

ΕΤΥ-114 Ηλεκτρονικοί Υπολογιστές Ι

Εισαγωγή στη γλώσσα προγραμματισμού Fortran 95

Σταμάτης Σταματιάδης
stamatis@materials.uoc.gr

Τμήμα Επιστήμης και Τεχνολογίας Υλικών,
Πανεπιστήμιο Κρήτης

Διαλέξεις

Παρασκευή 09:00-11:00 στην αίθουσα A1 του κτηρίου Επιστήμης Υπολογιστών.

Τμήματα εργαστηριακών ασκήσεων

- Τρίτη 11:00-14:00,
- Τετάρτη 09:00-12:00.

Διαλέξεις

Η παρακολούθηση των διαλέξεων είναι προαιρετική.

Εργαστηριακές ασκήσεις

- Γίνονται στην αίθουσα ΗΥ (απέναντι από τη Γραμματεία).
- Η συμμετοχή είναι **ΥΠΟΧΡΕΩΤΙΚΗ** για τους εισαχθέντες του 2018 και μετά.
- Έως **δύο** απουσίες είναι επιτρεπτές.
- Η εγγραφή σε τμήματα γίνεται στο eclass του μαθήματος.

Βιβλία

- Στην ιστοσελίδα διατίθενται οι σημειώσεις που θα ακολουθήσουμε.
- Παρέχεται επιπλέον βιβλίο μέσω Εύδοξου:
 - FORTRAN 77/90/95 ΚΑΙ FORTRAN 2003, Α. Σ. Καράκος
 - Εισαγωγή στην Fortran 90/95/2003, Ν. Καραμπετάκης

Ιστοσελίδα

Ιστοσελίδα TETY → Προπτυχιακά → Ιστοσελίδες Μαθημάτων

Eclass

<https://teleclass.materials.uoc.gr/courses/SEM1106>

Διεξαγωγή

- Στο τέλος κάθε εργαστηρίου (40%) και στην εξεταστική περίοδο (60%).
- Αποτελούνται από ασκήσεις συγγραφής κώδικα με **ανοιχτά βιβλία**.

Επιτυχία στο μάθημα

- Τουλάχιστον 10 παρουσίες (στα 12 εργαστήρια),
- **KAI** βαθμός κατά μέσο όρο ≥ 4.0 στις εξετάσεις των εργαστηρίων (υπολογίζονται οι 10 καλύτεροι βαθμοί),
- **KAI** βαθμός ≥ 5.0 συνολικά.

Αποτυχία στα εργαστήρια

- Όλα τα εργαστήρια **ΕΠΑΝΑΛΑΜΒΑΝΟΝΤΑΙ** σε άλλη χρονιά.
- Δεν επιτρέπεται η συμμετοχή στην εξέταση Ιανουαρίου ή Σεπτεμβρίου χωρίς επιτυχία στα εργαστήρια.

Αποτυχία στην τελική εξέταση

- Οι βαθμοί των εργαστηρίων διατηρούνται.
- Δεν υπάρχει η δυνατότητα να επαναληφθούν τα εργαστήρια.
- Συμμετέχετε μόνο στις τελικές εξετάσεις.

Εξετάσεις (για μη υπόχρεους)

- Η παρακολούθηση των εργαστηρίων είναι προαιρετική και μόνο αν υπάρχουν κενές θέσεις.
- Η βαθμολογία προκύπτει κατά 100% από την τελική εξέταση τον Ιανουάριο ή το Σεπτέμβριο (ή τον Ιούνιο). Όσοι συμμετάσχουν στις εξετάσεις των εργαστηρίων, θα έχουν το 40% από αυτές και το 60% από την τελική εξέταση.

Γιατί χρειάζεται ο προγραμματισμός υπολογιστών;

Αριθμητικά προβλήματα

Έχουμε N αριθμούς.

- Πόσο είναι το άθροισμά τους;
- Ποιος είναι ο μεγαλύτερος;
- Μπορούμε να τους γράψουμε σε αύξουσα σειρά;

Γιατί χρειάζεται ο προγραμματισμός υπολογιστών;

Αριθμητικά προβλήματα

Έχουμε N αριθμούς.

