

计算机语言袖珍指南之五

PASCAL

~~R. 亨特 著~~

D. 瓦特

上海翻译出版公司

计算机语言袖珍指南之五

PASCAL

【英】D. 瓦 特 著
陆 皓 译

上海翻译出版公司

Pocket Guide to PASCAL

D. Watt

PITMAN BOOKS LTD, 1982

计算机语言袖珍指南之五

PASCAL

〔英〕D. 瓦特

陆 皓 译

上海翻译出版公司出版

(上海福州路 390 号)

本书由上海发行所发行 华东师大印刷厂印刷

开本 787×1092 1/64 印张 1 24/32 字数 49,900

1985 年 8 月第 1 版 1985 年 8 月第 1 次印刷

印数 1—71,500

统一书号: 13311·6 定价: 0.39 元

出版说明

英国皮特曼图书公司于1982年出版了一套入门性质的计算机语言袖珍指南，共计五本。第一本介绍程序设计的基础知识，后四本分别介绍 BASIC, COBOL, FORTRAN, PASCAL 这四种计算机常用语言的基本概念、程序结构、基本语句等内容。PASCAL 语言是七十年代初在 ALGOL 语言的基础上发展起来的，很适合于教学、科学计算和编写各种系统软件等，在微机和其他机器上的应用日益普及。

作者针对不熟悉计算机的初学者想学而又无从着手这一特点，通过最简单的程序例子来介绍每一种语言中各语句的基本内容，而对使用语言中的编程技巧不花过多的笔墨，从而使初学者能很快掌握要领，所以这套指南确实是少而精的计算机常用语言入门读物。对于初步掌握一种语言又想涉猎其他几种语言的读者，这套书也提供了入门的捷径。

当然，由于篇幅上的限制，使这套书存在一些不足之处。好在国内已出版了不少关于计算机语言的专门著作，已经入门想进一步深造的读者，可以参考、钻研这些著作。

目 录

如何使用这本袖珍指南	1
一、词汇表：分隔符	2
二、词汇表：记号	4
三、类型和操作	7
四、顺序类型	14
五、布尔类型	17
六、字符类型	19
七、整数类型	20
八、实数类型	23
九、指针类型	25
十、数组类型	28
十一、串类型	31
十二、记录类型	33
十三、集合类型	37
十四、文件类型（包括非正文输入/输出）	40
十五、正文类型（包括正文输入/输出）	44
十六、分程序	49

十七、常数定义.....	50
十八、类型定义.....	52
十九、变量说明.....	54
二十、语句.....	56
二十一、赋值语句.....	57
二十二、复合语句.....	59
二十三、If(如果)语句.....	60
二十四、Case(情况)语句.....	62
二十五、While(当)语句.....	65
二十六、Repeat(重复)语句.....	66
二十七、For(循环)语句.....	67
二十八、With(开域)语句.....	70
二十九、Goto(转向)语句和标号.....	71
三十、表达式.....	74
三十一、变量.....	78
三十二、过程.....	80
三十三、函数.....	83
三十四、参数.....	88
三十五、值参数.....	89
三十六、可变参数.....	91
三十七、保形数组参数.....	92

三十八、过程参数.....	95
三十九、函数参数.....	98
四 十、程序.....	100
四十一、作用域规则.....	104

如何使用这本袖珍指南

PASCAL 的各个特征都有说明它的完整的节,大多数节包含下列子节:用法,它描述在程序设计中某些特征的使用方法,和这些特征怎样符合语言的使用方法。语法,它给出了语言特征的语法规则——这个规则在编译时将受到检验。语义,它描述这个特征在程序执行时行为方面的含义。例子,它也给出阐明特征的使用方法。

有关数据类型的各个节都包含一个子节操作,它概括了所有适合所描述的数据类型的操作。

表示法

PASCAL 符号是这样的集合: begin end

; : = .

下面是应用在语法规则中的表示法:

= 含义是‘定义为’。

| 分隔的择一结构,例如,‘字母 | 数字’表示或者是字母或者是数字。

(...) 用于择一归类。

[...] 表示一个任选结构,例如,‘[else 语句]’表示选择或者 ‘else’ 后面跟一条语句,或者根本不跟任何语句。

{... • ...} 表示一种序列相似结构,例如,

(a) ‘{标识符*,}’ 表示一种用 ‘,’ 分隔的序列标识符;

(b) ‘{数字*}’ 表示一个包含一个或多个数字的序列;

(c) ‘{* (字母 | 数字)}’ 表示一个包含零或多个字母或数字的序列。

在语义和操作子节中,小写斜体字母表示表达式和常数,大写斜体字母表示标识符,变量,语句,类型和其他结构。

一、词汇表: 分隔符

在词法一级,一个程序正文是由记号和分隔符组成的。这些记号是决定语法结构和程序含义的符号,分隔符主要是适合读者的习惯。

语法

程序正文 = { * { * 分隔符 } 记号 }

分 隔 符 = 空格 | 行结束符 | 注释

注 释 = { 注释正文 }

分隔符在一个程序的含义方面是无效的。空格和行结束符可以用于改善程序正文的编排。注释可以用来帮助读者看懂程序。一个注释正文可以包含任一字符和行结束符,但不能含‘}’及‘*)’。

