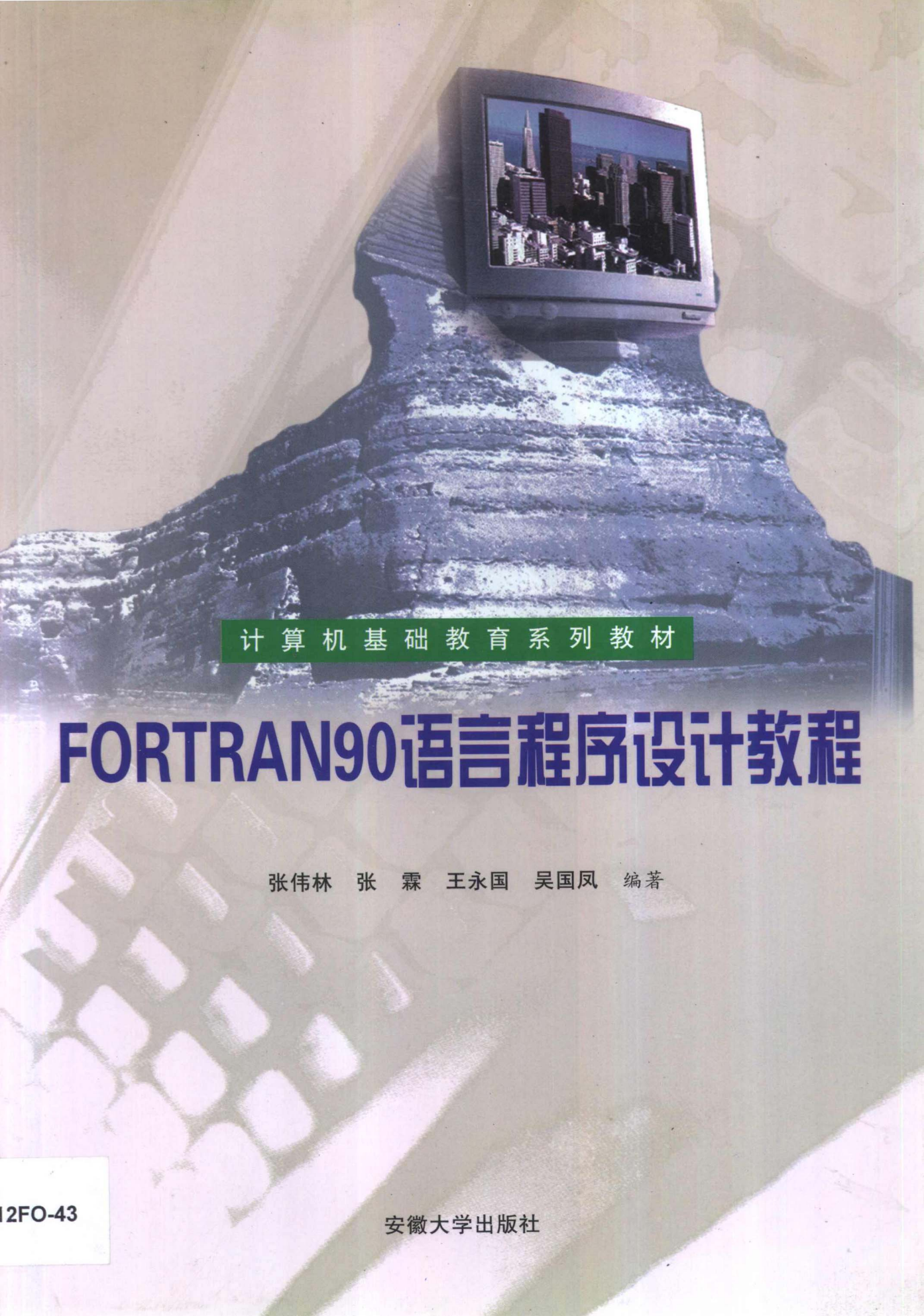




计算机基础教育系列教材

FORTRAN90语言程序设计教程

张伟林 张 霖 王永国 吴国凤 编著

12FO-43

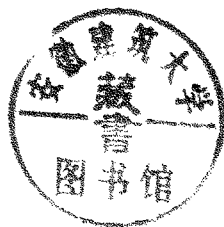
安徽大学出版社

302

计算机基础教育系列教材

FORTRAN 90 语言程序设计教程

张伟林 张 霖 王永国 吴国凤 编著



安徽大学出版社

图书在版编目(CIP)数据

FORTRAN 90 语言程序设计教程 / 张伟林等编著. — 合肥: 安徽大学出版社, 2002. 3
ISBN 7-81052-523-9

I . F… II . ①张… III . FORTRAN 语言—程序设计—教材 IV . TP312

中国版本图书馆 CIP 数据核字(2002)第 005981 号

FORTRAN 90 语言程序设计教程 张伟林 张 霖 王永国 吴国凤 编著

出版发行 安徽大学出版社
(合肥市肥西路 3 号 邮编 230039)
联系电话 总编室 0551-5107719
发行部 0551-5107784

E-mail: ahdxcchps@mail.hf.ah.cn

责任编辑 李 虹 钟 蕾

封面设计 张 霖

经 销 新华书店
印 刷 合肥育才印刷厂
开 本 787×1092 1/16
印 张 14.25
字 数 347 千
版 次 2002 年 3 月第 1 版
印 次 2002 年 3 月第 1 次印刷

ISBN 7-81052-523-9 / TP·63

定 价 17.80 元

如有影响阅读的印装质量问题,请与出版社发行部联系调换

内 容 简 介

FORTRAN 90 是 ISO(国际标准化组织)于 1991 年推出的标准。FORTRAN 90 在保留 FORTRAN 77 全部优点的基础上,增加了许多新功能。到目前为止,FORTRAN 语言仍然是工程分析和数值计算的最方便、最有效的计算机高级语言之一。

本书全面介绍了 FORTRAN 90 的语言特征,包括入门知识、三种基本控制结构、过程、数组、字符处理、派生结构、模块、接口块、指针和递归等。本书针对初学者的特点,在章节安排上由浅入深,循序渐进,通俗易懂,例题丰富。

