



朱 建 秦 军 唐素兰 编

PASCAL

程序设计

东南大学出版社

PASCAL 程序设计

朱 建 秦 军 唐素兰 编

东南大学出版社

内容简介

PASCAL 语言是一种语句功能强、数据类型丰富、程序结构严谨的高级程序设计语言,常用于开发系统软件、应用软件以及科学计算。PASCAL 语言是大专院校非计算机专业程序设计教学的最佳选择,也可以作为非计算机专业科技工作者计算机应用能力培养的起点。

本书从标准 PASCAL 语言的概念出发,循序渐进地介绍其基本数据和语句、程序结构、控制结构和子程序结构,在结合介绍自定义数据类型的同时,列举了大量应用实例。书中设有 TURBO PASCAL 对标准 PASCAL 的扩充和上机操作集成环境的介绍。

责任编辑:雷家煜

PASCAL 程序设计

朱 建 等编

*

东南大学出版社出版发行

(南京四牌楼 2 号 邮编 210096)

江苏省新华书店经销 南京邮电学院印刷厂印刷

*

开本 787×1092 毫米 1/16 印张 14.75 字数 368 千

1996 年 11 月第 1 版 1996 年 11 月第 1 次印刷

印数:1—4000 册

ISBN 7—81050—173—9/TP·29

定价:18.00 元

(凡因印装质量问题,可直接向承印厂调换)

前 言

PASCAL 语言是 70 年代初产生的一种结构化的计算机程序设计语言,且已成为世界上广泛流行的一种程序设计语言。PASCAL 语言具有数据类型丰富、语句功能较强、结构清晰、书写格式自由、表达能力强、移植性好等优势,使其成为一种较理想的教学用语言,极有利于培养学生良好的程序设计技术和风格。

PASCAL 语言的课程不仅可以作为高校计算机专业的专业基础课,而且也逐渐成为非计算机专业的一门必修基础课。PASCAL 语言作为计算机应用能力的一种入门语言工具,帮助学习者初步掌握程序设计的方法和技巧。本书是基于这一目的而编写的。

本书从标准 PASCAL 语言的语法入手阐述相关概念,同时通过广泛的例子,讨论 PASCAL 语言的各个组成部分和应用。算法是解决处理问题的钥匙,本书从算法的角度思考、分析问题。同时选择描述算法的结构化流程图——N—S 图,帮助读者掌握算法设计方法。对同一个问题的处理,既可以用 PASCAL 语言实现,也可以用其它任一种语言实现,请读者着力于掌握程序设计的方法,而不仅仅是学习 PASCAL 语言。

在本书中大部分概念和实例是基本的,同时也引入了一些较难的、为计算机后续课程服务的例子,初学者如若暂时放弃,并不影响对基本 PASCAL 语言的掌握。

本书由朱建、秦军、唐素兰编写,其中朱建编写第 1、4、5 章,秦军编写第 2、3、7 章,唐素兰编写第 6、8、9 章。南京邮电学院网络信息技术研究所张顺颐教授审阅了全书,并提出许多有益的意见,我们表示感谢。

本书的不足之处,敬请读者指正。

编 者

1996. 3

目 录

1	程序设计概述	1
1.1	绪论	1
1.2	程序设计	1
1.3	数据与算法	2
1.4	PASCAL 语言	4
	习题 1	6
2	数据基本处理	7
2.1	数据与数据类型	7
2.2	运算与表达式	12
2.3	标准函数	16
2.4	赋值语句	20
2.5	输入、输出语句	21
2.6	程序简例	27
	习题 2	29
3	控制语句	32
3.1	复合语句	32
3.2	选择控制	33
3.3	循环控制	39
3.4	标号说明与转向语句	46
3.5	例	48
	习题 3	51
4	过程与函数	55
4.1	子程序的概念	55
4.2	过程	56
4.3	函数	67
4.4	递归	73
4.5	过程参数和函数参数	79
4.6	例	85
	习题 4	90
5	枚举、子界和集合类型	93
5.1	自定义数据类型	93
5.2	枚举类型	94
5.3	子界类型	99
5.4	集合类型	103

