



## 第三讲 Fortran 程序设计基础 (3)

2011年9月7日



## 声明 (declaration)

### ◆ 强制类型声明

– Fortran 90 以前的版本:

- 变量在使用前不需要强制声明
- I~N规则: 以I、J、K、L、M、N开头的变量默认为整型变量

– Fortran 90 废除了隐含约定, 变量在使用前必须进行类型声明

- 好的程序员都会把程序当中所要用的变量首先进行声明, 因为如此才能明白地了解程序在运行当中, 内存的使用情形
- 变量如果不经声明就能使用, 写程序时很容易发生“人为错误”



## 声明 (declaration) (P13)

```
PROGRAM declare
  !IMPLICIT NONE
  !REAL INTEREST, COUNT
  INTEREST = 0.08
  COUNT = 0.07
  PRINT "(The interest is : ', F4.2)", INTEREST
  PRINT "(The count is : ', F4.2)", count
END
```



### ◆ Example:

implicit integer(A,B,C) ! A,B,C开头的变量都视为整数  
implicit integer(A-F,I,K) ! A到F及I,K开头的变量都视为整数

implicit real(M-P) ! M到P开头的变量都视为浮点数

implicit none ! 关闭隐含规则, 所有的变量都要事先声明

### ◆ 注意:

– IMPLICIT指令要马上接在PROGRAM指令的下一行, 不能放在其它位置

## 变量声明及其初始化

- ◆ 声明部分必须出现在执行部分之前，而不能将声明语句插在执行部分之中。
- ◆ Fortran 90:
- ◆ 数据类型 [[, 属性>::] 变量列表
  - DIMENSION、PARAMETER、TARGET、
  - POINTER、ALLOCATABLE、INTENT
  - ::可以省略，但如果在声明的同时给变量赋初值，则不能省略

## 变量声明及其初始化(P14)

```
PROGRAM initial
  IMPLICIT NONE
  INTEGER A
  REAL(8)      :: B=2.0
  COMPLEX(8)   :: C=(1.0, 2.0)
  CHARACTER(20) :: STR='FORTRAN 90'
  A = 6
  PRINT*, 'A=',A,'B=',B,'C=',C,'STR=',STR
END
```

## 进制

- ◆ 给常数赋初值时，除10进制外，还可以采用2进制、8进制，16进制：
- ◆ a=28
- ◆ A = **B**"11100"      Binary: 0, 1
- ◆ A = **O**"34"          Octal: 0~7
- ◆ A = **Z**"1C"          Z: 0~9, A, B, C, D, E

## DATA语句

- ◆ 如需赋初值的变量很多，程序就显得杂乱，可以用DATA/.../语句，按顺序给变量赋初值

```
PROGRAM initial2
  IMPLICIT NONE
  INTEGER A
  REAL(8) B
  COMPLEX(8) C
  CHARACTER(20) STR
  DATA A, B, C, STR /1, 2.0, (1.0, 2.0), 'FORTRAN 90'/
  PRINT*, 'A=',A,'B=',B,'C=',C,'STR=',STR
END
```

## 常量声明——parameter

- ◆ 程序中所用到的数据，有些是固定不变的常数，可以声明为常量，以减少错误并提高速度
- ◆ `real, parameter :: pi=3.14159265`
- ◆ 可将变量pi视为常数使用，通常目的在于简化代码、减少错误、方便修改，增加可读性
- ◆ `area = pi * radius**2` `area = 3.14159 * radius**2`
- ◆ 如需将3.14159265改为其它数字时，只需做如下修改：`pi=3.14`

## 常量声明——parameter

- ◆ 符号常量只能在声明时通过PARAMETER属性设置其值，而且只能设置一次。一旦设好，程序中就不能改变

```
program parameterExample
    implicit none
    real, parameter      ::PI=3.14159265
    print 10, sin(PI/2.)
10 format('sin( π /2)=', F6.3)
END
```

## 算术表达式

- ◆ 由运算符、常量和变量组成
- ◆ 计算数值的大小，结果为一标量
- ◆ 运算符
- ◆ + -
- ◆ \* /
- ◆ \*\*
- ◆ ( )
- ◆ 运算级相同：从左至右
- ◆ 连续的乘幂运算：从右至左

优先级增加

## 算术表达式

$2^2$  : `2**2`

$(A+B)(C+D)$  : `(A+B)*(C+D)`

$2A$  : `2*A`

$\frac{(A+B)(C+D)}{2(E+F)}$  : `((A+B)*(C+D))/(2*(E+F))`

$d = \sqrt{x^2 + y^2 + z^2}$

## Example

```
program algbic
  integer a
  a = 2**3**2
  print *, 'a=', a
end
```

## 类型转换——算术表达式

- ◆ 允许不同类型的数值型数据（整型、实型、复数型）之间进行算术运算，不允许在数值型数据与非数值型数据（字符型、逻辑型）之间进行算术运算
- ◆ 遇到不同类型之间的算术运算时，一般应显式地进行类型转换 `REAL(A)`, `INT(B)`
- ◆ 自动转换规则：低精度类型转换成高精度类型
- ◆ 数据类型的转换从左至右，当遇到不同类型时才开始转换

## Example

```
program convert
  real a
  a = 10 / 4 / 0.5
  print *, 'a = ', a
end
```

## 类型转换——赋值语句

- ◆ 表达式对变量赋值时，如果变量与表达式的结果类型相同，则直接进行赋值
  - `REAL A`
  - `A = 4.0 / 3.0`
- ◆ 如果变量与表达式的结果类型不同，则先进行表达式的类型转换，再进行赋值。
  - `REAL A`
  - `A = 4 / 3`



## Example (P17)

```
program convert2
  implicit none
  integer :: a = 3 / 2
  integer :: b = 3.0 / 2
  real    :: c = 3 / 2
  real    :: d = 3.0 / 2
  print *, 'a=', a, 'b=', b, 'c=', c, 'd=', d
end
```



## 上机作业

<http://chenbin.gr.xjtu.edu.cn>

交至:

[fortran\\_chen@foxmail.com](mailto:fortran_chen@foxmail.com)

邮件主题注明班级、姓名、学号、第几次作业  
源程序和结果拷入作业题文件，文件以班级姓名  
学号第几次作业命名。