- Πόσο είναι το άθροισμά τους;
- Ποιος είναι ο μεγαλύτερος;
- Μπορούμε να τους γράψουμε σε αύξουσα σειρά;

“Καθημερινά” προβλήματα

- Τι μέρα (Δευτέρα, Τρίτη, ...) πέφτει η Πρωτοχρονιά;
- Πότε είναι το Πάσχα για τα επόμενα 20 χρόνια;
- Καταθέτουμε κάθε μήνα 20€ σε έντοκο λογαριασμό τράπεζας. Τι ποσό θα έχουμε σε 5 χρόνια;
- Πρέπει να αγοράσω 10 είδη από διαφορετικά μαγαζιά. Με ποια σειρά πρέπει να τα επισκεφτώ ώστε να τελειώσω όσο πιο γρήγορα γίνεται;
- Έχω ένα οικόπεδο ακανόνιστου σχήματος. Ποιο είναι το εμβαδό του;

Αλγόριθμος

Η μέθοδος επίλυσης ενός προβλήματος. Ίσως δεν είναι μοναδική ή πρακτικά εφαρμόσιμη.

Αλγόριθμος

Η μέθοδος επίλυσης ενός προβλήματος. Ίσως δεν είναι μοναδική ή πρακτικά εφαρμόσιμη.

Προγραμματισμός υπολογιστών

Η “μετάφραση” ενός αλγόριθμου σε οδηγίες που μπορεί να ακολουθήσει ένας υπολογιστής.

Αλγόριθμος

Η μέθοδος επίλυσης ενός προβλήματος. Ίσως δεν είναι μοναδική ή πρακτικά εφαρμόσιμη.

Προγραμματισμός υπολογιστών

Η “μετάφραση” ενός αλγόριθμου σε οδηγίες που μπορεί να ακολουθήσει ένας υπολογιστής.

Κώδικας (ή Πρόγραμμα)

Οι οδηγίες προς τον υπολογιστή. Γράφονται σε κάποια από τις πολλές γλώσσες προγραμματισμού. Πρέπει να είναι **πλήρεις, λεπτομερείς, ακριβείς, σαφείς.**

Αλγόριθμος

Η μέθοδος επίλυσης ενός προβλήματος. Ίσως δεν είναι μοναδική ή πρακτικά εφαρμόσιμη.

Προγραμματισμός υπολογιστών

Η “μετάφραση” ενός αλγόριθμου σε οδηγίες που μπορεί να ακολουθήσει ένας υπολογιστής.

Κώδικας (ή Πρόγραμμα)

Οι οδηγίες προς τον υπολογιστή. Γράφονται σε κάποια από τις πολλές γλώσσες προγραμματισμού. Πρέπει να είναι **πλήρεις, λεπτομερείς, ακριβείς, σαφείς**.

Ο υπολογιστής είναι χαζός! Όμως, εκτελεί τον κώδικα χωρίς λάθη, χωρίς κόπωση και πιο γρήγορα από εμάς.

Εξέλιξη της Fortran

Fortran: Η παλαιότερη γλώσσα προγραμματισμού.

Δημιουργός

John W. Backus, IBM (1955)

Στόχος

Γλώσσα υψηλού επιπέδου για επιστημονικά προγράμματα, αντί για assembly: **FOR**mula **TRAN**slation

Standards

- 1966 (Fortran 66)
- 1977 (Fortran 77)
- 1990 (Fortran 90)
- 1995 (Fortran 95)
- 2003 (Fortran 2003)
- 2008 (Fortran 2008)
- 2018 (Fortran 2018)

Διαδικασία προγραμματισμού

- Δημιουργούμε κώδικα σε κειμενογράφο. Αποθηκεύουμε σε αρχείο με κατάληξη `.f90` ή `.f95`.

Διαδικασία προγραμματισμού

- Δημιουργούμε κώδικα σε κειμενογράφο. Αποθηκεύουμε σε αρχείο με κατάληξη `.f90` ή `.f95`.
- Μεταγλωττίζουμε σε κώδικα μηχανής με κατάλληλη εντολή.

