

计算机语言袖珍指南之四

FORTRAN

P. 里德勒 著

上海翻译出版公司

计算机语言袖珍指南之四

F O R T R A N

[英] P. 里 德 勒 著

季 学 翻 译

张 汉 良 校

上海翻译出版公司

Pocket Guide to FORTRAN

Philip Ridler

PITMAN BOOKS LTD, 1982

计算机语言袖珍指南之四

FORTRAN

〔英〕P. 里德勒 著

季学翻 译

张汉良 校

上海翻译出版公司出版

(上海福州路 390 号)

新华书店上海发行所发行 上海教育学院印刷厂印刷

开本 787 × 1092 1/64 印张 1 37/64 字数 45,700

1985 年 4 月第 1 版 1985 年 4 月第 1 次印刷

印数 1—74,800

统一书号: 13311·5 定价: 0.36 元

出版说明

英国皮特曼图书公司于1982年出版了一套入门性质的计算机语言袖珍指南，共计五本。第一本介绍程序设计的基础知识，后四本分别介绍BASIC、COBOL、FORTRAN、PASCAL这四种计算机常用语言的基本概念、程序结构、基本语句等内容。

作者针对不熟悉计算机的初学者想学而又无从着手这一特点，通过最简单的程序例子来介绍每一种语言中各类语句的基本内容，而对使用语言中的编程技巧不花过多的笔墨，从而使初学者能很快掌握要领，所以这套指南确实是少而精的计算机常用语言入门读物。对于初步掌握一种语言又想涉猎其它几种语言的读者，这套书也提供了入门的捷径。

当然，由于篇幅上的限制，这套书存在一些不足之处。好在国内已出版了不少关于计算机语言的专门著作，已经入门想进一步深造的读者，可以去参考、钻研这些著作。

目 录

如何使用这本袖珍指南.....	1
一、引言.....	2
1. 字符集.....	2
2. 名字.....	3
3. 程序行.....	3
4. 语句.....	5
5. 空白列.....	6
6. 顺序号.....	6
7. 数据卡片.....	7
二、常数, 变量和数组.....	7
1. 常数.....	7
2. 变量.....	10
3. 数组.....	13
三、算术和逻辑表达式.....	14
1. 算术运算符.....	15
2. 逻辑表达式.....	19
3. 函数引用.....	21
4. 溢出.....	21

四、赋值语句.....	22
1. 算术赋值语句.....	22
2. 逻辑赋值语句.....	25
五、控制语句.....	25
1. 无条件 GO TO 语句	26
2. 计算 GO TO 语句	26
3. 赋值 GO TO 语句	27
4. ASSIGN 语句	27
5. 算术 IF 语句	28
6. 逻辑 IF 语句	28
7. DO 语句	29
8. CONTINUE 语句	33
9. CALL 语句	33
10. PAUSE 语句.....	33
11. STOP 语句.....	34
六、程序结构.....	35
1. 段和函数.....	35
2. 主程序段.....	36
3. SUBROUTINE 段	36
4. FUNCTION 段	38
5. RETURN 语句.....	40
6. 函数.....	40

7. 变元.....	49
七、说明语句.....	52
1. 类型语句.....	52
2. DIMENSION 语句	53
3. EXTERNAL 语句	54
4. COMMON 语句	55
5. EQUIVALENCE 语句	57
6. DATA 语句	58
7. 输入和输出.....	60
8. READ 语句	62
9. WRITE 语句.....	64
八、FORMAT 说明	65
1. FORMAT 语句.....	66
2. FORMAT 说明	66
3. 复数变换.....	73
4. 重复计数.....	78
5. 比例因子.....	79
6. 字段分隔符.....	80
7. 格式控制.....	82
8. 作为数据的格式说明.....	86
9. 磁介质控制语句.....	88

九、标准 FORTRAN 语句表	90
十、段内语句的排列顺序	92

如何使用这本袖珍指南

FORTRAN 语言的各种特性都有专门的单元予以论述，文中出现的大写项（例如，INTEGER，SUBROUTINE 等）是 FORTRAN 程序中的关键字。本书描述了 FORTRAN 各种特性的用法（即语言的语法规则）以及编译程序的一些限制，在许多场合里还举了不少典型 FORTRAN 语句的例子。

本指南的最后部分附有标准 FORTRAN 语句表，并推荐一个大多数编译程序能接受的语句顺序结构。

注释

FORTRAN 关键字用大写字母印出

例如，LOGICAL 常数

每当字母“O”和数字“0”用于 FORTRAN 中时，为了将它们区别开来，通篇文章里的字母“O”写成“①”，数字“0”仍为“0”。（为了读者阅读方便，在译文中字母“O”仍写成“O”——译者）

本指南是专为标准 FORTRAN 而写的,但用户还应知道一些局部特性,譬如名字的长度、A 型转换的最大字符数以及所允许数组下标的个数等,这些经常与标准 FORTRAN 的规定是不同的。象输入输出设备的个数、整数和实数的最大范围、一个数据存储单元的大小以及 COMMON 语句的最大体积等也都是随机而异的。

一、引言

1. 字符集

标准 FORTRAN 语言可用下面这些字符:

英文字母表里的字母 A~Z

十进制数字 0~9

特殊字符 = + - / * () , · 空格 \$

“字母数字”字符指的是字母或数字。

注释

(1) 在该指南里,用 B 来表示一个空格字符。

(2) 为了避免字母“O”与数字“0”混淆,字母

“0”写成“①”。

(3) 十进制小数点必须写成“.”，不能写成“，”。

2. 名字

名字是一串字母数字序列，仅用来表示变量，数组，函数，子程序和 **COMMON** 区等项。名字的第一个字符必须是字母。在标准 **FORTRAN** 中，名字的长度可达 6 个字符。

