

PASCAL 程序设计语言 (修订版)

李启炎 宋秋杰 编著

TP312

PASCAL

程序设计语言

(修订版)

● 李启炎 宋秋杰 编著 ●

TP312
30-2

PASCAL

程 序 设 计 语 言

(修订版)

李启炎 宋秋杰 编著

同济大学出版社

内 容 提 要

PASCAL 语言是国内外广泛应用的一种结构程序语言, 常用于编写系统软件、应用软件以及科学计算等。本书从基本概念出发, 循序渐近地对 PASCAL 语言中功能较强的数据结构, 如枚举型、子界型、集合、记录、数组、文件、栈和队列、指针和动态数据结构以及它们在程序设计中的应用作了较详细的叙述。附录中还介绍了 MS-PASCAL 的编译程序, 以利读者在 IBM-PC 型微机上使用。全书分九章, 每章均有适量的例题和习题。

本书可作为大专院校有关专业的教材或参考书, 也可供科学技术人员及其他从事计算机程序设计的人员阅读。

责任编辑 洪建华

封面设计 王肖生

PASCAL 程序设计语言

(修订本)

李启炎 宋秋杰编著

同济大学出版社出版

(上海四平路1239号)

新华书店上海发行所发行

复旦大学印刷厂印刷

开本: 787×1092 1/16 印张: 14.75 字数: 377千字

1985年1月第1版 1989年6月第2版 1989年6月第1次印刷

印数: 1—31000

定价: 4.80

ISBN7-5608-0310-5/TP·20

初版序言

PASCAL 语言是在 60 年代末 70 年代初由瑞士 N. Wirth 教授提出的。它是在 ALGOL 60 的基础上发展的一种结构程序设计语言（语句结构化及数据结构化）。具有功能强，数据类型丰富，写出的程序简洁易读等特点。适用于数值和非数值问题的描述。既能用于应用程序设计，也能用于系统程序设计。因此，它已成为目前最流行的几种程序设计语言之一。在欧美国家中越来越多的大学选用 PASCAL 语言作为教学的第一计算机语言。

与 PASCAL 语言相近的 ALGOL 语言在我国已得到一定程度的普及，这使学习和推广 PASCAL 语言有了良好基础。鉴于 PASCAL 语言的上述特点，希望在我国能有更多的高等院校将它作为计算机教学语言，使学生一开始就培养良好的程序设计风格和掌握较全面的程序设计技巧。为此，在我校 PASCAL 语言培训班多次使用的讲义的基础上，我们编写了本书。编写方法采用子集扩充法，即每学完一章都能编写出完整的程序实例进行上机实习。由简到繁逐步扩充，以利于学习。

任何一种高级语言必须有一个编译程序将它翻译成机器指令程序，才能在一个特定的计算机系统中进行工作。而编译程序本身也是在一个特定的环境下工作的（即特定的计算机硬件及操作系统的支持）。故在一个特定的工作环境下，每一种高级语言就有一种具体的实现方案。一个实用的 PASCAL 实现方案除了包括标准 PASCAL 语法外，总还包括某些扩充的语法成份，以增强语言的功能。因此，我们在取材时，除了标准 PASCAL 语言成份外，还适当地将某些有代表性的实现方案中的一些重要的语法扩充部分作为附录编入本书。这些扩充部分来自 DEC 公司的 PDP-11 PASCAL V1.0 及 UCSD PASCAL。

