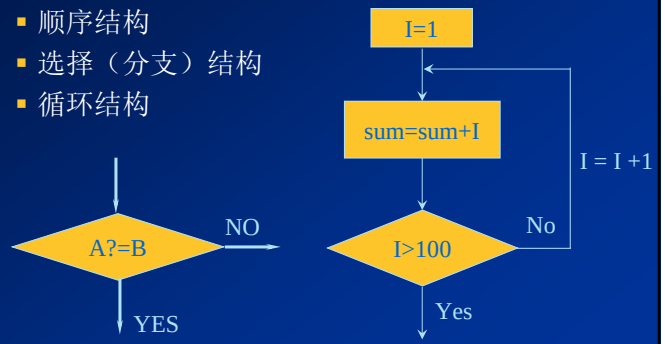
能源与动力工程学院

结构化程序设计——选择

陈　斌

结构化编程

- 结构化编程的三种基本结构：
  - 顺序结构
  - 选择（分支）结构
  - 循环结构



目录

第一节 逻辑运算

- 逻辑运算
  - 算术运算
  - 关系运算
  - 逻辑运算

运算优先级

- 关系运算符

F90	F77	功能/意义
==	.EQ.	判断是否「等于」
/=	.NE.	判断是否「不相等」
>	.GT.	判断是否「大于」
>=	.GE.	判断是否「大于或等于」
<	.LT.	判断是否「小于」
<=	.LE.	判断是否「小于或等于」

第一节 逻辑运算

- 逻辑运算符
  - 两个逻辑运算式间的运算关系

$80 \le A < 90$　　A>=80 .AND. A<90

逻辑运算符	含义		优先级
.NOT.	逻辑非	对后面的表达式结果取反	1
.AND.	逻辑与	两边的表达式都成立，整个表达式才成立	2
.OR.	逻辑或	两边的表达式只要有一个成立，整个表达式就成立	3
.EQV.	逻辑等	两边表达式的逻辑运算结果相同时，整个表达式就成立	4
.NEQV.	逻辑不等	两边表达式的逻辑运算结果不同时，整个表达式就成立	4

第一节 逻辑运算

逻辑A	逻辑B	.NOT.A	A.AND.B	A.OR.B	A.EQV.B	A.NEQV.B
T	T	F	T	T	T	F
T	F	F	F	T	F	T
F	T	T	F	T	F	T
F	F	T	F	F	T	F

第一节 逻辑运算

- 如果变量A在10~20之间时，条件成立
  (A>=10) .and. (A<=20)
- 如果变量key等于字符Y或y时，条件成立
  (key=='Y') .or. (key=='y')
- 变量A等于10时，条件不成立
  .not. (A==10)
- 变量a及b同为正数或同为负数，条件成立
  (a>0 .and. b>0) .or. (a<0 .and. b<0)

第二节 IF语句

- 行if语句：
  IF（逻辑判断式）执行语句
- 假如条件为真，就执行条件后的那条语句；否则，程序流程跳过IF语句，接着往下执行。
- 所需要执行的程序模块只能有一行程序代码

例 行IF语句

```
PROGRAM EX0401
    IMPLICIT NONE
    REAL (KIND=4) :: SPEED
    WRITE (*,*) "SPEED:" !信息提示
    READ (*,*) SPEED     !读入车速
    IF (SPEED > 100.0) WRITE (*,*) "SLOW DOWN."
END
```

第二节 IF语句

❖ 多重判断

```
if (a >= 60) print *, ”pass!”
print *, ”fail!”
```

第二节 IF语句

❖ 在F77以前配合goto语句实现流程与循环控制，容易造成程序结构混乱、降低可读性、并产生错误，所以不建议使用。

```
      if (a .GE. 60) goto 100
      print *, ”fail!”
      goto 200
  100 print *, ”pass!”
  200 stop
```

❖ 从上例可以看出，IF语句的功能极为有限，为了执行稍复杂的流程控制，不得不结合使用GOTO无条件转移语句，结果使程序代码的可读性大为降低。

第二节 IF语句

```
IF（逻辑判断式）THEN
    ……
    ……        !逻辑成立时，才会执行这里面的程序代码
    ……
END IF
```

