

● 高等学校教学用书 ●

FORTRAN 77 语言程序设计

姜汉民 王九玲 姜梅 编著

G A O D E N G
X U E X I A O
J I A O X U E
Y O N G S H U

冶金工业出版社

高等学校教学用书

FORTRAN77

语言程序设计

姜汉民
王九玲 编著
姜梅

青岛建筑工程学院

冶金工业出版社

图书在版编目(CIP)数据

FORTRAN77 语言程序设计/姜汉民等编著. -北京:冶金工业出版社,1997.8
高等学校教学用书
ISBN 7-5024-2049-5

I. F… I. 姜… III. FORTRAN 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字(97)第 09067 号

出版人 卿启云(北京沙滩嵩祝院北巷 39 号,邮编 100009)
中国人民警官大学印刷厂印刷;冶金工业出版社出版;各地新华书店发行
1997 年 8 月第 1 版,1997 年 8 月第 1 次印刷
787mm×1092mm 1/16; 12.5 印张; 301 千字; 193 页; 1-3200 册
14.80 元

前 言

FORTRAN 语言是国际上最流行的程序设计语言,在数值计算领域获得了最广泛的应用。FORTRAN77 与老版本 FORTRAN66(FORTRAN IV)兼容,但对 FORTRAN66 作了重大改进和扩充:改进了判断控制结构,使 FORTRAN77 成为一种结构化程序设计语言;增加了字符处理能力,使 FORTRAN 语言不仅具有很强的数值计算能力,而且也具有较强的数据处理能力;增加了表控输入/输出;增强了文件处理能力。随着计算机科学的迅速发展和普遍应用,工程技术人员和理工科学生掌握好 FORTRAN77 程序设计语言是十分重要的。

FORTRAN77 是一种功能较强的程序设计语言,但是对初学者常感内容庞杂、方法繁难。本书在兼顾 FORTRAN77 语言完整性的前提下,贯彻“打好基础、精选内容、优化结构、利于教学”的原则,力图做到结构合理,要点突出,内容适当,好教易学。几年来按照这一原则经过多轮教学实践和校际交流,编者进行多次改写。本书是在内部教材的基础上经过进一步修改、充实定稿的。本书适于理工科高校非计算机专业“FORTRAN 语言”课教学,也可供科技人员学习参考。

为了贯彻上述原则,本书以 FORTRAN77 标准文本为依据,以语言结构为主要线索,较全面系统地介绍了 FORTRAN77 全集语言的结构、主要成份以及使用该语言编写程序的基本方法。在内容上突出要点、兼顾一般、舍弃个别有违结构化程序设计原则的内容(多重入口和交叉返回等)。对于重要概念和语言成份按标准完整、准确地介绍,以利读者掌握语言内容;对于算法,本书只介绍最基本的数值运算和非数值运算的算法,而不涉及高深、复杂的算法和数据结构;对于程序设计的基本方法,本书突出结构化程序设计方法,以使编写的程序有良好的结构和风格,利于提高程序设计的效率和质量。书中所选例题紧紧围绕相应章节的内容,说明语言的基本概念、语法规则和编程技巧。

本书由姜汉民、王九玲和姜梅编写。姜汉民编写第四、五、六、七、八、九、十章,王九玲编写第一、二、三章,姜梅对第六~第十章内容作了补充、修改,并增补了较多例题,全书由姜汉民定稿。

我们感谢青岛海洋大学计算机科学系郝羽副教授,她对教材初稿作了认真细致的审核,并提出了许多宝贵意见。还要感谢我院教材科的关心和支持。

由于编者水平所限,书中难免有错误和疏漏,恳请读者批评指正。

编 者

1997 年 3 月

目 录

第一章 计算机和程序设计语言

- § 1-1 计算机的组成..... (1)
- § 1-2 计算机语言和编译程序..... (2)
- 习题一..... (4)

第二章 基础知识

- § 2-1 简单 FORTRAN77 程序分析 (5)
- § 2-2 FORTRAN77 程序的书写格式 (6)
- § 2-3 数据类型、常量和变量 (9)
- § 2-4 内部函数 (13)
- § 2-5 算术运算符和算术表达式 (16)
- § 2-6 算术赋值语句 (21)
- § 2-7 FORTRAN77 程序结构、几个常用语句 (22)
- 习题二 (24)

