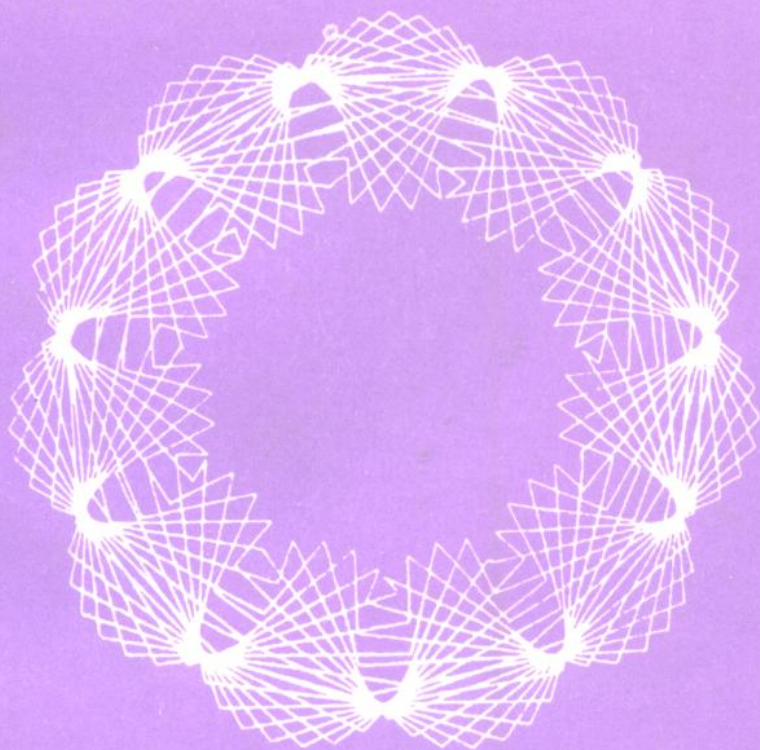


● 计算机应用基础系列丛书

Pascal 语言 程序设计基础

曹兰斌 王嘉祯 等编



电子工业出版社

计算机应用基础系列丛书

Pascal 语言 程序设计基础

曹兰斌 王嘉祯 等编

电子工业出版社

(京)新登字 055 号

内容简介

本书介绍了 Pascal 语言的数据类型、语句和程序设计方法,深入浅出、循序渐进、示例丰富、通俗易懂。

全书共分十二章,第一章概述了计算机必要的基本知识,第二章介绍了标准数据类型和标准函数,第三章讲述了各种语句和相应的程序,第四章讲述过程和函数,第五章至第八章讲述用户自定义数据类型和程序(枚举、子界、数组、集合、记录、文件、指针)。第九、十两章介绍 Pascal 语言与汇编语言的接口和编译程序运行过程,第十一章是 Turbo Pascal 基本说明,第十二章分析了全国计算机软件水平考试关于 Pascal 语言部分的试题。

本书可作为计算机软、硬件专业及应用专业的程序设计教学用书,也可作为计算机应用科技人员自学、培训或参考工具书。

计算机应用基础系列丛书 Pascal 语言程序设计基础

曹兰斌、王嘉祯 等编

责任编辑 王昌铭

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经售

河北省望都县印刷厂印刷

开本 850×1168 毫米 1/32 印张:7.625 字数:198 千字

1994 年 3 月第 1 版 1994 年 3 月第 1 次印刷

印数:1—8000 册 定价:6.2 元

ISBN7-5053-2443-8/TP·713

前 言

随着计算机技术的迅速发展,计算机的应用范围日益广泛。为适应计算机应用工作的普及和发展的需要,我们编写了本丛书。本丛书共十二册,涉及到计算机应用基础知识的各个方面,内容丰富、实用性强,是全面学习和掌握计算机应用知识难得的普及性读物。其内容包括:《操作系统及应用》、《微型机常用通用软件》、《计算机常用汉字输入方法》、《Pascal 语言程序设计基础》、《计算机硬件基础》、《计算机软件基础》、《C 语言应用基础》、《dBASE 数据库应用基础》、《ORACLE 数据库应用基础》、《计算机网络应用基础》、《微型机常见故障维修实例》、《微机屏幕提示信息英汉译义》。

Pascal 语言是沃斯(N. Wirth)在六十年代末提出来的。它较好地体现了结构程序设计的原理,是定义结构程序设计概念的第一个语言,在世界上,它是程序设计语言发展的一个里程碑。Pascal 语言通用性强,简明易懂,容易学习、使用,是高级语言程序设计知识的基础,近年来得到了较广泛的应用。在编写本书时,我们本着通俗易懂、深入浅出的原则,力求简明、实用。它既可作为高级程序设计教学用书,亦可作为各层次人员自学和参考的工具书。

