

Κεφάλαιο 7

Βασικές Έννοιες Προγραμματισμού

ΠΕΡΙΕΧΟΜΕΝΑ

Αλφάβητο και τύποι δεδομένων

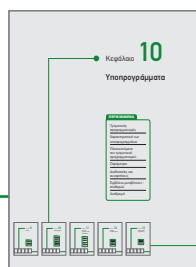
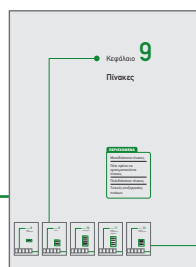
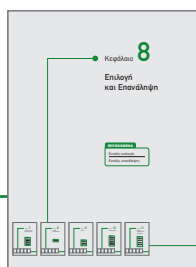
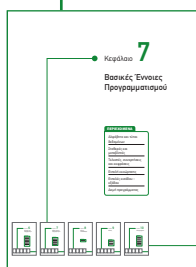
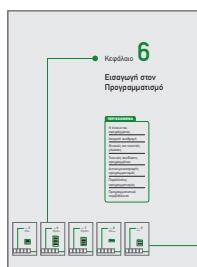
Σταθερές και μεταβλητές

Τελεστές, συναρτήσεις και εκφράσεις

Εντολή εκχώρησης

Εντολές εισόδου-εξόδου

Δομή προγράμματος



Εισαγωγή

Κάθε γλώσσα προγραμματισμού, όπως αναφέρθηκε, έχει το δικό της λεξιλόγιο και τα προγράμματά της ακολουθούν αυστηρούς γραμματικούς και συντακτικούς κανόνες. Για τη δημιουργία σωστών προγραμμάτων είναι απαραίτητη η γνώση των εντολών και του τρόπου σύνταξής τους.

Στο κεφάλαιο αυτό θα παρουσιαστούν τα βασικά στοιχεία της **ΓΛΩΣΣΑΣ**. Θα ασχοληθούμε με τους τύπους δεδομένων που υποστηρίζει, τα είδη των μεταβλητών της, τον τρόπο που υπολογίζονται οι παραστάσεις καθώς και τη δομή που πρέπει να ακολουθεί κάθε πρόγραμμα.

Επίσης θα παρουσιαστούν οι βασικές εντολές της **ΓΛΩΣΣΑΣ**, η εντολή εκχώρησης τιμών σε μεταβλητές και οι εντολές εισόδου εξόδου, με τις οποίες το πρόγραμμα επικοινωνεί με το χρήστη.

Διδακτικοί στόχοι

Να είναι σε θέση ο μαθητής:

- Να διακρίνει τις σταθερές από τις μεταβλητές.

- Να αναγνωρίζει τους διάφορους τύπους μεταβλητών.

- Να μετατρέπει τις αριθμητικές πράξεις σε εντολές προγράμματος.

- Να διατυπώνει τη δομή ενός προγράμματος.

- Να συντάσσει απλά προγράμματα, τα οποία εισάγουν δεδομένα, τα επεξεργάζονται και εμφανίζουν τα αποτελέσματα στην οθόνη.

Προερωτήσεις

- Έχει ένα πρόγραμμα συγκεκριμένους κανόνες στον τρόπο που γράφεται;

- Πώς διαχειρίζεται ένα πρόγραμμα τα δεδομένα;

- Πώς εκτελούνται οι πράξεις σε ένα πρόγραμμα;

- Με ποιον τρόπο επικοινωνεί το πρόγραμμα με το χρήστη κατά την εκτέλεσή του;

Εκατοντάδες γλώσσες προγραμματισμού χρησιμοποιούνται όπως αναφέρθηκε σήμερα για την επίλυση των προβλημάτων με τον υπολογιστή, τη δημιουργία σωστών προγραμμάτων. Η επιλογή της κατάλληλης γλώσσας δεν είναι εύκολη και εξαρτάται από το είδος του προγράμματος, το διαθέσιμο εξοπλισμό και σαφώς τις γνώσεις και τις ιδιαίτερες προτιμήσεις του προγραμματιστή. Συχνά το ίδιο πρόβλημα μπορεί να λυθεί εξίσου ικανοποιητικά με πολλές διαφορετικές γλώσσες προγραμματισμού.

Πρέπει να έχουμε πάντα υπόψη μας ότι:

Κάθε γλώσσα προγραμματισμού σχεδιάζεται για συγκεκριμένο σκοπό, δίνοντας ιδιαίτερη έμφαση σε ορισμένα χαρακτηριστικά σε βάρος βέβαια κάποιων άλλων. Δεν υπάρχει καλύτερη γλώσσα προγραμματισμού, απλά υπάρχει γλώσσα καταλληλότερη για την ανάπτυξη συγκεκριμένου τύπου εφαρμογών.

Οι γλώσσες προγραμματισμού περιέχουν πολλές πληροφορίες που σχετίζονται με τεχνικά θέματα. Αυτά τα χαρακτηριστικά αλλάζουν αρκετά συχνά, όπως η γλώσσα εξελίσσεται και εξαρτώνται σε μεγάλο βαθμό από τον εξοπλισμό και το λειτουργικό σύστημα. Οι νεώτερες εκδόσεις των γλωσσών συνήθως διαθέτουν πλουσιότερο ρεπερτόριο εντολών και άλλων δυνατοτήτων, χωρίς όμως να προσθέτουν οτιδήποτε στην εκμάθηση της δημιουργίας σωστών προγραμμάτων.

Σχεδόν όλες οι γλώσσες προγραμματισμού έχουν κοινά χαρακτηριστικά, επεξεργάζονται κατά κανόνα τους ίδιους τύπους δεδομένων, υποστηρίζουν τις ίδιες βασικές δομές και έχουν παρόμοιες εντολές.

Η γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε στα επόμενα κεφάλαια που ονομάζεται **ΓΛΩΣΣΑ**, είναι σχεδιασμένη έτσι ώστε να αποτελέσει ένα εργαλείο προγραμματισμού κατάλληλο για εκπαιδευτικούς σκοπούς. Περιέχει τα χαρακτηριστικά, τις δομές και τις εντολές που περιέχονται σε διάφορες σύγχρονες γλώσσες προγραμματισμού όπως η Pascal, Visual Basic, C, C++, Java και άλλες, χωρίς όμως να ασχολείται με τις τεχνικές λεπτομέρειες αυτών.

Έτσι ο προγραμματισμός με τη **ΓΛΩΣΣΑ** εστιάζεται στην ανάπτυξη του αλγορίθμου και τη μετατροπή του σε σωστό πρόγραμμα.

Σε όλο το βιβλίο οι εντολές της **ΓΛΩΣΣΑΣ** είναι γραμμένες με μπλε χρώμα και είναι πάντα με κεφαλαία, ενώ οι μεταβλητές είναι με πεζά ή κεφαλαία αλλά με το πρώτο γράμμα πάντα κεφαλαίο. Τα σχόλια των προγραμμάτων είναι γραμμένα με πράσινο χρώμα.

7.1 | Το αλφάβητο της ΓΛΩΣΣΑΣ

Το αλφάβητο της **ΓΛΩΣΣΑΣ** αποτελείται από τα γράμματα του ελληνικού και του λατινικού αλφαβήτου, τα ψηφία, καθώς και από ειδικά σύμβολα, που χρησιμοποιούνται για προκαθορισμένες ενέργειες, στις οποίες θα αναφερθούμε στη συνέχεια.

Συγκεκριμένα

Γράμματα

Κεφαλαία ελληνικού αλφαβήτου (Α-Ω)

Πεζά ελληνικού αλφαβήτου (α-ω)

Κεφαλαία λατινικού αλφαβήτου (A-Z)

Πεζά λατινικού αλφαβήτου (a-z)

Ψηφία

0-9

Ειδικοί χαρακτήρες

+ - * / = ' () . , ' ! & κενός χαρακτήρας ^



Στην πραγματικότητα τα δεδομένα καταχωρούνται στη μνήμη του υπολογιστή καταλαμβάνοντας συγκεκριμένο αριθμό θέσεων (bytes). Ανάλογα με τον τύπο του δεδομένου και το διατιθέμενο αριθμό bytes ποικίλλει και το εύρος τιμών που μπορούν να λάβουν. Έτσι στον υπολογιστή διαθέτουμε ένα υποσύνολο ακεραίων ή πραγματικών αριθμών. Συνήθεις τύποι δεδομένων στις διάφορες γλώσσες προγραμματισμού είναι ο ακέραιος (integer) σε 1, 2 ή 4 bytes και ο πραγματικός (real) σε 4 ή 8 bytes.

7.2 | Τύποι δεδομένων

Οι υπολογιστές επεξεργάζονται δεδομένα διαφόρων τύπων, γι' αυτό είναι σημαντικό να κατανοήσουμε τους διαφορετικούς τύπους δεδομένων που χειρίζεται η **ΓΛΩΣΣΑ**.

Οι τύποι δεδομένων που υποστηρίζει η **ΓΛΩΣΣΑ** είναι οι αριθμητικοί, που περιλαμβάνουν τους ακέραιους και τους πραγματικούς αριθμούς, οι χαρακτήρες και τέλος οι λογικοί.

Ακέραιος τύπος. Ο τύπος αυτός περιλαμβάνει τους ακέραιους που είναι γνωστοί από τα μαθηματικά. Οι ακέραιοι μπορούν να είναι θετικοί, αρνητικοί ή μηδέν. Παραδείγματα ακεραίων είναι οι αριθμοί 1, 3409, 0, -980.

