

跨世纪**中学生**计算机学习丛书

丁静 等 编著

PASCAL

语言程序设计



北京科学技术出版社

跨世纪中学生计算机丛书

PASCAL 语言程序设计

丁 静 王宇园 吴再陵
林 晖 陈 越 蒋新儿 编著

834-21

北京科学技术出版社

图书在版编目 (CIP) 数据

PASCAL 语言及程序设计/丁静编著. - 北京: 北京科学技术出版社,
1998.8

(跨世纪中学生计算机学习丛书)

ISBN 7-5304-2195-6

I . P… II . 丁… III . PASCAL 语言-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字 (98) 第 18611 号

北京科学技术出版社出版

(北京西直门南大街 16 号)

邮政编码:100035

各地新华书店经销

腾飞胶印厂印刷

*

787×1092 毫米 16 开本 17.5 印张 430 千字

1998 年 8 月第一版 1998 年 8 月第一次印刷

印数 1—6000 册

定价:20.00 元

编 委 会

主任委员：吕传兴（中国教育学会中小学计算机教育研究会
副理事长）

副主任委员：宋玉升（中国计算机学会普及工作委员会委
员）

编委：（以姓氏笔划为序）

毛国平（吉林省教育学院）

孙殿德（上海市长宁区教育学院）

姚期（上海市第三女子中学）

段青（海南省教委教科所）

陈平（南京市教学研究室）

高晓红（西安市教委教研室）

曾庆宇（北京科学技术出版社）

前 言

本书是为中学生编写的,其主要目的在于能让广大的中学生通过该书的学习,一方面了解计算机语言、程序、软件等知识,另一方面是为了让同学们掌握计算机的一种高级语言、程序设计的基本方法以及用计算机解题的技巧。

本书分为两大部分:第一部分介绍 PASCAL 语言的基本知识、基本语句以及程序设计的基本方法;第二部分介绍程序设计中的常用算法及其算法设计的技巧。青少年朋友们,我们希望通过此书的介绍,能够给你们带来许多有益的知识,开发智力、拓宽思路、激发创造思维,为进一步学习计算机知识打好基础;同时通过学习的迁移,提高发散思维,也能够促进其他知识的学习。

参加本书编写的有丁静、蒋新儿、吴再陵、陈越、王宇园、林晖等,由丁静主编。

编 者

1998 年 4 月

目 录

第一章 PASCAL 语言简介	(1)
§ 1.1 程序设计及描述方法	(1)
一、计算机语言、程序和软件	(1)
二、算法及其描述方法	(2)
§ 1.2 PASCAL 程序的结构	(5)
一、PASCAL 语言特点	(5)
二、PASCAL 程序	(5)
三、PASCAL 语法图	(6)
§ 1.3 PASCAL 语言的基本符号	(7)
§ 1.4 数据类型	(9)
一、数据的类型	(9)
二、标准数据类型	(10)
三、常量与变量	(13)
§ 1.5 标准函数及表达式	(17)
一、标准函数	(17)
二、运算符与表达式	(20)
习题	(23)
第二章 TURBO PASCAL 的基本操作	(24)
§ 2.1 TURBO PASCAL 简介	(24)
§ 2.2 TURBO PASCAL 6.0 系统的安装和启动	(24)
一、TURBO PASCAL 6.0 的安装	(24)
二、TURBO PASCAL 6.0 的启动	(26)
§ 2.3 TURBO PASCAL 6.0 的集成环境及菜单使用	(26)
一、菜单介绍	(26)
二、提示信息行介绍	(27)
三、窗口介绍	(27)
四、在 TURBO PASCAL 系统下初次编程	(28)
五、调试程序	(29)
第三章 PASCAL 语言及程序设计初步	(31)
§ 3.1 程序的三种基本结构	(31)
一、顺序结构	(31)
二、选择结构	(31)
三、循环结构	(32)
§ 3.2 顺序结构的程序设计	(32)
一、PASCAL 语言中的语句	(32)
二、输入、输出语句	(33)
三、综合应用	(39)

