

吴世琪 高 超 编著

FORTRAN 77

程序设计

科学出版社

FORTRAN 77 程序设计

吴世琪 高 超 编著

科 学 出 版 社

1 9 8 7

内 容 简 介

本书全面而系统地介绍了标准 FORTRAN 77 (全集)程序设计语言,并按照结构程序设计的要求讨论了该语言各种成分的意义、功能和应用。全书共十三章:第一、二两章介绍 FORTRAN 77 程序的基本概念,第三至九章描述该语言的各种语句结构及用法,第十、十一章讨论 FORTRAN 77 中过程的定义和引用,第十二章介绍文件,最后一章讨论了程序的移植和覆盖问题。

书中每章后均附有习题,以便读者加深对每章内容的理解。附录中还给出了 FORTRAN 77 与 FORTRAN 66 的比较和 FORTRAN 77 的全部语法图,以及在 HONEYWELL DPS 8 计算机上提交 FORTRAN 77 作业的方法。

本书可供广大研究、应用计算机的科技人员和大专院校师生,以及自学者阅读和参考。

FORTRAN 77 程序设计

吴世琪 高 超 编著

责任编辑 那莉莉

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

*

1987 年 11 月第 一 版 开本: 787×1092 1/32

1987 年 11 月第一次印刷 印张: 15 1/4

印数: 4,500 字数: 349,000

统一书号: 15031·876

本社书号: 4844·15-8

定价: 3.60 元

前 言

FORTRAN 语言是世界上最早提出并得到广泛应用的一种程序设计语言。在对其应用的二十多年的时间里,经过人们不断的扩充和改进,美国国家标准协会于 1978 年正式提出了新的标准 FORTRAN 语言文本——FORTRAN 77。这一新文本目前在国外已得到广泛应用,在国内各种大、中、小型计算机以及微处理机上,也多配有该语言系统。为了推广该语言,我们根据 FORTRAN 77 (全集)程序设计语言标准文本编写了此书。全书共十三章。第一和第二章介绍了 FORTRAN 77 语言的基本概念、语法成分、数据类型和程序格式及程序结构。希望读者通过这两章的学习,对 FORTRAN 77 程序设计的基本概念和整体结构有初步的了解。第三章至第十二章分别讨论 FORTRAN 77 语言的各种成分、语法和功能,并结合一些简明的实例说明它们的用法。其中第九和第十二两章分别介绍输入输出的基本部分和比较深入的部分。第十三章介绍结构程序设计的基本概念和要求,以及如何应用 FORTRAN 77 语言编写出一个结构清晰、易于阅读和调试的程序。在附录中给出了 FORTRAN 77 与 FORTRAN 66 (相当于 FORTRAN IV)的比较,以使那些熟悉 FORTRAN 66 的读者可以便捷地了解 FORTRAN 77 中新增加的成分以及它同 FORTRAN 66 的差别;附录中还介绍了在 Honeywell DPS 8 计算机上提交作业的方法,以有助于那些使用 DPS 8 计算机的读者在该机上提交和运行 FORTRAN 77 程序。

为了方便广大科技人员和大专院校师生对本书的使用,

书中系统而全面地介绍了 FORTRAN 77 语言(全集)的所有成分和语句,还讨论了程序的移植和覆盖问题,并在书后附有内容段索引,以便读者查找所要了解的内容。同时,为兼顾初学者自学,在每章开头指出了本章的内容要点和基本要求,并在每章末附有适量的习题。

本书第三、第六和第十一章(外部函数的定义及其引用一节除外)以及各章的习题、附录一、附录五和本书索引是高超同志编写的,其余部分是吴世琪同志编写的。本书完稿后,中国科学院软件所的张兆庆同志对全稿进行了认真、仔细的审阅和修改,并提出了许多宝贵意见;此外,北京大学的吴允曾教授和许卓群同志对本书的编写给予了热情的支持,许卓群同志还审阅了初稿;在此对他们表示衷心的感谢。

