，但不能写入新的信息去取代它。

半导体存储器做成的内存，除少量单元为 ROM 外，绝大部分为 RAM，它们的存取速度很快。但是，限于成本较高，以及要保持很高的存取速度，使得内存的容量一般不是很大。RAM 内存贮的信息，一旦关机断电，就会全部丢失。

上述 CPU 及内存构成计算机的主机。

三、输入输出设备及其接口

计算机与外界的信息交换，由系统中的输入输出设备通过相应的接口来完成。

1. 输入设备 (input devices)

键盘是最常用的输入设备，操作者用以向计算机输入指令、程序、数据等。

其它的输入设备还有光电扫描仪、光笔、鼠标器、模拟/数字信号变换器 (A/D) 等。通信设备也作为输入设备向计算机输入信息。

2. 输出设备 (output devices)

最常用的输出设备有显示器、打印机、绘图仪等。计算机也可以通过通信设备向外界或其它计算机输出信息。

3. 外存储器 (Auxiliary Storage)

计算机内存的信息是不能长期保存的，随着关机断电而消失。况且，内存容量较小，不敷使用。因此，计算机系统都带有外存储器。目前使用得最广泛的外存储器是磁带，磁盘与光盘。它们的成本比半导体存储器低，容量比内存大，且记录的信息可以长期保存。外存储器的容量从几百 KB 到几十 GB 不等。与内存相比，外存储器的存取速度要慢得多。

在向计算机输入信息时，外存储器可看作输入设备；而将内存的信息转存入外存时，又可将外存储器看作输出设备。

4. 输入/输出接口 (Input/Output interface)

接口是专门的器件与电路，把输入/输出等外围设备与计算机主机连接在一起，并按照一定的规定，使它们协调地工作，完成信息的交换。

上面所讨论的计算机主机 (包括 CPU 与内存贮器)、外围设备 (输入/输出设备、外

存贮器等), 连同接口, 构成了一个计算机的硬件系统。

在讨论计算机的硬件系统时, 很有必要单独提及微型计算机。

从 70 年代开始, 随着集成电路技术的发展, 一些厂家开始把本来由很多器件组成的 CPU 作在几块, 甚至一块半导体芯片上, 这就是微处理器 (Microprocessor)。由这种微处理器作成的计算机, 称为微型计算机 (Microcomputer), 简称微机。微机一出现, 就以其结构简单、成本低廉、使用方便而风靡全世界。可以毫不夸张地说, 正是由于微机的产生与发展, 在很大程度上促进了计算机的普及使用, 从而又推动了计算机自身的发展。

在短短的 20 来年中, 微机已更新了几代, 从最初的 8 位机, 发展成为 16 位机, 32 位机、64 位机等。运算速度提高的同时, 存贮器容量也成百上千倍的提高, 而成本与价格却不断大幅度下降。在硬件发展的同时, 大量的软件工作人员为其开发了难以数计的软件, 使得微机的使用越来越方便, 功能越来越强。

§ 1.4 计算机的软件系统

只有上面讨论的硬件系统, 计算机还是什么也不会作的。迄今为止, 计算机所作的每一件事, 无论复杂还是简单, 都是使用计算机的人指令它去作的。第一步作什么, 第二步作什么, 什么时候作, 如何作等等。人们用以指挥计算机的指令 (instruction) 序列, 称为程序 (program)。而软件 (software), 则是计算机程序、算法、规则、相关的文档、以及在计算机上运行的时必须的数据的总称。

一、计算机语言

在讨论软件系统之前, 我们先来介绍计算机语言。

语言是人们用以交流思想与信息的工具。所使用的语言, 只有在交流的双方都能理解时、交流才能成功。

当人们使用计算机, 命令它作预定的信息处理与交换工作时, 必须首先使计算机“理解”并执行指令。由此而产生计算机语言的问题。

1. 机器语言 (Machine Language)

前面已讨论过, 计算机中的各种元器件只有“通”、“断”两种状态, 用“0”、“1”来表示。因此, 计算机只能接受并执行用二进制方式写成的指令, 称为机器指令。每种计算机机器指令的序列就称为机器语言, 或低级语言 (low-level language)。

下面是一段用机器语言写成的程序:

```
00111010
01100000
00000000
00111100
00000000
00110010
01100000
00000000
```

它命令计算机把地址为 01100000 的存贮单元中的数调入运算器, 将其加 1, 把结果再存回原来的存贮单元。

这段程序所规定的操作非常简单, 但这种二进制代码的形式是很难为人们所理解与记忆的。

使用机器语言时, 计算机是可以接受并执行的, 但人们却难以理解与记忆, 因为它离我们日常使用的自然语言相去太远。另外一个大问题是, 每种机器的机器语言都不相同。也就是说, 为这种机器写的程序, 在另外一种机器上是无法运行的。

2. 符号语言 (Symbolic Language)

