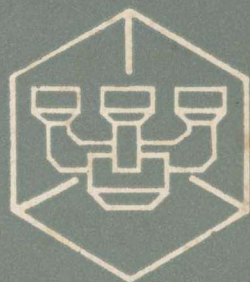


FORTRAN 77

程序设计及应用



中国建筑学会科技咨询中心

FOOTBALL

程序设计及应用



清华大学出版社

FORTRAN 77 程序设计及应用

霍 新 民 等

中国建筑学会科技咨询中心

一九八五年一月

内 容 提 要

本书全面系统地介绍 FORTRAN 77 语言(全集)和程序设计的基本方法。全书共十二章,前两章介绍程序设计的基础知识,后十章讲解 FORTRAN 77 全集语言。书末附有 FORTRAN 77 全集对 FORTRAN IV 的扩充与限制以及全集语言与子集语言的区别。

本书的特点是由浅入深,通俗易懂,内容详尽,例题丰富。对所有概念和程序设计实践中经常遇到的问题均有例题加以说明,对 FORTRAN 77 引入的新的特征予以重点介绍。

本书可供工程技术人员和大专院校师生进行程序设计时参考,亦可供初学者和自学者阅读。

FORTRAN 77 程序设计及应用

霍新民 汪琪美

*

中国建筑学会科技咨询中心出版
北 京 百万庄

*

责任编辑:王国泉 吴华强
封面设计:白建新

*

1985 年 4 月 内部发行 定价 4.62 元

前 言

新的技术革命正在蓬勃发展,电子计算机在这场革命中所起的作用,已日益被人们认识和肯定,学习电子计算机形成了热潮。为了帮助大家掌握电脑使用中普遍采用的 **FORTRAN 77** 语言,我们组织出版了霍新民等同志编写的《**FORTRAN 77** 程序设计及应用》一书。

本书不同于常见的标准文本,它以教科书的编写方式,依据美国国家标准 **ANSI X 3.9—1978 FORTRAN 77** 全集语言写成。书中全面介绍了 **FORTRAN 77** 语言,重点而详细地介绍了 **FORTRAN 77** 语言对 **FORTRAN IV** 语言的扩充部分及文件的使用,并提供了程序设计中常用的技巧。书中例题丰富,便于自学。应用面广,有较强的实用性,可供初学者及有一定基础的读者各取所需,是一本普及与提高相结合的书。

本书可供大专师生、科技人员、电算工作者阅读,可作教材使用,也可作为自学读本。

本书在出版过程中,得到了中国建筑学会的领导和各兄弟单位的支持,学会许多同志也给予了热情的帮助,这是本书得以在成稿后三个月内完成全部出版工作的重要原因。

由于时间紧促,若有错误与不当之处,敬请读者批评指正。

中国建筑学会科技咨询中心

1985 年 1 月

目 录

第一章 绪论	1
§ 1. 计算机的发展与应用	1
§ 2. 计算机系统	1
§ 3. 程序设计语言	3
§ 4. FORTRAN 语言的发展	4
§ 5. 程序设计	5
第二章 FORTRAN 语言的基础知识	7
§ 1. FORTRAN 77 字符集	7
§ 2. 语句	8
§ 3. 程序结构	9
§ 4. 语句和注释行的次序	9
§ 5. 源程序的书写格式	10
§ 6. 常数	12
§ 7. 变量	14
§ 8. 整型表达式与实型表达式	17
§ 9. 内部函数	21
第三章 最基本的 FORTRAN 语句	23
§ 1. 算术赋值语句	23
§ 2. STOP 语句、END 语句和 PAUSE 语句	25
§ 3. 表控输入/输出语句	26
§ 4. PARAMETER 语句	30
§ 5. PROGRAM 语句	31
第四章 控制转移语句	33
§ 1. GOTO 语句	33
§ 2. 逻辑 IF 语句	34
§ 3. 算术 IF 语句	41
§ 4. 块 IF、ELSE 和 END IF 语句	42
§ 5. ELSE IF 语句	52
§ 6. 计算 GOTO 语句	57
§ 7. 语句标号赋值语句和赋值 GOTO 语句	58
§ 8. 关于使用控制转移语句的一点建议	59
第五章 DO 循环	62
§ 1. DO 语句	62
§ 2. CONTINUE 语句	64

