

## 前 言

计算机的使用离不开操作。但真正使用计算机,仅会操作还显得不够,还需要掌握一些使用计算机的手段。例如:编制一个程序,让计算机执行你的程序,完成你所希望做的事情。计算机语言是一种常用的工具,为你提供了指挥计算机工作的手段。本书专为你介绍 PASCAL 语言。

本书与丛书中的其他部分一样,是为初学电脑者编写,其最大的特色是配有相应的学习软件。你可以边学边练,很轻松地掌握 PASCAL 语言,迅速地学会 PASCAL 程序的编制。

本书的起点很低,适用的读者面很广。只要你有一台装有 DOS 系统的 IBM PC 或兼容机(至少配置有单色图形显示卡或 EGA 卡),你就可以通过阅读本书及利用配套软件(运行 CAI 文件)很轻松地学会 PASCAL 程序设计。

你可能在书店里曾经看到过有关介绍 PASCAL 语言的书籍,这些书都很厚,给你的印象可能是:PASCAL 语言难掌握!其实不然!你现在拿着的书不厚,但它包含了

PASCAL 语言的基本内容,用这些内容编制一个常规程序已经丝毫没有困难了。对于那些属于锦上添花的内容,等你对 PASCAL 入门后,也就不难掌握了。

本书及配套软件的主线是 PASCAL 编程。对每一个语句、每一种结构,总是从实际例子出发,告诉读者这些语句或结构在程序中的作用。在有些章节中,对编程中的常见错误及时地给你提醒。学习软件能达到速成的目的,属于基本内容的单元后面都配有“自我练习”,通过它可以检验你的学习效果、对基本语句及结构的理解程度。除了编程训练外,在软件中也适当地安排有输入 PASCAL 程序及运行它的机会。但必须提醒读者,本书及配套软件中所涉及到的 PASCAL 程序的运行过程,是在 **TURBO PASCAL** 版本下进行的。

作者相信,学完本书及配套软件后,你一定会喜欢 PASCAL 语言的!

由于作者水平所限,本书及配套软件中难免有不妥及错误之处,恳请读者批评指正。

作者

1994 年 8 月

# 第一章 PASCAL 基本概念

## 1.1 PASCAL 语言特点

各种程序设计语言是人们使用计算机解决实际问题的工具,它们都有自己的特点以及适用范围。

那么 PASCAL 语言有什么特点呢?

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波兰数学家 Pascal。

PASCAL 语言是体现结构程序设计的第一个语言。可以说,PASCAL 语言的建立是程序设计语言发展史中的一个里程碑。

首先,PASCAL 语言具有较丰富的数据类型,简明的通用语句,清晰的程序结构。因此,这种语言的表达能力比较强,程序的可读性比较好,特别适合于程序设计教学。有人称这种语言为“学院式语言”。

其次,PASCAL 语言的规模比较小,语法严谨而简练,程序的连贯性比较好。有人形容用 PASCAL 语言编

制的程序为精致的艺术品,给人以“风度优美”的感觉。

另外,PASCAL 程序的书写格式比较自由,编译过程也比较紧凑,完全解除了初学者的烦恼感。并且,PASCAL 语言的应用也很广,它适用于教学、管理,编写各种系统软件和进行科学计算。也就是说,PASCAL 语言是一种没有应用偏向的语言。

由于以上特点,PASCAL 语言越来越博得人们的好评,应用也日益广泛。当读者学完了这种语言后,一定会喜欢上这种语言的。

如果读者在学习 PASCAL 语言之前已经学会了另外的一种或几种程序设计语言,例如 BASIC、FORTRAN、C 语言等,那么在开始学习 PASCAL 语言时,可能会感到在 PASCAL 语言中缺少了些什么。例如:PASCAL 语言中没有乘幂运算符,标准函数也很少等。在此有必要提醒读者,请你不要埋怨 PASCAL 语言缺少了什么,因为这是设计者有意删除的。只有删除了不十分必要或能用基本功能实现的那些功能,才能保证编译程序紧凑而高效。就是因为 PASCAL 语言中删除了这些功能(但你完全可以用其它基本功能来代替),才使得这种语言小巧、简洁,编出的程序比较精致。