第三章 判定结构和控制转移语句

- § 3-1 判定结构、程序流程图..... (28)
- § 3-2 关系表达式 (29)
- § 3-3 逻辑表达式、逻辑 IF 语句、GO TO 语句 (30)
- § 3-4 基本 IF—THEN—ELSE 结构 (34)
- § 3-5 多路判定结构、ELSE IF 语句 (37)
- § 3-6 其他控制语句 (41)
- 习题三 (43)

第四章 循环结构和循环控制语句

- § 4-1 “当型”循环 (46)
- § 4-2 “直到型”循环 (47)
- § 4-3 DO 循环和循环语句 (49)
- § 4-4 循环的嵌套 (53)
- § 4-5 程序举例 (55)
- 习题四 (59)

第五章 基本输入输出

- § 5-1 输入输出的几个基本概念 (62)
- § 5-2 表控输入 (63)
- § 5-3 表控输出 (66)
- § 5-4 格式输出语句和格式说明 (69)
- § 5-5 格式输出语句的执行 (76)
- § 5-6 格式输出的其他形式 (77)

§ 5-7 格式输入	(78)
§ 5-8 程序举例	(81)
习题五	(83)
第六章 数组	
§ 6-1 数组和数组元素	(85)
§ 6-2 数组的定义、数组的结构	(87)
§ 6-3 数组的输入输出	(90)
§ 6-4 DATA 语句	(94)
§ 6-5 程序举例	(96)
习题六	(100)
第七章 字符数据处理	
§ 7-1 字符型数据	(103)
§ 7-2 字符表达式和字符赋值语句	(106)
§ 7-3 字符关系表达式	(107)
§ 7-4 字符型数据的输入和输出	(109)
§ 7-5 字符型内部函数	(113)
§ 7-6 程序举例	(114)
习题七	(117)
第八章 过程	
§ 8-1 语句函数	(121)
§ 8-2 外部函数	(124)
§ 8-3 子例行程序	(134)
习题八	(142)
第九章 程序单位间的数据通讯	
§ 9-1 哑实结合	(145)
§ 9-2 公用区(块)	(161)
§ 9-3 BLOCK DATA 子程序(数据块子程序)	(165)
§ 9-4 EQUIVALENCE 语句、公用区的扩充	(166)
§ 9-5 SAVE 语句	(168)
习题九	(169)
第十章 文件	
§ 10-1 记录和文件	(172)
§ 10-2 文件的输入输出	(174)
§ 10-3 辅助输入输出语句	(175)
§ 10-4 程序举例	(181)
习题十	(184)
附 录	(186)
参考文献	(193)

第一章 计算机和程序设计语言

自从 1946 年第一台电子数字计算机问世以来,电子数字计算机经历了四个发展阶段:第一阶段(第一代)的电子数字计算机使用电子管作逻辑元件;第二阶段(第二代)使用晶体管作为逻辑元件;第三阶段(第三代)使用集成电路作逻辑元件;第四阶段(第四代)采用大规模集成电路作逻辑元件。计算机的硬件功能获得了迅速的发展和很大的提高。举例来说,1946 年研制出的第一台电子计算机 ENIAC,其速度仅为定点加法运算 5000 次/秒;而第四代机最快的运算速度已达几亿次/秒。伴随着硬件的发展,计算机的软件也经历了从无到有、从低级到高级的不断完善过程。计算机配备了日益完善的软件系统,大大增强了计算机的功能,方便了用户。

这四代计算机尽管其功能强弱差异很大,但其基本工作原理都是“存储程序”原理。这一原理是美籍匈牙利数学家冯·诺依曼(Von Noumann)于 1946 年提出来的。按照这一原理,要想让计算机自动地、快速地进行运算,必须事先把编制好的程序和数据存储在计算机内。计算机逐条执行程序中的指令就能完成预定的运算任务。冯·诺依曼明确提出,计算机至少应由五个基本部分组成:运算器、控制器、内部存储器、输入设备和输出设备。提出信息在计算机内部应以二进制数表示。

正在研制的第五代计算机将是一种非诺依曼型计算机,它采用全新的工作原理和体系结构,具有知识的智能处理等功能。第五代机的研制成功将是对科学技术的突破性贡献,被称为“第二次计算机革命”。

