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

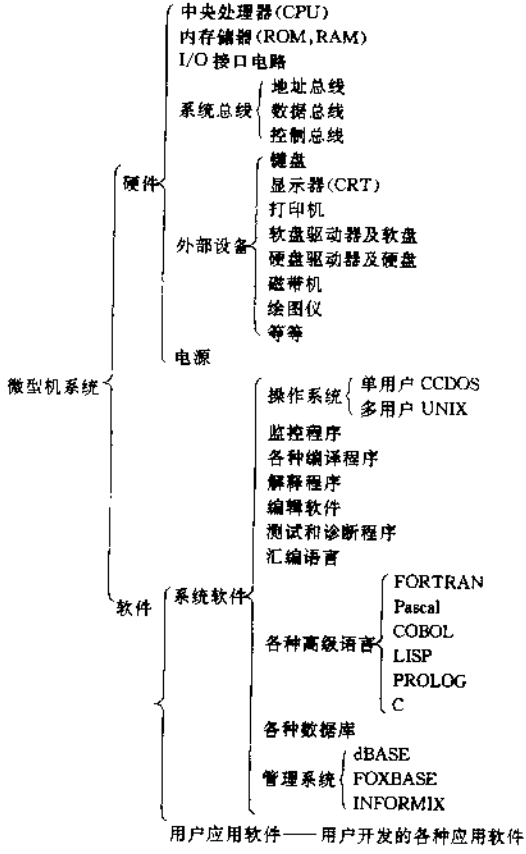


图 1-1　硬件和软件

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。

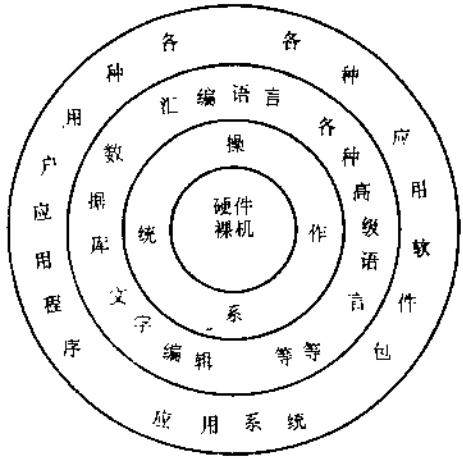


图 1-2 硬件和软件的关系

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；

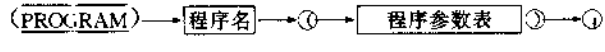
| 意思为“或”；

{ }花括号中的项可以重复多次。

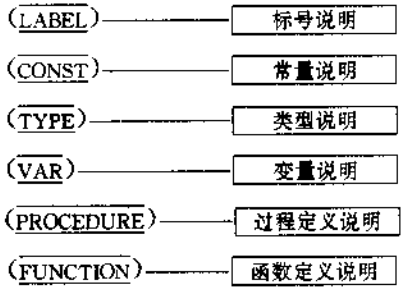
例如，⟨relation _ operation⟩::=<|>|>=|<=|=|< >| IN 表示关系运算符(非终结符)定义为小于或大于或大于等于或小于等于或等于或不等于或属于等等。

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

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：

sum	合法标识符
persons _ coun	合法标识符
hit16a	合法标识符
first time	非法，不能包含空格
next,word	非法，不能包含逗号
3may	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

明部分就是给直接量命名的，如对上面的直接量进行常量说明：

```
CONST
cyel1:=1;
cyel2:=100;
pi:=3.14159265;
ch1:='$';
ch2:='H';
```

此时，cyel1，cyel2，pi，ch1，ch2 称为常量标识符，它们是存放相应值的实体。上面程序可写成：

```
FOR j:=cyel1 TO cyel2 DO
x:=Pi*j;
ch:=ch1;
LG:=ch2;
```

顾名思义常量是一种在程序执行过程中其值不改变的量。一个好的程序设计，应除在常量说明部分之外，很少使用直接量，这给修改程序带来方便，欲修改一个直接量，只需修改常量的说明部分，勿需改动程序部分。

实数定义格式如下：

1，－1.5，1E5，1.5E－5 是合法的实数。但－5，E3，1.E－2，3.0E 是不合法的实数写法。

注意，整数或实数直接量总是以一个数字或数字符号开始的。

2.2.2 变量

在程序中，每一项数据不是常量就是变量，区别在于变量的值在程序执行过程中可以改变。

程序中每一个变量均必须而且只能与某一个数据类型相关，一个变量的类型定义了该变量可接受值的集合。

直接量和常量的类型由编译程序决定，用户不必去说明，但是变量的类型必须在使用前由用户说明。变量说明的例子如下：

```
VAR
  count,indc,i : integer;
  fris,last : real;
  enod:boolean;
  charac,endchare:char;
```

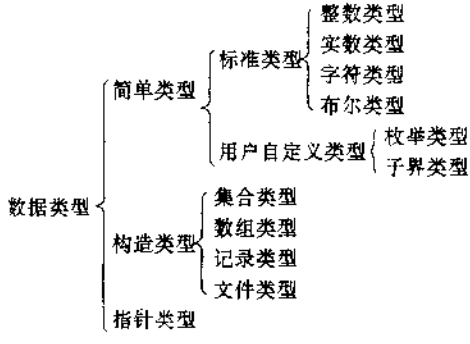
其中 VAR 是定义变量类型的保留字，通过以上变量说明，变量 count，indc，i 定义为整数类型变量，fris，last 定义为实数类型变量，enod 定义为布尔型变量，charac，endchare 定义为字符型变量。

§2.3 标准数据类型

Pascal 语言的优点之一是具有丰富的数据类型。这些数据类型可以分成三大类。

第一类是简单数据类型，这一数据类型构造简单，又称“非构造型数据类型”。它包括常用的四种标准的数据类型和两种用户自定义的数据类型。第二类是构造型数据类型，又称复杂型数据类型，其构造复杂，一般由其它数据类型按照一定的规则构造而成；第三类是指针类型，它与前两类数据类型不同，主要用于解决动态数据的建立、删除和修改等问题。

Pascal 数据类型，有标准类型和用户自定义类型共 11 种，如下所示：



本节首先介绍标准数据类型。

2.3.1 整数类型

整数的取值范围与特定的计算机的字长有关，但对每台特定的计算机都有一个表示该机所能允许的最大整数的标准标识符maxit，任何一个整数N必须满足：

$$-maxit \leqslant N \leqslant maxit$$

超出这个范围将产生溢出错误，例如：1000 * 100/50 的运算结果不会是2000 这是由于乘法运算1000 * 100 产生了溢出。

2.3.2 实数类型

实数的取值范围与特定机器有关，例如 IBM - PC Turbo pascal 中实数的正值范围为 1E - 38 到 1E - 38，精度为 11 位有效数字，在内存中占六个字节。在实数的运算过程中，若超出实数范围，则产生溢出错误，程序中止运行，系统显示执行错误。

Pascal 有两种形式的实数，一种是带小数点的形式，例如：

-3.5；0.0；-1234.56；$+147.532$

一种是带字母 'E' 的形式，例如：

9.1045E28；+5.0E+2；5E+2；5E2

-6.08E-15，-1.0E+2

都是合法实数，而下列实数：

6.08E+0.5，　+E-2，　-5E

都是非法实数。

字母 'E' 表示以 10 为底若干次幂，如 1.0E+2 表示实数 10^2；-6.08E-15 表示 -6.08×10^{-15}。

2.3.3 布尔类型

布尔类型的直接量只有两个：true(真)和 false(假)，因此布尔类型变量的取值为这两个值之一。

布尔型的变量在程序中十分有用，它主要用于控制程序的语句执行次序。

2.3.4 字符类型

一个字符直接量是写在引号之间的一个字符，例如：'A' 表示字母 A；'b' 表示字母 b。

字符型变量的值是一个字符，而不是一串字符，关于字符串处理在结构数据类型中讨论。字符型变量的取值范围是 ASCII 字符集（见附录），所有字符均按其 ASCII 码值排序，例如：'A'>'B'，因为字符 A 的 ASCII 码值为 65，而 B 的码值为 66。

§2.4 表达式和运算符

Pascal 语言有着广泛的数值计算功能，表达式是描述数值诸规则的算法结构，它是由操作数（变量名、常量名、函数名等）和运算符联结而成。

本节介绍如何用标准数据类型的量构成表达式，首先讨论 Pascal 语言所允许的运算符，按其运算优先级的大小，可以将运算符分为如下五类：

1. 一元减运算符 '—'，表示取操作数的负值，操作数只有一个，或是实型或是整型。例如表达式：—5，—0.5，—36 等称一元表达式。

2. 逻辑非运算符 NOT，表示取布尔操作数的逻辑非值。例如表达式：NOT true 取 false 值，NOT false 取 true 值。

3. 乘法等运算符如表 2-1 所示。

如果定义下列变量类型并给它们赋值：

```
VAR x1、x2:integer;
    y1、y2:REAL;
```

表 2-1

运算符	运算	操作数类型	结果类型
*	乘	实型	实型
		整型	整型
		实型、整型	实型
/	除	实型	实型
		整型	实型
		实型、整型	实型
DIV	整除	整型	整型
MOD	取模	整型	整型
*AND	算术与	整型	整型
AND	逻辑与	布尔	布尔
*SHL	左移	整型	整型
*SHR	右移	整型	整型

并且：

x1:=12 ;x2:=34;y1:=1.5;y2:=3.6;

则下面例子的结果为：

```
x1 * x2              =408
x2/x1                =2.83
x1 * y1              =18.0
123   DIV  4         =30
12    MOD  5         =2
true  AND  false     =false
12    AND  22        =4(01100 ∧10100=00100)
2     SHL  7         =256
256   SHR  7         =2
```

在乘法运算符表中，运算符前没打星号'*'的运算符是标准Pascal语言中所允许的运算，打'*'的运算符是turbo Pascal中允

许的运算。可见 turbo Pascal 扩充运算功能。

4. 加法等运算符如表 2－2 所示。

表 2－2

运算符	运算	操作数类型	结果类型
+	加	实型	实型
		整型	整型
		实型、整型	实型
-	减	实型	实型
		整型	整型
		整型、实型	实型
* OR	算术或	整型	整型
OR	逻辑或	布尔	布尔
* XOR	算术异或	整型	整型
* XOR	逻辑异或	布尔	布尔

5. 关系运算符：关系运算符有以下六个：

=（等于）、<>（不等于）、>（大于）、<（小于）、>=（大于等于）、<=（小于等于）。

关系运算符的操作数可以是标准数据类型的量，用以比较两个量的大小或相等。其结果只有两个值 true 或 false（真或假）。

设 a，b 为标准数据类型的变量，则：

a=b　　如果 a 等于 b 结果为真，否则为假。

a<>b　如果 a 不等于 b 结果为真，否则为假。

a>b　　如果 a 大于 b 结果为真，否则为假。

a<b　　如果 a 小于 b 结果为真，否则为假。

a>=b　如果 a 大于等于 b 结果为真，否则为假。

a<=b　如果 a 小于等于 b 结果为真，否则为假。

例如：

5=5　　结果为真。

5=10　结果为假。

false<true　　结果为真

false>true　　结果为假

'A'<'C'　　结果为真

12.5>=3　　结果为真

18.5<>-10.3　　结果为真

24>=24　　结果为真

10.3<10　　结果为假

运算符在表达式中是有优先次序的，各种运算符的优先级如下：

(1)圆括号(…(…(…)…))，按由内至外，逐层展开的规律进行。

(2)一元减；

(3)逻辑非：NOT

(4)乘法运算符：*，/，DIV，MOD，AND，SHL，SHR。

(5)加法运算符：+，-，OR 和 XOR。

(6)关系运算符：=，<>，<，>，<=，>=。

优先级相同的运算符，按在表达式中出现从左到右顺序计算。

§2.5　标准函数及其引用

在标准 Pascal 语言中，共定义十七种标准函数，用以实现最通用的数学函数与其它一些特殊的函数运算。这些函数的名称已经确定，作为 Pascal 语言中的标准保留字(预说明的标识符)，明确建议用户不把它们用作用户定义的标识符。

标准函数，用户不必对它们进行任何说明就可以直接引用。引用格式如下：

标准函数名---(---自变量---)　□

标准函数都只用一个自变量，其类型与取值范围必须符合相