

971860

TP312

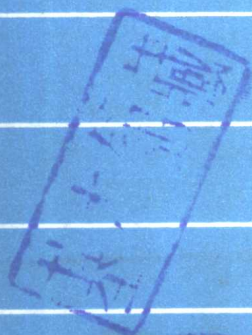
1022

第

高等学校规划教材

PASCAL 语言程序设计

贾作皆 吴秉均 赵铁群 编



煤炭工业出版社

971860

TP312
1022

TP312

1022

高等学校规划教材

PASCAL语言程序设计

贾作皆 吴秉均 赵铁群 编

煤炭工业出版社

(京)新登字042号

内 容 提 要

本书从基本概念出发,由浅入深,完整的介绍了标准PASCAL语言的内容和TURBO PASCAL语言扩展的新功能,结合具体实例,以结构化程序设计思想为指导,重点介绍语言,兼顾算法。

在介绍标准PASCAL语言的同时,引入TURBO PASCAL语言的新功能,是本教材的一个特点,目前这类教材尚少。

书中配有丰富的习题和例题,其中有些有一定的难度。例题讲解贯彻由算法到程序,自顶向下逐步求精的原则,这样能使學生熟练地掌握一种结构程序设计语言及实用程序的开发技术和规范,培养学生良好的编程风格,并能使學生在用高级语言开发系统程序方面得到良好的训练。

本书为计算机应用专业本科或专科学生的教材,同时也适用于非计算机专业的研究生和本科生,并且便于自学。

高 等 学 校 规 划 教 材 PASCAL 语 言 程 序 设 计

贾作皆 吴秉均 赵铁群 编

责任编辑:胡 玉 雁

煤炭工业出版社 出版

(北京安定门外和平里北街21号)

煤炭工业出版社印刷厂 印刷

新华书店北京发行所 发行

开本787×192mm¹/₁₆ 印张14⁵/₈

字数345千字 印数1—4,665

1993年8月第1版 1993年8月第1次印刷

ISBN 7-5020-0799-7/TD·738

书号 3567 A0227 定价 6.75元

前 言

为满足煤炭系统高等院校计算机语言教学的需要,在中国统配煤矿总公司教材编辑室的组织领导下,由山东矿业学院,中国矿业大学和西安矿业学院合作编写了这本《PASCAL语言程序设计》,作为计算机应用专业本科和专科或非计算机专业的本科和研究生的教材。

PASCAL是一种结构化程序设计语言,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,也能用于系统程序的设计。由于上述优点,它已成为我国最流行的一种程序设计语言。各类学校有关计算机专业都开设PASCAL语言课程。

本教材是以PASCAL标准文本为依据,吸收了国内外一些PASCAL语言教材的优点,总结了多年来在PASCAL语言教学中的经验,并引入了TURBO PASCAL新功能的基础上写成的。在编写时力求作到简明扼要,由浅入深地、完整地、系统地介绍PASCAL语言标准文本的全部内容以及TURBO PASCAL在某些方面的功能扩充,并在第十章对TURBO PASCAL的嵌入技术、覆盖技术和绘图功能加以重点介绍。全书贯穿了结构程序设计的思想方法,并在一些例题中力图使用自顶向下逐步求精的程序设计方法,以体现出结构化程序设计的良好风格,并可以使读者在这方面受到良好的训练。

在教材编写过程中得到山东矿业学院、中国矿业大学,西安矿业学院的领导和中国统配煤矿总公司教材编辑室的领导及编辑同志的大力支持和帮助,山东矿业学院软件教研室,中国矿业大学计算机教研室,西安矿业学院及阜新矿业学院电气工程系以及煤炭系统各院校的老师提出了宝贵意见,在此,一并表示衷心感谢。

全书由贾作皆、吴秉均、赵轶群编写,贾作皆任主编。其中赵轶群编写第一章,贾作皆编写第二、三、四、五、六章及附录部分,吴秉均编写第七、八、九、十章。由于我们水平有限,疏漏谬误在所难免,恳请读者批评指正。