Πραγματικός τύπος. Ο τύπος αυτός περιλαμβάνει τους πραγματικούς αριθμούς που γνωρίζουμε από τα μαθηματικά. Οι αριθμοί 3.14159, 2.71828, -112.45, 0.45 είναι πραγματικοί αριθμοί. Και οι πραγματικοί αριθμοί μπορούν να είναι θετικοί, αρνητικοί ή μηδέν.

Χαρακτήρας. Ο τύπος αυτός αναφέρεται τόσο σε ένα χαρακτήρα όσο και μία σειρά χαρακτήρων. Τα δεδομένα αυτού του τύπου μπορούν να περιέχουν οποιονδήποτε χαρακτήρα παράγεται από το πληκτρολόγιο. Παραδείγματα χαρακτήρων είναι 'Κ', 'Κώστας', 'σήμερα είναι Τετάρτη', 'Τα πολλαπλάσια του 15 είναι'.

Οι χαρακτήρες πρέπει υποχρεωτικά να βρίσκονται μέσα σε απλά εισαγωγικά, '. Τα δεδομένα αυτού του τύπου, επειδή περιέχουν τόσο αλφαβητικούς όσο και αριθμητικούς χαρακτήρες, ονομάζονται συχνά **αλφαριθμητικά**.

Λογικός. Αυτός ο τύπος δέχεται μόνο δύο τιμές: ΑΛΗΘΗΣ και ΨΕΥΔΗΣ. Οι τιμές αντιπροσωπεύουν αληθείς ή ψευδείς συνθήκες.

7.3 | Σταθερές

Οι σταθερές (constants) είναι προκαθορισμένες τιμές που δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του προγράμματος. Οι σταθερές είναι αντίστοιχου τύπου δεδομένων, δηλαδή ακέραιες, πραγματικές, αλφαριθμητικές ή λογικές.

Συμβολικές σταθερές

Η **ΓΛΩΣΣΑ** επιτρέπει την αντιστοίχιση σταθερών τιμών με ονόματα, εφόσον αυτά δηλωθούν στην αρχή του προγράμματος (στο τμήμα δήλωσης σταθερών, βλέπε παρακάτω).

Σύνταξη

ΣΤΑΘΕΡΕΣ

Όνομα-1 = σταθερή-τιμή-1

Όνομα-2 = σταθερά-τιμή-2

⋮

Όνομα-ν = σταθερά-τιμή-ν

Παραδείγματα

ΣΤΑΘΕΡΕΣ

ΠΙ=3.14159

ΦΠΑ=0.19

ΟΝΟΜΑ='Κώστας'

Λειτουργία

Αποδίδει ονόματα σε σταθερές τιμές. Κάθε ένα από αυτά τα ονόματα μπορεί να χρησιμοποιηθεί οπουδήποτε στο πρόγραμμα, αλλά δεν είναι δυνατή η μεταβολή της τιμής κατά τη διάρκεια εκτέλεσης του προγράμματος.

Η χρήση ονομάτων σταθερών κάνει το πρόγραμμα πιο κατανοητό και κατά συνέπεια ευκολότερο να διορθωθεί και να συντηρηθεί.

ΟΝΟΜΑΤΑ

Κάθε πρόγραμμα, καθώς και τα δεδομένα που χρησιμοποιεί (συμβολικές σταθερές και μεταβλητές) έχουν ένα όνομα, με το οποίο αναφερόμαστε σε αυτά. Τα ονόματα αυτά μπορούν να αποτελούνται από γράμματα πεζά ή κεφαλαία του ελληνικού ή του λατινικού αλφαβήτου (Α-Ω, Α-Z), ψηφία (0-9) καθώς και τον χαρακτήρα κάτω παύλα (underscore) (_), ενώ πρέπει υποχρεωτικά να αρχίζουν με γράμμα.

Επειδή μερικές λέξεις χρησιμοποιούνται από την ίδια τη **ΓΛΩΣΣΑ** για συγκεκριμένους λόγους, όπως οι λέξεις ΠΡΟΓΡΑΜΜΑ, ΑΚΕΡΑΙΕΣ, ΠΡΑΓΜΑΤΙΚΕΣ, ΑΝ κ.λπ., αυτές οι λέξεις δεν μπορούν να χρησιμοποιηθούν ως ονόματα. Οι λέξεις αυτές αποκαλούνται δεσμευμένες.

Παραδείγματα ονομάτων που είναι αποδεκτά από τη **ΓΛΩΣΣΑ** είναι: Α, Όνομα, Τιμή, Τυπική_Απόκλιση, Α100, ΦΠΑ, μέγιστο, Υπολογισμός_Ταχύτητας.

Παραδείγματα ονομάτων που δεν είναι αποδεκτά είναι: 100Α, Μέση Τιμή, Κόστος\$.

7.4 | Μεταβλητές

Η έννοια της μεταβλητής (variable) είναι γνωστή από τα μαθηματικά. Για παράδειγμα ο τύπος της γεωμετρίας

$$E = \alpha\beta$$

υπολογίζει το εμβαδόν (E) ενός ορθογωνίου με διαστάσεις, που συμβολίζονται με α και β . Αν στο α και στο β δοθούν οι αντίστοιχες τιμές, τότε ο τύπος αυτός υπολογίζει το εμβαδόν του ορθογωνίου.

Μια μεταβλητή λοιπόν παριστάνει μία ποσότητα που η τιμή της μπορεί να μεταβάλλεται.



Οι μεταβλητές που χρησιμοποιούνται σε ένα πρόγραμμα αντιστοιχούνται από το μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του υπολογιστή. Η τιμή της μεταβλητής είναι η τιμή που βρίσκεται στην αντίστοιχη θέση μνήμης και όπως αναφέρθηκε μπορεί να μεταβάλλεται κατά τη διάρκεια της εκτέλεσης του προγράμματος.

Μπορούμε να παρομοιάσουμε τη μεταβλητή και την αντίστοιχη θέση μνήμης σαν ένα γραμματοκιβώτιο, το οποίο εξωτερικά έχει ως όνομα το όνομα της μεταβλητής και ως περιεχόμενο εσωτερικά την τιμή που έχει εκείνη τη συγκεκριμένη στιγμή η μεταβλητή.

Ενώ η τιμή της μεταβλητής μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος, αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της μεταβλητής.

Η **ΓΛΩΣΣΑ** επιτρέπει τη χρήση μεταβλητών των τεσσάρων τύπων που αναφέρθηκαν, δηλαδή ακεραίων, πραγματικών, χαρακτήρων και λογικών, ενώ η δήλωση του τύπου κάθε μεταβλητής γίνεται υποχρεωτικά στο τμήμα δήλωσης μεταβλητών.

Το όνομα κάθε μεταβλητής ακολουθεί τους κανόνες δημιουργίας ονομάτων, δηλαδή αποτελείται από γράμματα, ψηφία καθώς και τον χαρακτήρα `_`, ενώ το όνομα κάθε μεταβλητής είναι μοναδικό για κάθε πρόγραμμα.

Σύνταξη

ΜΕΤΑΒΛΗΤΕΣ

τύπος-1: Λίστα-μεταβλητών-1

τύπος-2: Λίστα-μεταβλητών-2

·
·
·

Τύπος-ν: Λίστα-μεταβλητών-ν

Παραδείγματα

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Εμβαδόν, Α

ΑΚΕΡΑΙΕΣ: ΤΙΜΗ, Ν

ΧΑΡΑΚΤΗΡΕΣ: Όνομα

ΛΟΓΙΚΕΣ: Έλεγχος

Λειτουργία

Δηλώνει τον τύπο όλων των μεταβλητών που χρησιμοποιούνται στο πρόγραμμα.



Συνιστάται τα ονόματα των μεταβλητών και των σταθερών να ανάγουν στο περιεχόμενό τους.

Αν και όπως αναφέρθηκε, το όνομα των μεταβλητών μπορεί να είναι οποιοσδήποτε συνδυασμός χαρακτήρων, είναι καλή πρακτική να χρησιμοποιούνται ονόματα τα οποία να υπονοούν το περιεχόμενό τους, κάνοντας το πρόγραμμα ευκολότερο στην ανάγνωσή του και στην κατανόησή του.

Για παράδειγμα, στην περίπτωση του υπολογισμού του εμβαδού είναι προτιμότερη η χρήση του ονόματος ΕΜΒΑΔΟ για την αντίστοιχη μεταβλητή, από ένα όνομα που αποτελείται από ένα μόνο γράμμα όπως Ε ή Α ή ένα οποιοδήποτε τυχαίο όνομα που δεν ανάγει στο πραγματικό περιεχόμενο της μεταβλητής όπως Τιμή.

7.5 | Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές που υποστηρίζονται από τη ΓΛΩΣΣΑ καλύπτουν τις βασικές πράξεις: πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση, ενώ υποστηρίζεται και η ύψωση σε δύναμη, η ακέραια διαίρεση και το υπόλοιπο της ακέραιας διαίρεσης.