例4-1 块IF语句

```
PROGRAM EX0401
    IMPLICIT NONE
    REAL (KIND=4) :: SPEED
    WRITE (*,*) “SPEED:” !信息提示
    READ (*,*) SPEED      !读入车速
    IF (SPEED > 100.0) THEN
       !SPEED >100 时才会执行下面这一行程序
       WRITE (*,*) “SLOW DOWN.”
    END IF
END
```

第二节 IF语句

```
IF (逻辑判断式) THEN
  ……
  ……              !逻辑成立时，执行这一段程序代码
  ……
ELSE
  ……
  ……              !逻辑不成立时，则执行这一段程序代码
  ……
END IF
```

第二节 IF语句

例4-2 块IF语句

```
PROGRAM EX0402
   IMPLICIT NONE
   REAL (KIND=4) :: HEIGHT ! 记录身高
   REAL (KIND=4) :: WEIGHT ! 记录体重

   WRITE (*,*) “HEIGHT:”
   READ (*,*) HEIGHT    !读入身高
   WRITE (*,*) “WEIGHT:”
   READ (*,*) WEIGHT    !读入体重

   IF (WEIGHT > HEIGHT-100) THEN
   ! 如果体重大于身高减去100，会执行下面的程序
       WRITE (*,*) “TOO FAT ! ”
   ELSE
   ! 如果体重不大于身高减去100，会执行下面的程序
       WRITE (*,*) “UNDER CONTROL. ”
   ENDIF

END
```

第二节 IF语句

例4-3 块IF语句2：降雨量超过500厘米或者风力超过10级，可以停课

```
PROGRAM EX0403
   IMPLICIT NONE
   INTEGER RAIN, WINDSPEED

    WRITE(*,*,ADVANCE='no') “RAIN(mm):”
    READ(*,*) RAIN
    WRITE(*,*,ADVANCE='no') “WIND(scale):”
    READ(*,*) WINDSPEED

    IF (RAIN>=500 .OR. WINDSPEED>=10) THEN
        WRITE (*,*) “停止上班上课”
    ELSE
        WRITE (*,*) “照常上班上课”
    END IF
END
```

第二节 IF语句

```
IF (条件1) THEN
  ……          条件1成立时，执行这个模块程序
  ……
ELSE IF(条件2) THEN
  ……          条件2成立时，执行这个模块程序
  ……
ELSE IF(条件3) THEN
  ……          条件3成立时，执行这个模块程序
  ……
  ……
ELSE IF(条件4) THEN
  ……          条件4成立时，执行这个模块程序
  ……
ELSE            ELSE这个模块可以省略
  ……          每个条件都不成立时，才执行这个模块程序
  ……
END IF
```

ELSE IF 结构要求合理排列逻辑条件，使得一次只能有一个逻辑条件为真。

第二节 IF语句

例4-7 行If语句实现的多重判断

```
PROGRAM EX0407
  IMPLICIT NONE
  INTEGER SCORE
  CHARACTER GRADE

  WRITE (*,*) “SCORE:”
  READ (*,*) SCORE

  IF (SCORE>=90 .AND. SCORE<=100 ) GRADE='A'
  IF (SCORE>=80 .AND. SCORE<=90 )  GRADE='B'
  IF (SCORE>=70 .AND. SCORE<=80 )  GRADE='C'
  IF (SCORE>=60 .AND. SCORE<=70 )  GRADE='D'
  IF (SCORE>=0  .AND. SCORE<=60 )  GRADE='E'
  IF (SCORE>100 .OR.  SCORE<0 )    GRADE='?'

  WRITE (*,”('GRADE:',A1)”) GRADE

END
```

第二节 IF语句

例4-5 块If语句实现的多重判断(1)

```
PROGRAM EX0405
  IMPLICIT NONE
  INTEGER SCORE
  CHARACTER GRADE

  WRITE (*,*) "SCORE:"
  READ (*,*) SCORE

  IF ( SCORE>=90 .AND. SCORE<=100 ) THEN
    GRADE='A'
  ELSE IF ( SCORE>=80 .AND. SCORE<=90 ) THEN
    GRADE='B'
  ELSE IF ( SCORE>=70 .AND. SCORE<=80 ) THEN
    GRADE='C'
  ELSE IF ( SCORE>=60 .AND. SCORE<=70 ) THEN
    GRADE='D'
  ELSE IF ( SCORE>=0 .AND. SCORE<=60 ) THEN
    GRADE='E'
  ELSE
    GRADE='?'       ! SCORE<0或SCORE>100的不合理情况
  END IF

  WRITE (*,"('GRADE:',A1)") GRADE

END
```

第二节 IF语句

例4-6 块If语句实现的多重判断(2)

```
PROGRAM EX0406
 IMPLICIT NONE
 INTEGER SCORE
 CHARACTER GRADE

 WRITE (*,*) "SCORE:"
 READ (*,*) SCORE

 IF (SCORE>100) THEN
    GRADE='?'
 ELSE IF (SCORE>=90) THEN     !会执行到此，代表SCORE<=100
    GRADE='A'
 ELSE IF (SCORE>=80) THEN     !会执行到此，代表SCORE<=90
    GRADE='B'
 ELSE IF (SCORE>=70) THEN     !会执行到此，代表SCORE<=80
    GRADE='C'
 ELSE IF (SCORE>=60) THEN     !会执行到此，代表SCORE<=70
    GRADE='D'
 ELSE IF (SCORE>=0) THEN      !会执行到此，代表SCORE<=60
    GRADE='E'
 ELSE
    GRADE='?'
 END IF

 WRITE(*,"('GRADE:',A1)") GRADE

END
```

第二节 IF语句

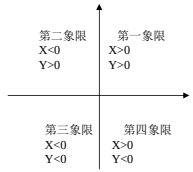
- 嵌套IF语句

```
IF ( …… ) THEN          !第1层IF开始

  IF ( …… ) THEN          !第2层IF开始

    IF ( …… ) THEN          !第3层IF开始
    ELSE IF ( …… ) THEN
    ELSE
    END IF                  !第3层IF结束

  END IF                  !第2层IF结束

END IF                  !第1层IF结束
```

第二节 IF语句

例4-8 判断一个点所在的象限

```
 IF ( X>0 ) THEN
   IF (Y>0) THEN ! X>0,Y>0
     ANS=1
   ELSE IF (Y<0) THEN ! X>0, Y<0
     ANS=4
   ELSE ! X>0, Y=0
     ANS=0
   END IF
 ELSE IF (X<0) THEN
   IF (Y>0) THEN ! X<0, Y>0
     ANS=2
   ELSE IF (Y<0) THEN ! X<0, Y<0
     ANS=3
   ELSE !X<0, Y=0
     ANS=0
   END IF
 ELSE ! X=0, Y=任意数
   ANS=0
 END IF
```



第二节 浮点数及字符的逻辑运算

例4-9 浮点数的判断

```
PROGRAM FloatingPoint
  IMPLICIT NONE
  REAL :: A
  REAL :: B = 3.0        !what will happen for 4.0

  A = SQRT(B)**2 - B     !理论上A应该等于0

  IF (A == 0.0) THEN
    WRITE(*,*) “A等于0”
  ELSE
    WRITE(*,*) “A不等于0”
  END IF
END
```

第二节 浮点数及字符的逻辑运算

例4-10 浮点数的判断

```
PROGRAM FloatingPoint
  IMPLICIT NONE
  REAL :: A
  REAL :: B = 3.0
  REAL, PARAMETER :: E = 1E-4 ! 设置误差范围

  A = SQRT(B)**2 - B     !理论上A应该等于0

  IF (ABS(A-0.0)<=E) THEN
    WRITE(*,*) “A等于0”
  ELSE
    WRITE(*,*) “A不等于0”
  END IF
END
```

第三节 select-case 语句

- 完成多重判断

```
select case(变量)
case(数值1)
  ..... ← 变量等于数值1时，执行这一段
case(数值2)
  ..... ←变量等于数值2时，执行这一段
  .....
case(数值n)
  ..... ←变量等于数值n时，执行这一段
case default
  ..... ←变量不等于上述数值时，执行该段
end select
```

