

高等学校计算机课程规划教材

(第2版)



# FORTRAN 95 程序设计 与数据结构基础教程

FORTRAN 95 CHENGXUSHEJI  
YUSHUJUJIEGOUJICHUJIAOCHENG

王庆节 葛卫民 王保旗 主编

高等学校计算机课程规划教材

# FORTRAN 95 程序设计 与数据结构基础教程

(第2版)

王庆节 葛卫民 王保旗 主编



天津大学出版社

TIANJIN UNIVERSITY PRESS

## 内 容 提 要

本书以 FORTRAN 95 国际标准为依据,全面介绍了 FORTRAN 95 程序设计语言。内容新颖,重点突出,例题丰富实用,通俗易懂。书中不仅介绍了 FORTRAN 95 的基本语句,而且阐述了 FORTRAN 95 的高性能计算和基于对象的程序设计特征,同时介绍了实用的程序设计方法和 FORTRAN 95 在数值计算、检索分类及数据结构等方面的应用。每章都配有大量、多类型的习题,并对其中有一定难度的编程题给出了完整的程序。在附录中给出了大部分习题的答案及供学生自测用的样卷,并列出了 FORTRAN 95 的全部内在(标准)过程。

本书可作为高等理工科院校 FORTRAN 程序设计课程的教材,也可作为教学人员、软件开发人员的参考书。

### 图书在版编目(CIP)数据

FORTRAN 95 程序设计与数据结构基础教程/王庆节,葛卫民,王保旗主编.

—天津:天津大学出版社,2010.9

ISBN 978-7-5618-3746-7

I. ①F… II. ①王… ②葛… ③王… III. ①FORTRAN 语言—程序设计—高等学校—教材 ②数据结构—高等学校—教材 IV. ①TP312 ②TP311.12

中国版本图书馆 CIP 数据核字(2010)第 188314 号

出版发行 天津大学出版社  
出 版 人 杨欢  
地 址 天津市卫津路 92 号天津大学内(邮编:300072)  
电 话 发行部:022-27403647 邮购部:022-27402742  
网 址 www.tjup.com  
印 刷 河北省昌黎县第一印刷厂  
经 销 全国各地新华书店  
开 本 185mm × 260mm  
印 张 21.5  
字 数 538 千  
版 次 2010 年 9 月第 2 版  
印 次 2010 年 9 月第 1 次  
印 数 1—4 000  
定 价 38.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换

版权所有 侵权必究

## 再版前言

《FORTRAN 95 程序设计与数据结构基础教程》首次出版以来,被许多院校选作 FORTRAN 程序设计课程的教材,受到广大教师和同学的好评。期间,编者感到书中一些内容还不能满足教学和读者自学的需要;而且发现其中还存在一些错误。对此,有关专家、教师,包括很多同学也给我们提出了许多意见和建议,我们由衷地表示感谢。鉴于此,我们重新编写了此书。

再版的《FORTRAN 95 程序设计与数据结构基础教程》在内容上进一步明确了学习各章内容应达到的目标;修改了原书中已经发现的错误;删去了一些教学中基本不讲,初学者暂不需要了解的内容和示例,特别是对 FORTRAN 文件部分的内容,删去了“内部文件”和“NAMELIST 语句”的全部内容;为便于教学和读者自学,在第 1 章中增加了在 Compaq Visual Fortran 6 环境下的上机实习步骤;在附录中新增加了一套学生进行自测的样卷,并对前五章习题中较难的编程题给出了解答;对书中的程序重新进行了调试。

再版的《FORTRAN 95 程序设计与数据结构基础教程》仍由 11 章内容组成,但对有些章的内容重新进行了小节的划分。各章内容分别为:FORTRAN 95 简单程序设计;分支程序设计;循环程序设计;数组、指针与派生类型;FORTRAN 文件;FORTRAN 过程程序设计;FORTRAN 模块;实用程序设计方法;字符串处理算法;排序、查找算法和数据结构基础。同时,为便于教学,我们制作了与此书配套的电子教案,需要者可直接与作者联系。

本书由王庆节、葛卫民、王保旗主编。多名从事 FORTRAN 语言教学的教师参加了编写:第 1、3、4 章及附录部分由王庆节编写;第 5 章由王庆节、张福强编写;第 2、9、10 章由葛卫民编写;第 7、8 章由张福强编写;第 6、11 章由罗韬、王保旗编写;书中的全部程序由王庆节调试。

在本书重新编写过程中,采用了原书作者(赵国瑞、王保旗、李英惠、刘捐献、薛晖)所撰写的很多内容,并且还参照引用了一些已经出版的教材内容,在此谨一并表示衷心的感谢!



