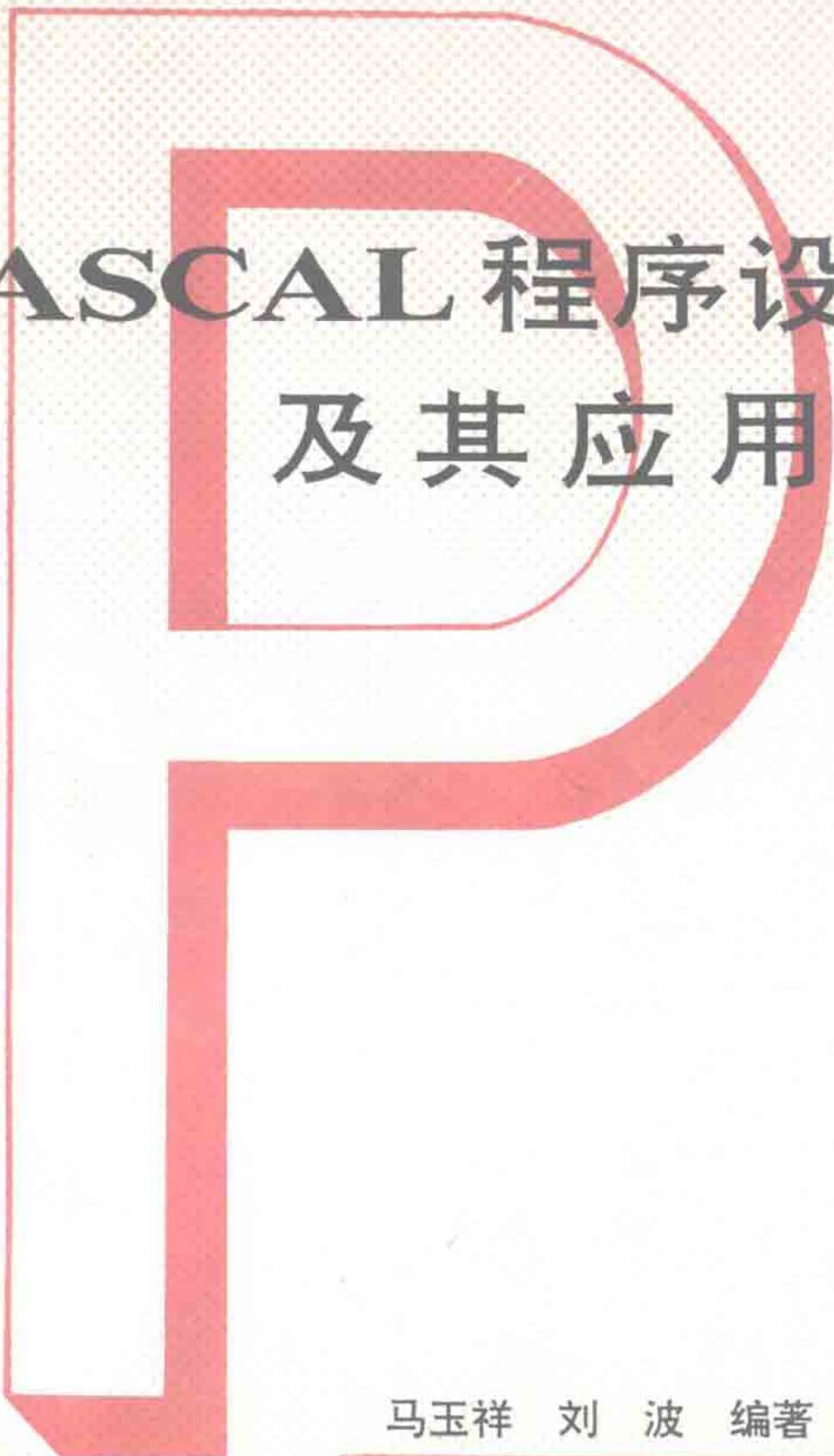




PASCAL 程序设计 及其应用

马玉祥 刘 波 编著

西安电子科技大学出版社

PASCAL 程序设计及其应用

马玉祥 刘 波 编著

西安电子科技大学出版社

1996

(陕)新登字 010 号

内 容 简 介

本书从基本概念和实际应用出发,由浅入深准确完整地讲述了标准 PASCAL 程序设计语言的语法、数据类型、通用语句;依据结构程序设计思想,循序渐进地叙述了 PASCAL 程序设计的方法和技巧;简明扼要地介绍了 TURBO PASCAL 6.0 版本的上机操作和使用方法;为了更好地掌握和应用 PASCAL 程序设计,给出各种例题程序 200 多个,书末还给出了各章习题、上机习题及部分参考答案。

