

伊增芬
陈燕平
编著

PASGAL
PASGAL
PASGAL
PASGAL

程序设计

PASGAL
PASGAL

海洋出版社

312

2F/1

TP312
Y=F/1

PASCAL程序设计

伊增芬 陈燕平 编著

海洋出版社

1988年·北京

【内 容 简 介】

PASCAL程序设计语言以其强大的生命力得到广泛的应用。该语言小巧、直观，运行效率高，兼有其他语言之长，又便于学习和使用。本书全面介绍了PASCAL程序设计语言的编制技巧和方法，并列举了较多的例题以供读者参考。全书共分十章，包括PASCAL程序设计概念，标识符及数据类型，基本语句，构造语句，枚举类型、子界类型及集合，数组，过程及函数，记录和文件等内容，除突出选编财经方面的例题外，还安排了具有实用价值的综合例题，文字叙述简洁，内容深入浅出。

本书可供财经战线的工作者学习和使用计算机用，也可作为高等院校非计算机专业广大师生的教学参考书。

责任编辑 阎世尊

JS398/08

PASCAL程序设计

伊增芬 陈燕平 编著

海洋出版社出版（北京市复兴门外大街1号）

新华书店北京发行所发行 新华印刷厂印刷

开本：787×1092 1/16 印张：9.625 字数：150千字

1988年9月第一版 1988年9月第一次印刷

印数：1—3000

ISBN 7-5027-0216-4/Z·41

统一书号：17193·1096 定价：3.80元

前 言

电子计算机是一种能够进行各种广泛意义上的“运算”、并具有存贮功能的先进电子设备。它已经被广泛地应用于工农业生产、科学技术和社会生活的各个领域。

使用电子计算机必须要学会程序设计,即利用程序设计语言把解题的要求、运算的步骤表示出来,以便机器按照程序规定的方法和步骤完成各种数值运算和数据处理任务。

程序设计语言是人和机器“对话”的必备桥梁。程序设计语言在冲破原始的、取决于机器本身构造的、繁琐的机器语言和汇编语言的羁绊之后,发展很快,各种高级算法语言应运而生,为电子计算机的普及推广应用做出了贡献。在数以百计的各种高级算法语言中,早期最受欢迎并被广泛应用的有:主要用于科研和工程技术方面数值计算的FORTRAN、ALGOL-60、ALGOL-68以及主要用于商业数据处理的COBOL语言。

上述这些专业化的程序设计语言把程序设计与更为广泛意义上的数据处理(例如情报资料检索)这两方面的应用分离开来,而使它们自身的通用性受到限制。实际上,由于在人类的生产和社会生活中没有脱离物质运动的“纯”数存在,所以,即使在不同领域内,程序设计所加工的基本对象存在着广泛的共性。

从上述这种客观存在的共性出发,瑞士的苏黎世Eidgenossische工科大学的沃斯(Niklaus Wirth)教授于1971年发表了数据类型丰富完备、简明灵活的PASCAL语言,并在1973和1974年先后对它进行了修订和补充。PASCAL语言小巧、直观、运行效率高,查错性能好。它较BASIC语言和ALGOL、FORTRAN语言数据类型丰富、结构化性能好,表达能力强;它又较COBOL语言使用的保留字少,书写格式也较COBOL更为灵活自由、简洁。较之COBOL语言易于学习和推广。

十余年来,PASCAL程序设计语言以其强大的生命力在人类生产和社会生活各个领域内生根、开花,受到各国学者的重视,成为目前国际上最受欢迎并被广泛应用的程序设计语言之一。金兰教授在为《PASCAL程序设计及其应用》这本书所写的序言中称这种语言为“前程似锦”的高级算法语言。

目前,由于计算机基础知识的迅速普及,我们必须尽快改变前些年由于历史的原因和局限造成的在高等学校广泛讲授BASIC语言的现状。

鉴于PASCAL语言有如前所述的兼有几种算法语言之长,而又易于学习和使用的特点,所以用它来做为高等学校非计算机专业计算机课程的启蒙语言是很合适的。通过对PASCAL语言的学习,可以使学生加深对程序设计的一些基本概念的理解,在程序设计的基本方法上受到良好的训练,也为进一步学习COBOL等不同范畴内的专用算法语言打下基础。

