

高校非计算机专业计算机教材丛书

FORTAN77

程序设计

陆诗娣 曹豫菽 张伏生 魏恒义



西安交通大学出版社

高校非计算机专业计算机教材丛书

FORTRAN 77 程序设计

陆诗娣 曹豫羲 张伏生 魏恒义

西安交通大学出版社

内 容 提 要

本书系统地介绍了 FORTRAN 77 程序设计语言,全书结构突破传统模式,力求内容更精炼、重点更突出。作者对各章内容及前后顺序的处理均作了精心安排,使概念清晰、深入浅出、通俗易懂。本书便于教学,也便于自学,较多地节省课内教学时数。

本书可作为高等院校本科及专科 FORTRAN 77 程序设计语言课程的教材,也可供初学者自学与参考。

(陕)新登字 007 号

FORTRAN 77 程序设计

陆诗娣 曹豫英 张伏生 魏恒义

责任编辑 陈 丽 叶 涛

*

西安交通大学出版社出版发行

(西安市咸宁西路 28 号 邮政编码 710049)

西安德力彩印厂印装

陕西省新华书店经销

*

开本:787×1092 1/16 印张:15 字数:354 千字

1995 年 9 月第 1 版 1996 年 4 月第 2 次印刷

印数:5001—11000

ISBN 7-5605-0757-3/TP·103 定价:15.00 元

序 言

在普通高等院校中对学生的计算机基础知识与应用能力的培养已经成为各学科各专业教学计划的重要组成部分。高等院校本、专科毕业生的计算机基础知识与应用能力的水平也已成为绝大多数用人单位选择录用人员的重要依据之一。我省各高等院校多年来在计算机基础课程的教学方面进行了精心的组织,工作在计算机基础课程教学第一线的广大教师、工程技术人员和实验室工作人员呕心沥血做了大量艰辛的工作。不过,由于人力、教材、计算机设备等等各方面条件的制约,各院校之间在计算机基础课程教学方面的进展还是不很平衡的。

为了把我省普通高等院校的计算机基础课程教学提高到一个更高的水平,陕西省教委非常重视普通高校计算机基础课程教学的课程建设和各学科各专业学生的计算机知识结构的研究和组织工作。在当前,教材建设又是一项当务之急的基础建设工作。为了使各院校在有关课程的教学内容方面能有一个比较合理、一致的基本要求,省教委邀请了省内近十所高校中工作在有关课程教学第一线的具有丰富教学经验的专家、教授经过认真切磋并参照国家教委工科计算机基础课程教学指导委员会制定的有关课程的教学基本要求,编写了一套“陕西省普通高校非计算机专业计算机教材丛书”。这套丛书包括:《计算机应用基础》(文科类);《计算机应用基础》(理工科类);《BASIC 语言程序设计》;《FORTRAN 语言程序设计》;《PASCAL 语言程序设计》;《COBOL 语言程序设计》;《C 语言程序设计》;《微机原理及接口技术》;《计算机软件基础》共九种教材。这不仅是一套适合普通高等院校作为有关课程教学使用的教材,也可作成人教育及各种专门培训班组织有关课程教学之用,当然也可作为社会上各行各业有关人员学习计算机基础课程的自学教材。这套教材普遍的特点是内容规范、取材精炼、便于组织教学和学生自学。我们相信,这套教材在大面积的使用过程中经过不断的听取意见和锤炼修改,定会成为一套受广大读者欢迎的好教材。

胡正家

1994.02.10

前 言

FORTRAN 是世界上最早出现的计算机高级语言,至今已有 40 年的历史了。它是最适用于数值计算的计算机语言之一,因此,目前许多理工科大学学生都要学习它。

从第一台电子计算机问世至今,计算机学科已经历了近 50 年的发展历程,有关计算机的知识愈来愈多,而且成为当代知识结构中不可缺少的重要部分。对大学生来说,在校学习的时间是有限的,他们在大学阶段不仅要学会掌握应用计算机的有关知识,还需要学会所学专业的知识,因此,用较少的时间让学生尽可能快地掌握计算机知识并应用到所学专业中去是计算机基础教学的目标。

