

第一章 计算机和程序设计介绍

1.1 引言

本章介绍计算机和程序设计的基本知识,包括计算机的发展与应用,计算机组成,计算机语言,PASCAL 介绍等内容。通过这些内容的学习,将使我们对计算机和程序设计有一个初步的了解,为以后各章的学习打下必要的基础。

计算机是程序式电子数字计算机的简称。从世界上第一台计算机诞生到现在不过几十年的时间,计算机已经历了四代,现正进入第五代。计算机已应用到工业、农业、商业、文教、军事、科学和人民生活等各行各业中。

计算机主要由五个部件——存储器、运算器、控制器、输入器和输出器组成。计算机只能按照人编写的程序进行工作。最基本的程序设计语言是机器语言。PASCAL 语言是一种高级程序设计语言,较适合于人的使用。本章将介绍 PASCAL 语言的特点,基本符号,保留字,标识符以及程序结构。

1.2 计算机的发展与应用

1.2.1 计算机的发展

计算机是程序式电子数字计算机的简称。世界上第一台程序式电子数字计算机是美国于 1946 年研制的 ENIAC 计算机。它由 18000 个电子管组成,占地 200m^2 ,耗电 100kW ,每秒运算 5000 次(人工计算每天约 2 千次,它相当于人工计算的 20 万倍)。

从世界上第一台计算机诞生到现在不过 50 年的时间,它已经历了四代,现正进入第五代。

第一代是电子管计算机。年代是 1946~1957 年。平均运算速度是每秒几千次到几万次。支撑软件是机器语言和汇编语言。

第二代是晶体管计算机。年代是 1958~1964 年。平均运算速度是每秒几万次到几十万次。支撑软件是算法语言、管理程序。

第三代是集成电路计算机。年代是 1965~1969 年。平均运算速度是每秒几十万到几百万次。支撑软件是操作系统。

第四代是大规模集成电路计算机。年代是 1970 年以后。平均运算速度是每秒几百万到几亿次。支撑软件是操作系统与数据库。

第五代是智能计算机。现正研究。平均运算速度可达每秒几亿到几千亿次。支撑软件是知识库。

由于采用了不同的电子元器件,新一代计算机比老一代计算机体积大大缩小,耗电量

大大减少，价格大大降低，而运算速度、内存容量、功能却大大提高。

1.2.2 计算机的应用

现在计算机已广泛应用于工业、农业、商业、文教、军事、科学和人民生活等各行各业。按其应用的特点可分为以下几个方面：

(1) 数值计算

例如天气预报，水坝应力分析，卫星轨道，原子反应堆计算等。这些问题都需要进行大量的计算。人工计算是很费时的，也很难得到准确的结果。因此可以利用计算机快速、准确的特点来完成这些计算。

(2) 数据处理

例如地震数据处理，人口普查数据处理，大型运动会数据检索与处理，银行数据处理，程控电话计费系统，工厂、企业、学校数据管理系统等。这些应用需要存储大量数据和进行复杂的比较、判断、处理。因此可以利用计算机的大容量存储和逻辑判断的功能来解决这些问题。

(3) 自动控制

例如炼钢、轧钢、电厂、程控机床、高炮和导弹等的自动控制。这些问题需要实时采集数据，迅速完成计算并及时控制对象作出相应的调整。这正好利用计算机的输入、计算和输出的功能完成其自动控制。

