

肖金声 李师贤 编著

科学出版社

PASCAL

PASCAL

PASCAL

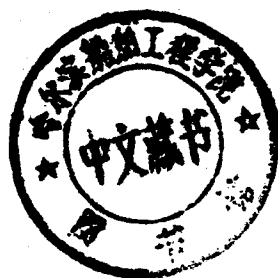
程序设计教程

346205

X38

PASCAL 程序设计教程

肖金声 李师贤 编著



科学出版社

1990

内 容 简 介

PASCAL语言是一种应用广泛的通用程序设计语言,用PASCAL语言书写程序,有助于提高程序的可读性、正确性和可靠性。

本书全面系统地介绍了 PASCAL 语言的数据类型、各种语句、程序结构以及程序设计的基本原理和程序设计方法。

本书可作为大专院校计算机有关专业教材和计算机应用软件人员水平考试的教材或参考书,也可供计算机软件及其他工程技术人员阅读参考。

JS142/08



中国科学院印刷厂印刷
新华书店北京发行所发行 各地新华书店经售

1990 年 7 月第 一 版 开本: 787×1092 1/32
1990 年 7 月第一次印刷 印张: 14
印数: 0001—12 200 字数: 319 000

ISBN 7-03-001854-0/TP·135

定价: 5.60 元

前 言

程序设计是驾驭数字计算机的方向盘,程序语言则是实施程序设计的工具。程序语言的种类数以百计,但一个人无论使用哪一种程序语言,程序设计的基本思想和方法都应该是一样的。

本书兼顾程序语言和程序设计方法这两个基本目标。

Pascal 语言创立于 70 年代初期。它是第一个体现结构程序设计思想的程序语言。创立者苏黎世联邦工业大学的 N. 沃思(Wirth)教授的原意是寻求一种有益于教学,且易于高效实现的程序语言。由于它语句简明、数据类型丰富、程序结构严谨,因而其客观效果已远远超出了创立者的初衷,而成为程序语言发展史上的一个里程碑。后来发展的并发 Pascal, MODULA-2 等并发程序语言,以及举世瞩目的 Ada,都以 Pascal 语言为基础。

时至今日,Pascal 语言已成为世界上最通用的程序语言之一,它为各种通用计算机系统所必备。十多年的经验证明,用 Pascal 语言书写程序,有助于提高程序的可读性、正确性和可靠性,有助于培养良好的程序设计风格。因此,它被公认为培养计算机专业人材的最佳入门语言。

本书讨论国家标准 GB7591-87《程序设计语言 Pascal》的 0 级规格,不涉及具体机种的扩充或删改。

作为入门教材,不要求读者具有计算机方面的必备知识。当然,具有高等数学方面的常识会有助于理解。

全书共十六章。第四、五、六、八、十一、十六章由肖金

• • •

声执笔,其余各章由李师贤执笔。除少数几章外,各章均有常识与技术一节。这一节对于掌握 Pascal 语言来说是可有可无的,但能丰富程序设计知识。

“过程”一章放得较前,是应结构程序设计的需要而为,有利于及早实施模块化。最后一章阐述程序验证的基本概念和技术。这对于年轻的学生来说,可能加大了一点学习的难度。但我们坚信,在克服了这不多的困难之后,他们的收获将是丰富、潜在,而又长久的。我们期望这一措施具有重要意义。

程序设计是一门实践性的课程。学习本书时,必须不断地配合以充分的上机实习;否则,可能会事倍功半,或者浮于表面。

本书初稿曾经赵立平副研究员、张治高级工程师审阅,特此表示感谢。

谬误之处,恳请读者和同行不吝指正。

编著者

1989 年冬

目 录

前言.....	v
第一章 计算机与程序.....	1
1.1 计算机系统	2
1.2 程序和程序语言	7
1.3 程序设计原理	10
1.4 程序语言的语法描述	11
习题与思考	13
第二章 程序初探	14
2.1 简单程序	14
2.2 标识符	19
2.3 预定义数据类型	21
2.4 程序结构	26
2.5 输入和输出	31
2.6 赋值语句	37
2.7 常识与技术: Pascal 程序格式	39
习题与思考	41
第三章 程序中的计算.....	45
3.1 表达式	45
3.2 预定义函数	53
3.3 常识与技术: 批处理和交互处理程序	59
习题与思考	60
第四章 简单类型.....	63
4.1 数据类型	63
4.2 枚举类型	65
4.3 子域类型	68