鉴于机器语言的难于理解与记忆, 人们使用一些符号来代表相应的二进制代码, 这就是符号语言, 也称汇编语言。比如, 上面所举的将一个数取出, 加 1 后再存回去的程序, 用符号语言可以表示为

```
LD  A, (60H)
INC  A
LD  (60H), A
```

其中, LD 是指装入, A 是指运算器, 60H 是地址 01100000 的 16 进制表示, INC A 是将 A 中的数加 1。因此, 执行这段程序时, 将地址为 60H 中的数调入运算器, 加 1 之后, 将结果存回 60H。可见, 这段程序与前面的机器语言程序完全是一回事, 只是采用了另外一种符号而已。

使用符号语言, 人们是容易理解一些了, 但计算机可不能接受与执行, 因为它只能接受二进制代码。因此, 需要一个程序, 把上述符号语言写成的程序逐条“翻译”成二进制代码。这种翻译程序就是汇编程序 (Assembler)。

应该指出的是, 符号语言只是在一定程度上改善了机器语言程序的可读性, 并没有改变程序对机器的依赖性, 因而同样是低级语言。

3. 高级语言 (High-level Language)

从计算机诞生起, 人们就在寻求一种计算机语言, 用它编制的程序, 既比较地接近人的自然语言, 又可以在各类计算机中具有通用性。这就是高级语言。

在计算机诞生之后不久的 50 年代中期, 诞生了第一种高级语言 FORTRAN (FORmular TRAN slation), 以后陆续产生并得到推广使用的主要高级语言还有 BASIC, ALGOL, COBOL, PASCAL, ADA 等。

上面举出的从某个存贮单元调出一个数, 加 1 后再存回去的例子, 用 PASCAL 语言来写, 只需要一条语句

```
X := X + 1
```

其中, X 是一个变量。它是一个存贮单元的抽象。至于它代表的是哪一个存贮单元, 则由执行该程序的机器自己去安排。这样就避免对某一种机器的依赖, 从而在各种机器中具有通用性。显然, 这个语句理解起来比机器语言或符号语言都要容易得多。

用高级语言编写的程序, 称为源程序 (Source program)。由于它比较接近人的自然语言与数学语言, 因而比较容易被人们所理解。但对计算机就不然了。因此, 也需要一个“翻译”, 来将其翻译成机器可以接受的二进制代码。这种翻译是由一个程序来完成的。这种程序称为编译程序 (Compiler)。它首先检查源程序中是否存在语法错误 (Syntax error),

检查无误后,将源程序翻译成二进制代码的目标程序(Object program),以便机器执行。

简单地介绍了计算机语言之后,我们再来讨论计算机软件。从计算机系统的角度讲,软件可以分为系统软件与应用软件。

二、系统软件(System Software)

系统软件是指管理、控制计算机及其外围设备,提供计算机与用户界面的软件。最常用的系统软件有操作系统、各种语言处理程序以及编辑程序等。

1. 操作系统(Operating System)

操作系统是由一系列程序组成的,它为用户提供一个使用方便的工作环境,其功能包括CPU管理、存储器管理、外围设备管理、文件管理与作业管理等等。

在微机系统中,目前使用得最广泛的操作系统有MS-DOS, Unix, Windows等。

2. 语言处理程序

前面所讨论的汇编程序与编译程序就是语言处理程序。它首先对源程序作语法、语义方面的检查,然后将其翻译为机器代码。

还有一种语言处理程序是解释程序(interpreter)。例如,使用得很广泛的BASIC解释程序。编译程序首先将源程序翻译成目标程序,然后再去运行目标程序,而运行时就与源程序及编译程序无关了。解释程序则不同,它对源程序每扫描一个语句,就翻译并执行一条语句。它不产生目标程序,每次运行都要求源程序与解释程序参加。

3. 编辑程序(editor)

用户将源程序输入计算机,以及对源程序作修改都依赖于编辑程序,它也是人一机之间的一个重要界面。

三、应用软件(Application Software)

在系统软件为用户提供了一个方便的工作环境之后,用户就可以编制程序来解决自己的特定问题,这些软件就是应用软件。

§ 1.5 程序设计过程

图1.2所示为一个程序的设计过程,其主要步骤为:

