

高等教育电教教材

程序设计语言 FORTRAN77

邓自立 编著

高等教育出版社

内 容 提 要

《程序设计语言 FORTRAN77》是配合程序设计语言 FORTRAN77 教学录象片编写的教材,也可作为独立的教材使用。全书共分十二章,介绍了计算机的一般概念和 FORTRAN77 全集语言的结构,主要内容以及编写程序的方法,并附有较多的例题与习题。该书可作为高等学校的教材,也适合于初学程序设计语言的人以及其他有关技术人员阅读参考。

高等教育电教教材

程序设计语言 FORTRAN77

邓自立 编著

高等教育出版社出版

新华书店北京发行所发行

北京印刷二厂印装

开本 787×1092 1/16 印张 22 字数 500,000

1985 年 6 月第 1 版 1985 年 8 月第 1 次印刷

印数 00,001—73,900

书号 13010·01047 定价 3.80 元

目 录

绪言	1	习题五	109
第一章 FORTRAN77 程序的基本结构	8	第六章 循环结构与循环控制语句	113
§ 1.1 一个简单的 FORTRAN77 程序	8	§ 6.1 循环结构	113
§ 1.2 FORTRAN77 程序的书写格式	10	§ 6.2 DO 语句与 DO 循环、CONTINUE 语句	115
§ 1.3 程序结构	14	§ 6.3 DO 循环的应用	121
习题一	17	§ 6.4 嵌套的循环结构	128
第二章 数据类型、常数、变量	18	习题六	137
§ 2.1 数据及其类型	18	第七章 数组与数组元素	140
§ 2.2 常数	18	§ 7.1 数组的引入	140
§ 2.3 变量	23	§ 7.2 数组	141
习题二	27	§ 7.3 数组元素	146
第三章 算术赋值语句、算术表达式	29	§ 7.4 数组的输入/输出	151
§ 3.1 算术赋值语句	29	§ 7.5 数组与数组元素的应用	158
§ 3.2 算术表达式	32	§ 7.6 变量、数组的初始化、DATA 语句	164
§ 3.3 算术赋值语句的赋值规则	41	习题七	169
§ 3.4 算术常数表达式与 PARAMETER 语句	43	第八章 FORTRAN77 过程	172
§ 3.5 函数、函数引用、内部函数	46	§ 8.1 过程	172
习题三	51	§ 8.2 语句函数	172
第四章 输入/输出初步	55	§ 8.3 外部函数及其引用	178
§ 4.1 输入/输出的几个基本概念和术语	55	§ 8.4 子程序及其引用	189
§ 4.2 表控输入	56	§ 8.5 可调数组与假定大小数组	196
§ 4.3 表控输出	60	§ 8.6 FORTRAN77 过程小结	199
§ 4.4 格式输入/输出语句的初步介绍	63	§ 8.7 程序举例	201
§ 4.5 程序举例	72	* § 8.8 辅程序的另外两个功能	205
习题四	74	习题八	208
第五章 判定结构与控制转移语句	76	第九章 程序单位间的数据通讯	211
§ 5.1 流程图	76	* § 9.1 变元的“哑实结合”	211
§ 5.2 逻辑表达式与逻辑赋值语句	77	§ 9.2 公用块	218
§ 5.3 无条件 GO TO 语句与逻辑 IF 语句	82	§ 9.3 变元法与公用块法比较	230
§ 5.4 判定结构及块 IF 语句、ELSE 语句、END IF 语句	85	习题九	234
§ 5.5 多路判定结构及 ELSE IF 语句	92	第十章 字符数据处理	236
§ 5.6 嵌套判定结构与 IF 级	97	§ 10.1 字符型数据及其类型说明	236
§ 5.7 其他控制语句	103	§ 10.2 字符子串	241
		§ 10.3 字符赋值语句与字符表达式	243

§ 10.4 含字符数据的 PARAMETER 语句与 DATA 语句.....	247	§ 11.4 格式输入/输出应用举例.....	296
§ 10.5 字符数据的输入/输出.....	248	习题十一	303
§ 10.6 字符关系表达式与字符排序序列.....	253	第十二章 输入/输出语句与文件的使用	307
§ 10.7 字符转换的内部函数	255	§ 12.1 文件的概念	307
*§ 10.8 字符哑元与实元的长度.....	258	§ 12.2 数据传输输入/输出语句.....	311
§ 10.9 字符数据处理应用举例.....	261	§ 12.3 辅助输入/输出语句.....	318
习题十.....	268	§ 12.4 文件的使用	328
第十一章 输入/输出的格式描述	271	习题十二	336
§ 11.1 格式说明与格式语句.....	271	附录 I 可执行和非执行语句一览	338
§ 11.2 编辑描述符	276	附录 II 程序单位中语句和注解行的顺序	340
§ 11.3 格式控制	292	附录 III 内部函数	341
		参考文献	344