## 1.2 从一个实例看 PASCAL 程序结构

下面是 PASCAL 程序的一个实例。这个程序的功能很简单：从标准输入设备（即键盘）输入两个整数，然后计算它们的和，最后在标准输出设备（即显示屏幕）上输出计算结果（即输入的两个整数的和）。

```
PROGRAM SIMPLE(INPUT,OUTPUT);  
{THIS PROGRAM IS USED FOR S=X+Y}  
VAR  
    X,Y,SUM:INTEGER;  
BEGIN  
    READ(X,Y);  
    SUM:=X+Y;  
    WRITELN('SUM=',SUM)  
END.
```

这个程序可以分为三部分：

### 1. 程序首部

这个程序的前两行就是程序的首部。程序的首部是一个程序的开头部分，它用来提供程序的主要特征。程序首部一般由四个小部分组成：

#### (1) 程序的标志

PASCAL 语言规定，PASCAL 程序一律以 PRO-

GRAM 开头,即以 PROGRAM 作为 PASCAL 程序的标志。

### (2) 程序的名称

程序名称是用户为程序起的名字。在本程序中,程序名称为 SIMPLE。

程序名称虽然可以由用户自行定义,但为了使程序名称更能够反映程序的功能,反映程序的编制者或与其它程序的联系等信息,最好使程序名称形象一些。本程序的名称 SIMPLE 表示是一个例子。在本书与配套软件中,后面的一些例子中的程序名称将用编号表示,如 PROG1、PROG2、…分别表示程序 1、程序 2、……。

程序标志 PROGRAM 与程序名称之间必须用一个空格隔开。

### (3) 程序的参数

程序参数用来表示程序与外界的联系。最常用的程序参数为 INPUT 与 OUTPUT,它们表示程序中有输入与输出操作。这两个程序参数一般可以省略(在以后的所有程序中,这两个参数都将省略)。一般来说,程序参数是文件变量名。INPUT 与 OUTPUT 是 PASCAL 语言已经定义了的用于输入与输出操作的文件变量。

程序参数用一对圆括号括起来,各参数之间用逗号“,”隔开。当程序参数省略时(默认为 INPUT 与 OUTPUT),一对圆括号也就不需要了。

程序参数后面(若程序参数缺省,则在程序名称后

面)必须有分号“;”。

#### (4) 程序的注释

本程序中的第 2 行为程序的注释。程序的注释对程序的执行不起作用,它只起注释作用。注释部分必须用一对花括号“{”与“}”括起来,注释的内容一般包括程序的名称、类型、主要功能、编制者、编制日期等,总之,凡是需要对程序作说明的信息,都可以放在注释部分。

对于一个复杂的程序,加上注释部分是很有好处的,可以提高程序的可读性,对以后阅读、修改程序十分有利。一个程序加上注释后,也便于交流。

事实上,PASCAL 程序的注释不仅可以出现在程序首部,在程序的任何位置都可以加注释。但别忘了,注释部分一定要由一对花括号括起来。

### 2. 程序说明部分

本程序的第 3、4 行是将程序中用到的 3 个变量 X、Y 与 SUM 说明为整数类型的变量。其中 VAR 是说明变量的关键字。

一般来说,PASCAL 程序说明部分包括以下几个方面:

标号说明——对程序中用到的所有标号进行说明;

常量说明——对程序中用到的所有常量进行说明;

类型说明——由用户定义新的数据类型;

变量说明——对程序中用到的所有变量进行说明  
(说明变量的类型);

