



全国成人高等教育规划教材

Pascal语言程序设计

教育部高等教育司 组编



高等教育出版社



全国成人高等教育规划教

Pascal 语言程序设计

教育部高等教育司 组编

陶龙芳 主编

陶龙芳 林成春 张纪勇 编著

高等教育出版社

内容简介

本书是根据教育部制定的“全国成人高等教育工学计算机程序设计课程教学基本要求”编写的。本书系统地介绍了 Pascal 语言的基本语法结构、各种数据类型、各种语句的语法及相应功能,并穿插了大量短小精炼的实例。全书始终贯穿结构化程序设计思想,注意程序设计的风格和程序设计基本技能的训练。为便于自学,每章前有内容提要、教学目标,每章后有内容小结、思考题和练习题。程序设计是实践性很强的课程,为此每章末还附有上机题,结合本章内容,给出实验目的、实验内容和实验要求。

本书适于作为成人高等学校 Pascal 语言程序设计课程教材,也可供各类工程技术人员阅读参考。

图书在版编目(CIP)数据

Pascal 语言程序设计/陶龙芳主编;林成春,张纪勇编
著. -北京:高等教育出版社,1999(2001重印)
全国成人高等教育规划教材
ISBN 7-04-006937-7

I. P… II. ①陶… ②林… ③张… III. Pascal 语言-程
序设计-成人教育:高等教育-教材 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 07734 号

Pascal 语言程序设计

教育部高等教育司组编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

经 销 新华书店北京发行所

印 刷 化学工业出版社印刷厂

开 本 787×1092 1/16

版 次 1999 年 7 月第 1 版

印 张 20.5

印 次 2001 年 12 月第 3 次印刷

字 数 490 000

定 价 23.70 元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,
请在所购图书销售部门联系调换。

版权所有 侵权必究

出版说明

为了加强成人高等教育教学的宏观管理,指导并规划成人高等教育的教学工作,保证达到培养规格,教育部于1998年4月颁布了全国成人高等教育公共课和经济学、法学、工学等学科门类主要课程的教学基本要求。教学基本要求是成人高等教育的指导性教学文件,是成人高等教育开展有关课程教学工作和进行教学质量检查的重要依据。为了更好地和更迅速地贯彻这些教学基本要求,我司又组织制订了全国成人高等教育主要课程教材建设规划。经过有关出版社论证申报和教育部组织的成人教育专家评审,确定了各门课程教材的主编人选及承担出版任务的出版社。

承担任务的出版社,遴选了学术水平高、有丰富成人教育经验的专家参加教材及教学辅助用书的编写和审定工作。新编教材尽可能符合成人学习特点,较好地贯彻了成人高等教育教学基本要求。推广使用这套教材,对于加强成人高等教育的教学工作,提高教学质量,促进成人高等教育的改革与发展具有十分重要的意义。

首批完成的有公共课和经济学、法学、工学三大学科门类共81门主要课程的教材。由于此项工作是一项基础性工作,具有一定的开创性,可能存在不完善之处。我司将在今后的教学质量检查评估中,及时总结经验,认真听取各方反馈意见,根据教学需要,适时组织教材的修订工作。

教育部高等教育司

1998年12月1日

前 言

Pascal 语言是 20 世纪 70 年代初期由瑞士苏黎世联邦技术学院 N·沃思教授开发的，并以历史上著名的数学家 B. Pascal 的名字命名。

Pascal 语言具有丰富的数据类型和简洁灵活的操作语句，适合于描述数值与非数值问题，可以用于书写系统软件和应用软件，为各种通用计算机系统所配备，成为世界上流行的程序设计语言之一。

Pascal 语言特别适合于教学，适于阐述程序设计的基本概念和结构化程序设计的方法。数据结构、编译原理、程序设计方法等课程中的概念、算法，都可以用 Pascal 语言描述。

本书系统地介绍了 Pascal 语言的基本概念，基本语法，各种数据类型，各种语句的格式、功能及特点；阐述了程序的基本结构及结构化程序设计的思想；通过大量短小精炼的实例说明了用 Pascal 语言进行顺序结构、选择结构、循环结构程序设计的方法。为便于自学，每章前有内容提要、教学目标，每章后有内容小结、思考题、练习题。程序设计是实践性很强的课程，为此每章末还附上上机题，结合本章内容给出实验目的、实验内容和实验要求。全书叙述力求深入浅出、通俗易懂。

