

# Fortran και Αντικειμενοστραφής προγραμματισμός

[www.corelab.ntua.gr/courses/fortran\\_navail/navail](http://www.corelab.ntua.gr/courses/fortran_navail/navail)

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

## 4η ενότητα

- ✓ Είσοδος -Έξοδος
- ✓ Μορφοποίηση Εξόδου
- ✓ Προσδιοριστές Μορφοποίησης
- ✓ Επαναλήψεις Προσδιοριστών

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

- ◆ 2 μορφές εισόδου

- ◆ READ \*, λίστα εισόδου
- ◆ READ(\*,\*) λίστα εισόδου

όπου λίστα εισόδου είναι μία μεταβλητή ή λίστα μεταβλητών που χωρίζονται με κόμμα

- ◆ 2 μορφές εξόδου

- ◆ WRITE (\*,\*) λίστα εξόδου
- ◆ PRINT \*, λίστα εξόδου

όπου λίστα εξόδου είναι μία ή περισσότερες μεταβλητές που χωρίζονται με κόμμα, ή συμβολοσειρές ή σταθερές

- ◆ `WRITE(*,*)`: εμφανίζει τις τιμές των αντικειμένων της λίστας εξόδου. Αν η λίστα εξόδου παραλείπεται εμφανίζεται μία κενή γραμμή. π.χ.  
`WRITE (*,*) 'η ταχύτητα του οχήματος ήταν', x , & 'χιλιόμετρα την ώρα'`
- ◆ `READ(*,*)`: διαβάζει τιμές από το πληκτρολόγιο και τις αναθέτει στις μεταβλητές ακολουθώντας τους παρακάτω κανόνες:
  - ◆ Κάθε φορά που ένα read εκτελείται μία ολόκληρη γραμμή δεδομένων προσπελάζεται.
  - ◆ Οι συνεχόμενες τιμές πρέπει να χωρίζονται με κόμμα ή με ένα ή περισσότερα κενά.

- ◆ Στην περίπτωση που οι είσοδοι σε μία γραμμή είναι λιγότερες από τις μεταβλητές της λίστας εισόδου η εντολή συνεχίζει στις επόμενες γραμμές εως ότου σε όλες τις μεταβλητές εισόδου ανατεθούν τιμές.
- ◆ Στην περίπτωση που οι είσοδοι σε μία γραμμή είναι περισσότερες από τις μεταβλητές της λίστας εισόδου η εντολή χρησιμοποιεί τις πρώτες τιμές που χρειάζεται και αγνοεί τις υπόλοιπες
- ◆ Οι τιμές σε μία γραμμή εισόδου δεδομένων πρέπει να είναι ίδιου τύπου με τις μεταβλητές στις οποίες ανατίθενται (εξαίρεση αποτελεί η ανάθεση μιας ακέραιας μεταβλητής σε μία πραγματική μεταβλητή όπου αυτόματα γίνεται η μετατροπή).

### ◆ Παράδειγμα

```
implicit none
real:: r1,r2,r3,r4,r5=-5.86753,r6
print *, "dwse times sta r1 r3 kai r6"
read *, r1, r3, r6
r2= 0.5*r5*r6**2+r3*r6+r1
r4=r5*r6+r3
print *, "an h timi ths metavlitis r6=", r6, " kai"
print *, "h timi ths r4 einai r4=", r4, " tote"
print *, "h timi ths r2 einai r2=", r2
end program
```

dwse times sta r1 r3 kai r6

4 5 2

an h timi ths metavlitis r6= 2. kai

h timi ths r4 einai r4= -6.7350597 tote

h timi ths r2 einai r2= 2.2649403

# Μορφοποίηση εξόδου

(i)

- ◆ Όταν θέλουμε τα αποτελέσματα μιας εντολής `WRITE (*,*) ...` ή `PRINT *, ...` να εμφανίζονται με συγκεκριμένο τρόπο τροποποιούμε τον δεύτερο `*` στη `write` ή τον `*` στην `print`.
- ◆ 2 τρόποι υλοποίησης της μορφοποίησης
- ◆ 1ος  
`WRITE(*, '(προσδιοριστές μορφοποίησης)')` ή  
`PRINT '(προσδιοριστές μορφοποίησης)', ...` και
- ◆ 2ος
  - ◆ `WRITE (*,x) ...` ή
  - ◆ `PRINT x, ...`
  - ◆ το `x` δηλώνει που θα βρούμε τη μορφοποίηση (`format`)
  - ◆ `x FORMAT (format εξόδου)`