编 者

1992年5月

目 录

第一章 概述	1
第一节 PASCAL语言的发展史	1
第二节 PASCAL语言的特点	2
第三节 PASCAL语言的程序结构	3
第四节 PASCAL语言的基本元素	5
第五节 PASCAL语言的语法图	7
习题	8
第二章 数据和数据类型、常量和变量	9
第一节 数据和数据类型	9
第二节 标准数据类型	9
第三节 常量和变量	11
第四节 标准函数	12
第五节 表达式	13
习题	16
第三章 简单的程序设计	18
第一节 赋值语句	18
第二节 数据的输入	19
第三节 数据的输出	20
第四节 程序举例	22
习题	25
第四章 控制语句	28
第一节 复合语句	28
第二节 条件语句	29
第三节 循环语句	36
第四节 程序举例	50
习题	53
第五章 过程与函数	55
第一节 过程	55
第二节 函数	60
第三节 值参数和变量参数	63
第四节 标识符的作用域	67
第五节 过程和函数的嵌套与递归	69
第六节 过程参数和函数参数	76
第七节 程序举例	78
习题	82
第六章 枚举与子界类型	86
第一节 枚举类型	86

第二节 子界类型	92
习题	95
第七章 构造数据类型	97
第一节 数组类型	97
第二节 集合类型	113
第三节 记录类型	118
第四节 程序举例	128
习题	136
第八章 文件	138
第一节 文件类型定义及文件变量的说明	138
第二节 文件操作	139
第三节 正文文件	148
第四节 无类型文件	152
第五节 程序举例	153
习题	167
第九章 动态数据	168
第一节 指针类型、指针变量及标准过程NEW和DISPOSE	168
第二节 链表	171
第三节 二叉树	189
第四节 程序举例	193
习题	199
第十章 TURBO PASCAL的嵌入覆盖及绘图技术	201
第一节 嵌入技术	201
第二节 覆盖技术	202
第三节 图形技术	207
习题	219
附 录	220
一、PASCAL语法图	220
二、保留关键字	224
三、标准标识符	224
四、ASCⅡ字符集	225
五、上机实习安排	225
参考文献	227

第一章 概 述

PASCAL语言问世以来,对程序设计语言的发展产生了巨大的影响,它已成为国内外流行最广泛的程序设计语言。

本章将对PASCAL语言产生的背景和发展情况、语言的基本元素和特点,程序的基本结构等方面加以介绍,使读者对PASCAL程序有一个初步认识。

第一节 PASCAL语言的发展史

一、软件危机与结构程序设计

随着第一台计算机的问世也就产生了人如何与计算机联系的问题。一开始人们直接用机器语言编写程序来解决这一问题。机器语言难记、难编程、效率低,但它是机器唯一能直接接受的语言。很自然地,在使用机器语言时、人们用助记符来帮助记忆和编程,导致汇编语言的形成。汇编语言的意义还不仅仅是增加了一些助记符,而在于它开辟了人机联系的新途径,即分两步来解决人机联系问题。首先用人易于接受但计算机不能直接接受的语言来编程,然后由计算机将这种程序自动翻译成计算机指令,而为计算机所接受。汇编语言由于还紧密依赖于计算机本身的特点,因而仍属于低级语言,但它体现的分两步走的思想却为以后出现的各种高级语言所发展。常见的有FORTRAN、ALGOL60、COBOL、BASIC等等。这些高级语言所写的程序都是要经过解释或编译才能变为可执行的机器指令序列。越是高级的语言,越是远离计算机的具体特征,越是接近自然语言。这正是高级语言有别于机器语言、汇编语言这些低级语言的地方。这些高级语言对于计算机各个应用领域的发展起了很大的推动作用,相应地软件开发也由早期少数人开发上百条语句的自用性程序发展到由几十人、上百人开发的大型软件系统。程序的正确性也从无需证明发展到无法证明的地步。到了60年代这一问题终于以“软件危机”的形式爆发了出来,也就是说随着大型软件的发展,程序的正确性和软件系统的可靠性越来越难以保证,软件开发生产率远远满足不了对于新系统的要求。问题是由1962年美国金星探测卫星水手二号发射失败引起的。经多方测试,发射水手一号不出错的程序在发射水手二号时出了问题,无法通过测试证明软件是正确的。于是许多计算机科学家转到程序正确性证明的研究,探索开发、生产软件的新方法,研究更加接近软件本质的理论。著名的荷兰科学家图林奖获得者E. W. Dijkstra提出了“GOTO语句是有害的。”这一著名论断引起了一场大争论。通过争论,人们认识到滥用“GOTO”语句会使程序无法分析、难于测试、不易修改。同时还对诸如全程变量的数据耦合效应、函数调用的副作用、类型隐含说明和自动转换造成的难于控制等潜在的不安因素进行了深入研究。加深了人们对软件本质、程序语言的理解,逐渐形成了以Dijkstra提出的“结构程序设计”为核心的软件工程学。结构程序设计就是抛弃传统的、手工式的开发软件的方法,寻求以工业化方式可靠地生产软件的方法论。人们积极研制反映新理论的程序设计语言,PASCAL语言是第一个实现并得到普遍承认的结构程序设计语言。

