

Fortran και Αντικειμενοστραφής προγραμματισμός

www.corelab.ntua.gr/courses/fortran_navai/navai

Δδάσκοντες: Άρης Παγουρτζής (pagour@cs.ntua.gr)
(Επίκουρος Καθηγητής ΣΗΜΜΥ)
Δώρα Σούλιου (dsouliou@mail.ntua.gr)
(ΕΔΙΠ ΣΗΜΜΥ)

3η ενότητα

- ✓ Βρόχοι επανάληψης
- ✓ Βρόχος DO
- ✓ Βρόχος DO WHILE
- ✓ Παραδείγματα

Αρχικές Διαφάνειες σε Pascal: Ε. Ζάχος, Ν. Παπασπύρου
Προσαρμογή σε Fortran - συμπληρώσεις: Α. Παγουρτζής, Δ. Σούλιου

Βρόχοι Επανάληψης (i)

Εντολή **DO ... END DO**

DO

Εντολή 1

Εντολή 2

...

Εντολή n

END DO

Δυνητικά *ατέρμονος βρόχος*, απαραίτητη η χρήση **EXIT**

Εντολές EXIT και CYCLE

Με την εντολή **EXIT** διακόπτεται η εκτέλεση του βρόχου και η εκτέλεση του προγράμματος συνεχίζει μετά το τέλος του βρόχου.

Με την εντολή **CYCLE** διακόπτεται η εκτέλεση της τρέχουσας επανάληψης και η εκτέλεση συνεχίζει με την επόμενη επανάληψη.

Εύρεση ΜΚΔ με βρόχο DO (i)

```
PROGRAM GCD1      ! Εύρεση ΜΚΔ δύο αριθμών
  IMPLICIT NONE
  INTEGER :: a, b, c

  WRITE(*,*) 'Δώσε 2 θετικούς ακераίους: '
  READ(*,*) a, b
  IF (a < b) THEN
    c = a
  ELSE
    c = b
  END IF
  .
  .
  .
```


Εύρεση ΜΚΔ με βρόχο DO (i)

```
! PROGRAM GCD1 συνέχεια  
.  
.  
.  
DO          ! Δοκιμές αριθμών c,c-1,... έως ότου  
              ! βρεθεί κοινός διαιρέτης  
IF ((MOD(a,c)==0) .AND. (MOD(b,c)==0)) THEN  
    EXIT  
ELSE  
    c = c - 1  
END IF  
END DO  
WRITE(*,*) 'Ο ΜΚΔ είναι ο', c  
END PROGRAM GCD1
```


Εύρεση ΜΚΔ με βρόχο DO (ii)

```
PROGRAM GCD2                                ! Εύρεση ΜΚΔ δύο αριθμών
  IMPLICIT NONE
  INTEGER :: a, b
  WRITE(*,*) 'Δώσε 2 θετικούς ακераίους: '
  READ(*,*) a, b
  DO
    IF ((a == 0) .OR. (b == 0)) EXIT
    IF (a > b) THEN
      a = a - b
    ELSE
      b = b - a
    END IF
  END DO
  WRITE(*,*) 'Ο ΜΚΔ είναι ο', a+b
END PROGRAM GCD2
```


Εύρεση ΜΚΔ με βρόχο DO (iii)

```
PROGRAM GCD3                                ! Εύρεση ΜΚΔ δύο αριθμών
  IMPLICIT NONE
  INTEGER :: a, b, t

  WRITE(*,*) 'Δώσε 2 θετικούς ακεραίους: '
  READ(*,*) a, b
  IF (a < b) THEN                            ! πρέπει a>=b αλλιώς
    t = a                                     ! οι τιμές τους ανταλλάσσονται
    a = b
    b = t
  END IF

  DO
    t = MOD(a, b)                            ! υπολογισμός του υπολοίπου t
    IF (t == 0) EXIT                         ! αν t = 0 ο ΜΚΔ είναι ο b
    a = b                                     ! αλλιώς συνεχίζουμε με
    b = t                                     ! τον ΜΚΔ των b και t
  END DO

  WRITE(*,*) 'Ο ΜΚΔ είναι ο', b
END PROGRAM GCD3
```