统观全书通俗易懂,难易得当,众多例题方便应用。本书可作为高等院校本科、专科和职工高等学校等计算机专业的教科书;陕西省高等教育自学考试委员会指定本书为自学考试教材,同时本书可作为非计算机专业的教科书和从事计算机方面工作的科技人员的参考书。

PASCAL 程序设计及其应用

马玉祥 刘 波 编著

责任编辑 王绍菊

西安电子科技大学出版社出版发行

西安长青印刷厂印刷

新华书店经销

开本 787×1092 1/16 印张 24 2/16 字数 578 千字

1993 年 8 月第 1 版 1996 年 10 月第 4 次印刷 印数 23001~31000

ISBN 7-5606-0267-3/TP·0091

定价 18.00 元

前 言

PASCAL 程序设计语言是由瑞士沃斯 (N. Wirth) 教授于 1971 年正式发表的结构程序设计语言, 是以 17 世纪著名的法国数学家帕斯卡 (B. Pascal) 的名字命名的。由于它具有丰富完备的数据类型, 简明灵活的通用语句, 清晰的模块化结构, 并且具有程序书写简洁明了、程序编译紧凑方便和程序运行效率高等优点, 因此, PASCAL 程序设计语言已成为世界上流行广泛的程序设计语言。PASCAL 程序设计语言适用于教学、科学计算、事物处理和各种系统软件的编写等。

本书从基本概念和实际应用出发, 结合具体实例, 由浅入深准确完整地讲述了标准 PASCAL 程序设计语言; 按照结构程序设计思想, 循序渐进地叙述了 PASCAL 程序设计的方法和技巧; 简明扼要地介绍了目前微机上应用较广的 TURBO PASCAL 6.0 版本的上机操作和使用方法, 该部分内容将使读者能很快地熟悉 TURBO PASCAL 集成环境, 能迅速掌握其主要功能和方法, 勿须购买昂贵的 TURBO PASCAL 6.0 用户手册; 为了更好地掌握和应用 PASCAL 程序设计, 给出了各种例题程序 200 多个; 为了能较快地熟悉和编写 PASCAL 程序, 书末还给出了部分习题和上机习题的参考答案。

在此, 特别感谢西安电子科技大学计算机系微机应用中心的同志们, 在本书的编写过程中给了我们极大的支持和帮助。

本书由马玉祥同志主编。第一章至第十章由马玉祥同志编写, 第十一章和第十二章部分由刘波同志编写。编者虽然经过了 PASCAL 程序设计十多年的教学实践, 但书中错误和不妥之处在所难免, 敬请读者批评指正。

编 者

1992.12.26

目 录

第一章 程序设计基本概念	1
§ 1.1 计算机语言	1
§ 1.2 PASCAL 语言简介	1
§ 1.3 程序设计基础	2
第二章 PASCAL 语言的基本概念	6
§ 2.1 PASCAL 语言的数据类型	6
§ 2.2 标准数据类型	7
§ 2.3 标准数据类型的特点	12
§ 2.4 标准函数	13
§ 2.5 PASCAL 程序结构	14
§ 2.6 简单的程序设计	15
第三章 PASCAL 语言的基本语句	32
§ 3.1 简单语句	32
§ 3.2 构造语句(1)	33
§ 3.3 构造语句(2)	45
§ 3.4 数组	52
§ 3.5 应用举例	56
第四章 过程和函数	66
§ 4.1 过程	66
§ 4.2 参数	71
§ 4.3 函数	74
§ 4.4 过程的作用	79
§ 4.5 应用举例	99
第五章 用户定义的简单类型	109
§ 5.1 类型定义	109
§ 5.2 枚举类型	110
§ 5.3 子界类型	115
第六章 集合类型与类型间的关系	122
§ 6.1 集合类型	122
§ 6.2 应用举例	127
§ 6.3 类型间的关系	131
第七章 数组和记录	136
§ 7.1 数组类型	136
§ 7.2 记录类型	157
§ 7.3 记录的变体	167