Διαδικασία προγραμματισμού

- Δημιουργούμε κώδικα σε κειμενογράφο. Αποθηκεύουμε σε αρχείο με κατάληξη `.f90` ή `.f95`.
- Μεταγλωττίζουμε σε κώδικα μηχανής με κατάλληλη εντολή.
- Διορθώνουμε τυχόν σφάλματα που υποδεικνύει ο μεταγλωττιστής. Αν κάνουμε αλλαγές στον κώδικα, μεταγλωττίζουμε ξανά.

Διαδικασία προγραμματισμού

- Δημιουργούμε κώδικα σε κειμενογράφο. Αποθηκεύουμε σε αρχείο με κατάληξη `.f90` ή `.f95`.
- Μεταγλωττίζουμε σε κώδικα μηχανής με κατάλληλη εντολή.
- Διορθώνουμε τυχόν σφάλματα που υποδεικνύει ο μεταγλωττιστής. Αν κάνουμε αλλαγές στον κώδικα, μεταγλωττίζουμε ξανά.
- Εκτελούμε το πρόγραμμα με κατάλληλη εντολή.

Διαδικασία προγραμματισμού

- Δημιουργούμε κώδικα σε κειμενογράφο. Αποθηκεύουμε σε αρχείο με κατάληξη `.f90` ή `.f95`.
- Μεταγλωττίζουμε σε κώδικα μηχανής με κατάλληλη εντολή.
- Διορθώνουμε τυχόν σφάλματα που υποδεικνύει ο μεταγλωττιστής. Αν κάνουμε αλλαγές στον κώδικα, μεταγλωττίζουμε ξανά.
- Εκτελούμε το πρόγραμμα με κατάλληλη εντολή.

Προτεινόμενοι μεταγλωττιστές (compilers)

για Windows Silverfrost Fortran FTN95

για Linux, MacOS GNU Fortran

για οποιοδήποτε σύστημα Online Fortran compiler

Προγραμματισμός στο Linux

1. Δημιουργία κώδικα σε κειμενογράφο (**emacs**, gedit, vi, ...) ή IDE (Code::Blocks, eclipse, ...).
Αποθηκεύουμε το αρχείο με κατάληξη **.f90**. Π.χ.
ονομα.f90

Προγραμματισμός στο Linux

1. Δημιουργία κώδικα σε κειμενογράφο (**emacs**, gedit, vi, ...) ή IDE (Code::Blocks, eclipse, ...).
Αποθηκεύουμε το αρχείο με κατάληξη **.f90**. Π.χ.
onoma.f90
2. Μεταγλώττιση (στο terminal ή εσωτερικά στο IDE):
gfortran onoma.f90

Προγραμματισμός στο Linux

1. Δημιουργία κώδικα σε κειμενογράφο (**emacs**, gedit, vi, ...) ή IDE (Code::Blocks, eclipse, ...).
Αποθηκεύουμε το αρχείο με κατάληξη **.f90**. Π.χ.
onoma.f90
2. Μεταγλώττιση (στο terminal ή εσωτερικά στο IDE):
gfortran onoma.f90
3. Διορθώνουμε τυχόν σφάλματα που υποδεικνύει ο μεταγλωττιστής.
Αν κάνουμε αλλαγές στον κώδικα, μεταγλωττίζουμε ξανά.

Προγραμματισμός στο Linux

1. Δημιουργία κώδικα σε κειμενογράφο (**emacs**, gedit, vi, ...) ή IDE (Code::Blocks, eclipse, ...).
Αποθηκεύουμε το αρχείο με κατάληξη **.f90**. Π.χ.
onoma.f90
2. Μεταγλώττιση (στο terminal ή εσωτερικά στο IDE):
gfortran onoma.f90
3. Διορθώνουμε τυχόν σφάλματα που υποδεικνύει ο μεταγλωττιστής.
Αν κάνουμε αλλαγές στον κώδικα, μεταγλωττίζουμε ξανά.
4. Εκτέλεση προγράμματος (στο terminal ή εσωτερικά στο IDE):
./a.out

