

Το Εγχειρίδιο της ΓΛΩΣΣΑΣ

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Γ' Λυκείου

Περιεχόμενα

1	Οι τύποι δεδομένων της ΓΛΩΣΣΑΣ	2
1.1	Παρατηρήσεις	2
1.2	Συνοπτικός Πίνακας Τύπων Δεδομένων	3
2	Οι τελεστές της ΓΛΩΣΣΑΣ	3
2.1	Αριθμητικοί τελεστές	3
2.1.1	Συνοπτικός Πίνακας Αριθμητικών Τελεστών	4
2.2	Συγκριτικοί τελεστές	4
2.2.1	Συνοπτικός Πίνακας Συγκριτικών Τελεστών	4
2.3	Λογικοί τελεστές	4
2.3.1	Πίνακας Αληθείας	5
2.4	Παρατηρήσεις για τους τελεστές	5
3	Τα μέρη του προγράμματος	5
3.1	Παράδειγμα προγράμματος	5
3.2	Δομή προγράμματος	6
3.3	Παρατηρήσεις	6
4	Η εντολή ανάθεσης τιμής (<-)	7
4.1	Σύνταξη	7
4.2	Παράδειγμα	7
4.3	Παρατηρήσεις	7
5	Η εντολή ΓΡΑΨΕ	8
5.1	Σύνταξη	8
5.2	Παράδειγμα	8
5.3	Παρατηρήσεις	8
5.4	Παραδείγματα χρήσης	9
6	Η εντολή ΔΙΑΒΑΣΕ	9
6.1	Σύνταξη	9
6.2	Παράδειγμα	9
6.3	Παρατηρήσεις	9
6.4	Συνοπτικός πίνακας τύπων που διαβάζονται	10
7	Η εντολή ΑΝ	11
7.1	Σύνταξη	11
7.2	Παράδειγμα	11
7.3	Παρατηρήσεις	11
8	Η εντολή ΕΠΙΛΕΞΕ	12
8.1	Σύνταξη	12
8.2	Παράδειγμα	12

8.3	Παρατηρήσεις	13
8.4	Σύγκριση AN και ΕΠΙΛΕΞΕ	14
9	Η εντολή ΓΙΑ	14
9.1	Σύνταξη	14
9.2	Παράδειγμα	15
9.3	Παρατηρήσεις	15
9.4	Ισοδυναμία με ΟΣΟ	15
9.5	Διαφορές με άλλες γλώσσες	16
10	Η εντολή ΟΣΟ	16
10.1	Σύνταξη	16
10.2	Παράδειγμα	16
10.3	Παρατηρήσεις	16
10.4	Χρήση μετρητή στην ΟΣΟ	17
10.5	Σύγκριση εντολών επανάληψης	17
11	Η εντολή ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ	17
11.1	Σύνταξη	17
11.2	Παράδειγμα	17
11.3	Παρατηρήσεις	18
11.4	Χρήση μετρητή στην ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ	18
11.5	Ισοδυναμία με ΟΣΟ	18
12	Ενσωματωμένες συναρτήσεις	18
12.1	Λίστα συναρτήσεων	19
12.2	Συνοπτικός πίνακας	19
12.3	Παράδειγμα	19
12.4	Παρατηρήσεις	19
13	Η εντολή ΚΑΛΕΣΕ	20
13.1	Σύνταξη	20
13.2	Παράδειγμα	20
13.3	Παρατηρήσεις	21
13.4	Σύγκριση διαδικασιών και συναρτήσεων	21
14	Διαδικασίες	21
14.1	Σύνταξη	21
14.2	Παράδειγμα	21
14.3	Παρατηρήσεις	22
15	Συναρτήσεις	23
15.1	Σύνταξη	23
15.2	Παράδειγμα	23
15.3	Παρατηρήσεις	24
15.4	Σύγκριση διαδικασιών και συναρτήσεων	25
16	Μεταβίβαση παραμέτρων σε υποπρογράμματα	25
16.1	Παράδειγμα	25
16.2	Κανόνες μεταβίβασης	26
16.2.1	Μεταβίβαση με αναφορά (μεταβλητή ως παράμετρος)	26
16.2.2	Μεταβίβαση με τιμή (έκφραση ως παράμετρος)	26
16.2.3	Μετατροπή μεταβλητής σε έκφραση	26

16.3	Συνοπτικός πίνακας	26
16.4	Ο μηχανισμός copy in - copy out	26
16.4.1	Διαφορές copy in - copy out από αναφορά	26
17	Λέξεις-κλειδιά της ΓΛΩΣΣΑΣ	27
17.1	Δομή προγράμματος	27
17.2	Τύποι δεδομένων (δήλωση μεταβλητών)	28
17.3	Τύποι δεδομένων (επιστροφή συναρτήσεων)	28
17.4	Λογικές τιμές	28
17.5	Αριθμητικοί τελεστές	28
17.6	Συγκριτικοί τελεστές	28
17.7	Λογικοί τελεστές	28
17.8	Εντολές εισόδου/εξόδου	29
17.9	Εντολή ανάθεσης	29
17.10	Δομή επιλογής (ΑΝ)	29
17.11	Δομή επιλογής (ΕΠΙΛΕΞΕ)	29
17.12	Δομή επανάληψης (ΓΙΑ)	29
17.13	Δομή επανάληψης (ΟΣΟ)	29
17.14	Δομή επανάληψης (ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ)	29
17.15	Υποπρογράμματα	30
17.16	Ενσωματωμένες συναρτήσεις	30
17.17	Ειδικά σύμβολα	30
17.18	Συνοπτική λίστα όλων των λέξεων-κλειδιών	30

1 Οι τύποι δεδομένων της ΓΛΩΣΣΑΣ

Όταν λέμε **τύπο δεδομένων**, εννοούμε από ποιο σύνολο τιμών παίρνει τιμές μια έκφραση. Στη ΓΛΩΣΣΑ υπάρχουν τέσσερις τύποι δεδομένων:

- ο **ακέραιος**, π.χ. -1234567890
- ο **πραγματικός**, π.χ. 3.14
- οι **χαρακτήρες**, π.χ. "N" ή 'Μυρτώ'
- ο **λογικός**, δηλαδή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ

1.1 Παρατηρήσεις

1. **Όρια ακεραίων και πραγματικών:** Αν και προγραμματίζοντας στο χαρτί δεν αντιμετωπίζουμε πρόβλημα μεγέθους στους αριθμούς, στον υπολογιστή έχουμε πάντα περιορισμένο μέγεθος, όσο μεγάλο και αν είναι αυτό. Στην τρέχουσα υλοποίηση του Διερμηνευτή:
 - Οι **ακέραιοι** είναι από $-9.223.372.036.854.775.808$ μέχρι $9.223.372.036.854.775.807$ (64 bit προσημασμένος)
 - Οι **πραγματικοί** από 5.0×10^{-324} μέχρι 1.7×10^{308}
2. **Ακρίβεια πραγματικών:** Πρόβλημα μεγέθους υπάρχει και στους πολύ μικρούς πραγματικούς αριθμούς. Δηλαδή ενώ στο σύνολο των πραγματικών αριθμών δε νοείται ο επόμενος του 1, στον υπολογιστή υπάρχει πάντα ο επόμενος του 1, και στην τρέχουσα υλοποίηση του Διερμηνευτή είναι ο 5.0×10^{-324} . Έτσι αν για παράδειγμα προσπαθήσουμε να προσθέσουμε έναν πολύ μικρό αριθμό στο 1, το άθροισμα θα είναι πάλι 1! Όταν το αποτέλεσμα κάποιας πράξης είναι τόσο μικρός αριθμός που δεν μπορεί να αναπαρασταθεί από τον υπολογιστή, προκύπτει **λάθος εκτέλεσης**.
3. **Χαρακτήρες και αλφαριθμητικά:** Δεν υπάρχει διαφορετικός τύπος δεδομένων για ένα χαρακτήρα και για πολλούς χαρακτήρες (π.χ. `char` και `string`). Η σύγκριση μεταξύ των αλφαριθμητικών γίνεται σύμφωνα με τη διάταξη του προτύπου Unicode, και έτσι:
 - Τα κεφαλαία είναι διαφορετικά από τα πεζά
 - Τα τονισμένα διαφορετικά από τα άτονα
4. **Εισαγωγικά σε αλφαριθμητικά:** Τα αλφαριθμητικά επιτρέπεται να περικλείονται είτε με απλά είτε με διπλά εισαγωγικά. Για να συμπεριλάβουμε ένα απλό εισαγωγικό σε ένα αλφαριθμητικό πρέπει να γράψουμε δύο απλά, π.χ.:

Παράδειγμα εισαγωγικών

```
'Γιάννενα πρώτα στ' 'άρματα'  
ή  
"Γιάννενα πρώτα στ' άρματα"
```

Το αντίστροφο συμβαίνει για τα αλφαριθμητικά που περικλείονται από διπλά εισαγωγικά.

5. **Προβλήματα ακρίβειας σε επαναλήψεις:** Οι πραγματικοί αριθμοί λόγω της περιορισμένης ακρίβειας δε συμπεριφέρονται πάντα όπως θα θέλαμε. Για παράδειγμα αν σε οποιαδήποτε γλώσσα προγραμματισμού προσθέσουμε στο 1 τριάντα φορές το 0.30 δεν θα πάρουμε το 10 αλλά το 10.000000001.

Αυτό δημιουργεί προβλήματα σε εντολές επανάληψης, αφού μπορεί να γίνει μία επανάληψη λιγότερη ή περισσότερη. Ο Διερμηνευτής αντιμετωπίζει αυτό το πρόβλημα «περιορίζοντας» την ακρίβεια των πραγματικών σε συγκρίσεις.

Σημείωση

Για περισσότερες λεπτομέρειες δείτε την περιγραφή της επιλογής «Σημαντικά δεκαδικά ψηφία πραγματικών σε συγκρίσεις» στο αρχείο βοήθειας του Διερμηνευτή.

1.2 Συνοπτικός Πίνακας Τύπων Δεδομένων

Τύπος	Δήλωση	Παράδειγμα
Ακέραιος	ΑΚΕΡΑΙΕΣ:	-1234567890, 42
Πραγματικός	ΠΡΑΓΜΑΤΙΚΕΣ:	3.14, -0.5
Χαρακτήρες	ΧΑΡΑΚΤΗΡΕΣ:	'Μυρτώ', "N"
Λογικός	ΛΟΓΙΚΕΣ:	ΑΛΗΘΗΣ, ΨΕΥΔΗΣ

2 Οι τελεστές της ΓΛΩΣΣΑΣ

2.1 Αριθμητικοί τελεστές

- **+** (συν): Ισχύει για τους ακέραιους και τους πραγματικούς.

Σημείωση

Προαιρετικά και χωρίς να προτείνεται, μπορείτε να ενεργοποιήσετε τη σχετική επιλογή του Διερμηνευτή ώστε να επιτραπεί η συνένωση χαρακτήρων με τον τελεστή +.

- **-** (πλην): Ισχύει για τους ακέραιους και τους πραγματικούς. Υφίσταται σαν δυαδικός (π.χ. 1 - 2) και σαν μοναδιαίος (π.χ. -1).
- ***** (επί): Ισχύει για τους ακέραιους και τους πραγματικούς.
- **/** (διά): Ισχύει για τους ακέραιους και τους πραγματικούς. Το αποτέλεσμα της διαίρεσης είναι **πάντα πραγματικός**, ακόμα και αν δεν υπάρχουν δεκαδικά ψηφία (π.χ. 4/2).
- **^** (δύναμη): Ισχύει για τους ακέραιους και τους πραγματικούς. Το αποτέλεσμα είναι συνήθως πραγματικός αριθμός, εκτός αν ο εκθέτης είναι θετικός ακέραιος σταθερής αποτίμησης.