第八章 文件	175
§ 8.1 顺序文件	175
§ 8.2 正文文件	182
§ 8.3 TURBO PASCAL 系统中的文件	189
§ 8.4 输入和输出	200
第九章 指针	212
§ 9.1 指针类型	212
§ 9.2 链表	215
§ 9.3 链表的插入和删除	220
§ 9.4 树	225
§ 9.5 应用举例	230
第十章 程序设计	249
§ 10.1 一般的程序设计方法.....	249
§ 10.2 自顶向下程序设计方法.....	252
§ 10.3 结构程序设计.....	259
第十一章 TURBO PASCAL 上机操作指南	271
§ 11.1 TURBO PASCAL 简介	271
§ 11.2 TURBO PASCAL 上机操作	272
§ 11.3 TURBO PASCAL 使用举例	278
§ 11.4 TURBO PASCAL 与 IBM PASCAL 的区别	281
第十二章 应用程序举例	289
§ 12.1 应用程序说明.....	289
§ 12.2 应用程序举例.....	289
习 题	320
部分习题参考答案	335
上机习题	356
部分上机习题参考答案	358
附 录	365
附录 A PASCAL 语法图	365
附录 B 保留关键字与标识符	369
附录 C 非字特定符号	370
附录 D TURBO PASCAL 编辑命令与例程(库函数)索引	371
附录 E IBM PASCAL 上机操作简介	376
参考文献.....	380

第一章 程序设计基本概念

一般的计算机系统由硬件和软件两部分组成。硬件包括：存储器、运算器、控制器和输入输出设备等。软件包括：操作系统、编译程序、系统程序和应用程序等。上述这些程序都是运用程序设计的方法和技巧，使用计算机语言编制的。本章将简要介绍 PASCAL 语言和程序设计基础的粗浅知识。

§ 1.1 计算机语言

计算机语言是人机信息交换的重要工具。计算机语言分为低级语言和高级语言，低级语言有机器语言、符号语言等。高级语言目前多达上千种，常用的也有 20 多种，要根据不同的用途选取不同的语言。常用的程序设计语言有：汇编语言、BASIC、FORTRAN、PL/I、COBOL、PASCAL、C、Ada 语言等。目前 PASCAL 语言是国内外较为流行的语言，尤其在大学里更为普遍。

§ 1.2 PASCAL 语言简介

PASCAL 语言最先是由瑞士沃斯 (N. Wirth) 教授在 1971 年公布的，为纪念法国的数学家 B. Pascal (1623~1662) 而取名。1975 年国际标准化组织承认并组织了“联合 PASCAL 委员会”。1986 年我国也制定了相应的 PASCAL 语言国家标准：GB ISO 7185—1982。经国家标准局批准公布。

1.2.1 对 PASCAL 语言的评价

PASCAL 语言的优点：

- (1) 层次结构清楚，容易阅读和理解；
- (2) 模块化结构，可分块设计；
- (3) 数据类型丰富完备，控制语句灵活；
- (4) 具有结构递归性质，可采用自顶向下的设计技术；
- (5) 程序结构清晰，语义简单，易于验证；
- (6) 编译紧凑简洁，易于修改，运行效率高。

PASCAL 语言的缺点：

- (1) 文件处理功能不完善；
- (2) 没有动态数组，只有在编译时分配的静态数组；
- (3) 循环结构没有非正常出口，只能引入布尔类型变量，用 IF 或 GOTO 语句转出。

1.2.2 PASCAL 语言的应用

PASCAL 语言是结构程序设计语言和系统程序设计语言。国内外应用日益广泛，在高等

院校的教学领域中应用也极其普遍。

目前,许多计算机系统都配备了 PASCAL 语言的编译系统,这对工程计算,事务管理及系统程序设计和系统软件设计开发等,起到了推动作用。

PASCAL 语言是应用普遍,发展迅速的程序设计语言之一,对于任何一个立志于计算机事业的人来说,尽快掌握 PASCAL 程序设计语言是迫切需要的事情。

§ 1.3 程序设计基础

1.3.1 程序与程序设计

一个程序是一个指令的序列,是对所要解决问题的各个对象和处理规则的描述,是由程序员编制设计的静态过程。而输入计算机实现指令的过程是动态的过程。

程序是实现人与计算机通讯的一种手段,编制程序要符合一种规约,这就是程序设计语言,PASCAL 语言就是其一。程序设计语言是用于书写计算机程序的语言。

程序设计就是设计、编制和调试程序的过程。

1.3.2 程序结构

程序设计语言发展的过程,就是把指令序列加上结构的过程。

程序员可以用常规的代数形式写表达式,其数据需要加以构造,要选用理想的指令结构将其输入给计算机,使之得到理想的结果。

程序有四种基本结构:一条接一条执行的顺序结构;根据某种条件的成立与否来决定是执行某些指令或跳过某些指令的判定结构;要多次执行某些指令的重复、循环结构;用一条指令替代一组指令的过程结构。因此指令自身具有顺序、判定、重复或循环以及过程等结构。

