

泉州科技

(增刊)

应用 PAD 编程技术
中文 dBASE 编程技巧

刘甲耀 编著
严桂兰



福建省泉州市科学技术委员会
福建省泉州市科学技术情报研究所

泉州科技

一九八七年

增刊

编辑：《泉州科技》编辑部

出版：泉州市科技情报研究所

（福建省泉州市九一路）

印刷：泉州科技印刷厂

一九八七年四月

晋地署文办(81)001号

应用PAD编程技术

内 容 简 介

本书阐述应用PAD编程技术,即阐述如何应用PAD进行结构化程序设计,首先阐述PAD的基本概念与算式基本成分的描述,然后详细介绍顺序、选择、循环三种基本结构的PAD/程序写法,最后介绍过程结构的PAD/程序写法,书末附有BASIC、PASCAL、FORTRAN、PL/I、COBOL以及dBASE II所用的PAD标准图式。

本书可作为大专院校计算机专业和相关专业教材,也可作为微型计算机应用培训班教材,还可供各行各业从事计算机应用的人员自学参考。

中文dBASE编程技巧

内 容 简 介

本书在科研与教学基础上,从基本编程方法入手,根据dBASE II汉化过程及其本身的若干缺陷,以及编程过程中容易出现的一些问题,提出熔技巧性、实用性与通用性为一体的相应措施及克服办法。最后,在综合各方面内容的基础上,给出一个较完整的通用型中文dBASE II事务管理系统,它包含着全部的源程序及必需的文档,读者从中可以得到一个管理系统的整体模式。购买一本此书,犹如得到一个丰富的管理软件。

本书第一章开设dBASE II程序设计的基本手段,最后还附上dBASE II各种命令的详细含义与格式,以及与dBASE II的相互转换。因而,本书内容不仅能适用于dBASE II的编程,同时,对dBASE III也有一定的指导和实用意义。

本书不仅适用于从事dBASE II编程的技术人员,也适用于具有高中以上水平的初学者,对大专学校师生也将是良师益友,因而,它有很广泛的适用性。

应用 PAD 编程技术

目 录

前 言	(1)
第一章 PAD基本概念	(2)
§ 1—1 PAD的基本原理、图式与写法	(2)
§ 1—2 PAD描述数据结构的方法	(6)
§ 1—3 PAD标准图式及藉助它编程的方法	(7)
§ 1—4 PAD的扩充图式	(8)
第二章 算式基本成分的描述	(11)
§ 2—1 常量、变量与算术函数	(11)
§ 2—2 算术表达式	(13)
§ 2—3 字符串常量、变量与字符串函数	(14)
§ 2—4 字符串相加表达式	(15)
第三章 顺序结构的PAD/程序写法	(16)
§ 3—1 算式处理的PAD/程序写法	(16)
§ 3—2 数据输入方式的PAD/程序写法	(19)
§ 3—3 数据输出方式的PAD/程序写法	(24)
第四章 分支结构的PAD/程序写法	(28)
§ 4—1 逻辑表达式的描述	(28)
§ 4—2 单、双分支选择结构的PAD/程序写法	(30)
§ 4—3 多分支选择结构的PAD/程序写法	(33)
第五章 循环结构的PAD/程序写法	(37)
§ 5—1 下标变量与数组说明的描述	(37)
§ 5—2 循环结构的PAD/程序写法	(38)
§ 5—3 循环嵌套结构的PAD/程序写法	(54)
第六章 过程结构的PAD/程序写法	(60)
§ 6—1 子程序结构的PAD/程序写法	(60)
§ 6—2 子程序嵌套结构的PAD/程序写法	(65)
附录一 BASIC用的PAD标准图式	(67)
附录二 PASCAL用的PAD标准图式	(69)
附录三 FORTRAN、PL/I、COBOL用的PAD标准图式	(68)
附录四 dBASE II用的PAD标准图式	(69)
参考文献	(69)

中文dBASE编程技巧

目 录

第一章 基本程序设计手段	(70)
§ 1—1 命令文件的建立、修改与运行	(70)
§ 1—2 中文dBASE II程序的基本语句及其结构	(72)
一、dBASE II程序的基本结构	(72)
二、输入/输出语句	(72)
三、条件判断语句——程序的分支结构	(78)
四、循环语句——程序的重复结构	(82)
五、过程语句——子程序结构	(83)
第二章 编程技术	(85)
§ 2—1 用户数据库结构的自动建立	(85)
§ 2—2 中文dBASE II程序设计中文段名汉字化的措施	(87)
§ 2—3 通用型的自由式报表编程技术	(90)
§ 2—4 中文dBASE II中文段名的使用	(93)
§ 2—5 中文dBASE II程序设计中宏代换符的功能	(96)
§ 2—6 循环语句在中文dBASE II编程中的应用	(101)
§ 2—7 打印报表中, 缺行、空走纸等问题的解决方法	(113)
第三章 通用型的中文dBASE II事务管理系统实例	(115)
§ 3—1 系统结构	(115)
§ 3—2 系统功能	(115)
§ 3—3 系统程序解释	(117)
§ 3—4 系统使用说明	(127)
附录一 dBASE II的命令	(129)
附录二 dBASE II到dBASE III的转换	(136)