§ 3.3 选择结构的程序设计	(40)
一、复合语句	(40)
二、逻辑运算及布尔表达式	(41)
三、选择语句	(44)
§ 3.4 循环结构程序设计	(50)
一、计数循环	(50)
二、当型循环	(53)
三、直到型循环	(56)
四、循环语句的嵌套	(57)
五、关于 GOTO 语句	(59)
§ 3.5 综合应用	(62)
习题	(67)
第四章 简单数据类型	(72)
§ 4.1 枚举类型	(72)
一、枚举类型的定义	(72)
二、枚举类型的性质	(74)
§ 4.2 子界类型	(76)
§ 4.3 类型间的相容性及综合应用	(77)
习题	(80)
第五章 数组	(82)
§ 5.1 数组的定义及使用	(82)
一、数组的概念	(82)
二、数组的类型说明及使用	(82)
§ 5.2 一维数组	(83)
一、一维数组定义及数据引用	(83)
二、一维数组应用举例	(84)
§ 5.3 二维数组	(90)
一、二维数组定义及数据元素引用	(90)
二、二维数组应用举例	(91)
§ 5.4 多维数组	(93)
§ 5.5 字符数组及字符串	(94)
一、字符数组	(94)
二、字符串	(96)
三、TURBO PASCAL 字符串变量	(96)
习题	(98)
第六章 过程和函数	(100)
§ 6.1 过程的定义及调用	(100)
一、定义过程	(100)
二、调用过程	(101)
三、实例	(101)
§ 6.2 函数的定义及调用	(103)

一、定义函数	(104)
二、函数调用	(104)
三、函数的应用	(104)
§ 6.3 全程量、局部变量及作用域	(106)
一、全程变量、局部变量及作用域	(106)
二、函数名、过程名的作用域	(109)
§ 6.4 值形参和变量形参	(110)
§ 6.5 子程序的递归调用和超前引用	(112)
一、子程序的递归调用	(112)
二、子程序的超前引用和间接递归	(116)
习题	(118)
第七章 集合和记录	(120)
§ 7.1 集合	(120)
一、集合的概念	(120)
二、集合的值	(121)
三、集合的运算	(121)
四、集合应用	(123)
§ 7.2 记录	(126)
一、记录的概念	(126)
二、开域语句	(128)
三、带变体的记录	(130)
四、记录的应用	(131)
习题	(134)
第八章 TURBO PASCAL 文件	(136)
§ 8.1 文件概述	(136)
一、文件的概念	(136)
二、文件的主要特点	(136)
三、文件分类	(137)
四、文件类型的定义	(137)
§ 8.2 对文件的操作	(138)
一、适合于所有文件类型的标准过程和函数	(138)
二、正文文件的标准过程和函数	(138)
三、类型文件的标准过程和函数	(138)
四、正文文件 TEXT 的操作	(139)
五、正文文件的定义	(139)
六、正文文件的操作步骤	(140)
七、应用举例	(141)
八、FILE 类型文件的操作	(142)
九、类型文件的应用举例	(143)
习题	(145)
第九章 指针变量及线性表	(146)

§ 9.1 动态存贮	(146)
一、静态存贮和动态存贮	(146)
二、指针类型和指针变量	(146)
三、指针变量的使用方法	(147)
四、指针变量的赋值和操作	(148)
§ 9.2 线性链表结构及操作	(150)
一、链表的基本结构	(150)
二、线性链表的建立	(150)
三、线性链表的插入、删除、求线性表的长度	(153)
四、归并运算	(158)
§ 9.3 双向链表和循环链表	(160)
一、双向链表	(160)
二、循环链表	(161)
习题	(168)
第十章 单元及面向对象的程序设计	(171)
§ 10.1 单元程序设计	(171)
一、单元的定义	(171)
二、单元的结构	(171)
三、单元的使用	(172)
四、标准单元简介	(174)
§ 10.2 面向对象的程序设计	(175)
一、如何设计	(176)
二、如何引用单元说明	(179)
三、综合应用举例	(181)
第十一章 解题的技法	(183)
§ 11.1 栈	(183)
§ 11.2 队列	(188)
§ 11.3 二叉树	(193)
§ 11.4 图	(197)
习题	(202)
第十二章 解题的算法设计	(207)
§ 12.1 什么是算法	(207)
§ 12.2 算法好坏的评价	(209)
§ 12.3 设计算法的常用方法	(212)
一、对应方法	(212)
二、穷举方法	(217)
三、递推递归方法	(227)
四、分治方法	(232)
五、建立数学模型的方法	(238)
§ 12.4 例题选解	(245)
习题	(265)