由于我们水平有限,书中一定存在缺点或错误,敬请读者批评、指正。

编者

1985 年 3 月

目 录

第一章 引论	1
1.1 计算机与程序设计语言	1
1.2 程序设计与 FORTRAN 语言	3
1.3 FORTRAN 77 基本概念	4
1.4 FORTRAN 77 程序的书写格式	10
1.5 程序流程图	15
习题	16
第二章 语句分类、数据类型及程序结构	17
2.1 语句分类和各类语句次序	17
2.2 数据类型和常数	21
2.3 计算机表示数据的精度和大小范围	26
2.4 数据的输入输出概念	30
2.5 程序结构和简单的程序实例	32
2.6 作业控制说明	39
习题	40
第三章 表达式和赋值语句	42
3.1 算术表达式和算术赋值语句	42
3.2 字符表达式和字符赋值语句	54
3.3 关系表达式	57
3.4 逻辑表达式和逻辑赋值语句	62
3.5 语句标号赋值语句	68
习题	69
第四章 控制语句(一)	73
4.1 块 IF, ELSE IF, ELSE 和 END IF 语句	73
4.2 逻辑 IF 语句、算术 IF 语句以及 STOP, PAUSE, CONTINUE 和 END 语句	89
4.3 三种 GO TO 语句	95

习题	100
第五章 控制语句(二)——DO 语句	102
5.1 DO 语句和基本循环程序	102
5.2 嵌套的循环结构	109
5.3 DO 循环和选择结构程序的结合	114
习题	122
第六章 说明语句	124
6.1 类型说明	124
6.2 PARAMETER 语句	135
6.3 EXTERNAL 语句	139
6.4 INTRINSIC 语句	143
6.5 EQUIVALENCE 语句	145
6.6 COMMON 语句	150
6.7 SAVE 语句	159
习题	162
第七章 数组	165
7.1 数组的特性和 DIMENSION 语句	165
7.2 数组元素	171
7.3 数组的输入输出——隐 DO 表	179
7.4 数组名在程序中的使用	183
习题	185
第八章 字符数据处理	188
8.1 字符变量和字符数组	188
8.2 量子串和数组元素子串	190
8.3 字符数据的表控输入输出	192
8.4 字符关系表达式和有关字符处理的内部函数	195
8.5 字符处理程序举例	199
习题	202
第九章 输入输出语句(一)	204
9.1 输入输出的基本概念	205
9.2 表控格式输入输出	216

9.3	格式输入输出语句和 FORMAT 语句	222
9.4	可重复格式码	232
9.5	格式说明及格式码重复	247
9.6	KP , / 和 : 格式码以及字符常数用作格式标识符	249
9.7	输入输出表, 记录和格式的相互关系	253
	习题	258
第十章	子程序及虚数组应用	260
10.1	FORTRAN 77 语言中的过程	260
10.2	子程序的定义及引用	262
10.3	实元与虚元结合	271
10.4	辅程序的多入口和 ENTRY 语句	294
10.5	可调数组和假定大小数组的应用	297
	习题	303
第十一章	函数及数据块辅程序	306
11.1	内部函数及其引用	306
11.2	语句函数的定义及其引用	315
11.3	外部函数的定义及引用	320
11.4	程序单位间的数据联系	337
11.5	DATA 语句	341
11.6	数据块辅程序	349
	习题	350
第十二章	输入输出语句(二)	352
12.1	变量格式标识符及其它格式码	352
12.2	文件的读写	357
12.3	辅助输入输出语句	364
12.4	磁带、磁盘和内部文件的读写	375
	习题	387
第十三章	FORTRAN 77 应用方面的一些问题	390
13.1	结构程序设计的概念和基本要求	390
13.2	关于 FORTRAN 77 的简评及其应用考虑	393
13.3	关于程序移植和程序覆盖问题	399
13.4	提交作业的一般方法	408

13.5 FORTRAN 77 程序应用举例	409
习题	419
附录一 FORTRAN 77 内部函数表	422
附录二 在 HONEYWELL DPS 8 机上提交和执行作 业的方法	430
附录三 FORTRAN 77 与 FORTRAN 66 的差别	442
附录四 ASCII 字符集	449
附录五 FORTRAN 77 (全集)语法图	454
参考资料	475
索引	476

第一章 引 论

本书按照程序设计语言 FORTRAN (全集) 美国国家标准文本 ANSI, X3.9—1978 讨论 FORTRAN 77 程序设计。本章和第二章将介绍高级程序设计语言的概念, FORTRAN 77 语言的基本术语、概念及其程序结构, 并给出简单的程序实例。我们希望读者读完这两章后, 对 FORTRAN 77 语言的总体框架和基本概念有所了解, 并能提出某些问题, 以便带着这些问题去阅读以后的章节。

