

国际标准化组织推荐文本

程序设计语言 FORTRAN

——
——

206

科学出版社

国际标准化组织推荐文本

程序设计语言 FORTRAN

韩淑娟 姚兆炜 译

曹德和 校

科学出版社

1980

内 容 简 介

FORTRAN 语言是目前国际国内十分流行的一种用于电子数字计算机的高级程序设计语言。为了便于交流算法和在各计算机上通用,对程序设计语言进行了标准化。本书译自被国际标准化组织(通常简称 ISO)一九七二年认可并出版的 ISO Recommendation R 1539, PROGRAMMING LANGUAGE FORTRAN。该标准语言由国际标准化组织的计算机和信息加工技术委员会 (ISO/TC97) 起草,并被当时 ISO 的二十一个成员国一致通过的一个 FORTRAN 标准语言。

本书可作为学习和了解 FORTRAN 语言及编制 FORTRAN 编译系统时的依据性资料,可供计算机程序设计人员、计算数学工作者、以及高等院校计算机软件专业的师生参考。

ISO Recommendation R 1539

PROGRAMMING LANGUAGE FORTRAN

Printed in Switzerland. 1972

国际标准化组织推荐文本 程序设计语言 FORTRAN

韩淑娟 姚兆炜 译
曹德和 校

*

科学出版社出版

北京朝阳门内大街 137 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1980 年 11 月第 一 版 开本: 787×1092 1/16

1980 年 11 月第一次印刷 印张: 5.1/4

印数: 0001—12,140 字数: 110,000

统一书号: 15031·312

本社书号: 1942·15—8

定价: 0.70 元

译者前言

FORTRAN (FORmula TRANslator) 是目前国际上十分流行的一种电子计算机高级程序设计语言。

1956年美国首先在 IBM 704 计算机上实现了当时的 FORTRAN II 语言。之后,各种大、中、小型计算机都配制有 FORTRAN 语言的编译系统,以供程序设计者用 FORTRAN 语言书写程序。

1966年美国标准化协会(ASA)公布了两级 FORTRAN 语言标准文本——X 3.9-1966 FORTRAN 和 X 3.10-1966 基本 FORTRAN。

1972年国际标准化组织(ISO)公布了三级标准 FORTRAN 推荐文本——基本级 FORTRAN、中间级 FORTRAN 和完全级 FORTRAN。此标准文本被国际上许多国家承认,各国也相应地制定了其本国的 FORTRAN 语言标准(如日、德、法等国)。

本手册是 1972 年 ISO 推荐的 FORTRAN 标准文本的中译本。

随着 FORTRAN 语言的广泛使用和编译技术的提高, FORTRAN 语言的标准化工作也在不断深入。1976年3月美国曾公布了一个 FORTRAN 语言的新标准草案。1978年4月该草案正式定稿,称为 X3.9-1978 FORTRAN (现许多文章中通称为 FORTRAN 77)。国际标准化组织将会相应地制定 FORTRAN 语言的新标准。然而,就目前情况来看,在国内外大量计算机上广泛使用的还是 1972 年的国际标准 FORTRAN。因此,本书可作为目前对于 FORTRAN 语言的编译、使用和教学的参考资料。

本文自 1973 年译成中文以来,曾在国内刊物上刊登和流传,许多同志对译文中的错误给予指正,在此对他们表示感谢。

1979 年 5 月

简 史

“国际标准化组织 (ISO)” 推荐文本 R 1539 《程序设计语言 FORTRAN》是由技术委员会 ISO/TC 97 (计算机和信息加工) 起草的, 委员会的秘书处由美国国家标准协会主持。

1968 年 4 月, ISO 推荐文本 No. 1539 草案曾在 ISO 全体成员国间征求过意见, 最后终于采用此草案。经少量编辑性的修改, 得到下列成员国的认可:

澳大利亚	德 国	葡萄牙
奥 地 利	爱尔兰	西班牙
比 利 时	以色列	瑞 典
巴 西	意大利	瑞 士
捷 克	日 本	土耳其
丹 麦	荷 兰	英 国
法 国	新西兰	美 国