前 言

软件的发展已进入了软件工程时代,用工程方法设计、制造、检查、维护软件具有特别重要的意义。软件工程方法有多种,本书主要介绍如何应用当前最有生命力的一种软件设计表现法——问题分析图(Problem Analysis Diagram,简称PAD)来进行程序设计。PAD是一种可视的逻辑树图,与流程图等原有的程序逻辑图相比,它有三个主要特点:(1)清晰(简洁地表现程序逻辑);(2)可视(程序逻辑透明,可与源程序对应);(3)易学、易读、易记、易理解,便于编程,便于测试,便于审查,便于维护。因而,应用PAD编程能大大提高程序的设计、制造、检查、维护等生产效率。PAD可应用于诸如BASIC、FORTRAN、PASCAL、COBOL、PL/I、LISP、dBASE II(III)等所有高级语言编程。为推广应用当代这一崭新的PAD编程技术,作者根据教学和科研的经验,特编成《应用PAD编程技术》一书。本书是以编制BASIC程序为例(书末还附有PASCAL、FORTRAN、PL/I、COBOL以及dBASE II使用的PAD标准图式),详细阐述如何应用PAD进行BASIC程序设计,首先阐述PAD的基本概念与算式基本成分的描述,然后详细阐述顺序、选择、循环三种基本结构的PAD/程序写法,最后介绍过程结构的PAD/程序写法。本书所有示例的PAD/程序均在IBM—PC机(配有PAD系统)上通过。本书的特点是:(1)应用PAD编程,编程技术新颖,程序逻辑清晰,直观透明,易学、易记、适合多层次读者;(2)PAD、程序(实例)、技巧三者並述,具有简练、紧凑、明晰的编程风格。本课程讲授的内容与顺序如图0—0所示。《应用PAD编程技术》是一门实践性很强的课程,应通过PAD人工地写出源程序或者直接利用PAD系统自动编程方式上机实践,这样,才能真正学会应用PAD编程技术。由于作者水平所限,书中错误之处,在所难免,敬请读者批评指正。

