

Σημειώσεις για την Γ' Γυμνασίου“Αλγόριθμοι και Δομές Δεδομένων”Τι είναι Αλγόριθμος

Ως αλγόριθμος ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος
2. Κατανόηση του προβλήματος
3. Λύση του προβλήματος
4. Διατύπωση του αλγόριθμου
5. Έλεγχος της λύσης

Μια παρουσίαση είναι αλγόριθμος όταν και μόνο όταν ικανοποιεί τα επόμενα κριτήρια:

- Έχει **Καθοριστικότητα (Definiteness)**. Κάθε κανόνας του ορίζεται επακριβώς και η αντίστοιχη διεργασία είναι συγκεκριμένη. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. π.χ. Σε μία διαίρεση να λαμβάνεται υπόψη και η περίπτωση όπου ο διαιρέτος λαμβάνει μηδενική τιμή. Τυπικές περιπτώσεις η διαίρεση με το μηδέν, υπόριξη ποσότητα αρνητική, κλπ. Προβλήματα καθοριστικότητας αντιμετωπίζονται συχνά με τη λογική της επιλογής, δηλ. Αν $a > 0$ τότε αλλιώς
- Έχει **Περατότητα (Finiteness)**. Κάθε εκτέλεση είναι πεπερασμένη, δηλαδή τελειώνει ύστερα από έναν πεπερασμένο αριθμό διεργασιών ή βημάτων. Μία διαδικασία που δεν τελειώνει μετά από συγκεκριμένο/πεπερασμένο αριθμό βημάτων λέγεται απλά υπολογιστική διαδικασία.
- Έχει **Αποτελεσματικότητα (Effectiveness)**. Είναι μηχανιστικά αποτελεσματικός, δηλαδή όλες οι διαδικασίες που περιλαμβάνει μπορούν να πραγματοποιηθούν με ακρίβεια και σε πεπερασμένο χρόνο "με μολύβι και χαρτί". Κάθε μεμονωμένη εντολή του αλγόριθμου να είναι απλή (και όχι σύνθετη). Δηλαδή μία εντολή δεν αρκεί να έχει οριστεί αλλά πρέπει να είναι και εκτελέσιμη.
- Έχει **Επεκτασιμότητα**
- Έχει **Είσοδο δεδομένων (Input)**. Κατά την εκκίνηση εκτέλεσης του αλγορίθμου καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δε δίνονται τιμές δεδομένων εμφανίζεται όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.
- Έχει **Έξοδο αποτελεσμάτων (Output)**. Δίδει τουλάχιστον ένα μέγεθος σαν αποτέλεσμα που εξαρτάται κατά κάποιο τρόπο από τις αρχικές εισόδους. Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή (δεδομένων) ως αποτέλεσμα προς το χρήστη ή προς ένα άλλο αλγόριθμο.

Ένας αλγόριθμος είναι πλήρης όταν:

1. Ικανοποιεί τα κριτήρια ορισμού του
2. Παρουσιάζεται με έναν από τους αποδεκτούς τρόπους
3. Πληρεί όλα τα κριτήρια αξιολόγησης

Σπουδαιότητα Αλγορίθμων

Η έννοια του αλγορίθμου είναι θεμελιώδης για την Επιστήμη της Πληροφορικής.

Η Πληροφορική μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές:

1. **Hardware (Υλικό)**. Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία αρχιτεκτονική τα διάφορα συστατικά του υπολογιστή.
2. **Γλωσσών προγραμματισμού (programming languages)**. Το είδος της γλώσσας προγραμματισμού που χρησιμοποιείται (χαμηλότερου ή υψηλότερου επιπέδου) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου.
3. **Θεωρητική (theoretical)**. Το ερώτημα που συχνά τίθεται είναι, αν πραγματικά υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος. Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική γιατί προσδιορίζει τα όρια της λύσης που θα βρεθεί σε σχέση με ένα συγκεκριμένο πρόβλημα.
4. **Αναλυτική (analytical)**. Μελετώνται οι υπολογιστικοί πόροι (computer resources) που απαιτούνται από ένα αλγόριθμο όπως για παράδειγμα το μέγεθος της κύριας μνήμης και της δευτερεύουσας μνήμης, ο χρόνος για λειτουργίες της CPU και για τις λειτουργίες εισόδου/εξόδου.

Σύμφωνα με όσα αναφέρθηκαν ο αλγόριθμος αποσκοπεί στην επίλυση ενός προβλήματος. Έτσι είναι απαραίτητο να γίνεται μια καλή ανάλυση του κάθε προβλήματος και να προτείνεται συγκεκριμένη μεθοδολογία και ακολουθία βημάτων. Βασικός στόχος μας είναι η πρόταση έξυπνων και αποδοτικών λύσεων.

Η ανάλυση ενός προβλήματος σε ένα σύγχρονο υπολογιστικό περιβάλλον περιλαμβάνει:

- την καταγραφή της υπάρχουσας πληροφορίας για το πρόβλημα
- την αναγνώριση των ιδιαιτεροτήτων του προβλήματος
- την αποτύπωση των συνθηκών και προϋποθέσεων υλοποίησης του
- την πρόταση επίλυσης του με χρήση κάποιας μεθόδου
- την τελική επίλυση με χρήση υπολογιστικών συστημάτων

Έτσι κατά την ανάλυση ενός προβλήματος θα πρέπει να δοθεί απάντηση σε κάθε μια από τις επόμενες ερωτήσεις:

1. ποια είναι τα δεδομένα του προβλήματος
2. ποιες είναι οι συνθήκες που πρέπει να πληρούνται για την επίλυση του προβλήματος
3. ποια είναι η πλέον αποδοτική μέθοδος επίλυσης τους (σχεδίαση αλγορίθμου)
4. πώς θα καταγραφεί η λύση σε ένα πρόβλημα (π.χ. σε ψευδογλώσσα)
5. ποιος είναι ο τρόπος υλοποίησης στο συγκεκριμένο υπολογιστικό σύστημα (π.χ. επιλογή γλώσσας προγραμματισμού)

Περιγραφή και αναπαράσταση αλγορίθμων

Τέσσερις είναι οι βασικοί τρόποι παρουσίασης (περιγραφής και αναπαράστασης) ενός αλγορίθμου:

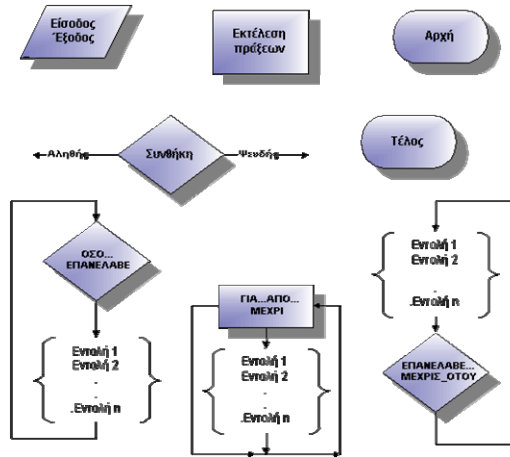
1. **Ελεύθερο κείμενο (free text)**, που αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Ελλοχεύει η δημιουργία μιας μη εκτελέσιμης κατάστασης παραβιάζοντας έτσι το κριτήριο της αποτελεσματικότητας.
2. **Διαγραμματικές τεχνικές (diagramming techniques)**, που συνιστούν ένα γραφικό τρόπο παρουσίασης αλγορίθμου. Η πιο γνωστή από αυτές είναι το **διάγραμμα ροής (flow chart)**. Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτελεί την καλύτερη λύση γι' αυτό και εμφανίζονται όλο και σπανιότερα στη βιβλιογραφία και πράξη.
3. **Φυσική γλώσσα (natural language)** που εκτελείται κατά βήματα. Σε αυτή την περίπτωση χρειάζεται προσοχή, γιατί μπορεί να παραβιαστεί το κριτήριο του καθορισμού μεταξύ των βημάτων.
4. **Κωδικοποίηση (coding)** του αλγορίθμου σε ψευδογλώσσα ή γλώσσα προγραμματισμού. Έτσι ο αλγόριθμος παρουσιάζεται πιο συνοπτικός, συμπαγής ενώ πληρεί και τις προϋποθέσεις του Δομημένου προγραμματισμού.

Σύμβολα ροής διαγράμματος

Ένα διάγραμμα ροής αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια ή λειτουργία. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών. Τα κυριότερα χρησιμοποιούμενα γεωμετρικά σχήματα είναι τα εξής:

- **έλλειψη**, που δηλώνει την αρχή και το τέλος του κάθε αλγορίθμου
- **ρόμβος**, που δηλώνει μία ερώτηση με δυο ή περισσότερες εξόδους για απάντηση
- **ορθογώνιο**, που δηλώνει την εκτέλεση μίας ή περισσότερων πράξεων
- **πλάγιο παραλληλόγραμμο**, που δηλώνει είσοδο ή έξοδο στοιχείων
- συγκεκριμένα σχέδια για τις επαναληπτικές δομές **ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ**, **ΟΣΟ...ΕΠΑΝΕΛΑΒΕ** και **ΕΠΑΝΕΛΑΒΕ...ΜΕΧΡΙΣ_ΟΤΟΥ**

Το επόμενο σχήμα αποτυπώνει όλα αυτά τα σύμβολα:



Μαζί με τα διαγράμματα ροής χρησιμοποιείται και η περιγραφή του αλγορίθμου υπό μορφή ψευδογλώσσας, που θα περιγραφεί λεπτομερειακά σε επόμενο κεφάλαιο.

Αξιολόγηση αλγορίθμων

Τα κριτήρια αξιολόγησης ενός αλγορίθμου είναι:

1. η σαφήνεια της παρουσιάσής του και ιδιαίτερα στην έκφραση των τύπων των δεδομένων
2. η ακρίβειά του
3. η εύκολη ανάγνωσή του
4. η εύκολη μετατροπή του σε γλώσσα προγραμματισμού

Οι αλγόριθμοι μπορούν να αξιολογηθούν κατά πόσο είναι καλοί ή όχι. Εάν δεν ικανοποιούν έστω ένα από τα πιο πάνω κριτήρια τότε ο αλγόριθμος κρίνεται ως «κακός αλγόριθμος».

Πιο κάτω υπάρχουν μερικά ερωτήματα για την κατανόηση της επίδοσης ενός αλγορίθμου. Αυτά είναι:

- πώς υπολογίζεται ο χρόνος εκτέλεσης ενός αλγορίθμου;
- πώς μπορούν να συγκριθούν μεταξύ τους οι διάφοροι αλγόριθμοι;
- πώς μπορεί να γνωρίζεις κανείς, αν ένας αλγόριθμος είναι «βέλτιστος»;

Οι πληροφορίες αυτές αφορούν κυρίως στην αναγνώριση της χειρότερης περίπτωσης του αλγορίθμου και στην αποτύπωση του μεγέθους του προβλήματος, με βάση το πλήθος των δεδομένων.

Η χειρότερη περίπτωση ενός αλγορίθμου αφορά το μέγιστο κόστος εκτέλεσης του αλγορίθμου, κόστος που μετράται σε υπολογιστικούς πόρους. Το κόστος αυτό πολλές φορές κρίνει την επιλογή και το σχεδιασμό ενός αλγορίθμου. Για να εκφραστεί αυτή η χειρότερη περίπτωση χρειάζεται κάποιο μέγεθος σύγκρισης και αναφοράς που να χαρακτηρίζει τον αλγόριθμο. Η πλέον συνηθισμένη πρακτική μέτρηση είναι η μέτρηση του αριθμού των βασικών πράξεων που θα πρέπει να εκτελέσει ο αλγόριθμος στη χειρότερη περίπτωση. Για παράδειγμα μία βασική πράξη μπορεί να είναι:

- ανάθεση τιμής
- σύγκριση μεταξύ δυο μεταβλητών, ή
- οποιαδήποτε αριθμητική πράξη μεταξύ δυο μεταβλητών

Η χειρότερη περίπτωση αντιπροσωπεύει τις τιμές εκείνες, που όταν δίνονται ως είσοδος στον αλγόριθμο, οδηγούν στην εκτέλεση μέγιστου αριθμού πράξεων.

Δομές δεδομένων

Δεδομένα: Είναι η αφαιρετική αναπαράσταση της πραγματικότητας και συνεπώς μιας απλοποιημένης όψης της. Είναι ακατέργαστα γεγονότα και κάθε φορά η επιλογή τους εξαρτάται από τον τύπο του προβλήματος. Η συλλογή των ακατέργαστων δεδομένων και ο συσχετισμός τους δίνει ως αποτέλεσμα την πληροφορία (**information**).

Η Πληροφορική θεωρείται η επιστήμη που μελετά τα δεδομένα από τις ακόλουθες σκοπιές :

- **Υλικό.** Το υλικό (**hardware**) δηλαδή η μηχανή, επιτρέπει στα δεδομένα ενός προγράμματος να αποθηκεύονται στη κύρια μνήμη και στις περιφερειακές συσκευές του υπολογιστή με διάφορες αναπαραστάσεις (**representations**). Τέτοιες μορφές είναι η δυαδική, με τη βοήθεια του κώδικα ASCII, το συμπλήρωμα του 1 ή του 2 κ.τ.λ.
- **Γλώσσες προγραμματισμού.** Οι γλώσσες προγραμματισμού υψηλού επιπέδου (**high level programming languages**) επιτρέπουν τη χρήση διάφορων τύπων (**types**) μεταβλητών (**variables**) για να περιγράψουν ένα δεδομένο. Ο μεταφραστής κάθε γλώσσας φροντίζει για την αποδοτικότερη μορφή αποθήκευσης, από πλευράς υλικού, κάθε μεταβλητής στον υπολογιστή.
- **Δομές Δεδομένων.** Δομή Δεδομένων (**data structure**) είναι ένα σύνολο δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών επί αυτών. Για παράδειγμα, μία τέτοια δομή είναι η εγγραφή (**record**), που μπορεί να περιγράψει ένα είδος, ένα πρόσωπο κ.τ.λ. Η εγγραφή αποτελείται από τα πεδία (**fields**) που αποθηκεύουν χαρακτηριστικά (**attributes**) διαφορετικού τύπου, όπως για παράδειγμα ο κωδικός, η περιγραφή κλπ. Άλλη μορφή δομής δεδομένων είναι το αρχείο που αποτελείται από ένα σύνολο εγγραφών. Μία επιτρεπτή λειτουργία σε ένα αρχείο είναι η σειριακή προσπέλαση όλων των εγγραφών του.
- **Ανάληψη Δεδομένων.** Τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων μελετώνται έτσι ώστε να αναπαρασταθεί η γνώση για πραγματικά γεγονότα. Οι τεχνολογίες των Βάσεων Δεδομένων (**Databases**), της Μοντελοποίησης Δεδομένων (**Data Modeling**) και της Αναπαράστασης Γνώσης (**Knowledge Representation**) ανήκουν σε αυτή την περιοχή μελέτης των δεδομένων.