本书可作为大中专院校和成人教育的计算机高级语言教材,也可供科研人员、工程技术人员作为自学参考书使用,又可作为实用工具书供 FORTRAN 90 用户参考和查询。

编 委 会 名 单

主 任:孙家启

委 员:	王忠仁	王世好	王志雁	王永国
	方潜生	尹荣章	刘 莉	冯年荣
	冯崇岭	孙家启	孙道德	仲 红
	江 涛	朱学勤	齐学梅	吴国风
	李 雪	何 明	张伟林	张国平
	林学华	陈桂林	陈高潮	郑尚志
	胡宏智	周鸣争	周恒忠	姚合生
	赵林玲	聂会星	徐奇观	徐精明
	黄海生	程承士	谢荣传	蔡之让
	潘 瑜	(按姓氏笔划为序)		

秘书长:王忠仁 聂会星

编写说明

为了支持计算机基础教育改革与建设,促进计算机基础课程教学与水平考试向纵深发展,我们按照计算机文化基础教育、技术基础教育和应用基础教育三个层次,组织编写了计算机基础教育系列教材。这套教材囊括了计算机文化基础、计算机文化基础上机操作教程、高级语言(QBasic, Visual Basic, C, Visual C++, PASCAL, FORTRAN77, FORTRAN90, FoxPro For Windows, Visual FoxPro等)程序设计、软件技术基础、微型计算机原理、计算机网络、微型机组装与维护、CAI课件制作及应用等方面内容,涵盖计算机水平考试的一、二、四级(全国等级考试的一、二、三级),因而具有广泛的适应性。这套教材所具有的突出特点是:紧扣计算机基础教育大纲(即计算机水平考试大纲),兼具普通教材与考试辅导材料的双重功能;立意创新,内容简练,其大量针对性极强的习题和典型例题分析为其他教材所少见;编写人员都是教学、科研等一线有着丰富教学与实践经验的教师,他们深谙相关知识点的张弛取舍。我们还聘请了三位知名专家担任高级顾问,以确保本系列教材的编写质量。

本系列教材的先期版本现已问世,第一辑各册已于1999年底全部出齐。由于计算机技术的发展比人们想象的还要快,所以本系列教材又增加了不少新内容,我们今后还将不断调整教材内容、平台和版本,使之与届时发展相适应,以便教材以更新更好的面目呈现在读者面前。