全体成员国一致同意本草案。

于是, 本草案提交给 ISO 委员会, 该委员会决定接受它为 ISO 的推荐文本。

目 录

引言	1
1. 范围	1
2. 基本术语	2
3. 程序形式	3
3.1 FORTRAN 字符集	3
3.1.1 数字	4
3.1.2 字母	4
3.1.3 字母数字字符	4
3.1.4 特殊字符	4
3.1.4.1 空白字符	4
3.2 行	4
3.2.1 注解行	5
3.2.2 结束行	5
3.2.3 初始行	5
3.2.4 继续行	5
3.3 语句	5
3.4 语句标号	6
3.5 符号名字	6
3.6 字符的排列	6
4. 数据类型	6
4.1 数据类型的结合	7
4.2 数据类型的特性	7
4.2.1 整型	7
4.2.2 实型	7
4.2.3 双精度型	8
4.2.4 复型	8
4.2.5 逻辑型	8
4.2.6 Hollerith 型	8
4.3 存贮单元与数据之间的对应	9
5. 数据和过程的标识	9
5.1 数据和过程的名字	9
5.1.1 常数	9
5.1.1.1 整常数	10
5.1.1.2 实常数	10
5.1.1.3 双精度常数	10
5.1.1.4 复常数	10
5.1.1.5 逻辑常数	10

5.1.1.6	Hollerith 常数	11
5.1.2	变量	11
5.1.3	数组	11
5.1.3.1	数组元素	11
5.1.3.2	下标	11
5.1.3.3	下标表达式	11
5.1.4	过程	12
5.2	函数引用	12
5.3	数据和过程标识符的类型规则	12
5.4	虚拟变元	13
6.	表达式	13
6.1	算术表达式	13
6.2	关系表达式	15
6.3	逻辑表达式	15
6.4	表达式的求值	16
7.	语句	16
7.1	可执行语句	16
7.1.1	赋值语句	17
7.1.1.1	算术赋值语句	17
7.1.1.2	逻辑赋值语句	18
7.1.1.3	标号赋值语句	18
7.1.2	控制语句	18
7.1.2.1	GO TO 语句	19
7.1.2.1.1	无条件 GO TO 语句	19
7.1.2.1.2	赋值 GO TO 语句	19
7.1.2.1.3	计算 GO TO 语句	20
7.1.2.2	算术 IF 语句	20
7.1.2.3	逻辑 IF 语句	20
7.1.2.4	CALL 语句	20
7.1.2.5	RETURN 语句	20
7.1.2.6	CONTINUE 语句	21
7.1.2.7	STOP 语句和 PAUSE 语句	21
7.1.2.7.1	STOP 语句	21
7.1.2.7.2	PAUSE 语句	21
7.1.2.8	DO 语句	22
7.1.3	输入/输出语句	24
7.1.3.1	READ 语句和 WRITE 语句	25
7.1.3.1.1	输入/输出表	25
7.1.3.1.2	格式 READ 语句	26
7.1.3.1.3	格式 WRITE 语句	26
7.1.3.1.4	无格式的 READ 语句	26
7.1.3.1.5	无格式的 WRITE 语句	27
7.1.3.2	辅助输入/输出语句	27

7.1.3.2.1	REWIND 语句	27
7.1.3.2.2	BACKSPACE 语句	27
7.1.3.2.3	ENDFILE 语句	27
7.1.3.3	格式记录的印刷	28
7.2	非执行语句	28
7.2.1	说明语句	28
7.2.1.1	数组说明符	29
7.2.1.1.1	数组元素后继函数和下标值	29
7.2.1.1.2	可调维	30
7.2.1.2	DIMENSION 语句	30
7.2.1.3	COMMON 语句	30
7.2.1.3.1	公用块的对应	31
7.2.1.4	EQUIVALENCE 语句	32
7.2.1.5	EXTERNAL 语句	32
7.2.1.6	类型语句	33
7.2.2	数据初值语句	33
7.2.3	FORMAT 语句	34
7.2.3.1	字段描述符	34
7.2.3.2	字段分隔符	35
7.2.3.3	重复说明	36
7.2.3.4	格式控制与输入/输出表的相互作用	36
7.2.3.5	比例因子	37
7.2.3.5.1	比例因子的作用	37
7.2.3.6	数的转换	38
7.2.3.6.1	整转换	39
7.2.3.6.2	实转换	39
7.2.3.6.3	双精度转换	41
7.2.3.6.4	复转换	42
7.2.3.7	逻辑转换	42
7.2.3.8	Hollerith 字段描述符	42
7.2.3.9	空白字段描述符	43
7.2.3.10	数组中的格式说明	43
8.	过程和副程序	44
8.1	语句函数	44
8.1.1	语句函数定义	44
8.1.2	语句函数引用	45
8.2	内部函数及其引用	45
8.3	外部函数	47
8.3.1	函数副程序定义	48
8.3.2	外部函数引用	49
8.3.3	基本外部函数	50
8.4	子程序	51
8.4.1	子程序副程序定义	51