Τα δεδομένα ενός προβλήματος αποθηκεύονται στον υπολογιστή, είτε στη κύρια μνήμη του είτε στη δευτερεύουσα. Η αποθήκευση αυτή δεν γίνεται κατά ένα τυχαίο τρόπο αλλά συστηματικά δηλαδή χρησιμοποιώντας μία δομή. Η έννοια της δομής δεδομένων (**data structure**) είναι σημαντική για την Πληροφορική και ορίζεται με τον ακόλουθο ορισμό:

Δομή δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

Κάθε μορφή δομής δεδομένων αποτελείται από ένα σύνολο κόμβων (**nodes**). Οι βασικές λειτουργίες (ή αλλιώς πράξεις) επί των δομών δεδομένων είναι οι ακόλουθες:

- **Προσπέλαση (access)**, πρόσβαση σε ένα κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχόμενό του.
- **Εισαγωγή (insertion)**, δηλαδή η προσθήκη νέων κόμβων σε μία υπάρχουσα δομή.
- **Διαγραφή (deletion)**, που αποτελεί το αντίστροφο της εισαγωγής, δηλαδή ένας κόμβος αφαιρείται από μία δομή.
- **Αναζήτηση (searching)**, κατά την οποία προσπελούνται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι που έχουν μια δεδομένη ιδιότητα.
- **Ταξινόμηση (sorting)**, όπου οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά.
- **Αντιγραφή (copying)**, κατά την οποία όλοι ή μερικοί από τους κόμβους μιας δομής αντιγράφονται σε άλλη δομή.
- **Συγχώνευση (merging)**, κατά την οποία δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
- **Διαχωρισμός (separation)**, που αποτελεί την αντίστροφη πράξη της συγχώνευσης.

Υπάρχει μεγάλη εξάρτηση μεταξύ της δομής δεδομένων και του αλγόριθμου και αυτό δικαιολογεί την εξίσωση που διατυπώθηκε το 1976 από τον Wirth (που σχεδίασε και υλοποίησε τη γλώσσα Pascal):

Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα

Γλώσσες Προγραμματισμού

Πρόγραμμα είναι ένα σύνολο από οδηγίες που κατευθύνουν τη λειτουργία του Η/Υ για την εκτέλεση μιας εργασίας και που είναι γραμμένο σε μια γλώσσα προγραμματισμού. Κάθε πρόγραμμα αποτελείται από 3 μέρη. Την είσοδο, την επεξεργασία και την έξοδο. Στην είσοδο δίνουμε τιμές στα δεδομένα του προβλήματος. Αυτές οι τιμές αποθηκεύονται σε μεταβλητές στη μνήμη του υπολογιστή. Στην επεξεργασία τα δεδομένα «συνδυάζονται» μεταξύ τους (γίνονται αριθμητικές και λογικές πράξεις) προκειμένου να υπολογιστούν οι τιμές των ζητούμενων του προβλήματος. Στην έξοδο παρουσιάζονται (οθόνη, εκτυπωτής κ.τ.λ.) στο χρήστη τα αποτελέσματα της επεξεργασίας (τα ζητούμενα του προβλήματος).

Ένα **πρόγραμμα** είναι η αναπαράσταση ενός αλγορίθμου γραμμένη σε γλώσσα κατανοητή για έναν υπολογιστή. Ένα πρόγραμμα, δηλαδή, αποτελείται από μία σειρά **εντολών** που δίνονται στον υπολογιστή με σκοπό να εκτελέσει κάποια συγκεκριμένη λειτουργία ή να υπολογίσει κάποιο επιθυμητό αποτέλεσμα. Η εργασία σύνταξης των προγραμμάτων ονομάζεται **προγραμματισμός**, ενώ τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται **προγραμματιστές**.

Οι γλώσσες που «καταλαβαίνουν» οι υπολογιστές είναι τεχνητές γλώσσες που ονομάζονται **γλώσσες προγραμματισμού**. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για την επικοινωνία του ανθρώπου με τη μηχανή, όπως οι φυσικές γλώσσες (ελληνική, αγγλική, γαλλική κ.λπ.) χρησιμοποιούνται για την επικοινωνία μεταξύ των ανθρώπων.

Γλώσσα Προγραμματισμού είναι ένα πρόγραμμα που μας επιτρέπει να δώσουμε εντολές στον υπολογιστή (να τον προγραμματίσουμε) χρησιμοποιώντας σύμβολα και λέξεις, που είναι πιο οικείες στον άνθρωπο από ότι μια ακολουθία από 1 και 0 (γλώσσα που καταλαβαίνει ο υπολογιστής).

Οι γλώσσες προγραμματισμού έχουν κι αυτές το δικό τους λεξιλόγιο και το δικό τους συντακτικό. Αν θέλουμε να προγραμματίσουμε τον υπολογιστή, για να εκτελεί πιστά τις λειτουργίες που του ζητάμε, πρέπει να μάθουμε κάποια γλώσσα προγραμματισμού. Δυστυχώς οι υπολογιστές δεν έχουν σχεδιαστεί, ώστε να καταλαβαίνουν τη γλώσσα που μιλάμε, δηλαδή τη φυσική γλώσσα. Η πρόοδος, όμως, στον τομέα αυτό είναι σημαντική και πιθανόν στο μέλλον να δίνουμε οδηγίες στον υπολογιστή με την ομιλία.

Όπως έχει αναφερθεί στη Β' Γυμνασίου, η λειτουργία των υπολογιστών βασίζεται στην αναπαράσταση μόνο δύο ψηφίων, των «0» και «1». Στα πρώτα βήματα της ιστορίας των υπολογιστών οι άνθρωποι, για να επικοινωνήσουν με τον υπολογιστή, έπρεπε να χρησιμοποιούν μία γλώσσα που είχε ως αλφάβητο το «0» και το «1». Αν ήθελαν λοιπόν να δώσουν μία απλή εντολή στον υπολογιστή, π.χ. να προσθέσει το 3+5 και να εμφανίσει το αποτέλεσμα, έπρεπε να μετατρέψουν όλη την εντολή σε μία γραμμή από 0 και 1. Η γλώσσα αυτή ονομάστηκε **γλώσσα μηχανής**. Η γλώσσα μηχανής είναι αρκετά δύσκολη για να την μάθει κάποιος, γιατί είναι πολύ

διαφορετική από τη φυσική μας γλώσσα. Επίσης δεν είναι ενιαία σε όλους τους υπολογιστές, μια και κάθε τύπος υπολογιστή (με διαφορετικό επεξεργαστή) έχει τη δική του γλώσσα μηχανής.

Χαρακτηριστικά Γλωσσών Προγραμματισμού

Με την πάροδο των χρόνων οι γλώσσες προγραμματισμού εξελίχθηκαν, ώστε να μοιάζουν όλο και περισσότερο με τη φυσική μας γλώσσα. Στις μέρες μας υπάρχουν διάφορες γλώσσες προγραμματισμού, που χρησιμοποιούνται για την ανάπτυξη γενικών εφαρμογών, ενώ άλλες είναι πιο εξειδικευμένες και χρησιμοποιούνται για πιο ειδικά επιστημονικά προβλήματα (ανώτερων μαθηματικών, μηχανικής, προσομοίωσης πειραμάτων κ.λπ.) και εξειδικευμένες εφαρμογές (προγραμματισμός ιστοσελίδων, διαχείριση εμπορικών δεδομένων κ.λπ.). Μερικές γνωστές γλώσσες προγραμματισμού είναι η Visual Basic, η Logo, η Pascal, η C ++, η Java.

Όπως και οι φυσικές γλώσσες, έτσι και κάθε γλώσσα προγραμματισμού έχει ως βασικά χαρακτηριστικά:

- το αλφάβητο,
- το λεξιλόγιο και
- το συντακτικό.

Το **αλφάβητο** μιας γλώσσας προγραμματισμού είναι το σύνολο των χαρακτήρων που χρησιμοποιούνται από τη γλώσσα.

Το **λεξιλόγιο** μιας γλώσσας είναι το σύνολο των λέξεων που αναγνωρίζει η γλώσσα και έχουν συγκεκριμένη και μοναδική σημασία. Στις γλώσσες προγραμματισμού το λεξιλόγιο είναι πολύ περιορισμένο (μερικές δεκάδες λέξεις), ώστε να μπορούμε να το μάθουμε εύκολα.

Το **συντακτικό** μιας γλώσσας προγραμματισμού είναι το σύνολο των κανόνων που πρέπει να ακολουθούμε, για να συνδέουμε λέξεις σε προτάσεις. Σε μια γλώσσα προγραμματισμού η σύνδεση λέξεων δημιουργεί ολοκληρωμένα; εντολές προς τον υπολογιστή. Αν δεν ακολουθήσουμε αυστηρά το συντακτικό μιας γλώσσας, είναι αδύνατο για τον υπολογιστή να καταλάβει ποια εντολή του δίνουμε.

Οι σύγχρονες γλώσσες προγραμματισμού μάς προσφέρουν ένα φιλικό περιβάλλον, έτσι ώστε γρήγορα να αναπτύσσουμε τα προγράμματά μας. Ένα **περιβάλλον προγραμματισμού** αποτελείται από διάφορα εργαλεία που βοηθάνε τον προγραμματιστή να γράψει και να διορθώσει το πρόγραμμά του.

Τα κύρια εργαλεία είναι:

- ένας εξειδικευμένος κειμενογράφος, που χρησιμεύει για τη σύνταξη και τη διόρθωση του προγράμματος και
- ένα πρόγραμμα-μεταφραστής που μετατρέπει τις οδηγίες μας στη μορφή που τις καταλαβαίνει ο επεξεργαστής δηλαδή σε μια σειρά από 0 και 1

Αυτή τη μετατροπή μπορούμε να την παρομοιάσουμε με τη διαδικασία επικοινωνίας μας με ένα κάτοικο της Κίνας. Αν δεν ξέρουμε Κινέζικα και έχουμε έναν Αγγλο μεταφραστή που μιλάει Κινέζικα, μπορούμε να του μιλήσουμε Αγγλικά και αυτός να μεταφράσει αυτό που θέλουμε στα Κινέζικα. Βέβαια μια τέτοια διαδικασία προϋποθέτει ότι ξέρουμε Αγγλικά. Παρόμοια, αν θέλουμε να επικοινωνήσουμε με τον υπολογιστή, πρέπει να μάθουμε μία γλώσσα προγραμματισμού με την οποία μπορεί να γίνει η απαραίτητη μετατροπή των οδηγιών μας σε σειρά από 0 και 1 (γλώσσα μηχανής).

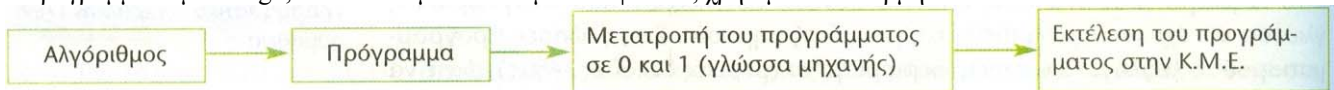
Αν σε κάποια οδηγία έχουμε κάνει λάθος στο αλφάβητο, στο λεξιλόγιο ή στο συντακτικό τότε το πρόγραμμα που μετατρέπει τις οδηγίες μας σε σειρά από 0 και 1 θα μας δώσει ένα κατάλληλο μήνυμα λάθους, ώστε να μας βοηθήσει να διορθώσουμε το λάθος μας. Τα λάθη αυτά ονομάζονται **συντακτικά λάθη**.

Τα προγράμματα που μετατρέπουν τις οδηγίες μας σε 0 και 1 μπορούν να χωριστούν σε δύο κατηγορίες:

- στους μεταγλωττιστές και
- στους διερμηνείς

Η διαφορά τους είναι ότι οι **μεταγλωττιστές (compilers)** θα ελέγξουν όλο το πρόγραμμα για συντακτικά λάθη και μετά θα το μετατρέψουν όλο σε μια κατάλληλη σειρά από 0 και 1, ώστε να μπορεί να εκτελεστεί από τον επεξεργαστή του υπολογιστή.

Αντίθετα οι **διερμηνείς (interpreters)** ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη οδηγία. Η γλώσσα προγραμματισμού Logo, που θα δούμε στο επόμενο κεφάλαιο, χρησιμοποιεί διερμηνέα.



Εικόνα 1: Στάδια για την εκτέλεση ενός αλγορίθμου από την Κ.Μ.Ε. του υπολογιστή

Δεν πρέπει να ξεχνάμε ότι ο υπολογιστής εκτελεί πιστά, όποιες συντακτικά ορθές εντολές και αν του δώσουμε. Αν το αποτέλεσμα, που τελικά προκύπτει από την εκτέλεση του προγράμματος, δεν είναι το αναμενόμενο, τότε το πρόβλημα δε βρίσκεται στον τρόπο εκτέλεσης, αλλά στον αλγόριθμο που κατασκευάσαμε για τη λύση του προβλήματός μας. Στην περίπτωση αυτή λέμε ότι έχουμε κάνει ένα λογικό λάθος και πρέπει να ελέγξουμε ένα προς ένα τα βήματα-εντολές του αλγορίθμου μας, ώστε να διαπιστώσουμε, αν δίνουμε τις κατάλληλες εντολές με τη σωστή σειρά.