第三节 select-case 语句

- 只能使用整数、字符、及逻辑变量，不能使用浮点数及复数
- 每个CASE中所使用的数值必须是固定的常量，不能使用变量
- 在CASE里的冒号前后放入两个数值时，代表在这两个数字范围中的所有数值
- CASE的括号里还可以用逗号来放入多个变量

第三节 select-case 语句

例4-13 select case 语句

```
SELECT CASE (SCORE)
  CASE(90:100)       ! 90到100分之间
    GRADE='A'
  CASE(80:89)        ! 80到89分之间
    GRADE='B'
  CASE(70:79)        ! 70到79分之间
    GRADE='C'
  CASE(60:69)        ! 60到69分之间
    GRADE='D'
  CASE(0:59)         ! 0到59分之间
    GRADE='E'
  CASE DEFAULT       ! 其他情况
    GRADE='?'
END SELECT
```

第三节 select-case 语句

例4-13 select case 语句

```
PROGRAM EX0413
  IMPLICIT NONE
  REAL A, B, ANS
  CHARACTER OPERATOR

  READ (*,*) A
  READ (*,"(A1)") OPERATOR  ! 不赋值格式时，有些机器会读不到除号"/"
  READ (*,*) B

  SELECT CASE (OPERATOR)
  CASE ( '+' )
    ANS = A + B
  CASE ( '-' )
     ANS = A - B
  CASE ( '*' )
     ANS = A * B
  CASE ( '/' )
     ANS = A / B
  CASE DEFAULT   ! 出入其他符号不处理
     WRITE ( *, " ( ' UNKNOWN OPERATOR ', A1 )" ) OPERATOR
     STOP   !  结束程序
  END SELECT

  WRITE ( * , " ( F6.2, A1, F6.2, '=', F6.2 ) " ) A, OPERATOR, B, ANS
END
```

第三节 select-case 语句

select case 语句

```
PROGRAM EX0413
  IMPLICIT NONE
  CHARACTER CH

  DO
    WRITE(*,'(A)',ADVANCE='NO')'Input a character:'
    READ*,CH
    IF(CH=='@')EXIT
    IF(CH>='A'.AND.CH<='Z'.OR.CH>='a'.AND.CH<='z') THEN
       SELECT CASE(CH)
      CASE('A','E','I','O','U','a','e','i','o','u')
          PRINT*,'Vowel'
      CASE DEFAULT
          PRINT*,'Consonant'
       END SELECT
    ELSE
       PRINT*,'Something else'
    END IF
    END DO
END
```

第四节 其他流程控制方式

- goto 语句标号
  - 跳转并执行语句标号所在行的代码
  - 容易造成程序结构混乱、降低可读性、并产生错误，所以不建议使用

例4-15

```
PROGRAM  EX0415
    IMPLICIT  NONE

    INTEGER  I   !  用来累加使用
    INTEGER  N   !  被当成常量，用来限定I的累加次数
    PARAMETER (N= 10)
    DATA  I /0/

10  WRITE ( * ,  ' (1X, A3, I2 )'  'I=', I
    I = I + 1
    IF (I. LT. N)  GOTO 10  !  I<10  就跳回代码为10的那一行
END
```

第四节 其他流程控制方式

- 算术IF语句
  - IF (算术表达式) 语句标号1，语句标号2，语句标号3
  - 语句标号1，2，3分别对应算术表达式的值小于0，等于0和大于0

例4-17

```
PROGRAM  ex0417
    IMPLICIT  NONE
    REAL  A, B, C
    DATA  A, B  /2.0, 1.0 /

    C = A - B
    IF ( C )  10, 20, 30
10  WRITE ( * ,* )  'A<B'
    GOTO  40
20  WRITE ( * ,* )  'A=B'
    GOTO  40
30  WRITE ( * ,* )  'A>B'
40  STOP
END
```

第四节 其他流程控制方式

- PAUSE
  - 程序执行到PAUSE时，会暂停执行，直到用户按下Enter键才会继续执行
- CONTINUE
  - 继续向下执行程序
- STOP
  - 结束程序执行