

Pascal语言 基础知识

朱涵梁 编著

Pascal
Pascal

2
1

机械工业出版社

Pascal 语言基础知识

朱涵梁 编著



机械工业出版社

Pascal 语言是新兴的计算机语言之一。它结构性强，数据类型丰富，易学好懂，是科学计算和数据处理的有效工具。本书以有限篇幅着重讲清 Pascal 的基本原理。同时以国内当前已广泛应用的 Apple II 机为上机工具，叙述编写、运行 Pascal 程序的步骤、方法和技巧。

本书可供从事计算机应用的工程技术人员阅读，也可作为非计算机专业大、中专学生学习 Pascal 语言的基础教材，还适合有关人员作在职短训之用。

Pascal 语言基础知识

朱涵梁 编著

★

责任编辑：秦起佑

封面设计：田淑文

★

机械工业出版社出版（北京阜成门外百万庄南街一号）

（北京市书刊出版业营业性印书字第 117 号）

机械工业出版社印刷厂印刷

新华书店北京发行所发行·新华书店经营

★

开本 $787 \times 1092^{1/32}$ · 印张 $4^{1/2}$ · 字数 95 千字

1987年10月北京第一版·1987年10月北京第一次印刷

印数 0.001—5,500 · 定价：1.30 元

★

统一书号：15033·6930

编 者 的 话

本书着重讲清基本原理，同时也注意训练上机操作。书中所列全部程序，都在 Apple II 机上通过，使本书读者既能学习 Pascal 语言，又可学会在 Apple II 机上使用 Pascal 语言编程。

本书可作为非计算机专业大、中专学生学习 Pascal 语言的基础教材，也适宜供有关人员作为计算机语言在职教育短训班用书。

本书选材扼要，叙述简明，力求使读者可以少依赖教师指导。对已具备 BASIC 语言编程能力的读者，可以自学本书，配以上机操作，即能掌握 Pascal 语言的基本知识及上机运行方法。

本书讲授20学时，上机练习10学时，共30学时。几次教学实践证明这样安排是可行的、合理的。

本书经华中工学院计算机系袁蒲佳副教授审阅并提了宝贵意见。

敬请各方面读者对本书不完善处给予指正。

编者

序 言

Pascal 语言是近年来新兴的计算机语言之一。它由瑞士 **Niklaus Wirth** 教授等于六十年代后期研制成功，七十年代初正式发表。该语言以十七世纪法国人 **Blaise Pascal** 的姓来命名，为的是纪念这位数学家在青年时代研制成世界上第一台加法机作出的贡献。

Pascal 语言在 **ALGOL** 的基础上发展而成，並吸收了 **FORTRAN** 和 **PL/1** 等语言的优点，因此，它结构性好，数据类型丰富。无论对科学计算或是数据处理，都是有效的工具。近年来，国内外对用 **Pascal** 进行工业控制也已相继开展工作。此外，**Pascal** 的编译程序较小，一般微型机都能配带。它还易学好懂，因此，一发表后即颇受各界欢迎。美国波士顿大学以及其他一些院校，自1980年起，先后决定以 **Pascal** 语言为大学生的第一计算机语言。美国 **TEXAS INSTRUMENT** 公司也定 **Pascal** 为其微型机产品的主要语言。

标准 **Pascal** 没有动态数组，文件处理功能也欠完善。美国加利福尼亚大学圣地亚哥分校在标准 **Pascal** 基础上研制成的 **UCSD Pascal**，在数组、文件以及其他一些方面作了改进。美国 **Apple** 公司又修改、扩充了 **UCSD Pascal**，制成了 **Apple Pascal**。另外，目前比较流行的还有 **DEC-10 Pascal** 和 **ASPECT Pascal** 等。不同的 **Pascal** 的基本原理都是相同的，但各在某些方面有其不一致处，使用前宜针对具体系统、具体机型，详阅其用户手册。除注明外，凡本书中

提到 Pascal, 均泛指各 Pascal 而言。

本书针对 UCSD Pascal, 讲清基本原理, 並以国内目前已广泛采用的 Apple II 机为上机工具, 介绍编写、运行 Pascal 程序的步骤、方法和技巧。

编写本书力求短小精悍, 以有限的篇幅, 讲清 Pascal 基础知识。对本书未包罗的内容, 请参阅书后所列有关参考书。Pascal 语言还很年轻, 它的好多术语的中译名尚未定论, 学习中以多记相应的英文术语为宜, 以利对照。计算机语言犹如生活用语, 学习时必须加强实践。制订 Pascal 语言国际标准的工作正在进行, 学习中宜密切注意信息, 跟上发展步伐。

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 KILOMETER-  
           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 后列出程序中将要用的所有变量的说明。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

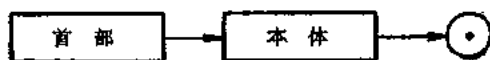


图1-1 书写程序的语法图

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。

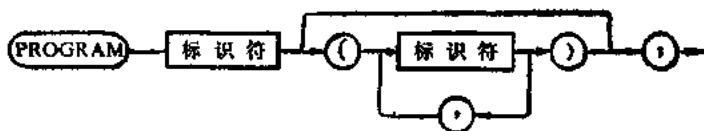


图1-2 书写程序首部的语法图

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。

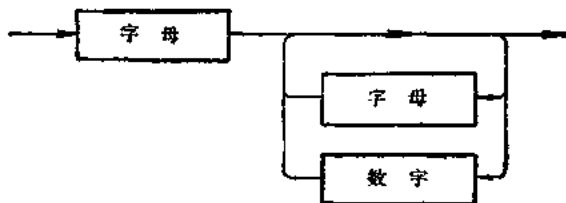


图1-3 标识符语法图

4. 程序本体语法图如图 1-4 所示。

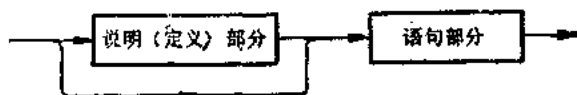


图1-4 书写程序本体的语法图

5. 变量说明语法图如图 1-5 所示。

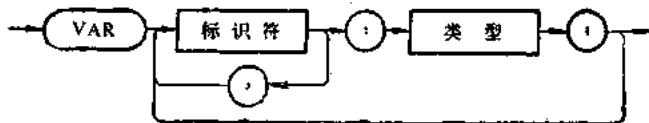


图1-5 变量说明语法图

6. 语句部分语法图如图 1-6 所示。

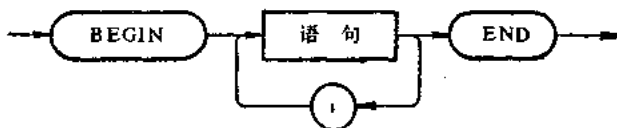


图1-6 书写语句的语法图

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, 並列出其可能取值, 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, 並知其取值范围。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。

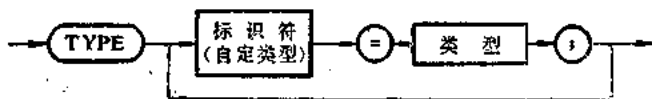


图2-1 类型定义语法图

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。没有下列说明 (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;
```