# Fortran και Αντικειμενοστραφής προγραμματισμός

www.corelab.ntua.gr/courses/fortran_naval/naval

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

## 7η ενότητα

- ✓ Είσοδος Έξοδος
- ✓ Άνοιγμα αρχείου
- ✓ Κλείσιμο αρχείου
- ✓ Διάβασμα από αρχείο
- ✓ Γράψιμο σε αρχείο

# Είσοδος – Έξοδος (1)

Άνοιγμα αρχείου:

```
open (unit = αριθμός, file = "όνομα_αρχείου")
```

*Αριθμός*: θετικός ακέραιος (εκτός του 6) με τον οποίο αναφερόμαστε στο αρχείο

*Όνομα αρχείου*: το όνομα του αρχείου (καλύτερα με όλο το path)

```
open (unit = 3, file = "dedomena")
open (unit = 1, file = "a:keimeno.txt")
open (11, file = "c:\\fortran\\data\\bathmoi.dat")
```

Κλείσιμο αρχείου:

```
close (αριθμός)
close (11)
```

# Είσοδος – Έξοδος (2)

Διάβασμα από αρχείο:

**read (*αριθμός, μορφοποιητής*) *ορίσματα***

```
   read (7,*) a, b

   read (7,5) n, x, y
5  format (i3,f4.1,f5.2)
```

Γράψιμο σε αρχείο:

**write (*αριθμός, μορφοποιητής*) *ορίσματα***

```
   write (3,*) "Oi lyseis einai ", x, " and ", y

   write (3,5) "Oi lyseis einai ", x, " and ", y
5  format (a,f14.5,a,f14.5)
```

# Αριθμός Μονάδας

- Αριθμός Μονάδας είναι ο ακέραιος αριθμός ή η ακέραια έκφραση που αντιστοιχούμε στις συσκευές από τις οποίες διαβάζονται ή στις οποίες γράφονται τα δεδομένα.
- Στα περισσότερα συστήματα το πληκτρολόγιο και η οθόνη έχουν έναν προκαθορισμένο αριθμό μονάδας (συνήθως 5 για το πληκτρολόγιο και 6 για την οθόνη) Έτσι όταν γράφουμε
  - `read(*,*)` είναι το ίδιο με το `read*`, ή `read(5,*)` ή `read(unit=*,fmt=*)`
  - `write(*,*)` είναι το ίδιο με το `print*` ή `write(unit=*,fmt=*)`

# Είσοδος – Έξοδος : παράδειγμα

```
program diairetes
! To programma vriskei tous diairetes enos arithmou
! kai tous grafei sto arxeio 'diairetes.dat'.
integer n, k, d(10)
open (unit = 1, file = "diairetes.dat")
print *, "Dwse thetiko akeraio :"
read (*,*) n
write (1,*) "Oi diairetes tou ", n, "einai :"
k = 0
do i = 1, n
       if (mod(n,i) == 0) then
              k = k + 1
              d(k) = i
       end if
end do
write (1,*) (d(j), j = 1, k)
close (1)
print *, "Oi diairetes vriskontai sto arxeio 'diairetes.dat'"
end
```

# Αρχικοποίηση αρχείων

- Αρχικοποίηση ενός αρχείου είναι η δημιουργία και το άνοιγμα όταν πρόκειται για νέο αρχείο ή το άνοιγμα αν πρόκειται για αρχείο που υπάρχει ήδη.
- `open(unit=unit_number, λίστα επιλογών)`
- Η λίστα επιλογών μπορεί να είναι αρκετά εκτεταμένη αλλά συνήθως περιλαμβάνει το όνομα του αρχείου, το status με τιμές old, new, replace, το form με τιμές formatted, unformatted, το access με τιμές read, write, readwrite και καθορίζει αν το αρχείο θα συνδεθεί με μορφοποιημένη ή μη είσοδο/έξοδο
- Αν το αρχείο άνοιξε σωστά στην επιλογή IOSTAT έχω την τιμή 0.
- Τα αρχεία κλείνουν αυτόματα μετά το τέλος του προγράμματος αλλά προεραιτικά χρησιμοποιείται η εντολή close
- `close(unit=unit_number, λίστα επιλογών)`
- Η λίστα επιλογών εδώ είναι πολύ μικρότερη