本书由军械工程学院曹兰斌、王嘉祯、张景芳、叶曲炜、谷绍洪合编。全书由曹兰斌教授统稿,由田秋成、木林林、王大刚主审。

由于编者水平有限,误漏之处在所难免,恳请批评指正。

编者

1993 年 8 月

目 录

第一章 概述	(1)
§ 1.1 微型计算机硬件及软件简介	(1)
§ 1.2 Pascal 语言简介	(1)
§ 1.3 Pascal 程序举例	(4)
习题一	(8)
第二章 Pascal 语言基本知识	(9)
§ 2.1 标识符和保留字	(9)
§ 2.2 直接量、常量和变量	(10)
§ 2.3 标准数据类型	(12)
§ 2.4 表达式和运算符	(14)
§ 2.5 标准函数及其引用	(17)
习题二	(21)
第三章 语句	(23)
§ 3.1 赋值语句	(23)
§ 3.2 读语句 READ 和 READLN	(24)
§ 3.3 写语句 WRITE 和 WRITELN	(25)
§ 3.4 条件语句 IF	(28)
§ 3.5 CASE 语句	(31)
§ 3.6 GOTO 语句和标号	(34)
§ 3.7 FOR 循环语句	(35)
§ 3.8 WHILE 循环语句	(37)
§ 3.9 REPEAT 重复语句	(37)
§ 3.10 循环出口语句和多重循环	(39)
习题三	(40)
第四章 过程与函数	(42)
§ 4.1 过程的概念	(42)

§ 4.2 变量与参数	(44)
§ 4.3 函数	(50)
§ 4.4 库过程与库函数	(52)
§ 4.5 过程参数与函数参数	(55)
§ 4.6 过程的嵌套和递归调用	(60)
习题四	(66)
第五章 枚举类型与子界类型	(69)
§ 5.1 枚举类型	(69)
§ 5.2 子界类型	(74)
习题五	(77)
第六章 构造型数据类型	(78)
§ 6.1 集合类型	(78)
§ 6.2 数组类型	(86)
§ 6.3 记录类型	(99)
习题六	(113)
第七章 文件	(115)
§ 7.1 文件的概念	(115)
§ 7.2 有类型文件	(116)
§ 7.3 文本文件	(123)
§ 7.4 输入和输出	(129)
习题七	(134)
第八章 指针类型	(136)
§ 8.1 指针的概念	(136)
§ 8.2 动态变量的使用方法	(140)
§ 8.3 指针的应用	(142)
习题八	(148)
第九章 Pascal 语言与汇编语言程序的接口	(149)
§ 9.1 问题的提出	(149)
§ 9.2 如何调用汇编语言子程序	(149)
习题九	(157)

第十章	Pascal 程序的编译和运行	(158)
§ 10.1	准备工作	(158)
§ 10.2	建立 Pascal 环境	(159)
§ 10.3	编译步骤	(160)
第十一章	Turbo Pascal 基本说明	(167)
§ 11.1	Turbo Pascal 简介	(167)
§ 11.2	Turbo Pascal 与标准 Pascal 比较	(169)
§ 11.3	Turbo Pascal 操作简介	(185)
§ 11.4	Turbo Pascal 程序的调试	(194)
第十二章	软件水平考试 Pascal 语言	
	部分的试题分析	(198)
§ 12.1	1990 年程序员级和高级程序员级试题分析	(198)
§ 12.2	1992 年程序员级试题分析	(208)
附录 A	Pascal 的字汇表	(214)
附录 B	标准标识符	(216)
附录 C	ASCII 字符集	(221)
附录 D	Pascal 语法图	(222)

第一章 概 述

§ 1.1 微型计算机硬件及软件简介

微型计算机从 1971 年问世以来,发展非常迅速,应用相当广泛。其运行速度和信息处理能力足以满足社会上的广泛需要。目前它的功能和应用领域正在进一步发展,形成了计算机应用的新时代。

微型机系统由硬件和软件两大部分组成。最常用的微型机有 IBM PC 机及兼容的 286、386 机,以及长城系列机等。本书主要结合微型机介绍 Pascal 语言。为便于了解 Pascal 语言在计算机系统中处于什么位置,首先简介一下微型机的硬件和软件及其它们的相互关系。