根据国家教委关于计算机基础教育分三层(计算机文化基础、软硬件基础和应基础)进行的有关精神,我们认为计算机语言教学应采用“精讲,多练”的方法进行。为了适应这样的教学方式,必须有相配套的教材,这本 FORTRAN 77 的教材就是在这种背景下编写的。

在这本教材中,我们力求用通俗的语言,学生容易理解的例子将算法,数据结构、计算机语言及结构化程序设计的思想融为一体,使读者能很容易看懂。在内容顺序的安排上也作了一定的改变,在分支结构之前就介绍了函数子程序的概念,这样可使读者早点了解一个完整的 FORTRAN 程序的结构;同时,通过之后诸多例子的运用使读者对子程序的使用更加熟练。我们在 10 多年的教学实践中体会到这样的安排对初学者来说是有效的。书中的例题不仅结合此部分的内容,而且还体现了一定的算法,在阅读这些例题时,希望读者注意从中学习编写程序的方法。书中每部分配置了形式多样的习题,以便读者加深对基本概念的理解和程序设计方法的掌握。

本书共 10 章,参加编写的有张伏生(第 1、2 章),陆诗娣(第 3~8 章),曹豫菽(第 9 章),魏恒义(第 10 章),全书由陆诗娣统稿。

为了做到教学与练习的一致性,我们还将配置相应的《FORTRAN 77 上机练习与指导》一书。

本书在编写过程中,自始至终得到了西安交通大学计算机系冯博琴教授的热情指导、关心和帮助,在此表示衷心的感谢。

由于水平所限,不妥之处在所难免,恳请读者提出宝贵意见。

编者 1995. 3

**陕西高校非计算机专业学生
计算机应用知识与应用能力等级考试
专家委员会名单**

顾问: 胡正家	西安交通大学教授
委员: 冯博琴	西安交通大学教授
张遵谦	西北工业大学教授
罗昌隆	西安电子科技大学教授
卞 雷	西北大学教授
曹豫菽	陕西师范大学副教授
魏文郁	西安冶金建筑学院副教授
王肇荣	陕西机械学院教授
孙明勤	西北农业大学副教授
陈 康	西安医学院高级工程师
李能贵	西安交通大学教务长 教授
鲍国华	西北工业大学副教授
肖兴民	西北大学副研究员
李汝峰	西安电子科技大学副研究员
孙 朝	省教委高教处副处长

目 录

第1章 FORTRAN 语言基础

1.1 计算机与程序设计语言	1
1.1.1 电子计算机的组成	1
1.1.2 程序设计和程序设计语言	3
1.2 FORTRAN 语言简介	5
1.3 FORTRAN 程序一瞥	6
1.3.1 FORTRAN 程序的一般结构	6
1.3.2 FORTRAN 源程序的书写格式	8
1.4 FORTRAN 语言的基本符号及常数	9
1.4.1 FORTRAN 语言的基本符号——字符集	9
1.4.2 常数及其类型	10
1.5 变量及其类型说明	14
1.5.1 FORTRAN 名字	14
1.5.2 变量的概念	14
1.5.3 变量类型	15
1.6 数组及其表示	17
1.6.1 数组的引用	17
1.6.2 数组和数组元素的表示方法	18
1.6.3 数组的说明	19
1.6.4 数组的存贮结构	20
1.7 子字符串	23
1.8 FORTRAN 内部函数	23
习题	25

第2章 简单的程序设计

2.1 算术表达式	27
2.1.1 算术运算符	27
2.1.2 算术表达式的构成和求值	28
2.1.3 算术表达式的类型	29

2.2	字符表达式	31
2.3	关系表达式	31
2.3.1	算术关系表达式	32
2.3.2	字符关系表达式	33
2.4	逻辑表达式	34
2.4.1	逻辑表达式的构成规则	35
2.4.2	逻辑运算符	35
2.4.3	逻辑表达式的运算次序	36
2.5	赋值语句	37
2.5.1	算术赋值语句	38
2.5.2	字符赋值语句	39
2.5.3	逻辑赋值语句	40
2.6	简单的输入、输出语句	40
2.6.1	简单的输入语句——表控输入	41
2.6.2	简单的输出语句——表控输出	42
2.7	赋初值语句——DATA 语句	43
2.8	END 语句、STOP 语句和 PAUSE 语句	44
2.8.1	END 语句	44
2.8.2	STOP 语句	45
2.8.3	PAUSE 语句	45
2.9	参数语句(PARAMETER 语句)	46
2.10	程序举例	47
	习题	49

