

PASCAL

程序设计基础

肖金声 陶葆兰 编著

中山大学出版社

312

JS-1

7/2
X-2/1

PASCAL程序设计基础

肖金声 陶葆兰 编著

中山大学出版社

PASCAL程序设计基础

肖金声 陶葆兰 编著

中山大学出版社出版

广东省新华书店发行

广东韶关新华印刷厂印刷

787×1092毫米 32开本 9.75印张20千字

1985年11月第1版 1985年11月第1次印刷

印数1——15,000册

书号: 13339·11 定价: 1.85元

内 容 简 介

Pascal语言是计算机的一种高级语言。它功能强、结构合理、使用方便,既能用于科学计算,又能用于数据处理和书写系统软件。因此,Pascal语言已成为当今最通用的程序语言之一。本书系统地讲述ISO标准PASCAL的基本内容,并介绍结构程序设计和程序正确性证明的基本知识。

本书取材精炼,叙述简明,宜用作有关专业的程序语言教科书或参考书,也适合于科技人员和自学者学习之用。

前 言

程序语言Pascal是由瑞士N.Wirth教授于七十年代初期在苏黎世联邦工业大学创立的。这一语言首次体现了结构程序设计的概念，是程序语言发展史上的一个里程碑。往前看，它源于另一个里程碑ALGOL60，但功能更强、结构更合理、使用也更方便；朝后看，它成为进一步发展程序语言的良好基础。后来的并发Pascal，MODULA-2，以及当今世界瞩目的ADA，都基于Pascal语言。

Pascal语言的优点是简单易学而又表达能力强，能高效实现而又有较好的可移植性，既能用于科学计算又能用于数据处理和书写系统软件。其缺点是没有动态数组，以及处理文件的功能不全。

两相比较，优点远胜于缺点。因而Pascal语言很快就被广泛接受，成为当今最通用的程序语言之一，为各种通用计算机（即使是微型机）系统所必备。

在中国，自从Pascal语言面世，就有人注视和研究，七十年代末期就有人开始运用这语言书写应用程序，也有人在国产计算机上实现Pascal，但人数不多，范围不广。进入八十年代后，Pascal语言为更多的人所认识，逐渐普及。1983年，原教育部把Pascal定为软件专业本科生学习的第一程序语言，使该语言在我国的广泛使用成为势所必然。我们坚信，随着

这一结构化程序语言的普及，人们的程序设计水平将会提到一个新的高度。

日前，介绍Pascal语言的教材和参考书已有多种(包括翻译的和自编的)，但本书的目的和性质有所不同。主要区别是：

1. 以充当第一程序语言为基点，不要求读者具备计算机方面的知识；

2. 贯彻原教育部颁布的“程序设计”教学大纲；

3. 加进程序设计方法学的基本内容，以期一开始就着重培养学生具有良好的程序设计习惯和风格。

编写时，我们力求简洁，对材料进行了慎重裁剪，以便使读者能在较短的时间内了解Pascal语言的全貌。

本书只讨论国际标准化组织ISO公布的标准Pascal语言，不涉及任何机种对Pascal语言功能的扩充或删改。

本书的章节可供作两种不同的选择。非计算机科学专业的学生可以免去第十、十三和十四这三章。计算机科学专业的学生则应完整地学习全书。此外，本书也适用于自学。

学习本书时，上机实践是重要环节，必须密切配合。没有上机实践的学习可能浮于表面。

藉此感谢闻人德泰同志为本书提供了一些有启发的习题。

由于水平所限，谬误之处在所难免，恳请读者不吝指正。如蒙赐教，吾等幸甚。

编者 1985年5月

目 录

第一章 绪 论	1
一、计算机的构成	1
二、程序和程序语言	4
三、用计算机解题	8
四、程序语言的语法描述	12
习题与思考	13
第二章 标识符、常量、变量、表达式	17
一、标识符	17
二、整数类型	19
三、实数类型	21
四、布尔类型	23
五、字符类型	25
六、常量和常量定义	27
七、变量和变量说明	28
八、表达式	29
习题与思考	32
第三章 简单程序	34
一、赋值语句	34
二、简单输入输出语句	36
三、程序的一般格式	39

习题与思考	45
第四章 简单类型	47
一、数据类型的概念	47
二、纯量类型	48
三、子界类型	53
四、类型定义	54
五、类型间的关系	56
习题与思考	59
第五章 选择与重复	61
一、IF语句	61
二、FOR语句	66
三、WHILE语句	72
四、REPEAT语句	83
五、CASE语句	87
六、语句的嵌套	91
习题与思考	93
第六章 数 组	98
一、数组的基本概念	98
二、多维数组	107
三、紧缩数组	111
习题与思考	118
第七章 记 录	122
一、记录的基本概念	122
二、WITH语句	127
三、记录的变体	130
习题与思考	135