4.4 类型定义	70
4.5 类型间的关系	72
习题与思考	74
第五章 选择	77
5.1 IF 语句	77
5.2 CASE 语句	82
5.3 空语句	84
5.4 常识与技术: 实现对 CASE 语句的改动	86
习题与思考	87
第六章 重复	91
6.1 FOR 语句	91
6.2 WHILE 语句	95
6.3 REPEAT 语句	98
6.4 正确设计循环	102
6.5 语句的嵌套	107
6.6 常识与技术: 伪码、输入检查和终止标志	109
习题与思考	114
第七章 过程	118
7.1 子程序概念	118
7.2 过程说明和过程调用	120
7.3 标识符的作用域	124
7.4 值参数和变量参数的传送	130
7.5 常识与技术: 程序的测试	133
习题与思考	135
第八章 数组	139
8.1 基本概念	139
8.2 多维数组	147
8.3 紧缩数组	151
8.4 常识与技术: 岗哨和用数组实现栈	158
习题与思考	161

第九章 记录	168
9.1 记录类型与记录变量	168
9.2 访问记录的方法	171
9.3 记录的变体	179
9.4 常识与技术: 变体记录的应用	182
习题与思考	188
第十章 函数	197
10.1 函数说明和函数引用	197
10.2 非局部量与副作用	204
10.3 过程和函数作参数	208
10.4 常识与技术: 布尔函数的应用	214
习题与思考	215
第十一章 程序的研制	219
11.1 三种基本控制结构	220
11.2 关于 GOTO 语句	222
11.3 抽象和逐步求精	228
11.4 黑盒观点	240
11.5 程序的效率	242
11.6 程序中的错误	251
11.7 常识与技术: 回溯算法	252
习题与思考	261
第十二章 文卷	267
12.1 序列	267
12.2 文卷类型和文卷变量	268
12.3 基本文卷操作	271
12.4 复合操作	281
12.5 正文文卷	282
12.6 常识与技术: 随机存取文卷	289
习题与思考	292
第十三章 集合	297

13.1 集合类型和集合变量	297
13.2 集合的表示	298
13.3 集合的运算	299
13.4 集合的应用	303
13.5 常识与技术: 程序结构与数据结构的关系	307
习题与思考	311
第十四章 递归程序设计	316
14.1 递归概念	316
14.2 递归过程和递归函数的设计	320
14.3 向前说明	334
14.4 常识与技术: 从递归到迭代	341
习题与思考	352
第十五章 动态数据结构	357
15.1 指针类型和指针变量	357
15.2 预定义过程 new 和 dispose	360
15.3 链表	364
15.4 二叉树	381
15.5 常识与技术: 理解程序	387
习题与思考	389
第十六章 程序验证初阶	397
16.1 断言与验证	398
16.2 Pascal 语句的验证法则	403
16.3 验证制导程序研制	412
习题与思考	426
附录 A 字符、预定义标识符和指示字	428
附录 B Pascal 语法规则	430
参考文献	438

第一章 计算机与程序

计算机问世至今,仅仅 40 多年的时间,它已深入人类生活的各个方面,对人类科学技术的进步与发展产生着巨大而深远的影响,并已成为现代化水平的一个重要的标志。

今天,计算机在社会生活的各个领域,正担负着越来越多的任务。在医学上,计算机化断层X射线扫视和核磁共振扫视能够提供人体重要器官的精细图像,从而协助医生作出更准确的诊断和制定好的治疗方案。在商业方面,计算机被用来传送信息,进行商业事务处理,并能够建立和访问包含有大量信息的数据库。在办公室里,计算机以“现代管理者”的身份出现,承担着种类繁多的信息处理和事物管理工作,如人事档案管理,工资管理,生产管理,市场经营管理,合同管理等。在科学研究和工程技术方面,计算机的应用以数值计算为主,其应用范围遍及科学技术的各个领域:从天文学到核子物理,从汽车动力计算到宇宙飞船的轨道计算,从分子生物学到工业自动化,几乎无所不在。在社会学的研究中,计算机同样大显身手,在人口统计,儿童健康分析等工作中起着重要作用。目前,计算机还被用来帮助鉴别文物及艺术品的真伪,在语言翻译工作中也显示了它的巨大潜力。