# Έξοδος σε αρχείο: παράδειγμα

```
program exodos_se_arxeio
implicit none
integer :: unit_number=12, in_out_status
real, dimension(5) :: dianysma=(/1.1, 2.2,   &
                                 3.3, 4.4, 5.5 /)
open(unit=unit_number, file='eisodos_exodos.txt', &
      status='replace', action='write', iostat=in_out_status)
write(*,*) in_out_status
      if (in_out_status==0) then
            write(unit_number, '(5f10.5)')   &
                                       dianysma
            close(unit_number)
      else
            print '(a70)', 'sfalma den anoigei &
                                       to arxeio'
      end if
END program
```

```
1.10000   2.20000   3.30000   4.40000   5.50000
```

```
Program ReadFILE
implicit none
integer :: i,small(5), big(15),unit_number
open(unit=unit_number,file='----.txt',access='sequential',&
                                        status='old')
write(*,*)
write(*,'(6X,A)') "Reading less than a record "
write(*,'(6X,A)') "============================"
      DO i=1,3
            read(unit_number, *) SMALL
            write(unit=*, fmt='(6X,5I4)' ) SMALL
      END DO
rewind (unit=unit_number)
write(*,*)
write(*,'(6X,A)') "Reading more than a record "
write(*,'(6X,A)') "============================"
      DO i=1,3
            read(unit_number, *) BIG
            write(unit=*, fmt='(6X,15I4)' ) BIG
      END DO
rewind (unit=unit_number)
END
```

Το αρχείο εισόδου :

```
11 12 13 … 19
21 22 23 … 29
31 32 33 … 39
41 42 43 … 49
51 52 53 … 59
61 62 63 … 69
71 72 73 … 79
81 82 83 … 89
91 92 93 … 99
```

Στην έξοδο τυπώνεται:

```
  Reading less than a record
  ==========================
   11  12  13  14  15
   21  22  23  24  25
   31  32  33  34  35

  Reading more than a record
  ==========================
  11  12  13  14  15  16  17  18  19  21  22  23  24  25  26
  31  32  33  34  35  36  37  38  39  41  42  43  44  45  46
  51  52  53  54  55  56  57  58  59  61  62  63  64  65  66
```

!διάβασμα από αρχείο χωρίς μορφοποίηση, εκτύπωση σε αρχείο

```
Program ReadWriteFILE
implicit none
integer :: i,small(5), big(15),unit_number1,unit_number2
unit_number1=10
unit_number2=15
open(unit=unit_number1,file='ReadFile.txt',access='sequential',status='old
open(unit=unit_number2,file='Writefile.txt',access='sequential',status='ne
write(unit_number2,*)
write(unit_number2,'(6X,A)') "1 Reading less than a record &
                                        (list-directed):"
write(unit_number2,'(6X,A)') "================================"
       DO i=1,3
              read(unit_number1, *) SMALL
              write(unit_number2, fmt='(6X,5I4)' ) SMALL
       END DO
rewind (unit_number1)
write(unit_number2,*)
write(unit_number2,'(6X,A)') "2 Reading more than a record &
(list-directed):"
write(unit_number2,'(6X,A)') "================================"
       DO i=1,3
              read(unit_number1, *) BIG
              write(unit_number2, fmt='(6X,15I4)' ) BIG
       END DO
rewind (unit_number1)
END
```

```
1 Reading less than a record (list-directed):
================================
  11  12  13  14  15
  21  22  23  24  25
  31  32  33  34  35

2 Reading more than a record (list-directed):
================================
  11  12  13  14  15  16  17  18  19  21  22  23  24  25  26
  31  32  33  34  35  36  37  38  39  41  42  43  44  45  46
  51  52  53  54  55  56  57  58  59  61  62  63  64  65  66
```

```
Program ReadFILE2
implicit none
integer :: i,small(5), big(15),unit_number1,unit_number2
open(unit=unit_number1,file='ReadFile.txt',access='sequential', &
status='old',form='formated')
write(*,*)
write(*,'(6X,A)') "1 Reading less than a record (formatted):"
write(*,'(6X,A)') "================================"
       DO i=1,3
              read(unit_number1, FMT='(5I4)') SMALL
              write(*, fmt='(6X,5I4)' ) SMALL
       END DO
rewind (unit_number1)
write(*,*)
write(*,'(6X,A)') "2 Reading less than a record (list-directed):"
write(*,'(6X,A)') "================================"

       DO i=1,3
              read(unit_number1, *) SMALL
              write(*, fmt='(6X,5I4)' ) SMALL
       END DO
rewind (unit_number1)
```

```
write(*,*)
write(*,'(6X,A)') "3 Reading more than a record (formatted):"
write(*,'(6X,A)') "================================"

        DO i=1,3
                read(unit_number1, FMT='(15I4)') BIG
                write(*, fmt='(6X,15I4)' ) BIG
        END DO

rewind (unit_number1)
write(*,*)
write(*,'(6X,A)') "4 Reading more than a record (list-directed):"
write(*,'(6X,A)') "================================"

        DO i=1,3
                read(unit_number1, *) BIG
                write(*, fmt='(6X,15I4)' ) BIG
        END DO
rewind (unit_number1)
END
```

```
 1 Reading less than a record (formatted):
 ============================
 11  12  13  14  15
 21  22  23  24  25
 31  32  33  34  35

 2 Reading less than a record (list-directed):
 ============================
 11  12  13  14  15
 21  22  23  24  25
 31  32  33  34  35

 3 Reading more than a record (formatted):
 ============================
 11  12  13  14  15  16  17  18  19   0   0   0   0   0   0
 21  22  23  24  25  26  27  28  29   0   0   0   0   0   0
 31  32  33  34  35  36  37  38  39   0   0   0   0   0   0

 4 Reading more than a record (list-directed):
 ============================
 11  12  13  14  15  16  17  18  19  21  22  23  24  25  26
 31  32  33  34  35  36  37  38  39  41  42  43  44  45  46
 51  52  53  54  55  56  57  58  59  61  62  63  64  65  66
```

```
PROGRAM create_new_file
IMPLICIT NONE
CHARACTER(20), DIMENSION(2)    :: FILE
INTEGER, DIMENSION(2)          :: OPENSTATUS
CHARACTER(20), DIMENSION(1)    :: STUDENTNAME
INTEGER                         :: INPUTSTATUS, STUDENTNUMBER
INTEGER                        :: GPA
INTEGER                        :: I
!Diavase ta onomata arxeiwn pou dinei o xrhshts
WRITE(*,*) "ENTER THE FILENAME AND THE PRODUCED FILE"
READ *,  FILE
!arxeio me dedomena create_new_file1.txt
!arxeio gia eggrafi apotelesmatwn create_new_file2.txt
OPEN (UNIT=10 , FILE=FILE(1), STATUS="OLD", ACTION="READ", &
POSITION ="REWIND", ACCESS="SEQUENTIAL", IOSTAT=OPENSTATUS(1))
IF (OPENSTATUS(1) >0) THEN
       WRITE(*,*) "*** FILE #", 1, "CAN NOT BE OPENED ***"
       STOP
END IF
OPEN (UNIT=30, FILE=FILE(2),STATUS="REPLACE", ACTION="WRITE",&
ACCESS="SEQUENTIAL",IOSTAT=OPENSTATUS(2))
```