第3章 过程(之一)

3.1	语句函数	54
3.1.1	语句函数的定义语句	54
3.1.2	语句函数的引用与举例	55
3.2	函数子程序	59
3.2.1	函数子程序的定义	59
3.2.2	函数子程序的引用和举例	60
	习题	63

第4章 数据的输入与输出

4.1	输出语句	65
4.2	格式语句	66
4.2.1	I 型格式	66
4.2.2	F 型格式	67
4.2.3	E 型格式	68
4.2.4	G 型格式	69
4.2.5	D 型格式	70

4.2.6	L 型格式	70
4.2.7	A 型格式	70
4.2.8	H 型格式	71
4.2.9	撇号格式符	71
4.2.10	X 型格式	72
4.2.11	走纸控制	72
4.2.12	斜杠格式符	73
4.3	输入语句	75
4.3.1	输入语句	75
4.3.2	整数的输入	75
4.3.3	实数的输入	76
4.3.4	复数的输入	77
4.3.5	双精度数的输入	77
4.3.6	逻辑型数据的输入	77
4.3.7	字符型数据的输入	78
	习题	79
第 5 章 分支		
5.1	流程图的概念	82
5.2	无条件转移语句	83
5.3	条件转移语句——算术 IF 语句	85
5.4	条件转移语句——逻辑 IF 语句	89
5.5	块 IF 结构	92
5.5.1	块 IF 结构	92
5.5.2	嵌套的块 IF 结构	96
5.5.3	多分支块 IF 结构	100
	习题	102
第 6 章 循环		
6.1	循环构造	107
6.2	直到型循环	109
6.3	当型循环	113
6.4	DO 循环	118
6.5	多重循环	129
6.6	循环与数组的输入输出	138
6.6.1	用 DO 循环来输入、输出数组	139
6.6.2	数组名出现输入、输出表中	139
6.6.3	输入、输出表中使用隐含 DO 循环	141
	习题	142
第 7 章 过程(之二)		
7.1	子例行程序	160

7.1.1 子例行程序的定义	160
7.1.2 子例行程序应用举例	162
7.1.3 函数子程序的“附作用”及与子例行程序的比较	166
7.2 可调数组	167
7.3 外部语句和内部语句	170
习题	173

第8章 程序块间的数据共享

8.1 等价语句	175
8.1.1 等价语句	175
8.1.2 数组的等价关系	176
8.1.3 使用等价语句时的注意事项	177
8.1.4 等价语句的用途	178
8.1.5 应用举例	178
8.2 公用语句	179
8.2.1 无名公用区的公用语句	179
8.2.2 有关公用语句的说明	179
8.2.3 公用语句的用途	180
8.2.4 公用语句的应用举例	180
8.2.5 有名公用区的公用语句	185
8.3 数据块子程序	186
8.3.1 数据块子程序的定义	186
8.3.2 有关的注意事项	186
8.3.3 数据块子程序举例	187
习题	187

第9章 文件

9.1 文件的基本概念	192
9.1.1 磁盘文件	192
9.1.2 磁盘数据文件	193
9.2 打开文件语句和关闭文件语句	193
9.2.1 打开文件语句(OPEN)	193
9.2.2 关闭文件语句(CLOSE)	196
9.3 回退语句(BACKSPACE)和反绕语句(REWIND)	197
9.3.1 回退语句(BACKSPACE)	197
9.3.2 反绕语句(REWIND)	198
9.4 文件结束语句(END FILE)和询问语句(INQUIRE)	199
9.4.1 文件结束语句(ENDFILE)	199
9.4.2 询问语句(INQUIRE)	199
9.5 内部文件	201
9.6 磁盘数据文件举例	202

