

微电脑学习与应用丛书·翁瑞琪 主编

FORTRAN 学习与应用

翁瑞琪 李俊旺 杨涛 编著



WEIDIANNAO XUEXI YU YINGYONG CONGSHU

国防工业出版社

TP312
W68

437374

FORTRAN 学习与应用

翁瑞琪 李俊旺 杨 涛 编著

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

FORTRAN 学习与应用/翁瑞琪等编著. —北京:国防工业出版社,1996.7

(微电脑学习与应用丛书/翁瑞琪主编)

ISBN 7-118-01636-5

I. F… I. 翁… II. FORTRAN 语言-基本知识 N. TP31
2F0

中国版本图书馆 CIP 数据核字(96)第 10606 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 14 1/4 319 千字

1996 年 7 月第 1 版 1996 年 7 月北京第 1 次印刷

印数:1—5000 册 定价:20.00 元

(本书如有印装错误,我社负责调换)

《微电脑学习与应用丛书》

编 委 会

主 编	翁瑞琪		
编 委	马丰宁	王艺海	王同胜
	阴仲朋	李永平	李俊旺
	杨 涛	杨 辉	杨丽英
	沈克成	张 炜	张洪儒
	林金栋	胡巧多	侯紫达
	曹焕林	阚瑞琪	

(以上按姓氏笔划排列)

31637
K07

31637
K07

31637
K07

31637
K07

31637
K07

31637
K07

31637
K07

丛书总序

当今人类正步入信息化时代,人们所面临的信息处理量之大、信息处理复杂程度之高,以致单靠人脑去处置已不能胜任。因此,人类迫切要求辅助脑力劳动的工具。电子计算机以其在信息处理上的独特优点而充当了人类脑力劳动的辅助工具,或者说,它起到了人类脑力活动在体外延长之作用。因此,人们美称电子计算机为电脑。

微型计算机(微电脑)的出现使计算机的应用得到极大普及。至今,微电脑已成为人类活动中不可缺少的有力助手。学习和应用微电脑是当今必然的趋势。为适应当今时代普及微电脑应用的需要,决定编写《微电脑学习与应用丛书》。

本丛书以教会如何使用微电脑为目的,帮助读者步入微电脑应用世界。

本丛书可供具有高中以上文化程度的微电脑使用者和广大爱好者自学,可用作微电脑应用短训班的培训教材,也可供大专院校广大学生学习计算机应用时参考。

本丛书由天津大学技术经济与系统工程系翁瑞琪教授主编,参加本丛书编写的是长期从事计算机教育和计算机科研工作的教学经验丰富、实践能力强的教师和专家。

本丛书的编辑出版得到国防工业出版社的大力支持,在此表示衷心的感谢。

期望本丛书的出版能为我国计算机应用范围的扩大和应用水平的提高起到促进作用。

热诚欢迎有关专家和广大读者对本丛书的编辑出版提出建设性的建议和改进意见。

翁瑞琪

前 言

本书是翁瑞琪教授主编的《微电脑学习与应用丛书》之一。

FORTRAN 语言是世界上最早推广应用的高级语言之一,至今仍然是最流行的高级语言。FORTRAN 语言不仅用于编制科学计算中的程序,还越来越多地用来编制商业和企业管理方面的程序。

《FORTRAN 学习与应用》可用作高级语言程序设计课程的学习教材,也可供计算机用户自学或用作相应的计算机培训班的培训教材。

本书分十二章,分别介绍 FORTRAN 基础、常用 FORTRAN 语句、数据的输入与输出、控制转移、判定与选择、循环、数组与下标变量、函数、子程序、文件等内容。附录中给出 FORTRAN 上机指导。

参加本书编写的有翁瑞琪、李俊旺、杨涛。第一、二、五、六、七、九、十一章和附录 A 由翁瑞琪执笔,第三、四章和附录 B 到附录 D 由李俊旺执笔,第八、十、十二章由杨涛执笔。全书由翁瑞琪统一定稿。