然而,许多不甚了解计算机的人会因此产生一种错觉:认为计算机无所不能。事实上,计算机不过是扩展人类智能的一种工具而已,也就是说,它并不能代替人的思考。人们对计算机的另一种错觉是,以为计算机的使用像电视机那样,按按开关,便能获得所需要的结果。其实不然,有了计算机仅是有

了解解决问题的物质条件,而具体要做什么,怎么去做,还需要人们用一系列的命令(称作**程序**)去操纵它才行。人们把计算机本身称作**硬件**,把人们编制的操作计算机工作的程序称作**软件**。

在这一章中,我们将介绍计算机系统的概况,以及使用计算机求解问题时所涉及的程序语言,程序设计原理和程序语言的语法描述等内容。

1.1 计算机系统

计算机系统由硬件和软件两部分组成。

1.1.1 计算机硬件

第一台电子计算机 ENIAC (Electronic Numerical Integrator And Calculator) 是 1946 年制成的。自那时以来,计算机不断地进行了更新换代的发展。其结果是计算机的运算速度越来越快,存贮容量越来越大,主机的体积越来越小,而价格却在大幅度地下降。这些变化基本上可以说是由于元器件由真空管到晶体管、集成电路、大规模集成电路、超大规模集成电路的半导体技术的进步引起的。

电子计算机种类繁多,而且型号不断翻新,从微型到大型,其型号、品种数以千百计。按计算机的工作原理,可分为两大类:一类称为**模拟式电子计算机**,它利用电压的高低来模拟数量的变化,利用电的物理变化过程模拟计算。这种计算机虽然可以很快地求解微分方程组,但由于精确度不高,应用范围较窄,因此没有得到广泛的使用。另一类称为**数字式电子计算机**,它通常用一串电信号的有无表示数或符号,并按照一定的规则进行处理。这一种就是我们通常所说的计算

机,有些人称之为电脑。

数字式电子计算机(以后我们简称为计算机)能做的工作是存贮、控制、处理、输入和输出。完成相应工作的部件如图 1.1 所示,其中用箭头表示信息的传输路径。

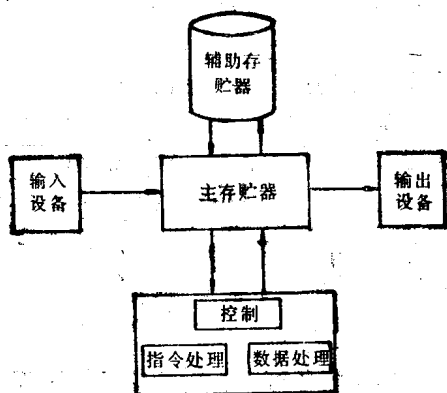


图 1.1 计算机结构

由计算机处理的信息首先经输入设备送入计算机的存贮器。存贮器内的信息由中央处理机 (CPU) 进行处理,而处理的结果仍存放在存贮器内。存贮器中的信息可用适当的输出设备展示出来。

存贮器

存贮器是一种用来存放指令和数据的存贮介质。**主存贮器**(或称内存)是能由中央处理机直接寻址的一种快速存贮介质。以微处理机为基础的系统可以只具有几百个单元(用以存放数据或指令的基本单位)的存贮器,一般计算机的存贮器可以有数万到数十万个单元不等,而大型计算机通常要有数十兆单元。存贮器还有随机存取存贮器 (RAM) 与只读存贮器 (ROM) 之分。**随机存取存贮器**可以写入和读出指令或数

据。**只读存储器**，故名思义，只能被 CPU 读出，它在制造时被永久地(不可改变)写上了指令或数据，并且在电源断电时也保持这些信息不变。

计算机的存储器可以看作由有顺序的一连串存储单元(或称字)组成。每个存储单元有它自己的编号，称为该单元的地址。

存储器中所存放的指令、数、字符、表格、图画等各种各样的信息，实际上都是 0 和 1 的串。每个 0 和 1 的串代表什么样的信息则取决于人们的解释(或者说约定)。

随机存储器单元有一种不同寻常的特性：当我们从某存储单元取出内容后，该单元的内容仍保持不变；每当把某种信息送入一个存储单元后，该单元便具备了新内容，原来的内容不复存在。

大多数计算机还配备有若干附加的**辅助存储器**，它们的存取时间比主存储器稍慢，但可以提供大得多的存储容量。应用最广泛的辅助存储器是磁盘，它由一片或多片绕中轴旋转的磁性盘片组成。磁盘的信息存取时间一般在毫秒量级内，容量从 25.6 万字节(即 8 位一组的信息单位)到 600 兆字节以上。磁泡存储器是一种固定的辅助存储设备，它可以大大减少大量数据存储的成本和复杂性。磁带是一种便于保存和携带的信息存储介质，价格较低，在许多处理系统中大量使用。

中央处理机 (CPU)

中央处理机是用来指挥整个计算机系统工作的核心装置。它主要由处理部件、存储控制部件和通道控制部件组成。

我们知道，袖珍计算器能做某一动作，它就有相应的按键；按下某一按键，它就能执行相应的一个动作。类似地，计算机的 CPU 能做某一基本操作，它就有一条相应的指令。所以，某 CPU 能执行多少种基本操作，相应地就有多少种不

同的指令。这些不同指令的集合，就称为该 CPU 的指令系统。

指令也是用 0 和 1 的串表示的，也能放在存储器中。CPU 能直接辨认和执行其指令系统中的每种指令。

CPU 的基本任务就是逐条顺序取出事先存放在主存储器中的指令，经分析后向计算机的相应部分发出控制信号，执行规定的操作。因此，中央处理机实际上是在完成由程序员编出的存放在内存的指令序列的动作。

中央处理机的处理能力，按每秒执行数十万到数千万条指令来量度。超级巨型机更具有亿次级的运算速度。

输入输出设备

输入输出设备是人们同计算机交换信息的工具。输入设备被用来向计算机送入要处理的各种信息，以及如何处理的指示；而输出设备则用来把处理的结果展示出来。

输入输出设备的种类相当多，常见的有终端键盘、显示器、行式打印机、绘图仪等。

中央处理机、存储器、输入输出设备通过总线相互连接，以传递指令、数据和维持通讯。这些装置总称为计算机硬件。

1.1.2 计算机软件

计算机刚刚问世的那些年，能够使用计算机的人屈指可数。即使是那些人，使用计算机也相当吃力，因为他们使用的是裸机——没有任何软件支持的计算机硬件。

如果说今天的计算机与当年有什么不同的话，除了前面提及的技术(主要是半导体和电磁)的进步之外，在很大程度上还由于现在具有日益完善的各类软件系统伴随着硬件服务于人类。这一变化使得千千万万非计算机专业人员得以进入使用计算机的行列。

所谓软件，实际上就是各种程序系统的泛称。由于它同硬件相辅相成、互相依赖、缺一不可，加之不象硬件那样看得见、摸得着，因而被诙谐地称为软件。如今，任何人在任何地方实际使用的“计算机”都是某个计算机系统，它由硬件和软件两大部分组成。

通常，人们把软件分为两大类，系统软件和应用软件。**系统软件**是为了有效地使用计算机，以及让计算机系统协调而充分地发挥它的潜力。系统软件一般包括操作系统、语言处理程序、数据管理系统等。

系统软件一般都比较庞大，仅就操作系统而言，随着计算机的发展，已日渐庞大和复杂。例如，1962年完成的 IBM 7090、7094 等计算机的操作系统共有 18.9 万条指令；1974年完成的 IBM 370 的操作系统 VS2 有 370 万条指令；1982年完成的 IBM 3081K 等计算机的操作系统 MVS/XA，据说实际超过 2000 万条指令。一般地说，如果一个人平均一天能开发的正确指令条数为 5 条。那么，像 MVS/XA 这样的操作系统就得需要 1000 个人工作 15 年，或者 3000 个人工作 5 年。这样庞大的软件靠少数人是无法胜任的。可是，人手一多就互相牵扯，很难协调地工作，其后果是开发时间延长，经费增加，错漏机会增多。因此，从 70 年代开始，人们提高了对可靠性的认识，正在摒弃传统的手工技艺，强调系统的、科学的程序设计方法。

应用软件解决的是计算机系统之外客观世界中的问题，如银行业务、售票系统、生产管理、过程控制之类。这类软件通常是应各种领域的需要而开发的。应用软件还可分为两类，**单项应用软件和应用程序包**。前者只适用于某一单位或某项工作，没有通用性。后者考虑到同行业的共性，在某一领域有一定的通用性。一般说来，后者更有利于做到高效、高质和低

成本。

1.2 程序和程序语言

人们做任何一件事情都有一定的“工序”。例如，若想从暖水瓶中倒出一杯水，就要遵循这样一个工序：

- (1) 准备好杯子；
- (2) 打开暖水瓶塞；
- (3) 倒出开水；
- (4) 把塞子塞回去。

如果谁不按这个工序办，就倒不成开水，甚至会出乱子。例如，没准备好杯子，开水就不知往哪里倒；倒开水后不把塞子塞回，就使整瓶开水都凉了等等。

当然，对于这个倒开水的工序人人心中有数，谁也不必用笔把它书写下来。但是，稍微复杂的事情，就有必要写好。譬如，一个大会的议程、食谱中一种菜的烹调方法、钢琴家的乐谱等。工序、议程、烹调方法、乐谱这四个词虽然各不相同，但却有一个共同点，它们都是指导人们去完成一件事的步骤或程序。

当我们试图用计算机去完成某一任务时，也需要一个能指导计算机如何完成该任务的程序。在前面提到的四种程序中，性质上与计算机程序更相近的是食谱和乐谱：

(1) 程序均有作者和执行者。乐谱的作者是作曲家，执行者是演奏家；食谱的作者可能是经验丰富的高级厨师，而执行者是一般厨师或家庭“厨师”；计算机程序的作者是程序员，执行者是计算机。程序是作者与执行者之间的通讯媒介。

(2) 执行程序均有效果。仅仅把程序写在纸上是不会有效果的，只有执行之后才会产生效果。乐谱的效果是听众听

到的乐曲声;菜谱的效果是餐桌上的佳肴;计算机程序的效果则是在打印机或显示屏上得到的计算结果等。

(3) 执行程序时应有处理的对象。做“木须肉”时,应用肉丝、鸡蛋、黄花菜、木耳等原料和油、盐、酱、葱等佐料;计算机程序处理的对象统称为**数据**。

(4) 程序的多样性。食谱中做“木须肉”和“炒鸡丁”的程序是不一样的;演奏家根据不同的乐谱奏出的乐曲各不相同;计算机程序的差异也会产生不同的结果。

上面谈的是各类程序共性的一面,在个性方面,值得一提的是程序的表达方式。食谱是用文字方式书写的,乐谱用五线谱表达,而计算机程序则用某些特定的语言(称为**计算机语言或程序语言**)来描述。

计算机语言有多种多样,既有与机器有关的,也有与机器无关的。本书仅介绍 Pascal 语言。它是 70 年代初按结构程序设计原则设计的一种语言,可用于编写系统软件、应用软件和科学计算程序。特别是,它作为讲授程序设计现代技术的一种入门语言而著称。Pascal 语言的创立者是瑞士的 N. 沃思教授。

现在来看一个求平方根的 Pascal 程序:

```
program SquareRoots (input, output);  
    {求一个数的平方根程序}  
    var x,                {说明 x 是实数}  
        y: real;          {说明 y 是实数}  
begin  
    writeln ('Hello. You are welcome.');    write ('Please enter a number to need square rooting:');  
    readln (x);           {读入一个要开平方的数}  
    y := sqrt(x);         {计算 x 的平方根,并赋给 y}  
    writeln ('The square rooting of' , x:12:4, ' is ', y:12:4)  
end.
```

• 3 •