PASCAL 语言,使用了这些构造技术,使得它成为一种功能较强的程序设计语言。

1.3.3 PASCAL 程序一般介绍

首先请阅读下例,然后分析其程序结构。

例 1 在输入的数中选取最大数输出。

程序如下:

```
PROGRAM max(input,output);  
  VAR x,y: integer;  
  BEGIN (* max number program *)  
    write('x='); readln(x);  
    write('y='); read(y);  
    IF x<y THEN x:=y;  
    writeln('MAX=',x)  
  END.
```

运行结果:

X=345

```
Y=5678
MAX=5678
```

这个简单的程序，由程序首部、说明部分和语句部分等三大部分组成，如图 1-1 所示。

程序首部：

```
PROGRAM max(input,output);
```

说明部分：

```
VAR x, y : integer;
```

语句部分：

```
BEGIN
```

```
:
```

```
END.
```

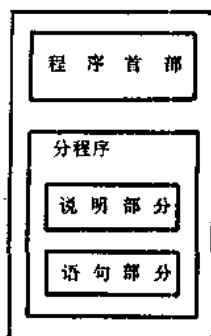


图 1-1 PASCAL 程序组成

说明部分和语句部分组成程序体，称为分程序。这个简单程序的语句部分具有顺序和判定结构。

例 2 输入一些整型数，对它们计数并找出最大数和最小数的程序。

程序如下：

```
PROGRAM minimax(input,output);
  VAR b : boolean;
      c,n,x,y : integer;
BEGIN
  b := true;      c := 0;
  x := maxint;    y := -maxint;
  writeln('Input any integer numbers : ');
  WHILE b ( * b=true * ) DO
    BEGIN
      read(n);
      IF n=0 THEN b := false
      ELSE
        BEGIN
          c := c+1;
          IF n<x THEN x := n;
          IF n>y THEN y := n;
        END
      END; ( * while * )
  writeln('Output results : ');
  writeln('C : 5, ' Numbers ');
  writeln('x : 5, ' Smallest ');
  writeln('y : 5, ' Largest ');
END. {mini_max}
```

运行结果：

Input any integer numbers;

123 234 -345 456 789 -987 0

Output results :

6 Numbers

-987 Smallest

789 largest

程序中, 变量 b 是布尔类型的, 只能有两个值: 真 (true) 或假 (false)。WHILE (当) 语句之后的 BEGIN 和 END 之间的语句, 只要 b 为真就重复执行。这是重复结构。

变量 c 是整型的, 起计数的作用, 在 n 不为零时, 表示输入数据个数的和; 当 $n=0$ 时, 作 WHILE 语句的终止条件。

变量 x 放当前最小数, y 放当前最大数。

1.3.4 PASCAL 语言的语法描述方法

PASCAL 语言的语法描述方法, 常用的是语法图和 BNF 范式这两种方法。

1. 语法图

PASCAL 语言的全部语法图, 详见附录 A。

语法图中的符号说明, 请看图 1-2。

(a) 表示不必定义的语法实体, 其内的字符是 PASCAL 语言规定的必须出现的实际字符。

(b) 表示由其他语法定义的语法实体。



图 1-2 语法图中符号说明

例: 程序语法图如图 1-3 所示。

2. BNF 范式

所谓 BNF 范式, 是使用元语言抽象化的描述方法。BNF 为 Backus - Naur Form 的缩写。1956 年 Chomsky 提出抽象文法规范, 1960 年 Backus 和 Naur 编辑 ALGOL-60 报告时使用了这种规范, 故称为 BNF 范式。它属于 Chomsky 的 I 型文法。BNF 范式将变元化简成终结符号。

例: 程序的语法用 BNF 范式描述为

$\langle \text{程序} \rangle ::= \langle \text{程序首部} \rangle \langle \text{分程序} \rangle .$

$\langle \text{程序首部} \rangle ::= \text{PROGRAM} \langle \text{标识符} \rangle (\langle \text{标识符} \rangle [, \langle \text{标识符} \rangle] \dots);$

$\langle \text{分程序} \rangle ::= \langle \text{说明部分} \rangle \langle \text{语句部分} \rangle$

$\langle \text{语句部分} \rangle ::= \text{BEGIN} \langle \text{语句} \rangle [; \langle \text{语句} \rangle] \dots \text{END}$