习题.....	205
第 10 章 MS FORTRAN 的使用	
10.1 MS FORTRAN 概述.....	207
10.2 MS FORTRAN V3.31 的使用	207
10.2.1 MS FORTRAN V3.31 系统的组织	207
10.2.2 MS FORTRAN V3.31 使用流程	208
10.2.3 输入、修改 FORTRAN 源程序.....	208
10.2.4 编辑、连接和运行程序	208
10.3 MS FORTRAN V5.0 的使用	210
10.3.1 MS FORTRAN V5.0 系统安装及组织	210
10.3.2 MS FORTRAN V5.0 编译器和连接器的使用	211
10.3.3 MS FORTRAN V5.0 出错处理	215
10.3.4 MS FORTRAN V5.0 使用示范	216
10.4 Microsoft FORTRAN V3.3 到 V4.0 以上版本的变更	217
10.5 常用的几种编辑工具.....	217
10.5.1 行编辑程序 EDLIN	218
10.5.2 VEDIT 编辑程序	219
10.5.3 QEDIT 编辑程序	220
附录 I 字符—ASCII 代码—EBCDIC 代码对照表	223
附录 II FORTRAN 77 内部函数	225
参考文献	

第 1 章

FORTRAN 语言基础

1.1 计算机与程序设计语言

1.1.1 电子计算机的组成

电子计算机由硬件(硬设备)和软件(软设备)两大部分组成。

1. 计算机硬件

计算机的硬件一般由主机和外部设备两部分组成。

(1) 主机由以下三部分构成:

① 存储器

存储器用来存放程序和数据。早期的存储器主要用磁芯存储器,目前都采用半导体存储器。通常它又被称为内存或主存。对于内存常用到以下几个术语:

位(bit):位是存储信息的最小单位,每一位可存放一个二进制代码,也被称为二进制位。

字节(byte):按照国际常用的习惯,一般把 8 个二进制位称为一个字节。

字(word):通常把存储单元称为字。一个存储单元由一个或几个字节组成。一个存储单元的位数称为字长,字长通常有 8 位、16 位、24 位、32 位、64 位等,随不同的机器而异。显然,字长越大,所表示的数值范围越大。

一个存储单元(字)中存放一个数据或一条指令。

地址:为了向指定的存储单元存入信息或取出信息,给存储单元从 0 开始依次编号,存储单元的编号称为地址。如果要向某个存储单元存入或取出信息,首先应指出它的地址,然后按地址才能找到该存储单元,再将信息存入或取出。

存储器的容量:存储单元的总数称为存储器的容量。容量以 K 为计算单位($1K=2^{10}=1024$)。例如某台计算机的内存容量为 32K,表示内存有 $32 \times 1024 = 32768$ 个存储单元。

存储器有一个重要的特点:取出存储单元信息后,该单元原有的信息并不破坏,只有存入新的信息,原有的内容才会被代替。

② 运算器

计算机的各种运算都由运算器完成。例如加、减、乘、除以及逻辑加、逻辑乘等,运算器还能

完成某些其它动作,如数码移位、传送等。

运算器的运算速度很快,一般为每秒几十万次到几百万次,对一些大型或巨型机,运算速度每秒可达几千万次到几亿次。

③控制器

控制器是计算机的指挥中枢,计算机的各种操作都是在控制器的指挥下进行的。控制器根据事先编制好存入计算机的程序向运算器、存储器、外部设备发出控制信号,使计算机的各部分能协调地自动工作。

运算器和控制器合称为中央处理器(Central Processing Unit),简称CPU。它一般由集成电路或大规模集成电路组成,是计算机的核心部分。

(2) 外部设备

计算机的外部设备一般包括:

①外存储器

外存储器又称为“辅助存储器”(简称外存)它是内存的后备,用于存放信息。常用的外存储器有磁盘和磁带等。

与内存相比,外存价格低,容量大,但存取速度慢,因此,计算机中暂时不用的大批信息常保存在外存中,在需要的时候通过磁带机或磁盘机把信息送入到计算机中去。由于磁盘、磁带可以拆卸,可以说外存的容量是无限的。