二、PASCAL语言的产生及发展

1964年, ALGOL工作组成员, 瑞士苏黎世联邦工业大学的Niklaus Wirth教授改进了ALGOL60, 提出了ALGOL W, 是一次追求结构化程序设计的有益尝试。ALGOL W由于结构简洁、完美而又结构化, 使其成为当时软件教程中的示例语言。接着 Wirth教授参加了从事开发新一代ALGOL的研究委员会——ALGOL68工作组。他们集当时语言和软件技术之大成, 取得了一些突破, 但因学究气太重, 程序员难于掌握而没有得到推广。Wirth教授经过这一次成功、一次失败之后, 回到瑞士潜心研究PASCAL语言。从一开始研制, Wirth教授就本着“简单、有效、可靠”的原则来设计完全符合结构化程序设计的语言。1971年PASCAL正式问世, 它较ALGOL60增加了非数值计算和数据处理方面的功能, 更重要的是PASCAL有完全结构化的控制结构、具有丰富的数据类型、简明通用的语句、清晰的程序结构、书写格式自由、直观易懂, 是程序设计的示范语言。在人们为摆脱软件危机而对结构化程序设计寄予极大希望的时候, PASCAL应运而生, 得到了很快的普及。

PASCAL语言本身也在不断发展。由于PASCAL只能编写顺序程序, 因此作为系统程序设计语言, 只能用来编写编译程序这类顺序型系统软件。1975年丹麦学者 B. Hanson开发了并发PASCAL, 将PASCAL向并发方面作了扩充。在控制方面提出了适应并发控制的进程类和管程的概念。通过管理实现进程通信, 有较强的静态类型检查, 可查出静态“死锁”, 获得了又一次成功。PASCAL语言还由于没有可分别编译的模块结构而在编写大型程序时造成困难。1975年Wirth教授开始开发Modula语言, 1977年改进为Modula-2。该语言具有可分别编译的模块结构, 增加了同步进程机制以支持并发程序设计, 取消了“GOTO”语句。但该语言与美国国防部耗费巨大人力物力研制的Ada语言非常近似, 而没有得到推广。另外一个更重要的发展是: 随着应用的发展, PASCAL的功能越来越强。诸如对系统功能的直接调用、直接的输入输出操作、中断的处理、窗口功能和图形功能等都使PASCAL语言的功能得到了很大的扩展。目前已推出了各种功能强大的版本, 如Microsoft公司的MS系列(V4.0), Borland公司的TURBO系列(V4.0、V5.0和V5.5)等。语言与其它语言的混合编程也得到愈来愈多的重视, 最近推出的PASCAL编译系统一般都具有混合编程的能力和接口, 使得与其它多种语言之间的交往变得更加容易。因此本书在讲解PASCAL语言本身的各种数据结构、语法和编程方法时(前八章), 以标准PASCAL为主, 而在讲解实用技术时, 以TURBO PASCAL为主(最后两章)。以便由浅到深, 完整地介绍标准PASCAL语言的内容, 又能介绍PASCAL语言最新扩展的功能。

第二节 PASCAL语言的特点

PASCAL语言是人们在解决软件危机的过程中, 为体现结构程序设计思想而设计的一种程序设计语言。它是70年代最有影响和最重要的一种程序设计语言。它具有以下特点。

一、结构化程序设计语言

PASCAL语言是第一个按照Dijkstra提出的结构程序设计思想设计的语言。它遵循结构程序设计所规定的一套程序设计准则, 反映了结构程序设计的要求和限制; 采用自顶向下、逐步求精的设计方法, 使所写程序具有结构合理、正确性高、易检验的特点。它虽保留不符合结构程序设计思想的“GOTO”语句, 但限制“GOTO”语句的使用。它只使用符合结构程序设计要求的三种基本控制结构来构造任何顺序程序。这些基本控制结构都是

单入口、单出口的，其中包括PASCAL扩展的三种控制语句。这样程序结构清晰、易读、易改，易于进行正确性检验。作为逐步求精的结果，程序结构是分层的。这不仅使程序结构清楚，而且使修改程序产生的影响局部化，从而使程序具有易改的特点。这些都体现了结构化程序设计的原则。

二、丰富的用户自定义数据类型

PASCAL语言不仅能解决数值计算问题，而且为更清楚、更精确地描述非数值应用问题和解决实际问题，提供了由用户按一定语法规则自己定义的一些数据类型。这些新的数据类型一旦被定义后，就如一些常见的数据类型一样，在程序中可以引用，便于编程。这些新的、丰富的自定义数据类型，大大增强了PASCAL的功能，充分显示了它的优点。