由于编者水平所限,错误在所难免,欢迎批评指正。

目 录

第一章 FORTRAN 基础	(1)	2.2.2 符号常数的命名	(14)
1.1 FORTRAN 程序	(1)	2.2.3 符号常数的类型	(14)
1.1.1 FORTRAN 程序示例	(1)	2.2.4 参数说明语句使用中需注意的 几个问题	(16)
1.1.2 FORTRAN 程序的结构与 构成	(1)	2.3 赋初值语句	(17)
1.1.3 源程序书写格式	(2)	2.4 标号赋值语句	(17)
1.2 FORTRAN 字符集	(3)	习题二	(18)
1.3 FORTRAN 常数	(3)	第三章 数据的输入	(20)
1.3.1 整型常数	(3)	3.1 表控格式输入	(20)
1.3.2 实型常数	(4)	3.2 格式输入	(23)
1.3.3 双精度型常数	(4)	3.2.1 直接指定输入格式的输入	(23)
1.3.4 复型常数	(4)	3.2.2 带格式语句的输入	(26)
1.3.5 逻辑型常数	(5)	习题三	(26)
1.3.6 字符型常数	(5)	第四章 数据的输出	(29)
1.4 FORTRAN 变量	(5)	4.1 WRITE 语句	(29)
1.4.1 变量名	(5)	4.1.1 表控格式输出	(30)
1.4.2 变量类型	(5)	4.1.2 带格式语句的输出	(31)
1.4.3 子字符串	(7)	4.1.3 直接指定输出格式的输入	(41)
1.5 FORTRAN 表达式	(7)	4.2 PRINT 语句	(42)
1.5.1 算术表达式	(7)	4.2.1 自由格式 PRINT 语句	(42)
1.5.2 关系表达式	(8)	4.2.2 带 FORMAT 语句的 PRINT 语句	(42)
1.5.3 逻辑表达式	(9)	4.2.3 直接指定输出格式的 PRINT 语句	(42)
1.5.4 字符表达式	(10)	习题四	(42)
习题一	(10)	第五章 结束、停、暂停与控制转移 语句	(47)
第二章 计算赋值语句、参数说明语句、 赋初值语句与标号赋值语句	(12)	5.1 结束、停与暂停语句	(47)
2.1 计算赋值语句	(12)	5.1.1 结束语句	(47)
2.1.1 数值型计算赋值语句	(12)	5.1.2 停语句	(47)
2.1.2 逻辑型计算赋值语句	(13)	5.1.3 暂停语句	(47)
2.1.3 字符型计算赋值语句	(13)		
2.2 参数说明语句	(14)		
2.2.1 一般格式	(14)		

5.2 控制转移语句	(48)	9.1 内部函数	(111)
5.2.1 无条件转移语句	(48)	9.2 语句函数	(121)
5.2.2 计算转移语句	(48)	习题九	(122)
5.2.3 赋标号转移语句	(49)		
5.2.4 算术 IF 语句	(50)	第十章 函数子程序与子例程序	
5.2.5 条件转移语句	(50)		(124)
习题五	(51)	10.1 引言	(124)
第六章 判定与选择	(52)	10.2 函数子程序	(125)
6.1 逻辑 IF 语句	(52)	10.2.1 函数子程序的定义	(125)
6.2 块 IF 结构	(53)	10.2.2 函数子程序的调用	(127)
习题六	(58)	10.3 子例程序	(132)
第七章 循环	(59)	10.3.1 子例程序的定义	(133)
7.1 当型循环	(59)	10.3.2 子例程序的调用	(134)
7.2 直到型循环	(62)	10.3.3 子程序中的数组	(139)
7.3 DO 循环	(64)	10.4 外部语句和内部语句	(142)
7.4 多重循环	(67)	10.4.1 FORTRAN 77 的 4 种 过程	(142)
7.5 隐 DO 表	(69)	10.4.2 EXTERNAL 语句和 INTRINSIC 语句	(143)
习题七	(70)	10.5 子程序中的 SAVE 语句 ...	(146)
第八章 数组与下标变量	(71)	10.6 子程序的多重入口和可变 返回点	(149)
8.1 引言	(71)	10.6.1 子程序的多重入口和 ENTRY 语句	(149)
8.2 数组	(73)	10.6.2 可变返回点	(150)
8.2.1 数组说明	(73)	10.7 程序举例	(151)
8.2.2 数组的大小	(76)	习题十	(160)
8.2.3 下标变量	(76)		
8.3 数组的逻辑结构和存储 结构	(79)	第十一章 等价语句、公用语句与 数据块子程序	(164)
8.3.1 数组的逻辑结构	(79)	11.1 等价语句	(164)
8.3.2 数组的存储结构	(80)	11.2 公用语句	(165)
8.3.3 下标值	(81)	11.3 等价语句与公用语句的 联用	(167)
8.4 数组的输入和输出	(82)	11.4 数据块子程序	(168)
8.4.1 下标变量值的输入和输出	(82)	习题十一	(169)
8.4.2 数组的输入和输出	(83)		
8.4.3 DATA 语句实现变量、数组 的初始化	(89)	第十二章 文件	(170)
8.5 数组与下标变量的应用	(91)	12.1 引言	(170)
习题八	(106)	12.2 文件的打开与关闭	(171)
第九章 内部函数与语句函数	(111)		