本书是根据教育部制定的“全国成人高等教育工学计算机程序设计课程教学基本要求”编写的，适于作为成人高等学校 Pascal 语言程序设计课程教材，也可供其它各类高等院校及从事计算机应用的人士使用。

本书共分十二章，第七、八、九章由林成春编写，第十、十一章由张纪勇编写，其余各章由陶龙芳编写。全书由陶龙芳主编。

本书定有不足之处，敬请读者批评指正。

编 者

1998 年 10 月

目 录

第一章 Pascal 语言概述	1
1.1 程序基本组成	1
1.1.1 程序举例	1
1.1.2 程序组成	2
1.2 字符和符号	2
1.2.1 字符	2
1.2.2 符号	3
1.3 语法图和巴科斯—诺尔范式	6
1.3.1 语法图	6
1.3.2 巴科斯—诺尔范式	6
1.4 标准数据类型	7
1.4.1 整数类型	8
1.4.2 实数类型	9
1.4.3 字符类型	10
1.4.4 布尔类型	11
1.5 常量定义	13
1.5.1 常量定义的格式	13
1.5.2 常量定义的意义	14
1.6 变量说明	14
1.6.1 变量说明的格式	14
1.6.2 变量说明的意义	15
1.7 标准函数	16
1.7.1 算术函数	16
1.7.2 转换函数	17
1.7.3 进退函数	18
1.7.4 逻辑判断函数	18
1.8 Pascal 表达式	19
1.8.1 算术表达式	19
1.8.2 字符表达式	21
1.8.3 布尔表达式	21
1.9 程序基本结构及 N—S 结构流程图	23
1.9.1 程序的三种基本结构	23
1.9.2 N—S 结构流程图	25
1.10 TURBO Pascal 集成环境	26
1.10.1 TURBO Pascal 主菜单	26
1.10.2 文本编辑	27
1.10.3 程序的编译、连接、运行	28
1.10.4 联机帮助	29
1.10.5 常用的菜单命令和等效的功能键	29
本章内容小结	30
思考题	32
练习题	33
上机题	35
第二章 基本语句及顺序结构程序设计	37
2.1 Pascal 语言的语句	37
2.2 赋值语句	38
2.3 输入语句	40
2.3.1 标准过程 Read 语句	40
2.3.2 标准过程 Readln 语句	41
2.4 输出语句	43
2.4.1 标准过程 Write 语句	43
2.4.2 标准过程 Writeln 语句	44
2.4.3 输出格式	44
2.5 复合语句	49
2.6 顺序结构的程序设计	49
本章内容小结	50
思考题	50
练习题	52
上机题	56
第三章 选择性语句及选择结构程序设计	58
3.1 IF 语句	58
3.1.1 基本概念	58
3.1.2 IF 语句的嵌套	60
3.2 CASE 语句	62
3.2.1 基本概念	62
3.2.2 CASE 语句的语义扩充	63
3.2.3 IF 语句与 CASE 语句的关系	64
3.3 选择结构的程序设计	65
3.4 分层缩进对齐的书写格式	71
本章内容小结	71
思考题	72
练习题	73

上机题	76
第四章 重复性语句及循环结构程序设计	79
4.1 WHILE 语句	79
4.2 REPEAT 语句	80
4.3 FOR 语句	82
4.4 三种重复性语句的比较	85
4.5 多重循环	86
4.6 循环结构的程序设计	87
4.7 标号说明和 GOTO 语句	92
4.7.1 标号说明	92
4.7.2 GOTO 语句	93
本章内容小结	94
思考题	95
练习题	96
上机题	104
第五章 函数与过程	107
5.1 标准函数与标准过程	107
5.1.1 标准函数	108
5.1.2 标准过程	108
5.1.3 标准函数与标准过程在使用上的区别	108
5.2 函数说明和函数调用	108
5.2.1 函数说明	109
5.2.2 函数调用	110
5.3 过程说明和过程调用	111
5.3.1 过程说明	111
5.3.2 过程调用	112
5.4 值参数和变量参数	113
5.5 标识符的作用域	115
5.5.1 标识符在程序中的两种出现	115
5.5.2 标识符的使用原则	115
5.5.3 标识符的作用域规则	116
5.5.4 全程量和局部量	117
5.6 子程序的嵌套与递归调用	122
5.6.1 子程序的嵌套	122
5.6.2 递归调用	123
5.7 “自顶向下，逐步求精”的程序设计 方法	128
本章内容小结	130
思考题	130
练习题	131
上机题	139