Ένα δεύτερο σημείο που πρέπει να γνωρίζουμε, όταν προγραμματίζουμε, είναι ότι για τον υπολογιστή τίποτα δεν είναι αυτονόητο. Ενώ εμείς οι άνθρωποι έχουμε την ικανότητα να συμπληρώνουμε τις οδηγίες κάποιου με τη λογική και την εμπειρία μας, ο υπολογιστής χρειάζεται να περιγράψουμε με μεγάλη ακρίβεια τις εντολές μας στον υπολογιστή, για να τις εκτελέσει. Αν, για παράδειγμα, του δώσουμε μία εντολή να υπολογίσει ένα άθροισμα, δεν είναι αυτονόητο ότι θα μας εμφανίσει και το αποτέλεσμα. Αν φαίνεται ότι οι υπολογιστές επιλύουν πολύ «έξυπνα» διάφορα προβλήματα, είναι, γιατί κάποιοι άνθρωποι τους προγραμματίσαν γι' αυτό και όχι γιατί οι μηχανές είναι «έξυπνες». Για να φτιάξουμε λοιπόν ένα καλό πρόγραμμα, πρέπει πρώτα να έχουμε σχεδιάσει έναν καλό αλγόριθμο. Ο ρόλος του αλγορίθμου είναι θεμελιώδης.

Το περιβάλλον προγραμματισμού MicroWorlds Pro

Το MicroWorlds Pro (προφέρεται "μάικρογουόρνλντς προ") είναι λογισμικό το οποίο προσφέρει στο χρήστη του ένα προγραμματιστικό πολυμεσικό περιβάλλον.

Αποτελείται από τα παρακάτω βασικά μέρη:

- Γραμμή Εντολών (menu),
- Γραμμή Εργαλείων (toolbar),
- Επιφάνεια εργασίας (Σελίδα),
- Κέντρο Εντολών,
- Περιοχή Καρτελών.

Τι είναι το Μενού Επιλογών (Γραμμή Εντολών) και σε τι χρησιμεύει:

Εδώ βρίσκονται όλες **οι εντολές του προγράμματος (όχι της γλώσσας προγραμματισμού αλλά του προγράμματος)**.

Τι είναι η Γραμμή Εργαλείων και σε τι χρησιμεύει:

Μια σειρά από **κουμπιά** με τις **βασικότερες** και τις πιο **συχνές εντολές** του προγράμματος.

Τι είναι η Επιφάνεια εργασίας (Σελίδα):

Είναι αυτό που παντού στον κόσμο ονομάζεται **Σελίδα** (Page). Επιφάνεια εργασίας ονομάζουμε το σύνολο του προγράμματος που φαίνεται στην οθόνη μας και πάνω στο οποίο μπορούμε να δουλέψουμε. Πρόκειται για το μέρος όπου εμφανίζονται και κινούνται τα **αντικείμενα** του προγράμματος όπως η χελώνα, τα κουμπιά, τα πλαίσια κειμένου, τα ρυθμιστικά κ.ά. Μία εργασία (Project) μπορεί να εμπεριέχει παραπάνω από μια σελίδες. Σκεφτείτε τις σελίδες σαν τις σελίδες ενός κειμενογράφου. Αν θέλουμε να γράψουμε κάτι το γράφουμε πάνω σε μια σελίδα. Αν χρειαστεί να γράψουμε και άλλα πράγματα, τότε γράφουμε σε άλλη σελίδα, έπειτα σε άλλη κ.ο.κ. Το ίδιο ακριβώς συμβαίνει και στο MicroWorlds Pro. Γι' αυτό και ονομάζεται Σελίδα και όχι Επιφάνεια εργασίας.

Τι είναι το Κέντρο Εντολών και σε τι χρησιμεύει:

Είναι το σημείο από το οποίο μπορούμε να **στέλνουμε απευθείας εντολές** στη Logo για εκτέλεση. Αυτό συμβαίνει επειδή το MicroWorlds Pro είναι ένας διερμηνέας (interpreter). Έτσι μπορούμε να κάνουμε κλικ στο κουμπί **Δημιουργία πλαισίου κειμένου** και να κάνουμε ένα απλό κλικ στη Σελίδα. Μόλις δημιουργήσαμε ένα πλαίσιο κειμένου. Έπειτα κάνουμε κλικ στο Κέντρο Εντολών και πληκτρολογούμε την εντολή:

ΔΕΙΞΕ [14 ΓΥΜΝΑΣΙΟ ΛΑΡΙΣΑΣ]

...και τέλος πατάμε το πλήκτρο ENTER

Μέσα στο πλαίσιο κειμένου θα εμφανιστεί η φράση 14 ΓΥΜΝΑΣΙΟ ΛΑΡΙΣΑΣ

Τι είναι η Περιοχή Καρτελών και σε τι χρησιμεύει:

Δεν είναι τίποτα άλλο παρά μια επέκταση των δυνατοτήτων του προγράμματος στο οποίο εμφανίζονται κάποιες ομαδοποιημένες εντολές και οι καρτέλες που εμφανίζουν επιπλέον λειτουργίες ή ιδιότητες του προγράμματος. Στο κάτω μέρος αυτής της περιοχής υπάρχουν τέσσερις καρτέλες με τις ονομασίες: Διαδικασίες, Εργασία, Διεργασίες, Γραφικά. Το τι κάνει η καθεμιά θα το δούμε παρακάτω.

Εντολές

Τις εντολές μπορούμε να τις γράφουμε είτε με κεφαλαία είτε με μικρά γράμματα. Καλό, όμως, θα ήταν να προτιμάμε να τις γράφουμε με κεφαλαία έτσι ώστε να ξεχωρίζουν. Στη σύνταξη των εντολών δεν ξεχνάμε να βάζουμε το διάστημα (κενό ή spacebar) μετά από κάθε εντολή ή όρισμα ή αριθμό ή σύμβολο πράξης. Οι αγκύλες [] λειτουργούν ως "δεσμοφύλακες κειμένου", δηλαδή ότι περικλείεται μέσα τους είναι κείμενο.

Στους υπολογιστές ισχύει η προτεραιότητα των πράξεων που ισχύει και στα μαθηματικά.

Ποιες εντολές ονομάζονται εξόδου:

Αυτές που έχουν ως αποτέλεσμα την **εμφάνιση** ενός αριθμού ή μιας λέξης στην **οθόνη** του υπολογιστή. Η εντολή ΔΕΙΞΕ είναι εντολή εξόδου.

Πώς μπορούμε να εμφανίσουμε ένα κείμενο στο Κέντρο Εντολών:

Δίνουμε την εντολή ΔΕΙΞΕ, η οποία συντάσσεται ως εξής:

ΔΕΙΞΕ λέξη ή ΔΕΙΞΕ "λέξη	προσοχή το " ισχύει για μια και μόνο λέξη (το σύμβολο " ΔΕΝ εμφανίζεται).
ΔΕΙΞΕ [λέξη λέξη λέξη]	εμφανίζει ότι περιέχεται μέσα στις αγκύλες []
ΔΕΙΞΕ αριθμός σύμβολο αριθμός	εμφανίζει το αποτέλεσμα της πράξης

Πώς μπορούμε να εμφανίσουμε ένα κείμενο σε μια Σελίδα:

Στο MicroWorlds Pro πρέπει πρώτα να δημιουργήσουμε ένα πλαίσιο κειμένου. Αυτό γίνεται ως εξής: Κάνουμε κλικ στο εικονίδιο.

Στη συνέχεια κάνουμε κλικ με το ποντίκι κάπου στη Σελίδα. Θα εμφανιστεί ένα πλαίσιο κειμένου. Δίνουμε την εντολή:

ΔΕΙΞΕ καλημέρα

...πατάμε ENTER και το **καλημέρα** θα εμφανιστεί μέσα στο πλαίσιο κειμένου που μόλις δημιουργήσαμε.

Πώς μπορούμε να κάνουμε μαθηματικές πράξεις:

Με την εντολή ΔΕΙΞΕ και με τη βοήθεια των παρακάτω συμβόλων:

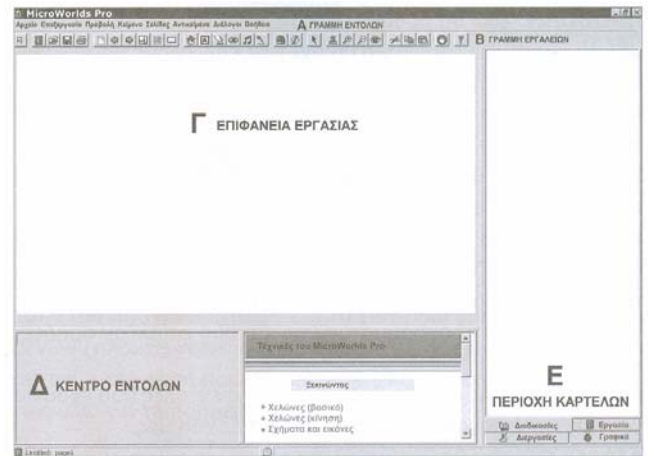
- + για πρόσθεση, - για αφαίρεση, * για πολλαπλασιασμό, / για διαίρεση

Το σύμβολο της πράξης πρέπει να ξεχωρίζει από τον προηγούμενο και από τον επόμενο αριθμό με ένα κενό (διάστημα ή space). Σε αντίθετη περίπτωση θα πάρουμε ένα μήνυμα λάθους.

Πώς μπορούμε να υψώσουμε σε δύναμη έναν αριθμό:

ΔΕΙΞΕ ΔΥΝΑΜΗ x y εμφανίζει το αποτέλεσμα της πράξης x^y , όπου y ο εκθέτης

ΔΕΙΞΕ ΔΥΝΑΜΗ 2 3



Πώς μπορούμε να εμφανίσουμε μηνύματα μαζί με τα αποτελέσματα αριθμητικών πράξεων;

Με τη χρήση της εντολής ΦΡΑΣΗ, η οποία συντάσσεται μαζί με την εντολή ΔΕΙΞΕ.

ΔΕΙΞΕ (ΦΡΑΣΗ [λέξη λέξη λέξη] πράξη)

ΔΕΙΞΕ (ΦΡΑΣΗ [Είμαι μόνο] 45 / 3 [χρονών])

θα εμφανίσει το: Είμαι μόνο 15 χρονών

Πώς μπορούμε να εμφανίσουμε ένα παράθυρο με ένα μήνυμα στην οθόνη του υπολογιστή μας μέσα από τη Logo;

Πρέπει να χρησιμοποιήσουμε την εντολή ΑΝΑΚΟΙΝΩΣΗ, η οποία συντάσσεται ως εξής:

ΑΝΑΚΟΙΝΩΣΗ [λέξη λέξη λέξη]

Δηλαδή πληκτρολογούμε: την εντολή ΑΝΑΚΟΙΝΩΣΗ, ένα διάστημα, [, το μήνυμα που θέλουμε να εμφανιστεί στο παράθυρο,] και πατάμε το πλήκτρο ENTER. Η εντολή ΑΝΑΚΟΙΝΩΣΗ είναι και αυτή εντολή εξόδου.

Πώς μπορούμε να εμφανίσουμε ένα παράθυρο που να προβάλλει ένα ερώτημα και να δέχεται την απάντηση του χρήστη;

Με την εντολή ΕΡΩΤΗΣΗ (τόρα αρχίζουμε να ασχολούμαστε με τον πραγματικό προγραμματισμό, την αλληλεπίδραση του προγράμματος με τον χρήστη).

ΕΡΩΤΗΣΗ [λέξη λέξη λέξη]

Η εντολή ΕΡΩΤΗΣΗ είναι ταυτόχρονα εντολή εξόδου και εισόδου, δηλαδή εμφανίζει κάτι στην οθόνη αλλά δέχεται και δεδομένα από το χρήστη προς το πρόγραμμα.

Πώς θα ξέρουμε ποια απάντηση έδωσε ο χρήστης στο ερώτημά μας;

Η χρήση της εντολής ΕΡΩΤΗΣΗ δεν θα μας βοηθούσε σε τίποτα αν δεν μπορούσαμε να δούμε ποια είναι η απάντηση του χρήστη στο ερώτημα που του υποβάλαμε. Η απάντηση του χρήστη αποθηκεύεται στη μνήμη του υπολογιστή, όχι, όμως, σε τυχαίο μέρος επειδή δεν θα μπορούσαμε να τη βρούμε. Το σημείο της μνήμης στο οποίο αποθηκεύεται η απάντηση του χρήστη μετά το ερώτημά μας ονομάζεται ΑΠΑΝΤΗΣΗ. Έτσι, μετά από ένα ερώτημα, μπορούμε να δώσουμε την εντολή: ΔΕΙΞΕ ΑΠΑΝΤΗΣΗ και θα εμφανιστεί η απάντηση του χρήστη στο Κέντρο Εντολών ή στο πλαίσιο κειμένου.