12.3 文件的输入与输出操作	(176)	习题十二	(189)
12.3.1 输入操作	(176)	附录 A FORTRAN 77 语句在程序	
12.3.2 输出操作	(177)	单位中的顺序	(193)
12.4 顺序文件和直接文件	(178)	附录 B FORTRAN 77 程序如何在	
12.4.1 有格式顺序文件的存取		IBM PC 及其兼容机上运行	
方法	(178)		(195)
12.4.2 有格式直接存取文件	(181)	附录 C FORTRAN 77 程序如何在	
12.4.3 无格式顺序文件和直接		VAX 计算机上运行	(203)
文件	(183)	附录 D Microsoft FORTRAN 5.0	
12.5 内部文件	(184)	的使用	(212)
12.6 程序举例	(186)		

第一章 FORTRAN 基础

FORTRAN 是英文 FORMula TRANslation(公式翻译)的缩略,它是世界上最早推广应用的高级语言,也是当前最流行的高级语言之一。

FORTRAN 有多种版本,流行的为 ANSI X3.9-1966 FORTRAN(被称为 FORTRAN 66)和 ANSI X3.9-1978 FORTRAN(被称为 FORTRAN 77)两种版本。

FORTRAN 66 在数值计算方面得到十分广泛的应用。

FORTRAN 77 是在 FORTRAN 66 基础上发展起来的,它与 FORTRAN 66 兼容。它包括了 FORTRAN 66 的几乎全部功能并进行了较大的扩充,因此,它不仅可用于数值计算,还适用于非数值计算。更为重要的是,它扩充了结构化语句,便于实现结构化程序设计。

本书主要介绍 FORTRAN 77(为叙述方便起见,本书中在不另加说明的情况下,FORTRAN 指 FORTRAN 77)。

1.1 FORTRAN 程序

1.1.1 FORTRAN 程序示例

下例给出的是计算边长为 A、B、C 的三角形面积 AREA 的 FORTRAN 程序。

例 1

C	CALCULATE THE AREA OF TRIANGLE	(注释行)
C	WITH SIDES A,B,C	(注释行)
	READ (*,10)A,B,C	(读语句)
10	FORMAT(3F10.5)	(格式语句)
	CALL AOT(A,B,C,AREA)	(调用语句)
	WRITE(*,20)AREA	(写语句)
20	FORMAT(1X,F10.5)	(格式语句)
	END	(结束行)
	SUBROUTINE AOT(X,Y,Z,S)	(子例程语句)
	P=(X+Y+Z)/2.0	(赋值语句)
	S=SQRT(P*(P-A)*(P-B)*(P-C))	(赋值语句)
	RETURN	(返回语句)
	END	(结束行)

1.1.2 FORTRAN 程序的结构与构成

1. 主程序与子程序

一个 FORTRAN 程序由一个主程序和若干个(可以是 0 个)子程序组成。FORTRAN

有三种子程序:子例程子程序、函数子程序和数据块子程序。例 1 程序中含一个主程序和一个子例程子程序。

在 FORTRAN 中,不论是主程序还是子程序,都是独立的程序单位(各程序单位可独立地进行编译)。在 FORTRAN 中,每个程序单位都要以一个 END 结束。

要运行一个 FORTRAN 程序,首先要分别对其各程序单位进行编译,然后把它们连接成一个统一的程序,于是才能运行。未编译前的程序称为源程序。

2. 语句行与非语句行

从例 1 程序可以看到,每一程序单位由若干行组成。有的行是语句行,有的行是非语句行。例 1 程序中以字母 C 开头的行为注释行,它们是非语句行,READ(读)、WRITE(写)等语句所在行为语句行。FORTRAN 规定,在一行内至多写一个语句,而一个长的语句可占多行。