§ 3. DO 循环的使用规则	64
§ 4. 循环的嵌套	66
§ 5. 两点说明	67
第六章 数组	70
§ 1. 数组与下标	70
§ 2. 数组的说明	71
§ 3. 数组元素的存储顺序	73
§ 4. 程序举例	74
§ 5. 隐 DO 表	84
第七章 基本的带格式的输入/输出语句	87
§ 1. 带格式的 READ 语句和 PRINT 语句	87
§ 2. FORMAT 语句	89
§ 3. 整型数据的输入/输出	92
§ 4. 实型数据的输入/输出	94
§ 5. 不可重复的编辑描述符	98
§ 6. 输入/输出表与格式说明的配合	107
第八章 双精度型、复型和逻辑型数据	114
§ 1. 双精度型数据	114
§ 2. 复型数据	117
§ 3. 逻辑型数据	121
第九章 字符型数据	124
§ 1. 字符常数、字符变量与字符数组	124
§ 2. 子串	127
§ 3. 字符表达式	128
§ 4. 字符赋值语句	129
§ 5. 字符型数据的输入和输出	132
§ 6. 字符表达式的比较	134
§ 7. 用于字符操作的内部函数	141
第十章 子程序	147
§ 1. 函数子程序	147
一. 函数子程序的定义	148
二. 函数子程序的调用和哑实结合	150
三. 字符型函数子程序	162
四. 函数的副作用	166
五. 语句函数	167
§ 2. 子例程子程序	169
一. 子例程子程序的定义规则	169
二. 子例程子程序的调用方式	170
三. 程序举例	171

§ 3. 与子程序有关的其它语句	173
一. 入口 (ENTRY) 语句	173
二. 子程序的交错返回	175
三. SAVE 语句	177
第十一章 数据联系语句与数据初值语句	179
§ 1. 等价语句	179
§ 2. 公用语句	182
§ 3. DATA 语句	190
§ 4. 数据块子程序	192
第十二章 文件	194
§ 1. 记录、文件与通道号	194
§ 2. 内部文件及其应用	201
§ 3. 输入/输出语句的通用形式	206
一. 输入/输出语句的通用形式	207
二. 无格式输入/输出语句	209
§ 4. 文件的定位	211
§ 5. 询问 (INQUIRE) 语句	213
§ 6. 程序举例	221
附录	234
附录1 ASCII 表	234
附录2 内部函数	236
附录3 FORTRAN 77 (全集) 与 FORTRAN IV 的主要不同	240
附录4 FORTRAN 77 (全集) 对 FORTRAN IV 的主要扩充	242
附录5 子集语言与全集语言之间的主要差别	243
编后	245

第一章 绪 论

本章的主要目的是为读者提供一点基础知识。主要是对计算机系统的发展, 程序设计的概念和 FORTRAN 语言发展的情况以及 FORTRAN 程序的结构进行非常粗浅的介绍。因为, 在有关计算机的绝大多数著作中都有类似的内容; 同时, 大多数读者也都具有这方面的知识。

§ 1 计算机的发展与应用

自从 1946 年第一台电子计算机问世以来, 迄今为止只有 38 年的历史, 但它却经历了四代的变化, 即电子管时代, 晶体管时代, 中、小规模集成电路时代和大规模集成电路时代。它每隔 5~7 年计算速度提高 10 倍, 存储容量扩大 10 倍, 体积缩小 10 倍, 价格降低 10 倍。它以其它工业部门所无可比拟的惊人速度向前发展, 给人类社会带来了巨大的影响和冲击。目前, 正向第五代迈进。