1.1 计算机与程序设计语言

任何一台电子计算机都有一套它自己的指令系统。人们要使用计算机解决某一问题, 可以先用计算机指令系统的各种指令(这种指令以代码形式提供, 称为指令码)编写出一段指令序列(这种指令序列就是所谓的程序), 再用某种方法(例如通过穿孔卡片)把它送入计算机, 然后让计算机执行这段指令序列, 这样就可以得到所要求的运算或操作结果。计算机可以不经翻译而直接执行的这种用机器指令编写的程序叫做机器语言程序。计算机的指令系统就是机器语言。

使用机器语言编写和调试解题程序, 对一般用户来说是很不方便的, 因为这要花很多的时间和精力, 首先要求用户了解并熟记各条机器指令的功能和用法, 至于修改程序就更是麻烦的事了。这就严重地妨碍了计算机应用的推广。为了解

决这个问题,人们提出并使用了高级程序设计语言。高级程序设计语言不依赖于具体的计算机及其指令系统,也就是说,它与具体机器无关。FORTRAN 语言就是一种高级程序设计语言,它比较接近普通的数学语言,FORTRAN 语言使用一些英语单词、数学符号及其它常用符号书写程序,很便于人们学习记忆和掌握运用。

用高级程序设计语言编写的程序叫做源程序。现有的计算机不能直接执行源程序,必须经过另外一个专门的程序把源程序翻译成计算机的指令序列(即机器语言程序)后才能由计算机执行。这种专门的程序称为编译程序,它和具体的机器有关,并和具体的高级语言相对应。例如对于 FORTRAN 77(全集)语言来说,就要根据 FORTRAN 77 语言的语法和语义规定考虑所用型号的计算机的指令系统,设计并编写一个 FORTRAN 77 编译程序系统,并且预先把它存入计算机中,使之成为某个计算机系统的一个组成部分。这样,该计算机系统就能够接受和执行 FORTRAN 77 源程序。

执行由 FORTRAN 77 语言编写的源程序可大致分两个阶段:第一个阶段由 FORTRAN 77 编译程序对源程序进行语法检查,如果符合各项语法规则,则对源程序进行编译,即把源程序翻译成机器语言程序,以得到所谓的目标程序;第二个阶段,计算机执行目标程序,其中包括输入原始数据,对数据进行加工处理和必要的运算,输出所要求的结果。第一阶段称为编译阶段,第二阶段称为运行阶段。在这两个阶段中,如果出现错误,计算机会输出相应的错误信息供使用者分析检查。学习使用高级语言进行程序设计不要求程序设计者或用户了解计算机的结构、计算机指令系统以及高级语言编译程序的内部结构。

1.2 程序设计与 FORTRAN 语言

使用计算机解决某个问题，首先要对该问题进行分析和描述，要把一个大问题分解为多个较小的问题，把一个大任务分解为多个较小的任务；然后对这些较小的问题和任务确定算法，也就是确定具体的解决步骤；最后用一种高级语言（例如 FORTRAN 77 语言）把这些具体步骤表达出来。这样就完成了解决一个问题的一项程序设计工作。换句话说，使用计算机执行解题任务时，计算机是按照人们对该项任务事先确定的描述步骤执行的。人们常说的“算法”一词，就是指对一项任务的精确描述。对于计算机来说，一个算法体现的是一组指令及其有关的数据。指令是计算机可以“理解”和执行的基本单位。使用计算机执行的各项任务必须事先用算法加以精确的描述，归根到底，是把一项任务分解成多个小任务，而每个小任务再分解成许多可用指令表达的最简单、最基本的步骤，然后提交计算机执行。

高级语言的每条语句一般都是由若干条计算机指令组成的。现有的计算机要通过编译程序的“翻译”才能够“理解”和执行高级语言的程序（未来的计算机可能不再需要编译程序的翻译，现在暂且不论）。因此，使用 FORTRAN 语言进行程序设计，最终表现为用 FORTRAN 源程序把描述任务的算法表达出来。关于算法主要与个人的专业知识、经验多少以及解决问题的技能有关。此外，人们已经积累有许多适用于各种领域、各个行业的 FORTRAN 语言算法供用户参考和选用。程序设计方法已经发展成一套程序设计方法学，它保证了程序的正确性和程序结构的清晰性，以便于人们阅读和理解、维护和修改。本书将着重从结构程序设计的角度来讨论