§ 1-1 计算机的组成

现代电子计算机系统是由硬件和软件两部分组成的。硬件是计算机的实际物理装置,它是由电子的、磁的、光的、机械的元件或器件组成。软件是指计算机的程序系统,它是为了充分发挥计算机系统的功能、提高系统的效率和方便用户而配备的。

一、计算机的硬件组成

计算机系统的硬件由控制器、运算器、内部存储器、输入设备和输出设备组成。由这五大部件组成一个严密的、有机的整体,称为硬件系统。控制器和运算器两部分合在一起称为中央处理单元(CPU)。硬件组成如图 1-1 所示。以下简述各部件的功能。

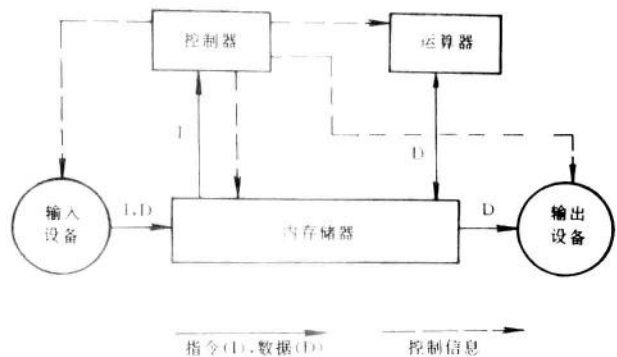


图 1-1

1. 内部存储器

内部存储器又叫主存储器,它是计算机的存储(记忆)部件,用来存放程序和数据。

程序是由指令组成的。指令和数据在计算机内部都用二进制数表示。

内部存储器由存储元件阵列和有关电路组成。用一个存储元件的两种不同电或磁的状态表示一位二进制数的值0或1,它称为二进制位(Bit)。用8个存储元件的状态表示一个8位二进制数,它称为一个字节(Byte)。由若干个字节组成一个字。例如八位微机的一个字由一个字节组成,十六位机的字由两个字节组成。一个字所占用的存储空间称为一个存储单元。

为了信息的存取方便,每个存储单元都编了地址号码,称为单元地址。按单元地址存取单元内容。

内部存储器中又包含随机存取存储器(RAM)和只读存储器(ROM)。随机存取存储器既可存入(写入)信息,又可以取出(读出)信息。只读存储器只能读出信息不能写入信息,用来存放不能随意改变的、最基本最常用的系统程序和程序。例如 IBMPC-XT 微机中,用 40KB 只读存储器存放了系统的基本输入/输出程序(BIOS)和 BASIC 语言解释程序。

一个存储器的规模(大小、容量)由它所能存储信息的最大字节数(或字数)表示。例如 IBMPC-XT 微机的内存储器的容量为 512KB 或 640KB。

存储器的存取信息的速度是另一个重要技术指标,它直接影响计算机系统的运行速度。

2. 控制器

控制器是控制全机协调工作的部件。控制器从内存储器取出指令,经过指令译码产生与此指令相应的一系列控制信息,将它们分送到各有关部件,控制各部件协调工作,完成指令规定的操作。

3. 运算器

运算器是计算机专门进行算术运算和逻辑运算的部件。它在控制器的控制下完成算术运算(加、减、乘、除)和逻辑运算(与、或、非、异或等)。

4. 输入设备

用户通过输入设备将程序、数据送入计算机,存入内部存储器。常用的输入设备有卡片读入机、终端键盘、磁盘驱动器、磁带机、光笔输入器等。

5. 输出设备

输出设备用来输出计算机运行程序的中间结果和最后结果。常用的输出设备有行式打印机、终端显示器、磁盘驱动器、磁带机、绘图仪等。

二、计算机软件

软件按其用途分为系统软件和应用软件两大类。

为了充分利用计算机资源,提高计算机效率和方便用户而设置的软件称为系统软件。它又分为语言编译系统、操作系统、支持软件和数据库管理系统。其中操作系统是系统软件的核心。