微型计算机系统的硬件和软件的内容如图 1-1 所示。

硬件和软件的关系用层次关系表示,如图 1-2 所示。

本书所介绍的 Pascal 语言是在操作系统控制管理下的一种系统软件,使用它用户可以开发各种具体问题的应用软件。

§ 1.2 Pascal 语言简介

Pascal 语言由瑞士的 N. Wirth(N. 沃思)教授于 1971 年在苏黎世联邦工业大学提出来的,是结构程序设计的一种描述算法的语言,为纪念著名哲学家、数学家 P. Pascal(P. 帕斯卡)将其命名为 Pascal 语言。十多年来它发展很快,74 年颁发了标准 Pascal 用户手册,随后在各种类型的计算机上都配备了 Pascal 语言。IBM PC 机

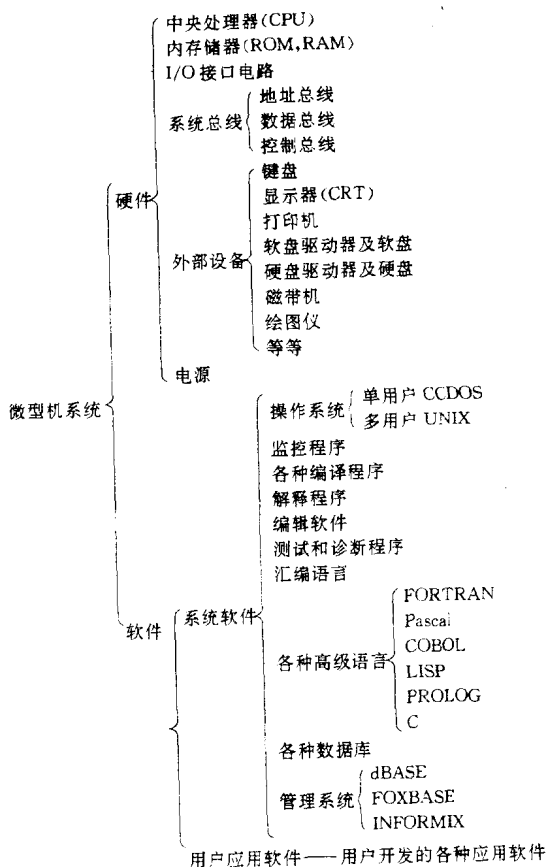


图 1-1 硬件和软件

生产后,立即开发了微机 PC Pascal 的各种版本,后来 BorLand International(博兰德公司)又开发了 Turbo - Pascal 版本,它的运行速度快,功能强,使用更加方便。

Pascal 语言具有丰富的数据类型,程序风格优美。它适用于教学,便于编写系统软件、应用软件以及科学计算等。

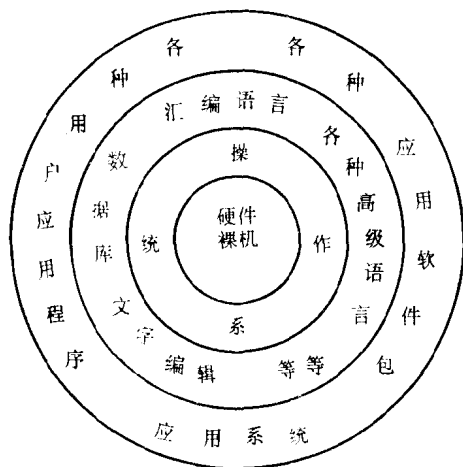


图 1-2 硬件和软件的关系

FORTRAN 语言的应用领域主要是数值计算,其原因是它的数据类型少且构造数据结构的方法简单,而 Pascal 语言提供了丰富的数据类型和构造数据结构的方法。除整型、实型、字符型、布尔型、数组型之外,还增加了子界类型、枚举型、集合型、记录型、文件型和指针型及字符串等。因此,Pascal 语言的应用范围很广、不仅适用于数值计算,而且运用于非数值计算方面的信息管理,编写系统软件等。

Pascal 语言的源程序,清晰易读,便于程序的修改和维护。它的程序为分块结构,由外至内按层次构成,可采用逐层锯齿形来书写,使程序的层次分明。又因为该语言提供了控制结构,运用这些控制结构可以构造出不使用 GOTO 语句的程序,使程序的静态结构和动态结构相一致,从而使读者容易阅读程序,理解程序内容。实践证明,使用 Pascal 语言编写的程序,在进行修改时,通常要比非结构式的语言程序如 BASIC 程序、FORTRAN 程序等省力得多。