本书在编写时,尽量避免使用其他程序设计语言知识,适合做为入门课程使用。为了帮助读者掌握用PASCAL语言编制程序的技巧和方法,书中列举了较多的例题,并在各个章节之后安排了相当数量的习题,可供教学使用。

本书在编写时除了注意选编财经方面的例题和习题以外，还在书的最后安排了具有实用价值的综合例题。

本书第一、二、三、四、六、七、十章由伊增芬编写，第五、八、九章由陈燕平编写，最后由伊增芬修改总纂。

本书所列重要例题的上机计算曾得到北京市计算中心徐承业同志的指导与帮助，在此表示感谢。

由于水平有限，欢迎广大读者批评指正。

编者

伊增芬

陈燕平

1985.12

目 录

第一章 PASCAL程序设计概念	(1)
1.1 程序设计基础	(1)
1.2 PASCAL语言的特点.....	(2)
1.3 语法图和程序基本结构	(2)
1.4 PASCAL程序的调试.....	(5)
本章习题.....	(6)
第二章 标识符及数据类型	(7)
2.1 PASCAL语言的字符集.....	(7)
2.2 标识符	(7)
2.3 常量及常量说明	(8)
2.4 变量	(10)
2.5 整数类型	(11)
2.6 实数类型	(12)
2.7 字符类型	(13)
2.8 布尔类型	(14)
2.9 PASCAL提供的标准函数.....	(16)
2.10 表达式及各种运算符的优先级别.....	(17)
本章习题.....	(19)
第三章 PASCAL程序的基本语句	(20)
3.1 赋值语句	(20)
3.2 读语句	(21)
3.3 写语句	(23)
3.4 空语句	(27)
本章习题.....	(27)
第四章 构造语句	(29)
4.1 复合语句	(29)
4.2 条件语句 (IF语句)	(30)
4.3 重复语句 (REPEAT及UNTIL语句)	(34)
4.4 分情形语句 (CASE语句)	(37)
4.5 当语句 (WHILE语句)	(41)
4.6 循环语句 (FOR语句)	(43)
本章习题.....	(48)
第五章 枚举类型、子界类型及集合	(50)
5.1 枚举类型	(50)

5.2 子界类型	(53)
5.3 集合	(56)
本章习题	(63)
第六章 数组	(64)
6.1 一维数组	(64)
6.2 多维数组	(73)
6.3 字符串、布尔数组	(76)
本章习题	(81)
第七章 过程及函数	(82)
7.1 过程	(82)
7.2 参数	(85)
7.3 函数	(90)
7.4 过程与函数的嵌套	(94)
7.5 递归	(99)
本章习题	(101)
第八章 记录 and 文件	(104)
8.1 记录	(104)
8.2 顺序文件	(111)
8.3 文本文件	(115)
8.4 输入和输出	(119)
本章习题	(120)
第九章 指针类型	(121)
9.1 指针	(121)
9.2 链表	(124)
9.3 节点的删除和插入	(128)
本章习题	(132)
第十章 小结	(133)
10.1 PASCAL 语言的特征	(133)
10.2 综合例题	(134)
附录:	
附录一 标准PASCAL标识符	(144)
一、保留字	(144)
二、标准过程及函数	(144)
三、标准常量	(146)
四、标准类型	(146)
五、标准文件	(146)
附录二 PASCAL 的运算符	(146)
附录三 ASCII 字符集	(147)
参考文献	(148)

第一章 PASCAL程序设计概念

1.1 程序设计基础

电子计算机虽然以它种种使任何传统的运算工具为之逊色的宝贵特性受到人们的钟爱，并且已经进入了人类生活的各个领域，但它必须在人的指挥下进行工作。这个指挥过程是通过程序设计来完成的。

程序设计的基本任务粗略地讲，就是：按照题目的要求，构造（或选择）适当的数学模型和算法，连同原始数据、工作步骤等“告诉”计算机。所谓程序就是那些能完成上述任务的一条条指令的总称。即：程序是一个指令序列。