Παράδειγμα προγράμματος Fortran

! Υπολογισμός του διπλασίου αριθμού που διαβάζει.

```
PROGRAM diplasio
  IMPLICIT NONE
  INTEGER :: a
  INTEGER :: b

  PRINT *, "Dose akeraio"
  READ *, a

  b = 2 * a

  PRINT *, "To diplasio einai"
  PRINT *, b
END PROGRAM diplasio
```


Γενικά

- Το κύριο πρόγραμμα γράφεται μέσα σε **PROGRAM . . .** και **END PROGRAM . . .**
- Οι εντολές εκτελούνται με τη σειρά που γράφονται, από πάνω προς τα κάτω.
- Το πρόγραμμα χωρίζεται στο τμήμα των δηλώσεων και στο τμήμα των εντολών.
- Πρώτη δήλωση είναι το **IMPLICIT NONE**.
- Κεφαλαία και πεζά γράμματα είναι **το ίδιο**.
- Δεν απαιτείται συγκεκριμένη στοίχιση. Καλή στοίχιση, όμως, καθιστά το πρόγραμμα ευανάγνωστο.
- Κείμενο που αρχίζει με το χαρακτήρα **!** και τελειώνει στο τέλος της γραμμής αγνοείται από το μεταγλωττιστή: είναι **σχόλιο**.

INTEGER

Τύπος για ακέραιες ποσότητες.

Τιμές ακέραιων

Σειρά αριθμητικών ψηφίων (0-9), χωρίς κενά ή άλλα σύμβολα, με πιθανό πρόσημο (+, -):

Π.χ. -12 , 54321

Παρατήρηση

Συνήθης μέγιστη τιμή: $2147483647 (= 2^{31} - 1)$.

Συνήθης ελάχιστη τιμή: $-2147483648 (-2^{31})$.

REAL

*Τύπος για πραγματικές ποσότητες, απλής ακρίβειας,
(6 σημαντικά ψηφία).*

Τύπος για πραγματικές ποσότητες, *απλής ακρίβειας*,
(6 σημαντικά ψηφία).

Τιμές πραγματικών ποσοτήτων απλής ακρίβειας

Σειρά αριθμητικών ψηφίων (0-9), χωρίς κενά, με πιθανό πρόσημο (+, -)
και

- *ή* τελεία (αντί για υποδιαστολή) που χωρίζει το ακέραιο από το δεκαδικό μέρος: Π.χ. 12.345, -1.02.
- *ή* το χαρακτήρα e (ή E) που χωρίζει τον αριθμό από τη δύναμη του 10 με την οποία πολλαπλασιάζεται. Π.χ. 123E2 ($\equiv 123 \times 10^2 \equiv 12300.0$), -12e-1 ($\equiv -1.2$),
- *ή* συνδυασμό των παραπάνω: Π.χ. -1.2E-2 ($\equiv -0.012$).

DOUBLE PRECISION

*Τύπος για πραγματικές ποσότητες, διπλής ακρίβειας,
(15 σημαντικά ψηφία).*

Τιμές πραγματικών ποσοτήτων διπλής ακρίβειας

Σειρά αριθμητικών ψηφίων (0-9), χωρίς κενά, με πιθανό πρόσημο (+, -),

- πιθανή τελεία (αντί για υποδιαστολή) που χωρίζει το ακέραιο από το δεκαδικό μέρος,
- **και** το χαρακτήρα d (ή D) που χωρίζει τον αριθμό από τη δύναμη του 10 με την οποία πολλαπλασιάζεται.

Παραδείγματα

123D2 (= 123×10^2) -12.3d-1 (= -12.3×10^{-1}).