Pascal 语言产生至今虽然只有二十年,但它已成为国际上应用最广泛的几种高级程序设计语言之一,并已普及于全世界,世界

各国的主要大学都采用 Pascal 语言进行程序设计教学。

§ 1.3 Pascal 程序举例

程序是对给定数据处理过程的描述。不同的语言系统有不同的成分和结构形式,使用 Pascal 语言描述的程序称为 Pascal 源程序。学习一门语言系统的最好办法是首先运行一个简单的程序,这包括如何按规定的结构书写程序,如何输入计算机并进行编辑,如何运行程序等。这一节给出一个简单示例程序,说明上述问题,并以此为模式,逐渐设计自己需要的程序。

例 1.1 对任意给定的整数找出其质因数,输出结果,试写出它的源程序,并编辑和运行。

第一步:编写程序。该问题的程序如下:

```
program exp1(input,output);           {程序首部}
label 10;                             {标号说明}
var c,i,j,x;integer;                 {变量说明}
begin                                {主程序开始}
    c:=1;
    readln(i);                       {输入整数}
    if i=0 then                      {i 为 0 转结束}
    begin
        writeln('the i is invaild');
        goto 10;
    end;
    write(i,'=');                    {准备输出格式}
    if i<0
    then begin
        write('-1 * ');              {处理负数}
        i:=-i;
    end;
```

```

x:=i;
while x div 2 * 2=x do           {若 i 能被 2 整除
    begin                       则连续用 2 整除}
        x:=x div 2;
        c:=c * 2;
        write('2 * ')
    end;
for i:=3 to i div 2 do          {在 3 至 i/2 间
    begin                       找 i 的质因数}
        while x div j * j=x do
            begin
                x:=x div j;
                c:=c * j;
                write(j);
                if c=i then
                    goto 10;      {若 c=i 表示质因数
                write(' * ')      分解完毕转结束}
            end
        end;
    10;
end.

```

Pascal 程序由程序首部、说明部分和执行部分三个部分组成，缺一不可。

第一行称为程序首部，保留字 Program 标志程序的开头，后面跟着自定义的程序名 exp1，小括号内是程序与外界联系的参数，本程序中 Input 表示有输入操作；Output 表示有输出操作。程序首部最后是“;”表示程序首部结束。

从第二行开始至 begin 之前的若干行为程序的说明部分，Label 定义标号；Var c,i,j,x:integer 定义了四个整型变量，还可以定义程序使用的各种类型的变量。除本程序的说明部分的内容外，还包括常量定义，类型定义，过程函数说明等。

从 begin 行开始至最后行的 end. 的中间部分是程序的执行部分,由若干语句构成,每个语句用“;”分隔。

程序中由(* *)括起来的文字是注解,也可以使用花括号 {}。注解可以放在一行的“;”之后,也可以作为单独一行,对程序的运行没有任何影响,打印出来是保留的,以备理解程序,便于维护程序。

第二步:将程序输入计算机,可以使用行编辑程序或字处理程序以及 Pascal 系统的编辑功能从键盘将程序输入。磁盘文件扩展名约定为 pas。

第三步:将程序文件进行 Pascal 编译,有错误则需排除,然后重新编译,直到没有错误为止。然后进行连接(有的 Pascal 不需要连接)。

第四步:运行,在操作系统下键入程序名则开始运行。

例如上述程序运行后,输入数据:

输入:6

输出: $6=2*3$

输入:-378

输出: $-378=-1*2*3*3*3*7$

输入:259

输出: $259=7*37$

上述程序虽然简单,但结构完整,步骤清楚。以此为范例可以模仿写出自己的程序并上机运行。它可以作为你学习 Pascal 的参考程序。

对于 Pascal 的程序结构一般用语法来说明,它的语法规则可采用两种方式说明:一种是巴克斯范式形式 BNF (Backus Naur Form),这是一种由非终结符、终结符和元符号组成的语法规则。终结符,指语言系统的字符集的字符或保留字及标准标识符,非终结符,指具有确定含义的语法成分,如标识符,表达式等,一般用尖括号括住。以下是 BNF 的元符号:

::=意思为定义为;

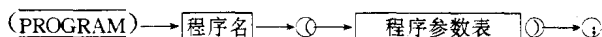
| 意思为“或”;

{ }花括号中的项可以重复多次。

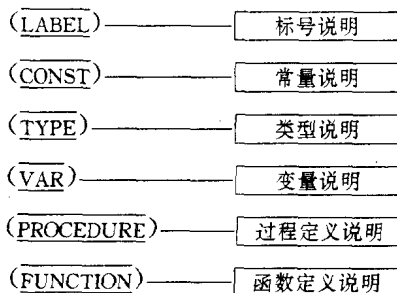
例如, $\langle \text{relation_operation} \rangle ::= < | > | > = | < = | = | < > |$
IN 表示关系运算符(非终结符)定义为小于或大于或大于等于或小于等于或等于或不等于或属于等等。

