

Pascal语言 基础知识

朱涵梁 编著

Pascal
Bazca

机械工业出版社

目 录

序言

第一章 Pascal 程序的基本组成部分1

1-1 首部和本体1

1-2 语法图3

第二章 基本数据类型6

2-1 标准数据类型6

2-2 列（枚）举数据类型7

2-3 类型定义的使用7

2-4 子域（界）数据类型8

2-5 有序数据类型9

2-6 练习10

第三章 基本运算12

3-1 整型运算12

3-2 实型运算12

3-3 比较运算13

3-4 布尔运算14

3-5 运算优先级别14

3-6 较复杂运算15

3-7 练习16

第四章 基本语句18

4-1 假定语句18

4-2 情况语句21

4-3 重复语句22

4-4 转向语句28

4-5 练习29

第五章	函数和过程	32
5-1	内部函数和内部过程	32
5-2	用户自定函数及形参和实参	33
5-3	用户自定过程	35
5-4	函数和过程的值参和变参	37
5-5	全局变量和局部变量	40
5-6	变量的适用范围	42
5-7	函数和过程的调用准则	44
5-8	简单递归	45
5-9	交互递归	45
5-10	函数和过程的中断退出	47
5-11	练习	48
第六章	数组和字符串	52
6-1	数组	52
6-2	多维数组和紧缩数组	54
6-3	字符串	55
6-4	练习	60
第七章	记录	62
7-1	记录	62
7-2	例子两则	63
7-3	记录变体	65
7-4	练习	69
第八章	集合	71
8-1	集合	71
8-2	例子两则	73
8-3	练习	76
第九章	文件	77
9-1	概述	77
9-2	与文件有关的若干函数和过程	78

9-3	过程 READ、WRITE 与 GET、PUT 的异同	82
9-4	文件应用实例	83
9-5	练习	90
第十章	指针	92
10-1	概述	92
10-2	指针变量的说明和定义	93
10-3	指针应用的准备知识	93
10-4	指针应用实例	95
10-5	练习	106
第十一章	总习题	109
附录		111
1	Pascal 的保留字	111
2	标准 Pascal 的内部函数及用法一览表	111
3	标准 Pascal 的内部过程及其功能简介	112
4	标准 Pascal 的标准标识符	113
5	Pascal 的运算符及其用法一览表	113
6	在 Apple 机双磁盘驱动系统上运行 Pascal 程序的操作要领	115
7	本书 Pascal 术语的英汉对照及索引	120
8	Pascal 语言的语法图	123
9	习题答案选录	129
参考文献		135

第一章 Pascal程序的 基本组成部分

1-1 首部和本体

Pascal 程序由程序首部和程序本体两大部分组成。下面是一个简单的 Pascal 程序。

```
PROGRAM LPKCAL(INPUT,OUTPUT);  
  VAR L:REAL;  
      K:REAL;  
      LPK:REAL;  
BEGIN  
  READ(L,K); (* READ STATEMENT *)  
  LPK:=L/K*100; (* ASSIGNMENT STATE-  
                  MENT *)  
  WRITELN ('LITERS PER 100 KILOMETE-  
           RS:', LPK)⊖  
  (* WRITE STATEMENT *)  
END.
```

上述程序中，第1行是程序首部。在Pascal语言保留字（参阅附录1）PROGRAM后跟一个程序名，两者要用空格隔开。空格至少一个，多则无妨。

程序名是一个标识符(IDENTIFIER)，本例中程序名为LPKCAL。程序名可由编程者任选字母和数字组成，但第

⊖ 程序中的上角逗号“'”，均应为撇号“'”，后同。

1个必须是字母，往后则随意，字母或数字均可，且长度没有限制。Apple Pascal规定只识别标识符的起首8个字符，更多的内容虽不拒绝，但不能识别。

紧跟在程序名后的是参数表，在本例中是 (INPUT, OUTPUT)。程序靠这些参数使计算机与其外围设备，诸如键盘、屏幕、打印机、磁盘驱动器等交换信息。INPUT代表键盘，OUTPUT代表屏幕。参数表中写上INPUT和OUTPUT表示程序将与键盘和屏幕打交道。

从上述程序的第2行起直到结束是程序本体。其中第2行到第4行是说明（定义）部分，这部分可以包含好些内容，诸如：标号 (LABEL) 说明部分；常量 (CONST) 定义部分；类型 (TYPE) 定义部分；变量 (VAR) 说明部分；函数 (FUNCTION) 与过程 (PROCEDURE) 说明部分等。凡说明（除函数和过程的外）均用冒号“:”，而定义都用等号“=”。在某一个程序中，上述各内容不一定全用上，但对用上的部分要求按上述次序排列。在PROGRAM LPKCAL中只有变量说明部分。在保留字VAR后列出程序中将用到的所有变量的说明。变量说明包括：变量名；冒号；该变量的所属类型。本例中L:REAL意即变量L的值是一个实数，L是一个实型变量。