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια;

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; `8.1d0`

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; $8.1d0$
- Πώς γράφουμε το 10^{-6} σε απλή ακρίβεια;
a) $10E-6$ b) $10E(-6)$ c) $E-6$ d) $1E-6$ e) $1E(-6)$

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; $8.1d0$
- Πώς γράφουμε το 10^{-6} σε απλή ακρίβεια;
a) $10E-6$ b) $10E(-6)$ c) $E-6$ d) $1E-6$ e) $1E(-6)$

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; **8.1d0**
- Πώς γράφουμε το 10^{-6} σε απλή ακρίβεια;
a) $10E-6$ b) $10E(-6)$ c) $E-6$ d) **$1E-6$** e) $1E(-6)$
- Πώς γράφουμε το 10^{-6} σε διπλή ακρίβεια;

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; **8.1d0**
- Πώς γράφουμε το 10^{-6} σε απλή ακρίβεια;
a) $10E-6$ b) $10E(-6)$ c) $E-6$ d) **$1E-6$** e) $1E(-6)$
- Πώς γράφουμε το 10^{-6} σε διπλή ακρίβεια; **1d-6**

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; **8.1d0**
- Πώς γράφουμε το 10^{-6} σε απλή ακρίβεια;
a) $10E-6$ b) $10E(-6)$ c) $E-6$ d) **$1E-6$** e) $1E(-6)$
- Πώς γράφουμε το 10^{-6} σε διπλή ακρίβεια; **1d-6**
- Ποιον πραγματικό τύπο να χρησιμοποιούμε;

Παρατηρήσεις στους πραγματικούς τύπους

- Το ακέραιο ή το δεκαδικό μέρος που είναι 0 μπορεί να παραλείπεται: $0.12 \equiv .12$
- Στους πραγματικούς διπλής ακρίβειας το d (ή D) είναι υποχρεωτικό.
- Πώς γράφουμε το 8.1 σε διπλή ακρίβεια; **8.1d0**
- Πώς γράφουμε το 10^{-6} σε απλή ακρίβεια;
a) $10E-6$ b) $10E(-6)$ c) $E-6$ d) **$1E-6$** e) $1E(-6)$
- Πώς γράφουμε το 10^{-6} σε διπλή ακρίβεια; **$1d-6$**
- Ποιον πραγματικό τύπο να χρησιμοποιούμε; **DOUBLE PRECISION**

Αριθμητικοί τελεστές μεταξύ ακέραιων ποσοτήτων

Άθροισμα

+

Π.χ. $5 + 7$

Διαφορά

-

Π.χ. $5 - 7$

Γινόμενο

*

Π.χ. $2 * 3$

Αριθμητικοί τελεστές μεταξύ ακέραιων ποσοτήτων

Άθροισμα

+

Π.χ. $5 + 7$

Διαφορά

-

Π.χ. $5 - 7$

Γινόμενο

*

Π.χ. $2 * 3$

Πηλίκο

/

Π.χ. $7/2$

→ 3 (πόσες φορές «χωρά» το 2 στο 7).

Αριθμητικοί τελεστές μεταξύ ακέραιων ποσοτήτων

Άθροισμα	+	Π.χ. $5 + 7$
Διαφορά	-	Π.χ. $5 - 7$
Γινόμενο	*	Π.χ. $2 * 3$
Πηλίκο	/	Π.χ. $7/2$ → 3 (πόσες φορές «χωρά» το 2 στο 7).
Υπόλοιπο	MOD(,)	Π.χ. MOD(7,2) → 1

Αριθμητικοί τελεστές μεταξύ ακέραιων ποσοτήτων

Άθροισμα	+	Π.χ. 5 + 7
Διαφορά	-	Π.χ. 5 - 7
Γινόμενο	*	Π.χ. 2 * 3
Πηλίκο	/	Π.χ. 7/2 → 3 (πόσες φορές «χωρά» το 2 στο 7).
Υπόλοιπο	MOD(,)	Π.χ. MOD(7,2) → 1
Ύψωση σε δύναμη	**	Π.χ. 3**2

Αριθμητικοί τελεστές μεταξύ ακέραιων ποσοτήτων

Άθροισμα	+	Π.χ. 5 + 7
Διαφορά	-	Π.χ. 5 - 7
Γινόμενο	*	Π.χ. 2 * 3
Πηλίκο	/	Π.χ. 7/2 → 3 (πόσες φορές «χωρά» το 2 στο 7).
Υπόλοιπο	MOD(,)	Π.χ. MOD(7,2) → 1
Ύψωση σε δύναμη	**	Π.χ. 3**2

Γενικά ισχύει (για ακέραια a,b με $b > 0$)

- $0 \leq \mathbf{MOD}(a,b) < b$,
- $a = (a/b) * b + \mathbf{MOD}(a,b)$.