Πώς μπορούμε να χρησιμοποιήσουμε την ΑΠΑΝΤΗΣΗ;

Η ΑΠΑΝΤΗΣΗ είναι μια μεταβλητή, είναι ένα κομμάτι της μνήμης που μπορούμε να το χρησιμοποιήσουμε στο πρόγραμμά μας. Αν η ΑΠΑΝΤΗΣΗ είναι αριθμός, τότε μπορούμε να κάνουμε πράξεις με αυτή. Για παράδειγμα, αν φέτος είναι το έτος 2010 μπορούμε να δώσουμε την παρακάτω εντολή για να δούμε την ηλικία του χρήστη:

ΕΡΩΤΗΣΗ [Ποια χρονιά γεννηθήκατε;]

ΔΕΙΞΕ 2010 - ΑΠΑΝΤΗΣΗ

βάζουμε όλες τις εντολές σε μία γραμμή

Η Logo **μπορεί να επεξεργαστεί παραπάνω από μία εντολές σε μία γραμμή**. Εκτελώντας αυτή την εντολή θα εμφανιστεί το παράθυρο διαλόγου και αμέσως μετά την απάντηση (το κλικ στο OK) θα εμφανιστεί η ηλικία.

Ας δώσουμε την εντολή:

ΕΡΩΤΗΣΗ [Ποια χρονιά γεννηθήκατε;] ΑΝΑΚΟΙΝΩΣΗ (ΦΡΑΣΗ [Είστε] 2010 - ΑΠΑΝΤΗΣΗ [ετών...])

Μ' αυτό εμφανίζεται ένα παράθυρο διαλόγου που ζητά το έτος γέννησης του χρήστη και μόλις εκείνος απαντήσει, εμφανίζεται ένα παράθυρο που λέει: Είστε xx ετών.

ΕΡΩΤΗΣΗ [Σου αρέσει η Logo;]

αφού απαντήσουμε στο παράθυρο που θα εμφανιστεί εμφανίζεται παράθυρο με μήνυμα

ΑΝΑΚΟΙΝΩΣΗ (ΦΡΑΣΗ [Απάντησες:] ΑΠΑΝΤΗΣΗ [Το εννοείς;])

Θα μπορούσαμε αντί της εντολής ΑΝΑΚΟΙΝΩΣΗ να χρησιμοποιήσουμε την ξανά την εντολή ΕΡΩΤΗΣΗ και έτσι ο «διάλογος» θα συνεχιζόταν.

ΕΡΩΤΗΣΗ (ΦΡΑΣΗ [Απάντησες:] ΑΠΑΝΤΗΣΗ [Το εννοείς αληθινά;])

Τι είναι η χελώνα στη γλώσσα Logo;

Είναι, ίσως, το πιο χαρακτηριστικό της στοιχείο. Είναι ένας δείκτης πάνω στη Σελίδα (Επιφάνεια εργασίας) τον οποίο μπορούμε να μετακινήσουμε με τις κατάλληλες εντολές. Έτσι, η γλώσσα Logo, εκτός από μαθηματικές πράξεις και εμφάνιση κειμένων μπορεί να παρουσιάζει και να μετακινεί **γραφικά** στην οθόνη. Η παρουσίαση των γραφικών γίνεται με το *ίχνος* (μια γραμμή) που αφήνει πίσω της η χελώνα, όπως το σαλιγκάρι.

Πώς εμφανίζεται μια χελώνα στη Σελίδα μας;

Πρέπει να κάνουμε κλικ στο εικονίδιο **Δημιουργία χελώνας**.

Πώς μπορούμε να αλλάξουμε το όνομα μιας χελώνας;

Κάνουμε δεξί κλικ πάνω στη χελώνα και επιλέγουμε *Επεξεργασία*. Στο παράθυρο που εμφανίζεται και στο πλαίσιο *Όνομα* πληκτρολογούμε το νέο όνομα που θ'έλουμε να έχει η χελώνα.

Πώς μπορούμε να βάλουμε μια άλλη εικόνα στη θέση της χελώνας;

Το MicroWorldsPro επιτρέπει την αλλαγή της εικόνας της χελώνας μέσα από την καρτέλα *Γραφικά*. Βρίσκουμε το σχήμα που μας αρέσει, κάνουμε κλικ επάνω του και μετά κάνουμε κλικ πάνω στη χελώνα.

Πώς μπορούμε να έχουμε δύο ή περισσότερες χελώνες στη Σελίδα μας;

Απλώς προσθέτουμε και δεύτερη και τρίτη και όσες χελώνες θέλουμε. Προσέχουμε μόνο να τους αλλάζουμε την εμφάνιση έτσι ώστε να ξεχωρίζουν, όχι μόνο από το όνομα στον προγραμματισμό αλλά και οπτικά στην Επιφάνεια εργασίας.

Με ποιες εντολές μπορούμε να μετακινήσουμε τη χελώνα;

Οι εντολές που μετακινούν τη χελώνα χωρίζονται σε δυο κατηγορίες:

- στον προσανατολισμό και
- στην κίνηση της χελώνας

Ο προσανατολισμός αφορά την επί τόπου στροφή της χελώνας σε μοίρες. Η κίνηση αφορά τη μετακίνηση της χελώνας σε pixels (εικονοστοιχεία).

Προσανατολισμός. Η χελώνα ξεκινάει με προσανατολισμό πάντοτε προς τα επάνω.

ΔΕΞΙΑ ή ΔΕ x**ΑΡΙΣΤΕΡΑ ή ΑΡ x**

(όπου x, ο αριθμός των μοιρών)

Μετακίνηση**ΜΠΡΟΣΤΑ ή ΜΠ x****ΠΙΣΩ ή ΠΙ x**


(όπου x, ο αριθμός των pixels)



Αν κάνουμε δεξί κλικ πάνω στη χελώνα και στη συνέχεια επιλέξουμε *Επεξεργασία*, μας εμφανίζεται το παρακάτω *Παράθυρο εντολών*:

Όπως βλέπουμε, κάθε χελώνα έχει ένα όνομα (π.χ. x1) και εκτελεί τις οδηγίες που θα πληκτρολογήσουμε (π.χ. μπ 100). Αν επιλέξουμε *Εκτέλεση Μία φορά* και πατήσουμε OK, η χελώνα θα ξεκινήσει.

Πώς σταματάμε τη χελώνα:

Με το κουμπί *Διακοπή Όλων*  η εκτέλεση του προγράμματος μας διακόπτεται.

Η χελώνα προχωράει αλλά δεν βλέπω κάποιο ίχνος. Γιατί γίνεται αυτό:

Εξ ορισμού η χελώνα είναι ρυθμισμένη να μην αφήνει κάποιο ίχνος πίσω της. Για να τη ρυθμίσουμε έτσι ώστε να αφήνει ίχνος, πρέπει να δώσουμε την εντολή: **ΣΤΚ**. Αν θέλουμε να σταματήσει να αφήνει ίχνη, τότε δίνουμε την εντολή **ΣΤΑ**.

Πώς μπορώ να επαναφέρω τη Σελίδα στην πρώτη της μορφή με τη χελώνα στο κέντρο της:

Δίνοντας την εντολή **ΣΒΗΣΕΓΡΑΦΙΚΑ** ή **ΣΒΓ**, το MicroWorlds Pro σβήνει τα σχέδια που έχουμε δημιουργήσει στην Επιφάνεια εργασίας και μεταφέρει τη χελώνα στο κέντρο με κατεύθυνση προς τα πάνω.

Πώς αλλάζει το χρώμα του στυλό της χελώνας:

Δίνοντας την εντολή **ΘΕΣΕΧΡΩΜΑ** ή **ΘΕΣΕΧΡ** η οποία συντάσσεται ως εξής: **ΘΕΣΕΧΡ αριθμός** ή **ΘΕΣΕΧΡ όνομαχρώματος**

Ποιοι είναι οι κωδικικοί αριθμοί των χρωμάτων:

Η αντιστοιχία των χρωμάτων στη Logo είναι η εξής: 0 = μαύρο, 1 = άσπρο, 2 = μπλέ, 3 = κόκκινο

Πώς αλλάζει το πάχος της γραμμής που αφήνει η χελώνα:

Δίνοντας την εντολή **ΘΕΣΕΠΑΧΟΣΣΤΥΛΟ** η οποία συντάσσεται ως εξής:

ΘΕΣΕΠΑΧΟΣΣΤΥΛΟ αριθμός

όπου αριθμός, είναι το πάχος της γραμμής σε pixels

Πώς μπορώ να «πα» στη Logo να επαναλάβει μια εντολή πολλές φορές χωρίς να χρειαστεί να την πληκτρολογήσουμε ξανά:

Η Logo έχει την εντολή **ΕΠΑΝΑΛΑΒΕ** μέσω της οποίας μπορούμε να ορίσουμε την επανάληψη μιας άλλης εντολής ή ομάδας εντολών, όσες φορές θέλουμε. Η εντολή αυτή συντάσσεται ως εξής:

ΕΠΑΝΑΛΑΒΕ x [εντολή]

όπου x, είναι ο αριθμός των επαναλήψεων και εντολή, η εντολή ή οι εντολές που θέλουμε να επαναληφθούν x φορές

Η εντολή **ΕΠΑΝΑΛΑΒΕ 3 [ΔΕΙΞΕ "Γιώργος]** θα έχει σαν αποτέλεσμα την εμφάνιση του ονόματος Γιώργος, τρεις φορές.

Βασικές Εντολές Logo

Εντολή	Περιγραφή	Παράδειγμα
Απόκρυψη Χελώνας (απχ)	Κάνει τη χελώνα αόρατη (η χελώνα εξακολουθεί να υπάρχει πάνω στην επιφάνεια εργασίας)	απχ
Εμφάνιση Χελώνας (εμχ)	Επανεμφανίζει	εμχ
Στυλό Κάτω (στκ)	Κατεβάζει το στυλό κάτω, πάνω στην επιφάνεια εργασίας (ώστε να γράφει καθώς η χελώνα μετακινείται)	στκ
Στυλό Άνω (στα)	Ανεβάζει το στυλό πάνω, (ώστε να μη γράφει καθώς η χελώνα μετακινείται)	στα
Μπροστά αριθμός (μπ)	Μετακινεί τη χελώνα μπροστά (προς την κατεύθυνση στην οποία είναι στραμμένη) κατά τόσα εικονοστοιχεία –pixels– όσα δείχνει ο αριθμός	μπροστά 100 ή μπ 100
Πίσω αριθμός (πι)	Μετακινεί τη χελώνα πίσω (προς την αντίθετη κατεύθυνση απ' την οποία είναι στραμμένη) κατά τόσα εικονοστοιχεία –pixels– όσα δείχνει ο αριθμός	πίσω 100 ή πι 100
Δεξιά αριθμός (δε)	Στρίβει τη χελώνα προς τα δεξιά κατά τόσες μοίρες όσες δείχνει ο αριθμός	δεξιά 90 ή δε 90
Αριστερά αριθμός (αρ)	Στρίβει τη χελώνα προς τα αριστερά κατά τόσες μοίρες όσες δείχνει ο αριθμός	αριστερά 90 ή αρ 90
Επανάλαβε αριθμός [Εντολές]	Επαναλαμβάνει τις εντολές που υπάρχουν μέσα στις αγκύλες τόσες φορές όσες δείχνει ο αριθμός	Επανάλαβε 4 [μπ 50 δε 90]
Κέντρο	Τοποθετεί τη χελώνα στο κέντρο της επιφάνειας εργασίας (στη θέση 0,0) και με κατεύθυνση προς τα πάνω.	κέντρο
Θέσε Κατεύθυνση αριθμός (ΘέσεΚτθ)	Στρέφει (προς τα δεξιά) τη χελώνα προς μια συγκεκριμένη κατεύθυνση κατά τόσες μοίρες όσες δείχνει ο αριθμός	ΘέσεΚατεύθυνση 60 ή ΘέσεΚτθ 60
Θέσε Θέση [x y]	Τοποθετεί τη χελώνα σε μια συγκεκριμένη θέση πάνω στην επιφάνεια εργασίας που υποδεικνύεται από τα x και y	ΘέσεΘέση [100 20]
Θέσε X αριθμός	Τοποθετεί τη χελώνα σε μια συγκεκριμένη θέση ως προς τον άξονα των X	ΘέσεX 100
Θέσε Y αριθμός	Τοποθετεί τη χελώνα σε μια συγκεκριμένη θέση ως προς τον άξονα των Y	ΘέσεY 100
Δείξε Κατεύθυνση	Στο κέντρο εντολών επιστρέφει ένας αριθμός σε μοίρες ° που δείχνει ποια κατεύθυνση έχει η χελώνα σε σχέση με το κέντρο	Δείξε Κατεύθυνση

	της σελίδας.	
(όνομα χελώνας), Δείξε Κατεύθυνση	Στο κέντρο εντολών επιστρέφει ένας αριθμός σε μοίρες που δείχνει πια κατεύθυνση έχει η χελώνα1 σε σχέση με το κέντρο της σελίδας	x1, Δείξε Κατεύθυνση
Πάγωσε	Η χελώνα μένει «παγωμένη»	Πάγωσε
Ξεπάγωσε	Η χελώνα «ξεπαγώνει»	Ξεπάγωσε
ΘέσεΜέγεθος (αριθμός)	Το μέγεθος της χελώνας γίνεται τόσα pixels όσα ο αριθμός (εξ ορισμού είναι 40 pixels)	ΘέσεΜέγεθος 15
ΘέσεΠάχοςΣτυλό αριθμός	Ρυθμίζει το πάχος του στυλό της χελώνας (όπου αριθμός, είναι το πάχος της γραμμής σε pixels)	ΘέσεΠάχοςΣτυλό 20
ΘέσεΧρώμα όνομα ή αριθμός (ΘέσεΧρ)	Ρυθμίζει το χρώμα του στυλό της χελώνας	ΘέσεΧρώμα "Ροζ ή ΘέσεΧρ 125
ΘέσεΦόντο όνομα ή αριθμός (ΘέσεΦντ)	Καθορίζει το χρώμα του φόντου της επιφάνειας εργασίας.	ΘέσεΦόντο "Μπλε ή ΘέσεΦντ 117
Πάγωσεφόντο ή πάγωσεφντ	Προστατεύεται το φόντο της εργασίας από διαγραφή ή σβήσιμο	Πάγωσεφόντο
Ξεπάγωσεφόντο ή ξεπάγωσεφντ	Το φόντο γίνεται διαθέσιμο για επεξεργασία και μεταβολές	Ξεπάγωσεφόντο
Σφρ ή Σφραγίδα	Η χελώνα αφήνει το αποτύπωμά της	Σφρ
(όνομα χελώνας), γέμισε	Η χελώνα γεμίζει μια περιοχή με το χρώμα της (εξ ορισμού μαύρο)	x1, γέμισε
Περίμενε αριθμός	Η χελώνα σταματάει για κάποιο χρονικό διάστημα την κίνησή της	Περίμενε 10
(όνομα χελώνας), Σφρ (λίστα οδηγιών) ή (όνομα χελώνας) Σφραγίδα (λίστα οδηγιών)	Η χελώνα αφήνει το αποτύπωμά της στην αρχική θέση, προχωρά 100 pixels μπροστά και στρέφεται 90ο δεξιά	x1, σφρ μπ 100 δε 90
"(όνομα χελώνας1), κοίταπρος "(όνομα χελώνας2)	Η χελώνα1 στρέφεται προς τη χελώνα2 (προσοχή αγγλικά εισαγωγικά)	"x1, κοίταπρος "x2
Σβήσε (σβ ή σβήσε)	Διαγράφει τα γραφικά χωρίς να αλλάζει τη θέση των χελωνών	σβ ή σβήσε
ΣβήσεΓραφικά (σβγ)	Καθαρίζει τα γραφικά από την επιφάνεια εργασίας και τοποθετεί τη χελώνα στο κέντρο με το κεφάλι προς τα πάνω	σβγ
ΣβήσεΕντολές (σβε)	Διαγράφει τα περιεχόμενα του Κέντρου εντολών	σβε