Παραδείγματα δύναμης

$2^2 \rightarrow$ ακέραιος (= 4)
 $2^{-2} \rightarrow$ πραγματικός (= 0.25)
 $2^{\iota} \rightarrow$ πραγματικός (γενική περίπτωση)

- **DIV** (πηλίκιο ακέραιας διαίρεσης): Ισχύει **μόνο** για τους ακέραιους.
Παράδειγμα: 7 DIV 2 = 3 (και περισσεύει 1)
- **MOD** (υπόλοιπο ακέραιας διαίρεσης): Ισχύει **μόνο** για τους ακέραιους.
Παράδειγμα: 7 MOD 2 = 1

2.1.1 Συνοπτικός Πίνακας Αριθμητικών Τελεστών

Τελεστής	Λειτουργία	Τύποι	Αποτέλεσμα
+	Πρόσθεση	Ακέραιοι, Πραγματικοί	Ανάλογα
-	Αφαίρεση	Ακέραιοι, Πραγματικοί	Ανάλογα
*	Πολλαπλασιασμός	Ακέραιοι, Πραγματικοί	Ανάλογα
/	Διαίρεση	Ακέραιοι, Πραγματικοί	Πραγματικός
^	Δύναμη	Ακέραιοι, Πραγματικοί	Συνήθως Πραγματικός
DIV	Ακέραια διαίρεση	Μόνο Ακέραιοι	Ακέραιος
MOD	Υπόλοιπο	Μόνο Ακέραιοι	Ακέραιος

2.2 Συγκριτικοί τελεστές

Γενικά οι συγκριτικοί τελεστές εφαρμόζονται μεταξύ των **ιδίων τύπων δεδομένων** (π.χ. χαρακτήρες >= χαρακτήρες) με εξαίρεση τους ακέραιους και τους πραγματικούς, που μπορεί να συγκρίνονται και μεταξύ τους (π.χ. ακέραιος = πραγματικός). Το αποτέλεσμα των συγκριτικών τελεστών είναι **πάντα λογικού τύπου**.

- **<= ή ≤ (μικρότερο ή ίσο):** Ισχύει για τους ακέραιους, τους πραγματικούς και τους χαρακτήρες.

Διάταξη χαρακτήρων (Unicode)

A (αγγλικό) < B < a < b < A < A (ελληνικό) < B < ά < α < β

- **< (μικρότερο):** Ισχύει για τους ακέραιους, τους πραγματικούς και τους χαρακτήρες.

Σημείωση για λογικές τιμές

Στις περισσότερες γλώσσες προγραμματισμού οι λογικές τιμές (ΑΛΗΘΗΣ και ΨΕΥΔΗΣ) έχουν διάταξη (ordinal). Το βιβλίο μαθητή αναφέρει ότι **δεν μπορεί** να εφαρμοστεί ανισότητα μεταξύ λογικών εκφράσεων.

- **= (ίσον):** Ισχύει για όλους τους τύπους δεδομένων.
- **<> ή ≠ (διάφορο):** Ισχύει για όλους τους τύπους δεδομένων.
- **> (μεγαλύτερο):** Ισχύει για τους ακέραιους, τους πραγματικούς και τους χαρακτήρες.
- **>= ή ≥ (μεγαλύτερο ή ίσο):** Ισχύει για τους ακέραιους, τους πραγματικούς και τους χαρακτήρες.

2.2.1 Συνοπτικός Πίνακας Συγκριτικών Τελεστών

Τελεστής	Unicode	Λειτουργία	Ακ.	Πραγ.	Χαρ.	Λογ.
<		Μικρότερο				
<=	≤	Μικρότερο ή ίσο				
=		Ίσον				
<>	≠	Διάφορο				
>		Μεγαλύτερο				
>=	≥	Μεγαλύτερο ή ίσο				

2.3 Λογικοί τελεστές

Οι λογικοί τελεστές (ΚΑΙ, Η, ΟΧΙ) εφαρμόζονται πάνω σε **λογικές εκφράσεις** (δηλαδή εκφράσεις που το αποτέλεσμά τους είναι ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ). Το αποτέλεσμά τους είναι πάλι λογικού τύπου δεδομένων.

2.3.1 Πίνακας Αληθείας

Στον παρακάτω πίνακα θεωρούμε ότι τα A και B είναι λογικές εκφράσεις (π.χ. $1 < 0$, $\beta = \pi$, ΑΛΗΘΗΣ).

A	B	A ΚΑΙ B	A Η B	ΟΧΙ A
ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ
ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	
ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ
ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	

- **ΚΑΙ (AND):** Επιστρέφει ΑΛΗΘΗΣ μόνο αν **και οι δύο** τελεστές είναι ΑΛΗΘΗΣ.
- **Η (OR):** Επιστρέφει ΑΛΗΘΗΣ αν **τουλάχιστον ένας** τελεστής είναι ΑΛΗΘΗΣ.
- **ΟΧΙ (NOT):** Αντιστρέφει τη λογική τιμή (μοναδιαίος τελεστής).

2.4 Παρατηρήσεις για τους τελεστές

1. **DIV και MOD με αρνητικούς:** Υπάρχει ασυμφωνία σχετικά με το αποτέλεσμα των DIV και MOD για αρνητικούς τελεστέους, καθώς και σχετική επιλογή του Διερμηνευτή για να προσαρμόσετε το αποτέλεσμα.

Οδηγία Παιδαγωγικού Ινστιτούτου (Φεβρουάριος 2008)

Δεν πρέπει να χρησιμοποιούνται σε ασκήσεις με αρνητικούς αριθμούς οι τελεστές DIV και MOD.

2. **Προσεταιριστικότητα δύναμης:** Υπάρχει ασάφεια σχετικά με το αν ο τελεστής ύψωσης σε δύναμη ^ εκτελείται από δεξιά προς τα αριστερά ή αντίθετα, καθώς και σχετική επιλογή στο διάλογο ρυθμίσεων του Διερμηνευτή.
3. **Προτεραιότητα ΚΑΙ/Η:** Υπάρχει ασάφεια σχετικά με το αν το ΚΑΙ έχει μεγαλύτερη ή ίση προτεραιότητα με το Η, καθώς και σχετική επιλογή του Διερμηνευτή.
4. **Unicode vs ANSI τελεστές:** Πριν την έκδοση 0.89 βήτα οι συγκριτικοί τελεστές μετατρέπονταν αυτόματα στους αντίστοιχους Unicode (\leq , \geq , \neq). Από την 0.89 βήτα και μετά υποστηρίζονται μεν οι Unicode, αλλά η **προεπιλεγμένη μορφή** των τελεστών είναι η ANSI ($>=$, $<=$, $<>$) για συμφωνία με το σχολικό βιβλίο.

3 Τα μέρη του προγράμματος

Μέρη του προγράμματος λέμε:

1. Την **επικεφαλίδα** (δηλαδή το όνομα)
2. Το **τμήμα δήλωσης σταθερών**
3. Το **τμήμα δήλωσης μεταβλητών**
4. Το **κυρίως σώμα** του προγράμματος

Στη συνέχεια βέβαια μπορεί να ακολουθούν και **υποπρογράμματα**. Τα τμήματα δηλώσεων είναι **προαιρετικά**.

3.1 Παράδειγμα προγράμματος

Ένα παράδειγμα προγράμματος που χρησιμοποιεί και τα τέσσερα τμήματα είναι το παρακάτω:

```

1 ΠΡΟΓΡΑΜΜΑ
2 ΕμβαδόνΚύκλουΣΤΑΘΕΡΕΣ
3
4 π = 3.14ΜΕΤΑΒΛΗΤΕΣ
5
6 ΠΡΑΓΜΑΤΙΚΕΣ: Εμ, ΑκΑΡΧΗ
7
8 ΓΡΑΨΕ Δώσε' την ακτίνα'
9 ΔΙΑΒΑΣΕ Ακ
10 Εμ <- πΑκ*^2
11 ΓΡΑΨΕ Το' εμβαδόν του κύκλου είναι ', ΕμΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
12 _

```

3.2 Δομή προγράμματος

Τμήμα	Λέξη-κλειδί	Υποχρεωτικό
Επικεφαλίδα	ΠΡΟΓΡΑΜΜΑ	Ναι
Σταθερές	ΣΤΑΘΕΡΕΣ	Όχι
Μεταβλητές	ΜΕΤΑΒΛΗΤΕΣ	Όχι
Κυρίως σώμα	ΑΡΧΗ ... ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ	Ναι

3.3 Παρατηρήσεις

1. **Αποδεκτά ονόματα αναγνωριστικών:** Τα ονόματα προγράμματος, σταθεράς ή μεταβλητής πρέπει να:

- Ξεκινούν με **γράμμα** (είτε ελληνικό είτε αγγλικό)
- Στη συνέχεια περιλαμβάνουν γράμματα, αριθμούς ή την κάτω παύλα (_)

Παραδείγματα αναγνωριστικών

Έγκυρα: Μεταβλητή1, x, Αρ_Μαθητών, counter

Μη έγκυρα: 1Μεταβλητή, Αρ-Μαθητών, my var

2. **Εκφράσεις στις σταθερές:** Στο τμήμα δήλωσης σταθερών επιτρέπεται και η χρήση τελεστών ή των ενσωματωμένων συναρτήσεων:

```

1 ΣΤΑΘΕΡΕΣ
2
3 N = 10
4 στ = 2 * TP_N()

```

3. **Πολλαπλές δηλώσεις τύπων:** Στο τμήμα δήλωσης μεταβλητών επιτρέπεται να χρησιμοποιούμε τους τύπους δεδομένων **όσες φορές θέλουμε**, δηλαδή μπορούμε για παράδειγμα να έχουμε δύο γραμμές με τη λέξη ΠΡΑΓΜΑΤΙΚΕΣ:

```

1 ΜΕΤΑΒΛΗΤΕΣ
2
3 ΠΡΑΓΜΑΤΙΚΕΣ: x, y
4 ΑΚΕΡΑΙΕΣ: i, j
5 ΠΡΑΓΜΑΤΙΚΕΣ: z ! Επιτρέπεται

```

4 Η εντολή ανάθεσης τιμής (<-)

Η εντολή ανάθεσης τιμής μπορεί να μεταφραστεί ως εξής:

Ερμηνεία εντολής ανάθεσης

«Υπολόγισε την τιμή της έκφρασης **δεξιά** από το σύμβολο ανάθεσης τιμής και βάλε το αποτέλεσμα στη **μεταβλητή** που βρίσκεται **αριστερά** από την ανάθεση τιμής.»