程序设计语言是联系人和计算机的桥梁。程序设计语言的发展也正如人类认识其他事物一样，经历了一个由简单到复杂，由低级到高级的过程。最原始、最直接的程序设计语言是由一系列二进制代码组成的机器语言，用这种语言编写程序非常繁琐，程序的直观性很差，查错和调试很困难，而且由于不同的机器的指令的结构不同，造成这种机器语言的通用性差，用一种机器的机器语言编写的程序，拿到另外的机器上去就不能用。到了五十年代初，人们改用一些简单形象的符号代替那些二进制代码串来编写程序，这种形象的符号叫做符号语言或者汇编语言。用这种符号语言来编写程序较前方便多了，程序也直观，易于阅读和修改，大大提高了编制程序的效率，便于更多的人掌握使用计算机。但汇编语言和人类的自然语言和数学公式还相差太远，编写程序的工作量仍然很大，编写程序仍嫌复杂，不易提高编写速度，而且由于计算机结构上的差异，使得各种计算机的汇编语言也各不相同，这样，提高计算机语言的通用性问题也还是有待于解决。在机器语言和汇编语言的基础上，人们又创造了更为完善的高级程序设计语言，这种高级程序设计语言有很多种，它们很接近人类的自然语言和常用数学公式表达法，编制程序的人不需要记忆任何机器指令，仅用一些容易理解的词汇和易于掌握的语法就能方便地描述自己的解题意图。编制程序，大大地节省了编制程序的时间，较好地解决了计算机的高速运行和编写程序的低效率之间的矛盾。这类高级程序设计语言对于计算机是相对独立的，与机器的内部结构无关，只要通过各类机器对应的翻译程序就可以在各类机器上运行，解决了困扰人们多年的通用性问题，为计算机的广泛应用提供了有利条件。

目前，国内外比较通用的高级程序设计语言有几十种，应用最普遍的有BASIC，FORTRAN，ALGOL，COBOL，PASCAL以及我国自己设计的BCY语言等。

无论使用哪种高级程序设计语言，编制的计算程序，一般来说有如下共同特征：

1. 程序的工作对象是“数据”。这里所说的数据是广义的，它既包括各种数字（例如整数、实数、常数等）也包括各种有固定意义的字符串（例如图书目录，商品编号等）。当然，不同的高级程序设计语言所能处理的数据类型的多少是不同的。例如，主要用于工程和

科技运算的FORTRAN语言所处理的数据主要是数字型数据,而被称为商用语言的COBOL语言的表处理的功能较强,便于处理一些字符串型数据。在数据类型丰富方面,PASCAL语言可以称得上各种高级程序设计语言中的佼佼者。

2. 程序是一系列指令的总称,一般说来,机器是按照指令的顺序一条条地依次执行的,除非编制程序的人预先安排了变更顺序的指令。

3. 计算机的工作结果都要由程序安排它的输出,当然,输出的方式可以有多种,例如启动打印机打印,在屏幕上显示,启动绘图仪绘图等。

4. 各种程序均可按人的意图安排某些重复(例如,大量报表中的相同行或相同列上的栏目的值、就是用相同的公式计算出来的),这些重复计算,既可预先指定重复的次数,也可以用某些相关条件来限定,由机器自己决定次数。对于PASCAL语言来说,为了实现程序的结构化,通常是利用过程和函数把一个大程序分割成许多独立的程序块,特别是那些在程序要多次应用的部分更是这样。这些程序块还可以互相嵌套、互相调用。这样,主程序就可以写得非常短小精悍,层次清楚,可读性强。

1.2 PASCAL语言的特点

为了保证大型复杂的程序系统有合理的结构,以保证程序尽可能正确,并较为方便地解决程序系统的查错验证问题,程序结构化是一种先进的有效手段,为此,人们规定了一系列程序设计准则,按照这些准则设计出来的程序叫结构化程序。

PASCAL语言是按照结构程序设计的原则设计的一种算法语言。用它能够方便地书写出层次分明,结构紧凑的结构化程序。

PASCAL语言的另一个特点是它提供了较为丰富的数据类型及多种构造数据的方法,特别是能够用严谨易懂的表达方式描述非数值性的数据结构及算法。这样,学习PASCAL语言不仅可以加深对数据结构的理解,编写出逻辑性强结构合理而又简洁易读的各种程序,而且有助于提高我们的思维能力和编程技巧,培养良好的程序设计风格。

此外,由于PASCAL语言的编译程序是用这种语言本身编写出来的(叫做自编译的编译程序),所以在各种型号的计算机上建立它的编译程序比较容易,也就是说PASCAL语言很容易在各种型号的计算机上推广使用。目前,在我国广泛使用的各种计算机系统、微型机都配有PASCAL的编译程序。