Τιμές του IOSTAT

0 αν δεν υπάρχει σφάλμα και δεν έχουμε φτάσει στο τέλος του αρχείου

>0 αν υπάρχει σφάλμα

<0 αν έχουμε φτάσει στο τέλος του αρχείου

```
IF (OPENSTATUS(2) >0) THEN
        WRITE(*,*) "*** FILE #", 2, "CAN NOT BE OPENED ***"
        STOP
END IF

!Diavase kathe eggrafi kai typwse th
DO
        READ (UNIT=10, FMT=100, IOSTAT=INPUTSTATUS) STUDENTNUMBER, &
        STUDENTNAME, GPA
        IF (INPUTSTATUS >0) THEN
                WRITE(*,*) "*** INPUT ERROR***"
        END IF
        IF (INPUTSTATUS<0) THEN
                WRITE(*,*) "End of input file"
                EXIT
        ELSE
                WRITE(30, FMT=100) STUDENTNUMBER, STUDENTNAME, GPA &
                100 FORMAT (I5,  1X,  A, 1X, I5)
        END IF
END DO

PRINT *
PRINT *, "FILE WRITING IS COMPLETE"

END PROGRAM create_new_file
```

```
PROGRAM create_new_file2
!anoigei arxeio me code name vathmo_ptyxiou etos kai ta grafei se
!neo kai ypologizei meso oro ana etos kai megalutero ana etos
IMPLICIT NONE
CHARACTER(20), DIMENSION(2)      :: FILE
INTEGER, DIMENSION(2)          :: OPENSTATUS
CHARACTER(20), DIMENSION(1)      :: STUDENTNAME
INTEGER                  :: INPUTSTATUS, STUDENTNUMBER, ETOS
INTEGER, DIMENSION(5)  :: A,D,B !pinakes gia meso oro, plhthos, …
INTEGER                          :: GPA
INTEGER                          :: I
!Diavase ta onomata arxeiwn pou dinei o xrhshts
WRITE(*,*) "ENTER THE FILENAME AND THE PRODUCED FILE"
READ *,  FILE
!onoma arxeiou me dedomena file.txt
!onoma arxeiou gia eggrafi apotelesmatwn ....txt
OPEN (UNIT=10 , FILE=FILE(1), STATUS="OLD", ACTION="READ", &
POSITION ="REWIND", ACCESS="SEQUENTIAL", IOSTAT=OPENSTATUS(1))
A=0
B=0
D=0
IF (OPENSTATUS(1) >0) THEN
       WRITE(*,*) "*** FILE #", 1, "CAN NOT BE OPENED ***"
       STOP
END IF
```

```
OPEN (UNIT=30, FILE=FILE(2),STATUS="REPLACE", ACTION="WRITE", &
ACCESS="SEQUENTIAL", IOSTAT=OPENSTATUS(2))
IF (OPENSTATUS(2) >0) THEN
        WRITE(*,*) "*** FILE #", 2, "CAN NOT BE OPENED ***"
        STOP
END IF
DO      !Diavase kathe eggrafi kai typwse th
        READ (UNIT=10, FMT=100, IOSTAT=INPUTSTATUS) STUDENTNUMBER, &
                                                    STUDENTNAME, GPA, ETOS
        IF (INPUTSTATUS >0) THEN
                WRITE(*,*) "*** INPUT ERROR***"
        END IF
        IF (INPUTSTATUS<0) THEN
                WRITE(*,*) "End of input file"
                EXIT
        ELSE
                IF (B(ETOS)<GPA) THEN ; B(ETOS)=GPA ; END IF
                A(ETOS)=A(ETOS)+GPA
                D(ETOS)=D(ETOS)+1
                WRITE(30, FMT=100) STUDENTNUMBER, STUDENTNAME, GPA, &
