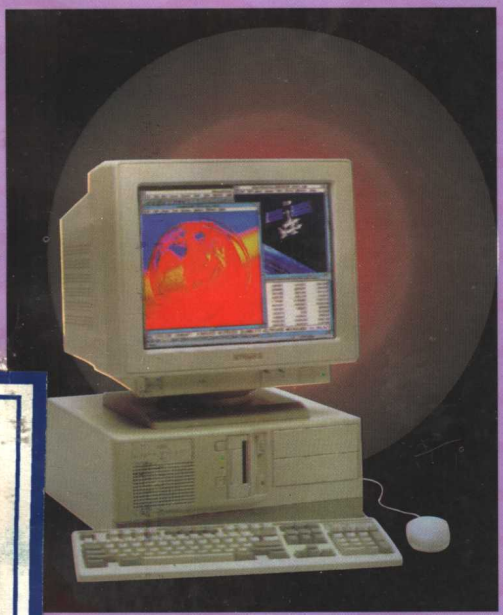
最新出版

中学生计算机自学丛书

跟我学PASCAL

蒋新儿　王晓敏　吴再陵　潘宏辉　编著



面向中学生
基本知识与实际操作

- Pascal语言的基本语句
- 自定义数据类型
- 数组、函数与过程
- 集合与记录
- 指针变量
- 程序设计方法
- 常用算法介绍

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

跟我学 PASCAL

蒋新儿　王晓敏
吴再陵　潘宏辉　编著

電子工業出版社

内容提要

本书是面向中学生讲PASCAL 语言。目的是为使高年级中学生学习和掌握一种模块化的程序设计方法，增强编程能力，为进一步学习计算机技术打下良好的基础。全书共分 10 章，内容包括：Pascal 语言的基本语句，数据类型，函数与过程，集合与记录，Turbo Pascal 文件，程序设计方法及常用算法等。书中提供了大量的编程实例与指导，每章后留有习题作为复习。

(中学生计算机自学丛书)

跟我学 PASCAL

蒋新儿　王晓敏　吴再陵　潘宏辉　编著

责任编辑：宋玉升

*

电子工业出版社出版（北京市万寿路）

电子工业出版社发行　各地新华书店经销

北京市顺义李史山胶印厂印刷

*

开本：850×1168 毫米　1/32　印张：8.5　字数：250 千字

1996 年 11 月第一版 1996 年 11 月第一次印刷

印数：6000 册　定价：10.50 元

ISBN 7-5053-3613-4/TP・1481

序　言

我最近在思考一个问题：引导青少年学电脑的最好形式是什么？听老师讲一些入门知识，参加一个学习班，有时很必要。但是，要想进一步学习一些比较深入的内容时，又该怎么办？我认为，自学可能是一个好办法。有人会问：电脑高深莫测，自学能懂得了吗？半个世纪以来电脑自始至终罩着一层神秘的面纱，许多介绍电脑学问的书籍也都属于"天书"，一般人很难问津。这样，就存在两个问题：一是，学用电脑可不可以通过自学？二是，怎样才能自学？

计算机科学是一门学科，和数学、物理一样。数学、物理可以自学，计算机自然也可以自学。它与其它学科不同的是，电脑是实践性很强的学科，不亲自动手上机实践是学不会的；与其它学科相比，所学的东西和要用于实践是统一的，这就是它便于自学的独具的优势，可以在计算机上"边做边学"。这种"边做边学"的学习方式效果是最好的。中国古代哲人在概括学习规律时曾提出：听而易忘，见而易记，做而易懂（I hear, and I forget. I see, and I remember. I do, and I understand）。的确是这样，你越是不敢去接触，越会感到它神秘；你在自学中多上机实践，你就会感到：学会不难，深造也办得到。