三、递归性

PASCAL语言中的函数或过程除与其它高级语言一样可以调用其它过程或函数外，还可以自己调用自己，称为递归调用。这种递归性在化复杂问题为一系列简单问题的递推时非常有用。如在计算阶乘、级数、幂指数运算等特别有效，可以大大简化程序、提高效率。

四、动态存储器分配

在PASCAL语言中有很丰富的各种简单类型和构造类型的数据，这些数据一旦加以说明，编译时即给这些变量分配存储器，这种分配在程序的执行过程中不能加以改变，称为静态存储器分配。为了对一些事先不能确定大小，又经常需要插入或删除其中一些数据的情况提供方便，PASCAL提供了一种新的数据类型，即指针类型。可以在程序的执行中动态地建立变量，也就是可以实现存储器的动态分配，方便地、高效地增加或删除若干数据。

由于PASCAL语言具有以上特点，给按照结构化程序设计思想编写程序提供了很多方便，因而得到广泛的应用。国内外大多数高等院校都把它作为高级程序设计语言的示范语言。在我国将PASCAL选作第一语言的院校也越来越多。学习它不仅仅是学习一门程序设计语言，更重要的是能学习到现代程序设计方法——结构程序设计方法。这对于尽快地掌握正确的程序设计方法与技巧、培养程序设计能力及良好的程序设计风格都有很大裨益的。

本书以介绍标准PASCAL语言为主，但是由于PC机的广泛应用，语言也形成了众多的PC机版本。针对主流微机IBM PC系列也已形成了Microsoft公司的MS PASCAL语言（V3.32或V4.0）、Borland公司的TURBO PASCAL语言（V3.01, V4.0, V5.0或V5.5）。为便于实际应用，本书也以一定篇幅介绍TURBO PASCAL语言。

第三节 PASCAL语言的程序结构

一个PASCAL程序由程序首部后接分程序构成。分程序中包括程序说明部分和语句部分。以一个简单的PASCAL程序来说明它的组成（程序见下页）。

这一简单程序是用来求一长方形的面积。程序中的英文字母也可用小写或大小写混合使用，在本书中，一般都采用大写。从这个程序可以看出程序包括程序首部与程序体两部分。在TURBO PASCAL中程序的首部也可以省略。

```

PROGRAM EXAMPLE(INPUT,OUTPUT);           }程序首部
VAR A,B,AREA,INTEGER;                    }变量说明
BEGIN                                     }
    WRITE('PLEASE INPUT TWO SIDES OF A RECTANGLE'); }语句部分
    READ(A,B); AREA:= A * B;
    WRITE('THE AREA OF THIS IS',AREA)
END.

```

一、程序首部

程序首部以保留字 PROGRAM 开头，紧跟程序名 EXAMPLE，在圆括弧内列出的 INPUT, OUTPUT 是程序参数，用来表示程序与外界有输入与输出的联系。即从标准文件 INPUT 读入数据，把结果送到标准文件 OUTPUT 中。文件的标识符 INPUT 和 OUTPUT 与具体运行环境有关。在微型计算机中 INPUT 可以是键盘，而 OUTPUT 是显示器屏幕；而在微机分时多用户系统中 INPUT 和 OUTPUT 可以是用户目录中已取名的文件。

二、程序体

程序体由程序说明部分和语句部分组成，分别完成对数据的描述和要实现的功能、动作的描述。构成了程序的主要部分。

1. 程序说明部分

PASCAL 语言中除一些标准常量、标准类型、标准过程和标准函数可以不加说明而直接引用外，用到的其它常量、变量、用户自定义的数据类型、过程和函数等均需在程序说明部分加以定义和说明。说明部分包括五个部分，必须严格按照规定顺序书写如下：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。

在一个程序说明部分中，不一定全部包含以上五个定义、说明成分，这要就具体情况而定。如所举例子中没有出现标号、常量、类型、过程和函数说明，仅有变量说明部分。

2. 语句部分

程序中语句部分是对要实现的功能需做的操作的描述，它是程序的执行部分，由一系列语句组成。每一个语句完成一定的动作，实现一定的功能。每个语句之间必须以分号“;”隔开。语句部分以保留字 BEGIN 开始，以 END 结束。在语句部分结尾的 END 后必须加上句号（.），以示整个程序结束。

