

300877

TCL PASCAL 手 册

(一九八〇年六月版)

广西科学院计算中心译

一九八一年四月

内部发行

序

计算机程序设计语言PASCAL与目前国内流行的ALGOL、FORTRAN和BASIC等语言相比，有许多显著的优点。除了和上述语言一样可用于工程计算外，PASCAL简单易学，功能强大，特别适用于教学；它可用于代替机器指令或汇编来编写系统软件，受到软件工作者高度重视；由于对数据类型的进一步扩充，使PASCAL特别适合于数据处理和文件处理。目前，它在国外已相当普遍地被使用，在国内也受到许多软件工作者的注意。

这个手册是COMMODORE公司用于PET的TCL PASCAL用户手册。TCL PASCAL对标准PASCAL作了一些扩充。本手册由罗海鹏、陈寿勤、符华儿、刘连芳翻译校对。限于译者水平，错误之处一定不少，希望读者批评指正。

一九八一年四月

版权

此磁盘和手册中资料的版权属Transam 有限分公司所有。复制转卖或交换是非法和严禁的，但原买主为了使用安全，可以复制磁盘付本。

告诫

这些程序发表前，虽经Commodore公司检验，但并不许诺软件准确性的要求权。因使用这些程序而引起的损失，Commodore公司和它的经销人不承担责任，仅仅在这种谅解的基础上才出售这些程序。为了提供低价软件，程序已标准化，因此在执行各个具体应用前，应详细地检验，同时也不一定能满足某一种特殊的要求。如果需要安装、维修和培训等项服务，其费用请与软件出售人商量。不经Commodore公司建议，对软件自行进行修改，可能被取消援助性服务。

硬件支援

建议在使用这些组件前，同出售人订一个现场维修合同。

程序安全只读存储器

这个程序装在Commodore公司程序安全只读存储器里，对此用户不必耽心。因为在软件交货前，Commodore公司出售者已把程序装入了用户的机器。

供出售者参考：这个只读存储器必须放在三个标准可用区中间空着的只读存储器位置上（十六进制的A000 - AFFF），这三个标准

可用区在32K PET逻辑板上。这个只读存储器往往必须由出售者装入，而不是由用户装入。

注

此手册用COMWORDPRO III生成的程序制备、编辑和打印。

磁盘维护

小型磁盘似乎不易磨损，事实上却非常娇嫩，需要小心操作，必须经常放在磁盘驱动器或它们的保护套里。磁盘任意放置，可能失去功能。

切勿接触磁面——要注意，跟一般的认识相反，程序和数据是记录在磁盘的底下那面的。

放置磁盘的环境温度不应低于 10°C (50°F)，也不应高于 52°C (125°F)。

磁盘绝不可以弯曲。

将磁盘插入驱动器中，必须动作轻缓，小心谨慎。

磁盘应远离所有的磁场（例如电动机、大电流电缆等…）。

初始化

用户有时会对磁盘初始化感到困难——这时中间的出错灯（发光二极管L.E.D.）明亮不熄。产生这个问题的最经常的原因是磁盘的腐蚀，而磁盘的腐蚀是由于处理不细心引起的。但早期的Commodore公司磁盘驱动器也会由于驱动器内的磁盘放得不妥当而偶然发生这种故障，发生这种故障时，应把磁盘驱动器的门或盖打开，只在初始化作用已经选择了工作驱动器，并点亮了发光二极管才关上。

目 录

I. Pascal 导言	(1)
II. Pascal 初学者指南	(5)
1. 开始	(5)
WRITE (写) 语句, 字符串	(5)
整型运算: +, -, *, DIV, MOD	(8)
函数: ABS, SQR, ODD	(9)
布尔表达式 >, <, >=, <=, <>, =, AND, OR, NOT	(10)
2. Pascal 语句	(11)
变量和赋值语句	(12)
FOR (循环) 语句	(13)
IF (如果) 语句	(14)
REPEAT (直到) 和 READ (读) 语句	(15)
CASE (分情形) 语句	(17)
出错信息和错误纠正	(18)
WHILE (当) 语句	(21)
3. 进一步介绍 Pascal 的数据类型	(22)
实型数	(22)
实型运算: /, SQRT, SIN, ARCTAN, LN, EXP, ROUND, TRUNC	(22)
格式化输出和常量	(23)
字符	(25)

