

TQ-16机

**FORTRAN**

**语 言**

大连工学院应用数学系  
大连工学院工程力学系 合编  
中国科学院计算中心

大 连 工 学 院

TQ—16 机  
FORTRAN 语 言

大连工学院

TQ-16 机 FORTRAN 语言

---

出版：大 连 工 学 院  
发行：大连工学院应用数学系  
印刷：大连工学院印刷厂

---

787×1092×1/32      字数：15 万字  
1980 年 11 月出版    1980 年 12 月印刷  
大工印刷编号：80-342    册数：4000 册

# 前 言

*FORTRAN* 语言是国际上应用最为广泛的程序设计语言之一。近年来，随着科学技术的发展和计算机的广泛应用，我国科技界要求使用 *FORTRAN* 语言的呼声越来越高。为适应这一需要，我们为国产的 TQ—16 机配制了 *FORTRAN* 语言。经过一年多的试算，我们对 *FORTRAN* 语言编译程序做了部分修改，使之尽量与国际标准 *FORTRAN* 一致。并且增加了语句函数和输入输出语句中的隐循环，配上了管理程序和标准算法库。为此，我们重新编写这本小册子。衷心希望广大用户在试用过程中提出宝贵的意见。

大连工学院数学、力学系

中国科学院计算中心

一九八〇年

# 目 录

<b>第一章 概 述</b> .....	( 1 )
§ 1—1 <i>FORTRAN</i> 程序举例.....	( 1 )
§ 1—2 程序的结构与语句的分类.....	( 4 )
§ 1—3 字符与基本符号.....	( 6 )
<b>第二章 数据及其类型</b> .....	( 13 )
§ 2—1 常数、变量与类型语句.....	( 13 )
§ 2—2 数组与维数语句.....	( 16 )
§ 2—3 说明语句的顺序.....	( 21 )
§ 2—4 函 数.....	( 22 )
§ 2—5 表达式.....	( 27 )
第二章 练习.....	( 31 )
<b>第三章 基本可执行语句</b> .....	( 33 )
§ 3—1 简单的输入输出语句的例.....	( 33 )
§ 3—2 赋值语句.....	( 34 )
§ 3—3 条件语句.....	( 37 )
§ 3—4 转向语句.....	( 41 )
§ 3—5 停语句与暂停语句.....	( 47 )
§ 3—6 继续语句.....	( 48 )
§ 3—7 循环语句.....	( 48 )
第三章 练习.....	( 61 )
<b>第四章 输入输出及格式语句</b> .....	( 63 )
§ 4—1 读语句和写语句.....	( 63 )

§ 4—2 格式语句	( 69 )
§ 4—3 格式语句与读写语句的关系	( 71 )
§ 4—4 各种格式介绍	( 72 )
附例	( 84 )
<b>第五章 程序段与程序段之间的联系</b>	( 89 )
§ 5—1 语句函数	( 90 )
§ 5—2 过程段	( 93 )
§ 5—3 函数引用, 调用语句	( 102 )
§ 5—4 外部语句	( 110 )
§ 5—5 公用语句和等价语句	( 111 )
第五章 练习	( 120 )
<b>第六章 程序库使用说明</b>	( 132 )
§ 6—1 库的调用	( 133 )
§ 6—2 库的扩充	( 134 )
<b>第七章 调试与修改措施</b>	( 136 )
§ 7—1 语法错误与修改措施	( 136 )
§ 7—2 调试措施	( 141 )
§ 7—3 调态与算态	( 145 )
<b>第八章 上机说明</b>	( 148 )
§ 8—1 纸带穿孔	( 148 )
§ 8—2 操作说明	( 160 )
§ 8—3 各种操作命令	( 165 )
§ 8—4 编译信息表	( 172 )
附录 1 控制台打字机输出/入符号表	( 176 )
附录 2 库过程目录表	( 177 )
附录 3 语法错误的编号与性质对照表	( 186 )