作为一个简单的例子，以上程序是计算一个长方形的面积。第一行是程序首部。给出程序名 EXAMPLE，并用程序参数 INPUT, OUTPUT 指明了标准输入、输出文件，可以从键盘输入数据，从显示器屏幕输出。第二行是变量说明部分。说明了 A, B, 和 AREA 三个整数变量，其它说明部分没有出现。由保留字 BEGIN 和 END 之间的四个语句构成程序的语句部分。每个句子描述一个动作。第一个动作在显示器屏幕上显示“PLEASE INPUT TWO SIDES OF RECTANGLE”，提醒使用者输入一个矩形的两条边长；第二个动作读入两个边长，分别送到变量 A 和 B 中；第三个动作是求矩形的面积并送入变量 AREA 中；最后一个动作是输出该矩形的面积；最后程序以句点（.）结束。

实际应用的 PASCAL 程序当然比以上的例子要复杂得多，但其语法结构却总是相同的。一个 PASCAL 程序可用如图 1-1 所示的程序结构语法规则图来表示。

在图 1-1 所示语法图中，带箭头的有向路径表示各个语法成分之间的连接关系。方框

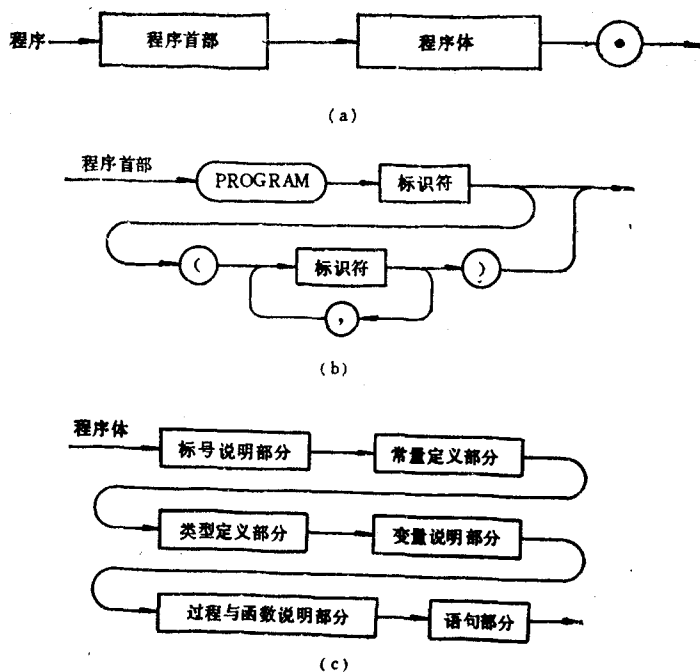


图 1-1 程序结构语法规则

内表示的语法成分是需要另外给出定义的。如程序首部、程序体分别表示程序的两个组成部分，都需要另外定义。要了解程序首部、分程序的语法结构可查阅说明这一语法成份的语法图。如图1-1所示。带圆弧的方框内写的是保留字；而在圆形框内是PASCAL的操作符或规定的其它特殊符号。这样从图1-1就可清楚地看出程序首部由保留字PROGRAM开始，紧跟一标识符，这里标识符就是程序名，再接着以一对圆括弧（运算符）括起来的表示程序参数的参数表。括弧内的标识符即程序的参数名，为INPUT, OUTPUT等。显然从图中还可看出参数可以是多个，每个参数名之间以逗号(,)分隔。从图1-1所示程序体的语法图可以清楚地看出说明部分应遵循的次序，也可以看出它可以缺少一些说明，甚至没有任何说明而直接达到语句部分。当然该图方框中的语法成分还需要进一步用语法图加以说明。最后，还应指出PASCAL语言的注释与其它语言的注释一样，可以写在程序的任何位置，因为它并不参与程序的运行，对程序的执行毫无影响。

第四节 PASCAL语言的基本元素

与其它程序设计语言一样，PASCAL也有它自己一套基本元素。这些基本元素可按照一定的语法规则构成PASCAL语句，进而组成PASCAL程序。基本元素可包括：基本字符、标识符、分隔符三类。

一、基本字符

基本字符包括字母、数字、特殊符号三类。

1) 字母：A、B、C……，X、Y、Z；a、b、c……x、y、z。PASCAL语言规定，一个字母可以是大写，也可以是小写，都被看成相同的字符。这一规定使我们在

书写PASCAL程序时可以比较自由地使用大、小写字母，使它比较接近自然语言的规定。但是需注意，在字符串中，大、小写字母是不能互换的，它们是不同的字符。

2) 数字：0、1、……9。

3) 特殊字符：由运算符、保留字两部分组成。

运算符：+、-、*、/、=、< >、<、<=、>、>=、()、[]、{}、:、:=、
..、,、;、..、'。

保留字：由字母拼成具有固定意义的单独符号。共有35个，它们是用来命名PASCAL语句、某些预定义的数据类型、某些操作符、说明部分及程序首部的。因此保留字绝不能移作它用。这35个保留字是：