全书共分九章和三个附录。第一章简单叙述了计算机系统的组成和各部分功能，程序设计和编译执行的基本概念，PASCAL 源程序的结构以及 PASCAL 基本字符集和基本语法单位；第二章概述了 PASCAL 语言中的数据类型以及变量、表达式等基本概念，介绍了赋值语句和输入输出语句；第三章和第四章介绍了 PASCAL 语言中的流程控制语句，包括 IF 语句、REPEAT 语句、WHILE 语句、FOR 循环语句、GOTO 语句和 CASE 语句；第五章对 PASCAL 子程序——函数和过程作了较详尽的叙述；第六章引进了用户定义的数据类型——枚举类型和子界类型，并介绍了结构化数据类型——数组类型以及字符串变量；第七章详细介绍了另外二种结构化数据类型——集合和记录；第八章叙述了文件的概念，对如何建立文件、更新文件、合并文件等常规文件操作通过实例作了详细讨论并对文本文件的概念和应用作了介绍；第九章引入了指针和动态数据结构的概念，重点描述了链表和树等动态数据结构以及对它们所进行的遍历、插入、删除等常规操作。在叙述中，从实例引出概念，由浅入深，对基本语法的叙述力求准确完整，但并未给出严密的语法定义范式。在附录 A 中给出了表示 PASCAL 语法结构的语法图。附录 B 和附录 C 分别介绍了 DEC PDP-11 PASCAL V1.0 以及在 APPLE-Ⅱ 等多种微型计算机上使用的 UCSD PASCAL。

书中列举了一些颇有实用价值的例题，并在一些例题中力图使用“自顶而下”的程序设计方法，以体现出结构化程序设计的良好风格。读者通过本书各章的学习，参阅有关附录或手册即能运用上述及其他各种版本的 PASCAL 语言编写程序。

在这里，特别要感谢同济大学计算中心黄足教授和上海市计算技术研究所王金汉同志，他们审阅了书稿并提出了许多宝贵的建议和修改意见。同时深切感谢同济大学计算中心智明同志，他为本书的出版做了许多有益的工作。

再版序言

本书第1版自1985年发行以来得到了广大读者的支持。不少高等院校选为教材，取得了较好的效果。很多教师根据教学实践和工作中所积累的经验，对本书提出了不少极其宝贵的修改意见。

本版本在修订时采纳了广大读者的意见，对原书中的错漏之处作了订正；增补、调整了一些例题和习题；考虑到 PASCAL 语言作为数据结构的前修课程，在“指针和动态数据结构”一章中增加了“栈和队列”一节；因为 IBM-PC 微型计算机在我国得到了广泛应用，成为微型计算机的主流机种，故增补了一节附录 D “MS-PASCAL 对于标准 PASCAL 的一些扩充”，介绍 PASCAL 语言在 IBM-PC 机上的使用。

衷心感谢广大读者的热情支持，恳切期望广大教师和读者批评指正。

同济大学计算中心智明同志参加了本书的修订工作。

目 录

初版序言	1
再版序言	3
第一章 计算机系统及程序设计概念	1
1.1 计算机系统概述	1
1.2 源程序的编译和执行	1
1.3 PASCAL 程序结构	3
1.4 基本语法单位	4
第二章 数据、表达式、赋值语句及输入输出	7
2.1 常数及常数说明	7
2.2 变量及其类型	9
2.3 表达式及赋值语句	13
2.4 关于输入输出	17
第三章 流程控制语句(I)	23
3.1 IF 语句	23
3.2 REPEAT 语句	28
3.3 WHILE 语句	30
3.4 程序举例	32
第四章 流程控制语句(II)	39
4.1 FOR 语句	39
4.2 GOTO 语句和标号说明	43
4.3 CASE 语句	46
第五章 函数和过程	51
5.1 函数	51
5.2 过程	56
5.3 变量作用域	62
5.4 递归子程序	66
5.5 子程序作为参数	70
第六章 枚举类型、子界类型、数组类型	76
6.1 枚举类型	76
6.2 子界类型	81
6.3 数组、字符串变量	84
第七章 集合和记录	103
7.1 集合类型	103
7.2 集合的几种运算	105

7.3 记录类型	112
7.4 WITH 语句	115
7.5 记录的数组和变体记录	121
第八章 文件	131
8.1 顺序文件及其说明	131
8.2 文件的建立和读入	132
8.3 文件的更新和合并	137
8.4 文本文件	145
第九章 指针 动态数据结构	152
9.1 指针 标准过程 new	152
9.2 链 表	156
9.3 节点的删除和插入	160
9.4 栈和队列	168
9.5 双向链表	172
9.6 树	175
附录 A PASCAL 语法图	185
附录 B PDP-11 PASCAL 对于标准 PASCAL 的一些扩充	190
附录 C UCSD PASCAL 对于标准 PASCAL 的一些扩充	202
附录 D MS-PASCAL 对于标准 PASCAL 的一些扩充	211