Οι τελεστές και οι αντίστοιχες πράξεις είναι:

ΑΡΙΘΜΗΤΙΚΟΣ ΤΕΛΕΣΤΗΣ	ΠΡΑΞΗ
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη
DIV	Ακέραια διαίρεση
MOD	Υπόλοιπο ακέραιας διαίρεσης

Ο τελεστής div χρησιμοποιείται για τον υπολογισμό του ηλίκου μιας διαίρεσης ακεραίων αριθμών, ενώ ο τελεστής mod για το υπόλοιπο.
Π.χ.
 $7 \text{ div } 2 = 3$ και $7 \text{ mod } 2 = 1$

7.6 | Συναρτήσεις

Πολλές γνωστές συναρτήσεις από τα μαθηματικά χρησιμοποιούνται συχνά και περιέχονται στη ΓΛΩΣΣΑ. Οι συναρτήσεις αυτές είναι:

HM(X)	Υπολογισμός ημιτόνου
ΣΥΝ(X)	Υπολογισμός συνημιτόνου
ΕΦ(X)	Υπολογισμός εφαπτομένης
T_P(X)	Υπολογισμός τετραγωνικής ρίζας
ΛΟΓ(X)	Υπολογισμός φυσικού λογαρίθμου
E(X)	Υπολογισμός του e^x
A_M(X)	Ακέραιο μέρος του X
A_T(X)	Απόλυτη τιμή του X

7.7 | Αριθμητικές εκφράσεις

Όταν μια τιμή προκύπτει από υπολογισμό, τότε αναφερόμαστε σε **εκφράσεις** (expressions). Για τη σύνταξη μιας αριθμητικής έκφρασης χρησιμοποιούνται αριθμητικές σταθερές, μεταβλητές, συναρτήσεις, αριθμητικοί τελεστές και παρενθέσεις. Οι αριθμητικές εκφράσεις υλοποιούν απλές ή σύνθετες μαθηματικές πράξεις.

Κάθε έκφραση παριστάνει μια συγκεκριμένη αριθμητική τιμή, η οποία βρίσκεται μετά την εκτέλεση των πράξεων. Γι' αυτό είναι απαραίτητο όλες οι μεταβλητές που εμφανίζονται σε μια έκφραση να έχουν οριστεί προηγουμένα, δηλαδή να έχουν κάποια τιμή.



Πάντα πρέπει να χρησιμοποιούνται ζεύγη παρενθέσεων. Διαφορετικός αριθμός αριστερών από δεξιές παρενθέσεις στην ίδια έκφραση είναι ένα από τα πιο συνηθισμένα λάθη.

Ιεραρχία

Οι πράξεις που παρουσιάζονται σε μια έκφραση εκτελούνται σύμφωνα με την επόμενη ιεραρχία

1. Ύψωση σε δύναμη
2. Πολλαπλασιασμός και διαίρεση
3. Πρόσθεση και αφαίρεση

Παραδείγματα

Μαθηματικά	ΓΛΩΣΣΑ
$a + 1$	$a + 1$
$1/2 a^3$	$1/2 * a^3$
$\frac{3x + 2y}{a - b}$	$(3 * x + 2 * y) / (a - b)$
$2 \eta \mu \chi$	$2 * \text{HM}(\chi)$

Όταν η ιεραρχία είναι ίδια, τότε οι πράξεις εκτελούνται από τ' αριστερά προς τα δεξιά. Σε πολλές όμως περιπτώσεις είναι απαραίτητο να προηγηθεί μια πράξη χαμηλότερης ιεραρχίας. Αυτό επιτυγχάνεται με την εισαγωγή των παρενθέσεων. Η πράξη που πρέπει να προηγηθεί περικλείεται σε ένα ζεύγος παρενθέσεων, οπότε και εκτελείται πρώτη. Π.χ. η έκφραση $2 + 3 * 4$ δίδει ως αποτέλεσμα 14, ενώ η $(2 + 3) * 4$ δίδει 20, διότι εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός.

7.8 | Εντολή εκχώρησης

Η εντολή εκχώρησης χρησιμοποιείται για την απόδοση τιμών στις μεταβλητές κατά τη διάρκεια εκτέλεσης του προγράμματος.



Σύνταξη

Όνομα-Μεταβλητής <- έκφραση

Παραδείγματα

```
A <- 132
ΜΗΝΑΣ <- 'Ιανουάριος'
ΕΜΒΑΔΟΝ <- A * B
```

Λειτουργία

Υπολογίζεται η τιμή της έκφρασης στη δεξιά πλευρά και εκχωρείται η τιμή αυτή στη μεταβλητή, που αναφέρεται στην αριστερή πλευρά.



Σε μια εντολή εκχώρησης η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου.

Μια εντολή εκχώρησης σε καμία περίπτωση δεν πρέπει να εκλαμβάνεται ως εξίσωση. Στην εξίσωση το αριστερό μέλος ισούται με το δεξιό, ενώ στην εντολή εκχώρησης η τιμή του δεξιού μέλους εκχωρείται, μεταβιβάζεται, αποδίδεται στη μεταβλητή του αριστερού μέλους. Για το λόγο αυτό ως τελεστής εκχώρησης χρησιμοποιείται το σύμβολο <- προκειμένου να διαφοροποιείται από το ίσον (=). Ωστόσο, ας σημειωθεί, ότι οι διάφορες γλώσσες προγραμματισμού χρησιμοποιούν διαφορετικά σύμβολα για το σκοπό αυτό.

7.9 | Εντολές εισόδου-εξόδου

Σχεδόν όλα τα προγράμματα υπολογιστή δέχονται κάποια δεδομένα, τα επεξεργάζονται, υπολογίζουν τα αποτελέσματα και τέλος τα εμφανίζουν.

Τα δεδομένα εισάγονται κατά τη διάρκεια της εκτέλεσης του προγράμματος από μία μονάδα εισόδου, για παράδειγμα το πληκτρολόγιο, και τα αποτελέσματα γράφονται σε μία μονάδα εξόδου, για παράδειγμα την οθόνη.

Η **ΓΛΩΣΣΑ** υποστηρίζει για την εισαγωγή δεδομένων από το πληκτρολόγιο την εντολή **ΔΙΑΒΑΣΕ** και για την εμφάνιση των αποτελεσμάτων την εντολή **ΓΡΑΨΕ**.

Σύνταξη

ΔΙΑΒΑΣΕ λίστα-μεταβλητών

Παραδείγματα

ΔΙΑΒΑΣΕ Ποσότητα, Τιμή

Λειτουργία

Η εκτέλεση της εντολής οδηγεί στην είσοδο τιμών από το πληκτρολόγιο και την εκχώρησή τους στις μεταβλητές που αναφέρονται.



Η εντολή **ΔΙΑΒΑΣΕ** ακολουθείται πάντοτε από ένα ή περισσότερα ονόματα μεταβλητών. Αν υπάρχουν περισσότερες από μία μεταβλητές, τότε αυτές χωρίζονται με κόμμα (,). Κατά την εκτέλεση του προγράμματος η εντολή **ΔΙΑΒΑΣΕ** διακόπτει την εκτέλεσή του και το πρόγραμμα περιμένει την εισαγωγή από το πληκτρολόγιο τιμών, που θα εκχωρηθούν στις μεταβλητές. Μετά την ολοκλήρωση της εντολής η εκτέλεση του προγράμματος συνεχίζεται με την επόμενη εντολή.

Σύνταξη

ΓΡΑΨΕ λίστα-στοιχείων

Παραδείγματα

ΓΡΑΨΕ 'Η τετραγωνική ρίζα του', Α, 'είναι: ', ΡΙΖΑ

Λειτουργία

Χρησιμοποιείται για την εμφάνιση σταθερών τιμών καθώς και των τιμών των μεταβλητών που αναφέρονται στη λίστα.



Η εντολή **ΓΡΑΨΕ** έχει ως αποτέλεσμα την εμφάνιση τιμών στη μονάδα εξόδου. Συσκευή εξόδου μπορεί να είναι η οθόνη του υπολογιστή, ο εκτυπωτής, βοηθητική μνήμη ή γενικά οποιαδήποτε συσκευή εξόδου έχει οριστεί στο πρόγραμμα. Για τα παραδείγματα αυτού του κεφαλαίου θεωρούμε ότι η εμφάνιση γίνεται πάντοτε στην οθόνη. Η λίστα των στοιχείων μπορεί να περιέχει σταθερές τιμές και ονόματα μεταβλητών.

Κατά την εκτέλεση του προγράμματος η εντολή **ΓΡΑΨΕ** προκαλεί την

εμφάνιση στην οθόνη των σταθερών τιμών. Όταν κάποιο όνομα μεταβλητής περιέχεται στη λίστα, τότε αρχικά ανακτάται η τιμή της και στη συνέχεια η τιμή αυτή εμφανίζεται στην οθόνη.

Η χρήση της εντολής ΓΡΑΨΕ είναι κυρίως η εμφάνιση μηνυμάτων από τον υπολογιστή, καθώς και αποτελεσμάτων που περιέχονται στις μεταβλητές.

7.10 | Δομή προγράμματος

Όπως κάθε εντολή ακολουθεί αυστηρούς συντακτικούς κανόνες, έτσι και ολόκληρο το πρόγραμμα έχει αυστηρούς κανόνες για τον τρόπο που δομείται. Η πρώτη εντολή κάθε προγράμματος είναι υποχρεωτικά η επικεφαλίδα του προγράμματος, η οποία είναι η λέξη **ΠΡΟΓΡΑΜΜΑ** ακολουθούμενη από το όνομα του προγράμματος. Το τελευταίο πρέπει να υπακούει στους κανόνες δημιουργίας ονομάτων της **ΓΛΩΣΣΑΣ**.

Στη συνέχεια ακολουθεί το τμήμα δήλωσης των σταθερών του προγράμματος, αν βέβαια το πρόγραμμά μας χρησιμοποιεί σταθερές.

Αμέσως μετά είναι το τμήμα δήλωσης μεταβλητών, όπου δηλώνονται υποχρεωτικά τα ονόματα όλων των μεταβλητών καθώς και ο τύπος τους.

