



高等学校计算机基础课程教材

FORTRAN

浩强创作室组织编写
杨微 刘新彦 编著
李盘林 主审

语言程序设计

大连理工大学出版社

312FO
147:1

TP312FO
H147:1



高等学校计算机基础课程教材

FORTRAN 语言程序设计

浩强创作室组织编写

编著 杨 微 刘新彦
主审 李盘林

大连理工大学出版社

图书在版编目(CIP)数据

FORTRAN 语言程序设计 / 杨微,刘新彦编著. — 大连:大连理工大学出版社,2000.2(2004.1 重印)

(高等学校计算机基础课程教材)

ISBN 7-5611-1669-1

I. F… II. ①杨… ②刘… 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(2000)第 03523 号

大连理工大学出版社出版

地址:大连市凌水河 邮政编码:116024

电话:0411-4708842 传真:0411-4701466 邮购:0411-4707961

E-mail: dzcb@dlutp.cn URL: http://dlutp.cn

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:14 字数:320千字

印数:6 001 ~ 7 000

2000年2月第1版

2004年1月第2次印刷

责任编辑:刘晓晶

责任校对:王 董

封面设计:孙宝福

定 价:18.00 元

序

我们即将进入 21 世纪,大家都在讨论:21 世纪需要什么样的人才?大学生应当具有什么样的素质?应当有什么样的知识结构和能力结构?怎样才能实现这些要求?

有一点是肯定的:21 世纪是信息时代。人们的工作、学习和思想都应当适应信息时代的要求。大学生的知识结构是随着社会的发展而不断变化的,决不是一成不变的。计算机是一门新兴的而又十分重要的学科。计算机的产生和发展,对所有领域的科学技术的发展起着巨大的推动作用。新世纪的知识分子应当站在新技术发展的前沿,掌握计算机应用技术,并且把它和自己从事的专业结合起来,有力地推动各领域科学技术及其应用的发展。

显然,目前许多高校所进行的计算机教育与日益增长的要求相比,差距是大的,难以满足新世纪的要求。近年来,许多高校进行了计算机基础教育的改革,在原有的基础上,提高了要求,增加了课程,更新了教学内容,改革了教学方法。许多学校按照以下三个层次设置计算机课程:

一、计算机公共基础;

二、计算机技术基础;

三、计算机应用课程。

不同学校不同专业根据不同需要开设了不同的课程。全国高校计算机基础教育研究会还提出了需要正确处理好的十个关系,即:

1. 理论与应用的关系;
2. 深度与广度的关系;
3. 当前与发展的关系;
4. 硬件与软件的关系;
5. 跟踪先进水平与教学相对稳定的关系;
6. 课内与课外的关系;
7. 课程设置与统一考试的关系;
8. 计算机课程与其他课程的关系;
9. 要求学生动手能力强与当前设备不足的关系;
10. 计算机技术迅速发展与师资现状的关系。

许多学校在以上这些方面摸索出了许多宝贵的经验。其中应当引起注意的是: 计算机应用课程的教学, 应该由教师传授为主逐步改变为学生自学为主, 由课堂讲授为主逐步改变为上机实践为主。此外, 除了规定的必修课以外, 还应设若干选修课, 以满足不同专业不同学生的需要。

为了推进高校计算机基础教学改革, 浩强创作室组织了这套“高等学校计算机基础课程教材”, 邀请高校中具有丰富教学经验的教师参加编写工作。在编写这些教材的过程中, 我们力求符合非计算机专业的特点, 而不是简单地将计算机专业的教材浓缩。非计算机专业的培养目标、学生特点、将来工作性质、教学内容、教学方法等各方面都和计算机专业有很大的不同。二十年的经验表明: 照搬计算机专业的一套是不成功的。非计算机专业的计算机教材必须有自己的体系, 根据需要精选内容, 叙述方法必须做到由浅入深通俗易懂。

编写这套丛书遵循的方针是: 内容新颖、概念清晰、实用性强、通俗易懂、层次配套。我们在确定丛书中各教材的选题时, 充分考虑到大多数高校当前和今后一个时期所需要开设的课程。各校情况不同, 有的可作为必修课, 有的可作为选修课, 也有的可以暂时不学。随着计算机科学技术的发展和教学改革的深入, 今后还将陆续增加新的教材。各本书的内容也将不断修改补充, 以跟上计算机技术的发展。