PET 制图	(25)
数组	(26)
枚举类型和子域 ORD, PRED, SUCC	(28)
集合	(29)
4. 过程和函数	(31)
过程	(31)
VAR (变量) 参数	(32)
函数	(33)
递归	(34)
行文文件	(35)
字符串	(36)
5. 新的特色	(39)
记录	(39)
指针和列表	(40)
GOTO (转) 语句	(42)
扩充	(43)
6. 建立在磁盘上的操作	(43)
7. 编辑命令一览表	(45)
8. 出错信息	(53)
9. 程序实例	(55)
III. TCL Pascal 参考手册	(63)
1. 一般概念	(63)
1.1 Pascal 关键字	(63)
1.2 Pascal 标识符	(63)
1.3 其他专用符号	(64)

1.4	注释	(64)
1.5	常量	(64)
	整常量	(64)
	实常量	(65)
	字符常量和字串常量	(65)
1.6	空格	(65)
2.	数据类型和运算符	(66)
2.1	整型	(66)
2.2	实型	(66)
2.3	字符型	(67)
2.4	用户定义 (枚举) 类型	(67)
2.5	子域类型	(68)
2.6	布尔型	(68)
2.7	运算符优先级	(69)
2.8	算术函数和转换函数一览表	(70)
3.	Pascal 说明和语句	(71)
3.1	Pascal 程序	(71)
3.1.1	常量说明	(71)
3.2	类型说明	(72)
3.3	变量说明	(72)
3.4	可执行语句	(73)
3.5	赋值语句	(73)
3.6	复合语句	(74)
3.7	“If” (如果) 语句	(74)
3.8	“Repeat” (直到) 语句	(75)

3.9 “While” (当) 语句	(75)
3.10 “For” (循环) 语句	(75)
3.11 “Case” (分情形) 语句	(76)
3.12 “Goto” (转) 语句	(77)
标号说明	(77)
4. 行文的输入和输出	(77)
4.1 写入行文文件	(78)
4.2 从行文文件读出	(78)
4.3 从行文文件读其它类型的数据	(79)
4.4 把其它类型的数据写入行文文件	(79)
4.5 缩写	(81)
4.6 文件处理	(81)
5. 构造数据类型	(81)
5.1 数组	(81)
5.2 集合	(83)
5.3 记录	(84)
5.4 紧缩构造	(87)
5.5 紧缩与松展	(88)
6. 函数和过程	(88)
6.1 函数和过程定义	(88)
6.2 过程和函数调用	(89)
6.3 参数表	(91)
6.4 局部量说明	(93)
6.5 递归和向前调用	(94)
7. 动态存储和指针	(95)

7.1 指针	(95)
7.2 “New” (生成变量) 和 “Dispose” (清除动态变量)	(96)
8. 磁盘文件	(97)
8.1 说明	(97)
8.2 顺序写	(97)
8.3 顺序读	(98)
8.4 外部文件	(99)
8.5 用其它的设备读或写 磁盘行文文件实例	(100) (101)
9. 对标准 Pascal 的扩充	(102)
9.1 十六进制常量	(102)
9.2 VDU存储器和通道访问	(102)
9.3 十六进制数的输入和输出	(103)
9.4 二进制运算	(104)
9.5 查找 (输入 / 输出) 错误	(105)
9.6 键盘中断	(105)
9.7 随机数发生器	(105)
9.8 下横杠	(106)
9.9 PET 机内时钟	(106)
9.10 串变量的输入	(106)
9.11 程序链接	(107)
10. TCL Pascal 接口指南	(108)
10.1 汇编语言格式	(108)
10.2 存储格式	(110)

I. Transam 分公司 Keith Frewin对TCL Pascal 的介绍

PASCAL 是一种功能强大的高级计算机语言, 它是由瑞士 苏黎士的 Niklaus Wirth 编著的*。

和在大型计算机上一样, 在小型计算机上也能有效地执行 PASCAL, 它比其它一般的微计算机语言(如BASIC)有更多的优点。

这些优点中的一些是;