5.5	类型之间关系	108
5.6	例	110
	习题 5	115
6	数组类型与串	117
6.1	数组类型	117
6.2	多维数组	122
6.3	字符数组和字符串	127
6.4	布尔数组、紧缩数组	131
6.5	例	137
	习题 6	143
7	记录与文件	144
7.1	记录类型	144
7.2	开域语句	146
7.3	记录的变体部分	151
7.4	文件	155
7.5	文件操作	157
7.6	文本文件	161
7.7	类型文件	164
7.8	例	167
	习题 7	171
8	指针与动态数据结构	173
8.1	指针类型	173
8.2	链表	177
8.3	二叉树	183
8.4	例	186
	习题 8	187
9	程序设计	190
9.1	逐步求精	190
9.2	调试	195
9.3	Turbo PASCAL 集成开发环境(IDE)简介	196
附录		
A	PASCAL 语法图	200
B	Turbo PASCAL 6.0 库函数和过程	205
C	Turbo PASCAL 6.0 错误信息和错误运行	217
参考书目		230

1 程序设计概述

1.1 绪 论

现代电子数字计算机是方便而快速地进行复杂、大量计算的工具。利用计算机处理实际问题时,有一个类似的模式:IPO;其中,I(Input)为输入反映客观事物特征的数据及其操作,P(Process)为按客观规律和计算机解决问题的原理的处理过程,O(Output)为按一定形式及格式输出处理的结果。

计算机具有逻辑判断、存储程序和大量数据,并由程序自动控制执行的特点。在使用计算机时,必须借助于计算机设计和制造者提供的环境和工具,处理实际问题。

程序设计语言就能代表一个能够支持该语言的抽象计算机,它是一种工具,程序设计就是使用工具的技术。

1.2 程序设计

一个程序是一个指令的序列。一个计算机程序就是一个通过计算机指令描述的序列。执行程序或称运行程序,就是让计算机执行指令序列。

程序是由程序设计者描述,并且由计算机来实现的序列。

程序中指令序列将由计算机依次执行,其指令的组织规则决定了执行顺序。计算机程序处理的对象称为数据,其处理的方法来源于一个称作算法的描述。

程序设计语言就是对算法和数据的描述。它包括计算机语言、汇编语言、高级程序设计语言和专门设计语言等。

用计算机所支持的形式,把算法和数据用程序设计语言描述出来的工作称为计算机程序设计,简称程序设计。它一般包括如下步骤:

(1)明确问题。有了明确的目标才能编制实现目标的方案。

(2)建立模型。这是对问题的抽象,可用数学的或物理的模型,准确地明确问题的目标和约束条件,利用计算方法获得求解过程。

(3)算法描述。将问题分解成一系列可解的步骤。

(4)编程。将算法步骤和数据,使用程序设计语言编制程序(简称编程),得到源程序。

(5)调试运行。由计算机编译程序、连接编辑程序进行语法语义检查,翻译成可执行的目标程序(机器指令序列),选择各种特定的数据作为输入,产生结果,分析判断,以期得到正确的结果。

(6)维护。编写技术文档资料,收集、调整数据信息。根据实际变化及时准确地进行增、删、改工作。

1.3 数据与算法

计算机处理的对象是一个数据的集合,数据是客观事物特征的抽象,由可识别的符号表示。

在程序设计语言中,数据可以以数值的形式出现,也可以用标识符表示。例如 3.3.14159, false, 'true' 等,或如 pi, x, name 等。

Pascal 语言中,标识符必须以字母开头,由字母、数字字符序列组成。用来表示程序、常量、变量、过程、函数等。

数据可以表示单一的值,称为简单数据;也可以表示若干值的组合,称为构造数据或称复杂数据,所有这些称为数据类型。程序设计语言提供了一些数据类型称为标准类型。用户(或称程序设计者)也可以自定义一些数据类型。

算法是指解决问题的有限操作步骤的描述。它可分数值运算和非数值运算。其描述方法可以有:自然语言、流程图、N-S 图等。

例如,数值计算 $y = \sqrt{x}$, $x \geq 0$ 的近似计算,可用牛顿迭代公式:




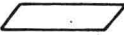

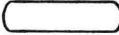
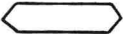

$$y_{n+1} = (y_n + x/y_n)/2$$

对给定的精度 ϵ , 求满足 $|y_{n+1} - y_n| \leq \epsilon$ 的 y 值。

自然语言描述此算法如下:

- (1) 初始化, $n=0$, 设 ϵ 的值(如 $\epsilon=0.0001$);
 - (2) 计算 $y_0 = (1+x)/2$;
 - (3) 迭代计算 $y_{n+1} = (y_n + x/y_n)/2$;
 - (4) 判别 $|y_{n+1} - y_n| \leq \epsilon$, 成立则 y_{n+1} 即为解。计算结束。
 - (5) 否则转向, $n=n+1$, 做(3);
- 流程图常用符号见下表 1.1。

表 1.1

名称	符号	含义
箭头		流向
矩形		处理功能
圆形		连接点
平行四边形		输入或输出
菱形		判断
圆形边框		开始或终止
尖头框		循环
双边矩形		调用子程序

上例算法流程图表示如下图 1.1。

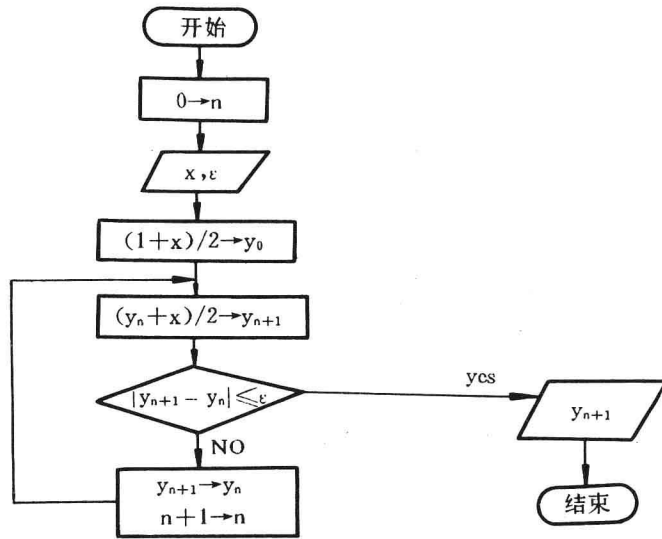


图 1.1 算法流程图例

N-S图是由美国科学家 Nassi 和 Skneiderman 提出的,有如下图 1.2 六类图构成:(a)顺序,(b)分支,(c)多分支,(d)、(e)循环,(f)调用子程序。

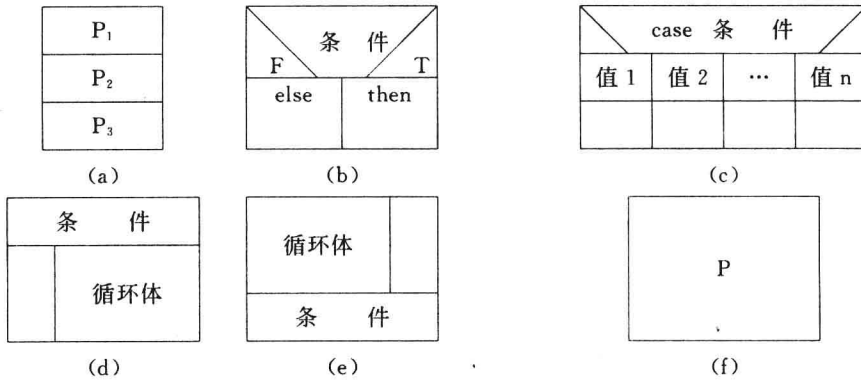


图 1.2 N-S图组成

上例用 N-S图描述如下图 1.3。

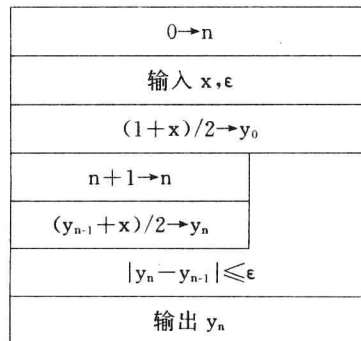


图 1.3 N-S图例

程序设计中的算法具有如下五个基本特征：

- (1) 有限性,必须在有限的操作执行步骤后结束;
- (2) 确定性,计算规则和步骤必须唯一确定,不可含糊定义;
- (3) 可行性,各步骤均必须在有限时间内完成操作;
- (4) 输入,初始化,以便生成原始数据;
- (5) 输出,至少有一个输出,得到结果,并鉴别问题解的正确性。