Αριθμητικοί τελεστές μεταξύ πραγματικών ποσοτήτων

Άθροισμα	+	Π.χ.	$5d0 + 7d0$
Διαφορά	-	Π.χ.	$5d0 - 7d0$
Γινόμενο	*	Π.χ.	$2d0 * 3d0$

Αριθμητικοί τελεστές μεταξύ πραγματικών ποσοτήτων

Άθροισμα	+	Π.χ.	$5d_0 + 7d_0$
Διαφορά	-	Π.χ.	$5d_0 - 7d_0$
Γινόμενο	*	Π.χ.	$2d_0 * 3d_0$
Λόγος	/	Π.χ.	$3d_0/2d_0$

Αριθμητικοί τελεστές μεταξύ πραγματικών ποσοτήτων

Άθροισμα	+	Π.χ.	$5d0 + 7d0$
Διαφορά	-	Π.χ.	$5d0 - 7d0$
Γινόμενο	*	Π.χ.	$2d0 * 3d0$
Λόγος	/	Π.χ.	$3d0/2d0$
Ύψωση σε δύναμη	**	Π.χ.	$3d0**2d0$

Σχετικές προτεραιότητες αριθμητικών τελεστών

Πολύ Υψηλή Παρενθέσεις ()

Υψηλή **

Μεσαία *, /

Χαμηλή +, -

Παρατήρηση

Τελεστές με ίδια προτεραιότητα σε μια έκφραση εκτελούνται από αριστερά προς τα δεξιά.

Σχετικές προτεραιότητες αριθμητικών τελεστών

Πολύ Υψηλή Παρενθέσεις ()

Υψηλή **

Μεσαία *, /

Χαμηλή +, -

Παρατήρηση

Τελεστές με ίδια προτεραιότητα σε μια έκφραση εκτελούνται από αριστερά προς τα δεξιά.

Παράδειγμα

$$(2d\theta+3d\theta)**3/2 \rightarrow ((2d\theta+3d\theta)**3)/2 \rightarrow 62.5d\theta$$

όχι $5.0^1 \rightarrow 5d\theta$ ούτε $5.0^{1.5} \approx 11.18d\theta$.

Γενικές παρατηρήσεις στους αριθμητικούς τελεστές

Κανόνες

- Τελεστές που δρουν μεταξύ ποσοτήτων **ίδιου τύπου** δίνουν αποτέλεσμα **αυτού του τύπου**.
- Τελεστές που δρουν μεταξύ ποσοτήτων **διαφορετικού τύπου** προκαλούν μετατροπή της **τιμής** της ποσότητας «χαμηλότερης» ακρίβειας στον τύπο με την «υψηλότερη» ακρίβεια.
Π.χ. πράξη μεταξύ ενός **DOUBLE PRECISION** και ενός **INTEGER** γίνεται αφού μετατραπεί ο **INTEGER** σε **DOUBLE PRECISION**.

Γενικές παρατηρήσεις στους αριθμητικούς τελεστές

Κανόνες

- Τελεστές που δρουν μεταξύ ποσοτήτων **ίδιου τύπου** δίνουν αποτέλεσμα **αυτού του τύπου**.
- Τελεστές που δρουν μεταξύ ποσοτήτων **διαφορετικού τύπου** προκαλούν μετατροπή της **τιμής** της ποσότητας «χαμηλότερης» ακρίβειας στον τύπο με την «υψηλότερη» ακρίβεια.
Π.χ. πράξη μεταξύ ενός **DOUBLE PRECISION** και ενός **INTEGER** γίνεται αφού μετατραπεί ο **INTEGER** σε **DOUBLE PRECISION**.

