

●何英

程序设计语言

A

华南理工大学出版社

内 容 提 要

本书全面介绍了PASCAL的语法规律,并着重论述了PASCAL所涉及的软件基本概念——过程和函数、嵌套和递归、数据类型和数据结构及静态结构和动态结构。此外,书中还介绍了结构程序的设计方法,分析了一些程序设计的典型例题。全书共分十一章,各章后都有习题,书末附有习题解答。

本书可作为计算机类和非计算机类各专业的教材或参考书,也适合从事于计算机应用的工程技术人员学习参考。

程序设计语言PASCAL

何廷枢 编著

*

华南理工大学出版社出版发行

(广州 五山)

广东省新华书店经销

广东省林业勘测设计院印刷厂印装

*

开本: 787×1092 1/32 印张: 13.25 字数: 296千

1988年4月第1版 1988年4月第1次印刷

印数: 1—3000册

ISBN7—5623—0036—4/TP·3(课)

定价: 2.20元

前 言

PASCAL程序设计语言是瑞士N.Wirth教授在1971年正式提出的一种高级语言,经过十多年的发展和推广,目前已成为使用普遍的计算机高级语言。由于它功能强,数据类型丰富,模块结构清晰,语言严谨,不仅适用于数值和非数值问题的处理,而且适用于计算机软件的教学,因而越来越受到广大计算机用户和计算机教育部门的欢迎。可以说,PASCAL语言的普及程度已成为计算机教育和应用的水平的反映。

我国对PASCAL语言的研究和推广起步较晚,与BASIC语言、COBOL语言和FORTRAN语相比,PASCAL语言的普及应用深度和广度都远远不及它们。笔者深感在我国宣传和推广PASCAL语言具有很大的必要性和迫切性,根据多年科研和教学的心得和经验编成本书,为的是使我们现在拥有的计算机能够成为人们忠实而顺从的朋友,使人能够用简洁明晰、层次分明的语言去支配它。

阅读本书不必有很丰富的计算机基本知识。本书可作为读者关于计算机软件方面的入门书,如果读者打算学习数据结构、程序设计或者系统软件结构,那么建议首先学习本书。

本书力图向读者阐明PASCAL的基本语法规律,以使读者能够用PASCAL语言和计算机“谈话”。除此之外,本书还向读者深入讲述PASCAL和数据结构的最基本的但又极为重要的概念,即过程和函数的概念、嵌套和递归的概念、数据类型和数据结构的概念以及静态和动态的概念。笔

者在论述中总是力求从概念出发，着重概念的分析，使读者在学完本书之后能够在脑海中对计算机软件的基本知识形成比较清晰的图画。

本书每章附有习题，书末的附录附有习题解答。计算机程序习题的答案往往不是唯一的，因此读者应该首先自己完成作业，然后才去翻阅习题解答。

本书在编写过程中，得到华南师范大学物理系领导同志的支持和帮助。该系的计算机教研室主任郭木森教授审阅了全书并提出了极为宝贵的意见。华南师范大学微电子学研究所的高级工程师王桂海、梁安煜和谢锦正讲师都曾为笔者提出过许多有益的建议；何杏山同志完成了本书的全部习题解答，并审核了书中的程序例；华南师范大学地理系马蓉同志为本书绘制了插图，笔者谨向他们表示诚挚的谢意。

笔者热诚地盼望读者对本书的缺点和错误之处提出批评和意见。

何廷枢

华南师范大学物理系

目 录

第一章 PASCAL简介	(1)
1-1 PASCAL发展史	(1)
1-2 PASCAL的特性	(2)
1-3 PASCAL的程序格式	(6)
1-4 PASCAL的语法结构图	(7)
1-5 PASCAL的基本符号	(10)
习题一	(12)
第二章 PASCAL的程序结构	(13)
2-1 PASCAL程序的定义	(13)
2-2 常数说明	(17)
2-3 变量说明	(20)
习题二	(23)
第三章 标准数据类型、函数与表达式	(25)
3-1 整数类型	(25)
3-2 实数类型	(26)
3-3 布尔类型	(28)
3-4 字符类型	(30)
3-5 标准函数	(31)
3-6 表达式	(33)
习题三	(36)
第四章 简单的输入输出	(38)
4-1 标准输入过程read和readln	(39)