Ακολουθεί το κύριο μέρος του προγράμματος, που περιλαμβάνει όλες τις εκτελέσιμες εντολές. Οι εντολές αυτές περιλαμβάνονται υποχρεωτικά ανάμεσα στις λέξεις **ΑΡΧΗ** και **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ**.

Τέλος, αν το πρόγραμμα χρησιμοποιεί διαδικασίες (βλ. κεφ. 10), αυτές γράφονται μετά το **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ**.

Κάθε εντολή γράφεται σε ξεχωριστή γραμμή. Αν μία εντολή πρέπει να συνεχιστεί και στην επόμενη γραμμή, τότε ο πρώτος χαρακτήρας αυτής της γραμμής πρέπει να είναι ο χαρακτήρας **&**.

Αν ο πρώτος χαρακτήρας είναι το θαυμαστικό (!), σημαίνει ότι αυτή η γραμμή περιέχει σχόλια και όχι εκτελέσιμες εντολές.

ΠΑΡΑΔΕΙΓΜΑ

Το επόμενο πρόγραμμα υπολογίζει το συνολικό κόστος παραγγελιών υπολογιστών. Το πρόγραμμα διαβάζει από το πληκτρολόγιο την ποσότητα της παραγγελίας και την τιμή του ενός υπολογιστή, υπολογίζει και γράφει το συνολικό κόστος καθώς και το αντίστοιχο κόστος του ΦΠΑ. Ο συντελεστής ΦΠΑ είναι 18%.

ΠΡΟΓΡΑΜΜΑ Κόστος_Υπολογιστών

! Πρόγραμμα υπολογισμού κόστους παραγγελίας υπολογιστών
ΣΤΑΘΕΡΕΣ

ΦΠΑ=0.18

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Ποσότητα, Τιμή_μονάδας, Κόστος

ΠΡΑΓΜΑΤΙΚΕΣ: Αξία_ΦΠΑ, Συνολικό_κόστος

ΑΡΧΗ

! Εισαγωγή δεδομένων

ΓΡΑΨΕ 'Δώσε την ποσότητα της παραγγελίας'

ΔΙΑΒΑΣΕ Ποσότητα

```

ΓΡΑΨΕ 'Δώσε την τιμή του υπολογιστή'
ΔΙΑΒΑΣΕ Τιμή_μονάδας
! Υπολογισμοί
Κόστος <- Ποσότητα* Τιμή_μονάδας
Αξία_ΦΠΑ <- Κόστος*ΦΠΑ
Συνολικό_κόστος <- Κόστος+Αξία_ΦΠΑ
! Εμφάνιση αποτελεσμάτων
ΓΡΑΨΕ 'Το κόστος των', Ποσότητα, 'υπολογ. είναι ', Κόστος
ΓΡΑΨΕ ' Η αξία του ΦΠΑ είναι', Αξία_ΦΠΑ
ΓΡΑΨΕ 'Το συνολικό κόστος είναι', Συνολικό_κόστος
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Στη συνέχεια παρουσιάζεται το πρόγραμμα αυτό σε γλώσσες προγραμματισμού Pascal και Basic.

Προγραμματιστικό περιβάλλον Pascal

```

PROGRAM computers;
CONST
    fpa=0.18;
VAR
    cost, value, quantity:INTEGER;
    total,cost_fpa:REAL;
BEGIN
    write('Δώσε την ποσότητα');
    readln(quantity);
    write('Δώσε την τιμή του υπολογιστή');
    readln(value);
    cost:=quantity*value;
    cost_fpa:=cost*fpa;
    total:=cost+cost_fpa;
    writeln ('Το κόστος των ', quantity, 'είναι:', cost);
    writeln ('Η αξία του ΦΠΑ: ', cost_fpa:7:0);
    writeln ('Το συνολικό κόστος είναι: ', total:7:0)
END.

```

Προγραμματιστικό περιβάλλον Basic