第一章 计算机系统及程序设计概念

1.1 计算机系统概述

计算机可高速地处理数据信息，它在现代经济建设和科学技术领域中的作用越来越为人们重视。要使用计算机首先有必要概括地了解计算机。一个计算机系统包括系统硬件和系统软件。

系统硬件包括：中央处理机，主存贮器和外围设备（图 1-1）。

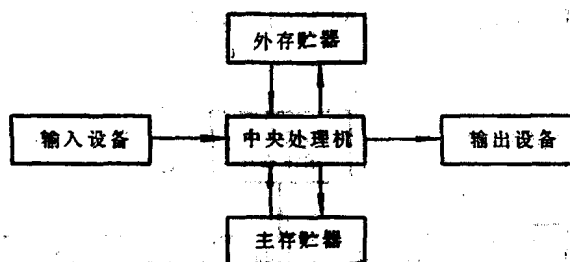


图 1-1 计算机硬件配置

中央处理机由运算器和控制器组成，它发出控制命令并按一定的顺序执行各种操作，如算术运算和逻辑运算。

主存贮器是用来存放系统的常驻信息和当前被中央处理机所处理的信息和数据的，它速度快但容量较小。外存贮器是用来存贮当前暂不处理的信息和数据的，它速度较慢但容量大。存贮器是按字节或字进行编址的。每个字节为 8 个二进位（即 8 bit）。一个字的长度对不同的计算机有不同的定义：微型或小型机一般是 8 bit 或 16 bit，中大型机一般是 32 bit，也有 48 或 64 bit。主存贮器的容量从几十千字节到几兆字节。外存贮器通常是磁盘、磁带，其容量可大到几百兆、几千兆字节。在外存贮器中，信息一般以文件形式存放。

输入输出设备是用来向计算机输入数据信息和由计算机输出结果的。常用的输入设备是纸带输入机，读卡机和终端键盘。常用的输出设备和宽行打印机及终端显示设备。

系统软件主要包括操作系统及其所支持的一些实用程序及语言编译程序。其作用是统一管理计算机系统的软硬件资源，包括输入输出设备的管理，存贮器管理，文件管理，作业调度及编译源程序等。它是用户和计算机硬件之间的接口，是整个计算机系统中的指挥部分。因此只有系统硬件而没有系统软件的计算机是不能有效地工作的。

1.2 源程序的编译和执行

对于任一特定的计算机来说，它只能识别特定的机器语言（即指令系统）。它的工作过程总是用机器指令编成的程序来处理输入数据，然后得到输出数据。如图 1-2 所示。

图 1-2 中的程序是按数据处理的要求用机器语言编写的指令串。解决各种不同的问题需

要各种不同的程序。由于机器指令繁琐难懂，对计算机的推广使用是一大障碍。PASCAL

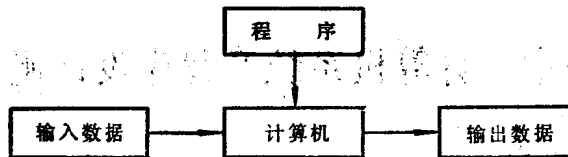


图 1-2 计算机工作过程

语言是一种高级语言，由于接近自然语言，语法规则简单清晰，故易于各类专业人员掌握使用。但计算机是不“懂得”这种高级语言的，因此用高级语言写的程序（源程序）必须有一个编译程序将它编译成机器指令编写的程序（目标程序），才能在一个特定的计算机系统中执行。翻译 PASCAL 源程序的程序称为 PASCAL 编译程序。