用户为了解决特定的实际问题专门编制的程序称为应用软件。把一些应用软件组合起来称为应用软件包。

§ 1-2 计算机语言和编译程序

信息(指令、数据等)在计算机内是用二进制编码表示的。以下先讨论二进制数和二进制编

码,然后再讨论计算机语言和编译程序。

一、二进制数和二进制编码

1. 二进制数

数值数据在计算机内用二进制数表示。二进制数是一种“逢二进一”的数制,只使用 0 和 1 两个数码表示所有二进制数。二进制数的基数是 2,任何一个二进制数都可以表示为 2 的幂的多项式形式。例如:

$$(1101.1)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$$

等号左边是一个二进制数,用角码“2”标出,以便和十进制数相区别。

在计算机内使用二进制数的原因有三:

(1)二进制数只使用数码 0 和 1,易于用一个电子元件的两个不同电状态来表示。

(2)二进制数的运算规则简单。例如加运算、乘运算的规则都仅有四条:

	$0+0=0$		$0 \times 0=0$
(加运算)	$0+1=1$	(乘运算)	$0 \times 1=0$
	$1+0=1$		$1 \times 0=0$
	$1+1=10$		$1 \times 1=1$

(3)使用二进制数,算术运算和逻辑运算可用同一运算器完成,简化了硬件电路。

二进制数书写起来不方便,因此在汇编语言源程序中常常使用十六进制数。十六进制数“逢十六进一”,基数为 16,使用 16 个数码:0,1,2,3,……,9,A,B,C,D,E,F。一个十六进制数可表示为 16 的幂的多项式的形式,例如:

$$(8FED)_{16} = 8 \times 16^3 + 15 \times 16^2 + 14 \times 16^1 + 13 \times 16^0$$

一般在一个十六进制数的末尾加“H”,以标明它是一个十六进制数,例如 8FEDH。一位十六进制数可以转换为四位二进制数,反之亦然。

2. 字符编码

计算机除了能进行数值数据的计算以外,还可以对字符型数据进行运算和处理。常用的字符有 A,B,……,X,Y,Z,0,1,2,……,9,+,-,*,/等。字符在计算机内是用二进制编码表示的。最常用的编码是 ASCII 码(美国标准信息交换码),它是用一个七位二进制编码表示一个字符。例如字符 0~9 的 ASCII 码为 30H~39H,A~Z 的 ASCII 码是 41H~5AH,详见附录中 ASCII 编码表。

二、从机器语言到高级语言

1. 机器语言

每一个型号的计算机都有一套指令,以规定用什么指令完成什么操作。这些指令的集合称为指令系统。

计算机只能直接识别和执行二进制编码形式的指令,这种形式的指令称为机器指令。例如在某计算机上执行机器指令 0000101011011101 实现一次加法运算。

机器指令的集合称为机器语言。机器语言是面向机器的语言,在不同型号的机器上不能通用。用机器语言编写的程序称为目标程序。

由于机器指令的形式是二进制编码,指令不直观、难记忆、易出错,因此用机器语言编制程序的工作较为繁重,而且编制出的程序不便阅读和修改,也无通用性。

2. 汇编语言

为了克服机器语言的缺点,人们设计出一种改进了的语言——汇编语言,汇编语言是一种符号语言,用助记符来表示指令。例如一条加指令的形式和功能是:

指令形式

指令功能

ADD 目标操作数,源操作数

目标操作数 ← 目标操作数 + 源操作数

使用汇编语言编制的程序称为汇编语言程序。汇编语言意义直观、便于记忆、不易出错,汇编语言程序便于阅读和修改。但是,汇编语言仍然是面向机器的语言,没有通用性。

汇编语言为编制程序带来了方便;但是汇编语言程序不能被计算机识别和执行,必须先把汇编语言程序“翻译”成机器语言目标程序,然后由计算机执行。这种“翻译”工作叫作代真,它可以由人工完成,也可以由计算机执行一个专用程序,自动地把汇编语言程序“翻译”成机器语言目标程序。这种完成“翻译”工作的专用程序称为汇编程序。

3. 高级语言

汇编语言和机器语言都是面向机器的语言,没有通用性。因此它们被称为低级语言。用低级语言编制程序繁难,也缺乏通用性,不便于推广。

后来人们创造了高级语言,又称为程序设计语言。高级语言是不依赖于机器的语言,可适用于不同类型的计算机。高级语言的语句和人们使用的习惯语言、数学语言相近,便于理解和使用。例如语句 READ 的作用是输入(读入)数据;语句 $A = (B + C) / 2$ 用来计算 $\frac{B+C}{2}$,并将结果存入变量 A 中。需要指出,高级语言一个语句的作用相当于若干条机器指令的作用。