第八章 过程与函数(I)	138
一、过程	138
二、函数	143
三、参数的传递	146
四、名字的作用域	151
习题与思考	155
第九章 程序的研制	160
一、三种基本控制结构	161
二、关于GOTO语句	163
三、抽象和逐步求精	168
四、程序中的错误	183
习题与思考	185
第十章 集 合	186
一、集合的基本概念	186
二、集合的表示	187
三、集合的运算	188
四、集合的应用	191
习题与思考	197
第十一章 文 件	200
一、序 列	200
二、文件类型与变量	201
三、基本文件操作	203
四、复合操作	210
五、text文件	211
习题与思考	215

第十二章 过程与函数(I)	219
一、非局部量与副作用.....	219
二、递归和向前引用.....	222
三、过程或函数作参数.....	228
习题与思考.....	231
第十三章 指针和动态变量	232
一、指针类型和变量.....	232
二、标准过程new.....	235
三、指针的应用.....	237
习题与思考.....	247
第十四章 程序验证初阶	250
一、断言和验证.....	250
二、Pascal语句的验证法则.....	256
三、程序设计新方法.....	263
习题与思考.....	279
参考资料	281
附 录	282
附录A 关键字和标准标识符.....	282
附录B 运算符一览表.....	283
附录C 语 法.....	284

第一章 绪 论

本章先简述计算机的基本构成，然后引进程序和程序语言的概念，介绍使用计算机解题的一般步骤，最后简介两种描述程序语言的方式。

一、计算机的构成

当今世界上生产的计算机，从微型到大型，其型号、品种以千百计，但在结构上却惊人地相似。任一台计算机均由四个基本部分组成，如图 1.1 所示，其中的矢径表示信息的传输路径。

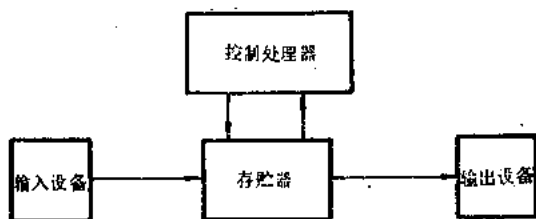


图1.1 计算机的基本组成部分

由计算机处理的信息首先必须经输入设备注入计算机的存贮器。存贮器内的信息由控制处理器加以处理，而处理的结果也存放在存贮器内。存贮器中的信息可用适当的输出设

备展示出来。

1. 存贮器

计算机的存贮器可以看作是有顺序的一连串存贮单元，每个单元有它自己的编号，称为该单元的地址。一台计算机的存贮器可以有数万到数十万个单元不等。有些计算机甚至会有数以百万计的存贮单元。

存贮单元是用来存放信息的，它能存放各式各样的信息，包括数、字符、表格等，甚至图画也能以某种形式存放在存贮器中。在一个存贮单元中存放的信息称为该单元的内容，实际上就是一串0和1。每个0、1串代表什么样的信息则取决于人们的解释(或者说约定)。这种种解释(约定)方式基本上是由计算机的设计者给出的，而对某个具体单元的内容采用哪一种解释，则是使用者的权利。

存贮单元有一种不同寻常的特性，而这种特性正是我们今后要加以利用的：每当我们从某单元取出内容后，该单元的内容仍保持不变；每当把某种信息放进一个单元之后，该单元便具备了新内容，原来的内容不复存在。

特别值得一提的是，要计算机如何处理信息的指示(由许多指令组成，每条指令要求计算机执行某一特定动作)也存放在存贮器内。计算机的控制处理部件可以从存贮器逐条取出这些指令，加以分析并执行。

基于用途和技术上的原因，存贮器又可分为随机存取的内存贮器和大容量的外存贮器两部分。内存贮器存取速度快，能配合控制处理部件的高速工作。外存贮器存取速度较慢，但容量很大，用以补充内存贮器容量的不足。内存贮器和外存贮器能成批交换信息。常见的内存贮器由磁芯或大规

模集成电路构成；外存贮器常采用磁带、磁盘或磁鼓。

2. 控制处理部件

计算机能处理各种信息，解决各类性质很不相同的问题，其处理功能的核心就是中央处理部件，常称之为 CPU (Central Processor Unit)。它负责指挥和协调计算机各部分的动作；它决定哪些操作应当执行，以及按什么顺序执行。

中央处理器中包含一个运算部件，称为运算器，它能执行各种算术和逻辑运算，并能同存贮器互相传递信息。

各类计算机内采用的 CPU 在功能上是有差异的，象型号不同的袖珍计算器功能有强有弱一样。各种 CPU 功能上的差别体现在各自的指令系统不同，这类似于不同型号的袖珍计算器上按键多寡和内容的不同。