过程说明——由用户定义过程；

函数说明——由用户定义函数；

在本程序中只用到了变量说明。

### 3. 程序执行部分

本程序的第 5 行到第 9 行是程序的执行部分，共有三个语句：

通过读语句 `READ(X, Y)` 由键盘输入两个整数 `X` 与 `Y`；

通过赋值语句 `SUM := X + Y` 计算两个整数 `X` 与 `Y` 之和，然后赋给整型变量 `SUM`；

通过写语句 `WRITELN('SUM=', SUM)` 输出计算结果。

PASCAL 语言规定：

(1) 程序执行部分必须包括在 `BEGIN` 与 `END` 之间；

(2) `END` 后必须用小数点“.”作为整个程序的结尾；

(3) 每两个语句之间要用分号“;”隔开。

从上面这个简单的例子可以看出 PASCAL 程序的概貌。

PASCAL 程序的一般结构如下面的方框中所示。

其中前两行为程序首部，第 3 行到第 8 行为程序说明部分，最后三行表示程序执行部分。



```
PROGRAM 程序名(程序参数表);  
    { 注 释 }  
    LABEL          标号说明;  
    CONST          常量说明;  
    TYPE           类型说明;  
    VAR            变量说明;  
    PROCEDURE     过程说明;  
    FUNCTION       函数说明;  
    BEGIN  
        执行语句部分  
    END.
```

最后,提醒读者注意以下几点:

(1) 在程序说明部分中,各说明部分的次序不能颠倒。例如,类型说明必须在常量说明之后,又必须在变量说明之前。但允许某些说明部分没有。例如,在前面的程序 SIMPLE 中,只有变量说明,而其它的说明都不需要。

(2) 说明部分中的每个说明后面必须有分号“;”。例如,在以下的变量说明中:

```
VAR  
    X,Y,SUM :INTEGER;  
    P,Q :REAL;
```

有两个说明:一是说明变量 X,Y,SUM 为整型;二是说

明变量 P, Q 为实型(有关 PASCAL 的数据类型在本书的后面会介绍)。每个说明后面都应有分号“;”。

(3) 程序的执行部分必须用 BEGIN 与 END 括起来,并在 END 后面用一个小数点“.”作为整个程序的结尾。

(4) 在执行部分中:

允许一个语句写在几行上,

也允许在一行上写多个语句,

但两个语句之间要用“;”分隔。

必须注意的是,执行部分最后的 END 不是语句,因此,在 END 前面的一个语句后面不应有分号“;”。

(5) PASCAL 程序的书写格式很自由,而且字符的大、小写是等价的。但在实际编写程序时,为使程序结构清晰,应采用一种好的格式,以提高程序的可读性。

## 1.3 PASCAL 符号

### 1. PASCAL 字符集

PASCAL 语言中,允许使用的基本符号,即 PASCAL 字符集包括以下三类:

(1) 26 个英文字母;

(2) 10 个数字符号 0~9;

(3) 特殊符号 22 个

+ - \* / := ↑ =

< > <= >= < > ( )  
[ ] { } , ; . ' /

## 2. PASCAL 标识符

PASCAL 标识符是以字母开头,后跟若干个字母或数字。例如:

X, SUM, SMP05, CTL16, A12B

等都是合法的 PASCAL 标识符;而

1A, A+B, P \* Q, L. 4

等都是非法的 PASCAL 标识符。

在 PASCAL 语言中,一般只识别前八个字符作为标识符。例如:

AB12EFCDG 与 AB12EFCDHKL

都被识别为同一个标识符 AB12EFCD。

通常,标识符是用来表示程序名、过程名、函数名、类型名、常量名、变量名等。

在使用标识符时,要提醒读者注意以下两点。

(1) 保留关键字不能作为用户定义的标识符,因为它们 PASCAL 语言中有特殊规定的意义,它们有的是 PASCAL 运算符,有的是 PASCAL 程序、过程或函数的标志,也有的是 PASCAL 语句定义符。

在 PASCAL 语言中,共有 35 个保留关键字,它们

是:

AND	ARRAY	BEGIN	CASE	CONST
OR	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

(2) 标准标识符在 PASCAL 语言中也有特殊规定的意义,同样不能作为用户定义的标识符。标准标识符是指 PASCAL 语言已经定义了的标准常量、标准类型、标准文件、标准函数与标准过程。

在 PASCAL 语言中,共有 40 个标准标识符,在后面的学习中,读者会认识其中一部分的意义。40 个标准标识符是:

标准常量

FALSE      MAXINT      TRUE

标准类型

BOOLEAN      CHAR      INTEGER      REAL  
TEXT

标准文件

INPUT      OUTPUT

标准函数

ABS    ARCTAN    CHR    COS    EOF    EOLN  
EXP    LN        ODD    ORD    PRED    ROUND  
SIN    SQR        SQR    SUCC    TRUNC

标准过程

DISPOSE    GET        NEW        PACK      PAGE

PUT            READ        READLN       RESET        REWRITE  
UNPACK        WRITE        WRITELN

## 1.4 TURBO PASCAL 操作

与所有程序设计语言一样,用 PASCAL 语言编写的程序,也要经过编译、链接后才能运行!负责将 PASCAL 程序翻译成计算机能够运行的程序的工具是 PASCAL 编译系统。

下面以 TURBO PASCAL 版本为例说明 PASCAL 程序的编译、链接与运行过程。

假设在计算机系统中已经装有 PASCAL 编译系统,1.2 节中的 PASCAL 程序 SIMPLE 已经输入进计算机,它的文件名为 P1.PAS。则对这个 PASCAL 程序进行编译、链接只要输入以下命令:

```
TPC ␣ P1.PAS ␣
```

此时, PASCAL 编译系统就对文件名为 P1.PAS 的 PASCAL 程序进行编译、链接,最后生成可执行文件 P1.EXE。在进行编译、链接过程中,屏幕上将显示如下信息:

```
Turbo Pascal Version 4.0 Copyright (c) 1987 Borland International
```

```
P1.PAS(9)
```

```
9 line, 0.4 seconds, 2384 bytes code, 591 bytes data.
```

其中第 1 行是告诉你 PASCAL 编译系统的版本,现在用的是 TURBO PASCAL 4.0 版本;第 2 行是告诉你现在编译的 PASCAL 程序的文件名为 P1.PAS,共有 9 行;第 3 行告诉你源程序有 9 行,编译的时间为 0.4 秒,2384 字节代码,591 字节数据。

如果在编译过程中发现有错误,则还显示出错信息。

在此要说明两点:

- PASCAL 程序的文件名后缀为 .PAS!
- 不同的编译系统版本其编译链接的操作可能不同!

要运行这个程序,只要执行可执行文件 P1.EXE,即输入以下命令:

P1 ↵

程序就执行。对于本例中的程序,运行过程中要等待你从键盘输入两个整数。如果你输入 1 2 ↵ 3 4 ↵

则在显示屏幕上输出结果:

SUM=46

由上述例子可以看出,PASCAL 程序的编译、链接是很简单的,只要输入一条命令就可完成编译、链接过程,这也是 PASCAL 语言的一个优点。

在配套的软件中,将为你提供 PASCAL 程序编译、链接、运行的操作机会。

## 第二章 PASCAL 标准数据类型

### 2.1 PASCAL 数据类型概述

1.1 节中曾经提到,PASCAL 语言的优点之一是具有丰富的数据类型。根据 PASCAL 数据类型的特点,可以将它们分为三大类。

#### 1. 简单的数据类型

简单数据类型的构造比较简单,因此也称为“非构造型数据类型”(又称“纯量数据类型”)。它又包括系统已经定义的四种标准数据类型以及用户定义型。其中

四种标准数据类型包括:

- 整数类型(INTEGER)
- 实数类型(REAL)
- 字符类型(CHAR)
- 布尔类型(BOOLEAN)

用户定义型有两种:

- 枚举类型
- 子界类型

本章主要介绍四种标准数据类型。

## 2. 结构型数据类型

结构型数据类型又称为“构造型数据类型”。这种数据类型的构造比较复杂,通常是由其它数据类型按照一定的规则构造而成。常用的四种结构型数据类型有:

- 集合类型(SET)
- 数组类型(ARRAY)
- 记录类型(RECORD)
- 文件类型(FILE)

其中由 10 个字符组成的数组被定义为标准的数据类型 ALFA;由字符组成的文件被定义为标准的数据类型 TEXT。

## 3. 指针型数据类型

这种数据类型使用比较灵活,主要用于解决动态数据的建立、删除和使用等问题。

PASCAL 数据类型的上述分类如图 2.1 所示。

本书是 PASCAL 入门,只介绍第一类的简单数据类型以及第二类中的数组类型。其它数据类型在对 PASCAL“进阶”时再介绍,或读者在学完本书后再进一步学习其它有关 PASCAL 的书籍。