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

除这些标准PASCAL的保留字外，有的方案还增加保留字OTHERWISE来扩充CASE语句的功能。

二、标识符

标识符就是用来标示、识别名称的符号。这一名称可以是程序名、常量名、变量名、过程名、函数名和类型名等等。标识符必须以字母开头，后面可以是字母、数字的任意组合。标识符可以分为标准标识符和用户定义的标识符两种。

1. 标准标识符

标准标识符是PASCAL语言预先给标准函数、标准过程、标准类型、标准常量及标准文件命名用的标识符，在程序中可直接使用。标准标识符有：

标准常量：FALSE, TURE, MAXINT。

标准类型：INTEGER, REAL, CHAR, TEXT, BOOLEAN。

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：READ READLN WRITELN WRITE PAGE PACK UNPACK
RESET REWRITE NEW DISPOSE GET PUT

2. 用户定义的标识符

用户定义的标识符是用来为常数、变量、用户定义的类型、过程、函数及程序等取名的。在取名时要尽量能反映被表示的对象的意义或功能并注意以下原则：

1) 绝不能与保留字同名。

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，即必须以字母开头，后跟以字母和数字的组合。

4) 标准PASCAL对一个标识符只识别前8个字符。因此须注意若两个标识符的前8

个字符相同，后几个字符不同，则仍认为是同一个标识符。如LARGESTTWO和LARGESTTHREE是同一标识符LARGESTT。

5) 标识符必须先定义后使用，即用户定义的标识符必须首先在程序的说明部分加以说明，才能在语句部分出现。

作为合法标识符的例子，可以举一些如X, Y, AREA, AVERAGE, BIG, SMALL, SUM等等。这些标识符都是易于理解、易于记忆，分别表示数字中的变量X、Y以及面积、平均值、大值、小值、和等等。

不合法的标识符如：BIG-ONE（含非字母、非数字的字符），1X（以数字开头），BEGIN（与保留字同名），CHAR SET（出现空格）。

3. 分隔符

空格符、行结束符和注释（即用{ }括起来的一串字符）称为分隔符。在PASCAL程序中任何两个相邻的保留字、标识符，数或两个相邻的符号之间都必须有一个或多个分隔符使编译程序能识别它们。为便于阅读，也需要如自然语言一样灵活地使用这些分隔符。

至此，PASCAL语言的发展史、它的主要特点，程序结构、基本元素等都介绍过了，最后介绍PASCAL语言的语法图。

第五节 PASCAL语言的语法图

PASCAL语言的一个特点是有非常精确的语法图。对PASCAL语言涉及的每一个语法概念、程序单元都通过语法图给以直观、精确、唯一性的解释，为学习PASCAL语言提供了极大的方便。语法图不仅给出准确无误的说明，而且也可以排除误解。因此要想学好PASCAL语言，必须认真地、反复地研究语法图，直到真正理解。语法图将贯穿本书，因此有必要较详细地阐述。

语法图由三部分组成：有向路径、方框（含带圆弧的方框）、圆框。

有向路径指明了各个语法成分之间的连接及顺序关系；方框内表示的是需由其它语法图定义的语法成分；带圆弧的方框内是PASCAL的保留字或无须进一步定义的语法成分；圆形框内是PASCAL的运算符和规定的其它特殊符号，为：，：，（，）等等。

例如标识符的语法规定是：必须以字母开头，后面跟以字母、数字的组合、用语法规则图表示如图1-2所示。通过该语法图可以很方便地检测出哪些标识符正确，哪些不正确。

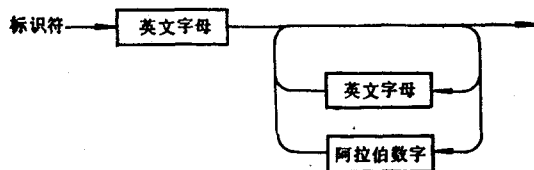


图 1-2 标识符的语法规则

图中方框内的字母、数字还须由其它语法定义，如图1-3、图1-4所示。于是标识符的语法就被完全清楚地表示出来，例如IBM PC是否符合标识符的语法，从语法图很容易看出是不合法的，因为语法图中不包含特殊字符-。再如x, x100显然是符合标识符语法的，而x(1)显然又不符合。本书在今后凡是遇到新的语法成分，都要用语法图来作唯一性的精确解释。