4.1 Σύνταξη

μεταβλητή <- έκφραση

4.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΑνάθεσηςΤιμήςΤο
3 ! πρόγραμμα αυτό δεν περιλαμβάνει Γράψε«». Για να δείτε τις τιμές
4 ! των μεταβλητών κοιτάξτε στην καρτέλα Μεταβλητές«.ΜΕΤΑΒΛΗΤΕΣ
5
6 ΑΚΕΡΑΙΕΣ: α
7 ΠΡΑΓΜΑΤΙΚΕΣ: π
8 ΧΑΡΑΚΤΗΡΕΣ: χ
9 ΛΟΓΙΚΕΣ: ΛΑΡΧΗ
10
11 α <- 1
12 π <- α2*           Επιτρέπεται! κι ας είναι ακέραια η έκφραση
13 χ <- Κείμενο' '
14 λ <- α >= π       Το! λ θα γίνει ΨευδήςΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
15 _
```

4.3 Παρατηρήσεις

1. Δεξί και αριστερό μέρος:

- Στο **δεξί μέρος** μπορεί να βρίσκεται οτιδήποτε έχει τιμή (έκφραση, σταθερά, μεταβλητή, κλήση συνάρτησης κ.λπ.)
- Στο **αριστερό μέρος** μπορεί να βρίσκεται μία και μόνο μεταβλητή (ή στοιχείο πίνακα) συμβατού τύπου δεδομένων με την έκφραση

2. Συμβατότητα τύπων: Η μόνη συμβατή ανάθεση μεταξύ διαφορετικών τύπων δεδομένων είναι από **ακέραιο** (δεξιά) σε **πραγματικό** (αριστερά).

Αριστερά (μεταβλητή)	Δεξιά (έκφραση)	Επιτρέπεται;
Ακέραιος	Ακέραιος	
Πραγματικός	Πραγματικός	
Πραγματικός	Ακέραιος	
Ακέραιος	Πραγματικός	×
Χαρακτήρες	Χαρακτήρες	
Λογικός	Λογικός	

3. Unicode vs ANSI: Πριν την έκδοση 0.90 βήτα ο τελεστής ανάθεσης τιμής μετατρεπόταν αυτόματα στη Unicode μορφή του (<—). Πλέον υποστηρίζεται και το βελάκι, αλλά η **προεπιλεγμένη μορφή** είναι η ANSI (<-) για συμφωνία με το σχολικό βιβλίο.

Σημείωση

Αυτή η συμπεριφορά μπορεί να προσαρμοστεί από το μενού **Εργαλεία** → **Επιλογές** του Διερμηνευτή.

5 Η εντολή ΓΡΑΨΕ

Η εντολή ΓΡΑΨΕ εμφανίζει στην οθόνη τις τιμές μιας λίστας εκφράσεων, χωρισμένων από κόμματα.

5.1 Σύνταξη

ΓΡΑΨΕ έκφραση1, έκφραση2, ..., έκφρασηN

5.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΓράψεΕμφανίζει
3 ! «10 3.00 κείμενο ΨΕΥΔΗΣ»ΜΕΤΑΒΛΗΤΕΣ
4
5 ΑΚΕΡΑΙΕΣ: αΑΡΧΗ
6
7 α <- 10
8 ΓΡΑΨΕ α, ' ', TR_α( - 1), ' κείμενο ', 1 = 3ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
9 _
```

5.3 Παρατηρήσεις

1. **Τύποι εκφράσεων:** Οι εκφράσεις μπορεί να είναι **οποιουδήποτε τύπου** δεδομένων. Αν είναι λογικού τύπου, στην οθόνη εμφανίζεται ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ.
2. **ΓΡΑΨΕ χωρίς παραμέτρους:** Επιτρέπεται εντολή ΓΡΑΨΕ χωρίς παραμέτρους. Απλά κατεβάζει κατά μία γραμμή το δρομέα (σελ. 180 του βιβλίου).

```
1 ΓΡΑΨΕ
2 ! Αλλαγή γραμμής
```

3. **Κενά μεταξύ τιμών:** Για ευκολία σε εισαγωγικές δραστηριότητες των μαθητών, μια εντολή ΓΡΑΨΕ 1, 2, 3 αφήνει κενά μεταξύ των εμφανιζόμενων αριθμών (δηλαδή εμφανίζει 1 2 3 αντί για 123).

Σημείωση

Αυτή η συμπεριφορά μπορεί να απενεργοποιηθεί από το μενού **Εργαλεία** → **Επιλογές**.

4. **Παραμονή δρομέα στην ίδια γραμμή:** Στη ΓΛΩΣΣΑ δεν υπάρχει εντολή αντίστοιχη της write της Pascal, η οποία να αφήνει το δρομέα στην ίδια γραμμή. Η λύση που υλοποιήθηκε είναι η εξής:

Κανόνες για τη θέση του δρομέα

Αφού αποτιμηθεί όλο το κείμενο που προκύπτει από τη ΓΡΑΨΕ, αν ο **τελευταίος χαρακτήρας** του είναι το κενό, τότε αυτή μεταφράζεται σε write (δρομέας στην ίδια γραμμή).

Εντολή	Αποτέλεσμα
ΓΡΑΨΕ 'Δώσε το όνομά σου: '	Δρομέας στην ίδια γραμμή
ΓΡΑΨΕ 'Δώσε το όνομά σου: '	Δρομέας στην επόμενη γραμμή

Προσοχή

Επειδή μερικές φορές χρειάζεται να μην υπάρχει κενό στο τέλος μίας ΓΡΑΨΕ, το τελευταίο κενό **διαγράφεται** και δεν εμφανίζεται στην οθόνη εκτέλεσης. Γι' αυτό το λόγο, αν θέλετε να εμφανιστεί ένα κενό στο τέλος, πρέπει να βάλετε **δύο κενά**.

Μπορείτε να απενεργοποιήσετε αυτήν τη συμπεριφορά από το μενού **Εργαλεία** → **Επιλογές**.

5.4 Παραδείγματα χρήσης

1 ΓΡΑΨΕ	
2 Καλημέρα '! '	! Εμφάνιση κειμένουΓΡΑΨΕ
3 'x = ', x	! Εμφάνιση μεταβλητήςΓΡΑΨΕ
4 α, β, γ	! Εμφάνιση πολλών τιμώνΓΡΑΨΕ
5 Αποτέλεσμα: ', α > β	! Εμφάνιση λογικής τιμήςΓΡΑΨΕ
6 Δώσε' τιμή: '	! Δρομέας στην ίδια γραμμή

6 Η εντολή ΔΙΑΒΑΣΕ

Η εντολή ΔΙΑΒΑΣΕ διαβάζει από το πληκτρολόγιο μία λίστα μεταβλητών, χωρισμένων από κόμματα. Για να το επιτύχει αυτό, σε κάθε ΔΙΑΒΑΣΕ η αντίστοιχη γραμμή της καρτέλας «Οθόνη εκτέλεσης» χρωματίζεται για να προσελκύσει την προσοχή του χρήστη και η εκτέλεση του προγράμματος παύει μέχρι να εισαχθούν τα δεδομένα και να πατηθεί το [Enter].

Οι μεταβλητές αυτές μπορεί να είναι οποιουδήποτε τύπου δεδομένων, **εκτός από λογικού**.

6.1 Σύνταξη

ΔΙΑΒΑΣΕ μεταβλητή1, μεταβλητή2, ..., μεταβλητήN

6.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΔΙΑΒΑΣΕΔιαβάζει
3 ! μία ακέραια μεταβλητή, μία πραγματική και μίααλφαριθμητική
4 !. Οι λογικές δεν είναι δυνατόν να διαβαστούν.Αντίθετα
5 ! με τις περισσότερες γλώσσες προγραμματισμού, ότανμία
6 ! ΔΙΑΒΑΣΕ έχει πολλά ορίσματα, αυτά πρέπει να εισαχθούνσε
7 ! ξεχωριστές γραμμές με( [Enter] ανάμεσα από τα ορίσματα).ΜΕΤΑΒΛΗΤΕΣ
8
9 ΑΚΕΡΑΙΕΣ: α
10 ΠΡΑΓΜΑΤΙΚΕΣ: Π[10]
11 ΧΑΡΑΚΤΗΡΕΣ: ΚΑΡΧΗ
12
13 ΔΙΑΒΑΣΕ α, Π[23 DIV 3]      διαβάζει! το α και το Π[7]
14 ΔΙΑΒΑΣΕ ΚΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
15 _
```

6.3 Παρατηρήσεις

1. **Μόνο μεταβλητές:** Δεν επιτρέπεται να διαβαστεί κάτι που δεν είναι μεταβλητή. Για παράδειγμα, η εντολή:

```

1 ΔΙΑΒΑΣΕ
2 α + β, π-, 24, κείμενο'' ! ΛΑΘΟΣ!

```

είναι τελείως λάθος.

2. **Όχι λογικές μεταβλητές:** Δεν είναι δυνατό να διαβαστούν λογικές μεταβλητές. Αντί αυτού μπορούμε να διαβάσουμε έναν ακέραιο ή ένα χαρακτήρα και στη συνέχεια να δώσουμε την αντίστοιχη τιμή στη λογική μεταβλητή με μια εντολή **AN**:

```

1 ΠΡΟΓΡΑΜΜΑ
2 ΔιάβασμαΛογικήςΜεταβλητήςΜΕΤΑΒΛΗΤΕΣ
3
4 ΑΚΕΡΑΙΕΣ: ι
5 ΛΟΓΙΚΕΣ: φύλοΑΡΧΗ
6
7 ΓΡΑΨΕ Δώσε' 0 αν είσαι γυναίκα και 1 αν είσαι άντρας: '
8 ΔΙΑΒΑΣΕ ι
9 AN ι = 0 ΤΟΤΕ
10     φύλο <- ΑΛΗΘΗΣ
11 ΑΛΛΙΩΣ
12     φύλο <- ΨΕΥΔΗΣ
13 ΤΕΛΟΣΑΝ_
14 Η! παραπάνω εντολή Αν ισοδυναμεί με την εντολή φύλο« <- ι = 0»
15 ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
16 _

```

3. **Έλεγχος δεδομένων κατά την είσοδο:** Στο Διερμηνευτή υπάρχει η επιλογή να ελέγχονται τα δεδομένα κατά την είσοδο. Αυτό σημαίνει ότι αν το πρόγραμμα προσπαθεί να διαβάσει:
- έναν **ακέραιο**: η περιοχή εισόδου επιτρέπει μόνο νούμερα
 - έναν **πραγματικό**: επιτρέπει νούμερα και μία τελεία
4. **Ξεχωριστές γραμμές για κάθε τιμή:** Σε εντολές του τύπου **ΔΙΑΒΑΣΕ χ, ψ, ω** δεν επιτρέπεται να εισαχθούν όλα τα δεδομένα σε μία γραμμή, αλλά πρέπει κάθε τιμή να εισάγεται χωριστά πατώντας ισάριθμες φορές το [Enter] (αντίθετα από τις περισσότερες γλώσσες προγραμματισμού).

Γιατί αυτή η συμπεριφορά;

Αυτό γίνεται για ευκολία των μαθητών, και για να μη δημιουργηθούν προβλήματα: για παράδειγμα η παράλληλη χρήση των **read** και **readln** της Pascal μπερδεύει εύκολα ακόμα και πεπειραμένους προγραμματιστές, ενώ στην C η **scanf** δε μπορεί να διαβάσει αλφαριθμητικά που να περιέχουν κενά.

Παράδειγμα προβλήματος

Σε μια εντολή **ΔΙΑΒΑΣΕ όνομα, ηλικία** αν ο χρήστης έδινε «Άλκης Γεωργόπουλος 35» που θα έπρεπε να σταματήσει το διάβασμα του αλφαριθμητικού;

6.4 Συνοπτικός πίνακας τύπων που διαβάζονται

Τύπος	Διαβάζεται;	Σημείωση
Ακέραιος		Μόνο αριθμοί
Πραγματικός		Αριθμοί και τελεία
Χαρακτήρες		Οτιδήποτε
Λογικός	×	Χρήση AN για μετατροπή

7 Η εντολή AN

Η εντολή AN χρησιμοποιείται όταν χρειάζεται **διακλάδωση** της ροής του προγράμματος, ανάλογα με κάποια συνθήκη.

7.1 Σύνταξη