自学就要有一本便于自学的书，不是所有的书都能适合于自学，这就要有人策划和组织专家来编写。电子工业出版社与"全国中小学计算机教育研究中心"联手推出的这套丛书，作者们大多是教学第一线的教师，有丰富的教学经验，都力图按照便于读者自学的思路来撰写。这套丛书的特点是：主题突出，编排合理，深入浅出，便于自学；既有基础知识的讲解，又有上机操作的指导，把知识的传授和实践结合起来；书的内容尽量反映出计算机科学技术最新的发展成就。

这套丛书有很多本是涉及计算机语言的。现在有一派意见认为：学电脑不必学程序设计语言。那么，仅仅学几个软件的使用，会敲敲键盘，是否就能驾驭电脑了呢？电脑是"人类通用的智力工具"，它将不分国界，为千千万万的人们所使用，改变着人们的生活方式和工作效率，并逐步成为一种文化，即所谓的"电脑文化"。如果说是文化，那么文化的一个重要特征是对语言的重构与再生。电脑的语言从形式上到逻辑上都不同于人类的自然语言，它带有语言的重构与再生的特点。文化是需要传播的，电脑技术的发展，特别是电脑网络技术使文化的传播更及时更快，彻底改变了空间对人的

约束。人类的创造性思维活动可以通过计算机的语言传给电脑。由电脑的强大的运算功能产生更多的思维成果，帮助人类认识世界和改造世界。显而易见，你要让电脑为你工作，你就要与电脑“对话”，要对话就要懂得电脑语言。这是顺理成章的事情。有的人一听要学语言就有点怕，其实并不难学。北京人初到上海，一句话也听不懂，呆上一年半载，在那种语言环境下很快就能懂了。学电脑语言也是这样，通过编程实践也不难学会。我以为，学电脑语言就像学画一样，可以先从临摹做起。别人写好的程序你认真分析、学习，上机运行，从中学习思路、算法，直到每条语句的作用。看得多了，做得多了，熟能生巧，你也就可以根据需要编写自己的程序了。这里最关键的问题就是动手实践。不动手你就会觉得很难，一动手你就会找到成功的感觉，甚至爱不释手。学用电脑贵在坚持，特别是自学，不可避免地会遇到难点，但只要你有坚定的信心和知难而上的勇气，你就会感到“世上无难事，只要肯攀登”。在你不懈地奋斗之后，电脑会俯首听命，为你所用，那时，你的心情会是多么欢畅！

世纪之交，电脑普及的浪潮一浪高过一浪，这是“科教兴国”、中华崛起的需要，我相信这套丛书一定会在普及电脑的事业中作出应有的贡献。

中国计算机学会普及委员会主任
国际信息学奥林匹克中国队总教练
清华大学计算机科学与技术系教授
吴文虎
1996.9.4　于清华园

前　言

本书是面向中学生讲PASCAL语言，目的是为使高年级中学生学习和掌握一种模块化的程序设计方法，为进一步学习计算机技术打下良好的基础。

PASCAL程序设计语言是瑞士N. Wirth教授于六十年代末七十年代初提出来的，经过二十多年的发展，PASCAL已成为目前广泛使用的程序设计语言之一。

PASCAL语言充分体现了结构化程序设计的概念，不仅具有通用语句简单灵活、模块结构清晰明了、数据类型丰富完备的特点，而且编程时的查错能力强，编写的程序结构好，运行效率高，便于移植。由于PASCAL语言具有良好的程序开发环境、高效的数值运算功能及多种系统级调用的手段，所以可以用来编写任何规模、任何类型的计算机应用程序。

面对中学生讲解PASCAL，对我们教师来说，确实动了一番脑筋。首先，根据中学生具备的数学知识，如何把PASCAL中的一些基本概念讲清楚；其次，如何把书中所讲的基本知识通过例题和习题巩固下来，让读者学会应用PASCAL解决一些简单的实际问题。为此，我们在书中安排了较多的编程实例，每章后都留有习题。

本书介绍了PASCAL语言及其程序设计，主要内容为：PASCAL程序简介、PASCAL语言的基本语句、数据类型、数组、函数与过程、集合和记录、TURBO PASCAL文件、指针变量及线性表、单元及面向对象的程序设计方法及常用算法等。