同样,计算机不能执行用高级语言编制的源程序,而必须先“翻译”成机器语言目标程序,然后才能执行。将高级语言源程序“翻译”成机器语言目标程序的工作由“编译程序”来完成。其工作过程如图 1-2 所示。

从 50 年代后期至今已出现了数百种高级语言,使用较广泛的有 FORTRAN、BASIC、PASCAL、COBOL、C 语言等。

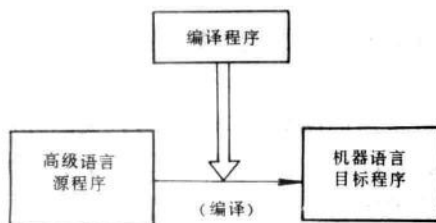


图 1-2

习 题 一

1. 名词解释:字节,硬件,软件,机器语言,汇编语言,高级语言,源程序,目标程序。
2. 计算机硬件由哪几部分组成?简述各部件的功能以及它们之间的关系。
3. 在计算机内为何使用二进制数?使用十进制数可否?为什么?
4. 机器语言、汇编语言、高级语言各有什么特点?
5. 何为编译?编译程序的作用是什么?

第二章 基础知识

FORTRAN 语言是目前在国际上最流行的程序设计语言。从 1954 年首次发表 FORTRAN 语言至今,语言功能几经扩展,已先后提出几种文本:美国 IBM 公司于 1958 年提出 FORTRAN I,1962 年提出 FORTRAN IV。美国国家标准化协会于 1966 年公布了两个标准文本:标准 FORTRAN(即 FORTRAN66,相当于 FORTRAN IV)和标准基本 FORTRAN(相当于 FORTRAN I)。

美国国家标准化协会于 1978 年公布了《美国国家标准程序设计语言 FORTRAN AN-SIX3.9—1978》,即 FORTRAN77。它是 FORTRAN66 的修订本,包括一个全集和子集,与 FORTRAN66 基本兼容,但对 FORTRAN66 作了重大改进:引入了字符处理和表控输入/输出等新功能,并增加了文件处理能力,从而大大增强了数据处理能力;尤其改进了判断控制结构,使 FORTRAN77 成为较好的结构程序设计工具;新的标准还提高了 FORTRAN 程序的可移植性,使之利于推广应用。

§ 2-1 简单 FORTRAN77 程序分析

首先,通过对一个简单的 FORTRAN77 程序的分析,对 FORTRAN77 程序有一个初步的了解。

例[2.1] 已知圆的半径 R,编制一个 FORTRAN77 程序计算圆的周长 C 和圆的面积 A。以下列出圆周长和圆面积的计算公式:

$$C=2\pi\times R$$

$$A=\pi\times R^2$$

根据题目要求,编制 FORTRAN 程序:

	1	2 5	6	7
1				PROGRAM CIRCLE
2	C	THI	S	PROGRAM IS USED TO COMPUTE
3	C	THE		CIRCUMFERENCE AND AREA OF A CIRCLE
4				PARAMETER(PI=3.14159)
5				READ(*,*) R
6				WRITE(*,*) 'THE RADIUS OF THE CIRCLE IS',
7			+	R
8				IF(R.LE.0.0) THEN
9				WRITE(*,*) 'THE RADIUS IS INVALID'
10				STOP
11				END IF
12				C=2.*PI*R

	1	2 5	6	7
13				A=PI*R**2
14				WRITE(*,*)'CIRCUMFERENCE=',C
15				WRITE(*,*)'AREA=',A
16				END

为了说明方便起见,在左边加上了行号,在上边加了列号,在编制的 FORTRAN77 程序中这些行号和列号是没有的。以上程序中,第 1 行是 PROGRAM 语句,它为该程序命名为 CIRCLE。第 2 行和第 3 行是注解行,用来说明该程序是用来计算圆的周长和面积的。第 4 行是为常数 3.14159 命名 PI 的参数语句。第 5 行是输入语句,输入圆半径 R 的值。第 6 行和第 7 行是输出语句,先输出信息:“THE RADIUS OF THE CIRCLE IS”,再输出半径 R 的值。第 8 行是块 IF 语句,根据 (R.LE.0.0) 值来决定程序的走向,如果 (R.LE.0.0) 的值为“真”,则执行第 9 行和第 10 行的两个语句;否则不执行这两个语句,从第 11 行开始继续执行。第 11 行的 END IF 语句与第 8 行的块 IF 语句配合构成一个判定结构,执行 END IF 语句时并不产生动作。第 12、13 行都是赋值语句,分别求出圆周长 C 和面积 A。第 14、15 行都是输出语句,分别输出圆的周长和圆的面积的值。第 16 行是 END 语句,表示程序结束。