说明:

$\langle \rangle$ —— 语法变量;

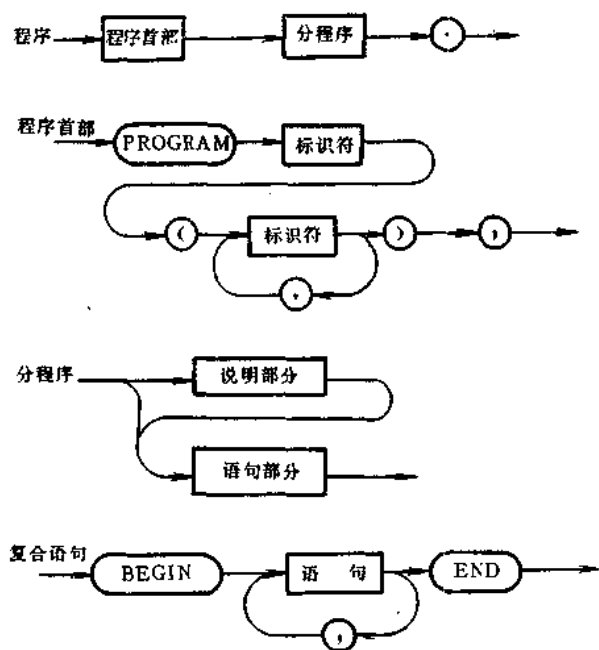


图 1-3 程序语法图

::=——定义作；

|——或；

[] ...——选择/重复。

1.3.5 PASCAL 程序的表示

任何一个 PASCAL 程序都由程序首部和分程序构成，并用一个句号（.）结束。因此 PASCAL 程序结构是层次分明的模块结构。

从前两例中，我们看到 PASCAL 程序是由保留关键字（如 PROGRAM, VAR）、标识符（如 integer, max）和一些非字特定的符号（如 :=, ;）所组成。详见附录 B 和 C。

敬请读者注意，本书的 PASCAL 程序中，保留关键字一律用大写字母，标识符大多数用小写字母，以便读程序时醒目。

第二章 PASCAL 语言的基本概念

本章主要介绍 PASCAL 语言的数据类型、标识符、变量、表达式、赋值语句、输入和输出语句等。这些都是 PASCAL 语言的基本概念。

§ 2.1 PASCAL 语言的数据类型

PASCAL 程序中每一个变量都必须在程序的变量说明部分说明其类型。类型确定了以后，变量可能取值的范围也就确定了，例如，整数类型的变量，其取值范围是该机器的整数子集；同时类型也决定了变量可参加哪些运算。

PASCAL 语言有丰富的数据类型，按其特点分成三大类，即简单类型、构造类型和指针类型。如图 2-1 所示。

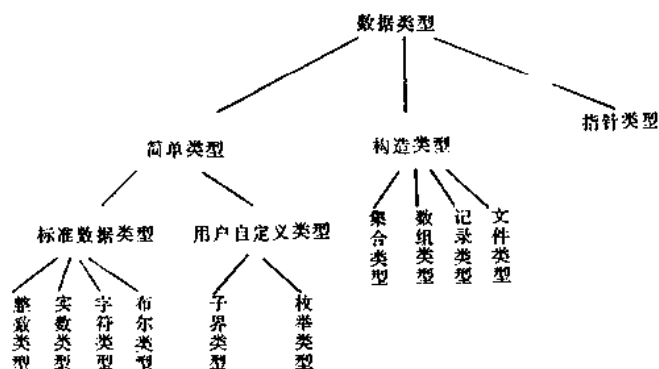


图 2-1 PASCAL 语言的数据类型

2.1.1 简单类型

简单类型包括常用的四种标准数据类型：整数类型（简称整型）、实数类型（简称实型）、字符类型和布尔类型。还包括两种用户自定义的数据类型：枚举类型和子界类型。

2.1.2 构造类型

构造类型是复杂的数据类型，是由其他类型按照一定的规则构造而成的。它包括集合类型（简称集合）、数组类型（简称数组）、记录类型（简称记录）和文件类型（简称文件）。

2.1.3 指针类型

指针类型（简称指针）为 PASCAL 语言提供处理动态变量的能力。其数据是动态数据。本章将介绍四种标准数据类型，其余将在后续章节介绍。

§ 2.2 标准数据类型

PASCAL 语言提供了四种标准数据类型,这四种类型的数据都是有序的,但实型没有整数序号。这里主要讨论四种标准数据类型的概念,它们的运算和标准函数。

在 PASCAL 语言中,这四种标准类型是:

integer (整数类型) real (实数类型)

boolean (布尔类型) char (字符类型)

