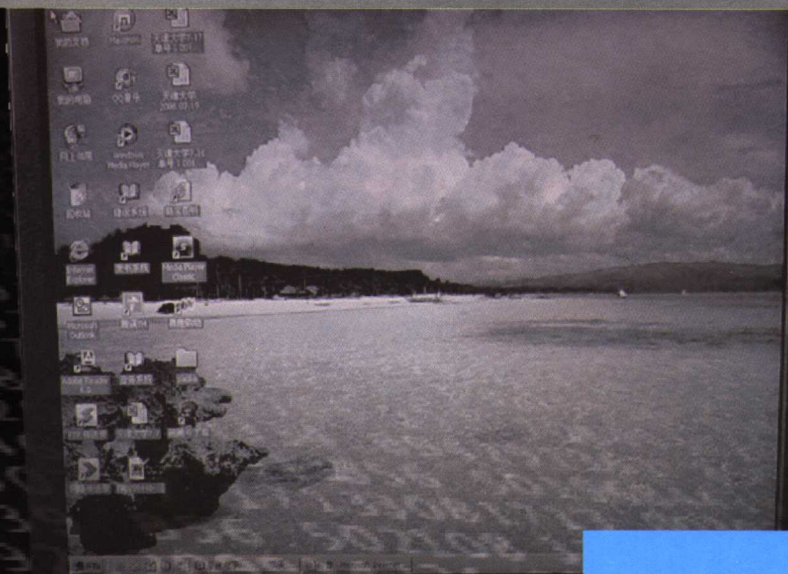


高等学校计算机课程规划教材

# FORTRAN 95程序设计 与数据结构基础教程



王保旗 主编



天津大学出版社  
TIANJIN UNIVERSITY PRESS

# FORTRAN 95 程序设计 与数据结构基础教程

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。内容新颖,重点突出,例题丰富,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、检索分类及线性表、树结构等方面的应用。每章都配有大量的单项选择题、程序填空题、编写程序题等,还对部分习题给出了解答。附录中列出了 FORTRAN 95 的全部内在(标准)过程。

本书可作为高等理工院校 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/王保旗主编.  
—天津:天津大学出版社,2007.3  
ISBN 978-7-5618-2425-2

I . F . . . II . 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 .  
TP312 TP311.12

中国版本图书馆 CIP 数据核字(2007)第 008903 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  
短信网址 发送“天大”至 916088  
印 刷 迁安万隆印刷有限公司  
经 销 全国各地新华书店  
开 本 185mm × 260mm  
印 张 20.5  
字 数 525 千  
版 次 2007 年 3 月第 1 版  
印 次 2007 年 3 月第 1 次  
印 数 1 - 2 500  
定 价 30.00 元

# 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,也支持基于对象的程序设计。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**。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程序设计特征。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 FORTAN 95 内容,可通过 <ftp://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

<b>第 1 章 FORTRAN 95 简单程序设计</b> .....	( 1 )
1.1 程序设计概述 .....	( 1 )
1.2 FORTRAN 语言简介 .....	( 4 )
1.3 FORTRAN 基本数据类型 .....	( 7 )
1.4 FORTRAN 表达式 .....	( 7 )
1.5 赋值语句 .....	( 13 )
1.6 简单的输入与输出 .....	( 14 )
1.7 程序举例 .....	( 17 )
习题 1 .....	( 20 )
<b>第 2 章 分支程序设计</b> .....	( 24 )
2.1 基本程序结构 .....	( 24 )
2.2 IF 分支结构 .....	( 26 )
2.3 SELECT CASE 结构 .....	( 32 )
2.4 分支结构的嵌套 .....	( 34 )
2.5 逻辑 IF 语句与 GOTO 语句 .....	( 36 )
2.6 程序举例 .....	( 37 )
习题 2 .....	( 40 )
<b>第 3 章 循环程序设计</b> .....	( 44 )
3.1 循环的概念 .....	( 44 )
3.2 DO 循环结构 .....	( 45 )
3.3 循环结构的嵌套 .....	( 52 )
3.4 辅助循环语句 .....	( 54 )
3.5 程序举例 .....	( 56 )
习题 3 .....	( 61 )
<b>第 4 章 数组、指针与派生类型</b> .....	( 67 )
4.1 数组概述 .....	( 67 )
4.2 数组的定义与引用 .....	( 68 )
4.3 数组的输入输出 .....	( 77 )
4.4 数组函数 .....	( 86 )
4.5 动态存储管理与指针的使用 .....	( 91 )
4.6 派生类型的定义与使用 .....	( 95 )
4.7 程序举例 .....	( 98 )
习题 4 .....	( 101 )
<b>第 5 章 FORTRAN 文件</b> .....	( 107 )
5.1 FORTRAN 文件概述 .....	( 107 )



5.2 其他关于文件的语句 .....	(111)
5.3 内部文件 .....	(115)
5.4 程序举例 .....	(116)
习题 5 .....	(120)
<b>第 6 章 FORTRAN 过程程序设计 .....</b>	<b>(125)</b>
6.1 FORTRAN 过程概述 .....	(125)
6.2 外部函数子程序 .....	(126)
6.3 外部子例行子程序 .....	(131)
6.4 哑实结合的进一步讨论 .....	(135)
6.5 数组哑元的进一步说明 .....	(148)
6.6 递归过程与递归调用 .....	(156)
6.7 内部过程和并行过程 .....	(159)
6.8 变量作用域与不同程序单元间的数据共享 .....	(163)
6.9 多文件与 INCLUDE 语句 .....	(168)
6.10 程序举例 .....	(169)
习题 6 .....	(171)
<b>第 7 章 FORTRAN 模块 .....</b>	<b>(179)</b>
7.1 模块概述 .....	(179)
7.2 模块的基本组成与使用 .....	(179)
7.3 面向对象的程序设计概述 .....	(185)
7.4 类的定义与对象的使用 .....	(186)
7.5 继承性和多态性 .....	(193)
习题 7 .....	(199)
<b>第 8 章 实用程序设计方法 .....</b>	<b>(205)</b>
8.1 程序设计概述 .....	(205)
8.2 结构化程序设计方法 .....	(206)
8.3 程序测试 .....	(213)
8.4 常用算法的设计方法 .....	(214)
8.5 软件工程概述 .....	(221)
习题 8 .....	(224)
<b>第 9 章 字符串处理算法 .....</b>	<b>(228)</b>
9.1 常用字符函数 .....	(228)
9.2 常用字符串操作 .....	(233)
9.3 字符串查找常规算法 .....	(237)
9.4 程序举例 .....	(240)
习题 9 .....	(244)
<b>第 10 章 排序、查找算法 .....</b>	<b>(250)</b>
10.1 常用排序算法 .....	(250)
10.2 常用查找算法 .....	(257)



习题 10 .....	(264)
<b>第 11 章 数据结构基础</b> .....	<b>(269)</b>
11.1 数据结构概述 .....	(269)
11.2 线性结构 .....	(272)
11.3 树形结构 .....	(283)
11.4 程序举例 .....	(293)
习题 11 .....	(297)
<b>附录 A 习题选答</b> .....	<b>(304)</b>
<b>附录 B FORTRAN 95 内在过程</b> .....	<b>(310)</b>
<b>附录 C ASCII 码表</b> .....	<b>(317)</b>
<b>参考文献</b> .....	<b>(319)</b>





# 第1章 FORTRAN 95 简单程序设计

本章介绍 FORTRAN 95 语言最基本的知识。通过本章的学习,应了解 FORTRAN 语言的基本概念,掌握 FORTRAN 95 语言的用途、程序的书写格式、FORTRAN 95 基本数据类型的表示方法、表达式的书写方法以及赋值语句和简单的输入、输出语句的用法,初步学会编写简单的 FORTRAN 95 程序。

## 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。编写程序有以下几种方法。

#### 1. 用机器语言编写程序

机器语言是指计算机指令的集合,用机器语言编写程序就是直接用机器指令组成程序。这种方式工作量大,十分繁琐,机器指令又难学难懂难记忆,而且每一种计算机都有自己特定的指令系统,互不通用,给计算机专业人员学习和使用带来极大困难。

#### 2. 用汇编语言编写程序

“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与用机器语言编写程序的方法类似,只是汇编语言的语句较机器指令易记,但仍然繁琐枯燥,工作量大,无通用性。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”。

#### 3. 用高级语言编写程序

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,通常说的“程序设计语言”就是这类语言。之所以称为“高级语言”,是因为它与机器指令不是紧密相关的,即不是依赖于具体的计算机。用高级语言编写程序,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。这给广大计算机使用者提供了很大方便,为计算机的推广使用扫清了障碍。

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(称目标程序),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。

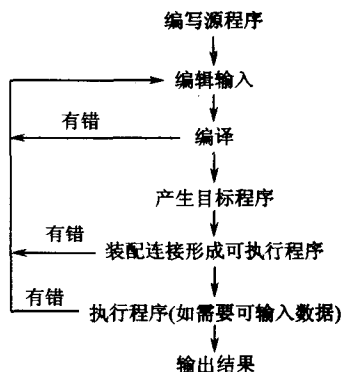


图 1-1 采用高级语言编程上机的过程



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。

## 1.1.2 程序设计过程

### 1.1.2.1 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,再把它用流程图画出来。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。

### 1.1.2.2 算法

所谓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。

以上步骤就能够执行这一“算法”。这是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,然后再根据计算方法编写程序。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人(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。

### 1.1.2.3 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。

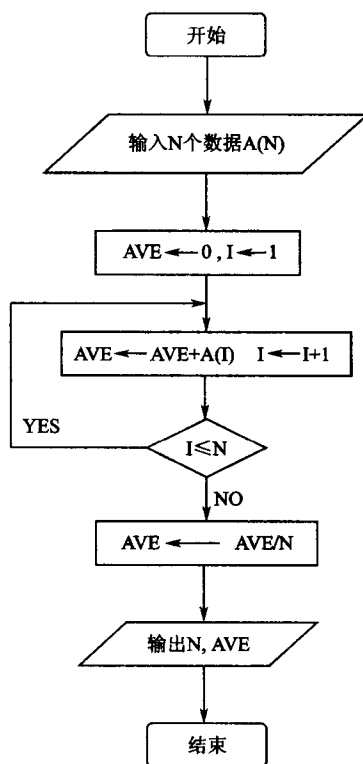


图 1-2 计算平均值的流程图



## 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, 这也就标志着 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 标准制订后, 程序设计语言的理论又有创新, 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 : = + - \* / ( ) , . ' ! " % & ; < > ? ¥ \_  $\lfloor$   
(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。



### 1.2.3 FORTRAN 程序的组成

#### 1.2.3.1 FORTRAN 程序结构

FORTRAN 程序是一种段式结构(或称为块结构)。每个 FORTRAN 程序由一个主程序段和若干个子程序段及模块组成。主程序段有且仅有一个,子程序段和模块根据需要可有零个或多个。每个程序段可以独立编写,实现不同的功能,程序运行总是从主程序段开始。每个程序段都有自己的段头语句,但主程序段的段头语句可以省略。

#### 1.2.3.2 FORTRAN 语句

每个程序段和模块由若干条 FORTRAN 语句组成,按照对 FORTRAN 程序进行编译、执行过程中所起的作用,这些语句又分为可执行语句和非执行语句。可执行语句在程序执行时使计算机产生某种特定的操作,如赋值、输入输出、控制转移等。非执行语句只是用于将某些信息(如变量的类型、数组的维数及大小、数据的输入输出格式等)通知编译程序,使编译程序在编译源程序时按这些信息的要求进行处理。程序执行时非执行语句不会使计算机产生任何操作。

FORTRAN 语句在使用时有基本的顺序要求,不能随意放置。表 1-1 给出了这些语句在 FORTRAN 程序中的顺序。各语句的位置如下:

表 1-1 FORTRAN 语句的顺序

注释行	PROGRAM、FUNCTION、SUBROUTINE、BLOCK DATA、MODULE		
	USE		
	FORMAT	IMPLICIT NONE	
		PARAMETER	IMPLICIT 及其他说明语句
		DATA	可执行语句
		CONTAINS	
		内部过程或模块过程	
	END		

①PROGRAM、FUNCTION、SUBROUTINE、BLOCK DATA、MODULE 均为程序段的段头语句,只能出现在每个程序段开始的位置上;

②END 语句是程序段的结束语句,只能出现在各程序段的最后;

③USE 语句是引用模块语句,只能出现在段头语句之后、其他语句之前;

④FORMAT 语句是格式说明语句,可以出现在 USE 语句和结束语句之间的任何位置上;

⑤IMPLICIT NONE 语句应紧挨在 USE 语句之后,在其他说明语句之前出现;

⑥PARAMETER 语句可出现在 DATA 语句和可执行语句之前、IMPLICIT NONE 语句之后的任何位置上;

⑦其他说明语句应出现在 DATA 语句和可执行语句之前;

⑧DATA 语句可以出现在说明语句之后、END 语句之前的任何位置上;

⑨注释行不是 FORTRAN 语句,仅为程序设计者提供了一种注释说明的手段,可以写在程



序的任何位置上。

#### 1.2.4 FORTRAN 程序书写格式

FORTRAN 程序的书写格式有两种:自由格式和固定格式。自由格式是目前最流行的书写格式。

##### 1.2.4.1 固定格式

固定格式是一种 FORTRAN 程序的旧式写法,采用这种写法的程序代码文件扩展名为“.FOR”或“.F”。

在“固定格式”中每一行至多写一个语句,且最多可写 72 个有效字符:第 1 个字符如果是字母 C 或 \* 号,说明此行为注释行,仅起说明作用,不会被编译;如果第 1~5 个字符是数字,则表示是该行的代号,称为语句标号,如不需要则只能是空格或全 0;如果第 6 个字符是数字“0”或空格以外的任何字符,表示该行是上一行的继续行;第 7~72 个字符是 FORTRAN 语句的书写区域;第 73 个字符及之后的字符将被忽略,有的编译程序会给出错误提示。以下是一个固定格式的 FORTRAN 程序实例。

```
C    Fixed Format
*    已知 a,b,c,求一元二次方程的根
      program main
      a = 1.0
      b = 3.0
      c = -5.6
      x1 = (-b + sqrt(b*b - 4.0*a*c))/(2*a)
      x2 = (-b - sqrt(b*b - 4.0*a*c))/
*      (2*a)
      write(*,100)x1,x2
100  format(1x,2f6.2)
      stop
      end
```

其中第 9 行的“\*”号表示该行内容“(2 \* a)”是上一行的继续。

##### 1.2.4.2 自由格式

自由格式是 FORTRAN 90 之后的新写法,取消了许多限制,没有规定每行的第几个字符有什么作用,但也有几点需要注意。这就是:①每行最多可写 132 个字符;②叹号“!”后的内容都是注释;③如果需要写语句标号,则标号可以紧挨着写在语句第 1 个字符前面;④一行字符的最后如果是符号“&”,则表示下一行是该行的继续;⑤如果一行程序代码开头是符号“&”,则其上一行的最后非空格符必须是一个 & 号,且 & 号前不能有空格,表示该行是上一行的继续。这种形式的续行允许把一个常数、变量名、关键字等分开放在两行上。在一行中可写上多个语句,但每两个语句之间必须用分号“;”隔开。用自由格式书写的 FORTRAN 程序文件的扩展名为“.F90”。以下是用自由格式书写的 FORTRAN 程序实例。

```
! Free Format
program main                                ! 已知 a,b,c,求一元二次方程的根
```



```
a = 1.0; b = 3.0; c = -5.6           ! 一行书写多个语句
x1 = (-b + sqrt(b * b - 4.0 * a * c)) / & ! 下一行是续行
      (2.0 * a)
x2 = (-b - sqrt(
&rt(b * b - 4.0 * a * c)) / 2.0 / a    ! 下一行是续行
write(*, 10) x1, x2
10  format(1x, 2f6.2)
      stop
end
```

### 1.3 FORTRAN 基本数据类型

在程序设计语言中数据有类型之分,不同类型的数据取值范围不同,处理方式不同,存储形式和书写形式也不同。FORTRAN 语言中有五种基本数据类型。

#### 1. 整型 (INTEGER)

整型表示整数的类型,分长整型和短整型。在 PC 机中长整型占 4 个字节,可保存的数值范围在  $-2\ 147\ 483\ 648 \sim 2\ 147\ 483\ 647$  之间 ( $-2^{31} \sim 2^{31} - 1$ );短整型占 2 个字节,保存的数值范围在  $-32\ 768 \sim 32\ 767$  之间 ( $-2^{15} \sim 2^{15} - 1$ )。

#### 2. 实型 (REAL)

实型数也称浮点数,分单精度和双精度两种。单精度实型数在 PC 机中占 4 个字节,有效位数为 6~7 位,可表示的最大正数为  $3.4 \times 10^{38}$ ,最小正数为  $1.18 \times 10^{-38}$ ;双精度实型数占 8 个字节,有效位数为 15~16 位,可表示的最大正数为  $1.79 \times 10^{308}$ ,最小正数为  $2.23 \times 10^{-308}$ 。

#### 3. 复型 (COMPLEX)

复型用以说明  $a + bi$  形式的复数。复数中  $a$ 、 $b$  的值为两个实型数,因此复型也分单精度和双精度复型。

#### 4. 字符型 (CHARACTER)

在程序设计语言中字符也是一种数据,这种数据称为字符型数据。字符型数据是一个字符串,其中一个字符占一个字节存储空间。

#### 5. 逻辑型 (LOGICAL)

逻辑型数据表示逻辑判断的结果,分别为“真”(TRUE)和“假”(FALSE)。在计算机中,通常以 1 或非 0 代表 TRUE,0 代表 FALSE。

### 1.4 FORTRAN 表达式

#### 1.4.1 常量与变量

##### 1.4.1.1 常量

常量指程序运行过程中其值固定不变的量。通常也将常量称为常数。不同类型的常数有不同的表示方法。



### 1. 整型常数

在整型常数中,除了正、负号外,不能包含其他任何非数字符号。例如:

128 + 256 - 128 0 -0 + 32767

都是正确的整型常数,而

128. - 123,567 100.0 - 4.0 0.

都是不正确的整型常数。

### 2. 实型常数

实型常数有多种不同的表示形式。

#### (1) 基本实常数

基本实常数形式与日常习惯写法相同,由一个正(或负)号、整数部分、小数点和小数部分组成。其中正号可以省略,并且允许没有整数部分或小数部分,但小数点是必需的。例如:

0.0 .02 314. -27.567 256.0 +0.05

都是正确的表示形式。

#### (2) 指数型实常数

指数型实常数一般用于表示一个较大或较小的数。一个指数型实常数由一个整型常数或基本实常数后跟一个字母 E, E 后跟一个正或负的整型常数组成。例如:

1.2E+0.8 -2.025E+16 0.4415E-05 1E+04 15E-06

都是正确的指数型实常数。使用指数型实常数时应注意:

- ① E 及其后的整型数表示 10 的乘幂, E 后不能是非整数;
- ② 单独的指数部分不能作为常数,如 E+02 不表示  $10^2$ , 应写成 1E+02。

#### (3) 双精度型常数

双精度型常数与上述指数型实常数表示方法基本相同,只是把指数部分的字母 E 换成字母 D。例如:

3.141592653589798D0 -1.23456789D-6 .01D-8 1.5D100

都是双精度型常数。双精度型常数中必须含有字母 D,它连同其后的整数同样表示 10 的乘幂。与上述指数型实常数不同,D 后最多可有三位整数,而 E 后最多只有两位整数。

### 3. 复型常数

一个复型常数由两个用括号括起来的整型或实型常数组成,且两个常数之间用逗号分隔。形式为(a,b),其中 a 为实部,b 为虚部。例如:

(1,2) (2.5, -3.07) (0.0, -0.455)

都是正确的复型常数,它们分别表示复数

$1 + 2i$   $2.5 - 3.07i$   $-0.455i$

复型常数的实部或虚部为零时,零不能省略。

### 4. 逻辑型常数

逻辑型常数只有两个值,即“真”值和“假”值。在 FORTRAN 语言中用 .TRUE. 表示“真值”,用 .FALSE. 表示“假”值,其中左右两个圆点不可省略。

### 5. 字符型常数

字符型常数是用英文的单引号或双引号括起来的字符串。字符串中的字符可以是 FORTRAN 字符集以外的字符(如 ASCII 码字符集中的任何可显示字符),也可以是汉字等。例如:





'FORTRAN' "123" "2003-12-5" "天津大学"

都是字符型常数。字符型常数的值就是引号之间的字符序列,在计算字符常数的长度时只计算该字符串中的字符个数,两端的定界符引号不计算在长度之内。例如'FORTRAN'的长度为7;"天津大学"的长度为8,这是因为一个汉字按两个字符处理。如果字符型常数中间有一个引号则应同时连写两个引号,计算长度时按一个字符计算,如'It's'的长度为4,而'It' 's'的长度为5。

### 1.4.1.2 变量

在程序中变量代表存放不同类型常数的内存空间。每个变量有一个名字,不同的变量代表不同的内存空间。

#### 1. 变量的命名规则

FORTRAN 语言中对变量命名有以下规则:

①变量名的第一个字符必须是字母,其余字符可以是字母、数字和下画线;

②变量名的长度最大为31个字符(与系统相关);

③变量名不能和程序名(包括主程序、子程序和模块)同名,同时尽量避免和 FORTRAN 语言中的语句名、标准函数名同名,这样在语法上虽然没问题,但容易引起混淆。

例如 A、BG5、TIME、SUM01、KK、S\_1 都是正确的变量命名,而下列都是错误或不宜的变量命名:5xy(以非字母字符开头)、A? C(含有非字母、数字字符)、F(X)(含有非字母、数字字符)、π(不是 FORTRAN 字符集中的字符)、WRITE(用 FORTRAN 保留字作为变量名是不合适的)。

#### 2. 变量类型的说明

变量表示存放数据的内存空间,数据有不同类型,占用不同大小的内存空间,说明变量的类型就是说明这个变量可存储什么类型的数据,同时告知编译程序预留多大空间以存储数据。说明变量的类型有以下几种方法。

##### (1) 用类型语句说明变量类型

用类型语句说明变量类型时的基本形式为:

类型名(kind = 字节数)::变量名表

其中变量名表是用逗号分隔的被说明的变量名;类型名用于说明变量名表中变量的类型,可以是 integer(整型)、real(实型)、complex(复型)、character(字符型)和 logical(逻辑型);“(kind = 字节数)”表示变量占用的内存空间的大小,根据实际应用的不同情况可以省略此项,此时对不同类型变量有不同的缺省长度规定。以下通过具体例子说明变量类型的说明方法。

##### 1° 整型变量的说明

说明格式如下:

```
integer(kind = 4)::X1, Y1
```

该语句说明两个整型变量 X1、Y1,其中 kind = 4 表示这两个变量分别用 4 个字节存放整数。

如果在说明整型变量时没有说明变量的长度,即没有说明变量占用的存储空间字节数,一般编译程序会默认为长整型,即一个整型变量占 4 个字节。例如可将这个语句写成

```
integer :: X1, Y1
```

效果和使用 kind = 4 说明变量长度相同。