绪 言

电子数字计算机是一种高度自动化的、用于信息处理的电子装置。它的主要特点是：能存储大量程序(运算步骤)和数据；在程序控制下高速度地处理大量的信息；并具有很高的精确度和很强的逻辑判断功能。由于电子计算机具有这些特点，它不仅在生产应用中可以节省大量人力，在同样的劳动时间里，生产出比过去多几十倍甚至几百倍的产品，而且在科学技术的许多领域中能够根据设计要求进行某些思维活动。今天，电子计算机正进入人们生活的各个方面，在人类认识自然和改造自然的活动中起着越来越突出的作用。

现代的电子计算机系统是由硬件设备和软件设备一起组成的。除了计算机硬件设备不断向大容量、高速度、多功能和微型化发展外，计算机软件也迅猛发展。正是由于软件的高速发展，计算机系统的功能才得以充分发挥，计算机的使用才能越来越方便和普及。

《程序设计语言 FORTRAN77》是计算机应用于科学计算的课程，主要目的是学习掌握 FORTRAN77 语言，学会编写解题程序。学习这门课程时，在硬件方面，不需过多了解计算机处理过程的电路原理、逻辑线路的结构和工作过程，只需要大略知道计算机系统的硬件设备的基本组成，各部件的主要功能和相互关系。在软件方面，也只需要了解计算机系统软件设备的主要作用。下面就围绕这些最基本的知识作些扼要的介绍。

一、计算机硬件设备的基本组成

计算机硬件是计算机实际的物理装置。它们由电子的、磁的、机械的或光的元件或器件组成。一个计算机系统能够正常工作，必须配备有五大部件：内存储器、控制器、运算器、输入装置和输出装置并由它们组成一个计算机硬件系统(图 1、图 3)。

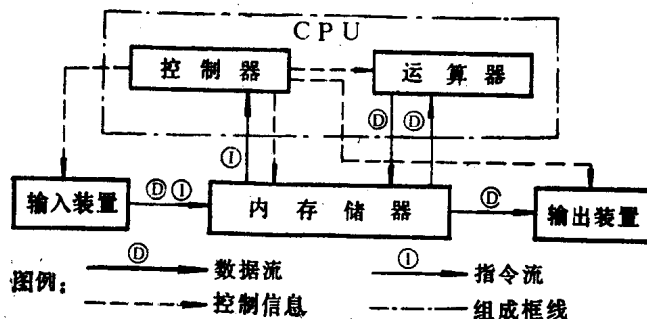


图 1 硬件设备基本组成

1. 输入装置

输入装置用来把程序和数据输入到内存储器。有各种各样的输入装置。常用的有：卡片读

人机、电传输机、终端键盘输入机、磁带机、磁盘机和光笔输入器等(图 3)。

2. 内存储器

内存储器是计算机的主要记忆装置。人们称计算机为电脑, 首先指的是内存储器能记忆大量程序和数据。存储器的记忆元件只有两种状态, 用二进制数字 0 或 1 表示。由一串 0 或 1 组成的代码称为**二进制代码**。存储一位二进制代码的信息单位称为**二进制位**(bit)。由若干个二进制位(常用八位)组成一个字节(byte), 它是用来处理一串二进制位数的单位。再由一个或多个字节组成一个字, 它是存储数据的一个单位, 也称为**存储单元**。它含有的二进制位数称为**字长**。如 IBM370 的一个字节含八位二进制位, 由四个字节组成一个字, 所以它的字长是 32 位。内存储器所能存储信息的总数称为**内存容量**。通常用所含有的存储单元(字)总数来衡量。例如, 一台计算机内存有 262144 个存储单元, 就说它的内存容量为 256K(字), 其中 $K=1024$ 。也有用字节总数来衡量的。如上例, 它的字节数为 $262144 \times 4 = 1048576$ 个字节。所以, 它的内存容量又可以说是 1024K(字节)。

计算机内存储器分成很多存储单元来保存信息。这好比一幢很大的集体公寓楼有许多小房间一样, 为了区分每一个房间, 公寓的房间都编有地址号码。同样, 内存储器的每一个存储单元也编有一个地址码。数据就是按照地址存入这些单元的。若要取出数据来参加运算也是按地址寻找并取出来参加运算的。我们要注意分清一个存储单元的地址和这个单元所存放的内容。这好比公寓房间的地址是一个号码而房间里面住的是人。我们用图 2 说明存储单元地址和单元内