# 第一章 概 述

*FORTRAN* 语言是一种描述科技计算问题的语言。假定我们要解决一个计算问题，当计算的方法和步骤确定了之后就可以用 *FORTRAN* 语言把它描述出来。这种用 *FORTRAN* 语言描述的计算步骤通常称之为程序。用 *FORTRAN* 语言来描述计算问题的工作叫做程序设计，或者叫做编程序。编程序是利用电子计算机解题的核心工作。这本小册子就是要介绍怎样用 *TQ-16* 机的 *FORTRAN* 语言编程序。

这一章，先给出一个例子，概括地说一说 *FORTRAN* 程序是什么样子的，并介绍一些最基本的术语和概念，以便于以后讨论。

## § 1—1 *FORTRAN* 程序举例

**例：**试求两个三角形的面积之和，这两个三角形的三个边分别是

$X$ ,  $2X$ ,  $Y$  和  $3(Z+W)$ ,  $19.7$ ,  $Z^2$

下面就是这个问题的 *FORTRAN* 程序和数据（这里假定  $X=4.5$ ,  $Y=7$ ,  $Z=3.2$ ,  $W=4.6$ ）

6\*6\*

光电输入开始标志

PAGEε001;

页标志，第一页

MASTER ε SUM;

名为 *SUM* 的主程序开始

{ CALCULATE AREA SUM; 注解，说明本程序

是求面积和的。

*REAL*  $\varepsilon$  *X, Y, Z, W*, 类型说明, 定义 *X, Y, Z, W*, *TOTA* 是实型量

*TOTA*;

*READ*  $\varepsilon$  (3, 10) 输入, 将纸带上的数据输入到变量 *X, Y, Z, W* 中。

10 *FORMAT*  $\varepsilon$  (*N*); 输入的格式是 *N* 格式。

*TOTA* = *AREA*(*X*, 2 \* *X*, 调用函数 *AREA*  
*Y*) + *AREA*(3 \* (*Z* + 计算两三角形的面积和  
*W*), 19.7, *Z* \* *Z*); 并赋给变量 *TOTA*。

*WRITE*  $\varepsilon$  (4, 20) *TOTA*; 按定点数格式将面积

20 *FORMAT*  $\varepsilon$  (F6.2); 和 *TOTA* 输出来。

*STOP*  $\varepsilon$ ; 停机。

*END*  $\varepsilon$ ; 主程序结束。

*PAGE*  $\varepsilon$  002; 第二页。

{ *CACULATE AREA*; 注解, 说明以下是计算面积。

*FUNCTION*  $\varepsilon$  *AREA* 计算三角形面积的函数开始。

(*A, B, C*);

*REAL*  $\varepsilon$  *A, B, C, S*; 类型说明, 定义 *A, B, C, S* 是实型量。

*S* = (*A* + *B* + *C*) / 2; 计算三边和之半。

*AREA* =  $\sqrt{S * (S - A) * (S - B) * (S - C)}$ ; 计算三角形的面积。

*RETURN*  $\varepsilon$ ; 返回主程序。

*END*  $\varepsilon$ ; 函数段结束。

*FINISH*  $\varepsilon$ ; 整个程序结束。

$\wedge \wedge \dots \wedge 6 * 6 *$  光电输入结束标志。



6*6*	光电输入开始标志
4.5, 7, 3.2, 4.6;	数据部分
^^.....^6*6*	光电输入结束标志

这是一个可以在 TQ—16 机上执行的 FORTRAN 程序。整个程序分为两部分：第一部分由第一个 6\*6\* 符号开始到第一个 ^^.....^6\*6\* 符号为止。这一部分称为程序部分（通常叫做源程序）。第二部分是数据部分，由第二个 6\*6\* 起到第二个 ^^.....^6\*6\* 止。

当要上机计算时，首先把上面的程序在穿孔纸带上穿孔，然后通过光电输入机输入到机内并执行就可算出结果：

#### 115.41

从上例可以看出，一个 FORTRAN 程序是由许多行组成的。每一行后面都跟一个 ‘；’ 号。除了下面的一些特殊的行外，每一行都叫做一个语句。每个语句都表达了一个完整的意思。