上述程序的第5行起直至结束是语句部分。第5行的BEGIN和最后一行的END都是Pascal保留字，它们起语句括号作用，把所有的语句包括在内。

第6行“READ(L, K);”是读语句。程序运行至此将停下，等待用户从键盘上送入信息，依次赋值给变量L和K。根据变量说明，应键入两个实数。每个数后要键入空格 (SPACE) 以示送数结束。语句最后的分号“;”起分隔

语句的作用。

第7行是赋值语句。在本例中，先计算出 $L/K * 100$ ，其中“/”、“*”分别表示“除以”和“乘以”。然后，将计算结果赋值给变量 LPK。

第8行是写语句。本例中，机器将在屏幕上复写出单引号中所列全部内容，接着（不加空格）显示 LPK 的值。因为下一行 END 并非语句，本写语句末不必加用分号。

Apple Pascal 规定，所有包括在（* 和 *）两个复合符号中的内容，除了用（* \$ 和 *）括起的情况外，机器一概不加理会，只供人阅读理解之用。

本例的变量说明，允许改写为：

```
VAR L, K, LPK; REAL;
```

以省篇幅。也可以在每个变量附近插入附注如下：

```
VAR L, (* Liters *)
```

```
    K, (* Kilometers *)
```

```
    LPK (* Liters Per 100 Kilometers * ); REAL;
```

说明各变量含义。

1-2 语 法 图

语法图借助图解方法阐明书写 Pascal 语句的准则。下面列举数例，至于更详尽的语法图例，请查阅本书附录 8。

1. 书写程序的语法图如图 1-1 所示。

上图指出，整个程序分首部和本体两部分。长方形框中

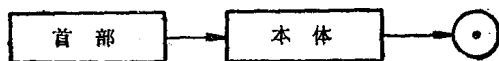


图1-1 书写程序的语法图

内容的书写准则，如本例中的首部和本体，还要再查阅有关语法图。圆形（或长圆形）框中所列，是必须在程序的该部位照写的内容。如本例中最右边的圆框中一点，表示程序終了必须写上句号，以示结束。

2. 首部语法图如图 1-2 所示。

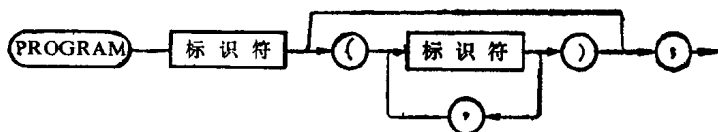


图1-2 书写程序首部的语法图

上图指出，长圆形框内的保留字 PROGRAM 必须在程序首部一开始就写上。接着是一个标识符，它给程序定名。标识符如何写要另查。程序名后有两个分支，根据情况选定某分支。直进分支用于要写参数表的情况。它指明，参数表由一个或多个标识符组成。其下回环指明各标识符要用逗号隔开。整个参数表用圆括号括起。程序名后的向上分支指的是 Apple Pascal 不要求在首部列明参数表，这点在后面第九章中还要细讲。最后的分号是将首部与其后内容分隔用的。