衡量一个算法的标准是多方面的。其中包括正确易读,实现简单;同时在资源占用时,占用空间要少,运算时间要短等等。

1.4 PASCAL 语言

Pascal 语言是 70 年代最有影响和最重要的一种程序设计语言。Pascal 语言由 N. Wirth 设计,以历史上著名数学家 B. Pascal 的名字命名。突出了 E. W. Dijkstra 提出的“结构化程序设计”思想。它注重结构化特征,以及语言的可靠性、易验证性、概念的清晰性和实现的简化性。

Pascal 语言除适用于科学计算外,特别适用于教学,使程序设计的基本概念和结构可以用 Pascal 所提供的一种系统的、精确的、合理的方法来描述,如计算机专业课,数据结构,编译技术,算法都可用 Pascal 来描述。适用于描述顺序型系统软件与应用软件,Pascal 语言不仅具有丰富的语法成份、丰富的数据类型,而且还提供构造新类型的方法,实现对各种各样数据结构的描述。

Pascal 语言与其它高级语言一样,有一套表示构成语言各种符号之间组合规律的语法结构或语法形式;有一系列反映特定定义的各种符号的语义。通过数据、运算、控制及输入输出来描述、表达程序。

一个 Pascal 语言源程序由程序首部和分程序两部分组成。分程序由程序说明部分和语句部分构成。

程序首部以 PROGRAM 开头,其后是标识符定义的一个程序名,以及用圆括号括起的逗号隔开的程序参数,参数用于说明程序中使用的文件变量标识符。

程序说明部分说明了程序中使用的数据类型、数据变量、过程或函数等成份。其中 Pascal 语言提供的标准常量、标准类型、标准过程和标准函数,可以直接使用,不必说明。

程序说明部分,依次可以出现:标号说明、常量说明、类型说明、变量说明、过程与函数说明部分。

语句部分是对操作的描述,由 BEGIN 开头,END 结尾,加上句点“.”结束,分号“;”隔开的语句序列构成。一般语句分为简单语句(赋值语句、过程调用、转语句、空语句),构造语句(复合语句、条件语句、重复语句、开域语句)。

Pascal 语言的语法规则也可用语法图描述,详见附录。程序结构语法图如图 1.4。

图 1.4 中,○,◯表示终结符号,□表示语法现象(待说明),→表示流向。

根据 Pascal 语言语法,描述 1.3 节中算法例子如下:

```
PROGRAM exal(input,output);
  CONST
    eps=0.0001;
```