8.4.2	子程序引用	52
8.5	数据块副程序	53
9.	程序	54
9.1	程序的组成成分	54
9.1.1	程序部分	54
9.1.2	程序体	55
9.1.3	副程序	55
9.1.4	数据块副程序	55
9.1.5	主程序	55
9.1.6	可执行程序	55
9.1.7	程序单位	56
9.2	正常的执行顺序	56
10.	程序间的关系和程序内的关系	56
10.1	符号名字	56
10.1.1	关于类的限制	57
10.1.2	在说明语句和数据语句中涉及的一些关系	57
10.1.3	数组和数组元素	58
10.1.4	外部过程	59
10.1.5	子程序	59
10.1.6	语句函数	59
10.1.7	内部函数	59
10.1.8	外部函数	60
10.1.9	变量	60
10.1.10	块名字	61
10.2	定义	61
10.2.1	过程的定义	61
10.2.2	影响定义的结合	61
10.2.3	影响定义的事件	62
10.2.4	无名公用块中的实体	63
10.2.5	有名公用块中的实体	64
10.2.6	不在公用块中的实体	64
10.2.7	基本块	65
10.2.7.1	最后一个可执行语句	65
10.2.8	第二级定义	66
10.2.9	函数副程序中的某些实体	66
10.3	使用实体的定义要求	66
表		
表 1	“到”的赋值规则	17
表 2	下标值	29
表 3	内部函数	47
表 4	基本外部函数	50
附录W	有关 FORTRAN 标准化目的的一些考虑	68

W.1	引言	68
W.2	FORTRAN 的发展历史和现状	68
W.3	总的目的	69
W.4	用于研制本 ISO FORTRAN 推荐文本的一些准则	69
W.5	FORTRAN II 和 FORTRAN IV	69
W.6	用于各种设备的 FORTRAN 语言	70
附录 X	各节的注记	70
X.1	节 1 的注记	70
X.2	节 2 的注记	71
X.3	节 3 的注记	71
X.4	节 4 的注记	71
X.5	节 6 的注记	71
X.6	节 7 的注记	72
X.7	节 8 的注记	73
X.8	节 10 的注记	73
附录 Y	ISO FORTRAN 三级间的主要差别	73
附录 Z	FORTRAN 字符集的卡片穿孔表示	74

引 言

本 ISO 推荐文本建立用 FORTRAN 语言表达的程序形式, 以及用这种语言写的程序的解释, 目的是: 促使在各种自动数据加工系统上用的这样一些程序可高度互换。

本 ISO 文本规定的语言分为三级。倘若加工程序接受并按规定解释的话(至少对某一给定级符合这里所描述的那些形式和关系), 那末, 它应遵从此 ISO 文本。

在不影响所描述的形式和一些关系的解释范围内, 任何要求的陈述, 如果这种要求不满足, 则可由推荐文本未提供的某种解释所表达的陈述代替。进而, 任何禁止的陈述, 当违反这种禁止时, 可由推荐文本未提供的某种解释所表达的陈述代替。

在本 ISO 推荐文本中, 三级 FORTRAN 语言是用同样节号刊印的。当文本内容对此三级语言完全一致时, 那末, 整行印刷一次。当内容相异时(表格除外), 按下述形式刊印:

- (1) 如果三级全异, 那末, 第 2 级和第 3 级的内容各列在 1/3 行宽的框里;
- (2) 如果只是第 3 级不一样, 那末, 相应的内容也列在 1/3 行宽的框里;
- (3) 如果第 2 级和第 3 级一样, 但第 1 级不同, 那末, 第 2 级和第 3 级相应的内容列在 2/3 行宽的框里。