4-2 标准输出过程write和writeln	(40)
习题四	(43)
第五章 PASCAL语句	(45)
5-1 赋值语句	(45)
5-2 复合语句	(47)
5-3 重复语句	(49)
5-4 条件语句	(63)
5-5 CASE语句	(71)
5-6 标号说明和转移语句	(76)
习题五	(78)
第六章 过程和函数.....	(81)
6-1 标准过程和标准函数	(82)
6-2 简单过程和简单函数	(85)
6-3 过程说明和过程语句	(88)
6-4 函数说明和函数调用	(97)
6-5 全程变量和局部变量.....	(101)
6-6 过程和函数的嵌套和调用原则.....	(105)
6-7 递归.....	(109)
6-8 过程参数和函数参数.....	(123)
6-9 结构化程序的设计.....	(127)
习题六	(152)
第七章 自定义数据类型	(155)
7-1 数据类型的概念.....	(155)
7-2 类型说明.....	(157)
7-3 枚举类型.....	(159)

7-4 子界类型	(167)
习题七	(171)
第八章 结构性数据类型——集合和数组	(173)
8-1 集合	(173)
8-2 集合类型	(176)
8-3 数组类型	(188)
8-4 多维数组	(199)
8-5 数组在内存中的存储和信息压缩	(203)
8-6 字符串	(207)
习题八	(209)
第九章 结构性数据类型——记录类型	(213)
9-1 记录类型	(214)
9-2 数组与记录的混合使用	(218)
9-3 记录类型的变体部分	(225)
习题九	(239)
第十章 文件类型	(240)
10-1 文件类型的概念	(240)
10-2 文件类型说明和文件类型变量说明	(242)
10-3 PASCAL文件的运算操作	(244)
10-4 文本文件类型和文本文件	(256)
习题十	(263)
第十一章 动态数据结构和指针类型	(265)
11-1 静态变量和动态变量	(265)
11-2 动态变量的建立和取消	(270)

11-3	动态数据结构	(272)
11-4	链表结构	(275)
11-5	双向链表	(284)
11-6	链表的递归定义和递归运算	(289)
11-7	树结构和二叉树	(291)
习题十一		(304)
附录A	PASCAL符号集	(307)
附录B	PASCAL语法结构图	(310)
附录C	ASCII表	(315)
附录D	习题解答	(316)
附录E	IBM PC微型计算机PASCAL的使用	(378)
参考文献		(415)

第一章 PASCAL简介

1-1 PASCAL发展史

PASCAL语言是计算机程序设计的一种高级语言,是瑞士苏黎世联邦工业大学沃斯(N. Wirth)教授于1968年依据ALGOL-60语言的构思而设计发展过来的。PASCAL语言比起FORTRAN语言、ALGOL语言虽然问世得都要晚,但由于它的丰富的数据类型、简明的通用语句、清晰的程序结构、自由的书写格式,因此博得了人们的好评。PASCAL语言编译紧凑,效能很强,特别适用于教学、编写系统软件和进行科学计算,使学生较容易地学到计算机软件的许多基本概念和技巧,使编程易于编写系统程序和科学工程程序。因此,自问世以来,PASCAL语言受到特别的欢迎,成为国际上应用最广泛的主要算法之一。目前,国内外许多高等院校都以PASCAL语言进行程序设计的教学,并以PASCAL语言作为学习数据结构、程序设计等计算机专业课程的先修课程,这不是没有道理的。

PASCAL语言是为了纪念十七世纪法国数学家帕斯卡而得名的。沃斯在1968年提出PASCAL语言的构思之后,经过两年的发展,1970年PASCAL编译程序(PASCAL Compiler)便设计完成。1973年沃斯和勒生(N. Jensen)共