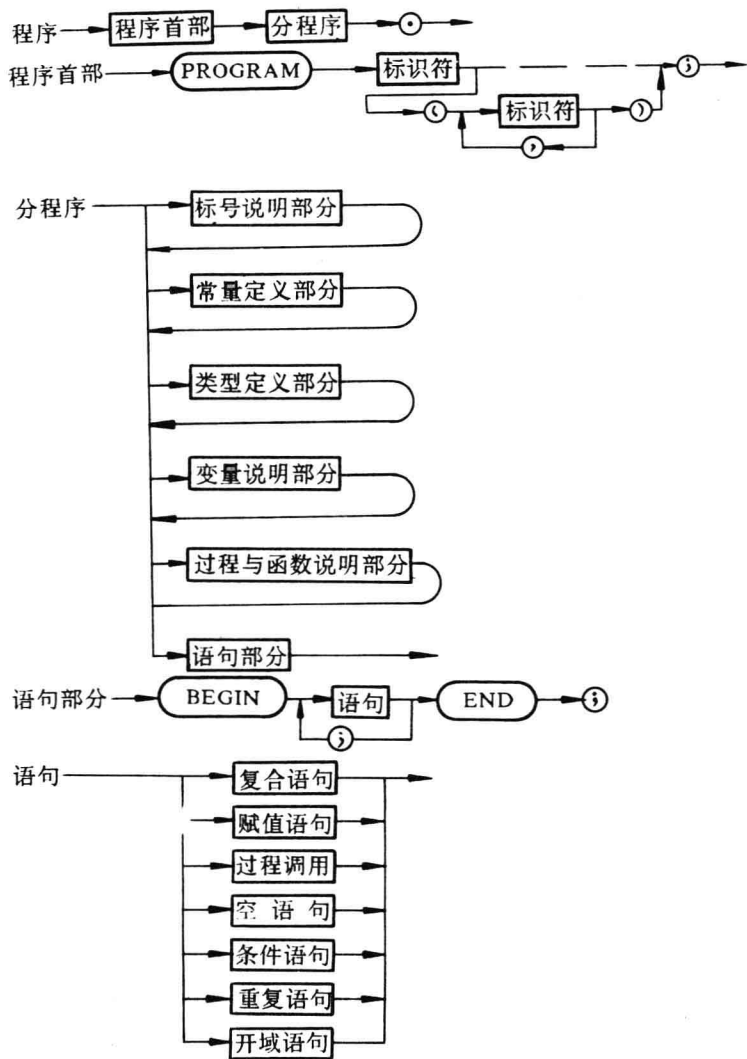


图 1.4 程序结构语法图

```

max=1000;
TYPE
  ar=ARRAY [0..max]OF real;
VAR
  n:integer;
  x:real;
  y:ar;
BEGIN
  read(x);
  n:=0;
  y[0]:=(1+x)/2;

```

```

REPEAT
  n:=n+1;
  y[n]:=(y[n-1]+x/y[n-1])/2
UNTIL abs(y[n]-y[n-1])<=eps;
write(n:4,y[n]:0:4)
END.

```

在这一例程中,给出程序名 exal,标识符 input,output 称为标准文件变量,分别代表键盘和显示器屏幕。定义了两个常量 eps 与 max。定义了一个数组类型 ar,该数组可含1001个实型数据元素,0..max 定义了下标变化范围。说明了变量 n,x,y 分别属于整型 integer,实型 real,数组类型 ar。语句部分描述了算法,读取 x,n 取初值0,数组元素 y[0]取值,反复执行 n 次,且每次变量增1,后一数组元素取前一数组元素的运算值。直到两者差的绝对值(abs 函数)小于或等于 eps 为止,有格式地输出 n 值与数组元素 y[n]。程序以句号“.”结束。

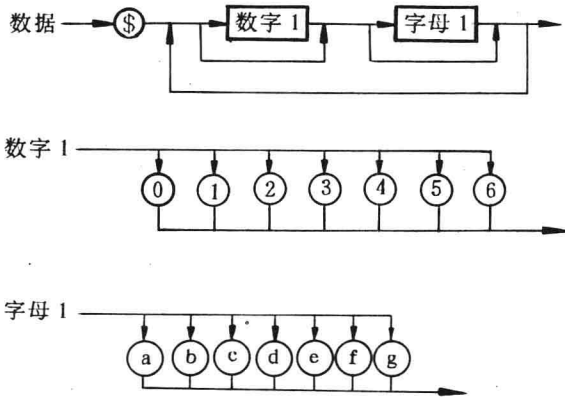
上述程序中用户定义了一系列标识符,分别为程序名 exal,常量 eps,max,类型 ar,变量 n,x,y。使用了系统标准标识符,其中变量 input,output,类型 integer,real,过程 read,write,函数 abs。使用的系统保留字有:程序首部定义 PROGRAM,常量定义 CONST,类型定义 TYPE,数组定义 ARRAY-OF,变量说明 VAR,语句标志 BEGIN-END,REPEAT-UNTIL。同时使用了直接量0.0001,1000,0,1,2,运算符+、-、/、<=等等一系列符号。

用户定义的标识符,根据需要、定义方法,先定义或说明,才能在程序中使用。

Pascal 语言程序对书写格式,几乎没有特别要求,一行可以写几个语句,一个语句可以写几行。但为了易读、易改,一般采用上例中分层缩位的书写方式。

习 题 1

- 1.1 试描述 Pascal 语言源程序的组成部分。列举几种不同的组成形式。
- 1.2 有一类数据定义如语法图描述,判断下列符合语法图的数据有哪些?