```

' Κόστος υπολογιστών
fpa = .18
INPUT "Δώσε την ποσότητα : ", Quantity
INPUT "Δώσε την τιμή του υπολογιστή : ", Value
Cost = Quantity * Value
CostFpa = Cost * fpa
Total = Cost + CostFpa
PRINT "Το κόστος των"; Quantity; "υπολογιστών είναι :";
Cost
PRINT USING "Η αξία ΦΠΑ είναι : #####"; CostFpa
PRINT USING "Το συνολικό κόστος είναι : #####"; Total
END

```

Λέξεις-κλειδιά

Πρόγραμμα,
Τύποι δεδομένων,
Μεταβλητή,
Σταθερά,
Εντολή,
Εκχώρηση τιμής,
Είσοδος-έξοδος
προγράμματος

Ανακεφαλαίωση

Σε αυτό το κεφάλαιο παρουσιάστηκαν τα βασικά χαρακτηριστικά της **ΓΛΩΣΣΑΣ**, το αλφάβητό της, οι τύποι δεδομένων που υποστηρίζει, οι κανόνες για τα ονόματα που χρησιμοποιούνται, οι αριθμητικές πράξεις, η εντολή εκχώρησης, οι εντολές εισόδου και εξόδου καθώς και η δομή που πρέπει να έχει κάθε πρόγραμμα.

Συγκεκριμένα:

- Οι τύποι δεδομένων που υποστηρίζονται είναι: Ακέραιοι, Πραγματικοί, Χαρακτήρες, Λογικοί.
- Οι μεταβλητές πρέπει να έχουν τον τύπο των δεδομένων που καταχωρούν.
- Κάθε μεταβλητή παίρνει τιμή με εντολή εκχώρησης ή με εντολή **ΔΙΑΒΑΣΕ**.
- Κάθε πρόγραμμα έχει τον τίτλο του, ακολουθεί το τμήμα δηλώσεων (σταθερών και μεταβλητών) και μετά ανάμεσα στις λέξεις **ΑΡΧΗ** και **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ** βρίσκονται όλες οι εκτελέσιμες εντολές.
- Η επικοινωνία του προγράμματος με τον χρήστη γίνεται με τις εντολές εισόδου εξόδου **ΓΡΑΨΕ** και **ΔΙΑΒΑΣΕ**.

Διαδίκτυο

<http://qbasic.com/>

Περιέχει εκπαιδευτικό οδηγό, κώδικα πολλών ασκήσεων και γενικές πληροφορίες για την Qbasic.

Ερωτήσεις - Θέματα για συζήτηση

1. Ποιους τύπους δεδομένων γνωρίζετε; Αναφέρατε δύο παραδείγματα για κάθε τύπο.
2. Σε ποια θέση του προγράμματος αναγράφονται οι δηλώσεις των σταθερών;
3. Ποια η διαφορά μεταβλητών και σταθερών;
4. Ποια η σειρά εκτέλεσης των πράξεων;
5. Ποιος ο σκοπός των εντολών εισόδου εξόδου;
6. Ποια η διαφορά των εντολών **ΔΙΑΒΑΣΕ** και **ΓΡΑΨΕ**;
7. Περιγράψτε τη δομή ενός προγράμματος.

Βιβλιογραφία

1. Θ. Αλεβίζος, Α. Καμπουρέλης, *Εισαγωγή με τη γλώσσα Pascal*, Αθήνα, 1984.
2. Γ. Βουτυράς, *Basic: Αλγόριθμοι και εφαρμογές*, Κλεψύδρα, Αθήνα, 1991.
3. Χρ. Κοίλιας, *Η QuickBasic και οι εφαρμογές της*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
4. R. Shackelford, *Introduction to Computing and Algorithms*, Addison-Wesley, USA, 1998.
5. S. Leestma-L.Nyhoff, *Turbo Pascal, Programming and Solving*, McMillan, New York, 1990.
6. N. Wirth, *Systematic Programming: An introduction*, Prentice Hall, 1973.

www.basicguru.com

Διεύθυνση που αναφέρεται αποκλειστικά στη Basic. Περιέχει πολλά έτοιμα παραδείγματα, πληροφορίες για εκδόσεις της γλώσσας, μεταφραστές για διάφορα λειτουργικά συστήματα.

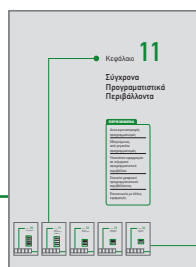
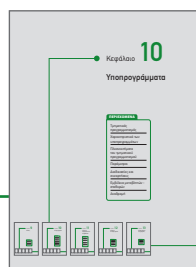
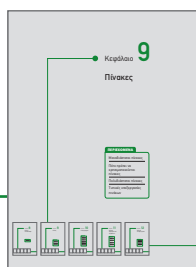
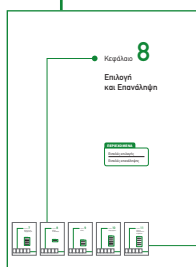
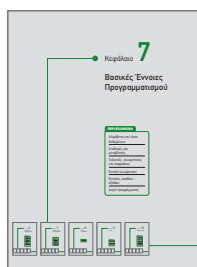
Κεφάλαιο 8

Επιλογή και Επανάληψη

ΠΕΡΙΕΧΟΜΕΝΑ

Εντολές επιλογής

Εντολές επανάληψης



Εισαγωγή

Στο προηγούμενο κεφάλαιο αναπτύξαμε προγράμματα, τα οποία ήταν πολύ απλά και οι εντολές των οποίων εκτελούνται η μία μετά την άλλη. Αυτή η σειριακή εκτέλεση των εντολών είναι κατάλληλη όμως μόνο για πολύ απλά προγράμματα, τα οποία εισάγουν δεδομένα, τα επεξεργάζονται και τυπώνουν το αποτέλεσμα, χωρίς να υπάρχει η δυνατότητα της επιλεκτικής εκτέλεσης τμημάτων του προγράμματος, σύμφωνα με την τιμή κάποιων δεδομένων ή την επανάληψη τμημάτων του προγράμματος. Όπως έχουμε αναφέρει, οι τρεις βασικές δομές είναι η δομή της ακολουθίας, της επιλογής και της επανάληψης. Οι δομές αυτές αποτελούν τη βάση του δομημένου προγραμματισμού και με τη χρήση αυτών μπορούν να υλοποιηθούν όλα τα προγράμματα υπολογιστών. Στο κεφάλαιο αυτό θα ασχοληθούμε με τις δύο αυτές βασικές δομές της επιλογής και της επανάληψης που θα μας επιτρέψουν τη συγγραφή πληρέστερων και πιο πολύπλοκων προγραμμάτων.

Διδακτικοί στόχοι

Να είναι σε θέση ο μαθητής:

Να σχηματίζει λογικές εκφράσεις, απλές και σύνθετες.

Να διατυπώνει τις μορφές της εντολής ελέγχου (επιλογής) ΑΝ.

Να διακρίνει τις διαφορές των μορφών της εντολής ΑΝ.

Να επιλέγει την καλύτερη μορφή της εντολής ΑΝ για το κάθε πρόγραμμα.

Να διατυπώνει τις εντολές επανάληψης.

Να επιλέγει την καλύτερη δομή επανάληψης και να χρησιμοποιεί την κατάλληλη εντολή.

Να συντάσσει προγράμματα τα οποία χρησιμοποιούν και τις τρεις βασικές δομές: της ακολουθίας, της επιλογής και της επανάληψης.

Προερωτήσεις

Γιατί χρησιμοποιούνται οι αλγοριθμικές δομές;

Νομίζεις ότι όλες οι αλγοριθμικές δομές έχουν τις αντίστοιχες εντολές σε μία γλώσσα προγραμματισμού;

Η δομή της επιλογής είναι σημαντική για την επίλυση προβλημάτων;

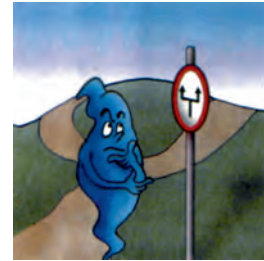
Αρκεί μία εντολή για να εκφράσει τη δομή της επανάληψης;

Πώς μπορεί να ελέγχεται ο τερματισμός μίας επανάληψης;

8.1 | Εντολές Επιλογής

Μία από τις βασικότερες δομές που εμφανίζονται σε ένα πρόγραμμα είναι η επιλογή. Σχεδόν σε όλα τα προβλήματα περιλαμβάνονται κάποιοι έλεγχοι και ανάλογα με το αποτέλεσμα αυτών των ελέγχων επιλέγονται οι ενέργειες που θα ακολουθήσουν.

Ας θεωρήσουμε το πολύ απλό πρόβλημα της καταμέτρησης των θετικών και των αρνητικών αριθμών. Πρέπει λοιπόν να γράψουμε ένα πρόγραμμα, το οποίο εισάγει αριθμούς και μετράει πόσοι από αυτούς είναι θετικοί και πόσοι αρνητικοί. Για να αποφασίσουμε αν ένας αριθμός είναι θετικός ή αρνητικός, πρέπει να τον συγκρίνουμε με το 0. Το αποτέλεσμα αυτής της σύγκρισης καθορίζει το είδος του αριθμού, αν είναι μεγαλύτερος από το 0, τότε ο αριθμός είναι θετικός, ενώ αντίθετα αν είναι μικρότερος από το 0, είναι αρνητικός.



Λογική Έκφραση

Για τη σύνταξη μιας λογικής έκφρασης ή συνθήκης χρησιμοποιούνται σταθερές, μεταβλητές, αριθμητικές παραστάσεις, συγκριτικοί και λογικοί τελεστές, καθώς και παρενθέσεις. Στις λογικές εκφράσεις γίνεται σύγκριση της τιμής μίας έκφρασης, που βρίσκεται αριστερά από το συγκριτικό τελεστή με την τιμή μιας άλλης έκφρασης που βρίσκεται δεξιά. Το αποτέλεσμα είναι μία λογική τιμή **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**.

Οι χρησιμοποιούμενοι συγκριτικοί τελεστές παρουσιάζονται στον επόμενο πίνακα.

ΣΥΓΚΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ		
Τελεστής	Ελεγχόμενη σχέση	Παράδειγμα
=	Ισότητα	Αριθμός=0
<>	Ανισότητα	Όνομα1 <> 'Κώστας'
>	Μεγαλύτερο από	Τιμή >10000
>=	Μεγαλύτερο ή ίσο	$X+Y \geq (A+B)/\Gamma$
<	Μικρότερο από	$B^2-4*A*\Gamma < 0$
<=	Μικρότερο ή ίσο	Βάρος <= 500

Οι συγκρίσεις γίνονται σε δεδομένα αριθμητικά, αλφαριθμητικά και λογικά.

Η σύγκριση μεταξύ δύο αριθμών γίνεται με προφανή τρόπο. Στην περίπτωση των πραγματικών αριθμών θεωρούμε ότι οι αριθμοί μπορούν να έχουν άπειρο αριθμό ψηφίων.

Η σύγκριση ατομικών χαρακτήρων στηρίζεται στην αλφαβητική σειρά, για παράδειγμα το 'α' θεωρείται μικρότερο από το 'β'.

Η σύγκριση αλφαριθμητικών δεδομένων βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση μέχρις ότου βρεθεί κάποια διαφορά, για παράδειγμα η λέξη 'κακός' θεωρείται μικρότερη από τη λέξη 'καλός' αφού το γράμμα κ προηγείται του γράμματος λ.

Η σύγκριση λογικών έχει έννοια μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>), αφού οι τιμές που μπορούν να έχουν είναι **ΑΛΗΘΗΣ** και **ΨΕΥΔΗΣ**.



Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μια έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες. Ακόμη, οι λογικοί τελεστές έχουν χαμηλότερη ιεραρχία από τους συγκριτικούς.

Σύνθετες Εκφράσεις

Σε πολλά προβλήματα οι επιλογές δεν αρκεί να γίνονται με απλές λογικές παραστάσεις όπως αυτές οι οποίες αναφέρθηκαν, αλλά χρειάζεται να συνδυαστούν μία ή περισσότερες λογικές παραστάσεις. Αυτό επιτυγχάνεται με τη χρήση των τριών βασικών λογικών τελεστών ΟΧΙ, ΚΑΙ, Ή.

Παραδείγματα

$0 < X < 5$

$X > 0$ ΚΑΙ $X < 5$

$X = 1$ ή 2 ή 3

$X = 1$ Ή $X = 2$ Ή $X = 3$

Η ιεραρχία των λογικών τελεστών είναι μικρότερη των αριθμητικών.

8.1.1 Εντολή AN

Η δομή επιλογής υλοποιείται στη ΓΛΩΣΣΑ με την εντολή **AN**. Η εντολή **AN** εμφανίζεται με τρεις διαφορετικές μορφές. Την απλή εντολή **AN...ΤΟΤΕ**, την εντολή **AN...ΤΟΤΕ..ΑΛΛΙΩΣ** και τέλος την εντολή **AN...ΤΟΤΕ..ΑΛΛΙΩΣ_ΑΝ**. Κάθε εντολή **AN** πρέπει να κλείνει με **ΤΕΛΟΣ_ΑΝ**.

Στην απλούστερη μορφή της η εντολή **AN** ελέγχει τη συνθήκη και αν αυτή ισχύει (είναι αληθής), τότε εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΤΕΛΟΣ_ΑΝ**.

Αν για παράδειγμα θέλουμε να υπολογίσουμε την τετραγωνική ρίζα των αριθμών που διαβάζουμε από το πληκτρολόγιο, τότε το αντίστοιχο τμήμα προγράμματος είναι

```
ΔΙΑΒΑΣΕ α
ΑΝ α >=0 ΤΟΤΕ
    Ρίζα <- Τ_Ρ(α)
ΤΕΛΟΣ_ΑΝ
```

Η γενική μορφή της εντολής **AN** είναι η εξής:

Σύνταξη

```
ΑΝ συνθήκη ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα

```
ΑΝ αριθμός > 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ο αριθμός είναι θετικός'
    Πλήθος_θετικών <- Πλήθος_θετικών+1
ΤΕΛΟΣ_ΑΝ
```

Λειτουργία

Αν η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΤΕΛΟΣ_ΑΝ**, σε αντίθετη περίπτωση οι εντολές αυτές αγνοούνται. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ_ΑΝ**.

Συχνά η εντολή **AN** εκτός από το τμήμα των εντολών, που εκτελούνται όταν η λογική έκφραση είναι Αληθής, περιέχει και το τμήμα των εντολών που εκτελούνται, αν δεν ισχύει η συνθήκη (είναι Ψευδής).

Η μορφή αυτής της εντολής ονομάζεται **AN...ΤΟΤΕ...ΑΛΛΙΩΣ**.