第一章 PASCAL 语言简介

§ 1.1 程序设计及描述方法

一、计算机语言、程序和软件

做任何一件事都有一定的步骤和过程，例如老师对某同学说：

$\pi = 3.14;$	{ 设定的常数 }
$r = 2;$	{ 半径 r 取值为 2 }
$s = \pi r^2;$	{ 由半径计算圆面积的公式 }
求 s 到小数点后 2 位。	{ 对结果的要求 }

老师对该同学说的内容实际上是老师指挥该学生完成计算圆面积这件工作的步骤，通常叫做程序。这位学生能否计算出要求的面积 s ，取决于该同学能否正确接受老师的程序，并正确完成程序中要求的操作或运算。如果该同学没学过 r^2 表示什么意义，那么就应该把公式 $s = \pi r^2$ 改写为 $s = \pi \cdot r \cdot r$ 。

程序是完成某项工作的步骤，在计算机中，程序是指解决某一问题的一系列命令的有序集合。即由执行程序的对象——计算机能理解并完成的一系列命令组成。

同样一件工作，让不同的对象完成，表达出来的程序在形式上可能差异很大。本书的程序，都是以 IBM PC 计算机为硬件，加上 PASCAL 翻译软件（如 TURBO PASCAL 6.0）为支撑环境。因此程序都必须以 PASCAL 翻译软件为基础来描述。

PASCAL 程序设计语言（简称 PASCAL）描述的程序称为 PASCAL 源程序或简称为 PASCAL 程序。

编写程序的过程通常叫做程序设计或编程。用 PASCAL 描述求圆面积的程序如下：

```
PROGRAM EX1_1 (input, output);    { 程序首部 }
CONST pi = 3.14;                  { 常量说明 }
VAR r, s: real;                   { 变量说明，半径 r，面积 s 取实数值 }
BEGIN
  r := 2;                          { 取 r = 2 }
  s := pi * r * r;                 { 计算圆面积 s }
  WRITELN ('s = ', s: 5: 2)       { 输出结果 }
END.
```

让计算机（软件为 TURBO PASCAL）执行程序输出的结果为： $s = 12.56$

一个外国人能够在中国按照汉语方式工作，并不能说明外国人能够完全接受汉语，实际上许多不懂汉语的外国人是通过翻译来达到工作目的的。

同样计算机能够接受并运行 PASCAL 程序，并不能说明计算机本身使用的语言是 PASCAL 语言（程序设计语言），通常计算机本身只能直接使用机器语言。所谓机器语

言,其实质是电路中电流的强、弱,电压的高、低,电路的断、通;为着叙述方便通常将机器语言用 0 和 1 两个符号的组合来表示,用机器语言表达的程序称为可执行程序(能在机器上直接执行的程序)。用机器语言编程是很繁琐的,所以通常总是用一种学起来容易,对一类问题编程十分方便的程序设计语言编写程序,写成的程序称为相应程序设计语言的源程序。由源程序到可执行程序的对应该转换工作则交给特别设计的、起翻译作用的软件去完成(软件本身通常也是一组可执行程序)。程序设计语言的翻译软件通常有两种工作方式:编译方式(相应的翻译软件称为编译程序)和解释方式(相应的翻译软件称为解释程序)。

编译方式的特点是源程序可以通过编译程序生成与源程序对应的目标程序(可执行程序)。目标程序可以直接在计算机上独立于编译程序运行出结果来。解释方式下,源程序运行的整个过程中,都必须有解释程序共同参与才能获得运算结果。