一般说来，空格是毫无意义的。例如，**AM3**，**A M3** 和 **AM 3** 被认为是相同的。

名字我们可自由选择，但第一个字母含有特殊意义。为保险起见，尽量避免把标准函数的名字作其它用。名字在一个程序段里只能有一个含义。用在不同程序段里的同一名字并不表示同一物理量。上述的后两条规则对 **COMMON** 区的名字不适用。

3. 程序行

程序行是一串 **FORTRAN** 字符，其中包括空格在内可达 72 个字符。拿穿孔卡片来说，字符的位置就是指列。在 **FORTRAN** 程序中，第 72 列以后的

字符是不起作用的。程序行有四种不同类型：起始行，续行，结束 (END) 行和注释行。

起始行

起始行是一个语句的第一行，它不是注释行也不是结束 (END) 行。第 1~5 列只允许写数字，第 6 列只允许写一个空格或零。

通常，一个程序的大部分是由起始行组成的。

严格地说，空白行是起始行，不应包括在程序里。然而，大多数编译程序是不理睬空白行的。

续行

续行是一个语句的任何一行，但不是起始行。续行的第 6 列上必须有一个数字或字符，但不能是空格或零。在续行的第 1~5 列上写任何字符编译程序都不理睬，但这不是一个好的编程习惯。

在标准 FORTRAN 语言里，一个语句最多可以有 19 个续行。

续行只能紧跟在起始行或其它续行之后。

END 行

END 行是在第 7~72 列上只出现字符 END 和空格的行。它不是语句，不能有标号。

END 行用来向编译程序表明此程序段到此结束，程序执行时一定不能执行到 END 行。因此，END 行的前一行必需是 STOP 或转移语句。

每个程序段必须用 END 行结束。

注释行

注释行是第 1 列上含字母 C 和第 2~72 列上含任何 FORTRAN 字符的行，它不是一个语句。

注释行对程序没有任何影响；只不过使程序员能在程序中间插入一些解释性的文字。注释行可在程序的任何地方出现，但不能插在一个有续行的语句的各行之间。

注释行没有续行，于是每个注释行必须用字母 C 开始。

4. 语句

FORTRAN 语句写在第 7~72 列上，每个语句写一行。如果语句太长，可以换行接着写，但每一续行的第 6 列上要写一个非空格或零的字符。续行

不能有标号。起始行的第 6 列上必须是空格或零。

语句标号（数字）

任何 FORTRAN 语句都可以有标号，该标号是 1~99999 中的任何一个整数，写在语句起始行的第 1~5 列的任何位置上。空格和前导零不起作用。

数字标号是标识一个语句的一种方法，使得该语句能被程序中其它语句引用。语句标号的数字是否按顺序排列其意义不大，但按顺序排列的语句标号看上去更清楚些。

在一个程序段中不允许出现两个相同的标号。

所有 FORMAT 语句都必须有标号，但是给任何其它非执行语句加个标号是没有意义的，因为程序无法利用这样的标号。

5. 空白列

在 FORTRAN 中空白列是没有意义的（字符常数里除外），它们可随便运用来提高可读性。

6. 顺序号

若输入卡片上含有一些 FORTRAN 程序行，则

第 73~80 列的内容不被理睬, 这些列可用来存放标识码或顺序号。

7. 数据卡片

穿孔卡片的 80 列全部用来存放数据, 但一张数据卡片并不是一个 FORTRAN 语句。

二、常数, 变量和数组

1. 常数

在 FORTRAN 中, 出现在语句里却又不是语句标号的任何数叫作常数。

分配给一个常数的变量名必须与该常数的类型一致。

整型 (INTEGER) 常数

INTEGER 常数是整数, 由一串数字组成, 前面可加上一个符号。它不可能出现小数点、逗号或指数。如果没有符号, 该常数被认为是正数。整数的机内表示值是没有误差的。

实型 (REAL) 常数

REAL 常数又名浮点型常数, 它是含有十进制小数点的数。

浮点型常数后面也可随意地跟一个十进制指数, 写成字母 E 再加上一个具有一位或二位数字的整型常数 (有符号或无符号), 表示 10 的幂。

由于机器字长的位数有限, 所以 REAL 常数机内的表示值可能不是它的确切值。

双精度型 (DOUBLE PRECISION) 常数

DOUBLE PRECISION 常数除了用 D 代替 E 外, 其书写形式与 REAL 常数的指数形式相同。两者唯一的区别是: DOUBLE PRECISION 常数有更多的数字, 所以能更精确地代表所表达的值。一个 DOUBLE PRECISION 常数占两个存储单元。

复型 (COMPLEX) 常数

COMPLEX 常数的形式是 (A, B), 其中 A 和 B 只能是合法的实型常数, 圆括号是不能省略的。A 是复数 $A + \sqrt{-1} B$ 的实部, B 是虚部。

一个 COMPLEX 常数占两个存储单元。

常数的范围

常数的取值范围取决于所用的机器。

对于一个存储单元为 16 位的机器来说，一个 INTEGER 常数不能超过 $\pm 32\,768$ ，而 32 位的机器能处理 $\pm 2\,147\,483\,648$ 范围内的整数。

REAL 常数的范围是相当不确定的，因为它取决于一个存储单元中有几位分给尾数，几位分给指数。典型的情况是：一个 32 位存储单元的范围可能是 $\pm 9.2 \times 10^{18}$ ，48 位的范围可能是 $\pm 5.8 \times 10^{76}$ 。

逻辑型 (LOGICAL) 常数

LOGICAL 常数只有两种，写作：

• TRUE •

和 • FALSE •

两个句点很重要。

LOGICAL 变量的值只能取这两个常数中的一个。

文字型 (Hollerith) 常数