由它们指明的变量说明语法如图 2-2 所示。

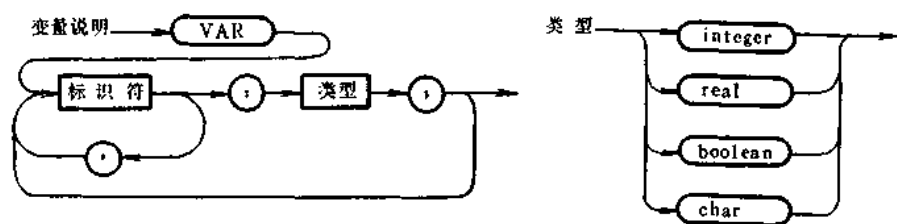


图 2-2 变量说明语法

例如: VAR i, j : integer;
x, y : real;
b1, b2 : boolean;
ch1, ch2 : char;

2.2.1 整数类型

整数包括正整数、负整数和整数零。其格式如图 2-3 所示。

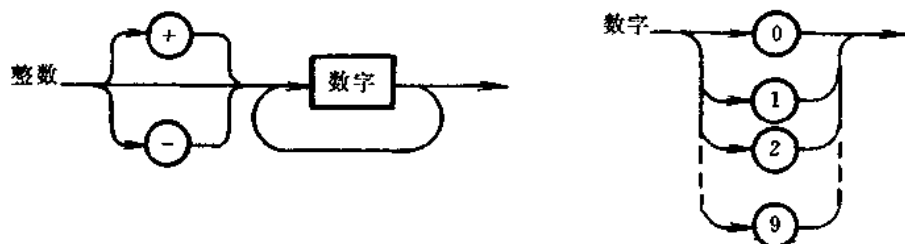


图 2-3 整数格式

一般情况下,数字用 0 到 9 的十进制表示。正整数和整数零可不写符号位。任何一台计算机只能表示整数的一个有限子集,把所能表示的最大整型数定义为标准常量 maxint。因此, PASCAL 语言中整型的值是整数中这样的子集:

$-maxint, -maxint+1, \dots, -1, 0, 1, \dots, maxint-1, maxint$ 在不同的 PASCAL 版本中, maxint 的值是不同的,通常在 32767 到 281474976710655 之间,即 $2^{15}-1$ 到 $2^{28}-1$ 之间。

整型变量的说明如下例:

term, count, num, i, j : integer;

与整型有关的运算符包括算术运算符和关系运算符。

算术运算符：+、-、*、DIV、MOD 分别表示加、减、乘、整除和取余。这五个运算符要求运算对象都是整型的，其运算结果的类型也是整型的。其中运算符 DIV 是两个整型值相除后截去小数部分后的整型结果。运算符 MOD 是取余运算符，它要求第二个运算量必须是正的。取余的意义是

$$a \text{ MOD } b = a - (a \text{ DIV } b) * b$$

例如：2 MOD 5 = 2 - (2 DIV 5) * 5

$$2 \text{ MOD } 5 = 2$$

同理 -2 MOD 5 = -2

算术运算举例如下：

$$5 + 3 \quad \text{结果为 } 8$$

$$5 - 3 \quad \text{结果为 } 2$$

$$5 * 3 \quad \text{结果为 } 15$$

$$5 \text{ DIV } 3 \quad \text{结果为 } 1$$

$$-5 \text{ DIV } 3 \quad \text{结果为 } -1$$

$$5 \text{ MOD } 3 \quad \text{结果为 } 2$$

$$-3 \text{ MOD } 5 \quad \text{结果为 } -3$$

关系运算符：=、<>、<、>、<=、>= 分别表示等于、不等于、小于、大于、小于等于和大于等于，共六种关系运算。关系运算的结果只有两种可能：成立或不成立，即真 (true) 或假 (false)。因此，关系运算的结果是布尔类型的值。

关系运算举例如下：

$$5 > 3 \quad \text{结果为 true}$$

$$5 \geq 3 \quad \text{结果为 true}$$

$$3 < > 5 \quad \text{结果为 true}$$

$$3 = 5 \quad \text{结果为 false}$$

与整数类型有关的标准函数有算术函数、转移函数和顺序函数。这些函数的值必须是整型的。详见后面的标准函数表 2-1。这里只介绍两个顺序函数，其他的留在 2.2.2 中介绍。