本系列教材编写目的明确,它特别适合于普通高校非计算机专业的本、专科教学用教材或成教、夜大、函大计算机专业的教材,也可供各地计算机水平考试考点使用,还可供广大计算机自学者、工程技术人员参考。

编写委员会

2001年3月

前 言

FORTRAN 语言是最早出现的,也是最富有生命力的计算机语言。FORTRAN 语言内容丰富、功能强大。自问世以来经多次改进、完善、更新和升级,目前仍是广泛应用于科技与工程计算的最有效、最方便的程序设计语言之一。

FORTRAN 90 是国际标准化组织于 1991 年推出的 FORTRAN 标准, FORTRAN 90 对以前的版本作了大量的改进,具有许多优秀特征。本书以 FORTRAN 90 标准(ISO/IEC 1539:1991)为基础,详细介绍了 FORTRAN 90 的语法结构、语句功能和程序设计方法与技巧。内容涵盖了计算机等级考试大纲,可作为大中专院校和成人教育的计算机高级语言教材,也可供科研人员、工程技术人员作为自学参考书使用,又可作为实用工具书供 FORTRAN 90 用户参考和查询。

本书在编写过程中力求做到内容系统、概念明确、条理清楚、由浅入深、循序渐进、语言流畅、重点突出,兼顾初学者的特点,列举大量的例题,给出必要的说明和例题分析。每章配有各类题型的习题,供读者学习和自测。

本书共分为 12 章。第 1 章介绍 FORTRAN 语言的发展概况、语言的特点和简单的 FORTRAN 程序分析;第 2 章介绍 FORTRAN 语言的基础知识;第 3 章介绍 FORTRAN 语言简单的程序设计;第 4 章讲述逻辑 IF 语句、块 IF 结构、CASE 结构等选择结构程序设计;第 5 章讲述循环结构程序设计;第 6 章介绍函数子程序和子例行子程序等过程知识、虚参与实参之间的数据传递;第 7 章介绍数组的定义与应用;第 8 章讲述字符数据处理;第 9 章介绍派生结构的应用;第 10 章介绍模块和接口块的应用;第 11 章介绍指针、链表与递归;第 12 章介绍文件的概念与应用。书中最后列出了三个附录供查询和参考。

本书共 12 章,由张伟林、张霖、王永国、吴国凤合作编著,最后由张伟林教授负责整理定稿。合肥工业大学的孙家启教授审阅了本书的编写大纲,给予了精心指导和支持,并认真审阅了全书,提出了宝贵意见。安徽大学出版社为本书的出版作出了巨大努力。在此表示衷心的感谢。

由于时间仓促和水平有限,本书可能有漏错和不足之处,我们真切希望得到广大读者的批评指正。

编 者
2002 年 2 月

目 录

第 1 章 FORTRAN 90 概述	1
1.1 FORTRAN 语言的发展概况	1
1.2 FORTRAN 90 语言的特点	2
1.3 简单的 FORTRAN 90 程序分析	3
习题	4
第 2 章 FORTRAN 90 语言基础	5
2.1 FORTRAN 90 字符集	5
2.2 固有数据类型	5
2.3 常量与变量	11
2.4 FORTRAN 函数	16
2.5 算术表达式	17
习题	19
第 3 章 FORTRAN 90 简单程序设计	21
3.1 FORTRAN 语句的分类	21
3.2 赋值语句	23
3.3 简单的输入输出语句	26
3.4 参数说明语句	33
3.5 格式输入输出	34
3.6 PROGRAM, END, STOP, PAUSE, GOTO 语句	40
3.7 简单程序设计举例	41
习题	44
第 4 章 选择结构程序设计	46
4.1 关系表达式与逻辑表达式	46
4.2 块 IF 结构与 IF 语句	51
4.3 CASE 结构	56
4.4 选择结构程序设计举例	59
习题	61
第 5 章 循环结构程序设计	64
5.1 有循环变量的 DO 循环结构	64
5.2 重复 DO 循环结构	66
5.3 DO - WHILE 控制的循环结构	69
5.4 循环的嵌套	70
5.5 循环结构程序设计举例	73
习题	75