(4) 计算机辅助设计与制造

例如集成电路的设计与制版，机械设计与制造，土木建筑设计等。它们也可利用计算机的输入、计算和输出功能来完成其所要求。

(5) 人工智能

例如语音、文字、图象识别与处理，机器人，医疗诊断，辅助教学，文字翻译，机器证明，下棋，作曲等。它们主要利用计算机的逻辑判断能力来完成这些需要复杂推理的应用。

1.3 计算机的组成

计算机主要由五个部件——存储器、运算器、控制器、输入器和输出器组成。另外为了扩大存储器容量和便于长期保存数据，通常还有外存储器。下面分别介绍这几个部件：

(1) 存储器

存储器是用来存放程序和数据的。为了让计算机算题，必须将初始数据和计算步骤（即程序）存入存储器。存储器由许多小单元（或称为字或字节）组成。每个单元可存放一个数或一条指令（一个程序是由许多指令组成的）。每个单元对应一个编号（称为地址）。每个计算机存储器所包含的单元数（称为存储容量）是不同的。如现在的个人计算机通常包括2兆到8兆字节（1兆是1百万）。从存储器中取出一个数或存入一个数是需要一定时间的，这个时间称为存取周期。存取周期的长短在很大程度上影响了计算机的平均运算速度。个人计算机的存取周期通常为1微秒（1微秒等于百万分之一秒）。从存储器的某个单元取出数以后，原单元中的数仍保持不变，可以再次取出。往存储器某单元中存放一个数，则

该单元中原来的数被新存入的数所取代。

(2) 运算器

运算器是用来对数据进行加、减、乘、除、比较等运算的。计算机中的数通常是以二进制形式存放的，它们的运算也是按二进制运算规则进行的。二进制运算规则较简单，实现起来较方便。进行加、减法较快，乘、除法较慢。计算机的平均运算速度是按加、减、乘、除各占一定比例计算出来的。如个人计算机完成一次运算大约需要 2 微秒。

(3) 控制器

控制器是计算机的指挥中心，由它发出控制信号，指挥计算机各部分协调工作。而控制器的工作又是由程序决定的。程序是由一系列指令组成的。计算机将存储器中的指令逐个送入控制器进行分析，根据分析结果决定：是否要从计算机存储器中取数到运算器进行运算；是否要将运算结果存储到存储器中或从输出设备输出；然后发出相应的信号给存储器、运算器、输入器和输出器。

通常把运算器和控制器统称为中央处理器（CPU）。

(4) 输入器

程序和数据是通过输入器输入到存储器的。常见的输入设备有终端打字机、卡片输入机、纸带输入机等。

(5) 输出器

计算机将计算结果通过输出器输出给人看，并可作长期保存。常见的输出设备有终端屏幕、打印机、绘图仪等。

(6) 外存储器

前面讲的存储器也叫内存储器。内存储器可以直接与运算器打交道。即在运算过程中，可以根据需要直接从内存储器取数到运算器中运算，运算结果也可以立即送回内存储器保存。但是，由于内存储器造价较高，容量不可能做得很大。且内存储器不能长期保存数据，一旦断电，所存数据会丢失。所以，为了扩大存储器容量、降低成本和便于长期保存数据，一般计算机都配有外存储器。外存储器不能直接与运算器打交道。但是可以将内存储器中的成批数据送入外存储器保存。或在需要时，将外存储器中的数据成批送入内存，再参加运算。

常见的外存储器是磁盘、磁带和光盘。它们的容量都很大，而造价较低。如个人计算机的磁盘容量通常为 100~200 兆字节。此外，许多计算机都配有软盘作为外设。软盘可以由用户保存，用时装入，不用时取下带走。这样方便了用户，也加强了保密性。

计算机各部件的关系如图 1.1 所示。

计算机的工作过程大致如下：首先将程序和数据通过输入器送入存储器保存。然后依次将存储器中的指令送入控制器分析，控制器根据分析结果，发出

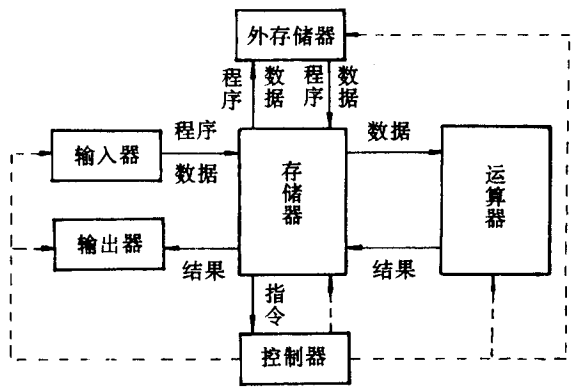


图 1.1 计算机的组成

相应的指示，从存储器取数到运算器运算，运算结果存入存储器。最后将存储器中的结果通过输出器输出。在必要时，可将内存储器的成批数据送入外存储器保存。或将外存储器中的成批数据调入内存储器，以便参加运算。

