

Variables lógicas y sentencia IF

Seminario de Computación – Verano de 2009

Variables lógicas: Solo pueden tomar 2 valores: .TRUE. y .FALSE. (verdadero o falso).

Existen operadores que aplicados a variables REAL, INTEGER o incluso CHARACTER y dan como resultado variables lógicas.

== (.eq.) , /= (.ne.) , > (.gt.) , >= (.ge.) , < (.lt.) , <= (.le.)

Otros operadores permiten operar con variables lógicas y dan como resultado variables lógicas:

.AND. , .OR. , .EQV. , .NEQV. , .NOT.

Ejemplo, determinar si un número entero es divisible por otro número entero.

```
PROGRAM divisible
IMPLICIT NONE
INTEGER :: a=-10 , b=3
INTEGER :: resto
LOGICAL :: es_divisible
!Calculo el resto de la division.
resto= a - b*INT(a/b)
!Ahora aplico el operador logico igualdad entre resto y 0. Si son iguales entonces el resultado
!sera TRUE, si son diferentes entonces sera FALSE.
es_divisible= ( resto == 0. )
WRITE(*,*)a,' es divisible por ',b,'? ',es_divisible
STOP
END PROGRAM divisible
```

También podemos usar la función intrínseca MOD y reemplazar la sentencia

resto=a-b*INT(b/a) por

resto=MOD(a,b)

Podemos modificar el programa anterior para determinar si un número a es divisible en forma simultanea por otros dos números b y c. (todos enteros).

```
PROGRAM divisible
IMPLICIT NONE
INTEGER :: a=12 , b=3 , c=2
INTEGER :: restob , restoc
LOGICAL :: es_divisibleb , es_divisiblec

Calculo el resto de la division.
restob=MOD(a,b)
restoc=MOD(a,c)

Ahora aplico el operador logico igualdad entre resto y 0. Si son iguales entonces el resultado
sera TRUE, si son diferentes entonces sera FALSE.
es_divisibleb = ( restob == 0 )
es_divisiblec = ( restoc == 0 )
es_divisiblebc = es_divisibleb .AND. es_divisiblec

WRITE(*,*)a," es divisible por ",b," y por ",c,"? :",es_divisiblebc

STOP
END PROGRAM divisible
```

Una de las aplicaciones más utilizadas de las variables lógicas y de los operadores lógicos es en el marco de la sentencia IF (condicional)

La sentencia IF permite ejecutar una o más instrucciones sujeto al cumplimiento de una condición. Esa condición puede ser el valor de una variable lógica o el resultado de una operación lógica.

Ejemplo: Quiero hacer un programa que me diga por pantalla si un número es par o impar.

```
PROGRAM par_impar
IMPLICIT NONE
INTEGER :: a=11 , resto
resto=MOD(a,2) !Si a es divisible por 2 entonces resto deberia ser 0.
IF ( resto == 0 ) THEN
    WRITE(*,*)"El numero ",a," es par"
ELSE
    WRITE(*,*)"El numero ",a," es impar"
END IF
STOP
END PROGRAM par_impar
```

También podemos prescindir de la variable resto.

En este ejemplo además el IF toma la decisión en función del contenido de la variable lógica "espar".

```
PROGRAM par_impar
IMPLICIT NONE
INTEGER :: a=11
LOGICAL :: espar
espar= ( MOD(a,2) == 0 )
IF ( espar ) THEN
    WRITE(*,*)"El numero ",a," es par"
ELSE
    WRITE(*,*)"El numero ",a," es impar"
END IF
STOP
END PROGRAM par_impar
```

La sentencia ELSE no es obligatoria. Por ejemplo, calculemos el valor absoluto de un número.

```
PROGRAM absoluto
IMPLICIT NONE
REAL :: X , ABS_X
WRITE(*,*)"Ingrese un numero real"
READ(*,*)X
ABS_X=X
IF( X < 0 )THEN
  ABS_X = -X
END IF
WRITE(*,*)"El valor absoluto de ",X," es: ",ABS_X
END PROGRAM absoluto
```

En este caso, todo el bloque IF se puede escribir de la siguiente manera abreviada:

```
IF( X < 0 ) ABS_X= -X
```

Ejemplo: Hacer un programa que pida ingresar un número entero de 1 a 12 y que determine el nombre del mes asociado.

Vamos a seguir determinados pasos:

- 1) Identificar que información va a entrar al programa y que información va a salir.
- 2) Vamos a determinar como va a ser el procedimiento ("algoritmo") para resolver este problema.
- 3) Podemos representar este algoritmo en forma de diagrama o también en forma de pseudo código.
- 4) Una vez que el sabemos que tenemos que hacer y como lo vamos a hacer escribimos el código fuente del programa.
- 5) Una vez que el programa está funcionando verificamos que este funcionamiento sea correcto.

1) **Variables de entrada y de salida: entra un número (INTEGER) y sale el mes correspondiente a dicho número (CHARACTER)**

2) **¿Cómo va a ser el procedimiento? (Esta es la parte generalmente más complicada del procedimiento).**

Es bueno dividir el procedimiento en pasos, por ejemplo en este caso.

- a) Tengo que leer el número ingresado por el usuario del programa.
- b) Puedo verificar si el número es 1, si es 1 entonces el mes correspondiente es ENERO y le asigno a la variable de salida el valor "ENERO".
- c) Repito el procedimiento anterior para los restantes meses.
- d) Verifico si el número ingresado esta entre 1 y 12 y si no es así muestro un cartel por pantalla (esto podría hacerse al principio del programa).

Otro ejemplo similar: Dado un número de mes y día ingresado por pantalla determinar la estación del año a la cual corresponde.

Establecer el problema: Me dan un número de día y un número de mes y tengo que determinar la estación sabiendo que las mismas cambian el 21 de marzo, 21 de junio, 21 de septiembre y 21 de diciembre.

Variables de entrada y de salida: número de día y número de mes (variables INTEGER). La variable de salida será la estación del año "VERANO", "OTOÑO", "INVIERNO" o "PRIMAVERA" variable CHARACTER.

¿Cómo va a ser el procedimiento?: Hay 8 meses que independientemente del día pertenecen a una estación determinada, los restantes (marzo, junio, septiembre y diciembre) la estación depende del día dentro de dichos meses. Primero pregunto si estoy los meses que pertenecen por entero a una estación y luego si no estoy en ninguno de esos meses me ocupo de los 4 restantes.

El pseudo-código sería algo así:

```
Inicio
INTEGER numero_dia , numero_mes
CHARACTER estacion
READ  numero_dia , numero_mes
IF ( numero_mes es 1 o 2 ) => estacion = "VERANO"
ELSEIF ( numero_mes es 4 o 5 ) => estacion = "OTOÑO"
ELSEIF ( numero_mes es 7 u 8 ) => estacion = "INVIERNO"
ELSEIF ( numero_mes es 10 u 11 ) => estacion = "PRIMAVERA"
!Ahora me ocupo de los otros cuatro meses.
ELSEIF ( numero_mes es 3 ) THEN
    IF ( numero_dia es menor que 21 ) => estacion = "VERANO"
    ELSE => estacion = "INVIERNO"
ENDIF
ELSEIF ( numero_mes es 6 ) THEN
    .....
ENDIF
ELSEIF ( numero_mes es 9 ) THEN
    .....
ENDIF
ELSEIF ( numero_mes es 12 ) THEN
    .....
ENDIF
IF ( numero_mes < 1 .OR. numero_mes > 12 .OR. numero_dia > 31 .OR. numero_dia < 1 )
    WRITE "Error: La fecha ingresada es incorrecta."
ELSE
    WRITE "La estacion es:", estacion
ENDIF
FIN DEL PROGRAMA
```

```
PROGRAM estaciones
MELICIT NONE
INTEGER dia , mes
CHARACTER (LEN=10) :: estacion
WRITE(*,*) 'Ingrese por pantalla el dia y el mes'
READ(*,*) dia , mes
principal: IF ( mes >= 1 AND mes <= 12 AND dia >= 1 AND dia <= 31 ) THEN
    nivel2: IF ( mes == 1 OR mes == 2 ) THEN
        estacion = "VERANO"
    ELSE IF ( mes == 4 OR mes == 5 ) THEN nivel2
        estacion = "OTOÑO"
    ELSE IF ( mes == 7 OR mes == 8 ) THEN nivel2
        estacion = "INVIERNO"
    ELSE IF ( mes == 10 OR mes == 11 ) THEN nivel2
        estacion = "PRIMAVERA"
    ELSE IF ( mes == 3 ) THEN nivel2
        IF ( dia <= 21 ) THEN
            estacion = "VERANO"
        ELSE nivel3_1
            estacion = "OTOÑO"
        END IF nivel3_1
    ELSE IF ( mes == 6 ) THEN nivel2
        IF ( dia <= 21 ) THEN
            estacion = "OTOÑO"
        ELSE nivel3_2
            estacion = "INVIERNO"
        END IF nivel3_2
    ELSE IF ( mes == 9 ) THEN nivel2
        IF ( dia <= 21 ) THEN
            estacion = "INVIERNO"
        ELSE nivel3_3
            estacion = "PRIMAVERA"
        END IF nivel3_3
    ELSE IF ( mes == 12 ) THEN nivel2
        IF ( dia <= 21 ) THEN
            estacion = "PRIMAVERA"
        ELSE nivel3_4
            estacion = "VERANO"
        END IF nivel3_4
    END IF
    WRITE(*,*) "La estacion del año es: ", estacion
ELSE principal
    WRITE(*,*) "Error: la fecha ingresada no es valida"
END IF principal
STOP
END PROGRAM estaciones
```

Otra sentencia que permite organizar el programa en bloques es la sentencia SELECT CASE.

La sintaxis de este bloque de sentencias es:

```

SELECT CASE (selector)
CASE (valores1)
sentencias1
CASE (valores2)
sentencias2
CASE (valores3)
sentencias3
.....
CASE (valores n)
sentencias-n
CASE DEFAULT !Si no es ninguno de los anteriores.
sentencias-DEFAULT
END SELECT

```

```

Ejemplo: Meses del año.
PROGRAM mes_del_año
INTEGER          :: numero_mes
CHARACTER(LEN=30) :: nombre_mes
LOGICAL          :: valido=.TRUE.
WRITE(*,*)'Ingrese el numero del mes'
READ(*,*)numero_mes
SELECT CASE ( numero_mes) !Comienza el CASE
CASE(1)
nombre_mes="ENERO"           !Que hago si la variable numero_mes es 1.
CASE(2)
nombre_mes="FEBRERO"        !Si es 2...
CASE(3)
nombre_mes="MARZO"          !Y así hasta 12
CASE(4)
nombre_mes="ABRIL"
CASE(5)
nombre_mes="MAYO"
CASE(6)
nombre_mes="JUNIO"
CASE(7)
nombre_mes="JULIO"
CASE(8)
nombre_mes="AGOSTO"
CASE(9)
nombre_mes="SEPTIEMBRE"
CASE(10)
nombre_mes="OCTUBRE"
CASE(11)
nombre_mes="NOVIEMBRE"
CASE(12)
nombre_mes="DICIEMBRE"
CASE DEFAULT !Que hago si el valor no corresponde a ninguno de los anteriores
nombre_mes="Numero invalido"
valido=.FALSE.
END SELECT !Termina el CASE
IF(valido)THEN
WRITE(*,*)'El mes ingresado es: ',nombre_mes
ELSE
WRITE(*,*)'El numero ingresado no es valido'
END IF
STOP
END PROGRAM mes_del_año

```

```

PROGRAM estaciones
IMPLICIT NONE
INTEGER :: dia , mes
CHARACTER(LEN=10) :: estacion
LOGICAL :: valido=.TRUE.
WRITE(*,*)'Ingrese el dia y el mes'
READ(*,*)dia,mes
SELECT CASE ( mes )
CASE(1:2)
estacion="VERANO"
CASE(4:5)
estacion="OTOÑO"
CASE(7:8)
estacion="INVIERNO"
CASE(10:11)
estacion="PRIMAVERA"
CASE(3)
IF(dia >= 21)estacion="OTOÑO"
IF(dia < 21)estacion="VERANO"
CASE(6)
IF(dia >= 21)estacion="INVIERNO"
IF(dia < 21)estacion="OTOÑO"
CASE(9)
IF(dia >= 21)estacion="PRIMAVERA"
IF(dia < 21)estacion="INVIERNO"
CASE(12)
IF(dia >= 21)estacion="VERANO"
IF(dia < 21)estacion="PRIMAVERA"
CASE DEFAULT
WRITE(*,*)'La fecha ingresada no es valida'
valido=.FALSE.
END SELECT
IF(valido)THEN
WRITE(*,*)'La estacion es:',estacion
END IF
STOP
END PROGRAM estaciones

```

Ejemplo: El programa que calcula la estación del año reloaded!

En este otro ejemplo utilizamos como variable para la estructura CASE a una variable CHARACTER.

```
PROGRAM otro_ejemplo
IMPLICIT NONE
Determinar si es fin de semana o no a partir del numero de dia de la semana.
CHARACTER (LEN=10) :: nombre_dia
WRITE(*,*)"Ingrese el nombre de un dia de la semana (minuscula)"
READ(*,*)nombre_dia
SELECT CASE ( nombre_dia )
CASE ( "lunes", "martes", "miercoles", "jueves", "viernes" )
WRITE(*,*)"No es fin de semana"
CASE ( "sabado", "domingo" )
WRITE(*,*)"Es fin de semana"
END SELECT
STOP
END PROGRAM otro_ejemplo
```