象ALGOL一样的块结构**

富有意义的变量名

功能强大的数据结构技术

用户定义的数据类型和常量

极好的函数和子程序连接

递归调用

清晰、新式的控制流

运行错误检查

动态变量分配

较多的标准化

TCL Pascal是为存储器小到32K字节的小型计算机特别设计的标准 PASCAL 的实现。它把这个强功能语言的所有特性和某些有效的改进一起提供给每个计算机用户。

PET版本有两种操作方法。最简单的方法是Pascal编辑程序同用户程序一起放在随机存取存储器(RAM)中。这个方法, 对于学习本语言

* 由JENSEN 和 WIRTH 写的 PASAL 用户手册和报告...

Springer - Verlag 1975

** PASCAL形式上象ALGOL, 结构上象FORTRAN, 是块结构。

或者写不必使用磁盘的小程序是理想的。采取这种方法，除了牵涉到磁盘文件的那些命令外，多数Pascal命令是有效的，对于更复杂的程序，建立在磁盘上的编辑程序可以用来提供这种语言的全部功能。

硬件必要配置

带有32K的随机存取存储器（RAM）和二级BASIC的CBM专用计算机，并附加一个带有一级磁盘操作系统（DOS）的3040型软磁盘驱动器。

一些执行信息

最大整数值 = 32767

整型 = - 32768...32767

字符型 = ASCII集合（扩充到包含PET制图所需的字型）

集合值：必须在0...127之间（因此字符集合必须在 Chr（0）... Chr（127）之间）

实数：精确度：9位数字

范围：大约从 $1E-38$ 到 $1E38$

不写出输出格式则输出：

整型：7个字符

实型：12个字符

布尔型：1个字符

字符串型：串的长度

程序的长度和复杂性：不限制，除非超过系统的总的存储器容量（这时印出STACK OVERFLOW（栈溢出））

标识符：仅区别前面的8个字符

标号：仅区别前面的8个数字

对标准Pascal的扩充

文件名的动态说明

字符串输入

十六进制数和十六进制输入／输出

二进制运算

机器语言接口

存储器和VDU屏幕访问

运行时输入／输出错误的检测

随机数发生器

程序链接

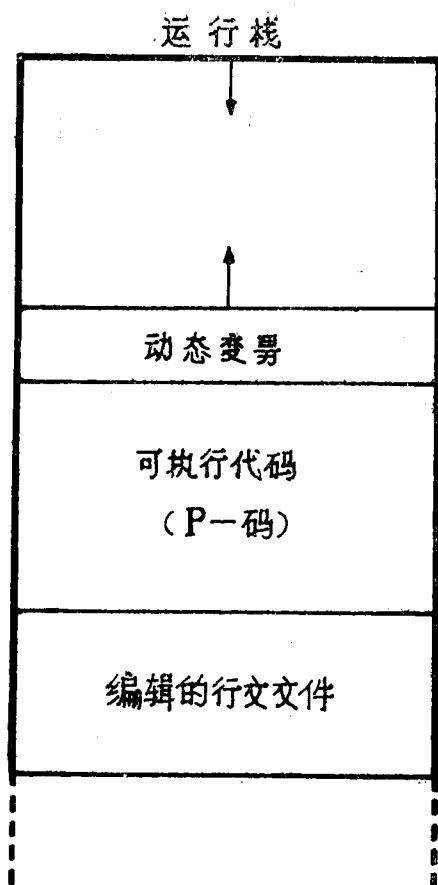
PET时钟接口

分别编译（连接）

PASCAL随机存取存储器使用情况

所有可用的存贮器被自动地优化使用

随机存取
存储器顶部



所有的存储器用完则印出 “Stack overflow” (栈溢出)。

II、Pascal初学者指南

这一章简单介绍Pascal的一些特点。

1、开始——写语句

开计算机。它应显示下列信息：

```
### COMMODORE BASIC ###
```

```
31743 BYTES FREE
```

```
REAPY.
```

要运行Pascal，就开磁盘驱动器，把Pascal磁盘插入右边的驱动器（驱动器0），然后打：

```
LOAD "*", 8
```

跟着打回车。计算机回答：

```
SEARCHING FOR *
```

```
LOADING
```

```
READY.
```

这样就把磁盘上的第一个程序装入内存中的Pascal系统区里。现在你只需打入：

```
RUN
```

接着打回车。在清晰的屏幕上，应出现TCL Pascal开始工作的信息：

```
TCL Pascal
```

```
(c)Copyright Transan 1980.
```

```
loading...
```

```
ready.
```

现代计算机功能很强，但需要由程序告诉它们做什么。计算机用

称作机器语言的语言工作，但对于人类来说，掌握机器语言通常十分困难，而且机器语言随着所用的计算机不同有很大差别。

用象COBOL或Pascal这样的高级语言和计算机通话要容易得多。这些语言有点类似英语，但有防止二义性语法的严格规则。

Pascal是瑞士苏黎士的Niklaus wirth于1968年发明的。（它以17世纪法兰西数学家Blaise Pascal的名字命名）。学习写计算机程序，Pascal是理想的语言。你的Pascal程序被计算机自动翻译成它可以解释的机器语言。

让我们马上从一个非常简单的程序例子开始吧：

〔例1〕

首先你得把这个程序放入计算机内存，这个工作是利用编辑程序而完成的。

打印如下所示的程序的第一行。后面跟着回车键（在此手册中称为〈return〉）：

```
10 begin
```

```
20 Write( 'Hi there! ' )
```

```
30 end.
```

第一行打印后，编辑程序将自动地用下一个行号提醒你；你不用重打这个行号：

```
20
```

这些行号在Pascal中没有意义——它们纯粹供编辑程序使用，并在将来所有的例子中都采用。记住，如果你打入的程序里有错误，你可以用屏幕编辑命令来修改它，这些命令是INST、DEL、指针上移、指针下移、指针左移和指针右移、这和BASIC一样。例如，按DEL键，将清除你打的最后的符号。完整的描述请看CBM用户手册。

现在打入程序余下的两行，加标点和拼字符要特别仔细，在结束时不要忘记句号“.”！

```
Write( 'Hi there! ' )  
end.
```

当你完成时，打入一个空行（仅仅打〈return〉）来结束行号。

为了运行你的程序，你必须做的是，打入r〈return〉。

如果一切都顺利，计算机将用类似下面的一些话来回答：

Compiling

Program 0 0509

0 error (s)

Compilation complete.

你不要操心细节，机内发生的事情是这样的：

计算机扫描你的程序，并把它转换成数字形式，这种数字形式计算机能够高效率地执行。（如果你没有得到信息：“0 error (s)”，那么很可能在你打入程序时有错误。你可以打“new〈return〉”重新开始！）。现在计算机自动地运行你的程序，并且打印信息：

Hi there !

一旦你的程序被编译，它被运行多少次都行，只要你打：

r〈return〉

每一次计算机都会输出

Hi there !

现在让我们更仔细地看看这程序。Pascal 程序的主体总是被括在BEGIN和END 两个字之间的，最后的 END 后面必须写一个句号。

Pascal 程序由一个“语句”的序列组成，这些“语句”按照它们写出的次序顺序地执行。例1有1个语句，即 WRITE 语句，它告诉计算机

在屏幕上写一些内容，在这时候写的是信息“Hi there！”，被括在单引号里的内容叫作字符串，它可以包含除〈return〉以外的任何字符序列。同样，如果字符串里，要包含单引号，就把单引号写两次，那么 Pascal 程序

```
begin
  write ( 'O "Brien" s string' )
end
```

就会在屏幕上输出信息

O 'Brien' s string

[例2] 加和减

除了字符串以外，其他的内容也可以输出。看下面的程序。我们将使用和例1相同的步骤，但是必须记住从计算机内存中清除例1。所以打入：

```
new 〈 return 〉
```

现在把例2打入计算机：

```
begin
  write ( 3 + 4 ) ;
  write ( 6 - 2 - 1 )
end.
```

接着打

r 〈 return 〉 (编译并运行程序) 当程序被运行时，计算机应输出

7 3

例2包含两个语句，它们必须用一个分号来隔开。这个例子是整型数运算的实例。