第六章 简单类型——枚举类型和子界类型	144
6.1 类型定义	144
6.2 枚举类型	145
6.2.1 枚举类型定义	145
6.2.2 枚举类型特性	145
6.3 子界类型	148
6.3.1 子界类型定义	148
6.3.2 子界类型特性	149
6.4 类型相容和赋值相容	150
6.4.1 类型相容性	150
6.4.2 赋值相容性	151
本章内容小结	152
思考题	152
练习题	152
上机题	154
第七章 构造类型之一——数组类型	156
7.1 一维数组	157
7.1.1 一维数组类型的定义和变量说明	157
7.1.2 一维数组元素	158
7.1.3 使用数组变量应注意的问题	159
7.1.4 引入数组类型的作用	159
7.1.5 应用举例	160
7.2 二维数组	165
7.2.1 二维数组的定义	165
7.2.2 二维数组元素	166
7.2.3 应用举例	168
7.3 多维数组	171
7.4 紧缩数组	173
7.5 字符串变量	175
7.6 可调数组参数	179
本章内容小结	181
思考题	182
练习题	183
上机题	188
第八章 构造类型之二——集合类型	191
8.1 集合类型的定义与集合变量说明	191
8.1.1 集合类型的定义	191
8.1.2 集合变量说明	192
8.1.3 集合值的表示	193
8.2 集合运算	193
8.2.1 集合的赋值运算	193
8.2.2 集合的并、交、差运算	195

8.2.3 集合的关系运算	198	11.3 NEW 过程	269
8.3 应用举例	199	11.4 动态存储单元和动态变量	270
本章内容小结	201	11.5 对指针变量的运算	271
思考题	202	11.5.1 赋值运算	271
练习题	202	11.5.2 比较运算	272
上机题	206	11.6 指针变量与动态变量的区别与联系	272
第九章 构造类型之三——记录类型	208	11.7 链表	274
9.1 记录类型的定义和变量说明	208	11.7.1 单向链表	275
9.2 记录变量的访问和赋值	210	11.7.2 循环链表	279
9.3 开域语句——WITH	213	11.7.3 双向链表	280
*9.4 记录的变体	219	11.7.4 程序举例	282
9.5 应用举例	223	本章内容小结	286
本章内容小结	229	思考题	287
思考题	230	练习题	289
练习题	232	上机题	293
上机题	236	第十二章 应用软件开发简介	297
第十章 构造类型之四——文件类型	237	12.1 应用软件的开发过程	297
10.1 文件的概念	237	12.1.1 问题定义	297
10.2 文件的类型定义和变量说明	238	12.1.2 算法描述	298
10.3 对文件处理的过程和函数	240	12.1.3 程序编制	298
10.4 文本文件	247	12.1.4 程序调试	298
10.4.1 文本文件的特点	247	12.1.5 程序维护	298
10.4.2 用于文本文件处理操作的过程和函数	247	12.2 程序设计的方法和风格	299
10.4.3 Input 文件和 Output 文件	251	12.2.1 自顶向下、逐步求精的程序设计方法	299
10.4.4 数据类型的自动转换	252	12.2.2 程序设计的风格	299
10.5 无类型文件	253	12.3 程序的检查与调试	300
10.6 应用举例	255	12.3.1 程序中常见的错误	300
本章内容小结	258	12.3.2 程序的检查	300
思考题	258	12.3.3 程序的调试	301
练习题	260	本章内容小结	302
上机题	263	思考题	303
第十一章 动态数据结构和指针类型	266	上机题	303
11.1 动态数据结构	266	附录 1 Pascal 语言语法图表	306
11.2 指针类型和指针变量	267	附录 2 扩展的巴科斯-诺尔范式	311
		参考文献	316

第一章 Pascal 语言概述

内 容 提 要

通过一个程序实例说明 Pascal 程序的基本组成。

介绍 Pascal 程序的一些基本知识, 包括: Pascal 主要符号、标准数据类型、常量定义、变量说明、标准函数和 Pascal 表达式。

简述程序的基本结构和 N-S 结构流程图。

说明 Pascal 程序的运行方法。

教 学 目 标

重点掌握

1. Pascal 程序的基本组成, 注意各部分之间以分号隔开, 程序最后以圆点结束。
2. Pascal 语言符号的表示方法、不同作用, 特别是自定义标识符时应遵循的原则。
3. 四种标准类型的类型标识符 (integer, real, char, Boolean)、取值范围及所能执行的操作。
4. 常量定义和变量说明的语法规则和使用方法。
5. Pascal 语言的各种基本运算, 表达式与常规代数式的区别与联系。PASCAL 表达式的使用与求值方法。
6. 程序的三种基本结构和 N-S 结构流程图。