## 2.2 整数类型

在 PASCAL 语言中,整数类型的数据包括正整数、



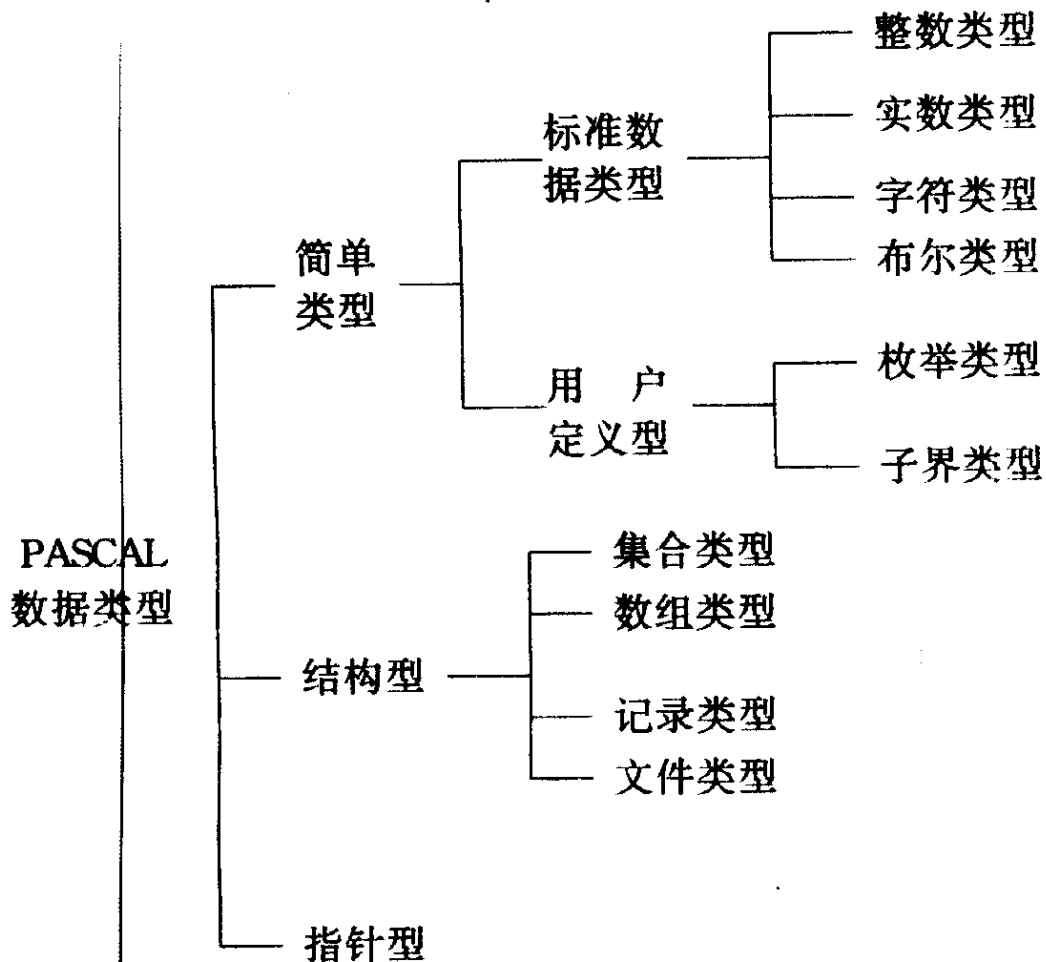


图 2.1 PASCAL 数据类型

负整数与整数零。在一般情况下,如果没有特殊说明,PASCAL 语言中的整数都用十进制表示。在本书中,所有的整数都用十进制表示。因此,PASCAL 整数应符合以下规定:

- 只能由符号+(正)与-(负)以及 10 个数字(0~9)组成;
- 在数字序列之间以及正、负号与左边第一个数字之间不允许出现其它字符;