随着软件技术的发展,目前翻译软件的这种工作方式已趋于一致,被集源程序输入修改、调试运行、编译生成于一体的集成环境所替代,成为多个程序组成的软件系统。

适应于不同用户和不同问题需要的程序设计语言(包括相应的翻译软件)可以列出几百种来,目前使用较多的有:汇编语言(相应的翻译软件称为汇编程序)、C 语言、C++ 语言、BASIC、FORTRAN、COBOL、PASCAL 等等,即使是同一种程序设计语言,因相应翻译软件的功能、研制公司、版本的不同,也有许多细节上差异。例如 PASCAL 常见的有 DOS PASCAL、UCSD PASCAL、TURBO PASCAL 3.0(5.0、5.5、6.0、7.0 for windows 等各种版本),但基本内容则是共同的,即都是在标准 PASCAL 基础上发展的,因此对于初学者来说,首先应学习 PASCAL 基本内容,在熟练的基础上再去查阅相应新版本的手册资料,加以活用就没有困难了。

二、算法及其描述方法

用具体的程序设计语言(如 PASCAL)编程,首先需要的设计工作是,理解问题的要求,弄清由已知数据到结果数据所需要的加工过程或方法。例如由半径 r 求圆面积的 s 的程序中,很关键的一步是公式 $s = \pi r^2$;不知道公式 $s = \pi r^2$,由 r 求 s 的问题,程序设计就会变得十分困难。

独立于具体的程序设计语言,寻找出求解问题的步骤或规则或公式通常称为算法,有了算法再去用具体的程序设计语言描述,困难就会大大减少。因此要掌握编程方法之前,应能够先将算法清楚地表达出来,这是程序设计的重要基本功。

一般说来算法描述可以采用由粗到细的思想方法。

例如求 $s = a_1 + a_1^2 + a_1^3 + a_2 + a_2^2 + a_2^3 + \cdots + a_{10} + a_{10}^2 + a_{10}^3$,求 s 的算法可粗分为如下步骤:

- (1) 输入 $a_1, a_2, \cdots, a_{10}$;
- (2) $s := s_1 + s_2 + s_3$; { $:=$ 表示将右边的式子值赋给 s }
- (3) 输出 s ;

其中 s_1, s_2, s_3 可以看作一个独立的算法(模块)

- | | |
|--|----------|
| $s_1 := a_1 + a_2 + \cdots + a_{10}$ | { 求和 } |
| $s_2 := a_1^2 + a_2^2 + \cdots + a_{10}^2$ | { 求平方和 } |

$$s_3 = a_1^3 + a_2^3 + \cdots + a_{10}^3 \quad \{ \text{求立方和} \}$$

以求和过程为例可将算法描述为：

```

s1.1   $s_1 := 0; i := 1;$ 
s1.2   $i > 10$  时转 s1.6 步, 否则做 s1.3 步;
s1.3   $s_1 := s_1 + a_i;$ 
s1.4   $i := i + 1;$ 
s1.5  转 s1.2 步;
s1.6  结束对  $s_1$  的求值;

```

s1.3 步第一次经过时, “:=” 右边为 “ $0 + a_1$ ”, 结果 s_1 的值为 a_1 ; 第二次经过时, “:=” 右边为 “ $a_1 + a_2$ ”, 结果 s_1 中为 “ $a_1 + a_2$ ”; 第十次经过时, “:=” 右边为 “ $(a_1 + a_2 + \cdots + a_9) + a_{10}$ ”, 结果 s_1 中为 “ $a_1 + a_2 + \cdots + a_{10}$ ”; 第十次经过 s1.4 时 $i := i + 1$, i 值已为 11, 由 s1.5 转 s1.2 步时 $i > 10$ 条件成立, 结果转 s1.6 步结束计算 s_1 。

为了算法描述能更方便地表达成 PASCAL 程序, 可采用 PASCAL 语言中的一些命令 (语句) 来表达算法中的步骤。这里我们做一个简要介绍, 读者可以先简要了解一些 PASCAL 语言的语句的作用。