计算机在国民经济各个领域中的应用可以归纳为四个主要方面。第一, 科学研究和工程设计的数值计算。如数理计算、统计分析、结构计算、需求预测、模拟计算等。由于计算机的精度高、速度快, 使过去无法计算的数学方程成为可能, 使过去需要花几个星期才能计算出结果的问题, 几分钟内即可得出结果。第二, 信息处理。由于计算机能存储大量的数据而又能快速处理, 因此广泛应用于信息处理, 企业管理和情报检索等方面。如生产管理、仓库管理、人事管理、财务管理、图书管理等。第三, 计算机辅助设计。计算机辅助设计是利用计算机的计算, 逻辑判断等功能帮助人们进行各种工程设计的一项专门技术。在光学设计、建筑、航空、船舶、汽车、化工、大规模集成电路、计算机设计自动化等方面有着广泛的应用, 使设计过程趋向半自动化或自动化。大大缩短了设计周期, 降低了成本、节省了人力、物力。第四, 实时控制。它代替人们对某些生产过程进行监测和控制, 有利于提高产品质量、减轻劳动强度、提高劳动生产率, 从而使工农业生产发生根本的变革。总之, 计算机所能处理的对象不胜枚举, 不论你工作在什么行业, 计算机都能成为你有力的工具。

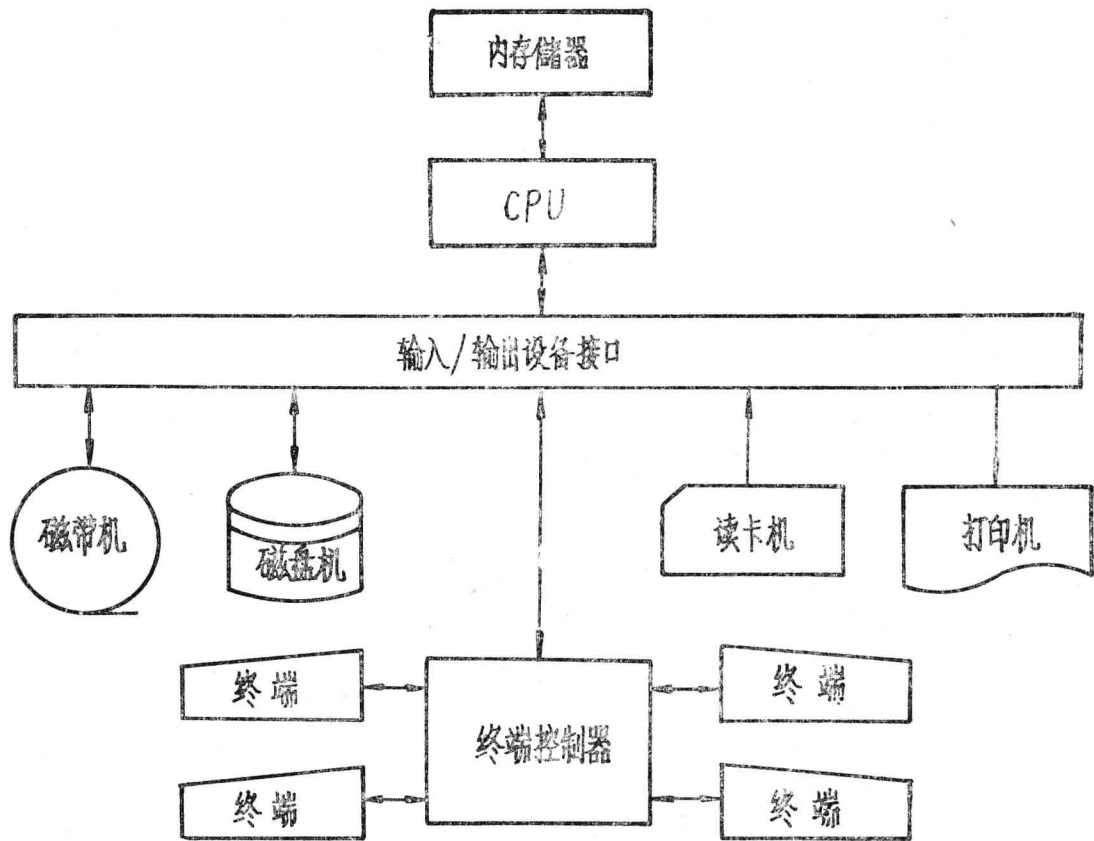
§ 2 计 算 机 系 统

一个完整的计算机系统由硬件和软件两部分组成。组成计算机的物理设备叫做硬件。如数据准备装置、输入设备、中央处理机、内存存储器、外存储器 and 输出设备。为使用计算机、发挥计算机的效率和维护计算机正常运行的程序系统称为软件。软件是组成计算机系统的逻辑设备。如操作系统、翻译程序以及其它实用程序。