②输入/输出设备

输入/输出设备是实现人机联系的设备。需输入到计算机中去的各种信息,如程序和数据要由输入设备送入计算机。运算过程中的中间结果和最终结果可以以各种形式输出,如显示、绘图、打印等,这要由输出设备实现。

常用的输入设备有:光电输入机、卡片读入机、磁盘机、CRT终端等。

常用的输出设备有:行式打印机、卡片或纸带穿孔输出机、自动绘图仪、电传打字机、CRT终端等。

计算机硬件各部分的联系如图 1.1 所示。

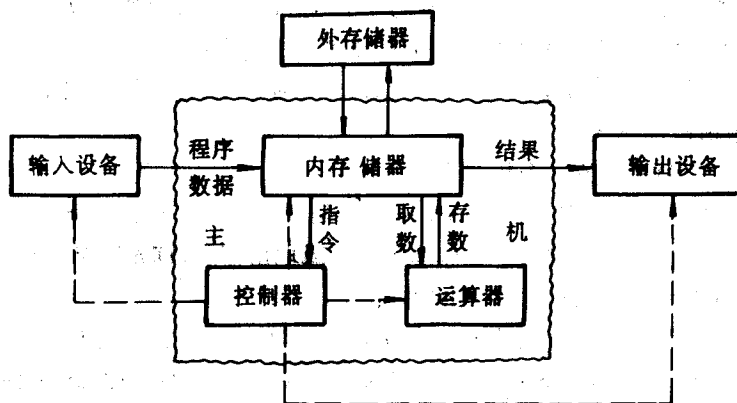


图 1.1

图 1.1 中实线表示信息的流向,虚线表示控制信号的流向。

2. 计算机软件

为了充分发挥计算机硬件的功能,功能完善的软件是不可缺少的。软硬件结合才能形成一台更理想的计算机,一个不配备任何软件的计算机称为“裸机”,作用是有限的,效率也很差。

通常可把软件分为两大类,即系统软件和应用软件。

(1) 系统软件

为了充分发挥计算机的最大效益,满足不同用户的要求,一般计算机出售时,还要向用户提供一系列软件或程序,即系统软件。系统软件又可分为三部分:

①操作系统

操作系统是系统软件的核心,它是一个庞大的管理程序,它负责对计算机的全部软硬件资源进行分配、控制和调度,最大限度地发挥计算机各部分的作用。

②语言处理系统

语言处理系统用来把用户编写的源程序翻译成计算机硬件能够理解并执行的机器指令程序。

③常用服务例行程序

这是绝大多数用户都要使用到的,能完成大多数用户所要求完成的任务的程序,如系统信息的维护、管理,软件的查错、修改等。

(2) 应用软件

应用软件是指用户为某一类应用的需要所编制的程序。随着计算机应用领域的日益扩大,应用软件发展很快,产生了种类繁多的应用软件系统。

如果说,系统软件是面向机器的,目的在于提高机器的使用效率,应用软件则是面向用户的,目的在于提高用户的工作效率。

1.1.2 程序设计和程序设计语言

人们要使计算机按照自己的意图工作,就必须将要解决的问题用计算机所允许的某种特定语言编写成程序,输入到计算机中,计算机根据程序自动工作。一般地,使用计算机解题的全过程如图 1.2 所示。

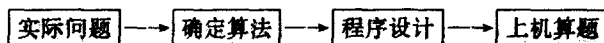


图 1.2

为计算机编写程序的过程就叫程序设计。程序设计的好坏与否将直接影响计算机工作的效率。

用于编写程序的语言称为程序设计语言。目前在程序设计中常用到下列几种语言:

①机器语言

任何一台电子计算机都有一套它自己的指令系统,称为机器指令。每条指令都产生一个相应的机器操作,例如做一个加法,从内存中取一个数,实现某种转移等,不同的机器指令的数目和表示方法也是不同的。显然指令越多,机器的功能越强。指令的集合称为计算机的指令系统,