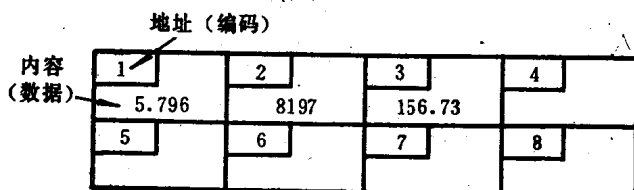


图 2

容的区别。每一长方框表示一个存储单元, 长方框的左上角有编号 1, 2, ... 表示单元地址编码(实际上是用二进制代码编码)。长方框里面的数字 5.796, 8197, 156.73, ... 是单元里面存放的内容。学习编写程序分清存储单元地址与存储单元内容之间的不同是很重要的。

内存储器的存储单元, 在存取信息时有这样的性质: “存得进, 取不尽, 挤得掉”这就是说, 向存储单元送入信息(数据或程序)总是可以存进去的(存得进); 若从单元中取出已存入的信息, 则取出后, 该单元原来存放的信息依然存在, 而且取多少次都是这样(取不尽); 而当向该单元送入新的信息后, 原存放的信息就不复存在了(挤得掉), 而被新存入的信息所取代。这个性质对我们编程很有用, 应该记住它。

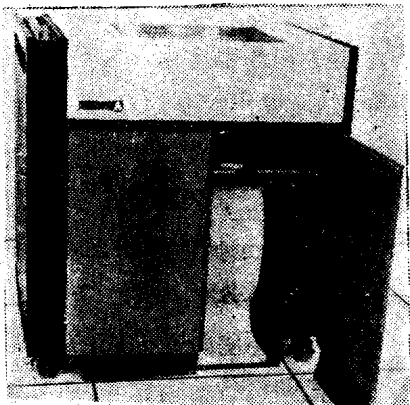
3. 运算器

运算器是计算机的主要运算装置。它在控制器控制下完成各种算术运算(加、减、乘、除)和逻辑运算。

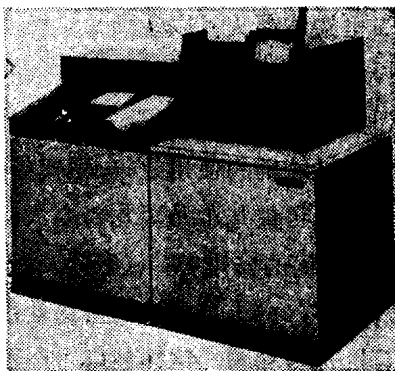
4. 输出装置



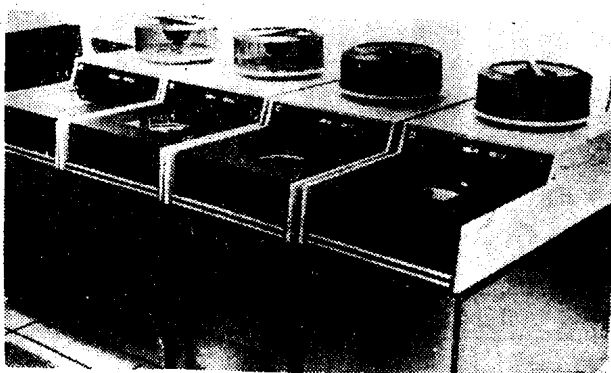
(1) 计算机硬件系统



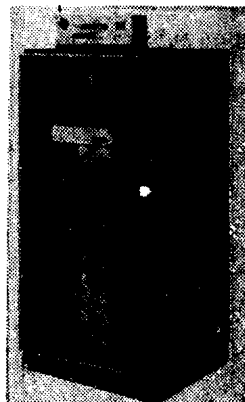
(2) 行打印机



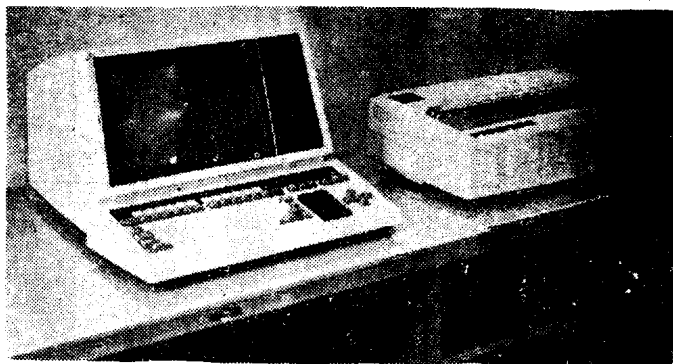
(3) 读卡机



(4) 磁盘机



(5) 磁带机



(6) 终端及针式打印机

图3 硬件设备

输出装置用来输出计算中间结果和最后结果。常用的有：终端显示器、行式打印机、热敏打印机、绘图仪、磁带机和磁盘机等(图3)。

5. 控制器

控制器是指挥全机工作的中枢装置。控制器根据存储器存入的程序指令(指令的含义后面讲)，用控制电路一条一条解释这些指令，然后向各个部件发送控制信号(图1中的虚线所示)，协调各部件的工作，完成程序指定的计算任务。