# Μορφοποίηση εξόδου

(ii)

- ◆ Τιμές που μπορεί να έχω στην δεύτερη παράμετρο του write ή στη θέση των προσδιοριστών μορφοποίησης της print
  - ◆ \* : σε αυτή την περίπτωση δεν έχω format
  - ◆ ένας ακέραιος : που αποτελεί label και οδηγεί σε μια εντολή format. Οι τιμές που μπορεί να πάρει ο ακέραιος είναι από 1 έως 99999
  - ◆ μια σταθερά ή μεταβλητή ή έκφραση ή πίνακα τύπου χαρακτήρα των οποίων οι τιμές καθορίζουν το format της εξόδου

παράδειγμα

```
INTEGER:: a=5468,b=4567 ; REAL:: x=23.567
```

```
WRITE(*,10) a, x, b, "συμβολοσειρά"
```

```
10 FORMAT (I4,1X,F6.3/I4,3x,A)
```

Το πρόγραμμα θα εκτυπώσει στην οθόνη

```
5468 23.567
```

```
4567  συμβολοσειρά
```

# Μορφοποίηση εξόδου

# Παράδειγμα

```
implicit none
integer:: num1=12345
real :: num2=12345.78
print *,num1,num2 !έξοδος εξαρτώμενη
!από τον compiler
print '(I5,1x,F8.2)', num1, num2
write(*,*) 'exactly'
print '(I5,F8.2)', num1, num2
write(*,*) 'exactly with space between'
write(*,'(I5,1X,F8.2)') num1,num2
print 10,num1,num2
10 FORMAT (1x,I8,F11.4)
print *,'less space'
print '(I4,1x,F6.2)', num1, num2
print *,'more space'
print '(I8,1x,F6.2)', num1, num2
End program
```

12345 12345.78  
12345 12345.78  
exactly  
1234512345.78  
exactly with space between  
12345 12345.78  
\_\_\_\_12345 12345.7803  
less space  
\*\*\*\* \*  
more space  
12345 12345.7803

# Κυριότεροι Προσδιοριστές Μορφοποίησης

---

**INTEGER Ix** το x δηλώνει πόσες θέσεις θα διατεθούν για την εκτύπωση του ακεραίου

**REAL Fx.w** το x δηλώνει πόσες θέσεις θα χρειαστεί συνολικά ο πραγματικός (και η τελεία) και το w πόσες θέσεις θα χρειαστούν μετά την υποδιαστολή.

**CHARACTER Ax** δηλώνει πόσες θέσεις χρειάζονται για τους χαρακτήρες

**LOGICAL Lx** δηλώνει πόσες θέσεις χρειάζονται για λογικές μεταβλητές

**nX** αφήνει n κενά

/ αλλάζει γραμμή

\ δεν αλλάζει γραμμή για το επόμενο **WRITE**

**n?x** (επαναλαμβάνει n φορές τη μορφοποίηση που ακολουθεί π.χ. 3I5 είναι το ίδιο με I5,I5,I5)

# Προσδιοριστής Ακεραίων

---

`nIx.m` : το `I` δηλώνει ακέραιο, το `x` δηλώνει πόσες θέσεις θα διατεθούν για την εκτύπωση του ακεραίου, το `n`, αν υπάρχει, δηλώνει πόσοι ακέραιοι με το ίδιο `format` θα εκτυπωθούν. Αν `n=1` η τιμή του `n` παραλείπεται. Το `m`, αν υπάρχει, δηλώνει το ελάχιστο μήκος που πρέπει να έχει ο ακέραιος που θα εκτυπωθεί π.χ. για `x=12345` η εντολή `print '(I10.8)', num1` θα τυπώσει τον ακέραιο `00012345`

Προσοχή! Αν ο ακέραιος έχει περισσότερα ψηφία από το μήκος που καθορίζεται στο `format` θα τυπωθούν αστερίσκοι. Αν έχει λιγότερα θα τυπωθούν στις πιο αριστερές θέσεις κενά.

# Προσδιοριστής Πραγματικών (με χρήση F)

---

nFx.w το F δηλώνει πραγματικό αριθμό, το x δηλώνει πόσες θέσεις θα διατεθούν συνολικά για την εκτύπωση του πραγματικού (και της τελείας) και το w πόσες θέσεις θα διατεθούν για τα ψηφία μετά την υποδιαστολή. Το n αν υπάρχει δηλώνει πόσοι πραγματικοί με το ίδιο format θα εκτυπωθούν. Αν πρόκειται για έναν πραγματικό μόνο η τιμή του n παραλείπεται.