Διαδικασίες

Για να μην πληκτρολογούμε ξανά και ξανά τις ίδιες εντολές δεν αρκεί η χρήση της εντολής ΕΠΑΝΑΛΑΒΕ. Η Logo μας δίνει τη δυνατότητα να ομαδοποιήσουμε τις εντολές μας, να δώσουμε ένα όνομα σε αυτή την ομάδα και όποτε καλούμε αυτό το όνομα να εκτελείται όλη η σειρά των εντολών αυτών. Η ομαδοποίηση αυτή των εντολών ονομάζεται **Διαδικασία**. Αφού δημιουργήσουμε μια Διαδικασία, αρκεί να δώσουμε το όνομά της στο Κέντρο Εντολών και αυτή θα εκτελεστεί.

Πώς δημιουργούμε μια διαδικασία στο MicroWorlds Pro:

Πηγαίνουμε στην Περιοχή Καρτελών και κάνουμε **κλικ** στην καρτέλα **Διαδικασίες**. Πάνω από τις καρτέλες παρατηρούμε ότι υπάρχει ένας μεγάλος, στενόμακρος, κενός χώρος, η *Περιοχή Καρτελών*. Εκεί μέσα (και όχι στο *Κέντρο Εντολών*) πληκτρολογούμε τον κώδικα από τον οποίο θα αποτελείται η διαδικασία ή οι διαδικασίες, αν γράψουμε παραπάνω από μία. Προσέξτε πως οι εντολές που πληκτρολογούμε δεν εκτελούνται κάθε φορά που πατάμε Enter – αυτό συμβαίνει μόνο στο Κέντρο Εντολών. Ας δούμε πώς γράφεται μία διαδικασία:

1. Πληκτρολογούμε την εντολή **ΓΙΑ** ακολουθούμενη από το **όνομα** της διαδικασίας. Για παράδειγμα, έστω ότι θέλουμε να γράψουμε μια διαδικασία που να ονομάζεται *τετράγωνο*. Η διαδικασία μας θα ξεκινά με το:

ΓΙΑ τετράγωνο

2. Πληκτρολογούμε τις εντολές.

3. Τελειώνουμε με την εντολή **Τέλος**. Η εντολή αυτή είναι απαραίτητη, έτσι ώστε να «γνωρίζει» η Logo σε ποιο σημείο τελειώνουν οι εντολές που ανήκουν στη διαδικασία αυτή. Στην ουσία, μια διαδικασία είναι ό,τι περιλαμβάνεται ανάμεσα στο ΓΙΑ όνομα και στο Τέλος. Δηλαδή:

ΓΙΑ τετράγωνο

ΣΤΚ

ΕΠΑΝΑΛΑΒΕ 4 [ΜΠ 60 ΔΕ 90]

Τέλος

Ας δούμε ένα παράδειγμα διαδικασίας που να φτιάχνει ένα εξάγωνο στα δεξιά της χελώνας και ένα οκτάγωνο στα αριστερά της χελώνας, όπως δείχνει η παρακάτω εικόνα.

ΓΙΑ διπλοπολύγωνο

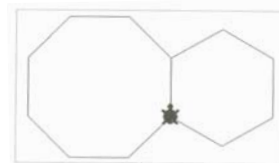
ΣΤΚ

ΕΠΑΝΑΛΑΒΕ 6 [ΜΠ 100 ΔΕ 60]

ΕΠΑΝΑΛΑΒΕ 8 [ΜΠ 100 ΑΡ 45]

Τέλος

Αν, τώρα πάμε στο Κέντρο Εντολών, πληκτρολογήσουμε *διπλοπολύγωνο* και πατήσουμε το πλήκτρο ENTER, η χελώνα μας θα εκτελέσει όλες τις παραπάνω εργασίες.



Πώς μπορούμε να εκτελέσουμε τη διαδικασία για «τετράγωνο» ώστε να ορίζουμε εμείς το μέγεθος της πλευράς του τετραγώνου;

Η Logo μας δίνει τη δυνατότητα να «εισάγουμε» δεδομένα μέσα στις διαδικασίες. Έτσι μπορούμε να εκτελέσουμε την ίδια διαδικασία με διαφορετικά μεγέθη κάθε φορά απλώς χρησιμοποιώντας μεταβλητές. Μεταβλητή ονομάζουμε ένα κομμάτι της μνήμης RAM στο οποίο αποθηκεύονται δεδομένα και μπορούμε να το διαβάσουμε ανά πάσα στιγμή χρησιμοποιώντας ένα όνομα που του έχουμε δώσει. Μπορούμε να δώσουμε όποιο όνομα θέλουμε σε μια μεταβλητή, εκτός από τα ονόματα των εντολών της Logo. Για να περάσουμε μια μεταβλητή σε μια συνάρτηση, πρέπει να ορίσουμε τη μεταβλητή και μετά το όνομα της διαδικασίας ως εξής:

ΓΙΑ όνομα_διαδικασίας :όνομα_μεταβλητής

Παρατηρήστε ότι η άνω και κάτω τελεία είναι κολλημένη στο όνομα της μεταβλητής. Στο παραπάνω παράδειγμα του τετραγώνου κάναμε τετράγωνα με πλευρές των 100 εικονοστοιχείων. Ας κάνουμε τη διαδικασία ως εξής:

ΓΙΑ τετράγωνο :πλευρά

ΣΤΚ

ΕΠΑΝΑΛΑΒΕ 4 [ΔΕ 90 ΜΠ :πλευρά]

Τέλος

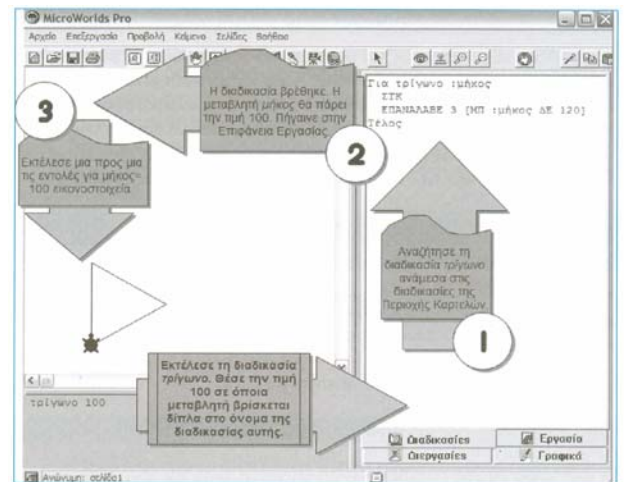
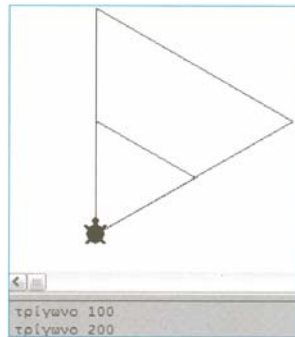
οπότε αρκεί να δώσουμε την εντολή: *τετράγωνο 150*

για να σχηματιστεί ένα τετράγωνο με πλευρά 150 εικονοστοιχεία μήκος.

Πώς μπορούμε να εκτελέσουμε τη διαδικασία «τρίγωνο» ώστε να ορίζουμε εμείς το μέγεθος της πλευράς του τριγώνου;

Στη διπλανή εικόνα φαίνονται τα βήματα που ακολουθούνται για να δημιουργηθεί (με τη χρήση της διαδικασίας *τρίγωνο*) ένα τρίγωνο στα δεξιά της χελώνας με πλευρά 100 εικονοστοιχεία.

Αν ο χρήστης επιθυμεί, μπορεί να δημιουργήσει ένα νέο *τρίγωνο* καλώντας τη διαδικασία *τρίγωνο*, δίνοντας αυτή τη φορά νέο μήκος. Π.χ. *τρίγωνο 200*:

**Μεταβλητές**

Μεταβλητή είναι ένα συμβολικό όνομα που αντιστοιχίζεται με ένα στοιχείο που εισάγεται ή ανασύρεται από μια μονάδα εισαγωγής ή αποθήκευσης ή αντιστοιχώς εξάγεται ή αποθηκεύεται σε μια μονάδα εξαγωγής ή αποθήκευσης και μπορεί να μεταβληθεί. Είναι δηλαδή ένα «κουτάκι», μια θέση στη μνήμη του υπολογιστή, η οποία έχει το δικό της όνομα και μέσα στη θέση αυτή μπορούν να αποθηκευτούν προσωρινά αριθμοί ή λέξεις. Η μεταβλητή στον προγραμματισμό δεν έχει την ίδια έννοια που έχει στα Μαθηματικά. Στον προγραμματισμό σε μία μεταβλητή X τοποθετούμε (εκχωρούμε) μία τιμή, δηλαδή, στη θέση μνήμης που αντιστοιχεί στη μεταβλητή X αποθηκεύουμε μία τιμή.

Αλφαριθμητικές Πληροφορίες: Οποιαδήποτε πληροφορία δεν αποτελεί ποσότητα με την οποία μπορεί να γίνουν πράξεις (αριθμητικές ή λογικές) χαρακτηρίζεται αλφαριθμητική πληροφορία και αποθηκεύεται σε αλφαριθμητικές μεταβλητές (μεταβλητή τυπου «χαρακτήρες»).

Μπορούμε να αλλάζουμε τα δεδομένα μιας μεταβλητής:

Τα δεδομένα (περιεχόμενο) μιας μεταβλητής είναι δυνατόν να αλλάζουν (μεταβάλλονται) κατά τη διάρκεια εκτέλεσης ενός προγράμματος. Γι' αυτό, άλλωστε, ονομάζεται και μεταβλητή (αλλιώς θα ονομαζόταν σταθερά). Φυσικά, όταν αλλάξουν τα δεδομένα, τα προηγούμενα χάνονται. Επίσης, η μεταβλητή μπορεί να έχει μία τιμή κάθε χρονική στιγμή, δηλαδή δεν μπορεί να έχει (ή παραπάνω) τιμές ταυτόχρονα.

Πώς ξεχωρίζει μια μεταβλητή από το όνομα μιας διαδικασίας ή από μια εντολή:

Κοινό χαρακτηριστικό όλων των μεταβλητών είναι ότι αρχίζουν με την άνω και κάτω τελεία κολλημένη στο όνομά τους. Προσοχή όμως! Όταν ορίζουμε μια μεταβλητή, και μόνο τότε, δεν συνοδεύεται από την άνω και κάτω τελεία.

Τι πρέπει να προσέχουμε όταν δίνουμε όνομα σε μια μεταβλητή:

Καλό είναι να δίνουμε σαφή ονόματα στις μεταβλητές μας, έτσι ώστε το πρόγραμμά μας να είναι και αυτό σαφές και κατανοητό. Επιπλέον, πρέπει να προσέχουμε να μη δίνουμε ονόματα μεταβλητών που να μπερδεύουν το πρόγραμμά μας, όπως *Δείξε*, *Για*, *Απάντηση* κ.ο.κ. Οι λέξεις αυτές είναι εντολές της Logo και η χρήσης τους ως ονόματα μεταβλητών θα οδηγήσει σε προβλήματα.

Ποιες είναι οι βασικές ιδιότητες μιας μεταβλητής:

Μια μεταβλητή:

- έχει συγκεκριμένο όνομα,
- έχει συγκεκριμένο περιεχόμενο που είναι δυνατόν να μεταβάλλεται κατά τη διάρκεια του προγράμματος και
- περιέχει μία και μόνο τιμή. Κάθε φορά που αλλάζει, οι προηγούμενες τιμές που περιείχε χάνονται για πάντα.