由于PASCAL语言有上述各方面的优越性,在它诞生后很短的时间内,就受到欧美许多国家的重视,成为目前国际上最受欢迎、广泛流行的程序设计语言之一,被誉为“后来居上”的程序设计语言。

目前,国际上流行的PASCAL语言有许多种版本,在不同的计算机系统中运行的PASCAL语言也存在一些差异,由于本书是按照标准PASCAL语言来编写的,而且所讲述的是有关PASCAL语言的一些基本内容——程序结构、数据类型、通用语句等,所以,上述差异并不影响本书的通用性。

1.3 语法图和程序基本结构

与其他高级算法语言一样,用PASCAL语言编写计算机程序也要符合这种语言的语法规定。我们除了可以用文字叙述来表达有关的语法规定以外,还可以用更为清晰简单的语法

图来描述这些语法规定。

下面我们通过一些具体实例来说明语法图的应用。

例 1: 9, 26, 32 767 都是 PASCAL 语言的合法无符号整数, 请用一个语法图来表示它。

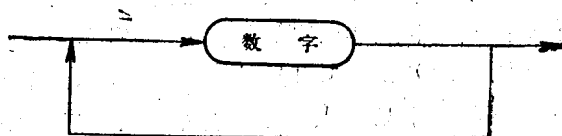


图 1-3-1 无符号整数

上面的语法图可以表示至少有一位的无符号整数, 也可以表示 PASCAL 允许范围内的任意位的无符号整数。

例 2: 有如下的一组有正负号的整数

0, +26, 78, -25, -7600

请用一个语法图来表示它们:

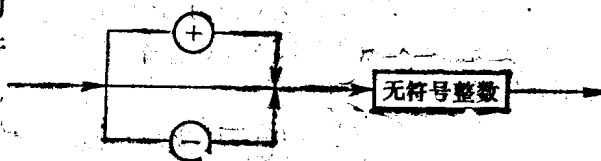


图 1-3-2 整数

从上面的语法图可以看出, 方框里的内容是由另一个语法图构成, 即: 方框表示的无符号整数是由例 1 中的语法图构成的。而语法图中的圆框中表示的是 PASCAL 语言的基本符号。

每个语法图, 从箭头的入口处开始经常有一条以上的路线, 可以沿着箭头所指的任何一条路径前进, 最后来到出口, 沿途所经过的圆框、方框依次连接起来就表示该语法图所表示的一种合法语法规则。

用语法图来描述语法规则不仅简单、直观、而且严密。在开始学习使用 PASCAL 语言编写程序时, 要经常参看语法图, 以减少编程中的语法错误。

在后面的章节中我们将通过语法图介绍有关的语法。

一个 PASCAL 程序主要由两部分组成, 前边是该程序用到的数据的描述 (说明部分), 后边是算法的描述 (执行部分)。

程序中的说明部分可以包含以下 6 个成分:

1. 标号说明 (LABEL)
2. 常量说明 (CONST)
3. 类型说明 (TYPE)
4. 变量说明 (VAR)
5. 过程说明 (PROCEDURE)
6. 函数说明 (FUNCTION)

在一个 PASCAL 程序中, 可以直接使用 PASCAL 语言定义的标准常量、标准数据类型、标准的过程和函数等等。但对于用户自己定义的标号、常量、类型、变量、过程和函数等都必须要在程序的说明部分加以说明, 才能在执行部分加以引用。

在一个 PASCAL 程序的说明部分中, 上述的 6 个成分不一定同时出现, 但有两个以上的成分出现时要按照上面介绍的顺序排列, 不得随意更动。

程序的执行部分，是用BEGIN开始，用END结尾的一系列语句组成。在BEGIN和END之间的各个语句都用分号隔开，END之后用实心句号结尾。

在PASCAL程序中的BEGIN和END之间的各个语句在书写时，允许一行写多个语句，也允许一个语句写成几行，这一点较之其他某些算法语言要灵活得多。

一个PASCAL程序除了要有上述说明部分和执行部分两个主要组成部分以外，在程序的开头还要有一个以保留字PROGRAM开始的程序首部，这个保留字是向计算机说明程序从这里开始。在PROGRAM之后要写上程序的名称，这个程序名称由用户自己定义，可以写英文，也可以写汉语拼音。程序名之后一般是书写用圆括号括起来的程序参数，最常用的程序参数是INPUT和OUTPUT。其中INPUT是表示这个程序是从键盘输入数据，OUTPUT表示程序的运行结果要送到输出设备（例如CRT）上输出。