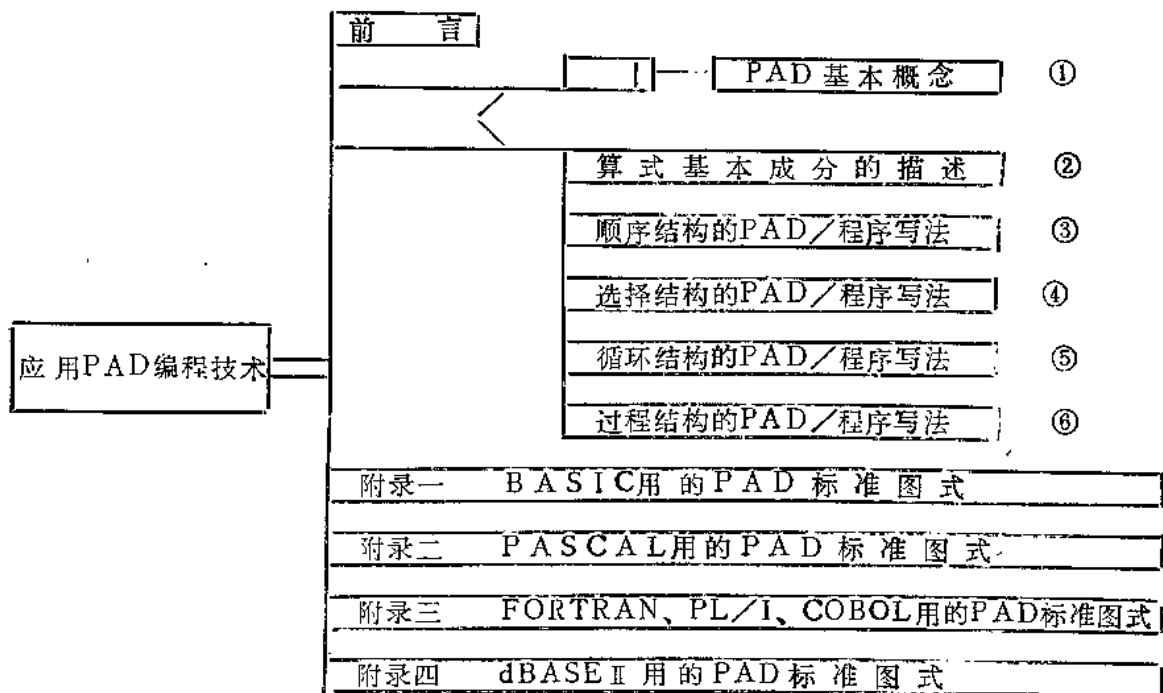


图 0—0 《应用PAD编程技术》讲授内容与顺序

应用PAD编程技术

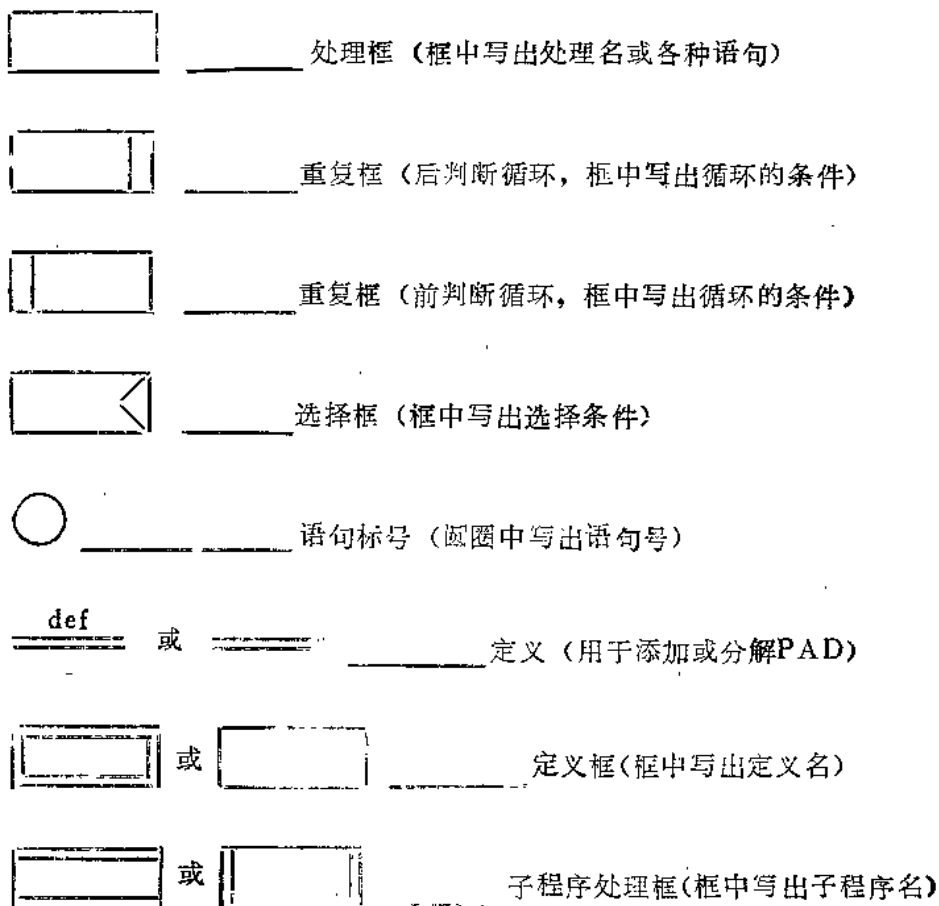
第一章 PAD的基本概念

PAD(Problem Analysis Diagram)是一种二维的、树形结构的软件设计表现法,藉助它进行结构化程序设计,能使程序易读、易记、易理解,从而大大提高程序的设计、制造、检查、维护等效率,然而,要应用它进行编程,首先就得对它有一基本概念。本章讲三个问题:一、PAD的基本原理、图式与写法;二、PAD描述数据结构的方法;三、PAD的标准图式及藉助它编程的方法。

§1-1 PAD的基本原理、图式与写法

一、PAD的基本原理

PAD的核心思想是利用自顶向下设计和反复的逐步求精的基本概念,把人们头脑中那种相当粗略、模糊的解决问题的想法转换成计算机能进行处理的明确而详尽的过程。PAD使用六种符号(如图1-1所示),它用于描述处理、重复(循环)、选择、语句标号以及定义和过程。



与其它规定的软件设计方法相类似，PAD为软件设计者提供了设计系统时所遵循的步骤：

1. 从设计程序或系统的一种模糊的概念开始，标识现有过程的各个部分的顺序，即在时间顺序上模模糊糊部分的前后关系——顺序(连接)的细化。
2. 标识将要重复的过程部分以及重复的初始和终止条件，即主要循环过程的开始和结束条件等——循环的细化。
3. 标识将要执行的各个过程部分的条件，即主要选择的条件——选择的细化。
4. 如果需要平行处理，则标识将要并行执行的那部分——平行性的细化。
5. 重复1~4步，直到过程完全确定和完全精确为止。