设 i 是整型变量，顺序函数是：

pred(i) 函数值是 i 的前趋，即 i-1；

succ(i) 函数值是 i 的后继，即 i+1。

称它们为前趋函数和后继函数。但要注意当 i 为 maxint 值时，pred(-maxint) 和 succ(maxint) 都是非法的。

2.2.2 实数类型

实数包括正实数、负实数和实数零。PASCAL 语言中，表示实数有两种方法，即十进制表示法和科学表示法。其格式如图 2-4 所示。

图中小数点前面为整数部分，后面为小数部分，正实数和实数零可不写符号位。PASCAL 语言规定，实型数的小数点前后都必须有数字。这是我们常用的十进制表示法。科学

表示法就是指数表示法, 字母 E 表示以 10 为底的指数, 与计算机的浮点表示是一致的。E 前面部分无论有无小数部分都被称为尾数, E 后面部分是阶码, 只允许是整数, 其正号可以不写。

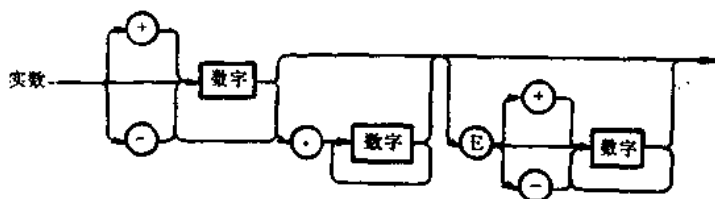


图 2-4 实数格式

计算机只能表示实数的一个有限子集。被表示的每一个实数都具有固定的有效数字位数, 有效数字的位数取决于用多少个二进制位表示一个实型数。整型数的算术运算结果是精确的, 而实型数的算术运算结果是近似的。

实型的变量说明如下例:

```
avg, x, y, s: real;
```

与实型有关的算术运算符为: $+$ 、 $-$ 、 $*$ 、 $/$ 。其中“/”表示除, 与 DIV 是完全不同的概念。即使两个运算对象都是整型的, 其结果类型还是实型的。例如:

```
10 DIV 5 = 2    10/5 = 2.0
```

```
10 DIV 4 = 2    10/4 = 2.5
```

“/”能自动将整型数转换为实型数参加运算。

对这四个算术运算符, 只要有一个运算对象是实型的, 则结果也是实型的。

六种关系运算符均适用于实型。它们的运算对象还可以一个是实型, 另一个是整型。关系运算后产生布尔类型结果值 (true 或 false)。但要注意, 对于实型数的相等比较要慎用。因为两个实数值非常接近时, 计算机由于存在舍入误差, 就不能区别它们了。应比较它们的差值是否在某一误差范围中 (见 2.2.4)。

与实型有关的标准函数有算术函数和转换函数。

算术函数中, 变元 (或称自变量) x 的类型可以是实型或整型的。对于函数 abs 和 sqr, 其结果类型与变元 x 的类型相同。其余所有的算术函数的结果类型总是实型的, 如:

abs(x)	x 的绝对值
sqr(x)	x 的平方
sin(x)	x 的正弦, x 用弧度表示
cos(x)	x 的余弦, x 用弧度表示
arctan(x)	x 的反正切的主值, x 用弧度表示
exp(x)	指数函数 e^x
ln(x)	x 的自然对数
sqrt(x)	x 的平方根

PASCAL 语言没有引入乘幂运算符, 但可利用标准函数来计算 x^y 的值。由于 $x^y = e^{y \ln x}$, 因此若要计算 x^y , 则可写成

```
exp(y * ln(x))
```

转换函数中, 变元 x 的类型是实型的, 函数的结果类型是整型的。

$\text{trunc}(x)$ 截去实型值 x 的小数部分, 转换为整型, 故常称为截尾函数。如:

$$\text{trunc}(4.5) = 4 \quad \text{trunc}(-4.5) = -4$$

$\text{round}(x)$ 对实型值 x 作四舍五入, 转换为整型。如:

$$\text{round}(4.5) = 5 \quad \text{round}(-4.5) = -5$$

2.2.3 字符类型

字符类型数据包括全部可以打印输出的字符。字符类型数据的格式如图 2-5 所示。符号 “'” 叫撇, 又叫单引号, 字符格式中必须成对出现。

如 “'” 是字符 “'” (撇) 的合法表示, 这是 PASCAL 语言规定的。



图 2-5 字符类型数据的格式

字符类型变量说明如下:

$\text{ch1}, \text{ch2} : \text{char};$

一个字符类型变量的值是一个字符, 该字符必须是计算机所用的字符集内的可打印输出的字符。

目前计算机中常用的字符集有两种, 就是 ASCII (美国标准信息交换码) 字符集和 EBCDIC (扩充二-十进制交换码) 字符集。前一种字符集由 128 个字符组成, 后一种字符集由 256 个字符组成。

ASCII 字符集较为常用, 是 ISO (国际标准化组织) 标准字符集。在其字符集内, 每一个字符对应一个次序值 (又称码值), 不同的字符, 其码值也不同。在 128 个字符中, 00 码至 31 码和 127 码是不能打印出来的, 其余的 95 个字符 (次序值由 32 到 126) 才能作为 PASCAL 语言的字符类型数据。例如:

'0' 表示数字字符 0, 其码值为 48;

',' 表示符号 , , 其码值为 58;

'A' 表示字母 A, 其码值为 65;

'a' 表示小写字母 a, 其码值为 97。

ASCII 字符集内, 数字、大写字母和小写字母关系为: '9' < 'A', 'Z' < 'a'。

必须注意, 字符类型数据只能是一个字符, 不能是一串字符。如 'ABC'、'780' 等, 都不是字符型数据, 而是字符串, 属于构造型数据。这种数据将在本书第六章数组一节中介绍。

与字符类型有关的运算符是六种关系运算符。这六种关系运算符都适用于字符类型数据。

适用于字符类型的标准函数只有顺序函数, 它们是前趋函数 (pred)、后继函数 (succ)、序号函数 (ord) 和字符函数 (chr)。例如:

$$\text{pred}('9') = '8' \quad \text{succ}('7') = '8'$$

$$\text{ord}('5') = 53 \quad \text{chr}(65) = 'A'$$

后两个顺序函数 ord 和 chr 在确定字符 c 的序号时 (次序值) 或者确定序号为 i 的字符时引用。

$\text{ord}(c)$ 给出字符 c 在字符集内的序号,

$\text{chr}(i)$ 给出字符集内序号为 i 的字符。

若 c 是数字字符, 则

$\text{ord}(c) - \text{ord}('0')$

是该数字字符的数值。例如:

$\text{ord}('5') - \text{ord}('0') = 5$

若 n 是小于 10 的正整数, 则对应字符的序号是

$\text{ord}('0') + n$

从而, 对应的字符是

$\text{chr}(\text{ord}('0') + n)$

例如:

$\text{chr}(\text{ord}('0') + 5) = '5'$

因此, 函数 ord 和 chr 可用于数字字符与其相应数值的转换。有些人将 ord 和 chr 函数定义为转换函数。

使用上述函数时应注意字符及其码值的范围。如 $\text{chr}(299)$, $\text{pred}('')$, $\text{succ}(' \sim')$ 均为非法表示。特别要注意数字字符不等于数字值, 如:

$\text{ord}('0') = 48$ 而不为 0

$\text{ord}('5') = 53$ 而不为 5

但

$\text{ord}('5') - \text{ord}('0') = 5$

$\text{ord}('0') - \text{ord}('0') = 0$

是正确的。

2.2.4 布尔类型

布尔类型数据只有两个: true 和 false 。它们是标准常量, 用于表示判断的结果, 规定其次序关系为

$\text{false} < \text{true}$

且序号规定为

$\text{ord}(\text{true}) = 1, \text{ord}(\text{false}) = 0$

布尔类型变量说明如下例:

$b1, b2 : \text{boolean};$

适合于布尔类型数据运算的有六种关系运算符和三个布尔运算符 NOT、AND、OR。其中 NOT 称为逻辑非, AND 称为逻辑与, OR 称为逻辑或。这些运算符的运算对象的类型都是布尔类型, 结果类型也是布尔类型。

通过上述运算符进行逻辑判断, 所产生的布尔类型结果, 也就是布尔表达式的值。例:

$\text{NOT} (10 > 5)$ 结果为 false

$(5 > 3) \text{ AND } (10.0 < 25.2)$ 结果为 true

$('A' > 'B') \text{ OR } (5.1 > 0.2)$ 结果为 true

$\text{NOT} (10 > 5) = \text{NOT} (10 < 10)$ 结果为 false

$(\text{false} < \text{true}) < ('A' > 'B')$ 结果为 true

虽然布尔类型数据比较简单, 但其应用却很广泛, 程序设计中主要用于布尔表达式的计算和逻辑判断。