本丛书由浩强创作室策划并组织编写。参加本丛书策划、讨论和编写的有: 谭浩强、李盘林、刘晓晶、朱桂兰、薛淑斌等。

希望本丛书的出版能推动高校计算机基础教育的发展。

全国高校计算机基础教育研究会理事长

谭浩强

1999.1

前 言

在当今信息化社会中,计算机技术飞速发展,计算机应用已渗入到各个领域。因此对高等学校所培养学生的知识结构有了新的更高的要求,同样教学模式也在不断地改革,已经由传统的黑板教学转化为多媒体教学,由传授知识为主转化为以培养能力为主,减少了教学学时,增加了实践环节。为适应 21 世纪培养人才的需要,贯彻“精讲多练”的教学思想,我们编写了这本教材。

为了使学生在少学时、新环境下掌握 FORTRAN 语言程序设计的基本内容,根据我们多年的教学经验,较详细地归纳总结了 FORTRAN 语言基本概念、基本语句、结构化程序设计的三种基本结构、数组、子程序、数据联系语句、字符等。另外,本教材中还介绍了程序调试和常见错误分析。本书中的例题都是在 MS-FORTRAN 5.0 编译系统下运行过的。

本教材的特点是,内容精练,重在练习,突出基本理论、基本知识和基本技能的训练。语言通俗易懂,习题例题丰富,实际操作性强。它不仅可以作为 FORTRAN 语言程序设计课程的教材,也可以作为自学参考书,还可以满足参加全国、省级计算机等级考试二级 FORTRAN 程序设计的需要。

本书共 13 章,第 1 章至第 11 章由杨微执笔,第 12、13 章、附录及全部习题由刘新彦执笔(第 12 章例题由曲勇提供)。

杨微、刘新彦共同审定了全稿。感谢李盘林、李世廉教授审阅了书稿,并提出了很好的意见和建议。

由于时间仓促和作者水平有限,书中难免有不妥或错误之处,恳请读者批评指正。

编 者

2000 年 1 月

目 录

前 言

第 1 章 绪 论	1
1.1 计算机语言	1
1.2 FORTRAN 语言发展概况	2
1.3 算 法	3
1.3.1 算法	3
1.3.2 程序设计基本步骤	3
习 题	4
第 2 章 FORTRAN 77 语言基础	5
2.1 FORTRAN 77 程序举例	5
2.2 FORTRAN 77 程序的构成	6
2.3 FORTRAN 77 源程序的书写格式	6
2.4 FORTRAN 77 字符集	7
2.5 常 数	7
2.5.1 整型常数	7
2.5.2 实型常数	8
2.6 变 量	9
2.6.1 变量的概念	9
2.6.2 变量的类型及说明	9
2.7 算术运算符和算术表达式	11
2.7.1 算术运算符	11
2.7.2 算术表达式	11
2.7.3 算术表达式使用规则	12
2.7.4 算术表达式的类型	12
2.8 标准函数	13
习 题	14
第 3 章 FORTRAN 77 基本语句	17
3.1 算术赋值语句	17
3.2 END 语句、STOP 语句 和 PAUSE 语句	18
3.2.1 END 语句(结束语句)	18
3.2.2 STOP 语句(停止语句)	18
3.2.3 PAUSE 语句(暂停语句)	18
3.3 GOTO 语句(无条件转移语句)	19
3.4 输出语句	19
3.4.1 表控格式输出	19

