第 1 章　PASCAL 语言程序设计介绍

PASCAL 语言是由瑞士的沃斯(N. Wirth)教授于 1971 年提出来的。它的命名是为了纪念法国数学家 Pascal。

本章将介绍 PASCAL 语言的特点、基本符号、保留字、标识符与程序结构。这些内容是很重要的，是今后学习和正确编写程序所必需的知识。

1.1　PASCAL 语言的特点

PASCAL语言是在 ALGOL60 基础上发展起来的，它有如下特点：

(1) 它是世界上第一个结构化程序设计语言

由戴克斯特拉(E. W. Dijkstra) 和霍尔(C. A. R. Hoare) 提出的结构化程序设计思想，是程序设计发展史上的一个里程碑。他们主张在程序设计中去掉 GOTO 语句,所有程序都可以由三种基本结构(顺序结构、选择结构、循环结构) 组成。后来有人把函数和过程作为第四种基本结构。这四种基本结构对外来看都只有一个入口,一个出口,结构清晰,避免了由 GOTO 语句所引起的混乱。另外在程序设计方法上，他们主张采取自顶向下，逐步求精的方法。即将一个大的复杂的问题，划分成若干小的易解决的问题。每个小问题，又可划分成一些更小的更易解决的问题。这样,每个小问题解决了,整个大问题也就解决了。这种方法还为多个人同时编程序提供了方便。

而 PASCAL 语言正是基于结构化程序设计思想建立的。它所提供的语句可以充分满足实现四种基本结构的需要。它的函数和过程又为进行自顶向下，逐步求精提供了方便。

由于 PASCAL 语言具有良好的结构化程序设计特性，所以它特别适合于教学，适合于培养学生掌握自顶向下逐步求精的结构化程序设计思想和方法，并养成良好的程序设计风格和习惯。因此，国内外许多大学都将 PASCAL 作为第一门程序设计教学语言。

(2) 功能强、应用广

PASCAL 语言提供了丰富的数据类型和语句，功能强、应用广。它不仅适合于教学，也可广泛用于编写各种系统软件和应用软件。

(3) 编译和运行效率高

在 PASCAL 语言中提供了必要的说明，并去掉了一些影响效率的因素 (例如去掉了乘幂运算,去掉了字符串运算,去掉了动态数组等),使得 PASCAL 语言编译和运行效率都较高。

(4)可移植、易推广

PASCAL 语言标准化程度高，不依赖于具体的机器，用 PASCAL语言写的源程序可以在各种具有 PASCAL 编译系统的机器上运行。如果某机型没有 PASCAL 编译系统,也可以通过用 PASCAL 语言写编译系统的自编译方法，为该机型产生 PASCAL 编译系统。

现在世界上几乎所有大、中、小型计算机都已配置了 PASCAL 编译系统，为 PASCAL 语言的广泛使用打下了基础。

1.2 基本符号、保留字、标识符

1.2.1 基本符号

PASCAL 语言只能使用以下几类基本符号：

(1) 大小写英文字母

A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z

a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z

(2)数 字

0,1,2,3,4,5,6,7,8,9

(3)其它符号

+,−,*,/,=,<>,<=,>=,<,>,(,),[,],{,}, :=,,,;,.,:,..,',↑

注意,PASCAL 语言除了能使用以上规定的基本符号外，不得使用任何其它符号。例如：

$\alpha,\beta,\gamma,\pi,\epsilon,\Sigma,\int$, $

等都不得在 PASCAL 语言中使用。

1.2.2 保留字

在 PASCAL 语言中，有些词具有特定的含义。用户必须了解其含义，以便正确地使用,否则会造成错误。这些具有特定含义的词被称为保留字。保留字一共有 35 个，它们是：

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

保留字除按规定的意义使用外，不得另作它用。在书写时，保留字可以用大写，也可以用小写。在本书中，为了醒目和提高可读性，将保留字用大写，其它符号用小写。

按标准 PASCAL 语言规定，除了引号中的字母外是不区分大、小写的。但是在上机时，必须了解你所使用机器的 PASCAL 编译系统，看它对大、小写字母是如何规定的，以便正确输入和编译。