同发表了PASCAL的使用说明及报告 (PASCAL USER MANUAL AND REPORT), 最终完成了PASCAL的严格定义和修订。1975年以此书为设计依据的PASCAL编译程序亦相继完成, 这是标准的PASCAL语言。

在PASCAL的发展过程中, 美国加州大学圣地亚哥分校(UCSD)对PASCAL进行开发, 成为USCD PASCAL。这种PASCAL可在各种计算机中运行, 也改进了它原有的一些缺点。

目前在微型机上也开发了PASCAL语言, 以Zilog公司、Intel公司和Motorola公司生产的微处理器组成的微型机大都可以使用PASCAL。例如IBM公司的IBM PC微型机可以运行IBM PASCAL。

1-2 PASCAL的特性

PASCAL语言有以下的特点:

(一) 结构化程序语言

PASCAL语言的程序结构是模块结构, 层次分明, 清晰整齐。它比BASIC语言、FORTRAN语言具有明朗得多的结构化结构。它摒弃了BASIC语言和FORTRAN语言使用GOTO语句的结构思想, 避免了GOTO语句对程序结构的破坏, 使PASCAL成为结构化程序的设计语言。

PASCAL语言的程序结构, 见图1-2-1。其一般可以分为三大部分:

(1) 首部 又称为程序头。程序首部是程序的开头部分, 它提供程序的主要特征。首部由程序标志PROGRAM、程序名称、程序参数和程序注释四个部分组成。

(2) 说明部分 说明部分说明用户自己定义的标号、常量类型、变量、过程和函数, 以便在程序的执行部分加以引用。

(3) 执行部分 在图1-2-1的程序结构中BEGIN和END之间的部分为程序的执行部分, 它由一系列语句组成, 每一条语句执行一定的操作。语句之间用分号隔开, 整个程序用句号结束。

(二) 可由用户自定义数据类型

我们学习BASIC语言可知道, 在基本BASIC语言中允许使用的数据类型有整数(Integer)类型、实数(Real)类型、字符(Character)类型和布尔(Boolean)类型(或称逻辑类型), 除此之外, 用户不能自定义其他数据类型。但PASCAL除了系统提供的数据类型外, 用户可以自己定义需要的数据类型。例如, 若定义枚举类型week:

TYPE

week = (sunday, monday, tuesday, wednesday,
thursday, friday, saturday);

则此后就可使用week这种数据类型。

(三) 递归性

如果一个程序或过程自己调用自己, 则称此程序或过程具有递归性。如, 下面是过程a的定义:

```

PROCEDURE a(x,y,z);    { 过程a }
    :
BEGIN
    :
    a(u,v,w);    { 过程a中调用过程a 自己 }
    :
END;

```

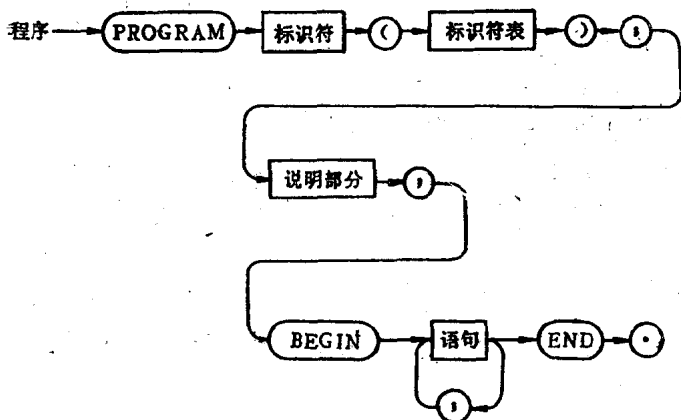


图1-2-1 PASCAL语言的程序结构

在过程a的定义中，过程a即PROCEDURE a调用它自己，因此过程a具有递归性。允许过程或函数的递归调用，是PASCAL的特点之一。利用PASCAL的这一特点，如果要解决的复杂问题具有递归性，那么编程者只需找出问题的递归关系和最简单时的情况，并按照这种递归关系和最简单情况的自然表示用递归程序实现，复杂问题的解决便转化为子程序的递归调用。例如求某正整n的阶乘 $n!$ ，可以化成