使用 FORTRAN 77 语言进行程序设计的问题, 不专门讨论算法问题和程序设计的方法问题。

FORTRAN 语言最初是由美国 IBM 公司于 1956 年实现的, 它主要被用于科学和工程计算, 以后得到了飞速发展并被广泛使用。1966 年美国制定了 FORTRAN 两级国家标准: 标准 FORTRAN 和标准基本 FORTRAN。后者是前者的子集。标准 FORTRAN 也称为 FORTRAN 66, 它和通常的 FORTRAN IV 相当。此后, 许多厂商和用户对 FORTRAN 66 进行了扩充。为适应数据处理、商业和企事业管理的需要以及结构程序设计的要求, 1977 年提出了新的语言标准 FORTRAN 77, 并于 1978 年由美国国家标准协会 (ANSI) 批准为两级美国国家标准 (ANSI, X3.9—1978): FORTRAN 77 全集和 FORTRAN 77 子集, 其中子集与 FORTRAN 66 基本相当。国际标准化组织 (ISO) 已经确定它们为国际标准, 我国也已采纳它们为我国国家标准。

1.3 FORTRAN 77 基本概念

1.3.1 FORTRAN 77 字符集

FORTRAN 77 字符集是 FORTRAN 77 语言允许使用的所有字符的集合。它包括从 A 到 Z 的 26 个英语字母, 从 0 到 9 的 10 个数字, 以及 13 个专用符号共 49 个字符, 现列表如下:

表 1.1 中任何一个字母或任何一个数字称为字母数字字符。FORTRAN 77 的所有语法项、语句和程序都只能由这 49 个基本字符组成 (字符常数例外)。字符集中的所有字符 (包括空白符) 在某种编码制 (例如 ASCII 编码制) 中一一对应不同的编码。

表 1.1 FORTRAN 77 字符集

字母数字字符	字 母	A 到 Z
	数 字	0 到 9
专 用 字 符	=	等号
	+	加号
	-	减号
	*	星号(乘号)
	/	斜杠(除号)
	(	左括号
	)	右括号
	,	逗号
	.	小数点
	\$	美元符号
符	'	撇号
	:	冒号
		空白

1.3.2 程序单位、语句和语法项

能够执行一项小任务的一段独立的程序称为一个程序单位。一个程序单位是由一系列语句组成的。执行一项任务往往需要若干个(至少一个)程序单位，且其中有一个是主程序单位。

程序单位是由语句序列组成的，而语句是由一系列语法项组成的。这些语法项由 FORTRAN 77 字符集中的各种字符组成。FORTRAN 77 的基本语法项是常数、符号名、语句标号、关键字、运算符及专用字符。

下面举一个最简单的 FORTRAN 77 程序单位的例子，以说明上述概念。假定要计算三个数的平均值，可以写出如下的程序：

```
PROGRAM ABC
```

```

REAL A, B, C, AV
READ *, A, B, C
5  AV=(A+B+C)/3.0
10 PRINT*, AV
END

```

这是一个简单而完整的程序单位,它由6个语句组成,以PROGRAM ABC语句开始,以END语句结束。PROGRAM ABC指明该程序单位为主程序单位,程序单位的名字为ABC。REAL A, B, C, AV是一种说明语句,它说明A, B, C和AV是四个实型变量。READ *, A, B, C是读语句,即输入语句,它把外部介质上的数据输入到计算机内存贮器中,并赋给变量A, B, C。例中 $AV=(A+B+C)/3.0$ 是一个赋值语句,它完成求平均值的计算并把计算结果赋给实型变量AV。PRINT*, AV是一种输出语句,它把计算结果传输到外部介质上并打印出来。最后的END是程序单位结束语句。例中的输入语句(READ语句)、赋值语句和输出语句(PRINT语句)以及END语句都是可执行语句,也就是说,计算机要执行这些语句以完成指定的操作或运算。PROGRAM语句和REAL语句则是非执行语句,它们为程序单位的编译提供必要的信息和说明。

例中的每个语句都是由若干个(至少一个)语法项构成的。下面分别介绍这些语法项的概念:

常数 常数的值是不变的。有算术常数、逻辑常数和字符常数等。例中的3.0就是一个实型算术常数。

符号名 符号名是标识程序中各种实体的名字。在一个程序单位中,所有的变量名、数组名、常数名(常数符号名)、语句函数名、内部函数名和虚过程名等,都是局限于该程序单位内的符号名。这些符号名的作用范围只局限于该程序单位,即它们所标识的实体名字的意义只在该程序单位内起作用,在

另外的程序单位中, 这些符号名就失去它原来的意义并可以被用来标识另外的实体名字. 这种符号名标识的是局部实体, 所以也称局部量. 例中的 A, B, C 和 AV 都是符号名, 它们标识的该程序单位中的四个实型变量都是局部量. 符号名标识全局实体时则称为全局量. 全局量的作用范围是一个可执行程序的所有程序单位. 例中的程序单位名 ABC 就是一个全局量的符号名. FORTRAN 77 中各种符号名的作用范围都有明确的规定, 我们将在适当的地方说明. 此外, FORTRAN 77 规定, 符号名由 1 到 6 个字符组成, 且第一个字符必须是字母.

关键字 FORTRAN 77 中的关键字是标识各种语句的标识符, 采用 1 到 2 个英语单词表示. 除去赋值语句和语句函数语句外, FORTRAN 77 的每个语句都是用一个关键字开头的. 如输入语句中的 READ, 输出语句中的 PRINT, 类型语句中的 REAL, PROGRAM 和结束语句 END 等都是关键字, 它们作为语句的开头而标识各个不同的语句. 虽然 FORTRAN 77 的关键字和内部函数名都是有确定意义的专用标识符, 但是 FORTRAN 77 并没有规定它们不能作其它符号名备用. 尽管如此, 我们还是提倡用户在编写程序时不要使用关键字或内部函数名去做其它变量、数组或函数的符号名, 否则容易引起混淆, 影响程序的可读性.

标识符一词在 FORTRAN 77 中是一般的术语. 关键字或符号名都是一种标识符. 一个具体的标识符可以标识某种特定的语句(关键字)或者标识某个特定的实体(符号名), 也可以标识某个特定的格式或部件. 因此, 标识符的形式可以是英语单词、字母数字串或者正整数、星号等.

语句标号 语句标号以 1 位到 5 位数字组成的十进制正整数表示的. 该正整数不可为零. 它放在语句的左边, 用以标识语句序列中的某个语句. 不同的关键字标识不同类型的

语句,不同的语句标号则标识语句序列中不同位置(即不同序号)的语句。语句标号提供了转移或引用某个语句的方便。语句序列中任何语句的左边都可以加上语句标号,而格式语句必须带有语句标号。语句标号的大小与语句的前后顺序并不一定统一。但是为方便阅读和易于检查错误,最好还是按语句书写的顺序,使语句的标号由小到大、有规律地递增排列。如上例中赋值语句带语句标号 5,其后的 PRINT 语句带语句标号 10,在需要修改程序时,还可以在语句标号 5 和 10 之间增加其它的语句和语句标号。语句标号的作用范围只局限于本程序单位,就是说,某个程序单位内的语句不能通过语句标号去引用或转向另外一个程序单位内的语句。同一形式的语句标号在不同的程序单位内标识完全不同的语句。

运算符和专用字符 组成 FORTRAN 77 语句的语法项目还包括各种运算符和专用字符。如上例中的加号和除号运算符以及星号、等号、左括号和右括号等专用字符。这里的专用字符就是 FORTRAN 77 字符集中的专用字符。运算符包括算术运算符、字符运算符、关系运算符和逻辑运算符等。

1.3.3 FORTRAN 77 中的实体

FORTRAN 77 中的实体就是程序中的运算对象,它们包括变量、数组、子串、过程以及虚元和实元等。为了以后讨论的方便,这里先简单介绍变量、数组和子串的初步概念,在第七章和第八章中将详细讨论数组和子串。第二章将简单介绍虚元、实元和过程的初步概念,第十章中再对它们加以详细讨论。

变量 FORTRAN 77 中变量的值不是固定不变的,可以用输入语句给它赋以初值,在程序的执行过程中,还可以用赋值语句给它赋以另外的新值。在程序执行的不同时刻,变量可以有不同的定义,即具有不同的确定的值。这些时刻都称