一般掌握

1. 常量与变量的区别, 变量与类型的联系。Pascal 程序要定义常量, 说明变量的原因。
2. 标准函数的含义及自变量的类型, 函数的类型。

一般了解

1. 语法图和巴科斯范式。
2. Pascal 数据类型分类。

1.1 程序基本组成

1.1.1 程序举例

计算半径为 r ^① 的圆的面积 s 。已知计算公式为 $s = \pi r^2$ 。

① 为使变量名在文字叙述中与程序中一致, 统一使用正体, 后边不再另行说明

Pascal 语言程序为:

```
PROGRAM AreaOfCircle (Input, Output);           {程序首部}
CONST Pi = 3.14159;                             {常量定义}
VAR s, r: Real;                                  {变量说明}
BEGIN
    Readln(r);                                   {输入 r 值}
    s: = Pi * Sqr(r);                            {语句部分}
    Writeln('s = ', s)                          {输出 s 值}
END.
```

} 说明部分
} 程序体

1.1.2 程序组成

一个 Pascal 程序是由程序首部和程序体两部分组成的。程序首部以分号结束，程序体以圆点结束。

程序首部包括三项内容，按照书写顺序，它们是：

- (1) PROGRAM——Pascal 程序标志，写在程序的开头。
- (2) 程序名——由程序设计者依据“标识符”规则（详见 1.2.2 节）自行定义，如上例中的 AreaOfCircle。

(3) 参数表——即本程序所用到的文件列表。写在程序名后面的括号内，文件之间用逗号分开，如上例中的 (Input, Output)。TURBO Pascal 参数表可以缺省。

程序体是程序的主体，也称“分程序”。程序体由说明部分和语句部分组成。

(1) 说明部分——写在语句部分前面，用来描述程序中所用到的数据的属性。说明部分顺序是标号说明部分、常量定义部分、类型定义部分、变量说明部分、过程与函数说明部分。有时说明部分的内容可以全部或部分地缺省，但书写顺序不得改变。

(2) 语句部分——程序体必须包括语句部分。语句部分是程序的执行部分，用来描述程序所执行的算法和操作。语句部分是一个复合语句。BEGIN 和 END 表示复合语句的开始和结束，其间包含一个或几个语句，语句间用分号“;”隔开。

1.2 字符和符号

任何一种高级语言，除规定一套严密的语法规则外，还必须规定一套基本的语法元素，以便按照语法规则将元素组织为语言的各种成分（如程序首部、常量定义、变量说明等）。Pascal 的基本语法元素称为符号，每个符号由一个或几个字符组成。在 Pascal 程序中，单个字符不一定具有确定的含义，但是符号具有确定的意义。

1.2.1 字符

Pascal 语言中的字符由以下三类组成：

1. 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

Pascal 中也可以使用大写字母,但除非在字符串(见下节)之内,否则大小写字母被认为是相同的字符。

2. 数字

0 1 2 3 4 5 6 7 8 9

3. 其它

+ - * / = < > . , : ; ↑ () [] { } '
空格等等

1.2.2 符号

Pascal 语言中的符号包括特定符号、标识符、数、字符串、注解、分隔符和标号等。

1. 特定符号

特定符号是指具有特殊意义的符号。它有两种,分别是字特定符号和非字特定符号。尤其需要指出的是,字特定符号虽然也是标识符,但这种特殊的标识符是由系统定义的,不能再做它用,因此特别取名为“保留关键字”,简称“保留字”或“关键字”,为醒目起见,本书采用大写字母印出。下面分别列出各特定符号。

1) 字特定符号(保留字)

AND	ARRAY	BEGIN	CASE	CONST
DIV	DO	DOWNT0	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

除了这 35 个特定符号外, TURBO Pascal 还有十几个保留字,例如用 STRING 来说明串变量(见第 7.5 节)等。

2) 非字特定符号

+ - * / < <= = <> >= > ↑
. , ' : ; := .. () [] { }

2. 标识符

标识符是以字母开头的字母数字序列。例如: A, B1, Char, CharNumber 都是标识符。

标识符分为预定义的标准标识符和用户定义的标识符两类。

1) 预定义的标准标识符

TURBO Pascal 预先定义了若干个标识符,下面列出 40 个,它们有特定的含义,用来作为标准的常量名、类型名、函数名、过程名和文件名,分列如下:

标准常量	False	Maxint	True				
标准类型	Boolean	Char	Integer	Real	Text		
标准函数	Abs	Arctan	Chr	Cos	Eof	Eoln	Exp

	Ln	Odd	Ord	Pred	Round	Sin	Sqr
	Sqrt	Succ	Trunc				
标准过程	Dispose	Get	New	Pack	Page	Put	Read
	Readln	Reset	Rewrite	Unpack	Write	Writeln	
标准文件	Input	Output					

为了增加程序的可读性，本书中预定义的标准标识符第一个字母大写，其余小写，只有用户定义的标识符才全部用小写字母表示。

2) 用户定义的标识符

这类标识符是指程序员根据需要所定义的，它用来为常量、变量、类型、函数、过程及程序取名。

用户在选用标识符时需注意如下几个问题：

(1) 必须严格按照标识符的定义规则来构成标识符，即必须是以字母开头的字母数字序列。例如：1A, 234, W·John, char set 都不是合法的标识符。前两个错在以数字开头，后两个错在中间插入了其它字符“·”和空格。

(2) 选用的标识符不能与保留字同名，即前面讲过的保留字不能当作标识符使用。

(3) 语法上允许将预定义的标准标识符作为用户定义的标识符使用。例如：Abs 本来是绝对值函数名，如果在程序定义部分将 Abs 定义为常量 50，则在程序中 Abs 不再代表绝对值函数，而代表常数 50，这是与保留字不同之处。但是为了使程序清晰易读，少出错误，定义的标识符应避免和标准标识符同名。

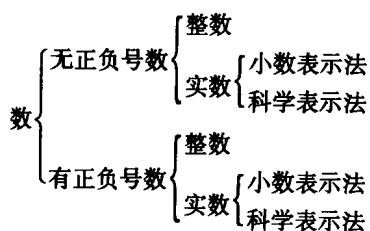
(4) Pascal 对标识符的长度没有限制，但是，在具体机器上实现时，对标识符的长度是有限制的。如 MS Pascal 规定，标识符最长为 31 个字符，如果标识符长度超过 31，则只识别前 31 个字符。

(5) 虽然 Pascal 对标识符中所用的字母没有规定，但是为了使程序容易理解，经常选用见名知义的标识符。例如用 V 标识体积，用 T 标识温度，用 sum 标识和数等。有时候为了使含义清晰，一个标识符往往由几个英文单词组成。例如用 charnumber 标识字符个数，它由两个英文单词组成。为了醒目起见，每个英文单词的第一个字母写成大写，如 charnumber 写成 CharNumber。对 Pascal 语言来说，大小写字母是等同的，所以 charnumber 和 CharNumber 是同一个标识符。

(6) 标识符必须严格遵照“先定义后使用”的原则，即一个用户定义的标识符必须先出现在程序的说明部分，然后才能出现在程序的语句部分（只有一个例外，见第十一章）

3. 数

在 Pascal 语言中，采用十进制表示，包括下列各类常用数。



例如：25，17.28，3E-1，36.5E+8，+2.7E3，-18.69E-10等都是Pascal中的数。其中第一个数是无正负号整数，最后两个是有正负号实数，其余三个是无正负号实数。与通常数的表示一样，其中正号可以省略，因此，有正号数与无正负号数的意义相同。例如：+2.7E3=2.7E3。若实数中不出现字母E，则是小数形式的表示法，否则为科学表示法，字母E后的整数表示10的幂次，如：3E-5=3×10⁻⁵。

Pascal中数的表示方法有特定的格式：

(1) 若数中出现小数点，则小数点的前面必须有数字。因此

.58 58.

都不是Pascal语言中的数。

(2) 若数中出现字母E，字母E之前必须有数字，E之后必须是整数。因此

E-3 -5E2.5

也都不是Pascal语言中的数。

4. 字符串

用一对撇号括起来的字符序列称为字符串。如

'a'，'PASCAL'，'This is a string'

若构成字符串的字符序列中有撇号，应将串内的撇号用两个撇号表示。如字符串：

'This is a "string"'

表示字符序列

This is a 'string'

在字符串中，允许出现计算机使用的系统字符集中的任意字符，例如\$虽不是Pascal字符，但只要它属于该计算机使用的系统字符集，就可以出现在字符串中。

在字符串内，大、小写字母表示不同的字符，具有不同的意义。

在第六章中可以看到，字符串是串类型的常量。如果字符串是由单个字符组成的，像'a'、'+',''''（中间两个撇号表示一个撇号），在下一节可以看到，它们是字符类型的常量。