```
AN συνθήκη ΤΟΤΕ
    εντολές
ΑΛΛΙΩΣ_ΑΝ συνθήκη2 ΤΟΤΕ
    εντολές
ΑΛΛΙΩΣ
    εντολές
ΤΕΛΟΣ_ΑΝ
```

Σημείωση

Τα τμήματα ΑΛΛΙΩΣ_ΑΝ και ΑΛΛΙΩΣ είναι **προαιρετικά**. Μπορούν να υπάρχουν πολλαπλά ΑΛΛΙΩΣ_ΑΝ, αλλά το πολύ ένα ΑΛΛΙΩΣ.

7.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΑνΔιαβάζει
3 ! ένα βαθμό και εμφανίζει στο
4 ! χρήστη αν πέρασε ή αν κόπηκε.ΜΕΤΑΒΛΗΤΕΣ
5
6 ΠΡΑΓΜΑΤΙΚΕΣ: βαθμόςΑΡΧΗ
7
8 ΓΡΑΨΕ Δώσε' βαθμό: '
9 ΔΙΑΒΑΣΕ βαθμός
10 ΑΝ βαθμός < 0 ΤΟΤΕ
11     ΓΡΑΨΕ Δεν' υπάρχει αρνητικός βαθμός'
12 ΑΛΛΙΩΣΑΝ_ βαθμός < 10 ΤΟΤΕ
13     ΓΡΑΨΕ Κόπηκες''
14 ΑΛΛΙΩΣΑΝ_ βαθμός <= 20 ΤΟΤΕ
15     ΓΡΑΨΕ Πέρασες''
16 ΑΛΛΙΩΣ
17     ΓΡΑΨΕ Ο' βαθμός πρέπει να είναι στην εικοσαβάθμια,',
18     & ' όχι στην εκατοσταβάθμια κλίμακα'
19 ΤΕΛΟΣΑΝ_ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
20 _
```

7.3 Παρατηρήσεις

1. **ΑΛΛΙΩΣ_ΑΝ vs ΑΛΛΙΩΣ ΑΝ:** Η εντολή ΑΛΛΙΩΣ_ΑΝ είναι **διαφορετική** από την ΑΛΛΙΩΣ ΑΝ.

- Η ΑΛΛΙΩΣ_ΑΝ είναι **μία εντολή** (else-if)
- Η ΑΛΛΙΩΣ ΑΝ είναι **δύο εντολές**: το μέρος ΑΛΛΙΩΣ μιας εντολής ΑΝ και μία δεύτερη ΑΝ στη συνέχεια

Προσοχή

Στην περίπτωση **ΑΛΛΙΩΣ ΑΝ**, η δεύτερη **ΑΝ** **δεν επιτρέπεται** να βρίσκεται στην ίδια γραμμή με το **ΑΛΛΙΩΣ** — πρέπει να γραφεί στην επόμενη γραμμή.

2. Αριθμός ΤΕΛΟΣ_ΑΝ:

- Με **ΑΛΛΙΩΣ_ΑΝ**: χρειάζεται **ένα** **ΤΕΛΟΣ_ΑΝ**
- Με **ΑΛΛΙΩΣ ΑΝ**: χρειάζονται **τόσα** **ΤΕΛΟΣ_ΑΝ** όσα και τα **ΑΝ**

Σύγκριση δομών

Με **ΑΛΛΙΩΣ_ΑΝ** (1 **ΤΕΛΟΣ_ΑΝ**):

```
ΑΝ
  x < 0 ΤΟΤΕ
  ...ΑΛΛΙΩΣΑΝ
_  x < 10 ΤΟΤΕ
_  ...ΤΕΛΟΣΑΝ
_
```

Με **ΑΛΛΙΩΣ ΑΝ** (2 **ΤΕΛΟΣ_ΑΝ**):

```
ΑΝ
  x < 0 ΤΟΤΕ
  ...ΑΛΛΙΩΣ

  ΑΝ x < 10 ΤΟΤΕ
  ...
  ΤΕΛΟΣΑΝ_ΤΕΛΟΣΑΝ
_
```

8 Η εντολή **ΕΠΙΛΕΞΕ**

Η εντολή **ΕΠΙΛΕΞΕ** χρησιμοποιείται αντί για πολλές **ΑΝ ... ΑΛΛΙΩΣ_ΑΝ ...** όταν θέλουμε να πάρουμε περιπτώσεις ανάλογα με την τιμή **μίας μόνο έκφρασης**. Δηλαδή δεν μπορεί να αντικαταστήσει πολλές **ΑΝ** οι οποίες αναφέρονται σε διαφορετικές συνθήκες.

8.1 Σύνταξη

```
ΕΠΙΛΕΞΕ έκφραση
  ΠΕΡΙΠΤΩΣΗ τιμή1
  εντολές
  ΠΕΡΙΠΤΩΣΗ τιμή2, τιμή3..τιμή4
  εντολές
  ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
  εντολές
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
```

8.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΕΠΙΛΕΞΕΜΕΤΑΒΛΗΤΕΣ
3
4 ΠΡΑΓΜΑΤΙΚΕΣ: βαθμόςΑΡΧΗ
```

```

5  ΓΡΑΨΕ Δώσε' βαθμό:  '
6  ΔΙΑΒΑΣΕ βαθμός
7  ΕΠΙΛΕΞΕ βαθμός
8
9  ΠΕΡΙΠΤΩΣΗ 15..20                                !15 <= βαθμός <= 20
10   ΓΡΑΨΕ Είσαι' άξιος'
11   ΠΕΡΙΠΤΩΣΗ 10..15                                !10 <= βαθμός < 15
12   ΓΡΑΨΕ Καλούτσικα' πήγες'
13   ΠΕΡΙΠΤΩΣΗ 0                                     βαθμός! = 0
14   ΓΡΑΨΕ Δεν' προσήλθες στις εξετάσεις'
15   ΠΕΡΙΠΤΩΣΗ 0..10                                !0 < βαθμός < 10
16   ΓΡΑΨΕ Κόπηκες' '
17   ΠΕΡΙΠΤΩΣΗ > 20
18   ΓΡΑΨΕ Ο' βαθμός θα πρέπει να είναι στην εικοσαβάθμια, ',
19   & όχι' στην εκατοσταβάθμια κλίμακα'
20   ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
21   ΓΡΑΨΕ Έδωσες' άκυρο βαθμό'
22 ΤΕΛΟΣΕΠΙΛΟΓΩΝ_ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
23 _

```

8.3 Παρατηρήσεις

1. **Τύπος έκφρασης:** Η έκφραση στην ΕΠΙΛΕΞΕ μπορεί να είναι **οποιοιδήποτε τύπου** δεδομένων.

Παραδείγματα από το σχολικό εγχειρίδιο

- Με **πραγματικούς αριθμούς**: σελίδα 75 του τετραδίου μαθητή
- Με **αλφαριθμητικά**: σελίδα 269 του βιβλίου μαθητή

2. **Συγκριτικοί τελεστές στις περιπτώσεις:** Επιτρέπεται η χρήση συγκριτικών τελεστών στις περιπτώσεις της ΕΠΙΛΕΞΕ, για παράδειγμα ΠΕΡΙΠΤΩΣΗ > 20. Σχετικό παράδειγμα υπάρχει στη σελίδα 75 του τετραδίου μαθητή.

Προσοχή

Δεν επιτρέπεται η χρήση **λογικών τελεστών** στις περιπτώσεις. Δηλαδή η εντολή:
ΠΕΡΙΠΤΩΣΗ > 20 ΚΑΙ < 40
δεν είναι αποδεκτή.

3. **Σειρά περιπτώσεων με πραγματικούς:** Προσέξτε τη σειρά των περιπτώσεων όταν έχουμε πραγματικούς αριθμούς!

Στο παράδειγμα, ο βαθμός θεωρείται πραγματικός, οπότε μια τιμή όπως 9.5 είναι αποδεκτή και πρέπει να εμφανίσει «Κόπηκες». Αν όμως η περίπτωση 0..10 προηγούταν της 10..15, τότε το 10 θα πήγαινε στην πρώτη περίπτωση και το αποτέλεσμα θα ήταν λάθος.

Κανόνας για πραγματικούς

Στην ΕΠΙΛΕΞΕ, όταν έχουμε να κάνουμε με **πραγματικούς**, πρέπει να προηγούνται τα **κλειστά διαστήματα** (π.χ. [10, 15)) και μετά τα **ανοιχτά** (π.χ. (0, 10)).

4. **Πολλαπλές εκφράσεις ανά περίπτωση:** Η κάθε ΠΕΡΙΠΤΩΣΗ μπορεί να περιέχει πολλές εκφράσεις χωρισμένες με κόμματα:

```

ΠΕΡΙΠΤΩΣΗ
  TP_N(/2) .. α, β + 25, 10 .. 9

```

Η παραπάνω περίπτωση ισοδυναμεί με την εντολή:

ΑΝ

επιλογέας(\geq TP_N(/2) ΚΑΙ επιλογέας \leq α) Η
επιλογέας(= β + 25) Η
επιλογέας(\geq 10 ΚΑΙ επιλογέας \leq 9) ΤΟΤΕ ...

Ανάλυση

- T_P(N/2) .. α: διάστημα τιμών
- β + 25: συγκεκριμένη τιμή (όχι διάστημα)
- 10 .. 9: κενό διάστημα (το κάτω άκρο > πάνω άκρο)

5. **ΕΠΙΛΕΞΕ χωρίς περιπτώσεις:** Από τον Διερμηνευτή της ΓΛΩΣΣΑΣ:

- **Αποδεκτή:** ΕΠΙΛΕΞΕ που περιέχει μόνο ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
- **Μη αποδεκτή:** ΕΠΙΛΕΞΕ χωρίς καμία ΠΕΡΙΠΤΩΣΗ

6. **Αποτίμηση έκφρασης:** Η έκφραση στην ΕΠΙΛΕΞΕ αποτιμάται **μία φορά**, όχι διαδοχικά σε κάθε περίπτωση.

Σημασία για συναρτήσεις

Αν η έκφραση περιέχει κλήση σε συνάρτηση, η συνάρτηση θα κληθεί **μόνο μία φορά** κατά την ΕΠΙΛΕΞΕ και όχι σε κάθε ΠΕΡΙΠΤΩΣΗ.

8.4 Σύγκριση ΑΝ και ΕΠΙΛΕΞΕ

Χαρακτηριστικό	ΑΝ	ΕΠΙΛΕΞΕ
Διαφορετικές συνθήκες		×
Μία έκφραση, πολλές τιμές		
Διαστήματα τιμών	Με λογικούς τελεστές	Με ..
Συγκριτικοί τελεστές		
Λογικοί τελεστές		×

9 Η εντολή ΓΙΑ

Η εντολή ΓΙΑ είναι η πιο απλή εντολή επανάληψης. Χρησιμοποιεί ένα **μετρητή** για να μετράει πόσες επαναλήψεις γίνονται. Επιλέγουμε να τη χρησιμοποιήσουμε όταν **γνωρίζουμε από πριν** πόσες επαναλήψεις θέλουμε.

9.1 Σύνταξη

ΓΙΑ μετρητής ΑΠΟ αρχή ΜΕΧΡΙ τέλος ΜΕ_ΒΗΜΑ βήμα
εντολές
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Σημείωση

Αν το ΜΕ_ΒΗΜΑ παραληφθεί, το βήμα θεωρείται ίσο με 1.

9.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΓιαΜΕΤΑΒΛΗΤΕΣ
3
4 ΠΡΑΓΜΑΤΙΚΕΣ: ΠΑΡΧΗ
5
6 ΓΙΑ π ΑΠΟ 10 ΜΕΧΡΙ 1 ΜΕΒΗΜΑ_ -0.5
7   ΓΡΑΨΕ π
8 ΤΕΛΟΣΕΠΑΝΑΛΗΨΗΣ_ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
9 _
```

9.3 Παρατηρήσεις

1. **ΜΕ_ΒΗΜΑ vs ΜΕ ΒΗΜΑ:** Στο βιβλίο μαθητή το μέρος ΜΕ ΒΗΜΑ της εντολής ΓΙΑ αναφέρεται τέσσερις φορές **χωρίς κάτω παύλα**, ενώ σε όλες τις υπόλοιπες αναφορές των σχολικών εγχειριδίων είναι **με κάτω παύλα**.