3.4.2	格式输出	20
3.4.3	其他编辑符	25
3.5	输入语句	28
3.5.1	表控输入	28
3.5.2	格式输入	30
3.6	程序举例	32
3.7	DATA 语句(赋初值语句)	33
3.8	PARAMETER 语句	34
3.9	流程图	35
	习 题	37
第 4 章	逻辑运算和选择结构	40
4.1	问题的提出	40
4.2	关系表达式	40
4.2.1	关系表达式的一般形式	40
4.2.2	关系运算符	40
4.2.3	关系表达式举例	41
4.3	逻辑运算	41
4.3.1	逻辑型常数	41
4.3.2	逻辑型变量	41
4.3.3	逻辑型表达式	42
4.3.4	逻辑型数据的输入、输出	44
4.4	逻辑 IF 语句	46
4.5	块 IF 结构	47
4.5.1	块 IF 结构的组成	47
4.5.2	块 IF 的嵌套	49
4.5.3	用 ELSE IF 语句实现多分支结构	51
4.6	程序举例	53
	习 题	54
第 5 章	循环结构	59
5.1	“当型”循环	59
5.1.1	用块 IF 和 GOTO 语句实现“当型”循环	59
5.1.2	用 WHILE 语句实现“当型”循环	60
5.2	“直到型”循环	61
5.2.1	用 UNTIL 语句实现“直到型”循环	61
5.2.2	用块 IF 结构和 GOTO 语句实现“直到型”循环	62
5.2.3	用逻辑 IF 语句实现“直到型”循环	63
5.3	DO 循环	63
5.3.1	DO 循环一般形式	64
5.3.2	DO 循环的执行过程	64
5.3.3	DO 循环次数的确定	64
5.4	循环的有关规定	66
5.5	CONTINUE 语句(继续语句)	67
5.6	循环的嵌套	68

5.7 程序举例	71
5.8 几种循环形式的比较	75
5.9 关于 GOTO 语句及其与循环的关系	77
习 题	77
第 6 章 数 组	83
6.1 数组的说明与引用	84
6.1.1 数组的说明	84
6.1.2 数组元素的引用	86
6.2 数组的存储结构和数组的输入输出	86
6.2.1 数组的存储结构	86
6.2.2 数组的输入及输出	87
6.3 程序举例	94
习 题	104
第 7 章 语句函数	112
7.1 问题的提出	112
7.2 语句函数的定义及使用规则	112
7.2.1 语句函数的定义	112
7.2.2 语句函数定义规则	113
7.3 语句函数的引用	114
7.4 程序举例	115
习 题	116
第 8 章 子 程 序	117
8.1 函数子程序	117
8.1.1 函数子程序的定义	118
8.1.2 函数子程序的调用	120
8.1.3 程序举例	120
8.2 子例程子程序	121
8.2.1 子例程子程序的定义	122
8.2.2 子例程子程序的调用	122
8.2.3 程序举例	123
8.2.4 函数子程序与子例程子程序对比	124
8.3 虚实结合	124
8.3.1 变量名作为形参	124
8.3.2 数组作为形参	125
8.3.3 过程名作为形参	131
8.3.4 字符型变量或字符型数组作为形参	133
8.4 程序举例	134
习 题	138
第 9 章 数据联系语句和数据块子程序	142
9.1 EQUIVALENCE 语句	142
9.1.1 EQUIVALENCE 语句的形式和作用	142
9.1.2 EQUIVALENCE 语句的使用规则	142
9.2 COMMON 语句	144

9.2.1	无名公用区	145
9.2.2	无名公用区使用规则及其特点	147
9.2.3	有名公用区	148
9.2.4	有名公用区使用规则及其特点	149
9.3	EQUIVALENCE 语句与 COMMON 语句的关系	150
9.4	BLOCK DATA 子程序	151
9.5	程序举例	152
习 题		154
第 10 章	双精度运算与复数运算	159
10.1	双精度运算	159
10.1.1	双精度型常量	159
10.1.2	双精度型变量	159
10.1.3	双精度型数据的输入、输出	160
10.2	复数运算	161
10.2.1	复型常量	161
10.2.2	复型变量	161
10.2.3	复型表达式	161
10.2.4	复型数据的输入和输出	162
习 题		164
第 11 章	字符处理	165
11.1	字符型常数	165
11.2	字符型变量与字符型数组	165
11.2.1	字符型变量	165
11.2.2	字符型数组	167
11.3	子字符串	167
11.4	字符变量的赋值及输入输出	168
11.4.1	字符变量的赋值	168
11.4.2	字符型数据的输入输出	169
11.5	字符型表达式	172
11.6	字符串的大小	172
11.7	字符型内部函数	173
11.8	程序举例	175
习 题		177
第 12 章	文 件	181
12.1	文件概念	181
12.1.1	记 录	182
12.1.2	文件指针	182
12.2	文件操作语句	182
12.2.1	文件打开语句 OPEN	182
12.2.2	文件关闭语句 CLOSE	184
12.2.3	文件返绕语句 REWIND	184
12.2.4	回退语句 BACKSPACE	185
12.2.5	文件输入/输出语句 (READ/ WRITE 语句)	185