5. 注解

在Pascal语言中，可以采取在程序中插入注解的方法来增加程序的可读性。注解是两端用字符{和}括起来的字符序列。为了避免含混不清，规定注解内不得包含右花括号“}”。注解可以出现在程序中的任意两个符号之间（注意不是两个字符之间）。注解内部允许换行。注解的内容十分丰富，可以说明程序的题目、功能、用法、算法要点、输入/输出数据，指明语句的功能、模块的功能及调用关系等。由于注解只起着解释的作用，对程序的执行不产生任何影响，因此，Pascal编译程序将不理睬程序中的所有注解。为了便于阅读，本书全部采用中文注解。有些计算机不具有花括号，这时可用（* *）代替。

6. 分隔符

Pascal程序是由具有一定意义的符号按一定的语法规则构成的，但Pascal程序不是一个长长的符号流。在有些符号之间，需要用其它符号隔开，这种用于隔开符号的符号称为分隔符。在Pascal语言中，空格、注解和行结束符都可作为分隔符。

在Pascal符号中，有不少符号本身既具有明确的意义，同时又有分隔相邻符号的双重功能，如分号、括号等。但对于另外一些符号则不然。例如在

IF x=y THEN BEGIN xy:=2.8; i:=4 END

中 $x=y$ 的等号“=”两边可以不留空格，但 x 的前面与 y 的后面都需要留空格。又如，若保留字 BEGIN 与其前面的保留字 THEN 或与其后面的标识符 xy 不隔开，就会被当作一个新的标识符 THENBEGIN 或 BEGIN xy 。一般来说，在一个 Pascal 程序中，任何相邻的两个符号之间，或程序正文的第一个符号之前，都可以插入一个或几个分隔符，而任何相邻的保留字、标识符、无正负数、标号或指示字之间至少要有一个分隔符。

需要注意的是：任何符号内部不允许插入分隔符。特定符号（包括保留字和非字特定符号）、标识符（包括预定义的标准标识符和用户定义的标识符）、数、标号和指示字等，它们的内部都不允许出现空格和注解，也不允许将它们任何一个分作两行书写，因为任何两行之间都隐含有一个行结束符。

1.3 语法图和巴科斯－诺尔范式

1.3.1 语法图

Pascal 语言的语法规则可以用语法图进行描述。语法图是由一些图形符号组成的。

下面以语法图描述程序的组成为例，说明语法图中的符号含义。

图 1.1 中最左边的文字“程序”是被定义的对象，它是由“程序首部”、后面跟分号“;”、“程序体”和句点“.”组成的。图中矩形框围起来的内容在别处另行定义，而端圆框（此图内为圆框）围起来的内容是必须出现的实际字符，它们是不必进一步定义的对象。图中的箭头只表示先后次序，不是语法规定的内容。

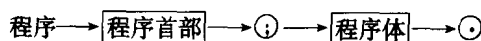


图 1.1 语法图举例

以后各章陆续讲到的 Pascal 定义、说明和语句都可用语法图描述，详见附录 1。

1.3.2 巴科斯－诺尔范式

语法图可以用图表示，也可以用其它方式表示。这里我们介绍扩展的巴科斯－诺尔 (Backus-Naur) 范式来表示语法规则。扩展的巴科斯－诺尔范式要用到一些记号，我们把这些记号以及它们的含义列出如下：

记 号	含 义
=	定义为
	或
()	与通常的括号相同
[]	括号内的内容可出现 0 次或 1 次
{ }	括号内的内容可出现 0 次或任意次
.	一个语法规则的结束

由于扩展的巴科斯-诺尔范式要用到 =、(、)、[、]、{、}、· 等符号,而这些符号又是 Pascal 语言的终结符号。因此,为了进行区分,我们把 Pascal 语言中的终结符号都用双引号括起来。

例如:

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

即数字定义为 0 或 1 或 2...或 8 或 9。

又例如:

常量定义部分 = ["count" 常量定义 ";" {常量定义 ";" }]