从上述 FORTRAN77 程序举例可以看到以下几点:

(1)FORTRAN77 程序由注解行和语句组成。注解行由 FORTRAN77 字符集的字符组成。语句由基本的语法成份和 FORTRAN77 字符集的一些专用符号,按照一定的语法规则组成。

(2)语句中的基本语法成份有:关键字、符号名、常数和运算符,它们也是由 FORTRAN77 字符集中的字符按照一定的语法规则组成的。在上述程序中,PROGRAM、PARAMETER、READ、WRITE、IF、THEN、ENDIF、STOP、FORMAT、END 是关键字,CIRCLE、PI、R、C、A 是符号名,3.14159 和 2 是常数,* 和 .LE. 是运算符。

(3)程序中的语句可分为两类,可执行语句和非执行语句。在上述程序中 READ 语句,WRITE 语句、块 IF 语句、ENDIF 语句、STOP 语句、END 语句和两个赋值语句都属于可执行语句的范围;PROGRAM 语句、PARAMETER 语句属于非执行语句。一般可执行语句规定了计算机系统应该完成的动作;而非执行语句仅向系统提供说明性的信息。

(4)每个语句行分为三个区:标号区,续行标志区,语句内容区。

(5)程序以 END 语句结束。

§ 2-2 FORTRAN77 程序的书写格式

一、FORTRAN77 字符集

FORTRAN77 全集语言的字符集共有 49 个字符。它们是:

英文大写字母: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

数字:0,1,2,3,4,5,6,7,8,9

专用字符见表 2.1 中所列。

表 2.1

字 符	字 符 名 称	字 符	字 符 名 称
	空白)	右括号
=	等号	,	逗号
+	加号	.	小数点
-	减号	\$	货币号
*	星号	'	撇号
/	斜线	:	冒号
(左括号		

为了书写和阅读程序方便,本书用“□”表示一个空白符。当输入源程序时,只需按一下空格键就输入一个空白符;输出空白符时,只输出一个字符宽度的空白。

二、关键字

关键字是一个英文字母序列,用来标识语句,供编译程序识别语句用。在 FORTRAN77 中,除赋值语句和语句函数语句外,每个语句都以一个关键字开头,例如 PROGRAM、READ、WRITE、FORMAT 等。

三、符号名

FORTRAN77 的变量、数组、函数、程序等都用一个符号名来命名,如 CIRCLE、CIRCUM、AREA 等。符号名是以英文字母开头由 1~6 个字母或数字字符组成的字符序列。

四、行

通常 FORTRAN77 源程序按照要求的格式书写在专用的源程序纸上(请参看图 2-1)。源程序纸的每一行有 80 个字符位置(即 80 列),称为第 1 列,第 2 列,……,第 80 列。其中第 1 列至第 72 列用于书写源程序。一行中第 1 列至第 72 列称为一个程序行,它或为一个注解行,或为一个语句行。第 73 列至第 80 列的范围供用户写说明或作标记用。

注解行用于对程序进行必要的注解或说明,以提高程序的可读性,根据需要可写在 END 语句前的任何位置。注解行虽然是源程序的组成部分,但编译程序并不对其进行编译,因此注解行对程序的执行不产生影响。注解行有严格的格式要求:在第 1 列要写字符“C”或“*”,第 2 列至第 72 列书写注解内容。注解行的内容由 FORTRAN77 字符集的任何字符组成。当注解内容在一行内写不完时,可从下一行的第 2 列接着写,但第 1 列也必须写字符“C”或“*”,以表示该行也是注解行。

每个语句行分为三个区。第 1 列至第 5 列称为标号区,用于书写语句标号。语句标号允许使用 1~99999 范围内的任何整数。语句标号只是语句的一个标记,其大小不表示语句的前后顺序。第 6 列称为续行标志区,用于书写续行标志。第 7 列~第 72 列称为语句内容区,用于书写语句的内容。