至少有一个分隔符必须位于两个依次记号之间,这些记号可以是关键字,标识符,无符号的整型常数或无符号的实型常数。在一个记号内不可以嵌入任何分隔符。

在这本小册子的其他节里,分隔符在语法规则上是被置之不理的。(不过,在这些其他节里,语法规则的编排提出了对于相应的一个程序结构的合适编排。)

二、词汇表：记号

语法

记 号 = 专门记号 |

标识符 |

指令 |

语句标号 |

无符号整型常数 |

无符号实型常数 |

字符常数 |

串常数

专门记号 = 关键字 | + | - | * | / |

= | < | <= | >= | > |

(|) | [|] | := | . | , | ; |

: | .. | †

关键字 = and | array | begin | case | const | div |

do | downto | else | end | file | for |

function | goto | if | in | label | mod |

nil | not | of | or | packed | procedure

program | record | repeat | set | then

to | type | until | var | while | with

标识符 = 字母{* (字母 | 数字)}

指令 = 字母{* (字母 | 数字)}

语句标号 = {数字*}

无符号整型常数 = {数字*}

**无符号实型常数 = {数字*}. {数字*} |
{数字*}. {数字*} [e [+ | -]
{数字*}]**

整型常数 = [+ | -] 无符号整型常数

实型常数 = [+ | -] 无符号实型常数

字符常数 = '字符映象'

串常数 = '字符映象{字符映象*}'

字符映象 = ' | 除省字号外的任一字符

**字母 = a | b | c | d | e | f | g | h | i | j | k | l | m | n |
o | p | q | r | s | t | u | v | w | x | y | z**

数字 = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

专门记号是具有固定意义的记号。

标识符用于程序名称,常数,类型,记录字段,变量,过程,函数,格式参数和界限。指令用于过程说明和函数说明。没有一个标识符或指令可具有象任一关键字那样的拼法。

语句标号可以用作语句的词头。语句标号是用十进制数值标识(以0开头是无意义的),语句标号的值在0到9999之间。

一个整型常数,实型常数或字符型常数表示一个简单类型的值。十进制表示法用于整型常数,实型常数以及代表‘10的幂’的字母e。一个包含n个字符映象($n > 1$)的串常数表示串类型的一个值:

`packed array [1...n] of char`

字符映象”表示省字号‘,’;任何其他字符映象表示它自己。一个行结束符不可以用于字符映象。一个字符常数和串常数中的空格和注释括弧也是字符映象。

等价表示

除在字符映象中外,下列字符具有等价表示:

字	符	等	价
abc...xyz		A B C...X Y Z	
↑		^ 或 (a)	
[		(.	
]		.)	
{		(*	
}		*.)	

例

标识符: A number X1 V 2000A

Oxygen Flow Rate

语句标号: 1 999

无符号整型常数: 0 1 123 32767

无符号实型常数: 0.0 1.234 1e-8 0.5E+6
3.33E10

字符常数: '' '' 'A' ''

串常数: '*' 'January' 'Murphy'
's Law'

三、类型和操作

用法

类型是每一个值、变量和函数的固有特性。类型是用一组值以及用可以在这些值上执行的操作来表示的。一个非构造类型(顺序, 实型或指针)的值是一种单一不可分的值。一个构造类型(数组, 记录, 集合或文件)的值由若干个成分值组成, 那些成分值可以是非构造的或者构造的。

语法

类型 = 类型标识符 | 类型描述符

类型描述符 = 顺序描述符 |

指针描述符 |

[Packed]数组描述符 |

[Packed]记录描述符 |

[Packed]集合描述符 |

[Packed]文件描述符 |

标准类型是由预先确定的类型标识符表示的。在一个程序内出现的每一个类型描述表示一种新的独特的类型（即使它和另一个类型描述的原文相同）。

对数组成分，记录成分和文件类型成分的类型几乎没有限制。不过，作用域规则防止了除通过指针外再发生递归类型的定义。

语义

前缀Packed在构造类型的语义方面是无效的。它要求对于构造类型的值表示为压缩存贮，是以减少有效存取到构造的成分为代价。对任何非紧缩构

造有效的操作，对除下列情况外的相应的紧缩构造也是有效的：(a)串操作(表11)仅对某些紧缩字符数组有效；(b)紧缩构造的直接成分不可以作为实变参数。

操作

PASCAL的种种类型类别构成层次结构(见图1)。对任何一类型类别有效的操作，对在层次结构内所有的子类别有效。

在表1—15中， $A, B, C, \dots$ 代表指定类型的变量，而 $a, b, c, \dots$ 代表指定类型的子表达式。

- (1) 所有类型的变量可作为可变参数使用。
- (2) 在表1中给出的操作对除了文件类型(和带有文件成分的构造类型)外的所有类型T都是有效的。

表1. copy(复制)操作

操 作	操 作 数	结 果	含 义
$V := e$	$V:T$ $e:T^+$	—	e 赋值给V (如果 e 值超界, 则失败)
$read(F, V)$	$F:T$ 的文件 $V:T^+$	—	读下一个F的成分到V里 (如果 $eof(F)$, 则失败)
$write(F, e)$	$F:T$ 的文件 $e:T^+$	—	一个具有 e 值的成分写 到F

* 应用赋值兼容的规则。

此外, 所有除文件类型外的值可以作为值参数那样使用。

(3) 在表2中给出的操作对所有非构造类型是有效的。

表2. Equality(等式)操作

操 作	操 作 数	结 果	含 义
$a == b$	兼容; real和integer	布 尔	a 等于 b
$a <> b$	同 上	布 尔	a 不等于 b