3. 标识符语法图如图 1-3 所示。

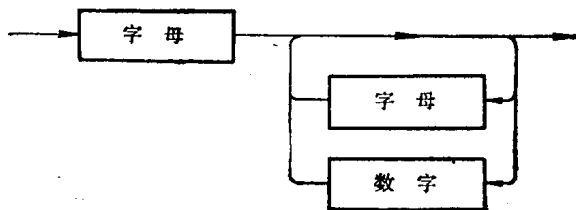


图1-3 标识符语法图

4. 程序本体语法图如图 1-4 所示。

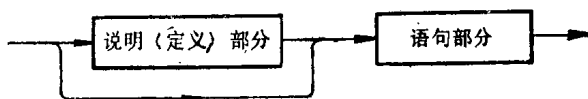


图1-4 书写程序本体的语法图

5. 变量说明语法图如图 1-5 所示。

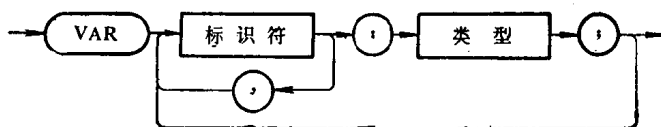


图1-5 变量说明语法图

6. 语句部分语法图如图 1-6 所示。

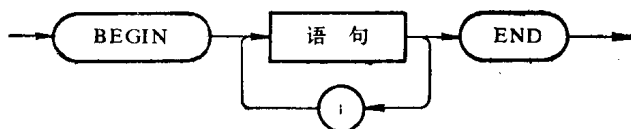


图1-6 书写语句的语法图

第二章 基本数据类型

变量的数据类型决定变量取值准则。若一个变量被说明为整数类型，则此变量应在所用机器规定的整数范围内取值。变量的数据类型不同，其运算可行范围也就有所不同。

2-1 标准数据类型

在 Pascal 语言中，下列四个称为标准数据类型：

1. INTEGER (整数类型)；
2. REAL (实数类型)；
3. BOOLEAN (布尔类型)；
4. CHAR (字符类型)。

例如，在某程序中的说明部分有：

```
VAR  X;CHAR;  
      Y;BOOLEAN;
```

则表示该程序中将用到的变量 X 属字符类型，变量 Y 属布尔类型。

对于不同的机器，INTEGER、REAL、CHAR 等数据类型可能有不同的取值范围，编程前宜查看使用手册。Pascal 语言用标准标识符 MAXINT 代表整数类型的最大取值。例如：Apple Pascal 以 MAXINT 代表最大整数 32767。附录 4 列有标准标志符表。

Apple Pascal 增设 LONGINTEGER (长整数类型)。

2-2 列（枚）举数据类型（ENUMERATED）

本数据类型借助列举方法来定义一个有序集合作为变量的取值范围。用户可以通过列举来设计自定的数据类型。设有下述变量说明：

```
VAR WEEKDAY: (MON, TUE, WED, THU, FRI, SAT);
```

则变量 WEEKDAY 即可在列举的范围内取值。因此，下列两个语句都是合法的。

```
WEEKDAY := WED;
```

```
WEEKDAY := FRI;
```

整数类型、布尔类型、字符类型是列举类型的特例。机器已知它们的全部可能取值，编程者不必列举。

列举类型也称纯量类型（SCALAR）。

2-3 类型定义的使用

如果将上节中的程序说明部分改写为：

```
TYPE DAY = (MON, TUE, WED, THU, FRI, SAT);
```

```
VAR WEEKDAY: DAY;
```

则效果完全相同。在上述说明（定义）部分的类型定义中，将 DAY 定义为列举类型，并列出其可能取值，而在变量说明中将变量 WEEKDAY 说明为 DAY 类型。DAY 不是标准类型，是用户自定的。根据已作的类型定义，即知 WEEKDAY 属列举类型，并知其取值范围。当程序的说明（定义）部分要多次用到某自定类型时，添写类型定义会使程序简洁，读写方便。

类型定义的语法图如图 2-1 所示。

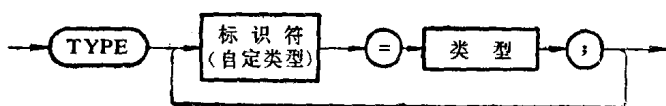


图2-1 类型定义语法图

2-4 子域 (界) 数据类型 (SUBRANGE)

子域类型规定的取值范围是列举类型取值范围的一个子域。子域类型是又一个用户自定的数据类型。设有下列说明 (定义) 部分:

```
TYPE DAY = (SUN, MON, TUE, WED, THU,
FRI, SAT); (* enumerated *)
WORKDAY = MON..SAT; (* subrange *)
VAR STARTDAY: WORKDAY;
    FWEEK: MON..WED; (* another subra-
nge *)
```

在上述类型定义中, 先将 DAY 定义为列举类型, 又将 WORKDAY 定义为 DAY 的一个子域, 从 MON 到 SAT。在变量说明中将变量 STARTDAY 说明为 WORKDAY 类型, 查类型定义即知 STARTDAY 属 DAY 的一个子域。接着又将变量 FWEEK 说明为 DAY 的另一个子域。

根据上述说明和定义, 下面两个语句是合法的;

```
STARTDAY := WED;
```

```
FWEEK := WED;
```

而下列语句是错的:

```
STARTDAY := SUN;
```

FWEEK_i = THU;

2-5 有序数据类型

四个标准数据类型中的 INTEGER、BOOLEAN、CHAR 加上枚举类型和子域类型，都是有序数据类型。有序类型取值范围中的每个单元，都有其唯一的前承单元和后继单元。当然，第一个单元没有前承单元，最末的单元没有后继单元。例如：整数类型中的 531 的后继单元是 532，字符类型中的 'Y' 的前承单元是 'X' 等等。注意，表达字符时，Pascal 规定要用单引号括起。

前承和后继单元可分别由前承函数 PRED 和后继函数 SUCC 求得。例如：下列赋值语句被执行后：

```
INTRi = SUCC(531);
```

```
CHi = PRED('Y');
```