1.2.3 标识符

标识符是以字母开头的字母、数字组合。例如：

x,y,max,min,sum,a15,a3b7

都是合法的标识符。而如：

5x,x－y,α,π,ε,ex10.5

则不是标识符，或称它们为非法的标识符。

为了描述合法的标识符，可以借助语法图来描述。语法图是描述算法语言形式语法的工具。在语法图中，用矩形框表示需要进一步定义的语法单位。用圆框或两头是圆的长方形框表示终结符（基本符号或保留字）。用箭头表示它们之间的连接关系。

标识符的语法图如图 1.1 所示。

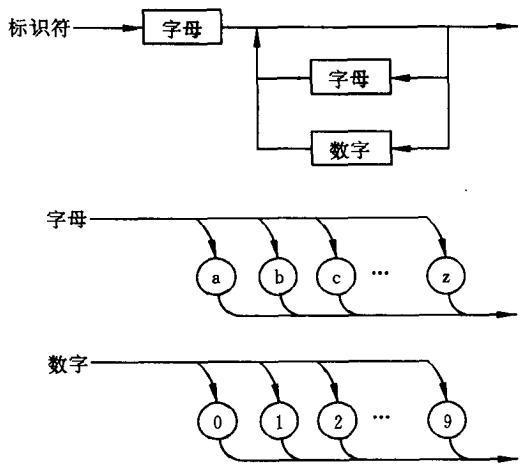

图 1.1　标识符的语法图

标识符可用来表示常量、变量、类型、文件、函数、过程或程序的名字。

标识符的长度(字符个数)是没有限制的。但标准 PASCAL 规定能区分的有效标识符长度是 8。超过 8 个字符的标识符也能使用，但是以前 8 个字符作为有效字符。因此，当两个标识符的前 8 个字符相同时，则认为是相同的标识符，多余的字符将不起作用。例如：

students1 和 students2

将被认为是相同的标识符。因为它们的前 8 个字符相同。

不同的 PASCAL 编译系统，对于可区分的有效标识符长度的规定可能是不同的，在使用时要注意。

标识符的选取最好有一定的含义，这样便于记忆，也增加了程序的可读性。标识符的书写可以用大写、小写或大、小写字母混合使用。在本书中采用小写字母。

在 PASCAL 语言中，有些标识符具有特殊的含义，称它们为标准标识符。标准标识符一共有 39 个，它们是：

标准常量（3 个）：

false,true,maxint

标准类型（5 个）：

integer,real,char,boolean,text

标准文件（2 个）

input,output

标准函数（17 个）

abs，arctan，chr，cos，eof，eoln，exp，ln，odd，ord，pred，round，sin，sqr，sqrt，succ，trunc

标准过程（12个）：

get，new，pack，page，put，read，readln，reset，rewrite，unpack，write，writeln

这些标准标识符的含义和用法，将在以后各章中陆续介绍。读者在定义自己的标识符时要注意不要与保留字和标准标识符重名，以免发生错误或引起混淆。

1.3 程 序 结 构

对于 PASCAL 程序的结构是有严格规定的。为了说明这个规定，我们先来看一个例子。

【例 1.1】 已知圆的半径，求圆的周长和面积。

设圆的半径为 r，周长为 l，面积为 s。根据数学公式：

$$l = 2\pi r$$

$$s = \pi r^2$$

其中 r，l，s 可以作为合法的标识符在程序中使用。 π 是一个常量，但它不是合法的标识符，可选用 pi 来代替。

PASCAL 语言规定，程序中用到的常量和变量都必须在程序中加以说明。即说明常量的值和变量的类型。对于计算的初始数据，可以通过读语句从终端键盘读入。计算可以利用赋值语句来实现。计算结果可以通过写语句输出到屏幕或打印机。

该例完整的 PASCAL 程序如下：

程序 1.1 已知半径，求圆周长和面积的程序

```
PROGRAM  circle (input,output);
   {已知半径求圆周长和面积}
  CONST
     pi=3.14159;
  VAR
     r,l,s:real ;
  BEGIN
     read(r) ;
     l :=2 * pi * r;
     s :=pi * r * r ;
     write(r,l,s)
  END.
```

1. 程序首部

程序 1.1 的第一行称为程序首部。PROGRAM 是保留字。每个 PASCAL 程序都必须以它开头。circle 是该程序的名字。每个程序的名字可以不同，但必须是合法的标识符。圆括号里的内容称为程序的参数。程序参数指明程序与外部联系的文件名。 input 是标准输入文件，例如键盘打字机。output 是标准输出文件，指屏幕显示器或打印机。为了读入数据和输出结果，必须写上文件参数 input 和 output。

程序中由花括号括起来的内容称为注释。该程序的第二行就是一个注释，它说明了该程序的目的。注释除了给人看，以增加程序的可读性外，对编译和运行都不起作用。一个程序可以包含多个出现在不同地方的注释，也可以没有注释。

2. 说明部分

在程序 1.1 中，从第三行到最后一行为程序的分程序。分程序一般由说明部分和语句部分构成。说明部分可以包括多种类型的说明。此例包括以 CONST 开始的常量说明和以 VAR 开始的变量说明。语句部分必须以 BEGIN 开始，以 END. 结束。中间是一些用分号分开的语句。

在该例的常量说明中，说明了常量 pi 的值为 3.14159。在以后的程序语句中，出现的 pi 都被当作是值 3.14159。

注意，在程序中只能使用已被说明的常量的值，而不得改变它的值。

常量说明（或称常量定义）的一般形式是：

CONST

〈常量标识符〉＝〈常量〉；

〈常量标识符〉＝〈常量〉；

解释 将指定的常量(值)赋给常量标识符。常量标识符的值可由后面的程序使用，但不得改变它的值。

常量标识符的类型与定义它的常量（值）的类型相同。因此，如果值是包括小数点的数，常量就是实型，如果值是不包括小数点的整数，常量就是整型。

常量说明的语法图如图 1.2 所示。

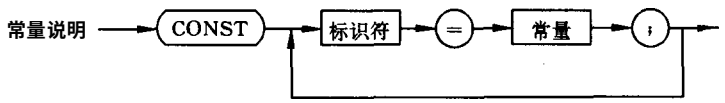

图 1.2 常量说明语法图

在程序中使用常量标识符而不使用数值本身的好处有二：其一是常量标识符的意义明确，使用它可以增加程序的可读性。其二是如果在程序的许多地方用到某个数，当要修改这个数时，必须到各个地方找出这个数，并逐一改成新数，还必须注意不要改动那些与其数值相同而意义不同的数，这是一件较困难的事。如果使用常量标识符，只需改动常量

说明中的数值即可。

在该例的变量说明中,说明 r,l,s 都是实型 (real)。

注意,在程序中出现的所有变量都必须说明它们的类型。变量类型有实型、整型、字符型、布尔型等。

变量说明的一般形式如下:

```
VAR
   (变量表):〈类型〉 ;

   (变量表):〈类型〉 ;
```

解释:变量说明的作用是说明变量的类型。变量表可以是单个变量或是用逗号分开的多个变量。此时表明这些变量的类型相同,可以一起说明。编译程序将为变量说明中的每个变量分配相应的存储单元。

变量说明的语法图如图 1.3 所示。

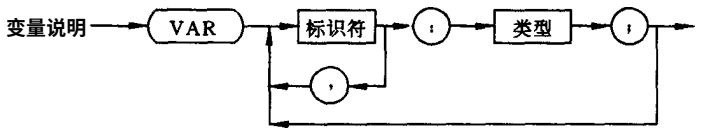

图 1.3 变量说明的语法图

3. 语句部分

从 BEGIN 开始,到 END. 结束是语句部分。程序 1.1 包括 4 个语句,每个语句之间用分号(;)分开。

第一个语句是读语句(或叫输入语句)。它的作用是读入半径 r 的值。当程序运行到此处时,将等待用户从键盘输入数据。这时如果用户从键盘输入:

```
12.5↙
```

则 r 取值为 12.5。符号↙代表回车键,输入数据结束应按此键。一个读语句也可读入多个变量的值,各变量以逗号分开。输入时各数值以空格分开。

第二个语句和第三个语句是赋值语句。它们的作用是按右端表达式计算其值,并将这个值赋给左端的变量。符号:=代表赋值号。在赋值语句中赋值只能用赋值号 (:=)而不能用等号(=)。符号 * 代表乘号。通过这两个语句,将计算出周长 l 和面积 s 的值。

表达式中最常见的运算是加(+)、减(-)、乘(*)、除(/)。在不加括号的情况下,按先乘除,后加减运算。括号只有一种,即圆括号'(' 与')'。遇括号时,先执行括号内的运算,再执行括号外的运算。

表达式必须以线型形式写出。因此,分子、分母、指数、下标等都必须写在同一行上。乘号必须明确写出,不能省略。下面是几个简单的数学表达式所对应的 PASCAL 表达式。

数学表达式	PASCAL 表达式
b^2-4ac	b*b-4*a*c
$\frac{a+b}{2}$	(a+b)/2

第四个语句是写语句(或叫输出语句)。它的作用是输出(显示或打印)r,l,s 的值。通常，读入的数据和计算的结果都要通过写语句显示给人看，或打印留下永久的记录。

在编写程序时，除了安排好运算的先后次序和正确地使用语句外，还必须正确地使用标点符号。例如用分号（；）将程序首部、各个说明、各个语句分开。常量说明中用等号(=),变量说明中用冒号(:),赋值语句中用赋值号(:=)，不要搞错。程序最后应以 END. 结束，最后一个圆点不可少。

另外，要注意程序的书写格式。PROGRAM 写在最左面，与它空一格写程序名。注释的括号 {与 CONST,VAR,BEGIN,END 上下左对齐，且它们比 PROGRAM 向右移两个字符。各个说明语句和程序语句也是上下左对齐，它们比 CONST,VAR,BEGIN 又向右移两个字符。CONST,VAR,BEGIN,END 都写在单独的行上。每个说明和每个语句也都写在单独的行上，即每行只写一个说明或写一个语句。这样写出的程序结构清楚，可读性好，不易出错。希望读者从一开始就养成良好的程序书写风格和习惯。

以上看到的只是一个简单的 PASCAL 程序。一个复杂的 PASCAL 程序将包括更多的说明部分和更多的语句。下面给出一个较完全的 PASCAL 程序结构形式。

程序 1.2 一个完全的 PASCAL 程序结构

```
PROGRAM 程序名(程序参数表);
  LABEL
    标号说明;
  CONST
    常量说明;
  TYPE
    类型说明;
  VAR
    变量说明;
  FUNCTION
    函数说明;
  PROCEDURE
    过程说明;
```

说明部分

BEGIN
语句；
语句；
⋮
语句
END.

语句部分

当然，对于一个具体的程序，它不一定包括以上全部说明。但是如果它们出现，必须以这里所指的先后次序出现。在每个分程序中，保留字 LABEL,CONST,TYPE,VAR 每个只允许出现一次，它们下面的说明语句可以出现多个。例如：

```
VAR
  x,y:real;
  i,j:integer;
```