如果把硬件比作一个人的躯体的话, 那么, 软件就是一个人的灵魂。硬件是计算机系统潜在的能力, 但必须要由软件进行驾驭和发挥。如果没有软件, 计算机硬件只不过是一

堆复杂的电子电路而已。过去，一谈起计算机，人们往往津津乐道于一个 CPU 可以集成在单个芯片上。而现在，人们已开始把注意力从七十年代的硬件集成转移到八十年代的系统集成上来了。

一个典型的硬件系统示意如下：



一. 存储器

存储器用来保存程序和数据，这就是计算机的“记忆”能力之所在。

内存器由许多存储单元组成，每个单元存放若干位二进制信息。能存放一个二进制数字(0 或1)称为一位(bit)，连续的8位称为一个字节(byte)。内存容量一般以 KB 为单位来表示。1 KB=1024 B，即 1024 个字节。每个存储单元有一个唯一的地址来标识，计算机按指定的地址存取信息。当存储器中信息被取出时，原来的信息并不消失；当存入新的信息时，原有的信息即被破坏。内存存在一定程度上决定着计算机的速度和精度。

内存的容量是有限的，它不可能记录下全部信息。这就要借助外存储器（如磁盘、磁带）来存储暂时不用的大量信息，外存的容量可以是内存的几千倍甚至几万倍。

二. 中央处理器(CPU)

通常把计算机的控制器和运算器称为中央处理器。

运算器是对二进制数进行算术运算和逻辑运算的装置。算术运算包括二进制加、减、

乘和除等运算；逻辑运算指两个数大小的比较等操作。

控制器是计算机的“指挥中心”，它控制输入设备把程序和数据输入内存，控制运算器、存储器有秩序地进行计算，并控制输出设备输出中间结果和最后结果。

三. 输入/输出设备

常用的输入设备有纸带输入机、卡片输入机和键盘。前两种输入设备要通过中间媒介纸带和卡片，因此，输入之前预先要用专门的装置把程序或数据记录在纸带或卡片上。键盘是在人们按键时，计算机直接把字符接收到机器内。

常用的输出设备是打印机和终端显示器。

磁盘和磁带既可作为输入设备，又可作为输出设备。

§ 3 程序设计语言

程序设计是伴随着计算机的出现而产生的一门学科。它也象计算机硬件的发展一样，经历着由低级到高级的发展过程。程序设计就是把完成特定任务的算法用计算机所能接收的语言表述出来。程序设计语言是人与计算机进行通讯的媒介。

一. 机器语言和汇编语言

在计算机发明初期，程序设计人员必须用二进制数字(0 或 1)编写计算机能够接收的一系列指令，即所谓“机器语言”。同时，还必须指定每条指令存储单元的地址。可想而知，用机器语言编写的程序形同“天书”，难编、难读、难改、难于移植。同时，程序设计人员必须彻底了解计算机的硬件组织。在这种情况下，计算机的应用和发展受到了极大的阻碍。

经过人们积极研究，对机器语言的形式进行改进，将机器指令中的操作码和地址用“助记符”来表示，比如，加、减分别用 ADD 和 SUB 来表示，即所谓“汇编语言”。这是一个很大的进步。使程序设计人员的工作大大改观，人容易看懂了，但计算机却完全看不懂了，汇编语言成了计算机完全不懂的“外国语”。因此，必须有一个程序担任翻译，把汇编语言翻译成机器语言，这就是“汇编程序”。汇编语言学习和使用仍很困难，用汇编语言编写的程序只适用于同类机器，它仍然是一种低级语言(有人亦称为中级语言)。

二. 高级语言

从机器语言向汇编语言的进步，刺激了人们创造更高级语言的积极性，不同种类的语言纷纷问世。经过长期的竞争和考验，有的逐渐被淘汰，有的生存下来，而真正在全世界广为采用的大约只有七、八种。

高级语言有的接近于人们的口语或文章，有的接近于数学公式。因此，能为人们所普遍接受。同时，高级语言的语句可以看成机器指令或汇编语言指令的浓缩体，一条高级语言的语句往往等于几条，甚至十几条汇编语言的指令。因而，程序设计的效率较高。更重要的是，高级语言几乎能用于各种不同的计算机上，这就扩大了程序的适用范围，便于推广和交流。