综上所述，在计算机上运行一个 PASCAL 源程序的过程分二步进行。

第一步：编译

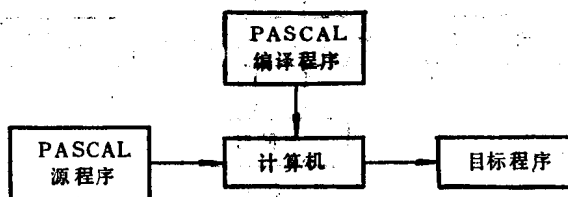


图 1-3 编译过程

第二步：运行

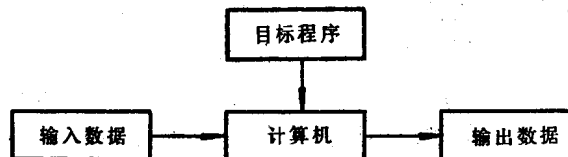


图 1-4 运行过程

由图 1-3 和图 1-4 可知，计算机被使用了二次。第一次使用时，所用的程序是 PASCAL 编译程序，输入数据是 PASCAL 源程序，得到的输出数据是目标程序。第二次使用计算机时，所用的程序是目标程序，对输入数据进行处理后得到所需的输出数据。

一般而言，在一个源程序中难免存在一些错误。这些错误可分为三类：

（一）在编译时发现的错误，称为编译错误。例如，对于一个 BEGIN 漏掉相应的 END 等。

（二）执行目标程序时发现的错误，称为运行错误。例如，在执行函数 $\text{sqrt}(x)$ 时， x 的值为负值。

（三）在编译和运行时，计算机不能发现的错误。例如，由于程序员的笔误，将 $\text{sqrt}(x)$ 写成 $\text{sqr}(x)$ ，则计算机将把 $\sqrt{x}$ 错误地计算成 x^2 。

当发生编译错误时，计算机将报告出错信息。严重的编译错误会导致不能产生目标程序。当发生运行错误时，计算机也会报告出错信息。

由于程序设计中难免发生错误，需要对程序进行调试，找出语法上和逻辑上的错误，以便进行修改。这样的过程往往要进行多次，才能得到一个较为完善的可执行程序。进行程序设计应力求使程序具有下列特点：

(1) 正确性。这要求程序员要熟悉所使用的程序设计语言，避免语法、语义上的错误，还要设计简单易行的算法来达到预期的目的。有些问题本身可能是很复杂的，这就要求程序员采用有效的程序设计方法，譬如流行的自顶向下 (TOP-DOWN) 逐步求精的方法，使复杂问题简化。

(2) 易读性。要调试一个程序就要求程序的结构清晰易读，这样就易于查找错误。将程序提供给他人阅读时，就更要求程序的易读性。

(3) 运行效率高。这里指的是运行时间短，占用存储空间小。这是人们力求达到的目标，需在不断地学习和实践中提高程序设计水平。

1.3 PASCAL 程序结构

在系统地学习 PASCAL 语言之前，我们先通过一个简单的 PASCAL 程序来初步认识完整的程序结构。虽然程序的书写格式对编译程序来说是不起作用的，但为了易于阅读，本书所有的程序实例中都用紧缩对齐格式书写。并用大写字体表示保留字，如 BEGIN, END, REPEAT, UNTIL 等。用小写字体表示标识符。

程序 minimax 读入一组整数，对它们计数，并记录其最小值和最大值。在尚未学习语法知识之前，我们可用英文词义来理解程序在做什么。

```
PROGRAM minimax (input, output);           程序首部
CONST
    min = - 32767;
    max =  32767; } 常量说明
VAR
    reading:boolean;
    number, minimum, maximum, count:integer; } 变量说明
} 说明部分
BEGIN
    reading:=true;    count:=0;
    minimum:=max; maximum:=min;
    WHILE reading DO
        BEGIN
            read(number);
            IF number=0 THEN reading:=false
                ELSE BEGIN
                    count:=count+1;
                    IF number<minimum THEN
                        minimum:=number;
                } 语句部分
```

```

        IF number > maximum THEN
            maximum := number;
        END;
    END; {WHILE}
    writeln(count, 'number read');
    writeln('The smallest was', minimum);
    writeln('The largest was', maximum)
END.