Στο παράδειγμα του υπολογισμού της τετραγωνικής ρίζας έχουμε

```
ΔΙΑΒΑΣΕ α
AN α >= 0 ΤΟΤΕ
    Ρίζα <- T_P(α)
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Η τετρ. ρίζα αρνητικού αριθμού δεν ορίζεται'
ΤΕΛΟΣ_ΑΝ
```

Η γενική μορφή της εντολής **AN...ΤΟΤΕ...ΑΛΛΙΩΣ** έχει ως εξής:

Σύνταξη

```
AN συνθήκη ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΑΛΛΙΩΣ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα

```
AN αριθμός > 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ο αριθμός είναι θετικός'
    Πλήθος_θετικών <- Πλήθος_θετικών+1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Ο αριθμός είναι αρνητικός ή 0'
    Πλήθος_μη_θετικών <- Πλήθος_μη_θετικών +1
ΤΕΛΟΣ_ΑΝ
```

Λειτουργία

Αν η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΑΛΛΙΩΣ**, διαφορετικά εκτελούνται οι εντολές μεταξύ **ΑΛΛΙΩΣ** και **ΤΕΛΟΣ_ΑΝ**. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ_ΑΝ**.

Η γενική μορφή της εντολής **AN** καλύπτει την επιλογή μιας από δύο εναλλακτικές περιπτώσεις.

Όταν οι εναλλακτικές περιπτώσεις είναι περισσότερες από τις δύο, τότε μπορούν να χρησιμοποιηθούν πολλές εντολές **AN** ή μία μέσα στην άλλη, οι εμφωλευμένες εντολές **AN**, όπως ονομάζονται.

ΟΡΙΣΜΟΣ

Εμφωλευμένα **ΑΝ** ονομάζονται δύο ή περισσότερες εντολές της μορφής **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ** που περιέχονται η μία μέσα στην άλλη.

Για παράδειγμα, οι παρακάτω εντολές προγράμματος

```
////////
ΔΙΑΒΑΣΕ Βάρος, Ύψος
ΑΝ Βάρος < 80 ΤΟΤΕ
  ΑΝ Ύψος < 1.70 ΤΟΤΕ
    ΓΡΑΨΕ 'Ελαφρύς, κοντός'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
////////
```

Η χρήση εμφωλευμένων εντολών **ΑΝ** οδηγεί συνήθως σε πολύπλοκες δομές που αυξάνουν την πιθανότητα του λάθους καθώς και τη δυσκολία κατανόησης του προγράμματος.

Πολύ συχνά οι εντολές που έχουν γραφεί με εμφωλευμένα **ΑΝ**, μπορούν να γραφούν πιο απλά χρησιμοποιώντας σύνθετες εκφράσεις ή την εντολή επιλογής **ΑΝ ... ΤΟΤΕ ... ΑΛΛΙΩΣ_ΑΝ**, που θα παρουσιαστεί στη συνέχεια.

Το προηγούμενο τμήμα προγράμματος μπορεί να γραφεί ως εξής

```
////////
ΔΙΑΒΑΣΕ Βάρος, Ύψος
ΑΝ Βάρος < 80 ΚΑΙ Ύψος < 1.70 ΤΟΤΕ
  ΓΡΑΨΕ 'Ελαφρύς, κοντός'
ΤΕΛΟΣ_ΑΝ
////////
```

Μία άλλη μορφή επιλογής είναι η εντολή **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ_ΑΝ**

Σύνταξη

```
ΑΝ συνθήκη-1 ΤΟΤΕ
  εντολή-1
  εντολή-2
  ...
  εντολή-ν
ΑΛΛΙΩΣ_ΑΝ συνθήκη-2 ΤΟΤΕ
  εντολή-1
  εντολή-2
  ...
  εντολή-ν
...
ΑΛΛΙΩΣ
  εντολή-1
  εντολή-2
  ...
  εντολή-ν
ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα

```
ΑΝ αριθμός > 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ο αριθμός είναι θετικός'
    Πλήθος_θετικών <- Πλήθος_θετικών+1
ΑΛΛΙΩΣ_ΑΝ αριθμός < 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ο αριθμός είναι αρνητικός '
    Πλήθος_αρνητικών <- Πλήθος_ αρνητικών +1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Ο αριθμός είναι 0'
    Πλήθος_0 <- Πλήθος_0 +1
ΤΕΛΟΣ_ΑΝ
```

Λειτουργία

Εκτελούνται οι εντολές που βρίσκονται στο αντίστοιχο τμήμα, όταν η συνθήκη είναι αληθής.

Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ_ΑΝ**

ΠΑΡΑΔΕΙΓΜΑ 1

Στο πρόγραμμα του προηγούμενου κεφαλαίου (πωλήσεις υπολογιστών) υποθέτουμε ότι η τιμή των υπολογιστών εξαρτάται από την ποσότητα παραγγελίας. Συγκεκριμένα ισχύουν οι παρακάτω τιμές αγοράς υπολογιστών.

ΠΟΣΟΤΗΤΑ	ΤΙΜΗ ΜΟΝΑΔΑΣ
1-50	580
51-100	520
101-200	470
πάνω από 200	440

Ο υπολογισμός με χρήση εμφωλευμένων εντολών ΑΝ είναι:

```
ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*580
ΑΛΛΙΩΣ
    ΑΝ Ποσότητα<=100 ΤΟΤΕ
        Κόστος <- Ποσότητα*520
    ΑΛΛΙΩΣ
        ΑΝ Ποσότητα<=200 ΤΟΤΕ
            Κόστος <- Ποσότητα*470
        ΑΛΛΙΩΣ
            Κόστος <- Ποσότητα*440
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
```



Να αποφεύγεται, αν είναι δυνατόν, η χρήση των εμφωλευμένων ΑΝ, και στη θέση τους να χρησιμοποιούνται απλούστερες δομές που διευκολύνουν την ανάγνωση και την κατανόηση του προγράμματος.



Στην πρώτη εντολή ΑΛΛΙΩΣ_ΑΝ ο έλεγχος της συνθήκης Ποσότητα>50 είναι περιττός.

Το ίδιο πρόγραμμα με τη χρήση της εντολής ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ_ΑΝ έχει ως εξής:

```
ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*580
ΑΛΛΙΩΣ_ΑΝ Ποσότητα<=100 ΤΟΤΕ
    Κόστος <- Ποσότητα*520
ΑΛΛΙΩΣ_ΑΝ Ποσότητα<=200 ΤΟΤΕ
    Κόστος <- Ποσότητα*470
ΑΛΛΙΩΣ
    Κόστος <- Ποσότητα*440
ΤΕΛΟΣ_ΑΝ
```

Ένα συχνό λάθος που παρατηρείται στα προγράμματα είναι ο έλεγχος περιττών συνθηκών. Οι επιπλέον έλεγχοι αυξάνουν την πολυπλοκότητα του προγράμματος.

Στο προηγούμενο παράδειγμα για το οποίο θεωρούμε ότι η ποσότητα είναι θετικός αριθμός, ένα παράδειγμα περιττών ελέγχων είναι το ακόλουθο:

```
ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*580
ΑΛΛΙΩΣ_ΑΝ Ποσότητα>50 ΚΑΙ Ποσότητα <=100 ΤΟΤΕ
    Κόστος <- Ποσότητα*520
ΑΛΛΙΩΣ_ΑΝ Ποσότητα>100 ΚΑΙ Ποσότητα <=200 ΤΟΤΕ
    Κόστος <- Ποσότητα*470
ΑΛΛΙΩΣ
    Κόστος <- Ποσότητα*440
ΤΕΛΟΣ_ΑΝ
```

8.1.2 Εντολή ΕΠΙΛΕΞΕ

Αν οι εναλλακτικές περιπτώσεις επιλογής είναι πολλές, μπορεί να χρησιμοποιηθεί η εντολή ΕΠΙΛΕΞΕ, η γενική μορφή της οποίας είναι:

Σύνταξη

```
ΕΠΙΛΕΞΕ έκφραση
    ΠΕΡΙΠΤΩΣΗ λίστα_τιμών_1
        εντολές_1
    ΠΕΡΙΠΤΩΣΗ λίστα_τιμών_2
        εντολές_2
    .....
    ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
        εντολές_αλλιώς
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
```

Παράδειγμα

```
ΔΙΑΒΑΣΕ αριθμός
ΕΠΙΛΕΞΕ αριθμός
    ΠΕΡΙΠΤΩΣΗ 0
        ΓΡΑΨΕ 'Μηδέν'
```

ΠΕΡΙΠΤΩΣΗ 1, 3, 5, 7, 9

ΓΡΑΨΕ 'Μονός αριθμός'

ΠΕΡΙΠΤΩΣΗ 2, 4, 6, 8

ΓΡΑΨΕ 'Ζυγός '

ΠΕΡΙΠΤΩΣΗ **ΑΛΛΙΩΣ**

ΓΡΑΨΕ 'Αριθμός < 0 ή >9 ή όχι ακέραιος'

ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

Λειτουργία

Υπολογίζεται η τιμή της έκφρασης και εκτελούνται οι εντολές που ανήκουν στην αντίστοιχη περίπτωση τιμών. Αν η τιμή της έκφρασης δεν αντιστοιχεί σε καμία περίπτωση, τότε εκτελούνται οι εντολές αλλιώς.

Στην εντολή αυτή οι λίστες τιμών που συνοδεύουν κάθε περίπτωση μπορούν να περιλαμβάνουν μία ή περισσότερες τιμές ή περιοχή τιμών από-έως.

Η χρήση της εντολής **ΕΠΙΛΕΞΕ** λόγω της συμπαγούς δομής της προσφέρει σημαντικά πλεονεκτήματα στον προγραμματισμό.

8.2 | Εντολές επανάληψης

Η τρίτη βασική δομή είναι η δομή επανάληψης, ο βρόχος, η οποία επιτρέπει την εκτέλεση εντολών περισσότερες από μία φορά. Οι επαναλήψεις ελέγχονται πάντοτε από κάποια συνθήκη, η οποία καθορίζει την έξοδο από το βρόχο.