当然，计算机是无法执行用高级语言编写的程序(称为源程序)的，它必须预先翻译成机器语言程序(称为目标程序)。担任翻译工作的程序一般称为编译程序，翻译过程称为编译。编译程序是由计算机厂商提供的，它是系统软件的一部分。

高级语言的翻译过程一般分为五步:

- (一) 把编译程序从磁盘调入内存。
- (二) 将磁盘(或其它设备)上的源程序读入内存。
- (三) 编译程序将源程序翻译成目标程序送入磁盘(或留在内存)。
- (四) 将目标程序调入内存并执行之。(需要时从键盘或磁盘等外设将数据读入内存)。
- (五) 把结果从内存输出。

显然,最理想的情况是人们能直接使用自然语言(例如英语、西班牙语、俄语等)与计算机通讯,但遗憾的是,自然语言大量的词汇,复杂的语法都给程序的翻译造成了极大的困难。二十世纪七十年代,科学家们对语义信息处理和模式识别的研究,为在使用自然语句作为人一机通讯的媒介奠定了初步的基础。迄今为止,所有的程序设计语言都是人工语言。每种人工语言都有一个不大的词汇表,并严格定义了文法以及语法和语义规则,以便于编译程序进行翻译。程序设计者必须严格遵守这些规则,才能和计算机进行对话。只有程序设计得正确,它才能为你得出正确的结论。计算机毕竟是机器,它只能机械地执行你的程序(而不一定是你主观上想)要它干的事。从这点看来,计算机是忠实的。

§ 4 FORTRAN 语言的发展

计算机是应数值计算的要求而诞生的,而最初的应用也是进行数值计算。1954年,为在计算机上实现公式转换(FORMula TRANslation)而研制成功了世界上第一个高级语言——FORTRAN语言。它的出现大大推动了计算机的普及和应用,难怪有人称它为50年代的“功臣”。

此后, FORTRAN语言被广泛使用,不同计算机上的“方言”种类繁多。为了给编译系统研制者提供标准、提高源程序的通用性并便于人们进行算法交流,要求FORTRAN标准化的呼声很高。为此,美国国家标准协会ANSI于1966年正式公布了两个美国标准文本,即

标准基本FORTRAN (ANSI X 3.10—1966)

标准FORTRAN (ANSI X 3.9—1966)

前者相当于1958年6月提出的FORTRAN II,而后者相当于FORTRAN IV。

在美国标准的基础上,国际标准化组织于1972年公布了国际标准程序设计语言FORTRAN, ISO R 1539文本,把FORTRAN描述为三种级别,即基本级、中间级和完全级。ISO R 1539基本级与ANSI X 3.10—1966相近,ISO R 1539完全级与ANSI X 3.9—1966相近,而ISO R 1539中间级介乎两者之间。

1976年,美国国家标准协会对ANSI X 3.9—1966进行了修订和补充,于1978年4月3日正式公布了美国国家标准程序设计语言FORTRAN, ANSI X 3.9—1978(即FORTRAN 77)。同时宣布取消标准基本FORTRAN (ANSI X 3.10—1966)。

1980年,国际标准化组织采用美国标准ANSI X 3.9—1978作为新的国际标准ISO 1539—1980。

中国国家标准局根据国际标准于1982年发布了FORTRAN国家标准GB 3057—82。

FORTRAN包括全集和子集语言。子集的提供是为了提高程序的可移植性,以便在

各种各样的数据处理系统上使用。子集 FORTRAN 是全集语言的一个子集。

FORTRAN 77 是对美国国家标准 FORTRAN ANSI X 3.9 — 1966 的修订。其增加的功能见附录 3。

FORIRAN 是世界上最流行的数值计算语言，成为世界通用的计算机语言。国外有人估计，进行数值计算的工程和科研人员中，90% 的人使用 FORTRAN 语言。

§5 程 序 设 计

一. 程序设计的步骤

程序设计大致包括以下几步。

(一) 明确确定所要解决的问题和所要达到的目的。

(二) 对所要解决的问题进行分析。对于数值计算问题。要选择合适的数值计算方法，并分析可能带来的问题。

(三) 编制程序流程图。拟出解决问题的步骤，确定程序的模块和调用关系以及模块之间传输的数据和数据传输的方法。

(四) 程序的编写。把各模块的功能要求用程序设计语言表述出来。程序常常写在专门的程序纸上。程序编好后，根据其使用计算机的输入方式，要么穿制成纸带或卡片，要么直接从键盘输入。

