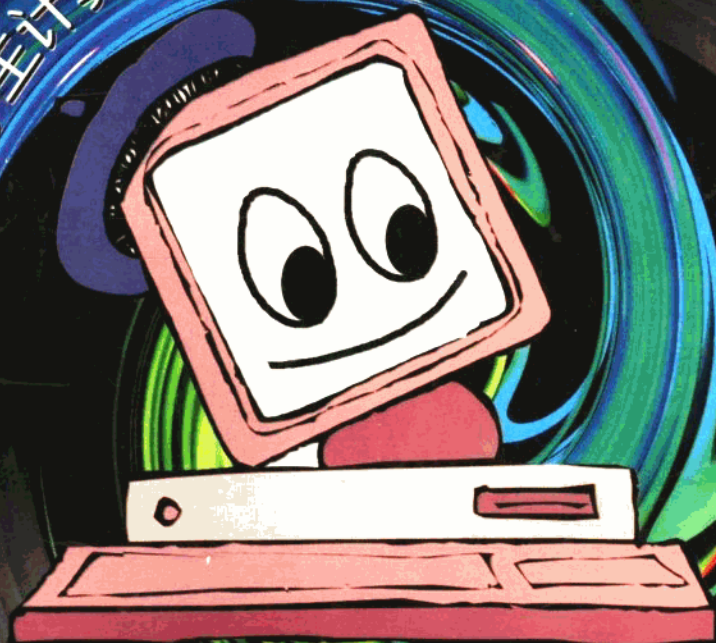


中学生计算机教程



# PASCAL 语言教程

克群 主编  
凯成 编著



中学生计算机教程

# PASCAL 语言教程

袁克群 主编  
程凯成 编著

本书较为详细地介绍了 TURBO PASCAL 6.0 语言的数据类型和基本语句,系统地讲述了程序设计的方法,采用深入浅出的写作方法,内容浅显易懂。

本书可作为中学和中等专业学校的教学用书,也可作为青少年的自学教材。

### 图书在版编目(CIP)数据

PASCAL 语言教程/程凯成编著. - 北京:机械工业出版社, 1999.1  
中学生计算机教程  
ISBN 7-111-06815-7

I . P... II . 程... III . PASCAL 语言-中学-教材  
IV . G634.67

中国版本图书馆 CIP 数据核字(98)第 35980 号

出 版 人:马九荣(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:何文军 齐福江

封面设计:姚 毅 责任校对:肖新民

北京京丰印刷厂印刷·新华书店北京发行所发行

1999 年 1 月第 1 版第 1 次印刷

787mm × 1092mm<sup>1</sup>/<sub>16</sub> · 9.625 印张 · 237 千字

0 001 - 6 000 册

定价:14.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

## 《中学生计算机教程》丛书

### 编委会名单

主 编 袁克群

副主编 程凯成

编 委 杨思禹 张凤岐  
只宏振 姜朝霞

## 序

计算机是一门新兴的学科，它千变万化的形式，无所不能的功能，无不与青少年活泼好奇的天性有着千丝万缕的联系，青少年才是计算机的主人。

计算机技术的飞速发展使目前国内中学生计算机教材远远滞后于计算机知识的高速更新，建立全新的中学生计算机课程体系势在必行，《中学生计算机教程》丛书就是我们的一个尝试。

丛书的写作原则是：以计算机的应用为主，开拓学生的眼界，培养学生的动手能力、创造性思维以及审美意识，全面提高学生的素质。

丛书突出了以下几个基本特色：

1. 着重于中学生计算机基础知识的培养，在《计算机基础教程》一书中介绍的计算机基础知识以全国计算机应用能力考核的范围为准，只要掌握了书中的有关内容就可很轻松地通过计算机应用能力的考核。

2. 为了进一步推动和促进中学生掌握和应用计算机的能力，适应全国计算机等级考试的要求，培养学生逻辑思维能力和抽象思维能力，学会一门计算机算法语言并掌握其编程技巧，是一项重要的内容。为此在《PASCAL 语言教程》一书中介绍了国际上公认为程序设计教学语言典范的 PASCAL 语言，书中的所有例题都不脱离中学生的知识范围，采用深入浅出的写作方法，内容浅显易懂。

3. 针对计算机技术日新月异的特点，本丛书力求介绍当前计算机最新的知识，如《Windows 95 教程》和《Word 97 教程》中介绍的 Windows 95、Windows 98、Word 97 等。对目前正在全球普及发展的国际互联网 Internet 也在《Internet 教程》一书中作了专门的介绍，所以本丛书不仅可以作为中学、中等专业学校的教材，也是学生活动课、选修课的首选教材。

4. 计算机辅助设计是计算机应用的一个重要方面，利用计算机绘制图形及制作动画，一直是学生最感兴趣的事，所以在《电脑动画制作——Animator Pro 教程》一书中介绍了目前国内十分普及的动画

制作软件 Animator Pro, 该软件简单易懂, 对计算机的硬件要求也较低, 是中学生学习动画制作的有力工具。

5. 针对中学生的年龄特点, 在大多数的书中采用看图速成的写作方法。同学不需很深的计算机基础, 也不需老师的指导, 只要有一台计算机, 按照书中图示进行相关的操作, 就可以十分轻松地学会有关的内容。目前计算机已普及到了家庭, 所以本丛书还可作为青少年的自学教材。

编 者

1999 年 1 月

## 前 言

目前在我国的中学,开设计算机课程的学校越来越多,搞好计算机课程教育,使学生学到必要的计算机应用知识,能熟悉和基本掌握计算机这一现代化的工具,对中学生在今后迎接二十一世纪信息化社会的挑战具有重要的战略意义。此外,计算机课程还可以全面提高广大青少年的科技素质,提高新一代劳动者的计算机意识。所以说计算机课程教学已成为中学素质教育的重要组成部分。

为了进一步推动和促进中学生掌握和应用计算机的能力,培养学生逻辑思维能力和抽象思维能力。学会一门计算机算法语言并掌握编程技巧,是中学阶段计算机课程的一项重要内容。

目前国际上公认为程序设计教学语言典范的语言是 PASCAL 语言,与 BASIC 语言相比,中学生在学习和运用该语言存在着一定的难度,所以我们采用深入浅出的写作方法,内容浅显易懂。

书中对 PASCAL 语言丰富的数据类型作了较为详细的介绍,针对中学生的特点,在内容的安排上有所侧重,重点是简单类型和构造类型中的数组类型和集合类型。在程序设计方法上作了系统地讲述,并着重培养学生养成良好的程序设计风格和习惯。为此在书中举了大量具有典型性的例题,且所有例题都不脱离中学生的知识范围,学生只要阅读和动手完成这些例题,就可以很轻松地学会程序设计的方法。

在书后的附录中我们介绍了单元的概念,供一部分学有余力的学生选学,为了减少学生上机时的困难,培养学生的自我纠错能力,我们在附录中加入了 TURBO PASCAL 的出错信息表,供学生在上机出错时查阅。

编 者

1999 年 1 月

# 目 录

## 序

## 前言

|                                |    |
|--------------------------------|----|
| 第1章 PASCAL 语言基础 .....          | 1  |
| 1.1 PASCAL 的特点 .....           | 1  |
| 1.2 PASCAL 语言的基本符号 .....       | 1  |
| 1.2.1 基本字符集 .....              | 1  |
| 1.2.2 标识符 .....                | 2  |
| 1.2.3 分隔符 .....                | 2  |
| 1.3 PASCAL 语言的程序结构 .....       | 2  |
| 1.3.1 程序首部 .....               | 2  |
| 1.3.2 程序体 .....                | 2  |
| 1.4 TURBOPASCAL 6.0 集成环境 ..... | 3  |
| 1.4.1 PASCAL 的进入 .....         | 3  |
| 1.4.2 PASCAL 菜单 .....          | 3  |
| 第2章 标准类型 .....                 | 7  |
| 2.1 整数类型 .....                 | 7  |
| 2.1.1 整型常量 .....               | 7  |
| 2.1.2 整型变量 .....               | 8  |
| 2.1.3 整数类型的运算 .....            | 8  |
| 2.1.4 标准函数 .....               | 8  |
| 2.2 实数类型 .....                 | 9  |
| 2.2.1 实型常量 .....               | 9  |
| 2.2.2 实型变量 .....               | 10 |
| 2.2.3 实数类型的运算 .....            | 10 |
| 2.2.4 实数类型标准函数 .....           | 10 |
| 2.3 字符类型 .....                 | 10 |
| 2.3.1 字符型常量 .....              | 10 |
| 2.3.2 字符型变量 .....              | 10 |
| 2.3.3 字符类型的运算 .....            | 11 |
| 2.3.4 字符类型标准函数 .....           | 11 |
| 2.4 布尔类型 .....                 | 11 |
| 2.4.1 布尔型常量 .....              | 11 |
| 2.4.2 布尔型变量 .....              | 11 |
| 2.4.3 布尔类型的运算 .....            | 11 |
| 2.4.4 布尔类型标准函数 .....           | 12 |
| 2.5 表达式 .....                  | 12 |

|                                    |           |
|------------------------------------|-----------|
| 2.5.1 定义                           | 12        |
| 2.5.2 表达式的优先级                      | 12        |
| <b>第3章 PASCAL 程序设计初步</b>           | <b>14</b> |
| 3.1 赋值语句                           | 14        |
| 3.1.1 赋值语句的一般形式                    | 14        |
| 3.1.2 说明                           | 14        |
| 3.2 输入语句——读语句                      | 15        |
| 3.2.1 输入语句的一般形式                    | 16        |
| 3.2.2 说明                           | 16        |
| 3.3 输出语句——写语句                      | 17        |
| 3.3.1 输出语句的一般形式                    | 17        |
| 3.3.2 说明                           | 18        |
| 3.3.3 输出格式                         | 18        |
| 3.4 程序设计初步                         | 20        |
| 练习题                                | 22        |
| <b>第4章 选择结构</b>                    | <b>23</b> |
| 4.1 复合语句                           | 23        |
| 4.1.1 复合语句的一般形式                    | 23        |
| 4.1.2 说明                           | 23        |
| 4.2 N-S 结构图                        | 24        |
| 4.3 IF 语句                          | 25        |
| 4.3.1 格式一: if ... then 语句          | 25        |
| 4.3.2 格式二: if ... then ... else 语句 | 25        |
| 4.4 IF 语句的嵌套                       | 28        |
| 4.4.1 格式一                          | 28        |
| 4.4.2 格式二                          | 29        |
| 4.4.3 格式三                          | 29        |
| 4.5 情况语句                           | 31        |
| 4.5.1 情况语句的一般形式                    | 31        |
| 4.5.2 说明                           | 31        |
| 练习题                                | 32        |
| <b>第5章 循环结构</b>                    | <b>34</b> |
| 5.1 REPEAT-UNTIL 循环                | 34        |
| 5.1.1 repeat-until 循环的一般形式         | 34        |
| 5.1.2 执行过程                         | 34        |
| 5.1.3 说明                           | 34        |
| 5.2 WHILE 循环                       | 36        |
| 5.2.1 while 循环的一般形式                | 36        |
| 5.2.2 执行过程                         | 36        |

|                             |    |
|-----------------------------|----|
| 5.2.3 说明                    | 36 |
| 5.3 FOR 循环语句                | 40 |
| 5.3.1 FOR 循环的一般形式           | 40 |
| 5.3.2 执行过程                  | 40 |
| 5.3.3 说明                    | 40 |
| 5.4 多重循环                    | 43 |
| 5.5 文本作图                    | 45 |
| 5.6 随机函数                    | 48 |
| 5.6.1 随机现象                  | 48 |
| 5.6.2 随机函数                  | 49 |
| 练习题                         | 51 |
| <b>第 6 章 过程和函数</b>          | 54 |
| 6.1 过程                      | 54 |
| 6.1.1 过程的定义                 | 54 |
| 6.1.2 过程的一般形式               | 54 |
| 6.1.3 说明                    | 54 |
| 6.1.4 值参数和变量参数              | 56 |
| 6.1.5 全程变量和局部变量             | 58 |
| 6.1.6 过程的嵌套                 | 62 |
| 6.2 函数                      | 63 |
| 6.2.1 函数的定义                 | 63 |
| 6.2.2 函数的一般形式               | 63 |
| 6.2.3 说明                    | 63 |
| 6.3 递归                      | 66 |
| 练习题                         | 69 |
| <b>第 7 章 枚举类型、子界类型和集合类型</b> | 70 |
| 7.1 枚举类型                    | 70 |
| 7.1.1 枚举类型的一般形式             | 70 |
| 7.1.2 说明                    | 70 |
| 7.2 子界类型                    | 72 |
| 7.2.1 子界类型的一般形式             | 72 |
| 7.2.2 说明                    | 72 |
| 7.3 集合类型                    | 75 |
| 7.3.1 集合类型的一般形式             | 75 |
| 7.3.2 说明                    | 75 |
| 7.3.3 集合的运算                 | 75 |
| 7.3.4 集合的关系运算               | 76 |
| 练习题                         | 80 |
| <b>第 8 章 数组类型 and 串</b>     | 81 |

|                              |            |
|------------------------------|------------|
| 8.1 一维数组                     | 81         |
| 8.1.1 一维数组类型的一般形式            | 81         |
| 8.1.2 说明                     | 81         |
| 8.2 数组在编程中的使用                | 85         |
| 8.2.1 排序                     | 85         |
| 8.2.2 查找                     | 88         |
| 8.3 多维数组                     | 89         |
| 8.3.1 二维数组的一般形式              | 89         |
| 8.3.2 二维数组的结构                | 89         |
| 8.4 字符数组和字符串                 | 92         |
| 8.4.1 字符数组                   | 92         |
| 8.4.2 字符串                    | 92         |
| 8.4.3 字符串的比较规则               | 93         |
| 8.4.4 字符串函数及过程               | 94         |
| 练习题                          | 99         |
| <b>第9章 记录类型、文件类型和指针类型</b>    | <b>100</b> |
| 9.1 记录类型                     | 100        |
| 9.1.1 记录类型的数据                | 100        |
| 9.1.2 开域语句                   | 103        |
| 9.2 文件类型                     | 105        |
| 9.2.1 file 类型文件              | 105        |
| 9.2.2 text 类型文件              | 109        |
| 9.3 指针类型                     | 111        |
| 9.3.1 指针类型和指针变量              | 112        |
| 9.3.2 动态存储单元的开辟和释放           | 112        |
| 9.3.3 指针变量的操作                | 113        |
| 9.3.4 链表结构                   | 115        |
| 练习题                          | 119        |
| <b>附录 A TURBO PASCAL 的单元</b> | <b>120</b> |
| A.1 概念                       | 120        |
| A.1.1 单元                     | 120        |
| A.1.2 标准单元                   | 120        |
| A.1.3 用户单元                   | 120        |
| A.1.4 单元的作用                  | 120        |
| A.2 单元定义                     | 120        |
| A.2.1 单元定义的一般形式              | 120        |
| A.2.2 说明                     | 121        |
| A.3 使用 TPUMOVER 文件           | 123        |
| A.3.1 命令格式                   | 123        |

|                               |            |
|-------------------------------|------------|
| A.3.2 复制单元到 TURBO.TPL .....   | 124        |
| A.3.3 删除 TURBO.TPL 中的文件 ..... | 124        |
| A.3.4 快速操作 .....              | 124        |
| <b>附录 B ASCII 码对照表 .....</b>  | <b>125</b> |
| B.1 控制字符 .....                | 125        |
| B.2 可打印的字符 .....              | 126        |
| <b>附录 C 常见错误信息 .....</b>      | <b>129</b> |
| C.1 编译错误信息 .....              | 129        |
| C.2 运行错误 .....                | 138        |
| C.2.1 DOS 错误 .....            | 138        |
| C.2.2 I/O 错误 .....            | 139        |
| C.2.3 严重错误 .....              | 139        |
| C.2.4 致命错误 .....              | 140        |

# 第 1 章 PASCAL 语言基础

## 1.1 PASCAL 的特点

PASCAL 是由瑞士苏黎世联邦工业大学的沃思(Niklaus Wirth)教授于 1969 年以 ALGOL-60 为基础扩充和改造而设计成功的,从 1970 年开始在计算机上使用,现已成为最广泛使用的算法语言之一。

与原有的算法语言相比, PASCAL 语言具有十分鲜明的特点。

首先 PASCAL 语言中可以使用多种数据类型,特别是允许用户自行定义最适合的数据类型。尤其是“记录”这种类型的引入使人们可以易于处理类型不同,但逻辑相关的一组信息。还有,它提供了其他语言中所不具有的指针类型和动态数据结构,具有独特的长处和强大的功能。

PASCAL 语言是结构化的语言,具有多种控制结构和定义子程序的功能,它保留了 ALGOL-60 语言中过程的概念和分层嵌套的程序结构,便于编写层次清晰结构化、模块化的程序。

PASCAL 语言除了适用于数值运算外,还适用于非数值运算领域。此外它还具有书写灵活、运行效率高、查错能力强等优点。所以 PASCAL 语言从一诞生起就受到计算机专业人员的欢迎,尤其是程序设计人员的欢迎。

作为一个初步接触程序设计的中学生来说, PASCAL 语言是一种十分理想的语言,利用它可以对学生进行严格和良好的编程训练,打好基本功。

## 1.2 PASCAL 语言的基本符号

任何程序设计语言都有一套自己的记号和规则, PASCAL 语言也是如此。

### 1.2.1 基本字符集

#### 1. 字母

A, B, C, D, ……Z, a, b, c, d, ……z。

#### 2. 数字

0, 1, 2, 3, 4, 5, 6, 7, 8, 9。

#### 3. 特殊符号

+ - \* / = < > ( ) [ ] { } . , ' ; : ^ 。

#### 4. 保留字

保留字又叫关键字,是指在 PASCAL 语言中具有固定意义的一批英文单词。每个保留字都有其特定的用途,不能作他用,下面列出本书涉及到的保留字:

AND, ARRAY, BEGIN, CASE, CONST, DIV, DO, DOWNT0, ELSE, END, FILE, FOR, FUNCTION, IF, IN, NOT, OF, OR, PACKED, PROCEDURE, PROGRAM, RECORD, REPEAT, SET, THEN, TO, TYPE, UNTIL, VAR, WHILE, WITH。

### 1.2.2 标识符

#### 1. 定义

用来表示程序、常量、变量、过程、函数和类型等名称的符号。

#### 2. 标识符名的规定

- 1) 标识符必须以字母开头,后面可以是字母和数字的任意组合,机器只识前八位。
- 2) 保留字不能作标识符,如 SUM, PRINT, X, A1, FILENAME 等都是合法的标识符。

以下为不合法的标识符:

CHER SET {空格不能作标识符}

1ABS {数字不能作开头}

MONEY% {%不能作标识符}

ARRAY {保留字不能作标识符}

### 1.2.3 分隔符

空格、行结束符、注释(用{ }括起来的一串字符)作为分隔符。

规定:任何两个相邻的保留字、标识符、数或两个相邻的符号之间或程序中第一个符号之前都必须插入一个或几个分隔符。

## 1.3 PASCAL 语言的程序结构

PASCAL 语言的程序由两部分组成:程序首部和程序体。

### 1.3.1 程序首部

以 PROGRAM 开头,后跟程序名,程序名由用户自己定义。

### 1.3.2 程序体

也可称作分程序,程序体由以下两部分组成:

#### 1. 程序说明部分

用来描述程序中用到的数据的属性,PASCAL 中规定,凡是程序中用到的变量、常量、数组、过程、函数等,都必须在说明部分进行定义。

标号说明部分: LABEL

常量定义部分: CONST

类型定义部分: TYPE

变量说明部分: VAR

过程和函数说明部分: PROCEDURE, FUNCTION

#### 2. 程序执行部分

BEGIN

<语句 1>;

<语句 2>;

:

<语句 N>;

END.

我们可以看到,程序的执行部分由保留字 BEGIN 开头,以 END 结尾,且要在后面加上

句点。其间有若干个语句，每个语句都用“;”号结尾。

下面我们来看一个完整的 PASCAL 程序：

```
例 1.3.1    program exe;                                {程序首部,定义程序名为 exe};
              var a, b, t: integer;                      {定义 a,b,t 三个变量为整型};
              begin
                a:=3;                                     {语句1};
                b:=5;                                     {语句2};
                t:=a;a:=b;b:=t;                           {语句3,4,5};
                write('a=',a,'b=',b);
              end.
```

这是一个交换两个变量值的程序，我们并不想在此就弄懂该程序的原理，我们可以在这个例子中看到 PASCAL 语言的书写风格。本程序在执行语句的书写上，采用了缩进的写法，这是为了使程序的结构更加清晰，容易阅读。在语句的执行部分，上面的程序中将语句3、语句4和语句5写在同一行上，这是完全可以的，因为它们都是用“;”号隔开的，由此我们可以充分体会到 PASCAL 语言灵活的书写风格。

#### 1.4 TURBOPASCAL 6.0集成环境

TURBOPASCAL 6.0提供了很强的功能。它把编辑、调试、运行各种功能集中于一体，通过菜单、窗口进行编程、程序存取、调试、运行等各种操作。这种编程环境称为集成环境。

##### 1.4.1 PASCAL 的进入

在 DOS 提示符下键入 TURBO，按 Enter 键，即可进入 PASCAL 集成环境。此时已处在编辑状态，可以输入和编辑程序。按(F10)键，即进入菜单状态。

##### 1.4.2 PASCAL 菜单

PASCAL 集成环境的核心是菜单操作，几乎每种操作都要通过选择菜单及相应的命令来实现。

PASCAL 的菜单是下拉菜单。进入 PASCAL 集成环境后，屏幕的顶部一行就是主菜单：

|      |      |        |     |         |       |         |         |      |
|------|------|--------|-----|---------|-------|---------|---------|------|
| File | Edit | Search | Run | Compile | Debug | Options | Windows | Help |
| ↑    | ↑    | ↑      | ↑   | ↑       | ↑     | ↑       | ↑       | ↑    |
| 文件管理 | 编辑   | 搜索     | 运行  | 编译      | 调试    | 选项      | 窗口      | 帮助   |

利用主菜单里的菜单命令，可以实现各种不同的功能。如果某菜单里还含有各种操作命令，则选中该命令时，还会出现一个子菜单，这个子菜单称为下拉菜单，又可以称为二级菜单。

##### 1. 常用 PASCAL 菜单内容简介

PASCAL 菜单命令比较多，但是不需要死记硬背。经常使用，自然就能灵活运用。我们在开始学习使用 PASCAL 时，需要掌握的最基本的功能是程序的输入、保存和运行。对初学者来说，应用比较多的菜单命令是：File, Compile, Run 和 Windows，所以这里只介绍这几个菜单。

File 菜单：

|            |
|------------|
| Open       |
| new...     |
| Save       |
| Save As... |
| :          |
| Print...   |
| :          |
| Exit       |

打开文件  
建立新程序(清除原先的程序)  
存储当前文件  
用指定名存储当前文件  
打印文件  
退出 PASCAL 环境, 返回 DOS

Run 菜单:

|     |
|-----|
| Run |
| :   |

运行当前的程序

compile 菜单:

|         |
|---------|
| Compile |
| :       |

编译当前的程序

Windows 菜单:

|             |
|-------------|
| :           |
| User Screen |
| :           |

转到用户屏幕

## 2. 菜单操作

选择菜单可以用键盘, 也可以使用鼠标。这里只介绍使用键盘选取菜单。

(1) 选取主菜单 按 F10 键(F10 键是一个“开关”, 按 F10 键后, 主菜单的首字符加亮, 再按 F10 键, 亮度消失。这一特点仅用在菜单上)。

按下高亮度字符相应的字母键或按 <→> 或 <←> 键, 从一个子菜单移到另一个子菜单。即打开相应的下拉菜单。

(2) 选取菜单命令 用 <↑> 或 <↓> 键移动光条到相应的命令处, 再按 ENTER 键。或按菜单中提示的快捷键(又称为简捷键, 取代通过屏幕菜单选取命令)。

(3) 关闭菜单 按 <ESC> 键, 即关闭当前打开的菜单, 回到先前状态。

## 3. TURBO PASCAL 的编辑器

进入 PASCAL 集成环境后, 机器就自动进入了编辑状态, 下面我们介绍一下编辑器的功能。

(1) 光标的移动 在我们编辑修改程序的时候, 需要频繁地移动光标, 下面给出光标移动按键表。

表1-1 光标移动按键表

| 光标移动       | 按 键     | 光标移动  | 按 键           |
|------------|---------|-------|---------------|
| 上, 下, 左, 右 | ↑、↓、←、→ | 移到下一页 | <PgDn>        |
| 移到行首       | <Home>  | 移到文件头 | <Ctrl>+<PgUp> |
| 移到行尾       | <End>   | 移到文件尾 | <Ctrl>+<PgDn> |
| 移到上一页      | <PgUp>  |       |               |

(2) 插入和删除 当<Insert>打开时,即进入了插入状态,在光标处可以插入任意多个字符,原光标处的字符将顺序右移。

当<Insert>关闭时,即进入了改写状态,在光标处改写原来的字符。

按一次<Backspace>键,可以删除光标前的一个字符。

按一次<Del>键,可以删除光标处的一个字符,后面的字符右移。

按<Ctrl>+<y>可以删除光标所在行。

按<Ctrl>+<n>可以在光标处插入一空行。

### (3) 块操作

表1-2 块操作功能表

| 块 功 能 | 按 键           | 块 功 能 | 按 键           |
|-------|---------------|-------|---------------|
| 定义块首  | <Ctrl>+<k>(b) | 块的复制  | <Ctrl>+<k>(c) |
| 定义块尾  | <Ctrl>+<k>(k) | 块的取消  | <Ctrl>+<k>(h) |
| 块的移动  | <Ctrl>+<k>(v) | 块的删除  | <Ctrl>+<k>(y) |

例1.4.1 在PASCAL集成环境下,输入、编辑、编译和运行例1.3.1。

步骤:

(1) 进入PASCAL语言集成环境。按<F10>及<→>或<←>键,进入File菜单,再按<↑>或<↓>键移动光条到New命令处,按回车。

(2) 此时已处在编辑状态,利用编辑器输入和编辑程序。

(3) 按<F10>及<→>或<←>键,进入Compile菜单,再按<↑>或<↓>键移动光条到compile命令处,按回车。此时对所编程序进行编译,在内存中形成机器码目标文件。

如果在程序中存在错误,可根据提示,查附录中的编译错误信息表,利用编辑器进行修改。

当编译通过时屏幕将出现提示窗口:

compiling

Main file:NONAMEOO.PAS

Compiling:NONAMEOO.PAS

Destination:memory                      Line nuber:8

Free memory:165K                        Tatalil line:8

Compie successful:press any key

上表指出该文件的文件名、编译后的文件名、编译的方式、目标文件的大小及文件所占的行数等。按任一键则编译结束。

(4) 编译通过的文件就可以运行了。按<F10>及<→>或<←>键,进入Run菜单,再按<↑>或<↓>键移动光条到Run命令处,按回车。此时运行所编译的文件,在运行中如果发现错误,根据机器提示查附录中运行错误表,利用编辑器进行修改。直至程序能正常运行。

(5) 按<F10>及<→>或<←>键,进入Windows菜单,再按<↑>或<↓>键移动光条到User Screen命令处,按回车。进入用户屏幕,这时可以看到运行的结果,再按回车即重新回到集成