在UCSD PASCAL和OMSI PASCAL中允许将程序名之后的参数省略。

需要注意的是以PROGRAM为首的程序首部的结尾必须书写分号，不写标点或写成句号都是错误的。

关于PASCAL程序的结构，还有一点要说明的是：在程序的首部，为了更为详细一些地标明本程序的一些特征，如程序的主要功能，使用数据的范围，编写的日期，编写人姓名

等等，都可以用注释的形式写在首部的下一行，注释的内容要用大括号“{...}”括起来。注释部分不是程序的必要成分，如果是极简单、只供练习用又没有保存价值的程序，不写注释也行。注释的内容由编写程序的人自己决定，可以用英语书写，也可以用汉语拼音书写。这部分内容机器并不执行，仅供阅读。因此，用大括号括起来的注释不仅可以放在程序的首部、做总说明用，也可以放在程序中的任何其他位置，用来增加程序的可读性。

上述PASCAL程序的基本结构可以用如左的语法图来表示：

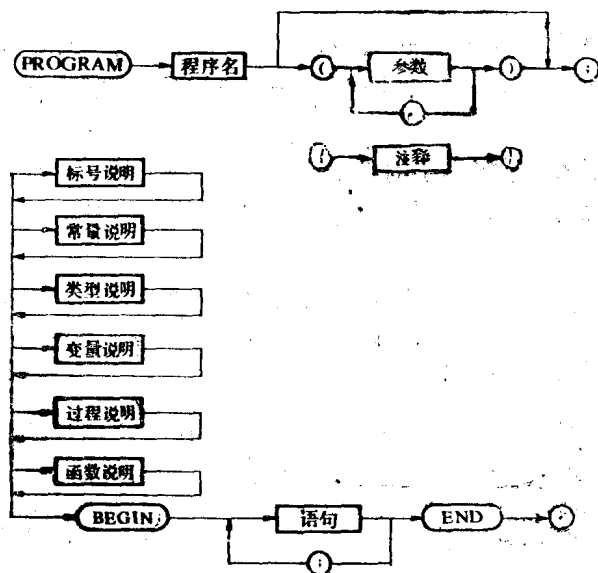


图 1-3-3 程序结构

例：计算一组存有定期一年存款的储户所得的利息及本利和。已知定期一年存款按月利率5厘7计息，存款时间按月计算。如果用汉语拼音字头BJ表示本金，LX表示利息，BLH表示本利和，YLL表示月利率，YSH表示月数，则其PASCAL程序是这样的：

```
PROGRAM LIXI CHENG XU (INPUT, OUTPUT); (首部)
CONST YLL=0.0057; (常量说明)
VAR BJ, LX, BLH: REAL; (变量说明)
    YSH: INTEGER;
```

说明部分
(数据描述)

BEGIN	(开始)	
REPEAT	(重复进行如下过程)	
READ (BJ);	(输入本金)	
READ (YSH);	(输入月数)	
LX := BJ * YEL * YSH;	(计算利息)	执行部分 (算法描述)
BLH := BJ + LX;	(计算本利和)	
WRITELN ('LX=', LX);	(输出利息)	
WRITELN ('BLH=', BLH);	(输出本利和)	
UNTIL BJ = 0	(直到本金为0)	
END,	(结束)	

首部中的程序名是汉语拼音利息程序，后边的INPUT表示数据（本金，月数）由键盘输入，OUTPUT表示运算结果由输出设备输出。

月利率是常数，在常数说明中说明。

在变量说明中，说明本金（BJ）、利息（LX）、本利和（BLH）都是可能带有小数的实数；而存款月数则恒为整数。

1.4 PASCAL程序的调试

和其他许多高级计算机语言一样，PASCAL语言也是采用编译方式，即用PASCAL语言编制的源程序送入计算机以后，先要通过PASCAL编译程序把它改编成机器能够识别的机器语言的程序——叫做目标程序，然后机器执行这个目标程序，输出运行结果。这个过程可用下面的框图来表示：