第 6 章 过程	77
6.1 概述	77
6.2 函数子程序	78
6.3 子例行子程序	83
6.4 虚参与实参之间的数据传递	87
6.5 过程的应用举例	92
习题	93
第 7 章 数组	94
7.1 数组的定义与操作	94
7.2 动态数组	101
7.3 数组在函数和子程序中的应用	102
7.4 数组的应用举例	106
习题	113
第 8 章 字符数据处理	115
8.1 字符型数据	115
8.2 字符型数据的运算	117
8.3 字符子串	119
8.4 字符型数据的应用举例	120
习题	123
第 9 章 派生类型	125
9.1 派生类型的定义	125
9.2 派生类型变量的说明	127
9.3 派生类型的使用	128
9.4 派生类型的应用举例	131
习题	134
第 10 章 模块	135
10.1 模块的定义	135
10.2 USE 语句	136
10.3 接口块	137
10.4 重载和定义操作符	139
10.5 模块的应用	144
习题	150
第 11 章 指针与递归	151
11.1 指针的概念	151
11.2 指针的使用	155
11.3 指针数组	160
11.4 链表	162
11.5 递归及其应用	173
习题	178

第 12 章 文件	180
12.1 文件的基本概念.....	180
12.2 文件操作语句.....	181
12.3 文件的操作.....	185
12.4 文件的应用举例.....	189
习题.....	195
附录 A ASCII 码表	197
附录 B Fortran 90 的内部过程	199
附录 C Fortran PowerStation 4.0 简介	203

第 1 章 FORTRAN 90 概述

1.1 FORTRAN 语言的发展概况

FORTRAN 是 Formula Translate(公式翻译)的缩写,是世界上最早出现的高级编程语言。它是为科学、工程中能够用数学公式表示的问题而设计的,具有较强的数值计算功能。FORTRAN 语言的思想首先由 John Backus 于 1953 年提出,称为 FORTRAN I。第一个 FORTRAN 程序运行于 1957 年 4 月。随后, FORTRAN 的应用迅速传播,形成了许多不同的版本。较为重要和流行的是 1958 年提出的 FORTRAN II,它在 FORTRAN I 的基础上增加了大量重要的扩充,例如引进子函数等概念。从 1958 年至 1963 年期间, FORTRAN 语言在许多计算机上得以实现,其间又出现了较为流行的 FORTRAN IV。但是 FORTRAN II 和 FORTRAN IV 并不兼容,从而使 FORTRAN 语言的标准化工作提上日程。

1962 年 5 月,美国国家标准学会(American National Standard Institute,简称 ANSI)成立了相关机构来进行 FORTRAN 语言标准化的工作。1966 年 ANSI 正式公布了两个美国标准文本:

- 美国国家标准 FORTRAN(ANSI X3.9-1966),相当于 FORTRAN IV
- 美国国家标准基本 FORTRAN(ANSI X3.10-1966),相当于 FORTRAN II

其中,美国国家标准基本 FORTRAN 是美国国家标准 FORTRAN 的一个子集,从而实现了语言的向下兼容。1972 年国际标准化组织(International Standard Organization,简称 ISO)在 FORTRAN 66 的基础上公布了 ISO FORTRAN 标准(ISO R1539)。它描述了三种级别:

- 基本级(接近于 ANSI X3.10-1966)
- 中间级(介于基本级和完全级之间)
- 完全级(接近于 ANSI X3.9-1966)

美国国家标准 FORTRAN(通常称为 FORTRAN 66)建立之后受到了国际上的广泛接受,统治了几乎所有的计算机领域。但是随着计算机软硬件技术的迅猛发展,尤其是在结构化程序设计方法提出以后, FORTRAN 66 日益不能满足需要。这是因为 FORTRAN 66 不是结构化的语言,尤其以 GOTO 语句为标志,没有很好地实现顺序、分支和循环这三种基本结构的语句。因此许多计算机厂商对 FORTRAN 66 进行了不同程度的扩充。针对这种情况,美国国家标准学会于 1976 年对 1966 年公布的 ANSI X3.9-1966 进行了修订,基本上把各厂商有效的功能都吸收进来,并且还增加了许多新的内容。1978 年 4 月美国国家标准学会正式公布了新的美国国家标准 ANSI X3.9-1978《程序设计语言 FORTRAN》,这就是通常称为 FORTRAN 77 的 FORTRAN 语言标准。FORTRAN 77 向下兼容 FORTRAN 66。1980 年, FORTRAN 77 被 ISO 正式采纳成为国际标准 ISO 1439-1980,该标准同样分为全集和子集。