整型变量 INTR 的值为 532，字符变量 CH 的值为 'X'。

有序类型的每一个单元都有它专有的序号，左起第 1 个单元为 0 号，第 2 个单元为 1 号，依次向右递增。用求序函数 ORD 可求得有序类型的单元序号。例如设有下列说明：

```
VAR FOODi : (BEEF, PORK, LAMB, CHICKEN,  
FISH);
```

则 ORD(BEEF) 的值为 0，ORD(PORK) 为 1，依此类推。

对字符类型讲，ORD 的反函数是 CHR。函数 CHR 能根据序号求得字符类型取值范围内相应的单元内容。例如：

```
ORD('A') 的值为 65,
```

```
CHR(65) 的值为 'A'。
```

2-6 练 习

1. 修改下列程序, 使它能在 Apple I 机上运行。上机操作要领见附录 6。

```
PROGRAM FIRST,
  BEGIN
    READ (A, B);
    C = A + B
    WRITELN (A, B, C)
  END
```

2. 运行下列程序, 弄清 WRITE 和 WRITELN 语句的区别。

```
PROGRAM TRYING (INPUT, OUTPUT);
  (* On Apple I, words INPUT and/or OUTPUT may
  be omitted *)
  VAR X, Y; REAL;
  BEGIN
    READ (X, Y);
    WRITELN (X);
    WRITELN (Y);
    WRITE (' X =', X);
    WRITE ('      ');
    WRITELN (' Y =', Y);
    WRITELN (' X + Y =', X + Y);
    WRITELN (X, ' + ', Y, ' = ', X + Y)
```

END.

3. 运行下列诸程序, 学会使用 PRED, SUCC, ORD, 及 CHR 等函数。

```
1) PROGRAM SCALAR,
  VAR X; INTEGER;
  BEGIN
    READLN(X); (* type RETURN to finish your
```

```

    number entering *)
WRITELN;
WRITELN (X);
WRITELN (PRED(X));
WRITELN (SUCC(X))

```

END.

2) PROGRAM MYTRY;

```

    VAR DAY; (SUN, MON, TUE, WED, THU, FRI,
              SAT);
    A, B; INTEGER;
BEGIN
    A; = ORD(WED); B; = ORD(FRI);
    WRITELN('WED=' , A, ' ', 'FRI=' , B)

```

END.

3) PROGRAM CHRFUNCTION;

```

    VAR X; INTEGER;
BEGIN
    X; = 65; WRITE (CHR(X), ' ');
    X; = 66; WRITE (CHR(X), ' ');
    X; = 70; WRITELN(CHR(X))

```

END.

4) PROGRAM TRYAGAIN;

```

    VAR DAY; (SUN, M, TU, W, TH, F, SAT);
    A; INTEGER;
BEGIN A; = ORD(M); DAY; = CHR(A) END.

```

(*Pay attention to error shown on screen when
this program is compiled*)

4. 用‘X’，或其他字符，借助写语句在屏幕上写出一个简单的中文字，如“中”、“王”等。

第三章 基 本 运 算

3-1 整 型 运 算

与整型数有关的加、减、乘运算分别使用 +、-、* 运算符号。如：

3 * 5,

COUNT + 1,

TOTAL1 + TOTAL2等。

此外，DIV（整除）运算得商，MOD（求模）运算得余数。如：

运算式	结果
20DIV 6	3
15DIV 2	7
9 MOD 2	1
21MOD 3	0

整型运算的输入和输出都是整型数。

3-2 实 型 运 算

与实型数有关的运算除众所周知的 +、-、* 外，还有/（实际）、ROUND（舍入）、TRUNC（去尾）等。如：

运算式	结果
15/2	7.5（实数）
9/3	3.0（实数）
ROUND(2.718)	3（整数）

ROUND(-2.718) - 3 (整数)

TRUNC(2.718) 2 (整数)

TRUNC(-2.718) - 2 (整数)

实除运算/允许输入实型数或整型数，但输出一定是实型数。ROUND 和 TRUNC 运算的输入是实型数，输出是整型数。

3-3 比较运算

Pascal 语言中有六种比较运算，分别使用下列六种运算符：

=, >, <, >=, <=, < >.

设有下列变量说明及程序片断：

VAR I, J, K; INTEGER;

X, Y; CHAR;

BEGIN

I := -4; J := 4; K := 0; X := 'C'; Y :=
'W',

则若干比较运算及其相应的运算结果如下：

比较运算	运算结果布尔值
I < J	TRUE
I + 8 = J	TRUE
X > Y	FALSE
I <= K	TRUE
I * 2 + 12 < > J	FALSE
K >= I + 5	FALSE

比较运算的输入可以是多种数据类型，但运算结果总是布尔类型。