Πώς ορίζουμε την τιμή μιας μεταβλητής (ή βάζουμε ή εκχωρούμε δεδομένα σε μια μεταβλητή):

Για να ορίσουμε την τιμή μιας μεταβλητής πρέπει να χρησιμοποιήσουμε την εντολή KANE (είτε στο Κεντρο εντολών, είτε μέσα σε διαδικασία στην Περιοχή καρτελών). Η σύνταξη της εντολής KANE για τον ορισμό μεταβλητής είναι η εξής:

KANE "όνομα_μεταβλητής τιμή_μεταβλητής για αριθμό
 KANE "όνομα_μεταβλητής "τιμή_μεταβλητής για κείμενο
 KANE "όνομα_μεταβλητής [τιμή_μεταβλητής] για μήνυμα

Αν, δηλαδή, θέλουμε να έχουμε μια μεταβλητή με το όνομα *ύψοςY* και θέλουμε να δώσουμε την τιμή 12, τότε θα δώσουμε την εντολή:
 KANE "ύψοςY 12

Αν έχουμε μια μεταβλητή με το όνομα *σχολείο* και θέλουμε να της δώσουμε την τιμή 14 *Γυμνάσιο*, τότε θα δώσουμε την εντολή:
 KANE "σχολείο "14 Γυμνάσιο

Αν έχουμε μια μεταβλητή με το όνομα *μήκος* και θέλουμε να της δώσουμε την τιμή *πάρα πολύ μεγάλο*, τότε θα δώσουμε την εντολή:
 KANE "μήκος [πάρα πολύ μεγάλο]

Η εκχώρηση μιας τιμής σε μια μεταβλητή αποτελεί εσωτερική διαδικασία στον υπολογιστή. Δε θα εμφανιστεί κάτι στην οθόνη κατά την εκτέλεσή της. Το αποτέλεσμα θα μπορέσουμε να το ελέγξουμε με εντολή εξόδου (π.χ. την ΔΕΙΞΕ).

Πώς προβάλλουμε (εμφανίζουμε) την τιμή που είναι αποθηκευμένη μέσα σε μια μεταβλητή:

Πρέπει να χρησιμοποιήσουμε την εντολή ΔΕΙΞΕ. Θυμηθείτε ότι τώρα πρέπει να χρησιμοποιήσουμε την άνω και κάτω τελεία:
 ΔΕΙΞΕ :ύψοςY

Ποιά είναι η λειτουργία της εντολής KANE "α :α + 1:

Λέει στον υπολογιστή να εκχωρήσει στη μεταβλητή *α* μια νέα τιμή. Αυτή η τιμή θα ισούται με $:α + 1$, δηλαδή με «ό,τι περιέχεται στην *α*» + 1. Στην ουσία, η εντολή αυτή **αυξάνει την τιμή της μεταβλητής *α* κατά 1**. Στη συνέχεια με την εντολή Δείξε :α η νέα αυτή τιμή θα εμφανίζεται στην οθόνη.

Εντολές του τύπου KANE "α :α + 1 χρησιμοποιούνται σε δομές επανάληψης για να γνωρίζει ο υπολογιστής σε ποια επανάληψη βρίσκεται κάθε στιγμή. Για το λόγο αυτό, βοηθητικές μεταβλητές όπως η *α* ονομάζονται μετρητές. Αξίζει να παρατηρήσει κανείς πως η εντολή KANE "α :α + 1 δεν ισοδυναμεί με τη μαθηματική έκφραση $α = α + 1$ (αδύνατο). Απλώς, με την εντολή αυτή αυξάνουμε τη μεταβλητή *α* κατά 1. Αυτό θα γίνεται σε κάθε επανάληψη.

Οι παρακάτω εντολές εμφανίζουν διαδοχικά τους ακέραιους αριθμούς από το 1 μέχρι και το 10.

KANE "α 1

ΔΕΙΞΕ :α

ΕΠΑΝΑΛΑΒΕ 9 [KANE "α :α + 1 ΔΕΙΞΕ :α]

Σχηματικά η εκτέλεση των εντολών από τον υπολογιστή απεικονίζεται ως εξής:

Τι είναι ο μεταβολέας:

Ο **μεταβολέας** είναι ένα ηλεκτρονικό ρυθμιστικό μέσα από το οποίο μπορούμε να δίνουμε τιμές σε μια μεταβλητή από την Επιφάνεια εργασίας (Σελίδα) πολύ απλά, κουνώντας το ποντίκι. Ο μεταβολέας, οπτικά, μοιάζει με τα ρυθμιστικά που υπάρχουν στις κονσόλες ήχου (μείκτες). Έχουν **αρχική** και **τελική τιμή**. Μετακινώντας το ειδικό ρυθμιστικό από τα αριστερά προς τα δεξιά (για οριζόντιο μεταβολέα) ή από επάνω προς τα κάτω (για κάθετο μεταβολέα) αλλάζουμε τιμές από την αρχική προς την τελική. Έτσι, αν η αρχική τιμή είναι το 100 και η τελική το 200 τότε, μετακινώντας το ρυθμιστικό του μεταβολέα, θα μπορέσουμε να έχουμε όλες τις ενδιάμεσες ακέραιες τιμές.

Πώς δημιουργούμε ένα μεταβολέα:

Κάνοντας κλικ στο κουμπί Δημιουργία μεταβολέα.



Θα παρατηρήσουμε ότι το ίχνος του ποντικιού μας αλλάζει. Εμείς δεν έχουμε παρά να κάνουμε κλικ στο σημείο της σελίδας που θέλουμε να εμφανιστεί ο μεταβολέας. Κάνοντας κλικ στο σημείο που θέλουμε, θα εμφανιστεί ένα παράθυρο με τις ιδιότητες του μεταβολέα.

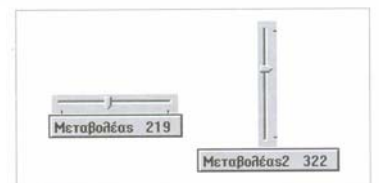
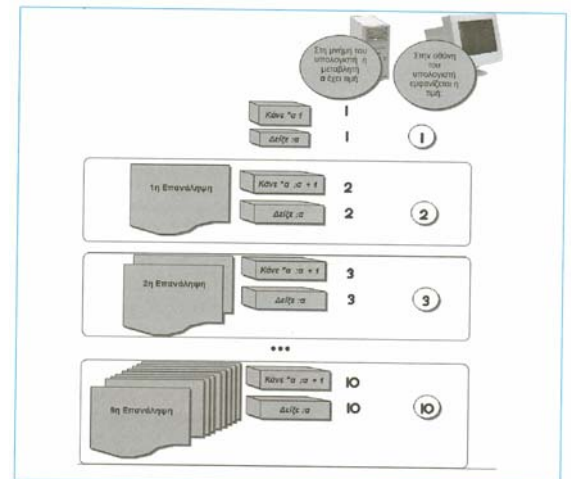
Στο σημείο αυτό μπορούμε να ορίσουμε τις παρακάτω ιδιότητες του μεταβολέα:

- το **όνομα** του μεταβολέα το οποίο μπορούμε να το χειριστούμε ως μεταβλητή
- την **ελάχιστη τιμή** που μπορεί να πάρει ο μεταβολέας
- τη **μέγιστη τιμή** που μπορεί να πάρει ο μεταβολέας
- την **τρέχουσα τιμή** που έχει ο μεταβολέας
- αν θα **εμφανίζεται** το όνομα του μεταβολέα στην οθόνη αλλά και η τιμή του
- αν ο μεταβολέας θα είναι **ορατός** ή **αόρατος**
- αν ο μεταβολέας θα είναι **οριζόντιος** ή **κάθετος**

Πώς μπορούμε να χρησιμοποιήσουμε ένα μεταβολέα:

Ο μεταβολέας χρησιμοποιείται ως **μεταβλητή**, αλλά δεν συντάσσεται όπως η μεταβλητή. Για να τυπώσουμε την τιμή μιας μεταβλητής στην οθόνη ή στο Κέντρο Εντολών πρέπει να βάλουμε την άνω και κάτω τελεία : μπροστά και κολλητά στο όνομα της μεταβλητής. Στο μεταβολέα κάτι τέτοιο δεν χρειάζεται. Απλώς χρησιμοποιούμε το όνομα του χωρίς την άνω και κάτω τελεία.

Αν έχουμε ένα μεταβολέα με το όνομα *Δοκιμαστικό* και δώσουμε την εντολή:



Όνομα:	Μεταβολέας1	Προβολή ονόματος	<input checked="" type="checkbox"/>
Ελάχιστη:	0	Ορατός	<input checked="" type="checkbox"/>
Μέγιστη:	99	Κάθετος	<input type="checkbox"/>
Τιμή:	50	OK	Άκυρο

ΔΕΙΞΕ Δοκιμαστικό

Τότε θα εμφανιστεί η τιμή που έχει το ρυθμιστικό του μεταβολέα. Αν μετακινήσουμε το ρυθμιστικό του μεταβολέα και δώσουμε πάλι την εντολή:

ΔΕΙΞΕ Δοκιμαστικό

Τότε θα εμφανιστεί η νέα τιμή του μεταβολέα.

Πώς μπορούμε να χρησιμοποιήσουμε ένα μεταβολέα σε μια διαδικασία:

Ο μεταβολέας εκτός από ρυθμιστικό, είναι και μια μεταβλητή που «μένει» στη μνήμη (οι επαγγελματίες προγραμματιστές τις ονομάζουν global ή public μεταβλητές). Αυτό σημαίνει ότι η μεταβλητή αυτή μπορεί να χρησιμοποιηθεί και μέσα σε διαδικασίες χωρίς να χρειαστεί να τις ορίσουμε ξανά. Για παράδειγμα, έστω ότι έχουμε μία διαδικασία που φτιάχνει ένα τετράγωνο και δέχεται μια μεταβλητή με το μήκος της πλευράς του τετραγώνου. Θα ήταν κάπως έτσι:

Για τετράγωνο :πλευρά

Μπορούμε να το φτιάξουμε και με διαφορετικό τρόπο, με τη χρήση ενός μεταβολέα. Φτιάχνουμε ένα μεταβολέα με το όνομα *πλευρά*. Η διαδικασία *τετράγωνο* θα γραφεί κάπως έτσι:

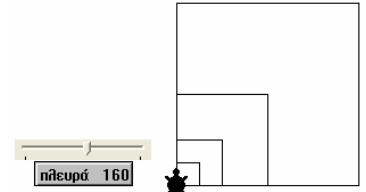
Για τετράγωνο

ΕΠΑΝΑΛΑΒΕ 4 [ΜΠ πλευρά ΔΕ 90]

Τέλος

Παρατηρήστε ότι **δεν ορίσαμε** παράμετρο-μεταβλητή δίπλα στο όνομα της διαδικασίας. Παρ' όλα αυτά χρησιμοποιούμε τη μεταβλητή *πλευρά*, που δεν είναι τίποτα άλλο παρά η τιμή του μεταβολέα με το όνομα *πλευρά*, και γι' αυτό, άλλωστε, δεν έχει άνω και κάτω τελεία.

Μπορούμε να χρησιμοποιήσουμε ακόμη περισσότερο τη διαδικασία δημιουργώντας ένα κουμπί με το οποίο θα τη συνδέσουμε. Έτσι, θα μετακινούμε το ρυθμιστικό και με ένα κλικ στο κουμπί θα δημιουργείται το τετράγωνο με μήκος πλευράς αυτό που έχουμε ορίσει με τον μεταβολέα.

**Πώς μπορούμε να χρησιμοποιήσουμε την τιμή ενός μεταβολέα σε μια μαθηματική έκφραση:**

Θέσαμε το ρυθμιστικό του μεταβολέα *πλευρά* στην τιμή 50. Με τις εντολές:

ΔΕΙΞΕ πλευρά

50

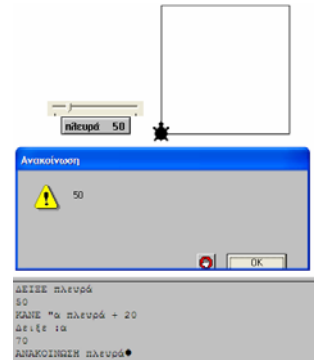
ΚΑΝΕ "α πλευρά + 20

Δείξε :α

70

ΑΝΑΚΟΙΝΩΣΗ πλευρά

διαπιστώσαμε όντως ότι η τιμή 50 έχει τεθεί στο μεταβολέα. Επιπλέον χρησιμοποιήσαμε τη βοηθητική μεταβλητή *α* για να δούμε πως η τιμή του μεταβολέα *πλευρά* μπορεί να χρησιμοποιηθεί σε μια μαθηματική έκφραση (*πλευρά* + 20). Το αποτέλεσμα της έκφρασης *πλευρά* + 20 μπορεί να εκχωρηθεί στη μεταβλητή *α* χωρίς κανένα πρόβλημα.

**Επιλέγοντας****Τι ονομάζουμε επιλογή σε έναν αλγόριθμο:**

Συχνά, συμβαίνει σε έναν αλγόριθμο να χρειαστεί να διακοπεί η βήμα προς βήμα εκτέλεση των εντολών του. Στις περιπτώσεις αυτές, θα πρέπει να γίνεται επιλογή των βημάτων-εντολών που πρέπει να εκτελεστούν. Το ποια θα είναι αυτά εξαρτάται κάθε φορά από τα δεδομένα εισόδου. Για παράδειγμα, έστω ο παρακάτω αλγόριθμος:

1. Ρώτησε το χρήστη πόσων χρονών είναι.
2. Διατήρησε την απάντηση που δόθηκε.
3. Αν η απάντηση είναι αριθμός μεγαλύτερος του 18, τότε εμφάνισε το μήνυμα "Ενήλικας"

Το αν θα εμφανιστεί μήνυμα στην οθόνη εξαρτάται αποκλειστικά από τα δεδομένα εισόδου που έδωσε ο χρήστης. Τα ενδεχόμενα είναι: ή να εμφανιστεί το μήνυμα «Ενήλικας» ή να μην εμφανιστεί τίποτα. Σχηματικά:

Η **Επιλογή** (συχνά αναφέρεται και ως **Δομή επιλογής**) λοιπόν είναι η δυνατότητα να επιλέγουμε ποιες εντολές θέλουμε να εκτελεστούν σε έναν αλγόριθμο (ή σε ένα πρόγραμμα) όταν ισχύει μια συνθήκη.