“完全 FORTRAN”、“中间 FORTRAN”和“基本 FORTRAN”三级分别标记为 1、2 和 3。

1. 范 围

本 ISO 推荐文本规定:

- (1) 用 FORTRAN 语言写的程序形式;
- (2) 由这种程序(它在自动数据加工系统上运行)加工的输入数据书写形式;
- (3) 解释这种程序含义的一些规则;
- (4) 只要解释规则规定了一种解释, 在自动数据加工系统上使用这种程序所得的输出数据形式。

本 ISO 推荐文本并不规定:

- (1) 为用于数据加工系统转换程序的方法(这种方法和数据加工系统合在一起称为加工程序);
- (2) 转录这种程序、输入这种程序、输出这种程序的方法, 以及把输出数据记到数据加工介质, 或从数据加工介质输出数据的方法;
- (3) 在数据加工设备上设置和控制这种程序的使用所要求的手动操作;
- (4) 当解释规则无法规定这种程序的一种解释时所得到的结果;
- (5) 程序的大小或复杂程度超过具体数据加工系统的容量, 或具体加工程序的能力;
- (6) 数量的范围或精确度。

2. 基本术语

本节介绍一些基本术语和概念。在以后有关各节中再给以严格的论述。这里列举的是关于语法形式及一些具体词含义的某些约定。

能用作完整计算过程的一个程序称为**可执行程序**(9.1.6)。

确切地说,一个可执行程序(9.1.6)是由一个主程序也可能再由一个或多个副程序所组成。

主程序(9.1.5)是不包含 FUNCTION 语句、SUBROUTINE 语句或 BLOCK DATA 语句的一组语句和注解构成的。

副程序(9.1.3 和 9.1.4)类似于主程序,所不同的是它用 BLOCK DATA 语句、FUNCTION 语句或 SUBROUTINE 语句开头以 BLOCK DATA 语句开头的副程序称为说明副程序。以 FUNCTION 语句或 SUBROUTINE 语句开头的副程序称为过程副程序。

主程序(9.1.5)是不包含 FUNCTION 语句或 SUBROUTINE 语句的一组语句和注解构成的。

过程副程序(9.1.3)类似于主程序,所不同的是它用 FUNCTION 语句或 SUBROUTINE 语句开头。过程副程序有时就称作副程序。

2

3

所谓**程序单位**(9.1.7)指的是主程序或副程序。

除说明副程序外,任何程序单位都可引用**外部过程**(见第 8 节)。

任何程序单位都可引用**外部过程**(见第 8 节)。

2

3

用 FORTRAN 语句定义的外部过程称为**过程副程序**。外部过程也可用别的方法定义。外部过程可以是一个外部函数或外部子程序。以 FUNCTION 语句开头的用 FORTRAN 语句定义的外部函数称为**函数副程序**。以 SUBROUTINE 语句开头的用 FORTRAN 语句定义的外部子程序称为**子程序副程序**(见 8、9 节)。

任何程序单位都是由若干**语句和注解**组成的。一个语句可分为若干自然段,每段称为**行**(3.2),第一行称为**初始行**,其余的都称为**继续行**。

有一种行称为**注解行**(3.2.1),它不是语句,只是为解释程序的作用而提供的信息。

FORTRAN 中的语句分为两大类:可执行语句和非执行语句(7.1 和 7.2)。可执行语句规定程序执行的动作;非执行语句描述程序的使用、运算对象的特点、编辑信息、语句函数及数据的排列。

语句的语法元素是**名字和运算符**。名字被用来指称一些对象,诸如数据或过程。运算符(包括命令动词)规定对命名的对象所施的动作。

数组名字应专门讨论。数组名字以及该数组的大小必须在数组说明符(7.2.1.1)中定义。只用下标限定的数组名字是用来标识该数组的某个具体元素(5.1.3)。

数据的名字和算术(或逻辑)运算符可联结成表达式。

数据的名字和算术运算符可联结成表达式。

计算这种表达式可得到一个值。这个值是对命名的数据执行规定的运算后获得的。

在 FORTRAN 中使用的标识符是名字和数。数据、过程都给以命名。语句是用数标记的。输入输出部件都给以编号(见 3、6、7 节)。

在本 ISO 推荐文本的不少地方,有一些语句带有有关的参数表。除明确说明外,在所有这样的情况里,都假定这种表至少要有一个参数。例如,在语句

SUBROUTINE *s* (a_1, a_2, \dots, a_n)