(五) 程序的调试和测试。编写的程序是否合乎所用程序设计语言的语法要求，这可以借助于翻译程序和其它程序来检查。更重要的是要检查程序的功能是否达到了预定的要求。

(六) 程序资料。要重视程序资料的编写和收集，这是程序使用和维护时所不可缺少的。例如，程序的功能说明、框图、使用说明、数学模型的说明以及程序所适用的范围等。

二. 程序流程图

流程图又称框图。

流程图分系统流程图和程序流程图两种。系统流程图是用来形象描述数据在整个数据处理系统中如何流动的一种图表。而程序流程图则用来说明计算机解题的逻辑和步骤。

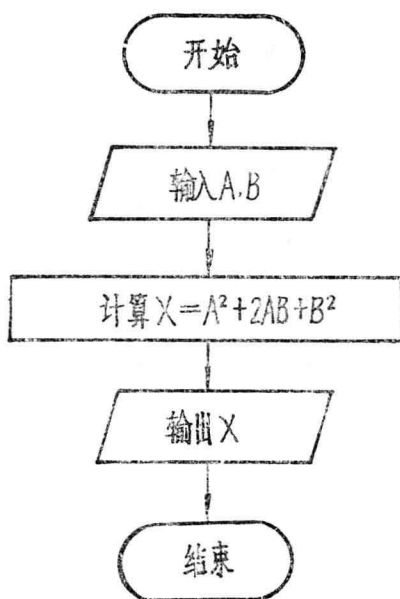
〔例〕输入 A, B 的值，根据

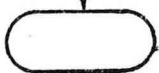
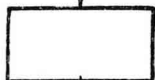
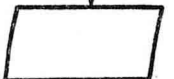
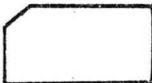
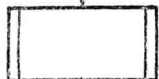
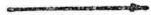
$$X = A^2 + 2AB + B^2$$

计算 X 的值，并将 X 的值打印出来。

程序流程图如右图：

现将美国国家标准局公布的标准 ANSI X 3.5 中的常用符号列于下表：



符 号	说 明	符 号	说 明
	程序开始		页外连接
	程序结束		键盘输入
	处理过程		显 示
	输入／输出		穿孔卡片
	条件分支		打印文件
	循环开始		穿孔纸带
	过程调用		磁 带
	程序流线		磁盘、磁鼓
	页内连接		

第二章 FORTRAN 语言的基础知识

§ 1 FORTRAN 77字符集

FORTRAN 77 字符集规定了 FORTRAN 程序中允许使用的字符，由 26 个大写字母，10 个数字和 13 个专用字符组成。

一. 字母

字母是下列 26 个字符之一：

ABCDEFGHIJKLMNOPQRSTUVWXYZ

二. 数字

数字是下列 10 个字符之一：

0123456789

三. 专用字符

专用字符是下列 13 个字符之一：

字 符	名 称	字 符	名 称
␣	空 格	)	右 括 号
=	等 号	,	逗 号
+	加 号	.	小 数 点
-	减 号	\$	美 元 符 号
*	星 号	'	撇 号
/	斜 线	:	冒 号
(	左 括 号		

四. 几点说明

(一). 书写程序时，一定要把字符写清楚。字母 O 和数字 0，字母 Z 和数字 2，字母 I 和数字 1，通常把数字 0 写作 ϕ 或 θ ，Z 写作 $\mathbb{Z}$ ，字母 I 注意上下加短横线。