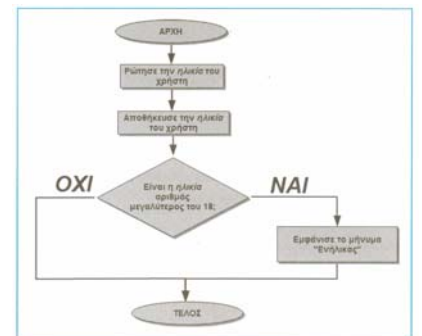
Πώς συντάσσεται μία συνθήκη:

Η συνθήκη έχει δύο αποτελέσματα: το 1 και το 0, το αληθές και το ψευδές. Είτε ισχύει (αληθής) είτε δεν ισχύει (ψευδής). Η σύνταξη μιας συνθήκης απαιτεί **δύο μέρη** (ας τα ονομάσουμε Α και Β) και τους **τελεστές** που κάνουν τη σύγκριση και είναι οι εξής:

- = (το σύμβολο της ισότητας και ελέγχει αν οι δύο τιμές/μεταβλητές είναι πανομοιότυπες)
- > (το σύμβολο «μεγαλύτερο από» και ελέγχει αν η πρώτη τιμή είναι μεγαλύτερη από τη δεύτερη)
- < (το σύμβολο «μικρότερο από» και ελέγχει αν η πρώτη τιμή είναι μικρότερη από τη δεύτερη)
- **ανήκει?** (η εντολή αυτή ελέγχει αν η πρώτη τιμή ανήκει μέσα στην ομάδα των τιμών που ακολουθεί – αντί για δεύτερη τιμή)

Τι είναι το «λογικό ΑΝ... ΤΟΤΕ...»:

Ο προγραμματισμός δεν είναι μια απλή ομάδα εντολών - διαδικασιών που πρέπει να εκτελεστούν με καθορισμένη σειρά. Ο σωστός προγραμματιστής πρέπει να λαμβάνει υπόψη του και το περιβάλλον στο οποίο «τρέχει» το πρόγραμμα καθώς, επίσης, και τις κινήσεις του χρήστη. Για παράδειγμα, σε προηγούμενες δραστηριότητες αυτού του κεφαλαίου φτιάξαμε διαδικασίες που δημιουργούν τετράγωνο, εφόσον ο χρήστης έδινε μια τιμή. Τι γίνεται αν ο χρήστης δώσει μια μηδενική ή μια τεράστια τιμή; Κανείς δεν του το απαγορεύει αυτό, οπότε είναι δυνατόν. Αν, συμβεί κάτι τέτοιο, τότε είτε το πρόγραμμα θα εμφανίσει στην οθόνη κάποιο περίεργο αποτέλεσμα είτε κάποιο μήνυμα λάθους. Αυτό δεν είναι σωστός προγραμματισμός. Ο επαγγελματίας προγραμματιστής πρέπει να



προβλέπει και να μην επιτρέπει να γίνονται σφάλματα. Ένας από τους μεγαλύτερους βοηθούς του προγραμματιστή σε αυτό τον τομέα είναι η διαδικασία AN ... ΤΟΤΕ ... η οποία συντάσσεται ως εξής:

AN συνθήκη ΤΟΤΕ εντολές

Το λογικό AN... ΤΟΤΕ... είναι μια διαδικασία της γλώσσας προγραμματισμού κατά την οποία το πρώτο κομμάτι (το AN ΣΥΝΘΗΚΗ) ελέγχει αν η συνθήκη που του βάλαμε είναι αληθής (ισχύει), οπότε εκτελούνται οι εντολές που ακολουθούν το ΤΟΤΕ.

Ποιες εντολές της Logo χρησιμοποιούν το λογικό AN... ΤΟΤΕ... (για την υποστήριξη της Επιλογής):

Υπάρχουν δύο εντολές που χρησιμοποιούν το λογικό AN... ΤΟΤΕ... στη Logo, η AN και η ΑΝΔΙΑΦΟΡΕΤΙΚΑ.

Πως χρησιμοποιείται η εντολή AN:

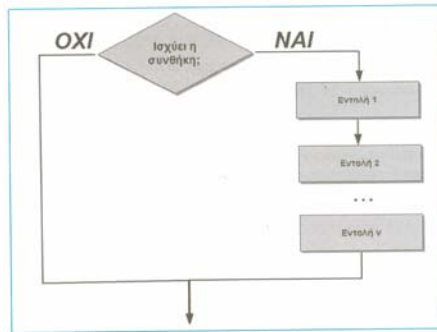
Η εντολή AN συντάσσεται ως εξής:

AN συνθήκη "διαδικασία

AN συνθήκη [εντολή_1 εντολή_2 ... εντολή_n]

όπου εντολή_1 εντολή_2 ... εντολή_n οι εντολές που θα πρέπει να εκτελεστούν στην περίπτωση που η συνθήκη ισχύει.

Σχηματικά:



Έστω, λοιπόν, ότι η διαδικασία είναι το τετράγωνο που φτιάξαμε πιο πάνω. Η συνθήκη θα μπορούσε να είναι: το να μην είναι η πλευρά μεγαλύτερη από 100. Θα είχαμε κάτι σαν αυτό:

AN πλευρά < 101 "τετράγωνο

Παρατηρήστε ότι βάλαμε πλευρά < 101 και όχι πλευρά < 100. Αυτό το κάναμε, επειδή θέλουμε να «τρέξει» η διαδικασία στην περίπτωση που η πλευρά έχει μήκος 100. Αν βάζαμε < 100, τότε η διαδικασία δεν θα έτρεχε αν δίναμε στην πλευρά την τιμή 100. Τέλος, δεν πρέπει να ξεχνάμε το διάστημα (κενό) ανάμεσα σε κάθε διαφορετικό μέλος αυτής.

Ας το εξηγήσουμε λίγο αυτό. Αν η συνθήκη (π.χ. πλευρά < 101) είναι αληθής, τότε θα εκτελεστεί η πρώτη εντολή (εντολή1). Αν η συνθήκη είναι λάθος (ψευδής), τότε θα εκτελεστεί η δεύτερη (εντολή2).

Πως χρησιμοποιείται η εντολή ΑΝΔΙΑΦΟΡΕΤΙΚΑ:

Για να καταλάβουμε τη διαφορά της ΑΝΔΙΑΦΟΡΕΤΙΚΑ από την AN, θα τροποποιήσουμε τον πιο πάνω αλγόριθμο ως εξής:

1. Ρώτησε το χρήστη πόσων χρονών είναι.
2. Διατήρησε την απάντηση που δόθηκε.
3. Αν η απάντηση είναι αριθμός μεγαλύτερος του 18, τότε εμφάνισε το μήνυμα "Ενήλικας", αλλιώς εμφάνισε το μήνυμα "Παιδί".

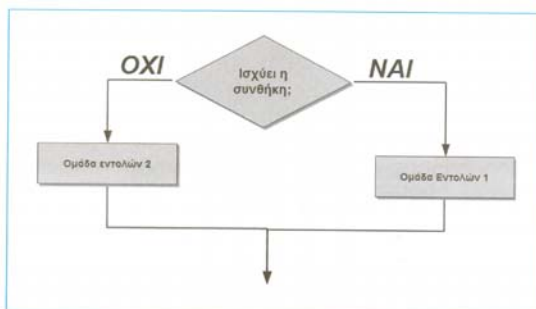
Με τρεις εντολές AN θα μπορούσαμε να έχουμε το ίδιο αποτέλεσμα (AN αριθμός <18 ΤΟΤΕ..., AN αριθμός =18 ΤΟΤΕ..., AN αριθμός >18 ΤΟΤΕ...), αλλά υπάρχει πιο εύκολος τρόπος. Η ΑΝΔΙΑΦΟΡΕΤΙΚΑ είναι η εντολή που χρησιμοποιείται αν θέλουμε να δούμε αν ισχύει μια συνθήκη ή αν ισχύει η αντίθετή της.

Η εντολή ΑΝΔΙΑΦΟΡΕΤΙΚΑ συντάσσεται ως εξής:

ΑΝΔΙΑΦΟΡΕΤΙΚΑ συνθήκη [εντολές_1] [εντολές_2]

όπου εντολές_1 οι εντολές που θα πρέπει να εκτελεστούν αν ισχύει η συνθήκη και εντολές_2 οι εντολές που θα πρέπει να εκτελεστούν στην περίπτωση που η συνθήκη δεν ισχύει.

Σχηματικά:




Δημιουργώντας πιο σύνθετες εφαρμογές με τη Logo

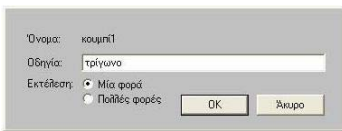
Τι είναι το κουμπί στο MicroWorlds Pro:

Κουμπί είναι ένα αντικείμενο (στον προγραμματισμό αντικείμενο ονομάζουμε οτιδήποτε έχει ξεχωριστές ιδιότητες) στην Επιφάνεια εργασίας το οποίο προσομοιώνει τη χρήση ενός πραγματικού κουμπιού και εκτελεί συγκεκριμένες εντολές ή διαδικασίες κάθε φορά που κάνουμε κλικ πάνω του.

Πώς δημιουργείται ένα κουμπί:

Αν θέλουμε να δημιουργήσουμε ένα κουμπί που να ενεργοποιεί τη διαδικασία τρίγωνο (ΣΤΚ ΕΠΑΝΑΛΑΒΕ 3 [ΜΠ 100 ΔΕ 120]), πατάμε το κουμπί *Δημιουργία Κουμπιού*  της Γραμμής εργαλείων αφού μετακινήσουμε το δείκτη στο σημείο της Επιφάνειας εργασίας όπου επιθυμούμε να εμφανίζεται το κουμπί μας, κάνουμε κλικ.

Θα εμφανιστεί το παρακάτω παράθυρο:




Το όνομα του κουμπιού που δίνει το MicroWorlds Pro είναι το «κουμπ1». Στο πλαίσιο κειμένου *Οδηγία:* μπορούμε να δώσουμε το όνομα της διαδικασίας (ή αναλυτικά μια σειρά εντολών) που θέλουμε να εκτελούνται κάθε φορά που πατάμε το κουμπί μας. Στο πεδίο αυτό θα γράψουμε το όνομα της διαδικασίας *τρίγωνο*.

Τέλος, το MicroWorlds Pro μας δίνει τη δυνατότητα οι εντολές αυτές να εκτελούνται μία φορά (τη στιγμή που θα κάνουμε κλικ πάνω στο κουμπί) ή πολλές φορές.

Όταν τελειώσουμε, πατάμε OK και βλέπουμε πως ένα νέο κουμπί έχει δημιουργηθεί πάνω στην Επιφάνεια εργασίας: το κουμπί. τρίγωνο Κάνουμε κλικ πάνω του και θα εκτελεστεί η διαδικασία *τρίγωνο*.

Παρατηρούμε πως το κείμενο που αναφέρεται πάνω στο κουμπί είναι και το όνομα της διαδικασίας (ή ολόκληρη η σειρά των εντολών) με την οποία έχει συνδεθεί. Καλό είναι λοιπόν τα ονόματα των διαδικασιών μας είναι να είναι αντίστοιχα του περιεχομένου τους, έτσι ώστε να μην μπερδεύουμε το χρήστη.


Πώς μπορούμε να χρησιμοποιήσουμε ένα κουμπί για τη διαρκή εκτέλεση εντολών:

Έστω πως θέλουμε να κάνουμε τη χελώνα μας να κινείται συνεχώς προς τα μπροστά. Δημιουργούμε ένα κουμπί: και μέσα στο πλαίσιο *Οδηγία:* γράφουμε την εντολή *ΜΠ 20*. Αν τώρα επιλέξουμε *Εκτέλεση: Πολλές φορές* και πατήσουμε OK, θα δημιουργήσουμε ένα νέο κουμπί ΜΠ 20 στην Επιφάνεια εργασίας. Κάνοντας κλικ πάνω του, η χελώνα ξεκινά να κινείται ασταμάτητα. Για να σταματήσουμε την κίνησή της, θα κάνουμε κλικ στο κουμπί  της Γραμμής εργαλείων.

Τι είναι το πλαίσιο κειμένου στο MicroWorlds Pro:

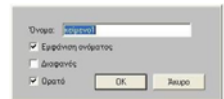
Πλαίσιο κειμένου είναι ένα αντικείμενο που μπορεί να τοποθετηθεί στην Επιφάνεια εργασίας και να περιέχει κείμενο. Ένα πλαίσιο έχει συγκεκριμένες διαστάσεις και όνομα. Μέσα σ' αυτό μπορούν να εκτελεστούν εντολές εξόδου ή να γράψει απευθείας ο χρήστης.

Πώς δημιουργούμε και τροποποιούμε ένα πλαίσιο κειμένου:

Κάνουμε κλικ στο κουμπί *Δημιουργία πλαισίου κειμένου*  πάνω στη Γραμμή εργαλείων. Στη συνέχεια κάνουμε κλικ στο σημείο που επιθυμούμε να εμφανιστεί το πλαίσιο μας και «τραβάμε» με το δείκτη του ποντικιού την κάτω δεξιά γωνία του για να του δώσουμε τις διαστάσεις που επιθυμούμε. Παρατηρούμε πως τα πλαίσιά μας έχει συγκεκριμένο όνομα και, αν κάνουμε κλικ στο όνομα αυτό, μπορούμε να πληκτρολογήσουμε κείμενο μέσα σε αυτό.