在编写过程中，我们得到了国家教委全国中小学计算机教研中心的指导和帮助，全书由中国软件行业协会余胜先生审阅，在此一并表示感谢。由于编者水平有限，书中不妥之处在所难免，请读者批评指正。

编　者

1996年5月

目 录

第一章 PASCAL程序简介

§1.1 程序设计概述

一、计算机语言、程序和软件

老师对A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A同学说的内容实际上是老师指挥A同学完成计算圆面积这件工作的步骤，通常叫做程序。A同学能否计算出要求的面积S,取决于A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。

程序是完成某项工作的步骤，它由执行程序的对象能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编译软件(如Turbo Pascal 5.5)为支撑环境。因此程序都必须以Pascal编译软件为基础来描述。

用Pascal语言(简称Pascal)编写的程序称为Pascal源程序或简称为Pascal程序。

设计程序的过程通常叫做编程。

用Pascal语言描述求圆面积的程序如下:

```
PROGRAM EX001(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.
```

让计算机执行此程序后，输出的结果为:

```
s=12.56
```

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。机器语言只使用0和1两个符号，用机器语言编程是很繁琐的，通常总是用一种学起来容易、编程十分方便的程序设计语言来编写程序，写成的程序称为该设计语言的源程序。由源程序到能在机器上直接执行的程序的转换工作则交给特别设计的、起翻译作用的软件去完成。程序设计语言的翻译软件通有如下两种工作方式:

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运行出结果来。在解释方式下，源程序的运行的都必须有解释程序的参与才能获得运算结果。随着软件技术的发展，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)一一问世，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，因功能、研制公司、版本的不同，也有许多细节上的差异。例如Pascal常见的有Dos Pascal、UCSD P Pascal、TurboPascal 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新版本的手册资料，在使用上也没有什么困难。

二、算法及语句

用具体的程序设计语言(如Pascal)编程，首先要理解问题的要求，弄清由已知数据到得出结果的过程或方法。例如由半径r求圆面积的程序中，很关键的一步是公式 $s=\pi r^2$；如果不知道公式 $s=\pi r^2$，那么由 r 求 s的程序设计就会变得十分困难。

找出求解问题的步骤、规则或公式，通常称为算法，有了算法再去设计程序，困难就会大大减少。因此编程之前先将算法清楚地表达出来，是

程序设计的基本功。

一般说来算法描述可以采用由粗到细的设计方法。

例如求 $s=a_1+a_1^2+a_1^3+a_2+a_2^2+a_2^3+\cdots\cdots+a_{10}+a_{10}^2+a_{10}^3$，可以粗分为如下步骤:

1. 输入 a_1, a_2, ……, a_{10};

2. s:=$s_1+s_2+s_3$;　　{:= 表示将右边的值赋给 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 }

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

以求和过程为例可将算法描述为:

s1.1　s1:=0; i:=1;

s1.2　i>10 时转 s1.6 ,否则转 s1.3 ;

s1.3　$s_1:=s_1+a_1$;

s1.4　i: =i+1;

s1.5　转 s1.2 ;

s1.6　结束对 s_1 的求值;

第一次经过 s1.3 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 $(a_1+a_2+\cdots\cdots+a_9)+a_{10}$，结果 s_1 中为 $a_1+a_2+\cdots\cdots+a_{10}$;第 10 次经过 s1.4 时 i:=i+1,结果 i 值已为 11;由 s1.5 转 s1.2 时 i>10 ，条件成立，结果转 s1.6 ，结束计算 S1。

为了把算法表达成 Pascal 程序，可采用 Pascal 语言中的一些语句来表达算法中的步骤。

(1) 赋值语句 x:=< 表达式 >,表示将 < 表达式 > 的值赋给变量 x 。使用符号: = 而不用 =,是避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 就成为一种数学上永远不可能成立的式子。