Σε περίπτωση που ο πραγματικός έχει περισσότερα ψηφία μετά την υποδιαστολή από την τιμή του w τότε στρογγυλοποιείται π.χ. έστω  $x=3.14156$

```
print '(F6.4 )' , x
```

```
3.1416
```

# Προσδιοριστής Πραγματικών (με χρήση E)

---

nEx.w

E δηλώνει πραγματικό αριθμό σε εκθετική μορφή,

x δηλώνει το συνολικό μήκος εκτύπωσης του πραγματικού

w δηλώνει τον αριθμό των δεκαδικών ψηφίων που θα εκτυπωθούν

n (αν υπάρχει) δηλώνει το πλήθος των δεκαδικών που θα εκτυπωθούν με την ίδια μορφή

παράδειγμα

```
real:: A=0.12345E8, B=0.0237, C=4.6E-12, D=-76.1684E12
```

```
print '(1X, 2E15.5, E15.4, E14.4)', A,B,C,D
```

```
_____0.12345E+08_____0.23700E-01_____0.4600E-11_____-  
0.7617E+14
```

5 κενά real

4 κενά real

5 κενά real

3 κενά real

# Προσδιοριστής Χαρακτήρων

---

nAx το A δηλώνει ότι πρόκειται για χαρακτήρα ή συμβολοσειρά, το n αν υπάρχει δηλώνει πόσες συμβολοσειρές ίδιου μήκους θα τυπωθούν, και το x αν υπάρχει δηλώνει πόσες θέσεις θα διατεθούν για την εκτύπωση της συμβολοσειράς

Η συμβολοσειρά μπορεί να αποτελεί τμήμα του προσδιοριστή μορφοποίησης π.χ.

```
real :: x1=0.32, y1=7.89
```

```
print '(1X, "x1=", F6.2, " y1=", F6.2) ', x1,y1
```

τυπώνει

```
_x1=___ 0.32__y1=___ 7.89
```

2 κενά

# Προσδιοριστές κενών και θέσης

---

`nX` εισάγει κενά τόσα όσα δηλώνει ο ακέραιος `n`

`Tx` όπου `x` δηλώνει τη θέση που θα τυπωθεί ο επόμενος χαρακτήρας από την αρχή της γραμμής. Κάποιες φορές η θέση αυτή μπορεί να είναι και πριν την τρέχουσα θέση. Έτσι μπορεί να τυπωθεί ο επόμενος χαρακτήρας σε θέση κάποιου προηγούμενου. Αυτό να αποφεύγεται γιατί σε πολλά συστήματα αυτό δίνει `run time error`

παράδειγμα

έστω `num1=123`

```
print 15, "abcdefghigk", "abcd", num1
```

```
15 FORMAT (1X, A11, 3X, A4 ,2X, I3)
```

τυπώνει στην έξοδο

```
_abcdefghigk___abcd___123
```

(1κενό chars 3 κενά chars 2 κενά integer)

το ίδιο θα πάρουμε και με τα παρακάτω formats

```
15 FORMAT (1X, A11, T16, A4, 2X, I3)
```

```
15 FORMAT (1X, A11, T22, I3, T16, A4) (προσοχή! να αποφεύγεται)
```

# Προσδιοριστής /

---

Το / ή n/ Χρησιμοποιούνται προκειμένου να αλλάξουμε γραμμή ή γραμμές το πλήθος των οποίων καθορίζεται από το n

Δεν είναι απαραίτητο να χρησιμοποιήσουμε κόμα προκειμένου να ξεχωρίσουμε τον προσδιοριστή / από τους υπόλοιπους προσδιοριστές

# Επαναλήψεις προσδιοριστών

---

Συμπιεσμένες μορφές format

3F10.2 τυπώνει 3 πραγματικούς με το ίδιο ακριβώς format

'(1X, 2(A, F6.3))' εκτυπώνει το ίδιο με

'(1X, A, F6.3, A, F6.3)'

και μπορούμε να προχωρήσουμε περισσότερο γράφοντας

'(1X, 2(E18.2, 2(I3,A)), F8.4) ' που είναι το ίδιο με

'(1X, E18.2, I3,A, I3,A, E18.2, I3,A,I3,A, F8.4) '