FORTRAN 77 公布之后在国际上得到了广泛应用,多数计算机系统都配备了 FORTRAN 77 编译程序,国内外关于 FORTRAN 77 的教材和参考资料也相当丰富。但是, FORTRAN 77 还不是完全结构化的语言。国际上为了推动 FORTRAN 语言的结构化和现代化作了不懈的努力。经过长时间酝酿,1991 年 5 月通过了 FORTRAN 90 标准,美国国家标准编号为 ANSI X3.198-1991,国际标准编号为 ISO/IEC 1539:1991。新标准中增加了许多新的特性和功能,其中值得一提的是由我国计算机和信息处理标准化技术委员会程序设计分会提出的多字节字符集数据类型及相应的内部函数。这一数据类型的增加为非英语国家在使用计算机方面提供了极大的方便。之后不久又出现了 FORTRAN 95,并且更新的标准又在积极准备之中。本书应用的主要是 FORTRAN 90 标准。

由此可以看出,FORTRAN 语言的历史虽然最为悠久,但仍在不断地改进和发展。这也是 FORTRAN 语言保持旺盛的生命力的原因之一。

1.2 FORTRAN 90 语言的特点

FORTRAN 90 除保持了 FORTRAN 77 的全部优点外,又加进了许多具有现代特征的新功能,对 FORTRAN 77 做了较大的扩充和完善,为 FORTRAN 语言注入了新的活力。FORTRAN 90 最显著的扩充主要有如下 7 个方面:

- (1)引入了数组运算。
- (2)提高了数值计算的功能。
- (3)内在数据类型的参数化。
- (4)用户定义的数据类型。
- (5)用户定义的运算和赋值。
- (6)引入模块数据及过程定义的功能。
- (7)引入指针概念。

另外,还包括了其他一些扩充。例如,改进了源程序的书写形式、引入了更多的控制结构和递归过程、新的输入/输出功能及动态可分配数组等。

FORTRAN 90 的先进性,体现在以下几个方面:

(1)增加了许多具有现代特点的项目和语句,用新的控制结构实现选择、分支与循环操作,真正实现了程序的结构化设计。

(2)增加了结构块、模块及过程调用的灵活性,使源程序易读易维护。

(3)吸收了 C,PASCAL 语言的长处,淘汰或拟淘汰 FORTRAN 77 中过时的语句,具有现代语言的特色。

(4)在 FORTRAN 77 数值计算的基础上,进一步发挥了数值计算的优势。新增了许多先进的调用手段,扩展了 FORTRAN 77 的操作功能。

(5)增加了多字节字符集的数据类型及相应的内部函数,允许在字符数据中选取不同种别,在源程序字符串中可以使用各国文字和各种专用符号,对非英语国家使用计算机提供了更大更有效的支持。

(6)FORTRAN 77 程序仍能在 FORTRAN 90 编译系统下运行,即具有对 FORTRAN 77 的向下兼容性。

1.3 简单的 FORTRAN 90 程序分析

相对于 FORTRAN 77 版本, FORTRAN 90 具备了较为自由的书写格式, 不再有标号区、续行标志区、语句区及注释区的划分, 在合乎语法的前提下, 让使用者拥有了更大的自主权。一个简单的 FORTRAN 90 程序基本结构如下:

```
[PROGRAM 程序名称]
[说明语句部分]
[执行语句部分]
END [PROGRAM [程序名称]]
```

其中, PROGRAM 语句称为主程序语句, 说明语句用于对用户命名的表示符进行定义或说明, 执行语句是程序运行时要进行的操作。END 语句是程序结束语句。

上述[]内的内容是可选的, 在本书中约定, 方括号[]中的内容表示是可选项。

可以看出, 只有 END 语句才是一个 FORTRAN 程序所必需的。对 FORTRAN 程序而言, END 语句意味着程序的编译到此为止, 即程序的结束。

注意: 在没有程序名称的时候, 程序开头的 PROGRAM 应该省略。如果使用了程序名称且在 END 语句中出现, 则 END 语句中的 PROGRAM 不能省略。