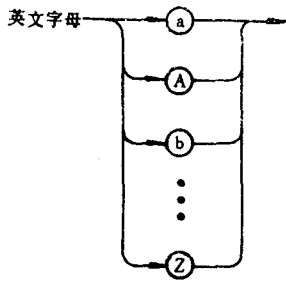


图 1-3 字母语法规则

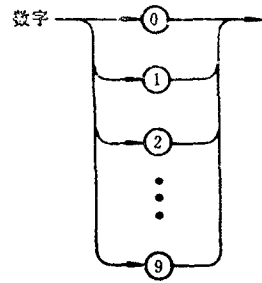


图 1-4 数字语法规则

习 题

1-1 PASCAL语言主要有哪些特点?

1-2 PASCAL语言的程序结构由几部分组成? 它与你所熟悉的算法语言的程序结构有什么不同?

1-3 PASCAL语言中规定的用户使用的标识符构成原则是什么? 并指出下列字符串中, 哪些是不合法的? 为什么?

ABC, ANNDB, FOR-TO, SIN, ST53D,

PL/I, C-11, APPLE, 4XY, A(I), PASCAL,

1992Y, DJS-130, BEGIN, PROGRAM.

第二章 数据和数据类型、常量和变量

构成PASCAL程序的最基本的成分，包括4种标准数据类型：常量和变量，标准函数和PASCAL语言中使用的表达式。这些都是学习PASCAL程序设计的基础。

第一节 数据和数据类型

数据是程序设计中的一个重要内容。一个程序在执行过程中一般都要提供初始数据和输出数值结果。PASCAL语言具有丰富的数据类型，按其特点可分为简单数据类型、构造数据类型和指针类型三大类，如图2-1所示。

一、简单数据类型

这一数据类型构造简单，它包括常用的4种标准函数类型，即整数类型、实数类型、字符类型、布尔类型。这4种类型在本章下一节将作详细介绍。简单数据类型还包括用户自定义的枚举和子界类型（见第六章）。

二、构造数据类型

这种数据类型的构造比较复杂，一般是由其它数据类型按照一定的规则构造而成，它包括常用的数组、集合、记录和文件4种类型，这些类型将在第七、八章中介绍。

三、指针类型

这是一种使用灵活的数据类型，主要用于解决动态数据的建立、删除和使用等问题，将在第九章介绍。

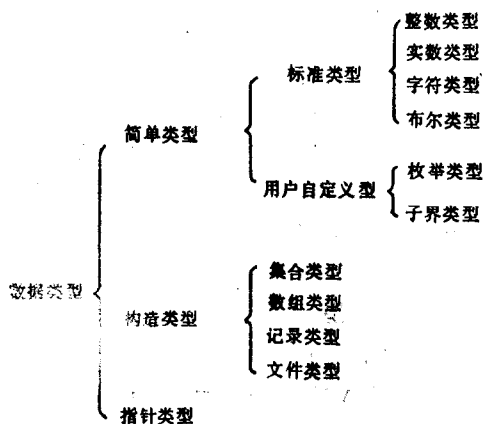


图 2-1 PASCAL数据类型

第二节 标准数据类型

一、整数类型

整数类型数据包括正整数，零整数和负整数。在数学中，整数是一个无限集合，但任何计算机只能表示整数的一个子集。在PASCAL语言中，预先定义一个标准标识符MAXINT，它是一个常数标识符，表示在计算机系统中整数的最大值。由于各类计算机的字长不同，MAXINT的值域也不同，一般MAXINT的最大值为32767至281474976710655之间。

一个整数，一般只能由正号或负号和数字序列组成，在序列中不允许出现任何其它符号。当为正整数时，正号“+”可以不写。

例如：123；253；-253；-1；1；0等都是合法的整数，而21.0；35A；±25；1.0等都是非法的。

在一个确定的计算机系统中, MAXINT的值是确定的。所以在具体计算机系统中使用的整数N的范围必定是

$$- \text{MAXINT} \leq N \leq \text{MAXINT}$$

任何超出这个范围的整数都是不允许的。

二、实数类型

实数是由正号或负号、小数点或E(指数分隔符)及数字字符组成。

实数的表示形式一般有 $\pm a.b$ 和 $\pm a.bE \pm c$ 两种,即小数形式和指数形式。其中“ $\pm$ ”号只能为“+”号或“-”号之一,且“+”号可以省略,a, b, c为无符号整数。