Συμπεριφορά Διερμηνευτή

Ο Διερμηνευτής θεωρεί και τα δύο αποδεκτά, εκτός εάν αλλάξετε τη σχετική επιλογή από το μενού **Εργαλεία** → **Επιλογές**.

2. **Τύποι εκφράσεων:** Στα μέρη ΑΠΟ, ΜΕΧΡΙ και ΜΕ_ΒΗΜΑ μπορούν να χρησιμοποιηθούν **ακέραιες** ή **πραγματικές** εκφράσεις.

Προσοχή

Αν ο μετρητής είναι **ακέραιος**, δεν επιτρέπεται πραγματική τιμή στο ΑΠΟ ή στο ΜΕ_ΒΗΜΑ.

3. **Αποτίμηση εκφράσεων:** Οι εκφράσεις στα ΑΠΟ, ΜΕΧΡΙ και ΜΕ_ΒΗΜΑ αποτιμούνται **μία μόνο φορά**. Αν περιέχουν κλήση συνάρτησης, αυτή θα κληθεί μία φορά.
4. **Τιμή μετρητή μετά το τέλος:** Μετά το τέλος της επανάληψης, ο μετρητής έχει τιμή **μία θέση πέρα** από την τελική τιμή.

Παράδειγμα

Μετά από ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10, το i είναι **11** (όχι 10).

5. **Όταν δεν γίνεται καμία επανάληψη:** Αν δεν πρόκειται να γίνει καμία επανάληψη (π.χ. ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 0), ο μετρητής **παίρνει την αρχική τιμή** έτσι κι αλλιώς.

9.4 Ισοδυναμία με ΟΣΟ

Η μετάφραση της ΓΙΑ με χρήση της ΟΣΟ είναι:

```
μετρητής
<- έκφρασηαρχής_τιμήτέλους
_ <- έκφρασητέλους_ ! Για να μη γίνεται πολλέςτιμήβήματος
_ <- έκφρασηβήματος_ ! φορές η αποτίμησηΟΣΟ
μετρητής( >= ή <= τιμήτέλους_) ΕΠΑΝΑΛΑΒΕ
  εντολές
μετρητής <- μετρητής + τιμήβήματος_ΤΕΛΟΣΕΠΑΝΑΛΗΨΗΣ
_
```

Επιλογή τελεστή σύγκρισης

- Αν τιμή_βήματος > 0: χρησιμοποιούμε <= (μικρότερο ή ίσο)
- Αν τιμή_βήματος < 0: χρησιμοποιούμε >= (μεγαλύτερο ή ίσο)

9.5 Διαφορές με άλλες γλώσσες

Η ΓΙΑ είναι η εντολή με τις πιο μεγάλες διαφορές στην υλοποίησή της στις γλώσσες προγραμματισμού:

Χαρακτηριστικό	ΓΛΩΣΣΑ	Pascal	Basic
Βήμα διαφορετικό από 1		× (μόνο -1)	
Τιμή μετρητή μετά for i := 1 to 10	11	10	11
Αρχική τιμή αν 0 επαναλήψεις	Ναι	Όχι	Ναι

10 Η εντολή ΟΣΟ

Η εντολή ΟΣΟ είναι η πιο **ισχυρή** εντολή επανάληψης. Χρησιμοποιείται συνήθως όταν **δεν ξέρουμε** τον αριθμό των επαναλήψεων, αλλά οι επαναλήψεις εξαρτώνται από κάποια συνθήκη.

Πότε επιλέγουμε την ΟΣΟ

Την προτιμούμε από την ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ όταν είναι πιθανό να **μη γίνει καμία επανάληψη**, γιατί στην ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ γίνεται πάντα τουλάχιστον μία.

10.1 Σύνταξη

ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ
εντολές
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

10.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2   ΗΕντολήΌσοΜετράει
3   ! με( χαζό τρόπο) πόσες φορές μπορεί να
4   ! διαιρεθεί διαδοχικά ένας αριθμός με το 2.ΜΕΤΑΒΛΗΤΕΣ
5
6   ΑΚΕΡΑΙΕΣ: ι, ΑΡΧΗ
7
8   ΔΙΑΒΑΣΕ α
9   ι <- 0
10  ΟΣΟ α >= 2 ΕΠΑΝΑΛΑΒΕ
11    α <- α DIV 2
12    ι <- ι + 1
13  ΤΕΛΟΣΕΠΑΝΑΛΗΨΗΣ_
14  ΓΡΑΨΕ Ο' αριθμός μπορεί να διαιρεθεί ', ι, ' φορές με το 2.'ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
15 _
```

10.3 Παρατηρήσεις

1. **Έλεγχος συνθήκης:** Η συνθήκη ελέγχεται **πριν** από κάθε επανάληψη. Αν είναι **ΨΕΥΔΗΣ** από την αρχή, δεν εκτελείται καμία επανάληψη.
2. **Χωρίς ενσωματωμένο μετρητή:** Η ΟΣΟ δεν περιλαμβάνει μετρητή όπως η ΓΙΑ.

10.4 Χρήση μετρητή στην ΟΣΟ

Αν θέλετε να βάλετε μετρητή στην ΟΣΟ, θυμηθείτε τα **τρία βήματα** του μετρητή:

Βήμα	Περιγραφή	Παράδειγμα
1	Αρχικοποίηση πριν την ΟΣΟ	<code>ι <- 1</code>
2	Μεταβολή πριν το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	<code>ι <- ι + 1</code>
3	Συνθήκη τέλους στην ΟΣΟ	<code>ΟΣΟ ι <= 10</code>

```

ι
<- 1                      ! 1. Αρχικοποίηση ΟΣΟ
ι <= 10 ΕΠΑΝΑΛΑΒΕ        ! 3. Συνθήκη τέλους
! εντολές...
ι <- ι + 1                ! 2. Μεταβολή ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
—

```

10.5 Σύγκριση εντολών επανάληψης

Χαρακτηριστικό	ΓΙΑ	ΟΣΟ	ΑΡΧΗ_ΕΠΑΝ.
Γνωστός αριθμός επαναλήψεων			
Ενσωματωμένος μετρητής		×	×
Έλεγχος στην αρχή			×
Έλεγχος στο τέλος	×	×	
Ελάχιστες επαναλήψεις	0	0	1

11 Η εντολή ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

Η εντολή ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ χρησιμοποιείται όταν δεν υπάρχει συγκεκριμένος αριθμός επαναλήψεων, αλλά οι επαναλήψεις εξαρτώνται από κάποια συνθήκη. Είναι η αντίστοιχη της `repeat - until` της Pascal και της `do - while` στην Basic και στη C.

Πότε επιλέγουμε την ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

Την προτιμούμε από την ΟΣΟ όταν θέλουμε να γίνει **τουλάχιστον μία** επανάληψη.

11.1 Σύνταξη

```

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  εντολές
ΜΕΧΡΙΣ_ΟΤΟΥ συνθήκη

```

11.2 Παράδειγμα

```

1 ΠΡΟΓΡΑΜΜΑ
2   ΗΕντολήΑΡΧΗΕΠΑΝΑΛΗΨΗΣ_Διαβάζει
3   ! έναν ακέραιο υποχρεώνοντας το χρήστη
4   ! να είναι μεταξύ 0 και 20. ΜΕΤΑΒΛΗΤΕΣ
5
6   ΑΚΕΡΑΙΕΣ: βαθμός ΑΡΧΗ
7
8   ΑΡΧΗΕΠΑΝΑΛΗΨΗΣ_
9     ΓΡΑΨΕ Δώσε' βαθμό: '
10    ΔΙΑΒΑΣΕ βαθμός
11    ΜΕΧΡΙΣ_ΟΤΟΥ_ βαθμός >= 0 ΚΑΙ βαθμός <= 20
12    ΓΡΑΨΕ Έδωσες' ', βαθμός ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

```

11.3 Παρατηρήσεις

1. **Έλεγχος συνθήκης:** Η συνθήκη ελέγχεται **μετά** από κάθε επανάληψη. Έτσι, οι εντολές μέσα στο βρόχο εκτελούνται **τουλάχιστον μία φορά**.
2. **Συνθήκη τερματισμού:** Η επανάληψη σταματά όταν η συνθήκη γίνει **ΑΛΗΘΗΣ** (αντίθετα από την **ΟΣΟ** που συνεχίζει όσο είναι **ΑΛΗΘΗΣ**).

Διαφορά από ΟΣΟ

- **ΟΣΟ:** Επαναλαμβάνει **όσο** η συνθήκη είναι **ΑΛΗΘΗΣ**
- **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ:** Επαναλαμβάνει **μέχρι** η συνθήκη να γίνει **ΑΛΗΘΗΣ**

3. **Χωρίς ενσωματωμένο μετρητή:** Η **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** δεν περιλαμβάνει μετρητή όπως η **ΓΙΑ**.

11.4 Χρήση μετρητή στην ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

Αν θέλετε να βάλετε μετρητή στην **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ**, θυμηθείτε τα **τρία βήματα** του μετρητή:

Βήμα	Περιγραφή	Παράδειγμα
1	Αρχικοποίηση πριν την ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ	<code>ι <- 1</code>
2	Μεταβολή πριν το ΜΕΧΡΙΣ_ΟΤΟΥ	<code>ι <- ι + 1</code>
3	Συνθήκη τέλους στη ΜΕΧΡΙΣ_ΟΤΟΥ	<code>ΜΕΧΡΙΣ_ΟΤΟΥ ι = 10</code>