- 数据:(1)23 (2)\$23 (3)2\$3 (4)23\$ (5)ab (6)AB (7)\$ab (8)\$a8 (9)\$a5 (10)\$³f (11)abc\$ (12)\$000 (13)\$bc45\$de (14)\$ (15)

2 数据基本处理

数据是计算机程序处理的对象。或者说,计算机程序的执行是对这些数据的处理过程。由于计算机系统处理的事务多种多样,因此有各种各样的数据。例如,在科学计算中,处理数据主要是整数和实数;在书刊的编辑中,处理数据主要是字符;在企事业管理中,处理的数据主要是表格等等。在 Pascal 语言中,数据的描述由程序中的说明部分来实现,本章介绍数据、数据类型及数据的简单处理手段。

2.1 数据与数据类型

数据作为程序处理的对象有不同的表达形式和取值范围,还具有不同的运算规则。数据的这种特征在高级语言程序设计中,是通过数据类型来定义的,各种数据类型表达了各种数据的不同性质,程序中所使用的每个数据都有一个和它相联系的类型。该类型决定了数据可能具有的值,也决定了对该数据能进行的操作。在 Pascal 语言中,数据类型可分为三大类,见图 2.1。

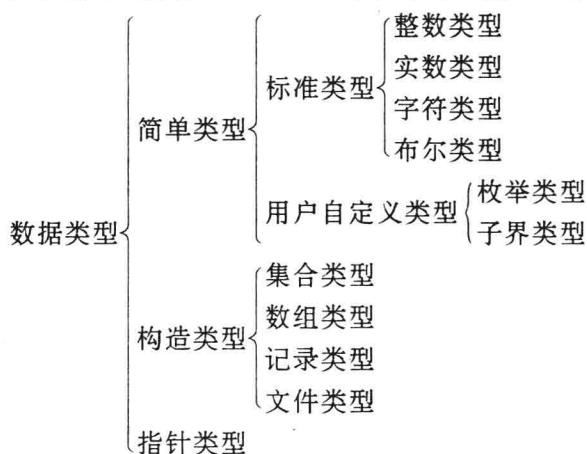


图 2.1 Pascal 数据类型

在详细介绍标准数据类型之前,先介绍有关 Pascal 的字符集及标识符的基本概念。其他数据类型将在以后各章节中叙述。

2.1.1 字符集及标识符

在 Pascal 语言中,组成单词、语句乃至整个程序的字符的全体称为 Pascal 字符集。它们是 ASCII 代码字符集的子集。Pascal 语言中包括三类字符。

数字: 0,1,2,3,4,5,6,7,8,9。这些数字字符可以组成整数和实数,也可以组成下面将要介绍的标识符。

字母: a,b,c,⋯,z(或 A,B,C,⋯,Z)。通常字母的大小写形式是不分的。

特殊符号(主要是一些运算符和分隔符): +, -, *, /, <, =, >, (,), [,], {, }, ,, ., ; 等。

Pascal 语言的单词就是由上面介绍的三类字符组成,其中最基本的单词是标识符,它是用来给常量、变量、过程、函数等起名用的。

一般地说,标识符是用以字母打头的字母、数字串表示,语法图如图 2.2 所示。

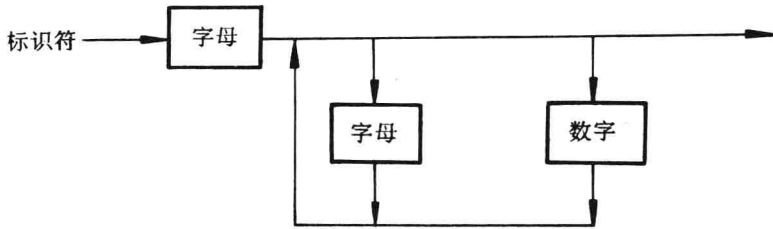


图 2.2 标识符的语法图

从语法图中可看出,标识符的第一个字符必须是字母,后面可以不跟字符,也可以跟任意多个字母和数字,但不可跟任何其它符号。

例 a,bookfile,X13,Begin 是合法的标识符。而 12P,IBM-PC,xy', \$ 40 是不合法的标识符。

对标识符的长度(即标识符中包含字符的个数)是没有限制的,但在具体的计算机系统中,对标识符的长度有自己的约定。通常仅取前面 8 个字符作为有效字符。

例 bookfile1,bookfile2 两标识符被认为是相同的。

在 Pascal 语言中,标识符按其性质可分为三种类型:

(1) 保留字

保留字在 Pascal 语言中有特定的含义,用户不能作其他用途。Pascal 语言的保留字有:

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

(2) 标准标识符

标准标识符在 Pascal 语言中有着特定的含义,用户可以根据自己的需要重新定义。但一般应尽量避免给标准标识符重新定义,以免引起混乱。在 Pascal 语言中,标准标识符有:

abs	arctan	boolean	char	chr
cos	dispose	eof	eoln	exp
false	get	integer	input	ln
maxint	new	odd	ord	output
pack	page	pred	put	read
readln	real	reset	rewrite	round

sin	sqr	sqrt	succ	text
true	trunc	unpack	write	writeln

关于以上所列的保留字和标准标识符的含义和用法将在后面的章节陆续介绍。

(3) 用户自定义标识符

为了编写程序的需要,用户可以自己定义一些标识符,给常量、变量、过程、函数等起名,用户在自定义标识符时应注意:

- ① 标识符应以字母开头,后面可不跟或跟任意字母和数字。
- ② 不能与 Pascal 的保留字相同,尽量避免与标准标识符相同。

例:

x12, blue, pi, loop,
a, b,

2.1.2 常量与变量说明

在程序执行过程中其值保持不变的数据,称为常量。在程序执行过程中其值可以改变的数据,称为变量。在 Pascal 语言中,常量和变量定义在程序说明部分加以说明。为使程序易读,常量或变量的命名应遵循“见名知义”的原则。

1) 整数类型常量

整数类型常量包括正整数、负整数和零,其语法图如图 2.3。

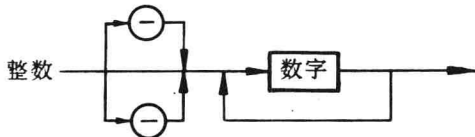


图 2.3 整数类型常量语法图

例 2.1 +1234,56,-99,0 都是合法的整数类型常量,而 31.0,2.78,20.5 则是非法的整数类型常量。

任何计算机只能表示整数的一个子集,其中最大的整数由 Pascal 语言中的标准标识符 maxint 来表示。它的值与计算机的字长有关,不同类型的计算机 maxint 的值为: $\text{maxint} = 2^{\omega L - 1} - 1$, ωL 为计算机字长。

2) 实数类型常量

实数类型常量包括正实数、负实数和实数零。其语法图如图 2.4 所示。

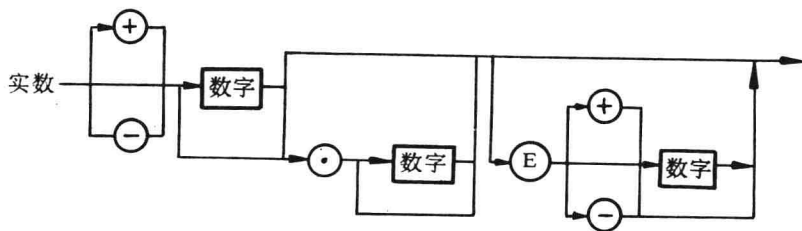


图 2.4 实数类型常量语法图

Pascal 的实数类型常量有两种表示形式。

(1) 十进制表示法,即日常使用的十进制小数形式。

例 2.0,0.0,-378.08,-63.7,988.6。

(2) 科学表示法,即将实型数据表示成指数形式。用 E 代表底数 10,E 后面的整数表示 10 的幂次。

例 7.2E13,-3.58E24,1.2E-8。

实数类型常量表示法要注意以下几点:

(1) 十进制表示法中,小数点的前后均要有数字。

(2) 科学表示法中,字母 E 前必须有一小数形式的实数,E 后必须有一整数。

例 2.2 12.15,-0.134,-986.78,-9.3E-7,83.1E12 等均是合法的实数,而 0.,.76,+E-2,6.07E-0.7,7E8 等均为不合法的实数。

在数学中,实数是一个无限连续集合。而计算机的表示范围受到机器字长的限制。如一个字长为 16 位的计算机,可接受的实型量的绝对值在 $1E-38$ 到 $1E+38$ 之间,超出范围,则将发生溢出错误。