Παρατηρήσεις

- Πόσο κάνει $4^{**}(-1)$;
a) -4 b) 0.25 c) 0 d) 3

Γενικές παρατηρήσεις στους αριθμητικούς τελεστές

Κανόνες

- Τελεστές που δρουν μεταξύ ποσοτήτων **ίδιου τύπου** δίνουν αποτέλεσμα **αυτού του τύπου**.
- Τελεστές που δρουν μεταξύ ποσοτήτων **διαφορετικού τύπου** προκαλούν μετατροπή της **τιμής** της ποσότητας «χαμηλότερης» ακρίβειας στον τύπο με την «υψηλότερη» ακρίβεια.
Π.χ. πράξη μεταξύ ενός **DOUBLE PRECISION** και ενός **INTEGER** γίνεται αφού μετατραπεί ο **INTEGER** σε **DOUBLE PRECISION**.

Παρατηρήσεις

- Πόσο κάνει $4^{**}(-1)$;
a) -4 b) 0.25 c) 0 d) 3

Γενικές παρατηρήσεις στους αριθμητικούς τελεστές

Κανόνες

- Τελεστές που δρουν μεταξύ ποσοτήτων **ίδιου τύπου** δίνουν αποτέλεσμα **αυτού του τύπου**.
- Τελεστές που δρουν μεταξύ ποσοτήτων **διαφορετικού τύπου** προκαλούν μετατροπή της **τιμής** της ποσότητας «χαμηλότερης» ακρίβειας στον τύπο με την «υψηλότερη» ακρίβεια.
Π.χ. πράξη μεταξύ ενός **DOUBLE PRECISION** και ενός **INTEGER** γίνεται αφού μετατραπεί ο **INTEGER** σε **DOUBLE PRECISION**.

Παρατηρήσεις

- Πόσο κάνει $4^{**}(-1)$;
a) -4 b) 0.25 c) 0 d) 3
- Πώς **ΔΕΝ** γράφουμε το 10^{-6} ;

Γενικές παρατηρήσεις στους αριθμητικούς τελεστές

Κανόνες

- Τελεστές που δρουν μεταξύ ποσοτήτων **ίδιου τύπου** δίνουν αποτέλεσμα **αυτού του τύπου**.
- Τελεστές που δρουν μεταξύ ποσοτήτων **διαφορετικού τύπου** προκαλούν μετατροπή της **τιμής** της ποσότητας «χαμηλότερης» ακρίβειας στον τύπο με την «υψηλότερη» ακρίβεια.
Π.χ. πράξη μεταξύ ενός **DOUBLE PRECISION** και ενός **INTEGER** γίνεται αφού μετατραπεί ο **INTEGER** σε **DOUBLE PRECISION**.

Παρατηρήσεις

- Πόσο κάνει $4^{**}(-1)$;
a) -4 b) 0.25 c) 0 d) 3
- Πώς **ΔΕΝ** γράφουμε το 10^{-6} ; $10^{**}(-6) \rightarrow 0$

Μεταβλητή

Μεταβλητή είναι θέση στη μνήμη για αποθήκευση δεδομένων.

Κανόνας

Κάθε μεταβλητή προτού χρησιμοποιηθεί **πρέπει να δηλωθεί**. Οι δηλώσεις γίνονται στην αρχή του προγράμματος.

Προτού χρησιμοποιηθεί σε πράξεις **πρέπει να έχει τιμή**.

Μεταβλητή

Μεταβλητή είναι θέση στη μνήμη για αποθήκευση δεδομένων.

Κανόνας

Κάθε μεταβλητή προτού χρησιμοποιηθεί **πρέπει να δηλωθεί**. Οι δηλώσεις γίνονται στην αρχή του προγράμματος.

Προτού χρησιμοποιηθεί σε πράξεις **πρέπει να έχει τιμή**.

Δήλωση

τύπος :: όνομα_μεταβλητής

Μεταβλητή

Μεταβλητή είναι θέση στη μνήμη για αποθήκευση δεδομένων.

Κανόνας

Κάθε μεταβλητή προτού χρησιμοποιηθεί **πρέπει να δηλωθεί**. Οι δηλώσεις γίνονται στην αρχή του προγράμματος.

Προτού χρησιμοποιηθεί σε πράξεις **πρέπει να έχει τιμή**.