1.4 计算机语言

计算机是按照人的要求进行计算的。我们必须以计算机所能接受的语言——计算机语言与其通信。告诉它对什么数进行什么运算，以及运算的先后次序。

计算机语言的发展已经历了几代，下面分别加以介绍。

(1) 第一代语言——机器语言

第一代语言是机器语言，它是依赖于机器的。即不同的计算机有不同的机器语言。机器语言程序由许多机器指令组成。在每条指令中要规定作什么运算（由操作码指示）和对哪个单元中的数进行运算（由地址码指示）。此外，数据和指令必须分别放在不同的单元（地址）中。

由于计算机只能存储和识别二进制的数据和指令，所以在机器语言中，每条指令的操作码和地址码都用二进制（或八进制）编码。存放数据和指令的地址也用二进制（或八进制）编码。数据也需预先转换成二进制。所以机器语言也称为二进制语言。

例如计算

$$x = \frac{3.57 + 1.25 \times 4.19}{3.24}$$

某计算机的机器语言如下：

数 据		程 序			
地 址	数	地 址	指 令		注 释
			操作码	地址码	
0100	3.57	0200	021	0101	取 1.25
0101	1.25	0201	012	0102	×4.19
0102	4.19	0202	010	0100	+3.57
0103	3.24	0203	013	0103	÷3.24
0104	x	0204	022	0104	存 x
		0205	035	0104	打印 x
		0206	007	0000	停

计算机可以直接识别和执行机器语言程序，执行效率较高。但是人工编写机器语言程序却较繁琐，易出错。且不同计算机有不同的机器语言，不能通用。

(2) 第二代语言——汇编语言

汇编语言也称符号语言。它是用符号代替机器语言中的二进制编码，这样看起来较直观，不易出错。

例如上例的汇编语言程序如下：

```
K: LDA      1.25
   ×        4.19
   +        3.57
   ÷        3.24
   STA      x
   PRINT    x
   STOP
```

计算机不能直接识别和执行汇编语言程序，它必须经过一个汇编程序（系统软件）转换成机器语言（目标程序）后，才能执行。如图 1.2 所示。

汇编语言仍旧是依赖于机器的，不同的计算机有不同的汇编语言，不能通用。并且汇编语言与机器语言是一一对应的，一个复杂的程序将包括许多汇编语言指令，写起来仍较繁琐。

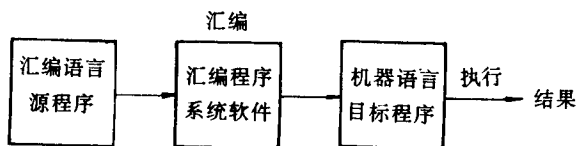


图 1.2 汇编语言的汇编与执行

(3) 第三代语言——算法语言

算法语言是按照算法书写的。一个算法语言程序由许多语句组成。一个语句可以对应多条机器语言指令。所以用算法语言编写程序更加简单、方便、直观、不易出错。另外，算法语言是不依赖于机器的。为一个计算机编写的算法语言程序可以在任何其它类型计算机上运行。所以算法语言也称为高级语言。

常用的算法语言有 BASIC, FORTRAN, ALGOL, COBOL, PASCAL, C 语言等。

如上例，用 BASIC 语言可以写成如下程序。

```
10  x = (3.57 + 1.25 * 4.19) / 3.24
20  PRINT x
30  END
```