此过程如图1-2所示。

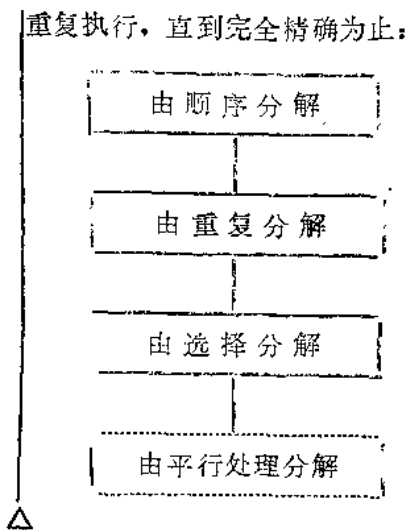


图1-2 PAD方法的处理流程

归结起来，PAD程序开发的原理可用图1-3来描述。

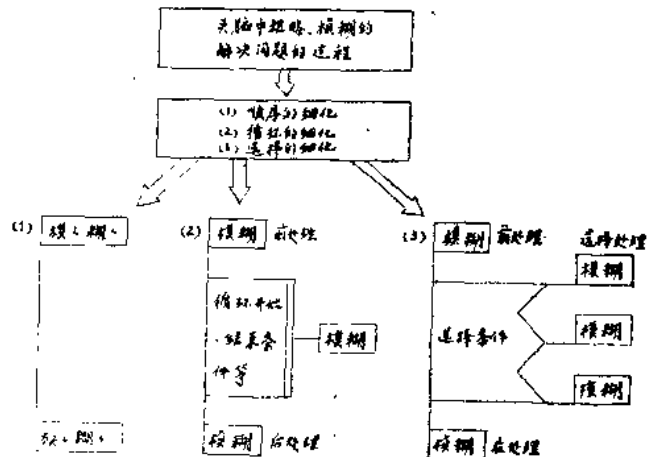


图1-3 PAD程序开发原理

二、程序结构和PAD的基本图式

任何程序都是由顺序、循环、选择三种基本形式所组成。

1. 顺序结构——处理两个事情以上的时间序列。
2. 循环结构——凡条件成立时就反复执行。
3. 选择结构——处理两个以上的事情，选择满足条件的处理。

这三种形式的程序结构及其对应的流程图(Flow Chart, 简写FC)与PAD图如表1-1所示；而在循环型中又分前判断型与后判断型以及面对问题型三种，它们的关系如表1-2所示。正如前面所指出，除顺序、循环、选择三种基本型式外，还有一种平行结构型式，用于进行平行性处理，如图1-4所示。

表 1—1 程序结构及PAD的基本形式

型式	PAD描述	FC描述
顺序型		
循环型		
选择型		

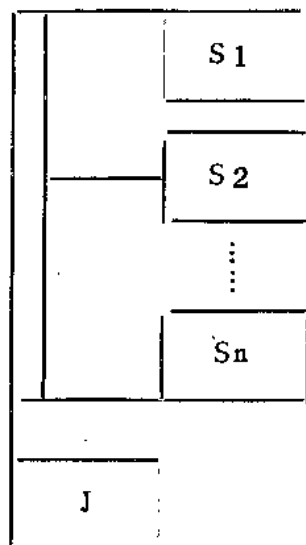
表 1—2 前判断循环与后判断循环的关系

循环图式	当条件成立时的处理	当条件不成立时的处理

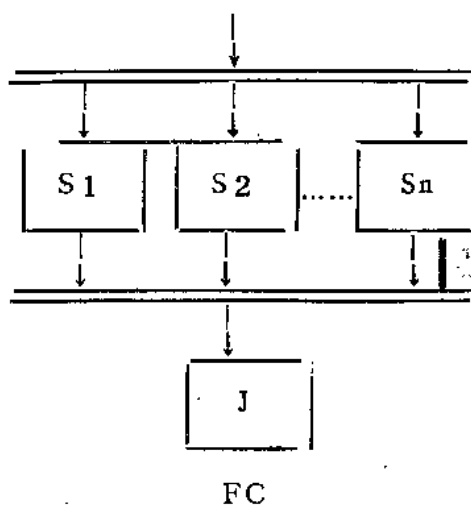
表 1—1 说明: (1) Y 表示条件成立, N 表示条件不成立;

(2) 读“Q 条件成立时”, 称“前判断循环”;

(3) 表示 Q 条件成立时执行上方的处理, 否则执行下方的处理。



PAD



FC

图 1—4 平行处理图式(含义: 平行处理 $S_1, S_2, \dots, S_n$, 然后再处理 J)

表 1—2 说明: (1)

Q

 读“直到Q条件成立为止”,称后判断循环;

(2)

I=M, N, K

 读“I从M到N,步长为K”,称前付问题循环。

注意:循环框中的M, N, K之间既可用逗号分隔,也可用分号分隔。

三、PAD的写法及注意事项

(一) PAD的写法及 def 的使用

PAD是一种二维的树形结构的程序表示法,它表示图形/概念的一种唯一的组合。纵向表示顺序信息,横向描述分支和嵌套层,两者组合即构成表示程序的PAD图。写PAD时,用户可从定义一个程序或程序组的顶层图开始,然后再用def符号添加详图。

def 或 def 符号用于定义PAD,其写法为:

S

def PAD 或

S

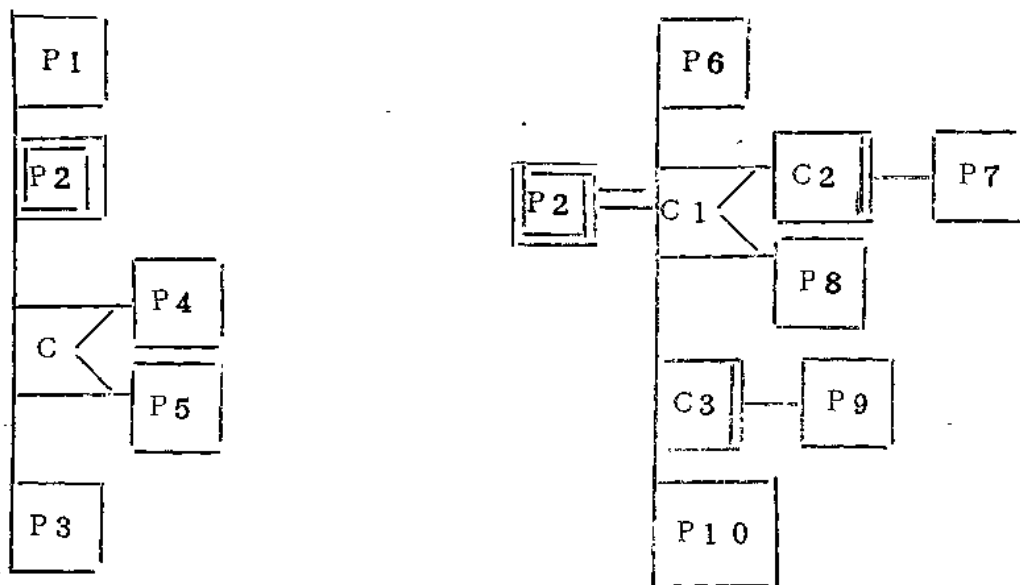
def PAD

其中 S为所定义PAD的名字。

def 用法有三:

1. 用于表示改进阶段的各个详细步骤,即逐步求精。
2. 用于把较大的PAD分解成较小的PAD。
3. 用于对某一处理添加PAD,或连接上下页,即在本页末尾和下页开始定义之。

例 1:定义PAD细部的用法(图 1—5)



(a) 初始的PAD图

(b) 使用def符号对P2求精

图 1—5 定义PAD细部的用法

例 2: 定义上下页的连接(图 1—6)

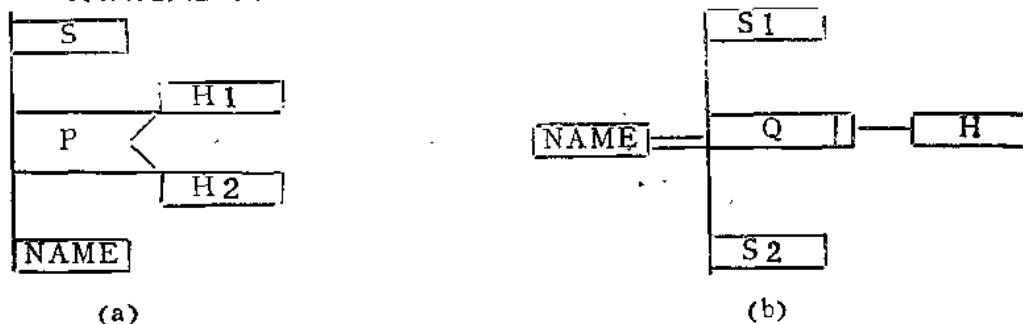
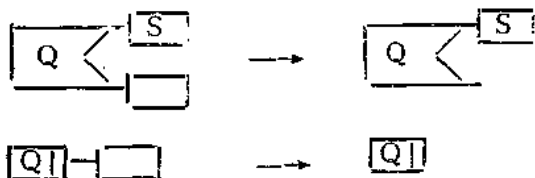


图 1—6 定义上下页连接的用法

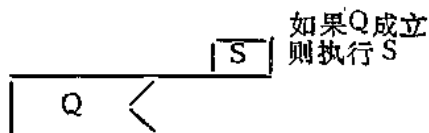
(二) 注意事项

1. 当处理框为空时, 可省略该框及其分支。

例如:



2. 在PAD外侧可写上注解。如:



3. 在横向写PAD时; (1) $\square$ 、 $\square$ 后可连接任何一种PAD框($\square$, $\square$, $\square$, $\square$, $\square$); (2) $\square$ 后不能连接任何一种框($\square$, $\square$, $\square$, $\square$, $\square$).

4. 在纵向写PAD时, 可以写一任何种PAD框($\square$, $\square$, $\square$, $\square$, $\square$).

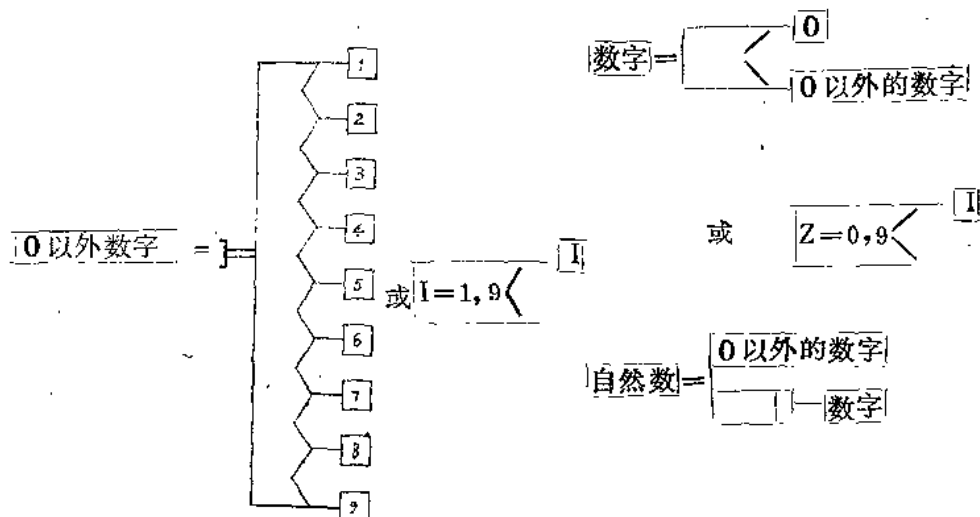
§ 1—2 PAD描述数据结构的方法

数据结构也可用象程序结构一样的那三种基本形式来描述, 如表 1—3 所示, 相当直观简单。

表 1—3 数据结构的基本形式

基本形式	PAD	含 义
顺 序		数据A之后为数据B
循 环		反复操作N次数据A, 如果N为未知数则可省略, 并且循环条件可写成 $\square$
选 择		根据条件选择数据A或B, 选择条件可写成 $\square$ 或 $\square$