这些特殊的行是：

(1) PAGE ε 是页标志。如果源程序太长（比如超过了100行）就得分页，即在源程序的适当地方写上 PAGE ε，PAGE ε 要单独做为一行。PAGE ε 是统计一页中行数的起点，它本身算是第 0 行。以后每遇一个 ‘；’ 就算一行。统计页行是为了指出程序中出错的确切位置。本语言规定整个程序不能超过 100 页，每页不能超过 100 行。

(2) 以 ‘〔 ‘ 开头的行称为注解行。在 〔 和其后的第一个 ‘；’ 之间的内容只起注释作用，不影响程序的执行。注解行主要是为了对程序加注解或说明以便于阅读和交流。注解行作为独立的一行可以写在程序的任何地方，但不能写在另一行的内部。也就是说 ‘〔 ‘ 号只能跟在 ‘；’ 的后面，或

者是整个程序的第一个符号。

(3) *END* ε 称为结束行。它标志着一个程序段的结束。

(4) *FINISH* ε 它表示整个程序的结束。它总是放在整个源程序的最后并且不能省略。

(5)  $\wedge \wedge \dots \wedge$  是由 15 个以上的  $\wedge$  组成的，这是为了保证程序（或数据）能全部输入到机器中去，因此这 15 个以上的  $\wedge$  号是必须要有的。

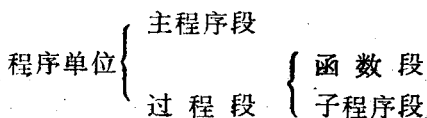
由此，从书写的角度来看，*FORTRAN* 程序是分页的（以 *PAGE* ε 为标志），页内是分行的（以；为标志）。

## § 1—2 程序的结构与语句的分类

这一节我们将从程序结构的角度来考察 *FORTRAN* 程序。

### 1—2.1 程序结构

§ 1—1 中的源程序可以分为两大段：第一段是从 *MASTER* ε 开始到第一个 *END* ε 为止。这一段称为主程序段。它是整个程序的控制部分。第二段是从以 *FUNCTION* ε 打头的行开始到第二个 *END* ε 为止，这一段称为过程段。过程段有两种，一种是以 *FUNCTION* ε 打头的，称为函数段。还有一种是以 *SUBROUTINE* ε 打头的，称为子程序段。一般地，一个 *FORTRAN* 源程序是由一个或几个相对“独立”的程序段所组成。这些程序段在 *FORTRAN* 语言里称为程序单位。程序单位分为主程序段和过程段。过程段又分为函数段和子程序段：



一个 *FORTRAN* 程序或者只由一个主程序段组成，或者由一个主程序段和若干个过程段组成。也就是说每一个程序都必须有一个主程序段而且只能有一个主程序段。主程序段的结构如下：

```
MASTER ε 主程序名；  
…… (语句或行，称为程序体)  
END ε；
```

象 *MASTER* ε 主程序名；和 *FUNCTION* ε 函数名 (参数)；等，它们是一个程序段的标志，称之为段头语句。一个最简单的程序是：

```
6*6*  
MASTER ε MST；           主程序语句  
STOP ε；                 停机语句  
END ε；                 结束行  
FINISH ε；  
∧∧……∧6*6*
```

其中 *STOP* ε；是停机语句 (§ 3—5) 它是主程序不可少的成份。这个程序什么工作也不做，但却是一个合法的程序，它表示了一个程序的必须要有的部分。

当一个程序是由主程序段和过程段组成时，它们之间的次序可以是任意的。主程序段可以放在任何位置上。我们把 *FORTRAN* 语言的这种结构称为段状结构。

### 1—2·2 语句的分类。

*FORTRAN* 语言的每个程序单位都由若干语句组成。语句又分为可执行语句和不可执行语句两类。如例中的 *MASTER* ε *SUM*；*REAL* ε *X*，*Y*，*Z*，*W*，*TOTA*；*FUNCTION* ε *AREA*(*A*，*B*，*C*)；等是不可执行语句。它们分别定义了 *SUM* 是一个主程序段，*X*，*Y*，*Z*，*W*，

*TOTA* 是实型变量，*AREA* 是以 *A*, *B*, *C* 为自变量的函数，例中的  $S = (A+B+C)/2$  是可执行语句。本语言的语句详细分类如下：（见第 7 页）

### 1—2.3 语句标号

有些语句的前面可以放上一个整数，以便于其它的语句来引用它。这个整数称为语句标号或简称为标号。例如，语句 *FORMAT*  $\epsilon$  (*N*) 前的整数 10 就是该语句的标号。

对于语句标号要注意以下几点：

(1) 语句标号只能是 1 到 99999 之间的整数，因此 0 不能作为标号。

(2) 整数作为标号来用时没有数值上的意义，也不表示语句出现的顺序。例如标号为 15 的语句可以在标号为 1980 的语句之后。标号左边的 0 不起作用，例如标号 015 与标号 15 是一样的。

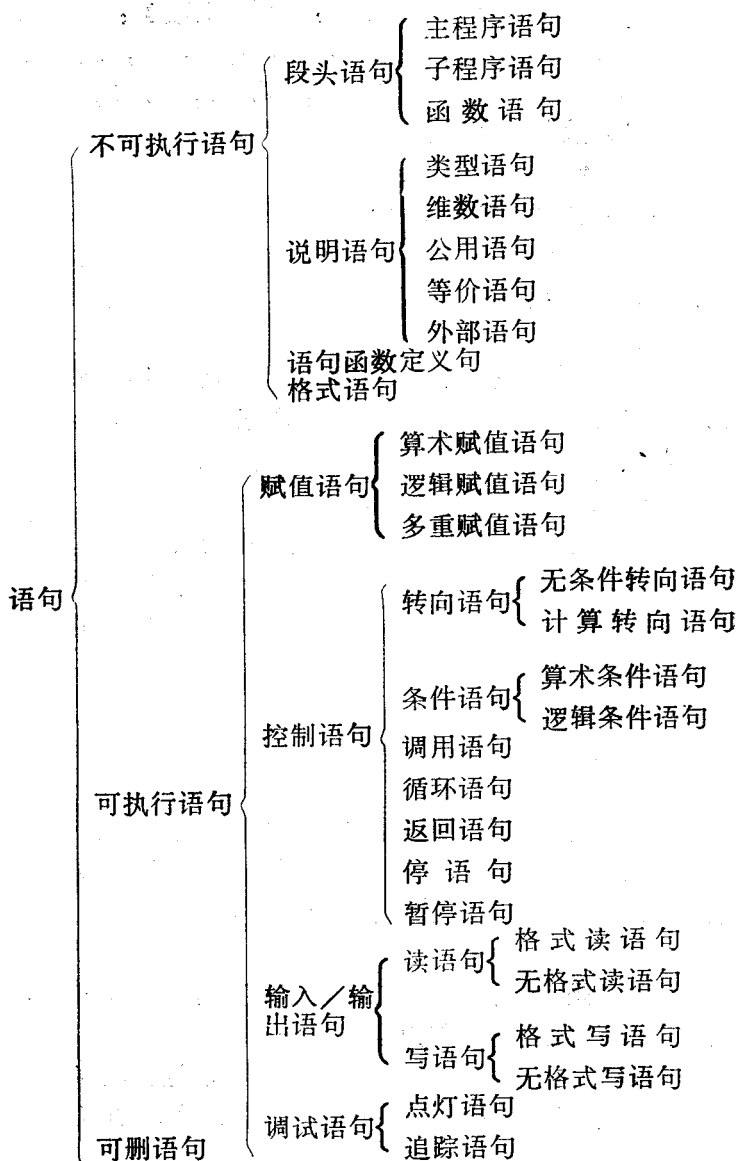
(3) 除了格式语句外，其它的不可执行语句是不能带标号的。结束行，与注解行也不能有标号。

(4) 在一个程序单位中，不同的语句不能有相同的标号。在两个不同的程序单位中可以使用相同的标号。也就是说，标号是局部于程序单位的。

## § 1—3 字符与基本符号

### 1—3.1 *FORTRAN* 字符集

由例子可以看出，书写一个 *FORTRAN* 程序的最基本的单位是象字母 *a*, *b*, ……*c*, 数字 0, 1…和 +, - 等这样一些符号。在 *FORTRAN* 中，这套符号叫做 *FORTRAN* 字符集。集中的每一个符号称为一个字符。*TQ—16* 机 *FORTRAN* 语言的字符集包括以下 64 个字符：（见第 8 页）



(1) 字母 (26 个拉丁字母, 大小写不分) :

*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.*

(2) 数字 (10 个) :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9。

(3) 其它字符 (28 个) :

+, -, \*, /, ÷, ↑, √, ∧, ⊥, <, ≤, =, ≥, >, ×, :, ;, #, (, ), [, ], , , ε, □, ·, 10, ‘。

其中 ‘□’ 表示空白。在印刷上空白是看不见的符号, 为了醒目起见, 在必要时用记号 ‘□’ 来表示空白。字母 O 容易与数字 0 混淆, 建议将字母 O 写为 ∅。

字符是 *FORTRAN* 程序的最基本的书写和印刷单位。程序中不允许出现字符集以外的字符。一般来说, 字符在语法上没有独立的意义。用它们组成整数, 符号名和定义符后, 在语法上就有独立的意义。我们可用它们来描述语句的各种成份。

### 1-3.2 整 数

整数是由一系列数字组成的。它的写法与通常数学上的写法是一样的。例如: 329, 2820, 075 等都是整数。

整数有两个用处: 一是用来组成常数 (见 §2-1.1)。二是用来标识语句, 即用来做语句的标号。此时整数就没有数值意义, 只有标识作用 (见 §1-2.3)。

### 13.3 符 号 名

符号名是由以字母开头的字母数字序列组成的。例如, 例子中的下列符号序列都为符号名:

*SUM*

表示主程序的名字

*X, Y, Z, W*

表示变量的名字

*AREA* 表示函数的名字

此外象 *A12*, *BIJ*, *D3I* 等都是符号名。

下列的符号序列就不是符号名:  $C+2$ ,  $a\cdot 2$ ,  $36B$ ,  $-a$  等。

符号名是专用来给变量, 数组, 函数, 子程序, 公用区, 哑元起名字的。

符号名的选取, 最好是与原来的计算问题或计算公式中的符号一样或相似, 以便于记忆。

例如

公式中的符号

符号名

$a, b, c$

$A, B, C$

$a_1, b_2$

$A_1, B_2$

$aij$

$AIJ$

$\epsilon$

$EPS$

$\alpha, \beta, \gamma$

$ALPHA, BETA, GAMMA$

关于符号名的使用, 要注意两点:

(1) 在同一个程序中, 不能用同一个符号名作为两个不同的量的名字 (名字不许用重)。

(2) 本语言规定符号名只有前四个字符有意义, 后面的字符可有可无。例如:

$ABSDF$ ,  $ABSDG$ ,  $ABSDGZ$  被认为是同一个符号名。

### 1-3.4 逻辑值

逻辑值只有两个, 它们是 *TRUE* (真) 和 *FALSE* (假) 逻辑值是用来表示逻辑常数的。

### 1-3.5 定义符

在描述 *FORTRAN* 语言的各种语句时要用到许多定义符。我们根据它们的作用把这些定义符分成四类。现在把





### 1-3.6 专用名

专用名是用来区别各种语句的。专用名一般是用一个英文字后面跟一个‘ε’组成。专用名中只有四个字母有意义，后面的字母可有可无，但‘ε’号不能丢掉。专用名可以简写。兹列表如下：

英文字	简写字	含义
MASTER ε	Me	主程序
FUNCTION ε	FU ε	函数
FORMAT ε	F ε	格式
SUBROUTINE ε	SU ε	子程序
STOP ε	S ε	停机
REAL ε	无	实型
READ ε	R ε	读
RETURN ε	RE ε	返回
INTEGER ε	I ε	整型
IF ε	无	如果
LOGICAL ε	L ε	逻辑型
COMMON ε	CO ε	公用
CALL ε	CA ε	调用
CONTINUE ε	C ε	继续
GOTO ε	G ε	转向
DIMENSION ε	D ε	维数
DO ε	无	循环
DD ε	无	点灯
WRITE ε	W ε	写
PAGE ε	PA ε	页
PAUSE ε	P ε	暂停
FINISH ε	FI ε	完