(2) 条件语句 IF < 条件 > THEN < 语句 1> ,表示条件成立时做

ELSE < 语句 2> 语句 1,

条件不成立时完成语句 2，例如

```
if x>0 then y:=1
```

```
        else y:=0;
```

表达了公式 $y=\begin{cases}1, & \text{当 } x>0 \text{ 时},\\ 0, & \text{当 } x\leqslant 0 \text{ 时},\end{cases}$

(3) 复合语句

在算法设计中一个很重要的思想是模块化，即将若干个操作看作一个整体，这就引出了PASCAL中的二个相应的语句：复合语句和过程。

复合语句是用 begin end 括住的若干个语句，可以将它看作一个语句。例如：

$$y=\begin{cases}1 & x>0\\ 0 & x\leqslant 0\end{cases}\qquad z=\begin{cases}10 & x<0\\ 100 & x\leqslant 0\end{cases}$$

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;
```

如果采用复合语句，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;
```

所谓过程是将一组语句看作一个独立的整体，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;        {过程的名称定为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;
```

定义成过程的好处在于不必重复书写多个语句。当程序中需要时，只需写上过程名即可调用。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) 重复语句

求和模块中第 s1.3 步重复 10 次是通过 s1.1 中 i:=1, s1.2 和 s1.5 来控制的。这一过程实际上是对 i 从 1 开始每次增加 1 直到 10 为止来控制 s1:=s1+a，重复 10 次。用 Pascal 的重复语句可写为:

```
FOR i:=1 TO 10 do s1:=s1+A[I];
```

这里 $a_1, a_2, \cdots, a_{10}$ 已定义成数组 A[1..10], 表示 A 有十个元素组成，这十个元素可通过下标 1,2,…, 10 来表示，即 A[1], A[2], …, A[10].

这样，上述求 $s=a_1+a_1^2+a_1^3+a_2+a_2^2+a_2^3+\cdots\cdots+a_{10}+a_{10}^2+a_{10}^3$ 的算法写成 Pascal 程序就方便了:

```
PROGRAM EX002(input,output);
```

(程序名定为EX002, input 表示用键盘输入， output 表示用显示器输出)

```
CONST n=10;                          (表示下标的上限取10)
VAR A:ARRAY[1 … n] of integer;(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 s1:=s1+A[i ]
  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 s2:=s2+A[ i]* A[ i]
  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]
  END;
  {主程序}
```

```
BEGIN
  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Niklans Wirth(沃思)1971年发表的，为了纪念17世纪法国著名哲学和数学家Blaisc Pascal而命名为Pascal程序设计语言。

Pascal语言充分体现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、分叉、重复三种基本结构的所谓结构化设计原则，提供了丰富的数据类型（处理对象）和清晰的模块化结构的语句，使得程序书写起来十分自由，风格优美，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进行交流的首选语言。掌握了Pascal的基本内容，将为进一步学习其他程序设计语言打下良好的基础。

二、Pascal程序的结构

从上一节的Pascal程序E001和E002可以看到，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，程序名是用户自己给程序加的名字，凡由用户命名的程序名、变量名、类型名、常量名、过程名等都称为标识符。标识符要求以字母开头，后面跟字母或数字组成（Turbo Pascal还允许使用连字符），例如exp－1，exp001，BirthDate等都是允许的标识符，而3rdRoot，TWO Word等则是不允许的标识符，因为数字开头，中间有空格都不允许。标识符的长度对具体翻译软件来说是有限制的。命名标识符时需要注意的是应有一定的表意性，不要随心所欲，以便于别人或自

已经过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，通常以标准输入输出设备(DOS 系统规定为键盘和显示器)作为输入输出时，取文件名为 Input 和 Output。对 TurboPascal 来说，这两个参数完全可以省略，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、过程和函数的有关说明，这样做的目的是能正确把握处理的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，翻译软件便指出其错误并提醒用户加以改正。

通常说明的内容有:

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 括住的语句，称为程序体，是加工数据的主体部分。对定义为过程或函数的模块，在程序体中通过过程名和函数名来调用。END 后有一圆点，不要忽略由于程序体是由一串语句构成的，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，应尽可能将它们之间的配对关系明了地表达清楚。

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，