计算机也不能直接识别和执行算法语言程序。算法语言程序（源程序）必须先经过编译程序（系统软件）编译成机器语言程序（目标程序）后，才能执行。如图 1.3 所示。

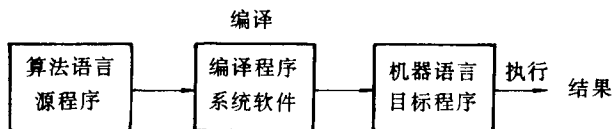


图 1.3 算法语言的编译与执行

(4) 第四代语言——非过程化语言

算法语言是为专业程序设计人员设计的，而不是为非专业程序设计人员设计的。一个复杂问题的求解，仍需要大量的语句行。为了调试它们也需要花费很多时间，要更改这些复杂系统也非常困难，因而不能及时响应用户的需求。

第四代语言旨在解决以上问题，且要实现下述目标：

- 提高应用系统开发的速度。

- 使应用系统能既容易又迅速地改变，从而降低维护费用。
- 将调试工作量降至最低限度。
- 从需求的高级描述中产生无差错的代码。
- 使语言对用户“友好”。非专业程序设计人员也能独自使用计算机解决自己的问题。

第三代语言是过程化语言，它必须描述问题是如何求解的？第四代语言是非过程化语言，它只需描述需求解的问题是什么？

例如需要将某班学生的成绩按从高到低的次序输出。用第四代语言只需写出这个要求即可，而不必写出排序的过程。利用数据库的查询语言就可以做到这一点。所以数据库的查询语言可以看成是最简单的第四代语言。由于第四代语言的效率较高，有时也将第四代语言称为高生产率语言。

(5) 第五代语言——智能化语言

第五代语言主要是为人工智能领域设计的。例如知识库系统、专家系统、推理工程、自然语言处理等。在这些领域内，将复杂的知识进行编码，使得计算机能从中得出推论，使得软件显出较高的智力。

第五代语言还处于萌芽状态。PROLOG 语言可以看作是它的一个例子。

1.5 PASCAL 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。

本节将介绍 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须的知识。

1.5.1 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。

1.5.2 基本符号、保留字、标识符

1.5.2.1 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

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

等都不准在 PASCAL 语言中使用。

1.5.2.2 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,以便正确的使用,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, DOWNT0, 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。

1.5.2.3 标识符

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

 $5x, x-y, \alpha, \pi, \epsilon, \text{ex}10.5$

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。

标识符可用来表示常量、变量、类型、文件、函数、过程或程序的名字。

students1 和 students2

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。

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):

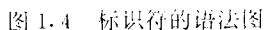
integer, real, char, boolean, text

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。

1.5.3 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打字机的键盘读入。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.
```

以上程序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

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。

从第三行到最后行为程序的分程序。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 1.5 所示。

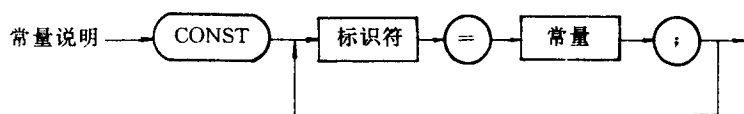


图 1.5 常量说明语法图

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这个数，当要修改这个数时，必须找出所有这些数，将它们逐一改成新数，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。除了实型外，还有整型、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1.6所示。

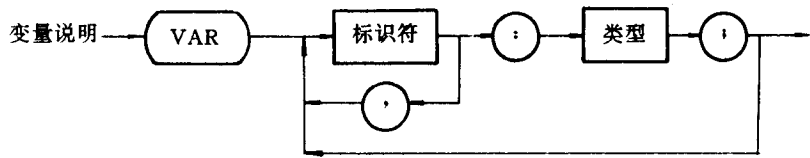


图1.6 变量说明的语法图

从 BEGIN 开始，到 END 结束是语句部分。该程序包括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。乘法必须明确写出，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;
    BEGIN
        语句;
        语句;
        :
        :
        语句
    END.
```