为了增加程序的可读性, 建议不要省略 PROGRAM 语句。

语句是一个程序的基础, FORTRAN 90 的语句行可以是 0 到 132 个字符。FORTRAN 90 允许出现空语句行, 使用空语句行可以使程序的可读性增强。早期的 FORTRAN 版本要求语句行的特定部分从特定的列开始, 而 FORTRAN 90 取消了这个要求。

在 FORTRAN 90 中, 除了赋值语句之外, 每个语句都要使用关键词开头。所谓关键词, 是指 FORTRAN 系统使用的具有一定含义和功能的词, 如 PROGRAM, END 等。

FORTRAN 90 对字母的大小写是不加以区分的, 用户可以根据自己的习惯使用大小写。为了便于阅读, 本书的程序一律采用大写。

一般而言, 每行有一条语句。如果希望一行中出现多个语句, 语句间要用分号隔开。为了增加程序的紧凑性, 较短的语句可以放在一行使用, 如简单的赋值语句:

```
BOOK = 4; NOVEL = 3; PROSE = 5
```

在编程的时候, 使用空格往往可以调整程序的格式, 如使语句缩进等, 这样可以增强程序的可读性, 给程序的调试带来了方便。但是, 应该注意的是, 空格并不是随处都可以使用的, 一个像关键词、变量、常量名以及操作符等的字符, 其内部是不能使用空格的, 空格会使字符失去其原有的含义。

FORTRAN 90 的注释语句是以感叹号为标志的, 注释语句的目的是为了增加程序的可读性。在程序编译的时候, 一行中感叹号后的所有字符都被编译器作为注释语句而忽略。注释语句可以单独占一行, 也可在程序的其他语句后面出现。

在 FORTRAN 90 中, 空行也可作为注释语句。

在编程的时候, 如果遇到一条语句的长度超过了 FORTRAN 90 所允许的行最大长度, 需要写成几行时, 第 1 行称为起始行, 从第 2 行开始以后的各行称为继续行。这时, 编译器

如何来辨认某一行是上一行的继续行而不是新语句呢? FORTRAN 90 提供了一个续行符 (&)来解决这个问题,通过在语句末尾添加续行符,编译器就会把下一行作为继续行来处理。如果是把一个如关键字这样的字符分成两行,那么需要在下行语句的开头再加一个续行符,这样编译器就能够把这个分离的字符当作一个完整的字符处理。不过,一个严谨的编程人员是不会轻易使用这种方法的,因为它破坏了程序中字符的完整性,给阅读程序带来一定的困难。

如果续行符出现在注释语句中,则失去了续行的功能。

下面给出一个简单的例子,以对上述内容做一个直观的解释。

PROGRAM SAMPLE

! 该程序为了直观解释上述内容

! 上述空行可作为注释语句

CHARACTER(LEN=20):: NAME

PRINT *

NAME='WYG'

PRINT *, '您好,', & ! 使用续行符 &

NAME

PRINT *

PRINT *, '欢迎进入 FORTRAN 90 世界!'

END PRO&

&GRAM SAMPLE ! 续行符也可连接一个关键字,但这种方法不推荐使用

习 题

1. FORTRAN 90 的特征与新功能。
2. FORTRAN 90 程序的基本结构。

第 2 章 FORTRAN 90 语言基础

2.1 FORTRAN 90 字符集

FORTRAN 90 中的字符集与 FORTRAN 77 相比,有较大的扩充和更大的灵活性,其中_(下划线)、!(叹号)、“(双引号)、%(百分号)、&(AND)、;(分号)、>(大于号)、<(小于号)、?(问号) 9 个字符是 FORTRAN 90 新增的。FORTRAN 90 字符集由下面字符组成:

(1)所有大写和小写英文字母:

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz

在控制程序方面不区分字母的大小写(除了字符常量)。

(2)十个阿拉伯数字:0123456789。

(3)下划线(_).