```

一个程序是由首部、说明部分和语句部分组成的。有时把说明部分和语句部分合称为程序块。一个程序总是以句点结束。首部是保留字 PROGRAM 后接程序名及程序参数表（即此程序所用到的文件列表）。说明部分包括各类说明，本程序中包括常量说明和变量说明，其主要作用是告诉编译程序一些必要的信息，以便安排存储空间及确定各变量的数据类型。语句部分是程序的执行部分，由一系列的语句组成，它们描述程序所执行的算法和动作。

由程序 minimax 可知，在首部、说明部分及语句部分之间均以分号隔开，说明部分中各说明之间及语句部分的各语句之间也均以分号隔开。程序中括在花括号中的字符串是程序的注解。注解无语法意义，编译程序将忽略注解。注解可出现在任何空格可出现的地方。在本书后面一些章节的某些较复杂的程序例中，我们还插入了一些括在花括号中的中文注释，这仅仅是为了便于阅读。该程序首先将计数器 count 置零，将计算机允许存放的最大整数 (32767) 送到存放最小值的结果变量 minimum 中去，将计算机允许存放的最小整数 -32768 送到存放最大值的结果变量 maximum 中去。在完成这些准备工作后，逐个读入数据，每读入一个数，计数器 count 加 1。若读入的数据小于 minimum 中的值或大于 maximum 中的值，则用新的数据代替 minimum 或 maximum 中的值。重复执行上述过程，直到读入一个零值为止。最后，该程序输出三个结果：总共读入的数据个数，它们中的最小值和最大值。

这一简单程序已包含了 PASCAL 程序的基本结构，从第二章开始，我们将对程序的每一组成部分进行详细的描述。

1.4 基本语法单位

上一节我们从总体上给出了一个 PASCAL 程序的基本结构。作为一种高级语言，除了规定一套严密的语法规则外，还必须规定一套基本的语法单位，以便按照语法规则将它们构造成语言的各种成份（如常数、变量、表达式、语句、说明等）。本节将叙述 PASCAL 的基本语法单位。

PASCAL 基本语法单位有两类：字和专用符号（如：+，-，*，/，:=，; 等）。它们是由 PASCAL 字符集中的字符组成的。PASCAL 字符集包括英文字母、数字及一些符号，实际上是 ASCII 字符集的一个子集。

字可分为两类：保留字和标识符。

保留字的作用是用来命名 PASCAL 语句，某些预定义的数据类型，某些操作符，说明段及程序首部的。现将所有的保留字分列如下：

AND	END	NOT	THEN
ARRAY	FILE	OF	TO
BEGIN	FOR	OR	TYPE
CASE	FUNCTION	PACKED	UNTIL
CONST	GOTO	PROCEDURE	VAR
DIV	IF	PROGRAM	WHILE
DO	IN	RECORD	WITH
DOWNT0	LABEL	REPEAT	NIL
ELSE	MOD	SET	

除这些标准 PASCAL 的保留字外，在某些实现方案中还增加了保留字 OTHERWISE 来扩充 CASE 语句的功能（详见第四章中关于 CASE 语句的叙述）。读者不必死记硬背这些保留字，通过本书的学习，读者自然会掌握每个保留字的具体用法。

标识符是以字母开头的字母与数字的序列。有两类标识符：预定义的标准标识符及用户定义的标识符。

预定义的标准标识符是用来命名标准函数，标准过程，标准类型，常数及标准文件的。现分列如下：

常数：

false, true, maxint

标准类型：

integer, boolean, real, char, text

标准文件：

input, output

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

用户定义的标识符是程序员根据需要按标识符的定义方法所定义的标识符。是用来为常数，变数，用户定义的类型、过程、函数及程序取名的。用户在选用标识符时需注意以下几个问题：

- (1) 不能与保留字同名。
- (2) 尽量避免与标准标识符同名，以免发生混淆。
- (3) 必须严格按照标识符的定义来构成标识符。
- (4) 必须遵照先定义后使用的原则，即一个用户定义的标识符必须首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。
- (5) 标准 PASCAL 对一个标识符只识别前 8 个字符。若两个标识符的前 8 个字符完全相同，则被认为是同一个标识符，如：abcdefgh1 与 abcdefgh2 是同一标识符 abcdefgh。