Η **ΓΛΩΣΣΑ** υποστηρίζει τρεις εντολές επανάληψης, την εντολή **ΟΣΟ** όπου η επανάληψη ελέγχεται από μία λογική έκφραση στην αρχή και εκτελείται συνεχώς όσο η συνθήκη είναι Αληθής, την εντολή **ΜΕΧΡΙΣ_ΟΤΟΥ** όπου η συνθήκη βρίσκεται στο τέλος του βρόχου και εκτελείται συνεχώς μέχρις ότου η συνθήκη αυτή γίνει Αληθής και τέλος την εντολή **ΓΙΑ**, με την οποία ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό φορών.



8.2.1 Εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

Η γενικότερη δομή επανάληψης υλοποιείται στη **ΓΛΩΣΣΑ** με την εντολή **ΟΣΟ... ΕΠΑΝΑΛΑΒΕ**. Σε αυτή, η συνθήκη που ελέγχει την επανάληψη βρίσκεται στην αρχή της επανάληψης και ο βρόχος επαναλαμβάνεται συνεχώς, όσο η συνθήκη αυτή ισχύει. Με τη δομή αυτή μπορούν να εκφραστούν όλες οι επαναλήψεις και γι' αυτό η εντολή **ΟΣΟ... ΕΠΑΝΑΛΑΒΕ** είναι η σημαντικότερη από όλες τις εντολές επανάληψης. Χαρακτηριστικό της επανάληψης αυτής είναι ότι ο αριθμός των επαναλήψεων δεν είναι γνωστός, ούτε μπορεί να υπολογιστεί πριν από την εκτέλεση του προγράμματος.

Σύνταξη

```
ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Παράδειγμα

```
Αθροισμα <- 0
ΟΣΟ Αθροισμα < 1000 ΕΠΑΝΑΛΑΒΕ
    ΔΙΑΒΑΣΕ Α
    Αθροισμα <- Αθροισμα + Α
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Λειτουργία

Ελέγχεται η συνθήκη και αν είναι Αληθής, εκτελούνται οι εντολές που βρίσκονται ανάμεσα στις **ΟΣΟ_ΕΠΑΝΑΛΑΒΕ** και **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**. Στη συνέχεια ελέγχεται πάλι η συνθήκη και αν ισχύει, εκτελούνται πάλι οι ίδιες εντολές. Όταν η λογική έκφραση γίνει Ψευδής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**.

Εφόσον μετά από κάθε επανάληψη ελέγχεται εκ νέου η συνθήκη, πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μία εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Σε αντίθετη περίπτωση η επανάληψη δε θα τερματίζεται και θα εκτελείται συνεχώς.

ΠΑΡΑΔΕΙΓΜΑ 2

Να γραφεί πρόγραμμα το οποίο διαβάζει από το πληκτρολόγιο μία σειρά μετρήσεων, ακεραίων μη μηδενικών αριθμών, υπολογίζει και τυπώνει το άθροισμά τους καθώς και το μέσο τους όρο. Ως τέλος της διαδικασίας εισαγωγής στοιχείων χρησιμοποιείται η τιμή 0.

```
ΠΡΟΓΡΑΜΜΑ Αθροισμα
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Χ, Αθροισμα, Πλήθος
    ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
ΑΡΧΗ
    Πλήθος <- 0
    Αθροισμα <- 0
    ΓΡΑΨΕ 'Δώσε Αριθμό'
    ΔΙΑΒΑΣΕ Χ
    ΟΣΟ Χ <> 0 ΕΠΑΝΑΛΑΒΕ
        Αθροισμα <- Αθροισμα + Χ
        Πλήθος <- Πλήθος + 1
        ΓΡΑΨΕ 'Δώσε Αριθμό'
        ΔΙΑΒΑΣΕ Χ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```

ΑΝ Πλήθος > 0 ΤΟΤΕ
    ΜΟ <- Αθροισμα/Πλήθος
    ΓΡΑΨΕ 'Το Αθροισμα είναι : ', Αθροισμα
    ΓΡΑΨΕ 'Ο Μέσος όρος είναι : ', ΜΟ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Δεν δόθηκαν στοιχεία'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Παρατηρήσεις

Η χρήση τιμών για τον τερματισμό μίας επαναληπτικής διαδικασίας, όπως στο παράδειγμα η αυθαίρετη επιλογή του 0, είναι συνήθης στον προγραμματισμό.

Η τιμή αυτή ορίζεται από τον προγραμματιστή και αποτελεί μια σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή είναι τέτοια, ώστε να μην είναι λογικά σωστή για το πρόβλημα, για παράδειγμα η τιμή 0 αποκλείεται από τις μετρήσεις σύμφωνα με την εκφώνηση του παραδείγματος. Η τιμή αυτή συχνά αποκαλείται "τιμή φρουρός".

8.2.2 Εντολή ΜΕΧΡΙΣ_ΟΤΟΥ

Η δεύτερη εντολή επανάληψης που χρησιμοποιεί η **ΓΛΩΣΣΑ** είναι η εντολή **ΜΕΧΡΙΣ_ΟΤΟΥ**. Σε αυτή οι εντολές του βρόχου εκτελούνται μέχρις ότου ικανοποιηθεί κάποια συνθήκη η οποία ελέγχεται στο τέλος της επανάληψης.

Σύνταξη

```

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΜΕΧΡΙΣ_ΟΤΟΥ λογική-έκφραση

```

Παράδειγμα

```

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ Α
    Αθροισμα <- Αθροισμα + Α
ΜΕΧΡΙΣ_ΟΤΟΥ Αθροισμα >= 1000

```

Λειτουργία

Εκτελούνται οι εντολές μεταξύ των **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**. Στη συνέχεια ελέγχεται η λογική έκφραση και αν δεν ισχύει (είναι ψευδής), τότε οι εντολές που βρίσκονται ανάμεσα στις **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**, εκτελούνται πάλι. Ελέγχεται ξανά η λογική έκφραση και αν δεν ισχύει, επαναλαμβάνεται η εκτέλεση των ίδιων εντολών.

Όταν η λογική έκφραση γίνει Αληθής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά από την **ΜΕΧΡΙΣ_ΟΤΟΥ**.

Πολύ συχνά η ίδια επαναληπτική διαδικασία μπορεί να γραφεί εξίσου σωστά χρησιμοποιώντας είτε τη δομή **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ** είτε τη δομή **ΜΕΧΡΙΣ_ΟΤΟΥ** και είναι προσωπική επιλογή του προγραμματιστή ποια από τις δυο θα χρησιμοποιήσει. Υπάρχουν όμως περιπτώσεις όπου η χρήση της εντολής **ΜΕΧΡΙΣ_ΟΤΟΥ** οδηγεί σε απλούστερα και πιο ευκολονόητα προγράμματα. Γενικά σε περιπτώσεις όπου η επανάληψη θα συμβεί υποχρεωτικά μία φορά, είναι προτιμότερη η χρήση της **ΜΕΧΡΙΣ_ΟΤΟΥ**.

Χαρακτηριστική περίπτωση όπου προτιμάται η εντολή **ΜΕΧΡΙΣ_ΟΤΟΥ** είναι στον έλεγχο αποδεκτών τιμών καθώς και στην επιλογή από προκαθορισμένες απαντήσεις ή μενού.

ΠΑΡΑΔΕΙΓΜΑ 3

Στο προηγούμενο παράδειγμα ας υποθέσουμε ότι οι μετρήσεις είναι υποχρεωτικά θετικοί αριθμοί και ότι μετά την εισαγωγή κάθε αριθμού υπάρχει η ερώτηση αν θα εισαγάγουμε άλλο. Η διαδικασία θα τελειώσει, όταν η απάντηση θα είναι Όχι (ο ή Ο).

ΠΡΟΓΡΑΜΜΑ Άθροισμα2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: X, Άθροισμα, Πλήθος

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΧΑΡΑΚΤΗΡΕΣ: Επιλογή

ΑΡΧΗ

Πλήθος <- 0

Άθροισμα <- 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

! Έλεγχος δεδομένων

ΓΡΑΨΕ 'Δώσε Αριθμό'

ΔΙΑΒΑΣΕ X

ΑΝ X<0 **ΤΟΤΕ**

ΓΡΑΨΕ 'Λάθος Αριθμός, Παρακαλώ δώστε ξανά...'

ΤΕΛΟΣ_ΑΝ

! Αν το X δεν είναι θετικό εισάγουμε νέο αριθμό

ΜΕΧΡΙΣ_ΟΤΟΥ X>0

Άθροισμα <- Άθροισμα+X

Πλήθος <- Πλήθος+1

ΓΡΑΨΕ 'Νέα μέτρηση ;'

ΔΙΑΒΑΣΕ Επιλογή

! Αν η επιλογή είναι Ο ή ο τότε σταματάει η επανάληψη

ΜΕΧΡΙΣ_ΟΤΟΥ Επιλογή='Ο' **Η** Επιλογή='ο'

ΜΟ <- Άθροισμα/Πλήθος

ΓΡΑΨΕ 'Άθροισμα =', Άθροισμα

ΓΡΑΨΕ 'Μέσος όρος =', ΜΟ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Η εντολή επανάληψης **ΜΕΧΡΙΣ_ΟΤΟΥ** εκτελείται υποχρεωτικά τουλάχιστον μία φορά.

Προγραμματιστικό περιβάλλον Pascal