$n * (n-1)!$ ，而 $(n-1)!$ 比 $n!$ 简单些；同样 $(n-1)!$ 又可以化成 $(n-1) * (n-2)!$ ，而 $(n-2)!$ 又比 $(n-1)!$ 简单些。用同样的递归公式化简下去，一直到 $2! = 2 * 1!$ 。我们知道 $1! = 1$ ，于是 $2! = 2 * 1!$ 可以算出为2， $3! = 3 * 2!$ 可以算出为6，依此类推，递归运算的结果可得出 $n! = n * (n-1)!$ 的值。

(四) 动态的存储器分配

在存储数据时系统要给数据分配存储器。在BASIC语言中，要存储的数据（例如数组）是静态结构的，存储器的分配是静态的。这就是说，编写程序时必须由程序员决定预留多少存储单元来存储数据，可是数据量又往往事先未可预测，只有在程序执行时才知道。程序员要么过大地预留存储空间而浪费空间，要么过小地预留空间以致不足使用。显然这种静态分配请求给程序员出了难题。

动态存储器分配请求中，存储单元的分配是动态的：程序执行时，有多少数据才请求分配多少单元，既不会分配过大，也不会分配过小。

(五) 嵌套模块结构

在PASCAL中，过程或函数中又可以再定义其他的 过程或函数而成为嵌套结构。如图1-2-2所示，其中M、P、A、B都是子程序模块，M中定义P，P中定义A，A中定义B，从而成为四层的嵌套结构。

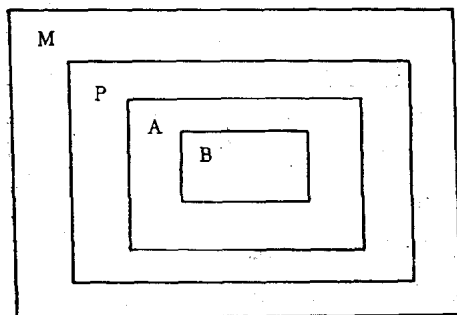


图1-2-2 模块结构的嵌套

1-3 PASCAL的程序格式

(一) 自由格式

PASCAL程序在书写时格式是自由的，就语法而言，各程序行都可以从任一系列开始来书写，不受任何限制。为了易读起见，程序行可写成阶梯式，但每一阶梯应保持整齐。

(二) 大小写不限

标准PASCAL语言对待大写或小写字体是同等的。例如程序中BEGIN和begin相同，可以互换。

(三) 分隔符

除字母A~Z和数字0~9外，其他符号如空格、+、-、*、/、=、(、)等在PASCAL程序中都可以作分隔符。例如，A=10在PASCAL中=的前后不必有空格，=本身就隔开了A和10；然而，在A DIV B中，DIV前后要用空格隔开，写成ADIVB是错的，因为字母A、D、V、B都不可

以用作分隔符。

(四) 空格

程序中在任何有空格存在的地方都可以任意加入空格。所以A DIV B可以写成A DIV B。有时为了程序好读,甚至可以在两个句子之间加入整行空格,形成空格行。

(五) 说明(注释)

为了程序易读,必须在程序中加入注释说明。PASCAL允许在空格的前后加入注释,但注释必须用{ }括起来。有些版本的PASCAL,则用(* 和 *)取代{ 和 }。

1-4 PASCAL的语法结构图

一个程序要成为正确的PASCAL程序,它至少必须符合已定义语言的“语法”。PASCAL采用语法结构图来表示它的语法结构。图1-4-1就是PASCAL程序的语法结构