语句行又有初始行与续行之分。在一个语句较长以致在一行内写不下而需占多行的情况下,它所占各行中的第一行称为初始行,其余行称为续行。短语句只占一行,只占一行的短语句所在行也称为初始行,它是无续行的初始行。注意,含 END(结束)语句的行不是初始行,称为结束行。

3. 语句号(标号)

由例 1 程序还可看到,有的语句前未标有号码,有的语句前标有号码。例如,例 1 程序的两个 FORMAT(格式)语句分别标有号码 10 与 20,这样的号码称为语句号(标号)。FORTRAN 规定,语句前可标语句号,也可不标语句号。标有语句号的语句,可通过其语句号而被其他语句所引用。因此,在设计程序时,对于以后要引用的语句要标以语句号。语句号选用 1~99999 之间的整数。注意,语句号的大小并不代表语句执行次序的先后。

4. 执行语句与非执行语句

从程序运行期间是否执行的角度来看,FORTRAN 语句可分为执行语句与非执行语句。例 1 程序中的赋值语句、READ(读)语句、WRITE(写)语句等是执行语句,而 FORMAT(格式)语句、SUBROUTINE(子例程)语句等为非执行语句。

5. END

在 FORTRAN 66 中,END 不是语句,是结束行,它是一个程序单位结束的标志,每一程序单位必须以一个 END 结束。由于结束行不是语句,该行不允许标有语句号。

在 FORTRAN 77 中,END 是语句,其执行导致程序执行的结束。另外,它兼有标志一程序单位结束的作用。也就是说,在 FORTRAN 77 中,每一程序单位仍以一个 END 结束。因此,FORTRAN 77 把含 END 语句的行仍称为结束行,该行不允许标有语句号。

1.1.3 源程序书写格式

为了便于编译程序对源程序进行编译,也为了便于阅读,FORTRAN 源程序必须按规定格式书写。

1. 标号区、续行标志区、语句区与注解区

标准的 FORTRAN 程序纸每行 80 列,分标号区、续行标志区、语句区和注解区 4 个区。

①标号区 每行第 1 列到第 5 列为标号区。凡需标出语句号的语句,可在该语句初始

行的标号区位置书写语句号,最多为 5 位数字。

②续行标志区 每行第 6 列为续行标志区。在一个语句过长而在其初始行写不下的情况下,可转写到下一行。在转行时,要在其各续行的续行标志区写一个续行标志,表示此行为续行。续行标志可用非 0 或非空格的任何字符表示。

③语句区 每行第 7 列到第 72 列为语句区。语句要写在语句区内,它可从第 7 列或第 7 列到第 72 列的任何列开始写。在一行写不下而需转行时,可在写满第 72 列后转行,也可在未写满第 72 列的某一合适位置转行。

④注解区 第 73 列到第 80 列为注解区。它是供程序设计者加注解用的。编译程序对源程序进行编译时忽略注解区的内容。

2. 初始行、续行、结束行与注释行

一个源程序是由若干行组成的,这些行有初始行、续行、结束行之分,还有注释行。在书写 FORTRAN 源程序时,它们是通过以下标写法来区分开的。

①初始行 语句初始行的续行标志区应留为空白或标以 0。在需标出一语句的语句号时把语句号写在该语句的初始行的标号区。

②续行 语句续行的续行标志区应标以非 0 或非空格的其他字符。续行的标志区应留为空白。

③结束行 一个程序单位的最后一行为结束行,用 END 表示,END 写在该行的语句区内。结束行的标号区、续行标志区均应留为空白。

④注释行 注释行的第 1 列标以 C 或 *。注释本身可从第 2 列或其后的位置开始写,最多可写到第 72 列。一条注释在一行内写不完时,将出现多行注释行,但每一注释行的第 1 列都要标以 C 或 *。注释行不是语句行,无续行的概念。

1.2 FORTRAN 字符集

FORTRAN 所能接受的全部字符称为 FORTRAN 字符集。FORTRAN 字符集包括:

英文字母 A~Z

阿拉伯数字 0~9

专用字符 空格 ' \$ () + - * / = , . : 等

1.3 FORTRAN 常数

在程序运行期间其值固定不变的量称为常数。FORTRAN 常数有整型、实型、双精度型、复型(以上 4 种统称为数值型)、逻辑型和字符型等类型。

1.3.1 整型常数

FORTRAN 整型常数(整数)是由一位或多位十进制数前面加上一个任意的正号“+”或负号“-”构成的。例如,0、-12、+34.567 等都是合法的整数。“+”号可以省略,例如,123 与 +123 是等效的。整数中不允许出现小数点,例如,在 FORTRAN 中,2 是整数,

而 2. 或 2.0 不是整数而是实常数。

1.3.2 实型常数

FORTRAN 实型常数(实常数)有基本实常数与指数形式实常数 2 种表示形式。

1. 基本实常数

FORTRAN 基本实常数是用定点形式表示的实常数,它包括一个任选的±号跟一个整数部分、一个小数点和一个小数部分。整数部分和小数部分由一位或多位十进制数构成。例如,0.12、-3.4、+45.67、89.0 等都是合法的实常数。基本实常数可以没有整数部分,例如,0.12 可省略记作“.12”;也可以没有小数部分,例如,89.0 可省略记作“89.”。但是,不能把整数部分与小数部分两者均省略。±号可以省略,例如,34.567 与 +34.567 是等效的。基本实常数中的小数点不可省略,例如,在 FORTRAN 中,3. 或 3.0 是基本实常数,而 3 不是基本实常数而是整数。

2. 指数形式实常数

FORTRAN 指数形式实常数是采用浮点形式表示的实常数。这种形式表示的实常数由有效数字部分和指数部分两部分组成。有效数字部分可以是基本实常数或整数,指数部分由字母 E 和后跟一个任选地冠以±号的整数构成,±号可以省略。指数表示其前面的值乘以 10 的整数次幂。例如, $1.2E-8 (=1.2 \times 10^{-8})$ 、 $3E10 (=3 \times 10^{10})$ 等都是合法的指数形式实常数。无论有效数字部分还是指数部分中的±号均可省略,例如, $3E10$ 、 $+3E10$ 、 $3E+10$ 与 $+3E+10$ 都是等效的。有效数字部分本身不可省略,例如,在数学中, 1×10^8 可记作 10^8 ,但在 FORTRAN 中, $1E8$ 是指数形式实常数,而 $E8$ 则是变量。有效数字部分可用整数表示,也可用基本实常数表示,例如, $3E10$ 与 $3.E10$ 、 $3.0E10$ 都是等效的。指数部分的指数只可以是整数,不允许是实常数,例如, $1.2E3.4$ 是非法的。

1.3.3 双精度型常数

为了提高精度,FORTRAN 提供双精度型常数(双精度常数),双精度常数由有效数字部分和指数部分两部分组成,有效数字部分可以是基本实常数或整数,指数部分由字母 D 和后跟一个任选的冠以±号的整数构成。例如, $1.234D-8$ 、 $5D12$ 等都是合法的双精度常数。

与指数形式实常数相比,双精度常数在机器中存储字节数增加一倍,其有效位数从而其精度也提高一倍左右。因此,它被称为双精度常数(也可称实常数为单精度常数)。单精度常数与双精度常数在书写上的区别是:单精度常数指数部分用字母 E 开头,双精度常数指数部分则用字母 D 开头。

1.3.4 复型常数

在科学与工程问题中,经常出现具有实部与虚部的复数,例如, $2.3+3.4i$ 。FORTRAN 提供复型常数(复数)来满足这一需要。FORTRAN 用括号内的一对实数来表示复数,例如, $2.3+3.4i$ 在 FORTRAN 中写作 $(2.3,3.4)$ 。有的机器要求表示实部、虚部的两个数必须是实常数,在这种机器中, $(2,3)$ 的写法是非法的,其正确写法是 $(2.,3.)$ 。有的机器则允许用整数来表示实部和虚部,在这种机器中, $(2,3)$ 的写法也是允许的。但是,所有机器都不允许用双精度常数来表示实部和虚部,例如, $(1.2E-2,1.3E-2)$ 是合法的 FORTRAN 复数,而 $(1.2D-2,1.3D-2)$ 是非法的。