12.2.6 写文件结束标志语句(END FILE 语句).....	186
12.3 程序举例.....	187
习 题.....	193
第 13 章 结构化程序设计方法及程序调试	196
13.1 结构化程序设计方法.....	196
13.2 编程风格.....	196
13.3 程序调试及常见错误分析.....	197
附 录	200
附录 1 常用 ASCII 码表.....	200
附录 2 FORTRAN 77 内部函数表	201
附录 3 程序单元中语句顺序表	203
附录 4 FORTRAN 77 可执行语句和非可执行语句	204
附录 5 MS-FORTRAN 5.0 错误信息表	205
参考文献	212

第1章 绪论

1.1 计算机语言

为了让计算机完成某项任务,必须向计算机发布一系列指令,正如人与人之间进行信息交流需要语言一样,人与计算机之间的信息交流也需要通过一种特定的语言来进行。这样的指令的集合就是一种计算机语言。为解决某一特定问题,用计算机语言所提供的一系列指令,按所规定的算法编制成程序送入计算机,使计算机按程序规定的流程进行工作,称做程序设计。因此,我们想让计算机自动完成某项任务,编制应用程序,至少应掌握一种计算机语言。

计算机语言可分为:机器语言、汇编语言、过程化语言及非过程化语言等几大类。

1. 机器语言

机器语言是计算机语言系列中最原始的语言。它的指令系统是用二进制数描述的,即每一类型的计算机都分别规定由若干个二进制数组成一条条指令。如规定以 10000000 表示“加法”操作,10010000 表示“减法”操作等。

计算机能直接识别和执行的二进制形式的指令称为机器指令,所有机器指令的集合称为机器语言。机器语言与计算机的硬件结构有关,对于不同的计算机,即使执行同一种操作,其指令也可能是不同的,也就是说,机器语言是依赖于具体计算机,是面向机器的。

用机器语言描述算法,计算机能直接识别和执行,占内存小且执行速度很快。但编制的程序难学、难记、难调试、难推广。

2. 汇编语言

针对机器语言的缺点,人们采用一些助记或表意符号来表示某些操作。例如,用 ADD 表示“加法”操作,“ADD A,B”表示将 $A+B$ 的结果再赋值给 A;用 SUB 表示“减法”操作,“SUB A,B”表示将 $A-B$ 的结果再赋值给 A。这种用助记符代替二进制指令的语言叫符号语言,又叫做汇编语言。

计算机不能直接识别和执行符号语言指令,必须转换成对应的二进制机器指令,计算机才能识别和执行。这项工作由“汇编程序”来完成,这种转换过程称为“汇编”,因此,符号语言又称做“汇编语言”。用汇编语言编写的程序称为“源程序”,汇编后得到的机器指令称为“目标程序”。

汇编语言也是面向机器的,不同类型的计算机有其特定的一套汇编语言系统,编程者需要了解计算机的硬件结构才能编制程序。汇编语言程序占内存小且执行速度快,但它的

通用性差,较难学。

由于机器语言、汇编语言都是面向机器的,因此,它们又统称为“低级语言”。

3. 过程化语言

随着计算机软硬件的不断发展,人们研制出了一种接近自然语言和数学过程的语言,称为“算法语言”。例如,用“ $A = A + B$ ”语句表示将 $A + B$ 的结果再赋值给 A ;“PRINT A, C”表示“打印 A 和 C 的值”。这种语言不必考虑机器硬件结构,只要按解题步骤写出解题过程即可。由于过程化语言接近自然语言和数学语言,因此,又称为“高级语言”。

计算机不能直接识别和执行高级语言编写的程序,需要“编译程序”将高级语言源程序编译成机器指令,机器才能识别与执行。

4. 非过程化语言

用过程化语言解决某一问题,既要描述“做什么”,又要描述“怎样做”,即要详细地描述解题的每一个步骤。例如,计算哪几个变量值,进行何种判断,循环多少次等。

为了使不熟悉计算机的人也能方便地编程,在程序中只要指明“做什么”,而不必指明“如何做”。例如,只要指明“将公司职工中基本工资大于 500 元的人的名单打印出来”,计算机就可以实现它,不必考虑计算机用何种方法实现它。这种语言称为“非过程化语言”。非过程化语言也是一种高级语言。

目前,国际上出现的高级语言有数百种,比较通用的主要有:

FORTRAN 语言 (适用于数值计算)

C 语言 (适用于编写应用软件和系统软件)

PASCAL 语言 (适用于讲解结构化程序设计)

BASIC 语言 (简单易学的会话式语言,适用于数值计算和非数值计算)

Visual Basic (面向对象的程序设计语言)

目前常见的非过程化语言有:

DBASE、FOXBASE 或 FOXPRO、SQL、ORICAL、INFORMIX 等数据库语言。

1.2 FORTRAN 语言发展概况

FORTRAN 语言是国际上第一个广泛流行的高级语言,主要适用于科学计算,具有标准程度高、通用性强、计算速度快等优点。FORTRAN 是英文 FORMula TRANslation 的缩写,意为“公式翻译”。

FORTRAN 语言第一个版本发表于 1954 年,称 FORTRAN I,1956 年实现。随后不断发展,更新了许多版本,其中较为流行的有 1958 年推出的 FORTRAN II 和 1962 年推出的 FORTRAN IV。1970 年美国国家标准化协会开始着手新的 FORTRAN 标准的工作,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,即 FORTRAN 77。FORTRAN 77 与 FORTRAN IV 基本兼容,即 FORTRAN IV 编写的程序基本不用修改,或者做很少修改,就可在 FORTRAN 77 编译系统下运行。1980 年 FORTRAN 77 被接受为国际标准,即“程序设计语言 FORTRAN ISO 1539-1980”,该标准分为全集和子集,全集基本上与 FORTRAN 77 一致,子集基本上与

FORTRAN IV一致。

我国制定的 FORTRAN 标准,基本采用了国际标准,于 1983 年 5 月公布执行,标准号为 GB3057-82。

FORTRAN 77 与 FORTRAN IV 相比,有了很大的改动,主要增加了块 IF 语句,提供了 IF-ELSE-ENDIF 形式的判断控制语句,扩充了字符处理功能,这样, FORTRAN 不仅可以用于数值计算,还可以进行非数值计算,扩大了 FORTRAN 的应用领域。

本书是以 FORTRAN 77 全集为基础进行介绍的。

1.3 算 法

1.3.1 算 法

广义地说,为解决一个问题采取的有效的方法和步骤就是算法。计算机算法可分为两大类:数值算法和非数值算法。数值运算的目的是求解问题的数值解,例如求方程的根、求函数的定积分等。非数值运算主要包括信息管理系统,如图书检索、人事工资管理等。

用计算机实现的算法应具有以下特点:

- 有穷性 一个算法所包含的操作步骤应是有限的。这包括两重含义:第一不能有死循环,第二必须在有限的时间内完成,如果一个算法要运行 1 年才能完成,显然是不合理的。

- 确定性 算法中的每一步骤都应是确定的,而不应是模棱两可的。

- 有一个或多个输出,没有输出的算法是没有意义的。

对于一个问题,可以设计出多种算法。例如,排序就有选择排序法、插入排序法等。算法的好坏一般视根据此算法编制的程序质量的好坏而定。

1.3.2 程序设计基本步骤

一般来说,程序设计应经过以下几个主要步骤:

- (1)对要解决的问题进行可行性研究,制定实施方案,搜集并整理原始数据。

- (2)依据原始数据确定正确的算法及选择合适的数据结构。

- (3)总体设计,确定程序系统的结构,画出总体程序结构图。

- (4)详细设计,根据总体设计划分的模块,分别对每个模块设计算法,并画出实现具体算法的程序流程图。

- (5)用适当的计算机语言编写程序。

- (6)将源程序输入到计算机系统中。

- (7)调试并修改程序,最后运行程序。

本书以 FORTRAN 语言为例,讲述程序设计的基本原理和方法。学习掌握程序设计方法,是学习掌握计算机技术的一项基本训练。有此基础,学习其他计算机语言的程序设计便会容易得多了。

第2章 FORTRAN 77 语言基础

2.1 FORTRAN 77 程序举例

【例 2.1】用 FORTRAN 77 语言编写程序,计算一元二次方程式 $ax^2+bx+c=0$ 的根。其中: $a=1, b=3, c=2$ 。

源程序编写如下:

```
* THE ROOTS OF THE QUADRATIC EQUATION      注释行
A=1.0                                          } 给 A,B,C 赋值
B=3.0                                          }
C=2.0                                          }
ROOT1=(-B+SQRT(B*B-4*A*C))/(2*A)             } 计算两个根
ROOT2=(-B-SQRT(B*B-4*A*C))/(2*A)             }
WRITE(*,*)ROOT1,ROOT2                        } 打印结果
END                                              } 程序结束
```

【例 2.2】用 FORTRAN 77 语言编写程序,计算 $5!, 10!$ 和 $18!$ 。

这个程序比例 2.1 复杂一些,源程序由两部分组成,即主程序和子程序。程序编写如下。主程序:

```
PROGRAM A1                                     开始部分
INTEGER C1,C2,C3                             说明部分
C1=JC(5)
C2=JC(10)
C3=JC(18)                                     } 语句部分
WRITE(*,*)C1,C2,C3
END                                             结束部分
```

子程序:

```
FUNCTION JC(N)                                开始部分
JC=1
DO 10 I=2,N                                  } 语句部分
  JC=JC*I
10 CONTINUE
END                                             结束部分
```

2.2 FORTRAN 77 程序的构成

由例 2.1、例 2.2 可以看出, FORTRAN 77 语言的结构有以下特点:

(1) 一个 FORTRAN 77 源程序由一个主程序或者一个主程序与若干个子程序组成。主程序与每一个子程序都是一个独立的程序单位, 称为一个程序模块。

(2) 每一个程序单位一般由以下几部分组成, 其顺序如下:

① 程序开始部分, 如 PROGRAM 语句、FUNCTION 语句。

② 程序说明部分, 如 INTEGER 语句。

③ 程序语句部分, 如 WRITE、CONTINUE、赋值语句等。

④ 程序结束部分, END 语句。

(3) 每一个程序单位必须有一个、也只能有一个 END 语句, 并且是一个程序单位的最后一条语句。

(4) 每一个程序单位由若干行组成。行有两种: 语句行和非语句行。语句行由 FORTRAN 语句组成, 程序中每行只能写一条语句, 一行写不下时, 可用继续行。FORTRAN 语句又分为可执行语句和非执行语句。非语句行在 FORTRAN 程序中即是注释行。在行的第一列写上字母“c”、“C”或星号“*”, 该行就为注释行, 注释行可以出现在一个程序单位的任何位置, 注释行不能有续行。注释行不是 FORTRAN 语句, 它只是对程序进行注释, 便于用户阅读程序, 不被翻译成机器指令, 也就不产生任何机器操作。

2.3 FORTRAN 77 源程序的书写格式

FORTRAN 77 源程序必须按一定的格式书写。

一个 FORTRAN 77 程序由若干行组成, 每一行有 80 列, 分为 4 个区: 标号区、续行区、语句区、注释区。

1. 标号区

1~5 列为标号区。用来书写语句标号。一条语句可以有标号也可以没有标号。如果有标号, 必须在该区内完成。FORTRAN 规定, 标号为 1~99999 的无符号正整数, 标号区内的空格不起作用。如: “123”或“1□2□3”或“1□□23”的作用相同(本书中“□”表示空格)。

使用语句标号时注意:

- 同一程序单位内, 不允许有相同的标号。
- 语句标号只是标识该语句在程序中的位置, 与语句的顺序无关。
- 语句标号的范围是 1~99999。
- 一条语句的第一行为起始行, 只有起始行才允许有标号。

2. 续行区

第 6 列为续行区。如果一条语句在一行写不下时, 可以使用继续行(简称续行), 续行符号为非零、非空格的任意字符。