运算器和控制器组成中央处理器(简称CPU)，中央处理器和内存储器构成计算机的主机，而输入/输出装置称为外部设备(图3)。

从图1里实线箭头可以看到计算机工作的流程：程序指令和数据从输入装置输入到内存，控制器按照程序规定的顺序，逐条取出存储器中的程序指令，然后发出控制信息，让存储器和运算器之间传送数据，进行各种运算，得出结果从输出装置输出到计算机外部。

为了提高计算机的使用效率，现代大型计算机系统都配备有几十个乃至几百个分时终端(图4)。一台计算机可供几十乃至几百个人同时使用，而计算机则通过操作系统来管理计算机资源的分配，运行时间的安排等等。例如，操作系统把执行时间分成一小片一小片，轮流执行每个终端上运行的程序。由于计算机的快速处理，使每个用户根本感觉不到时间的间隔，倒像他自己独占整台计算机进行算题。这样他就可以不慌不忙地输入、调试和运行他的程序，这种使用计算机的方法称为分时系统。分时系统使计算机使用更方便，为普及计算机的使用，充分发挥计算机快速处理的优势创造了条件。

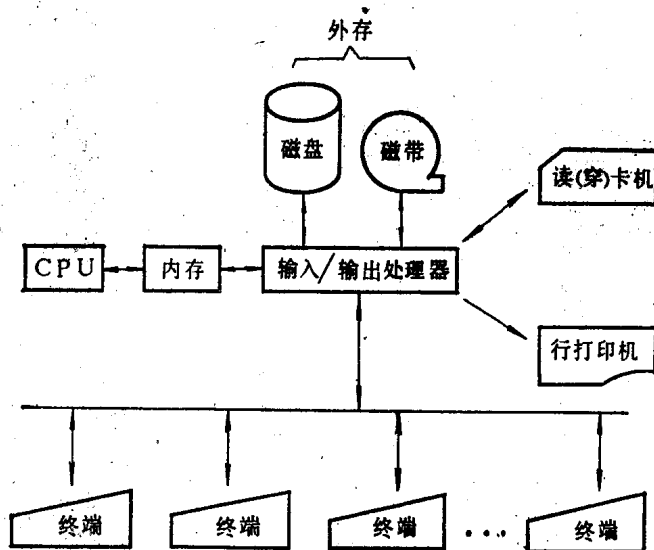


图 4

二、从机器语言到高级语言

上面已讲到，用计算机解算题目，就要通过某种方式告诉计算机要进行的步骤(程序)和所用的数据。就是说，人要与计算机进行通讯。通讯就涉及用什么语言。使用人类通常的语言文字，

计算机是不懂的。计算机只能识别机器语言。为了说明这个问题，下面讲一下关于“存储程序”原理。我们知道，计算机所以能自动连续运算，主要采用了“存储程序”原理。这个原理简单地就是在解算题目时，把复杂的解题过程分解成许多细小的步骤来完成。比如，算术运算分解成加、减、乘、除运算，每一个这种细小步骤作为计算机的一个基本操作。每一个基本操作就是设计计算机硬件时规定由电子线路能完成的一个命令，称为指令。计算机能实现的全部指令合起来称为指令系统。由许多这种指令排列起来构成解题步骤，称为程序。把这个程序存入计算机并启动计算机执行。计算机就是按照这个程序一步一步执行，完成整个解算任务。早期使用计算机就是用机器指令编写程序的。指令系统是计算机唯一能辨别和执行的语言，所以称为机器语言。而用指令编写的程序叫机器语言程序，因为都是手工编写的，也叫手编程序。机器指令是用二进制码书写的，通常一条指令包含操作码（即操作类型如：加法、减法等）和地址码（操作数的取数地址）。所以用机器指令编写程序非常麻烦，工作量很大。编出的程序形式难看、难懂、难记。而且不同的计算机，它的指令系统差别非常大。同一问题在一种计算机编出的程序无法在另一种计算机上运行。所以用机器语言编程序不是一般人能够做的工作，计算机的普及使用受到很大的限制。后来，人们采用一些符号代替操作码和操作地址来编程序。编出的程序称为假地址程序。然后，再用二进制码替换这些符号（这个过程称为“代真”），这种方法虽有改善，但仍然是使用机器语言编写程序，而且在程序代真过程中工作量且易出错。后来人们想到：既然计算机能高速地完成运算和判断工作，那么像“代真”这样的重复性工作是否可以由计算机来完成呢？于是，人们设计了一些有限的容易记忆的符号来表示指令中的操作码和地址码，然后用这些符号表达的指令编写程序。程序编好后输入到计算机，由计算机去“代真”和分配存储单元。但计算机不懂所用的符号，还要用机器语言编写一个“翻译”程序，用这个翻译程序来识别符号并进行代真工作，结果成功了。所用的符号称为符号语言或汇编语言。编出来的程序称为符号程序或汇编语言程序。所使用的翻译程序称为汇编程序。注意，汇编程序实际是机器语言程序，由它把汇编语言（即符号语言）程序翻译成机器语言程序，然后再执行运算。所以归根结底，计算机还是执行机器语言程序。汇编语言的成功无疑是一大进步，但毕竟还是依赖每一台计算机的指令系统。编程序工作量大。同样地在一种计算机编出的程序也很难用到另一种计算机上。所以使用起来仍然很不方便。计算机的普及仍然有很大障碍。通过进一步研究，人们想，既然计算机可以把符号程序翻译成指令程序，那么能不能把编写程序的工作也交由计算机完成呢？这就是程序设计自动化的设想。经过努力，1957年第一次实现了。这就是最早的FORTRAN语言。用这种语言编写的程序完全不管具体计算机的指令系统，而是按照数学运算公式直接写出程序，所以称为公式翻译。例如，把一个计算公式 $f=a \times b + c$ 直接写成 $F=A * B + C$ ，然后交给计算机编成指令程序。当然为了交由计算机编程序，总要有规则，这就是FORTRAN语言的语法规则。这种面向问题的语言称为计算机高级语言（所谓高级是指相对地脱离机器语言）。用这种语言编写的程序称为源程序。很显然，计算机是不认识这种语言的。同汇编语言相类似，也得有一个“翻译”程序把用高级语言编写的源程序翻译成机器语言程序才能执行。源程序翻译成机器语言程序称为目标程序。这个翻译程序的工作比汇编程序的工作要多且复杂。它不仅进行符

号代真等工作,而且还要把编出的“很粗”的高级语言程序编成目标程序。所以这个翻译程序称为**编译程序**。于是编写高级语言程序不必考虑某种计算机的具体细节。编好后,只要在一台配有由该计算机指令系统编写的编译程序的计算机上都可以运行。面向问题的高级语言的出现,为使用计算机提供了极大的方便,也为计算机的普及推广,创造了良好的条件。随着FORTRAN语言的发展,新的高级语言如雨后春笋般出现。当前国际上流行的通用和专用语言近数百种之多。其中使用较普遍的有用于科学计算的FORTRAN语言、ALGOL60语言,适用于数据处理的COBOL语言,适用于结构程序设计的PASCAL语言等。还有适用于一般数值计算和小型事务管理的交互会话语言BASIC(它是用解释程序,逐句翻译然后执行)也十分流行。

三、计算机软件

计算机软件是相对于计算机硬件而言的,它是计算机系统的不可缺少的组成部分。计算机软件主要分成两大类,一是系统软件,另外是应用软件。系统软件是计算机系统中用户共同使用的一些系统程序。例如,操作系统、编译程序、汇编程序、数据管理程序、诊断程序等。这些程序对扩大计算机的功能、提高计算机使用效率是必不可少的。如上面已看到,没有编译程序,高级语言程序就无法运行,而编译程序在计算机里面是为每一个用户服务的。应用软件的面很广,它是适用于某方面应用的专门程序或程序包。例如,数学程序库,统计专用程序包等。

四、用高级语言算题的步骤

概括地说,用高级语言(比如FORTRAN77)算题一般步骤如下:

1. 说明问题、分析问题、收集数据。
2. 建立数学模型(数学公式或逻辑模式)设计解题算法(解题步骤)。
3. 选用合适的高级语言编写程序。
4. 上机测试程序,并用可靠的数据进行试算。
5. 正式上机算题,评价所得结果。

我们不打算深入讨论每一步过程。这里只讲一点。程序编写好以后,应细心检查,直到确信没有错误时才上机测试。上机测试是重要的一步。上机测试就是由编译程序检查是否有语法上的错误,如果有语法错误,计算机打印出错信息,经修改后再上机测试,直到没有语法错误。但没有语法错误,不等于程序已经完全正确,有些算术运算中的错误,在语法检查过程是无法发现的(如用零值去除),只有在执行过程才发现。如果较大型的题目,立刻投入运行,可能花很多时间而得不到正确结果,所以最好使用一些可靠且已知运算结果的数据进行试算,试算成功了才正式上机运行。如果出现错误,还得认真分析算法或程序中存在的问题。如此反复地进行,直到测试成功为止。

五、FORTRAN 语言发展简介

FORTRAN 最初是为求解数值计算问题而设计的一门程序设计语言。今天在科学计算的

程序设计领域中它是国际上最广泛使用的程序设计语言。

FORTRAN 是英语 FORMula TRANslation 的缩写, 中文意思是“公式翻译”。第一个 FORTRAN 发表于 1954 年, 1957 年在 IBM 704 机上实现。随后不断发展, 1958 年实现了 FORTRAN II, 接着在 1958~1963 年间许多计算机都配了 FORTRAN 的编译程序。1962 年 FORTRAN IV 出现了, FORTRAN IV 大大扩充了语言的功能。后来的发展使 FORTRAN 语言很不统一。因此, 美国国家标准协会着手制定 FORTRAN 标准并于 1966 年 3 月公布了两个美国标准文本, 美国标准 FORTRAN(即 FORTRAN66, 相当于 FORTRAN IV)和美国标准基本 FORTRAN(相当于 FORTRAN II)。

随着 FORTRAN 语言在国际上广泛流行, 国际标准化组织于 1972 年 7 月发表了完全的、中间的和基本三级 FORTRAN 标准文本。其中完全的 FORTRAN 相当于 FORTRAN IV, 基本的相当于 FORTRAN II。1970 年美国国家标准协会开始着手新的 FORTRAN 标准形式化的工作, 于 1978 年 4 月公布了《美国国家标准程序设计语言 FORTRAN ANSI X3.9—1978》作为美国国家标准。新的标准称为 FORTRAN77, 它是 FORTRAN66 的修订本。它包括一个全集和子集。它与 FORTRAN66 基本兼容, 但对 FORTRAN66 作了重大改进。引进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 FORTRAN 的数据处理能力。尤其改进了判断和控制结构, 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。目前国际上已经广泛使用 FORTRAN77 语言。国内也已在许多计算机上配有 FORTRAN77 编译程序, 相信 FORTRAN77 不久将在我国得到广泛的应用。

本书介绍的是标准 FORTRAN77 全集语言。为了叙述方便, 下面把 FORTRAN77 语言简称为 FORTRAN77, 把用 FORTRAN77 语言编写的源程序称为 FORTRAN77 程序。

第一章 FORTRAN77 程序的基本结构

FORTRAN77 语言同其他计算机高级语言一样,是用来编写程序,实现求解问题的算法(解题步骤)的。要求写成的程序能够在计算机上进行处理。所以,程序的结构和书写格式都有严格的要求。本章通过两个引例扼要地介绍 FORTRAN77 程序的基本成分和结构。其目的是使读者在学习程序的具体内容前对 FORTRAN77 程序的全貌有所了解,从而明白学习后面的每一部分内容的目的性。所以对本章的内容不必过多追求细节,有些概念和术语暂时不理解也没有关系。因为,所介绍的内容在随后各章还会逐一详细讲述。

§ 1.1 一个简单的 FORTRAN 77 程序

引例 1 设飞机旅客交运行李的收费标准规定为:每张机票交运的行李,若不超过 20 公斤不再收费。若超过 20 公斤,则超过部分按每公斤收取机票票价 0.32% 的运费。编写一个程序,当旅客的机票票价和交运行李的重量输入计算机,计算机即可算出旅客应付的运费,并打印出付款清单。

设机票票价为 c ,交运行李重量为 w ,应付运费为 b ,则运费计算公式为:

$$b = \begin{cases} 0.0032 \times c \times (w - 20) & w > 20 \\ 0 & w \leq 20 \end{cases}$$

编写 FORTRAN77 程序时,机票票价用大写字母 C 表示,交运行李重量用大写字母 W 表示,应付运费用 $BILL$ 表示。票价 C 和重量 W 可以在终端键盘输入计算机。计算机根据重量 W 是否超过 20 公斤,作出收费计算处理。最后在打印机上打印输出付款清单。编出的 FORTRAN 77 程序如下:

```
C  AN EXAMPLE OF FORTRAN77 PROGRAM
PROGRAM EXAM1
READ *, 'C, W
PRINT *, 'COST:', C, 'bbbWEIGH:', W
IF (W.GT.20.0) THEN
BILL=0.0032*C*(W-20.0)
ELSE
BILL=0
END IF
WRITE (*, 4) BILL
FORMAT (1X, 'PAYbTHEbFREIGHT:',
```

+ F7.2)

END

程序编好以后,可以通过终端键盘输入到计算机。假设某旅客机票票价为 100 元,交运行李的重量为 35.5 公斤。可在终端键盘输入下面两个数:

100.0, 35.5

程序执行后,计算机打印付款清单如下(其中第一行数字表示打印内容在一行上的字符位置.):

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7			
COST:										100.0000000										WEIGH:										35.5000000									
PAY THE FREIGHT:										4.96																													

这个程序共写了十三行。

第一行的第一个字符是字母 C, 这是英语单词 Comment 的第一个字母, 表示注解的意思。这一行就是注解行。注解的内容意思是“一个 FORTRAN 77 程序的例”。

第二行的第一个单词是 PROGRAM, 在程序中称为**语句的关键字**。它表示这一行是一个主程序语句。关键字后面的字符串“EXAM1”是这个主程序的名字。

第三行是输入语句。关键字 READ 是读入的意思。星号“*”表示由计算机指定的输入装置(比如, 终端键盘), 用计算机规定的格式(称为**表控格式**)读入数据。READ 语句的执行用表控格式读入两个实数值分别传送给变量 C 和 W。

第四行是输出语句。关键字 PRINT 是印刷输出的意思。星号“*”表示由计算机指定的输出装置(比如, 行打印机), 用计算机规定的格式(也称表控格式)输出数据。PRINT 语句的执行用表控格式输出两组共四个数: 第一组先打印字符串“COST:”(程序中称为**字符常数**)意思是“票价:”, 然后打印变量 C 的数值 100.0000000; 第二组先打印字符串“WEIGH:”, 意思是“重量:”, 然后打印变量 W 的数值, 即行李重量值 35.5000000。注意, 在第二个字符常数书写时有三个小写字母 b, 表示空白字符, 下面会介绍。

第五行是块 IF 语句。关键字 IF 是如果的意思。它是用来判断下一步程序要执行什么语句的。进行判断时, 块 IF 语句不是单独起作用的, 而是由块 IF 语句中的 IF(如果)和 THEN(则)连同后面的 ELSE(否则)语句以及 END IF 语句结合在一起, 构成 IF—THEN—ELSE(如果—则—否则)判定结构共同起作用的。关键字 IF 后面的括号括住判定条件。判定即判断决定的意思。这里判定条件是“W.GT. 20.0”即重量 W 大于 20 吗? 其中“.GT.”是大于的意思, 是一个关系运算符。整个 IF—THEN—ELSE 判定结构的动作是: 如果条件 (W>20) 成立, 则执行 THEN 之后直到 ELSE 之前的语句组(这里只有第六行一个语句); 否则(条件不成立, 即 W≤20), 执行 ELSE 之后直到 END IF 之前的语句组(这里只有第八行一个语句)。

第六行是赋值语句。它没有关键字。它的作用是: 按照收费标准计算旅客应付运费, 并把计算结果送给变量 BILL, 这里的星号“*”表示乘法运算符。

第七行是 ELSE 语句。如上所述, 它是 IF—THEN—ELSE 结构中的一个语句。当块 IF 语

句中判定条件不成立时就转移到这个语句继续执行。

第八行也是赋值语句。它把零值赋给变量 BILL, 因为当条件不成立时才执行这个语句, 此时不再收取运费。

第九行是 END IF 语句。如上所述, 它是 IF—THEN—ELSE 结构中一个语句。它指出块 IF 语句作用终止。

第十行也是输出语句。关键字 WRITE 是写出的意思。WRITE 后面跟着一对括号, 括号里面第一个字符是星号“*”也表示使用计算机指定的输出装置, 后面的整数 4 是格式语句的编号, 指出要使用这个格式语句所提供的格式来输出数值。括号后面是要输出数据的变量名 BILL, 称为输出名表。

第十一、十二行是格式语句。关键字 FORMAT 是格式的意思。格式就是数值输出时的排列方式。最左边的数字 4 是语句的编号, 称为语句标号, 供给输出语句引用时使用。注意, 这个语句写了两行。一般情况下, 一个语句只写一行。但语句内容很长, 一行写不完时(或写得太长不美观), 可以紧接着把内容写在下一行。但在下一行接写内容之前, 要先写一个标志(称为续行标志)。第十二行左端所写的字符“+”就是续行标志。

格式语句提供的格式是在一对括号内指定的。括号内首先遇到 1X 表示在新的一行打印。接着遇到 'PAYbTHEbFREIGHT:' 表示打印出文字“PAY THE FREIGHT:”, 意思是“应付运费:”。最后遇到 F7.2 表示打印一个浮点十进制数值, 共占 7 个字符位, 其中小数部分占 2 位。

程序中 WRITE 语句和格式语句配合输出的结果可参看计算机输出的第二行。

第十三行是 END 语句。关键字 END 表示终止的意思, 它表示本程序单位的结束。

从阅读这个程序, 可以得到如下印象:

第一、程序是由注解行及一系列语句组成的。而注解行及语句又是由一些基本字符形成的。

第二、写程序时, 一行分成三部分: 第一部分书写语句标号(如果有的话), 第二部分书写续行标志(如果需要的话), 第三部分书写语句的内容。

第三、程序中出现的一些运算成分有: 常数(例如, 算术常数 0, 20.0, 0.0032 及字符常数 'COST:' 等); 变量(例如, C, W, BILL), 算术运算符, —, *; 关系运算符 .GT. 等。

第四、程序中出现的语句大致分成两类: 一类是产生某种动作的, 例如读入数据的输入语句, 印出结果的输出语句, 以及进行计算的赋值语句等。这类语句称为可执行语句。另一类是没有动作的, 而只是起说明作用的语句。例如, 说明是主程序单位的 PROGRAM 语句及提供格式安排的格式语句等。此类语句称为非执行语句。