                                                               ETOS
                100 FORMAT (I5,  1X,  A, 1X, I5, 1X, I1)
        END IF
END DO
```

```
A=A/D
WRITE(*,*) "O pinakas meswn wrwn einai", A
WRITE(*,*) "O pinakas megistwn einai", B
WRITE(*,*) "O pinakas foithtvn ana etos einai", D
PRINT *
PRINT *, "FILE WRITING IS COMPLETE"
END PROGRAM create_new_file2
```

```
PROGRAM MERGING_FILES
IMPLICIT NONE
CHARACTER(20), DIMENSION(3)       :: FILE
INTEGER, DIMENSION(3)             :: OPENSTATUS
CHARACTER(20), DIMENSION(2)       :: STUDENTNAME
INTEGER, DIMENSION(2)             :: INPUTSTATUS, STUDENTNUMBER
REAL, DIMENSION(2)                :: GPA
INTEGER                           :: I
!Διάβασε τα ονόματα των αρχείων και άνοιξέ τα
PRINT *, "ENTER THE FILENAMES TO BE MERGED &
          AND THE PRODUCED FILE".
READ *,  FILE
DO I=1,2
       OPEN (UNIT=10*I , FILE=FILE(I), STATUS="OLD", &
                    ACTION="READ", POSITION ="REWIND", &
                    ACCESS="SEQUENTIAL", IOSTAT=OPENSTATUS(I))
END DO
OPEN (UNIT=30, FILE=FILE(3),STATUS="NEW",&
            ACTION="WRITE",ACCESS="SEQUENTIAL",&
                     IOSTAT==OPENSTATUS(3))
DO I=1,3
       IF (OPENSTATUS(I) >0) THEN
              PRINT *, "*** FILE #", I, "CAN NOT BE OPENED ***"
              STOP
       END IF
END DO
```

```
!Διάβασε την πρώτη εγγραφή από κάθε αρχείο
DO I=1,2
        READ (UNIT=10*I, FMT=100, IOSTAT=INPUTSTATUS(I))&
                STUDENTNUMBER(I), STUDENTNAME(I), GPA(I)
        IF (INPUTSTATUS(I) >0) STOP "*** INPUT ERROR***"
END DO
100 FORMAT (I5,  1X,  A,  F4.2)
!Αν δεν έχει τελειώσει ούτε το ένα (FILE(1)) ούτε το άλλο αρχείο (FILE(2))  συνέχισε
DO
        IF(INPUTSTATUS(1)<0 .OR. INPUTSTATUS(2)<0) EXIT
        !IF AT THE END OF EITHER FILE TERMINATE THE REPETITION
        !OTHERWISE CONTINUE
        IF(STUDENTNUMBER(1)< STUDENTNUMBER(2)) THEN
        I=1
ELSE
        I=2
END IF
WRITE(UNIT=30, FMT=100) STUDENTNUMBER(I), STUDENTNAME(I),&
GPA(I)
READ (UNIT=10*I, FMT=100, IOSTAT=INPUTSTATUS(I))&
 STUDENTNUMBER(I), STUDENTNAME(I), GPA(I)
END DO
```

```
!Αν έχουν απομείνει εγγραφές σε κάποιο από τα δύο αρχεία αντέγραψέ τες στο τρίτο αρχείο
DO I=1,2
        DO
            IF (INPUTSTATUS(I)<0) EXIT
              WRITE(UNIT=30, FMT=100) STUDENTNUMBER(I), &
                                                STUDENTNAME(I), GPA(I)
            READ (UNIT=10*I, FMT=100, IOSTAT=INPUTSTATUS(I))&
                      STUDENTNUMBER(I), STUDENTNAME(I), GPA(I)
          END DO
END DO
PRINT *
PRINT *, "FILE MERGING IS COMPLETE"
END PROGRAM MERGING FILES
```