函数和过程说明两者之间，谁在前、谁在后要根据需要来定。但是它们必须在其它说明之后。函数和过程说明在程序中可以不出现，也可以出现任意多次。

语句部分是不可少的。它以 BEGIN 开始，以 END. 结束。里面可以包括任意多个由分隔符(;)隔开的语句。

PASCAL 程序的语法图如图 1.4 所示。

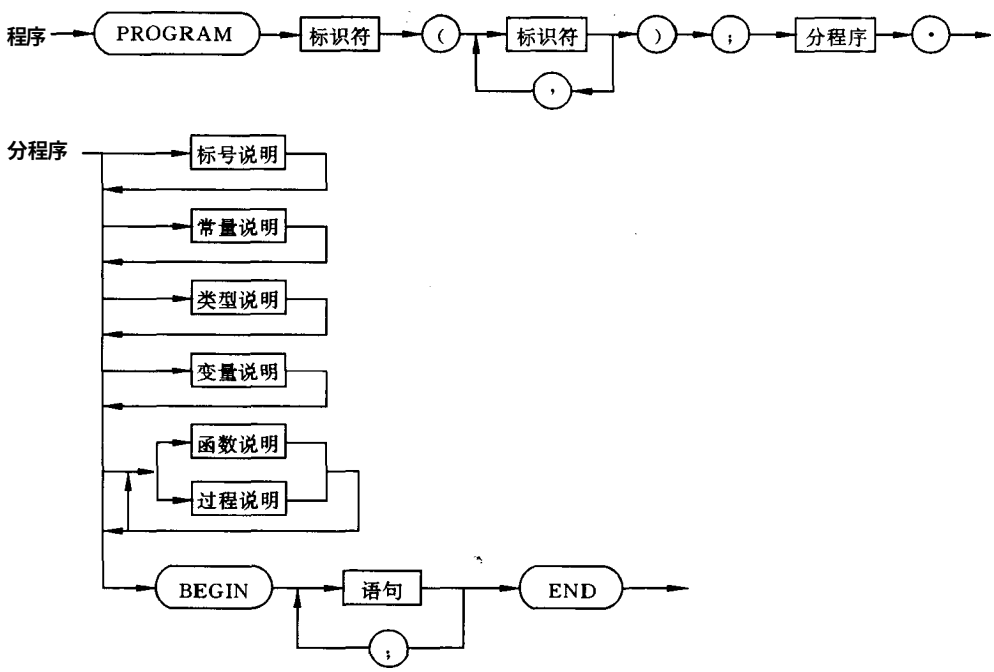

图 1.4 PASCAL 程序的语法图

PASCAL 程序编写完成并检查无误后，即可输入计算机运算。一般是先通过编辑命

令，由计算机终端键盘输入 PASCAL 源程序，建立PASCAL 源文件。然后执行编译命令，得到可执行文件。最后执行可执行文件，并在需要输入数据时，由键盘输入相应数据，完成计算并显示或打印计算结果。

1.4 小　　结

本章介绍了 PASCAL 语言的特点，基本符号，保留字，标识符和程序结构，并给出了一个简单的 PASCAL 程序的例子。

PASCAL 是世界上第一个结构化程序设计语言。特别适合于教学，也可广泛用于系统软件与应用软件的开发。PASCAL 程序只能使用规定的基本符号。保留字是一些具有特定含义的词，只能按规定的意义使用。标识符是以字母开头的字母、数字组合。有些标识符具有特定的含义，称为标准标识符。另外，用户可以为常量、变量、类型、函数、过程、文件及程序名选取合适的标识符，注意不要与保留字和标准标识符重名。

一个 PASCAL 程序由程序首部和分程序组成。分程序通常由说明部分和语句部分组成（在特殊情况下可以没有说明部分）。

说明部分可以包括多种类型的说明。各类说明出现的先后次序是有严格规定的，不得违反这个规定。常见的说明是常量说明和变量说明。对于程序中某些值不变的量可以说明为常量。对于那些值可能变化或预先不知道的量可说明为变量。要在变量说明中说明变量的类型。

语句部分由 BEGIN 开始，END. 结束，中间可以包括任意多个由分号(;)分开的语句。本章介绍了三种语句，它们是读语句、赋值语句和写语句。

在编程时无法确定其值的量，可以通过读语句，上机时从终端键盘输入。这样做也增加了程序的灵活性，可以对不同的输入值进行计算。

对于已经确定其值的量或需要计算的量，可以通过赋值语句为其赋值。

写语句用来输出计算结果。有时为了记录输入数据，将输入变量的值也输出。中间计算结果(不是最终计算结果)一般不输出。

习　　题

1.1 判断下列标识符，哪些是合法的？哪些是非法的？

x3，3x，a17，p5q，π，β，ε，abcd，x^2，ex9.5

1.2 输入三个数，计算并输出它们的平均值以及三个数的乘积，写出程序。

1.3 已知地球半径为 6371 km，计算并输出地球的表面积和体积，写出程序。

1.4 已知匀加速运动的初速度为 10 m/s，加速度为 2 m/s^2，求 20 s 以后的速度，20 s 内走过的路程及平均速度，写出程序。

1.5 读入摄氏温度 c，将它转换成华氏温度 f 输出，写出程序。已知：

$$f = \frac{9}{5}c + 32$$

1.6 一个 PASCAL 程序必须包括

A) 程序首部、说明部分、语句部分

B) 程序首部、说明部分

C) 程序首部、语句部分

D)说明部分、语句部分

1.7 下列常量说明中,合法的是

A)
```
CONST
    a :=17.25;
```

B)
```
CONST
    a=2 * 3+5;
```

C)
```
CONST
    a=sin(2.17);
```

D)
```
CONST
    a=21.54;
```

1.8 下列变量说明中,合法的是

A)
```
VAR
    x : =real;
```

B)
```
VAR
    x=real ;
```

C)
```
VAR
    x:real; i:integer;
```

D)
```
VAR
    x:real , i:integer;
```

1.9 下列说明中,合法的是

A)
```
CONST
    a=13.25;
VAR
    x:real;
VAR
    i:integer;
```

B)
```
VAR
    x:real;
    i:integer;
CONST
    a=13.25;
```

C)
```
CONST
    a=13.25;
VAR
```

```
    x:real;
    i:integer;
D) CONST
    a=4.325;
   CONST
    b=31.78;
   VAR
    x,y:real;
```