# 前 言

FORTRAN 95 是在 FORTRAN 77 语言基础上进行了重大扩展之后制订的新的 FORTRAN 语言标准,是当前高性能计算领域应用最广泛的程序设计语言。FORTRAN 95 吸取了其他高级程序设计语言中先进的程序设计思想,使其既支持传统的面向过程的程序设计,也支持基于对象的设计。当前,在科学计算方面,其程序运行速度是任何其他高级程序设计语言都无法比拟的。因此 FORTRAN 作为世界上第一个高级程序设计语言,至今仍保持着极强的生命力。目前许多国内外高等院校,特别是需要高速计算的理工类专业,都将 FORTRAN 语言作为程序设计课程的首选。

本书主要是为了适应大专院校理工类专业需要而编写的。全书以 FORTRAN 95 国际标准为依据,全面介绍了 FORTRAN 95。该书内容新颖,重点突出,例题丰富,通俗易懂。书中不仅详细介绍了 FORTRAN 95 的数据类型、表达式、程序控制、过程构造、文件操作等基本语句,而且阐述了 FORTRAN 95 的高性能计算和基于对象的设计特征。同时,为了提高学生的程序设计实践能力,并为今后进一步提高计算机应用水平打下良好基础,本教材还介绍了实用程序设计方法和一些基本的常用算法以及数据结构的基础知识。

为了便于组织教学和学生自学,全书共分 11 章,第 1 章到第 6 章系统介绍了 FORTRAN 95 的基础知识、FORTRAN 95 程序的基本结构和设计方法,第 7 章为 FORTRAN 95 的特色之一,即模块的概念和使用,第 8 章介绍了实用软件开发技术和方法,第 9 章和第 10 章介绍了一些最基本最常用的非数值算法,第 11 章介绍了数据结构的基础知识。另外,每章都配有大量的习题,并对部分习题给出了解答。为便于学生系统掌握、运用 FORTRAN 95 的内容,附录中列出了 FORTRAN 95 的全部内在(标准)过程。

本书的总体结构与编写思想由赵国瑞教授设计,由王保旗主编。多名从事 FORTRAN 语言教学的教师参加了编写:第 1、2 章由薛晖编写;第 3、4、5 章由王保旗编写;第 6、7、8 章和附录由赵国瑞编写;第 9、10 章由刘捐献编写;第 11 章由李英慧编写。此外,本书在编写过程中得到了天津

大学计算机科学与技术学院计算机基础教学部葛卫民主任的大力支持和帮助,本书中还借鉴了一些已公开出版的著作中的内容,在此一并表示衷心感谢!

由于作者水平所限,书中可能存在错误和不妥之处,敬请专家和读者指正。此外,由于受篇幅所限,一些教学中暂不讲授的内容均未包含在书中,读者若希望了解更全面的 FORTRAN 95 内容,可通过 <http://59.67.33.117> 下载。我们的 E-mail 地址是:[wangbq@tju.edu.cn](mailto:wangbq@tju.edu.cn)。