```
ι
<- 1                                ! 1. Αρχικοποίηση ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
! εντολές...
ι <- ι + 1                          ! 2. Μεταβολή ΜΕΧΡΙΣ_ΟΤΟΥ
ι = 10                             ! 3. Συνθήκη τερματισμού
```

11.5 Ισοδυναμία με ΟΣΟ

Η **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ_ΟΤΟΥ** μπορεί να γραφεί ισοδύναμα με **ΟΣΟ**:

! Εκτέλεση μία φορά πριν τον έλεγχο εντολές **ΟΣΟ**

ΟΧΙ συνθήκη **ΕΠΑΝΑΛΑΒΕ**
εντολές **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**

Σημείωση

Παρατηρήστε ότι η συνθήκη αντιστρέφεται με το **ΟΧΙ**, γιατί η **ΟΣΟ** συνεχίζει όσο είναι **ΑΛΗΘΗΣ**, ενώ η **ΜΕΧΡΙΣ_ΟΤΟΥ** σταματά όταν γίνει **ΑΛΗΘΗΣ**.

12 Ενσωματωμένες συναρτήσεις

Η **ΓΛΩΣΣΑ** έχει ενσωματωμένες τις εξής συναρτήσεις:

12.1 Λίστα συναρτήσεων

- **A_M(πραγματικός)**: το **Ακέραιο Μέρος** ενός πραγματικού. Δέχεται πραγματικό και επιστρέφει ακέραιο.
- **A_T(αριθμός)**: η **Απόλυτη Τιμή** ενός αριθμού. Δέχεται ακέραιο και επιστρέφει ακέραιο, ή δέχεται πραγματικό και επιστρέφει πραγματικό.
- **E(πραγματικός)**: η **Εκθετική** συνάρτηση (e^x). Επιστρέφει πραγματικό.
- **ΕΦ(πραγματικός)**: η **Εφαπτομένη** ενός πραγματικού. Επιστρέφει πραγματικό.
- **ΗΜ(πραγματικός)**: το **Ημίτονο** ενός πραγματικού. Επιστρέφει πραγματικό.
- **ΛΟΓ(πραγματικός)**: ο φυσικός **Λογάριθμος** ενός πραγματικού. Επιστρέφει πραγματικό.
- **ΣΥΝ(πραγματικός)**: το **Συνημίτονο** ενός πραγματικού. Επιστρέφει πραγματικό.
- **T_P(πραγματικός)**: η **Τετραγωνική Ρίζα** ενός πραγματικού. Ισοδυναμεί με $x^{1/2}$.

12.2 Συνοπτικός πίνακας

Συνάρτηση	Περιγραφή	Είσοδος	Έξοδος
A_M(x)	Ακέραιο μέρος	Πραγματικός	Ακέραιος
A_T(x)	Απόλυτη τιμή	Ακέρ./Πραγμ.	Ίδιος τύπος
E(x)	Εκθετική (e^x)	Πραγματικός	Πραγματικός
ΕΦ(x)	Εφαπτομένη	Πραγματικός (μοίρες)	Πραγματικός
ΗΜ(x)	Ημίτονο	Πραγματικός (μοίρες)	Πραγματικός
ΛΟΓ(x)	Φυσικός λογάριθμος	Πραγματικός (> 0)	Πραγματικός
ΣΥΝ(x)	Συνημίτονο	Πραγματικός (μοίρες)	Πραγματικός
T_P(x)	Τετραγωνική ρίζα	Πραγματικός (≥ 0)	Πραγματικός

12.3 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΕνσωματωμένεςΣυναρτήσειςΕπιδεικνύει
3 ! τις ενσωματωμένες συναρτήσεις της ΓλώσσαςΜΕΤΑΒΛΗΤΕΣ
4
5 ΠΡΑΓΜΑΤΙΚΕΣ: ΠΑΡΧΗ
6
7 π <- -3.2
8 ΓΡΑΨΕ Ακέραιο' μέρος του ', π, ': ', ΑΜ_π()
9 ΓΡΑΨΕ Απόλυτη' τιμή του ', π, ': ', ΑΤ_π()
10 ΓΡΑΨΕ Εκθετική' συνάρτηση του ', π, ': ', Επ()
11 ΓΡΑΨΕ Εφαπτομένη' του ', π, ': ', ΕΦπ()
12 ΓΡΑΨΕ Ημίτονο' του ', π, ': ', ΗΜπ()Στο
13 ! δεκαδικό λογάριθμο δεν επιτρέπονται αρνητικοί αριθμοί
14 ΓΡΑΨΕ Δεκαδικός' λογάριθμος του ', ΑΤ_π(), ': ', ΛΟΓΑ(Τ_π())
15 ΓΡΑΨΕ Συνημίτονο' του ', π, ': ', ΣΥΝπ()Στην
16 ! τετραγωνική ρίζα δεν επιτρέπονται αρνητικοί αριθμοί
17 ΓΡΑΨΕ Τετραγωνική' ρίζα του ', ΑΤ_π(), ': ', ΤΡ_Α(Τ_π())ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
18 _
```

12.4 Παρατηρήσεις

1. **Ακέραιοι ως πραγματικοί**: Όπου μπορεί να μπει πραγματικός, μπορεί και ακέραιος, λόγω του ότι οι ακέραιοι είναι υποσύνολο των πραγματικών. Το ίδιο ισχύει και στην ανάθεση τιμής $\pi <- 1$ όπου το π είναι πραγματικός και το 1 ακέραιος.

2. **Τριγωνομετρικές σε μοίρες:** Η παράμετρος των ΕΦ, ΗΜ και ΣΥΝ είναι σε **μοίρες**, όχι σε ακτίνια (σελίδα 62 του τετραδίου μαθητή).

Σημείωση

Μπορείτε να αλλάξετε αυτή τη συμπεριφορά από τις επιλογές του Διερμηνευτή, αλλά **δεν προτείνεται**.

3. **Ασάφεια στην A_M:** Υπάρχει ασάφεια σχετικά με το αν η A_M(-5.5) επιστρέφει -5 ή -6. Μπορείτε να ορίσετε τη συμπεριφορά που θέλετε από τις επιλογές του Διερμηνευτή.
4. **Περιορισμοί τιμών:**
- Στη ΛΟΓ δεν επιτρέπονται αρνητικοί αριθμοί ή μηδέν
 - Στην T_P δεν επιτρέπονται αρνητικοί αριθμοί
5. **Σταθερή αποτίμηση:** Οι ενσωματωμένες συναρτήσεις είναι **σταθερής αποτίμησης**, δηλαδή μπορούν να χρησιμοποιηθούν σε:
- Δηλώσεις σταθερών
 - Όρια πινάκων
 - Εντολές ΠΕΡΙΠΤΩΣΗ

ΣΤΑΘΕΡΕΣ

N = 10

ρίζαN_ = TP_N()ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: AA[M_T(P_(100))] ! Πίνακας 10 θέσεων

13 Η εντολή ΚΑΛΕΣΕ

Η εντολή ΚΑΛΕΣΕ μας επιτρέπει να καλέσουμε μία **διαδικασία**.

Σημείωση

Η ΚΑΛΕΣΕ χρησιμοποιείται **μόνο** για διαδικασίες, όχι για συναρτήσεις. Οι συναρτήσεις καλούνται μέσα από εκφράσεις.

13.1 Σύνταξη

```
ΚΑΛΕΣΕ ΌνομαΔιαδικασίας(παράμετρος1, παράμετρος2, ...)
```

13.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2 ΗΕντολήΚάλεσεΕπιδεικνύει
3 ! απλά τη χρήση της εντολής.ΑΡΧΗ
4
5 ΚΑΛΕΣΕ ΓράψεΌνομα('My name is Bond. James Bond.')ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
6 _ΔΙΑΔΙΚΑΣΙΑ
7
8 ΓράψεΌνομαόνομα()ΜΕΤΑΒΛΗΤΕΣ
```

ΧΑΡΑΚΤΗΡΕΣ: όνομαΑΡΧΗ

ΓΡΑΨΕ όνομαΤΕΛΟΣΔΙΑΔΙΚΑΣΙΑΣ

13.3 Παρατηρήσεις

1. **Παράμετροι:** Ο αριθμός και ο τύπος των παραμέτρων καθορίζονται από την αντίστοιχη διαδικασία.
2. **Διαδικασία χωρίς παραμέτρους:** Αν μία διαδικασία δεν έχει παραμέτρους, **δεν πρέπει** να μπουν παρενθέσεις κατά την κλήση της.

ΚΑΛΕΣΕ

ΔιαδικασίαΧωρίςΠαραμέτρους ! ΣωστόΚΑΛΕΣΕ

ΔιαδικασίαΧωρίςΠαραμέτρους () ! Λάθος !

13.4 Σύγκριση διαδικασιών και συναρτήσεων

Χαρακτηριστικό	Διαδικασία	Συνάρτηση
Κλήση με ΚΑΛΕΣΕ		×
Κλήση μέσα σε έκφραση	×	
Επιστρέφει τιμή	×	
Πολλαπλές έξοδοι (με παραμέτρους)		×

14 Διαδικασίες

Οι διαδικασίες είναι **υποπρογράμματα** που γράφονται μετά το κυρίως πρόγραμμα. Μπορούν να κληθούν με την εντολή ΚΑΛΕΣΕ από οποιοδήποτε σημείο του προγράμματος.

14.1 Σύνταξη

```
ΔΙΑΔΙΚΑΣΙΑ ΌνομαΔιαδικασίας(παράμετρος1, παράμετρος2, ...)
ΣΤΑΘΕΡΕΣ
...
ΜΕΤΑΒΛΗΤΕΣ
    τύπος: παράμετρος1, παράμετρος2, ...
    τύπος: τοπικές_μεταβλητές
ΑΡΧΗ
    εντολές
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Σημείωση

Οι παράμετροι αναφέρονται ονομαστικά κατά τη δήλωση της διαδικασίας και στη συνέχεια ο τύπος τους δηλώνεται στο τμήμα ΜΕΤΑΒΛΗΤΕΣ.

14.2 Παράδειγμα

1 ΠΡΟΓΡΑΜΜΑ

2 ΜίαΔιαδικασίαΔημιουργία

```

3 ! διαδικασίας και κλήση της με την Κάλεσε.ΑΡΧΗ
4
5 ΚΑΛΕΣΕ ΓράψεΌνομα('My name is Bond. James Bond.')
```

ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ

```

6 _ΔΙΑΔΙΚΑΣΙΑ
7
8 ΓράψεΌνομαόνομα()ΜΕΤΑΒΛΗΤΕΣ
9
10 ΧΑΡΑΚΤΗΡΕΣ: όνομαΑΡΧΗ
11
12 ΓΡΑΨΕ όνομαΤΕΛΟΣΔΙΑΔΙΚΑΣΙΑΣ
13 _
```

14.3 Παρατηρήσεις

1. **Τι μπορεί να περαστεί ως παράμετρος:** Σαν παράμετρος μπορεί να περαστεί οτιδήποτε έχει τιμή:
 - Μεταβλητές
 - Πίνακες
 - Σταθερές
 - Εκφράσεις
 - Εκφράσεις που περιέχουν κλήση συναρτήσεων
2. **Μεταβίβαση με αναφορά:** Αν μία διαδικασία κληθεί με **μεταβλητή** σαν παράμετρο και αλλαχθεί η τιμή της μέσα στη διαδικασία, τότε η τιμή της μεταβλητής στο καλών υποπρόγραμμα θα **ενημερωθεί** μετά την εκτέλεση της εντολής ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ.

Παράδειγμα μεταβίβασης

```

ΠΡΟΓΡΑΜΜΑ
  ΠαράδειγμαΜΕΤΑΒΛΗΤΕΣ

  ΑΚΕΡΑΙΕΣ: xΑΡΧΗ

  x <- 5
  ΚΑΛΕΣΕ Διπλασίασε(x)
  ΓΡΑΨΕ x ! Εμφανίζει 10ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
_ΔΙΑΔΙΚΑΣΙΑ

  Διπλασίασεα()ΜΕΤΑΒΛΗΤΕΣ

  ΑΚΕΡΑΙΕΣ: αΑΡΧΗ

  α <- α * 2ΤΕΛΟΣΔΙΑΔΙΚΑΣΙΑΣ
_
```

3. **Διαδικασία χωρίς παραμέτρους:** Μία διαδικασία χωρίς παραμέτρους δηλώνεται και καλείται **χωρίς παρενθέσεις**.

```

ΔΙΑΔΙΚΑΣΙΑ
  Χαιρετισμός ! Χωρίς παρενθέσειςΑΡΧΗ

  ΓΡΑΨΕ Γεια' σου!ΤΕΛΟΣΔΙΑΔΙΚΑΣΙΑΣ
_
```

! Κλήση: ΚΑΛΕΣΕ
Χαιρετισμός ! Χωρίς παρενθέσεις

4. **Αναδρομή:** Επιτρέπονται μέχρι **1000 αναδρομικές κλήσεις** (μία διαδικασία να καλεί τον εαυτό της). Αν γίνουν παραπάνω, εμφανίζεται το μήνυμα «Υπερχείλιση στοίβας» και το πρόγραμμα τερματίζεται.