1.3.5 逻辑型常数

FORTRAN 提供的两个逻辑型常数(逻辑常数)是“.TRUE.”与“.FALSE.”。“TRUE.”表示真,“.FALSE.”表示假。

1.3.6 字符型常数

FORTRAN 字符型常数(字符常数)是用一对单引号括起来的一个或多个字符,其值就是这对单引号所定界的字符串。例如,'RICHARD'的值是 RICHARD,'MALE'的值是 MALE,'SHANGHAI'的值是 SHANGHAI,'05/30/1932'的值是 05/30/1932,它们都是合法的字符常数。但需注意,若所定界的字符串内含单引号,一个单引号要用两个连续的单引号来表示。例如,其值为 IT'S 四个字符的字符常数的正确表示是'IT'S'而不是'IT'S'。

1.4 FORTRAN 变量

在程序运行期间其值可以变化的量称为变量。提供变量这一手段的实际意义在于用程序员定义的名字代表存储单元的绝对地址,一旦命名一个变量,就可方便地引用该名字来存取它所代表的存储单元。

1.4.1 变量名

FORTRAN 变量名由 1~6 个字符(字母或数字)组成,其首字符为字母。例如,A、B、C、S、AREA、A1、A2、A3 等都是合法的变量名,而 1X、L-10、 α 等是非法的。

1.4.2 变量类型

FORTRAN 变量有整型、实型、双精度型、复型(以上 4 种类型统称为数值型)、逻辑型和字符型等类型。在 FORTRAN 中,指定变量类型的方法有隐含说明、显式说明以及用隐含说明语句说明等多种方法。

1. 隐含说明

在未作其他说明(显式说明或隐含说明语句说明)的情况下,凡首字符为 I、J、K、L、M、N 之一的任何变量的类型被隐含说明为整型,凡首字符为任一其他字母的任何变量的类型被隐含说明为实型。这种隐含说明的规则被称为 I-N 规则。例如,AREA、A、B、C 等为实型变量,M1、NUM 等为整型变量。

2. 显式说明

FORTRAN 提供类型说明语句来显式说明变量的类型。经显式说明的变量的类型不再受 I-N 规则的限定,也不受隐含说明语句的限定。

FORTRAN 提供如下 6 种类型说明语句。

(1) INTEGER 语句

INTEGER(整型)语句用来说明指定变量的类型为整型。其一般格式如下:

INTEGER(变量表)

例如,INTEGER A,B,C 语句说明 A、B、C 均为整型变量。

(2) REAL 语句

REAL(实型)语句用来说明指定变量的类型为实型。其一般格式如下:

REAL(变量表)

例如, REAL I, J, K 语句说明 I, J, K 均为实型变量。

(3) DOUBLE PRECISION 语句

DOUBLE PRECISION(双精度型)语句用来说明指定变量的类型为双精度型。其一般格式如下:

DOUBLE PRECISION(变量表)

例如, DOUBLE PRECISION AREA, VOLUME 语句说明 AREA, VOLUME 均为双精度型变量。

(4) COMPLEX 语句

COMPLEX(复型)语句用来说明指定变量的类型的复型。其一般格式如下:

COMPLEX(变量表)

例如, COMPLEX Y, Z, RATIO 语句说明 Y, Z, RATIO 均为复型变量。

(5) LOGICAL 语句

LOGICAL(逻辑型)语句用来说明指定变量的类型为逻辑型。其一般格式如下:

LOGICAL(变量表)

例如: LOGICAL ONE, TWO 语句说明 ONE, TWO 均为逻辑型变量。

(6) CHARACTER 语句

CHARACTER(字符型)语句用来说明指定变量的类型为字符型,并说明其长度。它有如下几种常用格式:

格式 1 CHARACTER(变量表)

它说明指定变量为字符型变量,其长度为 1。例如, CHARACTER ALPH1, ALPH2 语句说明 ALPH1, ALPH2 均为长度为 1 的字符型变量。

格式 2 CHARACTER * (长度), (变量表)

它说明指定变量为字符型变量,其长度由(长度)给定。例如, CHARACTER * 10, CITY, NAME 语句说明 CITY, NAME 均为长度为 10 的字符型变量。

格式 3 CHARACTER(变量 1) * (长度 1), (变量 2) * (长度 2), ...