(1) $x := \langle \text{表达式} \rangle$ 该形式表示将 $\langle \text{表达式} \rangle$ 的值赋给变量 x ; 称为赋值语句。这时使用符号: $:=$ 而不用 $=$, 是为了更进一步强调动作的意义, 以免引起不必要的误解, 例如 $i := i + 1$; 表示将原有的 i 值加上 1 作为新的 i 值, 若写为 $i = i + 1$ 就会产生一种数学上永远不可能成立的式子的感觉。

(2) IF $\langle \text{条件} \rangle$ THEN $\langle \text{语句 1} \rangle$ 表示条件成立时做语句 1
ELSE $\langle \text{语句 2} \rangle$ 条件不成立时完成语句 2, 称为条件语句。

例如 IF $x > 0$ THEN $Y := 1$
ELSE $Y := 0$; 表达了公式 $Y = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$

由于算法设计中一个很重要的思想是模块化, 即将若干个动作可以看作一个整体, 这就引出了 PASCAL 中二个相应的概念: 复合语句和过程。

(3) 复合语句是用 BEGIN 和 END 这一对语句所括住的若干个语句, 即可以将它看作一对括号语句。例如:

$$y = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad z = \begin{cases} 10 & x > 0 \\ 100 & x \leq 0 \end{cases}$$

可表达为二个语句 (语句用分号结束):

```

IF  $x > 0$  THEN  $y := 1$ 
ELSE  $y := 0$ ;
IF  $x > 0$  THEN  $z := 10$ 
ELSE  $z := 100$ ;

```

采用复合语句则可写为一个条件语句:

```

IF  $x > 0$  THEN BEGIN  $y := 1; z := 10$  END
ELSE BEGIN  $y := 0; z := 100$  END;

```

其含义是: 当 $x > 0$ 时, 则做两条赋值命令: $y := 1; z := 10$, 而当 $x \leq 0$ 时做另两条赋值命令: $y := 0; z := 100$ 。

所谓过程是将一组语句看作一个独立的整体加以定义，给以名称，可以引用。例如将
 $y := 1; z := 10;$ 两个赋值语句定义为一个过程可写为：

```
PROCEDURE P1;           { 过程的名称为 P1; }
BEGIN
    y := 1;
    z := 10
END;
```

同样可将 $y := 0; z := 100;$ 定义为过程 P2:

```
PROCEDURE P2;
BEGIN
    y := 0;
    z := 100
END;
```

这时条件语句可写为：

```
IF x > 0 THEN P1
ELSE P2;
```

将程序段定义成过程的好处在于能被程序多处调用而不必重复书写过程体部分已定义的多个语句。当程序中需要调用时，只需写上过程名（可能包括参数）。如果过程类似于函数求值，这时还可以将过程定义为函数。

(4) 重复，求和模块中 s1.3 步重复 10 次是通过 s1.1 中 $i := 1$, s1.2 和 s1.5 实现控制的。这一过程实际上是对 i ，从 1 开始每次增加 1 直到 10 为止，控制 $s_1 := s_1 + a_i$ 重复（10 次）。用 PASCAL 的重复语句过程写为：

```
FOR i := 1 TO 10 DO  $s_1 := s_1 + a[i];$ 
```