编者

2006年10月

# 目 录

第1章 FORTRAN 95 简单程序设计	(1)
1.1 程序设计概述	(1)
1.1.1 程序设计语言	(1)
1.1.2 程序设计过程	(2)
1.2 FORTRAN 语言简介	(4)
1.2.1 FORTRAN 语言简介	(4)
1.2.2 FORTRAN 字符集	(4)
1.2.3 FORTRAN 程序组成	(5)
1.2.4 FORTRAN 程序书写格式	(6)
1.3 FORTRAN 基本数据类型	(7)
1.4 FORTRAN 表达式	(8)
1.4.1 常量与变量	(8)
1.4.2 FORTRAN 库函数	(11)
1.4.3 算术表达式	(12)
1.5 赋值语句	(13)
1.5.1 赋值语句	(13)
1.5.2 给变量赋初值	(13)
1.6 简单的输入与输出	(14)
1.6.1 表控输入语句	(14)
1.6.2 表控输出语句	(15)
1.7 程序举例	(16)
1.8 FORTRAN 语言上机实习步骤	(19)
习题	(22)
第2章 分支程序设计	(26)
2.1 基本程序结构	(26)
2.1.1 结构化程序设计概述	(26)
2.1.2 顺序结构	(26)
2.1.3 分支(选择)结构	(27)
2.1.4 循环结构	(27)
2.2 IF 分支结构	(28)
2.2.1 逻辑表达式	(28)
2.2.2 IF...ELSE...ENDIF 结构	(30)
2.2.3 IF...ELSEIF 结构	(32)
2.3 SELECT CASE 结构	(33)
2.4 分支结构嵌套	(34)



2.5	逻辑 IF 语句与 GOTO 语句	(35)
2.5.1	逻辑 IF 语句	(35)
2.5.2	GOTO 语句	(36)
2.6	程序举例	(36)
	习题	(39)
<b>第3章</b>	<b>循环程序设计</b>	<b>(44)</b>
3.1	循环的概念	(44)
3.2	DO 循环结构	(45)
3.2.1	DO 循环结构的形式	(45)
3.2.2	计数型 DO 循环	(45)
3.2.3	DO WHILE 循环(当型循环)	(49)
3.3	循环结构的嵌套	(51)
3.4	辅助循环语句	(53)
3.4.1	CYCLE 语句	(53)
3.4.2	EXIT 语句	(53)
3.5	程序举例	(54)
	习题	(57)
<b>第4章</b>	<b>数组、指针与派生类型</b>	<b>(63)</b>
4.1	数组概述	(63)
4.1.1	数组与数组元素	(63)
4.1.2	数组说明符	(63)
4.2	数组的定义与引用	(64)
4.2.1	数组的定义	(64)
4.2.2	数组的引用	(65)
4.2.3	WHERE 与 FORALL 结构和语句	(68)
4.3	数组的输入输出	(73)
4.3.1	数组的存储规则	(73)
4.3.2	格式化输入输出	(74)
4.3.3	数组的输入输出	(78)
4.4	数组函数	(83)
4.5	动态存储管理与指针的使用	(87)
4.5.1	可变大小的数组	(87)
4.5.2	指针的概念及简单应用	(88)
4.5.3	指针数组	(90)
4.6	派生类型的定义与使用	(91)
4.6.1	派生类型的定义	(91)
4.6.2	派生类型的使用	(92)
4.7	程序举例	(94)
	习题	(97)

第5章 FORTRAN 文件 .....	(104)
5.1 FORTRAN 文件概述 .....	(104)
5.1.1 文件的基本概念 .....	(104)
5.1.2 文件的基本操作 .....	(105)
5.2 其他关于文件的语句 .....	(107)
5.2.1 BACKSPACE、ENDFILE、REWIND 语句 .....	(107)
5.2.2 INQUIRE 语句 .....	(108)
5.3 程序举例 .....	(108)
5.4 有关文件语句的细节 .....	(113)
习题 .....	(117)
第6章 FORTRAN 过程程序设计 .....	(122)
6.1 FORTRAN 过程概述 .....	(122)
6.2 外部函数子程序 .....	(124)
6.2.1 外部函数子程序的定义 .....	(124)
6.2.2 外部函数子程序的调用 .....	(127)
6.3 外部子例行子程序 .....	(129)
6.3.1 外部子例行子程序的定义 .....	(129)
6.3.2 外部子例行子程序的调用 .....	(130)
6.4 关于哑实结合的进一步讨论 .....	(134)
6.4.1 实元和哑元的对应关系 .....	(134)
6.4.2 哑实结合的方法 .....	(136)
6.4.3 入口哑元和出口哑元 .....	(138)
6.4.4 关于字符型哑元和返回不定长度字符串的函数子程序 .....	(139)
6.4.5 关于过程哑元 .....	(142)
6.4.6 指针与过程 .....	(143)
6.4.7 过程中保持前次过程调用期间值的方法 .....	(146)
6.5 关于数组哑元的进一步说明 .....	(148)
6.5.1 哑元是数组名时的实元 .....	(148)
6.5.2 哑元数组的形式 .....	(149)
6.5.3 自动数组 .....	(156)
6.6 递归过程与递归调用 .....	(156)
6.6.1 递归函数子程序 .....	(156)
6.6.2 递归子例行子程序 .....	(158)
6.7 内部过程和并行过程 .....	(160)
6.7.1 内部过程 .....	(160)
6.7.2 并行过程 .....	(161)
6.8 变量作用域与不同程序单元间的数据共享 .....	(163)
6.8.1 变量作用域 .....	(163)
6.8.2 公用语句 .....	(164)

6.8.3	数据块程序单元	(167)
6.8.4	等价语句	(168)
6.9	多文件与 INCLUDE 语句	(168)
6.10	程序举例	(169)
	习题	(172)
<b>第7章</b>	<b>FORTRAN 模块</b>	<b>(180)</b>
7.1	模块概述	(180)
7.2	模块的基本组成与使用	(180)
7.2.1	模块的基本组成形式	(180)
7.2.2	模块的共享使用	(181)
7.2.3	公有共享和私有使用	(182)
7.2.4	别名共享和部分共享	(185)
7.3	面向对象的程序设计概述	(186)
7.4	类的定义与对象的使用	(187)
7.4.1	类的定义	(187)
7.4.2	对象的使用	(189)
7.4.3	组合一类对象的使用	(189)
7.5	过程重载与运算符重载	(193)
7.5.1	过程重载	(193)
7.5.2	运算符重载	(195)
7.5.3	重载赋值号(=)	(199)
7.6	继承性和多态性	(201)
7.6.1	基类和派生类	(201)
7.6.2	多态性	(201)
	习题	(204)
<b>第8章</b>	<b>实用程序设计方法</b>	<b>(210)</b>
8.1	程序设计概述	(210)
8.1.1	程序设计的概念	(210)
8.1.2	计算机程序设计的一般过程	(211)
8.2	结构化程序设计方法	(211)
8.2.1	结构化程序设计的概念	(211)
8.2.2	结构化程序设计的规则	(216)
8.2.3	结构化编程	(216)
8.3	程序测试	(218)
8.3.1	程序测试概述	(218)
8.3.2	程序测试的黑盒法	(218)
8.3.3	程序测试的白盒法	(219)
8.4	常用算法的设计方法	(219)
8.4.1	迭代法	(219)

8.4.2	穷举法	(220)
8.4.3	递推法	(221)
8.4.4	递归法	(223)
8.4.5	回溯法	(224)
8.4.6	分治法	(226)
8.5	软件开发方法简介	(226)
	习题	(227)
<b>第9章</b>	<b>字符串处理算法</b>	<b>(231)</b>
9.1	字符型数据的基本概念	(231)
9.1.1	字符型常数与变量	(231)
9.1.2	字符表达式	(231)
9.2	常用字符函数	(232)
9.2.1	概述	(232)
9.2.2	常用字符函数的使用	(233)
9.3	常用字符串操作	(236)
9.4	字符串查找的常规算法	(240)
9.5	程序举例	(244)
	习题	(247)
<b>第10章</b>	<b>排序、查找算法</b>	<b>(252)</b>
10.1	常用排序算法	(252)
10.1.1	选择排序	(252)
10.1.2	冒泡排序	(254)
10.1.3	直接插入排序	(255)
10.2	常用查找算法	(256)
10.2.1	顺序查找	(257)
10.2.2	二分查找	(258)
10.2.3	分块查找	(259)
	习题	(260)
<b>第11章</b>	<b>数据结构基础</b>	<b>(265)</b>
11.1	数据结构概述	(265)
11.2	线性结构	(267)
11.2.1	线性表	(267)
11.2.2	栈	(272)
11.2.3	队列	(274)
11.2.4	数组	(275)
11.3	树形结构	(278)
11.3.1	树的概念	(278)
11.3.2	二叉树的定义和基本性质	(281)
11.3.3	二叉树的存储结构	(283)

11.3.4	二叉树的遍历	(284)
11.3.5	树、森林与二叉树的转换	(284)
11.3.6	二叉排序树	(285)
11.4	程序举例	(288)
	习题	(293)
附录 A	习题选答	(299)
附录 B	模拟试题及答案	(316)
附录 C	FORTRAN 95 内在过程	(323)
附录 D	ASCII 编码表	(330)
	参考文献	(332)

# 第 1 章 FORTRAN 95 简单程序设计

本章介绍 FORTRAN 95 语言基础知识。通过本章的学习,应达到如下目标:

- 了解 FORTRAN 语言的基本概念;
- 掌握 FORTRAN 95 语言的用途;
- 掌握 FORTRAN 程序的书写格式;
- 掌握 FORTRAN 95 基本数据类型的表示方法;
- 掌握 FORTRAN 表达式的书写方法以及赋值语句和简单输入与输出语句的用法;
- 初步掌握简单 FORTRAN 95 程序的编写。

## 1.1 程序设计概述

### 1.1.1 程序设计语言

简单地讲,程序设计就是编写程序。计算机是通过执行程序完成各种任务的,因此利用计算机去解决实际问题,首先就应该编写出解决问题的各种程序。用计算机语言编写程序先后经历了三个阶段:机器语言、汇编语言、高级语言。

用机器语言编写程序工作量大,十分烦琐。汇编语言是用特定的助记符号来描述指令。用汇编语言编写程序仍然烦琐枯燥,工作量大。因为机器语言和汇编语言都依赖于具体机器,所以常被称为“低级语言”。其特点是:程序难读,开发效率低。

高级语言是接近人们使用的自然语言和数学语言的编程语言。用高级语言编写程序,是因为它与机器指令不紧密相关,即不依赖于具体的计算机。这样,可以采用比较抽象、与机器无关的方法解决各类问题,编写程序时不必了解计算机的具体结构和工作原理,只需考虑所解决的问题本身。这给广大计算机用户提供了很大方便,为计算机的推广应用扫清了障碍。

用高级语言编写的程序通常称为源程序。计算机不能直接识别和执行源程序,必须把它们先“翻译”成机器指令程序(目标程序),然后再由计算机执行目标程序。这个翻译工作是由一个称为“编译程序”的软件完成的。不同的高级语言有不同的编译程序。显然,用高级语言编程解决实际问题的前提是:在计算机中已装入该高级语言的编译程序。用户采用高级语言编程上机的过程如图 1-1 所示。其中“装配连接”是由装配连接程序完成的,任务是将若干个目标程序(包括库函数)连接在一起,并执行重定位操作及自动的库文件检索,然后形成可执行的程序。连接好的程序保存在外存,随时可装入主存中运行。

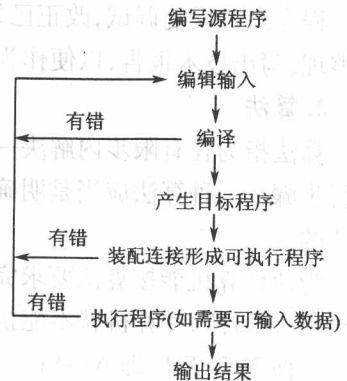


图 1-1 采用高级语言编程上机的过程

## 1.1.2 程序设计过程

### 1. 程序设计步骤

一项程序设计任务从提出到完成一般要经过以下步骤。

#### (1) 确定所要解决的问题及最后应达到的要求

接受程序设计任务后,首先要对解决的问题有详细的了解。应该注意以下问题:要解决的问题是否明确,有无二义性;用户提供了哪些原始数据和参数,这些原始数据和参数有什么用,有无多余和遗漏的参数;用户要求得到什么结果,结果的精度要求及结果输出的形式要求等。在对任务没有详细而确切了解、任务不明确的情况下不要急于编写程序,否则会出现一些意想不到的问题(如缺少参数、误差不符合要求、输出格式不合适等),从而返工修改,事倍功半。

#### (2) 分析问题构造模型

这一步的任务是根据实际问题确定物理模型,再用数学语言描述它,即列出解题的数学公式或方程式,也就是建立数学模型。

#### (3) 选择计算方法

对同样的数学模型,可能有不同的求解方法,如求定积分值的问题,可用梯形法、辛卜生法、矩形法等。因此,在编写程序之前应先确定用哪种方法计算。计算方法是研究数值计算近似方法的一门学科。

#### (4) 确定算法并绘制流程图

编写程序之前,应当整理好思路,设想好每一步怎样运算或处理,再用流程图画出来。流程图表示工作流程,能清楚表达人们的设计思路,减少编写程序时的错误。

#### (5) 编写程序

这一步是将流程图表示的解题步骤写成高级语言程序。

#### (6) 程序调试

所谓调试是指通过一些典型例子进行“试算”,以便找出程序中的错误并加以改正。一个程序往往需要反复试算修改,才能正确运行。

#### (7) 正式运行

程序经过反复调试,改正已发现的错误后就可以正式运行。在完成运算后还应对结果进行整理,写出技术报告,以便作为资料交流或保存。

### 2. 算法

算法指为在有限步内解决一个具体问题而规定的意义明确的步骤,是针对计算机设计的执行步骤。一种算法应当是明确且能够正确执行的,也就是说不是含义不清而使人感到无所适从的。

为使计算机能按算法要求完成所需要的动作,算法要足够详细。例如,需要计算 $5!$ ,如果只给出“求 $5!$ ”,计算机就不能执行,而应详细给出计算的每一步细节,比如给出:

- ①使 $N$ 等于 $1$ ,即 $N \leftarrow 1$ ;
- ②使 $I$ 等于 $2$ ,即 $I \leftarrow 2$ ;
- ③使 $N$ 等于 $N$ 乘以 $I$ ,即 $N \leftarrow N \times I$ ;
- ④使 $I$ 等于 $I$ 的原值加 $1$ ,即 $I \leftarrow I + 1$ ;



⑤如果  $I$  小于等于 5, 转第③步, 否则输出  $N$  后停止。

计算机能够执行这一“算法”, 因为算法中的每一步都可以用 FORTRAN 语言提供的语句来实现。

在接受任务和分析问题之后, 把解决问题的算法写出来, 然后再根据算法编写程序。总之, 一个算法应具有以下基本特征:

①有穷性, 即一个算法必须在执行有限步之后结束。

②确定性, 即算法每一步骤的顺序和内容都必须有确切的定义, 不能有模棱两可和二义性。

③有效性, 也称为可行性, 指算法的每一步都能使算法的执行者(人或机器)明确其含义并能去实现所规定的操作。

④有零个或多个输入。输入是指执行算法时需要从外界取得必要的初始数据。有的算法表面上无输入(即零个输入), 但实际上算法执行所需的初值已被嵌入在算法中了。

⑤有一个或多个输出。每个算法应至少有一个输出步骤, 告知解决问题的结果。

一个算法除了应具备以上基本特征外, 评价一个算法通常还考虑以下标准: 执行算法所需时间的长短、执行算法所需存储空间, 即所需计算机内存容量的大小。除此之外, 算法的正确性、最佳性和精确性等也是评价算法的标准。

### 3. 流程图

上述算法可以用文字表达, 也可以用流程图表示。有时用流程图表示算法更有利于编写程序, 可以减少编程过程中的错误。

最简单的流程图也称“框图”, 即用一个个特定的“框”代表某些动作, 框内写出各个步骤, 然后用带箭头的线把它们连接起来, 以表示执行的先后顺序。例如, 计算平均值的流程图如图 1-2 所示。

在画流程图时应当使图足够详细, 以便按照它可以顺利写出程序, 而不必在编写程序时临时构思。流程图说明了程序的逻辑结构, 不仅可以指导编写程序, 而且可以在调试时用来检查程序的正确性。如果流程图正确而运行结果不对, 则按照流程图逐步检查程序是很容易发现错误的。流程图还能作为程序说明书的一部分提供给别人, 以便帮助别人理解编写程序的思路和程序结构。

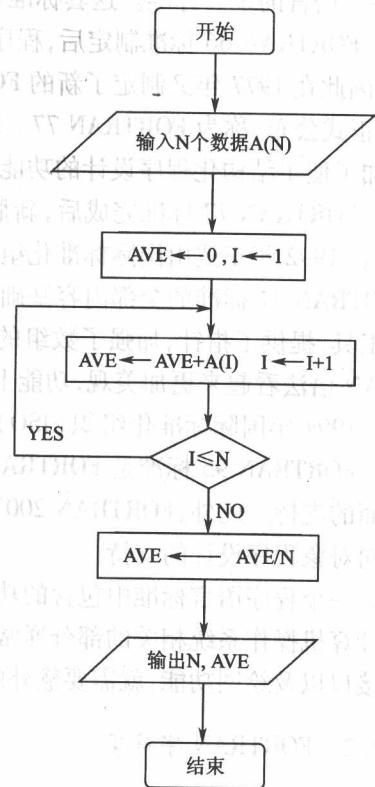


图 1-2 计算平均值的流程图



## 1.2 FORTRAN 语言简介

### 1.2.1 FORTRAN 语言简介

FORTRAN 语言是一种在国际上广泛流行的适用于科学计算的高级程序设计语言。FORTRAN 的名字来自英文的 Formula Translator,即数学公式翻译器。

FORTRAN 语言是 1954 年由美国 IBM 公司提出的,且在 1957 年开发出了第一个 FORTRAN 编译程序,这也标志着 FORTRAN 语言的诞生。此后, FORTRAN 语言成为 20 世纪 60 年代世界上最通用的计算机语言,很多软件公司推出了自己的编译程序,而且各家公司都强调自己产品的功能,都在原 FORTRAN 语言之外添加了一些自己的独门语法,这就导致了 FORTRAN 语言移植上的困难。鉴于这一原因,美国国家标准协会(ANSI)在 1966 年制定了 FORTRAN 语言的统一标准。这套标准后来被称为 FORTRAN 66。

FORTRAN 66 标准制定后,程序设计语言的理论又有所创新,且计算机硬件技术发展很快,因此在 1977 年又制定了新的 FORTRAN 语言标准,并在 1978 年由美国国家标准协会(ANSI)正式公布,称为 FORTRAN 77。FORTRAN 77 保留了 FORTRAN 66 标准的大部分内容,还添加了便于结构化程序设计的功能。

FORTRAN 77 标准完成后,新版本的修订工作也在同一时间开始。这一工作进行了 15 年,在 1992 年正式由国际标准化组织(ISO)公布,称为 FORTRAN 90。FORTRAN 90 在保留了 FORTRAN 77 标准的全部内容基础上添加了很多新内容。主要特色是加入了面向对象的概念及工具,提供了指针,加强了数组的功能,改良了原来 FORTRAN 语法中的编写格式,使 FORTRAN 语法看起来更加美观,功能上也有很大增强。

1997 年国际标准化组织(ISO)再次公布了 FORTRAN 语言的新标准,这就是 FORTRAN 95。FORTRAN 95 标准是 FORTRAN 90 的修正版,主要加强了 FORTRAN 在高性能并行运算方面的支持。此外, FORTRAN 2003 的标准正在制定中,不久将会正式发布。新标准将增强对面向对象程序设计的支持。

一个程序语言标准中包含的功能通常足以满足基本的使用要求,一些特殊的输入输出或与计算机操作系统相关的部分通常不会定义在语言标准中。例如,要使用窗口操作系统的图形接口以及绘图功能,就需要额外的程序库或扩充语法实现。

### 1.2.2 FORTRAN 字符集

“字符集”是指编写程序时所能使用的全部字符和符号。FORTRAN 语言的字符集包括的字符及符号有:

英文字母 A~Z 及 a~z(在 FORTRAN 程序中英文字母不分大小写)

数字 0~9

特殊符号 : = + - \* / ( ) , . ' ! " % & ; < > ? ¥ \_  $\square$

(表示一个不可显示的空格符)

其中的特殊符号除了作为数学运算符外,还有其他用法,后面各章将会详细介绍。