```
PROGRAM athroisma2;
VAR
  x, sum, count: INTEGER;
  avg: REAL;
  choice: CHAR;
BEGIN
  count:=0; sum:=0;
  REPEAT
    REPEAT
      write('Δώσε αριθμό:');
      readln (x);
      IF x<=0 THEN
        writeln ('Λάθος αριθμός, Δώσε ξανά..');
      UNTIL x>0;
      sum:=sum+x;
      count:=count+1;
      write('Νέα μέτρηση;');
      readln (choice);
    UNTIL (choice='ο') OR (choice='Ο');
    avg:=sum/count;
    writeln('Αθροισμα: ', sum:5);
    writeln('Μέσος όρος: ', avg:6:2);
  END
```

8.2.3 Εντολή ΓΙΑ.. ΑΠΟ...ΜΕΧΡΙ

Πολύ συχνά ο αριθμός των επαναλήψεων που πρέπει να εκτελεστούν είναι γνωστός από την αρχή. Αν και αυτού του είδους οι επαναλήψεις μπορούν να αντιμετωπιστούν με τη χρήση των προηγούμενων εντολών επανάληψης, η **ΓΛΩΣΣΑ** διαθέτει και την εντολή **ΓΙΑ**. Η εντολή αυτή χειρίζεται μια μεταβλητή, στην οποία αρχικά εκχωρείται η αρχική τιμή. Η τιμή της μεταβλητής συγκρίνεται με την τελική τιμή και εφόσον είναι μικρότερη από αυτή, τότε εκτελούνται οι εντολές που βρίσκονται στο βρόχο (ανάμεσα στις εντολές **ΓΙΑ** και **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**). Στη συνέχεια η μεταβλητή ελέγχου αυξάνεται κατά την τιμή που ορίζει το **ΒΗΜΑ**. Αν η νέα τιμή είναι μικρότερη της τελικής, τότε ο βρόχος εκτελείται ξανά. Η διαδικασία αυτή επαναλαμβάνεται συνεχώς, έως ότου η τιμή ελέγχου γίνει μεγαλύτερη της τελικής τιμής, οπότε τερματίζεται η επανάληψη και το πρόγραμμα συνεχίζει με την εντολή που ακολουθεί το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**.

Ας σημειωθεί ότι, αν η τιμή του βήματος είναι 1, τότε μπορεί να παραλειφθεί.



Η εντολή **ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ** χρησιμοποιείται στην περίπτωση που πρέπει να επαναληφθεί η εκτέλεση κάποιων εντολών για προκαθορισμένο αριθμό επαναλήψεων.

Σύνταξη

```
ΓΙΑ μεταβλητή ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή2 ΜΕ ΒΗΜΑ τιμή3  
    εντολή-1  
    εντολή-2  
    ...  
    εντολή-ν  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Παράδειγμα

```
ΓΙΑ Αριθμό ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ ΒΗΜΑ 2  
    Άθροισμα <- Άθροισμα+Αριθμό  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Λειτουργία

Οι εντολές του βρόχου εκτελούνται για όλες τις τιμές της μεταβλητής από την αρχική τιμή μέχρι την τελική τιμή, αυξανόμενες με την τιμή του βήματος. Αν το βήμα είναι ίσο με 1, τότε παραλείπεται.

ΠΑΡΑΔΕΙΓΜΑ 4

Το παρακάτω πρόγραμμα υπολογίζει το άθροισμα των περιττών αριθμών που είναι μικρότεροι από το 100.

```
ΠΡΟΓΡΑΜΜΑ Περιττοί  
ΜΕΤΑΒΛΗΤΕΣ  
    ΑΚΕΡΑΙΕΣ: Άθροισμα, Αριθμός  
ΑΡΧΗ  
    Άθροισμα <- 0  
    ΓΙΑ Αριθμός ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ ΒΗΜΑ 2  
        Άθροισμα <- Άθροισμα + Αριθμός  
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
    ΓΡΑΨΕ 'Άθροισμα περιττών αριθμών είναι: ', Άθροισμα  
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Πολύ συχνά για την επίλυση των προβλημάτων απαιτείται η χρήση εμφωλευμένων βρόχων. Σε αυτή την περίπτωση ο ένας βρόχος βρίσκεται μέσα στον άλλο.

Στη χρήση των εμφωλευμένων βρόχων ισχύουν συγκεκριμένοι κανόνες που πρέπει να ακολουθούνται αυστηρά για τη σωστή λειτουργία των προγραμμάτων.

Συγκεκριμένα:

- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος, πρέπει να ολοκληρώνεται πρώτος.
- Η είσοδος σε κάθε βρόχο υποχρεωτικά γίνεται από την αρχή του.
- Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.



Κάθε επανάληψη που εκτελείται με μία εντολή ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ, μπορεί να υλοποιηθεί και με τη χρήση των βασικών εντολών επανάληψης ΟΣΟ..ΕΠΑΝΑΛΑΒΕ και ΜΕΧΡΙΣ..ΟΤΟΥ.

ΠΑΡΑΔΕΙΓΜΑ 5

Να γραφεί πρόγραμμα το οποίο να εκτυπώνει την προπαίδεια του πολλαπλασιασμού.

ΠΡΟΓΡΑΜΜΑ Προπαίδεια

!Πρόγραμμα εκτύπωσης της προπαίδειας των αριθμών 1 έως 10

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, Γ

!A: Πολλαπλασιαστέος

!B: Πολλαπλασιαστής

!Γ: Γινόμενο

ΑΡΧΗ

ΓΙΑ A **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΓΙΑ B **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

Γ <- A*B

ΓΡΑΨΕ A, 'X', B, '=', Γ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ **!Εισαγωγή κενής γραμμής στην εκτύπωση**

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Προγραμματιστικό περιβάλλον Basic

\ Προπαίδεια

DEFINT A-Z

FOR a = 1 TO 10

FOR b = 1 TO 10

c = a * b

PRINT a; "x"; b; "="; c

NEXT b

PRINT

NEXT a

END

Λέξεις-κλειδιά

Λογική έκφραση,
Επιλογή,
Επανάληψη,
Βρόχος

Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκαν οι εντολές που χρησιμοποιεί η ΓΛΩΣΣΑ για να υλοποιήσει τις βασικές δομές της επιλογής και της επανάληψης.

Αρχικά παρουσιάζονται οι λογικές εκφράσεις καθώς και ο τρόπος που διατυπώνονται σύνθετες λογικές εκφράσεις με τη χρήση των λογικών τελεστών ΟΧΙ, Ή, ΚΑΙ. Η εντολή ΑΝ...ΤΟΤΕ υλοποιεί τη δομή της επιλογής.

Η εντολή αυτή εμφανίζεται με πιο σύνθετες μορφές: την εντολή ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ καθώς και την εντολή ΑΝ...ΤΟΤΕ... ΑΛΛΙΩΣ_ΑΝ. Μία άλλη εντολή επιλογής που υπάρχει είναι η εντολή ΕΠΙΛΕΞΕ.

Οι εντολές επανάληψης είναι τρεις. Η εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ, η εντολή ΜΕΧΡΙΣ_ΟΤΟΥ και τέλος η εντολή ΓΙΑ. Η εντολή ΓΙΑ χρησιμοποιείται για καθορισμένο αριθμό επαναλήψεων, ενώ ο αριθμός επαναλήψεων των άλλων δύο δεν είναι γνωστός εκ των προτέρων και εξαρτάται από τις συνθήκες που τις ελέγχουν. Η εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ ελέγχει τη συνθήκη στην αρχή της επανάληψης, ενώ η εντολή ΜΕΧΡΙΣ_ΟΤΟΥ κάνει τον έλεγχο στο τέλος της επανάληψης.

Ερωτήσεις - Θέματα για συζήτηση

1. Ποιες είναι οι τιμές που μπορεί να πάρει μία λογική έκφραση;
2. Ποιοι είναι οι βασικοί λογικοί τελεστές; Αναφέρατε δύο παραδείγματα για τη χρήση του καθενός.
3. Ποια είναι η σύνταξη της εντολής ΑΝ;
4. Ποια είναι η διαφορά της εντολής ΑΝ- ΑΛΛΙΩΣ και της ΑΝ- ΑΛΛΙΩΣ_ΑΝ;
5. Τι είναι τα εμφωλευμένα ΑΝ;
6. Πότε χρησιμοποιείται η εντολή ΕΠΙΛΕΞΕ;
7. Ποιες οι εντολές επανάληψης;
8. Ποιες οι διαφορές της εντολής ΟΣΟ και της εντολής ΜΕΧΡΙΣ_ΟΤΟΥ;
9. Πώς συντάσσεται η εντολή ΓΙΑ;
10. Ποια η βασική διαφορά της εντολής ΓΙΑ από τις άλλες δύο εντολές επανάληψης;

Βιβλιογραφία

1. Ι. Κάβουρας, *Δομημένος προγραμματισμός με Pascal*, Κλειδάριθμος, Αθήνα, 1997.
2. Κ. Γιαλούρης-Κ. Σταθόπουλος, *Προγραμματισμός σε Turbo Pascal*, Αθήνα, 1996.
3. Χρ. Κοίλιας, *Η QuickBasic και οι εφαρμογές της*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
4. R. Shackelford, *Introduction to Computing and Algorithms*, Addison-Wesley, USA, 1998.
5. S. Leestma-L.Nyhoff, *Turbo Pascal, Programming and Solving*, McMillan, New York, 1990.
6. N. Wirth, *Systematic Programming: An introduction*, Prentice Hall, 1973.