例 1：自然数的描述

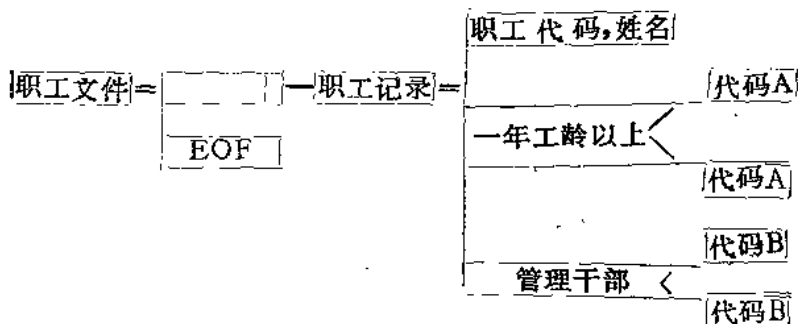


例 2：长为N的一维矩阵 $A(1), A(2), \dots, A(N)$ 。 $I=1, N \mid A(I)$

例 3： $N \times N$ 的二维矩阵 $A(1,1), A(2,1), \dots, A(N,1), A(1,2), \dots, A(N,N)$

$I=1, N \mid J=1, N \mid A(I, J)$

例 4：职工文件一例



§ 1—3 PAD的标准图式及借助它编程的方法

一、BASIC语言的PAD标准图式

PAD(写法)基本上与语言无关,除标准图式外,使用者可根据需要进行不同的扩充。BASIC的PAD标准图式如附录一所示。

二、借助PAD编程的方法

由于PAD是用于表示程序的一种二维树形结构,根据PAD能够很容易地转换为源程序。要借助PAD编程,首先要写出PAD,然后再按该图式进行编程,编程有两种方法。一种是由PAD人工地化为源程序;另一种是在有PAD系统支持的计算机上,直接输入PAD,机器即自动编码(自动写出源程序)。这里,我们介绍人工地根据PAD编程的方法,具体步骤是:

1. 写出PAD图,并标上语句标号。根据算法画出PAD图,然后在IF、GOTO、FOR、WHILE、ON语句的相应处记入○,再把PAD看作横向生长的树,沿着树迹前进,以10为单位递增地在○标记中写出上语句标号。这种沿着树形结构一步一步前进查找的

方法称为树迹法或沿树形前进法，这样画出的图称为树迹图。

2. 编写源程序

沿着树迹前进(箭头所示方向)，写出源程序，其原则是：

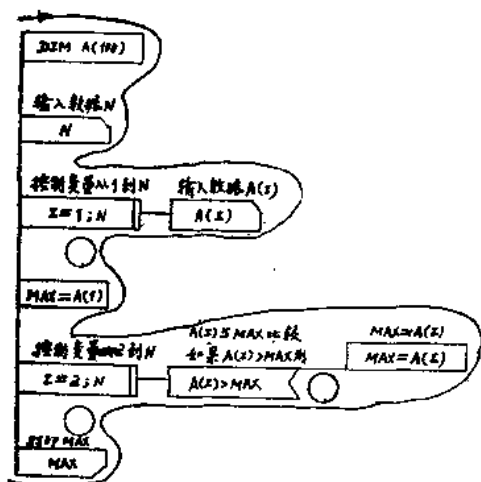
(1)遇到处理部分 ，写出其处理内容；

(2)遇到语句标号，如10，则写10 REM；

(3)遇到循环图  或选择图  则根据附录一所示的PAD标准图式写上FOR、IF、GOTO等语句。这样，沿树迹结束，程序即告完成。

示例：求输入N个数的最大值。

1. 根据题目要求写出的PAD及树迹：



2. 根据PAD写出源程序：

```

10 DIM A(100)
20 INPUT N
30 FOR I=1 TO N
40 INPUT A(I)
50 NEXT I
60 MAX=A(1)
70 FOR I=2 TO N
80 IF A(I)>MAX THEN 100
90 GOTO 110
100 MAX=A(I)
110 NEXT I
120 PRINT MAX
130 END
    
```

§ 1—4 PAD的扩充图式

任何计算程序，使用上述基本PAD图式

虽可满足要求，但要进一步提高程序的易写

性、易读性以及效率性，就得使用PAD扩充图式。为此，下面介绍四种PAD扩充图式。

一、中判断循环

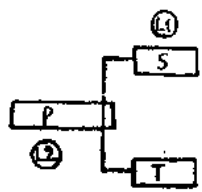
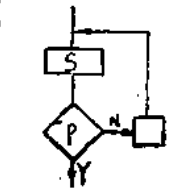
如果只用顺序、循环、连接三种基本型式描述PAD，那么，在多处出现同样的处理(这类PAD称为重复型或循环型)时，就得重复书写同样的程序，这样一来，程序就冗长。为了克服这一缺点，于是引进中判断循环或叫N+1/2循环，其描述方法如表1—3所示。

二、带限制的GOTO

为了提高程序效率和使非结构化程序的逻辑易于理解，引进了ENTER、EXIT、MERGE三种扩充图式，即在有条件的GOTO语句处加上名字ENTER、EXIT、MERGE。

在一个循环内或循环外允许三种向前转移，而在没有进入循环或离开循环的情况下，允许从一个PAD转移到另一个PAD。这三种带限制GOTO的用法如表1—4所示。注意：使用时要特别小心，否则会损坏PAD易读性。

表1—3 中判断循环

PAD	流程图	BASIC
		<pre> L1 REM S IF P THEN L2 T GOTO L1 L2 REM </pre>

1. ENTER使用示例:

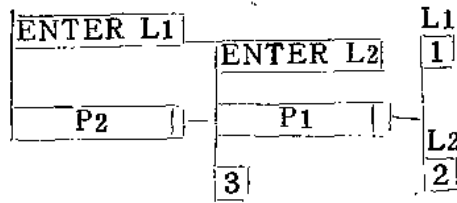
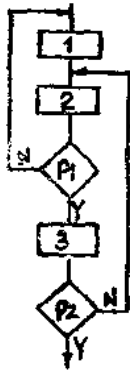
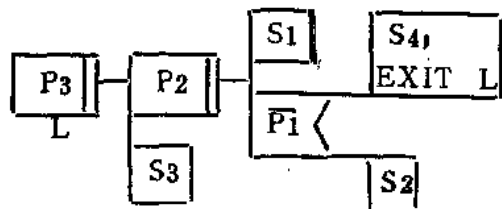
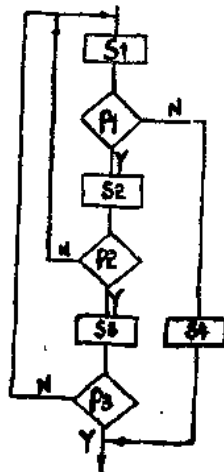


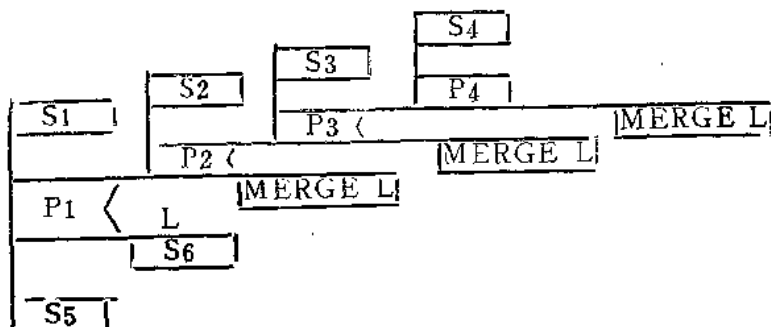
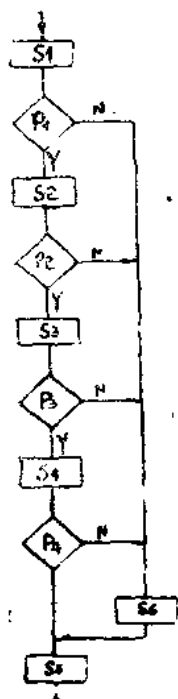
表 1—3 带限制的GOTO(向前转移)

形 式	功 能	示 例
ENTER L	从循环结构外部 转移到 内部L。 (注意:只允许进入较低层次的循环)	
EXIT L	从循环结构内部 转移到 外部L。 (注意:只允许跳 到本循环之下)	
MERGE L	从一个分支转移到另一个分支L。但不进入或退出任何循环结构。(注意:只允许在包含本框的最小循环内转移)	

2. EXIT使用示例

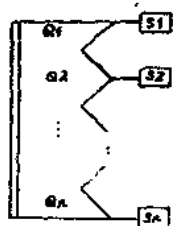


3. MERGE 使用示例

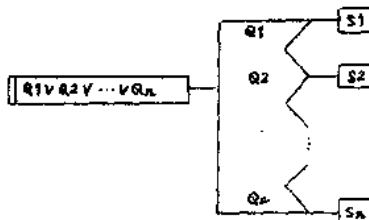


三、选择循环

选择循环的PAD为:



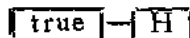
其含义为:



注:图中“V”表示逻辑或即表示BASIC中的“OR”

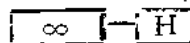
四、无穷循环

在程序设计中,有时要用到无穷循环,此时,用PAD基本结构



来描述,看起来好象没有无穷循环的感觉,

为此,引进无穷循环的PAD如下:



其相应的BASIC编程为

```

L REM
H
GOTO L
    
```

上面我们介绍了PAD的基本概念,对应用PAD编程技法有了一个初步的认识,下面我们
从算式基本成分的描述开始,对应用PAD编程技法逐一进行详细介绍。

第二章 算式基本成分的描述

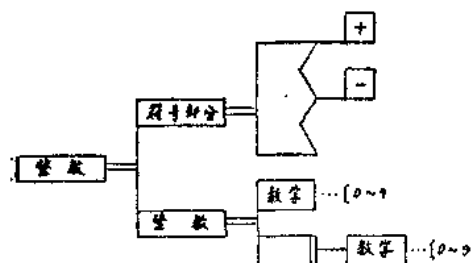
在实际计算中，经常会碰到各式各样的算式，而一般说来，算式中又含有常量、变量和标准函数这样一些基本成分，这在PAD/程序中如何描述呢？这是学习程序技法（应用PAD编程技法）首先要解决的问题。因此，本章讲四个问题：一、常量、变量和算术函数；二、算术表达式；三、字符串常量、变量与字符串函数；四、字符串相加表达式。

§ 2—1 常量、变量和算术函数

一、常量

常量（常数）是指在算式中具有固定值的数据项。在微型计算机的BASIC中，一般有三种类型的常量：整型常量、实型常量以及字符型常量。

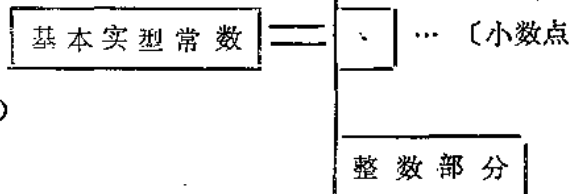
（一）整型常量（整数）



凡是不带小数点和指数的数都是整型常量，如：0，100，-8，+10。

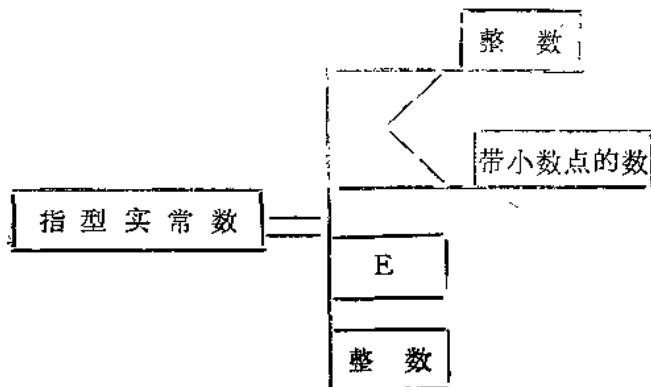
（二）实型常量（实型常数）

1. 基本实型常数（带小数点的常数）



凡是带有小数点的数都是基本实型常数（带小数点的常数）。如：3.1416，2.71828，123.45，-10.45，+7.3。

2. 指型实常数（带指数的常数）



凡是带有指数的数都是指型实常数（带指数的数）。如：1.53E3（即 1.5×10^3 ），0.6E-7（即 0.6×10^{-7} ）

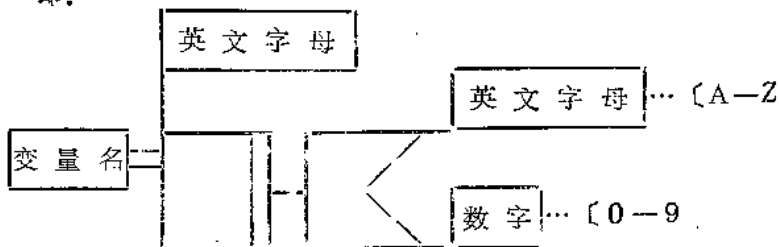
注意：E是作为数的指数符号，它只能出现在数的中间，否则无效，如E，E-5都不是实型常数。

二、变量

变量是指在算式计算中可以取不同数值的量。

(一) 变量名

为便于识别，在程序中需要给各个不同的变量取个名字(标识符)，叫做变量名。变量名可以用A—Z英文字母中任何一个字母来表示，也可用一个字母后缀一个数字或字母来表示即：

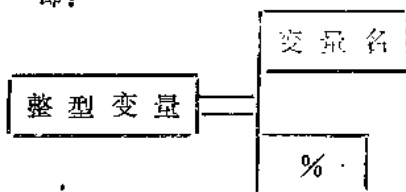


注意：在一般微型机BASIC中，变量名最多只能由两个字符（以英文字母开头的由字母数字组成的符号序列）组成，超过两个字符，计算机均不承认。

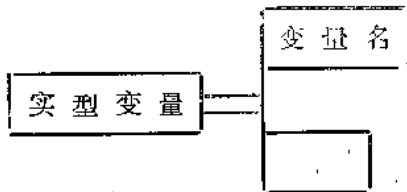
(二) 变量的类型

在BASIC中，变量的类型有整型、实型、字符串型三种，整型变量用于存贮整型常数，实型变量用于存贮实型常数，而关于字符串变量放到§2—3介绍。在程序中，对所用的各种变量，都要说明其类型。一般做法是，在变量名后面标以类型符(整型为%，实型为空)。

即：



例如：A%，B%，AB%，BA%

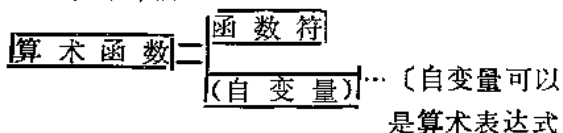


例如：A，B，AA，BA

三、算术函数

在算式中常常遇到一些算术函数，如 $\sin(x)$ ， $\cos(x)$ ， $\sqrt{x}$ 等，当用BASIC对其描述时，允许把这些函数直接写进去。

(一) 写法



即书写时，除标识符(函数符)外，必须用圆括号把初等函数的自变量括住，如右表所示：

算术函数	含 义
ABS(X)	$ x $ x 之绝对值
SQR(X)	$\sqrt{x}$ (x 为正数)
SIN(X)	$\sin(x)$
COS(X)	$\cos(x)$
TAN(X)	$\tan(x)$
ATN(X)	$\arctg(x)$
EXP(X)	e^x (指数函数)
LOG(X)	$\ln(x)$ (自然对数) $x > 0$
INT(X)	求不大于 x 的最大整数
SGN(X)	$\text{sgn}(x) = \begin{cases} -1 & \text{若 } x < 0 \\ 0 & \text{若 } x = 0 \\ 1 & \text{若 } x > 0 \end{cases}$ (符号函数)
RND(X)	产生(0, 1)的随机数

(二) 注意事项

1. 写算术函数时，一定要用圆括号把自变量括起来，如 $\text{SIN}(X)$ 不能写成 $\text{SIN}X$ 。
2. 圆括号中的自变量可以是一个算术表达式。