中,假定在括号内至少有一个符号名字是列在表里的。一个表就是一组可标识的元素,每个元素和它的后继元素用逗号分开。

引用一词如第 5 节中定义的那样,是具有特殊的意义而被使用的。

3. 程序形式

每个程序单位都是由组成行和语句的一些字符构成的。

3.1 FORTRAN 字符集

用下面一些字符来书写一个程序单位:

A、B、C、D、E、F、G、H、I、J、K、L、M、N、O、P、Q、R、S、T、U、V、W、X、Y、Z、
0、1、2、3、4、5、6、7、8、9 和

字 符	名 称	字 符	名 称
=	空 白	=	空 白
+	等 号	+	等 号
-	加 号	-	加 号
*	减 号	*	减 号
/	星 号	/	星 号
(斜 线	(斜 线
)	左 括 号)	左 括 号
,	右 括 号	,	右 括 号
.	逗 号	.	逗 号
“币符”	十进小数点	“币符”	十进小数点
币 符		币 符	

上面的字符顺序并不是经过整理后的顺序。

3.1.1 数 字

数字是下面十个字符之一：0、1、2、3、4、5、6、7、8、9。除另有规定外，当数制基的解释是合适的话，一个数字串将以十进制数来解释。

八进制数字是下面八个字符之一：0、1、2、3、4、5、6、7。它们只用在 STOP 语句(7.1.2.7.1)和 PAUSE 语句(7.1.2.7.2)中。

3.1.2 字 母

字母是下面二十六个字符之一：A、B、C、D、E、F、G、H、I、J、K、L、M、N、O、P、Q、R、S、T、U、V、W、X、Y、Z。

3.1.3 字母数字字符

字母数字字符指的是字母或数字。

3.1.4 特殊字符

特殊字符是下面十一个字符之一：空白、等号、加号、减号、星号、斜线、左括号、右括号、逗号、十进小数点和币符。

特殊字符是下面十个字符之一：空白、等号、加号、减号、星号、斜线、左括号、右括号、逗号和十进小数点。

2

3

3.1.4.1 空白字符

除特定使用外(见 3.2.2, 3.2.3, 3.2.4, 4.2.6, 5.1.1.6, 7.2.3.6 和 7.2.3.8), 空白字符是没有意义的, 只要遵守 3.3 中对继续行的限制, 可任意使用, 以改善程序的外观。

除特定使用外(见 3.2.2, 3.2.3, 3.2.4, 7.2.3.6 和 7.2.3.8), 空白字符是没有意义的, 只要遵守 3.3 中对继续行的限制, 可任意使用, 以改善程序的外观。

2

3

3.2 行

一行是由 72 个字符组成的字符串。除 5.1.1.6 和 7.2.3.8 中所讲的外, 所有字符必须为 FORTRAN 字符集里的字符。

一行是由 72 个字符组成的字符串。除 7.2.3.8 中所讲的外, 所有字符必须为 FORTRAN 字符集里的字符。

2

3

在一行中,各字符所在的位置称为列,并顺次记为 1, 2, 3, ..., 72. 每个数都表示一行中从左到右各字符的顺序位置.

3.2.1 注解行

位于一行的第 1 列的字母 C 表示该行为注解行. 注解行后必须紧跟初始行、另一注解行或结束行. 注解行丝毫不影响程序的原义,而其好处是为使用者提供方便.

3.2.2 结束行

结束行是在第 1 至 6 列中为空白字符的行,而在第 7 至 72 列中,字符按 E、N、D 的顺序各出现一次,在 E、N、D 前,或在 E、N、D 间,或在 E、N、D 后可有空白字符出现. 结束行指示加工程序所编写的程序单位(9.1.7)结束. 每个程序单位必须有一结束行以示终止.