Σημείωση

Οι **επαναληπτικές κλήσεις** (μία διαδικασία καλεί άλλη διαδικασία που καλεί την πρώτη κ.ο.κ.) επιτρέπονται απεριόριστες.

15 Συναρτήσεις

Η ΓΛΩΣΣΑ επιτρέπει τη δημιουργία **συναρτήσεων** από το χρήστη, οι οποίες μπορούν στη συνέχεια να χρησιμοποιηθούν όπως και οι ενσωματωμένες συναρτήσεις.

Το αποτέλεσμα κάθε συνάρτησης έχει πάντα τιμή κάποιου τύπου δεδομένων και δηλώνεται μετά τις παραμέτρους της συνάρτησης (ΑΚΕΡΑΙΑ, ΠΡΑΓΜΑΤΙΚΗ, ΧΑΡΑΚΤΗΡΑΣ ή ΛΟΓΙΚΗ).

Προσοχή

Οι συναρτήσεις **δεν επιτρέπεται** να επιστρέφουν πίνακες.

15.1 Σύνταξη

```
ΣΥΝΑΡΤΗΣΗ ΌνομαΣυνάρτησης(παράμ1, παράμ2, ...): ΤΥΠΟΣ  
ΣΤΑΘΕΡΕΣ  
...  
ΜΕΤΑΒΛΗΤΕΣ  
    τύπος: παράμ1, παράμ2, ...  
    τύπος: τοπικές_μεταβλητές  
ΑΡΧΗ  
    εντολές  
    ΌνομαΣυνάρτησης <- τιμή  
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Υποχρεωτική επιστροφή τιμής

Όλες οι συναρτήσεις **πρέπει** να περιέχουν μία εντολή του τύπου ΌνομαΣυνάρτησης <- τιμή, ώστε να επιστρέφουν κάποια τιμή στο καλών υποπρόγραμμα.

15.2 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ  
2   ΣυναρτήσειςΠαράδειγμα  
3   ! υλοποίησης συναρτήσεων.Υλοποιεί  
4   ! τις συναρτήσεις ΑπόλυτηΤιμή και Μέγιστος.ΑΡΧΗ  
5  
6   ΓΡΑΨΕ ΜέγιστοςΑπόλυτηΤιμή((-2), 1)                      Θα! γράψει «2»  
7   ΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ  
8   _ΣΥΝΑΡΤΗΣΗ
```

```

9  ΑπόλυτηΤιμή(χ): ΑΚΕΡΑΙΑΒρίσκει
10 ! την απόλυτη τιμή του χ χωρίς να χρησιμοποιήσει την AT_ΜΕΤΑΒΛΗΤΕΣ
11
12  ΑΚΕΡΑΙΕΣ: χΑΡΧΗ
13
14  ΑΝ χ >= 0 ΤΟΤΕ
15    ΑπόλυτηΤιμή <- χ
16  ΑΛΛΙΩΣ
17    ΑπόλυτηΤιμή <- χ-
18  ΤΕΛΟΣΑΝ_ΤΕΛΟΣΣΥΝΑΡΤΗΣΗΣ
19 _ΣΥΝΑΡΤΗΣΗ
20
21  Μέγιστος(α, β): ΑΚΕΡΑΙΑΕπιστρέφει
22 ! το Μέγιστο μεταξύ των α και βΜΕΤΑΒΛΗΤΕΣ
23
24  ΑΚΕΡΑΙΕΣ: α, βΑΡΧΗ
25
26  ΑΝ α >= β ΤΟΤΕ
27    Μέγιστος <- α
28  ΑΛΛΙΩΣ
29    Μέγιστος <- β
30  ΤΕΛΟΣΑΝ_ΤΕΛΟΣΣΥΝΑΡΤΗΣΗΣ
31 _

```

15.3 Παρατηρήσεις

1. **Υποχρεωτική επιστροφή:** Αν δεν υπάρχει εντολή τύπου ΌνομαΣυνάρτησης <- τιμή στο εσωτερικό μιας συνάρτησης, ο Διερμηνευτής εμφανίζει **σφάλμα εκτέλεσης**.
2. **Αναδρομή με το όνομα:** Αν το όνομα μιας συνάρτησης εμφανιστεί οπουδήποτε αλλού εκτός από το **αριστερό μέρος** μιας ανάθεσης τιμής, προκαλείται αναδρομή.

```

ΣΥΝΑΡΤΗΣΗ
Παραγοντικόν(): ΑΚΕΡΑΙΑΑΡΧΗ

ΑΝ ν <= 1 ΤΟΤΕ
  Παραγοντικό <- 1          ! Επιστροφή τιμής
ΑΛΛΙΩΣ
  Παραγοντικό <- ν * Παραγοντικόν(-1)  ! Αναδρομή!
ΤΕΛΟΣΑΝ_ΤΕΛΟΣΣΥΝΑΡΤΗΣΗΣ
_

```

3. **Τι μπορεί να περαστεί ως παράμετρος:** Σαν παράμετρος μπορεί να περαστεί οτιδήποτε έχει τιμή:
 - Μεταβλητές
 - Πίνακες
 - Σταθερές
 - Εκφράσεις
 - Εκφράσεις που περιέχουν κλήση συναρτήσεων
4. **Μεταβίβαση με τιμή:** Οι παράμετροι στις συναρτήσεις περνιούνται **με τιμή**, δηλαδή οποιαδήποτε αλλαγή στις τυπικές παραμέτρους μιας συνάρτησης **δεν επηρεάζει** τις πραγματικές παραμέτρους.

Διαφορά από διαδικασίες

Στις **διαδικασίες** οι παράμετροι περνιούνται με αναφορά (οι αλλαγές επηρεάζουν το καλών πρόγραμμα), ενώ στις **συναρτήσεις** με τιμή (οι αλλαγές δεν επηρεάζουν).

5. **Συνάρτηση χωρίς παραμέτρους:** Μία συνάρτηση χωρίς παραμέτρους δηλώνεται και καλείται χωρίς παρενθέσεις.
6. **Αναδρομή:** Επιτρέπονται μέχρι **1000 αναδρομικές κλήσεις**. Αν γίνουν παραπάνω, εμφανίζεται το μήνυμα «Υπερχείλιση στοίβας» και το πρόγραμμα τερματίζεται.

15.4 Σύγκριση διαδικασιών και συναρτήσεων

Χαρακτηριστικό	Διαδικασία	Συνάρτηση
Κλήση	με ΚΑΛΕΣΕ	μέσα σε έκφραση
Επιστρέφει τιμή	×	
Μπορεί να επιστρέψει πίνακα	×	×
Μεταβίβαση παραμέτρων	με αναφορά	με τιμή
Αλλαγές επηρεάζουν καλών		×
Χωρίς παραμέτρους	χωρίς ()	χωρίς ()
Όριο αναδρομής	1000	1000

16 Μεταβίβαση παραμέτρων σε υποπρογράμματα

Η ΓΛΩΣΣΑ μοιάζει λίγο με την Basic στη μεταβίβαση παραμέτρων, αλλά είναι πάρα πολύ διαφορετική από την Pascal. Η Basic ακολουθεί το μηχανισμό μεταβίβασης παραμέτρων **με αναφορά**, όποτε αυτό είναι δυνατό.

16.1 Παράδειγμα

```
1 ΠΡΟΓΡΑΜΜΑ
2  ΜεταβίβασηΠαραμέτρωνΜΕΤΑΒΛΗΤΕΣ
3
4  ΠΡΑΓΜΑΤΙΚΕΣ: π
5  ΑΚΕΡΑΙΕΣ: αΡΧΗ
6
7  π <- 1
8  ΓΡΑΨΕ π' = ', π                      Γράφει! 1
9  ΚΑΛΕΣΕ Δπ()                      Καλείται! με μεταβλητή, οπότε το π αλλάζει
10 ΓΡΑΨΕ π' = ', π                      Γράφει! 2
11 α <- 1
12 ΓΡΑΨΕ α' = ', α                      Γράφει! 1
13 ! Κάλεσε Δα()  Αυτό! δεν επιτρέπεται, προσπαθούμε να περάσουμε
14                  με! αναφορά ακέραιο ενώ απαιτείται πραγματικός.
15 ΚΑΛΕΣΕ Δα(())  Βάζοντας! το α σε παρένθεση το κάνουμε έκφραση, οπότε
16 ! περνιέται με τιμή. Το ίδιο θα γινόταν με Κάλεσε Δα( + 0).
17 ΓΡΑΨΕ α' = ', αΤΕΛΟΣΠΡΟΓΡΑΜΜΑΤΟΣ
18 _ΔΙΑΔΙΚΑΣΙΑ
19
20 Δβ()Αυξάνει
21 ! το β. Αν περαστεί μεταβλητή (= με αναφορά) αλλάζει και η
22 ! μεταβλητή του κυρίως προγράμματος. Αν περαστεί έκφραση (= με τιμή
23 !) το β αλλάζει μόνο τοπικά.ΜΕΤΑΒΛΗΤΕΣ
24
25 ΠΡΑΓΜΑΤΙΚΕΣ: βΑΡΧΗ
```

```

26
27  β <- β + 1 ΤΕΛΟΣΔΙΑΔΙΚΑΣΙΑΣ
28 _ Δ

```

16.2 Κανόνες μεταβίβασης

16.2.1 Μεταβίβαση με αναφορά (μεταβλητή ως παράμετρος)

Αν καλέσουμε με παράμετρο **μεταβλητή** κάποια **διαδικασία** (όχι συνάρτηση) και αυτή αλλάξει την τιμή της, η νέα τιμή θα επιστραφεί στο κυρίως πρόγραμμα.

Υποχρεωτική συμβατότητα τύπων

Η τυπική και η ουσιαστική παράμετρος **πρέπει** να είναι του **ίδιου τύπου** δεδομένων. Δεν επιτρέπεται καν να περάσουμε ακέραιο σε διαδικασία που περιμένει πραγματικό αριθμό!

16.2.2 Μεταβίβαση με τιμή (έκφραση ως παράμετρος)

Αν καλέσουμε με παράμετρο **έκφραση** κάποιο υποπρόγραμμα (είτε διαδικασία είτε συνάρτηση) και αυτό αλλάξει την τιμή της, η νέα τιμή **δεν μπορεί** να επιστραφεί στο κυρίως πρόγραμμα.

Ευελιξία τύπων στις εκφράσεις

Η έκφραση πρέπει να είναι του ίδιου τύπου με την τυπική παράμετρο, με τη διαφορά ότι **επιτρέπεται** πέρασμα ακεραίου σε υποπρόγραμμα που περιμένει πραγματικό (όπως και στην ανάθεση τιμής).

16.2.3 Μετατροπή μεταβλητής σε έκφραση

Αν θέλουμε να περάσουμε με τιμή κάποια μεταβλητή, πρέπει να κάνουμε κάποια «πράξη» ώστε να πάψει να είναι μεταβλητή:

Τρόπος	Παράδειγμα
Παρένθεση (απλούστερο)	ΚΑΛΕΣΕ Δ((χ))
Πρόσθεση μηδέν	ΚΑΛΕΣΕ Δ(χ + 0)
Πολλαπλασιασμός με ένα	ΚΑΛΕΣΕ Δ(1 * χ)

16.3 Συνοπτικός πίνακας

Χαρακτηριστικό	Μεταβλητή	Έκφραση
Μηχανισμός	Με αναφορά	Με τιμή
Αλλαγές επηρεάζουν καλών	(διαδικασίες)	×
Ακέραιος → Πραγματικός	×	

16.4 Ο μηχανισμός copy in - copy out

Το σχολικό βιβλίο (σελ. 218) περιγράφει μηχανισμό μεταβίβασης παραμέτρων με **αντιγραφή** (copy in - copy out), όχι με αναφορά. Αυτό ακριβώς έχει υλοποιηθεί στο Διεργμηνευτή.