一个语句行最多只能写一个语句。一个语句在一个语句行内写不完时,可在下一行继续写,语句占用的第一个语句行称为始行,继续占用的语句行称为续行。FORTRAN77 规定一个语句最多可占用 19 个续行。

对始行和续行的标号区、续行标志区有不同要求。始行标号区可以有语句标号,也可以没有语句标号(全是空白),其续行标志区必须为数字 0 或空白符。续行的标号区必须是空白,续

程序名称 计算圆的周长和面积

设计人第 页 共 页
日期 60 65 70 72 75 80

5	6	10	15	20	25	30	35	40	45	50	55	60	65	70	72	75	80
	PROGR	AM	CIRCLE	AM	IS	USED	TO	COM	PUTE								
C	I	S	RCUMF	R		AREA	OF	A	CIRCLE								
C	E	CI	METER	(P1=3.													
				R													
				'THE	RADIUS	OF	THE	CIRCLE	IS',								
	WRITE	(***)	R														
	+			R													
				(R.LE.	0.0)	THEN											
				'THE	RADIUS	IS	INVALI	D'									
	WRITE	(***)															
	STOP																
	END	IF															
	C=2.*	PI*R															
	A=PI*	R**2															
	WRITE	(***)	'CIRCU	MFER	ENCE	=',C											
	WRITE	(***)	'AREA	'AREA	=',A)												
	END																

图 2-1

行的续行标志区必须是除数字 0 和空白符以外的任何 FORTRAN77 字符集的一个字符。

由于空白符仅在字符常数中有意义,而在语句中无意义,因此可以按照程序结构的层次要求,把一个语句的内容写在第 7 列~第 72 列的适当位置。

§ 2-3 数据类型、常量和变量

由例[2.1]程序可见,FORTRAN77 程序由说明部分和执行部分组成。说明部分在前,由非执行语句组成;执行部分在后,由可执行语句组成。执行部分包含数据提供、数据的操作或处理以及数据的输出。因此,数据的定义和类型是 FORTRAN77 的基本概念。

一、数据类型

常数和变量都称为数据,是语句的基本成份之一。数据有两个最基本的特征:数据的类型和数据的值。数据类型的多少直接影响程序设计语言功能的强弱。FORTRAN77 所能处理的数据共有六种:

(1)整型	INTEGER
(2)实型	REAL
(3)双精度型	DOUBLE PRECISION
(4)复型	COMPLEX
(5)逻辑型	LOGICAL
(6)字符型	CHARACTER

二、常数

在程序运行期间保持其确定值不变的数据,称为常数。整型常数、实型常数在 FORTRAN77 中使用最频繁。

1. 整常数

整常数由正号或负号后跟一串数字(0~9)组成,它表示一个十进制的正的或负的整数,其中正号可省略。例如 0,554,+230,-897 是四个整常数。在整常数中不允许出现小数点或其它符号。

一个整常数在计算机内存中以两个字节或 4 个字节存贮,在 PC-XT 微机中一般以两个字节存贮,其中最高二进制位存放正负号,低 15 位存放数据,因此整数范围为 $-2^{15} \sim +(2^{15}-1)$ (即 $-32768 \sim +32767$)。由此可见,整常数在计算机内被精确表示,且形式简单,故使用整形常数运算准确、迅速;但由于整形常数占用内存的位数所限,整型常数的范围小,运算时超出范围即产生“溢出”错误。

2. 实常数

实常数有两种形式:小数形式和指数形式。

(1)小数形式的实常数由正负号、整数部分、小数点、小数部分组成,其中正号可省略。例如:+37.5, -10.023, 0.567, 0.0。

小数形式的实常数允许缺少整数部分或小数部分,但不允许两部分都缺少(即仅剩小数点)。例如,0. , -567. , .86 , -.425 都是合法的小数形式的实常数。

(2)指数形式的实常数是在小数形式的实常数(基本实常数)或整常数后跟一个实指数。例如:

指数形式的实常数	表示的十进制数
3.15E+3	3150.0
0.785E-1	0.0785
36E-2	0.36
-1E-5	-0.00001
395E2	39500.0