3.2.3 初始行

初始行既非注解行又非结束行,且在第 6 列处应为数字 0 或空白字符,在第 1 至 5 列中可写语句标号或全为空白字符.

3.2.4 继续行

继续行是这样的行,即第 6 列为除数字 0 或空白字符以外的任何字符,且第 1 列不为字符 C.

继续行只可跟在某一初始行或另一继续行之后.

3.3 语 句

语句是由初始行其后顺序跟最多 19 个继续行所组成.

语句是由初始行其后顺序跟最多 9 个继续行所组成.

语句是由初始行其后顺序跟最多 5 个继续行所组成.

语句写在各行的第 7 至 72 列中.

语句中各字符的顺序是: 初始行的第 7 至 72 列, 当需要时, 后跟第一继续行的第 7 至 72 列、下一继续行的第 7 至 72 列等等.

在逻辑 IF 语句的情形中, 上面所讲的形式适用于整个语句, 而不是其内嵌语句 (7.1.2.3)

3.4 语句标号

语句可任意给以标记,使得在其它语句中能引用它。

语句标号由 1 至 5 个数字组成。

语句标号由 1 至 4 个数字组成。

3

语句标号所表示的整值无数值意义,但它必须大于 0。语句标号可写在该语句初始行的第 1 至 5 列的随便哪个位置上。在一个程序单位中,同一语句标号不能标记一个以上的语句。在区分语句标号时,左 0 不起作用。

3.5 符号名字

符号名字由 1 至 6 个字母数字字符组成,但必须以字母开头。

符号名字由 1 至 5 个字母数字字符组成,但必须以字母开头。

3

见 10.1 到 10.1.10 关于符号名字分类的讨论及对它们的用法所作的一些限制。

3.6 字符的排列

字符的排列是指在一个程序单位内说的。于是,组成名字、行和语句的任何有意义的字符集成,都是作为整个有序集而存在的。这种排列是受 3.2 中字符位置规则所制约(3.2 规定一行中各字符的排列)。而实际上也就是对按这种顺序列出的行进行加工的。

4. 数据类型

定义六种不同的数据类型。它们是整型、实型、双精度型、复型、逻辑型和 Hollerith 型。每种类型都有不同的数学意义并有相异的内部表示。

定义三种不同的数据类型。它们是整型、实型和逻辑型。每种类型都有不同的数学意义并有相异的内部表示。

2

定义两种不同的数据类型。它们是整型和实型。每种类型都有不同的数学意义并有相异的内部表示。

3

于是,当解释与数据有关的运算时,数据类型就有了意义。函数的数据类型定义了该

函数在其中出现的那个表达式的一个数据的类型。

4.1 数据类型的结合

用来标志一个数或函数的名字本身就附有数据类型。表示一个常数的字符串既定义值又定义数据的类型。

对每个程序单位来说,表示函数、变量或数组的符号名字必须只有一种数据类型与之结合。一个特定的名字,一旦结合以某种具体数据类型后,那末,不管怎样使用它总是该类型,即要求在定义它的整个程序单位里只有一种数据类型。

对符号名字可通过类型语句(7.2.1.6)建立数据的类型,说明为整型、实型、双精度型、复型和逻辑型。这种专门的说明就废弃对整型和实型所用的隐含结合(见 5.3)。

对符号名字可通过类型语句(7.2.1.6)建立数据的类型,说明为整型、实型和逻辑型。这种专门的说明就废弃对整型和实型所用的隐含结合(见 5.3)。

对符号名字用该名字的第一个字符建立数据的类型(见 5.3)。

没有一种手段可把符号名字和 Hollerith 数据类型相结合。于是,和常数不一样,它是假借其它类型之一的名字来标识这种类型的数据的。

4.2 数据类型的特性

在下面几段中定义每种数据类型的数学性质和表示特点。

对实型、双精度型和整型数据来说,值 0 既不看作作为正的,又不看作为负的。

对实型和整型数据来说,值 0 既不看作作为正的,又不看作为负的。

4.2.1 整 型

整型数据总是一个整值的明确表示。它可为正的、负的和零值。但它只能为整值。

4.2.2 实 型

实型数据是一个实数值在加工程序表示的意义下的一个近似值。它可为正的、负的