Επιπλέον, μπορούμε να αλλάζουμε το όνομα του πλαισίου μας. Κάνουμε δεξιά κλικ πάνω στην περιοχή κειμένου του πλαισίου και μετά κάνουμε κλικ στην επιλογή *Επεξεργασία*. Θα εμφανιστεί το παρακάτω παράθυρο στο οποίο είναι δυνατόν να αλλάξουμε το όνομα και κάποιες παραμέτρους εμφάνισης του πλαισίου:

Ποιες βασικές εντολές ισχύουν για τα πλαίσια κειμένου:

Επειδή είναι δυνατόν να έχουμε περισσότερα από ένα πλαίσια κειμένου, θα πρέπει να είναι σαφές κάθε φορά σε ποιο πλαίσιο αναφέρονται οι εντολές μας. Για το λόγο αυτό, κάθε φορά που θέλουμε να εκτελέσουμε εντολές που αφορούν ένα πλαίσιο πληκτρολογούμε πρώτα το όνομά του πλαισίου σαν να ήταν εντολή. Αν, για παράδειγμα, έχουμε ένα πλαίσιο που ονομάζεται *αποτέλεσμα*, θα πρέπει να γράψουμε τη λέξη:

αποτέλεσμα,

πριν από τη χρήση κάποιων από τις παρακάτω εντολές. Προσέχουμε να μην ξεχάσουμε το χαρακτήρα «*,*» (κόμμα) ο οποίος υποδεικνύει στον υπολογιστή πως αναφερόμαστε σε πλαίσιο κειμένου και όχι σε όνομα διαδικασίας.

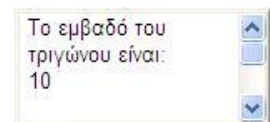
- Εντολή ΤΥΠΩΣΕ (ΤΥ): Εμφανίζει μια λέξη, ένα κείμενο ή ένα αποτέλεσμα μιας μαθηματικής πράξης μέσα στο πλαίσιο μας. Συντάσσεται ακριβώς όπως η εντολή ΔΕΙΞΕ. Για παράδειγμα, έστω πως έχουμε δημιουργήσει το πλαίσιο *αποτέλεσμα* και τη διαδικασία *εμβαδό*:

```
Για εμβαδό
αποτέλεσμα,
ΤΥ [Το εμβαδό του τριγώνου είναι:]
ΚΑΝΕ "βάση 5
ΚΑΝΕ "ύψος 4
ΚΑΝΕ "ε (:βάση * :ύψος) / 2
ΤΥ :ε
```

Τέλος

Η εκτέλεση της διαδικασίας αυτής θα μας δώσει τα εξής δεδομένα εξόδου μέσα στο πλαίσιο μας:

- Εντολή ΣΒΗΣΕΚΕΙΜΕΝΟ: Καθαρίζει το πλαίσιο κειμένου από οποιαδήποτε κείμενο περιέχεται σε αυτό.




αποτέλεσμα

Πώς μπορούμε να κάνουμε μια σειρά εντολών να επαναλαμβάνεται διαρκώς:

Υπάρχουν περιπτώσεις όπου θα χρειαστεί κάποιες εντολές να επαναλαμβάνονται διαρκώς. Με την εντολή ΣΥΝΕΧΩΣ μια ομάδα εντολών επαναλαμβάνεται χωρίς να σταματάει. Συντάσσεται ως εξής:

ΣΥΝΕΧΩΣ [εντολή/εντολές]

Αν για παράδειγμα, δώσουμε ΣΥΝΕΧΩΣ [ΜΠ 1] η χελώνα θα προχωράει προς τα μπροστά διαρκώς και θα σταματήσει μόνο όταν πατήσουμε το κουμπί *Διακοπή όλων*  στη Γραμμή εργαλείων.

Πώς μπορούμε να καταλάβουμε ότι δυο χελώνες βρίσκονται στο ίδιο σημείο στη Σείδα:

Αυτό μπορεί να γίνει χρησιμοποιώντας την εντολή *ΑΓΓΙΖΕΙ?* Η συνθήκη αυτή ελέγχει αν δύο χελώνες βρίσκονται στο ίδιο σημείο στην Επιφάνεια εργασίας (αν δηλαδή έχουν αγγίξει η μια την άλλη). Συντάσσεται ως εξής:

ΑΓΓΙΖΕΙ? "χελώνα1 "χελώνα2

Όπου *χελώνα1* και *χελώνα2* τα ονόματα των δύο χελωνών. Άρα, έχει νόημα αν συντάσσεται ύστερα από μια εντολή επιλογής AN ή ANΔΙΑΦΟΡΕΤΙΚΑ.

Για παράδειγμα, αν έχουμε δυο χελώνες x1 και x2:

ΑΓΓΙΖΕΙ? "x1 "x2

Πως μπορούμε να στρίψουμε μια χελώνα προς την κατεύθυνση μιας άλλης;

Η εντολή με την οποία στρέφεται μια χελώνα προς την κατεύθυνση μιας άλλης (χωρίς να χρειάζεται να υπολογίσουμε τις μοίρες) είναι η ΚΟΙΤΑΠΡΟΣ. Συντάσσεται ως εξής:

"χελώνα1, ΚΟΙΤΑΠΡΟΣ "χελώνα2

Όπου *χελώνα1* η χελώνα που θέλουμε να στραφεί προς τη *χελώνα2*. Προσοχή, το πιο συχνό συντακτικό λάθος είναι να ξεχάσει κανείς το χαρακτήρα «,» (κόμμα) μετά το όνομα της πρώτης χελώνας.

Πως μπορούμε να σταματήσουμε την εκτέλεση μιας διαδικασίας;

Η εντολή ΣΤΑΜΑΤΗΣΕΜΕ, αν δοθεί μέσα σε μια διαδικασία τερματίζει την εκτέλεσή της.

Πως μπορούμε να προκαλέσουμε μια χρονική καθυστέρηση στο πρόγραμμα;

Όταν έχουμε μια επανάληψη (όπως με την εντολή ΣΥΝΕΧΩΣ) υπάρχουν φορές που χρειάζεται να γίνει μία παύση στη διαδικασία. Η παύση αυτή γίνεται με την εντολή ΠΕΡΙΜΕΝΕ η οποία συντάσσεται ως εξής:

ΠΕΡΙΜΕΝΕ αριθμός

(όπου *αριθμός* ένας αριθμός ο οποίος αντιπροσωπεύει δέκατα του δευτερολέπτου)

Δηλαδή το ΠΕΡΙΜΕΝΕ 50 σημαίνει παύση για 5 δευτερόλεπτα.

ΠΕΡΙΕΧΟΜΕΝΑ

Σημειώσεις για την Γ' Γυμνασίου	1
“Αλγόριθμοι και Δομές Δεδομένων”	1
<i>Τι είναι Αλγόριθμος</i>	1
<i>Σπουδαιότητα Αλγορίθμων</i>	1
<i>Περιγραφή και αναπαράσταση αλγορίθμων</i>	2
<i>Σύμβολα ροής διαγράμματος</i>	2
<i>Αξιολόγηση αλγορίθμων</i>	2
<i>Δομές δεδομένων</i>	3
<i>Γλώσσες Προγραμματισμού</i>	3
Το περιβάλλον προγραμματισμού MicroWorlds Pro	5
<i>Τι είναι το Μενού Επιλογών (Γραμμή Εντολών) και σε τι χρησιμεύει;</i>	5
<i>Τι είναι η Γραμμή Εργαλείων και σε τι χρησιμεύει;</i>	5
<i>Τι είναι η Επιφάνεια εργασίας (Σελίδα);</i>	5
<i>Τι είναι το Κέντρο Εντολών και σε τι χρησιμεύει;</i>	5
<i>Τι είναι η Περιοχή Καρτελών και σε τι χρησιμεύει;</i>	5
Εντολές	5
<i>Ποιες εντολές ονομάζονται εξόδου;</i>	5
<i>Πώς μπορούμε να εμφανίσουμε ένα κείμενο στο Κέντρο Εντολών;</i>	5
<i>Πώς μπορούμε να εμφανίσουμε ένα κείμενο σε μια Σελίδα;</i>	5
<i>Πώς μπορούμε να κάνουμε μαθηματικές πράξεις;</i>	5
<i>Πώς μπορούμε να υψώσουμε σε δύναμη έναν αριθμό;</i>	5
<i>Πώς μπορούμε να εμφανίσουμε μηνύματα μαζί με τα αποτελέσματα αριθμητικών πράξεων;</i>	6
<i>Πώς μπορούμε να εμφανίσουμε ένα παράθυρο με ένα μήνυμα στην οθόνη του υπολογιστή μας μέσα από τη Logo;</i>	6
<i>Πώς μπορούμε να εμφανίσουμε ένα παράθυρο που να προβάλλει ένα ερώτημα και να δέχεται την απάντηση του χρήστη;</i>	6
<i>Πώς θα ξέρουμε ποια απάντηση έδωσε ο χρήστης στο ερώτημά μας;</i>	6
<i>Πώς μπορούμε να χρησιμοποιήσουμε την ΑΠΑΝΤΗΣΗ;</i>	6
<i>Τι είναι η χελώνα στη γλώσσα Logo;</i>	6
<i>Πώς εμφανίζεται μια χελώνα στη Σελίδα μας;</i>	6
<i>Πώς μπορούμε να αλλάξουμε το όνομα μιας χελώνας;</i>	6
<i>Πώς μπορούμε να βάλουμε μια άλλη εικόνα στη θέση της χελώνας;</i>	6
<i>Πώς μπορούμε να έχουμε δύο ή περισσότερες χελώνες στη Σελίδα μας;</i>	6
<i>Με ποιες εντολές μπορούμε να μετακινήσουμε τη χελώνα;</i>	6
<i>Πως σταματάμε τη χελώνα;</i>	7
<i>Η χελώνα προχωράει αλλά δεν βλέπω κάποιο ίχνος. Γιατί γίνεται αυτό;</i>	7
<i>Πώς μπορώ να επαναφέρω τη Σελίδα στην πρώτη της μορφή με τη χελώνα στο κέντρο της;</i>	7
<i>Πώς αλλάζει το χρώμα του στυλό της χελώνας;</i>	7
<i>Ποιοι είναι οι κωδικοί αριθμοί των χρωμάτων;</i>	7
<i>Πώς αλλάζει το πάχος της γραμμής που αφήνει η χελώνα;</i>	7
<i>Πώς μπορώ να «πω» στη Logo να επαναλάβει μια εντολή πολλές φορές χωρίς να χρειαστεί να την πληκτρολογήσουμε ξανά;</i>	7
Διαδικασίες	8
<i>Πώς δημιουργούμε μια διαδικασία στο MicroWorlds Pro;</i>	8
<i>Πώς μπορούμε να εκτελέσουμε τη διαδικασία για «τετράγωνο» ώστε να ορίζουμε εμείς το μέγεθος της πλευράς του τετραγώνου;</i>	9
<i>Πώς μπορούμε να εκτελέσουμε τη διαδικασία «τρίγωνο» ώστε να ορίζουμε εμείς το μέγεθος της πλευράς του τριγώνου;</i>	9
Μεταβλητές	9
<i>Μπορούμε να αλλάξουμε τα δεδομένα μιας μεταβλητής;</i>	9
<i>Πώς ξεχωρίζει μια μεταβλητή από το όνομα μιας διαδικασίας ή από μια εντολή;</i>	9
<i>Τι πρέπει να προσέχουμε όταν δίνουμε όνομα σε μια μεταβλητή;</i>	9
<i>Ποιες είναι οι βασικές ιδιότητες μιας μεταβλητής;</i>	9
<i>Πώς ορίζουμε την τιμή μιας μεταβλητής (ή βάζουμε ή εκχωρούμε δεδομένα σε μια μεταβλητή);</i>	10
<i>Πώς προβάλλουμε (εμφανίζουμε) την τιμή που είναι αποθηκευμένη μέσα σε μια μεταβλητή;</i>	10
<i>Ποιά είναι η λειτουργία της εντολής KANE "α :α + 1";</i>	10
<i>Τι είναι ο μεταβολέας;</i>	10
<i>Πώς δημιουργούμε ένα μεταβολέα;</i>	10
<i>Πώς μπορούμε να χρησιμοποιήσουμε ένα μεταβολέα;</i>	10
<i>Πώς μπορούμε να χρησιμοποιήσουμε ένα μεταβολέα σε μια διαδικασία;</i>	11
<i>Πώς μπορούμε να χρησιμοποιήσουμε την τιμή ενός μεταβολέα σε μια μαθηματική έκφραση;</i>	11
Επιλέγοντας	11
<i>Τι ονομάζουμε επιλογή σε έναν αλγόριθμο;</i>	11
<i>Πώς συντάσσεται μία συνθήκη;</i>	11
<i>Τι είναι το «λογικό ΑΝ... ΤΟΤΕ...»;</i>	11
<i>Ποιες εντολές της Logo χρησιμοποιούν το λογικό ΑΝ... ΤΟΤΕ... (για την υποστήριξη της Επιλογής);</i>	12
<i>Πως χρησιμοποιείται η εντολή ΑΝ;</i>	12
<i>Πως χρησιμοποιείται η εντολή ΑΝΔΙΑΦΟΡΕΤΙΚΑ;</i>	12
Δημιουργώντας πιο σύνθετες εφαρμογές με τη Logo	12
<i>Τι είναι το κουμπί στο MicroWorlds Pro;</i>	12

Πώς δημιουργείται ένα κουμπί;	13
Πώς μπορούμε να χρησιμοποιήσουμε ένα κουμπί για τη διαρκή εκτέλεση εντολών;.....	13
Τι είναι το πλαίσιο κειμένου στο MicroWorlds Pro;	13
Πώς δημιουργούμε και τροποποιούμε ένα πλαίσιο κειμένου;	13
Ποιες βασικές εντολές ισχύουν για τα πλαίσια κειμένου;	13
Πώς μπορούμε να κάνουμε μια σειρά εντολών να επαναλαμβάνεται διαρκώς;	13
Πώς μπορούμε να καταλάβουμε ότι δυο χελώνες βρίσκονται στο ίδιο σημείο στη Σείδα;	13
Πώς μπορούμε να στρίψουμε μια χελώνα προς την κατεύθυνση μιας άλλης;.....	14
Πώς μπορούμε να σταματήσουμε την εκτέλεση μιας διαδικασίας;.....	14
Πώς μπορούμε να προκαλέσουμε μια χρονική καθυστέρηση στο πρόγραμμα;.....	14