另一种是图解法,它称为语法图,语法图方法直观,容易理解,本书附录主要使用它来描述语法规则。

在语法图中以端圆框或圆框围起来的内容表示必须出现的实际字符,用户不能更改为其它字符,矩形框围起来的内容是由用户定义的实体。图 1-3 表示了 Pascal 程序的一般结构的语法图。



(a) 程序首部



(b) 程序说明部分



(c) 程序执行部分

图 1-3 Pascal 程序结构语法图

图 1-3(a)是程序首部语法,端圆框或圆框的字母符号串和符号是规定的必用符号,矩形框的程序名和参数表,用户自己定义。(b)是程序的说明部分,包括五部分,在一个程序中这五部分并非一定都出现,在程序执行部分中引用的对象才需要说明。但

Pascal 语言本身定义的标准常量、标准过程的函数等,不必说明就可以在程序的执行部分引用。(c)是程序执行部分语法图,它由一系列语句组成,每个相邻语句之间用 begin 和 end 表示程序执行部分的开始和结束,程序以“·”符号作为结尾。

上述 Pascal 程序结构中的内容是 Pascal 语言全部内容的概要,这些内容将在以后各章中逐一介绍。

习 题 一

- 1.1 简述 Pascal 语言的特点。
- 1.2 简述 Pascal 源程序的结构及各部分的作用。
- 1.3 给出“Pascal 源程序”语法成分的 BNF 语法规则。

第二章 Pascal 语言基本知识

本章主要介绍基础知识,前三节讨论标识符、保留字和基本量;后二节讨论 Pascal 语言的标准数据类型和程序结构。

§ 2.1 标识符和保留字

2.1.1 标识符

标识符是用来定义程序、过程、函数、数据类型、常量、变量等名称和符号。例如在第一章的程序中 `exp1` 表示程序的名称;`c`、`i`、`j`、`x` 表示整型变量的名称,它们都是标识符。

标识符由一个字母开头,后跟任意个字母、数字组成,在 Turbo Pascal 中还可以后跟任意个下划线,举例如下:

<code>sum</code>	合法标识符
<code>persons _ coun</code>	合法标识符
<code>hit16a</code>	合法标识符
<code>first time</code>	非法,不能包含空格
<code>next, word</code>	非法,不能包含逗号
<code>3may</code>	非法,以数字开头

标识符的长度仅由行的长度即 127 个字符所限制,但标准 Pascal 规定只有前 8 个字符是有效的, Turbo Pascal 规定标识符中所有的字符均有效。例如 `thisisapascal` 与 `thisisaprogram` 在标准 Pascal 中认为是同一标识符,而在 turbo pascal 中则认为不同。

在 Pascal 语言中,有一些特殊的标识符,它们的含意由编译程序规定,称之为标准标识符,例如标准常量标识符有 `false`, `true`,

maxint, 标准类型标识符有 integer, real, boolean, char, 还有标准文件标识符等, 详见附录。这些标识符, 用户可直接引用, 不需再定义。注意, 用户一旦重新定义标准标识符, 那么它将失去原有的含意。若按原来的定义使用将产生错误, 因此, 用户一般应避免对标准标识符重定义。

选定好的标识符是程序设计的重要工作之一, 什么是好的标识符呢? 一般应注意: 避免使用易混淆的字母和数字, 如字母“O”和数字“0”不易区别, to20 是一个不好的标识符; 标识符应有一定的意义, 如一个删除过程名可选定 delete 为其过程名; 一个长的标识符未必好, 如一个标识符在一段程序中只用几次, 就可以用单个字母表示之。

2.1.2 保留字

保留字是 Pascal 的组成部分, 有特定的意义, 用户绝对不能重新定义为用户使用的标识符, 除了在注释中以外, 它们只能用于语言中定义的用途。例如在 Pascal 程序中, 语句部分是由保留字 BEGIN 开始, 由保留字 END 结尾。循环语句的保留字为: WHILE DO(一种)等等。详细情况参看附录。

§ 2.2 直接量、常量和变量

2.2.1 直接量和常量

在程序中一个具体的值称为一个直接量, 例如下面的程序:

```
FOR j:=1 TO 100 DO  
  X:=3.14159265*j;  
  ch:='$'  
  LG:='H'
```

其中数值 1, 100, 3.14159265, 字符 '\$', 'H' 均为直接量, 用户可以给直接量命名, 命名了的直接量称为常量, 在 Pascal 中, 常量说