(4)其他特殊字符,如表 2.1 所示:

表 2.1 特殊字符

字符	名称	字符	名称
空格或(TAB)	空格或 TAB	:	冒号
=	等号	!	感叹号
+	加号	"	引号
-	减号	%	百分号
*	星号	&	AND(与)符号
/	斜线	;	分号
(	左括号	<	小于号
)	右括号	>	大于号
,	逗号	?	问号
.	句号(小数点)	\$	美元符号
'	撇号		

(5)其他可打印字符。

2.2 固有数据类型

FORTRAN 90 的数据类型十分丰富,既有 FORTRAN 语言预先定义的内部数据类型——固有数据类型,又有可由用户通过内部数据类型定义和使用的派生数据类型。本节将介绍固有数据类型,而派生数据类型将在以后的章节里加以介绍。

FORTRAN 90 的固有数据类型包括:

整型 INTEGER, INTEGER(1), INTEGER(2), INTEGER(4);
 实型 REAL, REAL(4), DOUBLE PRECISION, 和 REAL(8);
 复型 COMPLEX, COMPLEX(4), DOUBLE COMPLEX 和 COMPLEX(8);
 字符型 CHARACTER [(LEN = N)], N 是字符串长度;
 逻辑型 LOGICAL, LOGICAL(1), LOGICAL(2), LOGICAL(4)。

每种数值类型都包括 0, 0 既不认为是正值也不认为是负值。有符号的零与无符号的零相等。

每个固有类型都有一个种别(KIND)参数, 它进一步描述了数据类型。在数值类型中, 种别参数描述了精度和十进制指数范围。对于字符型只有一种种别。各种数据类型的缺省种别参数如表 2.2 所示。

表 2.2 各种数据类型的内存需求和缺省种别

数据类型	种别	字节	说 明
BYTE	1	1	和 INTEGER(1)相同
INTEGER	2, 4 或 8	2, 4 或 8	缺省字节为 4
INTEGER(1)	1	1	
INTEGER(2)	2	2	
INTEGER(4)	4	4	
REAL	4 或 8	4 或 8	缺省字节为 4
REAL(4)	4	4	
DOUBLE PRECISION	8	8	和 REAL(8)相同
REAL(8)	8	8	
COMPLEX	4 或 8	8 或 16	缺省字节为 8
COMPLEX(4)	4	8	
DOUBLE COMPLEX	8	16	和 COMPLEX(8)相同
COMPLEX(8)	8	16	
CHARACTER	1	1	CHARACTER 和 CHARACTER (1)相同, (1)是种别参数而不是字符串长度
CHARACTER * N CHARACTER(LEN = N)	1	1	N 是字符串长度; 在 INTEL CPU 上从 1 到 65535, 在 AL-PHA CPU 上从 1 到 $2 * 31 - 1$
LOGICAL	2, 4 或 8	2, 4 或 8	缺省字节为 4
LOGICAL(1)	1	1	
LOGICAL(2)	2	2	
LOGICAL(4)	4	4	

2.2.1 整型数据

整型可以定义为 INTEGER, INTEGER(1), INTEGER(2), INTEGER(4)。还可以指定为 INTEGER * 1, INTEGER * 2 或 INTEGER * 4。可以通过使用 /INTEGER _ SIZE:

SIZE 来改变整型的缺省长度。

整型定义语句的语法如下：

```
INTEGER [( [KIND= ]种别值) ] [ [ ,属性列表 ] :: ] 实体表
```

内部查询函数 KIND 可以返回整型数据的种别参数；内部查询函数 RANGE 可以返回指数范围；内部查询函数 SELECTED _ INT _ KIND 可以找出提供指定整数范围的种别值。如表 2.3 所示，列出了每种数据所占的字节和范围。

表 2.3 整型数据类型的字节和范围

数据类型(种别)	字节	范 围
INTEGER(1)	1	有符号：- 128 ~ 127；无符号：0 ~ 255
INTEGER(2)	2	有符号：- 32 768 ~ 32 767；无符号：0 ~ 65 535
INTEGER(4)	4	有符号：- 2 147 483 648 ~ 2 147 483 647 无符号：0 ~ 4 294 967 295