其中,实指数的形式是字母 E 后跟一个整常数,它代表 10 的方幂。但需指出,实指数不能单独表示一个实常数。

实常数在内存中一律用指数形式存放。例如,18.6 和 1.86E1 实际表示同一个实数,先将它们转化为规格化指数形式:0.186E2,然后存放在内存的四个相邻字节中(如图 2-2)。



图 2-2

当然,图中的十进制数在内存中是以二进制形式存放的。由于位数所限,数字部分过多的小数位数丢失,可见实常数在内存中是近似表示,且表示形式较整常数复杂。因此使用实常数是近似运算,且运算速度较整常数慢;但是实常数能表示范围大得多的数,例如 PC-XT 机实常数范围为:

$$\begin{aligned} \text{负数: } & -1.7\text{E}+38 \sim -3.0\text{E}-39 \\ & (\text{表示 } -1.7 \times 10^{38} \sim -3.0 \times 10^{-39}) \\ \text{正数: } & 3.0\text{E}-39 \sim 1.7\text{E}+38 \\ & (\text{表示 } 3.0 \times 10^{-39} \sim 1.7 \times 10^{38}) \end{aligned}$$

绝对值小于 3.0×10^{-39} 的数作零处理。

在编程中应根据实际情况来选用整常数或实常数,当数值不太大也不太小而要求绝对精确时,使用整常数;否则,数值范围大时选用实常数。

3. 双精度常数

双精度常数的形式是在整常数或小数形式的实常数后跟一个双精度指数,它较精确地表示一个实数。例如:

双精度常数	较精确表示的数
12D5	$12. \times 10^5$
2.5D-6	2.5×10^{-6}
-8.9D20	-8.9×10^{20}
-0.2708D+3	-0.2708×10^3

字母 D 后跟一个整常数称为双精度指数,表示 10 的方幂。双精度常数在内存中占用了比实常数更多的字节数,因此提高了数的表示精确度;但双精度常数的运算速度比实常数慢。

4. 复型常数

复型常数的形式是括号内两个有序的实常数或整常数,两者间用“,”分隔。它用来表示一

个复数。一个复常数中的两个常数或都为整型或都为实型,必须类型相同。例如:

复常数	表示的复数
(5,12)	5+12i
(-1.6,3.75)	-1.6+3.75i
(0.28E1,2E-2)	2.8+0.02i
(0,1)	i
(4.75,0.)	4.75

可见,括号内第一个实(整)常数表示复数的实部,第二个实(整)常数表示其虚部。

5. 字符常数

字符常数是用分界撇号括起来的一个字符序列,其中的字符是计算机系统允许使用的任何字符。一般,计算机系统允许使用的字符往往多于 FORTRAN77 字符集中的字符。一个字符常数至少含有一个字符。

嵌入分界撇号内的空白符是有效字符。字符常数的值是指分界撇号内的字符串,不包括分界撇号。撇号内使用撇号,该撇号必须使用双撇号“''”,以便和分界撇号相区别。字符常数的长度是指分界撇号内字符的个数,包含其中的空白符和撇号,但不包含分界撇号。

下面列出一些合法的字符常数以及它们的值和长度。

字符常数	值	长度
'ABC'	ABC	3
'A□B□C'	A B C	5
'DON'T'	DON'T	5
'54□RIGHT+1'	54 RIGHT+1	10

6. 逻辑常数

逻辑常数仅有两个值:真,其形式为.TRUE.;

假,其形式为.FALSE.。

三、变量

在程序运行期间其值可变的数据称为变量。例如在例[2.1]程序中 R、C 和 A 都是变量。

1. 变量名

为了识别变量必须给每个变量命名。变量是用符号名来命名的,即变量名是用英文字母开头的由不多于六个字母或数字组成的字符序列。

应注意,变量名不能与关键字相同,以免引起混乱。关键字是 FORTRAN77 规定的一些专用定义符,它们都有固定的含意。例如例[2.1]程序中出现的关键字:READ、WRITE、PROGRAM、END 等,此外还有用作内部函数名的关键字,如 SIN、SQRT 等。

为了使程序增加可读性,变量名应尽量反映其表示的意义或符合习惯,例如可用 AREA 表示面积,用 MAX 表示最大值等。以下列出一些合法的和非法的变量名。

合法的变量名:

ABCH2;X12;T34Y;CHAN。

非法的变量名:

ABC-1

3XY

错误原因

不允许出现字符“-”

不允许以数字开头