也称为机器语言。用机器语言编写的程序称为手编程序。手编程序中的每一条指令都是用二进制表示的机器指令。

用机器语言编写的程序,计算机能直接识别和执行,执行速度快,效率高。但是,用机器语言编写程序要求了解所用机器的各条机器指令的功能和用法,甚至还要考虑到为数据和程序分配相应的存储单元。这是一件繁琐而枯燥的工作,而且手编程序难读、难懂,还易出错,不利于交流,无通用性,这些缺点严重地妨碍了计算机应用的推广。

机器语言程序的执行过程如图 1.3 所示。

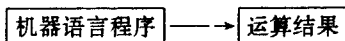


图 1.3

② 汇编语言(低级语言)

为了克服使用机器语言的种种困难,提高计算机的使用效率,人们采用了“汇编语言”。汇编语言实质上就是符号化了的机器语言。用便于记忆的符号来描述机器指令,而将分配存储单元交给计算机去完成。用汇编语言编写的程序称为源程序。

由于计算机只能识别和接受机器语言程序,而不认识用汇编语言编写的程序,因此,需要有一个“翻译员”将这些符号指令“翻译”成二进制的机器指令,计算机才能执行。这个“翻译员”实际上也是一个程序,即“汇编程序”。它由软件设计者事先编制好存入计算机,属于系统软件的一部分。翻译后的程序为目标程序。汇编语言编写的程序执行过程如图 1.4 所示。

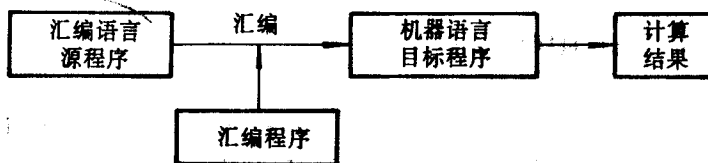


图 1.4

用汇编语言编写程序的步骤和手编差不多,所不同的是用符号代替了二进制代码,两者都要依赖于具体机器。由于不同的计算机通常具有不同的指令系统,所以它们的机器语言和汇编语言也各不相同,用汇编语言编写程序还是需要了解计算机的内部结构,因此相对机器语言来说,汇编语言并没有实质性的突破。

③ 高级语言

高级语言是一种比较接近于自然语言(主要是英语)和数学语言的计算机语言。例如 FORTRAN 语言就是一种出现最早的高级语言,它比较接近普通的数学语言,使用一些英语单词、数学符号及其它常用符号书写程序,很便于人们接受和运用。

计算机配置了高级语言后,人们不必学习机器指令,也无须了解计算机的内部结构与工作原理,便可以方便地使用计算机。由于高级语言不依赖于具体的机器和指令系统,因此,它可通

用于不同的计算机系统。用高级语言编写的程序一般称为源程序。

与机器语言、汇编语言不同,高级语言的每条语句都是由若干条机器指令组成的。计算机不能直接执行源程序,必须经过一个翻译程序把源程序翻译成计算机的指令系列(即目标程序),计算机才能执行。这个翻译程序称为编译程序。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,考虑所用计算机的指令系统,设计并编写一个编译程序系统;并且预先把它存入计算机中,使之成为计算机系统的一个组成部分。这样,该计算机就能接受和执行 FORTRAN 77 源程序了。用户并不需要了解编译程序的内部结构。

用高级语言编写的程序执行过程如图 1.5 所示。

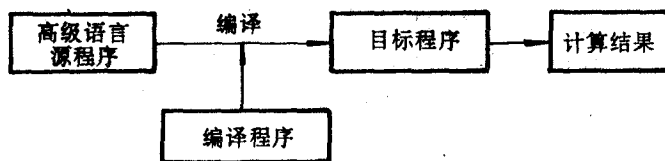


图 1.5

使用高级语言编写的源程序,增强了程序的可读性和通用性,编写速度也要快得多。

目前,国内外高级语言有数百种之多,其中一半以上是专用语言,如用于作图、模拟、计算机辅助设计、系统程序设计等,还出现了各种交互式会话语言,如 BASIC、APL 等,其后又出现了许多专门用于编写软件系统的语言,如 C 语言等。高级语言的出现及迅速发展大大推动了计算机应用的普及和推广,使人们不必深入了解计算机的内部结构和工作原理,对要解决的问题只要设计出算法,就很容易将它用高级语言表示。