它说明各指定变量为字符型变量,其长度分别由(长度 1)、(长度 2)、...指定。例如, CHARACTER NAME * 10, CLASS * 5 语句说明 NAME, CLASS 均为字符型变量,它们的长度分别为 10 和 5。

格式 2、格式 3 中指定的(长度)应是整型的量,可以用整数表示,也可用整型变量或整型表达式表示。

3. 用 IMPLICIT 语句说明

IMPLICIT 语句为隐含说明语句,它用来说明以指定字母为首字符的所有变量的类型。从而排除 I-N 规则对以这些字母为首字符的变量类型的限定。IMPLICIT 语句的一般格式如下:

IMPLICIT(类型 1)(字母范围 1), (类型 2)(字母范围 2), ...

其中, (类型)可取 INTEGER、REAL、DOUBLE PRECISION、COMPLEX、LOGICAL 或 CHARACTER * (长度)(CHARACTER * 1 可省略为 CHARACTER), 分别表示整型、实型、双精度型、复型、逻辑型或字符型。(字母范围)有两种表示法: 是逐个列出各字母,其间用逗号隔开;一是在字母范围为连续的情况下,列出两端的字母,中间加“—”

号。

例如,IMPLICIT INTEGER(A,C),REAL(I-K)语句说明以 A、C 之一为首字符的所有变量的类型为整型,以 I~K 之一为首字符的所有变量的类型为实型。

又如,IMPLICIT CHARACTER * 5(A),CHARACTER * 8(B-D)语句说明凡是由字母 A 开头的变量均是长度为 5 的字符型变量,凡是由 B~D 中任一字母开头的变量均是长度为 8 的字符型变量。

1.4.3 子字符串

一个字符变量的值为一串字符,有时只需引用其中一部分连续的字符,这在 FORTRAN 中可用子字符串(子串)来表示。子串实际上是字符型变量的一个片断。子串的一般格式如下:

⟨字符型变量⟩(⟨位置 1⟩:⟨位置 2⟩)

其中⟨位置 1⟩、⟨位置 2⟩分别给出子串的下界和上界,⟨位置 1⟩、⟨位置 2⟩取整型量,可以用整数表示,也可用整型变量或整型表达式表示。⟨位置 1⟩的缺省值为 1,⟨位置 2⟩的缺省值为该字符型变量的长度。

例如,CHARACTER * 10,STRING 语句定义 STRING 是长度为 10 的字符型变量,于是 STRING(2:5)表示字符型变量 STRING 从第 2 个字符到第 5 个字符的子串;STRING(:5)表示 STRING 从第 1 个字符到第 5 个字符的子串;STRING(4:)表示 STRING 从第 4 个字符到第 10 个(最末)字符的子串。

1.5 FORTRAN 表达式

FORTRAN 通过表达式来实现计算,因运算对象不同,FORTRAN 的表达式可分为算术表达式、关系表达式、逻辑表达式和字符表达式。

1.5.1 算术表达式

算术表达式是根据计算的需要,用算术运算符把算术运算对象(数值常数、数值型变量等)组合起来的计算式,其计算结果是一个数值,所以,也可把算术表达式看作是一种数值型数据。

1. 算术运算符与其优先级

FORTRAN 的算术运算符有+(加)、-(减)、*(乘)、/(除)、** (乘幂)。它们具有不同的优先级,优先级最高的为 **,其次为 *、/(* 和 / 优先级相同),优先级最低的为 +、-(+ 和 - 优先级相同)。优先级给出了表达式的计算顺序。当计算一个具有多个运算符的算术表达式时,首先计算优先级最高的运算,然后计算比它低的。也就是说,最先进行 ** 运算,然后进行 *、/ 运算,最后才进行 +、- 运算(遇括号优先)。对于连续乘除的情况,则先左后右,例如,2 * 3 / 2 等效于 (2 * 3) / 2,结果为 3;对于连续加减的情况,也是先左后右,例如,2 + 3 - 4 等效于 (2 + 3) - 4,得 1;但是,对于连续乘幂的情况,则是先右后左,例如,2 * * 3 * * 2 等效于 2 * * (3 * * 2),即 $2^9 = 512$ 。