对于一个大型程序来说，即使是由熟练的工作人员来编制，错误也是难免的。查出并纠正程序中的各种错误，是一项很重要的工作，为了解决这类问题，我们通常把程序中的错误分为三类：

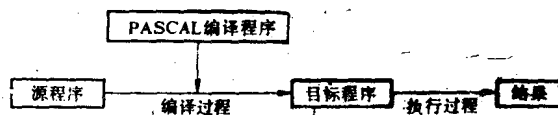


图 1-3-4 PASCAL 工作过程

第一类是语法错误，这类错误通常是由于编制程序的人对PASCAL的语法规则不熟悉或粗心造成的。例如两个语句间忘了写分号，把赋值号`:=`写成了`=`，写了BEGIN忘了写END或把某些保留字写错等等。对于这类错误，编译程序在对源程序进行编译的过程中都能检测出来。并且及时通过输出设备向操作人员报告，例如发现某个语句有这类错误，通常是先在这个语句的出错位置的下方给出一个`^^`标记，表示这里有错，然后在下一行的左端给出6个星号`*****`并用英文给出错误类型、性质的有关信息。在这之后计算机就停机中止编译过程，等待工作人员进行修正，直到错误被纠正才继续编译。全部源程序被编译完毕，这类语法错误就全都被清除了。

清除了语法错误并不意味着程序就绝对正确，运行后能够得到预期结果或正确结果了，这里还可能存在另一类错误，就是只有在程序运行过程中才能发现的错误，叫做逻辑错误。

所谓逻辑错误有许多种，例如，求某个中间计算值X的倒数时，X的取值等于0，或求

某个中间结果Y的平方根或对数时，Y的值为负数，以及程序中变量的值超过了机器允许的范围（发生了溢出）等等。对于这类运行过程中才能发现的错误，在多数计算机系统中都配有辅助调试的服务程序，做为系统软件的一个组成部分，利用这种诊断程序，就能及时检测出这类逻辑性错误，给出错误所在的位置和性质等信息。

此外，还有一类在编译和执行的过程中均不能发现的错误，例如，应该输入数值5000，而输入了数值500，产生输入数据错误，或应该写入一个计算某个数A的平方根（应写成 $\text{SQRT}(A)$ ）的符号而写成了求该数的平方（ $\text{SQR}(A)$ ），产生算法描述错误等等。对于这类错误，计算机无法检出，当然也就不会给出任何出错信息。为了解决这类问题，我们可以在程序中的某些特定位置插入一些输出语句，以便及时检测某些中间结果的取值，避免上述错误的产生。

本章习题

- 一、PASCAL程序的基本结构是怎样的？
- 二、有如下一组数字信息，请用一个语法图来表示它们：
-5, 0.62, 8.26A, -36.5B,

第二章 标识符及数据类型

2.1 PASCAL语言的字符集

PASCAL语言所使用的基本字符集包括下面三部分:

英文字母: A, B, C, ..., Z, a, b, c, ..., z;

数 字: 0, 1, 2, ..., 9;

特殊符号: 分为符号和保留字两类:

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 REPEAT SET  
THEN TO TYPE UNTIL VAR  
WHILE WITH PROGRAM RECORD  
PROCEDURE
```

这些保留字在PASCAL程序中均有固定的意义, 不允许用户再赋予它别的意义。

除了这些保留字以外, PASCAL语言还规定了一些标准标识符, 标准标识符包括INTEGER, REAL, CHAR, READ, BOOLEAN, WRITE等共40个。它们也有固定的意义, 用户可以根据需要直接使用它们。

标准标识符与保留字的区别在于, 允许用户在程序的说明部分定义它们。当然, 为了避免由于一个词的二义性造成的机器工作混乱, 一般是不对这些标准标识符重新定义的。万不得已时, 也只能对本程序中不需要按原意使用的那些标准标识符重新定义。

利用上述字符集中的字符, 按照PASCAL语言的语法规则就可以构造PASCAL符号、语句和程序。要想使程序容易阅读和检查, 恰当地选择标识符是一个很重要的措施。

2.2 标识符

标识符就是用来表示常量、变量、数据类型、过程、函数及程序名称的符号。

上节中介绍的PASCAL字符集中的保留字就是标识符的一种。除了保留字和标准标识符以外, PASCAL语言还允许用户根据需要自己定义标识符。

PASCAL语言规定: 用户自己定义的标识符必须由字母开头, 后面可以跟由数字和字

母组成的任何序列，还可以不跟任何字符，只由一个字母组成（可以认为字母后面跟的是空序列）。

上述关于用户自己定义标识符的规定可以形象地用下面的语法图来表示：

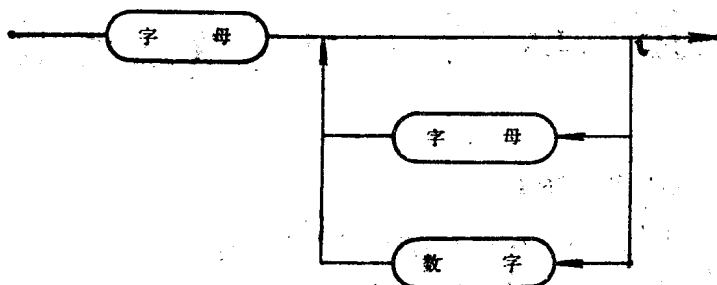


图2-2-1 标识符

按照这个构成标识符的规则，A，ABC，A₁₂，XY₄₀，FINE，NAME，good都可以做为合法的标识符，而2B，NAME-1，good bye，end等都是不合法的标识符：因为2B是非字母打头；NAME-1是出现了非字母及数字的符号；good bye是出现了空格；end是使用了保留字。

虽然，按照语法规则，构成标识符的字符的个数没有限制，但是由于一个标识符必须在一行之内写完，这就使得一个标识符充其量由80个左右的字符构成。而且在标准PASCAL语言中规定标识符中的前8个字符是有效的，即表示不同对象的标识符，应该在前8个字符有所区别才行。例如electricbell和electricclock由于前8个字符相同，在标准PASCAL程序中就无法区别了。如果改写成elecbell和elëcclock就可以区别了。

为了使编出的程序清晰易读，减少编写或修改程序时由于疏漏产生的错误，一般地说，是根据所描述的量所代表的含意用英文或汉语拼音来做标识符的，这样做便于程序的阅读和检查。只有在程序简短，使用的标识符个数不多或某个标识符在程序中使用的次数很少时，才使用单个字母做标识符。

由于标识符只能由字母和数字组成，中间不允许有其他字符，对于由两个或两个以上单词（或音阶）组成的标识符，可以用大写字母做字头，其他字符用小写的方法来书写，例如，“利息”这个量，可以写成汉语拼音LiXi，“单价”可以写成Dan Jia，应该注意，不能在两个字之间留空，把利息和单价写成LI XI和DAN JIA。

2.3 常量及常量说明

程序所加工的对象是数据，当然，这里所说的是广义的数据。数据粗略地可以分为常量和变量两类。本节讨论常量。

顾名思义，常量就是那些在程序中不改变的量，它包括那些固定不变的数（例如圆周率3.1415926）和固定不变的字符串（例如TRUE）。

PASCAL语言本身定义了三个常量，这就是机器的最大整数（与机器的型号有关PDP 11-73型机器规定为 $|X| = 32767$ ）和布尔常量TRUE及FALSE。此外，PASCAL语言还允许用户通过常量说明的形式自己定义一些标识符，用来代表一些常量。这些常量在说明时要

用单引号 ' ' 括起来。

常量定义是用保留字CONST开始的, 它可以用<标识符>=<常量>的形式定义一个或多个常数, 如果定义多个常数, 则应该用分号将它们隔开, 可以表示为:

```
CONST <标识符>=<常量>; {<标识符>=<常量>; }
```

用花括号{ }括起来的部分是可选项。

例如:

```
CONST    PI=3.14159;  
         LiXi=3.29;  
         T=TRUE;  
         CUO='YiCh'  
         G='Good';  
         KUENG='
```

都是合法的常量说明。这里要特别指出的是由于TRUE是PASCAL语言自己规定的常数, 可以直接书写在常量标识符的后面; 而Good是用户自己定义的常量, 书写时必须要用单引号 ' ' 括起来才行。' ' 表示的是空串。

此外, PASCAL语言为了方便用户, 还允许把已定义过的常量标识符加上正负号做为新的常量来定义。例如