1.2 FORTRAN 语言简介

FORTRAN 语言是一种适合于数值计算的程序设计语言。它是 FORMula TRANslator 的缩写,中文意思是“公式翻译”。在计算机高级语言发展史中,它是世界上使用最早,同时也是在科学、工程计算领域中使用最广泛的一种程序设计语言。

FORTRAN 语言最初是由美国 IBM 公司于 1956 年提出,以后得到了飞速发展和广泛应用,并不断推出新的版本。其中最流行的是 1958 年出现的 FORTRAN I 和 1962 年出现的 FORTRAN IV。为了对各种版本提出一个统一的标准,1966 年美国国家标准协会(American National Standard Institute,简称 ANSI),正式公布了两个美国标准文本:

- 标准 FORTRAN(ANSI X3.9-1966),大致相当 FORTRAN IV。
- 标准基本 FORTRAN(ANSI X3.10-1966),大致相当 FORTRAN I。

1972 年国际标准化组织(International Standard Organization,简称 ISO)又制定了三级国际标准:

- 完全级(一级)FORTRAN,相当于 FORTRAN IV。
- 中间级(二级)FORTRAN,介于 FORTRAN I 和 FORTRAN IV 之间。

• 基本级(三级)FORTRAN,相当于 FORTRAN I。

此后,许多厂商和用户对 FORTRAN N 进行了扩充。为了适应发展的需要,在各计算机厂商扩充 FORTRAN N 的基础上,美国标准化协会于 1976 年又重新修订了 FORTRAN (ANSIX3.9-1966),制定了新的国家标准,它包括一个全集和一个子集。这个新标准于 1978 年正式公布,定名为 FORTRAN 77。

FORTRAN 77 和 FORTRAN N 基本上是兼容的,用 FORTRAN N 编写的程序基本不用修改即可用 FORTRAN 77 编译程序进行编译和运行。与 FORTRAN N 相比,FORTRAN 77 在许多方面都有了新的发展,增加了不少新的内容。FORTRAN 77 增加了字符处理和表控输入/输出等功能,以适应数据处理、商业和企事业管理的需要。尤其是改进和增加了一些结构化语句,使 FORTRAN 77 成为较好的结构化程序设计语言。

目前,一般大、中型计算机系统配置的是 FORTRAN 77 全集,微型计算机大多配置 FORTRAN 77 子集。本书以 FORTRAN 77 全集为基础,介绍如何使用 FORTRAN 77 语言进行程序设计。凡书中出现 FORTRAN 的地方均指的是 FORTRAN 77。

1.3 FORTRAN 程序一瞥

1.3.1 FORTRAN 程序的一般结构

为使读者对 FORTRAN 程序有一个感性认识,我们通过两个简单的例子来看看 FORTRAN 程序是怎样的。

例 1 计算 $2+3$ 及 $-8.5\sqrt{2.5}$

FORTRAN 程序如下:

```
I=2+3
X=-8.5*SQRT(2.5)
WRITE(*,100)I,X
100 FORMAT(1X,I2,F10.4)
END
```

这是本书中出现的第一个 FORTRAN 程序,尽管我们还没有学习如何编写 FORTRAN 程序,但略加解释即可明白上面 FORTRAN 程序的含义。

这个程序共有 5 行,每一行是一个语句,每个语句占一行。

头两行是两个赋值语句,它俩分别把 $2+3$ 及 $-8.5\sqrt{2.5}$ 的值算出来后送到名字叫 I 和 X 的变量中去。第 2 行中的“*”是 FORTRAN 语言中的乘法运算符。SQRT()被称为一种标准函数,它的功能是对圆括号中的值求算术平方根。

后面三句中的 WRITE、FORMAT、END 都是本语句中的关键字。除了赋值语句外,每一个 FORTRAN 语句都是由一个关键字开头的,并且这些关键字的英文词意与该语句的性质是紧密相关的。

第 3 行是输出语句,关键字 WRITE 后面紧跟着的圆括号内有一个“*”和 100,这里的“*”表示系统隐含的输出设备,一般在微机上“*”表示屏幕或打印机,10 是一个格式语句的