在使用实数时要注意以下几点:

1) 不同的计算机系统所能表示的实数范围不同,在一般16位计算机系统中,实数X的范围约为 $1.0 \times 10^{-38} \leq 1X \leq 1.0 \times 10^{+38}$,且有效数字一般为7位。

2) 用小数形式表示实数时,小数部分或整数部分为零时都不得省略。例如5.0和0.8不能写成5.和.8。

3) 用指数形式表示实数时,E的前边必须是整数或者小数形式的实数,E的后边必须是一个整数。如:E5, 3E, 1.2E2.5都是错误的,2E3, 1.0E2, 1.5E-2都是正确的。

4) 要注意整数和实数的区别。如100是整数,若写成1.0E2或100.0则为实数。运算速度整数比实数快得多,凡能用整数解决的问题就尽量使用整数计算。但是实际上由于整数的范围较小,仅用整数计算的用途是很有限的。

三、字符类型

字符类型的所有可能取值是全部可显示字符。空格符也是可显示字符,因而空格属于字符类型。在PASCAL语言中,用两个单引号把字符集中的一个字符括起来,表示字符类型数据。例如'A', 'B', ' ', '*'等,分别表示字符A, B, 空格和*。“空格”是一个重要字符,它表示在该位置留出一个空格。在本节中,在表示空格字符的地方用“ ”代替,但实际上并不存在符号“ ”。

在PASCAL语言中,字符''表示单引号字符"'"。若用'''表示则是错误的。

字符类型的数据只能是一个字符,不能是一串字符。字符串属于构造型数据,将在第七章介绍。

四、布尔类型

布尔类型的数据共有两个:即TRUE和FALSE。它们分别表示逻辑判断结果是真(TRUE)或假(FALSE)。

例如: $10 > 1$ 为TRUE而 $10 < 1$ 为FALSE。

本节介绍了PASCAL语言中的4种标准数据类型,这4种数据中,不但整数和实数是有序的,字符数据,逻辑数据也是有序的。

在PASCAL语言中,所使用的字符集都是按规定排好的,每个字符都有一个序号,字符序号的大小代表着字符的大小,所以所有字符都可以比较大小。如:

'0' < '1' < ... < '9'

'A' < 'B' < ... < 'Z'

逻辑数据的两个值TRUE和FALSE顺序关系是: FALSE < TRUE。

第三节 常量和变量

一、常量

在程序执行中保持不变的量称为常量。在PASCAL语言中，常量在程序里出现的形式可以有两种，一种是以本身面值形式出现，如：

$N := N + 1$; $PI := 3.14159$;

其中1和3.14159都是常量，这种常量称为直接常量。

另一种是用标识符命名的常量，其目的一般是为了增加程序的可读性，为常量命名的句法称为常量定义。常量定义的语法规则如图2-2所示。

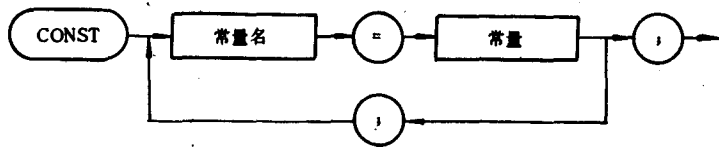


图 2-2 常量定义语法规则

图中CONST为常量定义的保留关键字。

常量名由用户自己定义，但必须符合用户标识符的规定。常量名要和常量一一对应，常量可以是任何一种标准数据类型的数据。例如：

CONST

TEN = 10;

PI = 3.14159;

BLANK = ' ';

T = TURE;

上述常量的定义是正确的。

在PASCAL语言程序中，允许定义若干个常量共用一个CONST，也允许每个被定义的常量都加上CONST。

使用定义常量便于修改程序。对在程序中多次出现的同一个常量如100，若定义了一个标识符BIG当需要把100改为500时，只需改变常数定义即可（即改为BIG = 500）。

二、变量

在程序执行中值可以改变的量称为变量。变量是以标识符来命名的。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和所能够执行的运算操作。在程序中用到的变量必须首先在变量说明部分进行说明，然后才能使用。变量说明的语法规则如图2-3所示。

图中，VAR是变量说明的保留关键字。在VAR下可以定义许多不同类型的变量，每一类型之间用分号分隔。每种类型允许定义一个以上的变量标识符，它们之间用逗号分隔。PASCAL系统本身已定义的数据类型称标准数据类型，它们是INTEGER（整数类型）、REAL（实数类型）、BOOLEAN（布尔类型）和CHAR（字符类型）。除此之外，还可