Εύρεση ΜΚΔ με βρόχο DO (iv)

```
PROGRAM GCD4           ! Εύρεση ΜΚΔ δύο αριθμών
  IMPLICIT NONE
  INTEGER  :: a, b, t
  WRITE(* ,*) 'Δώσε 2 θετικούς ακεραίους: '
  READ(* ,*) a, b

                                     ! Απλοποίηση: δεν χρειάζεται να ελέγξουμε αν a < b
  DO
    t = MOD(a, b)           ! υπολογισμός του υπολοίπου t
    IF (t == 0) EXIT      ! αν t = 0 ο ΜΚΔ είναι ο b
    a = b                   ! αλλιώς συνεχίζουμε με
    b = t                   ! τον ΜΚΔ των b και t
  END DO

  WRITE(* ,*) 'Ο ΜΚΔ είναι ο', b
END PROGRAM GCD4
```


Εύρεση ΜΚΔ με βρόχο DO (v)

```
PROGRAM GCD5           ! Εύρεση ΜΚΔ δύο αριθμών
  IMPLICIT NONE
  INTEGER  :: a, b, t

  WRITE(* ,*) 'Δώσε 2 θετικούς ακεραίους: '
  READ(* ,*) a, b

  ! Δομημένος προγραμματισμός: ορθότερη χρήση του IF
  DO
    t = MOD(a, b)       ! υπολογισμός του υπολοίπου t
    IF (t == 0) THEN
      EXIT             ! αν t = 0 ο ΜΚΔ είναι ο b
    ELSE
      a = b             ! αλλιώς συνεχίζουμε για να βρούμε
      b = t             ! τον ΜΚΔ των b και t
    ENDIF
  END DO

  WRITE(* ,*) 'Ο ΜΚΔ είναι ο', b
END PROGRAM GCD5
```


DO ... END DO με μεταβλητή ελέγχου (i)

DO *control variable* = αρχή, τέλος, βήμα

Εντολή 1

Εντολή 2

...

Εντολή n

END DO

Παράδειγμα

DO a=1,10,1 *! Εκτελείται 10 φορές*

 WRITE(*,*) a

END DO

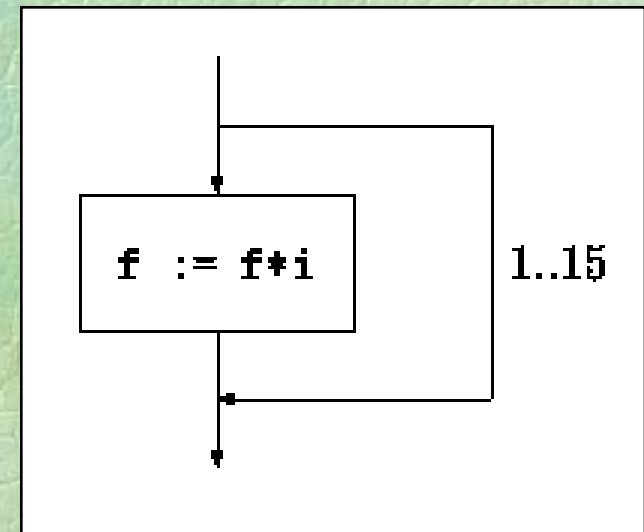
Εντολή DO ... END DO(ii)

Παραδείγματα

```
DO i=10,1,-1  
    WRITE(*,*)i  
END DO
```

```
DO i=41,-3,-1  
    WRITE(*,*)i  
END DO
```

```
f=2  
DO i=1,15  
    WRITE(*,*) f*i  
END DO
```



Εντολή DO ... END DO (iii)

Παραδείγματα (συνέχεια)

```
DO i=1,5  
  DO j=1,10  
    WRITE(*, '(A1\)' ) '*'  
  END DO  
  WRITE(*,*)  
END DO
```

```
*****  
*****  
*****  
*****  
*****
```

```
DO i=1,5  
  DO j=1,2*i  
    WRITE(*, '(A1\)' ) '*'  
  END DO  
  WRITE(*,*)  
END DO
```

```
**  
****  
*****  
*****  
*****
```


Εντολή DO ... END DO (iv)

Ειδικές περιπτώσεις για τα όρια:

DO i=10, 10 ... (* *μία φορά* *)

DO i=12, 10 ... (* *καμία φορά* *)

Η μεταβλητή ελέγχου δεν μπορεί να μεταβληθεί (π.χ. με ανάθεση) μέσα στο σώμα του βρόχου

Τα όρια υπολογίζονται **μία φορά** στην αρχή

Βρόχος DO WHILE ... END DO (i)

Επανάληψη βρόχου όσο ικανοποιείται μια συνθήκη

DO WHILE (συνθήκη)

Εντολή 1

Εντολή 2

...

Εντολή n

Παραδείγματα

END DO

DO WHILE (x>15); k=k+2; **END DO**

DO WHILE (state **.AND.** (x>15))

x=x-5; WRITE(*, *) 'a'

END DO

Εντολή DO WHILE... END DO (ii)

Παραδείγματα

i=1

DO WHILE (i<15)

WRITE(*, *) i; i=i+1

END DO

i=40

DO WHILE (i>2) !Θα τερματίσει ο βρόχος;

WRITE(*,*) i; i=i+5

END DO

Εντολή DO WHILE... END DO (ii)

Παραδείγματα

```
f=10; g=5
```

```
DO WHILE (g*f<200)
```

```
    WRITE(*,*) g*f; g=g+10
```

```
END DO
```

```
ans = 'Y'
```

```
DO WHILE (ans == 'Y')
```

```
    READ (*,*) n
```

```
    WRITE (*,*) 'square of ', n, ' is ', n*n
```

```
    WRITE (*,*) 'continue? (Y/N)'
```

```
    READ (*,*) ans
```

```
END DO
```


Εντολή DO WHILE ... END DO (iii)

Ο αριθμός επαναλήψεων γενικά δεν είναι γνωστός εκ των προτέρων

Αν η συνθήκη είναι αρχικά ψευδής, ο βρόχος τερματίζεται χωρίς να εκτελεστεί το σώμα.

Υπολογισμός του Παραγοντικού

! Πρόγραμμα που υπολογίζει το παραγοντικό

```
PROGRAM PARAGONTIKO
  IMPLICIT NONE
  INTEGER:: n, i, result
  result=1
  WRITE(*,'(A)') "Give number"
  READ(* , *) n
  DO i=1,n
    result = result * i
  END DO
  WRITE(* , *) result
END
```


Υπολογισμός των αριθμών FIBONACCI

```
PROGRAM FIBONACCI ! Πρόγραμμα εκτύπωσης αριθμών Fibonacci
  IMPLICIT NONE
  INTEGER:: n, i, fib, fib_prev=0, fib_last=1

  WRITE(*,'(A)') "Give number"
  READ(* , *) n
  WRITE(*,'(I5\\')') fib_prev
  WRITE(*,'(I5\\')') fib_last
  DO i=3,n
    fib = fib_prev + fib_last
    WRITE(*,'(I5\\')') fib
    fib_prev = fib_last
    fib_last = fib
  END DO
END
```


Ύψωση σε Δύναμη

PROGRAM POWER

! Πρόγραμμα ύψωσης σε δύναμη

IMPLICIT NONE

INTEGER:: a, b, i, p

p=1

WRITE(*,'(A)') "Give the first number"

READ(*,*) a

WRITE(*,'(A)') "Give the second number"

READ(*,*) b

DO i=1,b

 p=p*a

END DO

WRITE(*,*) p

END

Υπολογισμός του ΜΚΔ

```
PROGRAM GCD4
IMPLICIT NONE; INTEGER:: a,b,i,j
WRITE(*,'(a)') "Give number a"; READ(*,* ) a
WRITE(*,'(a)') "Give number b"; READ(*,* ) b
i=a; j=b
DO WHILE ((i>0).AND.(j>0))
  IF (i>j) THEN
    i=MOD(i,j)
  ELSE
    j=MOD(j,i)
  END IF
END DO
WRITE (*,'(I10)') i+j           ! Γιατί δουλεύει σωστά;
END
```


Τι μάθαμε

- Βρόχοι επανάληψης
 - ✓ Απλή μορφή: βρόχος DO. Πρέπει να φροντίζουμε για την έξοδο από το βρόχο με EXIT .
 - ✓ Βρόχος DO με καθορισμένο αριθμό επαναλήψεων.
 - ✓ Βρόχος DO WHILE: έξοδος υπό συνθήκη. Προτιμήστε τον για δομημένο προγραμματισμό!
- Οι βρόχοι επανάληψης είναι από τα πλέον βασικά συστατικά του (προστακτικού) προγραμματισμού!