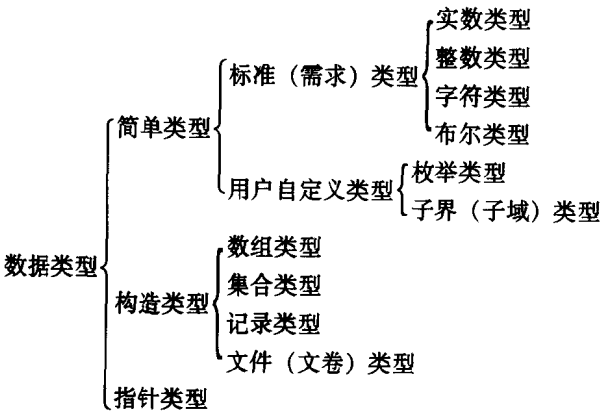
即常量定义部分或为空,或为 count 后面跟上常量定义、分号,或再跟上任意多个(可以为 0 个)常量定义、分号。这就是说,常量定义部分可以为空,也可以是 count 后面跟上一个或多个常量定义,每个常量定义后都要有一个分号。

Pascal 语言中的语法规则也都可以用巴科斯-诺尔范式描述,详见附录 2。

1.4 标准数据类型

用计算机进行计算,不可避免的要涉及到常量、变量以至更加复杂的各种计算对象,在 Pascal 中,一个程序所用到的每个常量、变量和函数标识符都或隐式地或显式地与一种数据类型(简称类型)相联系。每种数据类型是由一组值和能在这组值上进行的全部操作组成的。因此,类型既规定了与该类型相联系的常量、变量和函数标识符的取值范围,也规定了对它们能执行的运算。

Pascal 语言的数据类型极为丰富,共有 11 种类型,具体划分如下:



说明:

(1) Pascal 中的数据类型分为三大类:

第一类是简单类型。它是指值为有序的类型。它又分为标准类型(也称需求类型)和用户自定义类型两种。标准类型有实数类型、整数类型、字符类型和布尔类型。按照 Pascal 语言国家标准规定,标准类型是已经在系统中为用户定义好了的类型,用户不必进行定义或说明即可直接引用。用户自定义类型有枚举类型和子界类型(也称子域),是用户根据需要自定义并使用的类型。

第二类是构造类型。包括数组类型、集合类型、记录类型和文件(也称文卷)类型。因为它要由其它类型按一定的规则构造而成,所以结构比较复杂。

第三类是指针类型。它是一种动态的简单数据类型，可以用来处理动态数据结构。

(2) 关于有序类型和顺序类型。第一类的六种基本类型都是有序类型，因为它们的值都是有序的。例如，整数 $8 > -2$ ，实数 $25.3 > 13.8$ ，字符 'B' 在 'A' 之后。

除去实数类型之外，第一类的五种基本类型都是顺序类型，因为它们的值不仅是有序的，而且有顺序号。例如，整数 -8 的顺序号是 -8，或者说 -8 的“下一个整数”是 -7。同样，字符也是有顺序号的，例如，A 的“下一个字符”是 B。Pascal 规定，布尔值“真”(True) 的顺序号为 1，“假”(False) 的顺序号为 0。实数类型则不然，它的值虽然是有序的，但不能用顺序号标识。例如，32.5 的“下一个实数”是什么呢？是 32.6？还是 32.51？无法确定。

1.4.1 整数类型

整数类型简称整型，用标准标识符 Integer 标识。

1. 整数类型的值

在数学中，整数是一个无限集合。但是，计算机只能表示它的一个有限子集。在 Pascal 语言中，整数类型的值是整数中这样的子集：

$$- \text{Maxint}, -\text{Maxint} + 1, \dots, -1, 0, 1, \dots, \text{Maxint} - 1, \text{Maxint}$$

其中，Maxint 是一个由 Pascal 系统预定义的标准常量，它代表所使用的计算机系统 Pascal 语言允许使用的最大整数值，由具体的编译器决定。例如，TURBO Pascal 规定 Maxint 的值为 32767。

2. 适用于整数类型的运算

1) 算术运算

+	-	*	DIV	MOD
加	减	乘	整除	取模或取余

两个整型数据进行运算，结果仍为整型。前三个运算符的意义是明显的，第四个运算符 DIV 是将两个整型量相除后取其商的整数部分作为结果。若 DIV 的两个运算分量同号，则结果为正，否则结果为负。例如：

$$\begin{array}{ll} 14 \text{ DIV } 4 = 3 & 14 \text{ DIV } (-4) = -3 \\ 4 \text{ DIV } 14 = 0 & -14 \text{ DIV } (-4) = 3 \\ -14 \text{ DIV } 4 = -3 & -4 \text{ DIV } (-14) = 0 \end{array}$$