```
CONST  
    MAX=27 500;  
    MIN=-MAX;
```

也是合法的。

上述关于常量的语法规则, 我们也可以用下面的语法图来表示:

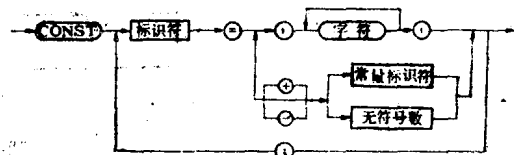


图2-3-1 常量

除了程序中只使用一次的常量以外, 在PASCAL程序中的常量一般都要在常量定义中加以说明, 因为这样做程序中用到的常量都可以用相应的常量标识符所取代, 当需要改变程序的应用范围及内部数据结构的容量时, 无需改动程序的执行部分, 只要改变常量说明中的有关常量的取值即可。

例如, 我们在1.3中介绍了一个计算定期一年的储蓄存款利息及本利和的程序, 我们在常量说明中, 对定期一年的月利率(YLL)进行了说明:

```
CONST  
    YLL=0.0057
```

现在我们想要利用这个程序为一个存有定期三年存款的储户计算利息及本利和, 就无需改动程序的执行部分, 只需把常量说明中的一年的定期月利率5厘7改为三年定期的月利率6厘6即可:

```
CONST  
    YLL=0.0066
```

在程序的数据描述部分使用常量说明还有利于增加复杂程序的可读性。例如在常量说明

部分我们给常量标识符汉语拼音“CUO”(错)定义为字符串常量“Yich”(溢出)以后,在执行条件语句或其他有关语句后要求打印。这个表示出错的字符串常量的值时,就会直接打印出“溢出”这个信息。对于解决查错的问题,有很大好处。

2.4 变量

在“数据”这个范畴内,除了常量之外,跟计算机程序关系更为密切的是那些可以不断改变它的值的量——变量。它是程序加工的主要对象。

给单个或一组可变值起的名字叫做变量的名称。从物理的角度来说,有一个(或一组)变量,就有一个(或一组)内存单元与它相对应,用来存放这个(或这组)变量的值。

根据变量所取值的性质,变量可以分成若干种类型。

按照PASCAL语言的语法要求,程序中所使用的变量都要在程序的说明部分预先加以说明,变量说明的内容是:变量名是什么,它属于哪种类型。

变量说明要以保留字VAR开始,在它后面写上变量名和变量的类型,并在它们之间加上冒号,然后用分号结尾,如果有几个变量同属于一种类型,可以用一个变量说明,只要在两个变量名之间用逗号隔开即可。例如:贷款的户数及贷款笔数均为整数类型,用INTEGER表示,贷款性质为字符类型,用CHAR来表示,则可以用一个变量说明语句来说明:

```
VAR HuShu, BiShu: INTEGER; XingZhi: CHAR;
```

上述关于变量说明的语法规则,可以形象地用下面的语法图来表示:

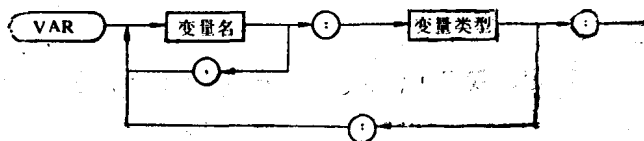


图2-4-2 变量

应当注意的是,在一个程序内部,一个变量只能属于一种确定的数据类型,不允许对变量类型做二次说明,例如,下面的变量说明

```
VAR x: INTEGER;
```

```
    x: CHAR;
```

是错误的。

PASCAL语言的一个重要特征就是数据类型丰富。它的数据类型可以分为三大类,即标量类型(也称为纯量类型或简单类型),构造类型和指针型。其中:

标量类型又可分为标准标量类型和用户自定义类型。标准标量类型数据是由PASCAL语言本身规定的一些最简单、最基本的数据类型。它又可分为整数类型、实数类型、布尔类型和字符类型四类;用户自定义类型分为枚举类型和子界类型两类。

构造类型数据是由其他类型的数据按一定规则构造而成,它包括四种类型,即数组类型,集合类型、记录类型和文件类型。

第三大类指针类型数据是用于处理动态变量的一种简单灵活的数据。

上述的数据的分类,可用下面的图来表示: