

PC 系列微机实用软件大全

(三)

金连甫 主编

浙江大学出版社

(浙)新登字 10 号

内 容 简 介

本书分三册,集 PC 系列微机软件之精华,详细介绍了基于 PCDOS 的“操作系统”、“计算机网络”、“字处理”、“数据库”、“工具软件”和“程序设计语言”等六大部份。内容涉及操作系统基础知识、PCDOS 3.30 的结构和命令、国内流行的汉字操作系统以及五笔字型等汉字输入方法、DOS 的中断和功能调用;网络基础知识、3+网、NOVELL 网和 NetBIOS;常用的汉字 WORDSTAR、WPS、WordPerfect、PE I、EDLIN 和 WSP 字拼写器等字处理软件;数据库基础知识、dBASE III PLUS、MFoxBASE、FoxPro 和 ORACLE 数据库;调试工具 DEBUG 和 SOURCER、系统工具 PCTOOLS 以及其它实用软件;Turbo BASIC 语言、FORTRAN 语言、Turbo PASCAL 语言、Turbo C 语言和汇编语言等。本书内容力求实用、系统、完整。

本书是 PC 微机用户和计算机室的必备参考书或工具书。

为便于读者练习,本社可提供书中有关软件,并准备制作有关软件上机录像带。详情与本社发行科联系(邮编 310027)。

编写人员:

主编: 金连甫

编委: 金连甫 袁 清 李小东 金 龙 王建国
叶耐霜 章振坚 刘 宁 张 钧 钱 涛

责任编辑:

洪保平 俞妙送 王文文 王 宇

PC 系列微机实用软件大全

(三)

金连甫 主编

* * *

浙江大学出版社出版

浙江印刷学校印刷厂印刷

浙江省新华书店发行

* * *

开本 787×1092 1/16 印张: 44.75 字数 1117 千

1994 年 2 月第 1 版 1994 年 2 月第 1 次印刷

印数: 0001—5000

ISBN 7-308-01166-6/TP·082

定价: 144.00 元

(共三册,每册 48.00 元)

前 言

电子计算机是 20 世纪科学技术的卓越成就之一,它的出现以及它在生产、科研领域中的广泛应用,正在深刻地影响着社会生产和生活的各个方面。计算机的科学水平、生产规模及其应用的广泛和程度,已成为衡量一个国家现代化水平的主要标志。

随着计算机科学技术的发展,电子计算机的应用越来越普遍,从数值计算发展到非数值计算,使计算机广泛应用于数据处理、自动控制、信息管理、办公自动化等领域,逐步渗透到企业、事业单位,乃至家庭,得到越来越广泛的应用。特别是 PC 系列微型计算机目前已在国内得到相当普及,其配备的系统软件和应用软件也极其丰富。为了使广大 PC 系列机的用户更好地学习和使用计算机,我们根据多年来从事 PC 系列微机的应用开发和教学工作的经验,特编写了《PC 系列微机实用软件大全》。

本书集 PC 系列微机软件之精华,按“操作系统”、“网络”、“字处理”、“数据库”、“软件工具”和“程序设计语言”等六大部份编写,全书分为 26 章。“操作系统”主要介绍操作系统基础知识、PCDOS 3.30 的结构和命令、国内流行的汉字操作系统以及五笔字型等汉字输入方法、DOS 的中断和功能调用。“网络”主要介绍网络基础知识、3+网、NOVELL 3.11 网和 NetBIOS。“字处理”主要介绍常用的汉字 WORDSTAR、WPS、WordPerfect、PE I、EDLIN 和 WSP 字拼写器等字处理软件。“数据库”主要介绍数据库基础知识、dBASE III PLUS、MFoxBASE、FoxPro 和 ORACLE 数据库。“工具软件”主要介绍调试工具 DEBUG、SOURCER、PCTOOLS 和其它实用软件。“程序设计语言”主要介绍 Turbo BASIC 语言、FORTRAN 语言、Turbo PASCAL 语言、Turbo C 语言和汇编语言等。

本书第一、二、三、四、五、十一、十四、二十一、二十四、二十五章由金连甫撰写;第七、八、九章由金龙撰写;第十、十二、十三章由李小东撰写;第十五、十六、十七章由袁清撰写;第二十二、二十六章由王建国撰写;第六章由刘宁撰写;第十八章由张钧撰写;第十九章由章振坚撰写;第二十章由钱涛撰写;第二十三章由叶耐霜撰写。全书由金连甫主持编写并最后修改定稿。

本书编写过程中力求实用、系统、完整,使其成为 PC 机用户的必备参考书或工具书。

在本书编写过程中,得到浙江大学计算机系软件研究所的关心和支持,得到浙江大学计算与信息中心办公自动化研究室的全力支持,在此表示衷心的感谢。感谢本书稿的全体撰写者,废寝忘食的工作和友好的配合,以使本书在较短的时间内完成。

由于编写时间仓促以及限于水平,书中错误和不妥之处在所难免,敬请读者批评指正。

金连甫

1993 年 3 月于浙大求是园

目 录

第六篇 程序设计语言(续)

第二十三章 FORTRAN

第一节 FORTRAN 语言的基本概念 1

一、字符集 1

二、FORTRAN 源程序的书写格式 1

三、常量、变量、算术表达式 2

四、数据类型及类型说明 4

第二节 基本的 FORTRAN 语句 9

一、赋值语句 9

二、PARAMETER 语句 10

三、PROGRAM 语句 10

四、简单的输出语句 11

五、简单的输入语句 11

六、FORMAT 语句 12

七、只带格式说明的 READ 和 PRINT 语句

..... 22

八、STOP 语句 23

九、PAUSE 23

十、END 语句 24

第三节 控制语句 24

一、GOTO 语句 24

二、计算 GO TO 语句 25

三、ASSIGN 语句和赋值 GO TO 语句

..... 25

四、条件语句 26

第四节 数组、循环 29

一、数组的概念 29

二、数组的说明 29

三、数组元素的引用 30

四、数组的输入输出 30

五、DO 语句 31

六、CONTINUE 语句 32

七、多重循环 33

八、隐含 DO 循环 33

第五节 文件及文件的输入输出 35

一、文件的概念 35

二、OPEN 语句 35

三、CLOSE 语句 36

四、READ 语句 37

五、WRITE 语句 38

六、REWIND 语句 40

七、BACKSPACE 语句 40

八、ENDFILE 语句 41

九、INQUIRE 语句 42

十、内部文件 43

第六节 函数和子程序 44

一、内部函数 44

二、语句函数 47

三、函数子程序 48

四、子例程序子程序 49

五、实参与虚参之间的数据传递 50

六、EXTERNAL 语句与 INTRINSIC 语句

..... 53

七、ENTRY 语句 55

八、SAVE 语句 56

第七节 数据联系语句与初值语句 56

一、EQUIVALENCE 语句 56

二、COMMON 语句 57

三、DATA 语句 59

四、数据块子程序 60

第八节 FORTRAN 程序的编译、连接与运行

..... 61

一、Microsoft FORTRAN 包含的系统文件

..... 61

二、编译元命令 65

三、编译 FORTRAN 文件 73

四、连接 FORTRAN 文件 78

五、运行 FORTRAN 文件 82

第二十四章 TURBO PASCAL

第一节 TURBO PASCAL 集成开发环境

..... 84

一、启动 84

二、文件菜单(File) 87

三、运行菜单(RUN) 88

四、编译菜单(Compile) 89

五、选项菜单(Option) 91

六、调试菜单(Debug)	95	第十一节 数值方法工具箱	284
七、Break/Watch 菜单	96	一、单变量方程的求根	284
八、集成调试器	97	二、插值	290
第二节 程序设计基本概念	104	三、数值微分	294
一、Turbo PASCAL 程序结构	104	四、数值积分	298
二、数据类型	115	五、矩阵例程	301
三、表达式	118	六、矩阵的特征值和特征向量	306
四、指针和动态内存分配	120	七、初始值和边界值法	310
第三节 程序设计控制语句	123	八、快速 Fourier 变换例程	319
一、IF 语句	123	第二十五章 TURBO C	
二、REPEAT 语句	124	第一节 Turbo C 的安装	324
三、WHILE 语句	126	一、Turbo C 2.0 的系统文件	324
四、FOR 语句	127	二、Turbo C 的安装	328
五、GOTO 语句	128	第二节 Turbo C 集成开发环境	329
六、CASE 语句	128	一、使用 TC	329
第四节 TURBO PASCAL 文件	131	二、文件菜单(File)	336
一、文件处理概念	131	三、运行菜单(Run)	338
二、正文文件	131	四、编译菜单(Compile)	340
三、有类型文件	134	五、工程菜单(Project)	342
四、无类型文件	137	六、选项菜单(Options)	343
五、文件的删除和改名	139	七、调试菜单(Debug)	349
第五节 正文与图形	140	八、Break/Watch	352
一、正文显示	140	第三节 Turbo C 程序设计	354
二、图形绘制	144	一、数	354
第六节 Turbo PASCAL 过程和函数库	155	二、表达式	364
第七节 面向对象的程序设计	188	三、语句和控制流	370
第八节 Turbo PASCAL 数据库工具箱	207	四、函数与程序结构	374
一、工具箱数据库过程	207	五、数组和指针	386
二、数据库低级命令一览	213	六、结构类型	393
三、数据库高级命令一览	217	七、其它构造类型	395
四、数据库工具箱排序例行程序	219	第四节 文件操作	399
第九节 Turbo PASCAL 图形工具箱	224	一、库函数	399
一、图形工具箱过程	224	二、标准文件的输入输出	408
二、终端过程	226	三、文件操作	413
三、图形窗口	228	第五节 图形功能	417
四、图形裁剪	231	一、窗口和视口	418
五、区域坐标系	231	二、字符屏幕管理	418
六、首部操作	232	三、图形功能	422
七、颜色	234	四、图形系统控制	423
八、绘图命令	234	五、绘图与填充	425
九、正文	239	六、屏幕及视口的管理	428
第十节 Turbo PASCAL 编辑工具箱	240	七、图形模式下的正文输出	429
		八、颜色控制	430
		九、图形模式下的错误处理	432

十、状态查询	432
第六节 Turbo C 程序的编译连接	433
一、命令行选择项	433
二、Turbo C 的存贮模式	441
三、混合语言编程	450
四、用 TCINST 工具设置 Turbo C 参数	460
第七节 TURBO C 实用工具	469
一、预处理程序 CPP	469
二、MAKE 实用程序	471
三、连接程序 TLINK	486
四、库管理程序 TLIB	489
五、Grep 实用程序	493
六、Touch 实用程序	496
七、驻留内存的 THELP 帮助工具	496
八、配置转送工具 CINSTXFR	501
九、图形驱动程序的字库转换工具 BGIOBJ	501
十、目标模块交叉引用工具 OBJXREF	504
第八节 Turbo C 库函数	504
第二十六章 汇编	

第一节 汇编语言的语法	628
一、概述	628
二、常量	629
三、变量	629
四、表达式	633
五、寻址方式	634
六、语句	637
七、标号	638
第二节 指令系统	640
第三节 伪指令	687
第四节 汇编的基本操作	702
一、程序编辑	702
二、程序编译(MASM)	702
三、程序连接(LINK)	705
四、库管理程序(LIB)	706
五、交叉参考程序(CREF)	707
第五节 汇编语言与 BIOS、DOS 的接口	707
一、汇编语言程序与 BIOS 的接口	707
二、汇编语言程序与 DOS 的接口	708

第 23 章

FORTRAN 语言

FORTRAN 是 Formula Translation 的缩写。它是适用于数值计算的一种高级语言,在社会上广泛流行。FORTRAN 语言的第一个版本是 1954 年提出的,1956 年开始使用,以后不断发展,陆续推出了许多不同版本,当时,最流行的有 FORTRAN I 和 FORTRAN N (即 FORTRAN 66)。1978 年美国标准化协会(ANSI)又公布了 FORTRAN(X3.9-1978),即 FORTRAN 77。1980 年被接受为国际标准。目前,又推出 FORTRAN 的新版本 FORTRAN 90,它是在 FORTRAN 77 基础上发展而产生的。掌握了 FORTRAN 77 以后,学习 FORTRAN 90 就比较容易了。本章介绍 IBM PC 机上的 FORTRAN 语言,IBM PC 机上的 FORTRAN 编译程序有许多版本,主要有 1.00 版本、3.2 版本、3.3 版本、4.0 和 5.0 版本等。本章介绍 Microsoft FORTRAN 4.0 的版本,它包含了 FORTRAN 77 的特性,还增加了许多功能。例如,增加了编译元命令,还支持 DOS 3.0 的网络功能,包括记录和文件加锁及文件共享等。

第一节 FORTRAN 语言的基本概念

一、字符集

FORTRAN 字符集是 FORTRAN 源程序中可以使用的字符的全体。

字符集包括下列字符:

字母:

52 个大写和小写英文字母(A~Z 和 a~z)

数字:

10 个数字(0~9)

专用字符:

␣(空格) \$ () = + - * (乘号) /(除号)
,(逗号) .(小数点) '(单引号) :(冒号)

二、FORTRAN 源程序的书写格式

FORTRAN 源程序要按规定格式书写。一行包含 80 列,可分为四个区:

1. 语句标号区——第 1~5 列。如果语句需要标号,可以写在第 1~5 列。标号是 1 到 99999 之间的无符号整数。

2. 续行区——第 6 列。如果一个语句在一行内写不下,可以在下一行继续写,在下一行的第 6 列写一个非空格和非零的任一字符,编译时就会把继续行第 6 列以后(不包括第 6 列)的字符紧接着上一行的最后一个字符。

3. 语句区——第7~72列。FORTRAN 语句从第7列~72列内任意列开始写,如语句超过第72列,则用继续行。编译程序忽略语句区内的空格(除了字符串内的空格)。

4. 说明区——第73列~80列。如果要在某一行写些注释,可以在第73列~80列中写,中文或英文都可以。编译时将忽略这部分内容。

FORTRAN 程序可以有注释行。在行的第1列写上字母C或字符*,则表示该行是注释行,一行全部是空格也是注释行,编译时对注释行忽略。

Microsoft FORTRAN 4.0版本中还可以有调试行。如果在行的第1列写上任意大写或小写字母(除了C和美元符号\$外),则该行就是调试行。如果编译元命令\$DEBUG说明了调试行中的第1列中的字符,则该行就被编译。如果\$DEBUG没有说明第1列中的字符,则该行就当作注释行处理。

三、常量、变量、算术表达式

1. 常量

在程序运行过程中其值始终不变的量称为常量。有数值型常量(整型常量、实型常量,双精度常量、复型常量),逻辑常量和字符常量。数值型常量又称常数,如5、1.2、-18.2、(65,27)等都是数值型常量,.TRUE.和.FALSE.为逻辑常量,如'ABC'是字符型常量。

2. 变量

(1) 变量的概念

在程序运行过程中其值可以变化的量称为变量。

如:A=10.2,A称为变量,它的值为10.2。

(2) 变量名

变量名由1~6个字母数字组成。第一个必须是字母,后面可以跟1~5个字母或数字。在Microsoft FORTRAN 4.0版本中,美元符号(\$)也看作字母,下划线(_)作为数字类。

如X、Y、A1、B1、AB12、A_1等都是合法的变量名。

如果变量名超过6个字符,则系统只承认前6个字符。如果两个变量名为ABCDEF和ABCDEFG,则系统认为它们是同一变量ABCDEF。但如果使用编译元命令\$NOTRUNCATE或在编译时使用/4NT选择,则名字可以有31个字符。

注意:不要用MAIN作为变量名,因为这是主程序的外部名字,也不要使用以QQ结尾的名字,因为这些名字由编辑程序保留(在一定条件下这些名字也能使用,但建议不使用为好)。

(3) 变量类型及其说明

变量分为整型变量、实型变量、双精度变量、复型变量、逻辑型变量。不同类型的变量分别用来存放不同类型的常量,如整型变量存放整型常量,实型变量存放实型常量。

变量属于哪种类型应在程序中说明,有三种方法说明变量类型。

①隐含约定、变量名由字母I、J、K、L、M、N开头的代表整型变量,由其它字母开头的代表实型变量。这个隐含的约定称为I-N规则。

如I、N、L为整型变量,而A1、ABC、X、Y₁为实型变量。

②变量类型也可以是显式说明,如:

```
INTEGER X,Y
```

变量X,Y按I-N规则应为实型变量,现在用INTEGER语句(类型说明语句)将X,Y变量规定为整型变量。

FORTRAN 中有 6 个类型说明语句:

INTEGER 语句(整型说明语句);

REAL 语句(实型说明语句);

DOUBLE PRECISION 语句(双精度说明语句);

COMPLEX 语句(复型说明语句);

LOGICAL 语句(逻辑型说明语句);

CHARACTER 语句(字符型说明语句)。

③用 IMPLICIT 语句(隐含说明语句)作隐式说明。

IMPLICIT 语句的功能:

IMPLICIT 语句用来将某一字母开头的全部变量规定为某种类型。

格式:

IMPLICIT type (a[,a]...),[,type(a[a]...)]...

说明:

类型 type 可以是 INTEGER, INTEGER * 1, INTEGER * 2, INTEGER * 4, REAL, DOUBLE PRECISION, COMPLEX, CHARACTER * 长度, LOGICAL 等之一。a 表示变量名、数组名等开头字母。Microsoft FORTRAN 4.0 版本中将美元符号(\$)看作一个字母。

举例:

IMPLICIT REAL(L,M)

将由 L 和 M 字母开头的全部变量规定为实型。

又如, IMPLICIT INTEGER(A~D), 将字母 A 到 D(A、B、C、D)开头的全部变量规定为整型。

以上三种类型说明的优先级是显式类型说明最高,其次是 IMPLICIT 语句,I-N 规则的隐含约定级别最低,如:

IMPLICIT INTEGER (A,B)

REAL ABC

如果没有这两个语句,则根据 I-N 规则,ABC 为实型,现在 IMPLICIT 语句规定以 A,B 开头的所有变量为整型,但显式类型说明语句 REAL 又规定 ABC 为实型,因为显式类型说明语句的优先级比 IMPLICIT 语句高,所以最后以 REAL 语句指定为准,即在程序中变量 ABC 为实型,其它由 A,B 开头的变量为整型。

类型说明仅仅对本程序单位有效。如在主程序中 REAL 语句说明 L 为实型,子程序中的 L 如不作说明,则按 I-N 仍为整型。

IMPLICIT 语句应写在所有类型说明语句之前,而所有类型说明语句应在所有可执行语句之前。

3. 算术表达式

(1) 算术运算符

算术运算符包括:

+	加(或正号)
-	减(或负号)
*	乘

/ 除
* * 乘方

(2) 算术表达式的含义

算术表达式包括算术常量,代表算术常量的符号名、变量名、数组元素引用或函数引用。

如 $A(I)$ 、 $X1$ 、 123 、 $SIN(B)$ 。用算术运算符和括号将算术表达式连结起来所得到的也是算术表达式。如 $A * (B + X)$ 、 $A + B$ 。

(3) 表达式的计算

表达式按照下列规则计算:

①先计算括号中的表达式(若括号嵌套,则先计算最里面的),②函数,③* (乘方),④*(乘)和/(除),⑤+(加)和-(减)。

对于相同级别的优先级按“先左后右”原则计算。如:

$8.5 - 1.2 + \text{SQRT}(8.0/2.0) * * 2$

①先计算括号中的表达式(得 4.0),②求平方根(得 2.0),③乘方(得 4.0),④由于加和减运算优先级相同,按先左后右原则,先实行减运算(得 7.3),⑤最后加运算(得 11.3)。

四、数据类型及类型说明

FORTRAN 中有 6 种数据类型:整型、实型、双精度型、复型、逻辑型和字符型。每种数据类型都有该类型所代表的值,可以对这些值进行操作。

1. 整型数据

(1) 整型常量和整型变量

整型常量也称整数,整数不应包括小数点“.”,如 10, -2, +86 都是整数。

整型变量是用来存放整型常量的。变量在内存中所占的字节数和数据存贮形式与类型说明语句有关。整型数据用 INTEGER 语句说明。

(2) INTEGER 语句

功能:

INTEGER 语句用来显式说明整型数据。

格式:

INTEGER[* bytes]v[* length][/values/][, v[* length][/value/]]...

说明:

v 是变量名、数组说明符、数组名或函数名(这里指语句函数或函数子程序,下同)。bytes 应为 1、2 或 4,也可以不写,即 INTEGER * 1, INTEGER * 2, INTEGER * 4 或 INTEGER。INTEGER * 1 用一个字节存贮一个整数,INTEGER * 2 用 2 个字节存贮一个整数,INTEGER * 4 用四个字节存贮一个整数,INTEGER 可以是 2 个字节或 4 个字节,由 STORAGE 命令确定(缺省值为 4 个字节)。1 个字节的整数取值范围: -127 ~ 127。2 个字节的整数取值范围: -32767 ~ 32767。4 个字节的整数取值范围: -2147483647 ~ 2147483647。length 说明 v 的长度,应为 1、2 或 4。如说明 length,则由 bytes 说明的长度属性就消除了。value 为常量或重复常量表,由逗号分隔,如说明 /value/,就对 v 初始化。INTEGER 语句应放在所有可执行语句之前。

举例:

INTEGER * 1 A, B * 2, C(2,3) * 4/6 * 1/, D

其中 A 为 1 个字节的整数, B 为 2 个字节的整数, 数组 C 中的 6 个数组元素是 4 个字节的整

数,每个数组元素的值都为 1,D 为 1 个字节的整数。

INTEGER X/2/

X 为 INTEGER 类型,长度由 \$STORAGE 命令确定,如果没有 \$STORAGE,则 X 为 4 个字节的整数,X 的值为 2。

2. 实型数据

(1) 实型常量和实型变量

实型常量简称实数,可以由两种形式表示,即小数形式和指数形式。小数形式的实数由整数、小数点和小数部分组成,如 1.2, -2.3, 0.1 等。小数点前也可以没数字,如 0.2 表示为 .2, 小数点后也可以没数字,如 5.0 表示为 5.。指数形式的实数由数字部分和指数部分组成。

用 E 表示以 10 为底的指数,一个实数后面加上 E,再加上一个带符号的整数(正号可省略),如:5.8E6 表示 5.8 乘以 10 的 6 次幂,即 5.8×10^6 ,用指数形式表示很大的数或很小的数很便利。

实型变量用来存放实型常量,实型变量用 REAL 语句说明。

(2) REAL 语句

功能:

REAL 语句用来说明实型数据。

格式:

REAL[* byte]v[* length][/value/][, v[* length][/value/]]...

说明:

v 是变量名、数组说明符、数组名或函数名。byte 以字节为单位说明长度,应为 4 或 8, length 也应为 4 或 8,如说明 length,则由 byte 说明的长度属性被它消除。REAL 或 REAL * 4 为单精度实数,占用内存 4 个字节,REAL * 8 为双精度实数,占用内存 8 个字节,value 为常量或重复常量表,由逗号分隔。如说明 /value/,就对 v 初始化。REAL 语句应放在所有可执行语句之前。

举例:

REAL I,L,X/2.0/

REAL * 4 A,B

3. 双精度型数据

(1) 双精度型常量和双精度型变量

双精度型数据形式、性质与实数类似,但有效位数比实数多,(实数为 6 位,双精度为 15 位),因此若要提高精度,则可以用双精度型。

在 FORTRAN 中,双精度常数用指数形式表示。表示方法与带 E 的指数型实数类似,只要把实数中表示指数的字母“E”改成字母“D”就行了。如 123.1234567D0 表示 123.1234567, 2.12345678D3 表示 2.12345678×10^3 ,在表达式中,双精度型常数一定要用指数形式表示,如:

A=8.12345678901D0

不能写成

A=8.12345678901

双精度变量用来存放双精度常量,双精度型数据可以用 DOUBLE PRECISION 语句说明。

(2) DOUBLE PRECISION 语句

功能:

DOUBLE PRECISION 语句用来说明双精度数据。

格式:

DOUBLE PRECISION v[/value/][,v[/value]]...

说明:

v 是变量名,数组说明符,数组名或函数名。value 是常量或重复常量表,用逗号分隔,DOUBLE PRECISION 语句应放在所有可执行语句之前。

举例:

DOUBLE PRECISION X₁Y₁/1.12345678D0/

规定 X₁,Y₁ 为双精度型,Y₁ 的值为 1.12345678。

4. 复型数据**(1) 复型常数和复型变量**

复型常数写成(a,b),其中 a,b 是实型常数,a 表示复数中的实部,b 表示复数中的虚部。如 (2.0,4.0) 表示复数 2.0+4.0i。复型变量用来存放复型常数,复型数据用 COMPLEX 语句说明。

(2) COMPLEX 语句**功能:**

COMPLEX 语句用来说明复型数据。

格式:

COMPLEX [* byte]v[* length][[/value/][,v[* length][[/value/]]]...

说明:

v 是变量名、数组说明符、数组名或函数名。byte 以字节为单位说明长度,应为 8 或 16, length 也应是 8 或 16,如说明了 length,则消除由 byte 说明的长度属性。value 为常量或重复常量表,用逗号分隔,如说明/value/,则对 v 初始化。COMPLEX 语句应放在所有可执行语句之前。

举例:

COMPLEX X

X=(1.2,8.2)

:

5. 四种数值型数据之间的转换和运算

(1) 如果表达式中的运算量是同类型的,则结果还是原类型。如 11/2 的值为 5(因为 11 和 2 都是整数,则结果也取整数部分)。又如 1/2 的值是 0,10*(-2)的值也为 0。

(2) 如果表达式中有不同类型的运算量,则按类型的级别进行转换,将优先级低的转变为优先级高的。类型的优先级如下:①COMPLEX * 16,②COMPLEX[* 8],③REAL * 8 或 DOUBLE PRECISION,④REAL[* 4],⑤INTEGER * 4,⑥INTEGER * 2,⑦INTEGER * 1。

6. 逻辑型**(1) 关系表达式**

关系表达式是最简单的逻辑表达式。

关系表达式的一般形式为:

<算术表达式><关系运算符><算术表达式>

FORTRAN 中的关系运算符有如下 6 个:

关系运算符	所代表的数学符号	英语原文
.GT.	>(大于)	Greater Than
.GE.	≥(大于等于)	Greater than or Equal to
.EQ.	=(等于)	Equal to
.LT.	<(小于)	Less Than
.LE.	≤(小于等于)	Less than or Equal to
.NE.	≠(不等于)	Not Equal to

如关系表达式 $I \text{ LE } 10$ 相当于数学式子 $I \leq 10$,

$x+y \text{ GE } 0.0$ 相当于数学式子 $x+y \geq 0$ 。

执行关系表达式的过程:先算出算术表达式的值,然后将两个算术表达式的值转换成同一类型(转换时将优先级低的类型向优先级高的类型转换),将两个表达式的值进行比较。复数比较,只能用 .NE. 和 .EQ.。

关系表达式的值是一个逻辑值。如果关系式满足,则值为“真”;如果关系式不满足,则值为“假”。如 $I \text{ LE } 3$,当 $I \leq 3$ 时,关系表达式的值为“真”,否则为“假”。

在关系表达式中,要注意实数在机内的近似表示。所以用 .EQ. 或 .NE. 运算符时要注意,可能理论上相等的两个量在机内不相等。所以,如果要用 .EQ. 和 .NE. 运算符对两个实型量进行比较时,可以用两者之差的绝对值小于等于某一个很小的数,则认为它们相等。如两者之差的绝对值大于某一个数时,认为它们不相等。

(2) 逻辑常量和逻辑变量

逻辑常量只有两个值:真与假。TRUE. 表示“真”,FALSE. 表示“假”。关系表达式的值是逻辑常量,如:当 $I=8$ 时,“ $I \text{ GT } 3$ ”的值为 TRUE. (真),而“ $I \text{ LT } 2$ ”的值为 FALSE. (假)。

逻辑变量是用来存放逻辑常量的。它的值也只有 TRUE. (真)或 FALSE. (假)两种。

(3) LOGICAL 语句

功能:

LOGICAL 语句用来说明逻辑型数据。

格式:

LOGICAL [** bytes*]*v* [** length*][*/value/*][*, v* [** length*][*/value/*]]...

说明:

v 是变量名,数组名,数组说明符或函数名。byte 以字节为单位说明由 LOGICAL 语句说明的项的长度,可以取 1、2 或 4。如说明 *length*,则消除由 *bytes* 说明的长度属性。*value* 是常量或重复常量表,用逗号分隔。LOGICAL 语句应放在所有可执行语句之前。

举例:LOGICAL X,Y,规定 X,Y 为逻辑变量。

(4) 逻辑表达式

(a) 逻辑表达式的基本形式为:

<逻辑量><逻辑运算符><逻辑量>

其中逻辑量有逻辑常量、逻辑变量引用、逻辑数组元素引用和逻辑函数引用;关系表达式。FORTRAN 77 的逻辑运算符有五个: .AND. , .OR. , .NOT. , .EQV. , .NEQV. 。含义如下:

.AND. 称为“逻辑与”，假定 A 和 B 为逻辑变量，则 A .AND. B 的含义是当 A 为“真”且 B 也为“真”时，A .AND. B 的值为“真”。

.OR. 称为“逻辑或”，A .OR. B 的含义是当 A 或 B 有一个为“真”时，则 A .OR. B 为“真”。

.NOT. 称为“逻辑非”，.NOT. A 的含义是如 A 为“真”，则 .NOT. A 为假，如 A 为假，则 .NOT. A 为“真”。

.EQV. 称为“逻辑等值”，A .EQV. B 的含义是 A 和 B 的值都为“真”或者都为“假”时，A .EQV. B 为“真”。

.NEQV. 称为“逻辑不等值”，A .NEQV. B 的含义是 A 和 B 的值一个为“真”，一个为“假”时，A .NEQV. B 为“真”。

(b) 逻辑表达式的运算

逻辑表达式按下列次序进行运算：

先计算括号里的，再计算算术表达式的值，然后求关系表达式的值，最后进行逻辑运算（次序是：.NOT.，.AND.，.OR.，.EQV. 或 .NEQV.）。

例如：

设 I=3, J=2, L=4, M=5

I .LE. 0 .OR. I+J .GT. M-L .AND. .NOT. TRUE 运算次序：①运算 I+J=5, M-L=1, ②求“I .LE. 0 的值为“假”，“I+J .GT. M-L”的值为“真”，③. NOT. TRUE. 的值为“假”，④进行 .AND. 运算，“TRUE .AND. FALSE.”的值为“假”，⑤进行 .OR. 运算，最后结果为“假”。

7. 字符型数据

(1) 字符型常量和字符型变量

字符型常量是单引号括起来的字符序列，也称字符串。如‘ABCDE’，‘A.’，汉字也可以出现在字符串中，如‘输入数据’。如果在字符串中出现单引号，则要用两个连续的单引号表示，如果表示字符串 AB/CD 则应写成‘AB’ ‘CD’，两个连续的单引号只表示一个字符，认为一个单引号。字符型变量用来存放字符型常量，使用之前要先说明。

(2) CHARACTER 语句

功能：

CHARACTER 语句用来说明字符型数据。

格式：

CHARACTER[* bytes]v[* length][[/value/]][, v[* length][[/value/]]]...

说明：

bytes 为字符长度，是整数表达式，其值在 1~32767 之间，如果表达式中有几项，则要用括号括起来，如果说明 length，则消除由 bytes 说明的长度。value 是常量或重复常量表，用逗号分隔，/value/对 v 初始化。如果没有说明 bytes，也没有说明 length，则长度为 1。CHARACTER 语句应放在所有可执行语句之前。

举例：CHARACTER * 8 A, B 表示字符变量 A, B 可以存放 8 个字符，如果存放汉字，则为 4 个汉字。

CHARACTER * (1+5) A 表示 A 可以存放 6 个字符。

CHARACTER A * 6, B * 8 表示 A 长度为 6, B 长度为 8。

CHARACTER A, B 隐含指定 A, B 长度为 1。

(3) 子字符串

字符串中一部分连续的字符串称为该字符串的子串,它的一般形式为:

$v([e_1,] : [e_2])$

说明:

v 是字符变量名或字符数组元素。 e_1, e_2 是算术表达式, e_1 表示子字符串在字符中的开始位置, e_2 表示子字符串在字符串中的结束位置。如果 e_1, e_2 为非整数表达式, 则在子串操作执行前, 被隐含使用的 INT 内部函数截断。 e_1 的缺省值为 1, e_2 的缺省值为串长度。如果 e_1 和 e_2 都省略, 则取字符串的全部字符, 如果有 \$ STRICT 或 /4YS 选择, 则 e_1, e_2 只能是整型表达式。

举例:

```
CHARACTER * 12 A
```

```
A = 'ABCDEFGHJKLM'
```

则子串 A(5:8) 表示 'EFGH', A(:4) 表示 'ABCD', 与 A(1:4) 相同。A(10:) 表示 'JKL', 与 A(10:12) 相同。A(:) 与 A(1:12) 相同。

(4) 字符表达式

字符表达式由字符型数据(包括字符常量、字符变量引用、字符数组元素引用、字符子串及字符函数引用)和字符运算符组成。字符运算符“//”叫做并置运算符。字符表达式的值是字符型常量。

举例:

```
CHARACTER * 6 A, A1 * 3
```

```
A1 = 'AAA'
```

```
A = A1 // 'BBB'
```

```
WRITE(*, *) A
```

```
END
```

则输出 A 的值为 AAABBB。

```
CHARACTER * 8 A, A1 * 3, A2 * 5
```

```
A1 = 'AAA'
```

```
A2 = 'ABCDE'
```

```
A = A1 // A2(2:4) // 'BB'
```

```
WRITE(*, *) A
```

```
END
```

则输出 A 的值为 'AAABCDBB'。

第二节 基本的 FORTRAN 语句

一、赋值语句

功能:

先算出表达式的值, 然后把它赋给一个变量或数组元素名。

格式:

$v = e$

说明:

v 是变量名或数组元素名,“=”是“赋值”的符号,不是数学中的等号,e 可以是算术表达式、逻辑表达式或字符表达式。算术赋值语句中的变量 v 和表达式 e 的类型可以一致,也可以不一致。如果 v 与 e 同类型,则把 e 的值直接赋给 v。如果 v 与 e 类型不一致,则先求表达式的值,再将表达式的值转换为“=”号左边变量的类型,再赋给变量。赋值语句是可执行语句。

举例:

I=3*5(两边都是整型,直接赋值)

X=2.0*Y(两边都是实型,直接赋值)

J=6.8(J 是整型,6.8 是实型,结果 J 为 6)

如果 X,Y 已定义为逻辑变量,则可以这样写:

X=.TRUE.

Y=AA.LT.2.0

如果 A 定义为字符变量,则可以这样写 A='ABCDE'。

二、PARAMETER 语句

功能:

PARAMETER 语句(参数说明语句)用来给一个常量赋予符号名,即用一个名字代表一个常量。

格式:

PARAMETER (name=e[,name=e]...)

说明:

name 是符号名,取名规则与变量名相同。符号名的类型遵循 I-N 规则,也可以用类型语句说明。符号名可以是数值型、逻辑型或字符型。e 是由常数或符号名组成的常数表达式。name 与 e 必须具有相同类型。符号名不能用于格式说明,也不能作为编辑符前的重复系数。PARAMETER 语句属于说明语句类,应写在所有可执行语句之前。

举例:

PARAMETER(U=2.0)

A=5.5*U(相当于 5.5×2.0)

B=8.0*U(相当于 8.0×2.0)

三、PROGRAM 语句

功能:

PROGRAM 语句(程序语句)用来给主程序取名。

格式:

PROGRAM pgm

说明:

pgm 是主程序名,取名规则与变量名类似。它不能与另一个过程名或公用区名相同,也不能与该主程序中的其它名字(如变量名等)相同。不要取 QQ 结尾的名字。在一个程序中,PROGRAM 语句可有可无,如果有,应是主程序的第一条语句。

举例:

PROGRAM A1

INTEGER A,B