第 2 章　顺序结构程序设计

2.1　引　言

所有计算机程序都可以用四种基本结构表示。这就是顺序结构、选择结构、循环结构以及函数与过程结构。顺序结构是一组按书写顺序执行的语句。选择结构能根据运行时的情况自动选择要执行的语句组。循环结构允许多次重复执行一组语句。函数与过程结构使得可以通过简单的函数或过程调用执行一组复杂的语句。这四种结构对外界来讲都是一个入口，一个出口。用这四种基本结构编出的程序结构清楚，可读性好。也便于调试和修改。

从本章开始，将用四章分别研究这四种基本结构的程序设计。这四章是本书的核心，掌握了这四章的内容也就掌握了程序设计的基本方法。

本章将研究最简单的结构——顺序结构的程序设计。为此要介绍用计算机解题的基本方法，PASCAL 的标准数据类型(实型、整型、字符型、布尔型)，表达式与赋值语句，读语句，写语句等。并最后给出几个顺序程序设计的例子。

2.2　用计算机解题的基本方法

2.2.1　问题分析

计算机是用于表示和处理数据的工具。因此解题的前两个任务是定义在计算机存储器中表示的数据和描述算法——列出处理这些数据的步骤。这两个任务不是完全无关的。当描述算法时，可以改变数据定义。然而在构造算法前，应尽可能完全和仔细地执行数据定义。如果数据定义有错，描述算法是困难的，甚至是不可能的。

定义数据要求清楚地了解问题。首先必须确定由计算机计算和打印的信息，然后标识输入给计算机的信息。一旦标识了输入输出，我们要问，为了从给定的输入计算输出，是否有足够的信息可利用？如果没有，必须确定附加的信息，以及提供这些信息的手段。

当标识与问题有关的数据项时，要为每一项赋一个助忆名。它可以用来代表存储数据项的计算机存储单元，而不必考虑与每个变量名相联系的实际存储单元。编译程序将为每个变量名赋一个唯一的存储单元。

为了说明问题，我们来看一个例子。

【例 2.1】写一个程序，计算和打印三个数的和及平均值。

讨论：首先应分析和了解这个问题，并标识问题的输入和输出数据。然后确定如何从输入数据得到输出数据的算法。

该问题要求计算和打印三个数的和及平均值。显然，和及平均值是该问题的两个输出项。而为了得到这两个输出项，必须首先输入三个数，这三个数就是该问题的输入项。

用num1,num2 和 num3 标识三个输入数据，其值由读语句输入。

用sum 标识三个数的和,用 ave 标识三个数的平均值。它们的值代表问题所要求的最后结果，可以通过适当的计算得到。

2.2.2 问题解的描述

在对例 2.1 的问题有了清楚的了解后，可以仔细地构造求解步骤——算法。算法可以自顶向下逐步求精。

例 2.1 的一级算法如下：

1. 读数据到变量 num1,num2 和 num3 中,并回打这些数据
2. 计算 num1,num2,num3 的和,存储结果在变量 sum 中
3. 计算 num1,num2,num3 的平均值,存储结果在变量 ave 中
4. 打印变量 sum 与 ave 的值

一级算法只是问题解的一个轮廓。有些问题较复杂，只根据一级算法还难以写出 PASCAL 程序。这时可以对一级算法逐步求精，将它的某些步骤扩展成更详细的步骤，直到可以写出显然的 PASCAL 语句为止。

对于某些程序员是显然的程序语句，对于其他程序员可能不显然。因此，算法求精是因人而异的。当你取得开发算法和将它们转变成 PASCAL 程序的经验时，算法求精的步骤就可以越来越少。

例 2.1 的一级算法只有第 3 步需要进一步求精。

二级求精

第 3 步　求平均值

用求和的项数(3)去除和(sum)

现在可以一步一步地写出实现该算法的 PASCAL 程序了。

程序 2.1　求三个数的和及平均值的程序

```
PROGRAM add(input,output);
    {计算三个数的和与平均值}
  VAR
     num1,num2,num3,sum,ave:real ;
  BEGIN
      {读和打印 num1,num2,num3}
    read(num1,num2,num3);
    write(num1,num2,num3);
      {计算和与平均值}
    sum :=num1+num2 + num3;
    ave :=sum/3;
```

```
    {打印和与平均值}
    write(sum,ave)
END.
```