下面是一些合法的标识符：

a a1 nameofcar line3

下面是一些非法的标识符：

1a 不是以字母开头

begin 与保留字同名

up. to 出现非字母及非数字的字符

last name 空格不能出现在一个标识符的中间。

除注意上述五点外，为使程序清晰易读，在选用标识符时应适当取具有相应含义的英文名作为标识符。这样做既便于阅读又便于记忆。故在某些 PASCAL 实现方案中，把标识符的有效识别长度从 8 个字符扩大到更长，还允许在一个标识符中嵌入符号‘-’，‘\$’等，这样写出的标识符更清晰易读，如：type-of-name（详见附录 B）。

习 题 一

1. 计算机系统由哪几部分组成？计算机硬件系统包括哪几部分？各部分的作用是什么？
2. 什么叫机器语言，高级程序设计语言？各有什么特点？
3. PASCAL 语言的程序结构由哪几部分组成？试述各部分的构成形式和作用。
4. PASCAL 基本字符集由哪些成分构成？PASCAL 保留字是什么？举例说明保留字的作用。
5. PASCAL 语言中有哪些标准类型、标准文件、标准函数、标准过程以及标准常数。
6. PASCAL 语言中标识符是怎么构成的？下列符号中哪些是正确的 PASCAL 标识符？哪些是错误的？为什么？

A5B, 5H4, PEL.1, A31, BEGIN, ABS, x*y
A+B, \$500, G1(x), CI, VAR, xzy, E-10

第二章 数据、表达式、赋值语句及输入输出

在 PASCAL 中，一个程序所用到的每个常数、变量和函数标识符都或隐或显地与一种数据类型（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运算的一组运算符组成的。例如在字长为 2 个字节的计算机中，预定义的标准类型 integer（整型）是由 -32767 到 32767 的所有整数及能在这些值上进行运算的运算符 +、-、*、DIV、MOD 组成的。因此，类型既规定了与该类型相联系的常数、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。

常数的类型是由该常数本身隐含的，如：5 的类型是标准类型 integer，3.14159 的类型是标准类型 real。变量标识符和函数命名符的类型是在程序的说明部分显式地说明的。

PASCAL 中类型可分为三大类：标量类型、结构类型及指针类型。具体划分如下：

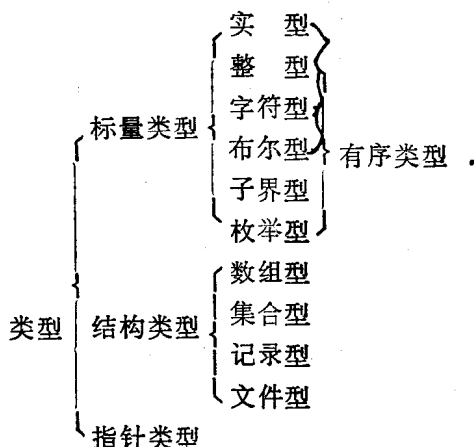


图 2-1 数据类型的分类

如图 2-1 所示，PASCAL 语言的数据类型极为丰富，共有十一种具体的类型。本章将介绍整型、实型、字符型和布尔型这四种预定义的标准类型。其它类型将在以后各章叙述。

2.1 常数及常数说明

在程序中可出现一些在程序执行前就已知的常数值，如：78，-5.63，'a'，true 等。下面叙述 4 种标准类型的常数。

（一）整型常数

整型常数是带或不带符号（+ 或 -）的数字串。如

6 -13 +7800 0

都是整数，而

7.3 -0.0 15,000

都不是整数。前两个是实数，最后一个数中的逗号是非法的。

(二) 实型常数

实数有两种表示法。一种是习惯写法，如

0.0 + 0.965 - 74.6 890

都是实数。实数的另一种写法是科学表示法（即指数形式），规则是在一个习惯写法的实数后面接字母 E 再后接一个整数。如：

6.25 E + 08	(= 6.25 × 10 ⁸)
- 51 E - 1	(= - 51 × 10 ⁻¹)
0 E 0	(= 0 × 10 ⁰)
0.6378 E 2	(= 0.6378 × 10 ²)

都是合法的实数。

必须注意的是：在小数点前后必须都有数字；在一个实数中除了正负号、数字、小数点及字母 E 外不能有其它字符；在科学表示法中，字母 E 前必须有一个习惯写法的实数，E 后必须有一个整数；一个整数也可看成是一个实数（如：890）。下面是一些非法的实数例：