3) 布尔类型常量

布尔类型常量有真和假两个值,分别用标准常量 true 和 false 表示。

布尔类型是有序类型:

false 的序号:0

true 的序号:1

并规定:false<true

在计算机内部常用一个字节表示布尔类型数据。

4) 字符类型常量

字符类型常量是括在两个单引号之间的 Pascal 字符集中的一个字符,其语法图如图 2.5。字符类型数据是有序的,它以相应的 ASCII 码作为序号。

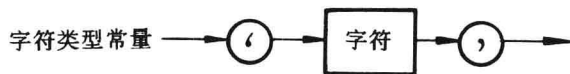


图 2.5 字符类型常量语法图

例 2.3 'a','B','4',';'均为字符类型常量。当单引号自身也作为字符时,对应的字符常量必须书写二个引号,再用单引号括起来即''。

5) 符号类型常量

前面我们介绍的各种类型常量的表达形式,就其该常量本身,这类常量通常也称为字符常量。换句话说,这类常量可以在程序中直接使用。Pascal 语言既允许常量的直接引用,也可以用常量标识符说明后使用,常量标识符和它所代表的量必须在程序说明部分事先定义。符号型常量语法图如图 2.6。

例 2.4

```
CONST
```

```
    Pi=3.1415926;
```

```
    black=' ';
```



```
ch='a';
m=true;
```

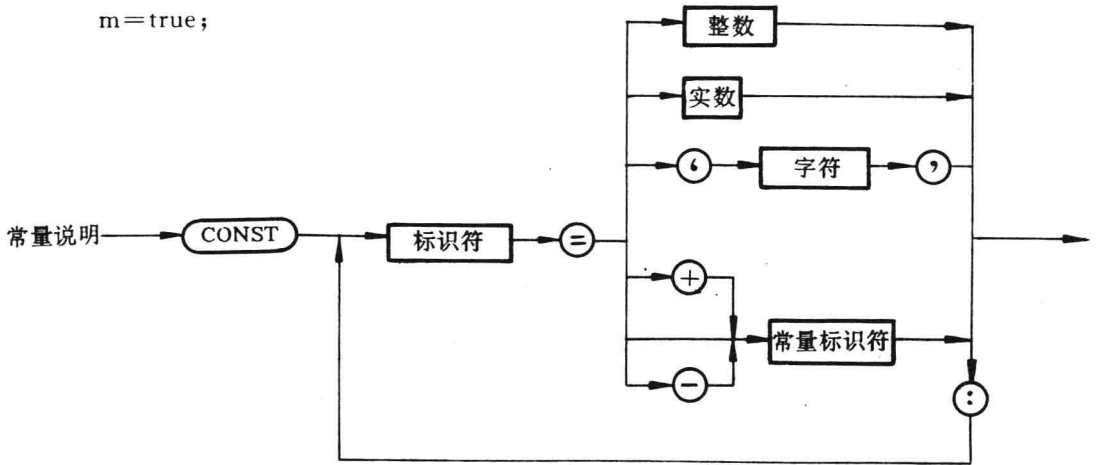


图 2.6 符号类型常量语法图

由此可见,使用常量标识符,可使程序中每个用到常量的地方,不必书写常量值,而代之以使用常量标识符,一方面避免写错,另一方面在修改大程序时,若要修改某一常量,只需在常量定义中修改一次(如将 $\text{Pi}=3.1415926$ 改成 $\text{Pi}=3.14$),就可改变全部程序中所有的 Pi 值。使程序易读易改。

6) 变量说明

程序中每个数据不是常量,就是变量。变量必须由用户自己为它定义一个变量名,用以标识变量,Pascal 语言使用标识符作为变量名,标识符是以字母打头的一个字符串。同时每个变量都具有自己的类型。变量名与变量类型均在程序说明部分加以定义,以便编译程序为其分配内存,如编译程序遇到未说明的变量名,将给出错信息,变量说明的语法图如图 2.7。

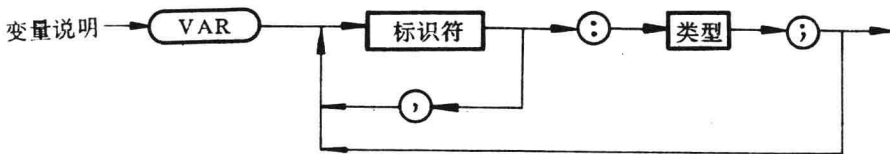


图 2.7 变量说明语法图

其中 VAR 是 Pascal 的保留字,表示变量说明段的开始,每个变量标识符或由逗号隔开的多个变量标识符,必须在它的后面说明其类型。

例 2.5

```
VAR
  x,y,z:real;
  i,j,k:integer;
  c:char;
  b:boolean;
```

其中 real、integer、char、boolean 分别是 Pascal 语言中实数类型、整数类型、字符类型和布尔类