要注意注释的使用。由‘ { ’字‘ } ’括起来的部分表示注释。有的系统用‘(* ’字‘ *) ”或‘ / * ’字‘ * / ”括起来表示注释。编译程序在编译时将忽略它们。列出注释是为了帮助程序员标识或确认程序体每一步的目的。

每个注释描述了跟在它后面的程序语句的目的。然而，太多的注释可能引起程序的混乱。一个好的经验是用注释标识一级算法中的每一步的 PASCAL 实现,以及要求进一步求精的其它步骤，使得算法和它的 PASCAL 实现之间的对应变得显然。

注释也可以帮助标识在程序段中重要变量的使用。至少应该有一个注释出现在程序的开始，以表明程序的目的。

注释除了给程序编者看以外，也给其它想阅读该程序的人提供了方便。注释可以根据条件与可能用中文或英文写出。

还要注意回打输入数据。在程序 2.1 中回打了 num1,num2 与 num3 的值。回打输入数据可以使你确信输入了正确的值。

归纳问题求解的步骤如下：

(1) 了解问题的需求。特别是要了解该问题已知什么？需要求什么？

(2) 确定计算方法。包括计算公式与计算步骤的确定。

(3) 选择合适的数据结构。即确定数据类型和数据的组织方式。

(4) 设计算法并根据需要，自顶向下，逐步求精。

(5) 编写程序。

(6) 上机调试和执行程序。

(7) 分析结果与总结。

2.3 标准数据类型

PASCAL 最重要的特性之一是它提供程序员各种标准数据类型（实型、整型、字符型、布尔型)和允许程序员自己定义新的数据类型。在 PASCAL 中必须说明每个标识符的类型。

可以在数据项上执行的运算依赖于数据的类型。如果运算和它的数据类型不一致，编译程序将给出错误诊断信息。

2.3.1 实型(real)

实型是最常用的数据类型。

在 PASCAL 中,实数有两种表示方法 小数表示法和指数表示法(或称科学表示法)。

以小数表示法表示的实数例子是：

1.25,132.67,0.0025,—1.56,0.0 ,100.0

以科学表示法表示这些数，可以写成：

1.25e0,1.3267e+2,2.5e-3,-1.56e0,0e0,1e2

在科学表示法中，e 后的数字代表 10 的幂。上列各数可以解释成：

$1.25e0=1.25\times10^{0}=1.25$　　$1.3267e+2=1.3267\times10^{2}=132.67$

$2.5e-3=2.5\times10^{-3}=0.0025$　　$-1.56e0=-1.56\times10^{0}=-1.56$

$0e0=0\times10^{0}=0.0$　　$1e2=1\times10^{2}=100.0$。

在小数表示法中必须有小数点，且小数点前后必须有数字(0～9)。因此 1,.5,23. 都不是合法的 PASCAL 实数。

在科学表示法中必须有 e，且 e 前后必须有数字。因此 25e，e9 都不是合法的 PASCAL 实数。

当实数值太大或太小(按绝对值来说)时，用科学表示法较方便。

合法的 PASCAL 实数可以用图 2.1 的语法图表示。在语法图中，用圆圈表示终结符，方框表示非终结符。非终结符是需要进一步定义的，如数字可进一步定义。

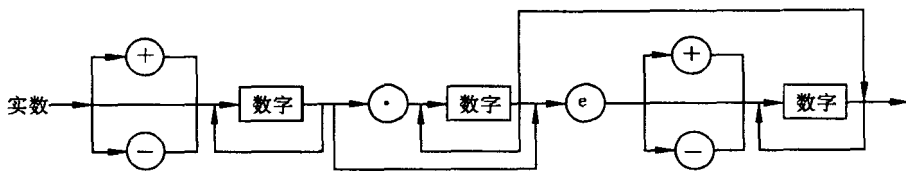

图 2.1　实数语法图

根据以上语法图可以检验一个数是否为合法的 PASCAL 实数。

对于可以取得实型值的常量和变量，可以按下列形式定义和说明。

```
CONST
    pi=3.14159;
VAR
    r,l,s:real ;
```

以上说明 pi 是实型常量，其值为 3.14159。它的值不得在程序中改变。r,l,s 为实型变量。它们可以取得任意实型值。　real 是实型标准类型标识符。

实型量的运算有：

＋ (加) 、－(减)、＊ (乘)、／(除)