袖珍计算器能做某一动作，它就有一个相应的按键，按下这一按键，它就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”。若要计算机执行一个操作，就要写一条相应的指令。所以，一种 CPU 能执行多少种基本操作，相应地就有多少种不同的指令。这些不同指令的汇集就称为它的指令系统。指令系统直接体现了 CPU 功能的强弱。

指令也是用 0、1 串表达的，因而也能存放在存贮器中。计算机能直接辨认和执行其指令系统中的每种指令，但对人来说这种 0、1 串是不直观的，因而也是不易使用的。

3. 输入输出设备

如果我们不能同计算机通讯，计算机的快速处理能力就会无法利用。输入输出设备就是人同计算机交换信息的工

具。输入设备被用来向计算机注入要处理的各种信息，以及应如何处理的指示；而输出设备则用来把处理的结果以可读的形式展示出来。

输入输出设备的种类相当多，常见的有纸带输入机、卡片读入机、终端键盘(或控制台打字机)、行式打印机、快速穿孔机、绘图仪等。有些计算机的终端还具有绘图功能，甚至允许用户用光笔同计算机进行有限度的通讯。此外，让计算机接受书面文件和声音的研究，目前已颇有成效，相信不久的将来就会进入实用阶段。

以上仅是计算机的粗略组成状况，当你需要使用某台计算机时，还应当对该机进行具体的了解。

二、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开水，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) 把塞子盖回去。

如果谁不按这个工序办，那么他倒开水或者不成功、或者会出乱子。例如，没准备好杯子，开水就不知往哪里倒；倒开水后不盖回塞子，就会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，谁也不必用笔把它写下来。但是，写在纸上的工序也是有的，譬如一个大会的议程、食谱中一种菜的烹调步骤、钢琴家的乐谱等。工序、议程、烹调步骤、乐谱这四个词虽然各不相同，但却有一个共同的含义，它们都是指导人们去完成某一件事

的程序，亦即倒开水的程序、开会的程序、做菜的程序和演奏钢琴的程序。

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程序员期望得到的某种结果(如果程序正确的话)，通常都在打印机或萤光屏上输出。

3) 执行程序时应有处理的对象。做木须肉时，应有肉丝、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(称为程序语言)来描述。本书仅介绍Pascal语言，它是一种公认较好的程序语言，其创立者是瑞士的Niklaus Wirth教授。

往后我们只讨论计算机程序，而不再讨论其他各类程序，故程序一词指的就是计算机程序。

现在来看看Pascal程序的一个简单例子，它是可以在计算机上真正运行的。

```
program squareroots (input, output);  
  var   x, y : real ;  
begin  
  read (x) ;  
  y := sqrt (x) ;  
  write (y)  
end.
```

C>squareroots

37.87

6.1538600E+00

这个程序向计算机发出的指示是

(1) read (x)

它要求从输入介质上读入一个值，赋给变量x。输入介质是什么，随所用的计算机而定，一般可假定是终端键盘。

(2) y := sqrt (x)

它要求把sqrt(x) 的值赋给变量y。其中sqrt是个标准函数(后面会解释)，sqrt(x)的值是x的平方根。

(3) write (y)

它要求在输出设备上输出变量y 之值。输出设备是什么也随计算机而定，一般可理解为终端显示屏。

上面这三条指示都称为语句。语句的书写是有顺序的。一般情况下，语句按书写的先后顺序执行。至于不按书写顺

序执行的情况，将在第五章详细介绍。

x 和 y 这两个变量可以获得什么样的值，由程序第二行的说明决定。那里指明 x 和 y 都是实型变量，也就是说，它们只能取实数为值。

程序头一行括号中的input指明这个程序要从输入介质上读入数据；output则指明要使用输出设备。整个程序的头一个单词program是所有程序必备的，其后的sguareroots是这个具体程序的名字。

于是，每当这个程序运行时，只要在输入介质上输入一个实数，就会在输出设备上得到该数的平方根。也就是说，这程序处理的数据是实数，解决的问题是求平方根。

用Pascal语言写出的程序并不能由计算机直接执行，因为计算机能直接执行的只是一台计算机所特有的一组命令，即其指令系统。不同的计算机具有不同的指令系统。人们用机器的指令系统写程序，就需了解计算机的许多细节，并用令人费解的0和1这两个数字的各种组合来表达自己的意图。这是一种繁琐而易错的工作方式。

与之相反，用Pascal语言写程序则不用理会计算机的细节，并用接近数学用语的方式来表达意图。因此，常把这类程序语言称为高级语言，而计算机的指令系统则称为机器语言。

然而，为了执行用Pascal这类高级语言写出的程序，就必须先把它翻译成机器语言的程序。这个翻译工作由一个专门的程序来完成。于是，这里出现了三种程序：由程序员用Pascal语言写出的程序，称为源程序；等价的机器代码（即计算机能直接遵照执行的机器语言），称为目标程序；还有