Δήλωση

τύπος :: όνομα_μεταβλητής

Παραδείγματα

INTEGER :: a

DOUBLE PRECISION :: b

Κανόνες σχηματισμού ονόματος

- Επιτρεπτοί χαρακτήρες: a-z, A-Z, 0-9, και _.
- Μήκος: το πολύ 31 χαρακτήρες.
- Δεν επιτρέπεται να αρχίζει από αριθμητικό ψηφίο.
- Κεφαλαία και πεζά γράμματα **είναι ίδια**.

Κανόνες σχηματισμού ονόματος

- Επιτρεπτοί χαρακτήρες: a-z, A-Z, 0-9, και _.
- Μήκος: το πολύ 31 χαρακτήρες.
- Δεν επιτρέπεται να αρχίζει από αριθμητικό ψηφίο.
- Κεφαλαία και πεζά γράμματα **είναι ίδια**.

Παραδείγματα

Μη αποδεκτά ονόματα: ena lathos onoma, άλφα, 1q, .onoma.

Αποδεκτά ονόματα: timi, a12, ena_onoma_me_megalo_mikos

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση]

- Το αριστερό μέλος είναι (υποχρεωτικά) μεταβλητή.

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση]

- Το αριστερό μέλος είναι (υποχρεωτικά) μεταβλητή.
- **Πρώτα** εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση]

- Το αριστερό μέλος είναι (υποχρεωτικά) μεταβλητή.
- **Πρώτα** εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.
- **Κατόπιν**, το αποτέλεσμα **μετατρέπεται** (αν χρειάζεται) στον τύπο της μεταβλητής του αριστερού μέλους και η τιμή που προκύπτει εκχωρείται σε αυτή.

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση]

- Το αριστερό μέλος είναι (υποχρεωτικά) μεταβλητή.
- **Πρώτα** εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.
- **Κατόπιν**, το αποτέλεσμα **μετατρέπεται** (αν χρειάζεται) στον τύπο της μεταβλητής του αριστερού μέλους και η τιμή που προκύπτει εκχωρείται σε αυτή.

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση]

- Το αριστερό μέλος είναι (υποχρεωτικά) μεταβλητή.
- **Πρώτα** εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.
- **Κατόπιν**, το αποτέλεσμα **μετατρέπεται** (αν χρειάζεται) στον τύπο της μεταβλητής του αριστερού μέλους και η τιμή που προκύπτει εκχωρείται σε αυτή.

Παραδείγματα

INTEGER :: a

REAL :: b

a = 2.5d0 * 3 ! a ← 7

b = 3.1415926535897d0 ! b ← 3.141593

Είσοδος/έξοδος δεδομένων

Ανάγνωση δεδομένων από το πληκτρολόγιο

```
READ *, a, b, c
```

ή, ισοδύναμα,

```
READ (*,*) a, b, c
```

όπου a, b, c μεταβλητές.

Είσοδος/έξοδος δεδομένων

Ανάγνωση δεδομένων από το πληκτρολόγιο

```
READ *, a, b, c
```

ή, ισοδύναμα,

```
READ (*,*) a, b, c
```

όπου a, b, c **μεταβλητές**.

Εκτύπωση δεδομένων στην οθόνη

```
PRINT *, "To apotelesma einai", a+2*b
```

ή, ισοδύναμα,

```
WRITE (*,*) "To apotelesma einai", a+2*b
```

Τυπώνουμε κείμενο, αποτέλεσμα πράξεων, κλπ.

Είσοδος/έξοδος δεδομένων

Ανάγνωση δεδομένων από το πληκτρολόγιο

```
READ *, a, b, c
```

ή, ισοδύναμα,

```
READ (*,*) a, b, c
```

όπου a, b, c **μεταβλητές**.

Εκτύπωση δεδομένων στην οθόνη

```
PRINT *, "To apotelesma einai", a+2*b
```

ή, ισοδύναμα,

```
WRITE (*,*) "To apotelesma einai", a+2*b
```

Τυπώνουμε κείμενο, αποτέλεσμα πράξεων, κλπ.

Παρατήρηση

Τι νόημα έχει ο κώδικας

```
READ *, a  
a = 4
```