例如：

```
3.25+2.17*1.28
pi*r*r
(a+b)/(c+d)
```

在 PASCAL 语言中规定，表达式的计算按先乘除后加减的次序进行。通过加括号，可以改变运算的先后次序。在有括号时，先执行括号内的运算，再执行括号外的运算。此外，表达式必须线性地写出，分子、分母应写在同一行上。乘号(*)必须明确写出，不得省略。下面是几个 PASCAL 表达式的例子：

数学表达式	PASCAL 表达式
b^2-4ac	b*b-4*a*c
$\frac{a+b}{c+d}$	(a+b)/(c+d)
$\frac{a+b}{c}+d$	(a+b)/c+d
$\frac{ab}{cd}$	a*b/(c*d)或 a*b/c/d

用于实型量的标准函数有：

abs(绝对值),sqr(平方),sqrt(开方),sin (正弦),cos(余弦),arctan(反正切),exp (以 e 为底的指数),ln(自然对数),trunc(取整),round (舍入取整)

PASCAL 规定，所有函数的自变量必须写在括号中。例如：

$\sin 2x$ 应写成 sin(2*x)

$(a+b)^2$ 应写成 sqr(a+b)

$\sqrt{b^2-4ac}$ 应写成 sqrt(b*b-4*a*c)

sin,cos 函数的自变量应为弧度。若是度，应转换成弧度。例如：

$\sin 32°15'$ 应写成 sin(32.25*3.14159/180)

在 PASCAL 中没有正切(tan)函数，为了计算正切，可以利用 sin,cos。例如：

$\tan x$ 应写成 sin(x)/cos(x)。

arctan 函数的结果为弧度。

exp 是以 e 为底的指数。因此

$e^{2.5}$应写成 exp(2.5)

ln 是以 e为底的自然对数，以其它数为底的对数，应做相应的转换。因此

$\ln x$ 写成 ln(x)

$\lg x$ 写成 ln(x)/ln(10)

在 PASCAL 中没有通常意义下的指数运算。为了计算 x^y,当 y 为小整数时，可以利用乘法(*)或平方(sqr)计算。当 y 较大或是非整数时，可以利用指数(exp) 和对数(ln)来计算,例如：

x^3 可以写成 x*x*x

$(a+b)^3$ 可以写成 sqr(a+b)*(a+b)

x^y 可以写成 exp(y*ln(x))

这最后一个式子是因为有

$$x^y = e^{\ln(x^y)} = e^{y\ln x}$$

此时，x 必须为正数。

trunc是去掉小数部分，取其整数。round 是将小数部分四舍五入后变为整数（即得到最接近于它的整数）。因此

```
trunc(1.2)=1          round(1.2)=1
trunc(1.7)=1          round(1.7)=2
trunc(-3.7)=-3        round(-3.7)=-4
```

2.3.2 整型（integer）

整型数包括正、负整数和零。例如：

25，−32，0

整数的语法图如图 2.2 所示。

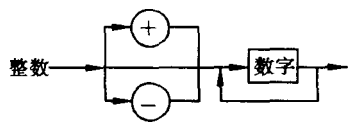

图 2.2 整数的语法图

从整数的语法图可以看出，在整数中不得包括小数点(.)及逗号(,)。例如：

15.0，1.325

都不是合法的 PASCAL 整数。

整型常量和变量可以按如下形式定义和说明。

```
CONST
   long=150;
   wide=65;
VAR
   i,j,k:integer;
```

以上说明 long 和 wide 是整型常量，其值分别为 150 和 65。它们的值不得在程序中改变。

i,j,k 是整型变量。它们可以取得任何整型值。 integer 是整型标准类型标识符。

PASCAL 系统定义了一个特殊的整型常量 maxint。其值为最大整数（不同的计算机系统具有不同的最大整数）。例如，有的计算机最大整数为 32767。用户不必定义，就可以直接使用 maxint。

整型量的运算有：

+(加)，−(减)，*(乘)，DIV(整除)，MOD(取余)

例如：

```
2*i+1
long*wide
```

DIV 的结果是整数商。MOD 的结果是整除后的余数。因此有：

```
8 DIV 3=2          8 MOD 3=2
7 DIV 3=2          7 MOD 3=1
6 DIV 3=2          6 MOD 3=0
```