(二). 在本书中，用 ␣ 来代表空格字符。在下列情况下，空格是有意义的：

1. 在字符常数中，撇号编辑描述符和 H 型编辑描述符中。
2. 在程序的第六列，用空格来标记初始行。
3. 计算每一行或每一语句中字符总数时。

在其它地方，可加以空格来改善程序的可读性。

(三). 不同的计算机系统可能对字符集有不同的限制和扩充。例如，DUAL/68000 系统还允许使用 26 个小写字母、反斜线字符“\”和和号“&”。

(四). 在字符常数、撇号编辑描述符和 H 型编辑描述符以及注释行中, 可以出现计算机能打印出的任何字符。

§ 2 语 句

FORTRAN 的语句可以分为可执行语句和非执行语句两类。

可执行语句说明要执行的实际操作。包括赋值语句、控制语句和输入/输出语句。

非执行语句向编译程序提供如下信息: 数据的特点、排列和初值; 程序单元的类型; 子程序的入口点等。

FORTRAN 提供的可执行语句和非执行语句如下表所示。

可 执 行 语 句		非 执 行 语 句	
赋 值 语 句	算术赋值语句 逻辑赋值语句 字符赋值语句 标号赋值语句	说 明 语 句	INTEGER 语句 REAL 语句 DOUBLE PRECISION 语句 COMPLEX 语句 LOGICAL 语句 CHARACTER 语句 IMPLICIT 语句 DIMENSION 语句 PARAMETER 语句 EXTERNAL 语句 INTRINSIC 语句 SAVE 语句 EQUIVALENCE 语句 COMMON 语句
控 制 语 句	无条件GO TO 语句 计算GO TO 语句 赋值GO TO 语句 逻辑 IF 语句 算术 IF 语句 块 IF 语句 ELSE IF语句 ELSE 语句 END IF语句 CONTINUE 语句 DO 语句 CALL 语句 RETURN 语句 PAUSE 语句 STOP 语句 END 语句		PROGRAM 语句 FUNCTION 语句 SUBPROGRAM 语句 BLOCK DATA 语句
输 入 输 出 语 句	READ 语句 PRINT 语句 WRITE 语句 OPEN语句 CLOSE 语句 INQUIRE 语句 REWIND 语句 BACKSPACE 语句 ENDFILE 语句		DATA 语句
			FORMAT 语句 语句函数语句

§ 3 程 序 结 构

FORTRAN 是块状结构。一个可执行程序是由一个主程序或由一个主程序和若干个
子程序组成。

程序单元(又称程序段)由语句序列和任选的注释行序列组成。一个程序单元或者是一个
主程序, 或者是一个子程序。

主程序的第一个语句不能是 FUNCTION 语句、SUBROUTINE 语句或 BLOCK DATA
语句。它可以用 PROGRAM 语句作为其第一个语句。一个可执行程序只能有一个主程序。

子程序有三种,第一个语句为 FUNCTION 语句的称为函数子程序,第一个语句是 SUB-
ROUTINE 语句的称为子例程子程序,第一个语句为 BLOCK DATA 的被称为数据块子程序。

§ 4 语句和注释行的次序

在任何程序单元中, 语句出现的次序必须遵循如下规则:

一. PROGRAM 语句必须是主程序的第一个语句。而 函数子程序、子例程子程序和
数据块子程序的第一个语 句必须 分别是 FUNCTION 语句、SUBROUTINE 语句和 BLO
CK DATA 语句。

二. 程序单元的最后 一个语句必须是 END 语句。

三. 注释行可以出现在程序的任何地方, 甚至可在第一条语句之前。

四. FORMAT 语句可以出现在程序的任何地方。

五. 所有说明语句必须出现在 DATA 语句、语句函数语句和可执行语句的前面。

六. 所有语句函数语句必须放在可执行语句之前。

七. DATA 语句可出现在说明语句之后的任何地方。

八. ENTRY 语句可出现在程序单元的任何地方(除了 DO 循环和 THEN 块、ELSE 块
或 ELSE IF 块外)。

九. IMPLICIT 语句必须放在所有说明语句之前(除了 PARAMETER 之外)。

十. 符号名的类型说明语句必须放在定义符号名的 PARAMETER 语句之前。

以上规则如下表所示:

注 释 行	PROGRAM, FUNCTION, SUBROUTINE 或BLOCK DATA语句		
	FORMAT 语句 或 ENTRY 语句	PARAMETER 语句	IMPLICIT 语句
		DATA 语句	其它说明语句
			语句函数语句
			可执行语句
END语句			

在表格中, 由垂直线隔开的各类语句可以混合使用, 用水平线隔开的各类语句不能混
合使用。