在标准 Pascal 中规定，MOD 是取模运算。但在有些版本中，如 TURBO Pascal，MOD 是取余运算。

(1) 取模运算：规定 $a \text{ MOD } b$ 中的 b 必须大于零，否则无意义。其运算规则如下：

① $a \geq b$ 时，不断地从 a 中减去 b ，直到得到一个小于 b 的非负数。例如：

$$14 \text{ MOD } 4 = 2$$

② 当 $a < b$ ，且 $a \geq 0$ 时，结果为 a 。例如：

$$4 \text{ MOD } 14 = 4$$

③ 当 $a < b$ ，且 $a < 0$ 时，将 b 不断地加到 a 上，直到结果是一个小于 b 的非负数止。例如：

$$-4 \text{ MOD } 3 = 2 \quad -6 \text{ MOD } 3 = 0 \quad -14 \text{ MOD } 4 = 2$$

$a \text{ MOD } b$ 取模的结果，其值必 ≥ 0 ，大小在 $0 \sim (b-1)$ 之间。

(2) 取余运算：规定取 a 整除 b 以后的余数。例如：

$$\begin{array}{ll}
 14 \bmod 4 = 2 & 4 \bmod 14 = 4 \\
 -4 \bmod 3 = -1 & -6 \bmod 3 = 0 \\
 -14 \bmod 4 = -2 & 4 \bmod (-14) = 4 \\
 -14 \bmod (-4) = -2 & -4 \bmod (-14) = -4
 \end{array}$$

如果有 $a \bmod b$ ，所得结果的符号与 a 相同，其绝对值在 $0 \sim |b| - 1$ 之间。

$a \bmod b$ 相当于 $a - (a \text{ DIV } b) * b$ 的运算。例如：

$$13 \bmod 4 = 13 - (13 \text{ DIV } 4) * 4 = 1$$

将以上的取模和取余结果相比较，可以看到：当 $a \geq 0$ 、 $b > 0$ 时， $a \bmod b$ 无论按“取模”或“取余”运算，得到的结果是相同的，只是当 $a < 0$ 时，两种运算结果才可能不同。

利用 $a \bmod b$ ，可以判断 a 能否被 b 整除。当 $a \bmod b = 0$ 时， a 能被 b 整除。

利用 DIV 和 MOD 运算，可以求出一个正整数 n 的个位、十位、百位、千位……位数。例如：将四位数 7341 分离出个、十、百、千位的程序段为

$$\begin{array}{ll}
 7341 \bmod 10 = 1 & \text{(个位数)} \\
 (7341 \bmod 100) \text{ DIV } 10 = 4 & \text{(十位数)} \\
 (7341 \bmod 1000) \text{ DIV } 100 = 3 & \text{(百位数)} \\
 7341 \text{ DIV } 1000 = 7 & \text{(千位数)}
 \end{array}$$

2) 关系运算

$$\begin{array}{cccccc}
 = & < > & < & < = & > = & > \\
 \text{等于} & \text{不等于} & \text{小于} & \text{小于或等于} & \text{大于或等于} & \text{大于}
 \end{array}$$

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)。标识符 True 和 False 属于布尔类型。

1.4.2 实数类型

实数类型简称实型，用标准标识符 Real 标识。

1. 实数类型的取值范围及运算精度

在数学中，实数是一个无限的、连续的集合。但是计算机由于受字长的限制，只能表示它的一个有限的子集。这是同整数类型类似的。例如有些 16 位的计算机，能接受的实型量的绝对值约在 $10^{-38} \sim 10^{38}$ 之间。超过这个范围，则发生溢出错误。绝对值大于 10^{38} 的数称为“上溢”数；绝对值小于 10^{-38} 的数称为“下溢”数，以输出零表示。它们可以用数轴表示，如图 1.2 所示。

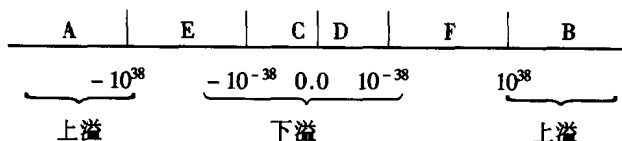


图 1.2 实数类型的取值范围

E 区和 F 区中的实数，是计算机所能表示的范围。A 区和 B 区是上溢区，C 区和 D 区是下溢区。

计算机不仅限制了实数的范围，而且限制了它的运算精度。例如，某一计算机系统，最多