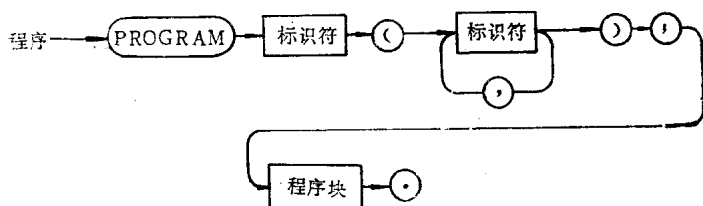
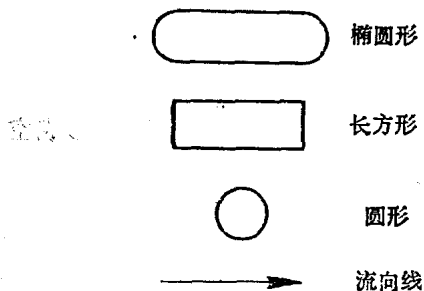


图1-4-1 PASCAL程序语法结构图

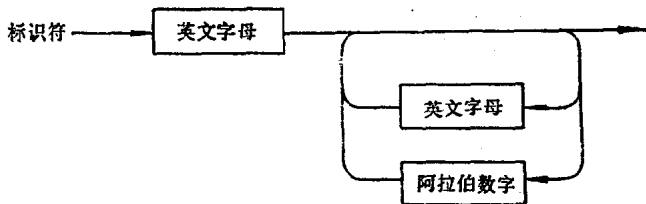
图。如果写出来的程序符合此定义,则是符合语法的程序,反之则是不符合语法的。图中我们可以看到三个图形符号和

一个箭头，它们是语法结构图所用到的标志。

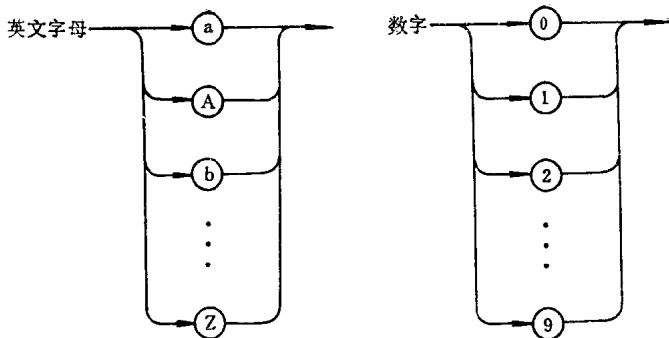


(1) 椭圆形 椭圆形里的字是保留字,如PROGRAM。保留字就是PASCAL专用字,既不允许拼错,也不允许作用其他用途。

(2) 长方形 长方形里的字是尚未定义清楚的字。我们还可以找到对此字的更详细的定义,如:长方形中的标识符在长方形中就表明是尚未定义清楚的,所以一定还可以找到对标识符的更详细的定义:



其中英文字母和阿拉伯数字也在长方形中,因此也可以再找到对它们的更详细的定义!



(3) 圆形 圆形里的字是终端字，即已经定义清楚不必再定义的字。例如a、b、c、 $\cdot$ 、 $\cdot$ 、 $\cdot$ 、 $\cdot$ 、0、1、2等都可写在圆形中。

(4) 流向线 流向线的箭头表示语法结构图中的流向。若有两个或两个以上的箭头则表示顺此方向可有多条以上流向的选择。例如，按照箭头的流向，在上述的PASCAL语法结构图中程序的第一个字是保留字PROGRAM，PROGRAM后接标识符，然后是左括号，左括号后接一个或多个标识符，而若是多个标识符，则两个标识符之间要用逗号隔开，标识符之后再接右括号，……。一个程序是否符合语法，就应看它是否能从相应的语法结构图中随着箭头的流向任意选择路径行走而最终走出语法结构图。例如，如果在程序中第一个保留字采用PROCEDURE而不是PROGRAM，或者；号换成，，那么按照语法结构图的流向，就走不出来，因而是错误的语法。