2. 算术运算对象与计算结果的类型

算术表达式的运算对象是数值型数据,其类型有整型、实型、双精度型和复型(这 4 种类型按等级由高到低的排列次序为复型、双精度型、实型、整型),其计算结果也是数值型

的。

在一个算术表达式中各算术运算对象为单一类型的情况下,其计算结果的类型同运算对象的类型。例如,算术表达式 $20+30$ 的计算结果 50 为整型, $2 * 3$ 的计算结果 8 为整型, $20. + 30.$ 的计算结果 50. 为实型, $2. * 3.$ 的计算结果 8. 为实型, $2D1+3D1$ 的计算结果 $5D1$ 为双精度型, $(1.2, 2.3) + (1.3, 2.4)$ 的计算结果 $(2.5, 4.7)$ 为复型。

特别要注意的是,两个整型量相除的结果为整型。因此, $7/4$ 的计算结果为 1 而不是 1.75, $7 * 6/3$ 等效于 $(7 * 6)/3$, 其计算结果为 14, 而 $7/3 * 6$ 等效于 $(7/3) * 6$, 其计算结果为 12, 可见 $I * J/K$ 与 $I/K * J$ 的结果不一定相同。

因为 $1/2$ 得 0 而不是 0.5, 可见, $X * (1/2)$ 等效于 X^0 而不是 $X^{1/2}$, $X^{1/2}$ 在 FORTRAN 中要写成 $X * 0.5$ 或 $X * (1.0/2.0)$ 。同理, $(1/2) * \sin(X)$ 等效于 $0 * \sin x$ 而不是 $\frac{1}{2} \sin x$, $\frac{1}{2} \sin x$ 在 FORTRAN 中要写成 $0.5 * \sin(X)$ 或 $\sin(X)/2.0$ 。 $2 * (-1)$ 同 $1/2$, 其结果为 0 而不是 0.5。 $5 * (-2)$ 同 $1/5 * 2$, 其结果为 0。

在一个算术表达式中算术运算对象为混合类型的情况下,出现混合运算。FORTRAN 对于混合运算是这样处理的:每进行一次运算前,先将等级较低的运算对象的类型转换为与等级较高的运算对象同一类型,然后才进行运算,结果取等级较高者的类型。例如, $7.5 * 6$ 要转化为 $7.5 * 6.0$ 后再计算,结果为 45.0。由于类型转换要费机器时间,此例最好直接写成 $7.5 * 6.0$ 。又如, $2 * 3.$ 要转化为 $2. * 3.$ 后再计算,结果为 8.。

要注意的是,双精度型数据不能与复型数据进行混合运算。在乘幂运算中,指数不能取复型数据。在乘幂运算的基数为复型数据时,其指数必须是整型数据。

1.5.2 关系表达式

关系表达式用来实现两个数值型数据(数值常数、数值型变量、算术表达式等)的比较或两个字符型数据(字符常数、字符型变量、子串、字符表达式等)的比较,其比较结果为 .TRUE. (真)或 .FALSE. (假),所以,可把关系表达式看作是逻辑型数据。

FORTRAN 提供的关系运算符有 .LT. (小于)、.GT. (大于)、.LE. (小于等于)、.GE. (大于等于)、.EQ. (等于)、.NE. (不等于)6 个。若指定的关系成立,比较结果为 .TRUE., 否则结果为 .FALSE.。

1. 数值型数据的比较

在两个数值型数据进行比较的情况下,比较对象有整型、实型、双精度型、复型 4 种类型,因此可能出现两个比较对象类型不一致的情况(例如,一个为整型,另一个为实型)。在此情况下, FORTRAN 先将等级较低对象的类型转化为与等级较高对象同一类型后才进行比较。例如,假设 X 为实型变量,其值为 5.0, I 为整型变量,其值为 5, 关系表达式 $X/2.0.GT.2.3$ 的值为 .TRUE.; 关系表达式 $I/2.GT.2.3$ 的运算中,先求出 $I/2$ 的值为 2, 经转化为 2. 后,才做是否大于 2.3 的比较,其结果为 .FALSE.。

在关系表达式中含算术表达式的情况下,要先完成算术运算,然后才进行关系运算。

要注意的是,双精度型数据不可与复型数据进行比较。复型数据只可与整型数据,实型数据或另一复型数据进行是否相等(.EQ.)或是否不相等(.NE.)的比较。

2. 字符型数据的比较

在计算机中,一个字符是以其代码来表示的。大多数计算机采用 ASCII 码或 EBCDIC