16.4.1 Διαφορές copy in - copy out από αναφορά

1. Χρονική στιγμή ενημέρωσης:

- **Με αναφορά:** Όταν αλλάζει η τιμή μιας παραμέτρου, **ταυτόχρονα** γίνεται η αλλαγή στην αντίστοιχη μεταβλητή του κυρίως προγράμματος (δεν υπάρχουν δύο μεταβλητές, απλά δύο ονόματα για την ίδια).
- **Copy in - copy out:** Υπάρχουν **δύο μεταβλητές** και η δεύτερη (της διαδικασίας) αντιγράφεται στην πρώτη **μόνο** με την εντολή ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ.

Πείραμα στο Διερμηνευτή

Σταματήστε την εκτέλεση αφού ξεκινήσει η ΚΑΛΕΣΕ Δ(π) αλλά πριν το ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ. Το β θα έχει γίνει 2, αλλά το π θα έχει ακόμα την τιμή 1. Γίνεται 2 **μετά** το ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ.

2. Κατανάλωση μνήμης:

Η δημιουργία διαφορετικών μεταβλητών για κάθε κλήση σπαταλά μνήμη. Π.χ. ταξινόμηση πίνακα 5000 ονομάτων με QuickSort (1 Mb) και 20 αναδρομικές κλήσεις θα χρειαστεί 20 Mb RAM μόνο για τον πίνακα.

3. Διπλή παράμετρος με ίδια μεταβλητή:

Κρίσιμη διαφορά

Αν έχουμε διαδικασία που αυξάνει δύο παραμέτρους ($\alpha \leftarrow \alpha + 1, \beta \leftarrow \beta + 1$) και την καλέσουμε με την ίδια μεταβλητή (ΚΑΛΕΣΕ Αύξηση(α, α)):

- **Με αναφορά:** το α θα αυξηθεί κατά 2
- **Copy in - copy out:** το α θα αυξηθεί κατά 1

Αυτό συμβαίνει επειδή δημιουργούνται δύο αντίγραφα του α, αυξάνονται και τα δύο κατά ένα, και τελικά το ένα αντικαθιστά το άλλο.

Σύσταση

Καλύτερα να αποφεύγεται η κατασκευή ασκήσεων που χρησιμοποιούν αυτήν τη δυσνόητη συμπεριφορά.

17 Λέξεις-κλειδιά της ΓΛΩΣΣΑΣ

Σε αυτήν την ενότητα παρουσιάζονται όλες οι **λέξεις-κλειδιά** (δεσμευμένες λέξεις) της ΓΛΩΣΣΑΣ, οργανωμένες σε κατηγορίες για εύκολη αναφορά.

17.1 Δομή προγράμματος

Λέξη-κλειδί	Περιγραφή
ΠΡΟΓΡΑΜΜΑ	Έναρξη προγράμματος (επικεφαλίδα)
ΣΤΑΘΕΡΕΣ	Τμήμα δήλωσης σταθερών
ΜΕΤΑΒΛΗΤΕΣ	Τμήμα δήλωσης μεταβλητών
ΑΡΧΗ	Έναρξη κυρίως σώματος
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ	Τέλος προγράμματος

17.2 Τύποι δεδομένων (δήλωση μεταβλητών)

Λέξη-κλειδί	Περιγραφή
ΑΚΕΡΑΙΕΣ	Δήλωση ακέραιων μεταβλητών
ΠΡΑΓΜΑΤΙΚΕΣ	Δήλωση πραγματικών μεταβλητών
ΧΑΡΑΚΤΗΡΕΣ	Δήλωση αλφαριθμητικών μεταβλητών
ΛΟΓΙΚΕΣ	Δήλωση λογικών μεταβλητών

17.3 Τύποι δεδομένων (επιστροφή συναρτήσεων)

Λέξη-κλειδί	Περιγραφή
ΑΚΕΡΑΙΑ	Επιστροφή ακέραιου από συνάρτηση
ΠΡΑΓΜΑΤΙΚΗ	Επιστροφή πραγματικού από συνάρτηση
ΧΑΡΑΚΤΗΡΑΣ	Επιστροφή αλφαριθμητικού από συνάρτηση
ΛΟΓΙΚΗ	Επιστροφή λογικού από συνάρτηση

17.4 Λογικές τιμές

Λέξη-κλειδί	Περιγραφή
ΑΛΗΘΗΣ	Λογική τιμή «αληθής»
ΨΕΥΔΗΣ	Λογική τιμή «ψευδής»

17.5 Αριθμητικοί τελεστές

Τελεστής	Όνομα	Περιγραφή
+	Συν	Πρόσθεση
-	Πλην	Αφαίρεση (ή αρνητικό πρόσημο)
*	Επί	Πολλαπλασιασμός
/	Διά	Διαίρεση (αποτέλεσμα πραγματικός)
^	Δύναμη	Ύψωση σε δύναμη
DIV	Ακέραια διαίρεση	Πηλίκο ακέραιας διαίρεσης
MOD	Υπόλοιπο	Υπόλοιπο ακέραιας διαίρεσης

17.6 Συγκριτικοί τελεστές

Τελεστής	Εναλλακτικό	Περιγραφή
=		Ίσον
<>	≠	Διάφορο
<		Μικρότερο
>		Μεγαλύτερο
<=	≤	Μικρότερο ή ίσο
>=	≥	Μεγαλύτερο ή ίσο

17.7 Λογικοί τελεστές

Λέξη-κλειδί	Αγγλικά	Περιγραφή
ΚΑΙ	AND	Λογικό «και»
Η	OR	Λογικό «ή»
ΟΧΙ	NOT	Λογική άρνηση

17.8 Εντολές εισόδου/εξόδου

Λέξη-κλειδί	Περιγραφή
ΓΡΑΨΕ	Εμφάνιση στην οθόνη
ΔΙΑΒΑΣΕ	Ανάγνωση από το πληκτρολόγιο

17.9 Εντολή ανάθεσης

Σύμβολο	Εναλλακτικό	Περιγραφή
< -	←	Ανάθεση τιμής σε μεταβλητή

17.10 Δομή επιλογής (ΑΝ)

Λέξη-κλειδί	Περιγραφή
ΑΝ	Έναρξη συνθήκης
ΤΟΤΕ	Μετά τη συνθήκη
ΑΛΛΙΩΣ_ΑΝ	Εναλλακτική συνθήκη (else-if)
ΑΛΛΙΩΣ	Εναλλακτικό τμήμα (else)
ΤΕΛΟΣ_ΑΝ	Τέλος δομής ΑΝ

17.11 Δομή επιλογής (ΕΠΙΛΕΞΕ)

Λέξη-κλειδί	Περιγραφή
ΕΠΙΛΕΞΕ	Έναρξη πολλαπλής επιλογής
ΠΕΡΙΠΤΩΣΗ	Μία περίπτωση τιμής
ΠΕΡΙΠΤΩΣΗ_ΑΛΛΙΩΣ	Προεπιλεγμένη περίπτωση
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ	Τέλος δομής ΕΠΙΛΕΞΕ

17.12 Δομή επανάληψης (ΓΙΑ)

Λέξη-κλειδί	Περιγραφή
ΓΙΑ	Έναρξη επανάληψης με μετρητή
ΑΠΟ	Αρχική τιμή μετρητή
ΜΕΧΡΙ	Τελική τιμή μετρητή
ΜΕ_ΒΗΜΑ	Βήμα μετρητή (προαιρετικό)
ΜΕ_ΒΗΜΑ	Εναλλακτική γραφή (χωρίς _)
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	Τέλος βρόχου

17.13 Δομή επανάληψης (ΟΣΟ)

Λέξη-κλειδί	Περιγραφή
ΟΣΟ	Έναρξη επανάληψης με συνθήκη
ΕΠΑΝΑΛΑΒΕ	Μετά τη συνθήκη
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	Τέλος βρόχου

17.14 Δομή επανάληψης (ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ)

Λέξη-κλειδί	Περιγραφή
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ	Έναρξη επανάληψης (do-while)
ΜΕΧΡΙΣ_ΟΤΟΥ	Συνθήκη τερματισμού

17.15 Υποπρογράμματα

Λέξη-κλειδί	Περιγραφή
ΔΙΑΔΙΚΑΣΙΑ	Δήλωση διαδικασίας
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	Τέλος διαδικασίας
ΣΥΝΑΡΤΗΣΗ	Δήλωση συνάρτησης
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ	Τέλος συνάρτησης
ΚΑΛΕΣΕ	Κλήση διαδικασίας

17.16 Ενσωματωμένες συναρτήσεις

Συνάρτηση	Περιγραφή	Τύπος αποτελέσματος
A_M(x)	Ακέραιο μέρος	Ακέραιος
A_T(x)	Απόλυτη τιμή	Ίδιος με είσοδο
E(x)	Εκθετική (e^x)	Πραγματικός
EΦ(x)	Εφαπτομένη (μοίρες)	Πραγματικός
ΗΜ(x)	Ημίτονο (μοίρες)	Πραγματικός
ΛΟΓ(x)	Φυσικός λογάριθμος	Πραγματικός
ΣΥΝ(x)	Συνημίτονο (μοίρες)	Πραγματικός
T_P(x)	Τετραγωνική ρίζα	Πραγματικός

17.17 Ειδικά σύμβολα

Σύμβολο	Περιγραφή
!	Σχόλιο (μέχρι τέλος γραμμής)
&	Συνέχεια εντολής στην επόμενη γραμμή
..	Διάστημα τιμών (στην ΕΠΙΛΕΞΕ)
[]	Δείκτης πίνακα
()	Παρενθέσεις (ομαδοποίηση, παράμετροι)
,	Διαχωρισμός (παραμέτρων, μεταβλητών)
:	Μετά τον τύπο δεδομένων στη δήλωση

17.18 Συνοπτική λίστα όλων των λέξεων-κλειδιών

Παρακάτω παρατίθενται **αλφαβητικά** όλες οι λέξεις-κλειδιά της ΓΛΩΣΣΑΣ:

ΑΛΗΘΗΣ	ΑΛΛΙΩΣ	ΑΛΛΙΩΣ_ΑΝ	ΑΝ
ΑΠΟ	ΑΡΧΗ	ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ	A_M
A_T	ΓΙΑ	ΓΡΑΨΕ	ΔΙΑΒΑΣΕ
ΔΙΑΔΙΚΑΣΙΑ	DIV	E	EΦ
ΕΠΑΝΑΛΑΒΕ	ΕΠΙΛΕΞΕ	H	ΗΜ
ΚΑΛΕΣΕ	ΚΑΙ	ΛΟΓΙΚΕΣ	ΛΟΓΙΚΗ
ΛΟΓ	ΜΕ_ΒΗΜΑ	ΜΕΤΑΒΛΗΤΕΣ	ΜΕΧΡΙ
ΜΕΧΡΙΣ_ΟΤΟΥ	MOD	ΟΣΟ	ΟΧΙ
ΠΕΡΙΠΤΩΣΗ	ΠΡΑΓΜΑΤΙΚΕΣ	ΠΡΑΓΜΑΤΙΚΗ	ΠΡΟΓΡΑΜΜΑ
ΣΤΑΘΕΡΕΣ	ΣΥΝΑΡΤΗΣΗ	ΣΥΝ	ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ	ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ	ΤΟΤΕ	T_P	ΧΑΡΑΚΤΗΡΑΣ
ΧΑΡΑΚΤΗΡΕΣ	ΨΕΥΔΗΣ	ΑΚΕΡΑΙΕΣ	ΑΚΕΡΑΙΑ