说明部分

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 1.7 所示。

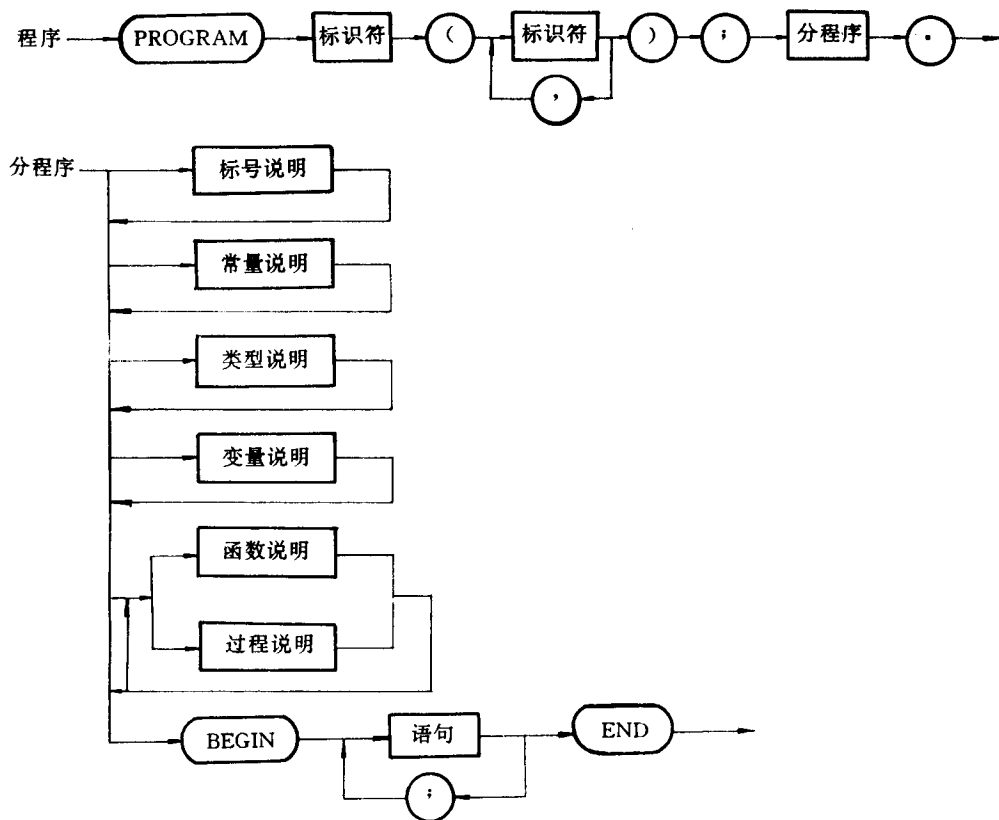


图1.7 PASCAL 程序的语法图

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。

详细上机操作说明见郑启华编著的《PASCAL 程序设计习题与选解（新编）》清华大学出版社，1995年12月。或查看你所使用的计算机的有关上机操作手册。

1.6 小 结

本章介绍了计算机的发展与应用，计算机组成，计算机语言，PASCAL 介绍。重点是 PASCAL 介绍。

在 PASCAL 介绍中，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？哪些是非合法的？

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6371km，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10m/s，加速度为2m/s²，求20s以后的速度，20s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

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

第二章 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“{”与“}”括起来的部分表示注释。有的系统用“(* ”与“ *)”或“/ * ”与“ * /”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，也给其他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 = $1.25 \times 10^0 = 1.25$, 1.3267e+2 = $1.3267 \times 10^2 = 132.67$

2.5e-3 = $2.5 \times 10^{-3} = 0.0025$, -1.56e0 = $-1.56 \times 10^0 = -1.56$

0e0 = $0 \times 10^0 = 0.0$, 1e2 = $1 \times 10^2 = 100.0$ 。