0.	小数点后面没有数字
.736	小数点前面没有数字
2,765.63	出现一个逗号
. E 6	字母 E 前缺一个实数
5 E 1.3	字母 E 后接一个非整数
.6 E 3	字母 E 前的实数出错

(三) 字符型常数

一个字符型常数是括在两个单引导之间的 PASCAL 字符集中的一个字符。如 'a' 表示字母 a, ' ' 表示空格字符, '7' 表示数字 7, 等等。不同的实现方案对 PASCAL 字符集有不同的定义，但一般都包括：26 个大小写的英文字母，10 个数字及一些标点符号、运算符号等专用符号。

除了字符型常数外，在程序中还可使用字符串常数。字符串常数是括在引号中的一串 PASCAL 字符。如：

'His name is Li Min'.

字符串常数可由字符型数组来表示，用法见第六章。

(四) 布尔型常数

布尔常数只有两个：false 和 true

由于 PASCAL 中除上述四种标准类型的常数外，还有其它类型的常数，如：子界型、枚举型等常数。这些类型的常数的定义将在以后各章中讲到相应的类型时介绍。

在 PASCAL 中，还可由用户对常数取名，即用一个标识符来代表一个常数。这种常数是在程序的常数说明中定义的。如

```
CONST
    pi = 3.141593;
```

```

biggest = 1000;
smallest = - biggest;
blank = ' ';
date = 'monday, january 16';

```

此常数说明规定 pi 为实型常数 3.141593; biggest 为整型常数 1000; smallest 为整型常数 - biggest (即 -1000); blank 为字符型常数 ' '; date 为字符串常数 ' monday, january 16'。有了这些常数说明后, 在程序的后面部分需要出现这 5 个常数的地方可以用这些相应的常数标识符来代替。

由上例知, 常数说明的语法规则是:

CONST

<常数标识符> = <常数>;

⋮

<常数标识符> = <常数>;

其中保留字 CONST 表示开始一个常数说明段。每个常数标识符是程序员选择的标识符。等号后面是该标识符代表的常数值。常数说明之间用分号隔开。一个常数标识符不能与程序中其它任何标识符同名。等号后的常数可以是带有符号的前面已说明过的常数标识符 (如上例中的 - biggest)。在程序执行中不能改变常数标识符的值。

由于字长的限制, 对于一个特定的计算机, 只能表示整数的有限子集, 故 PASCAL 有一个不必说明的标准整型常数 maxint, 此常数标识符表示计算机所能接受的最大整数。以 16 位字长的机器为例, 所能接受的最大整数为

$$\text{maxint} = 2^{16-1} - 1 = 32767$$

即该计算机的整数取值范围为:

$$-\text{maxint} \leq n \leq \text{maxint}$$

有了常数说明使程序易于修改。我们可在程序中将多次出现的同一个常数 (如 1000) 在常数说明中定义为一个常数标识符 (biggest), 然后在用到此常数的地方用该常数标识符来代替。此后, 若程序员要修改此常数时 (如从 1000 改为 500), 只需修改常数说明中的那个常数就可以了 (即改为 biggest = 500)。

2.2 变量及其类型

程序中的每一项数据不是常数就是变量, 上面讨论的是常数。变量的值在程序执行中是可改变的。变量是以标识符来取名的。每个变量都具有一个相应的类型, 这样就规定了该变量的取值范围及它所能执行的运算操作。程序中所用到的变量必须首先在变量说明段中加以说明, 然后才能使用。变量说明段的语法规则是:

VAR

<变量标识符表>: <类型>;

⋮

<变量标识符表>: <类型>;

其中保留字 VAR 表示开始一个变量说明段。每个变量标识符表是一个或多个由逗号隔开的

Var: 表示开始一个变量说明段