§ 1.2 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(表 1.1):

表 1.1

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, 本书用小写字母 b 来表示空白符。书写源程序时, 空白符一般不用书写的符号 b, 只要在该位置留出空白即可。只在需要时(如在格式语句或字符串中)才用符号 b 表示空白符, 以便识别。机器印刷字符串含有空白符时, 并无此符号 b 印出, 而是在该字符位置上留出空白。

怎样用这些字符书写程序, 将在下面各章详细讲述。必须指出, 字符书写要力求工整, 清晰, 注意区分容易混淆的字符。如, 字母 O 与数字 0, 字母 I 与数字 1, 字母 Z 与数字 2, 字母 B 与数字 8, 数字 3 与数字 8, 小写字母 b 与数字 6, 字母 U 与字母 V 等。

二、关键字

FORTRAN77 中除了赋值语句和语句函数语句(第八章)外, 每个语句都用一个关键字开始。例如, 在引例 1 中的 PROGRAM、FORMAT、READ、PRINT、WRITE、IF 等都是关键字。它是由一串字母组成的特殊字符序列。它主要用来给编译程序标识 FORTRAN77 的语句, 一般不作它用。全部关键字可以在附录 I 上找到。

三、符号名

FORTRAN 77 的变量、数组、函数、程序等均用一个符号名来起名字, 如引例 1 中的 C, W,

表 1.2

符 号	错 误 原 因
BB-1	不允许出现连字符
5QR	领头不是字母
Y9.5	不允许出现小数点
Q(1)	不允许出现括号
CU+1	运算符不能出现
'OUR'	领头不是字母, 撇号不能出现
temp	字母不能小写
CONDENSER	超过六个字符

BILL, EXAM1 等。符号名由 1~6 个字母和数字字符串组成,但第一个字符必须是字母。

例 1.1 下列符号名是正确的:

X, Y1, SUM, Q23, SMAX, I, J2U

USER, P23Z, TEMP, ALPHA, MIN, U1Z

CHINA, NUMBER, VOLUME, ARRAY5

例 1.2 下列符号用作符号名是错误的(表 1.2):

四、行

FORTRAN 77 程序通常按格式书写在专用的源程序纸上,如图 1.1 所示。图 1.1 上写的是引例 1 编写的计算行李运费程序。

一张源程序纸(图 1.1)可以包括若干行(行数可视纸的长短而定,无统一要求),每行必须有 80 格,每格写一个字符。每一行横向从左往右数,称为第 1 列,第 2 列, ..., 第 80 列;竖向从上往下数,称为第 1 行,第 2 行, ...。

FORTRAN 77 程序是由一系列行组成,按行的先后次序提交给编译程序处理。每行从第 1 到 72 列组成的字符序列称为程序行。它又分三个区:

标号区,指第 1 至第 5 列,用来书写语句标号。

续行区,指第 6 列,用来书写续行标志。

语句区,指第 7 至第 72 列,用来书写语句内容。

此外,从第 73 至第 80 列,可以是空白,也可以写上行的顺序号或其他说明。对这部分内容编译程序不予加工。

FORTRAN 77 程序有三种不同的行:注解行、始行、续行。

注解行:一行的第 1 列有字符 C 或 * 或全行都是空白的任意行。这行就是注解行。如果是第 1 列有字符 C 或 * 的注解行,注解的内容可以从第 2 列开始书写,直到第 72 列。可以用字符集中的任意字符组成。如果注解内容一行写不完,可以接着写下一行,但每一行第 1 列仍要有字符 C 或 *。

注解行仅提供程序的说明,方便使用者阅读和存查,对程序的编译和执行没有任何影响。所以注解行可以出现在程序单位的 END 语句之前任何地方。可以放在程序单位的第一个语句的前面,也可以出现在任一个语句的前面(有时还可以插入一个语句中间)。

始行:指每一个语句开始写的那一行,即如果语句需分行书写的话,第一行就是始行。始行的第 1 列到第 5 列的标号区一定全是空白或写上语句标号。第 6 列的续行区或者是空白,或者是数字 0。语句内容就从第 7 列开始书写。

续行:指语句延续行,即如果一个语句分多行书写的话,从第 2 行开始,第 3 行, ..., 都是续行。续行的续行区不能写空白或数字 0,应写上 FORTRAN 77 字符集的其他任何字符。本书用字符“+”作为续行标志。续行的第 1 列至第 5 列必须全是空白符。

因为语句内容可以写在一行的第 7 列至第 72 列,而空白符只在字符常数中有意义,在语句中没有意义。所以书写程序语句时,就可以留出适当的间隙。如果一行写不下,还可以利用续