下面的例子说明了整型变量的定义方式：

```
INTEGER, DIMENSION( : ), POINTER :: DAYS, HOURS  
INTEGER(2), POINTER :: K, LIMIT  
INTEGER(1), DIMENSION(10) :: MIN
```

属性定义的例子：

```
INTEGER DAYS, HOURS  
INTEGER(2) K, LIMIT  
INTEGER(1) MIN  
DIMENSION DAYS( : ), HOURS( : ), MIN(10)  
POINTER DAYS, HOURS, K, LIMIT
```

2.2.2 实型数据

实型数据以单精度数和双精度数来近似数学上的实数。这些数据也称为浮点数。语句 REAL 可以定义实型变量或函数。单精度用 REAL(4) 定义，双精度用 REAL(8) 或 DOUBLE PRECISION 定义。实型定义语句的语法如下：

```
REAL [( [KIND= ]种别值) ] [ [ ,属性列表 ] :: ] 实体表
```

缺省的种别值为 4，可以通过使用 /REAL _ SIZE:SIZE 来改变实型的缺省长度。内部查询函数 KIND 可以返回实型数据的种别参数；内部查询函数 SELECTED _ REAL _ KIND 可以找出提供指定精度和指数范围的种别值；内部查询函数 RANGE 可以返回指数范围。

下面的例子说明了实型变量的定义方式：

```
REAL(KIND=HIGH), OPTIONAL :: TESTVAL  
REAL, SAVE :: A(10), B(20,30)
```

属性定义的例子：

```
REAL(KIND=HIGH) TESTVAL  
REAL A(10), B(20,30)  
OPTIONAL TESTVAL  
SAVE A, B
```


如表 2.4 所示,列出了每种数据所占的字节和范围。

表 2.4 实型数据类型的字节和范围

数据类型	字节	精 度	范 围
REAL(4)	4	最左边的 7 位	负数: $-3.40282347\text{E}+38 \sim -1.17549435\text{E}-38$ 正数: $+1.17549435\text{E}-38 \sim +3.40282347\text{E}+38$
REAL(8)或 DOUBLE PRECISION	8	最左边的 15 位	负数: $-1.7976931348623158\text{D}+308$ $\sim -2.2250738585072013\text{D}-308$ 正数: $+2.2250738585072013\text{D}-308$ $\sim +1.7976931348623158\text{D}+308$

下面的实型双精度常数都代表了 0.052:

5.2D-2, +.00052E+2 _ 8, .052D0, 52D-3, .052 _ 8, 52.000E-3 _ DBL

其中参数 DBL 是在前面定义的符号常数,用来指定双精度,即 PARAMETER DBL=8。

2.2.3 复型数据

COMPLEX 或 COMPLEX(4)(COMPLEX * 8)数据类型是一对有顺序的单精度实数。DOUBLE COMPLEX 或 COMPLEX(8)(COMPLEX * 16)数据类型则是一对有顺序的双精度实数。例如:

```
COMPLEX(4):: C
```

```
C = (3.0,4.0)
```

其中前一个数是实部,后一个是虚部。实部和虚部的种别是相同的。单精度复数在内存中占 8 字节,双精度占 16 字节。由于复数是由两部分组成,所以在输入/输出时要给予特别注意。

复型数据的定义语法为:

```
COMPLEX [( [KIND= ]种别值)] [ [,属性列表] :: ] 实体表
```

复型数据定义的例子:

```
COMPLEX(4),DIMENSION(8):: CZ,CQ
```

属性定义的例子:

```
COMPLEX(4)CZ,CQ
```

```
DIMENSION(8)CZ,CQ
```

复型常量的形式:(C,C)

C 的取值为:对于单精度复型常量,C 是整型或 REAL(4)型常量;对于双精度复型常量,C 是整型、REAL(4)型、或双精度(REAL(8))型,其中至少有一个数据是双精度。

例如:(1.0, -1.0),(4,4.2E3),(5.0 _ 8,8.3E9 _ 8)。

2.2.4 字符型数据

字符型数据又称为字符串,字符型数据值的集合为字符。每个字符在内存中占一个字节,定义字符型变量的关键词是 CHARACTER,其定义的形式是:

```
CHARACTER [类型参数] [ [,属性列表] :: ]变量名
```

类型参数可以使用下面两种形式: