第二章 FORTRAN 语言基础

2.1 字符集与保留字

引子 让我们先从一个简单的示例程序展示 Fortran 语言的基本结构。

简单的 Fortran 示例小程序

```
    program main              开始语句，关键字为 program，必要
        ! example             注释行，可写程序说明
        implicit none         表示不需要程序自定义变量
    REAL st1,st2,st3,stave    自行变量申明，定义实型变量，必要
        st1=8.5               变量赋值
        st2=9.0
        st3=8.7
stave=(st1+st2+st3)/3.0       计算语句，运算式
    print *,'stave=',stave    或者： write(*,*) 变量 或 '字符串' 输出到屏幕
        end                   结束语句，必要
```

2.1.1 字符集

合法字符 Fortran 允许使用的字符包括（解释语句/注释中可随意使用中文等字符）：

① 英文字母：A-Z 与 a-z（大小写在 Fortran 中不作区分，如 REAL，real，Real 是完全一致的）

② 阿拉伯数字：0-9

③ 特殊符号：空格 = + - * ( ) , . ' ' : " " ! & ; < > $? _

2.1.2 保留字

保留字 又称为关键字，是 Fortran 中具有特定意义的字符串，例如语句关键字、内部函数名。

语句关键字 有 IF, THEN（选择语句）, PROGRAM（程序开始）, INTEGER（整型）, REAL, READ, PRINT, WRITE, DO（执行循环）, END, SUBROUTINE（子程序）, FUNCTION 等。

内部函数名 系统内部固有的函数库及其函数名，例如 ABS, SIN, LOG 等。

注意 Fortran 允许保留字作为其他实体的名称（变量名、数组名、函数名、程序名等），但很不推荐。例如 program program 中的第一个为保留字，第二个为主程序单元名称。

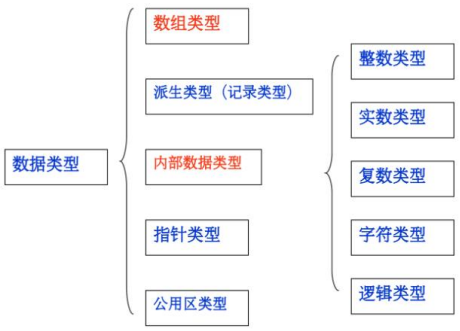
2.2 基本数据类型

2.2.1 提供的数据类型

主要类型 数组类型、派生类型（记录类型）、内部数据类型、指针类型、公用区类型。

内部数据 整型、实型、复数类型、字符类型、逻辑类型。

注意 不同类型数据具有不同特性，其处理方式有所不同，取值范围也不同，在处理数据前，必须先声明数据特性。这与 Python 的某些数据自动转换不同。



记录

目前常用的数据库（如 SQL）均为关系数据库，由表组成，第一行为字段名（表征事物基本特征的属性），其余行每行称为一条记录。一个关系数据库由很多条记录组成，读写可以以记录为单位，这就是记录类型。

2.2.2 数据类型的性质

四条性质

① 每个数据类型具有唯一确定的名称。
② 每个数据类型规定了一个取值范围（值的集合）。
③ 每个数据类型规定了其常量数据的表示方法。
④ 每个数据类型规定了一组操作。

2.3 常量与变量

2.3.1 常量

常量的概念 在程序运行过程中，其值不能被改变的量称为常量。它在程序中直接生成并直接用于计算和处理。

常量的类型 包含整型、实型、复数、字符型、逻辑型。

2.3.1.1 数值型常量

整型常量 又称为整型常数或整数，包括正数、复数和零值，例如+5、-36、0 等。

实型常量 又称为实型常数或实数，它具有两种形式：

① 小数形式（100.　.56）　② 指数形式（5.35E5 表示5.35×10^5）

当单精度实数不足以表示一个数的大小或精度时，可以使用双精度实数。只需要将指数部分的 E 改变为 D 即可，例如 6.85746304857D5 具有双精度。

复型常量 又称为复型常数或复数，例如(1.0,1.0)表示$1.0+1.0i$，(2.1,-4.5)表示$2.1-4.5i$。

2.3.1.2 字符型常量

字符型常量 又称为字符串，使用一对单引号或双引号括起来的数个非空字符串。例如'a', 'A', 'x+y'等。

注意

① 与变量名不同，字符串内部字母区分大小写。
② 字符串中间若带有撇号，需要使用转义符或双引号字符串，例如 'I''m a boy.' 或 "I'm a boy"。
③ 字符串长度包含空格，''为空字符串，' ' 长度为 1。

2.3.1.3 逻辑型常量

逻辑型常量 在 Fortran 中，逻辑常量有且仅有两个：**.TRUE.** 和 **.FALSE.** ，两侧有两个小点。

注意 对于逻辑值.TRUE.，其在存储单元字节内每位为 1，可视为整数-1；对于.FALSE.，则为 0，它们均能够直接参与到整数运算。例如：7+FALSE=7, 1+TRUE=0。

2.3.1.4 符号常量

符号常量 例如圆周率Π，重力加速度 g 等数据，关键字为 **PARAMETER**：。

定义方式	
REAL pi, x, y, z	此时为多个变量
PARAMETER(pi=3.1415926, x=1, y=2, z=3)	此时变为多个常量，其值不可再次更改

2.3.2 变量

变量的概念 变量在程序运行期间值可以变化，系统为程序中的每一个变量开辟一个内存空间，用来存放值。因此，使用变量前必须进行定义，否则数据将无处可放。

变量的命名 Fortran 中规定，变量必须以字母开头，随后可接多达 30 个字母、数字或下划线。

例如，Sum\average\student_name 是合法变量名，_total\M.D.John 都是非法变量名。

变量的类型 总体分为三类五种，即数值型变量（包含整型、实型、复型）、字符型变量与逻辑型变量。

2.3.2.1 整型变量

变量定义　下面为合法声明整型变量的语句，关键字为 **INTEGER**：

代码定义	
INTEGER(KIND=2) a,b,c,d	批量声明长度为 2 的 4 个整型变量
INTEGER(1) e	声明长度为 1 的 1 个整型变量
INTEGER f	声明长度为 4（缺省，由计算机位数决定）的 1 个整型变量
INTEGER ::g=123	声明长度为 4 的 1 个整型，且初始值为 123

注意　符号::在声明中可有可无，若有则可赋初值，否则不可赋初值。如 INTEGER f=123 是非法语句。

2.3.2.2 实型变量

变量定义　下面为合法声明实型变量的语句，关键字为 **REAL**：

代码定义	
REAL(KIND=4) a,b,c,d	声明长度为 4 的 4 个实型变量
REAL(8) e	声明长度为 8 的 1 个实型变量
REAL f	声明长度为 4（缺省）的 1 个实型变量
REAL ::g=1.23	声明长度为 4 的 1 个实型，且初始值为 1.23

注意　KIND 值为 8 的实型变量为双精度变量，可由 DOUBLE PRECISION 声明取代。

2.3.2.3 复型变量

变量定义　下面为合法声明复型变量的语句，关键字为 **COMPLEX**：

代码定义	
COMPLEX(KIND=4) a,b,c,d	声明长度为 4 的 4 个复型变量
COMPLEX(8) e	声明长度为 8 的 1 个复型变量
COMPLEX f	声明长度为 4（缺省）的 1 个复型变量
COMPLEX ::g=(3,4)	声明长度为 4 的 1 个复型，且初始值为$3+4i$

2.3.2.4 字符型变量

变量定义　下面为合法声明字符串的语句，关键字为 **CHARACTER**：

代码定义	
CHARACTER a	声明长度为 1（缺省默认值）的 1 个字符型变量
CHARACTER (8) b,c	声明长度为 8 的 2 个字符型变量
CHARACTER (len=4) e,f,g	声明长度为 4 的 3 个字符型变量
CHARACTER *6 h	声明长度为 6 的 1 个字符型变量
CHARACTER ::a='a'	声明初值为'a'的字符型变量
CHARACTER (7) ::b='Fortran',c	字符串 b 的初值为 Fortran，c 初值为 7 个空格

注意　形如声明语句 CHARACTER *7 h='student' 为非法语句（没有双冒号）。

2.3.2.5 逻辑型变量

变量定义　下面为合法声明逻辑型变量的语句，关键字为 **LOGICAL**：

代码定义	
LOGICAL(KIND=4) a	声明长度为 4 的 1 个逻辑型变量
LOGICAL (4) a	声明长度为 4 的 1 个逻辑型变量
LOGICAL a	声明长度为 4（缺省）的 1 个逻辑型变量
LOGICAL ::a=.True.	声明长度为 4 的 1 个逻辑型，且初始值为真

2.3.3 变量的声明

声明方式 在 Fortran 中，变量类型需要通过类型声明语句来定义，且有两类形式：显式声明和隐式声明。
显式声明即为上文各类型变量代码定义中的声明规则，我们来重点介绍隐式声明（隐含约定）。

I-N 规则 在程序中，凡是变量名用 I, J, K, L, M, N, i, j, k, l, m, n 开头的变量，均被默认为整型变量，以其他字母开头的变量均被默认为实型变量。例如 id 为整型，total 为实型。
这种隐式声明的方式在 Fortran 90/95 中不被提倡使用，建议在变量声明前使用 IMPLICIT 取消该规则。

IMPLICIT 这种语句可以禁止 I-N 规则或重新定义 I-N 规则，它的具体使用方式如下：

IMPLICIT 使用方法	
IMPLICIT NONE	关闭默认类型功能，任何变量都需要事先声明
IMPLICIT INTEGER(a,b,c)	a,b,c 开头的变量默认为整型
IMPLICIT REAL(m-p)	从 m 到 p 开头的变量都认为是实型

注意
① 在所有变量声明方法中，类型显式声明语句优先级最高，IMPLICIT 语句次之，I-N 规则最低。
② 类型说明语句和 IMPLICIT 语句都是非执行语句。
③ 类型说明只在本程序单位内有效。
④ IMPLICIT 命令必须置于 PROGRAM 命令的下一行，不能把它放在其他位置。

初始化 直接把数值写在声明的变量后面，使用该方法时，不能省略定义语句中间的冒号 :: 。
或者在声明后，单起一行，例如 real a； a=1

批量初始化 使用 DATA 命令批量按顺序设置：DATA a, b, c, string/1, 2.0, (1.0,2.0), "FORTRAN"/

2.4 运算符和表达式

一般概述 运算符号包括算术运算、字符运算、关系运算和逻辑运算。

2.4.1 算术运算符及其表达式

运算符 **+正号、-负号、*乘号、/除号、**乘方**，不同运算符有优先级顺序。例如：(a-b)/c**2+sin(x+y)。

注意
① 由于用 / 号作为除号，因此在写除法运算式子时应加上必要的括号。
② 乘号不能省略。如 asinx，必须写成 a*sin(x)。
③ FORTRAN 中无大、中、小括号之分，一律用小括号。
④ 乘方按先右后左原则处理。
⑤ 对单项运算符（±）相当于在它前面有一个运算量 0，如-a**2 相当于 0-a**2，而不是(-a)**2。

求值运算
① 同类型的操作数之间运算的结果仍保持原类型。特别要注意：两个整数相除的商也是整数。
例如，5/2 的值是 2 而不等于 2.5，4**(-1) 等于 0，应写为 5**(1./3.) 而不是 5**(1/3)。
② 如果参加运算的两个操作数为不同类型，则编译系统会自动将它们转换成同一类型后进行运算。转换的规律是：将低级类型转换成高级类型。类型的转换时从左向右进行的，在遇到不同类型的操作数时才进行转换。例如，1/2*1.0 等于 0，而 1./2*1 等于 0.5。

优先级 COMPLEX>REAL>INTEGER，同一类中长度长的高于长度短的。

2.4.2 关系运算符及其表达式

运算符 **.LT. <　　.LE. <=　　.EQ.==　　.NE./=　　.GT.>　　.GE.>=**

格式 表达式 1　关系运算符　表达式 2

注意
① 如果两个表达式都为算术表达式，则进行关系运算前将其转换成同一类型。
② 如果两个表达式都为字符表达式，则进行关系运算前将其转换成等长字符串，不足末尾补足空格。
③ 复数的关系运算只有两种：等于和不等于。
④ 对算术表达式进行关系运算，根据它们值的大小决定运算结果。
⑤ 对字符表达式进行关系运算，依次比较两字符串相应位置字符的 ASCII 码值大小决定运算结果。

例如	
12>34	结果为 .FALSE.
(4+5*2).LE.10	结果为 .FALSE.
(4.2,7.3).NE. (7.3,4.2)	结果为 .TRUE.
MOD(4,2).EQ.0	4 除以 2 的余数是否等于 0。结果为.TURE.
‘banana’<=’apple’	结果为 .FALSE.
‘is a pen.’<=’is a pencil.’	字符 . 的 ASCII 为 46，而 c 的 ASCII 为 99，结果为.True.