这里 $a_1, a_2, \dots, a_{10}$ 已定义成数组 $a[1..10]$ ，表示 a 由十元素组成，这 10 个元素可通过下标 1, 2, $\dots$, 10 来表示，即 $a[1], a[2], \dots, a[10]$ 。这样上述求 $s = a_1 + a_1^2 + a_1^3 + a_2 + a_2^2 + a_2^3 + \dots + a_{10} + a_{10}^2 + a_{10}^3$ 的算法写成 PASCAL 程序就方便了：

```
PROGRAM EX1_2 (input, output); { 程序名为 EX1_2, input, 表示输入从键盘, output 表示输出
                                用显示器, 分别代表标准输入输出文件 }
```

```
CONST n = 10;
```

```
VAR A: ARRAY [1..n] of integer; { 表示下标上限取 10, 存放 10 个数的数组定义 }
```

```
s, i: integer; { i 为控制循环（重复）的下标变量, s 为结果 }
```

```
FUNCTION s1: integer; { 定义 s1 为函数, 函数值为整数 }
```

```
BEGIN
```

```
    s1 := 0;
```

```
    FOR i := 1 TO 10 DO  $s_1 := s_1 + a[i];$  { 求和 }
```

```
END;
```

```
FUNCTION s2: integer; { 定义 s2 为函数, 函数值为整数 }
```

```
BEGIN
```

```
    s2 := 0;
```

```
    FOR i := 1 TO 10 DO  $s_2 := s_2 + a[i] * a[i];$  { 求平方和 }
```

```
END;
```

```

FUNCTION s3: integer;           { 定义 s3为函数, 函数值取整数 }
BEGIN
    s3 = 0;
    FOR i: = 1 TO 10 DO s3: = s3 + a [i] * a [i] * a [i]; { 求立方和 }
END;
BEGIN { 主程序 }
    WRITELN ( 'input a1, a2, ..., a10: ');
    FOR i: = 1 TO 10 DO READ ( a [i] );           { 第 1 步 }
    s: = s1 + s2 + s3;                           { 第 2 步 }
    WRITELN ( 's= ', s)                          { 第 3 步 }
END.

```

§ 1.2 PASCAL 程序的结构

一、PASCAL 语言特点

PASCAL 语言是瑞士苏黎士工科大学的 Niklaus Wirth (沃思) 1971 年发表的, 为了纪念 17 世纪法国著名哲学和数学研究者 Blaise Pascal 而命名为 PASCAL 程序设计语言。

PASCAL 语言充分考虑了算法设计中的自顶向下, 由粗到细的模块化思想, 以及总可以将算法步骤剖解为顺序、分支、重复三种基本结构流程的所谓结构化设计原则, 提供了丰富的数据类型 (描述处理对象) 和清晰的描述模块化结构的语句。使得相应程序书写起来十分自由, 风格优美, 且有紧凑易读、翻译效率高等优点, 近十几年来已成为编程课程中的必修科目, 也是编程竞赛和算法描述研究成果进行交流的首选语言。

由于编程技术研究的进展, 结构化程序设计方法、标准子程序库、面向对象的程序设计方法等等, 都不断地在新版本的 PASCAL 语言中得到扩充, 如果掌握了 PASCAL 的基本内容, 将为进一步学习其他程序设计语言和不断学习新的程序设计方法打下良好的基础。

二、PASCAL 程序

从上一节的 PASCAL 程序 EX1_1 和 EX1_2 可以知道, 一个 PASCAL 程序由三部分组成。

(1) PROGRAM 开头的第一行称为程序首部。程序首部固定地由保留字 PROGRAM 引导, 空一格后跟着程序名称和用圆括号括住的程序参数表, 即调用的文件名, 最后以分号结束。

这一部分的作用主要是标识程序, 程序名是用户给自己程序加的名字。凡由用户命名程序中所涉及处理对象时的名称如: 程序名、变量名、类型名、常量名、过程名等等我们都称之为标识符。所命名的标识符要求以字母开头, 其后跟字母或数字组成 (TURBO PASCAL 还允许使用连字符), 如下面 EX1_1, EXP001, BirthDate 等都是允许的标识符, 3rdRoot, TWO Word 等则是不允许的标识符。因为前者数字开头, 后者中间有空格, 这都是不允许的。虽然标识符的长度对具体翻译软件来说总是有限制的, 但用户使用时超长的情况是很少的。命名标识符时需要注意的是应该有一定的表意性, 不要随心所欲, 以