1. 分析要解决的问题,明确已知的条件和要求

就象是求解一个数学上的应用题,首先应分析问题,弄清已知条件与求解的要求,包括结果的精度等。

2. 构造模型

根据对问题的分析,确定要使用的模型,相当于根据应用题给的条件列写方程式。

3. 确定算法

算法是指解决问题时可机械执行的一系列步骤。例如,计算从1开始的前100个自然数的和

$$S_{100} = 1 + 2 + 3 + \cdots + 100$$

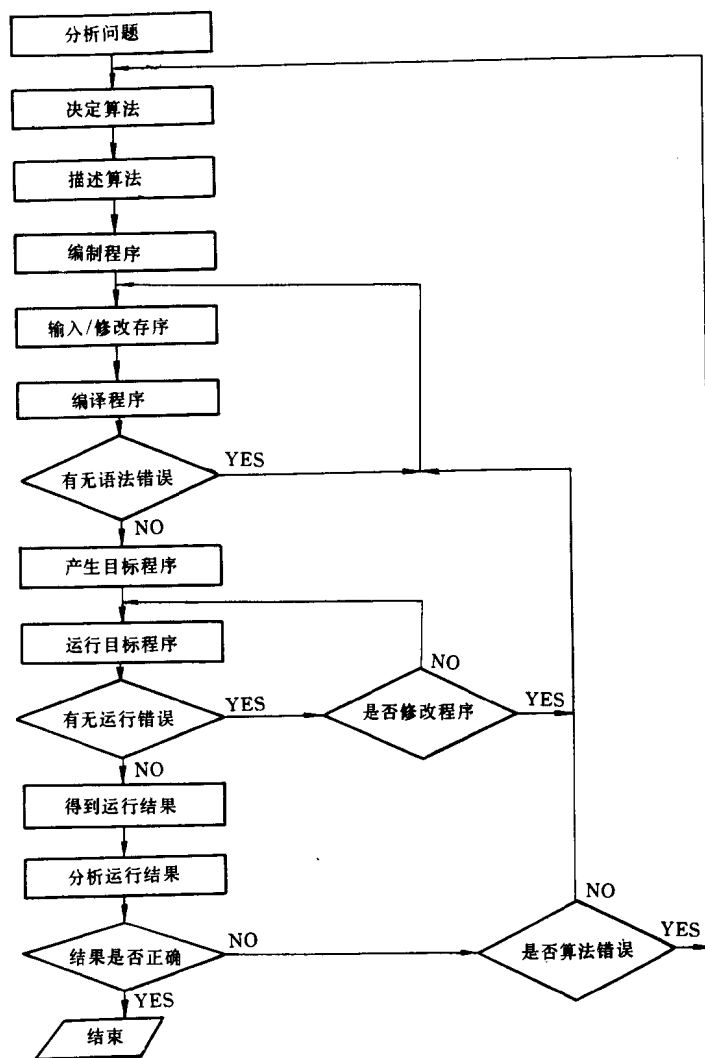


图 1.2 程序设计过程

当然可以先作 $1+2$ ，然后再加 3，再加 4...一直加到 100，也就是说，作 99 次加法，得到最后的结果。

然而，也可以用等差级数前 n 项和的公式去作

$$S_{100} = \frac{100 \times (1 + 100)}{2} = 50 \times (1 + 100)$$

结果当然是一样的,但这次只需要作一个加法与一个乘法。可见,在这个问题上,第二个算法比第一个运算量小,效率高。

一般说来,程序的效率基本上是由算法确定的。因此,根据实际问题确定适当的算法,是程序设计中的重要环节。有关算法的问题,我们将在以后的章节中陆续加以讨论。

4. 算法的描述

算法确定之后,编制程序也就是用计算机语言把算法加以精确地描述。为了保证编程时的思路清晰,往往用文字或图形方法将算法中的主要步骤加以描述。最常用的方法是流程图与伪码等。

图 1.2 是一个流程图,它使整个程序设计过程一目了然。

伪码则是按照程序的结构,用自然语言来对处理过程进行描述。它十分贴近人的思维过程,而又很容易按照其结构写出源程序,是一种很好的算法结构描述方法。

5. 确定程序设计语言,编制程序

不同的程序设计语言,有各自的特点。用户应从要解决的问题出发,选择合适的语言,根据上面所确定的流程与结构,编写出源程序。

6. 调试与运行程序

(1) 用编辑程序将源程序输入计算机

(2) 用编译程序编译源程序。如果有语法错误,则用编辑程序修改源程序,然后再编译,如此反复进行,直至没有语法错误,生成目标程序。

(3) 运行目标程序。如果有运行错误(run-time error),则根据错误的性质进行适当处置,直至没有运行错误,得到运算结果。

(4) 分析运算结果,看是否与预期结果一致。如果有问题,则往往需要修改源程序,乃至改变算法,直至得到满意的结果。

§ 1.6 PASCAL 语言

和其它高级语言相比,PASCAL 语言是较新的一种,由瑞士苏黎世联邦工业大学沃思(N. Wirth)教授于 1968 年设计。它是作为计算机的教学语言而提出的,但由于它在系统设计程序验证以及普通程序设计方面都取得了相当的成功,被公认为 70 年代以来最有影响的语言之一。

在介绍 PASCAL 语言的特点之前,我们先简单介绍一下结构化程序的设计方法。

在计算机使用的初期,机器的速度较低,存储器容量较小,这和求解的问题的复杂性形成尖锐的矛盾。因此,评判一个程序的好坏,往往把效率放在第一位。程序设计者千方百计追求程序设计的技巧,为减少一条指令、节省一个存储单元而努力。这样的程序带有强烈的程序设计者的个人色彩,往往晦涩难懂。不要说别人来读,就是自己过一段时间来读,都会感到困难。因此,对其正确性的检验难以进行,其可靠性与可维护性也难以得到保证。