方便别人或自己一段时间后回过头来阅读和理解程序，或对程序纠错、修改。程序名后圆括号中的参数，主要用来列出程序调用的文件名，通常以标准输入输出设备（DOS 系统规定标准输入设备为键盘、标准输出设备为显示器）作为输入输出时，取标准文件名 input 和 output。对 TURBO PASCAL 来说，这两个参数完全可以不写，甚至整个程序首部都可以省略。

(2) PASCAL 程序的第二个部分是说明部分，说明部分要求列出程序中引用的全部常量、变量、转移标号、类型、过程和函数等有关说明，这样做的目的是要求用户能正确把握处理对象——数据的特征和对它们进行的操作保持一致，例如操作 $x := a + b$ ；若变量 x 在说明部分没有说明，翻译软件便能指出其错误并提醒用户加以改正。

通常说明部分说明的内容有：标号说明；常量说明；类型说明；变量说明；过程和函数说明。

(3) 程序的第三个部分是用 BEGIN 和 END 括住的一串语句，称为程序体。程序体是加工数据的主体部分，对定义为过程或函数的模块，程序体中通过过程名和函数名（有时还需给出相应参数值或变量名）加以调用。对应程序体结束的 END 后有一圆点。由于程序体是由顺序书写的一串语句构成的，语句与语句之间以分号隔开，语句本身又可能是复合语句，因此程序体内部 BEGIN END 可能嵌套着多个层次，书写程序时应尽可能将它们之间的配对关系明了地表达清楚。

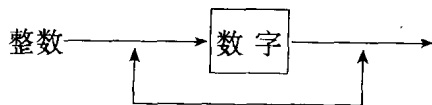
这样，一个 PASCAL 程序的结构可归纳如下：程序首部；说明部分；程序体。

说明部分包括：

标号说明；	{ 指明程序体中用于转移目的地的标号，如 GOTO 8, 8 是标号 }
常量说明；	{ 定义程序中使用标识符表示常量，如 $\pi = 3.14$; }
类型说明；	{ 定义用于变量说明中的类型标识符，如 DIM = ARRAY [1..10] of integer; }
变量说明；	{ 定义程序体中使用变量的类型，如 A: DIM; }
过程和函数说明；	{ 定义作为独立程序的模块，它可能被程序体中的过程语句或函数调用。 }

三、PASCAL 语法图

用 PASCAL 语言将算法描述成 PASCAL 程序，且能上机运行出结果来，必须按照 PASCAL 语言规定的语法规则使用语句（用于说明部分和程序体中的命令）。为了把语言中涉及的概念（语法成份）表达得更明确，PASCAL 常常采用语法图的形式描述，例如整数是一个语法成份，它由一个或多个数字组成，用语法图描述为：



矩形框中的数字本身也是一个语法成份，可以进一步叙述为 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; 这里 0~9 已经是 PASCAL 语言的基本字符了。用语法图可表示为下图。

可见语法图只是帮助理解语言成份的一种工具，对判定和纠正程序中的错误能起一定的作用。语法图并不是表达 PASCAL 语言规则（语法）的唯一方法，例如数字这一概念

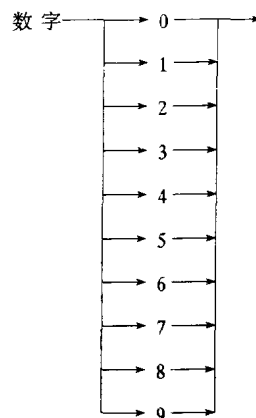
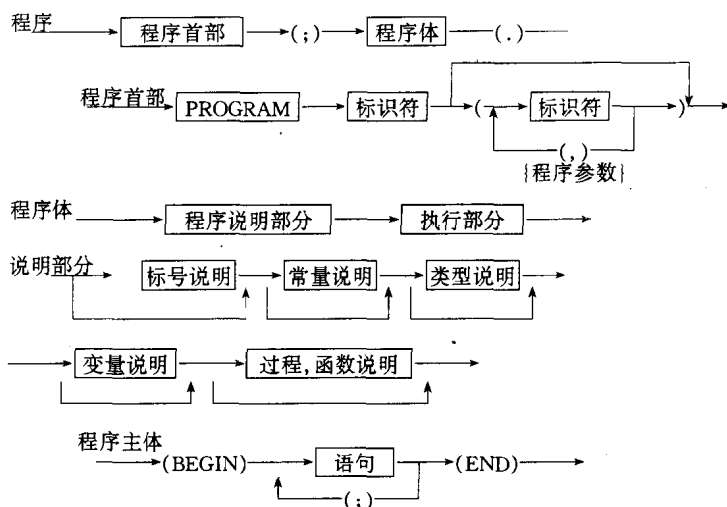
表达为:

* 数字指 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 之一;

* $\langle \text{数字} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

也同样十分清楚明了 (后者用 $::=$ 表示定义为, $|$ 表示或, 这种表示方法又称为 BNF——巴科斯范式)。

我们可以用语法图来描述程序的组成。



通过语法图我们可以更进一步理解 PASCAL 程序的结构了。在后面的章节中, 我们还将一一介绍有关 PASCAL 语言的一些语法图。

§ 1.3 PASCAL 语言的基本符号

PASCAL 程序中可使用的基本符号有如下几类:

1. 数字、字母

大写字母 A~Z, 小写字母 a~z

数字 0~9

2. 运算符、专用符

运算符 + - * / = ()

特殊符 [] { } . , ; ' \$

赋值符号 : =

关系运算符 > = 代表 $\geq$, < = 代表 $\leq$, < > 代表 $\neq$, =, >, < ,

子界符号 ..

注解符号 (* , * , 或用 { })

这里有的基本符号是采用双字符作为一个整体使用的, 拆成两个字符便失去原有的意义。

3. 保留词

保留词是 PASCAL 选定具有固定意义和用法的专用单词或缩写, 在程序中不允许作规定意义以外的使用, 使用时用大写还是小写则不影响其意义, 例如 DIV 或 div (写成 Div 也没关系) 则表示两个整数相除, 结果也取整数的除号。标准 PASCAL 使用的保留词有:

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

forward (也有把这一单词称为指示词的)

4. 标准标识符

程序中有些经常引用的处理对象, PASCAL 为了减少在程序中重复说明, 规定了一些标准标识符, 它们都有约定的意义, 可以直接在程序中使用, 不必作解释。常见的标准标识符有 (使用时大小写无关):

标准常量:	false	true	maxint		
标准类型:	integer	boolean	real	char	text
标准文件:	input	output			
标准函数:	abs	actan	chr	cos	eof
	elon	exp	ln	odd	ord
	pred	round	sin	sqr	sqrt
	succ	trunc			
标准过程:	get	new	dispose	pack	page
	put	read	readln	reset	rewrite
	unpack	write	writeln		

随着不同版本的 PASCAL 翻译软件的发行, 保留词和标准标识符都会有许多补充, 进一步使用时可参考有关手册。

5. 标识符

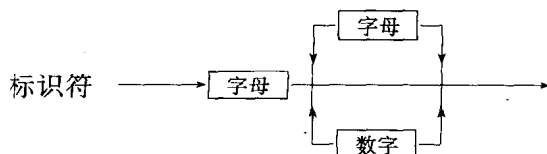
标识符是用来标识处理对象的符号名称, 如: 对常量、变量、类型、过程和函数、程序、文件等。标识符是编程者命名的, 自由度很大, 不过命名中应尽量注意如下约束:

不要使用保留词或标准标识符作为标识符, 这样做会引起程序含义的混淆;

命名的标识符应考虑其他人阅读程序或自己阅读或修改的方便, 若能从标识符本身即可体现出所代表对象的特征, 是较为理想的。如常量说明:

CONST pi = 3.14; { 一看 pi 即可联系到 π }

标识符规定要以字母开头, 后面限以字母或数字, 其语法图可表示为:



一般地, 一个具体的翻译软件对标识符的长度都是有限制的, 通常能识别前 8 个字符, 因此标识符也不能用得过长。

有些版本的 PASCAL 标识符中还允许使用连字符 (-), 例如 TURBO PASCAL 便是