

η Logo

στην εκπαιδευτική
διαδικασία

3

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
2000



ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ



026000318501

005.133

ΜΙΚ



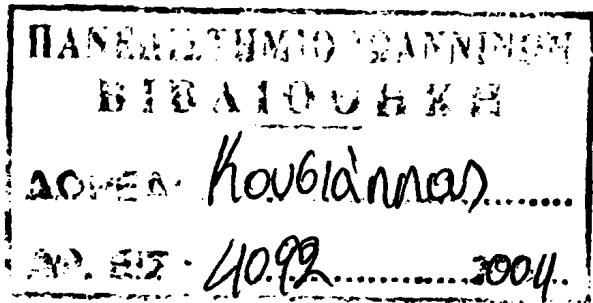
Τάσος Μικρόπουλος - Τάσος Λαδιάς

Η LOGO
στην
εκπαιδευτική
διαδικασία



Πανεπιστήμιο Ιωαννίνων 2000





«Η LOGO στην εκπαιδευτική διαδικασία»

© Τάσος Μικρόπουλος - Τάσος Λαδιάς

ISBN 960 - 233 - 040 - 6

Επιτρέπεται η αναπαραγωγή του παρόντος βιβλίου ή μέρος αυτού, υπό οποιαδήποτε έντυπη ή ηλεκτρονική μορφή, μόνο μετά από έγγραφη άδεια των συγγραφέων.

Τάσος Μικρόπουλος
Παιδαγωγικό Τμήμα Δημοτικής Εκπαίδευσης,
Πανεπιστήμιο Ιωαννίνων
Πανεπιστημιούπολη
45 110 Ιωάννινα
e-mail: amikrop@cc.uoi.gr

Τάσος Λαδιάς
1ο Τ.Ε.Ε. Κορυδαλλού

e-mail: ladias@ath.forthnet.gr



**Στον Αριστοτέλη,
το Λεωνίδα,
το Νικόλα
και το Δημήτρη**



Περιεχόμενα

Πρόλογος	
Εισαγωγή	1
Επεξεργασία δεδομένων	15
Τεχνολογία ηλεκτρονικών υπολογιστών	31
Λογισμικό	45
Εφαρμογές ηλεκτρονικών υπολογιστών	55
Πληροφορική και εκπαίδευση	69
Η ουσία των πραγμάτων	97
Πλίνθοι και κέραμοι	131
Εκδρομή στην Ελατού	159
Το μηδέν και το άπειρο	177
Animation / Εμ-ψυχο-ποίηση	209
Ιεραρχικός σχεδιασμός	221
Αναδρομικές διαδικασίες	247
Fractals	273
Παράλληλος προγραμματισμός	289
Λίστες. Μια δυναμική δομή της Logo	297
Μικρόκοσμοι	311
ΠΑΡΑΡΤΗΜΑΤΑ	
Η LogoWriter 2.0 της LCSΙ	326
Πρωτογενείς διαδικασίες - εντολές της LogoWriter	328
Πλήκτρα ειδικών λειτουργιών - Συνδυασμοί πλήκτρων	334
Προειδοποιητικά μηνύματα	335



Πρόλογος

Το παρόν πόνημα αποτελεί την εξέλιξη του βιβλίου “Πληροφορική και Εκπαίδευση. Δημιουργίες νοητικών μοντέλων στο ανοικτό περιβάλλον της γλώσσας Logo”. Στο πρώτο ξεφύλλισμα η δομή των δύο βιβλίων φαίνεται η ίδια. Είναι όμως δύο διαφορετικά βιβλία με διαφορετικό περιεχόμενο, με το παρόν να περιέχει τη διδακτική πείρα και χρήση του πρώτου επί πέντε έτη. Το “Πληροφορική και Εκπαίδευση. Δημιουργίες νοητικών μοντέλων στο ανοικτό περιβάλλον της γλώσσας Logo” έχει διδαχθεί στα Παιδαγωγικά Τμήματα Δημοτικής Εκπαίδευσης και Νηπιαγωγών των Πανεπιστημίων Ιωαννίνων, Θεσσαλίας και Αιγαίου, στο Τμήμα Φιλολογίας Ιωαννίνων, σε τμήματα εξομοίωσης και προγράμματα επανεκπαίδευσης δασκάλων, και σε αρκετά προγράμματα επιμόρφωσης εκπαιδευτικών (Π.Ε.Κ.). Η πείρα και εμπειρικές έρευνες σε εξέλιξη, καθώς και η συνεχής αλληλεπίδραση με εκπαιδευτικούς και φοιτητές οδήγησαν στη συγγραφή ενός νέου ουσιαστικά βιβλίου, διατηρώντας τη δομή του πρώτου. Έτσι έχει μειωθεί ο όγκος του πρώτου μέρους σχετικά με τα εισαγωγικά θέματα πληροφορικής, και έχει εμπλουτισθεί και αλλάξει το μέρος που αφορά στην ανάπτυξη πνευματικών δεξιοτήτων στο περιβάλλον της γλώσσας προγραμματισμού Logo. Έχει γίνει προσπάθεια αποφυγής παραδειγμάτων που άπτονται άμεσα θεμάτων μαθηματικών για να γίνει φανερό ότι ο προγραμματισμός δεν αφορά μόνο αυτά. Υπάρχει ένα εδάφιο με τη δημιουργία εργαλείων λογισμικού για την υπερπήδηση προβλημάτων της έκδοσης της Logo που χρησιμοποιείται με κύριο παράδειγμα την είσοδο τιμών από το πληκτρολόγιο. Περιλαμβάνεται ένα εκτενές κεφάλαιο σχετικό με αναδρομικές διαδικασίες που χρησιμοποιούνται στην καθημερινή ζωή και σπάνια πραγματεύονται στον προγραμματισμό. Υπάρχει ένα κεφάλαιο σχετικό με παράλληλο προγραμματισμό με το παράδειγμα ενός αγώνα δρόμου του λαγού και της χελώνας. Επίσης πραγματεύεται το θέμα δομών λίστας με ένα παράδειγμα εισόδου - εξόδου αυτοκινήτων από γκαράζ όπου εμπεριέχεται το σπέρμα της τεχνητής ευφυΐας με το πρόγραμμα να αποφασίζει και να αλλάζει τη δομή του, να αυτοβελτιώνεται και να γίνεται ευφυέστερο από μόνο του.

Ο εξελληνισμός των διαδικασιών έχει προχωρήσει, με τις περισσότερες δια-



δικασίες του βιβλίου να έχουν ελληνικές ονομασίες για μεγαλύτερη λειτουργικότητα στη χρήση τους ως ρουτίνες - εργαλεία σε άλλες.

Η έκδοση της Logo είναι η LCSI Logo v 2.0 για περιβάλλον DOS (παράρτημα Α), συμβατή με τις νεότερες εκδόσεις για περιβάλλοντα Windows x. Επιπλέον, οι εντολές και η σύνταξή τους είναι παρόμοιες με όλων των τύπων της Logo που κυκλοφορούν.

Και επανέρχεται το ερώτημα. Γιατί Logo;

Η Logo έχει πέσει θύμα της ίδιας της επιτυχίας στην πρωτοβάθμια εκπαίδευση. Έχει αποκτήσει τη φήμη της απλοϊκής γλώσσας για μικρά παιδιά. Γιατί τότε χρησιμοποιείται ως η βάση για ανάπτυξη πνευματικών δεξιοτήτων και εισαγωγή στην επιστήμη των υπολογιστών σε επίπεδο τριτοβάθμιας εκπαίδευσης;

Η αλήθεια είναι ότι η Logo είναι από τις ισχυρότερες γλώσσες προγραμματισμού για προσωπικούς υπολογιστές. Είναι μια διάλεκτος της LISP, της γλώσσας που χρησιμοποιείται σε ερευνητικά προγράμματα της επιστήμης των υπολογιστών. Τι σημαίνει όμως ότι μια γλώσσα είναι ισχυρή; Δε σημαίνει ότι μπορούν να γραφούν προγράμματα για εργασίες που δε μπορούν να γίνουν με άλλες γλώσσες προγραμματισμού. Υπό αυτήν την έννοια, όλες οι γλώσσες είναι ίδιες. Αν κάποιος προγραμματίζει σε Logo μπορεί και σε Pascal ή BASIC, και το αντίστροφο. Η ισχύς μιας γλώσσας φαίνεται στο κατά πόσο βοηθά στη συγκέντρωση του χρήστη στην επίλυση του συγκεκριμένου προβλήματος χωρίς να τον αποσπά με τεχνικά θέματα και τους περιορισμούς της. Αυτό αποτελεί το κύριο χαρακτηριστικό της Logo, η οποία βέβαια δεν περιορίζεται στα γραφικά της χελώνας.

Η ουσία στην εκμάθηση και αξιοποίηση του προγραμματισμού ως εργαλείου ανάπτυξης πνευματικών δεξιοτήτων βρίσκεται στην ισορροπία μεταξύ της θεωρητικής και πρακτικής προσέγγισης [B. Harvey, "Computer Science Logo Style" vol. 1, The MIT Press, Cambridge 1986]. Ένα λάθος είναι η ανάγνωση του βιβλίου χωρίς την επαφή με τον υπολογιστή, και ένα δεύτερο η μεταφορά των προγραμμάτων του βιβλίου στον υπολογιστή χωρίς μελέτη.

Το βιβλίο δεν περιέχει ασκήσεις. Αυτό οφείλεται κυρίως στο ότι ο αναγνώστης ενδιαφέρεται πράγματι για τη Logo ώστε να ερευνήσει μόνος του, και για να δοθούν κίνητρα για ανάπτυξη της δημιουργικότητας με την εφεύρεση ασκήσεων από τον ίδιο τον αναγνώστη. Σε ορισμένα μέρη του βιβλίου όμως υπάρχουν



ερωτήματα για παρακίνηση του χρήστη να δοκιμάσει παραλλαγές των διαδικασιών και να ερευνήσει παραπέρα το θέμα.

Ευχαριστούμε το φυσικό Δημήτρη Αγγελή για την κριτική ανάγνωση του κειμένου και τον έλεγχο των διαδικασιών. Επίσης ευχαριστούμε την παιδαγωγό Ευγενία Νικολού για τη συμβολή της στη σχεδίαση του εξωφύλλου του βιβλίου.



1 Εισαγωγή

1.1 Ορισμοί

Τι είναι και πως ορίζεται η πληροφορική; Είναι μια καθολική επιστήμη αφού όλα είναι πληροφορίες; Και αν ναι, είναι αυτόνομη επιστήμη; Είναι τεχνολογία αφού ίσως δε θα υπήρχε χωρίς τους υπολογιστές; Ή μήπως είναι ένα νέο σχήμα ανάλυσης;

Όπως έχει αρχίσει να γίνεται φανερό, ο συνδυασμός όλων των παραπάνω και όχι μόνο αυτών συνιστούν την πληροφορική.

Ο ορισμός της πληροφορικής ως η μηχανοποιημένη επεξεργασία και μετάδοση των πληροφοριών είναι πολύ γενικός και μεταθέτει το πρόβλημα στην κατανόηση της έννοιας της πληροφορίας.

Ετυμολογικά η λέξη “πληροφορία” (information στα Αγγλικά και Γαλλικά) έχει λατινικές ρίζες και πρωτοεμφανίζεται στον Κικέρωνα. Στα λατινικά η λέξη “informatio” προέρχεται από τη “forma” που σημαίνει την εξωτερική μορφή ενός αντικειμένου, ένα σύνολο ιδιοτήτων του και δηλώνει την “απόδοση μορφής” ή την “πράξη διαμόρφωσης”. Το ρήμα “πληροφορώ” σημαίνει ακόμη “διδάσκω” με την έννοια του παιδαγωγώ, διαμορφώνω το πνεύμα. Στη γαλλική γλώσσα του 13ου αιώνα “enformer” σημαίνει διδάσκω, συνδέεται με το δικαστικό κόσμο και έχει την έννοια της πληροφόρησης μιας υπόθεσης [Breton, 1991].

Στη σύγχρονη εποχή υπάρχει μια σύγχυση και στον όρο “πληροφορία” αποδίδεται ερμηνεία από την πλευρά της ενημέρωσης και της δημοσιογραφίας, αλλά και από καθαρά τεχνική σκοπιά.

Η πληροφορία είναι καταγραμμένη από το Διεθνή Οργανισμό Προδιαγραφών (ISO) με τον κωδικό 2382 ως η σημασία που δίνεται από τους ανθρώπους στα δεδομένα, με κοινή συμφωνία για την αναπαράστασή τους. Με άλλα λόγια, πληροφορία είναι η ερμηνευμένη είδηση.

Επιστημονική σημασία και ορισμός στην πληροφορία δίνεται το 1940 με τη ραγδαία



ανάπτυξη των ηλεκτρονικών τηλεπικοινωνιών και της ηλεκτρονικής επεξεργασίας των δεδομένων.

Η θέση που πήρε είναι μεταξύ των θεμελιωδών εννοιών της φυσικής, την ύλη και την ενέργεια. Η πληροφορία αποθηκεύεται, μεταφέρεται και υφίσταται επεξεργασία όπως και η ύλη. Επίσης έχει τη δυνατότητα επίδρασης στους αποδέκτες της, σε αντιστοιχία με την ενέργεια που έχει τη δυνατότητα παραγωγής έργου.

Το 1948 ο Αμερικανός μαθηματικός Κλοντ Σάννον (Claude Shannon) δημοσιεύει τη μαθηματική θεωρία της πληροφορίας στην οποία πραγματεύεται και γενικά θέματα όπως η τάξη και η αταξία, τα σφάλματα και ο έλεγχός τους, οι δυνατότητες και η πραγματοποίησή τους, η αβεβαιότητα και τα όριά τους. Στην προσπάθειά του να λύσει προβλήματα που σχετίζονταν με τις τηλεπικοινωνίες, ο Σάννον στη μαθηματική έκφρασή του για την ποσότητα της πληροφορίας κατέληξε στη μορφή που έχει και το φυσικό μέγεθος της εντροπίας. Η εξίσωσή του δείχνει την τάση που έχουν τα πάντα γύρω μας να γίνουν ακατάστατα όταν αφήνονται να εξελιχθούν αυθόρμητα. Η ενέργεια δε μεταβάλλεται ως προς την ποσότητά της, αλλά υποβαθμίζεται η ποιότητά της. Γίνεται δηλαδή περισσότερο ανοργάνωτη, λιγότερο χρήσιμη κατά τη φυσική ροή των πραγμάτων. Με άλλα λόγια, το σύμπαν οδηγείται προς μια κατάσταση αταξίας. Στην έννοια της εντροπίας βρίσκεται και η αναλογία μεταξύ της ενέργειας και της πληροφορίας, όπου το μη τυχαίο στοιχείο είναι η πληροφορία [Campbell, 1985].

Όπως φαίνεται, η πληροφορία συνδέει πολλούς επιστημονικούς τομείς και αρκετοί από τους κλάδους τους ενώθηκαν για να δημιουργήσουν ένα καινούριο πεδίο γνώσης. Πάντως οι ιδέες που γέννησαν και μετεξέλιξαν την έννοια της πληροφορίας περιλαμβάνουν τρεις άξονες.

Πρώτο, τη διαφοροποίηση μεταξύ της μορφής και του νοήματος [Breton, 1991]. Η διάκριση μεταξύ αυτών των χαρακτηριστικών ενός μηνύματος γίνεται φανερή από το ότι τα διάφορα σύμβολα από τα οποία κατασκευάζεται και παίρνει την τελική του μορφή, υποβάλλονται σε επεξεργασία ανεξάρτητα από τη σημασία τους που αποτελεί το νόημα του μηνύματος.

Δεύτερο, τις συνθήκες αλήθειας λογικών προτάσεων που εμπεριέχουν την ακριβή φύση των συλλογισμών.



Τρίτο, τις τεχνικές μετάδοσης μηνυμάτων με στόχους την κωδικοποίηση των συμβόλων, τη μετατροπή τους σε ηλεκτρικά συνήθως σήματα και την προσπάθεια της αναλλοίωτης μεταφοράς τους σε απόσταση.

Ίσως τώρα να γίνεται κατανοητός ο ορισμός που δόθηκε στην αρχή για την πληροφορική, σίγουρα όμως δεν καλύπτει όλο το φάσμα της. Εξάλλου ένας από τους σκοπούς του βιβλίου είναι να απαντήσει στο σύνολό του σ' αυτήν την ερώτηση.

Η πληροφορική ως όρος πρωτοεμφανίζεται στη δεκαετία του 1960 στη Γερμανία (informatik) και τη Γαλλία (informatique) όπου δημιουργείται από τη συνένωση των λέξεων πληροφορία (inform-ation) και αυτοματική (autom-atique). Ο τελευταίος, είναι ο κλάδος που μελετά τις αρχές και εφαρμογές του αυτοματισμού που ουσιαστικά δηλώνει την αυτορύθμιση ενός συστήματος με ανάδραση.

Για το ίδιο αντικείμενο, οι Αγγλόφωνοι θεωρούν περισσότερο δόκιμο τον όρο επιστήμη των υπολογιστών (computer science) ή το γενικότερο επιστήμη των υπολογισμών (computing science) και σπανίως χρησιμοποιούν τον "informatics". Αυτό δείχνει ότι η πληροφορική είναι στενά συνδεδεμένη με τους υπολογιστές και δεν είχε εμφανισθεί ως ορολογία όσο δεν υπήρχαν οι ηλεκτρονικοί υπολογιστές (Η/Υ).

ΟΠΤΙΚΗ ΓΩΝΙΑ	ΟΡΙΣΜΟΣ
Τεχνολογία λογισμικού	Επιστήμη αλγορίθμων και δομών δεδομένων
Τεχνολογία υλικού	Επιστήμη των υπολογιστών
Θεωρητική	Επιστήμη της μηχανοποιημένης επεξεργασίας συμβόλων
Τεχνητή νοημοσύνη	Επιστήμη μηχανοποίησης της σκέψης
Ανθρωπιστική	Επιστήμη αλληλεπίδρασης του ανθρώπου με τον κόσμο
Τεχνική	Επιστήμη αυτοματοποίησης και προσομοίωσης

Σχήμα 1.1 Ορισμοί της πληροφορικής

Εξαιτίας της ταχύτατης εξέλιξης της τεχνολογίας και των επιπτώσεων της στον άνθρωπο, με το νέο αυτόν τομέα άρχισαν να ασχολούνται επιστήμονες διαφόρων κλάδων που προέρχονται από χώρους όχι μόνο των λεγόμενων θετικών επιστημών αλλά και



παιδαγωγοί, κοινωνιολόγοι και φιλόσοφοι. Αυτό έχει ως αποτέλεσμα ο καθένας να δίνει έναν ορισμό από τη σκοπιά του που ρίχνει φως σε μια ακόμη όψη της πληροφορικής.

Ο πίνακας του σχήματος 1.1 δείχνει συνοπτικά τους ορισμούς της πληροφορικής που προέρχονται από διαφορετικούς χώρους, πράγμα που δηλώνει το πόσο ζωντανή και εξελισσόμενη είναι αφού ο κατάλογος συμπληρώνεται συνεχώς.

Αναζητώντας ένα γενικό ορισμό της πληροφορικής με την ανάγκη για συνεχή επαναπροσδιορισμό του κλάδου, αναφέρεται αυτός του Caruipio ότι “πληροφορική είναι η ερμηνευτική επιστήμη που αποστολή της είναι η τεχνική υποστήριξη της αλληλεπίδρασης του ανθρώπου με τον κόσμο” που φανερώνει ότι τελικά αντικείμενο της είναι ο άνθρωπος [Rechenberg, 1992]. Και αφού η κατάληξη είναι ότι η πληροφορική είναι επιστήμη, που κατατάσσεται; Παρότι έχει κοινά σημεία με τις φυσικές και τις ανθρωπιστικές επιστήμες, οι διαφορές μ’ αυτές είναι μεγάλες και φαίνεται να προσεγγίζει περισσότερο τα μαθηματικά. Έτσι όπως αυτά, θεωρείται ότι ανήκει στις δομικές επιστήμες.

1.2 Ιστορική αναδρομή

Δύο είναι οι κύριοι λόγοι για την ανάγκη μιας ιστορικής αναδρομής της ταχύτατα ακόμη και σήμερα εξελισσόμενης πληροφορικής, που όπως φάνηκε δεν είναι μόνο απόρροια της αλματώδους τεχνολογικής ανάπτυξης.

Ο πρώτος έχει να κάνει με τις πολιτισμικές κυρίως αλλαγές του ανθρώπου στον εικοστό αιώνα που είχαν ως συνέπεια την εφεύρεση νέων μηχανών. Ο δεύτερος λόγος είναι ότι η πληροφορική προσεγγίζει όλο και περισσότερους ανθρώπους που δεν έχουν τεχνικά μόνο ενδιαφέροντα, δείχνοντας ότι αρχίζει να πλέκεται μια διαλεκτική σχέση μεταξύ της τεχνολογίας και του πολιτισμού.

Οι πίνακες των σχημάτων 1.2 και 1.3 παρουσιάζουν διαγραμματικά τα πιο σημαντικά στάδια στην πορεία εξέλιξης της πληροφορικής, αφήνοντας έτσι χώρο για μια κατάδυση στην ιστορία της που προσπαθεί να τη συνδέσει με τα ιστορικά και κοινωνικά γε-



γονότα δείχνοντας τις μεταξύ τους αλληλεπιδράσεις. Για μια όσο γίνεται ολοκληρωμένη παρουσίαση του πολυσύνθετου κλάδου της πληροφορικής, γίνεται η προσέγγισή της από τη σκοπιά του αυτοματισμού, των υπολογισμών και των υπολογιστών.

ΧΡΟΝΟΣ	ΓΕΓΟΝΟΤΑ
προϊστορία	
5ος αι. π.Χ.	Εμφάνιση δεκαδικού συστήματος στην Ινδία.
3ος αι. π.Χ.	Αλγόριθμος του Ευκλείδη.
1ος αι. π.Χ.	Αστρολάβος των Αντικυθήρων. Αστρονομικοί υπολογισμοί.
1623 μ.Χ.	Ο W. Schickard κατασκευάζει μηχανή με γρανάζια που εκτελεί τις τέσσερις βασικές πράξεις της αριθμητικής.
1641	Ο μαθηματικός Blaise Pascal κατασκευάζει μηχανή με γρανάζια που προσθέτει εξαψήφιους αριθμούς.
1674	Ο φιλόσοφος και μαθηματικός Leibnitz κατασκευάζει υπολογιστική μηχανή με οδοντωτούς τροχούς για τις τέσσερις πράξεις και ασχολείται με το δυαδικό σύστημα αρίθμησης.
1805	Ο μηχανικός Joseph-Marie Jacquard κατασκευάζει έναν αυτοματοποιημένο αργαλειό. Δυαδική λογική, προγραμματισμός με διάτρητες ξύλινες πλάκες.
1822	Ο μαθηματικός Charles Babbage σχεδιάζει την αναλυτική μηχανή.
1886	Ο Hermann Hollerith εφαρμόζει τις διάτρητες κάρτες. Πρώτες συσκευές αποθήκευσης δεδομένων.
ιστορία	
1934	Ο μηχανικός Konrad Zuse σχεδιάζει μια υπολογιστική μηχανή με αποθηκευμένο πρόγραμμα βασισμένη σε δυαδική λογική.
1944	Ο μαθηματικός H.Aiken κατασκευάζει τον MARK I βασισμένο στο δεκαδικό σύστημα που προγραμματίζεται και διαθέτει μνήμη.
1946	Οι J.Eckert και J.Mauchly κατασκευάζουν τον πρώτο πλήρως ηλεκτρονικό υπολογιστή, τον ENIAC.
1946	Ο John von Neumann προτείνει το αποθηκευμένο πρόγραμμα.
1949	Ο M.Wilkes κατασκευάζει τον EDSAC, τον πρώτο ψηφιακό υπολογιστή γενικής χρήσης με αποθηκευμένο πρόγραμμα.
1951	Οι J.Eckert και J.Mauchly παρουσιάζουν τον ηλεκτρονικό υπολογιστή UNIVAC 1 που είναι βασισμένος στο δεκαδικό σύστημα. Αρχίζει η βιομηχανική παραγωγή των Η/Υ.

Σχήμα 1.2 Χρονολογική εξέλιξη της πληροφορικής



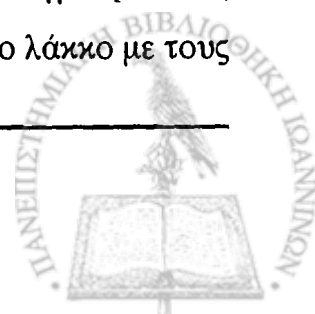
ΓΕΝΙΑ	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ
1η 1950 - 1959	Κυκλώματα με καθοδικές λυχνίες. Κεντρική μνήμη μερικών εκατοντάδων λέξεων μηχανής. Ταχύτητα 50000 - 200000 πράξεις το δευτερόλεπτο.
2η 1959 - 1963	Κυκλώματα με τρανζίστορ. Κεντρική μνήμη λίγων χιλιάδων λέξεων μηχανής. Ταχύτητα 200000 - 1000000 πράξεις το δευτερόλεπτο. Δευτερεύουσα μνήμη, λειτουργικά συστήματα, γλώσσες προγραμματισμού (Fortran, Cobol). Πολυεπεξεργασία.
3η 1963 - 1970	Ολοκληρωμένα κυκλώματα. Χρήση οθονών, γραφικά. Ταχύτητα 1000000 - 4000000 πράξεις το δευτερόλεπτο. Λειτουργία κατανεμημένου χρόνου.
4η 1971-σήμερα	Ολοκληρωμένα κυκλώματα μεγάλης κλίμακας (LSI). Μικροεπεξεργαστές, προσωπικοί υπολογιστές. Δίκτυα.
5η 1985-σήμερα	Ολοκληρωμένα κυκλώματα πολύ μεγάλης κλίμακας (VLSI). Νευρωνικά δίκτυα, τεχνητή νοημοσύνη, internet, εικονική πραγματικότητα, ποιός ξέρει;

Σχήμα 1.3 Γενιές υπολογιστών. Η ασύλληπτη ταχύτητα των εξελίξεων οδήγησε στην κατηγοροποίηση σε γενιές

1.2.1 Υπολογιστικές μηχανές

Κύριο μέλημα του ανθρώπου έκδηλο στην αρχαία Ελλάδα ήταν η μίμηση της φύσης, η κατασκευή μηχανικών πλασμάτων και αυτόματων μηχανών με στόχο τη χαλιναγωγή του χρόνου και της κίνησης. Από την ιστορία του αυτοματισμού αρχίζει να φαίνεται η άμεση σύνδεσή του μέσα από το βασικό γνώρισμά του που είναι η ρύθμιση, με την πληροφορική και ιδιαίτερα με τις αρχές του προγραμματισμού. Και τα δύο χαρακτηριστικά έχουν να κάνουν με μια πεπερασμένη ακολουθία διαδικασιών που αναπαράγεται από μια διάταξη.

Το πρώτο πληροφορικό σύστημα είναι η παγίδα του πρωτόγονου κυνηγού [Breton, 1991]. Το υλικό μέρος του συστήματος αποτελείται από το σκεπασμένο λάκκο με τους



μυτερούς πασσάλους και τα δεδομένα του κυνηγού - χρήστη είναι η γνώση του για το είδος του θηράματος και η διαδρομή που αυτό ακολουθεί. Η επεξεργασία αυτών των δεδομένων με τη βοήθεια της παγίδας κάνει περιττή την παρουσία του κυνηγού και αυτοματοποιεί τη σύλληψη του ζώου που αποτελεί το αποτέλεσμα της επεξεργασίας.

Το πρώτο βοήθημα του ανθρώπου για υπολογισμούς ήταν τα χέρια. Χρησιμοποιήθηκαν όλα τα μέρη του χεριού και όχι μόνο τα δάκτυλα, με την κωδικοποίηση των κινήσεών τους με τρόπο παρόμοιο με τη γλώσσα των κωφαλάλων. Το αποτέλεσμα δεν είναι καθόλου ευκαταφρόνητο, αφού παρέχεται η δυνατότητα για υπολογισμούς με αριθμούς που φτάνουν τις χιλιάδες.

Δύο μεγάλοι σταθμοί στη μηχανοποίηση των υπολογισμών αποτελούν ο άβακας και το αριθμητήριο. Ο άβακας επινοήθηκε πιθανώς πριν από το 4000 π.Χ. και αποτελείται από μια ξύλινη πλάκα που πάνω της χαράζονται αυλάκια σε παράλληλη διάταξη. Το δεξιότερο αυλάκι αναπαριστάνει τις μονάδες, το επόμενο του προς τα αριστερά τις δεκάδες, κ.ο.κ. Στα αυλάκια τοποθετείται ένας ορισμένος αριθμός από βότσαλα και ο συνδυασμός τους δίνει το αποτέλεσμα της πράξης. Η λατινική ονομασία για τα βότσαλα είναι "calculi" που θυμίζει υπολογισμούς και αριθμομηχανές.

Το αριθμητήριο χρησιμοποιείται ακόμη και σήμερα στην Ασία και ακολουθεί την ίδια λογική. Τα αυλάκια έχουν αντικατασταθεί από σύρματα και τα βότσαλα από χάντρες. Μ' αυτήν την υπολογιστική μηχανή γίνονται προσθέσεις, αφαιρέσεις, πολλαπλασιασμοί, διαιρέσεις και υπολογισμοί τετραγωνικών ριζών.

Την ίδια φιλοσοφία ακολουθούν οι υπολογιστικές μηχανές αλλά και οι ηλεκτρονικοί υπολογιστές που βασίζονται στο δεκαδικό αριθμητικό σύστημα. Τα μόνα που αλλάζουν είναι τα τεχνικά μέσα που χρησιμοποιούνται για την υλοποίηση των μηχανών.

Το 1623 ο Γερμανός Βίλχελμ Σίκαρντ κατασκευάζει την πρώτη υπολογιστική μηχανή που λειτουργεί με οδοντωτούς τροχούς και εκτελεί μηχανικά προσθέσεις και αφαιρέσεις. Έχει ακόμη τη δυνατότητα της απομνημόνευσης ενδιάμεσων υπολογισμών και ένα κουδούνι που σημαίνει όταν η υπολογιζόμενη ποσότητα υπερβαίνει τη χωρητικότητα της "μνήμης" της συσκευής. Αυτή συμπληρώνεται με τη διάταξη του Σκοτσέξου Τζον Νέπερ που έχει τη δυνατότητα να πολλαπλασιάζει.



Όμως τη δόξα του εφευρέτη της πρώτης υπολογιστικής μηχανής την πήρε ο Μπλαιζ Πασκάλ που το 1643 σχεδίασε και κατασκεύασε την “pascaline”, σε ηλικία δεκαεννέα ετών. Η μηχανή του Πασκάλ είχε δυνατότητες πρόσθεσης και αφαιρέσης. Μερικά χρόνια αργότερα ο Γερμανός φιλόσοφος και Μαθηματικός Βίλχελμ φον Λάιμπνιτς κατασκεύασε τη δική του μηχανή βελτιώνοντας αυτήν του Πασκάλ, προσθέτοντας τη δυνατότητα για πολλαπλασιασμούς. Το 1805 ο μηχανικός Ζακάρ αυτοματοποίησε έναν αργαλειό προγραμματίζοντάς τον χρησιμοποιώντας διάτρητα ξύλινα πλακίδια που η θέση των τρυπών καθόριζε και το σχέδιο του πλεκτού. Αυτού του είδους ο προγραμματισμός χρησιμοποιεί τη δυαδική κωδικοποίηση με την έννοια της ύπαρξης ή όχι τρύπας και έκανε αθάνατο το όνομα του Ζακάρ που αναφέρεται σε κάθε συζήτηση σχετική με πλεκτά με σχέδια.

Παρακινούμενος από διαφορετικά κίνητρα ο Άγγλος φιλόσοφος Φράνσις Μπέικον εφευρίσκει το δυαδικό τρόπο κωδικοποίησης στις αρχές του 17ου αιώνα. Είχε δει την αξία των δύο μόνο καταστάσεων σε έναν κώδικα κατά την κρυπτογράφηση διπλωματικών μηνυμάτων. Επιδίωξή του ήταν η μεταβίβαση της σκέψης σε απόσταση με δυαδική κωδικοποίηση και τεχνικά ηχητικά και οπτικά μέσα της εποχής. Αρχίζει λοιπόν να γίνεται φανερή η προσέγγιση της πληροφορικής που ακόμη δεν έχει κάνει την εμφάνισή της, από τελείως διαφορετικές σκοπιές.

Η προαιώνια ανάγκη του ανθρώπου να κατασκευάσει έμψυχα αγάλματα και μηχανικά όντα δείχνει ίσως τη ματαιοδοξία του να εξισωθεί με τους θεούς αλλά και να προστατευθεί από κινδύνους με την αξιοποίηση τέτοιων δημιουργημάτων. Εδώ μπορεί να αναζητηθεί και η βαθιά ανάγκη της ανάπτυξης των σύγχρονων υπολογιστών, που η εξέλιξή τους είχε αφορμή τις στρατιωτικές απαιτήσεις κατά το δεύτερο παγκόσμιο πόλεμο και τις μετέπειτα οικονομικές εφαρμογές. Αυτό φαίνεται και από την αρχική ονομασία των υπολογιστών, “ηλεκτρονικοί εγκέφαλοι”.

Τα προγραμματιζόμενα όντα με δυνατότητα για αυτόματους συλλογισμούς, φαίνονται και στις ιδέες του 17ου και 18ου αιώνα των υλιστών φιλοσόφων του Διαφωτισμού οι οποίοι περιγράφουν τον άνθρωπο ως ένα αυτόματο. Ο Καρτέσιος και ο Τόμας Χομπς αναφέρουν ότι η τυπική γλώσσα γίνεται ένα μέσο που χωρίς την επέμβαση του ανθρώ-



που μπορεί από μόνο του να παράγει συλλογισμούς. Έχει αρχίσει να γίνεται φανερή η επιθυμία του ανθρώπου για τη δημιουργία της τεχνητής νοημοσύνης.

Ως σημαντικότερος πρόδρομος της πληροφορικής θεωρείται ο Άγγλος καθηγητής των μαθηματικών Τσαρλς Μπάμπατζ ο οποίος το 1822 σχεδίασε δυο πολύπλοκες υπολογιστικές μηχανές, τη μηχανή των διαφορών και την αναλυτική μηχανή. Ο χρόνος και ο τόπος της εμφάνισής τους δεν είναι τυχαίος αφού έχει αρχίσει η εκβιομηχάνιση της παραγωγής και η Αγγλία είναι μια ισχυρή ναυτική δύναμη με απαιτήσεις για ακριβείς αριθμητικούς υπολογισμούς στη ναυσιπλοΐα.

Η επαναστατική ιδέα ήταν η καθοδήγηση των λειτουργιών της αναλυτικής μηχανής από διάφορα αποθηκευμένα προγράμματα με τη μορφή διάτρητων καρτών όμοιων με αυτών του Ζακάρ και εξαιτίας αυτού η συσκευή αναφέρεται ως ο πρόγονος των σύγχρονων υπολογιστών. Η μαθηματικός Αυγούστα Άντα Μπάιρον, κόρη του λόρδου Βύρωνα, χειρίστηκε τη μηχανή του Μπάμπατζ και το όνομα της έμεινε αθάνατο στην ονομασία της γλώσσας προγραμματισμού ADA.

Η υλοποίηση και των μηχανών του Μπάμπατζ προσέκρουσε σε τεχνικές αδυναμίες της εποχής και μόνο μερικά τμήματά τους κατασκευάστηκαν. Παρ' όλα αυτά, τα σχέδια τους αποτελούν ένα από τα μεγαλύτερα τεχνολογικά επιτεύγματα του 19ου αιώνα. Η χρήση των διάτρητων καρτών που εισήγαγε, επεκτάθηκε στις επόμενες μηχανές που βρήκαν τη θέση τους σε γραφεία και εργοστάσια.

Αυτήν την περίοδο ο Άγγλος μαθηματικός Τζωρτζ Μπουλ δημιουργεί την άλγεβρά του που βασίζεται σε νόμους της λογικής και δύο ειδών προτάσεις, τις αληθείς και τις ψευδείς, εμπνεόμενος ίσως από το συμπατριώτη του Μπέικον ως προς τη δυαδική λογική. Σκοπός του Μπουλ δεν ήταν η χρήση της λογικής του από τις υπολογιστικές μηχανές, αλλά η προσπάθειά του για καλύτερη κατανόηση των μηχανισμών της φυσικής γλώσσας για την απόδειξη της αλήθειας, εργασία που είναι συνέχεια των φιλοσοφικών αναζητήσεων στην αρχαία Ελλάδα.

Στα τέλη του 19ου αιώνα σημειώθηκε μια έκρηξη ιδεών σχετικά με τις υπολογιστικές μηχανές με στόχο την κατασκευή εύχρηστων επιτραπέζιων μηχανών απαλλαγμένων από πολύπλοκους χειρισμούς.



Ο Αμερικανός Ντορ Φελτ εισήγαγε το πληκτρολόγιο και η είσοδος των αριθμών άρχισε να γίνεται με πλήκτρα σε αντίθεση με τις προηγούμενες μηχανές στις οποίες έπρεπε να στραφούν γρανάζια σε συγκεκριμένες θέσεις. Αυτήν την περίοδο, γίνεται το πέρασμα από τις υπολογιστικές μηχανές αριθμών στις μηχανές επεξεργασίας δεδομένων. Ο κύριος λόγος είναι οι στατιστικές απαιτήσεις της εποχής που δεν αρκούνται μόνο σε αριθμητικούς υπολογισμούς.

Την αρχή την έκανε ο Αμερικανός Χέρμαν Χόλλεριθ που εργαζόταν στη στατιστική υπηρεσία των ΗΠΑ και αντιμετώπισε έναν τεράστιο όγκο δεδομένων για επεξεργασία κατά τη διάρκεια της απογραφής του πληθυσμού. Το 1887 παρουσίασε μια ηλεκτρική υπολογιστική μηχανή τη μηχανή πινάκων, που χρησιμοποιούσε διάτρητες κάρτες. Στην απογραφή του 1890 σαράντα τρεις τέτοιες μηχανές τέθηκαν σε λειτουργία, δούλεψαν ακούραστα επί δέκα εβδομάδες διαβάζοντας 50 με 80 κάρτες απογραφής το λεπτό και κάνοντας υπολογισμούς. Στις 10 Αυγούστου η υπηρεσία απογραφής ανακοίνωσε τον πληθυσμό της χώρας, 62.622.650 κάτοικοι. Μέχρι εκείνη την ημερομηνία δεν είχε υπολογισθεί ο πληθυσμός που προέκυπτε από τα στοιχεία της απογραφής της προηγούμενης δεκαετίας. Το 1896 ο Χόλλεριθ ιδρύει μια εταιρία κατασκευής υπολογιστικών μηχανών, η οποία το 1924 μετατρέπεται στην IBM.

Οι ανάγκες για υπολογισμούς συνεχώς αυξάνονταν όχι μόνο στους χώρους των βιομηχανιών και των επιχειρήσεων, αλλά και στο χώρο των θετικών επιστημών με σημαντικό παράδειγμα τα πρώτα τριάντα χρόνια του 20ου αιώνα που θεωρούνται τα χρόνια που συγκλόνισαν τη φυσική, με την κβαντική θεωρία που άλλαξε την αντίληψη μας για τη δομή της ύλης [Gapow, 1971]. Έτσι δημιουργήθηκε η ανάγκη για μια μηχανή που να δίνει αποτελέσματα χωρίς να κάνει υπολογισμούς, να μπορεί δηλαδή να λύνει διαφορικές εξισώσεις οι οποίες προβλέπουν τη συμπεριφορά ενός κινούμενου αντικειμένου. Έφθασε πλέον το πλήρωμα του χρόνου για την ανάπτυξη των επιστημονικών υπολογιστικών μηχανών. Δύο ακόμη ισχυροί λόγοι για τη σχεδίαση νέων μηχανών είναι οι μεγάλες λογιστικές ανάγκες των στρατιωτικών υπηρεσιών και ο ανταγωνισμός των διαφόρων κατασκευαστών για την κυριαρχία τους στη νέα αγορά.

Σημαντικό βήμα στην εξέλιξη της σχεδίασης των μετέπειτα υπολογιστών έκανε ο Γερ-



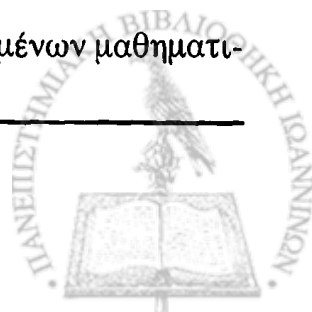
μανός μηχανικός Κόνραντ Ζυζ που εισήγαγε το δυαδικό σύστημα αρίθμησης στις υπολογιστικές μηχανές. Μέχρι τότε η αναπαράσταση των αριθμών γινόταν στο αυτονόητο για τη λογική του ανθρώπου δεκαδικό αριθμητικό σύστημα. Ο Ζυζ εφάρμοσε τη δυαδική κωδικοποίηση με την απλότητα των δύο μόνο ψηφίων της (0 και 1) για την αναπαράσταση κάθε αριθμού, με αντίτιμο τη χρήση περισσότερων ψηφίων για κάθε αριθμό από ότι στο δεκαδικό σύστημα με τον πλούτο των δέκα ψηφίων του.

Από το 1936 μέχρι το 1938 ο Ζυζ χωρίς να έχει ιδέα για τις εργασίες των ερευνητικών ομάδων στην άλλη πλευρά του Ατλαντικού με μοναδικό κίνητρο την επιθυμία του για αυτοματοποίηση των υπολογισμών, κατασκεύασε δύο πρωτότυπα της μηχανής του. Το μηχανικό Z1 και το Z2 που χρησιμοποιούσε ηλεκτρομηχανικούς διακόπτες τους γνωστούς τηλεφωνικούς ηλεκτρονόμους (ρελέ) που άλλαζαν θέση σε κλάσματα του δευτερολέπτου. Οι δύο θέσεις επαφής του ηλεκτρονόμου αντιστοιχούν στα δυαδικά ψηφία 0 και 1 με ταχύτητα εναλλαγής της τάξης του δεκάτου του δευτερολέπτου. Έτσι υλοποιείται η δυαδική κωδικοποίηση με ηλεκτρομηχανικό τρόπο και ανοίγει το κεφάλαιο των ηλεκτρονικών υπολογιστών.

Πλησιάζει ο δεύτερος παγκόσμιος πόλεμος και κάθε είδους τεχνολογική έρευνα έχει ως άμεσο αποδέκτη τις στρατιωτικές εφαρμογές αλλά και οι στρατιωτικές ανάγκες δίνουν ώθηση στην ταχεία πρόοδο της επιστημονικής έρευνας. Όλοι σχεδόν οι επιστήμονες στην Αμερική, Αγγλία και Γερμανία έχουν επιστρατευθεί και δουλεύουν σε πανεπιστήμια και ερευνητικά κέντρα του στρατού.

Το 1938 στην Αγγλία ο ιδιόρρυθμος μαθηματικός και μαθητής του Αϊνστάϊν Άλαν Τιούρινγκ καλείται από τις μυστικές υπηρεσίες της χώρας του να αναλάβει την αποκρυπτογράφηση των μηνυμάτων του εχθρού, που διακινούνταν κωδικοποιημένα με τη βοήθεια της μηχανής "αίνιγμα" που είχε τη δυνατότητα παραγωγής 22 δισεκατομμυρίων συνδυασμών. Ο Τιούρινγκ αναβάλλει τα σχέδια του για την κατασκευή μιας παγκόσμιας μηχανής με τον φον Νόμαν και το 1940 κατασκευάζει τη "βόμβα", το αντίδοτο στην "αίνιγμα". Είναι ένας υπολογιστής εξειδικευμένος στην αποκωδικοποίηση που όπως λέγεται συνέβαλε στην αρχή του τέλους του πολέμου.

Το 1937 στην Αμερική, ο Χάουαρντ Άικεν καθηγητής των εφαρμοσμένων μαθηματι-



κών στο πανεπιστήμιο Χάρβαρντ, σχεδιάζει έναν αριθμητικό υπολογιστή λαμβάνοντας υπόψη του την ξεχασμένη δουλειά του Μπάμπατζ. Τον Ιανουάριο του 1944 παρουσιάζεται ο MARK 1, ένας υπολογιστής γενικής χρήσης με βάση το δεκαδικό σύστημα αρίθμησης. Με διαστάσεις όσο ένα μεγάλο δωμάτιο, εκτελούσε έναν πολλαπλασιασμό δυο δεκαδικών αριθμών των 23 ψηφίων σε τρία δευτερόλεπτα.

Στο μεταξύ είναι η σειρά της ηλεκτρονικής να μπει στο παιχνίδι. Το δελεαστικό στοιχείο που προσφέρει είναι οι ηλεκτρονικές λυχνίες που δεν περιλαμβάνουν μηχανικά κινούμενα μέρη και μπορούν να χρησιμοποιηθούν ως ταχύτατοι διακόπτες. Ταχύτητα μεταγωγής και επομένως διάρκειας κατάστασης ενός κυκλώματος σε ανοιχτό ή κλειστό, ένα δέκατο του χιλιοστού του δευτερολέπτου, χίλιες φορές γρηγορότερα από τους ηλεκτρονόμους. Όπως φαίνεται, έχει αρχίσει ο πόλεμος της ταχύτητας εκτέλεσης πράξεων μεταξύ των διάφορων επιστημονικών ομάδων και εταιριών.

Τις ηλεκτρονικές λυχνίες αλλά και τις ιδέες του καθηγητή μαθηματικών Τζον Ατάναςοφ εκμεταλλεύτηκαν οι Έκερτ και Μόσλυ στο πανεπιστήμιο της Πενσυλβάνια και στις 15 Φεβρουαρίου 1946 παρουσίασαν τον προγραμματιζόμενο και γενικής χρήσης ENIAC. Σημαντικά στοιχεία του αποτελούν η δυαδική λογική και η ύπαρξη ενός ηλεκτρονικού ρολογιού για το συγχρονισμό των εσωτερικών του λειτουργιών. Με έκταση όση ενός μεγάλου διαμερίσματος, κύριος σκοπός του ήταν ο υπολογισμός τροχιών βλήματων, πράγμα που το έκανε επιτυχώς βρίσκοντας μέσα σε 20 δευτερόλεπτα την τροχιά ενός βλήματος που χρειάζονταν 30 δευτερόλεπτα να φτάσει στο στόχο του και τριών ημερών ανθρώπινους υπολογισμούς.

Ακολουθώντας την κρυφή επιθυμία του ανθρώπου για κατασκευή ενός τεχνητού εγκεφάλου, αρχίζουν οι συγκρίσεις των επιτευγμάτων κάθε περιόδου με τον ανθρώπινο εγκέφαλο με τις 10 δισεκατομμύρια περίπου λειτουργικές μονάδες του, τους νευρώνες. Για την κατασκευή μιας μηχανής στα τέλη της δεκαετίας του 1940 με τον ίδιο αριθμό στοιχείων, απαιτείται μέγεθος ίσο με μια πόλη σαν το Παρίσι και απαίτηση σε ενέργεια ισοδύναμη με αυτήν του μετρό του.



1.2.2 Ηλεκτρονικοί υπολογιστές

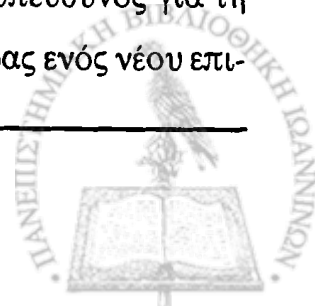
Ο ENIAC είναι μια προγραμματιζόμενη μηχανή και αποτελεί στρατιωτικό μυστικό. Στόχος και αυτής παραμένει ο ταχύς και ακριβής υπολογισμός τροχιών βλημάτων. Ο προγραμματισμός για την υλοποίηση μιας νέας εργασίας της απαιτεί δουλειά τριάντα μηνών, εικοσιτέσσερις ώρες το εικοσιτετράωρο και για μικρές αλλαγές στο πρόγραμμα, τεχνικοί - προγραμματιστές αλλάζουν τις θέσεις των καλωδιώσεων σε συγκεκριμένα τμήματα του υπολογιστή.

Ένα από τα σπουδαιότερα άλματα της πληροφορικής που υπερπήδησε το παραπάνω πρόβλημα και καθόρισε τις βασικές αρχές της αρχιτεκτονικής των υπολογιστών μέχρι σήμερα, είναι η ιδέα του Ουγγροαμερικανού μαθηματικού Τζον φον Νόουμαν να αποθηκεύσει το πρόγραμμα στη μνήμη του υπολογιστή.

Εδώ φαίνεται και η σύζευξη της πληροφορικής με τον αυτοματισμό, αφού κύριος στόχος είναι πλέον η μηχανή να απαλλάξει τον άνθρωπο και να αναλάβει η ίδια τον έλεγχο διαδοχής των υπολογισμών. Ο συντονισμός όλων των λειτουργιών μεταξύ των κατάλληλα σχεδιασμένων και οργανωμένων τμημάτων του υπολογιστή γίνεται από ένα κομμάτι του. Ο υπολογιστής περιέχει στη μνήμη του τα πεπερασμένα βήματα που πρέπει να ακολουθήσει για τη διεκπεραίωση διαφόρων διαδικασιών, δηλαδή τους αλγόριθμους.

Ο Νόουμαν συνέλαβε αυτήν την επαναστατική ιδέα ακολουθώντας τη γενικότερη φιλοσοφία της εποχής, την προσπάθεια για την κατασκευή ενός μοντέλου του ανθρώπινου εγκεφάλου. Έτσι η δυαδική λογική που ακολουθεί βασίζεται στην αντίληψη ότι ο ανθρώπινος εγκέφαλος λειτουργεί με δυαδικό τρόπο με την έννοια των ανταλλαγών ηλεκτρικών σημάτων μεταξύ των νευρώνων, των νευρικών κυττάρων.

Οι αλλαγές που έφερε ο ήδη διάσημος φον Νόουμαν στη σχεδίαση των υπολογιστών ήταν η μονάδα εσωτερικού ελέγχου και το αποθηκευμένο πρόγραμμα. Αυτές υλοποιήθηκαν το 1949 στο πανεπιστήμιο Κέιμπριτζ της Αγγλίας με τη μηχανή EDSAC και ανοίγουν την εποχή των υπολογιστών τύπου von Neumann. Αυτός είναι υπεύθυνος για τη διαμόρφωση της κουλτούρας της πληροφορικής και θεωρείται ο πατέρας ενός νέου επι-



στημονικού κλάδου, της κυβερνητικής, που ασχολείται με τον έλεγχο και την επικοινωνία στα ζώα και τις μηχανές [Wiener, 1961].

Έχει ήδη φθάσει η δεκαετία του '50 και οι κατασκευαστές υπολογιστών δεν έχουν να ανησυχούν για τις βασικές αρχές στην αρχιτεκτονική των συστημάτων τους. Το 1951 οι Έκερτ και Μόσλυ παρουσιάζουν τον UNIVAC 1 (UNIVersal Automatic Computer, καθολικός αυτόματος υπολογιστής) βασισμένο στο δεκαδικό σύστημα, με εξωτερική μνήμη μαγνητικών ταινιών και εκτυπωτές, προσανατολισμένο σε πολιτικές εφαρμογές. Ο τελευταίος λόγος ήταν υπεύθυνος για τις ενοχλήσεις που είχαν από την επιτροπή αντιαμερικανικών υποθέσεων που ερευνούσε τις ενέργειες ενάντια στην ασφάλεια του κράτους.

Η τεχνολογική πρόοδος αναπτύσσεται πλέον ραγδαία και δίνει ιδέες και προϊόντα στην πληροφορική που εξελίσσεται και αυτή με παρόμοιο ρυθμό και αρχίζει να γίνεται λόγος για τις γενιές των υπολογιστών. Η ανάπτυξη του λογισμικού ακολουθεί το υλικό χωρίς να το φτάσει αφού το σημαντικότερο χαρακτηριστικό του πρώτου είναι η ανθρώπινη νοημοσύνη και τα όριά της. Εδώ αρχίζουν να φαίνονται τα όρια της ανθρώπινης νόησης. Αυτό μάλλον είναι περιστασιακό πρόβλημα αφού η μέχρι τώρα εξέλιξη της πληροφορικής έχει διδάξει πολλά στον άνθρωπο και έχει επηρεάσει τον τρόπο σκέψης του [Haugeland, 1992]. Αποτελεί άραγε επιστημονική φαντασία η σχεδίαση και κατασκευή ενός ηλεκτρονικού, οπτικού, ή βιολογικού υπολογιστή με ικανότητες σχεδίασης και κατασκευής καλύτερων από αυτόν υπολογιστών;

Αναφορές

Breton P., "Ιστορία της Πληροφορικής", Δίαυλος, Αθήνα 1991

Campbell J. "Άνθρωπος & Πληροφορικά Συστήματα", Χατζηνικολής, Αθήνα 1985

Gamow G., "30 χρόνια που συγκλόνισαν τη Φυσική", Σελλούντος, Αθήνα 1971

Haugeland J., "Τεχνητή Νοημοσύνη", Κάτοπτρο, Αθήνα 1992

Rechenberg P. "Εισαγωγή στην Πληροφορική", Κλειδάριθμος, Αθήνα 1992

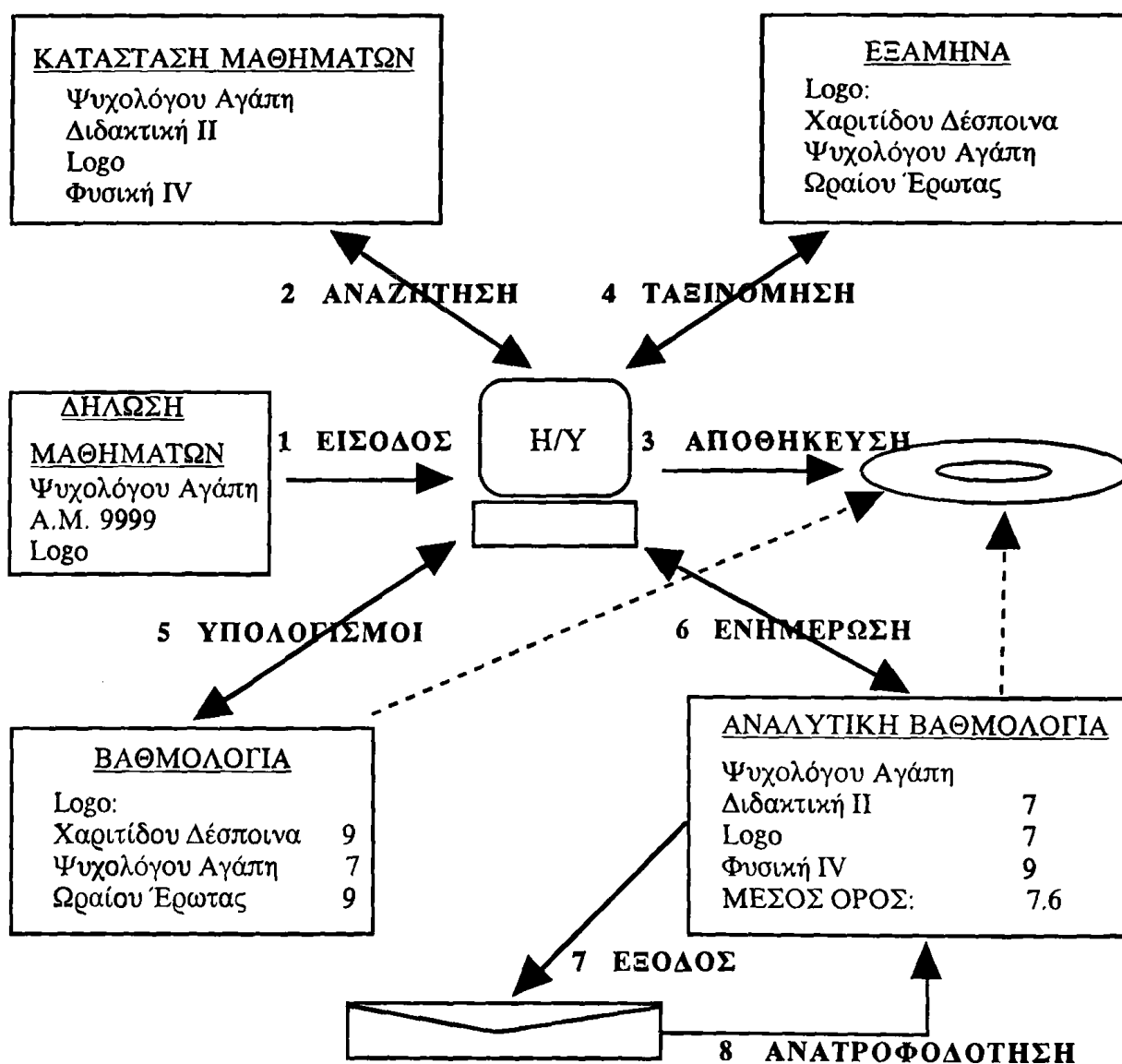
Wiener N., "Κυβερνητική ή έλεγχος και επικοινωνία στα ζώα και στις μηχανές",

Καστανιώτης 1961



2 Επεξεργασία δεδομένων

Ετυμολογικά η λέξη δεδομένα (data) είναι ο πληθυντικός αριθμός της λατινικής λέξης “datum” που σημαίνει γεγονός. Μ’ αυτήν τη μορφή συναντάται σε πολλές γλώσσες και όταν πρέπει αναγκαστικά να χρησιμοποιηθεί στον ενικό αριθμό, αναφέρεται ως στοιχείο δεδομένων.



Σχήμα 2.1 Μοντέλο επεξεργασίας δεδομένων



Τα δεδομένα είναι στοιχεία σε πρωτογενή μορφή που χρησιμοποιούνται για την αναπαράσταση καταστάσεων. Ο ορισμός αυτός βρίσκεται ένα βήμα πίσω από τον ορισμό της προδιαγραφής ISO 2382 για την πληροφορία και δηλώνει την ύπαρξη ενός ενδιάμεσου μεταξύ τους σταδίου, την επεξεργασία των δεδομένων (data processing).

Στο σχήμα 2.1 φαίνεται το μοντέλο επεξεργασίας των δεδομένων με ορισμένα από τα ενδιάμεσα στάδιά του, μέσα από ένα παράδειγμα βαθμολογίας και αρχειοθέτησης των στοιχείων φοιτητή.

Σε πολλές περιπτώσεις, οι έννοιες των δεδομένων και της πληροφορίας συγχέονται και είναι συχνό φαινόμενο ο όρος “επεξεργασία πληροφοριών”. Είναι βέβαια φανερό ότι πρόκειται για επεξεργασία δεδομένων με αποτέλεσμα την πληροφορία. Η κατανόηση της διαφοράς των δύο εννοιών ξεκαθαρίζει τα πράγματα παρά τις αλλαγές της ονοματολογίας και δεν αφήνει περιθώρια για σύγχυση.

2.1 Κωδικοποίηση δεδομένων

Τα δεδομένα αποτελούνται από χαρακτήρες που συνήθως ονομάζονται σύμβολα και μπορεί να είναι αριθμοί, γράμματα, ειδικά σύμβολα ή συνδυασμός τους. Ένα πεπερασμένο σύνολο διακριτών συμβόλων απαρτίζει το αλφάβητο. Ο αριθμός και το είδος των συμβόλων ενός αλφαβήτου, καθορίζει το συγκεκριμένο αλφάβητο. Στο σχήμα 2.2 παρουσιάζονται ορισμένα αλφάβητα. Ο κανόνας που ορίζει την αντιστοίχιση των χαρακτήρων ενός αλφαβήτου προς τους χαρακτήρες ενός άλλου αλφαβήτου, λέγεται κώδικας.

ΑΛΦΑΒΗΤΟ	ΣΥΝΟΛΟ ΧΑΡΑΚΤΗΡΩΝ
ΔΕΚΑΔΙΚΟ ΣΥΣΤΗΜΑ	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
ΔΕΚΑΕΞΑΔΙΚΟ ΣΥΣΤΗΜΑ	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
ΕΛΛΗΝΙΚΟ ΑΛΦΑΒΗΤΟ	{Α, Β, Γ, ..., Ω, α, β, γ, ..., ω}
ΛΑΤΙΝΙΚΟ ΑΛΦΑΒΗΤΟ	{Α, Β, C, ..., Z, a, b, c, ..., z}
ΕΠΟΧΕΣ ΤΟΥ ΕΤΟΥΣ	{Άνοιξη, Καλοκαίρι, Φθινόπωρο, Χειμώνας}

Σχήμα 2.2 Παραδείγματα αλφαβήτων



Στην πληροφορική και ιδιαίτερα στην επιστήμη των ηλεκτρονικών υπολογιστών, το αλφάβητο που χρησιμοποιείται είναι αυτό που αποτελείται από δύο μόνο σύμβολα και λέγεται δυαδικό. Τα σύμβολά του είναι τα 0 και 1, όχι όμως αυτά του δεκαδικού αριθμητικού συστήματος, και συνήθως καλούνται διεθνώς “bit” από την Αγγλική ορολογία “binary digit” που σημαίνει δυαδικό ψηφίο.

Η δυαδική κωδικοποίηση είναι επακόλουθο της μαθηματικής λογικής, των αναγκών κρυπτογράφησης και της ανάπτυξης του βιομηχανικού αυτοματισμού. Τα πλεονεκτήματα του δυαδικού αλφαβήτου κυρίως ως προς τον τρόπο της αναπαράστασής τους είναι αυτά που εδραίωσαν τη χρήση του στους ηλεκτρονικούς υπολογιστές.

1. Τα σύμβολα οποιουδήποτε αλφαβήτου μπορούν να εκφραστούν ως σύνολα δυαδικών χαρακτήρων με πρώτο παράδειγμα τα σήματα Μορς, το δυαδικό αλφάβητο με σύμβολα την τελεία και την παύλα. Η δυαδική κωδικοποίηση με τη χρήση μόνο των ψηφίων 0 και 1 είναι αρκετή για τη σύνθεση ενός τεράστιου φάσματος δεδομένων που χρησιμοποιούνται στους ηλεκτρονικούς υπολογιστές. Το μειονέκτημά της είναι ότι χρειάζονται πολλά ψηφία για την κωδικοποίηση αλφαβήτων με πολλούς χαρακτήρες.

Με n δυαδικά ψηφία πραγματοποιούνται 2^n συνδυασμοί, επομένως κωδικοποιούνται 2^n χαρακτήρες ενός αλφαβήτου. Έτσι, με οκτώ δυαδικά ψηφία παριστάνονται $2^8 = 256$ σύμβολα. Αυτά μπορεί να είναι τα δέκα ψηφία του δεκαδικού συστήματος, τα εικοσιέξι πεζά και άλλα τόσα κεφαλαία γράμματα του λατινικού αλφαβήτου, σημεία στίξης, μαθηματικά και ειδικά σύμβολα.

Όπως φαίνεται, ένας κώδικας των 8 bit καλύπτει τις συνηθισμένες απαιτήσεις και γι' αυτό έχει τυποποιηθεί διεθνώς. Ένας από τους πρώτους κώδικες είναι ο εκτεταμένος δυαδικά κωδικοποιημένος δεκαδικός κώδικας ανταλλαγής ή Ιμπ-Σι-Ντικ, ο γνωστός EBCDIC (Extended Binary Coded Decimal Interchange Code). Αργότερα, εμφανίστηκε ο κώδικας ASCII (American Standard Code for Information Interchange, Αμερικάνικος πρότυπος κώδικας για ανταλλαγή πληροφοριών) που χρησιμοποιείται και σήμερα. Ο πίνακας του κώδικα ASCII φαίνεται στο σχήμα 2.3. Το σχήμα 2.4 δείχνει την αντιστοίχιση δεκαδικών με δυαδικούς αριθμούς, καθώς και τον αριθμό των δυαδικών ψηφίων που απαιτούνται για την κωδικοποίηση.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
176							Α	·	Ε	Η	Ι		Ο	Υ	Ω	
192	ι	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
208	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
224	υ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
240	π	ο	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

Σχήμα 2.3 Τμήμα της ελληνικής τυποποίησης (ΕΛΟΤ 928) που αντιστοιχεί στον κώδικα ASCII. Κάθε χαρακτήρας αντιστοιχεί σε έναν αριθμό που προκύπτει από το άθροισμα γραμμής και στήλης (ο W αντιστοιχεί σε 80+7)

ΔΕΚΑ-ΛΙΚΟΙ	ΔΕΚΑ-ΕΞΑΔΙΚΟΙ	ΔΥΑΔΙΚΟΙ			
		1 ΨΗΦΙΟΥ	2 ΨΗΦΙΩΝ	3 ΨΗΦΙΩΝ	4 ΨΗΦΙΩΝ
0	0	0	0	0	0
1	1	1	1	1	1
2	2		10	10	10
3	3		11	11	11
4	4			100	100
5	5			101	101
6	6			110	110
7	7			111	111
8	8				1000
9	9				1001
10	A				1010
11	B				1011
12	C				1100
13	D				1101
14	E				1110
15	F				1111

Σχήμα 2.4 Αντιστοίχιση δυαδικών και δεκαεξαδικών με δεκαδικούς αριθμούς



Σχετικά με τον κώδικα ASCII, πολλές χώρες χρησιμοποιούν παραλλαγές του για την απόδοση ειδικών χαρακτήρων της γλώσσας τους. Το πρώτο από αριστερά bit είναι σε κάθε σύμβολο μηδέν, που σημαίνει ότι ο κώδικας είναι "επτάμπιτος," μπορεί δηλαδή να απεικονίσει $2^7 = 128$ χαρακτήρες. Μ' αυτόν τον τρόπο, μεταβάλλοντας την τιμή μηδέν σε τιμή ένα σε ορισμένα σύμβολα, δημιουργούνται οι επιπλέον χαρακτήρες για τις παραλλαγές αλφαβήτων όπως του Ελληνικού. Επίσης, το πρώτο από αριστερά bit χρησιμοποιείται ως ψηφίο ελέγχου για τη μεταφορά πληροφοριών σε απόσταση χωρίς σφάλματα. Χαρακτηριστικό παράδειγμα είναι η επιλογή του ψηφίου ελέγχου έτσι ώστε το άθροισμα των ψηφίων που αντιστοιχούν σε ένα σύμβολο να είναι άρτιο.

Οκτώ bit παριστάνουν ένα χαρακτήρα και ονομάζονται "byte", με Ελληνική απόδοση του όρου ως ψηφιολέξη. Αυτή είναι διαφορετική από τον όρο λέξη ή λέξη μηχανής που αποτελείται από πολλά byte μαζί, ο αριθμός των οποίων εξαρτάται από τον υπολογιστή. Συνήθως, όλα τα bit της λέξης μηχανής μεταφέρονται ταυτόχρονα (παράλληλα) μεταξύ των κυκλωμάτων του υπολογιστή μέσα από ομάδες ηλεκτρικών γραμμών.

2. Η δυαδική κωδικοποίηση υλοποιείται εύκολα με διάφορα τεχνικά μέσα. Όπως φάνηκε, οι ηλεκτρονόμοι και οι ηλεκτρονικές λυχνίες λειτουργούν ως διακόπτες δύο θέσεων (on/off) για την επίτευξη των 0 και 1 με τη λογική της διέλευσης ή όχι ηλεκτρικού σήματος. Το ίδιο ισχύει για τα τρανζίστορ και επομένως τα ολοκληρωμένα κυκλώματα με κέρδος τη μεγαλύτερη ταχύτητα μετάβασης από τη μια στην άλλη κατάσταση.

ΜΕΣΟ	1	0
Διάτρητες κάρτες	Τρύπα	Απουσία τρύπας
Ηλεκτρικά κυκλώματα	Ηλεκτρικό φορτίο	Απουσία ηλεκτρικού φορτίου
Ηλεκτρικά κυκλώματα	Τάση	Απουσία τάσης
Μαγνητικές μνήμες	Μαγνήτιση	Μαγνήτιση αντίθετης φοράς
Διαμόρφωση πλάτους	Πλάτος σήματος α	Πλάτος σήματος β
Διαμόρφωση συχνότητας	Συχνότητα σήματος α	Συχνότητα σήματος β
Οπτικός δίσκος	Φως	Σκοτάδι

Σχήμα 2.5 Τρόποι και μέσα για δυαδική κωδικοποίηση



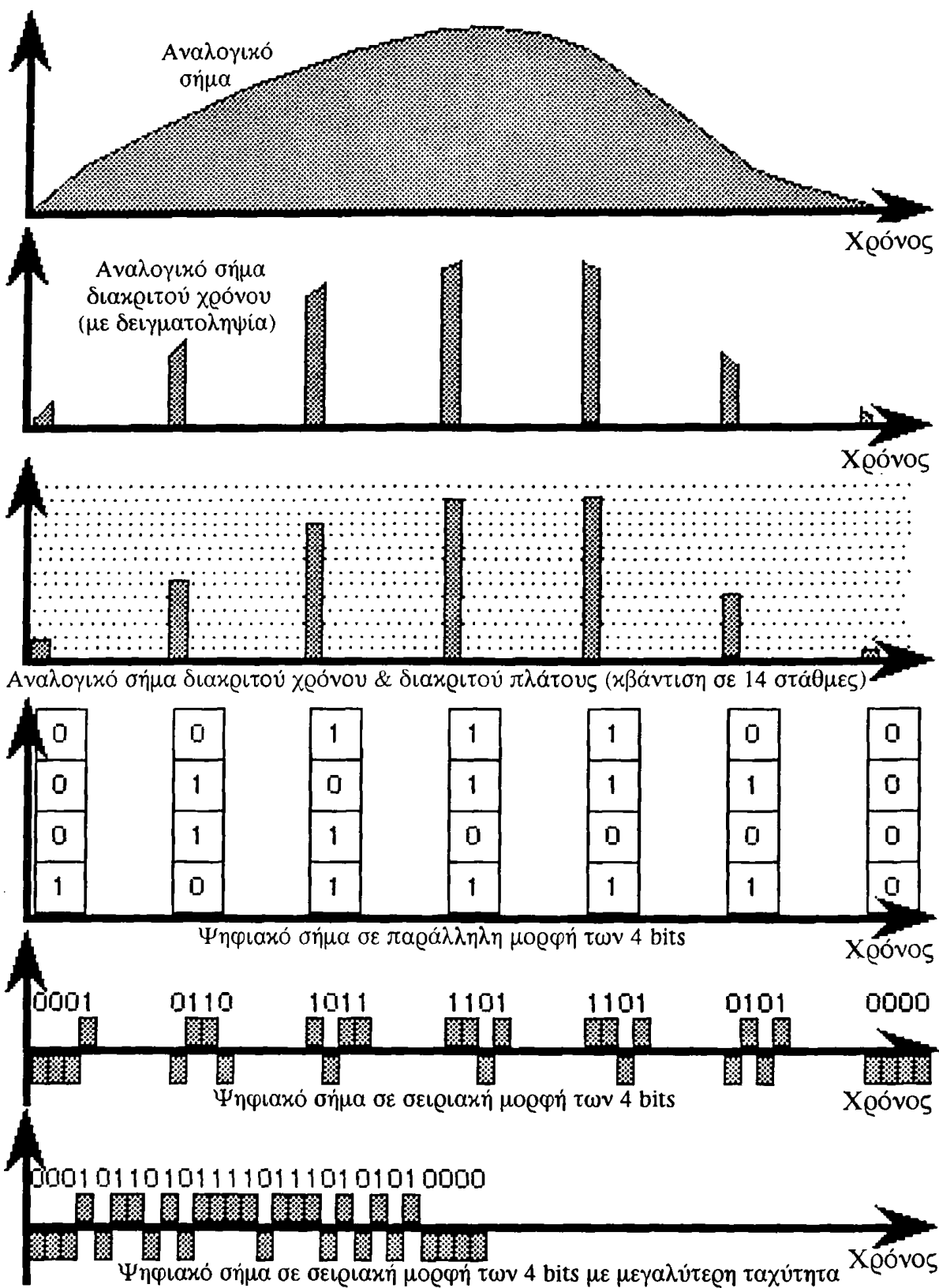
Αντίστοιχα, εύκολα και με αξιοπιστία πραγματοποιείται η αποθήκευση και ανάκτηση των δυαδικά κωδικοποιημένων δεδομένων και πληροφοριών με πολλούς τρόπους. Με μηχανικό τρόπο στις διάτρητες κάρτες, ηλεκτρικό στα ολοκληρωμένα κυκλώματα, μαγνητικό στα μαγνητικά μέσα αποθήκευσης όπως οι ταινίες, οι δίσκοι και οι δισκέτες, και οπτικό στους οπτικούς δίσκους. Το σχήμα 2.5 δείχνει τη δυαδική κωδικοποίηση διαφόρων φυσικών μεγεθών.

3. Όλα τα μετρήσιμα ή απαριθμήσιμα καθώς και τα φυσικά μεγέθη, μπορούν να κωδικοποιηθούν δυαδικά. Επομένως ο υπολογιστής επεξεργάζεται αριθμούς, γράμματα, αλλά και εικόνα και ήχο. Η διαφορά των φυσικών μεγεθών ως προς την κωδικοποίησή τους οφείλεται στο ότι όλα σχεδόν μεταβάλλονται με συνεχή τρόπο, λαμβάνουν δηλαδή άπειρο πλήθος τιμών και αναφέρονται ως αναλογικά. Κοινά παραδείγματα αποτελούν η ταχύτητα, η ένταση του ηλεκτρικού ρεύματος, η πίεση και η ένταση του φωτός.

Στη δυαδική απεικόνιση των φυσικών μεγεθών για την ηλεκτρονική επεξεργασία τους υπάρχει ένα ενδιάμεσο στάδιο, κατά το οποίο γίνεται μια επιλογή του άπειρου πλήθους των τιμών τους με ένα πεπερασμένο που περιορίζει την ακρίβεια, ανταποκρίνεται όμως στην πραγματικότητα. Η εργασία αυτή λέγεται ψηφιοποίηση. Η κωδικοποίηση αυτών των επιλεγμένων τιμών λέγεται ψηφιακή μετατροπή και καταλήγει στην ψηφιακή αναπαράσταση του μεγέθους. Όλη η διαδικασία πραγματοποιείται με κατάλληλα ηλεκτρονικά κυκλώματα, τους μετατροπείς αναλογικού σε ψηφιακό σήμα (Analog to Digital Converters, ADC) [Millman & Χαλκιάς, 1978].

Το σχήμα 2.6 δείχνει τη διαδικασία της ψηφιοποίησης και ψηφιακής μετατροπής ενός αναλογικού φυσικού μεγέθους.



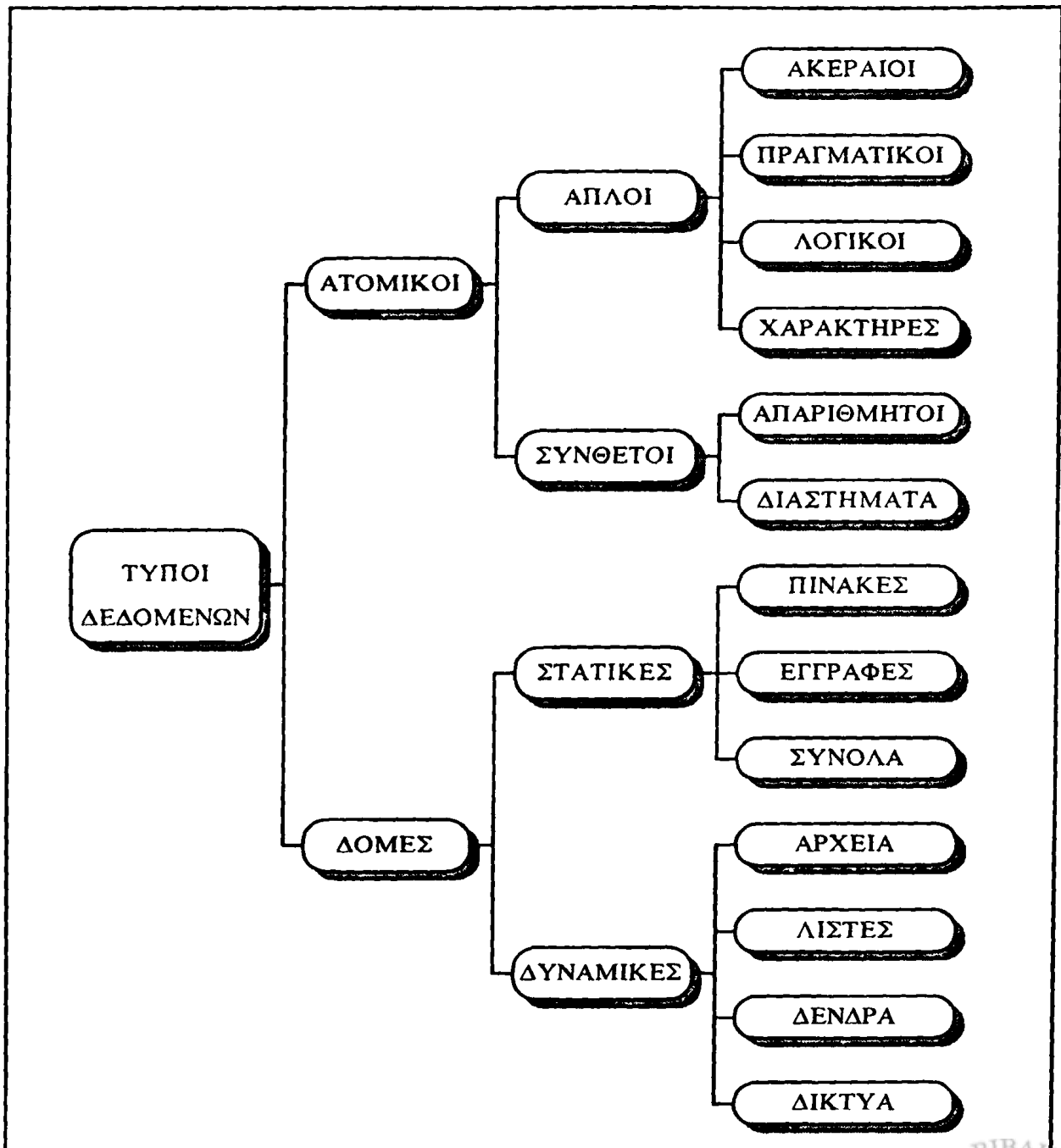


Σχήμα 2.6 Μετατροπή αναλογικού μεγέθους σε ψηφιακό και κωδικοποίησή του



2.2 Τύποι και δομές δεδομένων

Ένας τύπος δεδομένων ορίζεται ως ο συνδυασμός του συνόλου τιμών και των ιδιοτήτων που παίρνουν τα δεδομένα που τον απαρτίζουν. Το διάγραμμα του σχήματος 2.7 παρουσιάζει τα διάφορα είδη τύπων δεδομένων κατά κατηγορίες.



Σχήμα 2.7 Τύποι δεδομένων



Ως σύνθετοι τύποι δεδομένων αναφέρονται αυτοί που ορίζονται από απλούς ή άλλους ήδη ορισμένους σύνθετους.

Οι απλοί ή στοιχειώδεις και οι σύνθετοι τύποι δεδομένων ονομάζονται ατομικοί τύποι δεδομένων γιατί λαμβάνουν τιμές που δεν αναλύονται σε άλλες τιμές δεδομένων.

Η άλλη μεγάλη κατηγορία δεδομένων είναι οι δομές δεδομένων ή οι δομημένοι τύποι δεδομένων. Ως δομή δεδομένων ορίζεται ο τύπος δεδομένου που αναλύεται σε άλλα δεδομένα τα οποία είναι ατομικοί τύποι ή δομές δεδομένων και έχουν κάποια σχέση μεταξύ τους.

Ακέραιος τύπος (integer)

Οι ακέραιοι αριθμοί ανήκουν στους απλούς τύπους δεδομένων. Στο δεκαδικό αριθμητικό σύστημα κωδικοποιούνται δυαδικά σύμφωνα με τον κώδικα ASCII, διαδικασία διαφορετική από τη μετατροπή ενός συγκεκριμένου αριθμού του δεκαδικού συστήματος στο δυαδικό [Γαλάνης, 1993].

Όπως στο δεκαδικό σύστημα οι αριθμοί παριστάνονται ως αθροίσματα δυνάμεων του 10, κατ' αντιστοιχία στο δυαδικό αριθμητικό σύστημα με τα δύο μόνο ψηφία του οι αριθμοί παριστάνονται ως αθροίσματα δυνάμεων του 2. Έτσι ουσιαστικά ο αριθμός 456 του δεκαδικού συστήματος γράφεται

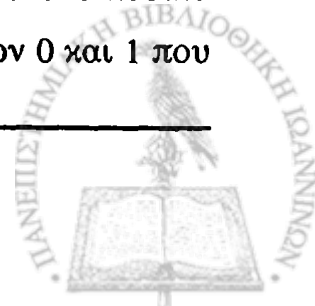
$$456 = 4 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$$

αριθμώντας τις δεκάδες ξεκινώντας από δεξιά και από το μηδέν, δηλαδή αποτελείται από έξι μονάδες, πέντε δεκάδες και τέσσερις εκατοντάδες.

Ο 456 στο δυαδικό σύστημα γράφεται ως

$$111001000 = 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

Η μετατροπή γίνεται πρακτικά δημιουργώντας τον πίνακα του σχήματος 2.8. Στην πρώτη γραμμή του παρατίθενται οι δυνάμεις του δύο κατ' αύξουσα σειρά ξεκινώντας από δεξιά. Η επόμενη γραμμή περιλαμβάνει τις αντίστοιχες τιμές στο δεκαδικό σύστημα. Στην τρίτη και τέταρτη γραμμή βρίσκονται οι συνδυασμοί των 0 και 1 που



δίνουν την τιμή του κάθε αριθμού στο δυαδικό, που προκύπτει θέτοντας 1 στους δεκαδικούς αριθμούς που έχουν ως άθροισμα τον επιθυμητό και 0 στους υπόλοιπους.

	10 2	9 2	8 2	7 2	6 2	5 2	4 2	3 2	2 2	1 2	0 2
	1024	512	256	128	64	32	16	8	4	2	1
456	0	0	1	1	1	0	0	1	0	0	0
139	0	0	0	1	0	0	0	1	0	1	1

Σχήμα 2.8 Πίνακας μετατροπής αριθμών από το δεκαδικό στο δυαδικό σύστημα. Παράδειγμα οι $456_{\text{δεκαδικός}}$ ($456 = 256+128+64+8$) και ο $139_{\text{δεκαδικός}}$ ($139 = 128+8+2+1$)

Σε αντίθεση με τη μετατροπή του αριθμού από το ένα αριθμητικό σύστημα στο άλλο, η δυαδική απεικόνιση του $456_{\text{δεκαδικό}}$ σύμφωνα με τον κώδικα ASCII έχει ως αποτέλεσμα τον εξής 24ψήφιο αριθμό με μορφή δυαδικών ψηφίων:

456: 00110100 00110101 00110110

Είναι εμφανές ότι ο αριθμός στο δυαδικό σύστημα είναι κομψότερος και επειδή και οι πράξεις είναι απλούστερες σ' αυτό το σύστημα, οι υπολογιστές χειρίζονται απευθείας τους δυαδικούς αριθμούς. Για έναν υπολογιστή, ο ακεραίος τύπος δεδομένων έχει ως σύνολο τιμών του ένα πεπερασμένο υποσύνολο των ακεραίων αριθμών και εξαρτάται από το είδος του υπολογιστή.

ΠΡΟΣΘΕΣΗ	ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ
$0 + 0 = 0$	$0 * 0 = 0$
$0 + 1 = 1$	$0 * 1 = 0$
$1 + 0 = 1$	$1 * 0 = 0$
$1 + 1 = 10$	$1 * 1 = 1$

Σχήμα 2.9 Πρόσθεση και πολλαπλασιασμός μονοψήφιων δυαδικών αριθμών. Ο αριθμός 10 δεν είναι το δέκα αλλά ο $1x2^1 + 0x2^0 = 2$. Στο δυαδικό σύστημα δεν υπάρχουν δεκάδες, παρά μόνο μονάδες και δυάδες. Έτσι το 10 σημαίνει 0 και 1 το κρατούμενο που γράφεται μια θέση αριστερά, μια μεγαλύτερη τάξη μεγέθους



Οι ιδιότητες του ακεραίου τύπου δεδομένων είναι οι γνωστές πράξεις από την αριθμητική, πρόσθεση και αφαίρεση, πολλαπλασιασμός και διαίρεση. Στο σχήμα 2.9 παρουσιάζονται οι κανόνες για την πρόσθεση και τον πολλαπλασιασμό των μονοψήφιων δυαδικών αριθμών. Σχετικά με τις δύο άλλες πράξεις, η μεν αφαίρεση θεωρείται ως πρόσθεση με αρνητικούς αριθμούς, η δε διαίρεση ως επαναλαμβανόμενες αφαιρέσεις και προσθέσεις.

Λογικός τύπος (Boolean)

Τα δεδομένα που εκφράζουν την τιμή που παίρνουν οι λογικές προτάσεις και είναι αληθείς ή ψευδείς, λέγονται λογικού τύπου. Όπως φαίνεται οι τιμές είναι μόνο δύο και συνήθως ονομάζονται 1 και 0 σε αντιστοιχία με το δυαδικό σύστημα.

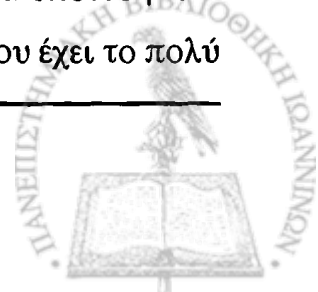
p	q	$p \wedge q$	$p \vee q$	$\neg p$
Αληθής (1)	Αληθής (1)	Αληθής (1)	Αληθής (1)	Ψευδής (0)
Αληθής (1)	Ψευδής (0)	Ψευδής (0)	Αληθής (1)	Ψευδής (0)
Ψευδής (0)	Αληθής (1)	Ψευδής (0)	Αληθής (1)	Αληθής (1)
Ψευδής (0)	Ψευδής (0)	Ψευδής (0)	Ψευδής (0)	Αληθής (1)

Σχήμα 2.10 Οι βασικές πράξεις μεταξύ των λογικών προτάσεων p και q

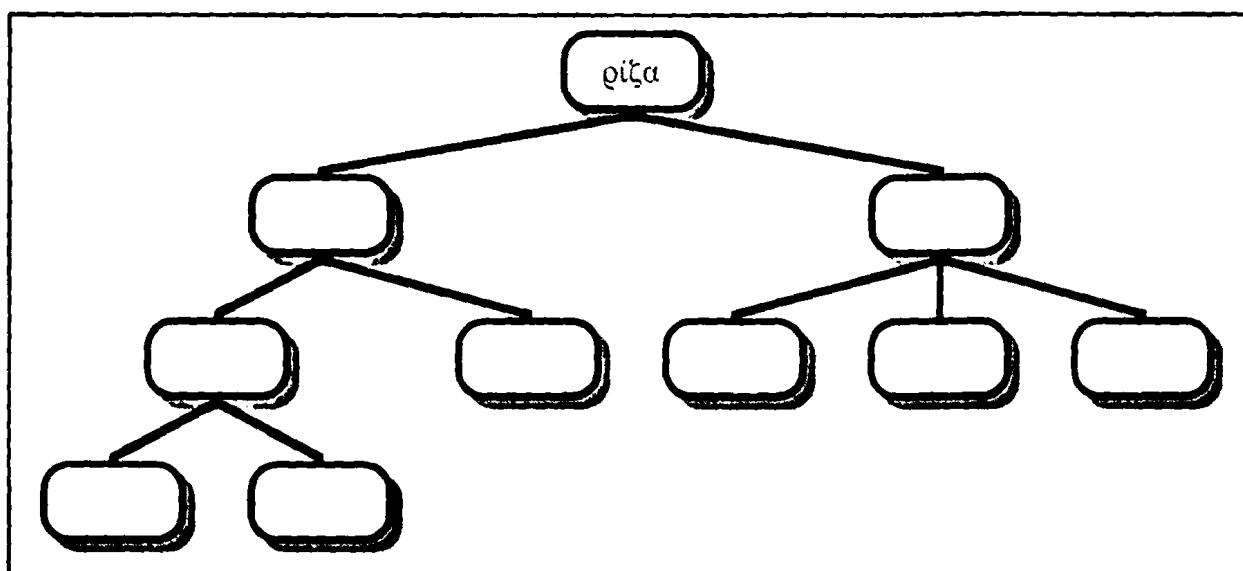
Οι ιδιότητες αυτού του τύπου δεδομένων είναι οι πράξεις σύζευξη (\wedge), διάζευξη (\vee) και άρνηση (\neg). Το σχήμα 2.10 παρουσιάζει τον πίνακα τιμών των πράξεων μεταξύ των δύο λογικών προτάσεων p και q .

Δένδρα (Trees)

Τα δένδρα είναι μια δυναμική δομή δεδομένων. Ο όρος δυναμική δηλώνει ότι ο αριθμός των μελών της μεταβάλλεται κατά την εκτέλεση του προγράμματος. Επίσης τα δένδρα είναι ιεραρχική και όχι γραμμική δομή δεδομένων, με την έννοια ότι κάθε στοιχείο τους (κόμβος) σχετίζεται με πολλά άλλα. Συγκεκριμένα, κάθε στοιχείο εκτός από ένα, τη ρίζα, έχει ένα μόνο προηγούμενο και ένα ή περισσότερα επόμενα. Συνέπεια αυτού είναι ότι υπάρχει μια μόνο διαδρομή από τη ρίζα προς ένα οποιοδήποτε στοιχείο του δένδρου (σχήμα 2.11). Το δένδρο στο οποίο κάθε κόμβος του έχει το πολύ



δύο επόμενους λέγεται δυαδικό και χρησιμοποιείται για την αναπαράσταση αριθμητικών παραστάσεων έχοντας το πλεονέκτημα της παρουσίασης της ιεραρχίας των πράξεων. Επειδή οι ιεραρχικές δομές αναπαριστούν με φυσικό τρόπο τις σχέσεις μεταξύ των στοιχείων του φυσικού κόσμου, η χρήση τους στην επεξεργασία δεδομένων έχει ιδιαίτερη σημασία. Παραδείγματα αυτής της δομής αποτελούν το οικογενειακό γενεαλογικό δένδρο, εφαρμογές βάσεων δεδομένων και πληροφοριών, ανάπτυξη εκπαιδευτικών εφαρμογών λογισμικού υπερμέσων/πολυμέσων.



Σχήμα 2.11 Σχηματικό διάγραμμα δομής δένδρου

Λίστες (Lists)

Η δομή λίστας είναι ένα ταξινομημένο σύνολο στοιχείων [Μαρινάκης & Τασόπουλος, 1985]. Στη στατική της μορφή που ο αριθμός των στοιχείων της είναι εκ των προτέρων καθορισμένα, ονομάζεται πίνακας. Όταν ο αριθμός των στοιχείων της λίστας μπορεί να μεταβάλλεται επιτρέποντας προσθέσεις νέων και αφαιρέσεις άλλων στοιχείων, η λίστα είναι μια δυναμική δομή δεδομένων. Μια λίστα μπορεί να διαμεριστεί σε άλλες, δύο ή περισσότερες λίστες μπορούν να συνδυαστούν για να φτιάξουν μια άλλη. Στη γλώσσα προγραμματισμού Logo οι λίστες χρησιμοποιούνται συνήθως για εφαρμογές σχετικές με τη φυσική γλώσσα, περιέχοντας γράμματα, λέξεις, ή και προτάσεις ως στοιχεία τους.



2.3 Αλγόριθμοι

Η λύση των προβλημάτων της καθημερινής ζωής και ιδιαίτερα με όσων ασχολείται η πληροφορική με την αυστηρότητα που τη διακρίνει, απαιτεί μια ολοκληρωμένη αντιμετώπιση ακολουθώντας συγκεκριμένη πορεία.

Το βασικότερο βήμα προς τη λύση είναι η κατανόηση του προβλήματος, η αντίληψη δηλαδή των δεδομένων του αλλά και των ζητούμενων, ο γενικός τρόπος επίλυσης και οι επιμέρους ιδιαιτερότητές του.

Ακολουθεί η σχεδίαση της μεθόδου επίλυσης που προϋποθέτει ορθολογική σκέψη και επιλογή της κατάλληλης μεθοδολογίας.

Επόμενο στάδιο αποτελεί η κωδικοποίηση της πορείας της λύσης ώστε να είναι κατανοητή από τον υπολογιστή. Αυτό περιλαμβάνει το πρόγραμμα που γράφει ο άνθρωπος, και τη μετάφραση του που κάνει η μηχανή για να φέρει το πρόγραμμα σε κατανοητή γι' αυτήν μορφή.

Έχει ήδη φανεί η ανάγκη της καταγραφής των διαδοχικών βημάτων που απαιτούνται για τη σωστή αντιμετώπιση κάθε προβλήματος, τον αλγόριθμο, που αποτελεί τον πυρήνα της πληροφορικής.

Αλγόριθμος είναι μια απλή και πεπερασμένη διαδοχική σειρά βημάτων για τον υπολογισμό ζητούμενων μεγεθών από δεδομένα. Κάθε βήμα αποτελείται από ορισμένες μονοσήμαντες και εκτελέσιμες πράξεις. Το στοιχείο του αλγόριθμου, που τον διακρίνει από τις μεθόδους των κλασικών μαθηματικών είναι ο βρόχος που εμφανίζεται σε προβλήματα όπου απαιτείται επανάληψη υπολογισμών. Ο βρόχος είναι μια κυκλική επαναληπτική λειτουργία -εκτέλεση εντολών- που περιλαμβάνει έλεγχο για την ισχύ ή μη κάποιας συνθήκης που καθορίζει την επανάληψη κάποιων βημάτων [McKeown, 1991].

Η αναπαράσταση του αλγόριθμου σε κατάλληλη μορφή αποτελεί ένα σημαντικό θέμα.

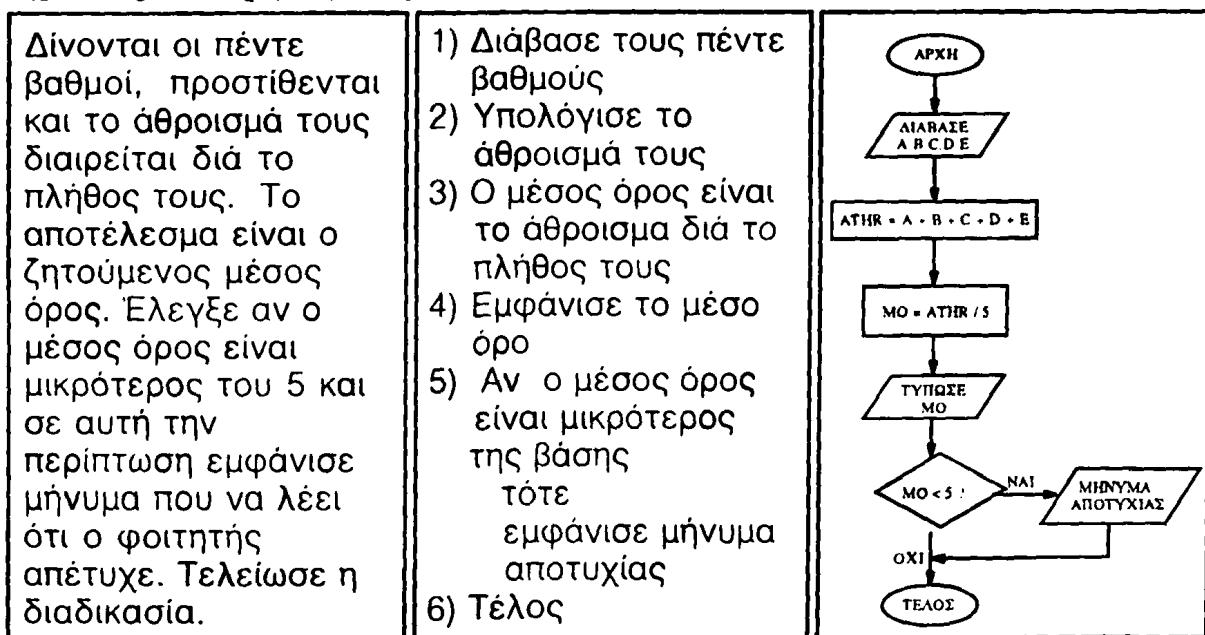
Έχουν επικρατήσει τρεις τρόποι καταγραφής των αλγορίθμων. Η φραστική μέθοδος, ο ψευδοκώδικας και το λογικό διάγραμμα.



Ο πρώτος χρησιμοποιεί τη φυσική γλώσσα και παραθέτει ταξινομημένες και με σειρά τις σκέψεις που απαιτούνται για την επίλυση του προβλήματος με τη μορφή προτάσεων γραπτού λόγου. Πλεονεκτήματα της φραστικής μεθόδου είναι η απλότητα και η ευχρηστία της, ενώ μειονεκτήματά της η πολυλογία και ο μεγάλος χρόνος για την καταγραφή της.

Ο ψευδοκώδικας στην προσπάθεια του να απομακρύνει τα μειονεκτήματα της φραστικής μεθόδου, χρησιμοποιεί μικρές και απλές προτάσεις σε συνδυασμό με σύμβολα γλωσσών προγραμματισμού. Θετικό στοιχείο του αποτελεί η γρήγορη καταγραφή του αλγόριθμου σε μορφή εύκολα κατανοητή. Στα μειονεκτήματά του περιλαμβάνονται ο μεγάλος χρόνος καταγραφής και η έλλειψη εποπτικής εικόνας, απαραίτητης σε πολύπλοκα προβλήματα.

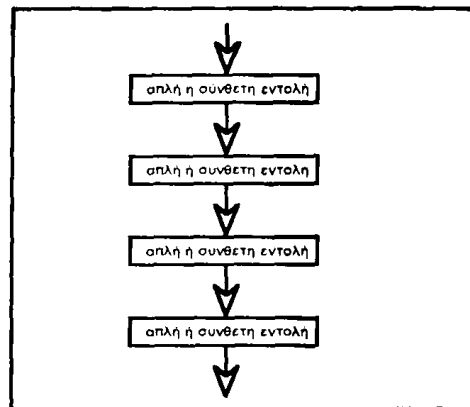
Το λογικό διάγραμμα χρησιμοποιεί απλά γεωμετρικά σχήματα συνδεδεμένα με βέλη που απεικονίζουν συγκεκριμένες έννοιες και περικλείουν απλές λέξεις ή φράσεις. Μ' αυτόν τον τρόπο υπάρχει εποπτεία της πορείας λύσης και λειτουργεί καλύτερα για το χρήστη. Το βασικό αρνητικό του σημείο είναι η προσοχή για τη λεπτομερή σχεδίαση της λύσης του προβλήματος.



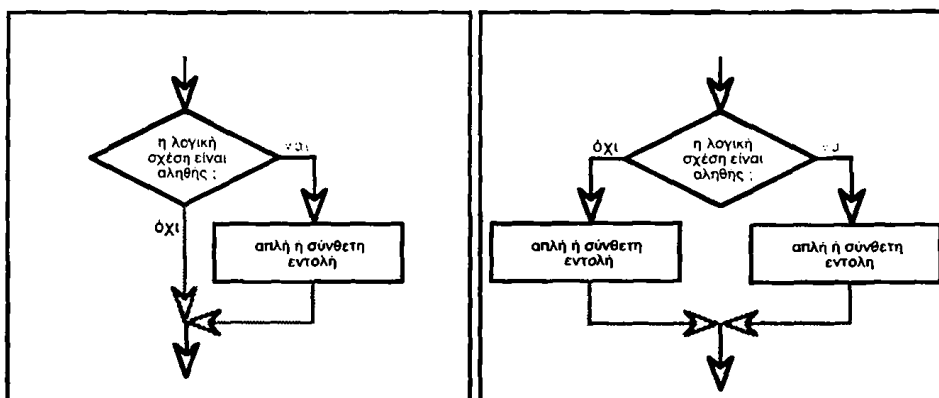
Σχήμα 2.12 Τρόποι αναπαράστασης αλγορίθμων (φραστική μέθοδος, ψευδοκώδικας, λογικό διάγραμμα). Α,Β,С, D, E: Μεταβλητές που κρατούν τους βαθμούς. ΑΘΡ, ΜΟ: Μεταβλητές για το άθροισμα και το μέσο όρο



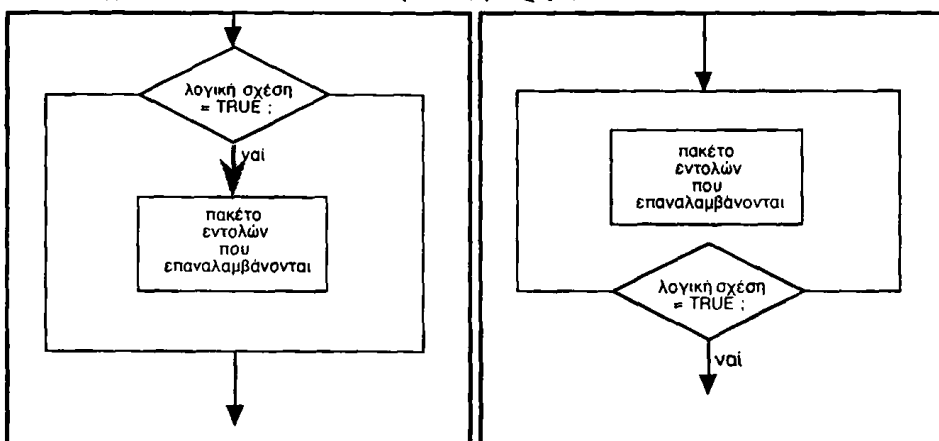
Το σχήμα 2.12 παραθέτει τους τρεις τρόπους αναπαράστασης για έναν αλγόριθμο που χρησιμοποιείται για το πρόβλημα της εύρεσης του μέσου όρου της βαθμολογίας πέντε ερωτήσεων σε ένα μάθημα ενός φοιτητή. Το αποτέλεσμα πρέπει να ελέγχεται για το αν είναι μικρότερο από το πέντε, που σημαίνει ότι ο φοιτητής αποτυχαίνει.



(α) Διαδοχή. Μια εντολή διαδέχεται την άλλη.



(β) Συνθήκες ελέγχου. Υλοποιούνται με τη μορφή IF - THEN και IF - THEN - ELSE.



(γ) Επανάληψη. Είναι μηχανισμός βρόχου. Οι εντολές του εκτελούνται εφόσον ισχύει η συνθήκη ελέγχου (αριστερά), ή μέχρις ότου παύει να ισχύει (δεξιά).

Σχήμα 2.13 Βασικά χαρακτηριστικά του δομημένου προγραμματισμού



Η διάρθρωση του αλγόριθμου για την επίλυση ενός προβλήματος περιλαμβάνεται στη σχεδίαση του υπόψη προγράμματος. Τρία είναι τα βασικά είδη σχεδιασμού προγραμμάτων. Ο ιεραρχικός, ο τμηματικός και ο δομημένος προγραμματισμός.

Σε περιπτώσεις πολύπλοκων προγραμμάτων που περιλαμβάνουν βρόχους, επαναληπτικές λειτουργίες, και λήψη αποφάσεων ενδείκνυται ο δομημένος προγραμματισμός. Ένα δομημένο πρόγραμμα είναι γραμμένο σε ανεξάρτητα μεταξύ τους λογικά τμήματα με μια είσοδο και μια έξοδο το καθένα. Μια τέτοια σχεδίαση επιτρέπει ευκολότερη ανάπτυξη, ανάγνωση, έλεγχο και διόρθωση του προγράμματος. Βασικά χαρακτηριστικά του δομημένου προγραμματισμού είναι η διαδοχή (sequence), οι συνθήκες ελέγχου (control structures), και η επανάληψη (iteration) (σχήμα 2.13) [Λαδιάς, 1991].

Αναφορές

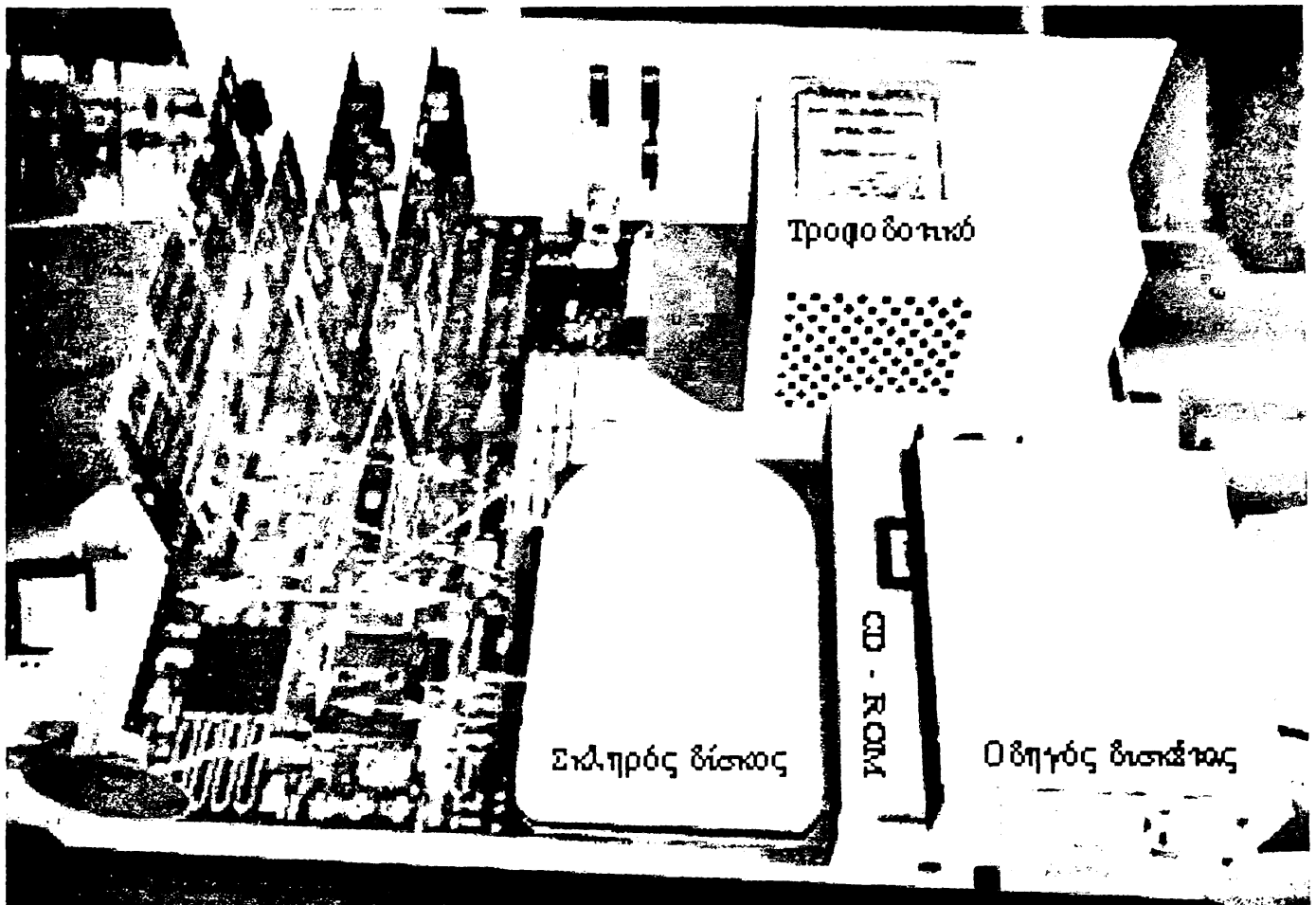
- Γαλάνης Σ., "Υπολογιστικά Μαθηματικά", Πανεπιστήμιο Ιωαννίνων 1993
Λαδιάς Γ. "Pascal. Μια Μεθοδική Προσέγγιση", Κλειδάριθμος, Αθήνα 1991
Μαρινάκης Κ., Τασόπουλος Α., "Οργάνωση Αρχείων", Αθήνα 1985
McKeown P. G., "Living with Computers", HBJ, USA 1991
Millman J., Χαλκιάς Χ., "Ωλοκληρωμένη Ηλεκτρονική", ΤΕΕ, Αθήνα 1978



3 Τεχνολογία ηλεκτρονικών υπολογιστών

3.1 Δομή και αρχιτεκτονική

Βλέποντας τον υπολογιστή ως ένα σύνολο συνεργαζόμενων τμημάτων που φέρουν μια ολοκληρωμένη εργασία σε πέρας, ο Ντέιβιντ Κρένκε προτείνει το μοντέλο των πέντε τμημάτων. Το υλικό (hardware), το λογισμικό (software), τα δεδομένα (data), τις διεργασίες (procedures), και τον άνθρωπο [Kroenke, 1987]. Το πρώτο τμήμα αφορά τον υπολογιστή ως μηχανή και παρουσιάζεται σ' αυτό το κεφάλαιο.



Δίαυλος Δεδομένων



Δίαυλος Διευθύνσεων



Δίαυλος Ελέγχου

Σχήμα 3.1 Ένας τυπικός μικροϋπολογιστής

Από όλα τα είδη των υπολογιστών ως προς τα δομικά τους στοιχεία γίνεται αναφορά στους πλέον διαδεδομένους, τους ηλεκτρονικούς υπολογιστές και ειδικότερα στους ψηφιακούς ηλεκτρονικούς υπολογιστές. Από την εξωτερική του εμφάνιση, ένας τυπικός μικροϋπολογιστής φαίνεται ότι αποτελείται από την κύρια μονάδα, ένα πληκτρολόγιο, ένα ποντίκι, μια οθόνη και έναν εκτυπωτή.

Μέσα στο κουτί της κύριας μονάδας βρίσκονται τα απαραίτητα εξαρτήματα και συσκευές που είναι υπεύθυνες για την επεξεργασία των δεδομένων, τη διαχείριση των πληροφοριών, την προσωρινή τους αποθήκευση και την εμφάνιση των αποτελεσμάτων. Υπάρχει μια μεγάλη "μητρική" πλακέτα με προσαρμοσμένα ηλεκτρονικά εξαρτήματα, και άλλες μικρότερες πλακέτες. Ακόμη υπάρχουν υποδοχές για περισσότερες, μονάδες αποθήκευσης δεδομένων και πληροφοριών όπως οδηγό δισκετών, σκληροί δίσκοι και οπτικοί δίσκοι CD. Τα υπόλοιπα ορατά τμήματα του συστήματος που συνδέονται με την κεντρική μονάδα είναι οι περιφερειακές συσκευές, που χρησιμεύουν στην εύκολη επικοινωνία του ανθρώπου με τη μηχανή. Το πληκτρολόγιο και το ποντίκι χρησιμοποιούνται για την εισαγωγή δεδομένων και λέγονται μονάδες εισόδου. Η οθόνη και ο εκτυπωτής εμφανίζουν τις πληροφορίες και ονομάζονται μονάδες εξόδου. Ένας τυπικός προσωπικός υπολογιστής φαίνεται στο σχήμα 3.1.

Το σημαντικότερο τμήμα του υπολογιστή είναι η αριθμητική και λογική μονάδα (Arithmetic and Logic Unit, ALU). Σ' αυτήν πραγματοποιούνται οι διεργασίες της επεξεργασίας των δεδομένων και του μετασχηματισμού των πληροφοριών. Τέτοιες είναι οι αριθμητικές και οι λογικές πράξεις και οι συγκρίσεις.

Ένα ακόμη σημαντικό τμήμα του υπολογιστή αποτελεί η μονάδα ελέγχου που συντονίζει τη λειτουργία των υπόλοιπων μονάδων του, και μαζί με την αριθμητική και λογική μονάδα από τεχνική άποψη αποτελούν την κεντρική μονάδα επεξεργασίας ή τον επεξεργαστή του υπολογιστή (Central Processing Unit, CPU).

Η αριθμητική και λογική μονάδα συνδέεται με την κεντρική ή κύρια μνήμη από όπου παίρνει τα δεδομένα και τις πληροφορίες, τα συνδυάζει κατάλληλα, ακολουθεί καθορισμένες εντολές, εξάγει τα αποτελέσματα και τα αποθηκεύει πάλι στην κεντρική μνήμη.

Στην κεντρική μνήμη αποθηκεύονται προσωρινά τα προς επεξεργασία δεδομένα, τα αποτελέσματα της επεξεργασίας, καθώς και το πρόγραμμα για την καθορισμένη εργασία. Επίσης επικοινωνεί με το εξωτερικό περιβάλλον μέσω των μονάδων εισόδου και εξόδου από όπου λαμβάνει τα δεδομένα και στέλνει τα αποτελέσματα.

Η μονάδα εισόδου αναλαμβάνει τη μετατροπή των εισερχόμενων δεδομένων στη δυαδική αναπαράσταση, ενώ η μονάδα εξόδου αποκωδικοποιεί τα αποτελέσματα σε μορφή κατανοητή από τον άνθρωπο και τα προωθεί στις εξωτερικές συσκευές παρουσίασής τους.

Η αρχιτεκτονική δομή του υπολογιστή βασίζεται λογικά σε τρεις διαύλους (busses) που στην πράξη είναι μια ομάδα παράλληλων ηλεκτρικών καλωδίων, τα οποία συνδέ-



ουν όλα τα δομικά μέρη του υπολογιστή και εξασφαλίζουν τη μεταξύ τους επικοινωνία.

Ο δίαυλος δεδομένων (data bus) είναι καλώδιο διπλής κατεύθυνσης και μεταφέρει δεδομένα και πληροφορίες από και προς τον επεξεργαστή.

Ο δίαυλος διευθύνσεων (address bus) είναι καλώδιο μιας κατεύθυνσης από τον επεξεργαστή προς την κύρια μνήμη και τις μονάδες εισόδου/εξόδου. Στο δίαυλο διευθύνσεων παρουσιάζονται ηλεκτρικά σήματα που καθορίζουν συγκεκριμένες θέσεις, τις διευθύνσεις, στην κύρια μνήμη ή κάποια συσκευή εισόδου/εξόδου.

Από το δίαυλο ελέγχου (control bus) διακινούνται ηλεκτρικά σήματα ελέγχου από και προς τον επεξεργαστή που είναι απαραίτητα για το συγχρονισμό των εργασιών όλων των τμημάτων του υπολογιστή [Δεληγιάννης κ.α., 1989].

3.2 Ηλεκτρονική

Η δυαδική κωδικοποίηση που χρησιμοποιείται στους ψηφιακούς ηλεκτρονικούς υπολογιστές υλοποιείται με δύο τιμές ηλεκτρικής τάσης. Το δυαδικό ψηφίο 0 παριστάνεται με τάση 0 Volt και το δυαδικό 1 με 5 Volt περίπου. Αυτές είναι οι μοναδικές τιμές της τάσης που βρίσκονται σε κάθε στιγμή στις ηλεκτρικές γραμμές των διαύλων.

Ο επεξεργαστής έχει ένα ρολόι που παράγει παλμούς με συγκεκριμένη συχνότητα, συγχρονίζει όλες τις λειτουργίες του υπολογιστή και εκφράζει κατά κύριο λόγο την ταχύτητα επεξεργασίας του [Zaks & Wolfe, 1990].

Σήμερα ένας τυπικός μικροϋπολογιστής έχει επεξεργαστή με ρολόι συχνότητας 200 MHz, που σημαίνει ότι ο χρόνος μεταξύ δύο διαδοχικών παλμών είναι 5 δισεκατομμυριοστά του δευτερολέπτου ή ότι σε κάθε δευτερόλεπτο παράγονται 200 εκατομμύρια παλμοί από το ρολόι, οι κύκλοι ρολογιού. Σε κάθε κύκλο γίνεται κάποια στοιχειώδης διεργασία στον υπολογιστή. Η αφαίρεση δύο αριθμών που για τον άνθρωπο είναι μια βασική πράξη, στον υπολογιστή ολοκληρώνεται μέσα σε λίγους κύκλους ρολογιού ανάλογα με τον τύπο του και διαρκεί λιγότερο από ένα χιλιοστό του δευτερολέπτου.

Βλέποντας τα πράγματα από την πλευρά της τεχνητής ευφυΐας, η ηλεκτρονική είναι για τους υπολογιστές ό,τι και η βιολογία για τον άνθρωπο.

Θεμέλιος λίθος στη βιολογία είναι ο άνθρακας και οι ενώσεις του ενώ η ηλεκτρονική βασίζεται σε ένα παρόμοιο στοιχείο, το πυρίτιο. Στη φύση το πυρίτιο εμφανίζεται σε μορφή κρυστάλλων με τα άτομά του “καρφωμένα” σε συγκεκριμένες θέσεις στο χώρο να σχηματίζουν ένα πλέγμα. Με προσμίξεις διαφορετικών ατόμων στον καθαρό κρύσταλλο του πυριτίου δημιουργούνται ημιαγωγοί τύπου n (ομάδες ατόμων με περίσσεια αρνητικού ηλεκτρικού φορτίου) και τύπου p (ομάδες ατόμων με περίσσεια θετικού ηλεκτρικού φορτίου). Με κατάλληλη διάταξη τέτοιων ημιαγωγών σχηματίζε-



ται το τρανζίστορ. Η συμπεριφορά του όταν λειτουργεί σε ακραίες καταστάσεις (κόρος, αποκοπή) είναι όμοια με ενός διακόπτη δύο θέσεων τύπου "on/off". Συγκεκριμένες συνδεσμολογίες από τρανζίστορ μαζί με άλλα ηλεκτρονικά στοιχεία όπως αντιστάσεις και πυκνωτές, αποτελούν τα συνδυαστικά και σειριακά λογικά κυκλώματα με τα οποία υλοποιείται ένας ηλεκτρονικός υπολογιστής.

Τα συνδυαστικά ηλεκτρονικά κυκλώματα είναι διακόπτες που συνδυάζουν δύο δυαδικά σήματα για να παράγουν ένα νέο. Τα ηλεκτρονικά στοιχεία που πραγματοποιούν τέτοιες εργασίες λέγονται λογικές πύλες (gates). Οι πύλες εκτελούν πράξεις που ακολουθούν τους κανόνες της μαθηματικής λογικής του Boolean προκειμένου να επεξεργαστούν αληθείς ή όχι καταστάσεις που συμβολίζονται με λογικά 1 ή 0 και να καταλήξουν σε αποτελέσματα.

Τρία είδη πυλών χρησιμοποιούνται ευρέως και οι συνδυασμοί τους υλοποιούν και τα πιο πολύπλοκα κυκλώματα για την πραγματοποίηση λογικών συναρτήσεων που απαιτούνται κατά τη σχεδίαση των υπολογιστών. Αυτές είναι οι AND, OR και NOT που εκτελούν τις λογικές πράξεις σύζευξη, διάζευξη και άρνηση αντίστοιχα όπως παρουσιάζονται στο σχήμα 2.10.

Περισσότερο σύνθετες λειτουργίες υλοποιούνται με συνδυασμούς των παραπάνω βασικών πυλών κατά διάφορους τρόπους. Έτσι σχηματίζεται η πύλη XOR που δηλώνει αποκλειστική διάζευξη (exclusive OR) και αποδίδει αληθή έξοδο όταν μια και μόνο μια από τις εισόδους της είναι αληθής. Συνδυασμός επίσης της AND ή της OR με τη NOT έχει ως αποτέλεσμα τις πύλες NAND (NOT AND) και NOR (NOT OR) με εξόδους αντίθετες από ότι οι AND και OR αντίστοιχα, αφού πραγματοποιούνται όταν στις εξόδους τους συνδεθεί μια πύλη NOT με την αρνητική δράση της.

Η απαίτηση για ολοένα και περισσότερο πολύπλοκους και μεγάλους συνδυασμούς λογικών πυλών οδήγησε στην κατασκευή τυποποιημένων ολοκληρωμένων κυκλωμάτων με πύλες, βασισμένα σε τρανζίστορ (Transistor - Transistor Logic, TTL). Τα συνδυαστικά λογικά κυκλώματα δεν έχουν μνήμη, δεν έχουν δηλαδή τη δυνατότητα να θυμούνται προηγούμενες καταστάσεις.

Η ιδιότητα της απαραίτητης για την υλοποίηση ενός υπολογιστή μνήμης περιέχεται στην άλλη κατηγορία κυκλωμάτων, τα σειριακά λογικά κυκλώματα. Σ' αυτά το σήμα που εμφανίζεται στην έξοδο τους δεν εξαρτάται μόνο από τα σήματα που υπάρχουν εκείνη τη στιγμή στην είσοδο, αλλά και από την κατάσταση που βρίσκεται το κύκλωμα, που μπορεί και μεταβάλλεται ανάλογα με τα σήματα εισόδου. Το απλούστερο σειριακό λογικό κύκλωμα μνήμης πραγματοποιείται με κατάλληλη σύνδεση λογικών πυλών (δύο OR και δύο NOT) και αποθηκεύει μια στοιχειώδη ποσότητα πληροφορίας, ένα bit. Αυτό το "κύτταρο μνήμης" ονομάζεται δισταθές κύκλωμα ή Flip-Flop. Αυτό αποθηκεύει ένα 0 ή 1 που θα εφαρμοσθεί στη είσοδό του και το εμφανίζει στην έξοδό του όταν του ζητηθεί.

Ουσιαστικά λοιπόν, ένας υπολογιστής είναι ένα σύνολο από διακόπτες τύπου



“on/off”. Ο συνδυασμός των καταστάσεων λειτουργίας των διακοπών αναπαριστά πληροφορίες, που χρησιμοποιούνται για τη μεταβολή της κατάστασης σε άλλους διακόπτες.

3.3 Τα συστατικά του υπολογιστή

Ο επεξεργαστής

Ο επεξεργαστής είναι το βασικότερο στοιχείο για τη λειτουργία του υπολογιστή. Όλα τα υπόλοιπα στοιχεία όπως η μνήμη και οι περιφερειακές συσκευές υπάρχουν για την “επικοινωνία” του επεξεργαστή με το χρήστη. Η κεντρική μονάδα επεξεργασίας δεν είναι ο μοναδικός επεξεργαστής σε έναν υπολογιστή. Επεξεργαστές υπάρχουν στις κάρτες διαχείρισης των γραφικών, του ήχου και στο πληκτρολόγιο, με έργο τους την αποδέσμευση του κύριου επεξεργαστή. Ένα τυπικό παράδειγμα μικροεπεξεργαστή αποτελεί το ολοκληρωμένο κύκλωμα Pentium της εταιρείας Intel. Έχει μέγεθος περίπου 2.5 cm² και περικλείει 3,100,000 τρανζίστορες. Όλες οι λειτουργίες στον μικροεπεξεργαστή εκτελούνται από ηλεκτρικά σήματα που θέτουν σε κατάσταση “on” ή “off” διαφορετικούς συνδυασμούς των διακοπών. Ένα τμήμα του, η αριθμητική και λογική μονάδα, εκτελεί αριθμητικές και λογικές πράξεις. Κάποια άλλα, οι καταχωρητές, σχηματίζουν γρήγορες μονάδες μνήμης για την προσωρινή αποθήκευση δεδομένων και πληροφοριών. Τέτοια είναι οι εντολές, οι θέσεις της μνήμης όπου βρίσκονται τα δεδομένα, τα ίδια τα δεδομένα.

Η μνήμη

Ο επεξεργαστής είναι ένα άβουλο όργανο. Για τη διεκπεραίωση οποιασδήποτε εργασίας είναι απαραίτητο να του δοθούν βήμα προς βήμα οι κατάλληλες οδηγίες - το πρόγραμμα που υλοποιεί τον αλγόριθμο, ο οποίος τροφοδοτείται με δεδομένα και εξάγει αποτελέσματα.

Το σύνολο των εντολών που υλοποιεί τον αλγόριθμο, τα δεδομένα και τα αποτελέσματα, είναι τοποθετημένα σε ένα μέσο αποθήκευσης, τη μνήμη. Από αυτήν αντλεί ο επεξεργαστής ό,τι χρειάζεται.

Σχηματικά αλλά και λειτουργικά η μνήμη αποτελείται από ένα σύνολο θέσεων που η κάθε μια έχει συνήθως μήκος ενός byte. Το μέγεθος και επομένως η χωρητικότητά της εξαρτάται από τον αριθμό αυτών των θέσεων που είναι αριθμημένες με αμφιμονοσήμαντη αντιστοίχιση. Σε κάθε μια αντιστοιχεί και ένας μοναδικός αριθμός που ονομάζεται διεύθυνση. Μ’ αυτόν τον τρόπο καλύπτεται η ανάγκη για προσπέλαση σε συγκεκριμένες θέσεις και δημιουργείται μια δομή με δυνατότητες αποθήκευσης όχι μόνο μεμονωμένων χαρακτήρων αλλά ολόκληρων “κειμένων”.



Η χωρητικότητα της μνήμης μετρείται σε byte και τα πολλαπλάσια του και εκφράζεται με δυνάμεις του δύο. Έτσι 1 Kilobyte (KB) δεν ισούται με 1000 byte αλλά με $2^{10} = 1024$ byte, διαφορά όχι μεγάλη. Τυπική χωρητικότητα της κύριας μνήμης των σύγχρονων μικροϋπολογιστών είναι τα 32 Megabyte (MB) ή αλλιώς $32 \cdot 2^{20} = 32 \cdot 1048576 = 33554432$ byte που αντιστοιχεί σε ισάριθμους χαρακτήρες, όσοι δηλαδή περιέχονται σε τριάντα δύο περίπου τόμους σαν αυτόν.

Σε περιπτώσεις όπου ο αλγόριθμος πρέπει να βρίσκεται πάντοτε στη μνήμη χρησιμοποιείται ένα είδος μνήμης που γράφεται ανεξίτηλα από τον κατασκευαστή της και ο επεξεργαστής μπορεί μόνο να τη διαβάσει. Βέβαια τα δεδομένα και τα αποτελέσματα μεταβάλλονται συνεχώς. Έτσι απαιτούνται μνήμες όπου ο επεξεργαστής μπορεί να διαβάσει αλλά και να γράψει. Είναι προφανές ότι σε μια τέτοια μνήμη γραφής/ανάγνωσης μπορούν εκτός από δεδομένα και αποτελέσματα να γραφούν και προγράμματα. Σε περίπτωση όμως διακοπής της ηλεκτρικής τροφοδοσίας του τσιπ της μνήμης, τα πάντα χάνονται.

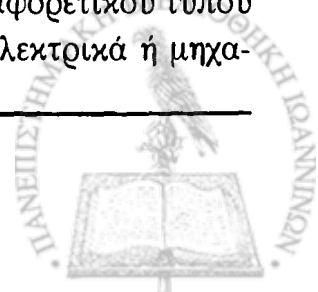
Οι μνήμες τις οποίες ο επεξεργαστής μπορεί μόνο να διαβάσει λέγονται μνήμες μόνο ανάγνωσης (Read Only Memories, ROM). Οι μνήμες που διαβάζονται και γράφονται από τον επεξεργαστή έχει καθιερωθεί κακώς να λέγονται μνήμες τυχαίας προσπέλασης (Random Access Memories, RAM). Αυτό δηλώνει ότι η προσπέλαση σε κάποια τυχαία θέση μνήμης γίνεται απευθείας και όχι ψάχνοντας τη σειρά από την αρχή. Αλλά αυτή η απευθείας προσπέλαση συμβαίνει και στις ROM. Έτσι μια πετυχημένη ονομασία για τις μνήμες γραφής/ανάγνωσης θα ήταν "Read Write Memories, RWM".

Μονάδες εισόδου/εξόδου

Ο υπολογιστής εκτελεί έναν καθορισμένο αριθμό λειτουργιών εφόσον αυτές αποτελούν μια αναπαράσταση του φυσικού κόσμου. Έχει λοιπόν ανάγκη από διαδικασίες, κυκλώματα και συσκευές που του επιτρέπουν να επικοινωνεί με το περιβάλλον. Να λαμβάνει "ερεθίσματα", να τα επεξεργάζεται και να αντιδρά ανάλογα με τα αποτελέσματα της επεξεργασίας. Κάτι ανάλογο ισχύει με τα αισθητήρια όργανα του ανθρώπου και τις περιοχές του εγκεφάλου που τα εξυπηρετούν.

Τα κυκλώματα εισόδου/εξόδου των υπολογιστών είναι ολοκληρωμένα κυκλώματα μέσα από τα οποία γίνεται η επικοινωνία του με το περιβάλλον. Αυτό υλοποιείται με τις περιφερειακές συσκευές που στέλνουν δεδομένα ή λαμβάνουν τα αποτελέσματα της επεξεργασίας τους.

Το γεγονός ότι ο έξω κόσμος δεν είναι συμβατός με τον υπολογιστή οδήγησε στη δημιουργία εξειδικευμένων κυκλωμάτων που φροντίζουν για την ομαλή επικοινωνία των δύο κόσμων και λέγονται "interfaces". Ο όρος interface αναφέρεται γενικά κατά τη σύνδεση δύο μονάδων και παίζει το ρόλο του μεσάζοντα. Στόχος του είναι να γεφυρώσει τις διαφορές των χαρακτηριστικών επικοινωνίας μεταξύ δύο διαφορετικού τύπου συσκευών. Τα χαρακτηριστικά της επικοινωνίας μπορεί να είναι ηλεκτρικά ή μηχανικά.



νικά ή ακόμη να αφορούν τους σκοπούς και τους τρόπους της επικοινωνίας [Μικρόπουλος & Λαδιάς, 1988].

Συνήθως τα σήματα στο φυσικό κόσμο είναι αναλογικής μορφής ενώ στο εσωτερικό του υπολογιστή ψηφιακά. Έτσι χρειάζονται εξειδικευμένα κυκλώματα που μετατρέπουν αναλογικά σήματα σε ψηφιακά (Analog/Digital Converters) για να μπορεί να τροφοδοτείται άμεσα ο υπολογιστής με δεδομένα από το περιβάλλον. Η αντίστροφη μετατροπή γίνεται με τους μετατροπείς από ψηφιακό σε αναλογικό σήμα (Digital/Analog Converters).

Η επικοινωνία μέσα στον υπολογιστή μεταξύ επεξεργαστή, μνήμης και κυκλωμάτων εισόδου/εξόδου γίνεται με ανταλλαγή πληροφοριών με τη μορφή των byte μέσα από τους διαύλους. Η ταυτόχρονη μεταφορά πληροφοριών ομαδοποιημένων σε byte λέγεται παράλληλη επικοινωνία. Το πρόβλημα μετάδοσης της πληροφορίας στο περιβάλλον με διάφορα μέσα όπως τα τηλεφωνικά δίκτυα και οι δορυφόροι, οδήγησαν στην κωδικοποίησή της. Νέες συσκευές αναπτύχθηκαν, τα modems, (MODulators - DEModulators, διαμορφωτές - αποδιαμορφωτές) που κωδικοποιούν και ελέγχουν τη μετάδοση της πληροφορίας σύμφωνα με τους κανόνες των πρωτοκόλλων επικοινωνίας.

3.4 Περιφερειακές συσκευές

Για την πλήρη αξιοποίηση των υπολογιστών είναι απαραίτητη η επικοινωνία τους με τον έξω κόσμο για τη λήψη δεδομένων, την παροχή πληροφοριών, τη μόνιμη αποθήκευσή τους και γενικότερα την εκτέλεση οποιουδήποτε έργου.

Γι' αυτούς τους σκοπούς χρησιμοποιούνται συσκευές που είναι γνωστές με το γενικό όνομα περιφερειακά ή περιφερειακές συσκευές. Η ανομοιογένεια τους είναι τόσο μεγάλη όσο απαιτεί το πλήθος των σκοπιμοτήτων που εξυπηρετούν, των τεχνικών που χρησιμοποιούνται και του δαιδαλώδους τρόπου σκέψης του ανθρώπινου εγκέφαλου. Ανάμεσα σε όλα αυτά βρίσκεται ο υπολογιστής για να συνδυάσει και να συντονίσει τη λειτουργία τους και λειτουργεί έτσι ώστε να εμφανίζεται μια αρμονική σύζευξη από τις πιο διαφορετικές τεχνολογίες.

Η κατάταξη των περιφερειακών τα ομαδοποιεί ως προς την εργασία που εκτελούν σε τέσσερις κατηγορίες:

1. Εξωτερική ή δευτερεύουσα μνήμη ή μονάδες αποθήκευσης.

Παραδείγματα αποτελούν οι δισκέτες, οι σκληροί δίσκοι και οι οπτικοί δίσκοι.

2. Μονάδες εισόδου/εξόδου που είναι απαραίτητες για την επικοινωνία του ανθρώπου με τον υπολογιστή. Τέτοιες είναι το πληκτρολόγιο, το ποντίκι, η οθόνη και ο εκτυπωτής.



3. Συνδέσεις με άλλους υπολογιστές για απευθείας επικοινωνία τους.
4. Συνδέσεις με συσκευές για τη συλλογή μετρήσεων και την παραγωγή σημάτων ελέγχου που στέλνονται στο περιβάλλον.

Μονάδες αποθήκευσης - Δίσκοι

Οι δίσκοι αποτελούν τις πλέον κοινές μονάδες αποθήκευσης στοιχείων. Οι χωρητικότητές τους κυμαίνονται από λίγες εκατοντάδες ΚΒ μέχρι μερικά GB.

Οι μηχανισμοί με τους οποίους δημιουργούνται τα δυαδικά 0 και 1 μπορεί να διαφέρουν, αλλά ο τελικός στόχος είναι οι μικροσκοπικές μεταβολές μικρών περιοχών της επιφάνειας του δίσκου έτσι ώστε κάποιες περιοχές να αναπαριστούν το 0 και κάποιες άλλες το 1.

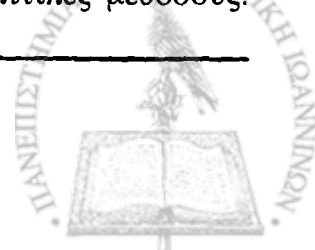
Κοινό χαρακτηριστικό των δίσκων είναι ο τρόπος οργάνωσης των στοιχείων στο δίσκο.

Για την αποθήκευση δεδομένων στο δίσκο, προηγείται η διαμόρφωση του (formatting). Με αυτόν τον τρόπο δημιουργείται ένας λογικός χάρτης που επιτρέπει την αποθήκευση και αναζήτηση των στοιχείων. Ο χάρτης αποτελείται από μαγνητικούς κώδικες ενσωματωμένους στη μαγνητική ταινία στην επιφάνεια του δίσκου. Οι κώδικες χωρίζουν την επιφάνεια σε τομείς (sectors) και ίχνη (tracks, ομόκεντροι κύκλοι). Αυτή η διαίρεση οργανώνει το δίσκο ώστε τα δεδομένα να γράφονται με ένα λογικό τρόπο και να προσπελαύνονται γρήγορα από την κεφαλή γραφής/ανάγνωσης που κινείται μπρος - πίσω πάνω στον περιστρεφόμενο δίσκο. Ο αριθμός των τομέων και των ιχνών, καθορίζει τη χωρητικότητα του δίσκου.

Το πιο διαδεδομένο μέσο για μόνιμη αποθήκευση και εύκολη μεταφορά σε συστήματα μικροϋπολογιστών είναι ο εύκαμπτος δίσκος (floppy disk) ή δισκέτα. Είναι πλαστικός δίσκος με μαγνητική επίστρωση και στις δύο πλευρές και προστατεύεται από σκληρό πλαστικό τετράγωνο κάλυμμα διαμέτρου συνήθως 3.5".

Στην ίδια κατηγορία μαγνητικού μέσου αποθήκευσης βρίσκεται και ο σκληρός δίσκος (hard disk). Είναι άκαμπτος από αλουμίνιο, με επίστρωση μαγνητικού υλικού και στις δύο πλευρές του και διάμετρο συνήθως 3.5". Συνήθως διατάσσονται πολλοί δίσκοι ο ένας πάνω από τον άλλον και σχηματίζεται μια δέσμη δίσκων (disk pack) για μεγαλύτερη χωρητικότητα. Η εγγραφή και ανάγνωση των δεδομένων γίνεται με τρόπο παρόμοιο με αυτόν στους οδηγούς δισκέτας, μόνο που εδώ υπάρχουν περισσότερες από δύο κεφαλές που δεν εφάπτονται στην επιφάνεια των δίσκων αλλά αιωρούνται πάνω τους σε απόσταση της τάξης του χιλιοστού του χιλιοστού του μέτρου. Οι σκληροί δίσκοι περικλείονται από μεταλλικό σταθερό περίβλημα και είναι εξωτερικοί ή εσωτερικοί στην κεντρικά μονάδα, σταθεροί ή μεταφερόμενοι.

Οι οπτικοί δίσκοι για υπολογιστές είναι παρόμοιοι με τους γνωστούς από τη μουσική οπτικούς δίσκους μόνιμης αποθήκευσης (Compact Disks, CD-ROMS). Η εγγραφή των δεδομένων γίνεται από τον κατασκευαστή και βασίζεται σε οπτικές μεθόδους.



Μια σχετικά ισχυρή δέσμη φωτός laser προσπίπτει στον πλαστικό δίσκο με το ανακλαστικό περίβλημα και προξενεί εσοχές που αντιστοιχούν σε δεδομένα. Η ανάγνωση γίνεται στον οδηγό του οπτικού δίσκου από ένα laser χαμηλής ισχύος, που αναγνωρίζει τις εσοχές από τη διαφορετική ανακλαστικότητά τους. Η χωρητικότητα ενός τέτοιου δίσκου είναι περίπου χίλιες φορές μεγαλύτερη από αυτήν της δισκέτας. Χαρακτηριστικό παράδειγμα αποτελεί ο οπτικός δίσκος "θησαυρός της Ελληνικής γλώσσας" που έχει αποθηκευμένα τα κείμενα όλων των αρχαίων Ελλήνων συγγραφέων και χωρητικότητα 57 εκατομμυρίων λέξεων.

Εκτός από τους δίσκους μόνο για ανάγνωση, υπάρχουν οι εγγράψιμοι (CD-R, CD-Recordable) και οι οπτικοί δίσκοι πολλαπλών εγγραφών/διαγραφών. Το 1997 εμφανίζεται ο ψηφιακός βιντεοδίσκος (Digital VideoDisk, DVD), παρόμοιος με το CD-ROM, με χωρητικότητα που φτάνει τα 17GB και δυνατότητες αποθήκευσης ολόκληρων κινηματογραφικών ταινιών.

Πληκτρολόγιο

Αποτελεί την πιο συνηθισμένη μονάδα εισόδου. Εξωτερικά έχει μόνο πλήκτρα αλλά στο εσωτερικό του τα ηλεκτρονικά του κυκλώματα περιμένουν το πάτημα ενός πλήκτρου. Κάθε πάτημα προκαλεί τη μεταβολή του ηλεκτρικού σήματος στα κυκλώματα που συνδέονται με το συγκεκριμένο πλήκτρο και ενημερώνεται το λογισμικό BIOS (Basic Input Output System, βασικό σύστημα εισόδου - εξόδου). Σε κάθε πλήκτρο αντιστοιχεί ένας κωδικός που στα περισσότερα πληκτρολόγια αντιστοιχεί στον κώδικα ASCII. Ο αριθμός των πλήκτρων στα συνηθισμένα πληκτρολόγια περιλαμβάνει γράμματα του αλφαβήτου, αριθμούς, ειδικά σύμβολα, πλήκτρα ειδικών λειτουργιών και προγραμματιζόμενα πλήκτρα.

Ποντίκι

Το ποντίκι (mouse) αποτελεί μια δημοφιλή μονάδα εισόδου δεδομένων στους υπολογιστές εξαιτίας των διαρκώς εξαπλούμενων γραφικών περιβαλλόντων εργασίας και επικοινωνίας του χρήστη με τη μηχανή. Είναι μικρή περιφερειακή συσκευή που χειρίζεται με το χέρι και μετατοπίζεται πάνω σε μια οριζόντια επιφάνεια όπως αυτή του γραφείου.

Το λογισμικό που συνοδεύει το ποντίκι μετατρέπει τα ηλεκτρικά σήματα που παράγονται από την κίνηση σε μετακίνηση ενός δείκτη στην οθόνη του υπολογιστή. Μ' αυτόν τον τρόπο ο υπολογιστής παίρνει συγκεκριμένες οδηγίες είτε από τη μετατόπιση του ποντικιού ή από το πάτημα ενός από τα ένα ως τρία πλήκτρα του που προσομοιώνουν βασικές λειτουργίες του πληκτρολογίου.

Οθόνη

Η πιο γνωστή περιφερειακή συσκευή εξόδου είναι η οθόνη (monitor). Στις περισσότερες περιπτώσεις είναι τύπου καθοδικού σωλήνα (Cathod Ray Tube, CRT), τεχνολο-



γία στην οποία είναι βασισμένες και οι σημερινές τηλεοράσεις. Βέβαια ο μηχανισμός απεικόνισης στον υπολογιστή είναι διαφορετικός από της τηλεόρασης αφού υπάρχει η ανάγκη μετάφρασης των bit σε κατάλληλα σήματα για τη λειτουργία της οθόνης.

Η απεικόνιση των χαρακτήρων γίνεται σε γραμμές και στήλες με τη διαδοχική σάρωση όλης της οθόνης από μια δέσμη ηλεκτρονίων. Ο κάθε χαρακτήρας σχηματίζεται από μια μήτρα κουκίδων που ονομάζονται εικονοστοιχεία (pixels).

Το μέγεθος της οθόνης μετριέται σε ίντσες που δηλώνουν τη διαγώνιο της. Χαρακτηριστικό της οθόνης με μεγάλη σημασία στην απόδοση γραφικών είναι η ανάλυσή της (resolution). Ορίζεται ως το γινόμενο του αριθμού των εικονοστοιχείων που απεικονίζονται στην οριζόντια διεύθυνση επί τον αντίστοιχο της κατακόρυφης. Μια καλή τιμή ανάλυσης που συναντάται σε σύγχρονες οθόνες είναι η 1280*1024.

Ένα δεύτερο χαρακτηριστικό των έγχρωμων οθονών είναι ο αριθμός των χρωμάτων που μπορούν να απεικονισθούν ταυτόχρονα στην οθόνη και στις πολύ καλές φθάνει μέχρι και τα αρκετά εκατομμύρια.

Η λειτουργία και ιδιαίτερα η απόδοση μιας οθόνης στην απεικόνιση γραφικών και χρωμάτων συνδέεται με την κάρτα γραφικών που κύριος ρόλος της είναι η μετατροπή των δεδομένων που λαμβάνει από τον επεξεργαστή σε κατάλληλη μορφή για την απεικόνισή τους στην οθόνη. Πρόκειται για ένα μετατροπέα ψηφιακού σήματος σε αναλογικό που μετατρέπει bit σε αναλογικό σήμα εικόνας. Οι σύγχρονες κάρτες γραφικών περιλαμβάνουν δικό τους επεξεργαστή που αναλαμβάνει την επεξεργασία των δεδομένων που προορίζονται για την οθόνη απαλλάσσοντας τον κύριο επεξεργαστή ο οποίος έχει πλέον χρόνο για σπουδαιότερες εργασίες. Αποτέλεσμα είναι η συνολική αύξηση της ταχύτητας του υπολογιστικού συστήματος.

Μια διαφορετική τεχνολογία οθονών που συναντάται κυρίως σε φορητούς υπολογιστές είναι οι οθόνες υγρών κρυστάλλων (Liquid Crystal Display, LCD). Η λειτουργία τους βασίζεται στη διαφορετική συμπεριφορά νηματοειδών κρυστάλλων κατά την εφαρμογή ή όχι τάσης, που ανακλούν ή όχι το πολωμένο φως με αποτέλεσμα να εμφανίζεται στην οθόνη το αντίστοιχο εικονοστοιχείο φωτεινό ή σκοτεινό.

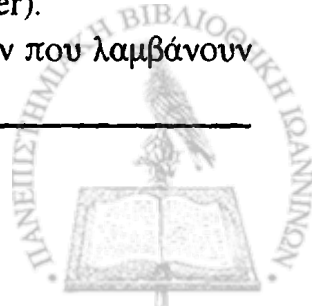
Πλεονεκτήματα τους αποτελούν ο μικρός όγκος, η χαμηλή κατανάλωση, η απουσία παραμόρφωσης και βλαβερής ακτινοβολίας και η επίπεδη σχεδιάσή τους. Στα μειονεκτήματά τους συγκαταλέγονται το μεγάλο κόστος και το μικρό τους μέγεθος.

Ο τύπος των συνηθισμένων οθονών είναι ο SVGA (Super Variable Graphics Array).

Εκτυπωτές

Η οθόνη είναι κατάλληλη μόνο για προσωρινή πληροφόρηση. Για τη μόνιμη αποτύπωση δεδομένων και πληροφοριών χρησιμοποιείται το χαρτί και το περιφερειακό που μεταφέρει τα κωδικοποιημένα σήματα πάνω του, ο εκτυπωτής (printer).

Οι εκτυπωτές ανάλογα με τον τρόπο διαχείρισης των δεδομένων που λαμβάνουν



κατατάσσονται σε τρεις κατηγορίες. Τους εκτυπωτές χαρακτήρων, γραμμής και σελίδας. Οι εκτυπωτές γραμμής διαχειρίζονται τα προς εκτύπωση δεδομένα γραμμή προς γραμμή, ενώ αυτοί της σελίδας τα διαχειρίζονται ανά σελίδα και γενικά είναι οι ταχύτεροι από όλους αφού χαρακτηρίζονται από παράλληλη λειτουργία λαμβάνοντας από τον υπολογιστή τους χαρακτήρες μιας σελίδας που τυπώνονται όλοι μαζί.

Οι τεχνολογίες που ακολουθούνται για την κατασκευή των εκτυπωτών είναι πολλές αλλά όλοι οι τύποι των εκτυπωτών διέπονται από κοινά βασικά χαρακτηριστικά, την ταχύτητα εκτύπωσης με συνηθισμένη μονάδα μέτρησης τις σελίδες ανά λεπτό και την ανάλυση της εκτύπωσης που μετριέται σε κουκίδες που τυπώνονται σε μια ίντσα (dots per inch, dpi).

Ως προς τον τρόπο εκτύπωσης, οι εκτυπωτές διακρίνονται σε δύο κατηγορίες. Τους κρουστικούς εκτυπωτές (impact printers) και τους μη κρουστικούς ή χωρίς επαφή (non-impact printers).

Στους πρώτους η εκτύπωση γίνεται με επαφή του μηχανισμού εκτύπωσης που είναι μια κεφαλή με ακίδες (pins) με το χαρτί που το κτυπούν μέσω μιας μελανοταινίας. Τυπικός εκπρόσωπός τους είναι ο εκτυπωτής ακίδων (dot matrix). Οι χαρακτήρες που μπορεί να τυπώσει είναι ενσωματωμένοι σ' αυτόν και ο καθένας σχηματίζεται από έναν ορισμένο συνδυασμό ακίδων. Η ποιότητα εκτύπωσης εξαρτάται από τον αριθμό των ακίδων που συνήθως είναι 24 με τυπική ανάλυση τα 360x180 dpi.

Ένας τύπος μη κρουστικού εκτυπωτή γραμμής είναι ο έγχυσης ή ψεκασμού μελάνης (ink jet). Για την αποτύπωση στο χαρτί χρησιμοποιείται μια γεννήτρια σταγονιδίων που αποτελείται από 50 περίπου ακροφύσια που εκτοξεύουν μελάνη. Οι σταγόνες της μελάνης φορτίζονται ηλεκτρικά και αποκλίνουν περνώντας από ηλεκτρικά πεδία ώστε να πέσουν στο κατάλληλο σημείο του χαρτιού για την απόδοση ενός συγκεκριμένου χαρακτήρα. Κύρια χαρακτηριστικά του εκτυπωτή ψεκασμού είναι το μικρό κόστος και η πολύ καλή ποιότητα εκτύπωσης.

Ο κυριότερος εκπρόσωπος των μη κρουστικών εκτυπωτών σελίδας που χαρακτηρίζεται από υψηλή ποιότητα και ταχύτητα είναι ο εκτυπωτής laser. Η τεχνική της εκτύπωσης είναι παρόμοια με αυτήν της ξηρογραφίας (φωτοαντίγραφο). Μια δέσμη laser πέφτει στη φωτοευαίσθητη εξωτερική επιφάνεια ενός κυλίνδρου και σχηματίζει χαρακτήρες φορτίζοντας την ηλεκτρικά. Ο κύλινδρος στη συνέχεια περνά από ένα χώρο με κόκκους άνθρακα (toner) οι οποίοι έλκονται από τις φορτισμένες περιοχές. Ο κύλινδρος έρχεται σε επαφή με το χαρτί όπου αποτίθενται οι κόκκοι του άνθρακα και μεταφέρονται οι χαρακτήρες. Καλή ποιότητα εκτύπωσης είναι τα 600 dpi και ταχύτητα οι 20 σελίδες ανά λεπτό.

Άλλες περιφερειακές συσκευές

Η αλματώδης εξέλιξη της τεχνολογίας δημιουργεί συνεχώς νέους τομείς της επιστήμης των ηλεκτρονικών υπολογιστών. Ένας από αυτούς είναι η εικονική πραγματι-



κότητα (virtual reality). Γενικός στόχος της είναι η καλύτερη αλληλεπίδραση ανθρώπου - μηχανής και έχει φθάσει σε σημείο όπου ο χρήστης αποκόβεται από τον πραγματικό κόσμο, βυθίζεται σε έναν τεχνητό και αλληλεπιδρά με τα εικονικά αντικείμενά του. Αυτό γίνεται με τη βοήθεια εξειδικευμένων περιφερειακών συσκευών που προσομοιώνουν τις ανθρώπινες αισθήσεις. Ένα τέτοιο περιφερειακό είναι το γάντι δεδομένων (data glove) το οποίο φοριέται στο χέρι του χρήστη, μεταδίδει στον υπολογιστή τις κινήσεις του και του επιστρέφει την αίσθηση των αντικειμένων του τεχνητού κόσμου που χειρίζεται. Η στερεοσκοπική όραση παρέχεται από ειδικά γυαλιά ή κράνη που αντικαθιστούν την οθόνη και περιλαμβάνουν συστήματα ανίχνευσης της θέσης του χρήστη.

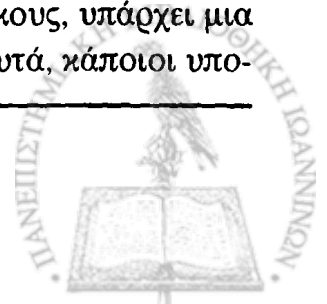
3.5 Δίκτυα

Ο διαρκώς αυξανόμενος όγκος των δεδομένων και οι ολοένα και πολυπλοκότερες απαιτήσεις για την επεξεργασία τους, οδηγούν στο τέλος της δεκαετίας του 1960 στην ανάγκη για άμεση μεταφορά δεδομένων και πληροφοριών από έναν υπολογιστή σε άλλον, ανεξάρτητα από τη φυσική θέση και την απόστασή τους. Οι υπολογιστές που βρίσκονται στον ίδιο χώρο ανταλλάσσουν δεδομένα συνδεδεμένοι με ένα καλώδιο. Όταν βρίσκονται σε μακρινές αποστάσεις, η μεταβίβαση των δεδομένων γίνεται μέσα από τηλεφωνικές γραμμές, με τη χρήση των μόντεμ που συνδέονται στους υπολογιστές και επιτρέπουν τη μεταφορά των ψηφιακών δεδομένων. Η λύση αυτή ονομάζεται μηδενικής θύρας (zero-slot).

Οι ανάγκες για ολοκληρωμένη επικοινωνία μεταξύ των υπολογιστών αυξάνουν και εμφανίζονται τα δίκτυα των υπολογιστών (networks). Τα δίκτυα αποτελούνται από προσωπικούς υπολογιστές, μεγάλα υπολογιστικά συστήματα, ή και άλλα δίκτυα που μπορεί να βρίσκονται στον ίδιο ή σε διαφορετικούς χώρους και είναι συνδεδεμένα μεταξύ τους. Χαρακτηριστικά της συνδεσμολογίας αποτελούν η επικοινωνία μεταξύ όλων των μηχανών, η κοινή χρήση πακέτων λογισμικού και περιφερειακών, αλλά και η ανεξαρτησία και αυτονομία της κάθε μιας.

Τα ομότιμα δίκτυα (peer to peer networks) αποτελούν το επόμενο στάδιο της εξέλιξης από τη σύνδεση μηδενικής θύρας και συνδέουν περισσότερους από δύο υπολογιστές. Σε αντίθεση μ' αυτήν όπου οι υπολογιστές συνδέονται με τις σειριακές τους θύρες, κάθε υπολογιστής εξοπλίζεται με μια ειδική κάρτα που αναλαμβάνει τη σύνδεση. Όλοι οι υπολογιστές ενός ομότιμου δικτύου έχουν τα ίδια δικαιώματα στη χρήση του λογισμικού και των περιφερειακών.

Εξαιτίας των περιορισμών των ομότιμων δικτύων όπως η ανάγκη να βρίσκονται σε λειτουργία όλοι οι υπολογιστές για πρόσβαση σε όλους τους δίσκους, υπάρχει μια δεύτερη κατηγορία δικτύων, τα δίκτυα με εξυπηρετητή (server). Σ' αυτά, κάποιοι υπο-



λογιστές είναι οι εξυπηρετητές και άλλοι οι πελάτες (clients) που εξυπηρετούνται από τους πρώτους. Ένα από τα βασικότερα χαρακτηριστικά των δικτύων με εξυπηρετητή είναι η δυνατότητα στη χρήση κοινών αρχείων. Οι πελάτες εκμεταλλεύονται όλες τις υπολογιστικές δυνατότητές τους χωρίς να παρενοχλούνται από τους άλλους χρήστες.

Ο τρόπος σύνδεσης των υπολογιστών για την υλοποίηση του δικτύου λέγεται τοπολογία (topology), κάθε υπολογιστής του αναφέρεται ως κόμβος (node) και αυτοί στους οποίους μπορούν να εργάζονται οι χρήστες, σταθμοί εργασίας (workstations). Οι κανόνες κωδικοποίησης των δεδομένων για την κατανοητή επικοινωνία των κόμβων του δικτύου, λέγονται πρωτόκολλα επικοινωνίας (communication protocols). Για τη σύνδεση δικτύων χρησιμοποιούνται οι γέφυρες (bridges) που δημιουργούν ένα μεγάλο ενιαίο δίκτυο, και οι δρομολογητές (routers) που μπορούν και συνδέουν δίκτυα που χρησιμοποιούν διαφορετικά πρωτόκολλα επικοινωνίας.

Γενικά υπάρχουν δύο κατηγορίες δικτύων υπολογιστών. Τα δίκτυα στα οποία οι κόμβοι είναι εγκατεστημένοι σε μακρινές μεταξύ τους αποστάσεις και για την επικοινωνία τους απαιτούνται τηλεφωνικά ή δορυφορικά συστήματα, και λέγονται δίκτυα ευρείας περιοχής (Wide Area Networks, WAN). Τα δίκτυα με τους κόμβους τους να βρίσκονται σε περιορισμένη φυσική θέση όπως ένα κτίριο και για την επικοινωνία τους χρησιμοποιούν ειδικές καλωδιώσεις, και ονομάζονται τοπικά δίκτυα (Local Area Networks, LAN).

Εξέλιξη των δικτύων αποτελούν τα καταναμημένα συστήματα (distributed systems). Σε αντίθεση με τα πρώτα όπου οι κόμβοι μπορούν να λειτουργούν και αυτόνομα και γνωρίζουν που βρίσκονται τα αρχεία τους, ένα καταναμημένο σύστημα παρουσιάζεται στους χρήστες του ως ένας υπολογιστής. Η φυσική σύνδεση των υπολογιστών παραμένει κρυφή και οι χρήστες δε γνωρίζουν που βρίσκονται τα δεδομένα τους και που γίνεται η επεξεργασία τους. Στα πλεονεκτήματά τους συμπεριλαμβάνονται η καλύτερη απόδοση και αξιοπιστία τους και η εύκολη επέκτασή τους.

Από τη δομή και τις ιδιότητες των δικτύων είναι φανερά τα πλεονεκτήματά τους. Αξιοσημειώτες είναι οι υπηρεσίες του παγκόσμιου διαδικτύου (internet) με υπηρεσίες όπως του ηλεκτρονικού ταχυδρομείου (electronic mail, e-mail) και του παγκόσμιου ιστού (world wide web, www). Κάθε χρήστης έχει μια ηλεκτρονική διεύθυνση και ένα κωδικό όνομα και μπορεί να επικοινωνεί με οποιονδήποτε έχει πρόσβαση στο δίκτυο και γνωρίζει τα στοιχεία του. Σημαντική εφαρμογή του διαδικτύου αποτελεί η εξ' αποστάσεως εκπαίδευση που φέρνει σε επικοινωνία και συνεργασία μαθητές και εκπαιδευτικούς που βρίσκονται σε διαφορετικές φυσικές θέσεις [Γεωργίου κ.ά., 1999].

Η σελίδα web του Πανεπιστημίου Ιωαννίνων για πρόσβαση σε πληροφορίες σχετικές μ' αυτό από οποιοδήποτε σημείο του κόσμου, έχει διεύθυνση <http://www.uoi.gr>.



Αναφορές

- Γεωργίου Θ., Κάππος Ι., Λαδιάς Α., Μικρόπουλος Α., Τζιμογιάννης Α., Χαλκιά Κ.,
“Πολυμέσα - Δίκτυα”, ΥΠΕΠΘ, ΟΕΔΒ, Αθήνα 1999
- Δεληγιάννης Φ., Λαδιάς Τ., Μικρόπουλος Τ., Μπέλλου Ι., Σαριδάκη Α, Χαλκίδης Α.,
“Computers. Περιπλάνηση στον κόσμο τους”, Φυσικός Κόσμος 117, 7 1989
και τεύχη 118, 8. 119, 8. 120, 8. 121, 4
- Kroenke D.M., Dolan K. “Business Computer Systems: An Introduction”, Mitchell, CA
1987
- Μικρόπουλος Τ., Λαδιάς Τ., “Προγραμματισμός του VIA (Versatile Interface Ada-
pter)”, ΓΤΕ ΕΛΚΕΠΑ, Αθήνα 1988
- Zaks R., Wolfe A., “Από τα Τσιπ στα Συστήματα”, Κλειδάριθμος, Αθήνα 1990



4 Λογισμικό

Το υλικό μέρος ενός υπολογιστικού συστήματος ανεξάρτητα από την ισχύ και τις προδιαγραφές του είναι αδύνατο να επεξεργαστεί οποιαδήποτε δεδομένα χωρίς τη βοήθεια του λογισμικού. Αυτό είναι που “εμφυσά” πνοή στο υλικό, εκμεταλλεύεται τις δυνατότητες του και δημιουργεί ένα αρμονικά συνεργαζόμενο σύνολο, το ολοκληρωμένο υπολογιστικό σύστημα.

Ο τρόπος με τον οποίο ένας υπολογιστής επεξεργάζεται τα δεδομένα εξαρτάται από το είδος του και από τις ανάγκες και σκοπούς που εξυπηρετούν οι πληροφορίες που προκύπτουν [Carson, 1990].

Ο πολυπρογραμματισμός (multiprogramming) χαρακτηρίζεται από τη διαχείριση πολλών προγραμμάτων που βρίσκονται στην κεντρική μνήμη, με κάθε περιφερειακή συσκευή να απασχολείται από ένα διαφορετικό πρόγραμμα. Ο επεξεργαστής εκτελεί ταυτόχρονα διαφορετικές εργασίες από διαφορετικά προγράμματα. Ο πολυπρογραμματισμός χρησιμοποιείται στους περισσότερους υπολογιστές, και ιδιαίτερα στους προσωπικούς που διαθέτουν μια κεντρική μονάδα επεξεργασίας. Εφαρμογές πολυπρογραμματισμού είναι συνήθως τα λογιστικά προγράμματα.

Ο καταμερισμός χρόνου επεξεργασίας (time sharing) αποτελεί μια περίπτωση του πολυπρογραμματισμού και πολλοί χρήστες χρησιμοποιούν ταυτόχρονα τον υπολογιστή. Όλα τα προγράμματα βρίσκονται στην κεντρική μνήμη και εκτελούνται τμηματικά και διαδοχικά με μεγάλη ταχύτητα ώστε ο κάθε χρήστης θεωρεί ότι απασχολεί αποκλειστικά τη μηχανή και ότι εργάζεται ταυτόχρονα με τους υπόλοιπους. Το σύστημα χαρακτηρίζεται ως πολλών χρηστών (multiuser) με καθένα να έχει το δικό του τερματικό που επικοινωνεί με τον κεντρικό υπολογιστή. Τυπική εφαρμογή αποτελεί ο έλεγχος πιστωτικών καρτών.

Στην τεχνική πραγματικού χρόνου (real time) η επεξεργασία των δεδομένων γίνεται αμέσως με την εισαγωγή τους στον υπολογιστή και τα αποτελέσματά της είναι διαθέσιμα σε πολύ μικρό χρόνο. Τυπικά συστήματα πραγματικού χρόνου χρησιμοποιούνται για τις τραπεζικές συναλλαγές και τις κρατήσεις θέσεων από αεροπορικές εταιρίες.

Η παράλληλη επεξεργασία (parallel processing) ή πολυεπεξεργασία (multiprocessing) απαιτεί τουλάχιστο δύο επεξεργαστές που επεξεργάζονται ταυτόχρονα διαφορετικά δεδομένα του ίδιου προγράμματος. Είναι ο τρόπος που επιτρέπει πραγματική ταυτόχρονη επεξεργασία και αυξάνει την ταχύτητα λειτουργίας του υπολογιστή.

Η επεξεργασία κατά ομάδες (batch processing) χαρακτηρίζεται από την επεξεργασία ενός μόνο προγράμματος που βρίσκεται στην κεντρική μνήμη του υπολογιστή και



τη δέσμευση όλων των περιφερειακών για την εξυπηρέτησή του. Συναντάται σε μικρο-υπολογιστές και σε mini συστήματα.

Το λογισμικό χωρίζεται σε δύο μεγάλες κατηγορίες. Το βασικό λογισμικό ή προγράμματα συστήματος, και τα προγράμματα εφαρμογών.

4.1 Λογισμικό συστήματος

Το λογισμικό συστήματος είναι ένα απαραίτητο κομμάτι σε κάθε ολοκληρωμένο υπολογιστικό σύστημα που διευκολύνει την επικοινωνία του ανθρώπου με τη μηχανή. Αποτελεί το μεσάζοντα για τα δεδομένα που εισάγει και τις εργασίες που επιθυμεί να διεκπεραιώσει με τα διάφορα περιφερειακά ο χρήστης, και τον υπολογιστή που το μόνο που καταλαβαίνει είναι η διέλευση ή όχι ηλεκτρικού σήματος από τα ηλεκτρονικά του κυκλώματα ή η μαγνήτιση και απομαγνήτιση των μαγνητικών μέσων του.

Λειτουργικό σύστημα

Στις πρώτες προσπάθειες ανάπτυξης των ηλεκτρονικών υπολογιστών η επικοινωνία ανθρώπου-μηχανής γινόταν στη γλώσσα του μηχανήματος με την εφαρμογή ηλεκτρικών σημάτων κατευθείαν στα ηλεκτρονικά του κυκλώματα.

Ο απάνθρωπος αυτός τρόπος επικοινωνίας οδήγησε τους ειδικούς αφενός να δημιουργήσουν τα λειτουργικά συστήματα και αφετέρου να βελτιώσουν τις γλώσσες προγραμματισμού.

Τα πρώτα λειτουργικά συστήματα εμφανίσθηκαν στα τέλη της δεκαετίας του 1940. Ήταν απλά προγράμματα για την είσοδο και την έξοδο δεδομένων και αποτελούνταν από μερικές εκατοντάδες εντολών σε γλώσσα μηχανής. Στα μέσα της δεκαετίας του 1950, οι περισσότεροι υπολογιστές δούλευαν με τρόπο ομαδικής επεξεργασίας. Μ' αυτόν τον τρόπο το λειτουργικό σύστημα συγκέντρωνε πρώτα όλα τα προγράμματα που υπέβαλαν οι χρήστες και στη συνέχεια τα εκτελούσε διαδοχικά και γρήγορα, εξαλείφοντας έτσι τις καθυστερήσεις που δημιουργούνταν με το φόρτωμα ενός προγράμματος κάθε φορά από τον προγραμματιστή που ήταν υποχρεωμένος να βρίσκεται δίπλα στο μηχάνημα για να αντιμετωπίσει τυχόν λάθος. Η εξέλιξη των λειτουργικών συστημάτων ακολούθησε την τεχνολογική ανάπτυξη και σήμερα η επικοινωνία ανθρώπου-μηχανής δεν απαιτεί την παράθεση εντολών με αυστηρό συντακτικό, αλλά χειρισμό εικόνων και γραφικών.

Ως λειτουργικό σύστημα (operating system) ορίζεται η συλλογή προγραμμάτων για την ομαλή διαχείριση και χρήση διαφόρων λειτουργικών μέσων όπως ο επεξεργαστής, η κύρια μνήμη, τα μέσα μαζικής αποθήκευσης και η διάταξη εισόδου/εξόδου. Πρόκειται δηλαδή για τα προγράμματα που ελέγχουν τη λειτουργία του υπολογιστή.



Το λειτουργικό σύστημα αποτελεί το “ζωντανό” μέσο που μεσολαβεί στην επικοινωνία του ανθρώπου με τη μηχανή και βελτιώνει τη συνολική απόδοση του συστήματος.

Ειδικότερα ασχολείται με τις εξής εργασίες:

Διεύθυνση της εκτέλεσης των προγραμμάτων

Διαχείριση επεξεργαστών για ικανοποίηση διακοπών

Διαχείριση της μνήμης

Διαχείριση αρχείων

Διαχείριση διατάξεων

Διαχείριση εργασιών

Ανεξαρτητοποίηση της γραφής προγραμμάτων από τις ιδιότητες του υλικού

Αναγνώριση και επισήμανση λανθασμένης συμπεριφοράς προγραμμάτων και ενεργειών.

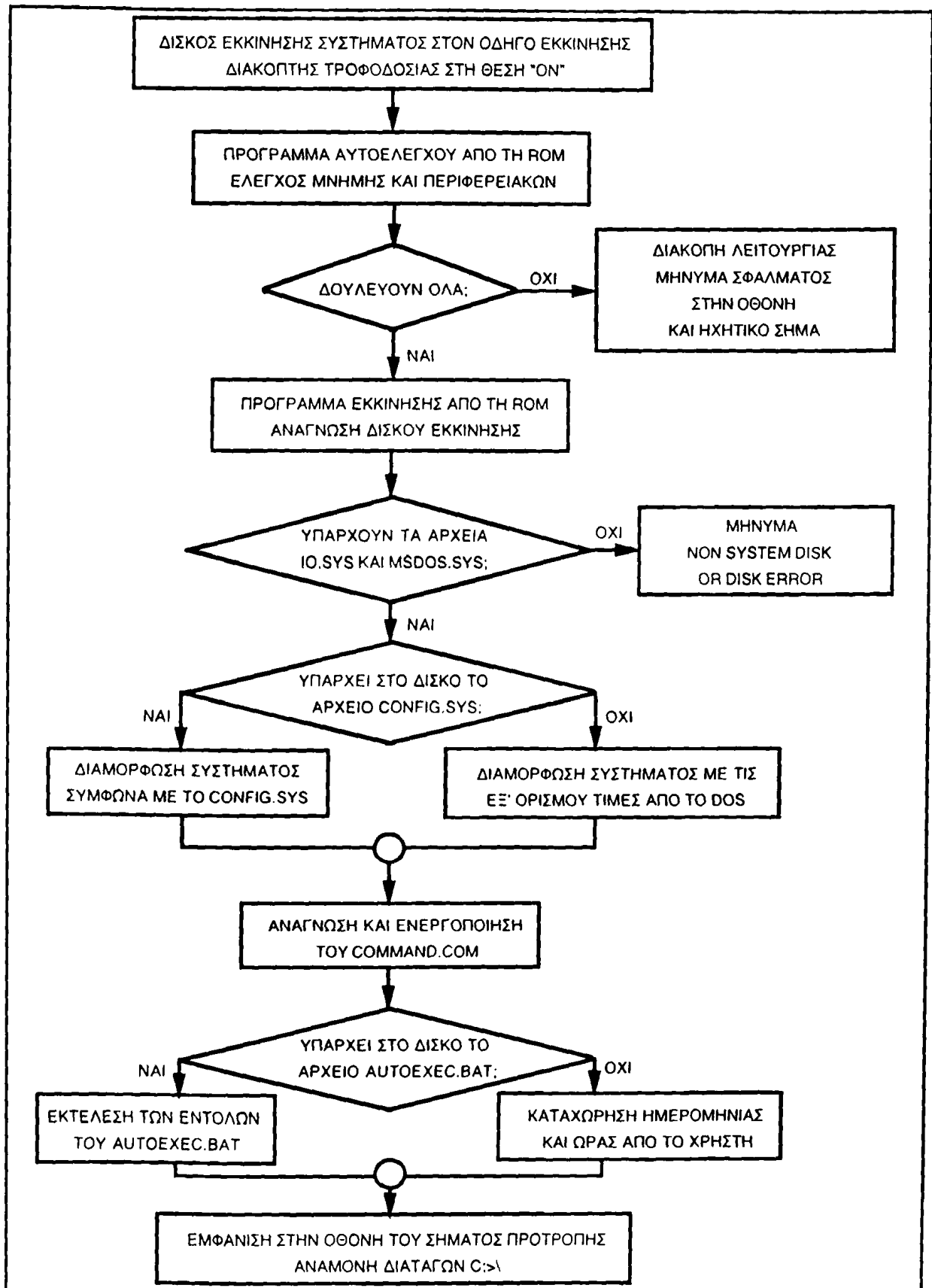
Το λειτουργικό σύστημα συνδέει τα διαφορετικής πολυπλοκότητας επίπεδα που υπάρχουν σε ένα υπολογιστικό σύστημα, σε μια ολότητα. Έτσι μερικά τμήματα του που σχετίζονται απευθείας με το υλικό δίνουν αποτελέσματα σε χρόνο της τάξης του δισεκατομμυριοστού του δευτερολέπτου, ενώ άλλα που δέχονται τις επιδράσεις του χρήστη έχουν ρυθμούς της τάξης του δευτερολέπτου.

Για την αντιμετώπιση της πολυπλοκότητας και την οργάνωση του λειτουργικού συστήματος, η σχεδίαση του βασίζεται στη δημιουργία μιας ιεραρχίας επιπέδων καθένα από τα οποία χαράζει τις κατευθυντήριες γραμμές μιας εργασίας αγνοώντας τις λεπτομέρειες υλοποίησής τους. Αυτήν την αναλαμβάνουν τα επίπεδα που βρίσκονται χαμηλότερα στην ιεραρχία. Το ανώτατο επίπεδο αυτής της δομής το κατέχει ο χρήστης που δε χρειάζεται να γνωρίζει τον τρόπο που το μηχάνημα πραγματοποιεί τις απαιτήσεις του.

Η ιεραρχική δομή ενός σύγχρονου λειτουργικού συστήματος κατανέμει τις λειτουργίες του ανάλογα με την πολυπλοκότητά τους, την κλίμακα χρόνου εκτέλεσής τους και το δικό του βαθμό αφαίρεσης [Λαδιάς, Μικρόπουλος, 1988].

Στο σχήμα 4.1 παρουσιάζεται η διαδικασία εκκίνησης ενός μικροϋπολογιστή με το λειτουργικό σύστημα MS-DOS, με τις διεργασίες που εκτελεί ώστε ο υπολογιστής να βρεθεί σε κατάσταση ετοιμότητας για λειτουργία. Παρότι το DOS είναι ξεπερασμένο, η διαδικασία που περιγράφεται στο σχήμα δεν παύει να ισχύει γενικά.





Σχήμα 4.1 Η διαδικασία εκκίνησης υπολογιστή με το λειτουργικό σύστημα DOS



Μεταγλωττιστές

Τα προγράμματα που εισάγονται στον υπολογιστή και είναι γραμμένα σε οποιαδήποτε γλώσσα εκτός της “μητρικής” του πρέπει να μεταφραστούν σε γλώσσα μηχανής, σε δυαδικό κώδικα κατανοητό και εκτελέσιμο από τον υπολογιστή.

Η διαδικασία κατά την οποία το αρχικό πρόγραμμα αποθηκεύεται, αναγνωρίζονται τα επιμέρους στοιχεία του και μετασχηματίζεται σε ένα νέο πρόγραμμα σε γλώσσα μηχανής χωρίς την αλλοίωση του αλγόριθμου, λέγεται μετάφραση.

Το αρχικό πρόγραμμα αναφέρεται ως πρόγραμμα-πηγή ή πρόγραμμα πηγαίου κώδικα (source code program) και το τελικό ως αντικείμενο πρόγραμμα ή αντικείμενος κώδικας (object code). Το αντικείμενο πρόγραμμα παράγεται μόνο στην περίπτωση που το πηγαίο δεν εμφανίζει συντακτικά λάθη και ασάφειες, γεγονός που συμβαίνει συχνά. Μόλις ο μεταφραστής (compiler) επισημάνει λάθη, τα εντοπίζει και ενημερώνει τον προγραμματιστή που τα διορθώνει.

Ένας διαφορετικός τρόπος μεταγλώττισης είναι αυτός του διερμηνέα (interpreter) που όπως φαίνεται και ετυμολογικά, μεταγλωττίζει το αρχικό πρόγραμμα σε δυαδικό κώδικα βήμα προς βήμα. Σε αντίθεση με το μεταφραστή ο διερμηνέας δε δημιουργεί αντικείμενο πρόγραμμα και αναλαμβάνει την εισαγωγή του προγράμματος, τη μεταγλώττιση και την επίβλεψη της εκτέλεσης του. Βρίσκεται σε διαρκή επικοινωνία με τον προγραμματιστή και τον ενημερώνει για πιθανά λάθη αμέσως μετά τη μεταγλώττιση των εντολών μιας προς μια. Το βασικό μειονέκτημα του είναι ότι δεν αποθηκεύει τις μεταγλωττισμένες εντολές και επομένως κάθε νέα εκτέλεση του προγράμματος σημαίνει και διερμηνεία του.

Σε περίπτωση που το αρχικό πρόγραμμα είναι γραμμένο σε συμβολική γλώσσα για τη μεταγλώττιση του χρησιμοποιείται ένας εξειδικευμένος μεταφραστής, ο συμβολομεταφραστής (assembler) με χαρακτηριστικό του την αντιστοίχηση μιας εντολής του για κάθε μια της γλώσσας μηχανής.

Για κάθε γλώσσα υψηλού επιπέδου ή συμβολική συγκεκριμένης μηχανής, υπάρχει και ο κατάλληλος μεταφραστής.

4.2 Προγράμματα εφαρμογών

Τα προγράμματα εφαρμογών χωρίζονται σε δύο μεγάλες κατηγορίες. Τις γλώσσες προγραμματισμού και τα πακέτα εφαρμογών.

Γλώσσες προγραμματισμού

Γλώσσα προγραμματισμού είναι ο αυστηρός και πλήρης τρόπος αναπαράστασης των αλγορίθμων για την απευθείας εισαγωγή και εκτέλεσή τους από τον υπολογιστή.



Εξαιτίας της ακρίβειας που απαιτεί η περιγραφή του αλγόριθμου, οι γλώσσες προγραμματισμού περιλαμβάνουν και μη αλγοριθμικά στοιχεία. Έτσι κάθε γλώσσα προγραμματισμού περιλαμβάνει το διαδικαστικό κώδικα που υλοποιεί τον αλγόριθμο και τον ορισμό των δεδομένων στοιχείων. Περιέχει επίσης βοηθητικά χαρακτηριστικά όπως δυνατότητα σχολιασμού των εντολών που αποτελούν το πρόγραμμα και εντολές προς το πρόγραμμα-μεταφραστή που μεταφράζει τον κώδικα που εισάγει ο χρήστης σε κώδικα που καταλαβαίνει η μηχανή.

Οι γλώσσες προγραμματισμού διακρίνονται σε δύο μεγάλες κατηγορίες. Τις γλώσσες χαμηλού και τις γλώσσες υψηλού επιπέδου. Παρουσιάζουν μεγάλες διαφορές με γενικότερη ότι οι πρώτες είναι άμεσα "κατανοητές" από τη μηχανή, ενώ οι δεύτερες πλησιάζουν τη φιλοσοφία του ανθρώπου με τάση να μοιάζουν με τη φυσική γλώσσα και τρόπο σκέψης του. Οι γλώσσες διαβαθμίζονται σε πέντε επίπεδα ή γενιές, όπου κάθε επόμενη έπεται χρονολογικά των προηγούμενων και χρησιμοποιείται συχνότερα πιο πρόσφατα [Carson, 1990].

1. Γλώσσα μηχανής (machine language)
2. Συμβολικές γλώσσες (assembly languages)
3. Γλώσσες υψηλού επιπέδου (high - level languages)
4. Γλώσσες πολύ υψηλού επιπέδου (very high - level languages)
5. Φυσικές γλώσσες (natural languages).

Επιπλέον, υπάρχει και μια ακόμη κατηγοριοποίηση των γλωσσών σε διαδικαστικές και μη διαδικαστικές. Οι διαδικαστικές γλώσσες (procedural languages) χρησιμοποιούνται στο δομημένο προγραμματισμό και παρουσιάζουν βήμα προς βήμα με τις κατάλληλες εντολές τη διαδικασία για την επίλυση ενός προβλήματος. Οι μη διαδικαστικές γλώσσες (nonprocedural languages) δεν περιλαμβάνουν διαδικασίες και η επίλυση του προβλήματος γίνεται με την περιγραφή του τελικού στόχου χωρίς να δίνονται λεπτομέρειες σχετικά με τον τρόπο που πρέπει να επιτευχθεί. Οι μη διαδικαστικές γλώσσες είναι κυρίως αυτές που χαρακτηρίζουν την τέταρτη γενιά των γλωσσών πολύ υψηλού επιπέδου.

Η γλώσσα μηχανής είναι γλώσσα χαμηλού επιπέδου (low-level language), η μόνη που ουσιαστικά κατανοεί ο υπολογιστής. Το λεξιλόγιο ή το σύνολο των εντολών της αποτελείται από ομάδες δυαδικών αριθμών που ερμηνεύονται απευθείας από τον αποκωδικοποιητή προγράμματος. Χαρακτηριστικό της είναι ότι εξαρτάται άμεσα από το είδος του υπολογιστή στον οποίο γράφεται το πρόγραμμα και περιγράφει συγκεκριμένες εργασίες που αφορούν τμήματα του υλικού του μέρους. Ο προγραμματιστής πρέπει να γνωρίζει την αρχιτεκτονική δομή της μηχανής όπως συγκεκριμένους καταχωρητές, τρόπους διευθυνσιοδότησης και θέσεις μνήμης. Επίσης σε ένα πρόγραμμα σε γλώσσα μηχανής που είναι μια σειρά από δυαδικούς αριθμούς, είναι δύσκολο να κατανοηθεί ο αλγόριθμος που περιγράφεται.

Ο προγραμματισμός σε γλώσσα μηχανής συνηθίζονταν στην περίοδο ανάπτυξης



της τεχνολογίας των υπολογιστών που προγραμματιστές ήταν οι ίδιοι οι τεχνικοί. Στη σύγχρονη εποχή τέτοιου είδους προγραμματισμός γίνεται μόνο σε ειδικές περιπτώσεις που απαιτούν άμεσο και γρήγορο έλεγχο διεργασιών και περιφερειακών που συνδέουν τον υπολογιστή με τον έξω κόσμο.

Για την απομάκρυνση από τη φιλοσοφία της μηχανής με τους περιορισμούς που θέτει, αναπτύχθηκαν οι συμβολικές γλώσσες. Αυτές αν και μηχανοστραφείς αντικαθιστούν το δυαδικό κώδικα με συμβολικές λέξεις, τα μνημονικά (mnemonics), που συνήθως είναι συντμήσεις αγγλικών λέξεων που θυμίζουν τη λειτουργία της κάθε εντολής. Μια γραμμή σε συμβολική γλώσσα αντιστοιχεί σε μια γραμμή γλώσσας μηχανής.

Με τη διάδοση των υπολογιστών εμφανίζεται η ανάγκη για γλώσσες προγραμματισμού που αφενός πλησιάζουν τον τρόπο σκέψης του ανθρώπου και αφετέρου δεν εξαρτώνται από το είδος της μηχανής στην οποία αναπτύσσονται τα προγράμματα. Έτσι εμφανίζονται οι γλώσσες υψηλού επιπέδου. Χαρακτηριστικά τους αποτελούν η απαίτηση ενός μεταφραστή από αυτές στη γλώσσα μηχανής, η αντιστοίχιση μιας εντολής τους με περισσότερες της γλώσσας μηχανής και ο καθορισμός ενός αλγόριθμου από το χρήστη για τη λύση του προβλήματος. Με τις γλώσσες υψηλού επιπέδου, ο προγραμματιστής δεν απασχολείται πλέον με καταχωρητές και διαύλους. Μαθηματικοί τελεστές όπως του εκθετικού και του πολλαπλασιασμού είναι ενσωματωμένοι στο λεξιλόγιο των γλωσσών. Παρέχονται επίσης έτοιμες οι δυνατότητες για τη διαχείριση δομών δεδομένων όπως οι πίνακες και τα αρχεία.

Στις φυσικές γλώσσες της πέμπτης γενιάς, ο χρήστης δίνει εντολές στον υπολογιστή χρησιμοποιώντας προτάσεις της φυσικής γλώσσας. Περιλαμβάνουν μια βάση γνώσης (knowledge - based system) που αναγνωρίζει γραμματικά και συντακτικά λάθη και μεταφράζει τις προτάσεις σε κώδικα κατανοητό από τον υπολογιστή. Συστήματα βάσεων γνώσεων οδηγούν στα έμπειρα συστήματα (expert systems) που ειδικεύονται σε συγκεκριμένα είδη εφαρμογών και ανοίγουν το δρόμο προς την τεχνητή ευφυΐα (artificial intelligence).

Από την εμφάνισή τους στη δεκαετία του 1960 μέχρι και σήμερα έχουν αναπτυχθεί αρκετές εκατοντάδες γλωσσών προγραμματισμού υψηλού επιπέδου παρά την προσδοκία για την ύπαρξη μιας και μόνο καθολικής γλώσσας. Δύο σημαντικοί λόγοι που δικαιολογούν την κατάσταση ιδίως για τα πρώτα χρόνια της ιστορίας τους, είναι ο περιορισμός και η εξάρτηση της γλώσσας από τη μηχανή στην οποία αναπτύσσεται και το μεγάλο εύρος εφαρμογών που προσανατολίζουν τον τρόπο έκφρασης της κάθε γλώσσας.

Παρά τις διαφορές που υπάρχουν μεταξύ των πολλών αλγοριθμικών γλωσσών, κάθε πρόβλημα που μπορεί να λυθεί με υπολογιστή μπορεί να αντιμετωπισθεί κατά κανόνα από οποιαδήποτε γλώσσα υψηλού επιπέδου [Λαδιάς, 1991].

Το σχήμα 4.2 παρουσιάζει χρονολογικά τις περισσότερο διαδεδομένες γλώσσες προγραμματισμού υψηλού επιπέδου.



ΟΝΟΜΑΣΙΑ	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	ΕΦΑΡΜΟΓΕΣ
FORTRAN	1957 Ακρίβεια υπολογισμών	Επιστημονικές
COBOL	1960 Μεγάλο λεξιλόγιο	Εμπορικές
BASIC	1962 Απλή για αρχάριους	Γενικής χρήσης
LISP	1962 Επεξεργασία καταλόγων	Τεχνητή νοημοσύνη
LOGO	1970 Γεωμετοία χελώνας	Εκπαίδευση
PASCAL	1971 Δομημένη, σύγχρονη	Γενικής χρήσης
PROLOG	1972 Λογικά μοντέλα	Τεχνητή νοημοσύνη
C	1973 Αντικειμενοστραφής	Γενικής χρήσης
SMALLTALK	1980 Αντικειμενοστραφής	Εκπαίδευση
SCRIPT LANGUAGES	1980 Πακέτα εφαρμογών	Γενικής χρήσης
JAVA	1994 Αντικειμενοστραφής	Internet

Σχήμα 4.2 Γλώσσες υψηλού επιπέδου, χρόνος δημιουργίας, χαρακτηριστικά και συχνές εφαρμογές τους

Πακέτα εφαρμογών

Μέχρι τη δεκαετία του 1960 ο κάθε χρήστης ηλεκτρονικού υπολογιστή ήταν και προγραμματιστής, γράφοντας με δικές του τεχνικές τα προγράμματα για να αντιμετωπίσει τις ανάγκες του. Δεν υπήρχε μεθοδολογία προγραμματισμού και κανόνες για την ανάπτυξη μεγάλων προγραμμάτων.

Οι διαρκώς αυξανόμενες απαιτήσεις σε διάφορους τομείς της επιστήμης, τεχνολογίας και οικονομίας οδήγησαν σε προσπάθειες ανάπτυξης τεράστιων εφαρμογών λογισμικού που απαιτούσαν ομαδική εργασία. Έτσι, το 1969 εμφανίσθηκε ένας νέος κλάδος της πληροφορικής, η τεχνολογία ή μηχανική λογισμικού (software engineering). Οι ειδικοί επιστήμονες του κλάδου, αναλυτές και προγραμματιστές, αναπτύσσουν μεγάλα πακέτα προγραμμάτων για την αντιμετώπιση κάθε προβλήματος για το οποίο ενδείκνυται η χρήση της πληροφορικής. Αυτά γενικά δεν απαιτούν γνώση προγραμματισμού από το χρήστη παρά μόνο την εκμάθηση χειρισμού τους.

Μεταξύ των έτοιμων πακέτων λογισμικού συγκαταλέγονται επιστημονικά προγράμματα, στατιστική επεξεργασία δεδομένων, βάσεις δεδομένων, εκπαιδευτικό λογισμικό, αυτοματισμός γραφείου και αυτόματος έλεγχος εργαστηρίων και παραγωγικής διαδικασίας.

Η ανάπτυξη λογισμικού οποιουδήποτε τύπου, περιέχει ορισμένα βασικά στοιχεία για την ολοκλήρωσή του [Νικολινάκος, 1985].

Την ανάλυση του προβλήματος και τον καθορισμό των απαιτήσεων του τελικού προγράμματος.



Τη σχεδίαση της συνολικής δομής, την ανάλυση σε τμήματα και τη σχεδίασή τους.
Τον προγραμματισμό κάθε τμήματος με τη γραφή του κώδικα στη γλώσσα που έχει επιλεγεί και τον έλεγχό του.

Τη δοκιμή του συνολικού πακέτου με τη βοήθεια ενός προγράμματος δοκιμής.

Την τεκμηρίωση με τη συνοδεία του πακέτου από ένα εύχρηστο εγχειρίδιο αναφοράς.

Τη συντήρηση που περιλαμβάνει τη διόρθωση λαθών και την τροποποίηση του προγράμματος για την αντιμετώπιση μελλοντικών αναγκών.

Αναφορές

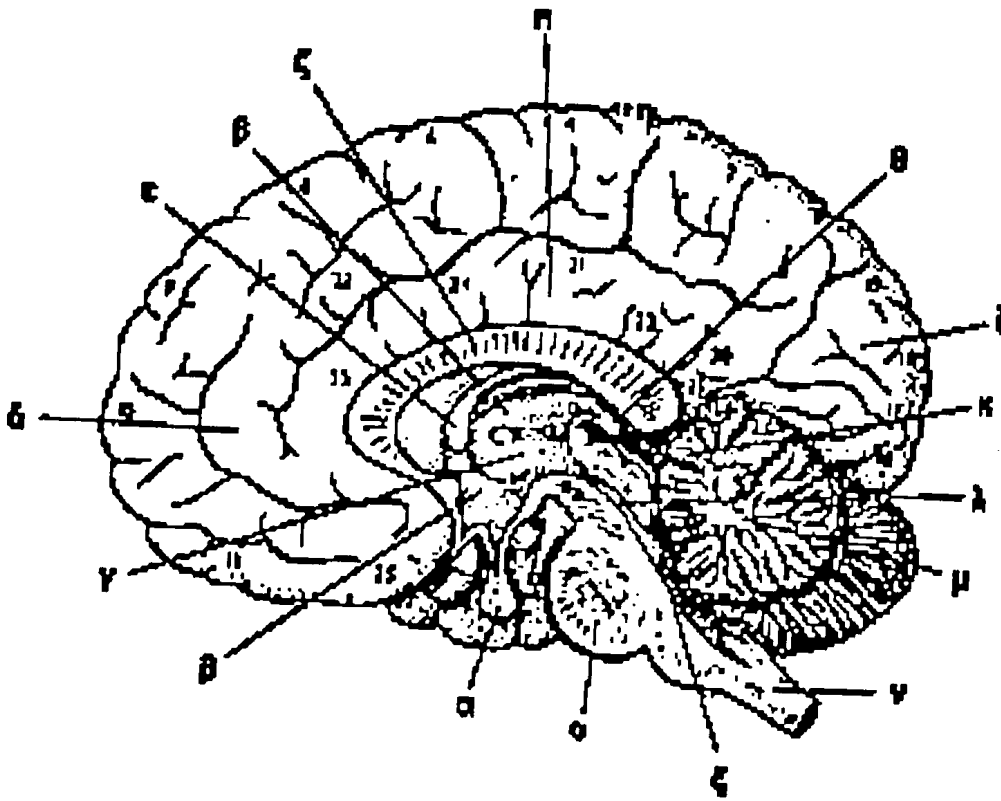
Capron H. L., "Computers. Tools for an Information Age", Benjamin/Cummings, CA 1990

Λαδιάς Τ. "Pascal. Μια Μεθοδική Προσέγγιση", Κλειδάριθμος, Αθήνα 1991

Λαδιάς Τ., Μικρόπουλος Τ., "Λειτουργικά Συστήματα", RAM, 1, 132 2/1988

Νικολινάκος Γ. "Ανάλυση και Σχεδίαση Συστημάτων", Αθήνα 1985





Ο ισχυρότερος επεξεργαστής
πραγματικού χρόνου



5 Εφαρμογές ηλεκτρονικών υπολογιστών

Όπως φαίνεται από την ιστορική εξέλιξη της επιστήμης των υπολογιστών, η τεχνολογική τους ανάπτυξη οδηγήθηκε τις περισσότερες φορές από τις στρατιωτικές απαιτήσεις, αλλά και οι πρώτες εφαρμογές της πληροφορικής αφορούν τα στρατιωτικά πράγματα.

Οι εφαρμογές αυτές σχετίζονται με υπολογισμούς τροχιών βλημάτων, αυτόματα αμυντικά συστήματα, στατιστικές αναλύσεις και οικονομικά στοιχεία. Με την ανάπτυξη μικρών και εύχρηστων υπολογιστών εμφανίζονται και εφαρμογές που καλύπτουν πεδία όπως η οικονομία, οι επιχειρήσεις, η βιομηχανία, οι επιστήμες, η ιατρική, η εκπαίδευση, οι τέχνες και η διασκέδαση. Ουσιαστικά σήμερα δεν υπάρχει τομέας στον οποίο δεν έχει εισχωρήσει η πληροφορική ή δεν έχει ζητηθεί η αρωγή της.

5.1 Τομείς εφαρμογής

Σχεδόν όλα τα πεδία εφαρμογής της πληροφορικής μοιράζονται κοινές εξελίξεις της τεχνολογίας των υπολογιστών που τις εξειδικεύουν ανάλογα με τις ανάγκες τους. Οι βασικότερες είναι οι προσομοιώσεις, τα γραφικά και η “αλληλεπίδραση” ανθρώπου-μηχανής. Σ’ αυτές συμπεριλαμβάνονται οι μιμήσεις και μοντελοποιήσεις φαινομένων και καταστάσεων, ο αυτόματος έλεγχος περιβάλλοντος εργασίας, η επιτραπέζια τυπογραφία, η επεξεργασία εικόνας, οι βάσεις δεδομένων, και οι επικοινωνίες.

Για τους παραπάνω λόγους οι τομείς εφαρμογής των υπολογιστών διακρίνονται σε τέσσερις κατηγορίες ανάλογα με την αρχιτεκτονική των συστημάτων που χρησιμοποιούνται.

1. Συστήματα υπολογιστών
2. Καταναλωτικές συσκευές
3. Βιομηχανικές εφαρμογές
4. Ειδικές εφαρμογές.

Οι πλέον διαδεδομένοι εκπρόσωποι των συστημάτων υπολογιστών είναι οι προσωπικοί υπολογιστές (Personal Computers, PCs). Οι περισσότεροι είναι μικροϋπολογιστές μιας κάρτας, της μητρικής πλακέτας (motherboard), στην οποία βρίσκεται ο επεξεργαστής και υπάρχει δυνατότητα σύνδεσης διαφόρων συσκευών εισόδου/εξόδου. Χαρακτηριστικό των προσωπικών υπολογιστών αποτελεί η ευελιξία τους σε πολύ μεγάλο φάσμα εφαρμογών με τη χρησιμοποίηση του κατάλληλου λογισμικού ή και τη διασύνδεση εξειδικευμένου υλικού.

Η πλειονότητα των καταναλωτικών συσκευών χρησιμοποιεί ή και βασίζεται σε



μικροϋπολογιστές. Ο τύπος της συσκευής και τα τεχνικά της χαρακτηριστικά, καθορίζουν και το είδος των υπολογιστών που χρησιμοποιούνται. Οι βασικές προδιαγραφές που είναι το μικρό κόστος, η μικρή απαίτηση σε ενέργεια, η αντοχή και η ευχρηστιά τους, οδήγησαν σε μικροϋπολογιστές που απαρτίζονται από ένα ή δύο τσιπ.

Οι μικροϋπολογιστές στις καταναλωτικές συσκευές ελέγχουν τον τρόπο λειτουργίας τους, παρέχουν προστασία από λάθη του χρήστη, του δικτύου τροφοδοσίας ή του περιβάλλοντος λειτουργίας, και αντικαθιστούν τον ηλεκτρομηχανικό τρόπο λειτουργίας τους. Συναντώνται σε συσκευές όπως ραπτομηχανές, φούρνοι μικροκυμάτων, τηλεοράσεις, πλυντήρια, κλιματιστικά, τηλέφωνα, μηχανήματα τηλεομοιοτυπίας, αυτοκίνητα και ηλεκτρονικά παιχνίδια [Morgan].

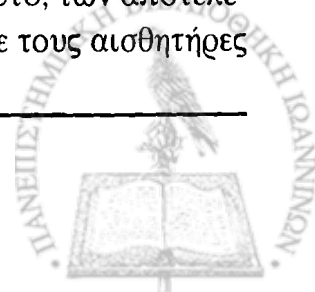
Σχετικά με τις βιομηχανικές εφαρμογές, οι μικροεπεξεργαστές αντικαθιστούν μεγάλα και πολύπλοκα συστήματα υπολογιστών και παρέχουν αυτοματοποίηση της διαδικασίας παραγωγής και αυτόματο έλεγχο του περιβάλλοντος και των συνθηκών εργασίας. Επειδή ο φυσικός κόσμος περιγράφεται από αναλογικά κυρίως μεγέθη, οι μικροϋπολογιστές πρέπει να διαθέτουν περιφερειακά τα οποία δέχονται και παράγουν τέτοιου είδους σήματα, καθώς και αναλογικές εισόδους και εξόδους [Λαδιάς, Μικρόπουλος, 1988].

Η επικοινωνία των μικροϋπολογιστών με το περιβάλλον γίνεται με τη βοήθεια των αισθητήρων και των ενεργοποιητών. Οι αισθητήρες (sensors) είναι όργανα που μετατρέπουν ένα φυσικό ή χημικό μέγεθος σε ηλεκτρικό σήμα, ενώ οι ενεργοποιητές (actuators) μετατρέπουν το ηλεκτρικό σήμα σε άλλου είδους φυσικό μέγεθος.

ΟΡΟΛΟΓΙΑ	ΕΡΜΗΝΕΙΑ
CAD	Computer Aided Design: Σχεδίαση με Η/Υ
CAM	Computer Aided Manufacturing: Κατασκευή με Η/Υ
CAE	Computer Aided Engineering: Μελέτη με Η/Υ
CAP	Computer Aided Planning: Προγραμματισμός με Η/Υ
CAT	Computer Aided Testing: Έλεγχος με Η/Υ
CAQ	Computer Aided Quality assurance: Ποιοτικός έλεγχος με Η/Υ
CIM	Computer Integrated Manufacturing: Παραγωγή με Η/Υ

Σχήμα 5.1 Τεχνικές που υποστηρίζονται από ηλεκτρονικό υπολογιστή (Η/Υ)

Απαραίτητο στοιχείο σε ένα σύστημα αυτόματου ελέγχου αποτελεί η ανάδραση (feedback) ή ανατροφοδότηση, που σημαίνει ότι ο έλεγχος μιας μηχανής γίνεται με βάση την πραγματική της συμπεριφορά και όχι την αναμενόμενη. Η ανάδραση είναι μια μέθοδος ελέγχου ενός συστήματος με την επανακαταχώρηση σ' αυτό, των αποτελεσμάτων της προηγούμενης συμπεριφοράς του. Το σύστημα συλλέγει με τους αισθητήρες



του στοιχεία από το περιβάλλον, ενεργεί σ' αυτό και το μεταβάλλει. Στη συνέχεια το σύστημα πληροφορείται με τον ίδιο τρόπο για τα αποτελέσματα της δράσης του και τροποποιεί τη συμπεριφορά του ώστε να επιφέρει την επιθυμητή κατάσταση λειτουργίας στο περιβάλλον. Στις εφαρμογές που σχετίζονται κυρίως με τον αυτοματισμό και τη ρομποτική συναντώνται σε πολλές από τις φάσεις ολοκλήρωσης του ολικού συστήματος τεχνικές, που υποστηρίζονται από υπολογιστές (Computer Aided, CA) (σχήμα 5.1).

Οι ειδικές εφαρμογές απαιτούν συνήθως επεξεργαστές με εξειδικευμένη σχεδίαση και κατασκευή, εξαιτίας των περιορισμών που προκύπτουν από το είδος και το περιβάλλον της εργασίας που επιτελούν. Συναντώνται σε στρατιωτικές, διαστημικές, ιατρικές και επιστημονικές χρήσεις. Κύριο χαρακτηριστικό τους είναι η αντοχή σε ειδικές συνθήκες λειτουργίας (ακτινοβολίες, μεγάλες επιταχύνσεις, διάστημα, ανθρωπινο σώμα).

Οι βηματοδότες που εμφυτεύονται στο ανθρώπινο σώμα περιέχουν μικροεπεξεργαστή ο οποίος με τη βοήθεια αισθητήρων ελέγχει την κυκλοφορία του αίματος και διεγείρει την καρδιά ανάλογα με τον ρυθμό της αναπνοής ώστε να μεταβάλλεται και ο ρυθμός άντλησης του αίματος.

5.2 Εφαρμογές

Οι περισσότερες εφαρμογές των υπολογιστών συναντώνται σε περισσότερους από έναν κλάδους και συνδυάζονται με τις ιδιαίτερες εφαρμογές του για την επίτευξη του τελικού στόχου. Τέτοιου είδους πακέτα λογισμικού γενικού ενδιαφέροντος είναι οι προσομοιώσεις, τα γραφικά, ο αυτοματισμός γραφείου, οι τράπεζες πληροφοριών, οι επικοινωνίες.

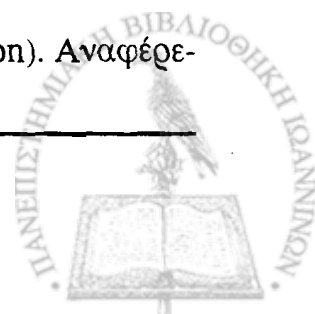
Προσομοίωση

Η πρόοδος στην επιστήμη, την τεχνολογία αλλά και την επιστημονική πλέον προσέγγιση της γνώσης, έχουν μεταβάλλει το ρόλο του ηλεκτρονικού υπολογιστή. Δεν είναι πλέον μόνο ένα μέσο για την ηλεκτρονική επεξεργασία δεδομένων που βασίζονται στην εκτέλεση συγκεκριμένων αλγορίθμων για τον υπολογισμό των επιθυμητών αποτελεσμάτων.

Ο υπολογιστής αποτελεί ένα μέσο προσομοίωσης, δημιουργίας και ελέγχου μοντέλων των δραστηριοτήτων που συμβαίνουν στο φυσικό περιβάλλον.

Η προσομοίωση (simulation) ορίζεται ως η μίμηση της συμπεριφοράς ενός συστήματος από ένα άλλο σύστημα. Η μίμηση μπορεί να πάρει τη μορφή ενός μοντέλου τελείως αφηρημένου που μπορεί να το χειριστεί υπολογιστής.

Ειδική περίπτωση της προσομοίωσης αποτελεί η εξομοίωση (emulation). Αναφέρε-

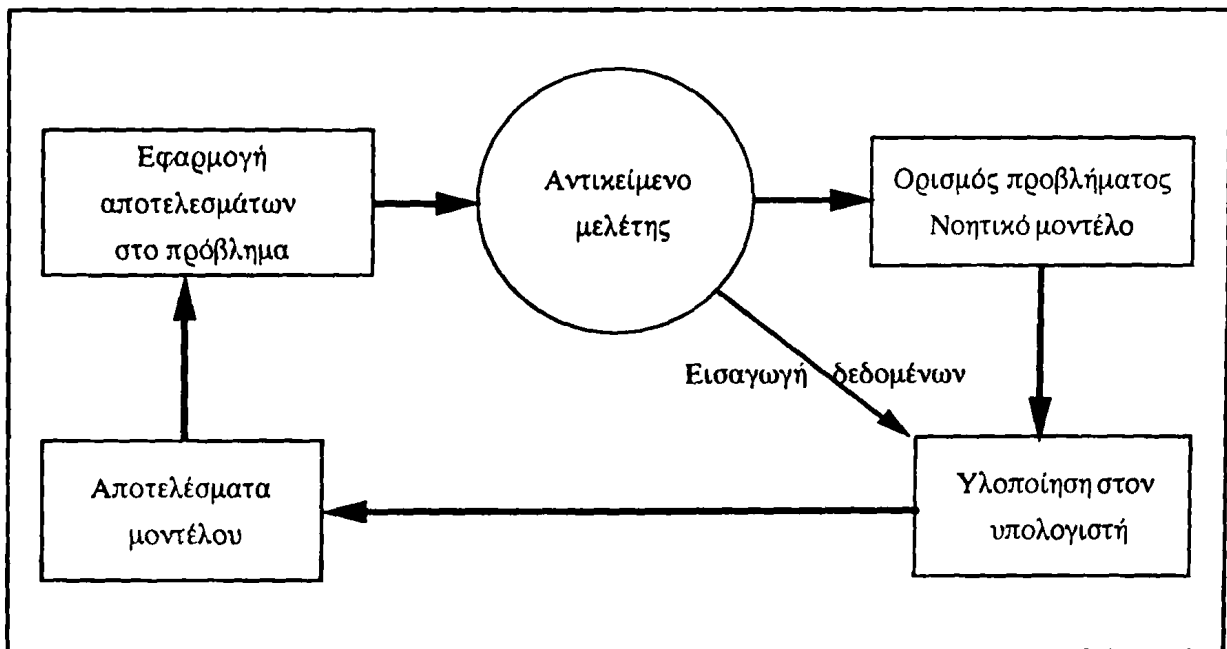


ται στη μίμηση ενός συστήματος υπολογιστή ή μέρους του από έναν άλλο υπολογιστή, με τρόπο ώστε το σύστημα που κάνει τη μίμηση να εκτελεί τα ίδια προγράμματα, να δέχεται ταυτόσημα δεδομένα και να παράγει ταυτόσημα αποτελέσματα με το πρωτότυπο σύστημα.

Οι νόμοι της φύσης περιγράφονται με αλγόριθμους ή διαδικασίες που περιγράφουν τη συμπεριφορά των διαφόρων συστημάτων. Φυσικά φαινόμενα και μαθηματικές δομές μπορούν να αναπαρασταθούν με αριθμούς και σύμβολα που τα διαχειρίζεται το κατάλληλο πρόγραμμα σύμφωνα με τους δεδομένους αλγόριθμους. Κατά την εκτέλεση του προγράμματος οι αριθμοί και τα σύμβολα τροποποιούνται ακολουθώντας τους νόμους που έχουν καταγραφεί και εμφανίζουν ως αποτελέσματα τις συνέπειες τους.

Η εκτέλεση ενός προγράμματος είναι διεργασία παρόμοια με τη διεξαγωγή ενός πειράματος. Σε αντίθεση όμως με τα αντικείμενα που πραγματεύεται το πείραμα, όσα συμβαίνουν σε ένα πείραμα που πραγματοποιείται στον υπολογιστή δεν περιορίζονται από τους φυσικούς νόμους. Έτσι παρέχεται η δυνατότητα για εκτέλεση υποθετικών πειραμάτων για τον έλεγχο θεωριών.

Το σχήμα 5.2 παρουσιάζει μια διαγραμματική εικόνα της προσομοίωσης των δραστηριοτήτων του φυσικού κόσμου.










Σχήμα 5.2 Η διαδικασία της προσομοίωσης των δραστηριοτήτων του φυσικού κόσμου

Χαρακτηριστικό της προσομοίωσης αποτελεί η αφαίρεση. Κατά την ανάπτυξη του μοντέλου για την αναπαράσταση μιας συγκεκριμένης διεργασίας, αγνοούνται οι λιγότερο ενδιαφέροντες παράγοντες και επιδράσεις και μελετώνται αυτοί που θεωρούνται για τη συγκεκριμένη περίπτωση σημαντικοί.



Μια προσομοίωση μπορεί να είναι αιτιοκρατική, δηλαδή το αποτέλεσμα της είναι μονοσήμαντο από τη στιγμή που έχουν ορισθεί όλες οι παράμετροι εισόδου. Μπορεί επίσης να είναι στοχαστική, κατά την οποία υπάρχει πιθανότητα το αποτέλεσμα της να είναι διαφορετικό για κάθε φορά που εκτελείται.

φυσικό πρόβλημα	ψευδοκώδικας	πρόγραμμα Pascal	πρόγραμμα Logo
	Το αριστερό ποτήρι έχει λεμονάδα και το δεξί βουσινάδα	Procedure SWAP (ari, dex: INTEGER); Var mes: INTEGER;	TO SWAP :ari :dex
 	Χύνουμε τη λεμονάδα στο μεσαίο ποτήρι	BEGIN mes := ari ;	MAKE "mes :ari
 	Χύνουμε τη βουσινάδα στο αριστερό ποτήρι	ari := dex ;	MAKE "ari :dex
 	Χύνουμε τη λεμονάδα από το μεσαίο στο δεξί ποτήρι	dex := mes ; END;	MAKE "dex :mes END

Σχήμα 5.3 Αντιμετάθεση των τιμών δύο μεταβλητών με τη βοήθεια προσομοίωσης φυσικού προβλήματος.

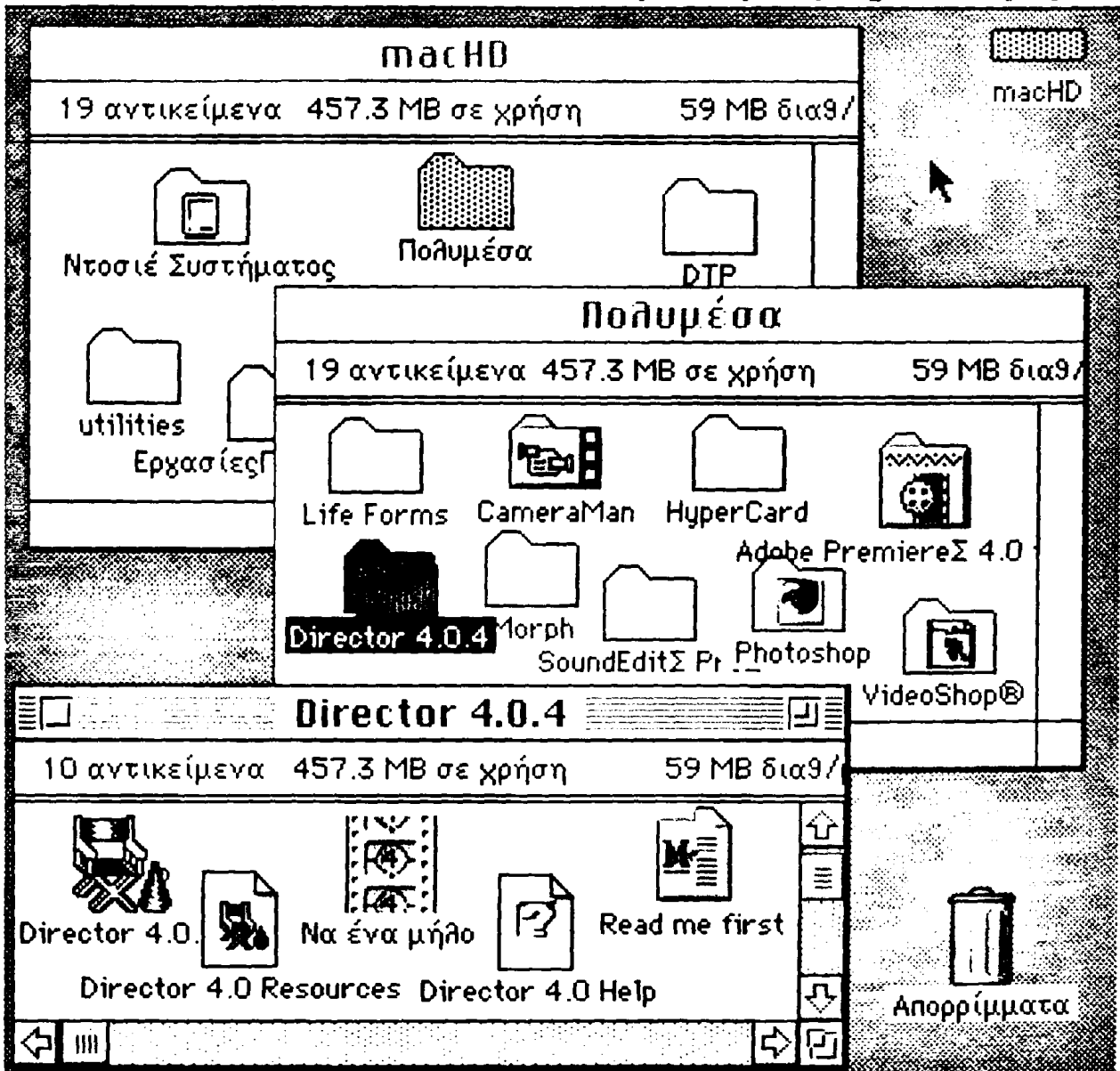
Παράδειγμα στοχαστικής προσομοίωσης είναι ένα μοντέλο που αφορά το στρίψιμο ενός νομίσματος, όπου υπάρχουν ίσες πιθανότητες να έρθει “κεφαλή” ή “γράμματα” όσες φορές και αν επαναληφθεί η διαδικασία. Η υλοποίηση στον υπολογιστή σε μια γλώσσα υψηλού επιπέδου γίνεται με τη χρήση μιας συνάρτησης που παράγει τυχαίους αριθμούς και την επανάληψη της όλης διαδικασίας πολλές φορές ώστε να περιέχεται ο στατιστικός έλεγχος. Στο σχήμα 5.3 παρουσιάζεται το κλασικό πρόβλημα αντιμετάθεσης των τιμών (περιεχομένων) δύο μεταβλητών. Η λύση στον προγραμματισμό γίνεται με τη βοήθεια της προσομοίωσης του προβλήματος με την αντιμετάθεση των περιεχομένων δύο ποτηριών με λεμονάδα και βουσινάδα [Λαδιάς, 1992].



Γραφικά

Το 1965 στην ομιλία του με τίτλο “Το απόλυτο μέσο παρουσίασης” (The Ultimate Display) στο συνέδριο του διεθνούς οργανισμού για την επεξεργασία των πληροφοριών, ο πρωτοπόρος των γραφικών Ιβάν Σάδερλαντ (Ivan Sutherland) τονίζει τη σπουδαιότητα του γραφικού τρόπου επικοινωνίας του ανθρώπου με τη μηχανή και παρομοιάζει την οθόνη των γραφικών με ένα παράθυρο στον κόσμο των θαυμάτων της Αλίκης [Sutherland, 1965].

Αρχείο Επεξ/σία Απεικόνιση Χαρ/σμός Διάφορα



Σχήμα 5.4 Τυπική οθόνη γραφικού περιβάλλοντος εργασίας

Συγκεκριμένα ως προς το μέσο παρουσίασης των γραφικών και τους διάφορους τρόπους επικοινωνίας ανθρώπου-μηχανής, αναφέρει ότι αυτό θα μπορούσε να είναι



ένα δωμάτιο μέσα στο οποίο ο υπολογιστής θα μπορεί να ελέγχει την ύπαρξη της ύλης. Μια καρτέλα που θα παρουσιάζεται σε ένα τέτοιο δωμάτιο θα μοιάζει τόσο πραγματική ώστε να μπορεί να καθίσει κάποιος σ' αυτήν.

Τα γραφικά με υπολογιστή (computer graphics) περιλαμβάνουν τη δημιουργία και διαχείριση σχεδίων και εικόνων με υπολογιστή και η θεωρία του Σάδερλαντ γίνεται πραγματικότητα με την τρομερή ανάπτυξη του υλικού και λογισμικού σ' αυτόν τον τομέα.

Τα γραφικά μπορεί να είναι δύο ή τριών διαστάσεων. Οι διδιάστατες απεικονίσεις είναι οι απλούστερες με κυριότερα παραδείγματα τις γραφικές παραστάσεις. Η απαίτηση για αναπαράσταση του φυσικού χώρου οδηγεί σε τεχνικές υλοποίησης της τρίτης διάστασης στη διδιάστατη οθόνη που πετυχαίνεται με την προοπτική και τη γραμμοσκίαση. Συνηθισμένα προβλήματα στα τρισδιάστατα γραφικά αποτελούν η έλλειψη ρεαλιστικότητας που την επιφέρει η επίδραση του φωτισμού των αντικειμένων και το κρύψιμο των μη ορατών πλευρών των αντικειμένων από τις διάφορες οπτικές γωνίες. Η τεχνολογία έχει προχωρήσει και σε αυτούς τους τομείς και αναπτύσσονται πολύπλοκες τεχνικές και αλγόριθμοι για την επίτευξη αληθοφανών εικόνων. Η διαχείρισή τους είναι σχετικά εύκολη και προσφέρει απεριόριστες δυνατότητες όπως αλλαγές στο χρώμα, περιστροφή, μετασχηματισμό της εικόνας και αλλαγή του μεγέθους της.

Ο κυριότερος παράγοντας για την εξάπλωση των γραφικών με υπολογιστή βασίζεται στο κινεζικό ρητό "μία εικόνα αξίζει όσο χίλιες λέξεις", δηλαδή στη φιλικότητα που παρέχουν κατά την αλληλεπίδραση ανθρώπου-μηχανής.

Σπουδαιότερη εφαρμογή της παραπάνω φιλοσοφίας αποτελούν τα γραφικά περιβάλλοντα εργασίας, που πλαισιώνουν τα λειτουργικά συστήματα όλων σχεδόν των τύπων των υπολογιστών και βασίζονται στην εξομοίωση της οθόνης με την επιφάνεια ενός γραφείου (desktop). Σ' αυτήν υπάρχουν όλα τα απαραίτητα όργανα και αντικείμενα για κάθε εργασία με τη μορφή παραλληλόγραμμων περιοχών, των παραθύρων (windows). Μέσα στα παράθυρα γίνεται η διαχείριση των εφαρμογών λογισμικού και των εγγράφων τους. Η επιφάνεια του γραφείου είναι όπως η αντίστοιχη ενός πραγματικού. Μπορούν να μετακινηθούν αντικείμενα πάνω της, να προστεθούν νέα ή να απομακρυνθούν άλλα και να μεταφερθούν δεδομένα και πληροφορίες από μια εφαρμογή σε άλλες. Το σχήμα 5.4 δείχνει μια τυπική οθόνη ενός γραφικού περιβάλλοντος εργασίας.

Τα γραφικά με υπολογιστή συναντώνται σε πολλών ειδών εφαρμογές. Τα παιδιά αλλά και οι ενήλικες μαθαίνουν να κατανοούν και να αντιδρούν σε πολλά ερεθίσματα μέσω των γραφικών ηλεκτρονικών παιχνιδιών και εκπαιδευτικών προγραμμάτων βασισμένων σε οπτικοποίηση (visualization) δεδομένων και πληροφοριών που περιέχουν κινούμενα σχέδια και αλληλεπίδραση με το χρήστη.

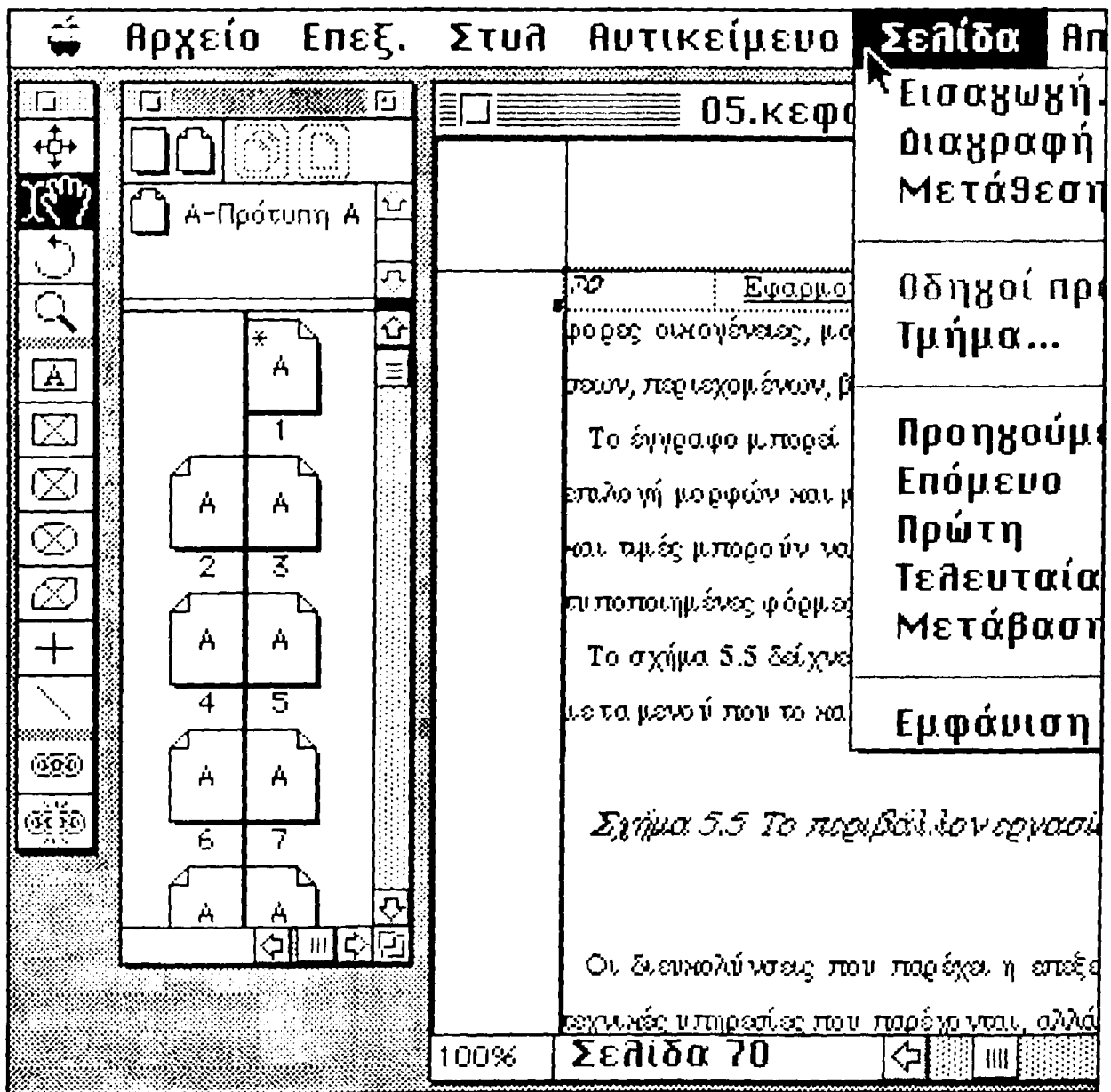
Καλλιτέχνες και τεχνικοί παράγουν ειδικά οπτικά εφέ βασισμένα εξ' ολοκλήρου σε τεχνικές γραφικών με υπολογιστή για διαφημίσεις και ταινίες επιστημονικής φαντασίας.



Αυτοματισμός γραφείου

Ο αυτοματισμός γραφείου ή ηλεκτρονικό γραφείο, περιλαμβάνει την αυτοματοποίηση των εργασιών που συνηθίζονται σε ένα γραφείο, τη διαχείριση στοιχείων και εγγράφων και την επεξεργασία δεδομένων με ηλεκτρονικό υπολογιστή. Αποτελεί μέρος της προσπάθειας για το γραφείο χωρίς χαρτιά, αν και η ευχρηστία που παρέχει ο υπολογιστής οδηγεί στο αντίθετο αποτέλεσμα βασιζόμενο στην ευκολία της λήψης και παρουσίασης των πληροφοριών στο τελικό μέσο που είναι το χαρτί.

Εφαρμογές που συναντώνται στο ηλεκτρονικό γραφείο είναι η επεξεργασία κειμένου, τα λογιστικά φύλλα, οι βάσεις δεδομένων και οι επικοινωνίες [Λαδιάς & Παντελή, 1992].



Σχήμα 5.5 Το περιβάλλον εργασίας ενός επεξεργαστή σελίδας (DTP)



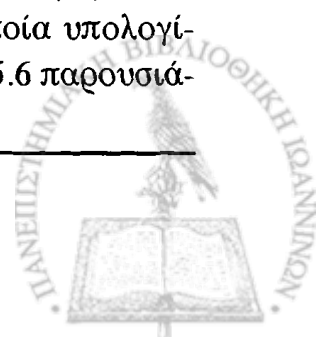
Η επεξεργασία κειμένου (word processing, wp) έχει να κάνει με τη δημιουργία και διαμόρφωση κάθε είδους εγγράφου.

Το λογισμικό υλοποίησης τέτοιων εργασιών είναι οι επεξεργαστές κειμένου (word processors) με πολλές επιλογές που αφορούν κείμενο, μαθηματικούς τύπους, γραφικά και εικόνες, καθώς και τα προγράμματα επιτραπέζιας τυπογραφίας (Desk Top Publishing, DTP) ή ηλεκτρονικών εκδόσεων με όλες τις δυνατότητες ενός συμβατικού τυπογραφείου.

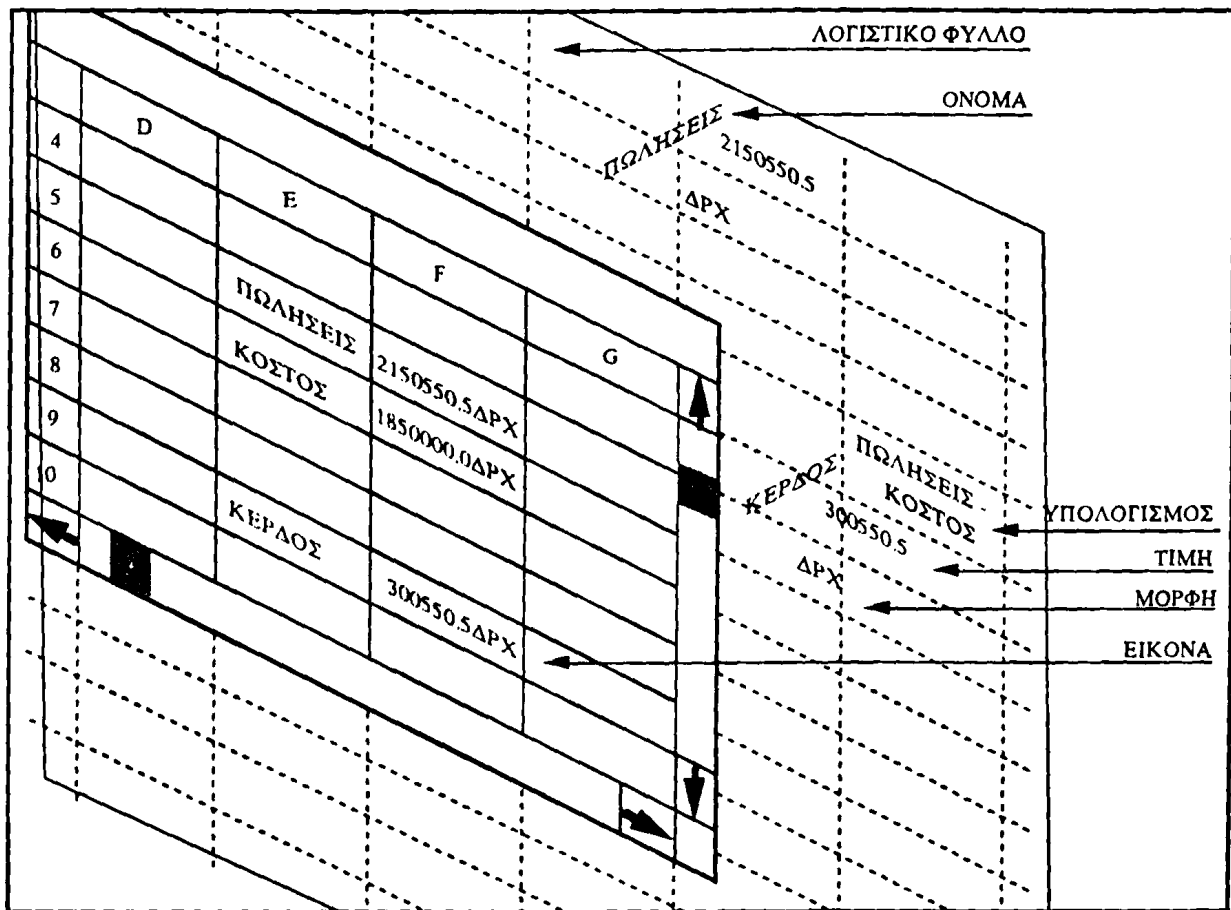
Το έγγραφο εμφανίζεται στην οθόνη του υπολογιστή καθώς εισάγεται από το πληκτρολόγιο και μπορούν να διορθωθούν, παρεμβληθούν, ή να διαγραφούν λέξεις ή και τμήματά του. Το σύστημα παρέχει δυνατότητες καθορισμού των περιθωρίων των σελίδων, στοίχισης του κειμένου με πολλούς τρόπους, αυτόματου χωρισμού των λέξεων στο τέλος της γραμμής και ελέγχου της ορθογραφίας. Επίσης γίνεται αναζήτηση και αντικατάσταση χαρακτήρων και λέξεων, προσθήκη κειμένου και εικόνων από άλλα αρχεία, γραφή με διάφορες οικογένειες, μορφές και μεγέθη τυπογραφικών στοιχείων, εισαγωγή υποσημειώσεων, περιεχομένων, βιβλιογραφίας και αυτόματης αρίθμησης των σελίδων. Το έγγραφο μπορεί να αποθηκευθεί για μελλοντική χρήση και να τυπωθεί με ορισμένη επιλογή μορφών και μεγεθών χαρτιού. Επίσης πληροφορίες όπως ονόματα, διευθύνσεις και τιμές μπορούν να αντικατασταθούν για την εκτύπωση ώστε να παράγονται εύκολα τυποποιημένες φόρμες εγγράφων. Το σχήμα 5.5 δείχνει τη μορφή της οθόνης στο περιβάλλον ενός επεξεργαστή σελίδας με τα μενού που το καθένα κρύβει μια σειρά εντολών.

Οι διευκολύνσεις που παρέχει η επεξεργασία κειμένου δεν περιορίζονται μόνο στις τεχνικές υπηρεσίες που παρέχονται, αλλά έχουν επεκτάσεις και στη διδασκαλία της γραφής και το χειρισμό του λόγου. Με τον υπολογιστή, ο μαθητής και συγγραφέας είναι δημιουργός του περιεχομένου αλλά και της μορφής του γραπτού λόγου. Το βάρος της εργασίας πέφτει στη σύνθεση του λόγου και την εκφραστική παρουσίασή του. Ο δημιουργός συγκεντρώνει τις σκέψεις του σε μια πρώτη γραπτή μορφή και ακολουθεί η γραμματική, συντακτική, νοηματική και εκφραστική επεξεργασία του κειμένου. Το κέντρο βάρους του έργου μετατοπίζεται ακόμη περισσότερο από την τεχνική στη σημασιολογική πλευρά.

Το λογιστικό φύλλο (spreadsheet) ή φύλλο εργασίας, αποτελείται από ένα σύνολο ενεργών αντικειμένων οργανωμένων σε έναν παραλληλόγραμμο πίνακα με γραμμές και στήλες που σχηματίζουν κελιά το καθένα με το δικό του σημείο αναφοράς, τη διεύθυνσή του. Στα κελιά καταχωρούνται αριθμοί ή κείμενο, καθώς και κανόνες υπολογισμού συγκεκριμένων ποσοτήτων. Κάθε φορά που η τιμή ενός κελιού μεταβάλλεται σε οποιοδήποτε σημείο του φύλλου, όλες οι τιμές σε οποιαδήποτε θέση του που εξαρτώνται από αυτό ξαναυπολογίζονται αυτόματα και εμφανίζονται οι νέες τιμές τους. Κάθε κελί μπορεί να θεωρηθεί ότι έχει αρκετά επίπεδα πίσω του τα οποία υπολογίζουν την τιμή του και καθορίζουν τη μορφή παρουσιάσής της. Το σχήμα 5.6 παρουσιάζει



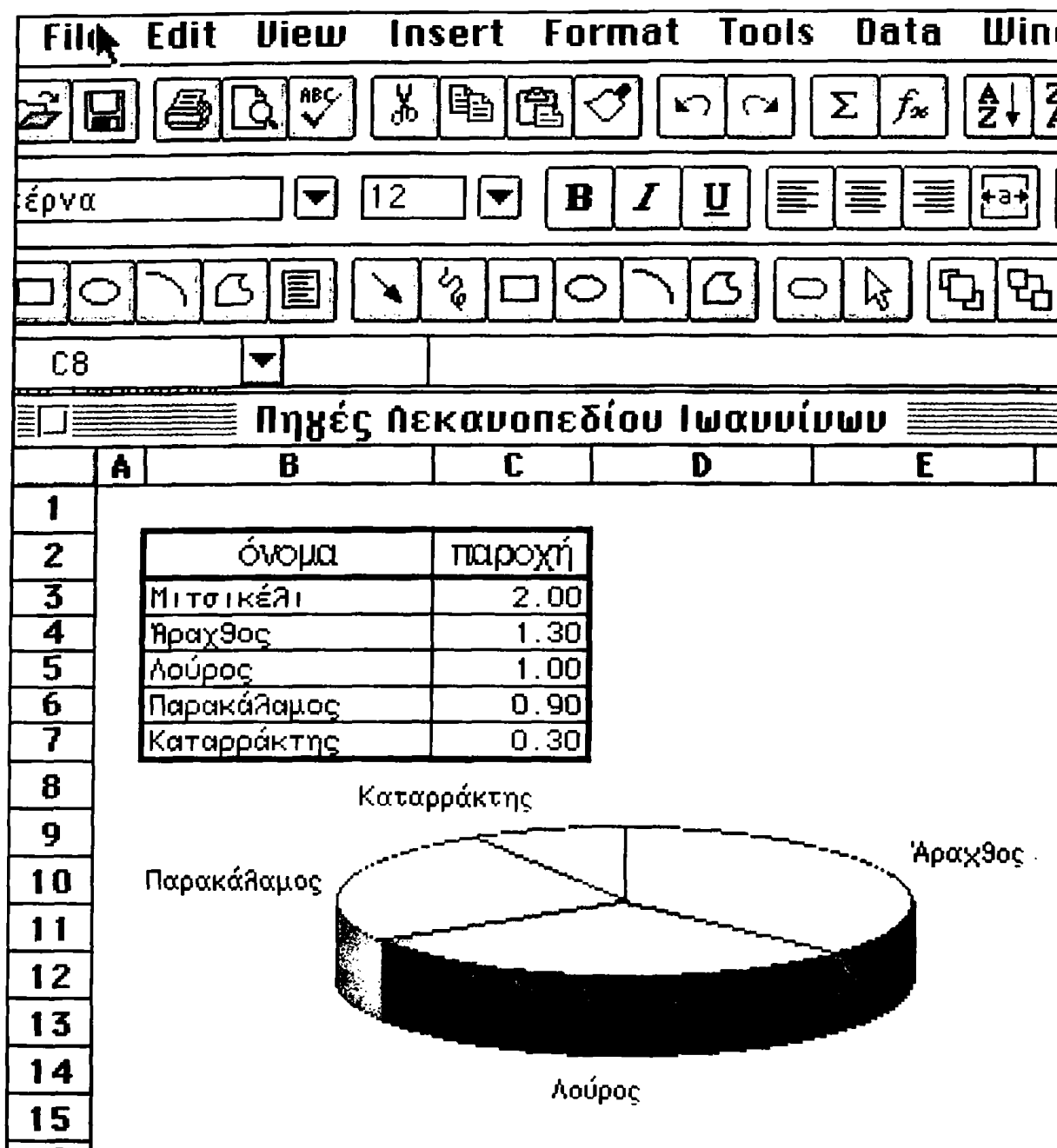
ζει τη δομή ενός λογιστικού φύλλου, τον τρόπο δημιουργίας των αποτελεσμάτων και τα αποτελέσματα όπως φαίνονται στην τελική τους μορφή.



Σχήμα 5.6 Η δομή ενός λογιστικού φύλλου. Σε πρώτο επίπεδο φαίνεται η τελική εικόνα. Τα πίσω επίπεδα που συμβολίζονται με τις διακεκομμένες γραμμές, δείχνουν τον τρόπο δημιουργίας των αποτελεσμάτων

Τα φύλλα εργασίας εκτός από τον υπολογισμό συναρτήσεων και την παρουσίαση των αποτελεσμάτων τους με αριθμητική μορφή, παρέχουν τη δυνατότητα για εποπτική παρουσίαση τους με διαγράμματα. Οι ενσωματωμένες συναρτήσεις είναι έτοιμες μαθηματικές παραστάσεις που συνήθως είναι αριθμητικής, τριγωνομετρικής, στατιστικής, οικονομικής και λογικής φύσης. Με την τροφοδοσία τιμών στο όρισμα των συναρτήσεων, τα αποτελέσματα εμφανίζονται άμεσα μετά από αυτόματη επεξεργασία. Ο διαγραμματικός τρόπος παρουσιάσής τους περιλαμβάνει ιστογράμματα, γραφικές παραστάσεις, κυκλικά διαγράμματα.

Στο σχήμα 5.7 φαίνεται ένα φύλλο εργασίας καθώς και η παρουσίαση των αποτελεσμάτων της επεξεργασίας των δεδομένων σε μορφή κυκλικού διαγράμματος.



Σχήμα 5.7 Στοιχεία λογιστικού φύλλου και γραφική παρουσίαση αποτελεσμάτων

Η διαχείριση της πληροφορίας είναι μια από τις σημαντικότερες χρήσεις των υπολογιστών και το λογισμικό γι' αυτόν τον σκοπό εξελίσσεται διαρκώς.

Η ανάπτυξη νέων τύπων βάσεων δεδομένων με πολλές δυνατότητες αναζήτησης των πληροφοριών που περιέχουν, ακολουθεί δύο βασικές αρχές. Η πρώτη σχετίζεται με το γεγονός ότι ο καλύτερος τρόπος οργάνωσης εξαρτάται από το περιεχόμενο των πληροφοριών και από τον τρόπο που θα χρησιμοποιηθούν. Ο δεύτερος συνδέεται με τη φυσική δομή του μέσου αποθήκευσης που θέτει περιορισμούς κατά την εισαγωγή και την ανάκτηση των πληροφοριών.



Η ανάγκη για την οργάνωση των δεδομένων γίνεται εμφανής σε περιπτώσεις που απαιτείται αποθήκευση και ταξινόμηση μεγάλου όγκου δεδομένων και γρήγορος εντοπισμός και ανάκτηση ορισμένων που επιλέγονται με συγκεκριμένα κριτήρια.

Τα δεδομένα οργανώνονται και αποθηκεύονται με τη μορφή αρχείων (files), που αποτελούν το ηλεκτρονικό ανάλογο του συμβατικού αρχείου ή της καρτελοθήκης και είναι δομές δεδομένων γραμμικού τύπου.

Μια ομάδα αντικειμένων που σχετίζονται μεταξύ τους σε ένα μέσο αποθήκευσης, αναφέρεται ως εγγραφή (record). Σε ένα αρχείο των φοιτητών ενός τμήματος, μια εγγραφή αφορά τα στοιχεία ενός φοιτητή και μπορεί να περιλαμβάνει το ονοματεπώνυμο, τη διεύθυνση, τον αριθμό μητρώου, το έτος φοίτησης και τον αριθμό των μαθημάτων που έχει περάσει.

Κάθε στοιχείο μιας εγγραφής, όπως το έτος φοίτησης, λέγεται πεδίο (field).

Με βάση τους παραπάνω ορισμούς, αρχείο είναι μια συλλογή από εγγραφές που αναφέρονται σε αντικείμενα με κοινά μεταξύ τους χαρακτηριστικά και έχουν μια οργάνωση που εξυπηρετεί ένα συγκεκριμένο σκοπό.

Η ανάκτηση μιας συγκεκριμένης εγγραφής από το αρχείο γίνεται με τη βοήθεια ενός κλειδιού (key). Αυτό μπορεί να είναι ένα πεδίο, μέρος από ένα πεδίο, ή και ένας συνδυασμός πεδίων.

Η βάση δεδομένων (data base) αποτελείται από ένα ή περισσότερα αρχεία και είναι η βάση για την ανάπτυξη της διακίνησης της πληροφορίας.

Ακολουθώντας τις τεχνολογικές εξελίξεις, την ανάπτυξη του λογισμικού και τις αυξανόμενες ανάγκες πληροφόρησης και επικοινωνίας, έχουν αναπτυχθεί και βρίσκονται σε ευρεία χρήση οι τράπεζες δεδομένων (data banks). Αυτές είναι ολοκληρωμένα πληροφορικά συστήματα που δίνουν απαντήσεις και στοιχεία σε προκαθορισμένες ή μη ερωτήσεις σχετικά με ένα σύνολο δεδομένων που αποτελούν μια βάση δεδομένων. Οι τράπεζες δεδομένων χαρακτηρίζονται από τις παρακάτω γενικές ιδιότητες:

1. Έχουν διάφορα επίπεδα ασφάλειας ώστε να εμποδίζεται η ανεξέλεγκτη πρόσβαση.
2. Παρέχουν πρόσβαση στους χρήστες χωρίς να απαιτείται από αυτούς γνώση της εσωτερικής οργάνωσης των δεδομένων.
3. Οι τροποποιήσεις της οργάνωσης των δεδομένων δε γίνονται αντιληπτές από τους χρήστες.

Παράδειγμα παγκόσμιων τραπεζών δεδομένων είναι αυτές που σχετίζονται με βιβλιογραφικά δεδομένα σε διάφορους τομείς, έχουν αποθηκευμένες βιβλιογραφικές αναφορές που ενημερώνονται συνεχώς και περιέχουν στοιχεία όπως τους τίτλους άρθρων, συγγραφείς, περιοδικά, ημερομηνίες δημοσίευσης και περιλήψεις των εργασιών. Η ανάκτηση της συγκεκριμένης πληροφορίας γίνεται με τη βοήθεια λέξεων-κλειδιών (keywords) που σχετίζονται με κάθε θέμα και με τη βοήθεια λογικών πράξεων που περιορίζουν τις αδιάφορες πληροφορίες και βοηθούν στον εντοπισμό των επιθυμητών στοιχείων.



Αναφορές

- Λαδιάς Τ., "BASIC Μία μεθοδική εισαγωγή στον προγραμματισμό των ηλεκτρονικών υπολογιστών", Πατάκης 1992
- Λαδιάς Τ., Μικρόπουλος Τ., "Οι Υπολογιστές στα Συστήματα Αυτόματου Ελέγχου", RAM 99 Απρίλιος 1988
- Λαδιάς Τ. & Παντελή Ε., "Γρήγορη εκμάθηση του QuarkXPress", Κλειδάριθμος 1992
- Morgan E. "Microprocessors A Short Introduction", Dept. of Industry, London
- Sutherland I.E., "The Ultimate Display", Proc. IFIP 65, 506 1965





6 Πληροφορική και εκπαίδευση

Τον Ιούνιο του 1978 ο Άλφρεντ Μπορκ [Bork, 1979] αρχίζει την ομιλία του στην Ένωση Αμερικανών Δασκάλων Φυσικής ως εξής:

“Βρισκόμαστε στο ξεκίνημα μιας σημαντικής επανάστασης στην εκπαίδευση, της σημαντικότερης από την εφεύρεση της τυπογραφίας. Το όπλο αυτής της επανάστασης θα είναι ο ηλεκτρονικός υπολογιστής. ...Μέχρι το έτος 2000, ο σημαντικότερος τρόπος μάθησης σε όλες τις βαθμίδες της εκπαίδευσης και σε όλα τα γνωστικά αντικείμενα θα είναι μέσα από τη χρήση των υπολογιστών σε αλληλεπίδραση με το χρήστη”.

Ο Μπορκ προσδιορίζει τη θέση της πληροφορικής στην εκπαίδευση, στον άμεσο και ενεργό τρόπο αλληλεπίδρασης του ανθρώπου με τη μηχανή.

Η προσφορά της πληροφορικής στην εκπαίδευση βρίσκεται πολύ μακρύτερα από τους υπολογισμούς και τις εξελίξεις της τεχνολογίας. Βρίσκεται στο νόημα της εκπαίδευσης, της μάθησης, της διδασκαλίας, της παιδαγωγικής.

6.1 Εκπαίδευση και η προσφορά της πληροφορικής

Από ιστορική πλευρά, είναι γνωστό ότι ο κυριότερος όγκος της μάθησης προέρχεται από την αλληλεπίδραση του κάθε ατόμου με το περιβάλλον του [Σιμάτος, 1995]. Η παιδική ηλικία αποτελεί το πρώτο στάδιο της εκπαίδευσης ως η φυσική συνέπεια του παιχνιδιού που προσφέρει ξεχωριστές εμπειρίες σε κάθε παιδί. Οι απαρχές λοιπόν της εκπαίδευσης δείχνουν ότι περιέχει το χαρακτήρα του προσωπικού, του ατομικού (individual) ή διαφορετικά της εξατομίκευσης.

Η αλληλεπίδραση του ατόμου με το περιβάλλον αποτελεί μια ενεργή (active) διεργασία. Οι εμπειρίες που λαμβάνει ο καθένας είναι ξεχωριστές και οδηγούν σε διαφορετικά αποτελέσματα από ότι στους άλλους.

Οι περισσότερες πληροφορίες που λαμβάνει κατά την αλληλεπίδραση είναι οπτικές (visual) και αυτές αντιλαμβάνεται και αφομοιώνει με μεγαλύτερη απόδοση.

Ένα ουσιαστικό στοιχείο στη δυναμική της μάθησης αποτελεί ο ρόλος του δασκάλου με τη Σωκρατική έννοια της διαλεκτικής.

Μεταγενέστερη κρίσιμη στιγμή στην εξελικτική πορεία της εκπαίδευσης αποτελεί η εμφάνιση των διδακτικών εγχειριδίων και ιδιαίτερα η ανάπτυξη τους με τη βοήθεια της τυπογραφίας.

Ένας από τους σπουδαιότερους στόχους της εκπαίδευσης στον οποίο μάλιστα συμβάλλει ιδιαίτερα η πληροφορική, είναι η επίλυση των προβλημάτων (problem solving). Σκοπό της επομένως αποτελεί η ανάπτυξη δεξιοτήτων από το μαθητή για την



επίτευξη αυτού του στόχου. Σημαντικό στοιχείο γι' αυτό είναι η διδασκαλία της μεθοδολογίας που οδηγεί στη λύση των προβλημάτων, σε αντίθεση με την απλή παράθεση παραδειγμάτων και παρόμοιων προς το υπό μελέτη αντικείμενο περιπτώσεων.

Η επικρατούσα κατάσταση, απέχει πολύ από την εκπαίδευση υψηλού επιπέδου όπως αυτή δηλώνεται παραπάνω. Το γεγονός οφείλεται στη δομή του εκπαιδευτικού συστήματος και στις παθητικές μεθόδους που ακολουθούνται στην εκπαιδευτική θεωρία και πράξη.

Η πλειονότητα των μαθημάτων είναι βασισμένη σε διαλέξεις οι οποίες όπως έχει δειχθεί [Bogk 1979], παρέχουν φτωχή ικανότητα στη μετάδοση της πληροφορίας. Η μελέτη εγχειριδίων αποτελεί το βασικό τρόπο μάθησης, αν και πολλοί μαθητές μαθαίνουν δύσκολα διαβάζοντας. Εξάλλου ο τρόπος γραφής των βιβλίων ακολουθεί συγκεκριμένα πρότυπα και όσα μεταφέρουν νέες διδακτικές τάσεις και προσεγγίσεις στα θέματα που πραγματεύονται δεν τυχαίνουν ευρείας αποδοχής. Το σημαντικότερο όμως αρνητικό στοιχείο της εκπαιδευτικής διαδικασίας είναι τα φτωχά εργαλεία και εποπτικά μέσα για την αντιμετώπιση και επίλυση προβλημάτων. Προβλήματα απορρέουν από τη σχεδίαση των αναλυτικών προγραμμάτων, τη διδακτική μεθοδολογία, τους διδακτικούς στόχους που θέτει ο εκπαιδευτικός, τα χαρακτηριστικά των μαθητών, την υπάρχουσα τεχνολογία, την τεχνογνωσία και την προδιάθεση γύρω από αυτήν.

Η προσφορά της πληροφορικής στην εκπαίδευση καθορίζεται από τις λύσεις που παρέχει στα παραπάνω προβλήματα κυρίως μέσα από τις δυνατότητες μεταφοράς διδακτικών τεχνικών, αλλά και την ανάπτυξη νέων εκπαιδευτικών μεθόδων. Οι τρόποι προσέγγισης του προβλήματος με τον υπολογιστή είναι πολλοί, με σπουδαιότερο κοινό χαρακτηριστικό το πλεονέκτημα του ενεργού τρόπου μάθησης μέσα από την επικοινωνία με ανάδραση με το χρήστη. Οι μαθητές από απλοί θεατές παίρνουν ενεργό μέρος στη διαδικασία μετάδοσης των πληροφοριών απαντώντας άμεσα σε ερωτήσεις, έχοντας υποδείξεις, αναζητώντας περισσότερα στοιχεία. Ουσιαστικά η αλληλεπίδραση δεν είναι μεταξύ μαθητή και μηχανής, αλλά μεταξύ μαθητή και συγγραφέων του λογισμικού. Αυτοί συνήθως απαρτίζουν μια διεπιστημονική ομάδα από ειδικούς επιστήμονες στο προς μελέτη αντικείμενο, παιδαγωγούς, ψυχολόγους, πληροφορικούς και καλλιτέχνες.

Η ενεργή συμμετοχή και αλληλεπίδραση παρέχει στο μαθητή δυνατότητα διαχείρισης του υλικού και τρόπου διδασκαλίας, αποφεύγοντας έτσι το πρόβλημα του διαφορετικού γνωστικού υπόβαθρου των μαθητών που παρατηρείται στις αίθουσες διδασκαλίας.

Με την επικοινωνία του κάθε μαθητή με τον υπολογιστή εφαρμόζεται η εξατομίκευση, το χαρακτηριστικό που συνοδεύει την εκπαίδευση από τα πρώτα στάδιά της. Ο μαθητής φεύγει από το μαζικό σύστημα παραγωγής γνώσης, τις διαλέξεις, και έχει τη δυνατότητα να συμπεριφερθεί ελεύθερα αναλύοντας τις πληροφορίες που του παρέχονται, αναζητώντας τη λύση στα προβλήματα με το ρυθμό και τον τρόπο που επιθυμεί.



Η εμπειρία ως στοιχείο μάθησης ουσιαστικά λείπει από τη διδακτική διαδικασία σε όλες τις βαθμίδες. Ο υπολογιστής ως μηχανή προσομοίωσης έχει τη δυνατότητα να προσομοιώνει φαινόμενα και καταστάσεις αλλά και να δημιουργεί νέους κόσμους σύμφωνα με τις επιθυμίες και εντολές του χρήστη. Με τέτοιου είδους διαδικασίες τηρείται ο κανόνας “πρώτα η ιδέα και μετά η ονομασία” και δίνεται η ευκαιρία στο μαθητή να περιηγηθεί και να δράσει στον κόσμο που δημιουργεί. Γνωρίζει και ελέγχει ο ίδιος οδηγούμενος στην κατανόηση εννοιών και την απόδοση ορισμών. Ιδιαίτερα στις φυσικές επιστήμες αντιμετωπίζει τα φαινόμενα χωρίς την πίεση από τους θεωρούμενους ως προϋπάρχοντες ορισμούς και μαθηματικές εκφράσεις, αλλά έχει τη δυνατότητα της φυσιολογικής διεξαγωγής τους. Έτσι αναπτύσσονται η διορατικότητα και η διαίσθηση που συμπληρώνουν τη μεθοδολογία για την επίλυση προβλημάτων [Arons, 1992].

Ο υπολογιστής ως πνευματικό εργαλείο είναι μια άλλη προσφορά της πληροφορικής που συνεισφέρει στους παραπάνω τομείς και ανοίγει νέους δρόμους στη μαθησιακή διαδικασία. Ο προγραμματισμός και ιδιαίτερα ο δομημένος, δίνει τα φόντα για ένα δομημένο τρόπο σκέψης και αντιμετώπισης προβλημάτων σε όλα σχεδόν τα γνωστικά αντικείμενα [Λαδιάς κ. α., 1989].

Ο έλεγχος των γνώσεων του μαθητή αποτελεί το κυριότερο κριτήριο για την αξιολόγηση και τη διαβεβαίωση για την επιτυχή αντιμετώπιση των προβλημάτων. Με τον υπολογιστή οι ίδιοι οι μαθητές ελέγχουν τις γνώσεις τους με άμεση ανάδραση. Οι σωστές απαντήσεις δε δηλώνονται απλά από τη μηχανή, αλλά παρουσιάζουν και συμπληρωματικές γνώσεις, εναλλακτικές λύσεις και πληροφορίες. Οι λανθασμένες δικαιολογούνται και εμφανίζεται το σκεπτικό για τη λύση, καθώς και τις απαιτούμενες γνώσεις γι’ αυτήν. Επειδή ο υπολογιστής από τη φύση του δεν έχει προκαταλήψεις, αποφεύγονται οι προσωπικοί παράγοντες μεταξύ δασκάλου και μαθητή που επηρεάζουν συνήθως την κρίση του πρώτου. Μ’ αυτόν τον τρόπο ελέγχου και αξιολόγησης, καθώς και των άλλων διαδικασιών που αφορούν μια τάξη, γίνεται άμεσα όλη η επεξεργασία των δεδομένων και πετυχαίνεται η διαχείριση της συνολικής λειτουργίας της τάξης ή και όλου του σχολείου.

Τα δομικά στοιχεία με βάση τα οποία οι σύγχρονες εκπαιδευτικές θεωρίες και τεχνικές οδηγούν στο χειρισμό των εννοιών και τη λύση προβλημάτων αποτελούν και χαρακτηριστικά γνωρίσματα των ηλεκτρονικών υπολογιστών. Εξάλλου, οι θεωρίες της εκπαίδευσης λαμβάνουν υπόψη τους την επιστημονική έρευνα στην εξέλιξη της γνώσης, τις διδακτικές μεθόδους, και την επίδραση της πληροφορικής στην τεχνολογική σκέψη της εκπαίδευσης [Bertrand, 1994].

Αρχικά γίνεται παρουσίαση και ερμηνεία των γενικών ιδεών και φυσικών αρχών που διέπουν το υπό μελέτη αντικείμενο και τις διεργασίες, μέσα από τις οποίες αναγνωρίζονται και ανακατασκευάζονται. Δεύτερο βήμα είναι η ιεραρχική οργάνωση της γνώσης για ευκολία ανάκτησης των απαραίτητων πληροφοριών. Επόμενο στάδιο απο-



τελεί η κατανοητή τοποθέτηση του προβλήματος με δυνατότητα επανεξέτασης και ελέγχου της λύσης. Τέλος είναι η δομημένη αμφίδρομη εκπαίδευση (principled instruction) που χαρακτηρίζεται από την αμοιβαία διδασκαλία υπολογιστή - μαθητή με ανάδραση.

Οι φυσικές επιστήμες έχουν εισαγάγει την επιστημονική προσέγγιση στη σύγχρονη διδακτική που επεκτείνεται σε όλα τα γνωστικά αντικείμενα.

Οι εξελίξεις στις φυσικές επιστήμες και σε επιστήμες όπως η ψυχολογία, η τεχνητή νοημοσύνη και η γλωσσολογία, έχουν οδηγήσει σε νέες αντιλήψεις για τη σύγχρονη σκέψη. Ιδιαίτερα η αλματώδης πρόοδος στην πληροφορική τεχνολογία προσφέρει νέα ισχυρά εργαλεία στην επιστημονική έρευνα και την εκπαίδευση και παρέχει σπουδαία μέσα για τον καθορισμό και την επίλυση εκπαιδευτικών προβλημάτων.

Η σύγχρονη διδακτική δίνει μεγάλη βαρύτητα στην επιστημονική προσέγγιση της διδασκαλίας των φυσικών επιστημών. Ο F. Reif [Reif, 1986] θεωρεί τη διδασκαλία και τη μάθηση ως μια διεργασία μετασχηματισμού και τροποποίησης κατά την οποία ο μαθητής μεταβαίνει από μια αρχική κατάσταση σε μια τελική, με κύριο χαρακτηριστικό τη βελτίωση της διανοητικής του απόδοσης. Η σημαντικότερη παράμετρος που επηρεάζει αυτήν τη διαδικασία είναι οι διδακτικές τεχνικές που ακολουθούνται για την επίτευξη του επιθυμητού στόχου. Η πολυπλοκότητα και η δυσκολία του θέματος επισημαίνεται κυρίως στο ότι ένας μαθητής που βρίσκεται στην αρχική κατάσταση δεν είναι ένα άγραφο βιβλίο που πρέπει να γραφεί σωστά, αλλά έχει ήδη κάποια γνώση που πρέπει να τροποποιηθεί, να βελτιωθεί, να ενισχυθεί.

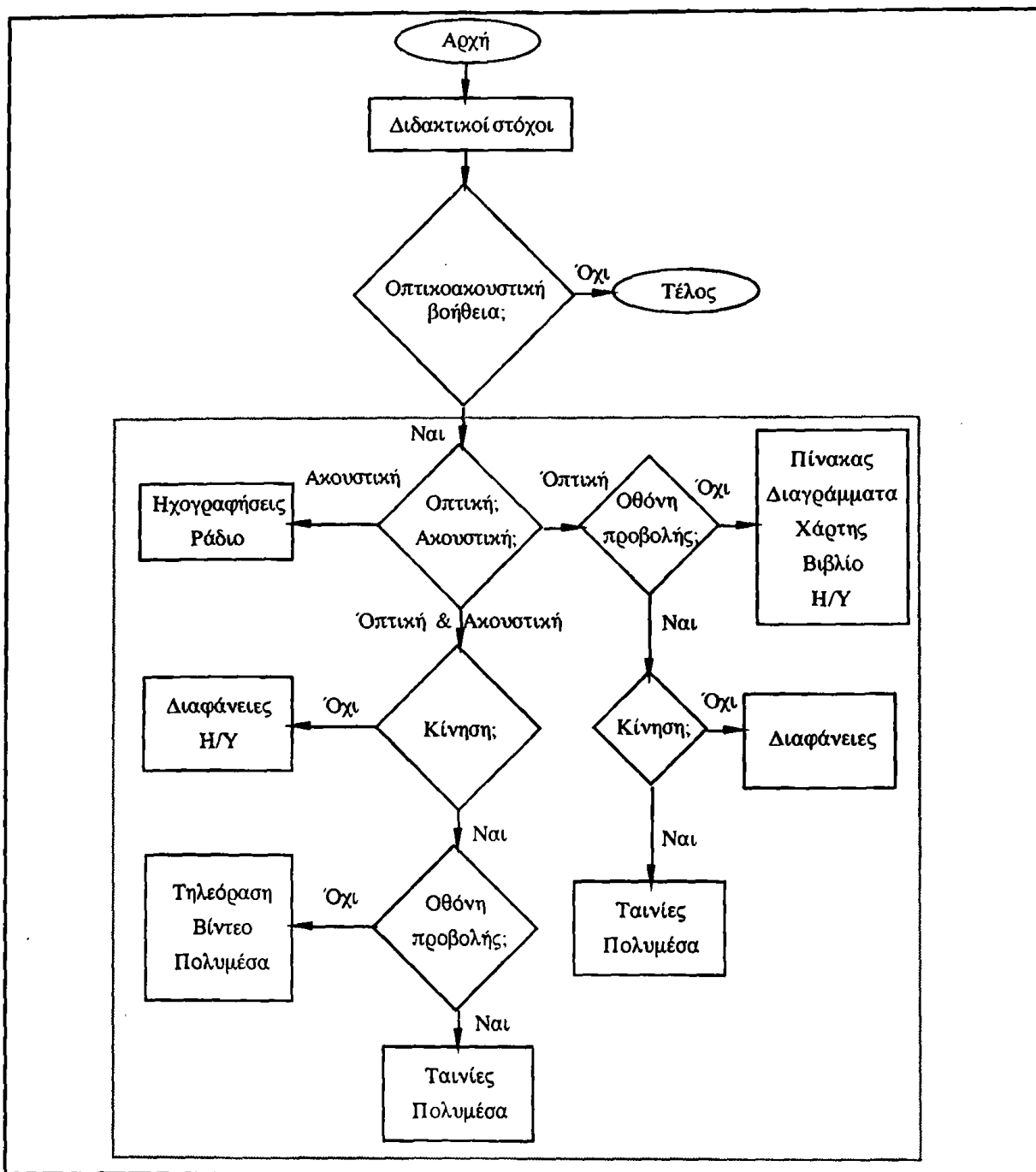
Σχετικά με τη φυσική, μελέτες έχουν δείξει ότι οι μαθητές ανεξάρτητα από ηλικία και βαθμίδα εκπαίδευσης έχουν μια γενική και πρωτόλεια ιδέα για το φυσικό κόσμο. Η τροποποίηση και αναδόμηση της υπάρχουσας αυτής γνώσης είναι δυσκολότερη από την αποδοχή νέας και απαιτεί σύγχρονες εκπαιδευτικές μεθόδους. Ουσιαστική είναι η εμπέδωση των αρχών που διέπουν το αντικείμενο μελέτης και το σκεπτικό με το οποίο ο μαθητής θα οργανώσει την επιστημονική γνώση για να φτάσει στον τελικό σκοπό του που συνήθως είναι η επίλυση προβλημάτων.

Η πληροφορική δεν αποτελεί και ούτε πρέπει να θεωρείται πανάκεια για την εκπαίδευση, αλλά η θέση της είναι στο να συμβάλλει με τα μέσα που διαθέτει, και διαθέτει πολλά, στην εξέλιξη της διδακτικής θεωρίας και την ανάπτυξη νέων εκπαιδευτικών τεχνικών.

Οι υπολογιστές δεν αποτελούν το μόνο τεχνολογικό μέσο στην υπηρεσία της εκπαίδευσης, αλλά το πλέον σύγχρονο με τις μεγαλύτερες δυνατότητες.

Τα σπουδαιότερα εκπαιδευτικά μέσα με τα οποία υλοποιείται η διδασκαλία μέσα από μαθήματα, εξάσκηση και προσομοιώσεις είναι τα οπτικοακουστικά μέσα συμπεριλαμβανομένων του έντυπου υλικού και της πληροφορικής τεχνολογίας. Η τελευταία υπερέρχει των προηγούμενων αφού εκτός από την άμεση αλληλεπίδραση με το χρήστη, συνδυάζει και τα θετικά στοιχεία του βιβλίου και των οπτικοακουστικών μέσων με τα πολυμέσα και υπερμέσα (multimedia, hypermedia).





Σχήμα 6.1 Επιλογή μέσου διδασκαλίας με βάση την τεχνολογία του. Οι απαιτήσεις που περιλαμβάνονται στο διακεκομμένο πλαίσιο μπορούν να υλοποιηθούν με έναν υπολογιστή πολυμέσων

Το σχήμα 6.1 παρουσιάζει διαγραμματικά την πορεία για την επιλογή ενός μέσου διδασκαλίας με βάση την τεχνολογία του. Όπως φαίνεται ένας υπολογιστής πολυμέσων αποτελεί το "ιδεατό" μέσο που επικουρεί στην επίτευξη του στόχου του εκπαιδευ-



τικού, με κατάλληλη εκμετάλλευση των συνόλων των συστημάτων συμβόλων και τις δυνατότητες επεξεργασίας του.

Οι δύο μεγάλες κατευθύνσεις της πληροφορικής στην εκπαιδευτική διαδικασία είναι η πληροφορική ως γνωστικό αντικείμενο και η πληροφορική ως διδακτική τεχνική. Η πρώτη έχει να κάνει με την εκπαίδευση στην επιστήμη των υπολογιστών και περιλαμβάνει όλους τους τομείς και τις ειδικεύσεις της πληροφορικής.

Η δεύτερη κατεύθυνση αφορά στην εκπαίδευση με υπολογιστές και απευθύνεται σε όλα σχεδόν τα γνωστικά αντικείμενα και βαθμίδες της εκπαίδευσης. Διαιρείται σε δύο μεγάλες κατηγορίες, τη διδασκαλία βασισμένη σε υπολογιστές (Computer Based Instruction-CBI) και τη διδασκαλία/μάθηση με τη βοήθεια των υπολογιστών (Computer Assisted Instruction-CAI, Computer Aided Learning-CAL).

Κύριο χαρακτηριστικό της πρώτης είναι η διδασκαλία που βασίζεται εξ' ολοκλήρου στους υπολογιστές με τη μετάδοση της γνώσης να γίνεται από τη μηχανή με την καθοδήγηση βέβαια του εκπαιδευτικού. Υπάρχει πολύς σκεπτικισμός ακόμα γι' αυτόν τον τρόπο εκπαίδευσης αν και η σύγχρονη τεχνολογία με τα νευρωνικά δίκτυα, την τεχνητή ευφυΐα και τους νέους τρόπους επικοινωνίας ανθρώπου-μηχανής μεταβάλλει γρήγορα τα πράγματα.

Η διδασκαλία με τη βοήθεια των υπολογιστών ήδη εφαρμόζεται σε πολλές χώρες και οι σχετικές έρευνες έχουν δείξει ότι οι στόχοι της εκπαίδευσης εκπληρώνονται σε μεγάλο βαθμό με την εισαγωγή της επιστήμης των υπολογιστών και ιδιαίτερα με τη χρήση κατάλληλου λογισμικού που οδηγεί στην κατανόηση και οργάνωση των εννοιών και τη λύση προβλημάτων. Στη Μεγάλη Βρετανία η εισαγωγή της πληροφορικής στα σχολεία με σκοπό τη βελτίωση της διδακτικής σε κάθε γνωστικό αντικείμενο έχει φέρει θετικά αποτελέσματα και από το 1989 το Εθνικό Συμβούλιο για την παιδεία συστήνει τη χρήση της όχι μόνο ως αντικείμενο μελέτης αλλά και ως διδακτική τεχνική σε όλα τα μαθήματα. Στις Η.Π.Α. έχει γίνει επιπλέον και η εκτίμηση (evaluation) της πληροφορικής σε σχέση με άλλα εκπαιδευτικά μέσα [Plomp et al, 1996]. Στην Ελλάδα προς το παρόν η εισαγωγή της πληροφορικής στην εκπαίδευση έχει γίνει ως γνωστικό αντικείμενο [ΥΠΕΠΘ, 1994/95 & ΥΠΕΠΘ, 1993]. Εισαγωγικά θέματα διδάσκονται σε Γυμνάσια της χώρας που διαθέτουν βασικό εξοπλισμό και σε Ενιαία Πολυκλαδικά και Τεχνικά Επαγγελματικά Λύκεια σε τμήματα ειδίκευσης στην πληροφορική.

Η διδασκαλία με τη βοήθεια υπολογιστών όπως δηλώνει και ο όρος δεν αποτελεί αυτόνομη πρακτική, αλλά συνοδεύει συμπληρώνοντας και βελτιώνοντας την κλασική μεθοδολογία.



6.2 Διδασκαλία με τη βοήθεια υπολογιστή

Οι διδακτικοί στόχοι υποδείχνουν τις στρατηγικές στη χρήση των υπολογιστών στην τάξη, καθώς και το κατάλληλο λογισμικό. Συνήθως χρησιμοποιούνται προγράμματα διδασκαλίας (tutorials), ασκήσεων (drills), προσομοίωσης (simulation) και εργαλείων λογισμικού (tool applications).

Τα προγράμματα διδασκαλίας ενεργούν όπως ο δάσκαλος και τα εγχειρίδια. Ο υπολογιστής παρέχει πληροφορίες κατά τμήματα για το υπό μελέτη αντικείμενο. Κάθε κομμάτι πληροφορίας ζητά από το μαθητή απάντηση σε συγκεκριμένα θέματα, με την επόμενη κίνηση της μηχανής να εξαρτάται από την απάντηση. Έτσι μπορεί να συνεχίσει με πληροφόρηση, να οδηγήσει το μαθητή στην κατανόηση του υλικού με διαφορετικό τρόπο και παραδείγματα, να εμφανίσει συμπληρωματικές πληροφορίες, να υποδείξει το σκεπτικό για την επίλυση προβλημάτων.

Τα πλεονεκτήματα ενός προγράμματος διδασκαλίας με υπολογιστή σε σύγκριση με την αποκλειστική μελέτη εγχειριδίων, περιλαμβάνουν την άμεση ανάδραση μεταξύ μαθητή και μηχανής, την προσήλωση στο υπό μελέτη θέμα και την ενσωμάτωση φωτογραφιών, κινούμενων γραφικών, ήχου και βίντεο (πολυμέσα). Τα τελευταία χρόνια αναπτύσσεται μια μέθοδος διδασκαλίας που εφαρμόζει στοιχεία της τεχνητής ευφυΐας και ονομάζεται έξυπνη διδασκαλία με τη βοήθεια υπολογιστή (Intelligent Computer Assisted Instruction-ICAI). Το υπολογιστικό σύστημα περιλαμβάνει μοντέλα μαθητών, αναζητά τις εσφαλμένες αντιλήψεις του κάθε μαθητή και του παρέχει πληροφορίες για τη βελτίωση της κατανόησης των αντίστοιχων θεμάτων.

Οι ασκήσεις που παρέχουν επαναλαμβανόμενη πρακτική και ανάδραση, ξεφεύγουν από την παρεξηγημένη έννοια που τους δίνεται από διδάσκοντες και μαθητές. Τέτοιου είδους προγράμματα αναπτύσσουν τις δεξιότητες του μαθητή και του παρέχουν ένα επιπλέον εφόδιο πέρα από την αρχική διδασκαλία και κατανόηση των εννοιών. Συνήθως έχουν τη μορφή ερωτήσεων-απαντήσεων, κουίζ και παιχνιδιών. Τα πλεονεκτήματα της εξάσκησης με τον υπολογιστή είναι παρόμοια με αυτά της διδασκαλίας, παρέχοντας κίνητρα για ενεργή συμμετοχή του μαθητή.

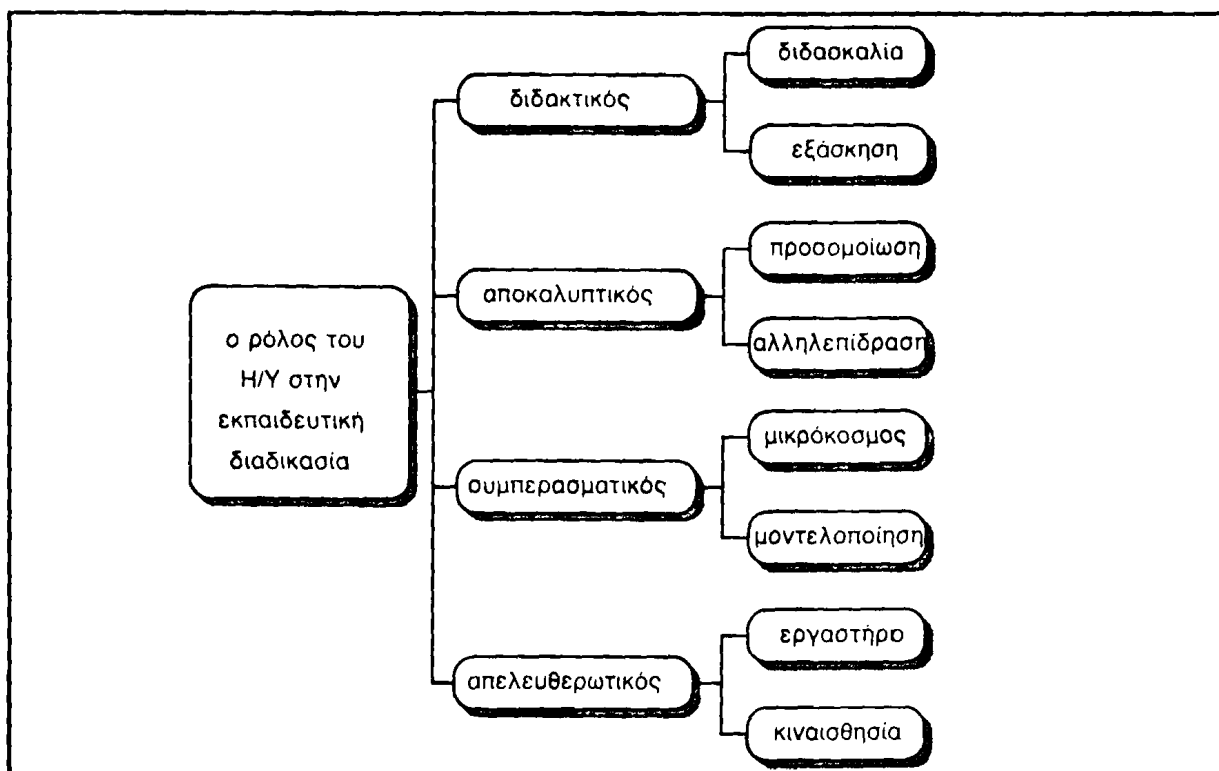
Οι προσομοιώσεις φαινομένων και καταστάσεων στον υπολογιστή αποτελούν μια από τις σημαντικότερες προσφορές του στην εκπαίδευση. Δραστηριότητες που είναι αδύνατο να γίνουν στην τάξη επειδή είναι χρονοβόρες (ανάπτυξη ενός φυτού), μεγάλου κόστους (σύγκρουση αυτοκινήτου), επικίνδυνες (εξάπλωση ραδιενέργειας), ή “ανήθικες” (πειράματα γενετικής), προσομοιώνονται με μικρό κόστος, στο διατιθέμενο χρόνο και με ασφάλεια. Η συνεισφορά της προσομοίωσης φαίνεται ιδιαίτερα στις φυσικές επιστήμες που μελετούν φυσικά, χημικά και βιολογικά φαινόμενα στα οποία η κλίμακα του χρόνου και της ενέργειας μπορεί να είναι από απειροελάχιστη μέχρι άπειρη.



Οι σωστές προσομοιώσεις εστιάζουν στο υπό μελέτη αντικείμενο και δεν συμπεριλαμβάνουν παράγοντες που τροποποιούν τα αποτελέσματα της μελέτης ή επηρεάζουν με διάφορους τρόπους το φαινόμενο. Επιπρόσθετοι παράγοντες που φέρνουν το αντικείμενο που προσομοιώνεται προς την πραγματικότητα ή υποθετικές συνθήκες, ενσωματώνονται κατά τη βούληση του χρήστη.

Εργαλεία λογισμικού όπως επεξεργαστές κειμένου, βάσεις δεδομένων, φύλλα εργασίας και γλώσσες προγραμματισμού χρησιμοποιούνται για την ανάπτυξη πνευματικών δεξιοτήτων και την επίλυση προβλημάτων που σχετίζονται με διάφορα γνωστικά αντικείμενα όπως η γλώσσα, τα μαθηματικά, οι φυσικές επιστήμες και οι τέχνες.

Τέλος, τα παιχνίδια και ιδιαίτερα τα παιχνίδια δράσης (arcade games) και στρατηγικής εκπληρώνουν ορισμένους εκπαιδευτικούς στόχους. Αποτελούν δημοφιλείς και ευχάριστες δραστηριότητες στον υπολογιστή παρακινώντας το χρήστη σε γρήγορη δράση και αντίδραση, εξασκώντας ανακλαστικά και παρατηρητικότητα, προσφέροντας γνώσεις σε συγκεκριμένα θέματα. Το σημαντικότερο πρόβλημα των παιχνιδιών είναι οι πολλές ώρες που αφιερώνει ο χρήστης μέχρι να φτάσει στο τέλος τους ή να πετύχει τον επιθυμητό στόχο. Τέτοιου είδους κίνητρα μπορούν να μετατραπούν σε πλεονέκτημα με την ανάπτυξη παιχνιδιών συγκεκριμένου εκπαιδευτικού περιεχομένου (ιστορικού, μαθηματικού, τεχνολογικού) και την ενασχόληση των μαθητών στον κατάλληλο χρόνο.



Σχήμα 6.2 Ο ρόλος του υπολογιστή στην εκπαιδευτική διαδικασία



Σχετικά με την εισαγωγή της πληροφορικής στην εκπαιδευτική διαδικασία, σημαντικός παράγοντας είναι ο ρόλος που έχει ο υπολογιστής στην παρακίνηση και προώθηση των διδακτικών δραστηριοτήτων. Αυτός παρουσιάζεται στο σχήμα 6.2 σε συνδυασμό με τα είδη εφαρμογών ανά τομέα.

Ο ρόλος του υπολογιστή στην εκπαιδευτική διαδικασία είναι η μετάδοση πληροφοριών στο μαθητή πληρώντας τις απαιτήσεις που θέτουν ο εκπαιδευτικός και το αναλυτικό πρόγραμμα. Παραδείγματα εφαρμογών είναι η διδασκαλία και οι ασκήσεις με τον τρόπο που έχουν αναφερθεί παραπάνω.

Με τον όρο αποκαλυπτικός εννοείται ο ρόλος κατά τον οποίο ο υπολογιστής μεσολαβεί μεταξύ του μαθητή και ενός ποιοτικού ή ποσοτικού μοντέλου των φαινομένων του πραγματικού κόσμου. Ένας τέτοιος ρόλος υλοποιείται με τις προσομοιώσεις και την ενεργή συμμετοχή του χρήστη στα δρώμενα στον υπολογιστή με άμεση απόκριση.

Στο συμπερασματικό ρόλο, ο μαθητής επεξεργάζεται τις πληροφορίες και βγάζει συμπεράσματα θέτοντας ερωτήσεις του τύπου "τι θα συνέβαινε αν..." στον υπολογιστή και ελέγχοντας τις υποθέσεις του. Μ' αυτόν τον τρόπο δημιουργεί αλληλεπιδρώντα μοντέλα σε διάφορα γνωστικά αντικείμενα. Εφαρμογές τέτοιου τύπου είναι οι μικρόκοσμοι με χαρακτηριστικό παράδειγμα αυτούς που αναπτύσσει ο ίδιος ο μαθητής στο ανοικτό προγραμματιστικό περιβάλλον της γλώσσας Logo και η κατασκευή μοντέλων διαφόρων καταστάσεων.

Με τον απελευθερωτικό ρόλο τέλος εννοείται η δημιουργική επεξεργασία των πληροφοριών και έχει ως σκοπό την αποφυγή κάθε μη παραγωγικής διαδικασίας όπως οι επαναλαμβανόμενοι υπολογισμοί και η οργάνωση των δεδομένων. Σημαντικό παράδειγμα εφαρμογής σ' αυτόν τον τομέα αποτελούν τα εργαστήρια που βασίζονται σε μικροϋπολογιστές (microcomputer based laboratories). Σε τέτοια εργαστήρια ο υπολογιστής αυτοματοποιεί το πείραμα και συλλέγει δεδομένα, τα ταξινομεί, τα αναλύει και τα παρουσιάζει με τον επιθυμητό τρόπο σε πραγματικό χρόνο. Έτσι ο μαθητής παρακολουθεί πραγματικά φαινόμενα, συμμετέχει σ' αυτά και βλέπει άμεσα τα αποτελέσματά τους, λαμβάνοντας απευθείας γνώση από τις προσωπικές του εμπειρίες. Οι εφαρμογές της κιναισθησίας περιλαμβάνουν την ενεργή συμμετοχή του μαθητή σε δραστηριότητες του πραγματικού κόσμου με τη σύνδεση του ίδιου με τον υπολογιστή, μεταφέροντάς του τις κινήσεις του και μελετώντας τα αποτελέσματά τους.

6.3 Ο υπολογιστής στην εκπαιδευτική διαδικασία

Έρευνες της τελευταίας εικοσαετίας έχουν δείξει ότι η γνώση της χρήσης του υπολογιστή για διδασκαλία σημαίνει γνώση της ίδιας της διδασκαλίας. Ο υπολογιστής είναι ένα εργαλείο που επεκτείνει τις δυνατότητες του δασκάλου, με συνέπεια διαφορετικοί δάσκαλοι να χρησιμοποιούν τον υπολογιστή με διαφορετικούς τρόπους.



Η πορεία της διδασκαλίας για την προσφορά, κατανόηση και εμπέδωση της γνώσης ακολουθεί αρχές πολλές από τις οποίες πραγματώνονται δημιουργικά μέσα από τη διδασκαλία με τη βοήθεια υπολογιστή (CAI).

Μια δυνατότητα του υπολογιστή πέρα από τις συμβατικές της επεξεργασίας των δεδομένων, είναι η τεχνολογία των πολυμέσων (multimedia) και υπερμέσων (hypermedia). Μέχρι πρόσφατα, ο όρος πολυμέσα σήμαινε τη χρήση διάφορων ανεξάρτητων συσκευών, όπως συγχρονισμένη προβολή διαφανειών, βίντεο και ηχητικής κάλυψης. Η τεχνολογική ανάπτυξη συνέβαλε στο συνδυασμό όλων των μέσων με τρόπο ώστε η πληροφορία που παρέχεται από αυτά να ολοκληρώνεται σε ένα, τον ηλεκτρονικό υπολογιστή. Με την ανάπτυξη των πολυμέσων, ο υπολογιστής βγήκε από το σιωπηλό του κόσμο, και ουσιαστικά άλλαξε τον τρόπο διαχείρισης και χρησιμοποίησης της πληροφορίας. Για την ολοκλήρωση των τεχνολογιών που αναφέρθηκαν παραπάνω, ένας υπολογιστής πολυμέσων περιλαμβάνει έναν οδηγό δίσκων CD (compact disk), κάρτα ήχου, ηχεία, και το απαραίτητο υλικό και λογισμικό για την επεξεργασία του ήχου και τη διαχείριση κινούμενων σχεδίων και βίντεο. Όπως φαίνεται μέχρι τώρα, τα πολυμέσα δε διαφέρουν πολύ από ένα σύστημα βίντεο και τηλεόρασης. Αυτό όμως αφορά τον τομέα της αναπαραγωγής, αφού ένα τέτοιο σύστημα δεν έχει δυνατότητες επεξεργασίας των σημάτων. Ένα σύστημα πολυμέσων, εκτός των άλλων παρέχει ένα εικονικό στούντιο ηχοληψίας και κινηματογράφου, αφού επιτρέπει μέσω του κατάλληλου λογισμικού την επεξεργασία ήχου και εικόνας. Το πλέον ισχυρό όμως χαρακτηριστικό του υπολογιστή πολυμέσων είναι η πρόσβαση στις πληροφορίες που διαχειρίζεται με άμεσο, τυχαίο τρόπο. Το πλεονέκτημα που προσφέρουν τα πολυμέσα είναι συνδυασμός ήχου, εικόνας και κειμένου και εύκολη και γρήγορη πρόσβαση σε οποιοδήποτε σημείο της εφαρμογής. Τα πολυμέσα ολοκληρώνονται με την ενσωμάτωση υπερκειμένου. Ως υπερκείμενο (hypertext) ορίζεται το μη γραμμικό κείμενο, με δυνατότητα σύνδεσης ανεξάρτητων σε πρώτη άποψη κειμένων - κόμβων, μέσω λέξεων κλειδιών, των συνδέσμων. Με αυτόν τον τρόπο περνάμε στα υπερμέσα (hypermedia) που εκτός από τις δυνατότητες των πολυμέσων, προσφέρουν άμεση σύνδεση μεταξύ πληροφοριών με την ενεργοποίηση περιοχών της εφαρμογής με τη μορφή κειμένου ή γραφικών.

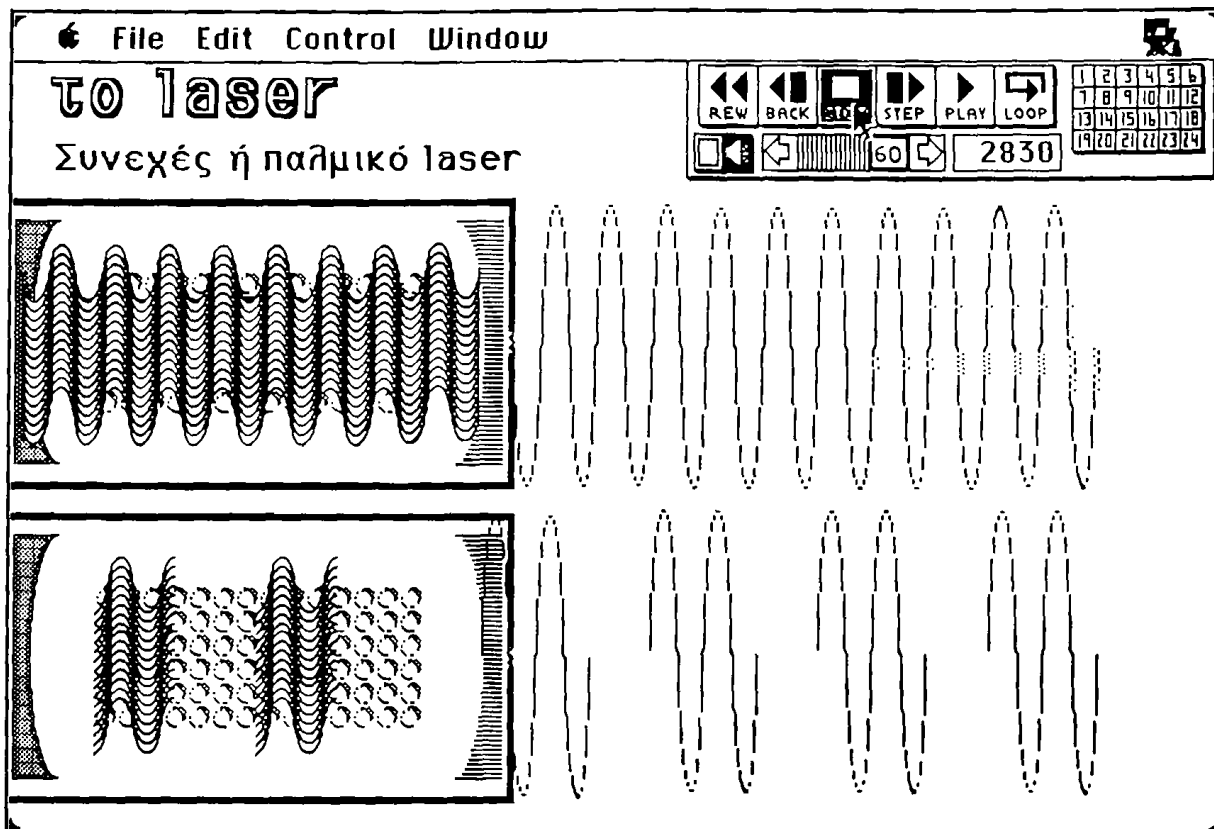
Ορισμένες από τις αρχές της παιδαγωγικής που έχουν χαρακτηριστικά τα οποία αναδεικνύονται μέσα από την εκμετάλλευση των δυνατοτήτων του υλικού και λογισμικού των υπολογιστών, αναφέρονται παρακάτω. Ο υπολογιστής αποτελεί πλέον ένα ισχυρό εργαλείο για το χειρισμό αυτών των αρχών και την εμφάνιση των αποτελεσμάτων τους στο μαθητή [Vockel, 1992].

Πρωτογενής διαδικασία και βασική γνώση

Όλοι οι μαθητές κατανοούν σε γενικά πλαίσια το κυρίως θέμα, στον κατάλληλο χρόνο. Το λογισμικό παρέχει επιπλέον βοηθήματα και πρακτική εξάσκηση στην πορεία της κατανόησης των εννοιών και καταγράφει την απόδοση του μαθητή.



Το σχήμα 6.3 παρουσιάζει στιγμιότυπα από την παρουσίαση ενός μαθήματος πολυμέσων σχετικά με τις βασικές αρχές των lasers. Το θέμα παρουσιάζεται μέσα από πολλά συστήματα συμβόλων σε μορφή κινουμένων σχεδίων, με τις απαραίτητες γνώσεις να εμφανίζονται στην κατάλληλη στιγμή, συνοδευόμενες από χαρακτηριστικούς ήχους και κείμενο [Λαδιάς κ.α. 1989].



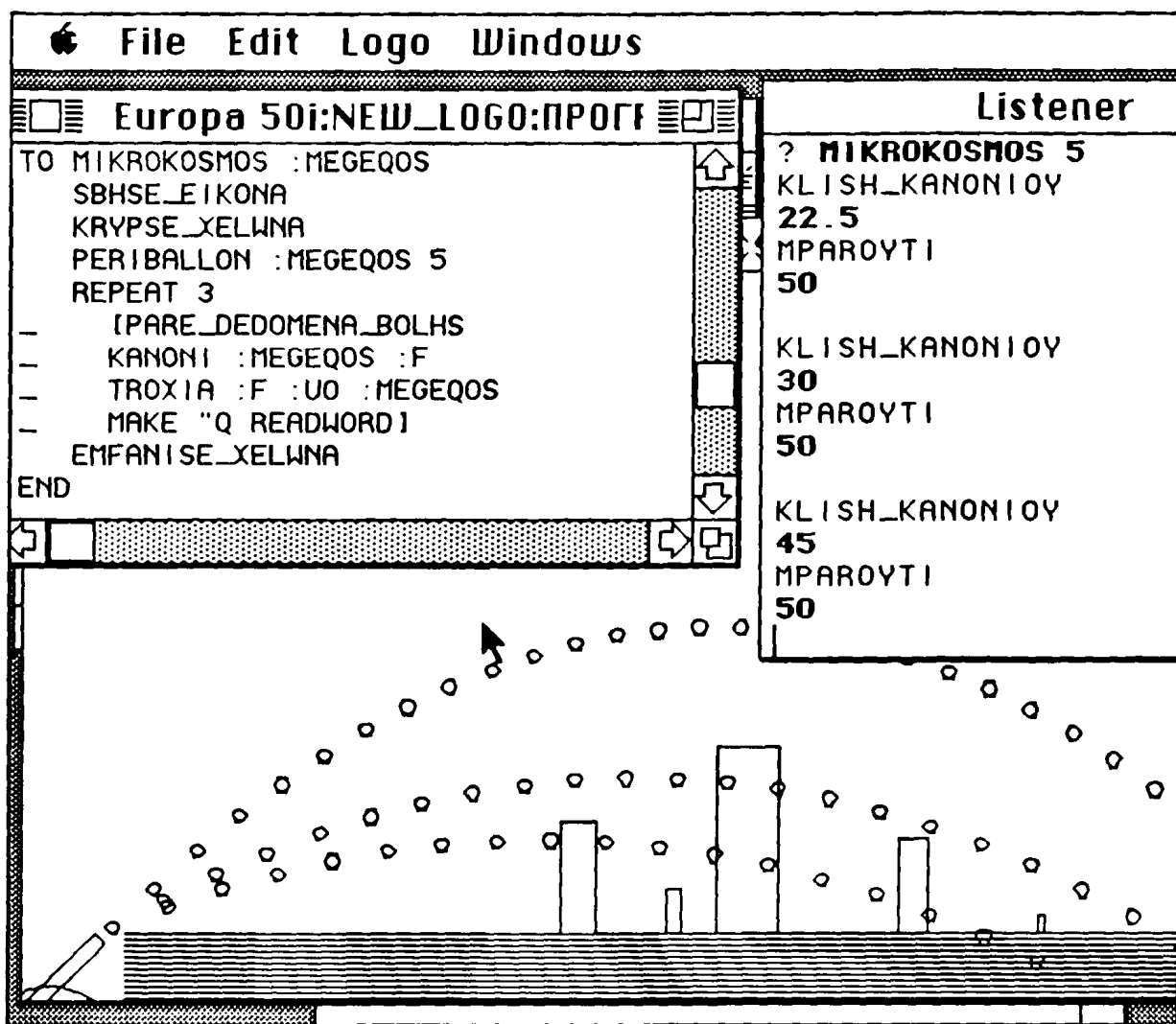
Σχήμα 6.3 Οι βασικές αρχές των lasers. Όταν ικανοποιούνται οι κατάλληλες συνθήκες στην κοιλότητα του laser (αριστερά), εκπέμπεται φως laser ως συνεχής (επάνω) ή παλμική (κάτω) ακτινοβολία

Άμεση διδασκαλία (direct instruction)

Οι μαθητές αναπτύσσουν δεξιότητες ακολουθώντας τα στάδια που καθορίζει ο εκπαιδευτικός κατά την περιγραφή του τελικού στόχου. Το λογισμικό καθορίζει τα ακριβή βήματα και η εκμάθηση των αρχών του δομημένου προγραμματισμού και η εξάσκηση σε κάποια γλώσσα παρέχουν το σκεπτικό για τη σωστή αντιμετώπιση προβλημάτων. Το σχήμα 6.4 παρουσιάζει ένα περιβάλλον εργασίας στη γλώσσα προγραμματισμού Logo, έναν μικρόκοσμο φυσικής στο θέμα της πλάγιας βολής. Το φαινόμενο προσομοιώνεται με κανονιοβολισμούς σε συγκεκριμένο στόχο, πάνω από τα κτίρια. Ο μικρόκοσμος περιλαμβάνει τα αντικείμενα με τις ιδιότητές τους, και περιορισμούς. Ο μαθητής προσπαθεί να πετύχει το στόχο ρυθμίζοντας την κλίση και τον αριθμό επανά-



ληψης των βολών, εκμεταλλευόμενος αρχές της φυσικής και των μαθηματικών χωρίς να βρίσκεται αντιμέτωπος με ένα ψυχρό πρόβλημα του βιβλίου.



Σχήμα 6.4 Ένας μικρόκοσμος της logo για μάθηση στο αντικείμενο της πλάγιας βολής. Το περιβάλλον δράσης "κρύβει" το ψυχρό πρόβλημα της φυσικής, δίνοντας κίνητρα στο μαθητή

Δευτερογενής διαδικασία μάθησης - εξειδικευμένη γνώση (overlearning)

Για την εμπέδωση των εννοιών απαιτείται πρακτική εξάσκηση και ενασχόληση πέρα από την αρχική διδασκαλία. Το λογισμικό παρέχει εξάσκηση με το χαρακτηριστικό της εξατομίκευσης, κεντρίζει το ενδιαφέρον με τη δομή παιχνιδιού και του περιβάλλοντος προς εξερεύνηση, και προσφέρει πολλές προσεγγίσεις στο ίδιο αντικείμενο. Στο σχήμα 6.5 ο χρήστης αλληλεπιδρά με ένα ανοικτό περιβάλλον εικονικής πραγματικότητας (virtual reality) [Mikropoulos, 1997]. Ο χρήστης "βρίσκεται" σε ένα τρισδιάστατο περιβάλλον με δυνατότητα να το εξερευνήσει με πλήρη ελευθερία κινήσεων.

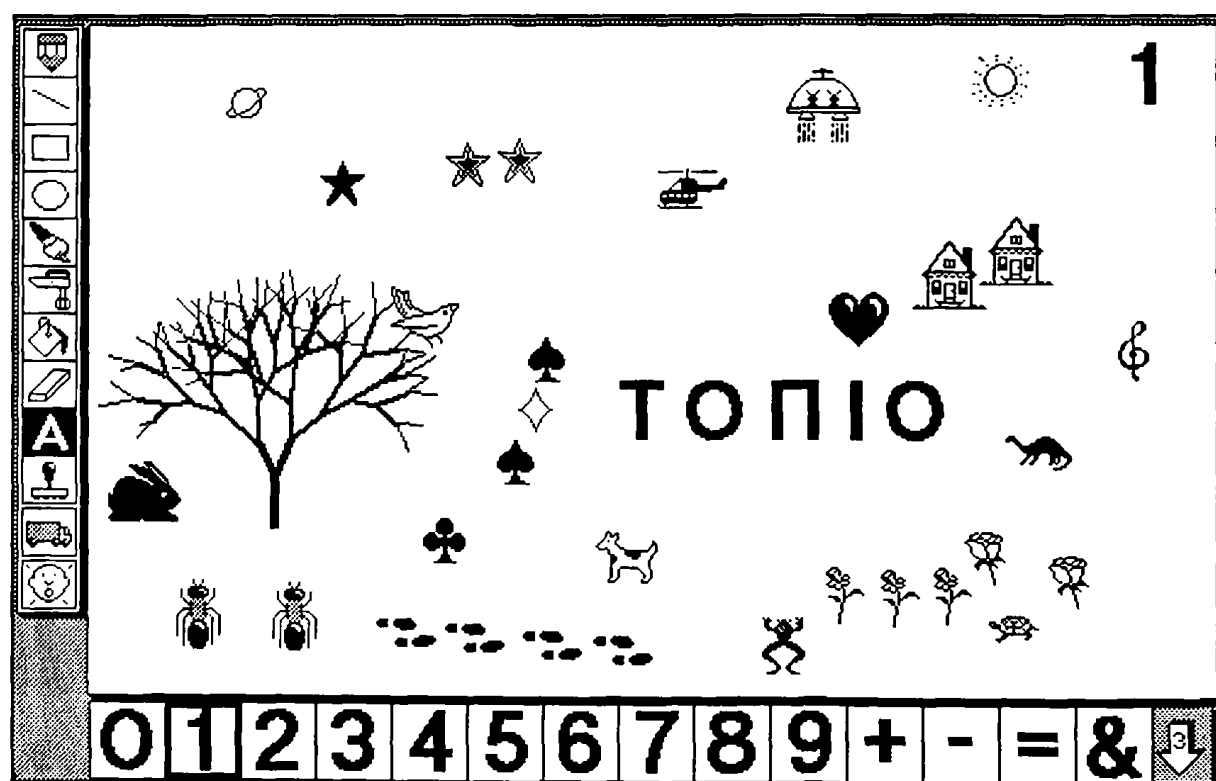




Σχήμα 6.5 Πλοήγηση σε εικονικό εκπαιδευτικό περιβάλλον με χρήση κατάλληλων περιφερειακών συσκευών

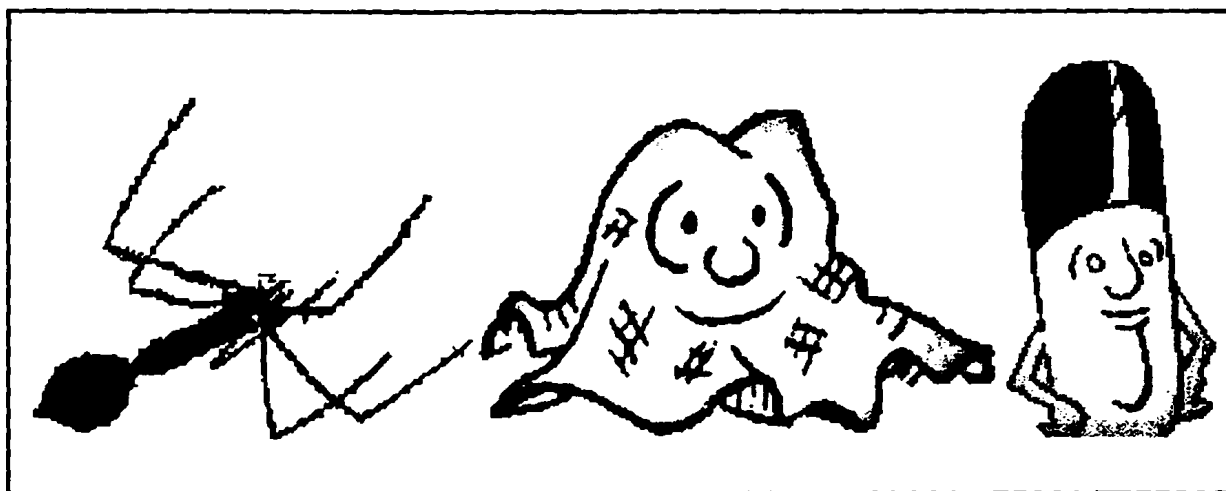
Κατανόηση διεργασιών απομνημόνευσης (memorization skills)

Η απομνημόνευση συντελεί στην εμπέδωση της βασικής πληροφορίας στην οποία βασίζεται η κατανόηση και η κριτική σκέψη. Το λογισμικό διευκολύνει την εκμάθηση και την απομνημόνευση με την επαναλαμβανόμενη πρακτική εξάσκηση.



Σχήμα 6.6 Το πρόγραμμα KidPix (BrΩderbund Software) για διδασκαλία και ανάπτυξη δεξιοτήτων παιδιών ηλικίας 5-6 ετών

Το σχήμα 6.6 δείχνει μια οθόνη του προγράμματος KidPix που απευθύνεται σε παιδιά ηλικίας 5-6 ετών. Το παιδί έχει μια μεγάλη ποικιλία εργαλείων για σχεδίαση που συνοδεύονται από ευχάριστους ήχους, καθώς και το αλφάβητο και τους αριθμούς που όταν επιλέγονται προφέρεται η αντίστοιχη ονομασία.



Σχήμα 6.7 Οθόνη πακέτου πολυμέσων για την αγωγή υγείας σε παιδιά προσχολικής ηλικίας. Οι χρησιμοποιούμενες αναπαραστάσεις και προσομοιώσεις συντελούν σε διεργασίες απομνημόνευσης από τα παιδιά

Το σχήμα 6.7 παρουσιάζει στιγμιότυπο ενός πακέτου πολυμέσων για την αγωγή υγείας σε παιδιά προσχολικής ηλικίας [Μίκγορουλος et al, 1994]. Τα μικρόβια και οι ιοί αναπαριστάνονται ως έντομα - γεγονός όχι μακριά από την πραγματικότητα. Η αλληλεπίδραση των παιδιών με το πακέτο γίνεται μέσω ομιλίας, εντολών που προφέρουν τα παιδιά στον υπολογιστή.

Ενσωμάτωση προαπαιτούμενων γνώσεων (prerequisite knowledge)

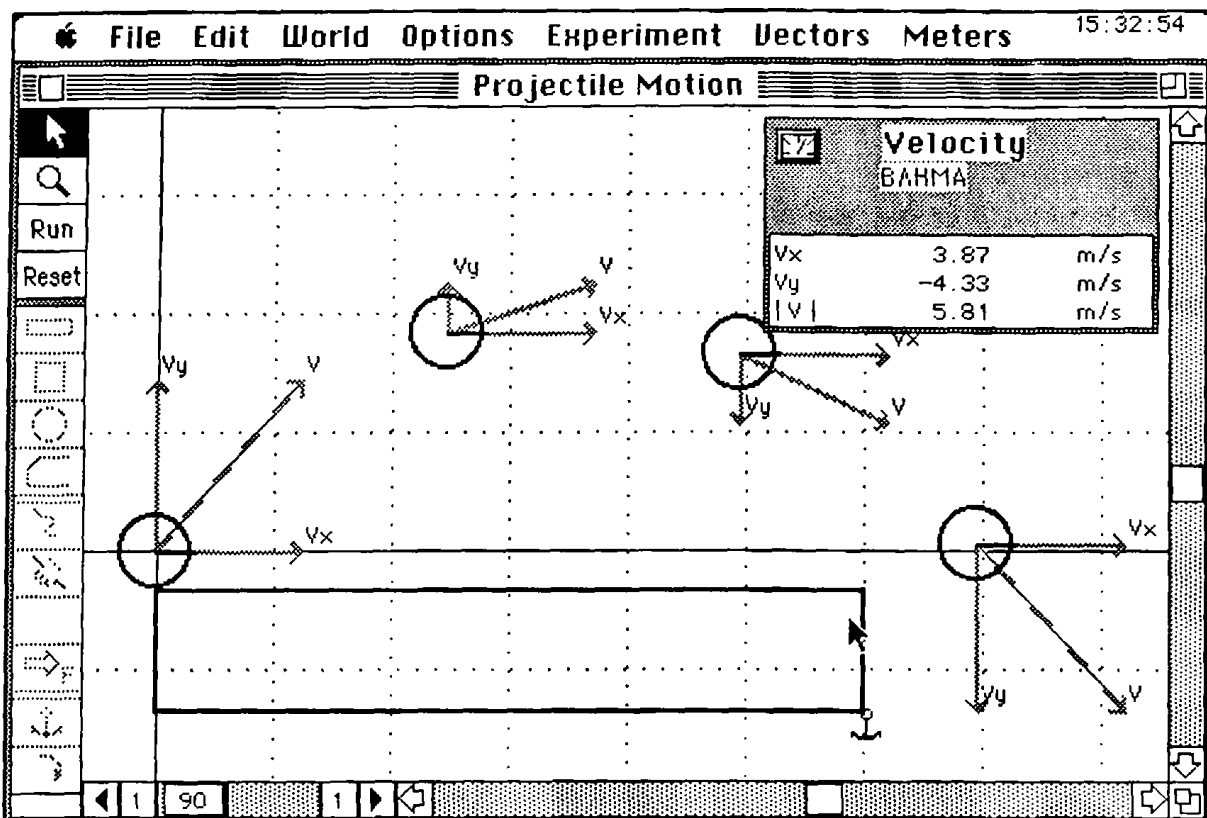
Εξαιτίας της ιεραρχικής δομής της γνώσης, απαιτείται κάποιο βασικό επίπεδο για την κατανόηση νέων ιδεών. Το λογισμικό αξιολογεί τη βασική γνώση και δεξιότητες και συμπληρώνει τα υπάρχοντα κενά.

Άμεση ανάδραση (immediate feedback)

Η ενεργή και άμεση αλληλεπίδραση μαθητή με διδάσκοντα βελτιώνει τη μαθησιακή διεργασία. Το λογισμικό παρέχει άμεση και σωστή ανάδραση στις ενέργειες του μαθητή.

Στο σχήμα 6.8 παρουσιάζεται μια οθόνη του λογισμικού Interactive Physics στο οποίο ο μαθητής υλοποιεί προσομοιώσεις πειραμάτων μηχανικής, βλέπει άμεσα τα αποτελέσματά τους και κατανοεί τη δράση των νόμων της φυσικής στον πραγματικό κόσμο [Τζιμογιάννης κ. α., 1995].





Σχήμα 6.8 Το πρόγραμμα Interactive Physics (Knowledge Revolution). Στιγμιότυπα από την τροχιά σε πλάγια βολή. Σημειώνονται τα διανύσματα της ταχύτητας και ένα ψηφιακό όργανο μέτρησης των συνιστωσών και του μέτρου της

Τύποι μάθησης (learning styles)

Οι μαθητές έχουν διαφορετικούς τρόπους να μαθαίνουν. Κάποιοι επιθυμούν αργή ροή πληροφοριών, άλλοι θέλουν δύσκολες ασκήσεις, κάποιοι άλλοι είναι οπτικοί τύποι. Το λογισμικό προσαρμόζεται στις ανάγκες και απαιτήσεις κάθε μαθητή, παρουσιάζοντας τις ίδιες πληροφορίες με πολλούς τρόπους.

Διδασκαλία με ισότιμη αλληλεπίδραση (peer tutoring)

Η σωστά δομημένη ισότιμη διδασκαλία παρέχει οφέλη και στο δάσκαλο και το μαθητή. Το λογισμικό βοηθά και τον πρώτο παρέχοντας πληροφορίες, υποδείξεις και ανάδραση.

Συνεργασία (cooperative learning)

Η συνεργασία μεταξύ των μαθητών είναι περισσότερο παραγωγική από τον ανταγωνισμό. Οι μαθητές εργάζονται σε ομάδες με τον υπολογιστή, που παρέχει κίνητρα για συνεργασία.



Έλεγχος προόδου (progress monitoring)

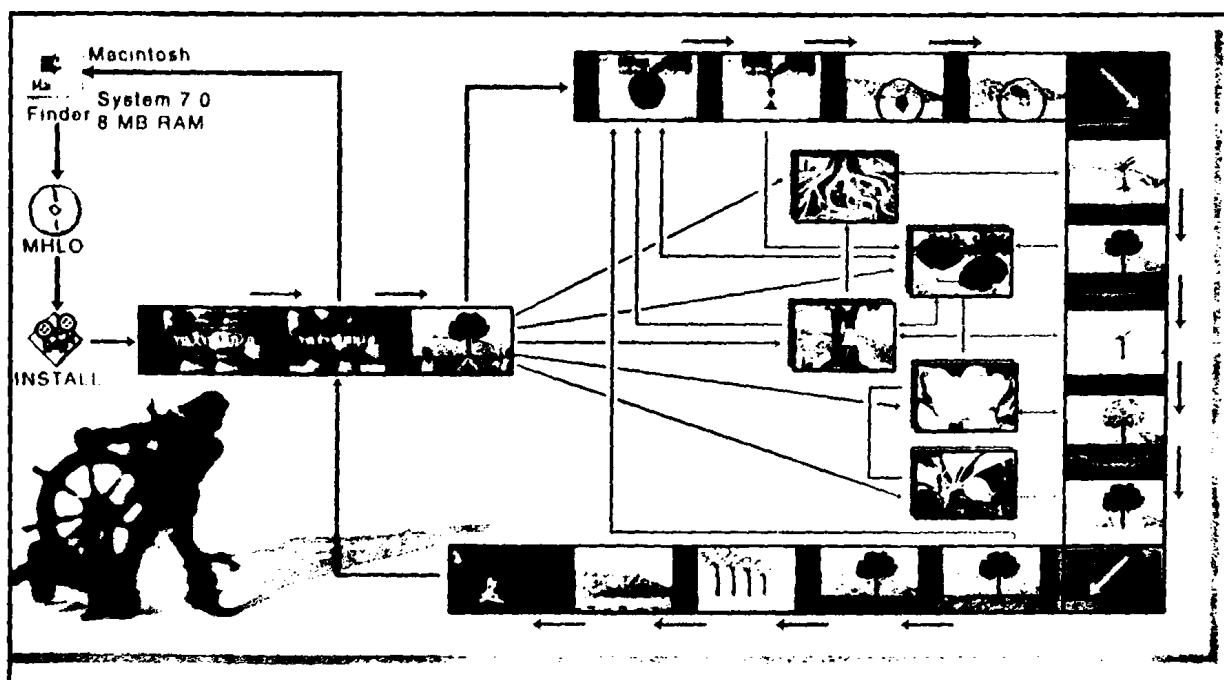
Η συνεχής παρακολούθηση της προόδου του μαθητή βοηθά το μαθητή, το δάσκαλο και τους γονείς να εντοπίσουν τις αδυναμίες του πρώτου. Το λογισμικό περιλαμβάνει σύστημα διαχείρισης της πορείας του μαθητή με ή χωρίς γνώση του για άμεσο έλεγχο και διαπίστωση της προοδευτικής διαδικασίας μάθησης.

Εσφαλμένη αντίληψη (misconception)

Ο εντοπισμός των σημείων που έχουν κατανοηθεί λανθασμένα διορθώνει τα λάθη και βελτιώνει τη γνώση. Το λογισμικό εντοπίζει τα σημεία που προέρχονται από εσφαλμένη αντίληψη και υποδείχνει τρόπους για διόρθωση των λαθών και σωστή κατανόηση.

Μεταφορά γνώσης (transfer of learning)

Οι μαθητές πρέπει να μπορούν να μεταφέρουν τη γνώση που αποκτήθηκε από ένα γνωστικό αντικείμενο σε άλλο και να συνδυάζουν τις γνώσεις τους για τη διεξαγωγή συμπερασμάτων.



Σχήμα 6.9 Οι δυνατότητες πλοήγησης στην εκπαιδευτική εφαρμογή πολυμέσων «ένα μήλο». Η εφαρμογή εκτός από το ότι παρέχει πληροφορίες για διάφορα μέρη ενός δένδρου (ρίζες, κορμός, φύλλα, άνθη, καρποί), παρέχει στο χρήστη δομές οργάνωσης δεδομένων και για άλλα παραδείγματα ή γνωστικά αντικείμενα

Το λογισμικό χρησιμοποιεί κοινές στρατηγικές και τρόπους σκέψης σε διάφορα γνωστικά αντικείμενα και προτρέπει το μαθητή να τις ακολουθήσει για τη διεξαγωγή συμπερασμάτων. Το σχήμα 6.9 παρουσιάζει μια οθόνη ενός πακέτου λογισμικού υπερμέσων/πολυμέσων, σχετικά με την ανάπτυξη μιας μηλιάς [Λαδιάς, 1997]. Η σχεδίαση του παραδείγματος εφαρμογής είναι τέτοια ώστε το μάθημα της φυτολογίας να οργανώνεται με τέτοιο τρόπο που να προκύπτει συστηματικά για το μαθητή ένα πλαίσιο διατεταγμένων κατηγοριών ανάλυσης και κατανόησης δεδομένων. Έτσι αποκτά τη δεξιότητα να μεταφέρει την αποκτηθείσα γνώση σε άλλα παραδείγματα, αλλά και άλλα γνωστικά αντικείμενα [Δήμου, Κατσίκης, Μικρόπουλος, 1995].

Με την εισαγωγή του υπολογιστή στην τάξη, η δραστηριότητα του μαθητή βρίσκεται στο κέντρο της μαθησιακής διαδικασίας. Ο μαθητής έχει τη δυνατότητα να επιλέξει, αναλύσει και να οργανώσει τα δεδομένα, για να οικοδομήσει τις προσωπικές του απαντήσεις. Βρίσκεται μπροστά σε καταστάσεις που του δημιουργούν ερωτήματα και συγκρούσεις, έχει στη διάθεσή του επιχειρήματα και πειράματα, που εντάσσονται λειτουργικά στις αναζητήσεις του.

Η εισαγωγή των υπολογιστών στην εκπαιδευτική διαδικασία και η εκπλήρωση των στόχων της όσον αφορά κυρίως στην CAI, ολοκληρώνεται σε δύο σημαντικά στάδια που περιλαμβάνουν τον αλφαριθμητισμό στους υπολογιστές (computer literacy) και την επέκταση του προγράμματος σπουδών.

Ο αλφαριθμητισμός στους υπολογιστές περιλαμβάνει την εξοικείωση με τη λειτουργία, χρήση και τον προγραμματισμό με στόχο τη χρησιμοποίησή τους ως εργαλεία σε διάφορα γνωστικά αντικείμενα και διαδικασίες. Συνήθως περιλαμβάνει εξοικείωση με το πληκτρολόγιο και το ποντίκι, χρήση έτοιμων πακέτων λογισμικού όπως επεξεργασίας κειμένου, λογιστικών φύλλων, βάσεων δεδομένων, σχεδίασης και ζωγραφικής και προγραμματισμό σε κάποια γλώσσα υψηλού επιπέδου.

Η επέκταση του προγράμματος σπουδών υλοποιείται με δραστηριότητες ανέμφικτες με άλλους τρόπους στο χώρο του σχολείου όπως προσομοιώσεις φυσικών, χημικών και βιολογικών φαινομένων και ανάπτυξη νοητικών μοντέλων σε προγραμματιστικά περιβάλλοντα. Επέκταση επίσης σημαίνει και τροποποίηση του προγράμματος σπουδών για τη θετική εισαγωγή της πληροφορικής τεχνολογίας χωρίς αρνητικές συνέπειες για όλα τα γνωστικά αντικείμενα και τη λειτουργία του σχολείου.

Η στάση των εκπαιδευτικών απέναντι στους υπολογιστές και ιδιαίτερα στην εισαγωγή τους στην εκπαίδευση δεν είναι πάντα θετική. Αυτό οφείλεται σε τρεις κυρίως λόγους. Στην έλλειψη οικειότητας με αυτό το είδος της τεχνολογίας, στην ανασφάλεια και στις πιθανά αρνητικές συνέπειες της νέας αυτής τεχνολογίας.

Η πλειονότητα των εν ενεργεία διδασκόντων κάθε βαθμίδας δεν έχει έλθει σε επαφή με υπολογιστές είτε κατά τις σπουδές τους ή μετά από αυτές και θεωρεί ότι απαιτείται να γίνουν ειδικοί και σ' αυτόν τον τομέα για να μπορέσουν να κάνουν χρήση της πληροφορικής στην εκπαιδευτική διαδικασία. Αυτός είναι ο κυριότερος



δισταγμός για την ένταξη των υπολογιστών στα σχολεία και την ενασχόληση μαζί τους, αντίληψη που θεραπεύεται με τον κατάλληλο τρόπο εισαγωγής και επιμόρφωσης των εκπαιδευτικών στη σύγχρονη τεχνολογία.

Πρώτα αποτελέσματα σχετικά με τη στάση Ελλήνων εκπαιδευτικών απέναντι στην εισαγωγή των υπολογιστών στη γενική εκπαίδευση, αφορούν σε ένα δείγμα 466 εκπαιδευτικών γενικών λυκείων και γυμνασίων όλων των ειδικοτήτων από τη Μακεδονία [Μπίκος, 1995]. Η στάση των εκπαιδευτικών είναι θετική απέναντι στην εισαγωγή της πληροφορικής στην εκπαίδευση. Το 37% των ερωτηθέντων διατείνεται ότι η επαφή του μαθητή με τον υπολογιστή θα πρέπει να αρχίζει από την πρωτοβάθμια εκπαίδευση, με την πλειονότητά τους (48.2%) να θεωρεί ως καταλληλότερη βαθμίδα, το γυμνάσιο. Σχετικά με τις προτιμήσεις για τον τρόπο ενσωμάτωσης των υπολογιστών στο υπάρχον αναλυτικό πρόγραμμα, το 43.5% του δείγματος συμφωνεί με τη λύση του ξεχωριστού μαθήματος. Το 40.3% συμφωνεί με την προοπτική της πληροφορικής ως βοηθητικού μέσου για τη διδασκαλία, 13.8% προτιμά την ένταξη θεμάτων της πληροφορικής και των νέων τεχνολογιών σε υπάρχοντα μαθήματα. Τέλος, ελάχιστοι (2.4%) δε δέχονται την εισαγωγή των υπολογιστών σε καμία περίπτωση. Παρά την έλλειψη δεδομένων για τις γνώσεις πληροφορικής των ερωτηθέντων, το μεγάλο ποσοστό που έχει θετική στάση για την εισαγωγή της πληροφορικής στην εκπαιδευτική διαδικασία και μάλιστα από το δημοτικό, αλλά και ως μέσο για τη διδασκαλία, μπορεί να ερμηνευθεί είτε ως ετοιμότητα εμπλοκής σε μια διαδικασία εξοικείωσης ή ως έκφραση προσδοκιών για δραστηριοποίηση στον τομέα της εκπαιδευτικής τεχνολογίας. Σχετικά με την ανασφάλεια, οι διδάσκοντες πιστεύουν σε πιθανή αντικατάστασή τους από τις μηχανές. Η εσφαλμένη αυτή αντίληψη διορθώνεται με την κατανόηση ότι ο υπολογιστής αποτελεί ένα ισχυρό εργαλείο στην υπηρεσία του διδάσκοντα που το εκμεταλλεύεται για αποδοτικότερη μάθηση έστω και με τις καινοτομίες που επιφέρει.

Ο τρίτος λόγος του σκεπτικισμού απέναντι στην εισαγωγή των υπολογιστών στο σχολείο έχει να κάνει με το μόνο γεγονός που ανταποκρίνεται στην πραγματικότητα. Τις αρνητικές συνέπειες από τη χρήση των υπολογιστών. Αυτές όμως εμφανίζονται με τη λανθασμένη χρήση των μηχανών. Σ' αυτές περιλαμβάνονται η αλόγιστη χρήση των μηχανών και ο κίνδυνος μετατροπής του υπολογιστή από μέσο σε αυτοσκοπό. Ένα γενικό πρόβλημα από το οποίο απορρέουν αρκετά άλλα, είναι η μεγάλη εμπιστοσύνη που αναπτύσσουν κυρίως οι μαθητές απέναντι στους υπολογιστές που οδηγεί σε κατάχρησή τους και έχει ως αποτέλεσμα τη συρρίκνωση ή υποβάθμιση ικανοτήτων που αναπτύσσονται χωρίς τη χρήση των υπολογιστών. Προβληματισμοί υπάρχουν και σχετικοί με θέματα εργονομίας, σχεδίασης των χώρων εργασίας με υπολογιστές, και τις επιπτώσεις στην υγεία των χρηστών. Αυτά αντιμετωπίζονται με την κατάλληλη σχεδίαση των εργαστηρίων πληροφορικής και τον προσδιορισμό του χρόνου χρησιμοποίησης των υπολογιστών. Σχετικές επιστημονικές έρευνες ασχολούνται με τέτοιου



είδους προδιαγραφές και αφορούν όχι μόνο σε θέματα υγείας αλλά και σωστής παιδαγωγικής αλληλεπίδρασης στην τάξη [Green et al, 1996].

Τα μειονεκτήματα της εισαγωγής της πληροφορικής στην υπηρεσία της εκπαίδευσης και ιδιαίτερα στο χώρο του σχολείου που εμφανίζονται ή προβάλλονται από φορείς, διδάσκοντες, γονείς και μαθητές αποφεύγονται με την κατανόηση του ρόλου του υπολογιστή ώστε να γίνεται σωστή εκμετάλλευσή αυτού και της φύσης κάθε προβλήματος ώστε να γίνεται δυνατή η πρόληψή του.

6.4 LOGO. Ένα ανοικτό περιβάλλον προγραμματισμού και δημιουργίας

Η πληροφορική ως η δομική επιστήμη με αποστολή της την υποστήριξη της αλληλεπίδρασης του ανθρώπου με τον κόσμο εμπεριέχει την εκπαίδευση και αποτελεί ένα ισχυρό μέσο για την εκπλήρωση των στόχων της. Αυτό γίνεται ιδιαίτερα αισθητό από τη θεώρηση της επιστήμης των υπολογιστών όχι μόνο από την πλευρά της επεξεργασίας δεδομένων, αλλά και της προσομοίωσης.

Ο υπολογιστής στην εκπαιδευτική διαδικασία συνεισφέρει στις παιδαγωγικές αρχές όπως η αναπαράσταση, η σύγκριση, η δημιουργικότητα, η αυτενέργεια, η προσομοίωση, προσφέροντας νέα εργαλεία και τρόπους θεώρησης της μαθησιακής διαδικασίας.

Η συμβολή του υπολογιστή όπως παρουσιάζεται στην προηγούμενη ενότητα βασίζεται αφ' ενός σε έτοιμα πακέτα λογισμικού και περιβάλλοντα εργασίας με τα οποία αλληλεπιδρά ο μαθητής και αφ' ετέρου σε προγραμματιστικά περιβάλλοντα, τις γλώσσες προγραμματισμού και τα εργαλεία ανάπτυξης λογισμικού. Τα δύο τελευταία επιτελούν ουσιαστικά δύο σκοπούς. Αποτελούν τον τρόπο για την κατανόηση της λειτουργίας της μηχανής, των δυνατοτήτων και των ορίων της και της σε βάθος εισαγωγής του χρήστη στο πνεύμα εργασίας με το συγκεκριμένο μέσο. Ο μαθητής-προγραμματιστής μαθαίνει μέσα από τον προγραμματισμό το δομημένο και ιεραρχικό τρόπο σκέψης και για την αντιμετώπιση προβλημάτων εκτός υπολογιστή. Ακόμη δημιουργεί ο ίδιος το περιβάλλον εργασίας της προτίμησής του για διάφορα θέματα. Έτσι είναι σε θέση να κατασκευάζει παιχνίδια εκπαιδευτικού περιεχομένου, να κτίζει βάσεις δεδομένων, να προγραμματίζει προσομοιώσεις, να δημιουργεί μικρόκοσμους σε διάφορα γνωστικά αντικείμενα. Μ' αυτόν τον τρόπο ο μαθητής αισθάνεται κυρίαρχος της νέας τεχνολογίας και ταυτόχρονα έρχεται σε επαφή με τις ιδέες της επιστήμης και τη δημιουργία διανοητικών μοντέλων. Έτσι οι άνθρωποι που θεωρούν τους εαυτούς τους "ανθρωπιστές" δε θα αισθάνονται αποξενωμένοι αλλά μέρος της διαδικασίας κατασκευής πληροφορικής παιδείας [Papert, 1991].

Ένα σημαντικό θέμα που προκύπτει είναι η επιλογή του προγραμματιστικού περι-



βάλλοντος και συγκεκριμένα η γλώσσα προγραμματισμού που ενδείκνυται ως εργαλείο που ικανοποιεί τις παραπάνω απαιτήσεις και επηρεάζει άμεσα την πληροφορική παιδεία. Το πρόβλημα γίνεται δυσκολότερο όταν ο χρήστης δεν προέρχεται ή δεν ακολουθεί τις λεγόμενες θετικές επιστήμες και δεν έχει συναντήσει αντικείμενα παραπλήσιας λογικής. Τις περισσότερες φορές ο χρήστης που είναι μαθητής μικρής ηλικίας, χρειάζεται απλά ένα ισχυρό εργαλείο για την αντιμετώπιση των προβλημάτων του και ένα περιβάλλον δράσης για συμβολή στην ανάπτυξη των νοητικών του μοντέλων.

Από τις καταλληλότερες γλώσσες προγραμματισμού για σωστή πληροφορική παιδεία θεωρείται η Logo. Πατέρας της θεωρείται ο Σείμους Παπέρ (Seymour Papert) που την ανέπτυξε όχι ως τεχνοκράτης μηχανικός λογισμικού αλλά ως μαθηματικός και παιδαγωγός. Καταλύτης για την ανάπτυξη της Logo, ήταν ένα από τα πρώτα εκπαιδευτικά προγράμματα για τα μαθηματικά στο οποίο συμμετείχε ο Παπέρ το 1965 [Solomon, 1996]. Παρατήρησε ότι οι μαθητές είχαν δυσκολίες στην άλγεβρα με έννοιες όπως οι μεταβλητές, όπως παρουσιάζονταν σε μια γλώσσα προγραμματισμού (TELCOMP) όπως η BASIC. Μια γλώσσα σχεδιασμένη για μηχανικούς και επιστήμονες δεν ήταν το καταλληλότερο εργαλείο για να μάθουν τα παιδιά μαθηματικά.

Η Logo έχει αποκτήσει τη φήμη της απλοϊκής γλώσσας για παιδιά, εξαιτίας της μεγάλης επιτυχίας της στην πρωτοβάθμια εκπαίδευση. Παρ' όλα αυτά, είναι μια από τις πλέον ισχυρές γλώσσες προγραμματισμού για προσωπικούς υπολογιστές. Αποτελεί μια διάλεκτο της LISP, της γλώσσας που χρησιμοποιείται σε ερευνητικά προγράμματα της επιστήμης των υπολογιστών, και ιδιαίτερα στην τεχνητή ευφυΐα. Μια γλώσσα προγραμματισμού είναι ισχυρή όχι γιατί μπορεί κάποιος να γράψει ένα πρόγραμμα που δε μπορεί σε άλλη γλώσσα, κάτω από αυτό το πνεύμα όλες οι γλώσσες είναι ίδιες. Η ισχύς της φαίνεται από τον τρόπο που βοηθά τον προγραμματιστή να συγκεντρωθεί στο συγκεκριμένο πρόβλημα, παρά να ασχολείται κυρίως με τους περιορισμούς της γλώσσας. Για παράδειγμα, στην Pascal, τη BASIC και τις γλώσσες που προέρχονται από τη FORTRAN, ο προγραμματιστής πρέπει να είναι σαφής ως προς τη διαχείριση της μνήμης του υπολογιστή. Έτσι, για την ομαδοποίηση 20 αριθμών ή χαρακτήρων ως μια ενότητα, πρέπει να ορισθεί ένας πίνακας (array) με 20 προκαθορισμένες θέσεις. Σε μελλοντική ανάγκη για 21 θέσεις, πρέπει ο πίνακας να ορισθεί από την αρχή. Στη Logo, όλη η διαδικασία γίνεται αυτόματα. Αν το πρόγραμμα δημιουργεί μια λίστα 20 αριθμών ή χαρακτήρων, ο χώρος για αυτήν τη λίστα παρέχεται αυτόματα. Η προσθήκη ενός επιπλέον στοιχείου στη λίστα, γίνεται και αυτή αυτόματα χωρίς παρέμβαση του προγραμματιστή.

Η Logo αποτελεί μια ολόκληρη θεωρία μάθησης βασισμένη στη γνώση και την επιστημολογική άποψη του έργου του Πιαζέ (Piaget) και την τεχνητή νοημοσύνη με την έννοια της γνωστικής επιστήμης. Μεταξύ των σκοπών της είναι και η απομυθοποίηση του υπολογιστή και του προγραμματισμού. Δεν είναι μόνο ένα εργαλείο προγραμματισμού αλλά ένας χώρος για να σκέφτεται ο προγραμματιστής σ' αυτόν.



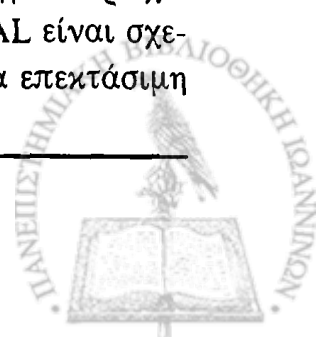
Η Logo είναι ένα περιβάλλον εργασίας, ένας τρόπος σκέψης για τους υπολογιστές, τη μάθηση και το συνδυασμό τους. Το περιβάλλον της Logo περιλαμβάνει τον υπολογιστή, τη γλώσσα προγραμματισμού, αλλά και συσκευές που ελέγχονται από τον υπολογιστή όπως χελώνες ρομπότ και συστήματα έκφρασης του μαθητή με λέξεις, ήχους, εικόνες, και κίνηση. Ο μαθητής γίνεται σχεδιαστής και ερευνητής. Έννοιες της πληροφορικής όπως μεταβλητές, διαδικασίες, συνδέονται με τον ανθρώπινο τρόπο σκέψης και γίνονται ζωντανά εργαλεία για επίλυση προβλημάτων. Η επίλυση προβλημάτων είναι μια δυναμική διεργασία ελέγχου ιδεών, παρατήρησης της εξέλιξής τους, αλλαγών, και αναμόρφωσής τους. Τα λάθη δεν είναι μη αποδεκτά, αφού αναδεικνύουν νέα αποτελέσματα, και οδηγούν σε νέους τρόπους σκέψης. Η διόρθωση λαθών είναι εμπειρία του φυσικού τρόπου σκέψης, αφού τα παιδιά μαθαίνουν κάνοντας και σκέπτονται αυτό που κάνουν.

Είναι ένα ανοικτό, επεκτάσιμο περιβάλλον προγραμματισμού και ανάπτυξης. Εκτός από τις εντολές της, ο χρήστης δημιουργεί εύκολα νέες εντολές-διαδικασίες σύμφωνα με τις ανάγκες και επιθυμίες του. Υπάρχει η δυνατότητα κατασκευής δομών ελέγχου και μετάφρασης των εντολών της γλώσσας σε άλλη φυσική γλώσσα όπως η Ελληνική.

Οι διαδικασίες προσφέρουν μικρές αυτόνομες δομές που συνδυάζονται και παράγουν μεγαλύτερες και πολυπλοκότερες. Ο τρόπος γραφής των διαδικασιών (π.χ. ΤΟ ΠΟΛΥΓΩΝΟ) δηλώνει μια μεταφορά, ότι αυτό που κάνει ο προγραμματιστής είναι να "διδάσκει" στον υπολογιστή μια νέα δεξιότητα. Στην εκπαιδευτική διαδικασία, οι μεταφορές είναι τα βασικά εργαλεία για την κατανόηση νέων εννοιών [Harvey, 1986]. Μ' αυτόν τον τρόπο δομείται ο τρόπος αντιμετώπισης των προβλημάτων προσφέροντας τμηματικό, ιεραρχικό και σταδιακό τρόπο σχεδίασης της λύσης τους. Επιπλέον γίνεται ευκολότερος ο έλεγχος και η εκσφαλμάτωση του τελικού προγράμματος.

Ο δομημένος τρόπος σκέψης διευκολύνει την επίλυση προβλημάτων και βοηθά στην ανάπτυξη πνευματικών δεξιοτήτων υψηλού επιπέδου. Ανεξάρτητες διαδικασίες της Logo που φτιάχνονται από τον ίδιο ή άλλους μαθητές, μπορούν να χρησιμοποιηθούν με κατάλληλους συνδυασμούς για την επίλυση νέων περισσότερο πολύπλοκων προβλημάτων. Γενικευμένες διαδικασίες που αντιμετωπίζουν μια ομάδα προβλημάτων σχεδιάζονται και αναπτύσσονται μια φορά από το μαθητή και χρησιμοποιούνται όταν απαιτείται. Η λύση του νέου προβλήματος κτίζεται κομμάτι - κομμάτι, οικοδομώντας τη γνώση [Papert, 1991]. Έτσι η εκπαιδευτική διαδικασία επιταχύνεται χωρίς να μηχανοποιείται αφού η επίλυση προβλημάτων βασίζεται στις ιδέες των ίδιων των μαθητών.

Κάθε εντολή μιας γλώσσας προγραμματισμού αποτελεί ένα δομικό κομμάτι για την ολοκλήρωση συγκεκριμένων διεργασιών. Η COBOL για παράδειγμα περιέχει εντολές χρήσιμες για προβλήματα σχετικά με επιχειρήσεις, ενώ η PASCAL είναι σχεδιασμένη για την εκμάθηση εννοιών προγραμματισμού. Η Logo είναι μια επεκτάσιμη



γλώσσα. Ο εκπαιδευτικός αλλά και ο μαθητής μπορούν να φτιάξουν δικές τους εντολές, μια προσωπική έκδοση της γλώσσας που ικανοποιεί τις ανάγκες τους. Αυτό προσθέτει μια νέα διάσταση στην επίλυση προβλημάτων που συμβαδίζει με τις τάσεις της εκπαίδευσης. Στο κοντινό μέλλον, ένα σύνολο συγκεκριμένων εργαλείων λογισμικού θα χρησιμοποιείται για την υποστήριξη της διδακτικής πράξης σε κάθε γνωστικό αντικείμενο [Yoder & Moursund, 1990]. Εκπαιδευτικοί και μαθητές που έχουν ασχοληθεί με την ανάπτυξη τέτοιου είδους εργαλείων στο περιβάλλον της logo αποκτούν τις πνευματικές δεξιότητες και μεταφέρουν τις γνώσεις σε νέα γνωστικά αντικείμενα και καταστάσεις.

Με τη χρήση της logo στην εκπαιδευτική διαδικασία, ο εκπαιδευτικός βλέπει τον εαυτό του να συμμετέχει ισότιμα με τους μαθητές σε ένα κοινό πρόγραμμα έρευνας, σχεδίασης και ανάπτυξης. Ο εκπαιδευτικός αντιμετωπίζει καταστάσεις και προβλήματα για πρώτη φορά. Δε γνωρίζει εκ των προτέρων τη λύση του προβλήματος ή τον τρόπο λύσης που επιλέγει ο κάθε μαθητής. Για τα περισσότερα παιδιά, η ιδέα μιας τιμίας και ισότιμης σχέσης με τον εκπαιδευτικό είναι κάτι νέο που εμπνέει. Έτσι η παρουσία του υπολογιστή ξεπερνά την απλή χρήση, συνεισφέρει στη δημιουργία μιας νέας κουλτούρας. Ο εκπαιδευτικός γνωρίζοντας τους κώδικες αυτής της κουλτούρας, χρησιμοποιεί τη γλώσσα της για να συζητά με ενδιαφέροντες τρόπους για πράγματα που όλοι γνωρίζουν και ενδιαφέρονται. Αποτελεί ενεργό μέρος της κουλτούρας της "τεχνολογικής" τάξης. Έχει γνώση ισχυρών ιδεών, σχεδιάζει προγράμματα εργασίας, καθοδηγεί τους μαθητές. Ο παιδαγωγός γνωρίζει πως να παρατηρεί αυτούς που εμπλέκονται σε διαδικασίες σκέψης, μάθησης, αγωνίας, χαράς, πως και πότε να παρεμβαίνει ή να αφήνει το μαθητή να προσπαθεί μόνος ή σε συνεργασία με συμμαθητές του. Έτσι υλοποιείται ο κύριος στόχος που είναι η βελτίωση της εικόνας που έχει ο κάθε μαθητής για τον εαυτό του ως μαθητή και ως ενεργού μέλους στην εκπαιδευτική διαδικασία.

Για τη επίλυση προβλημάτων απαιτείται γνώση στο πεδίο του προβλήματος και ο κατάλληλος συνδυασμός δεξιοτήτων χαμηλού και υψηλού επιπέδου. Κατά την εργασία με τη logo ο μαθητής ανακαλύπτει τον τρόπο μάθησής του. Χρησιμοποιεί μεμονωμένες εντολές, δημιουργεί δικές του, κτίζει με βάση άλλες διαδικασίες. Συνεργάζεται με συμμαθητές και τον εκπαιδευτικό.

Η logo προσφέρει το περιβάλλον για την ελεύθερη δράση του μαθητή και την ενεργή συμμετοχή του στην εκπαιδευτική διαδικασία. Ένα περιβάλλον για αξιολόγηση του μαθητή από τον ίδιο, για έρευνα στην επίλυση προβλημάτων.

Βασικό χαρακτηριστικό της logo αποτελεί το γραφικό περιβάλλον επικοινωνίας (Graphics User Interface, GUI) με το χρήστη, που υλοποιείται με τη χελώνα και τα γραφικά της. Στα πρώτα στάδια ανάπτυξης της γλώσσας, το περιβάλλον της βασιζόταν σε χαρακτήρες, λέξεις και σύμβολα με αποτέλεσμα να μην υπάρχει η επιθυμητή αμεσότητα στην επικοινωνία με το χρήστη. Το όνομα logo προέρχεται από την ελληνική λέξη



“λόγος” αφού το περιβάλλον σχεδιάστηκε αρχικά για διαχείριση της γλώσσας, σε αντίθεση με τις άλλες γλώσσες προγραμματισμού με έμφαση στους αριθμητικούς υπολογισμούς. Ο Παπέρ εισήγαγε τα γραφικά της χελώνας, μια έμπνευση από ένα κινούμενο ρομπότ. Έτσι η Logo αποκτά γραφικό περιβάλλον επικοινωνίας, κάτι που αποτελεί τη σύγχρονη τάση στο λογισμικό κάθε είδους. Διεργασίες κάθε είδους αναπαριστούνται με τα γραφικά και τα σχήματα που δημιουργεί η χελώνα με τις οδηγίες του χρήστη. Οι οδηγίες για τις θέσεις και τον προσανατολισμό της περιέχουν διαφορεικό τρόπο σκέψης και εισάγουν τις ιδέες του διαφορετικού λογισμού. Γεωμετρικά αντικείμενα περιγράφονται με διαδικασίες χωρίς την απαίτηση για “προκαθορισμένες” εξισώσεις της αναλυτικής γεωμετρίας που εξ’ άλλου διδάσκεται σε προχωρημένα μαθηματικά και όχι στην απλή αριθμητική. Μ’ αυτό το σκεπτικό της Logo ο μαθητής οδηγεί και προγραμματίζει τον υπολογιστή και δε συμβαίνει το αντίθετο. Επιπλέον, για το υπό μελέτη αντικείμενο πρώτα από όλα αναπτύσσεται και ολοκληρώνεται η ιδέα και στο τέλος ακολουθεί ως λογική συνέπεια ο ορισμός που συνήθως αποτελεί συμπέρασμα του μαθητή.

Οι μικρόκοσμοι είναι ένα δεύτερο γνώρισμα της Logo. Ο Παπέρ τους ονομάζει χαρακτηριστικά ως εκκολαπτήρια γνώσης και ουσιαστικά αποτελούν προσομοιώσεις του κόσμου. Αποτελούνται από ένα σύνολο αντικειμένων και τελεστών που διαμορφώνουν ένα εσκεμμένα “φτωχό” περιβάλλον, το οποίο ο μαθητής μπορεί να εξερευνήσει ώστε να φτάσει στην κατανόηση των πραγμάτων, των διεργασιών και των σχέσεων του πραγματικού κόσμου, που αναπαριστούνται από αυτά τα αντικείμενα και τους τελεστές. Για το μαθητή ο μικρόκοσμος φαίνεται ως παιχνίδι, ένα σύνολο προγραμμάτων για εκτέλεση, ή ως μια συλλογή υπολογιστικών εργαλείων - διαδικασιών που έχει στη διάθεσή του για να δημιουργήσει νέες διαδικασίες και παιχνίδια. Ο μαθητής μπορεί να δημιουργεί τους δικούς του μικρόκοσμους σε κάθε γνωστικό αντικείμενο και να εξερευνά τα θέματά του “από μέσα”. Το κύριο παιδαγωγικό χαρακτηριστικό ενός μικρόκοσμου είναι η οργάνωσή του γύρω από φαινόμενα παρά γύρω από προβλήματα [Miller & Bliss, 1993]. Ο Papert προτείνει ότι ένας μικρόκοσμος πρέπει να είναι πλήρως ορισμένος και να περιέχει όλα τα απαραίτητα στοιχεία, υποθέσεις και περιορισμούς. Οι μικρόκοσμοι του Papert περιέχουν αντικείμενα όπως οι χελώνες και τα γραφικά, που μπορούν να ενεργούν όπως να μεταβάλλουν το σχήμα τους ή να κινούνται. Περιέχουν πρώτιστα “κομμάτια γνώσης” σε διάφορα σημεία, που πρέπει να ανακαλύψει ο μαθητής.

Έτσι η εκμετάλλευση των παραπάνω ιδεών στην κατανόηση των νόμων του Νεύτωνα για την κίνηση, οδήγησε στη δημιουργία της δυναμοχελώνας, μιας χελώνας που δεν είναι πλέον γεωμετρικό σημείο αλλά αντικείμενο με μάζα και ταχύτητα [Papert, 1991]. Με τη μάθηση με τη χρήση των μικρόκοσμων, οι μαθητές μπορούν να μεταφέρουν τη γνώση από το μικρόκοσμο στον πραγματικό κόσμο, μπορούν να συνδυάσουν τα “κομμάτια γνώσης” που αποκτούν ώστε να παράγουν μια “ολοκληρωμένη



γνώση", μπορούν να ανταλλάσσουν πληροφορίες και γνώσεις με μαθητές άλλων ηλικιών, αφού οι πρωτογενείς διαδικασίες του μικρόκοσμου αποτελούν ένα συνεχές και ενιαίο σύνολο.

Με τη Logo, τα ερεθίσματα από το φυσικό κόσμο μεταφέρονται εύκολα στην οθόνη του υπολογιστή και η γνώση αποκτάται μέσω των απαραίτητων εντολών για την καθοδήγηση της χελώνας στην εκτέλεση του έργου της.

Ένας από τους πλέον σημαντικούς λόγους για τη χρήση του υπολογιστή στην επίλυση προβλημάτων είναι η αναπαράστασή τους στον υπολογιστή. Η Logo με τη γεωμετρία χελώνας είναι ένα ιδιαίτερα ισχυρό εργαλείο για την αναπαράσταση γραφικά προσανατολισμένων προβλημάτων. Συνεισφέρει επιπλέον και στη μετατροπή άλλων ώστε το αποτέλεσμα τους να δίνεται με οπτικοποιημένο τρόπο μέσω γραφικών. Ο χρήστης σχεδιάζει μια εικόνα στην οθόνη, την αντιπαραβάλλει με αυτό που έχει υπόψη του και την αλλάζει εύκολα βλέποντας άμεσα το αποτέλεσμα της δράσης του.

Η αναπαράσταση προβλημάτων στον υπολογιστή, η μοντελοποίηση (computer modeling), διαφέρει από το μοντέλο ενός αντικειμένου υπό κλίμακα, ένα μοντέλο στο χαρτί, ή ένα νοητικό μοντέλο. Για ένα συγκεκριμένο πρόβλημα μπορούν να αναπτυχθούν πολλά διαφορετικά μοντέλα. Τροποποιούνται εύκολα και γρήγορα, εμφανίζοντας νέα αποτελέσματα. Σε ένα μοντέλο για τη θερμομόνωση ενός κτιρίου, μεταβάλλοντας το μονωτικό υλικό φαίνονται τα αποτελέσματα για την εξοικονόμηση ενέργειας μέσα σε λίγα δευτερόλεπτα στην οθόνη του υπολογιστή.

Στη Logo, η εντολή RANDOM που παράγει τυχαίους αριθμούς, χρησιμεύει συχνά στη μοντελοποίηση προβλημάτων. Σε μια τάξη με 20 μαθητές για την επιλογή κάποιου από αυτούς στην τύχη, γίνεται κλήρωση. Η χρήση της εντολής RANDOM στο κατάλληλο πρόγραμμα της Logo, παράγει έναν τυχαίο αριθμό ή όνομα μαθητή σε ελάχιστο χρόνο. Η διαφοροποίηση από άλλες γλώσσες προγραμματισμού που περιλαμβάνουν γεννήτρια τυχαίων αριθμών, είναι ότι το αποτέλεσμα μπορεί να αποδοθεί γραφικά εύκολα από το μαθητή, δίνοντας μια "ανθρωπομορφική" υπόσταση σε έναν αριθμό.

Στόχος του βιβλίου δεν είναι ο προγραμματισμός στη γλώσσα Logo, αλλά η χρήση της ως εργαλείου στην εκπαιδευτική πράξη. Έτσι, δεν παρουσιάζονται όλες οι δυνατότητες της γλώσσας, ούτε όλες οι εντολές της. Συγκεκριμένα, χρησιμοποιούνται περίπου τριάντα πρωτογενείς διαδικασίες (εντολές) με τις οποίες υλοποιούνται από πολύ απλές μέχρι πολύπλοκες διεργασίες όπως δημιουργία φράκταλς και προγράμματα με στοιχεία τεχνητής ευφυΐας. Εξάλλου είναι δυνατή η ανάπτυξη προγραμμάτων με ένα μικρό σύνολο εντολών της Logo και στρατηγικές προγραμματισμού για συνδυασμό και συνένωσή τους. Οι μικροί μαθητές πρέπει να συγκεντρώνονται σε ιδέες που φέρουν άμεσα αποτελέσματα και κεντρίζουν το ενδιαφέρον τους, από την εκμάθηση όλου του συντακτικού της γλώσσας. Είναι σημαντικό να δουν τα παιδιά από την πρώτη τους κιόλας επαφή, ότι μπορούν να κάνουν κάτι με τον υπολογιστή που δε μπορούσαν να το κάνουν χωρίς αυτόν.



Αναφορές

- Arons A. "Οδηγός διδασκαλίας της Φυσικής", Τροχαλία 1992
- Bertrand Y., "Σύγχρονες Εκπαιδευτικές Θεωρίες", Ελληνικά Γράμματα, Αθήνα 1994
- Bork A., "Interactive learning: Milikan Lecture, American Association of Physics Teachers, London, Ontario, June, 1978", Am. J. Phys. 47, 5 1979
- Δήμου Γ., Κατσίκης Α., Μικρόπουλος Α., "Ερευνητική προσέγγιση της χρήσης αλληλεπιδραστικών περιβαλλόντων στη διδασκαλία της γεωγραφίας", Πρακτικά Β Πανελ. Συνεδρίου Διδακτική των μαθηματικών & πληροφορική στην εκπαίδευση, 557-566, Λευκωσία 1995
- Green E. E., Cook P. F., Bolt L., "Fitting New Technologies into Traditional Classrooms: Two Case Studies in the Design of Improved Learning Facilities", Educ. Techn. 27-38, July-August 1996
- Harvey B., "Computer Science Logo Style. Volume 1: Intermediate Programming", The MIT Press, Cambridge, MA 1986
- Λαδιάς Τ., Χαλκίδης Α., Σαριδάκη Α., Μπέλλου Ι., Μικρόπουλος Α., "Εκπαιδευτικά κινούμενα σχέδια με τη βοήθεια Η/Υ. Ένα παράδειγμα: Εισαγωγή στη φυσική των laser", Ε' Παν. Συν. Φυσικής, Αθήνα 1989
- Λαδιάς Τ., "Να ένα μήλο", ΤΕΧΝΟΜΑΘΕΙΑ Ι, ΕΠΕΤ ΙΙ 1997
- Lysaught J.P., Williams C.M. "A Guide to Programmed Instruction", J. Wiley 1963
- Μικρόπουλος Α., "Η πληροφορική ως διδακτική τεχνική. Εκπαιδευτικά κινούμενα σχέδια", τα εκπαιδευτικά 27-28, 143 1992
- Mikropoulos T. A., Kossivaki P., Katsikis A., and Savranides C., "Computers in Pre-school Education: An Interactive Environment", J. Computing in Childhood Education, 5(3/4), 339-351 1994
- Mikropoulos T. A., "Virtual Environments in Science Education", "Virtual Reality in Education & Training", Intl. Conf. Loughborough, UK 1997
- Miller H. G., Bliss J., "Expressing the student's concepts versus exploring the teacher's: Issues in the design of microworlds for teaching", J. Edu. Comp. Res. 9(1) 89-113, 1993
- Μπίκος Κ., "Εκπαιδευτικοί και Ηλεκτρονικοί Υπολογιστές", Κυριακίδη, Θεσσαλονίκη 1995
- Papert S., Νοητικές Θύελλες, ΟΔΥΣΣΕΑΣ, Αθήνα 1991
- Plomp T, Anderson R. E., and Kontogiannopoulou-Polydorides G., "Cross National Policies and Practices on Computers in Education", Kluwer Academic Publishers, Dodrecht 1996
- Reif F., "Scientific approaches to science education", Phys. Tod. 48, Nov. 1986
- Σιμάτος Α., "Τεχνολογία και Εκπαίδευση", Εκδόσεις Πατάκη, Αθήνα 1995



- Solomon C., in "Designing multimedia environments for children" by Druin A., Solomon C., John Wiley & Sons, NY 1996
- Τζιμογιάννης Α., Μικρόπουλος Α., Κουλαϊδής Β., "Ο υπολογιστής στη διδασκαλία της Φυσικής: Μια άμεση εφαρμογή με χρήση φύλλων εργασίας", Σύγχρονη Εκπαίδευση, 85, 38-46, 1995
- Yoder S., and Moursund D., "LogoWriter for Educators: A Problem Solving Approach", ISTE Publications, Oregon 1990
- ΥΠΕΠΘ, Εγκύκλιος 10/Γ2/4294/ 27-893 "Ανακοίνωση του μεταβατικού προγράμματος Πληροφορικής για το Γυμνάσιο, έτη 1993/94 και 1994/95"
- ΥΠΕΠΘ, Εγκύκλιος Γ2/2713/3/5/1993 "Επέκταση εξοπλισμού σε σχολικές μονάδες (Γυμνάσια) και Π.Ε.Κ."
- Vockel E.L, Schwartz E.M. "The computer in the classroom", McGraw-Hill 1992

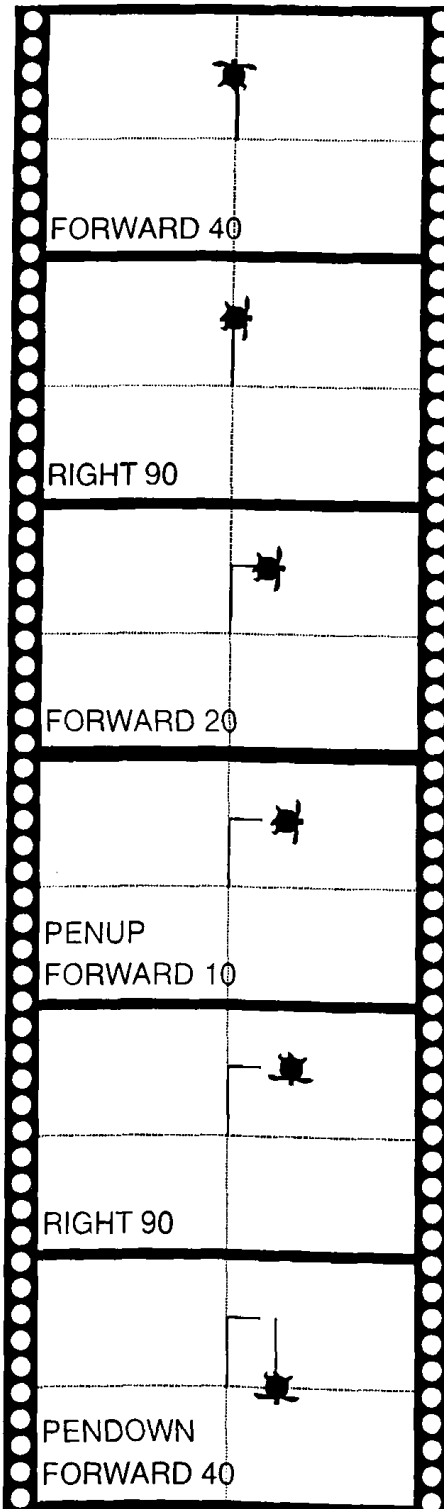




7 Η ουσία των πραγμάτων

Η χελώνα ζωγραφίζει τη λέξη «ΓΗ»

Η χελώνα εκτελώντας τις οδηγίες μας ζωγραφίζει τα γράμματα Γ και Η.

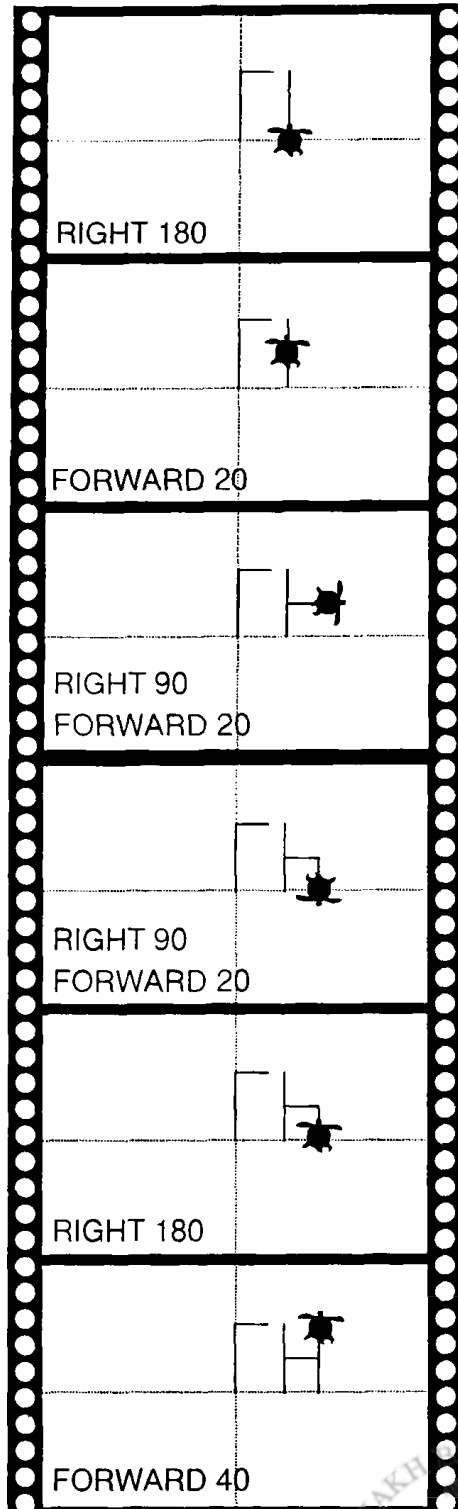


Οι οδηγίες προς τη χελώνα στη γλώσσα logo λέγονται εντολές. Το FORWARD είναι μια εντολή που διατάζει τη χελώνα να κινηθεί προς τα εμπρός. Ο αριθμός που ακολουθεί (το 40), δηλώνει τα βήματα που θα κάνει η χελώνα.

Μια άλλη εντολή είναι η RIGHT που στρίβει τη χελώνα δεξιά. Ο αριθμός που ακολουθεί (το 90), δηλώνει τη γωνία σε μοίρες.

Η χελώνα αφήνει ίχνη που φανερώνουν τη διαδρομή που έκανε. Η χελώνα μπορεί να μην αφήνει ίχνη με την εντολή PENUP.

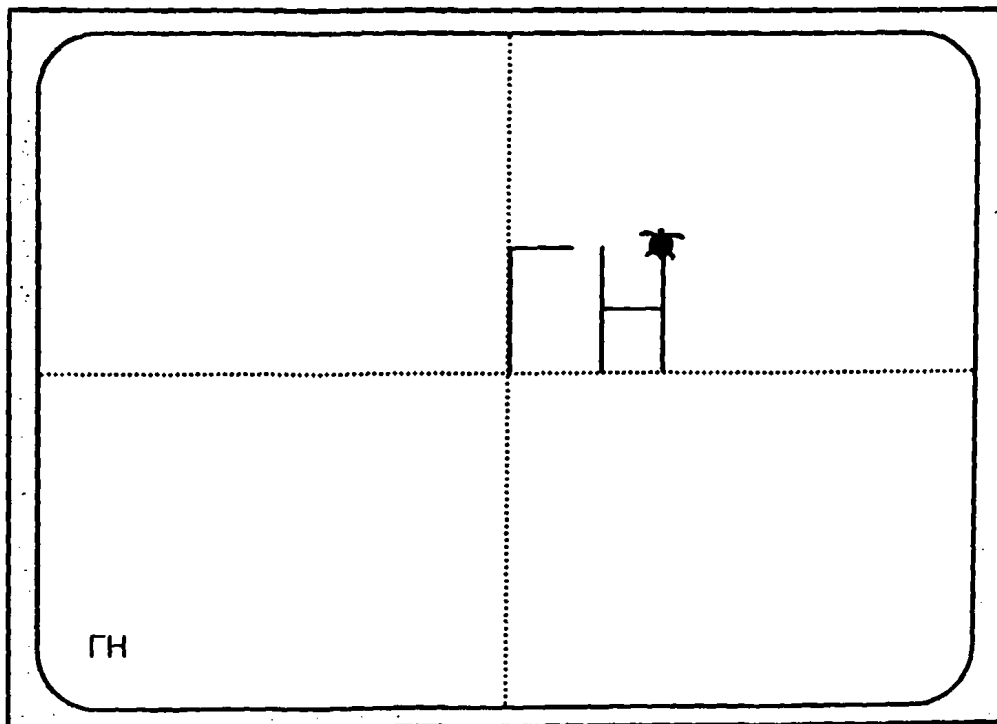
Με την εντολή PENDOWN η χελώνα μπορεί να αφήνει ίχνη. Η γλώσσα logo έχει ένα συγκεκριμένο λεξιλόγιο, δηλαδή διαθέτει προκαθορισμένες εντολές. (FORWARD, RIGHT, PENUP, PENDOWN).



Η αναγκαιότητα της διαδικασίας

Αν χρειαστεί να ξανασχεδιάσουμε τη ΓΗ θα πρέπει να επαναλάβουμε μια-μια όλες τις προηγούμενες εντολές. Για να το αποφύγουμε μπορούμε να φτιάξουμε μια δικιά μας διαταγή που να σχεδιάζει το ΓΗ. Η διαταγή που ορίζεται μ' αυτόν τον τρόπο λέγεται διαδικασία (procedure). Μια διαδικασία χρησιμοποιώντας το λεξιλόγιο της γλώσσας, πρέπει να δίνει σαφείς και αναλυτικές οδηγίες στον υπολογιστή για τον τρόπο που θα εκτελέσει κάποιο έργο.

```
TO ΓΗ
FORWARD 40
RIGHT 90
FORWARD 20
PENUP
FORWARD 10
PENDOWN
RIGHT 90
FORWARD 40
RIGHT 180
FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 180
FORWARD 40
END
```



Ο ορισμός μιας διαδικασίας ξεκινάει με το «TO» και το όνομά της (στη συγκεκριμένη περίπτωση το ΓΗ).

Στη συνέχεια ακολουθεί ένα τμήμα με εντολές που περιγράφουν τι πρέπει να κάνει η χελώνα ώστε να φέρει σε πέρας το συγκεκριμένο έργο που θα πραγματοποιεί η διαδικασία.

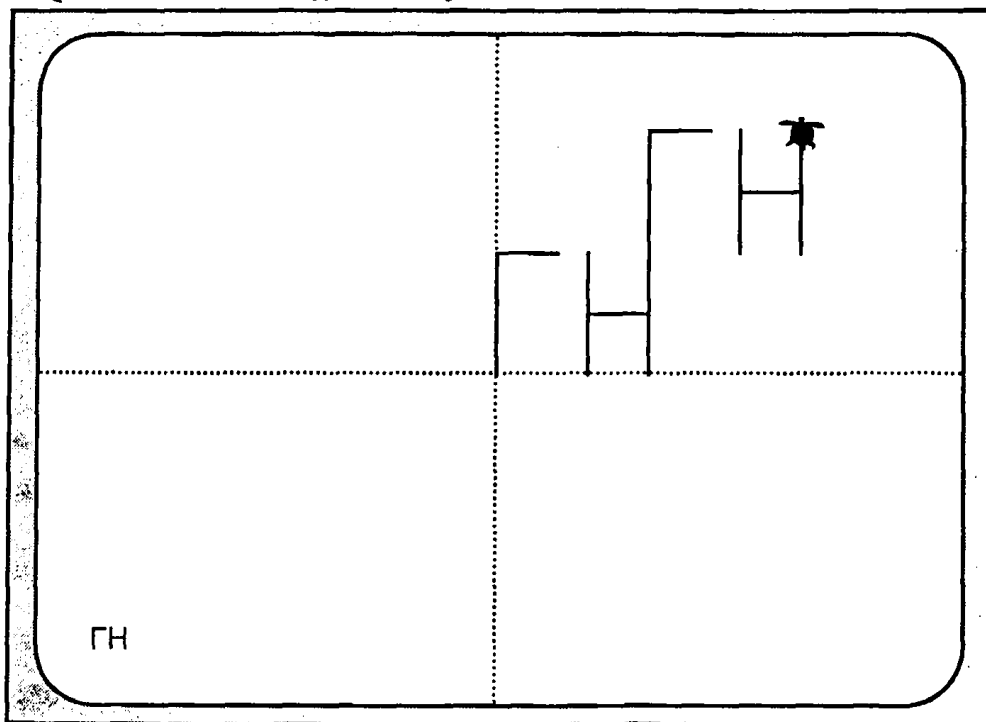
Η διαδικασία τελειώνει με την εντολή END.

Από εδώ και στο εξής όταν θα "λέμε" στη χελώνα ΓΗ, αυτή θα σχεδιάζει το ΓΗ.

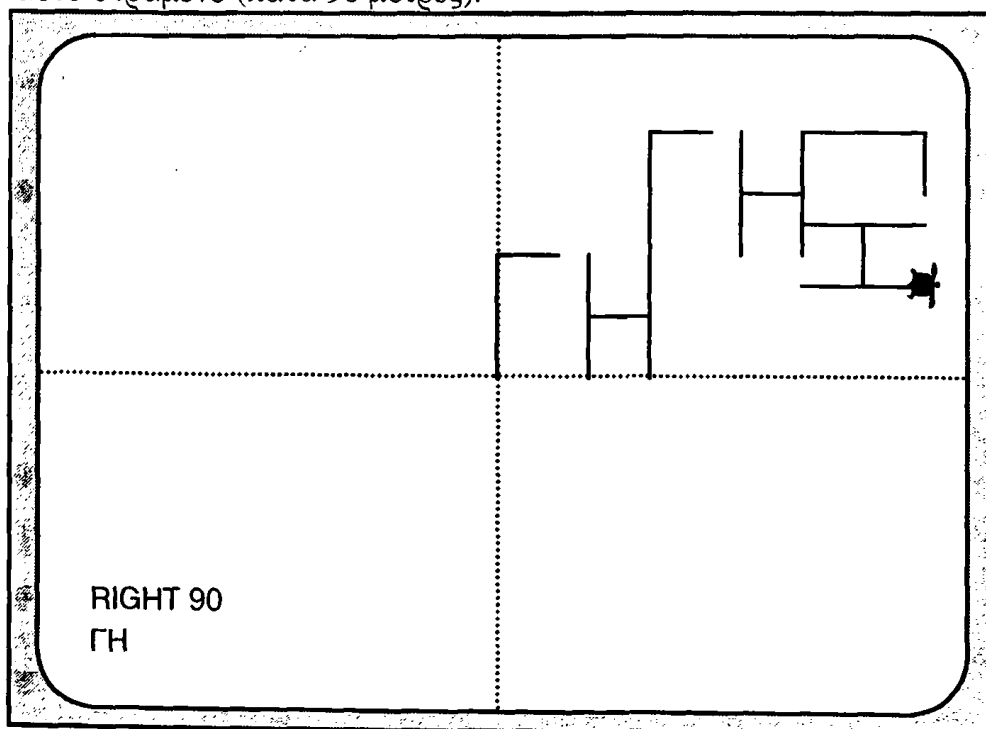


"Εκτελώντας" τη διαδικασία ΓΗ

Προηγουμένως δώσαμε τη διαταγή ΓΗ και η χελώνα ξεκίνησε από εκεί που ήταν και σχεδίασε το ΓΗ. Αν ξαναδώσουμε τη διαταγή ΓΗ, η χελώνα θα ξεκινήσει από το σημείο που βρίσκεται (στο πάνω δεξιά άκρο του Η) και θα σχεδιάσει ξανά το ΓΗ.



Αν στρίψουμε τη χελώνα δεξιά (90 μοίρες) και τη διατάξουμε να εκτελέσει τη διαταγή ΓΗ, τότε το ΓΗ που θα προκύψει θα είναι και αυτό στραμένο (κατά 90 μοίρες).



Καθαρίζοντας την οθόνη

Αν στην οθόνη υπάρχουν κάποια σχέδια και θέλουμε να "καθαρίσει η οθόνη" χρησιμοποιούμε την εντολή CG. Στο εξής όμως θα προσπαθήσουμε να "εξελληνίσουμε" τις εντολές και τις διαδικασίες που θα χρησιμοποιούμε. Έτσι θα αντικαταστήσουμε τις εντολές της Logo (όσες μπορούμε) με διαδικασίες που θα φτιάξουμε εμείς και θα έχουν ελληνικά ονόματα που θα είναι περισσότερο κατανοητά.

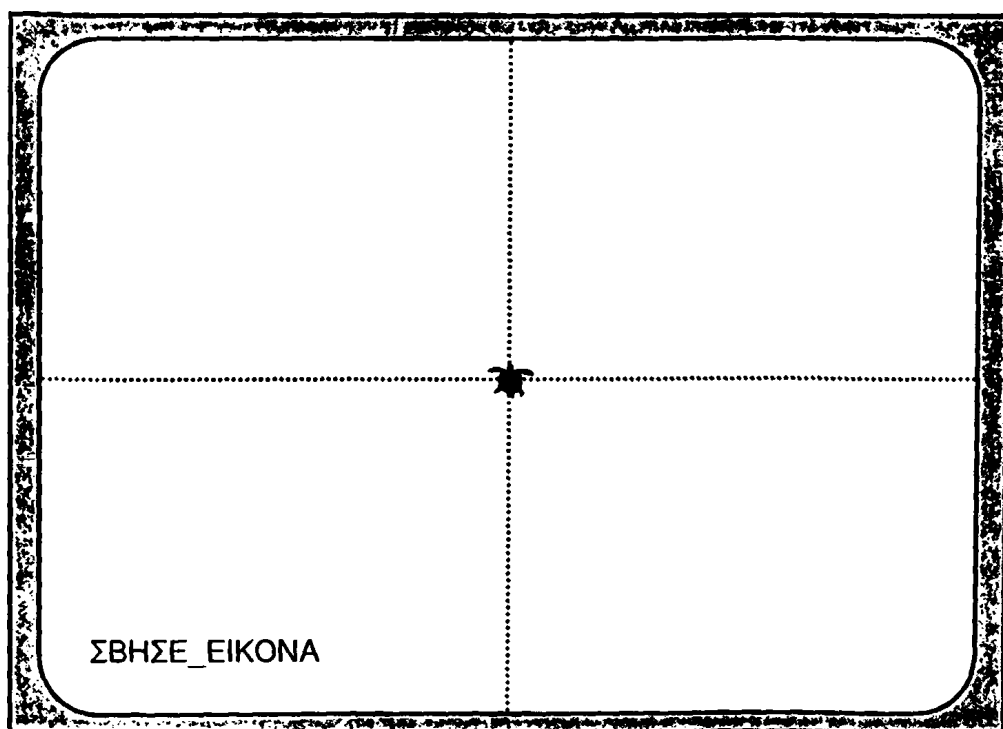
Ας φτιάξουμε τώρα μια διαδικασία που θα καθαρίζει την οθόνη σβήνοντας ό,τι σχήμα υπάρχει σ' αυτή.

```
ΤΟ ΣΒΗΣΕ_ΕΙΚΟΝΑ  
CG  
END
```

Η εντολή CG προέρχεται από τα αρχικά της ισοδύναμης εντολής Clear Graphics που στα αγγλικά σημαίνει καθάρισε την οθόνη.

Από εδώ και στο εξής όταν θέλουμε να καθαρίσουμε την οθόνη θα πληκτρολογούμε «ΣΒΗΣΕ_ΕΙΚΟΝΑ» όπως φαίνεται στο κάτω μέρος της οθόνης.

Αν το όνομα της διαδικασίας αποτελείται από περισσότερες από μια λέξεις αυτές πρέπει να συνδέονται με το χαρακτήρα της υπογράμμισης.



Η χελώνα κινείται σε ευθεία γραμμή

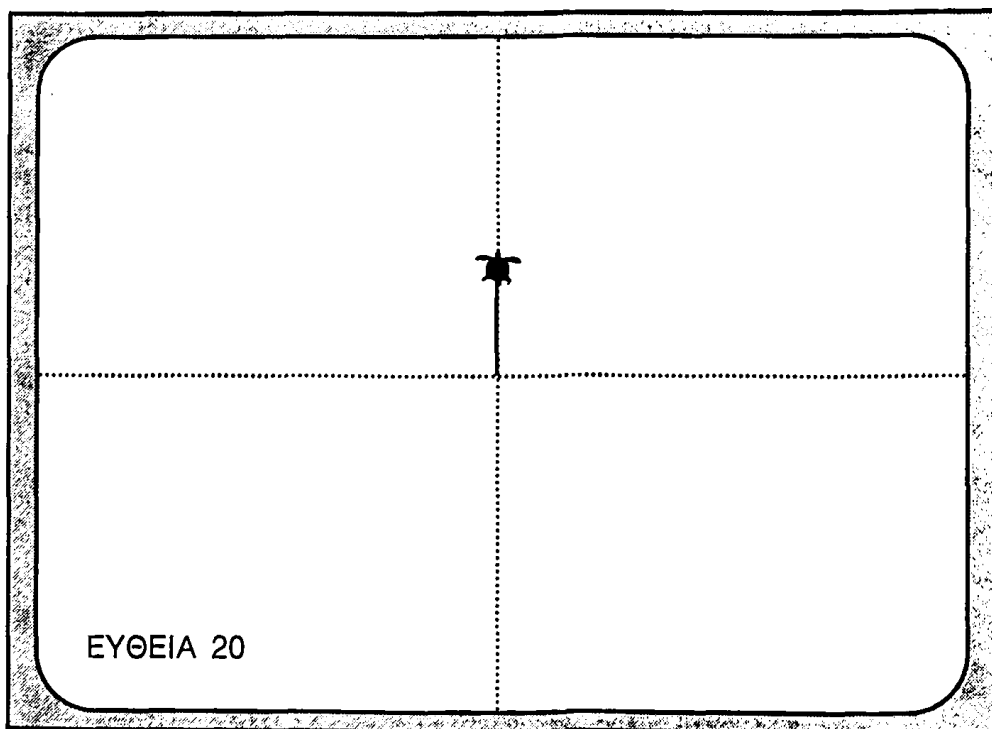
Ας μάθουμε τον υπολογιστή να κινεί τη χελώνα. Θα ορίσουμε μια διαδικασία που θα κάνει τη χελώνα να κινείται σε ευθεία γραμμή. Αυτό που χρειαζόμαστε είναι να καθορίζουμε (κάθε φορά που διατάζουμε τη χελώνα να κινηθεί) πόσα βήματα να κάνει πριν σταματήσει. Χρειάζεται λοιπόν κατά τον ορισμό της διαδικασίας εκτός από το όνομά της, να της δώσουμε και το όνομα μιας μεταβλητής (το ΜΗΚΟΣ) όπου θα φυλάει ο υπολογιστής το πλήθος των βημάτων που θα του λέμε να προχωρήσει η χελώνα κάθε φορά.

```
TO ΕΥΘΕΙΑ :ΜΗΚΟΣ
FD :ΜΗΚΟΣ
END
```

Η εντολή FD της Logo προέρχεται από την ισοδύναμη εντολή ForwarD που στα αγγλικά σημαίνει εμπρός.

Πρέπει να ξεκαθαριστεί ότι άλλο το όνομα μιας μεταβλητής και άλλο η τιμή της.

Το 20 είναι η τιμή της μεταβλητής (το περιεχόμενό της) και αναφερόμαστε σ' αυτή μέσα στο πρόγραμμα με το :ΜΗΚΟΣ.



«Μια βασική μαθηματική έννοια, της οποίας η κατανόηση διευκολύνεται με τη χελώνα είναι η ιδέα της μεταβλητής: η ιδέα της χρήσης ενός συμβόλου για να ονομαστεί μια άγνωστη οντότητα»
Seymour Papert

ΜΗΚΟΣ

20

Όταν διατάξουμε τον υπολογιστή να εκτελέσει τη διαδικασία ΕΥΘΕΙΑ αυτή θα πρέπει να συνοδεύεται και με τον αριθμό των βημάτων που θα κάνει η χελώνα.



Μαθαίνοντας τη χελώνα να στρίβει δεξιά

Το επόμενο βήμα για να κατευθύνουμε τη χελώνα μέσα από το πρόγραμμα, είναι να ορίσουμε μια διαδικασία που θα την κάνει να στρίβει.

Όπως στη διαδικασία ΕΥΘΕΙΑ πρέπει να πούμε στον υπολογιστή πόσα βήματα να κάνει, έτσι και στη διαδικασία της στροφής θα πρέπει να του πούμε πόσο να στρίψει. Το όνομα της μεταβλητής που θα περιέχει το πόσο θα στρίψει η χελώνα είναι MOIRES.

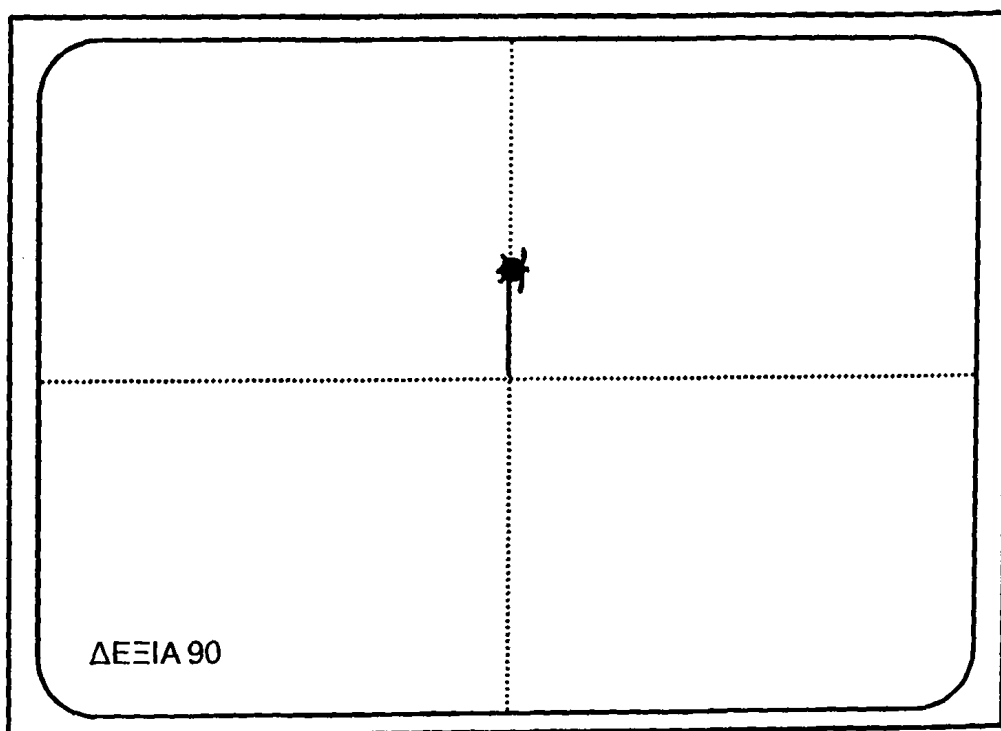
```
ΤΟ ΔΕΞΙΑ :MOIRES  
RT :MOIRES  
END
```

Η εντολή RT της Logo, προέρχεται από την ισοδύναμη εντολή Right που στα αγγλικά σημαίνει δεξιά.

Η μονάδα μέτρησης που χρησιμοποιείται εδώ είναι οι μοίρες.

«Ο ενήλικος θα αναγνωρίσει γρήγορα αυτούς τους αριθμούς ως μετρήσεις γωνίας σε μοίρες.

Για τα περισσότερα παιδιά αυτοί οι αριθμοί πρέπει να εξερευνηθούν και αυτό είναι μια διασκεδαστική διαδικασία παιχνιδιού.»
Seymour Papert

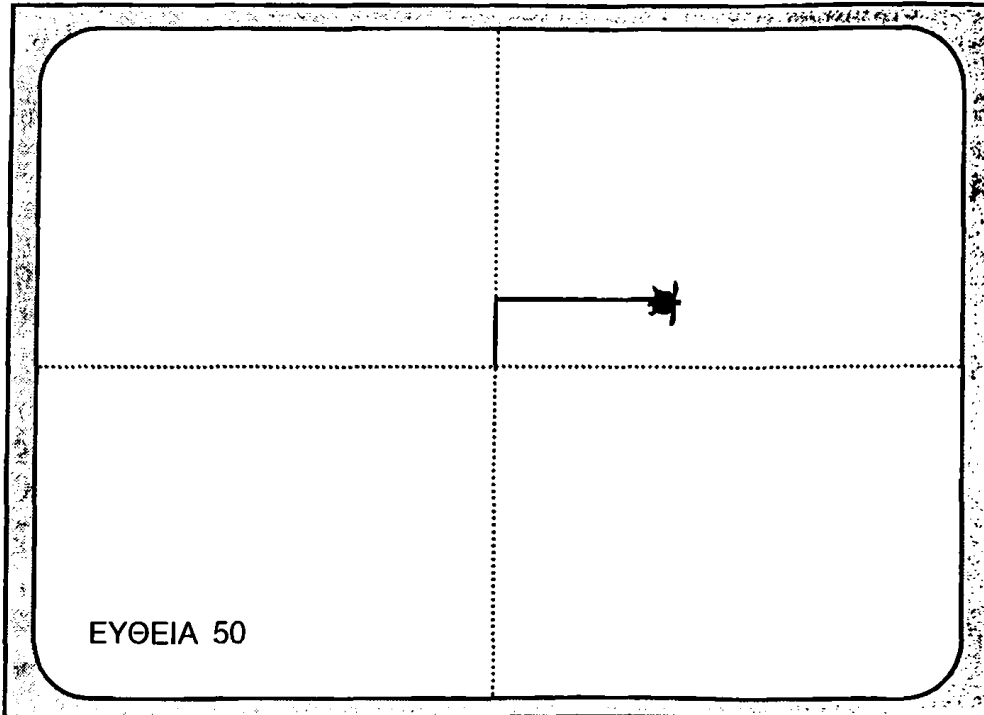


Όταν δώσουμε τη διαταγή ΔΕΞΙΑ, η χελώνα στρίβει προς τα δεξιά. Επειδή έχουμε διατάξει η στροφή να είναι 90 μοιρών, η χελώνα θα στραφεί κατά μια ορθή γωνία. Η χελώνα δε μετακινήθηκε αλλά περιστράφηκε.



Πόσο να στρίψει η χελώνα;

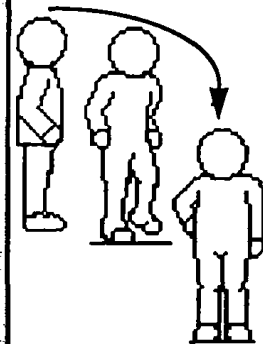
Μετά την προηγούμενη στροφή που έκανε η χελώνα, αν τη διατάξουμε να κινηθεί σε ευθεία γραμμή συνολικά η διαδρομή που θα σχηματισθεί από το ίχνος της θα είναι μια ορθή γωνία.



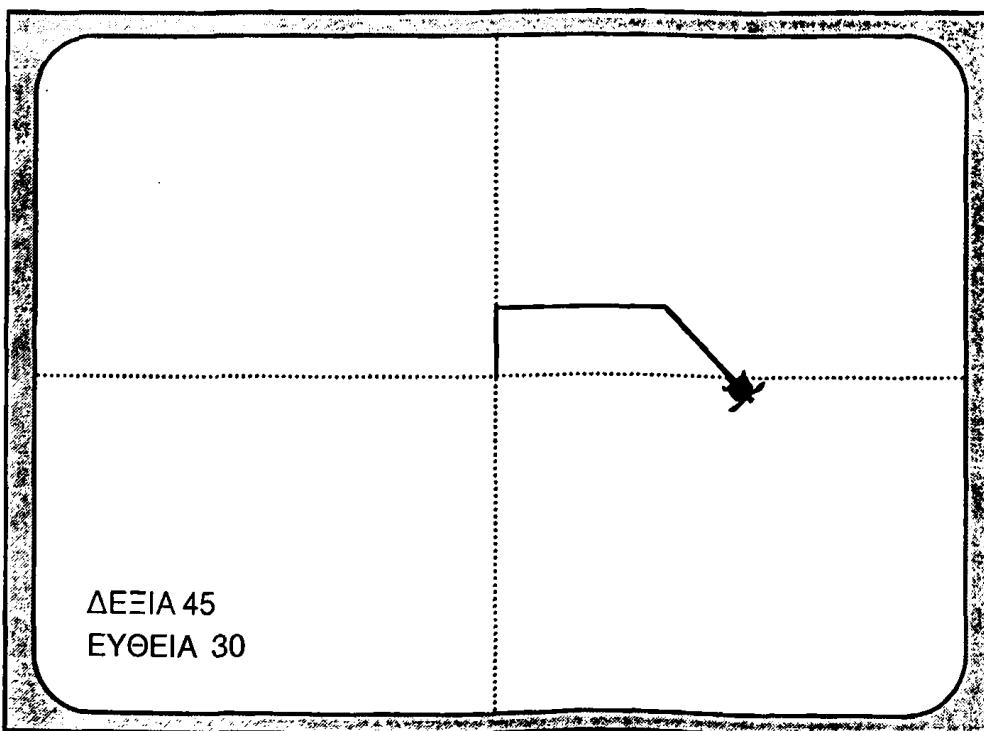
Μια στροφή 90 μοιρών αντιστοιχεί σε στροφή μιας ορθής γωνίας.

MOIRES

90



Οι στροφές που μπορεί να κάνει η χελώνα μπορούν να είναι και μικρότερες των 90 μοιρών. Με τη διαταγή ΔΕΞΙΑ 45 βλέπουμε τη χελώνα να στρίβει κατά μισή ορθή γωνία.



Διαπιστώνουμε ότι εφόσον ορίσαμε κάποια διαδικασία π.χ. την ΕΥΘΕΙΑ ή τη ΔΕΞΙΑ, μπορούμε να την καλούμε οπότε θέλουμε βάζοντας ό,τι τιμή θέλουμε στη μεταβλητή που συνοδεύει το όνομά της, π.χ. ΕΥΘΕΙΑ 20 ή ΕΥΘΕΙΑ 50 ή ΕΥΘΕΙΑ 30.

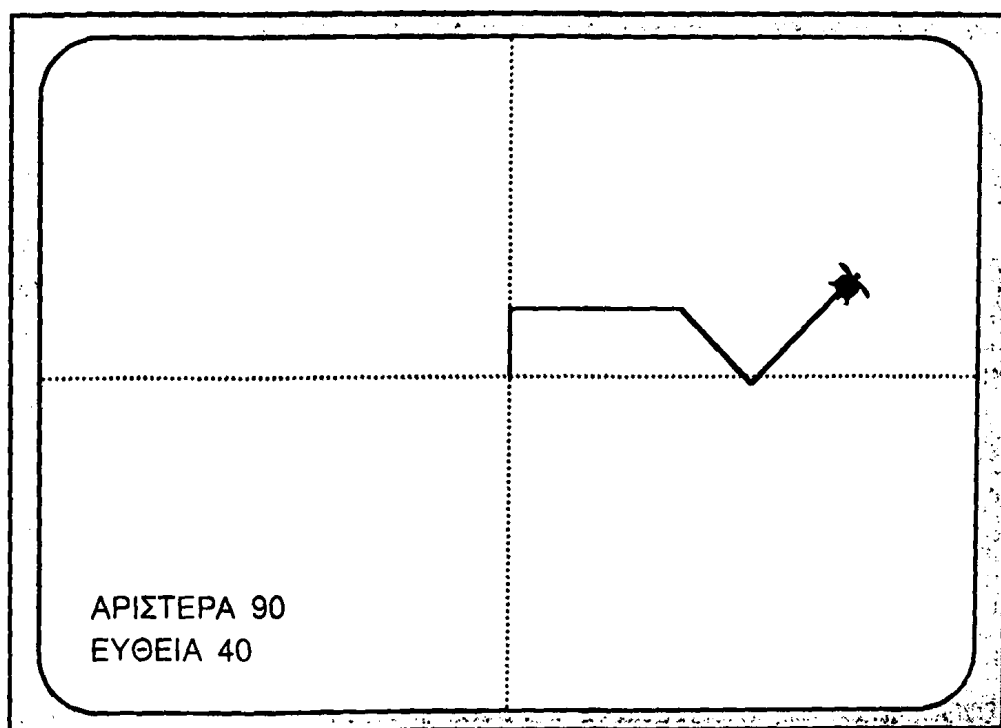


Η χελώνα στρίβει αριστερά

Η στροφή που έμαθε η χελώνα να κάνει είναι προς τα δεξιά. Θα τη μάθουμε τώρα να πραγματοποιεί στροφές και προς τα αριστερά. Αυτό γίνεται με το να ορίσουμε μια νέα διαδικασία, την ΑΡΙΣΤΕΡΑ, που όπως και η ΔΕΞΙΑ χρειάζεται μια μεταβλητή για τον προσδιορισμό του μεγέθους της στροφής.

```
TO ΑΡΙΣΤΕΡΑ :ΜΟΙΡΕΣ
LT :ΜΟΙΡΕΣ
END
```

Η εντολή LT της logo προέρχεται από την ισοδύναμη εντολή Left που στα αγγλικά σημαίνει αριστερά.



Προτείνουμε να πειραματιστείτε καλώντας την εντολή ΑΡΙΣΤΕΡΑ με διάφορες τιμές για τη γωνία όπως 180, 270, 360, 45, 30, 60, 135, 120, 150, 315, 330, 300, 450, 390, 420, 720. Ποιές από αυτές προκαλούν το ίδιο αποτέλεσμα;

Επιβεβαιώστε πειραματιζόμενοι ότι αριστερή στροφή 270 μοιρών και δεξιά στροφή 90 μοιρών έχουν το ίδιο τελικό αποτέλεσμα.

Πως θα στρίψουμε τη χελώνα 120 μοίρες αριστερά αν διαθέτουμε μόνο τη διαταγή ΔΕΞΙΑ;



Η χελώνα στρίβει και προχωρεί

Η διαδικασία ΕΥΘΕΙΑ κινεί τη χελώνα σε ευθεία γραμμή στην κατεύθυνση προς την οποία «κοίταγε» η χελώνα. Αν επιθυμούμε να κινηθεί προς άλλη κατεύθυνση θα πρέπει να στρίψουμε τη χελώνα με τη διαδικασία ΔΕΞΙΑ. Μπορούμε να ενσωματώσουμε αυτές τις δύο διαδικασίες σε μια που θα στρίβει τη χελώνα προς την κατεύθυνση που θέλουμε και στη συνέχεια θα την προχωράει όσα βήματα της πούμε.

Ας ονομάσουμε αυτή τη διαδικασία ΓΡΑΜΜΗ. Από τα προηγούμενα είναι προφανές ότι θα χρειάζεται τιμή και για το μέγεθος της στροφής και για το πόσα βήματα θα κάνει.

ΤΟ ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :ΜΗΚΟΣ

ΔΕΞΙΑ :ΜΟΙΡΕΣ

ΕΥΘΕΙΑ :ΜΗΚΟΣ

END

Στον ορισμό μιας διαδικασίας μπορούν να χρησιμοποιηθούν και άλλες διαδικασίες (όπως οι εντολές της Logo) που έχουν ορισθεί προηγουμένως.

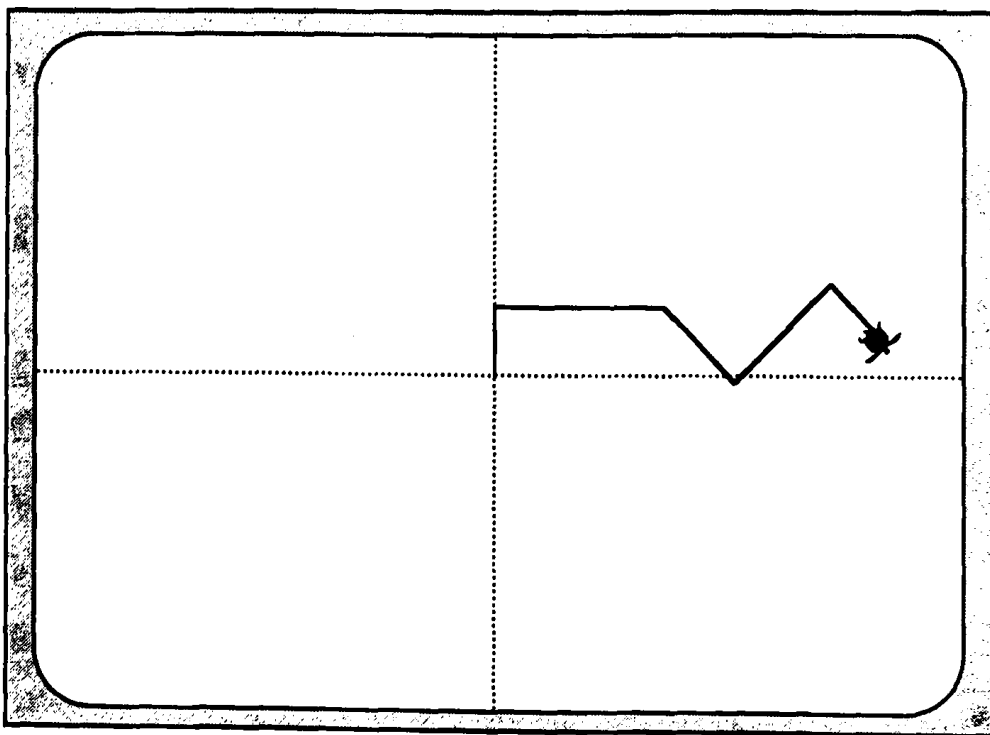
Δεν είναι απαραίτητο (επειδή κατά τον ορισμό π.χ. της διαδικασίας ΕΥΘΕΙΑ χρησιμοποιήθηκε το όνομα ΜΗΚΟΣ για τη μεταβλητή) και στο νέο ορισμό να χρησιμοποιηθεί το ίδιο όνομα. Για παράδειγμα ο προηγούμενος ορισμός θα μπορούσε να είναι:

ΤΟ ΓΡΑΜΜΗ :F :X

ΔΕΞΙΑ :F

ΕΥΘΕΙΑ :X

END



Τα ονόματα των διαδικασιών που χρησιμοποιούνται μέσα στη διαδικασία πρέπει να είναι όπως ακριβώς έχουν ορισθεί προηγουμένως.

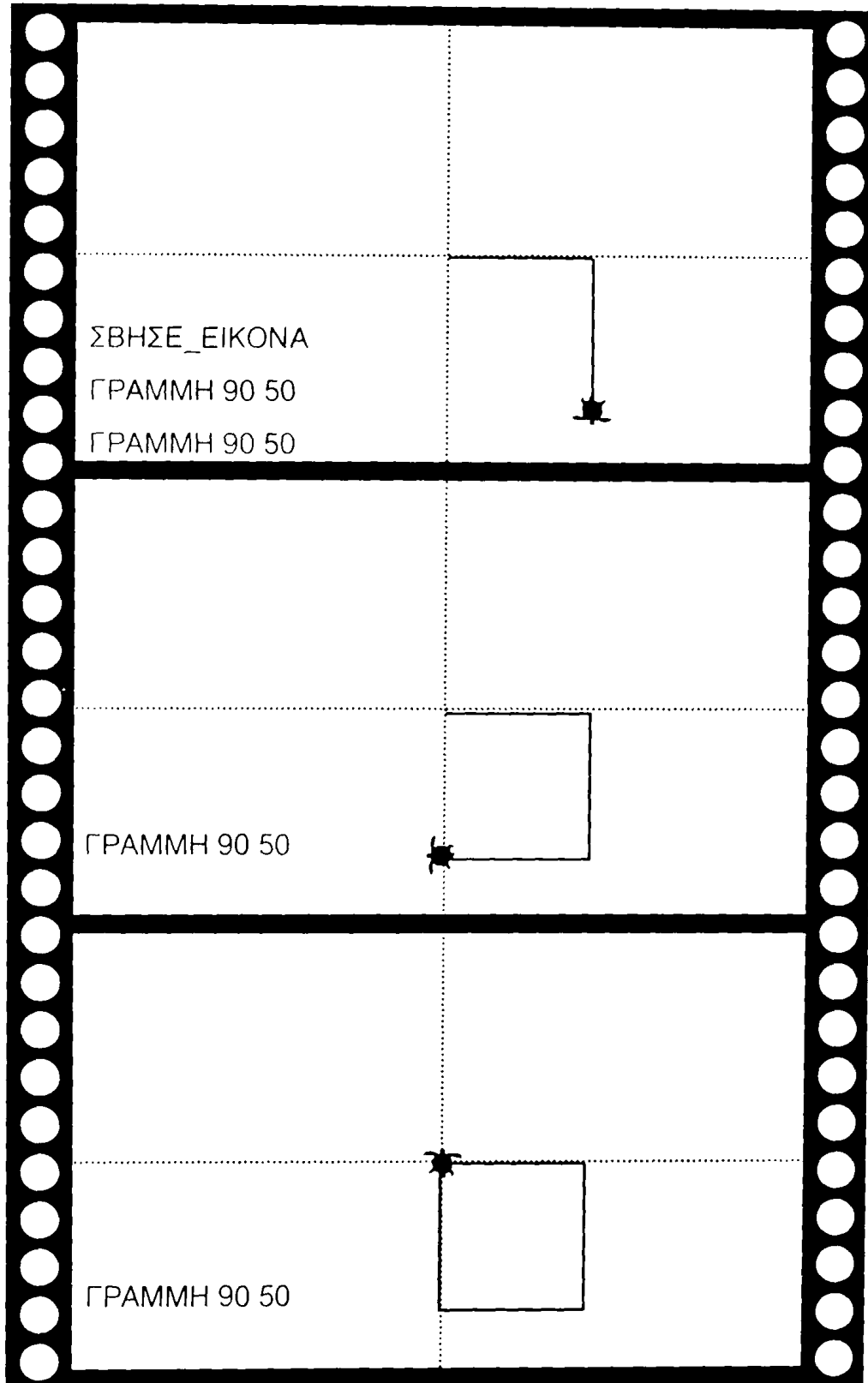
Τα ονόματα των μεταβλητών που χρησιμοποιούνται μέσα στη διαδικασία πρέπει να είναι τα ίδια με αυτά που δηλώνονται στην επικεφαλίδα.

Ποιές από τις παρακάτω διαταγές είναι λάθος και γιατί;
ΓΡΑΜΜΗ
ΓΡΑΜΜΗ 45
ΓΡΑΜΜΗ 00
ΓΡΑΜΜΗ 1 1
ΓΡΑΜΜΗ 0 0
ΓΡΑΜΜΗ 20 9
ΓΡΑΜΜΗ 45 23
ΓΡΑΜΜΗ 90 K 7
ΓΡΑΜΜΗ 90 50 3



Η χελώνα χαράζει ένα τετράγωνο

Μπορούμε δίνοντας κάθε φορά μια εντολή να βάλουμε τη χελώνα να σχηματίσει ένα τετράγωνο.



Αν η χελώνα στρίψει δεξιά 90 μοίρες και προχωρήσει κάποια απόσταση και μετά στρίψει πάλι δεξιά 90 μοίρες και προχωρήσει την ίδια απόσταση η τροχιά της θα σχηματίσει τις πλευρές μιας ορθής γωνίας.

Η χελώνα με τη διαταγή ΓΡΑΜΜΗ 90 50 συνεχίζει και ξαναστρίβει δεξιά 90 μοίρες και προχωράει την ίδια απόσταση.

Τέλος η χελώνα στρίβει δεξιά 90 μοίρες και προχωράει την ίδια απόσταση. Το αποτέλεσμα είναι να βρεθεί στην αρχική της θέση έχοντας κάνει μια πλήρη περιστροφή με διαδρομή σε σχήμα τετραγώνου.



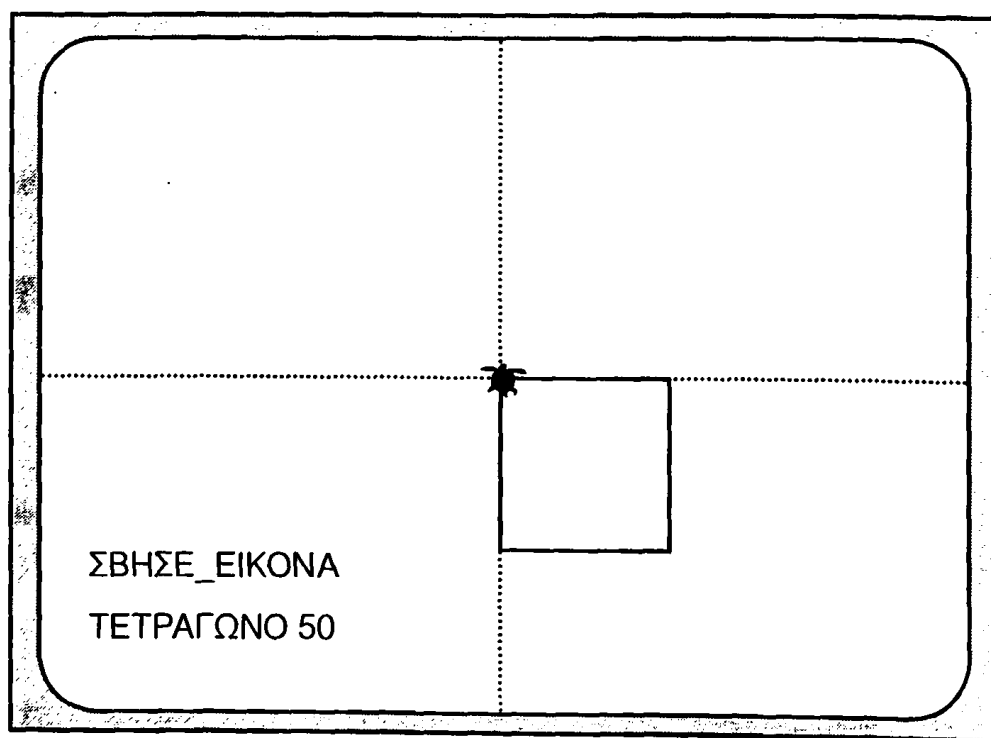
Η χελώνα κινείται σε τετράγωνο

Τώρα θα διδάξουμε τη χελώνα να κινείται μόνη της και να σχηματίζει το τετράγωνο. Ας φτιάξουμε μια νέα διαδικασία, την ΤΕΤΡΑΓΩΝΟ, που θα κάνει τη χελώνα να διαγράψει ένα τετράγωνο.

```
ΤΟ ΤΕΤΡΑΓΩΝΟ :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
END
```

Παρατηρούμε ότι η διαδικασία ΓΡΑΜΜΗ 90 :PLEYRA χρησιμοποιείται τέσσερις φορές στη διαδικασία τετράγωνο. Τι θα συνέβαινε αν η διαδικασία ΤΕΤΡΑΓΩΝΟ είχε ορισθεί ως εξής;

```
ΤΟ ΤΕΤΡΑΓΩΝΟ :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
END
```



Στη διαδικασία ΓΡΑΜΜΗ (που χρησιμοποιείται στον ορισμό της διαδικασίας ΤΕΤΡΑΓΩΝΟ) η τιμή της μεταβλητής ΜΟΙΡΕΣ είναι 90 ενώ η μεταβλητή PLEYRA θα παίρνει τιμή τη στιγμή που θα διατάζουμε τη χελώνα να κινηθεί σε τετράγωνο.

Το άθροισμα των γωνιών των τεσσάρων στροφών που θα κάνει η χελώνα για να σχηματίσει τετράγωνο είναι:
 $90+90+90+90=$
 $=360$ μοίρες.

Η τιμή 50 είναι το μήκος της πλευράς του τετραγώνου. Το 50 πηγαίνει ως τιμή στη μεταβλητή PLEYRA της διαδικασίας ΓΡΑΜΜΗ κάθε φορά που τη χρησιμοποιεί η διαδικασία ΤΕΤΡΑΓΩΝΟ.



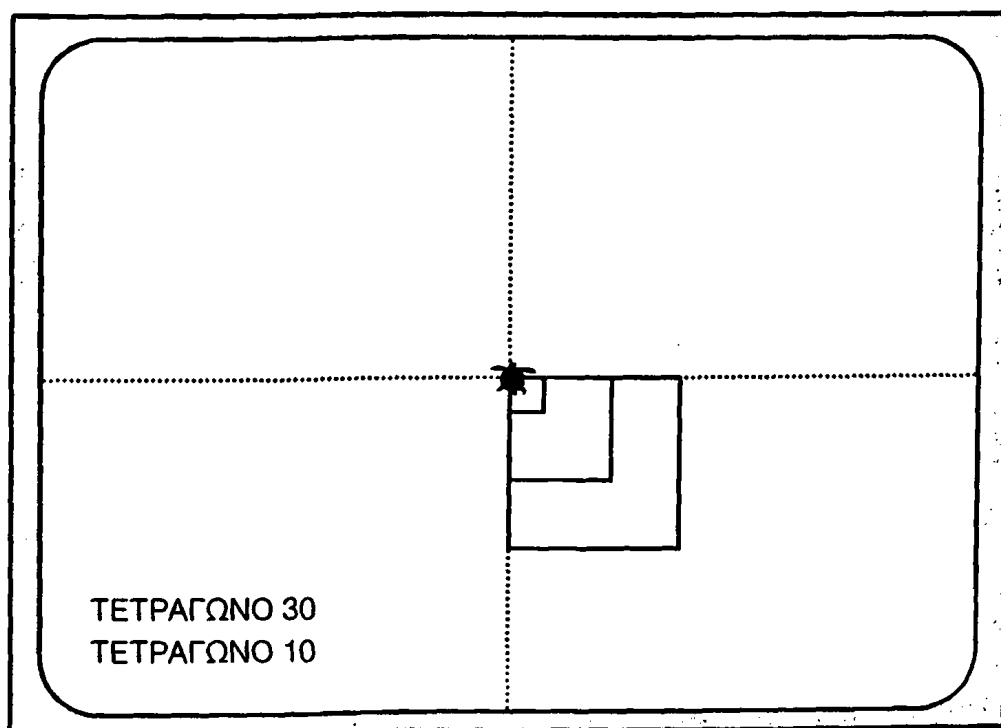
Η διαδικασία δίνει τιμή σε μια μεταβλητή

Προηγουμένως η στροφή που έκανε η χελώνα ξεκινώντας τη διαδικασία ΓΡΑΜΜΗ ήταν 90 μοίρες. Γενικά μπορούμε να βάλουμε μέσα στη μεταβλητή MOIRES την τιμή 90 με την εντολή MAKE και να χρησιμοποιούμε τη διαδικασία ΓΡΑΜΜΗ όπως φαίνεται στον παρακάτω νέο ορισμό της διαδικασίας ΤΕΤΡΑΓΩΝΟ.

```
ΤΟ ΤΕΤΡΑΓΩΝΟ :PLEYRA
MAKE "MOIRES 90
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
END
```

Η εντολή MAKE της logo, στα αγγλικά σημαίνει κάνε και εννοεί να δώσει την τιμή (το 90) ως περιεχόμενο σε κάποια μεταβλητή (τη MOIRES).

Όταν αναφερόμαστε στην τιμή (δηλαδή το περιεχόμενο) της μεταβλητής MOIRES, χρησιμοποιούμε το :MOIRES, ενώ όταν αναφερόμαστε στη μεταβλητή χρησιμοποιούμε το "MOIRES.

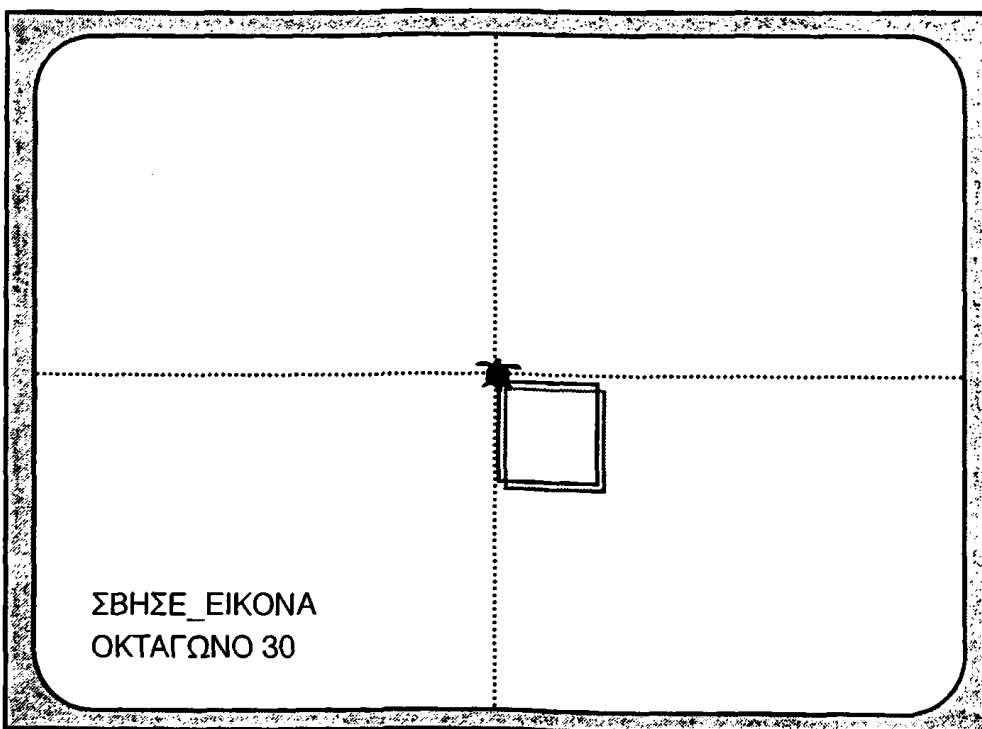


Προσπάθεια για περισσότερες πλευρές

Στη διαδικασία ΤΕΤΡΑΓΩΝΟ χρειάστηκε να χρησιμοποιήσουμε τέσσερις φορές τη διαδικασία ΓΡΑΜΜΗ. Για να φτιάξουμε ένα κανονικό σχήμα με οκτώ πλευρές, το οκτάγωνο, θα πρέπει να χρησιμοποιήσουμε τη διαδικασία ΓΡΑΜΜΗ οκτώ φορές. Ας γράψουμε λοιπόν τη διαδικασία ΟΚΤΑΓΩΝΟ.

```
ΤΟ ΟΚΤΑΓΩΝΟ :PLEYRA
MAKE "MOIRES 90
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
END
```

Θα διαπιστώσουμε ότι δε σχηματίστηκε ένα σχήμα με οκτώ πλευρές αλλά ότι η χελώνα διέτρεξε δύο φορές τις τέσσερις πλευρές ενός τετραγώνου. Τι δεν πήγε καλά;



Το πρόγραμμα που φαίνεται δίπλα είναι λάθος. Λάθει κάνουμε ώλυ. Αξία αίχι να δηδασκώμασται από αφτά. Μελετώντας τα λάθη μας μπορούμε να εντοπίσουμε τα αδύνατα σημεία του τρόπου που αντιμετωπίσαμε το πρόβλημα και να αναζητήσουμε τρόπους όχι μόνο να διορθώσουμε το συγκεκριμένο λάθος αλλά και να γενικεύσουμε την εμπειρία μας ώστε να μην το επαναλάβουμε.

Το άθροισμα των γωνιών των οκτώ στροφών που κάνει η χελώνα είναι:
 $90+90+90+90+$
 $+90+90+90+90 =$
 $= 720$ μοίρες
ενώ μια πλήρης περιστροφή είναι 360 μοίρες.

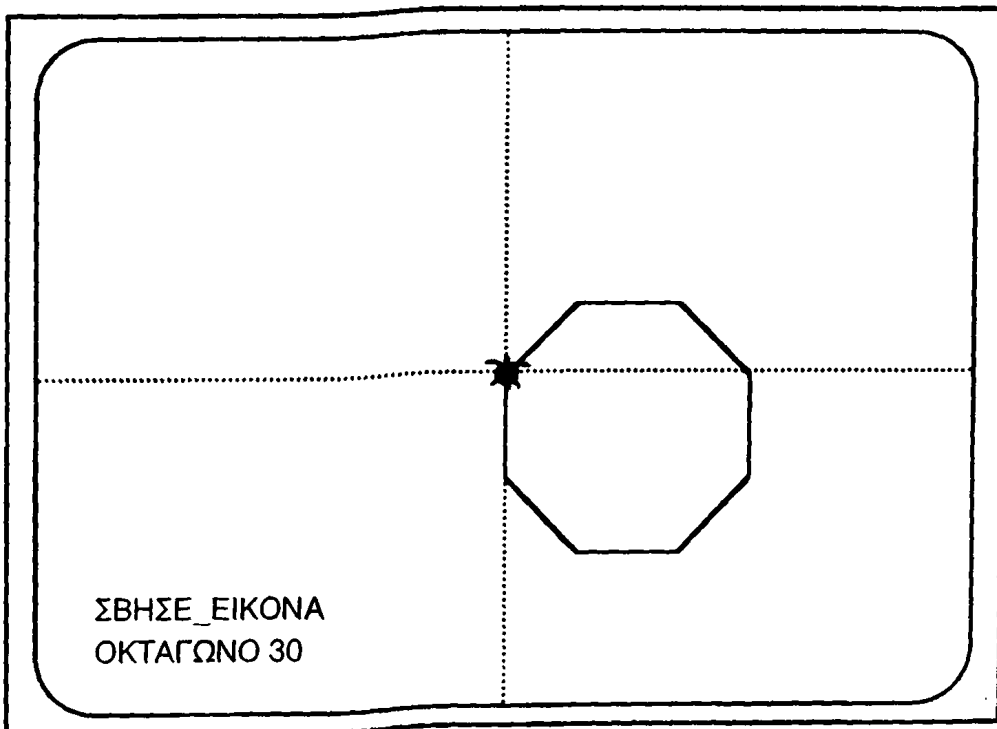


Περισσότερες πλευρές, μικρότερη γωνία

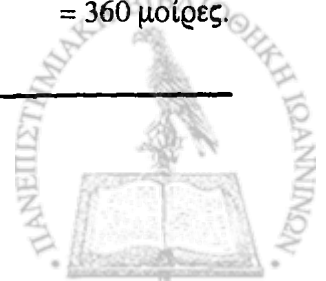
Όπως είδαμε, η χελώνα έκανε οκτώ στροφές των 90 μοιρών πραγματοποιώντας δύο περιστροφές δηλαδή γωνία 720 μοιρών ενώ θέλουμε να κάνει μόνο μια περιστροφή (δηλαδή 360 μοίρες) στρίβοντας 8 φορές. Αυτό θα το πετύχουμε αν κάθε φορά αντί να στρίβει 90 μοίρες στρίβει το μισό, 45 μοίρες. Μ' αυτήν την παρατήρηση ας διορθώσουμε το πρόγραμμα της διαδικασίας ΟΚΤΑΓΩΝΟ καταχωρώντας ως τιμή στη μεταβλητή MOIRES το 45.

```
TO ΟΚΤΑΓΩΝΟ :PLEYRA
MAKE "MOIRES 45
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
END
```

«...Το να μάθεις να είσαι έμπειρος προγραμματιστής σημαίνει να μάθεις πως να γίνεις ικανός στην απομόνωση και διόρθωση "σφαλμάτων", σημείων που αποτρέπουν την εκτέλεση του προγράμματος. Η ερώτηση σχετικά με ένα πρόγραμμα δεν είναι το αν είναι σωστό ή λανθασμένο, αλλά αν διορθώνεται.»
Seymour Papert



Το άθροισμα των γωνιών των οκτώ στροφών που κάνει η χελώνα για να σχηματίσει οκτάγωνο είναι:
 $45+45+45+45+$
 $+45+45+45+45 =$
 $= 360 \text{ μοίρες.}$



Γωνία και πλήθος πλευρών

Το επόμενο βήμα είναι να μάθουμε τη χελώνα να σχεδιάζει ένα κανονικό εξάγωνο. Φυσικά θα χρησιμοποιήσουμε τη διαδικασία ΓΡΑΜΜΗ έξι φορές. Δε μένει παρά να υπολογίσουμε τη γωνία που θα στρίβει η χελώνα κάθε φορά.

Είδαμε ότι για να κάνει η χελώνα μια πλήρη περιστροφή (είτε διαγράφοντας τετράγωνο ή οκτάγωνο) το άθροισμα των γωνιών είναι 360 μοίρες.

Δηλαδή

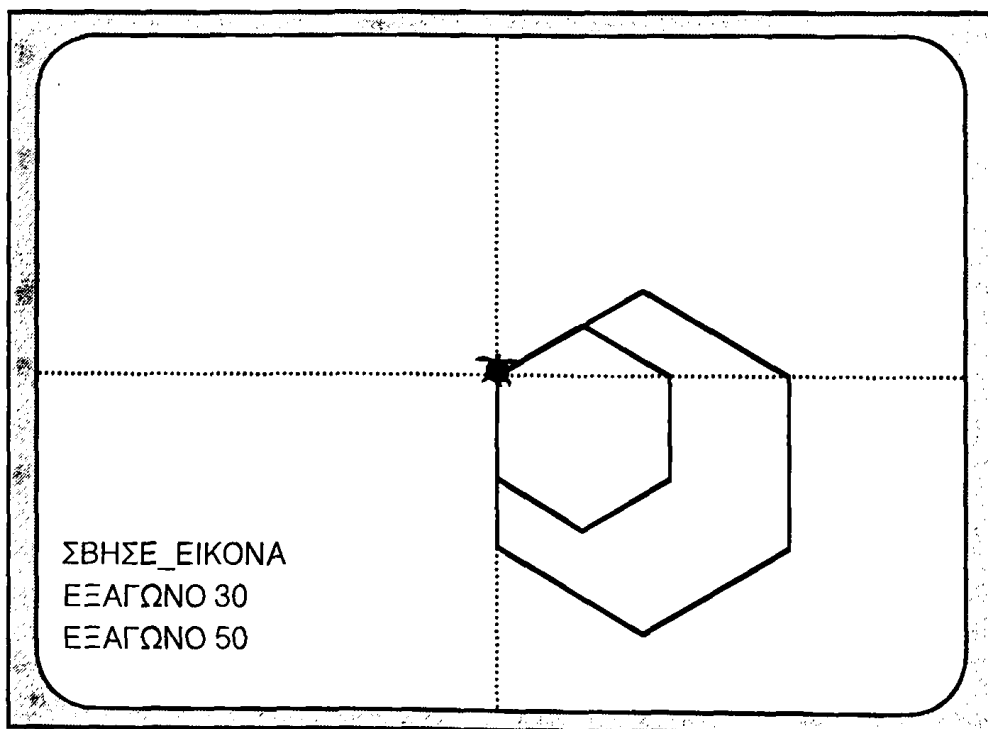
$$90+90+90+90 = 4 \times 90 = 360 \text{ ή } 360/4 = 90 \text{ μοίρες}$$

$$45+45+45+45+45+45+45+45 = 8 \times 45 = 360 \text{ ή } 360/8 = 45 \text{ μοίρες}$$

Άρα και η γωνία για το εξάγωνο θα είναι $360/6 = 60$ μοίρες.

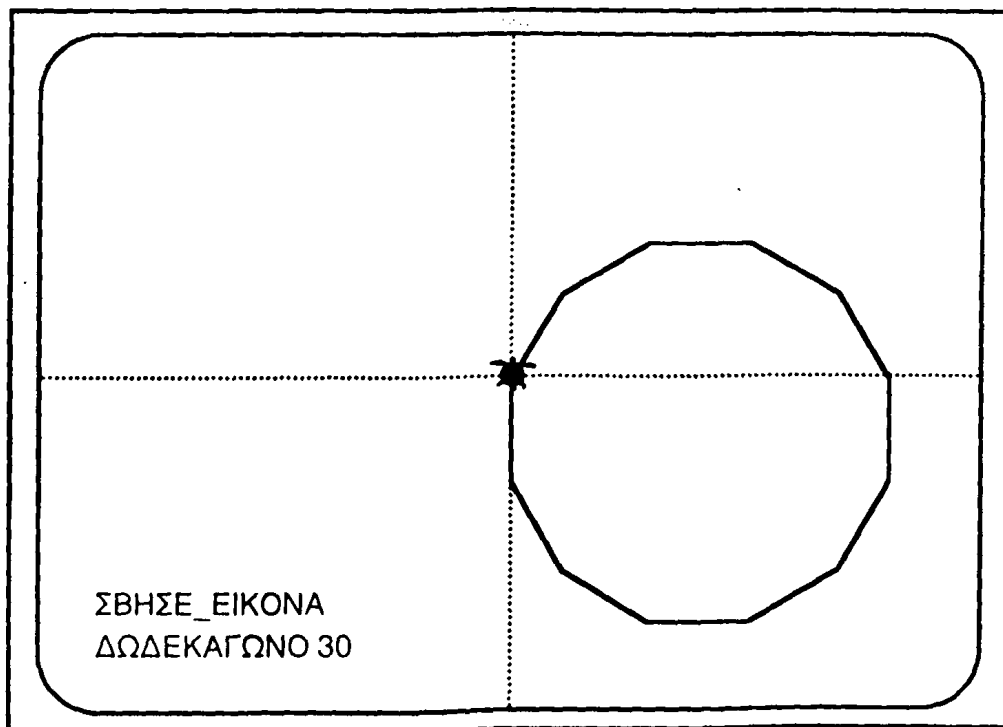
```
TO ΕΞΑΓΩΝΟ :PLEYRA
MAKE "MOIRES ( 360 / 6 )
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
END
```

Στην εντολή
MAKE
υπολογίζεται το
αποτέλεσμα της
παράστασης
360/6 και
καταχωρείται
ως τιμή στη
μεταβλητή
MOIRES.



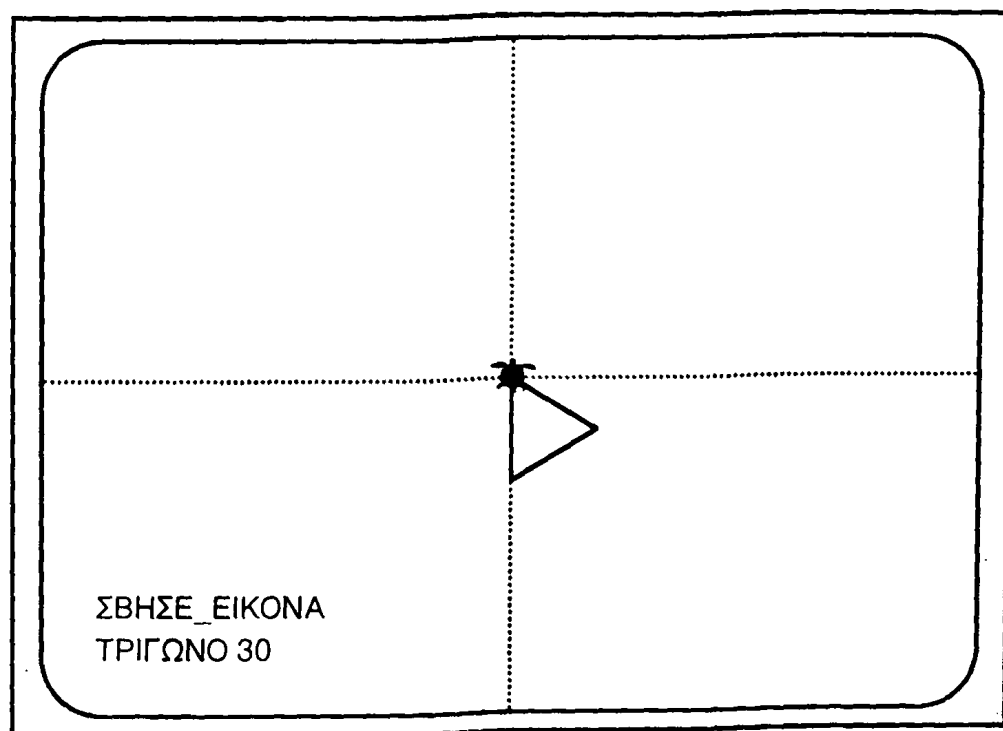
Δωδεκάγωνο και τρίγωνο

Γράψτε το πρόγραμμα της διαδικασίας που θα μαθαίνει τη χελώνα να σχεδιάζει ένα κανονικό δωδεκάγωνο. Τι τιμή θα δώσετε στην μεταβλητή MOIRES και γιατί;



«Για να λύσεις
ένα πρόβλημα
ψάξε να βρεις
κάτι που να
μοιάζει με αυτό
το οποίο ήδη
καταλαβαίνεις»
Seymour Papert

Γράψτε το πρόγραμμα της διαδικασίας που θα μαθαίνει τη χελώνα να σχεδιάζει ένα "κανονικό" τρίγωνο.



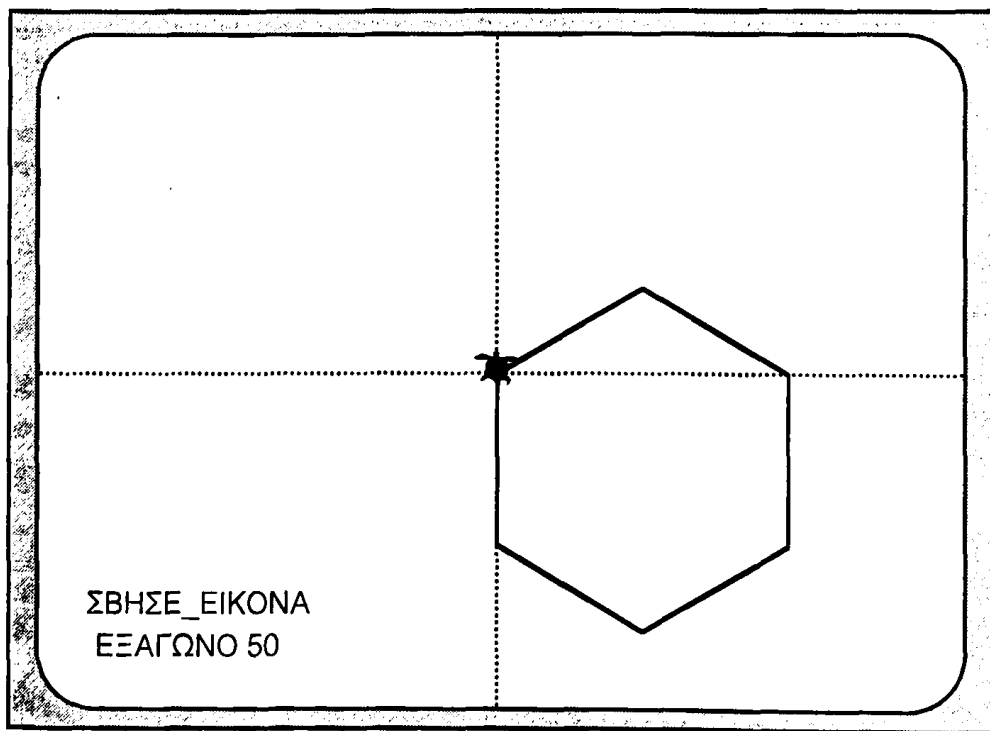
Ένας τρόπος επανάληψης

Στο πρόγραμμα για το δωδεκάγωνο, η ίδια διαδικασία (ΓΡΑΜΜΗ) επαναλαμβάνεται 12 φορές. Το να γράφεις 12 φορές το ίδιο πράγμα μάλλον θυμίζει τιμωρία παρά δημιουργική εργασία. Όλες οι γλώσσες προγραμματισμού διαθέτουν μια τουλάχιστον εντολή με τη βοήθεια της οποίας μπορούμε εύκολα να επαναλάβουμε κάποιο σύνολο εντολών.

Η logo διαθέτει την εντολή REPEAT. Ας δούμε πως μπορεί να γραφεί το πρόγραμμα της διαδικασίας ΕΞΑΓΩΝΟ χρησιμοποιώντας την.

```
TO ΕΞΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES ( 360 / 6 )  
REPEAT 6 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

Η εντολή REPEAT της logo στα αγγλικά, σημαίνει επανέλαβε. Μετά τη λέξη REPEAT ακολουθεί ένας ακέραιος αριθμός που καθορίζει πόσες φορές θα γίνει η επανάληψη. Στη συνέχεια μέσα σε αγκύλες υπάρχει η εντολή ή οι εντολές που θα επαναληφθούν.



Ίδιες διαδικασίες με τη βοήθεια του repeat

Ας δούμε πως μπορούν να ξαναορισθούν οι διαδικασίες ΤΡΙΓΩΝΟ, ΤΕΤΡΑΓΩΝΟ, ΟΚΤΑΓΩΝΟ και ΔΩΔΕΚΑΓΩΝΟ με τη βοήθεια της εντολής REPEAT.

```
TO ΤΡΙΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 3  
REPEAT 3 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

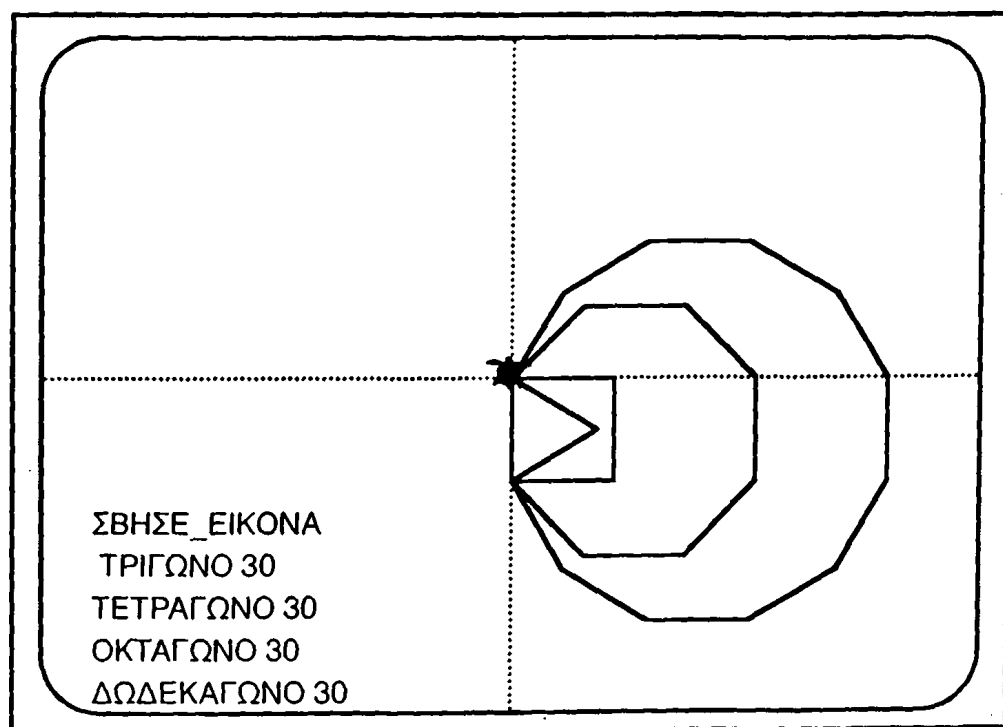
```
TO ΤΕΤΡΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 4  
REPEAT 4 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

```
TO ΟΚΤΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 8  
REPEAT 8 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

```
TO ΔΩΔΕΚΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 12  
REPEAT 12 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

Τα τέσσερα αυτά προγράμματα διαφέρουν μόνο ως προς τον αριθμό που εκφράζει το πλήθος των πλευρών (και των γωνιών) του κανονικού σχήματος. Δηλαδή τα προγράμματα όλων των κανονικών σχημάτων έχουν την ίδια δομή.

«Ο καλύτερος τρόπος κατανόησης της μάθησης είναι να καταλάβουμε πρώτα τις ειδικές επιλεγμένες περιπτώσεις και μετά να νοιαστούμε για το πως θα γενικευτεί αυτή η κατανόηση.»
Seymour Papert



Γιατί όλα τα σχήματα έχουν μια κοινή πλευρά;



Τρι-τετρα-εξα-οκτα...γωνο ή πολύγωνο

Από τα προηγούμενα προγράμματα ας κρατήσουμε ό,τι είναι κοινό σε όλα:

```
ΤΟ όνομα σχήματος :PLEYRA
MAKE "MOIRES 360 / πλήθος γωνιών
REPEAT πλήθος πλευρών [ΓΡΑΜΜΗ :MOIRES :PLEYRA ]
END
```

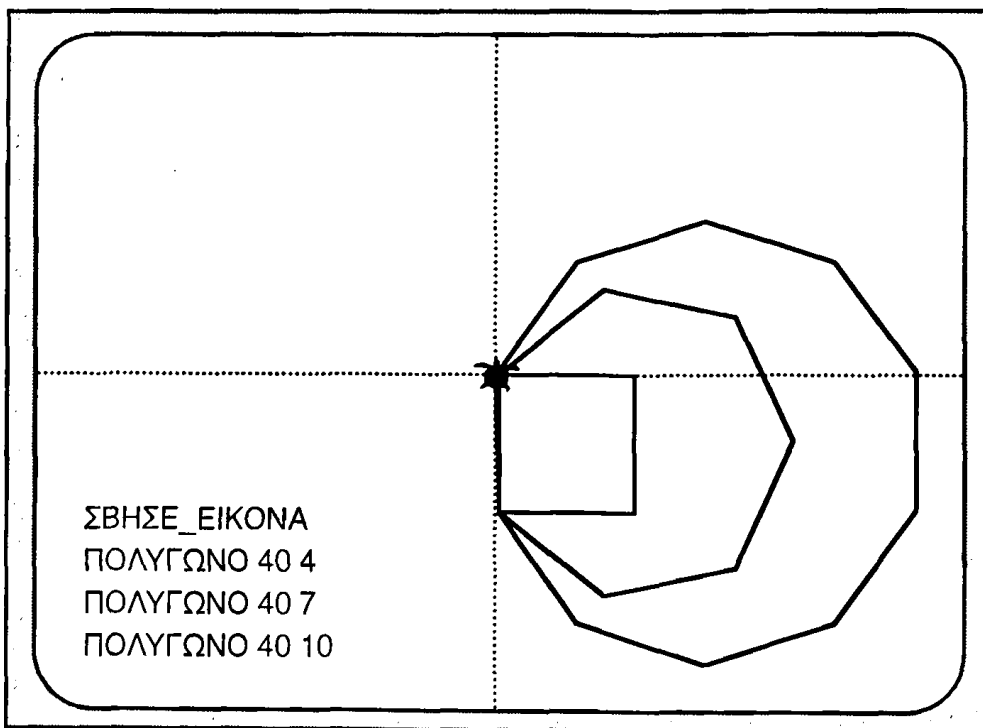
Το πλήθος των γωνιών είναι ίδιο με το πλήθος των πλευρών.

Με τη διαδικασία ΤΕΤΡΑΓΩΝΟ η χελώνα μαθαίνει να σχεδιάζει ένα τετράγωνο με μέγεθος ανάλογο προς το μήκος της πλευράς που θα της πούμε όταν τη διατάξουμε.

Ας προχωρήσουμε ένα βήμα πιο κάτω. Ας διδάξουμε τη χελώνα να σχεδιάζει οποιοδήποτε κανονικό πολύγωνο, δηλώνοντας και το πλήθος των πλευρών ή γωνιών του. Χρειάζεται λοιπόν κατά τον ορισμό της διαδικασίας αυτής να δίνεται στη χελώνα εκτός από το μήκος της πλευράς και το πλήθος των γωνιών.

```
ΤΟ ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
MAKE "MOIRES ( 360 / :PLHQOS )
REPEAT :PLHQOS [ΓΡΑΜΜΗ :MOIRES :PLEYRA ]
END
```

Η παράσταση (360/:PLHQOS) βρίσκεται μέσα σε παρενθέσεις για να είναι περισσότερο κατανοητό στον προγραμματιστή ότι στη μεταβλητή MOIRES εκχωρείται το αποτέλεσμα της διαίρεσης και όχι μόνο το 360.



Η δομή του προγράμματος αναπαριστάει στη μνήμη του υπολογιστή τη δομή του φυσικού κόσμου.

«Διδάσκοντας τον υπολογιστή πως να σκέφτεται, τα παιδιά ξεκινούν για μια εξερεύνηση του δικού τους τρόπου σκέψης. Η εμπειρία μπορεί να είναι μεθυστική. Η σκέψη για τη σκέψη κάνει το παιδί επιστημολόγο, μια εμπειρία άγνωστη ακόμα και στους περισσότερους ενήλικες.»
Seymour Papert

Παρατηρούμε ότι με την ίδια διαδικασία σχηματίζονται διαφορετικά σχήματα. Τι κοινό έχουν αυτά τα σχήματα;

Τι σχήμα θα δημιουργηθεί αν ζητήσουμε το πολύγωνο να έχει 2 πλευρές; Που θα διαφέρει αυτό από το πολύγωνο με 1 πλευρά;



Προσανατολισμένο πολύγωνο

Όλα τα προηγούμενα σχήματα δημιουργούνται δεξιά και κάτω από το σημείο που ξεκινά η χελώνα. Αυτό οφείλεται στο ότι η πρώτη εντολή που κινεί τη χελώνα είναι η ΓΡΑΜΜΗ και περιέχει την εντολή ΔΕΞΙΑ.

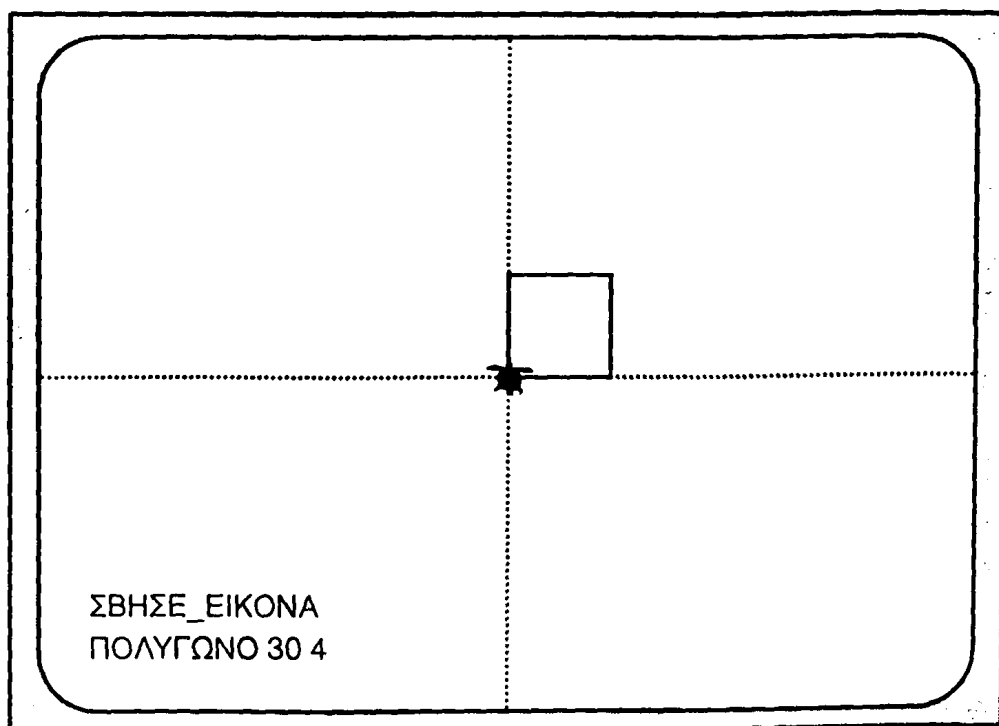
Αν θέλουμε η αρχή των σχημάτων να είναι πάνω από το σημείο αφητηρίας της χελώνας θα πρέπει να "εξουδετερώσουμε" αυτήν τη στροφή δεξιά με μια στροφή 90 μοιρών προς τα αριστερά.

Στη συνέχεια αφού σχηματισθεί το πολύγωνο, θα πρέπει να αποκαταστήσουμε τον αρχικό προσανατολισμό της χελώνας στρέφοντας την 90 μοίρες προς τα δεξιά.

Με βάση αυτές τις παρατηρήσεις ας βελτιώσουμε τη διαδικασία ΠΟΛΥΓΩΝΟ ώστε να δημιουργεί σχήματα που η αρχή τους να στρέφεται προς τα πάνω.

```
ΤΟ ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
ΑΡΙΣΤΕΡΑ 90
ΜΑΚΕ "ΜΟΙΡΕΣ ( 360 / :PLHQOS )
ΡΕΡΕΑΤ :PLHQOS [ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA]
ΔΕΞΙΑ 90
ΕΝΔ
```

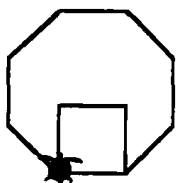
Πως αλλιώς
μπορεί να
προσανατολιστεί
το πολύγωνο;



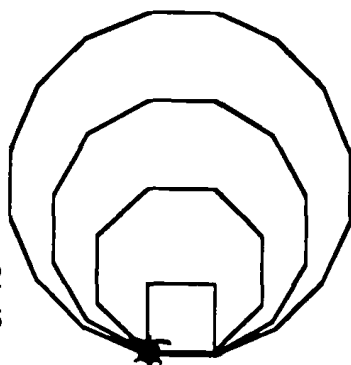
Το πλήθος των πλευρών τείνει στο άπειρο

Στα παρακάτω σχήματα, αυξάνεται διαδοχικά το πλήθος των πλευρών της διαδικασίας ΠΟΛΥΓΩΝΟ.

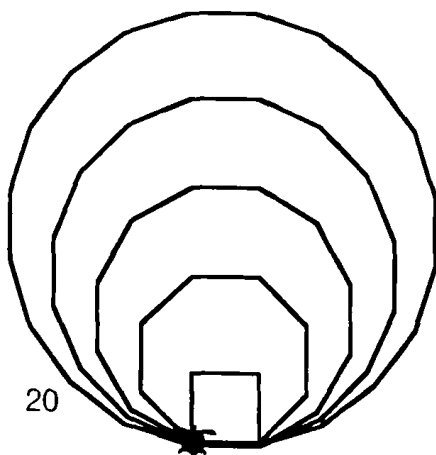
ΠΟΛΥΓΩΝΟ 20 4
ΠΟΛΥΓΩΝΟ 20 8



ΠΟΛΥΓΩΝΟ 20 12
ΠΟΛΥΓΩΝΟ 20 16



ΠΟΛΥΓΩΝΟ 20 20



Η χελώνα σχεδιάζει ένα τετράγωνο και ένα οκτάγωνο με μήκος πλευράς 20 βημάτων. Για να το επιβεβαιώσετε μετρήστε πόσες γωνίες έχει το κάθε σχήμα.

Στη συνέχεια σχηματίζουμε ένα δωδεκάγωνο και ένα δεκαεξάγωνο. Προσπαθείστε πάλι να επιβεβαιώσετε μετρώντας τις γωνίες τους, ότι έχουν το σωστό αριθμό γωνιών.

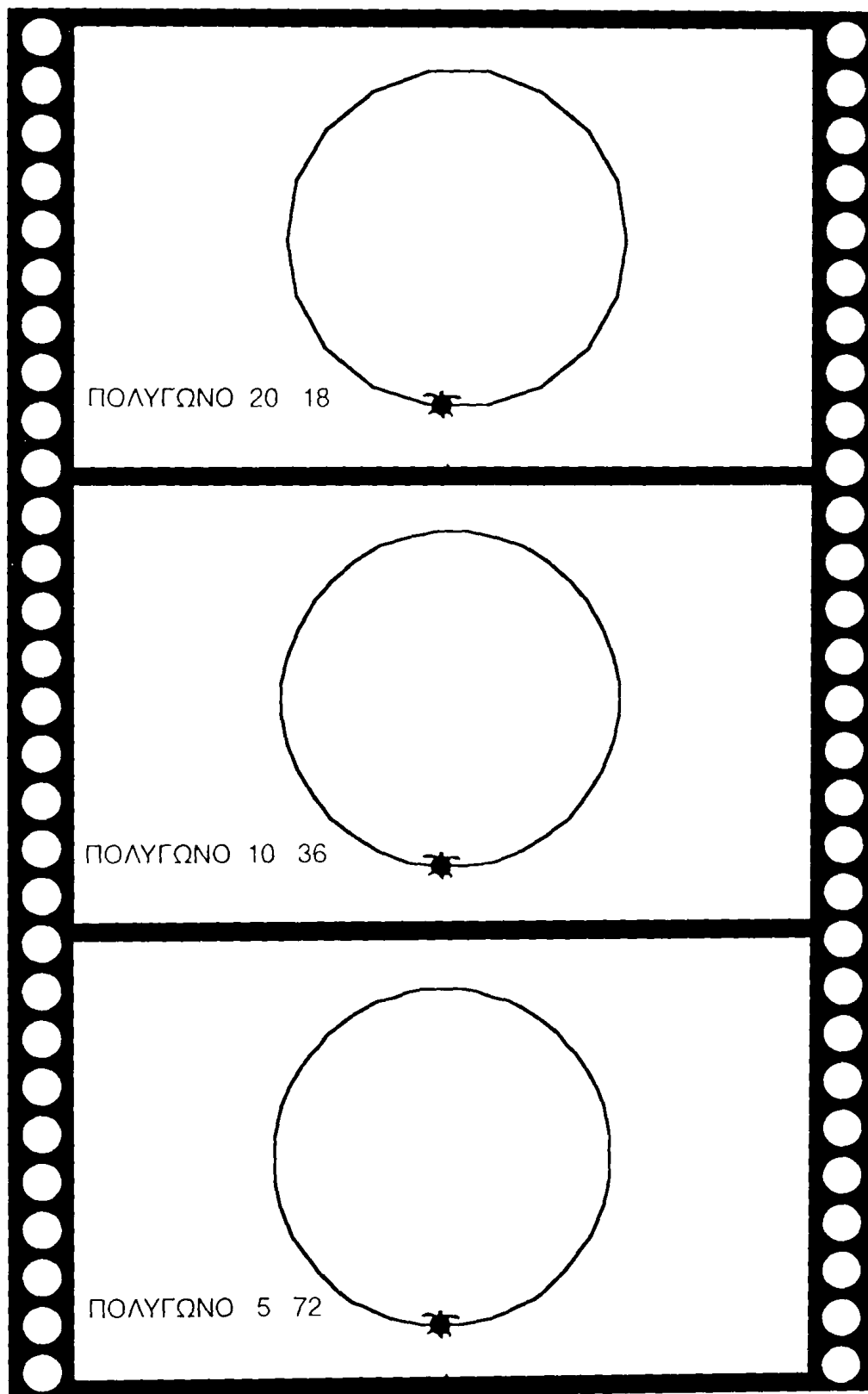
Τέλος, ζητάμε να σχηματιστεί ένα εικοσάγωνο. Είναι εύκολο να μετρήσετε τώρα το πλήθος των γωνιών του σχήματος;

Με τι μοιάζει το σχήμα; Φτιάξτε πολύγωνο με εικοσιτέσσερις γωνίες.



Προσεγγίζοντας μια περιφέρεια κύκλου

Ας φτιάξουμε τρία πολύγωνα με 18, 36 και 72 πλευρές αντίστοιχα και ας παρατηρήσουμε τις διαφορές τους.



Φαίνεται ότι πλησιάζουμε τα όρια της διακριτικής ικανότητας του μέσου απεικόνισης αλλά και της ίδιας της αναπαράστασης. Σιγά-σιγά μειώνεται η ικανότητα διάκρισης μεταξύ εικόνων δύο αντικειμένων που έχουν πολύ μικρές διαφορές.

Μεταξύ του 18-γωνου και του 36-γωνου υπάρχουν κάποιες μικρές διαφορές.

Μεταξύ του 36-γωνου και του 72-γωνου δεν υπάρχουν σημαντικές διαφορές στα σχήματα όπως εμφανίζονται στην οθόνη.

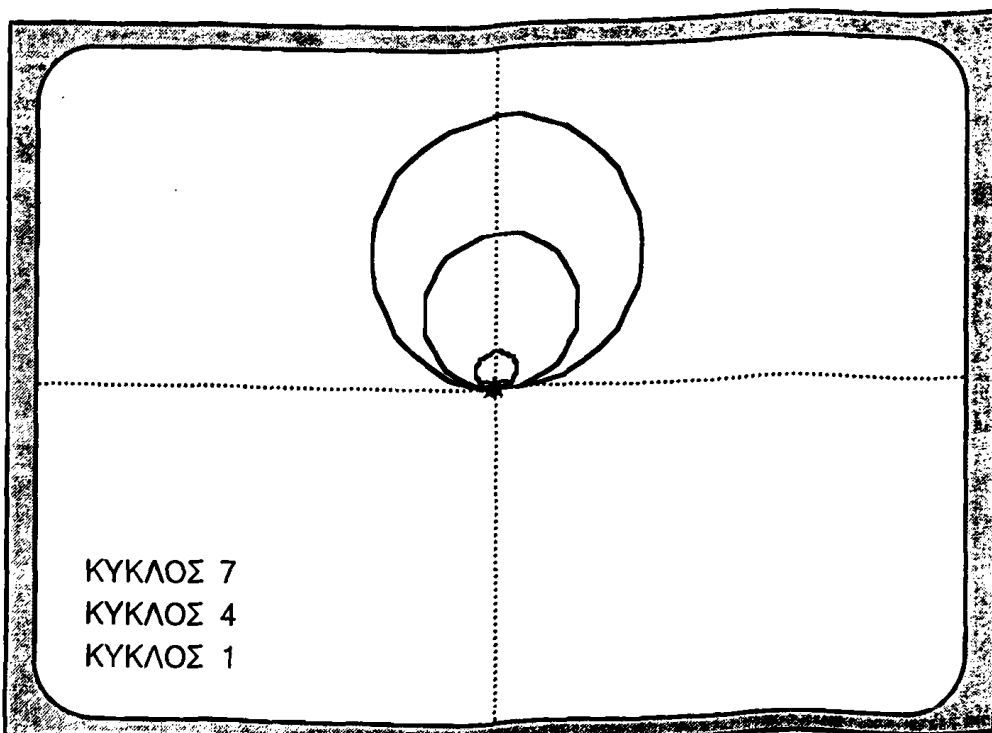
Σχήμα που μοιάζει με περιφέρεια κύκλου

Από τα σχήματα της προηγούμενης σελίδας καταλήγουμε στο συμπέρασμα ότι το 36-γωνο είναι μια ικανοποιητική προσέγγιση του κύκλου.

Επειδή ο κύκλος είναι ένα σχήμα που χρειάζεται συχνά, το ορίζουμε ξεχωριστά.

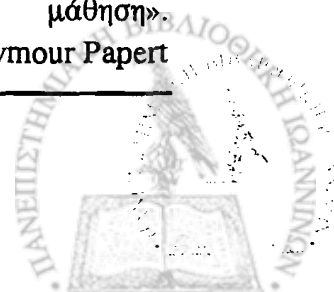
Ένας κομψός τρόπος είναι να χρησιμοποιήσουμε την υπάρχουσα διαδικασία του πολυγώνου, οπότε η διαδικασία ΚΥΚΛΟΣ ορίζεται ως εξής:

ΤΟ ΚΥΚΛΟΣ :PLEYRA
ΠΟΛΥΓΩΝΟ :PLEYRA 36
END



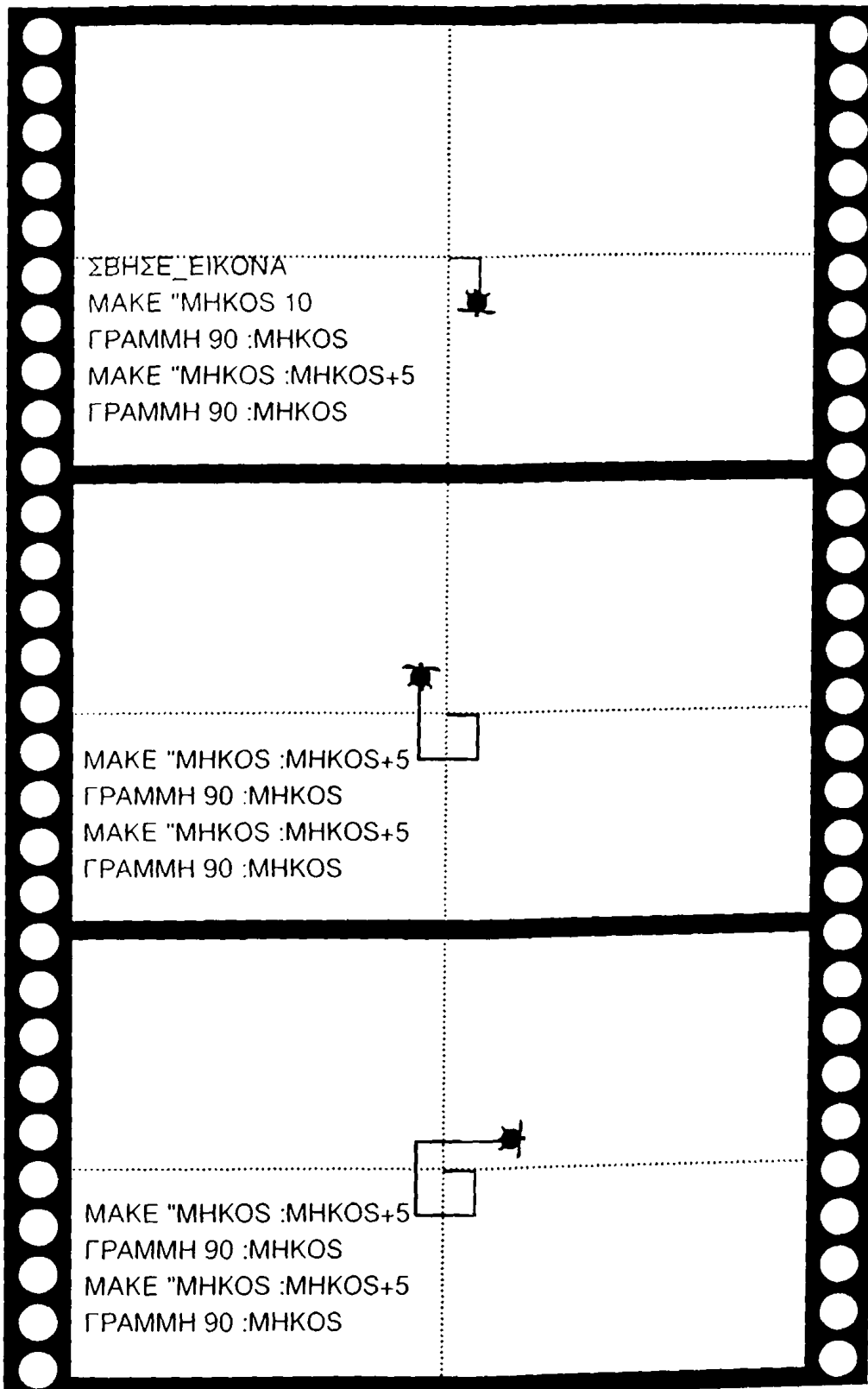
«Τα παιδιά... αν έφτιαξαν μόλις έναν κύκλο δίνοντας εντολές στη χελώνα να κάνει μικρά μικροσβήματα και μικρές δεξιές στροφές, είναι έτοιμα να επιχειρηματολογήσουν με το δάσκαλο ότι ο κύκλος είναι στην πραγματικότητα ένα πολύγωνο. Κάποιος που ακούει μια τέτοια συνομιλία στην πέμπτη τάξη της logo, δεν απομακρύνεται χωρίς να εντυπωσιαστεί από την ιδέα ότι η αλήθεια ή ψευδότητα της θεωρίας είναι δευτερεύουσας σημασίας ως προς τη συμβολή της στη μάθηση».

Seymour Papert



Επαναλαμβάνοντας τον εαυτό μας

Αυξάνοντας διαδοχικά την τιμή της μεταβλητής ΜΗΚΟΣ στη διαδικασία ΓΡΑΜΜΗ, το αποτέλεσμα μοιάζει με σπείρα.



Ο κόσμος του αύριο θα είναι ο κόσμος της τεχνητής νοημοσύνης. Μια τεχνική που χρησιμοποιείται ευρέως στον τομέα αυτό είναι η αναδρομή. Εμείς οι μεγάλοι καταλάβαμε την αναδρομή σε προχωρημένη ηλικία. Η αναδρομή δεν μας είναι οικεία όπως η επανάληψη. Πιστεύουμε ότι αν μπολιάσουμε τον τρόπο σκέψης της γενιάς που θα ζήσει με τις νοήμονες μηχανές με την έννοια της αναδρομής, αυτό θα την εξοικειώσει με την τεχνητή νοημοσύνη.

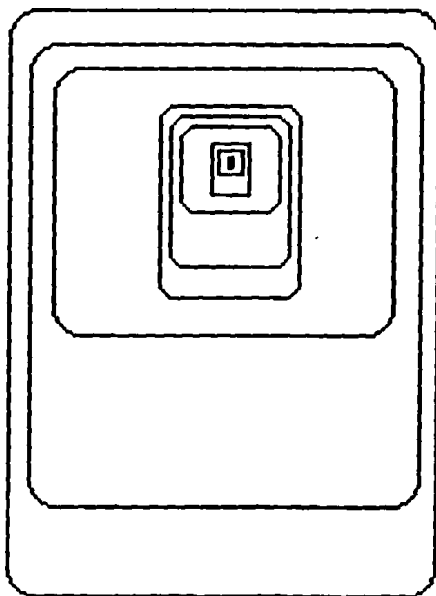


Μια αναδρομική διαδικασία

Αναδρομή είναι η τεχνική κατά την οποία μέσα στον ορισμό μιας διαδικασίας χρησιμοποιείται η ίδια διαδικασία. Η αναδρομή γεννιέται από τη στιγμή που ένα σχήμα κατασκευάζεται από αναπαγωγή του ίδιου του εαυτού του σε άλλη κλίμακα ή διαφορετικό επίπεδο, και αναδρομικότητα είναι η έκφραση μέσα από την αναδρομή. Αναδρομικές διαδικασίες συναντώνται στην καθημερινή εμπειρία με παραδείγματα την ανθρώπινη γλώσσα, τις ακρογιαλιές, τις νιφάδες του χιονιού, το φύλλωμα ενός δένδρου, και όλες τις μορφές φράκταλ από τους παλμούς της ανθρώπινης καρδιάς μέχρι την κατανομή των γαλαξιών και τις χαστικές διαδικασίες.

Είναι επόμενο λοιπόν, η λύση προβλημάτων των μαθηματικών και της πληροφορικής να βασίζονται στον αναδρομικό τρόπο σκέψης, με την αναδρομή να αποτελεί βασική αρχή του προγραμματισμού για εύκολη, γρήγορη και κομψή αντιμετώπιση προβλημάτων.

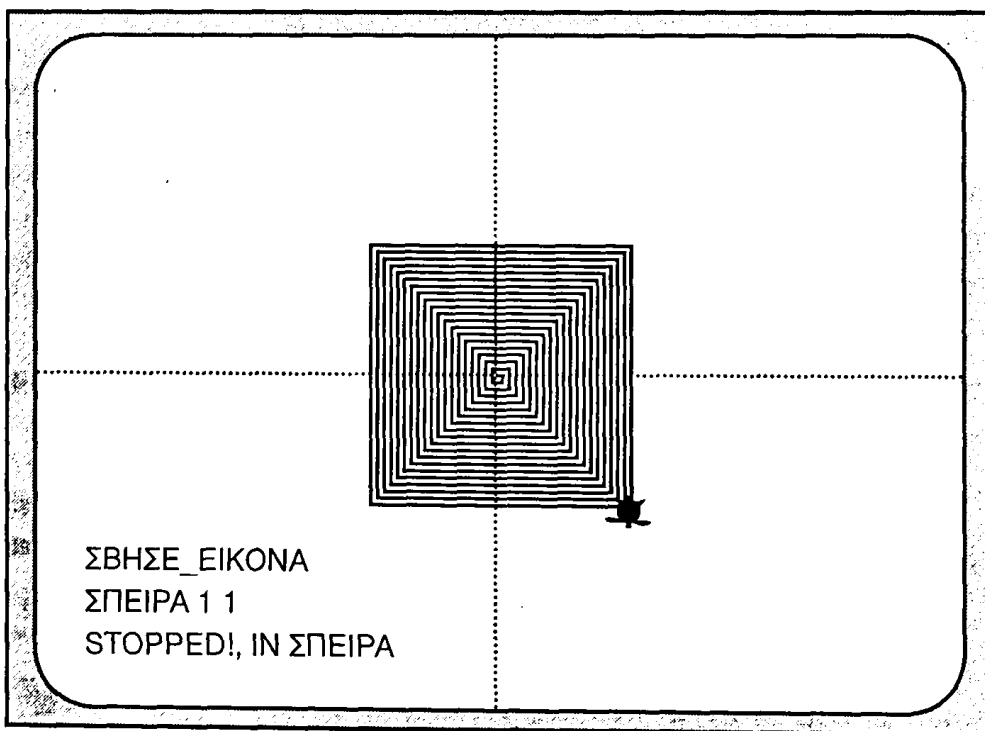
Ως εργαλείο για την κατανόηση και εξοικείωση με την έννοια της αναδρομής στην εκπαίδευση, προτείνεται το ανοικτό περιβάλλον της Logo. Σ' αυτό προγραμματιστικά προβλήματα αντιμετωπίζονται με απλούστερο τρόπο, και επιπλέον εμφανίζεται στο μαθητή ο ανθρώπινος τρόπος σκέψης και η ομορφιά της φύσης.



Μορφή μέσα στη μορφή

Η προηγούμενη διαδικασία μπορεί να γίνει πιο γενική αν αντί του 5 θέσουμε μια νέα μεταβλητή, τη ΒΗΜΑ, που θα αντιστοιχεί στο ποσό που θα αλλάζει κάθε φορά το μήκος της γραμμής.

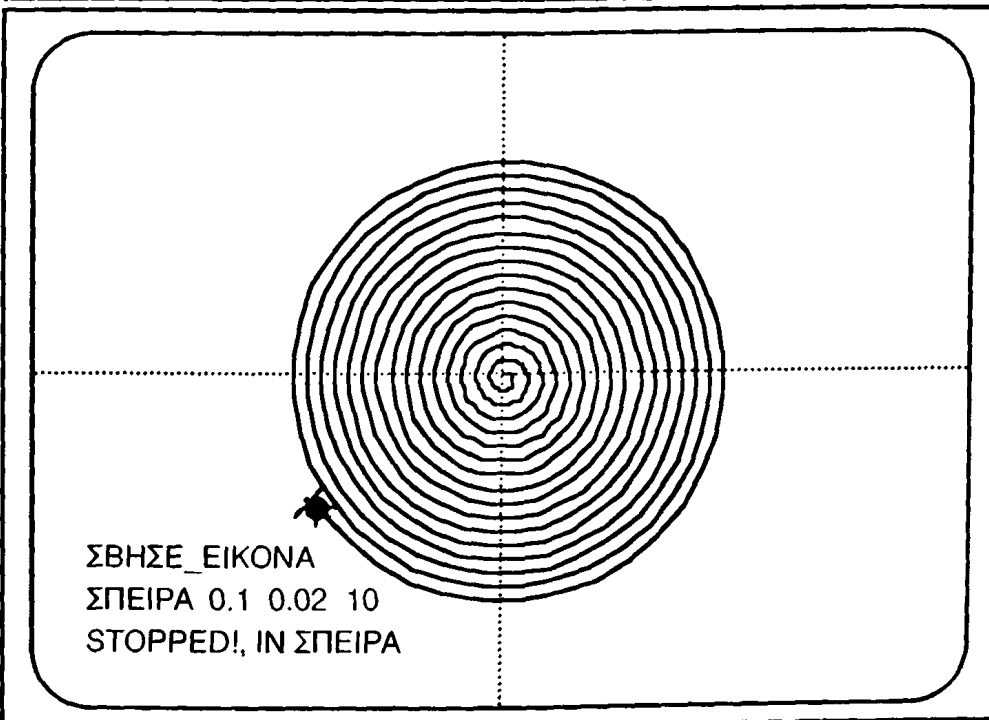
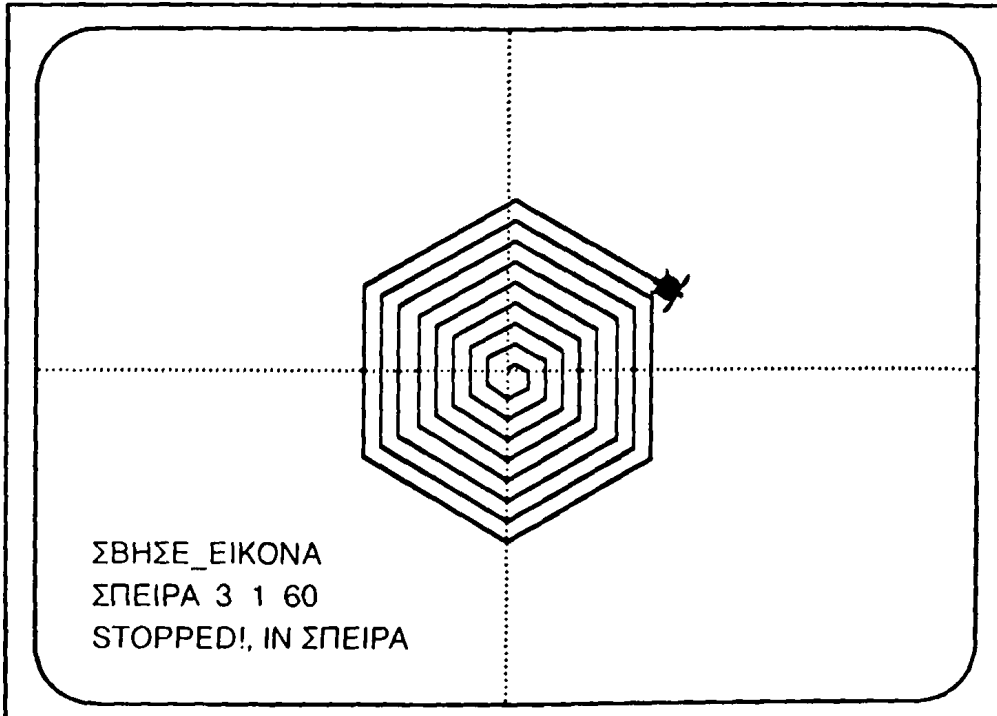
```
ΤΟ ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ  
ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ  
ΜΑΚΕ "ΜΗΚΟΣ :ΜΗΚΟΣ + :ΒΗΜΑ  
ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ  
END
```



Δεσμώτες του ιλίγγου της αναδρομής

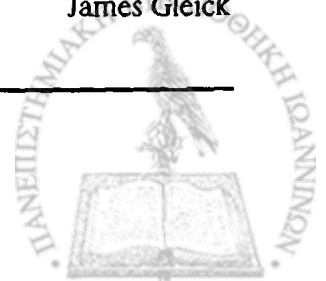
Μπορούμε να προσθέσουμε μια ακόμη μεταβλητή, τη MOIRES, που θα καθορίζει τη σταθερή γωνία που θα στρίβει η χελώνα.

```
ΤΟ ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ  
ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :ΜΗΚΟΣ  
ΜΑΚΕ "ΜΗΚΟΣ :ΜΗΚΟΣ + :ΒΗΜΑ  
ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ  
ΕΝΔ
```



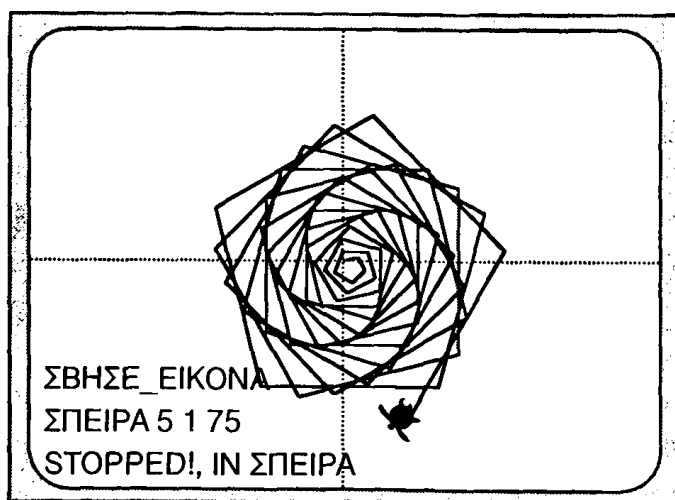
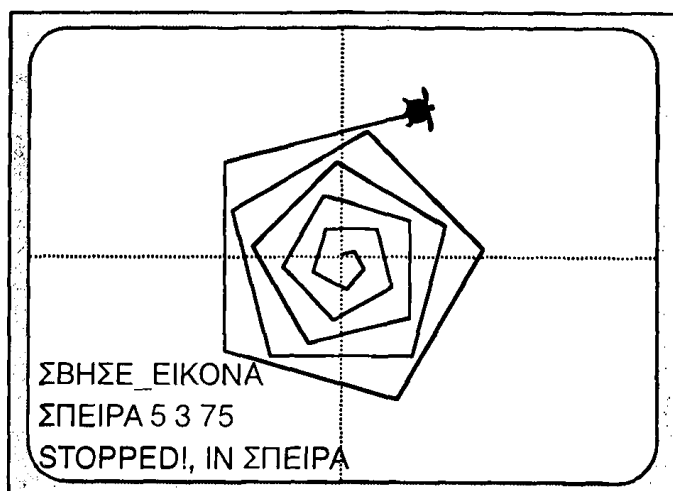
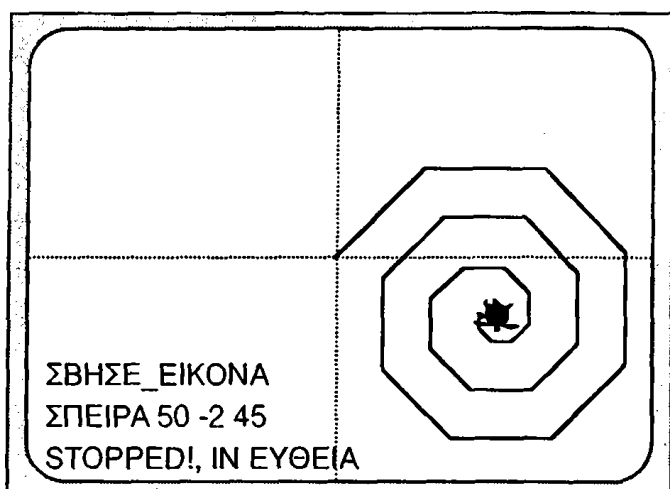
«Το μυαλό δε μπορεί να συλλάβει στο σύνολό της την άπειρη επανάληψη της πολυπλοκότητας μέσα στον εαυτό της. Αλλά για κάποιον που σκέφτεται τη μορφή με τον τρόπο του γεωμέτρη, αυτό το είδος επανάληψης της δομής σε όλο και πιο μικρές κλίμακες μπορεί να αποκαλύψει έναν ολόκληρο κόσμο.»

James Gleick



Ασήμαντες αιτίες με μεγάλα αποτελέσματα

Παίζοντας με διάφορες τιμές στις παραμέτρους εισόδου της διαδικασίας, θα παρατηρήσουμε ποικιλία σχημάτων.



Αρνητική τιμή
στο βήμα κάνει
τη σπείρα να
στρέφεται προς
το εσωτερικό της.

«Η σύγχρονη
μελέτη του χάους
άρχισε με τη
βαθμιαία
συνειδητοποίηση,
στη δεκαετία του
1960, ότι πολύ
απλές
μαθηματικές
εξισώσεις
μπορούσαν να
απεικονίζουν
συστήματα βίαια
σαν ένα
καταρράκτη.
Ελάχιστες
διαφορές στην
είσοδο
μπορούσαν
γρήγορα να
μετατραπούν σε
τεράστιες
διαφορές στην
έξοδο...»
James Gleick



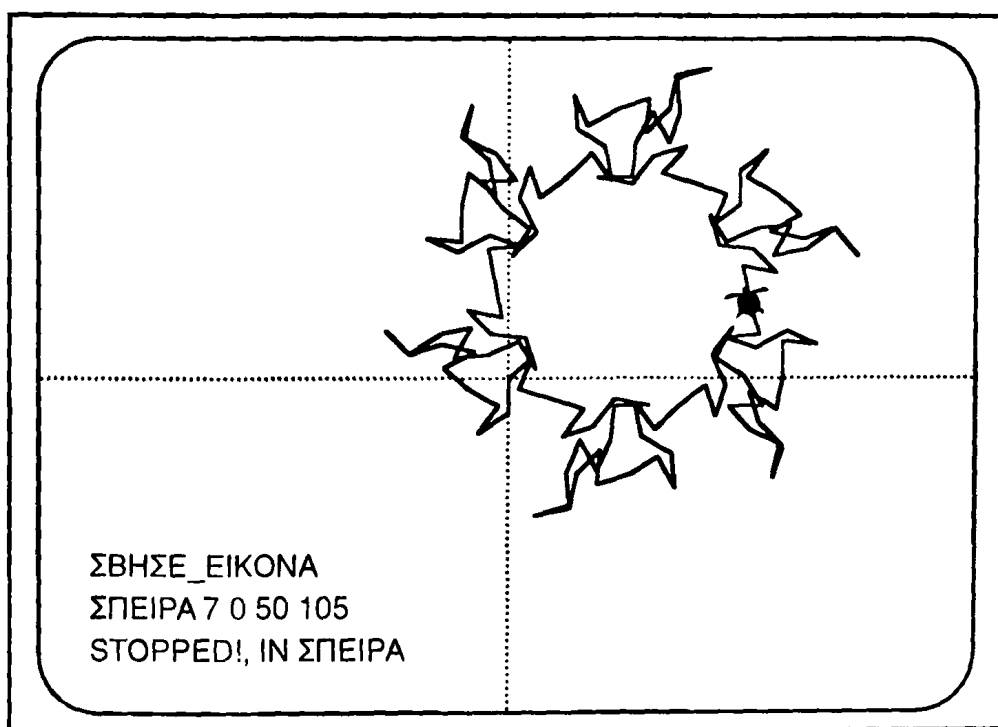
Ο χορός των μαγισσών

Μπορούμε να τροποποιήσουμε την προηγούμενη αναδρομική διαδικασία δίνοντάς της τη δυνατότητα να μεταβάλλεται κάθε φορά η γωνία στροφής. Αυτό το πετυχαίνουμε προσθέτοντας μια ακόμη μεταβλητή, την D_GWNIA που θα καθορίζει το ποσό της μεταβολής της στροφής που θα κάνει η χελώνα. Έτσι, η διαδικασία ΣΠΕΙΡΑ διαμορφώνεται ως εξής:

```
TO ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ :D_GWNIA
ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :ΜΗΚΟΣ
MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + :ΒΗΜΑ
MAKE "ΜΟΙΡΕΣ :ΜΟΙΡΕΣ + :D_GWNIA
ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ :D_GWNIA
END
```

Μια προσωπική εμπειρία.

Τα προγράμματα που φτιάχνουμε κατασκευάζονται αφού προσχεδιάσουμε τι ακριβώς θέλουμε να κάνουμε και που να καταλήξουμε. Συχνά όμως δε μπορούμε να αντισταθούμε στον πειρασμό και να παίξουμε μ' αυτές τις διαδικασίες αλλάζοντας τις τιμές των μεταβλητών τους. Το αποτέλεσμα συνήθως μας αφήνει έκπληκτους. Ο υπολογιστής και αυτή η τόσο απλή γλώσσα -η Logo- μας δίνουν την ευκαιρία να ξεπεράσουμε τη διαίσθηση και τη φαντασία μας. Βλέπουμε στην οθόνη πράγματα που μπορεί να τα ξέρουμε γιατί τα είχαμε προσεγγίσει μαθηματικά, αλλά ποτέ δεν τα είχαμε δει.



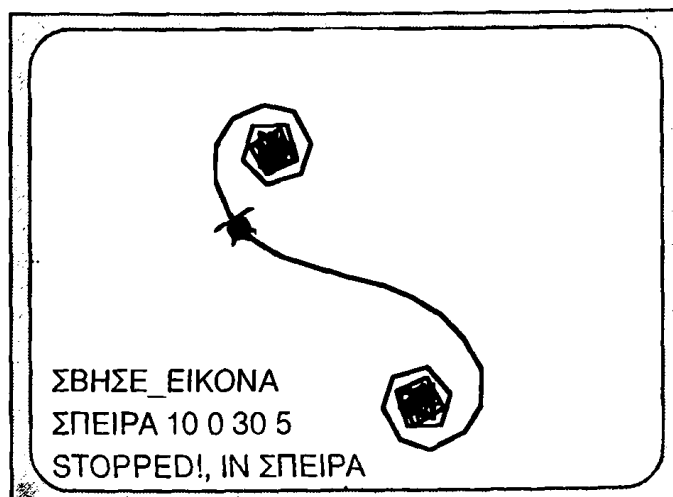
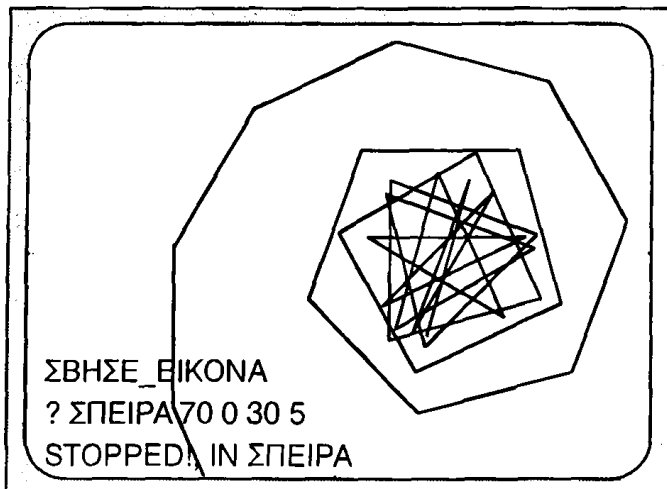
«Θέλετε να πείτε» είπε το παιδί, «ότι πραγματικά δεν ξέρετε πως να το φτιάξετε;» Το παιδί δεν ήξερε ακόμα πως να το πει αλλά αυτό που είχε ανακαλύψει ήταν πως μαζί με τη δασκάλα είχαν ασχοληθεί με ερευνητικό έργο... Στο περιβάλλον της logo, τα παιδιά μαθαίνουν ότι και ο δάσκαλος είναι μαθητευόμενος και ότι όλοι μαθαίνουν από τα λάθη τους» Seymour Papert

«Ακόμα και η τάξη εμφανίζεται αυθόρμητα σε αυτά τα συστήματα - χάος και τάξη μαζί.» James Gleick



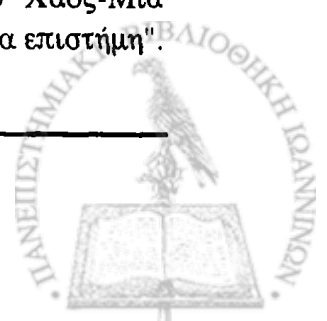
Το δένδρο και το δάσος

Μεταβάλλοντας μόνο μια παράμετρο όταν καλούμε τη διαδικασία μπορούμε να δούμε το δένδρο χωρίς να χάσουμε το δάσος.



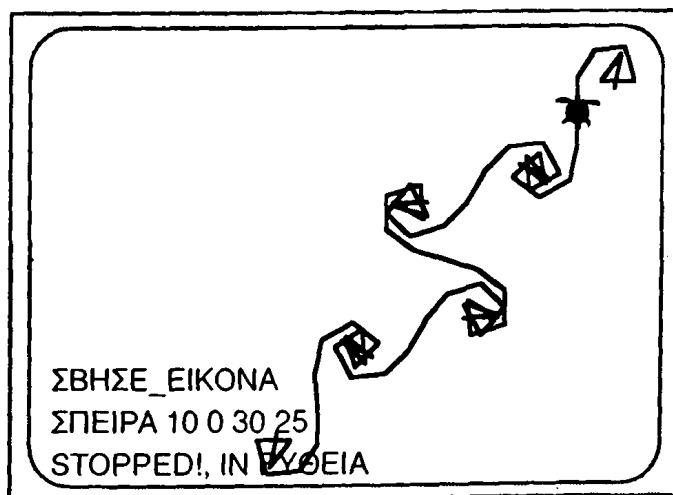
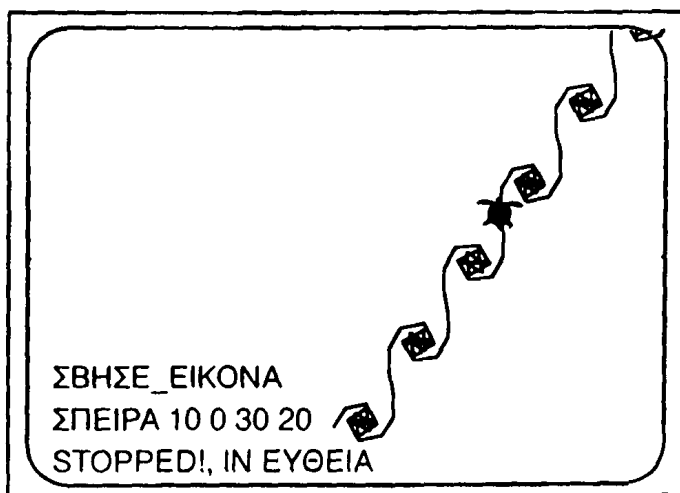
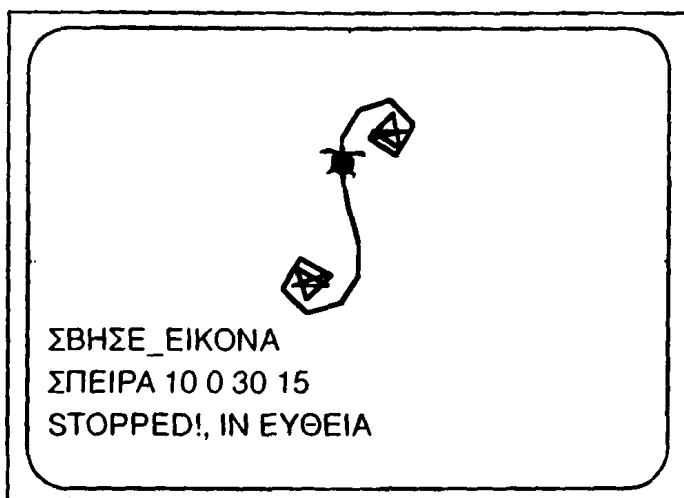
«Το απόλυτο μέγεθος των αντικειμένων ή η συγκεκριμένη χρονική διάρκεια της κίνησής τους δεν είναι αυτά που θα έπρεπε να μας ενδιαφέρουν. Η κεντρική ουσία της δομής και της δυναμικής πολλών αντικειμένων είναι ανεξάρτητη από την κλίμακα κάτω από την οποία τα παρατηρούμε. Η αποδέσμευσή μας από το ειδικό και τη λεπτομέρεια μας καθιστούν ικανούς να διακρίνουμε την παγκοσμιότητα των κοινών νόμων που διέπουν ένα μεγάλο πλήθος φαινομενικά διαφορετικών φυσικών συστημάτων.»

Τάσος Μπούντης,
από τον πρόλογο
του "Χάος-Μιά
νέα επιστήμη".



Μεταβάλλοντας τη μεταβολή της γωνίας

Μεταβάλλοντας την τιμή της μεταβολής της γωνίας παρατηρούμε τη χελώνα να διαγράφει διαφορετικές τροχιές κάθε φορά.



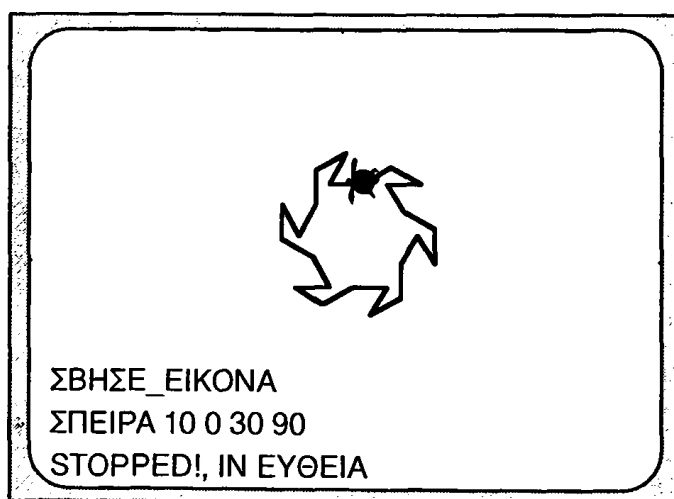
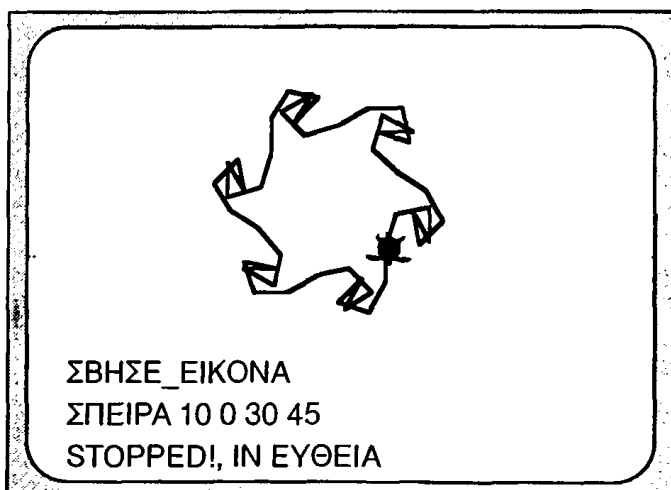
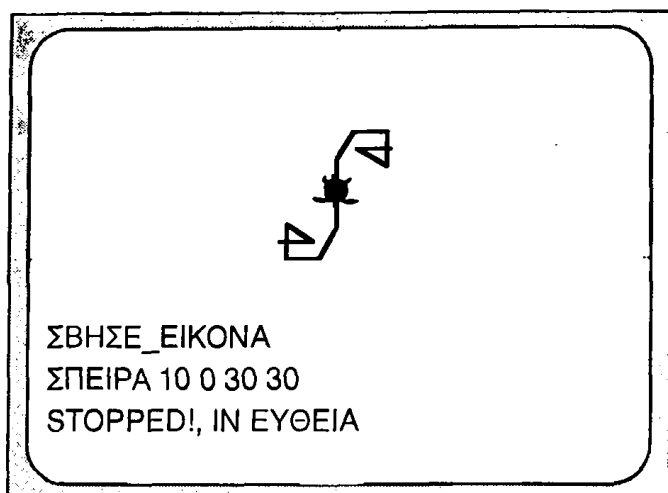
Εδώ η χελώνα επαναλαμβάνει αενάως τη διαδρομή που φαίνεται στο σχήμα, από τη μια άκρη στην άλλη.

Μπορούμε να ομολογήσουμε στους μαθητές μας ότι δεν ξέρουμε πού θα καταλήξει η χελώνα. Μπορούμε όμως να επιχειρήσουμε να ψάξουμε μαζί τους μικραίνοντας την τιμή του μήκους ώστε η καμπύλη να χωράει στην οθόνη.

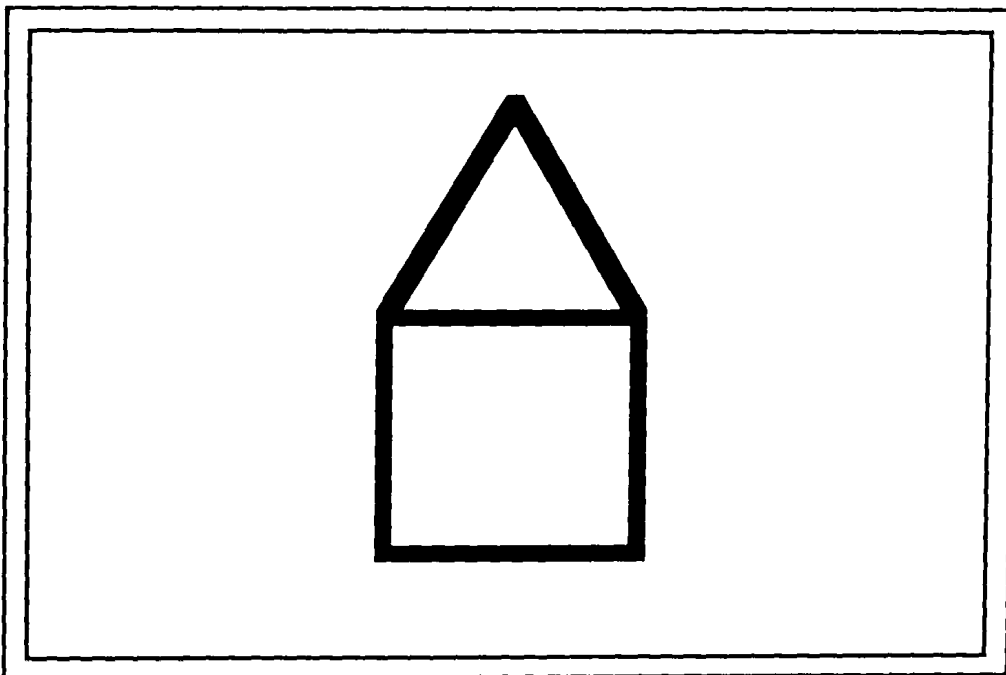
Η χελώνα επαναλαμβάνει αενάως την περισσότερο περίπλοκη διαδρομή που φαίνεται στο σχήμα, από τη μια άκρη στην άλλη.

Μεταβάλλοντας τη μεταβολή της γωνίας

Συνεχίζουμε και εδώ να μεταβάλλουμε την τιμή της μεταβολής της γωνίας.



«Οι μελέτες του Μάντελμπροτ σχετικά με τις ακανόνιστες μορφές των φυσικών διαδικασιών και η εξερεύνηση των απεριόριστα πολυπλόκων σχημάτων βασίζονταν σε μια κοινή έννοια: στη σταθερότητα κλίμακας, στην ιδιότητα της αυτο-ομοιότητας».
James Gleick



8 Πλίνθοι και κέραμοι

Τα δομικά στοιχεία του σπιτιού: Ο όροφος

Ένα απλό σκίτσο σπιτιού αποτελείται από τον όροφο, ένα τετράγωνο, και τη σκεπή, ένα τρίγωνο.

Ας δημιουργήσουμε μια διαδικασία που να σχεδιάζει έναν όροφο.

```
ΤΟ ΟΡΟΦΟΣ  
ΠΟΛΥΓΩΝΟ 30 4  
END
```

Μπορούμε να βελτιώσουμε την προηγούμενη διαδικασία αντικαθιστώντας το σταθερό αριθμό 30 που καθορίζει το ύψος του τετραγώνου με τη μεταβλητή *ΥPSOS*.

```
ΤΟ ΟΡΟΦΟΣ :ΥPSOS  
ΠΟΛΥΓΩΝΟ :ΥPSOS 4  
END
```

Η σκεπή, απεικονίζεται με ένα ισόπλευρο τρίγωνο. Ας δημιουργήσουμε μια διαδικασία που να σχεδιάζει μια σκεπή.

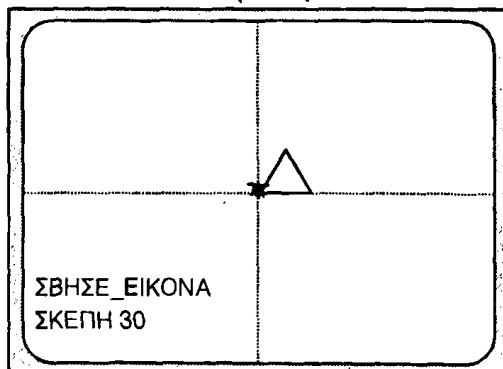
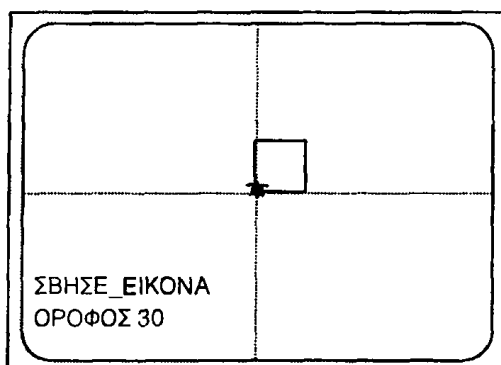
```
ΤΟ ΣΚΕΠΗ  
ΠΟΛΥΓΩΝΟ 30 3  
END
```

Παρόμοια με τη διαδικασία *ΟΡΟΦΟΣ* αντικαθιστούμε τον αριθμό 30 που καθορίζει το μήκος της πλευράς του τριγώνου με τη μεταβλητή *PLEYRA*.

```
ΤΟ ΣΚΕΠΗ :PLEYRA  
ΠΟΛΥΓΩΝΟ :PLEYRA 3  
END
```

Επειδή η διαδικασία *ΟΡΟΦΟΣ* θα συνδυαστεί με τη *ΣΚΕΠΗ* και η μεταβλητή της πρώτης παίρνει την ίδια τιμή με τη μεταβλητή *PLEYRA* της δεύτερης, μετονομάζουμε στον ορισμό της διαδικασίας *ΣΚΕΠΗ* τη μεταβλητή *PLEYRA* σε *ΥPSOS*.

```
ΤΟ ΣΚΕΠΗ :ΥPSOS  
ΠΟΛΥΓΩΝΟ :ΥPSOS 3  
END
```



Συναρμολόγηση του όλου από τα μέρη

Ένας όροφος και μια σκεπή από μόνα τους δε σχηματίζουν ένα σπίτι. Αφού δημιουργήσαμε τα τμήματα του σπιτιού θα πρέπει να τα συναρμολογήσουμε βάζοντάς τα στις σωστές τους θέσεις ώστε να σχηματιστεί το σπίτι.

Μπορούμε να σχεδιάσουμε πρώτα τον όροφο και μετά τη σκεπή.

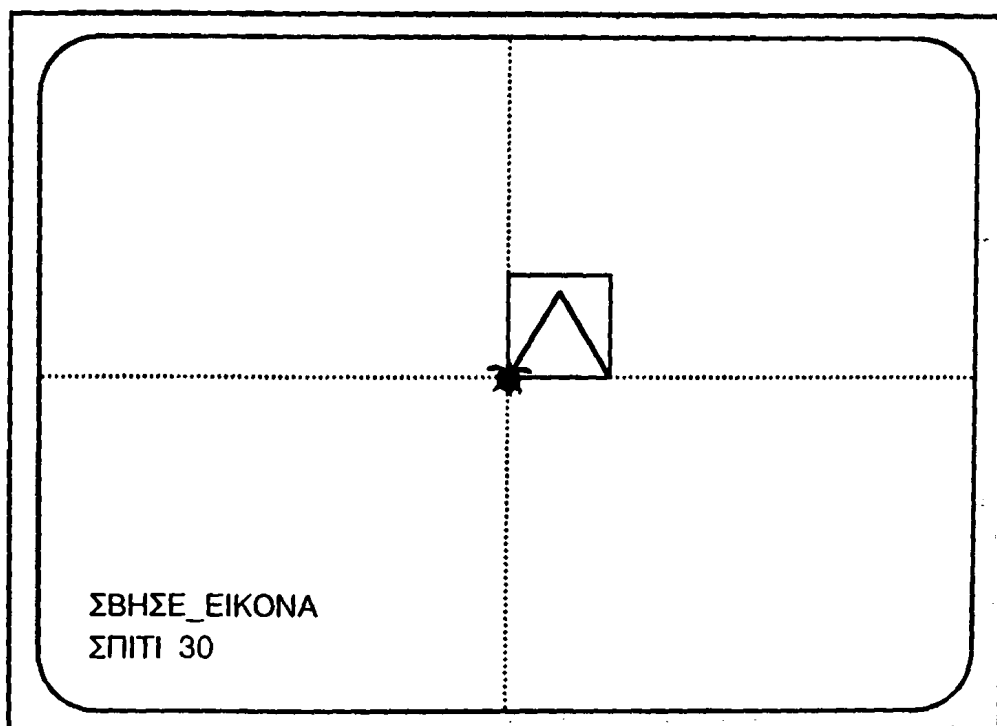
```
ΤΟ ΣΠΙΤΙ
ΟΡΟΦΟΣ
ΣΚΕΠΗ
END
```

Για τους λόγους που αναφέρθηκαν προηγουμένως η διαδικασία ΣΠΙΤΙ μπορεί να αντικατασταθεί με την παρακάτω:

```
ΤΟ ΣΠΙΤΙ :YPSOS
ΟΡΟΦΟΣ :YPSOS
ΣΚΕΠΗ :YPSOS
END
```

Όπως φαίνεται στο ίχνος που άφησε η χελώνα, τα πράγματα δεν ήρθαν όπως τα υπολογίσαμε.

Δεν πειράζει! Ας προσπαθήσουμε να καταλάβουμε γιατί έγινε το λάθος.



Μια διαδικασία μπορεί να περιέχει άλλες, οι οποίες εκτελούνται διαδοχικά με τη σειρά που εμφανίζονται.

Με την ανάλυση διακρίνουμε καταστάσεις που δε δηλώνονται άμεσα στην περιγραφή του αντικειμένου. Η ανάλυση μας επιτρέπει την επαναδόμηση του σχήματος με τα πραγματικά στοιχεία που δεν είναι τίποτε άλλο παρά μια σύνθεση.

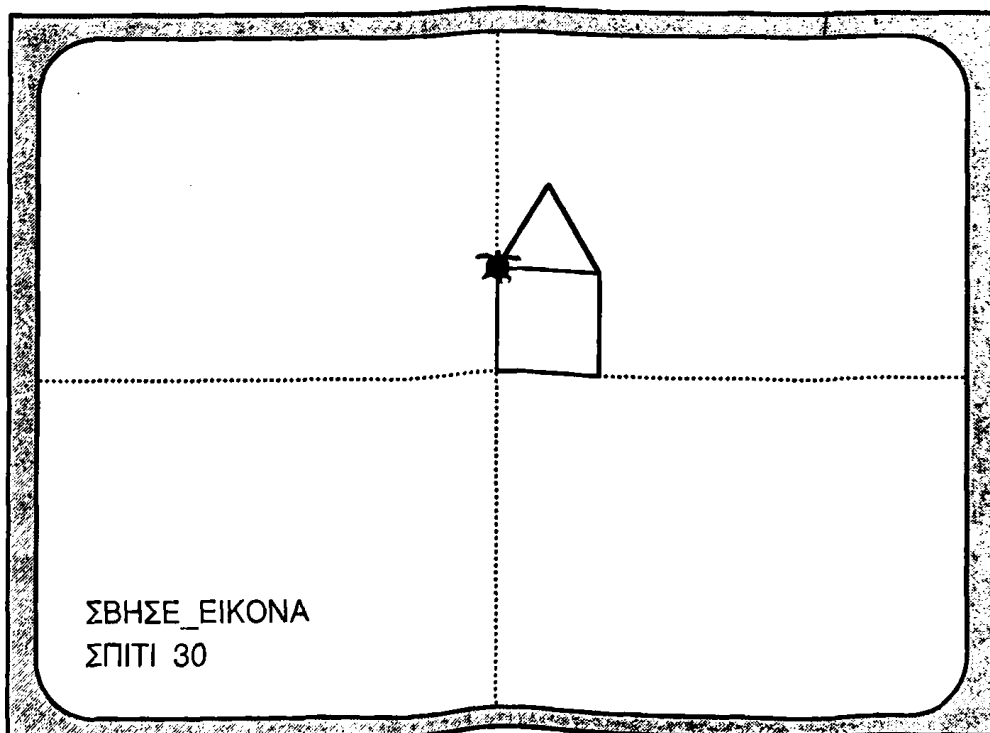
Βάζοντας τα πράγματα στη θέση τους

Για να εντοπίσουμε το λάθος στην προηγούμενη διαδικασία ας ακολουθήσουμε βήμα προς βήμα τις εντολές που εκτελεί η χελώνα.

Στην αρχή η πορεία της είναι ένα τετράγωνο, ο όροφος, με επιστροφή στην αφετηρία της. Στη συνέχεια από το σημείο που βρίσκεται σχεδιάζει ένα τρίγωνο, τη σκεπή.

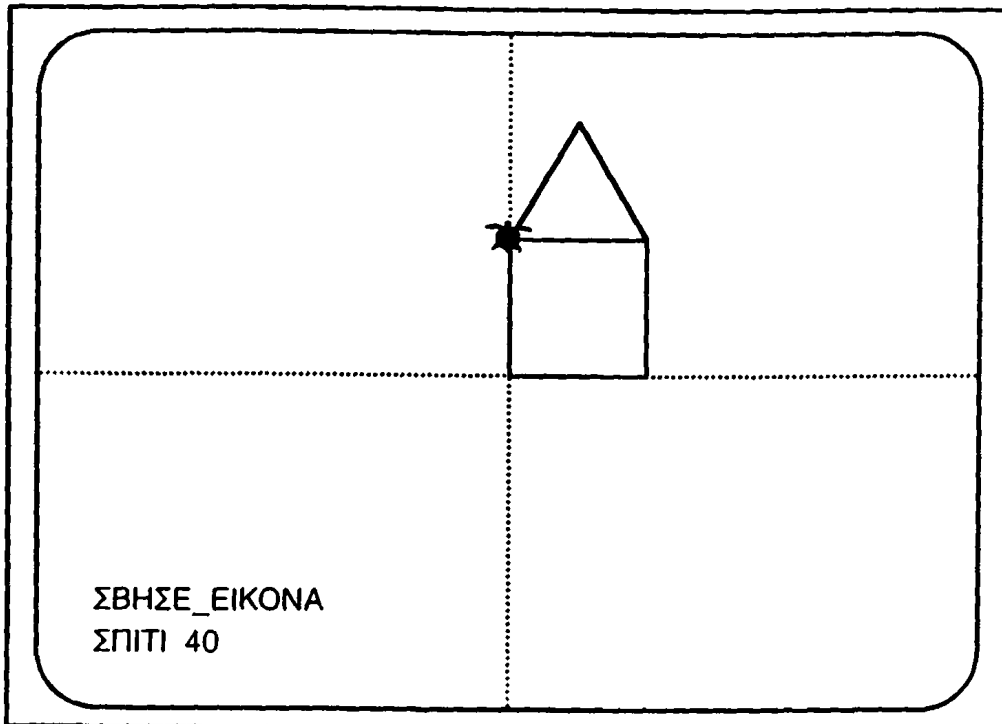
Θα έπρεπε όμως πριν σχεδιάσει τη σκεπή να είχε μετακινηθεί στην πάνω αριστερή γωνία του ορόφου ώστε η σκεπή να τοποθετηθεί πάνω στον όροφο. Χρειάζεται λοιπόν μετά τον όροφο να γίνει αυτή η μετακίνηση με την εντολή ΕΥΘΕΙΑ και για μήκος όσο είναι το ύψος του ορόφου. Ας δούμε λοιπόν πως θα διορθώσουμε τη διαδικασία ΣΠΙΤΙ.

```
ΤΟ ΣΠΙΤΙ :YPSOS  
ΟΡΟΦΟΣ :YPSOS  
ΕΥΘΕΙΑ :YPSOS  
ΣΚΕΠΗ :YPSOS  
END
```

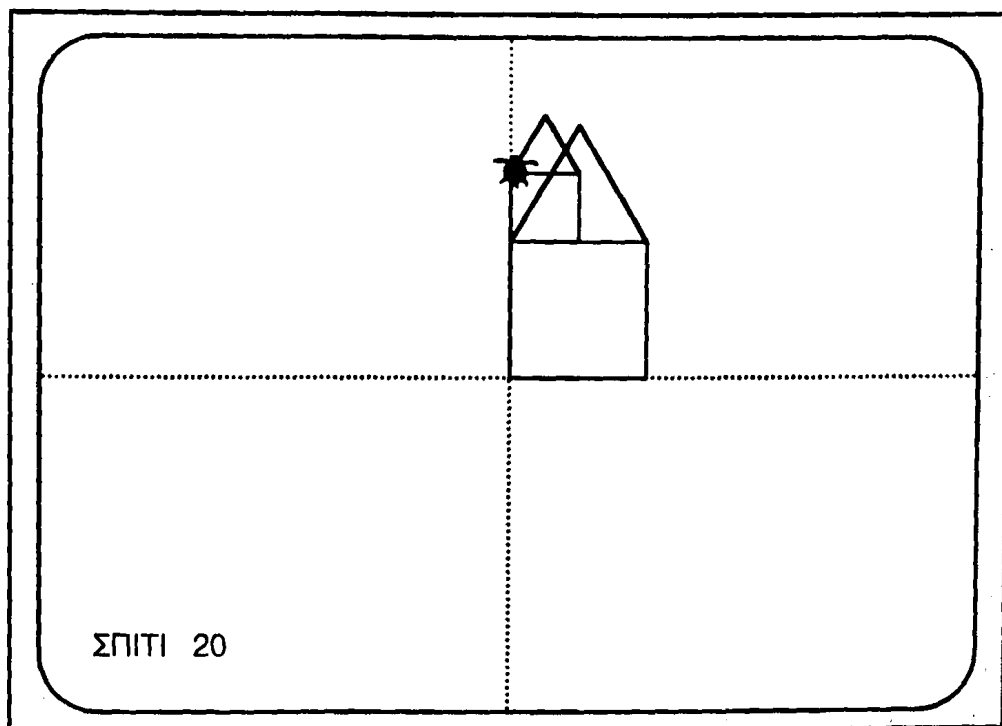


Διαπίστωση ότι κάτι δεν πάει καλά

Ας βάλουμε τη χελώνα να σχεδιάσει ένα σπίτι που ο όροφος να έχει ύψος 40.



Στη συνέχεια χωρίς να σβήσουμε το προηγούμενο σχήμα ας βάλουμε τη χελώνα να σχεδιάσει ένα ακόμα σπίτι με ύψος ορόφου 20. Βλέποντας το αποτέλεσμα μπορούμε να πούμε τι δεν πήγε καλά;



Πάμε πίσω...

Στην προηγούμενη σελίδα αυτό που δεν πήγε καλά ήταν το γεγονός ότι όταν τελείωσε η σχεδίαση ενός σπιτιού η χελώνα έμεινε στη στέγη κοιτάζοντας προς τα πάνω, αντί να κατεβεί στο έδαφος. Πρέπει λοιπόν να κάνουμε τη χελώνα να κινηθεί προς τα πίσω. Αυτό μπορούμε να το κάνουμε με δύο τρόπους.

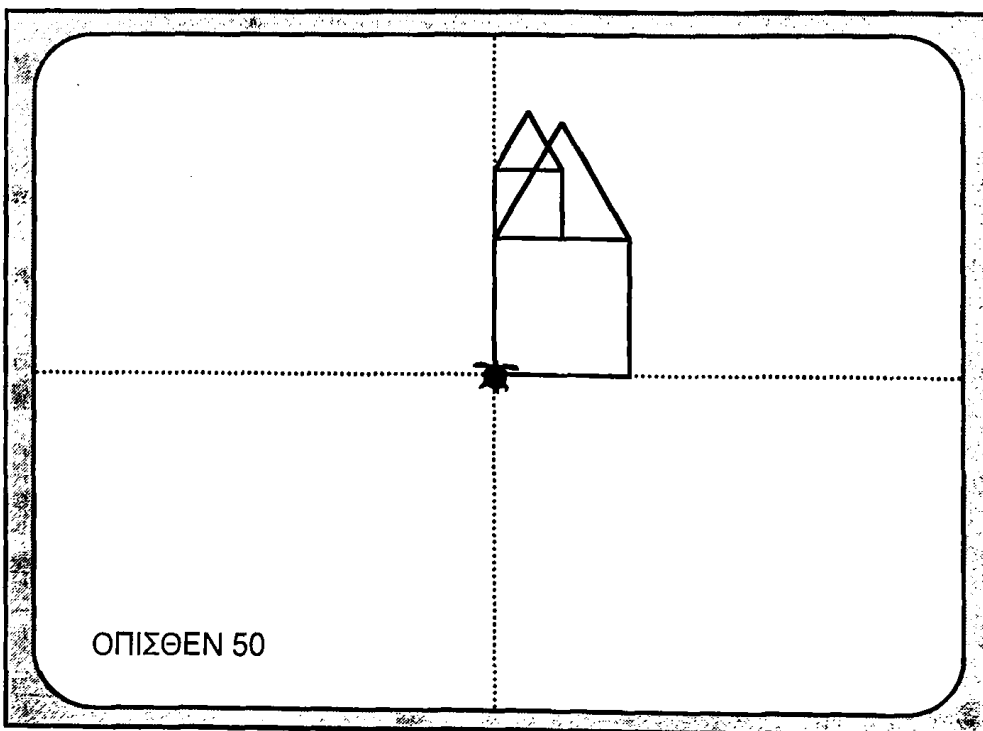
Ο πρώτος είναι να στρίψουμε τη χελώνα προς τα κάτω, να τη μετακινήσουμε ευθύγραμμα και μετά να την ξαναστρίψουμε ώστε να κοιτάζει προς τα πάνω. Αυτή η κίνηση γίνεται με την επόμενη διαδικασία που ονομάζουμε ΟΠΙΣΘΕΝ.

```
ΤΟ ΟΠΙΣΘΕΝ :ΜΗΚΟΣ
ΔΕΞΙΑ 180
ΕΥΘΕΙΑ :ΜΗΚΟΣ
ΔΕΞΙΑ 180
END
```

Ο άλλος τρόπος για να κινηθεί η χελώνα με την όπισθεν είναι να χρησιμοποιήσουμε την εντολή BK της Logo.

```
ΤΟ ΟΠΙΣΘΕΝ :ΜΗΚΟΣ
BK :ΜΗΚΟΣ
END
```

Η εντολή BK της Logo που υπάρχει στο προηγούμενο πρόγραμμα, προέρχεται από την ισοδύναμη εντολή **Back** που στα αγγλικά σημαίνει πίσω.



Λέγοντας στη χελώνα να κινηθεί προς τα πίσω θα πρέπει να της πούμε και πόσα βήματα να κάνει. Στο παράδειγμα που φαίνεται στην οθόνη είπαμε στη χελώνα να υποχωρήσει 50 βήματα και η χελώνα δεν κατάφερε να φτάσει στην κάτω αριστερή γωνία του σπιτιού. Μπορούμε να υπολογίσουμε πόσα βήματα πρέπει να υποχωρήσει ακόμα η χελώνα;

Με το παράδειγμα αυτό μπορούμε να κάνουμε αισθητή την αναγκαιότητα της λύσης της εξίσωσης πρώτου βαθμού.

Η μονοκατοικία

Ας ενσωματώσουμε τις προηγούμενες παρατηρήσεις στον ορισμό της διαδικασίας που σχεδιάζει ένα σπίτι και ας της αλλάξουμε το όνομα ονομάζοντάς την ΜΟΝΟΚΑΤΟΙΚΙΑ.

ΤΟ ΜΟΝΟΚΑΤΟΙΚΙΑ :YPSOS

ΟΡΟΦΟΣ :YPSOS

ΕΥΘΕΙΑ :YPSOS

ΣΚΕΠΗ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS

END

Για να καταλάβουμε τη δομή αυτής της διαδικασίας ας την «τρέξουμε με το χέρι».

Έστω ότι δίνουμε τη διαταγή ΜΟΝΟΚΑΤΟΙΚΙΑ 40.

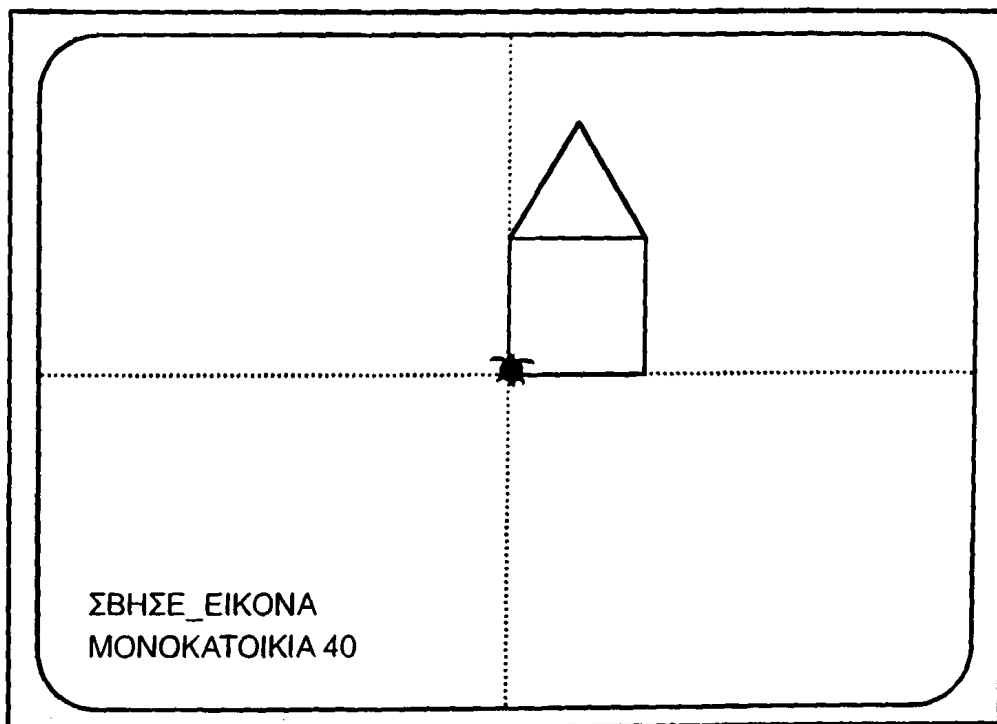
Ο αριθμός 40 μπαίνει ως τιμή στη μεταβλητή YPSOS.

Στη συνέχεια εκτελείται η πρώτη εντολή της διαδικασίας, η εντολή ΟΡΟΦΟΣ η οποία σχεδιάζει ένα τετράγωνο με πλευρά 40 και η χελώνα καταλήγει στην κάτω αριστερή γωνία του.

Με την εκτέλεση της επόμενης εντολής ΕΥΘΕΙΑ η χελώνα θα βρεθεί στην πάνω αριστερή γωνία.

Η εκτέλεση της εντολής ΣΚΕΠΗ θα σχεδιάσει ένα τρίγωνο και θα φέρει τη χελώνα πάλι στην επάνω αριστερή γωνία.

Τέλος η εντολή ΟΠΙΣΘΕΝ θα επιστρέψει τη χελώνα στην κάτω αριστερή γωνία από όπου είχε ξεκινήσει όταν άρχισε να εκτελεί τη διαδικασία ΜΟΝΟΚΑΤΟΙΚΙΑ 40.



«Μια πιο εμφανής διαφορά συνίσταται στο γεγονός ότι ορισμένες εντολές αφήνουν τη χελώνα στην αρχική της κατάσταση. Προγράμματα γραμμένα με αυτό το καθαρό ύφος είναι πολύ πιο κατανοητά και μπορούν ευκολότερα να χρησιμοποιηθούν σε ποικίλες περιπτώσεις. Και παρατηρώντας αυτές τις αρχές τα παιδιά παίρνουν δύο μαθήματα. Μαθαίνουν μια γενική "μαθητειακή αρχή", η οποία είναι να κάνουν τα συστατικά να διευκολύνουν την προσαρμογή. Και μαθαίνουν τη χρήση της πολύ δυναμικής ιδέας "κατάσταση"». Seymour Papert



Το διόροφο

Ας δούμε πως μπορούμε να μάθουμε τη χελώνα να σχεδιάζει ένα κτίριο με δύο ορόφους και σκεπή.

Δεν έχουμε παρά να προσθέσουμε έναν όροφο μετά το ισόγειο μαζί με την αντίστοιχη μετακίνηση προς τα πάνω. Φυσικά στο τέλος θα πρέπει να υπάρχει μια ακόμα ΟΠΙΣΘΕΝ η οποία θα αντιστοιχεί σε αυτήν τη μετακίνηση του επιπλέον ορόφου.

ΤΟ ΔΙΟΡΟΦΟ :YPSOS

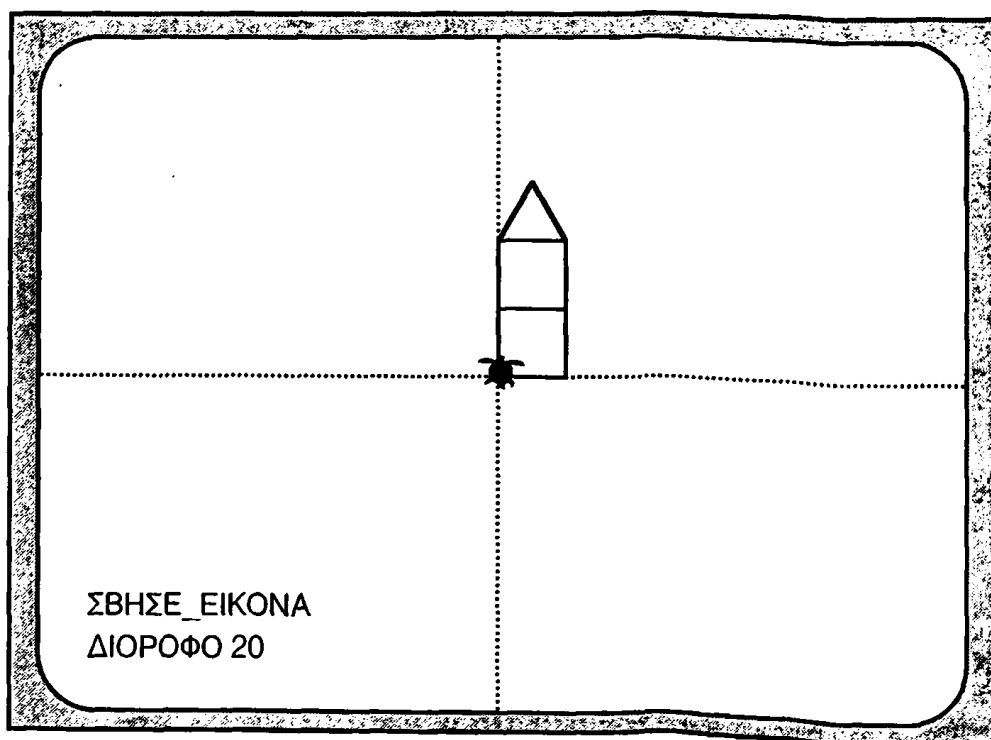
ΟΡΟΦΟΣ :YPSOS
ΕΥΘΕΙΑ :YPSOS

ΟΡΟΦΟΣ :YPSOS
ΕΥΘΕΙΑ :YPSOS

ΣΚΕΠΗ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS
END



...και το τριόροφο

Ας βάλουμε έναν ακόμα όροφο στο προηγούμενο κτίριο.

Δεν έχουμε παρά να προσθέσουμε τον όροφο μετά τους άλλους δύο που υπάρχουν ήδη μαζί με την αντίστοιχη μετακίνηση προς τα πάνω. Φυσικά στο τέλος θα πρέπει να υπάρχει μια ακόμα ΟΠΙΣΘΕΝ η οποία θα αντιστοιχεί σ' αυτήν τη μετακίνηση του επιπλέον ορόφου.

ΤΟ ΤΡΙΟΡΟΦΟ :YPSOS

ΟΡΟΦΟΣ :YPSOS
ΕΥΘΕΙΑ :YPSOS

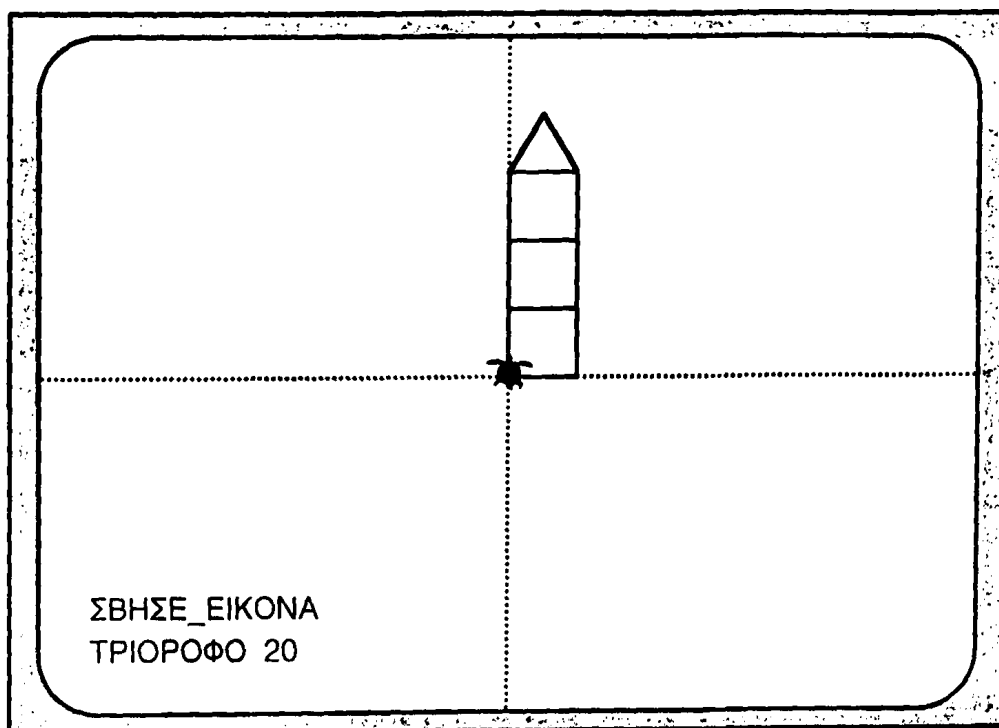
ΟΡΟΦΟΣ :YPSOS
ΕΥΘΕΙΑ :YPSOS

ΟΡΟΦΟΣ :YPSOS
ΕΥΘΕΙΑ :YPSOS

ΣΚΕΠΗ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS
ΟΠΙΣΘΕΝ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS
END



Ένα πολυόροφο κτίριο

Παρατηρούμε ότι στις προηγούμενες διαδικασίες το ζευγάρι των εντολών ΟΡΟΦΟΣ :YPSOS και ΕΥΘΕΙΑ :YPSOS επαναλαμβάνεται τόσες φορές όσοι είναι οι όροφοι του κτιρίου. Θα μπορούσαμε λοιπόν ολόκληρο το πακέτο αυτό με το παραπάνω ζευγάρι εντολών να το αντικαταστήσουμε με την εντολή:

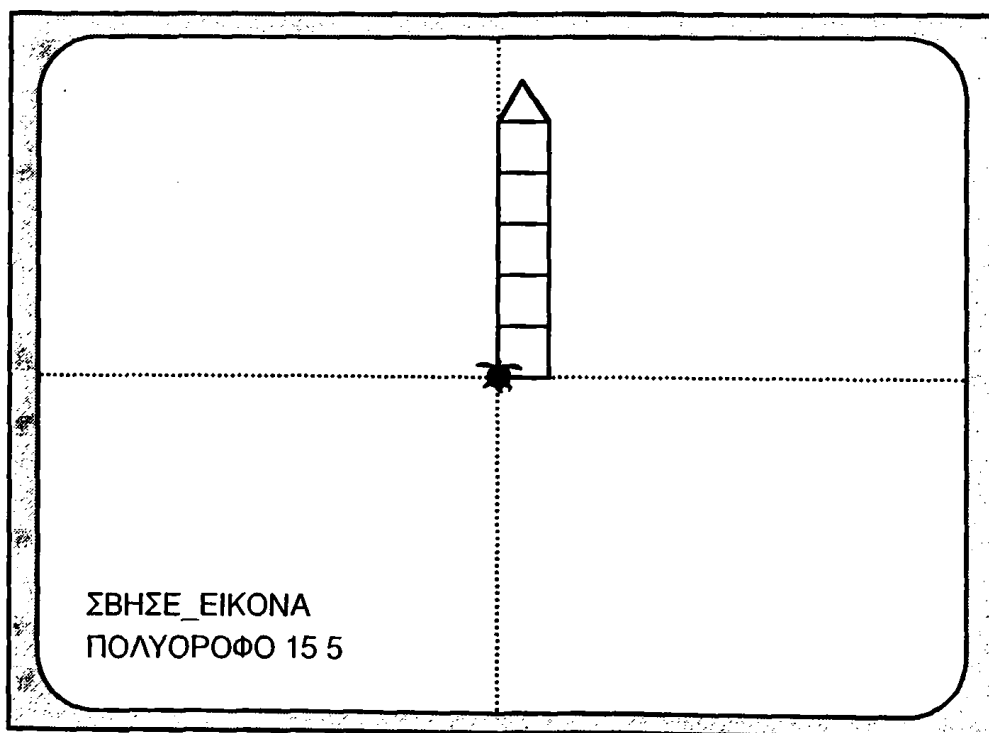
```
REPEAT πλήθος ορόφων [ΟΡΟΦΟΣ :YPSOS ΕΥΘΕΙΑ :YPSOS]
```

Παρόμοια η εντολή ΟΠΙΣΘΕΝ :YPSOS επαναλαμβάνεται τόσες φορές όσοι είναι οι όροφοι του κτιρίου. Θα μπορούσαμε λοιπόν να αντικαταστήσουμε αυτό το πακέτο εντολών με την εντολή:

```
REPEAT πλήθος ορόφων [ΟΠΙΣΘΕΝ :YPSOS]
```

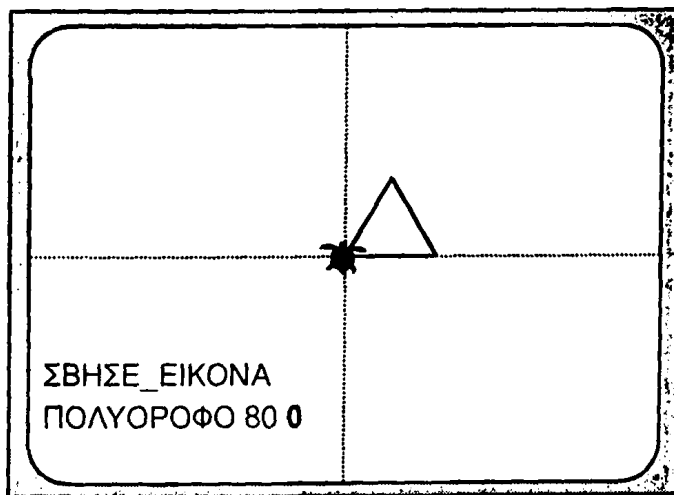
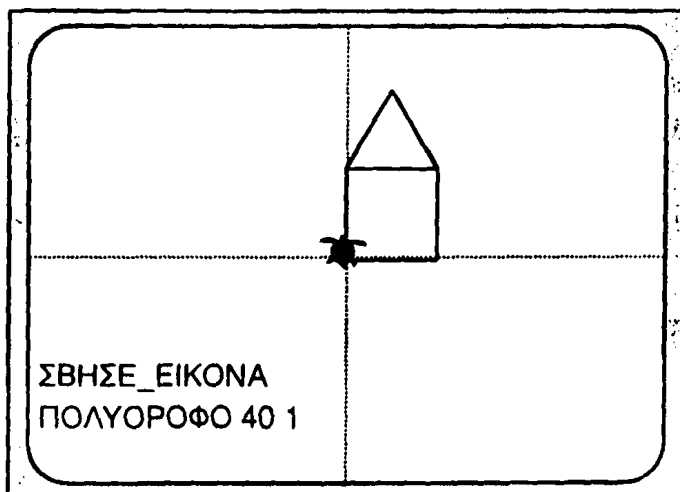
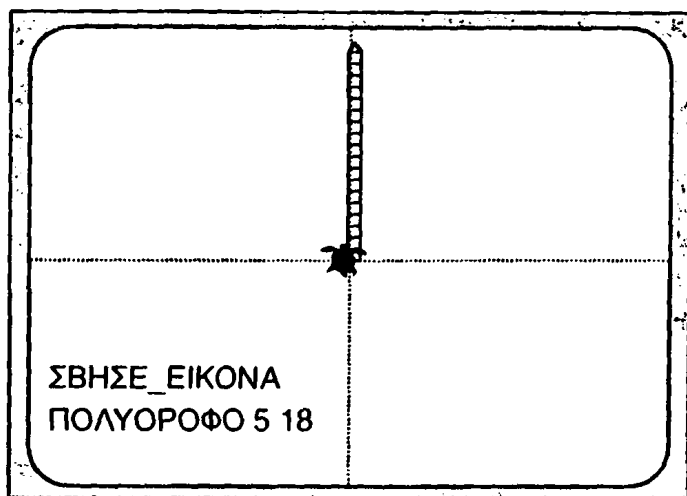
Μπορούμε επομένως να αντικαταστήσουμε τις προηγούμενες διαδικασίες με μια, την ΠΟΛΥΟΡΟΦΟ, που είναι πιο γενική και χρησιμοποιεί μια ακόμα μεταβλητή, την PLHQOS_OROFWN, που θα καθορίζει το πλήθος των ορόφων του κτιρίου.

```
ΤΟ ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
REPEAT :PLHQOS_OROFWN [ΟΡΟΦΟΣ :YPSOS ΕΥΘΕΙΑ :YPSOS]  
ΣΚΕΠΗ :YPSOS  
REPEAT :PLHQOS_OROFWN [ΟΠΙΣΘΕΝ :YPSOS]  
END
```



Ουρανοξύστης ή τσαντίρι;

Ας σχεδιάσουμε μερικά κτίρια δίνοντας διάφορες τιμές στο πλήθος των ορόφων και στο ύψος τους.



Ένα από τα σημεία στα οποία εντοπίζονται συχνά λάθη στον προγραμματισμό είναι οι περιπτώσεις που το πρόγραμμα παίρνει οριακές τιμές.

Ένα βελτιωμένο πολυόροφο κτίριο

Διαπιστώσαμε ότι η διαδικασία ΠΟΛΥΟΡΟΦΟ όταν λάβει το 0 ως τιμή της μεταβλητής PLHQOS_OROFWN σχεδιάζει μια σκεπή. Φυσικά δε νοείται ως κτίριο η σκεπή χωρίς έστω το ισόγειο.

Θα πρέπει λοιπόν το πρόγραμμα να λαμβάνει υπόψη του αυτόν τον περιορισμό. Έτσι,

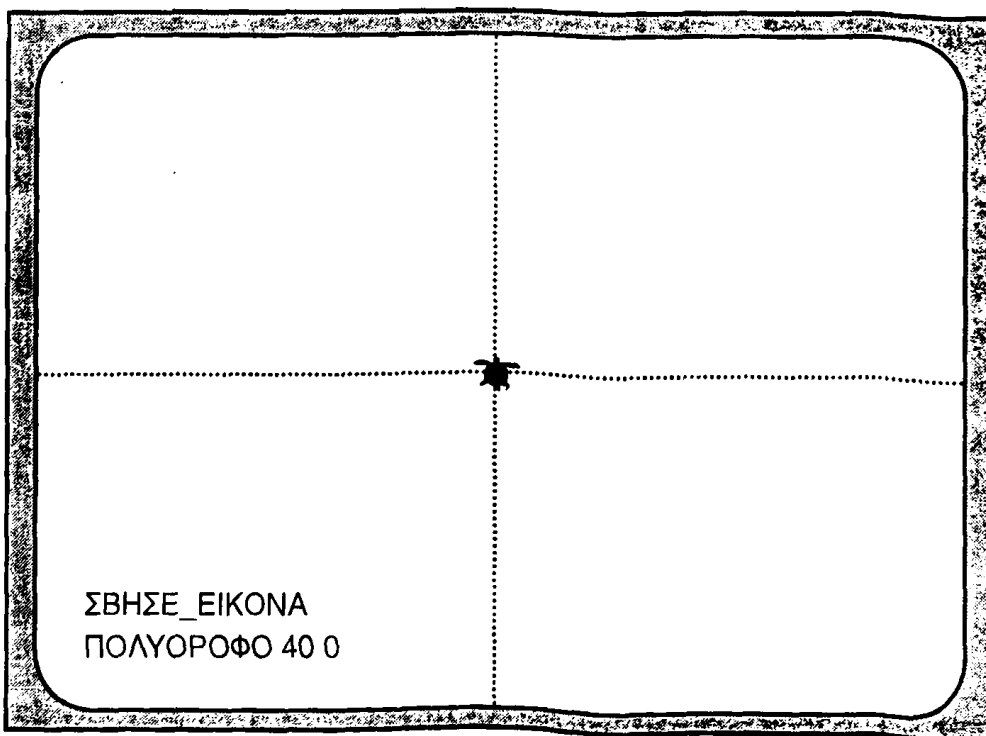
αν το πλήθος των ορόφων είναι μεγαλύτερο του μηδενός
τότε σχεδιάζεται μόνο η σκεπή.

Στον προγραμματισμό αυτή την υπό συνθήκη ενέργεια την υλοποιούμε με την εντολή IF.

Η γενική σύνταξη της εντολής IF είναι:

IF ισχύει η λογική σχέση
THEN κάποιες ενέργειες
ELSE κάποιες άλλες ενέργειες.

```
ΤΟ ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
REPEAT :PLHQOS_OROFWN [ΟΡΟΦΟΣ :YPSOS ΕΥΘΕΙΑ :YPSOS]  
IF (:PLHQOS_OROFWN > 0) [ ΣΚΕΠΗ :YPSOS ]  
REPEAT :PLHQOS_OROFWN [ΟΠΙΣΘΕΝ :YPSOS]  
END
```



Μεταξύ διαφορετικών εκδόσεων της logo γενικά υπάρχουν διαφορετικοί τρόποι σύνταξης των εντολών. Στην LCSII-logo που χρησιμοποιείται εδώ, αν δεν υπάρχει κλάδος else, μετά τη λογική συνθήκη ακολουθούν μέσα σε αγκύλες οι εντολές που θα εκτελεστούν αν ισχύει η συνθήκη.



Μετατόπιση της χελώνας, απόπειρα πρώτη

Βλέπουμε στην προηγούμενη διαδικασία ότι το πακέτο των εντολών

ΔΕΞΙΑ 90

ΕΥΘΕΙΑ :ΥΡΣΟΣ

ΑΡΙΣΤΕΡΑ 90

επαναλαμβάνεται τρεις φορές και επιπλέον ότι χρησιμεύει για να μετακινεί τη χελώνα σε κάποιο διπλανό σημείο.

Μπορούμε με αυτές τις εντολές να ορίσουμε μια νέα διαδικασία την ΚΙΝΗΣΗ_ΔΙΠΛΑ, που θα μετακινεί τη χελώνα δίπλα.

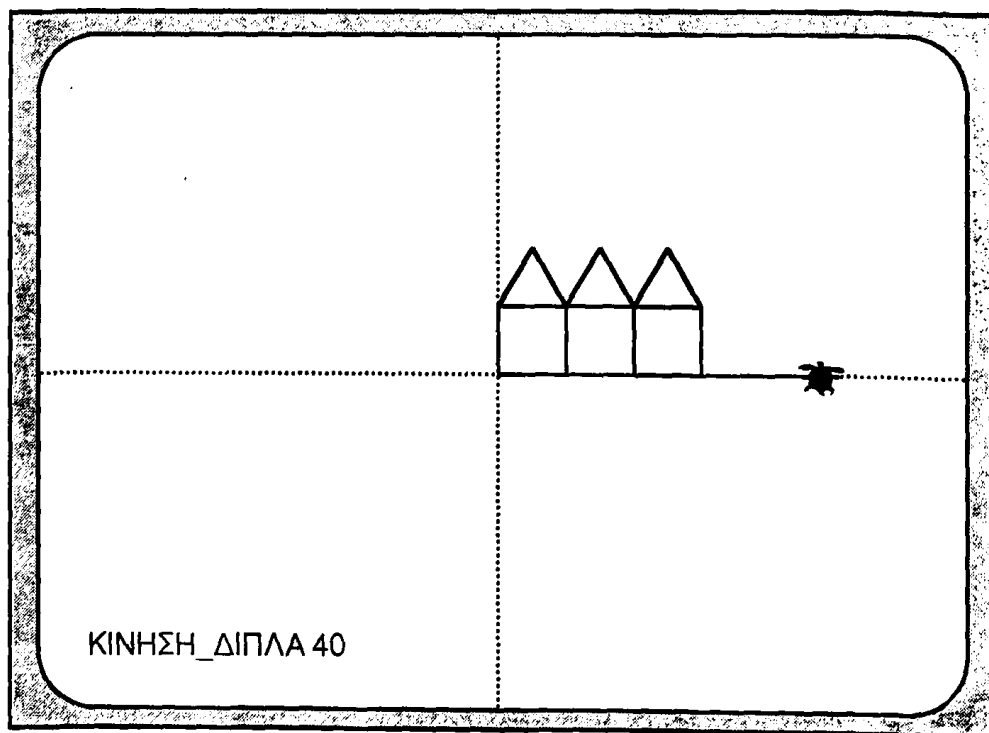
ΤΟ ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΑΡΟΤΑ΢Η

ΔΕΞΙΑ 90

ΕΥΘΕΙΑ :ΑΡΟΤΑ΢Η

ΑΡΙΣΤΕΡΑ 90

ΕΝΔ



Μια πιο “κομψή” γειτονιά

Θα χρησιμοποιήσουμε τη διαδικασία ΚΙΝΗΣΗ_ΔΙΠΛΑ που ορίσαμε προηγουμένως μέσα στον ορισμό της διαδικασίας ΓΕΙΤΟΝΙΑ. Έτσι, η διαδικασία ΓΕΙΤΟΝΙΑ διαμορφώνεται ως εξής:

```
ΤΟ ΓΕΙΤΟΝΙΑ :ΥΡSOS  
ΜΟΝΟΚΑΤΟΙΚΙΑ :ΥΡSOS  
ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΡSOS
```

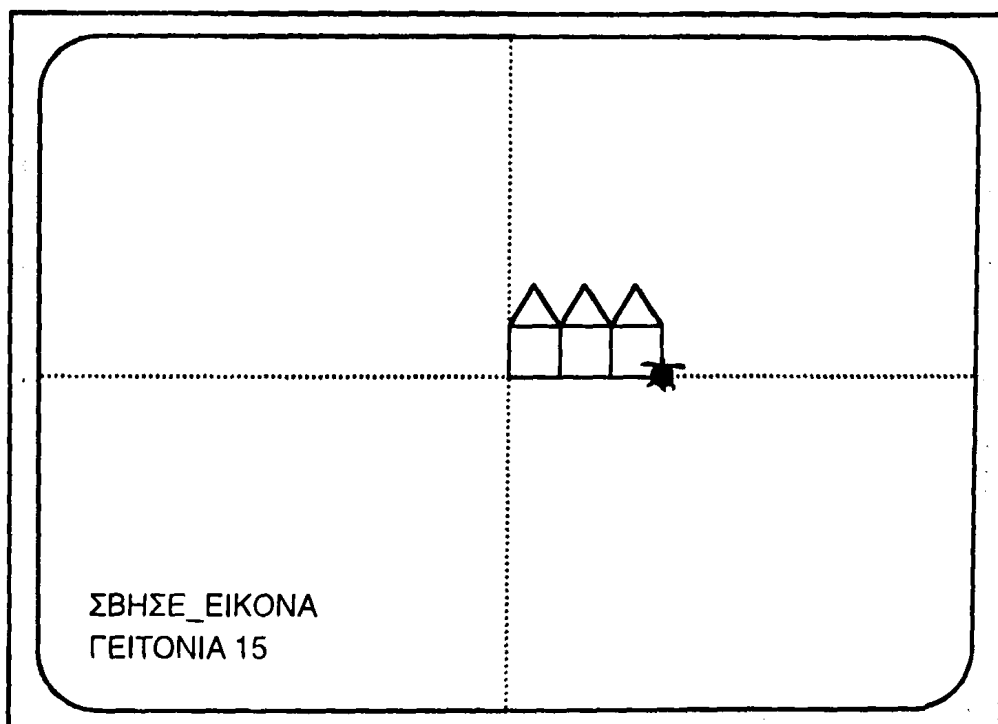
```
ΜΟΝΟΚΑΤΟΙΚΙΑ :ΥΡSOS  
ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΡSOS
```

```
ΜΟΝΟΚΑΤΟΙΚΙΑ :ΥΡSOS  
ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΡSOS  
END
```

Περισσότερο ευανάγνωστο πρόγραμμα.

Συγκρίνετε τις δύο τελευταίες εκδόσεις της διαδικασίας ΓΕΙΤΟΝΙΑ. Η δεύτερη μπορεί να δώσει αρκετές πληροφορίες για το τι κάνει με μια απλή ανάγνωση έστω και από κάποιον που δεν ξέρει από logo.

Τα προγράμματα πρέπει να είναι ευανάγνωστα και ευκολονόητα στον αναγνώστη-προγραμματιστή ώστε να μπορεί να αναζητήσει τα τυχόντα λάθη ή να κάνει βελτιώσεις πολύ εύκολα.



Μια γειτονιά αναπτύσσεται

Μπορούμε να κάνουμε περισσότερο ευέλικτη την προηγούμενη διαδικασία γενικεύοντάς την, έτσι ώστε το πλήθος των σπιτιών που θα σχεδιάζει η χελώνα να καθορίζεται όταν καλούμε τη διαδικασία να εκτελεστεί.

Αυτό γίνεται αν βάλουμε μέσα σε μια εντολή REPEAT το πακέτο των εντολών

```
MONOKATOIKIA :YPSOS
```

```
KINHSEH_DIPLA :YPSOS
```

που αντιπροσωπεύει κάθε σπίτι της γειτονιάς. Το πλήθος των επαναλήψεων είναι το πλήθος των σπιτιών που θα έχει η γειτονιά.

Έτσι η διαδικασία ΓΕΙΤΟΝΙΑ διαμορφώνεται ως εξής:

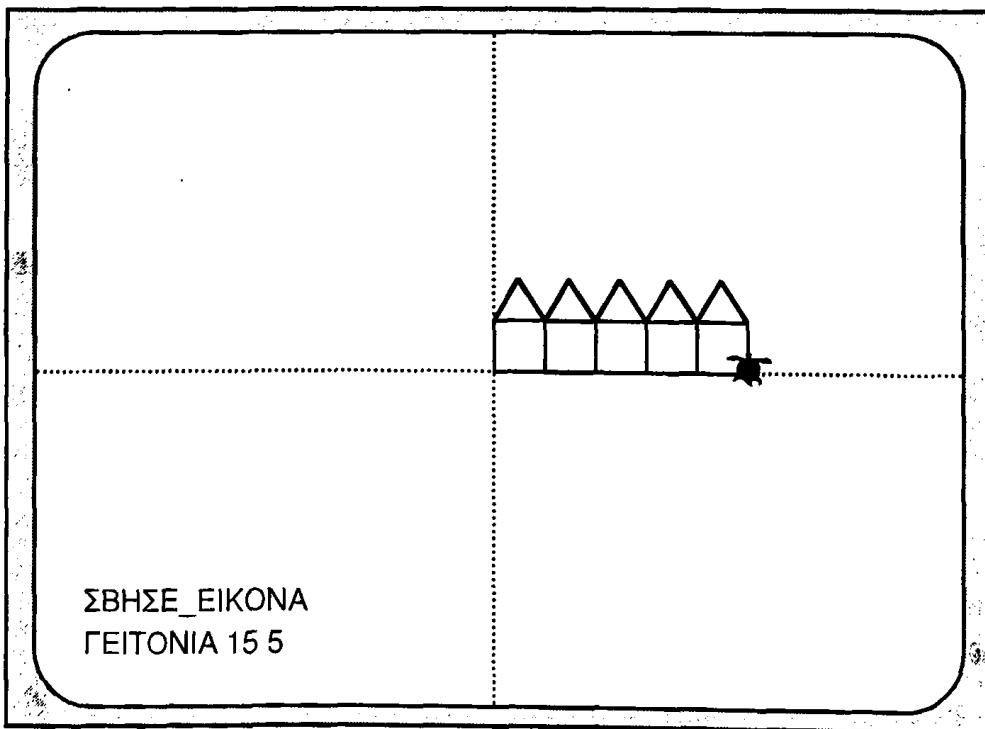
```
TO ΓΕΙΤΟΝΙΑ :YPSOS :PLHQOS_SPITIWN
```

```
REPEAT :PLHQOS_SPITIWN
```

```
  [MONOKATOIKIA :YPSOS
```

```
  KINHSEH_DIPLA :YPSOS]
```

```
END
```



Από τη γειτονιά στη συνοικία

Στη γειτονιά ενός χωριού σχεδόν όλα τα σπίτια είναι μονοκατοικίες.

Πως θα βάλουμε τη χελώνα να σχεδιάσει τη συνοικία μιας πόλης όπου κατά κανόνα υπάρχουν πολυόροφα κτίρια;

Αλλά στη διαδικασία ΓΕΙΤΟΝΙΑ θα αντικαταστήσουμε τη ΜΟΝΟΚΑΤΟΙΚΙΑ με το ΠΟΛΥΟΡΟΦΟ. Φυσικά επειδή η διαδικασία ΠΟΛΥΟΡΟΦΟ για να ξεκινήσει χρειάζεται να έχει και το πλήθος των ορόφων του κτιρίου με την εντολή MAKE θα δώσουμε στη μεταβλητή PLHQOS_OROFWN έστω την τιμή 4.

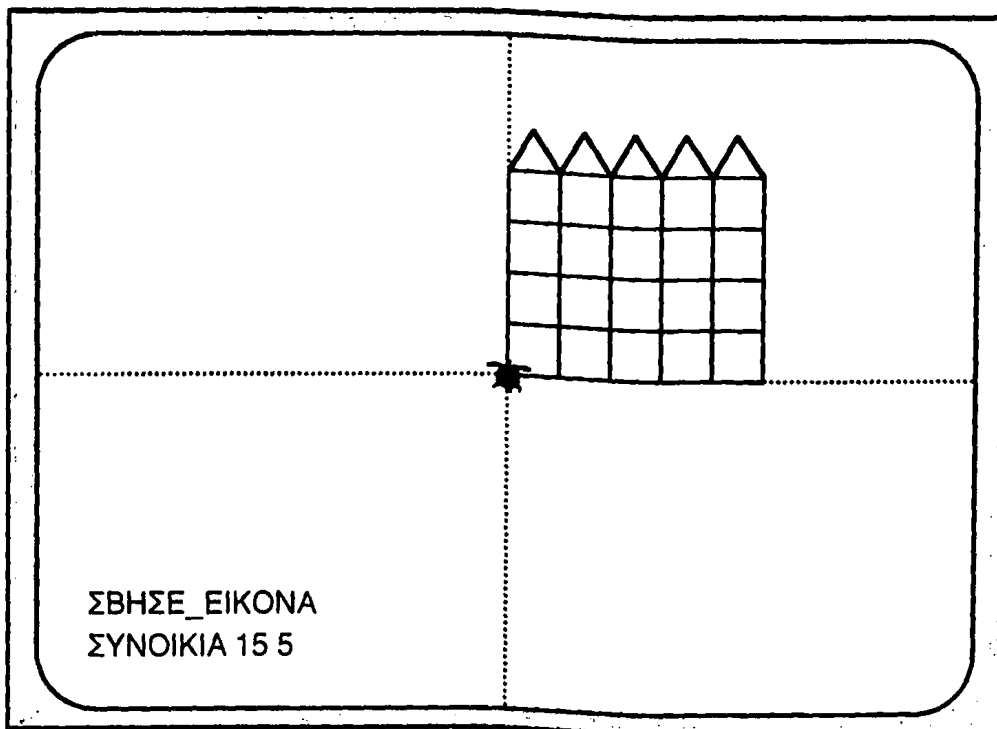
Ας δούμε πως ορίζεται η διαδικασία ΣΥΝΟΙΚΙΑ.

```
TO ΣΥΝΟΙΚΙΑ :YPSOS :PLHQOS_SPITIWN
```

```
MAKE :PLHQOS_OROFWN 4
```

```
REPEAT :PLHQOS_SPITIWN  
  [ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
   ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]
```

```
END
```



Μια πιο πραγματική εικόνα συνοικίας

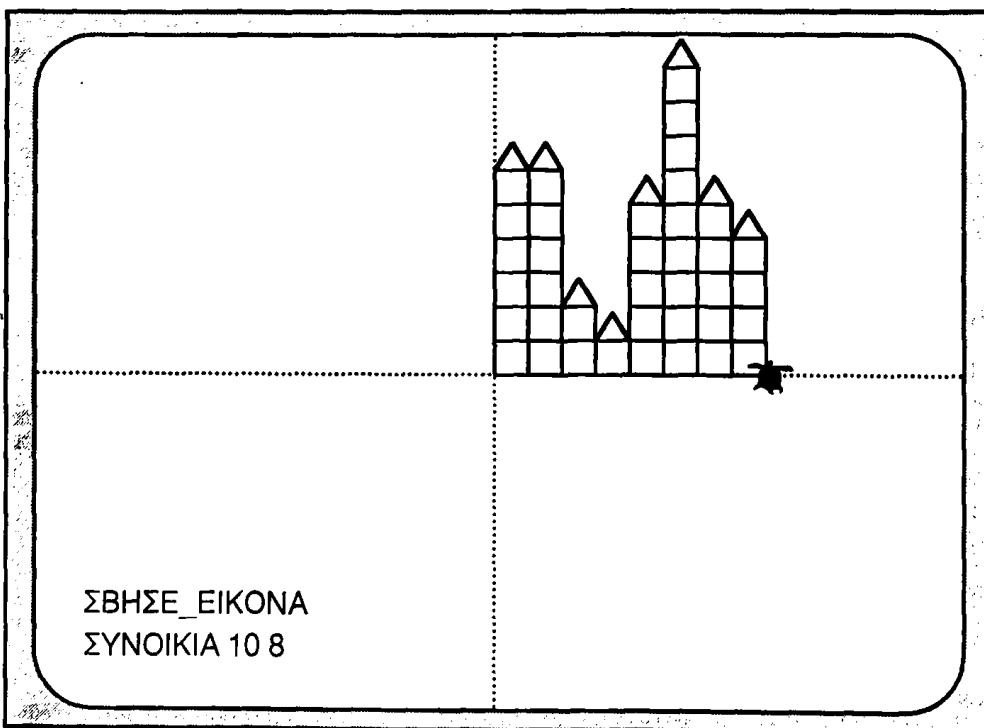
Στην προηγούμενη διαδικασία που σχεδιάζει μια συνοικία, όλα τα κτίρια είναι όμοια. Θα θέλαμε να αφήσουμε την τύχη να αποφασίζει πόσους ορόφους θα έχει κάθε κτίριο της συνοικίας. Αυτό γίνεται αν η μεταβλητή PLHQOS_OROFWN παίρνει τιμή με τη βοήθεια της συνάρτησης RANDOM. Ας δεχθούμε ότι δεν επιτρέπεται κανένα κτίριο να έχει πάνω από 10 ορόφους (το μέγιστο επιτρεπόμενο όριο είναι οι 9 όροφοι).

Ας δούμε πως θα βελτιωθεί η διαδικασία ΣΥΝΟΙΚΙΑ με τη χρήση της συνάρτησης RANDOM.

```
TO ΣΥΝΟΙΚΙΑ :YPSOS :PLHQOS_SPITIWN  
REPEAT :PLHQOS_SPITIWN  
  [MAKE "PLHQOS_OROFWN RANDOM 10  
  ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]  
END
```

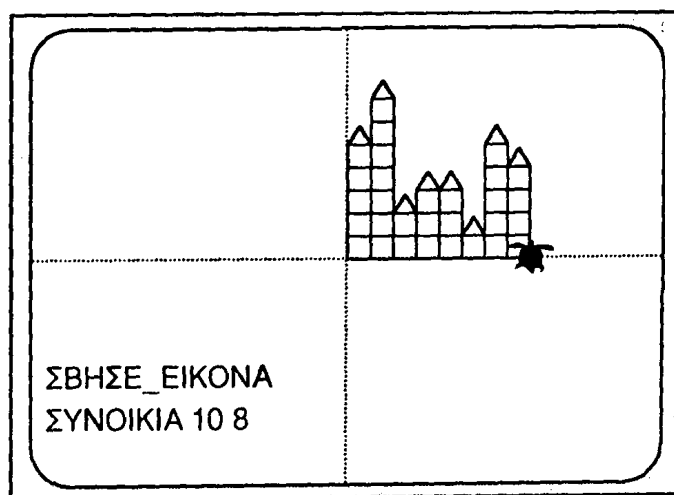
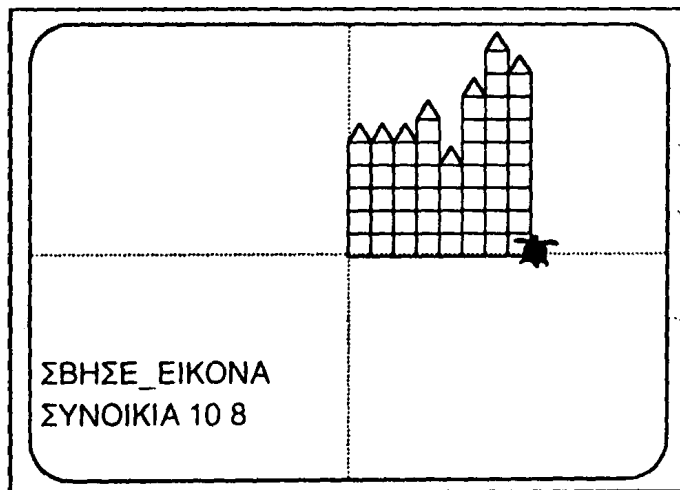
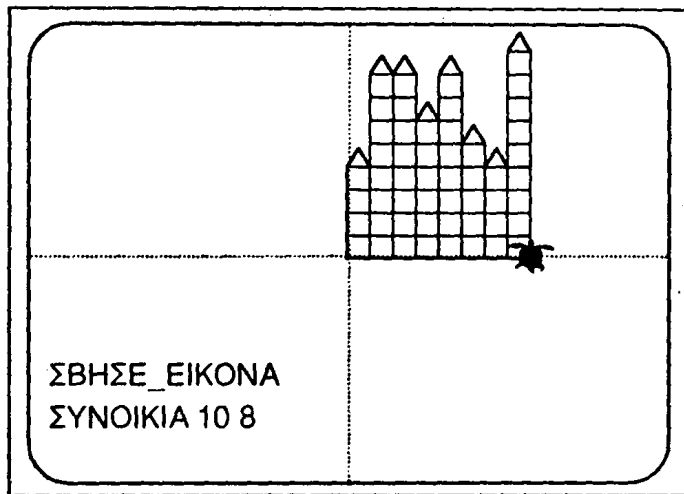
Η συνάρτηση RANDOM επιστρέφει στην τύχη έναν ακέραιο αριθμό που κυμαίνεται από το 0 μέχρι τον αμέσως μικρότερο ακέραιο του αριθμού με τον οποίο την τροφοδοτούμε.

Η συνάρτηση RANDOM 5, μπορεί να επιστρέψει έναν από τους παρακάτω αριθμούς: 0, 1, 2, 3, 4.



Το τυχαίο και η ομορφιά του (1)

Θα δούμε τώρα μερικές συνοικίες που σχεδιάζονται με την ίδια διαδικασία η οποία καλείται με τις ίδιες τιμές των παραμέτρων.



Μια γενίκευση της διαδικασίας

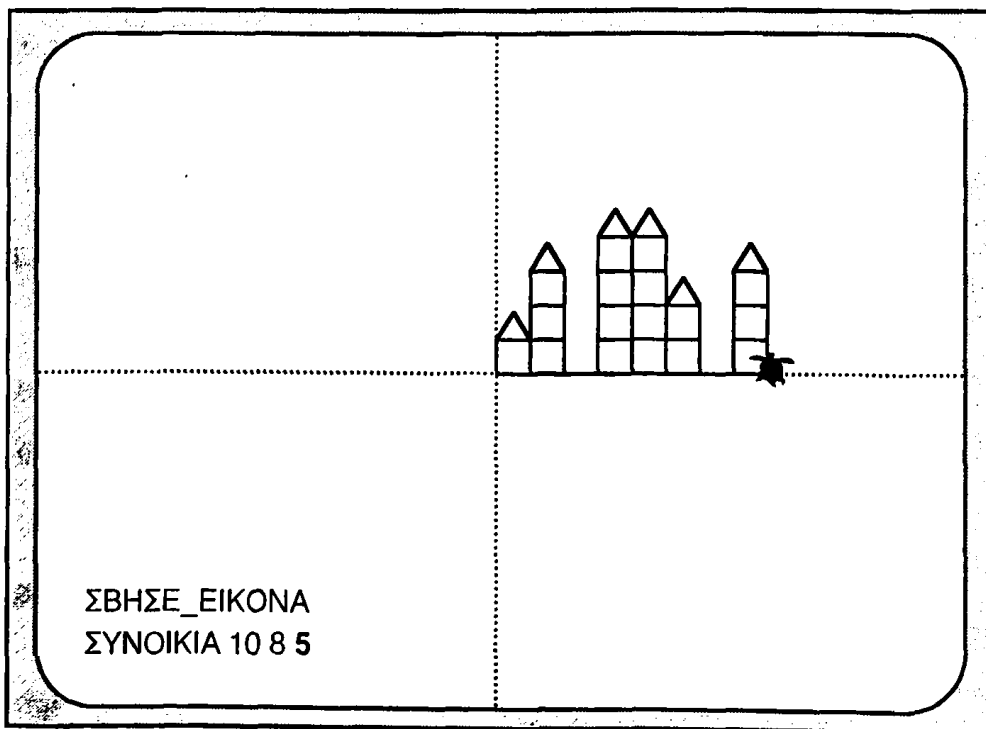
Στην προηγούμενη διαδικασία ως όριο των ορόφων χρησιμοποιείται στη συνάρτηση RANDOM το 10. Κάθε φορά που επιθυμούμε όμως να αλλάξουμε αυτό το όριο θα πρέπει να επεμβαίνουμε στο πρόγραμμα και να βάζουμε κάποια άλλη τιμή.

Για να γίνει περισσότερο ευέλικτη η διαδικασία, πρέπει να αντικαταστήσουμε τη σταθερή τιμή 10 με μια μεταβλητή την ORIO_OROF που θα δηλώνει το όριο του πλήθους των ορόφων.

Έτσι, η διαδικασία τροποποιείται ως εξής:

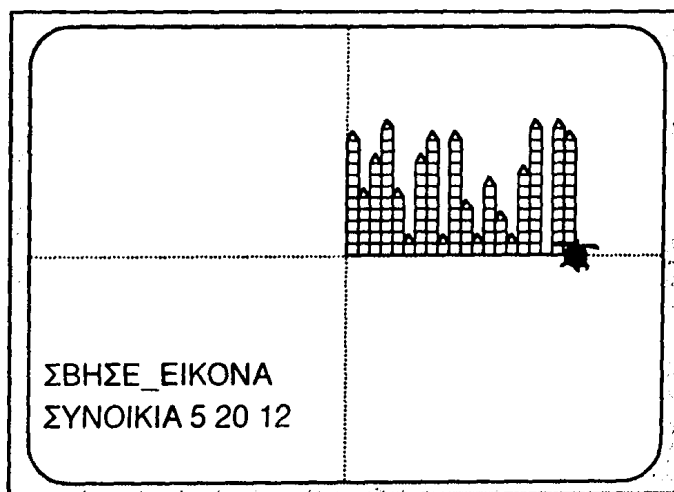
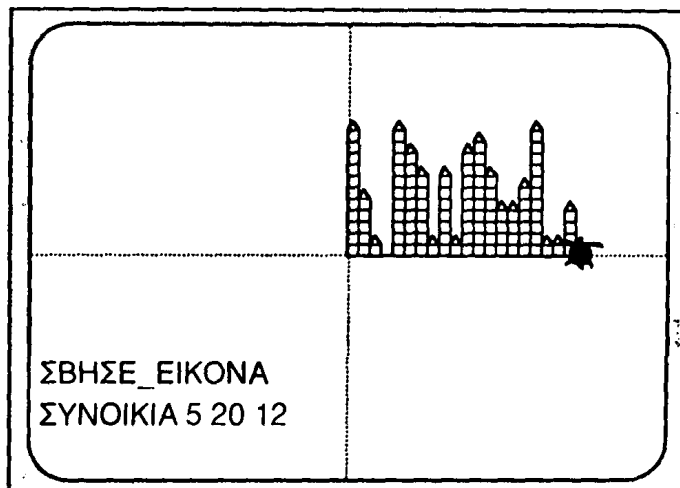
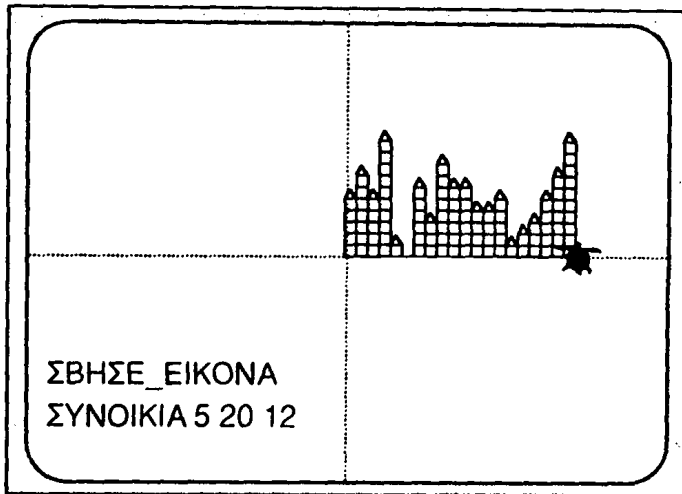
```
TO ΣΥΝΟΙΚΙΑ :YPSOS :PLHQOS_SPITIWN :ORIO_OROF
REPEAT :PLHQOS_SPITIWN
  [MAKE "PLHQOS_OROFWN RANDOM ORIO_OROF
  ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]
END
```

Για να αποφύγουμε συνεχείς επεμβάσεις στον κώδικα του προγράμματος, εισάγουμε ως μεταβλητές εισόδου τις επιθυμητές παραμέτρους.



Το τυχαίο και η ομορφιά του (2)

Ας διασκεδάσουμε βάζοντας τη χελώνα να σχεδιάσει μερικές συνοικίες.



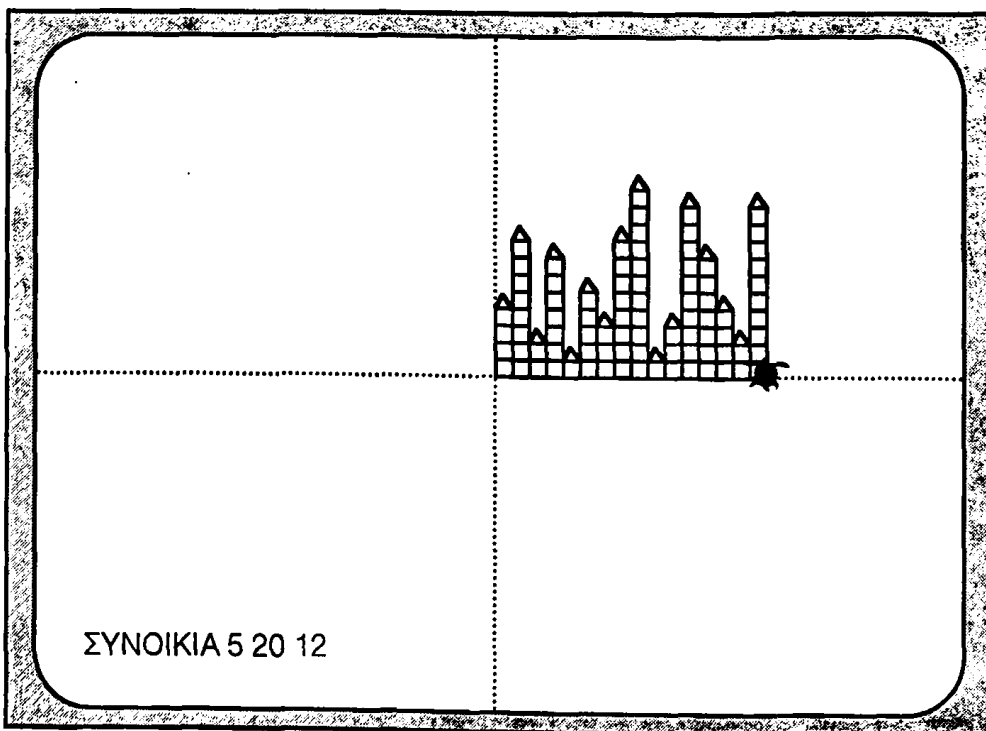
Μια ακόμα βελτίωση της συνοικίας

Οι προηγούμενες συνοικίες είχαν ένα εκ των προτέρων καθορισμένο πλήθος κτιρίων. Τι θα λέγατε αν θέλαμε να έχουμε τυχαίο αριθμό κτιρίων σε μια συνοικία;

Όταν θα καλούμε τη διαδικασία, θα της δίνουμε ένα μέγιστο πλήθος κτιρίων που θα μπορεί να έχει. Το πλήθος των κτιρίων θα προκύπτει τυχαία με τη χρήση της συνάρτησης RANDOM.

Ας δούμε πως θα διαμορφωθεί η διαδικασία ΣΥΝΟΙΚΙΑ.

```
TO ΣΥΝΟΙΚΙΑ :ΥΠΣΟΣ :ΟΡΙΟ_ΣΠΙΤΙΩΝ :ΟΡΙΟ_ΟΡΟΦ
MAKE "ΠΛΗΘΟΣ_ΣΠΙΤΙΩΝ RANDOM :ΟΡΙΟ_ΣΠΙΤΙΩΝ
REPEAT :ΠΛΗΘΟΣ_ΣΠΙΤΙΩΝ
  [MAKE "ΠΛΗΘΟΣ_ΟΡΟΦΩΝ RANDOM ΟΡΙΟ_ΟΡΟΦ
  ΠΟΛΥΟΡΟΦΟ :ΥΠΣΟΣ :ΠΛΗΘΟΣ_ΟΡΟΦΩΝ
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΠΣΟΣ]
END
```



Η πόλη

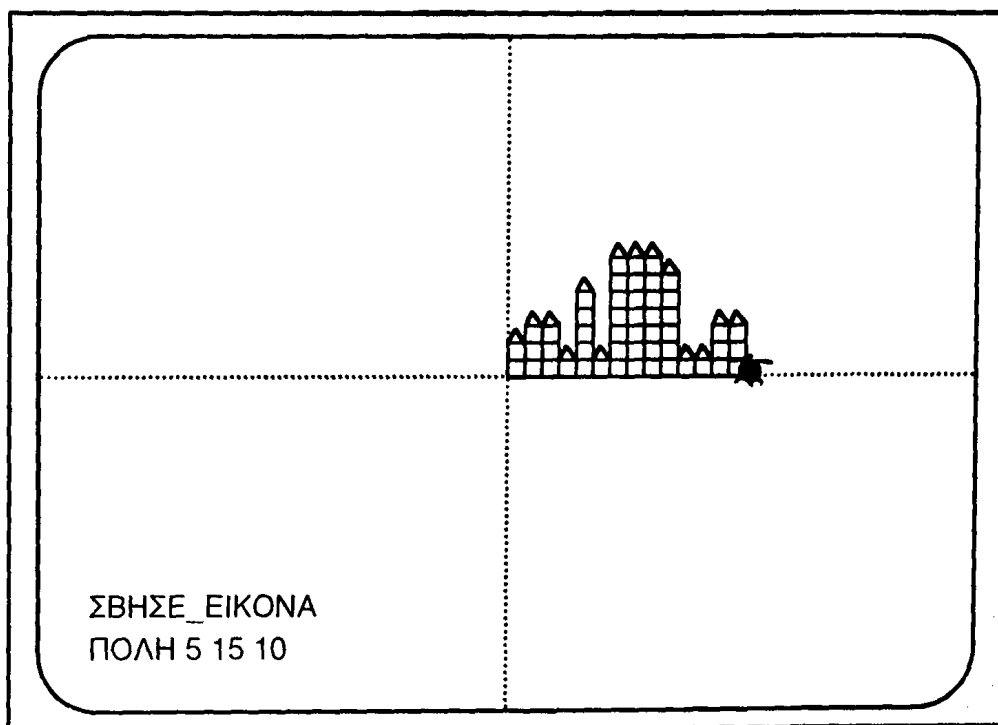
Μια συνοικία είναι ένα τμήμα μιας πόλης. Θα μπορούσαμε να διδάξουμε τη χελώνα να σχεδιάζει μια πόλη βάζοντας μια διαδικασία να σχεδιάζει πολλές συνοικίες.

Θα εκμεταλλευτούμε την εμπειρία που αποκτήσαμε προηγουμένως με τη συνάρτηση RANDOM και θα χρησιμοποιήσουμε τη μεταβλητή ORIO_SYNK ως την ανώτατη τιμή του πλήθους των συνοικιών που θα μπορεί να έχει η πόλη.

Ας δούμε τη διαδικασία ΠΟΛΗ.

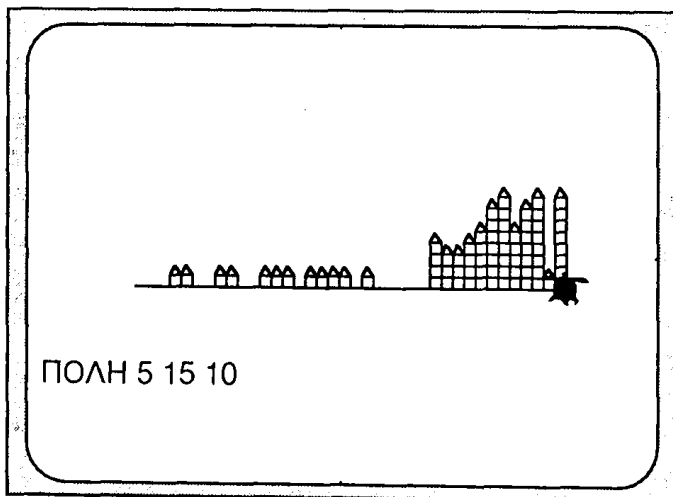
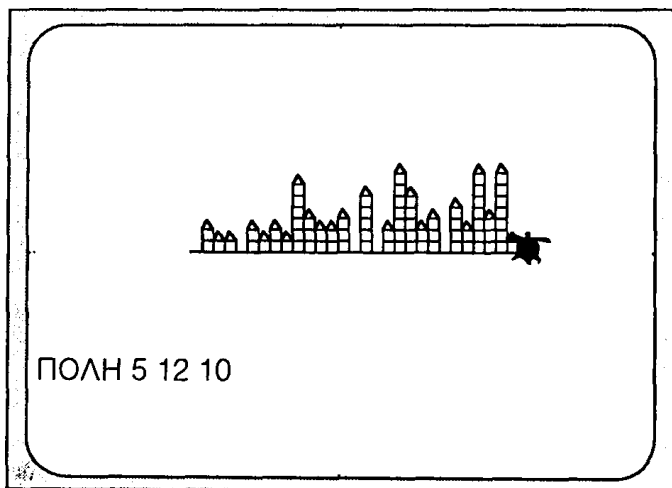
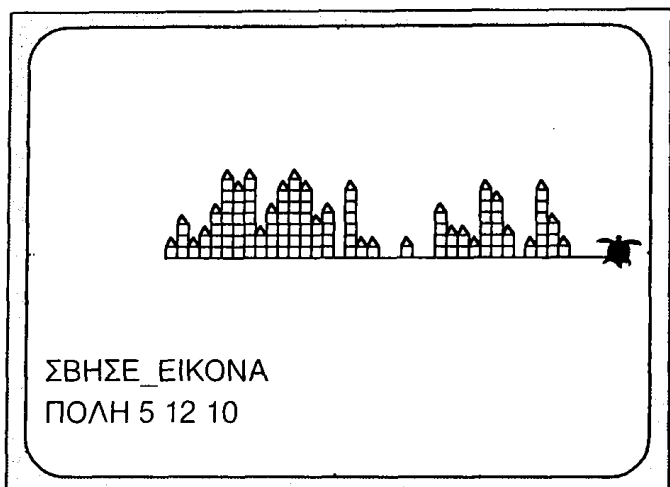
```
TO ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN  
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN  
REPEAT :PLHQOS_SYNOIKIWN  
  [MAKE "ORIO_OROF RANDOM 10  
  ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF]  
END
```

Στη διαδικασία ΠΟΛΗ υπάρχουν τρεις απρόβλεπτοι παράγοντες: Το πόσες συνοικίες θα έχει η πόλη (RANDOM ORIO_SYNOIKIWN), το πόσα κτίρια θα έχει κάθε συνοικία (RANDOM ORIO_SPITIWN) και το πόσους ορόφους θα έχει κάθε κτίριο (RANDOM 10).



Το τυχαίο και η ομορφιά του (3)

Ας βάλουμε τη χελώνα να σχεδιάσει μερικές πόλεις.



Εντοπισμός και διόρθωση του σφάλματος

Στη συγκεκριμένη περίπτωση το μήνυμα «RANDOM doesn't like 0 as input, in ΣΥΝΟΙΚΙΑ» μας πληροφορεί ότι δόθηκε το 0 ως στοιχείο εισόδου της συνάρτησης RANDOM της διαδικασίας ΣΥΝΟΙΚΙΑ.

Μάλιστα παρατηρώντας το σχέδιο που έφτιαξε η χελώνα μπορούμε να υποθέσουμε ότι πριν τύχει να δοθεί το 0, η διαδικασία συνοικία θα πρέπει να είχε εκτελεστεί αρκετές φορές.

Ας δούμε λοιπόν πως συνδέονται οι διαδικασίες ΠΟΛΗ και ΣΥΝΟΙΚΙΑ.

```
TO ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN  
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN  
REPEAT :PLHQOS_SYNOIKIWN  
  [MAKE "ORIO_OROF RANDOM 10
```

```
    ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF]  
END
```

```
TO ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF  
MAKE "PLHQOS_SPITIWN RANDOM :ORIO_SPITIWN  
REPEAT :PLHQOS_SPITIWN  
  [MAKE "PLHQOS_OROFWN RANDOM ORIO_OROF  
    ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
    ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]  
END
```

Το πρόβλημα δηλαδή εμφανίζεται σε ένα από τα δύο RANDOM. Το πρώτο RANDOM έχει σχέση με τη μεταβλητή ORIO_SPITIWN η οποία διατηρεί την τιμή που πήρε από την αρχή όταν ο χρήστης διέταξε τη χελώνα να σχεδιάσει την ΠΟΛΗ. Άρα το λάθος πρέπει να είναι σε κάποιο 0 που μπαίνει ως τιμή στη μεταβλητή ORIO_OROF. Όμως η μεταβλητή ORIO_OROF της διαδικασίας ΣΥΝΟΙΚΙΑ παίρνει τιμή από τη μεταβλητή ORIO_OROF της διαδικασίας ΠΟΛΗ. Άρα το 0 πρέπει να προέρχεται από εκεί. Στη διαδικασία ΠΟΛΗ η μεταβλητή ORIO_OROF παίρνει τιμή στην εντολή MAKE "ORIO_OROF RANDOM 10.

Πράγματι οι πιθανές τιμές που μπορεί να πάρει η μεταβλητή ORIO_OROF είναι 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Έτσι εξηγείται και το γεγονός ότι το σφάλμα δεν εμφανίζεται αμέσως ή και καθόλου αλλά μετά τη σχεδίαση μερικών συνοικιών.



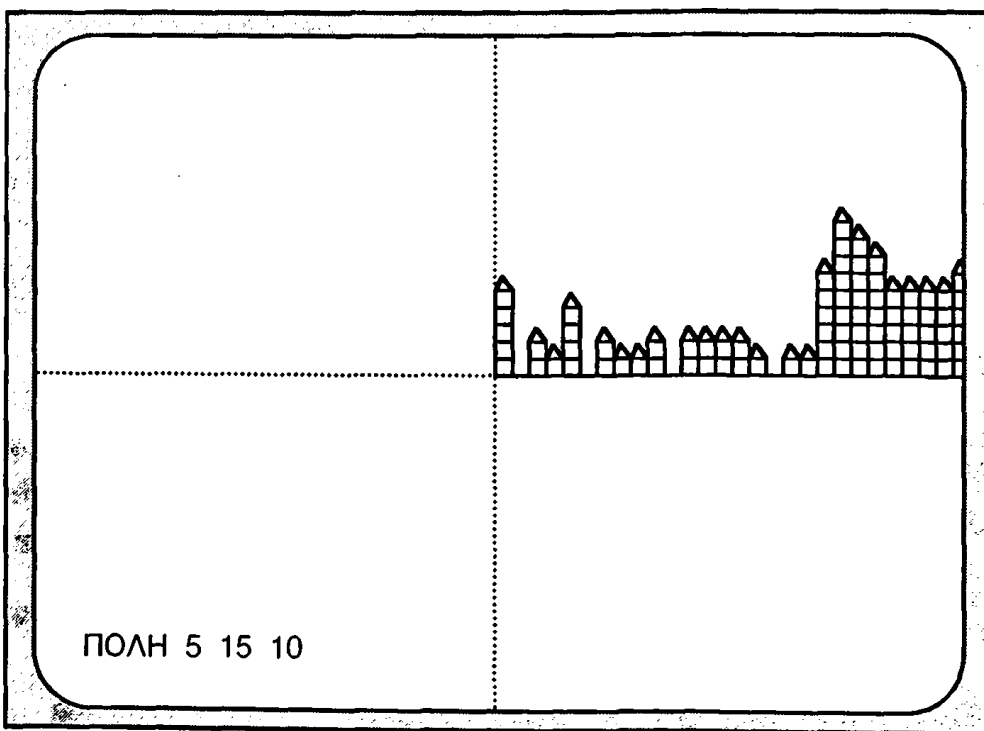
Η «διορθωμένη» πόλη

Όπως είδαμε θα πρέπει να κάνουμε τις απαραίτητες αλλαγές στο πρόγραμμα έτσι ώστε να αποκλείσουμε την περίπτωση η μεταβλητή ORIO_OROF να παίρνει την τιμή 0. Ένας απλός τρόπος είναι στην τιμή που θα προκύψει από τη συνάρτηση RANDOM να προσθέσουμε μια μονάδα μετασχηματίζοντας τις τιμές ως εξής:

0 + 1	μετασχηματίζεται σε	1
1 + 1	μετασχηματίζεται σε	2
2 + 1	μετασχηματίζεται σε	3
3 + 1	μετασχηματίζεται σε	4
4 + 1	μετασχηματίζεται σε	5
5 + 1	μετασχηματίζεται σε	6
6 + 1	μετασχηματίζεται σε	7
7 + 1	μετασχηματίζεται σε	8
8 + 1	μετασχηματίζεται σε	9
9 + 1	μετασχηματίζεται σε	10

```

TO ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN
REPEAT :PLHQOS_SYNOIKIWN
  [MAKE "ORIO_OROF RANDOM 10
  MAKE "ORIO_OROF :ORIO_OROF + 1
  ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF]
END
  
```



Ένα από τα περισσότερα σημαντικά καθήκοντα του προγραμματιστή είναι να εντοπίσει το λάθος και να το διορθώσει. Στον εντοπισμό του λάθους συνήθως μας βοηθάει ο υπολογιστής με διάφορα μηνύματα τα οποία μας πληροφορούν για το είδος του λάθους και το σημείο όπου κάτι δεν πήγε καλά. Η δουλειά του προγραμματιστή είναι να βρει την αιτία που δημιούργησε το λάθος και η οποία μπορεί να βρίσκεται όχι στο σημείο που παρουσιάστηκε η ανωμαλία αλλά κάπου αλλού.

Πόσες συνοικίες έχει αυτή η πόλη;



Κενές εκτάσεις μεταξύ συνοικιών

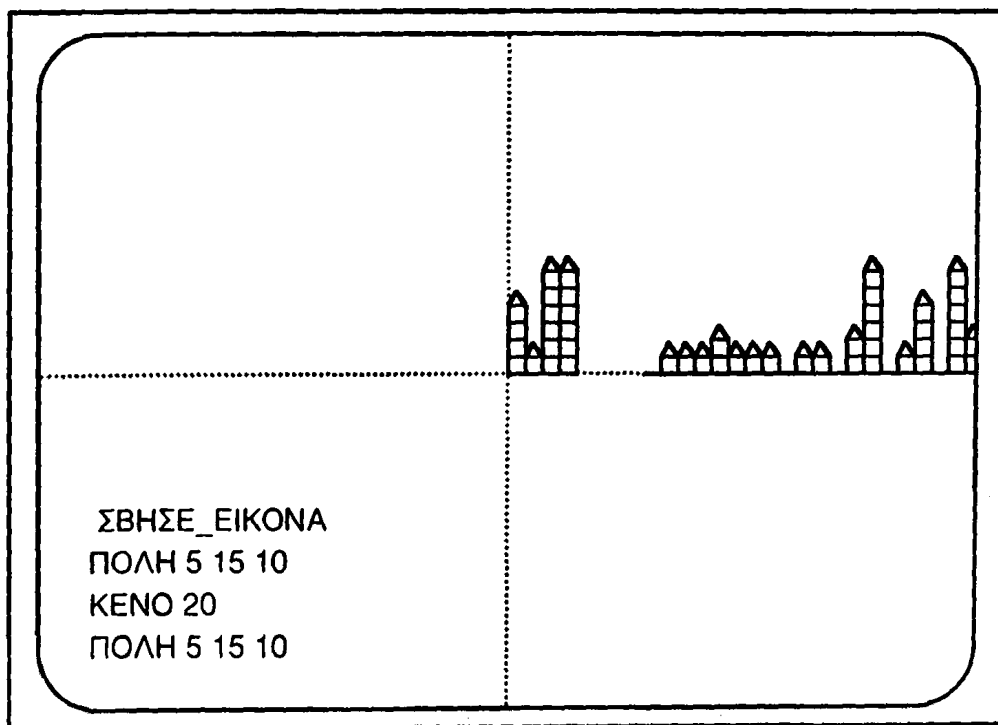
Στη διαδικασία της πόλης η μια συνοικία κολάει ακριβώς δίπλα στην άλλη με αποτέλεσμα να μην ξεχωρίζουν μεταξύ τους. Θα θέλαμε λοιπόν μεταξύ των συνοικιών να υπάρχει ένα κενός χώρος. Για το λόγο αυτό όταν τελειώνει μια συνοικία πρέπει η χελώνα να μετακινείται δεξιότερα χωρίς να αφήνει ίχνη. Αυτό γίνεται με την εντολή PenUp ή PU που σημαίνει σήκωσε την πένα (και μη γράφεις). Εμείς προσπαθώντας να εξελληνίσουμε το περιβάλλον της Logo θα ορίσουμε τη διαδικασία ΜΗ_ΓΡΑΦΕΙΣ. Η αντίθετη ενέργεια κατά την οποία η χελώνα κατεβάζει την πένα και μπορεί να αφήνει ίχνη γίνεται με την εντολή Pen Down ή PD και την εξελληνίζουμε με τη διαδικασία ΓΡΑΨΕ.

```
ΤΟ ΜΗ_ΓΡΑΦΕΙΣ
PU
END
```

```
ΤΟ ΓΡΑΨΕ
PD
END
```

Χρησιμοποιώντας αυτές τις δύο εντολές-διαδικασίες μαζί με τη διαδικασία ΚΙΝΗΣΗ_ΔΙΠΛΑ, μπορούμε να ορίσουμε τη διαδικασία ΚΕΝΟ που να μετακινεί δεξιότερα τη χελώνα χωρίς να αφήνει ίχνη στη διάρκεια της μετακίνησης.

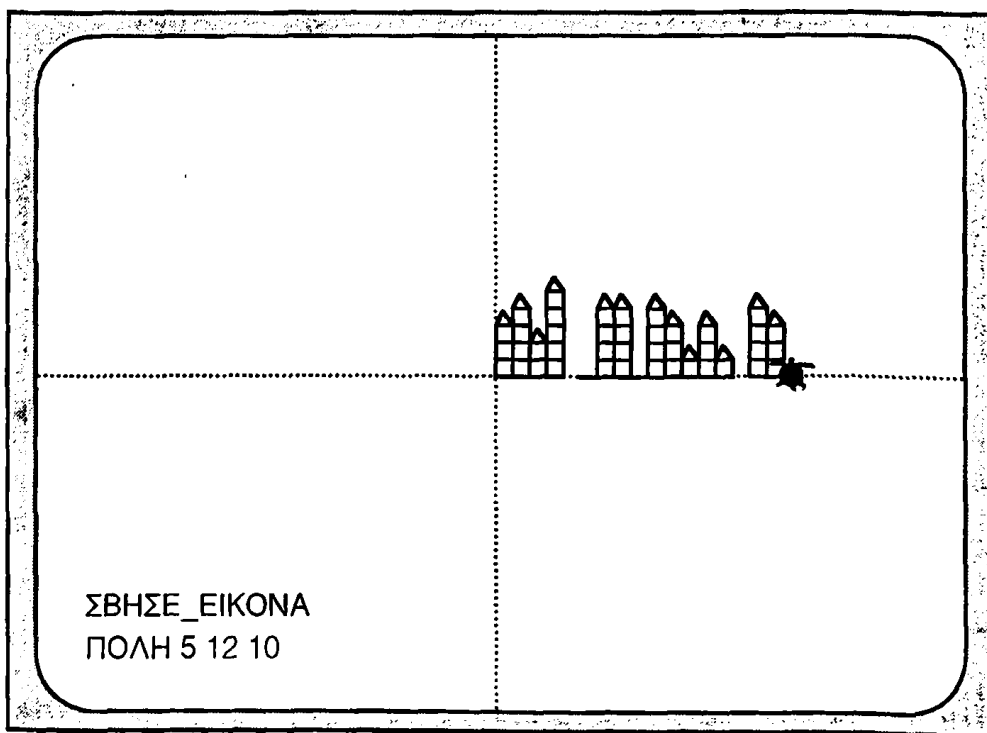
```
ΤΟ ΚΕΝΟ :ΑΡΟΤΑΨΗ
ΜΗ_ΓΡΑΦΕΙΣ
ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΑΡΟΤΑΨΗ
ΓΡΑΨΕ
END
```



Εμπρός για μια καλύτερη πόλη

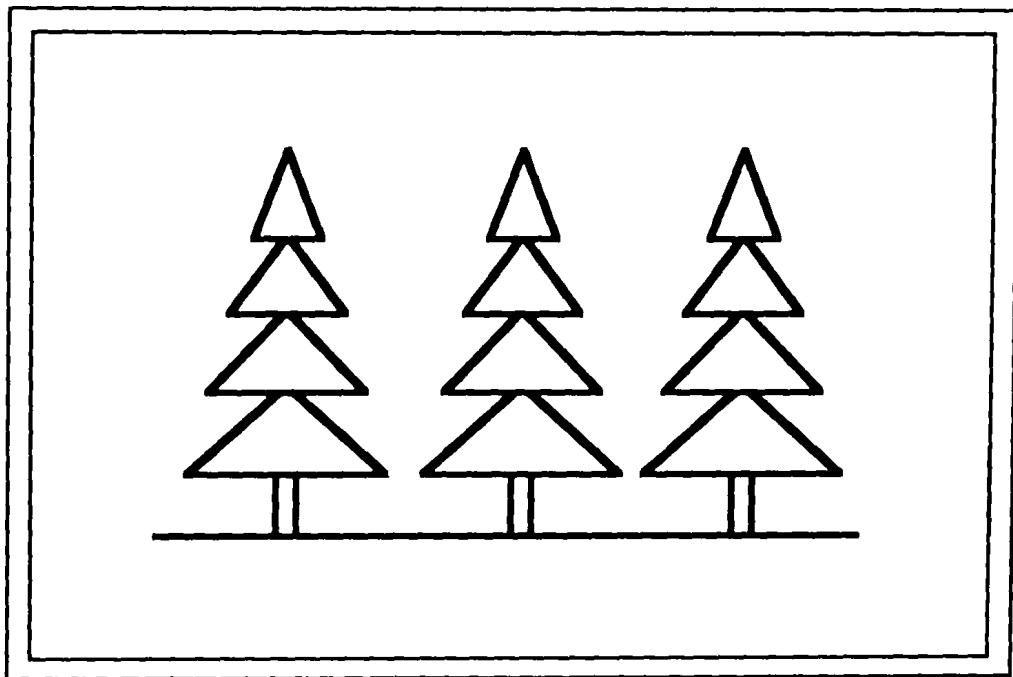
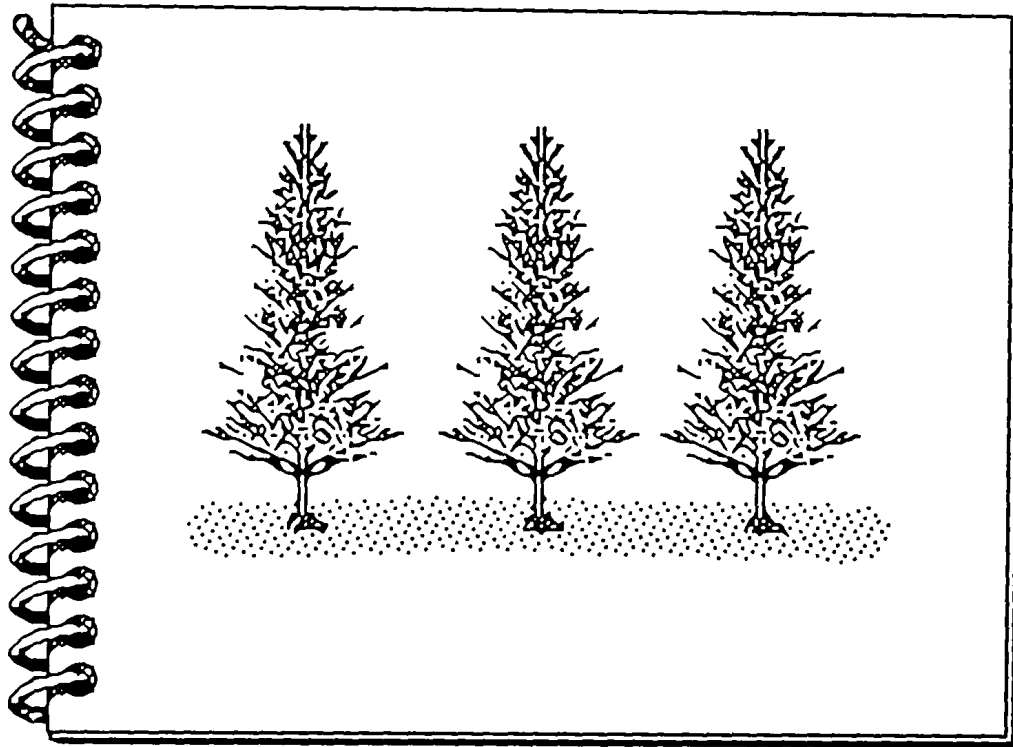
Στην οθόνη της προηγούμενης σελίδας ξεχωρίσαμε τις δύο πόλεις παρεμβάλλοντας ένα κενό χώρο μεταξύ τους. Το ίδιο μπορούμε να κάνουμε μεταξύ των συνοικιών μιας πόλης. Μπορούμε λοιπόν να ενσωματώσουμε στη διαδικασία ΠΟΛΗ τη διαδικασία ΚΕΝΟ μετά τη διαδικασία ΣΥΝΟΙΚΙΑ έτσι ώστε μετά το τέλος κάθε συνοικίας να αφήνεται λίγος κενός χώρος.

```
TO ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN  
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN  
REPEAT :PLHQOS_SYNOIKIWN  
  [MAKE "ORIO_OROF RANDOM 10  
  MAKE "ORIO_OROF :ORIO_OROF + 1  
  ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF  
  ΚΕΝΟ :YPSOS]  
END
```



Μετά το τέταρτο κτίριο μένει κενός χώρος και ακολουθεί μια γραμμή πριν το επόμενο κτίριο. Που διαφέρει (στον τρόπο που κατασκευάστηκε από το πρόγραμμα) η γραμμή από το κενό;





9 Εκδρομή στην Ελάτη

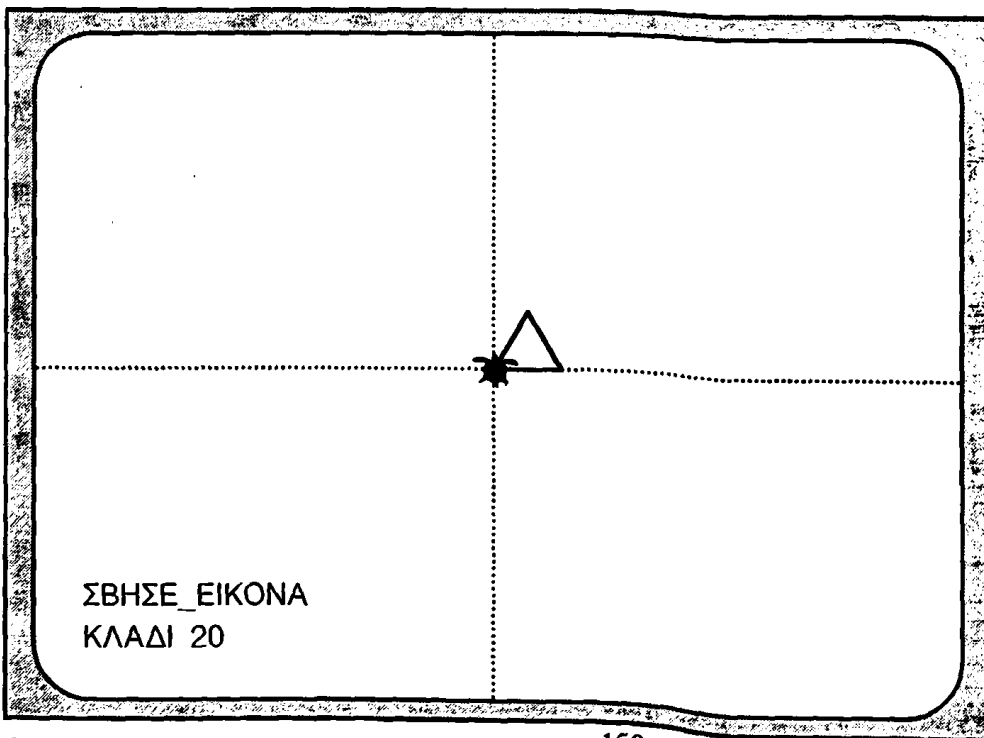
Το κλαδί

Ας αρχίσουμε να σχεδιάζουμε ένα έλατο. Το έλατο μπορεί να αναπαρασταθεί από τον κορμό και τα κλαδιά του. Κάθε κλαδί είναι ένα τρίγωνο. Γνωρίζοντας ότι το τρίγωνο είναι ένα πολύγωνο με τρεις πλευρές, μπορούμε να γράψουμε τη διαδικασία ΚΛΑΔΙ ως εξής:

```
ΤΟ ΚΛΑΔΙ :ΜΕΓΕΘΟΣ  
ΠΟΛΥΓΩΝΟ :ΜΕΓΕΘΟΣ 3  
END
```

Ας διατάξουμε τη χελώνα να σχεδιάσει ένα κλαδί.

Αργότερα, κάτω από το τρίγωνο-κλαδί θα βάλουμε τον κορμό. Αυτός θα πρέπει να βρίσκεται στη μέση της βάσης του τριγώνου. Μας διευκολύνει λοιπόν το τρίγωνο-κλαδί να είναι "κεντραρισμένο". Δηλαδή η χελώνα όταν τελειώνει τη σχεδίαση του κλαδιού να βρίσκεται στη μέση της βάσης του τριγώνου.



Το κλαδί στη σωστή θέση

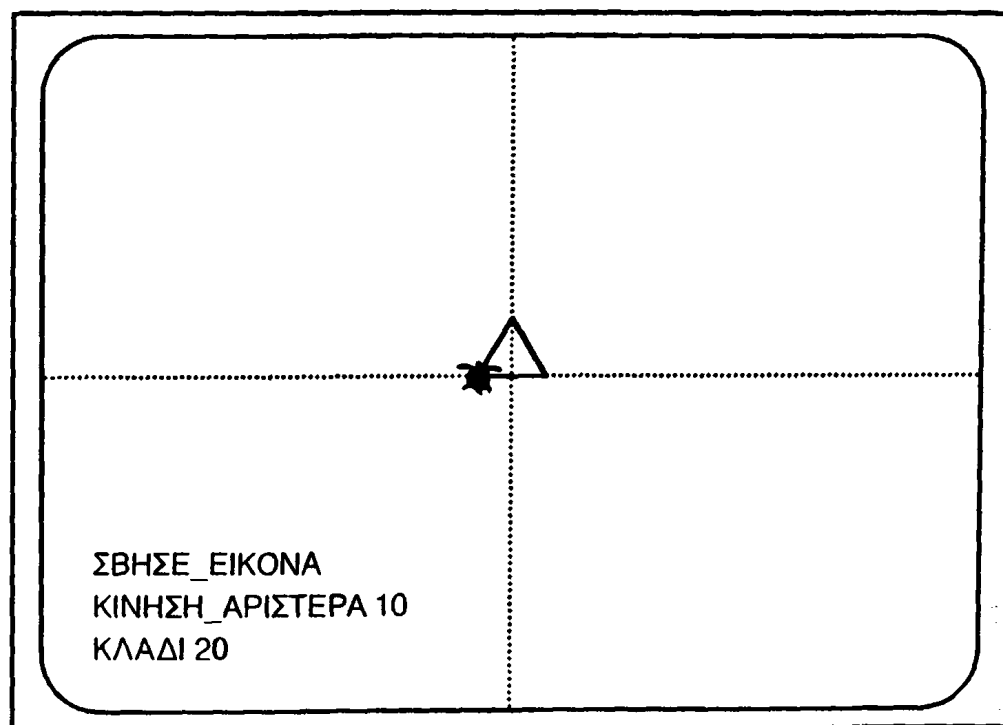
Για ευκολία ας προσπαθήσουμε να μεταφέρουμε το τρίγωνο έτσι ώστε η κορυφή του να περνάει από την ίδια κατακόρυφη η οποία θα περνάει από τη μέση της βάσης του.

Αυτό το πετυχαίνουμε αν πριν ξεκινήσει η σχεδίαση του τριγώνου έχουμε μετατοπίσει (οριζόντια) προς τα αριστερά τη χελώνα σε απόσταση ίση με το μισό της πλευράς του τριγώνου.

Αυτήν την κίνηση της χελώνας προς τα αριστερά θα την κάνουμε με τη διαδικασία ΚΙΝΗΣΗ_ΑΡΙΣΤΕΡΑ.

```
ΤΟ ΚΙΝΗΣΗ_ΑΡΙΣΤΕΡΑ :ΑΡΟΤΑΐΗ  
ΜΗ_ΓΡΑΦΕΙΣ  
ΑΡΙΣΤΕΡΑ 90  
ΕΥΘΕΙΑ :ΑΡΟΤΑΐΗ  
ΔΕΞΙΑ 90  
ΓΡΑΨΕ  
ΕΝΔ
```

Αφού ορίσθηκε η διαδικασία ΚΙΝΗΣΗ_ΑΡΙΣΤΕΡΑ ας τη χρησιμοποιήσουμε για να πετύχουμε τη μετατόπιση του κλαδιού προς τα αριστερά. Δεν πρέπει να ξεχάσουμε ότι η μετακίνηση αυτή θα γίνει για απόσταση μισή από το μήκος της πλευράς του τριγώνου-κλαδιού.



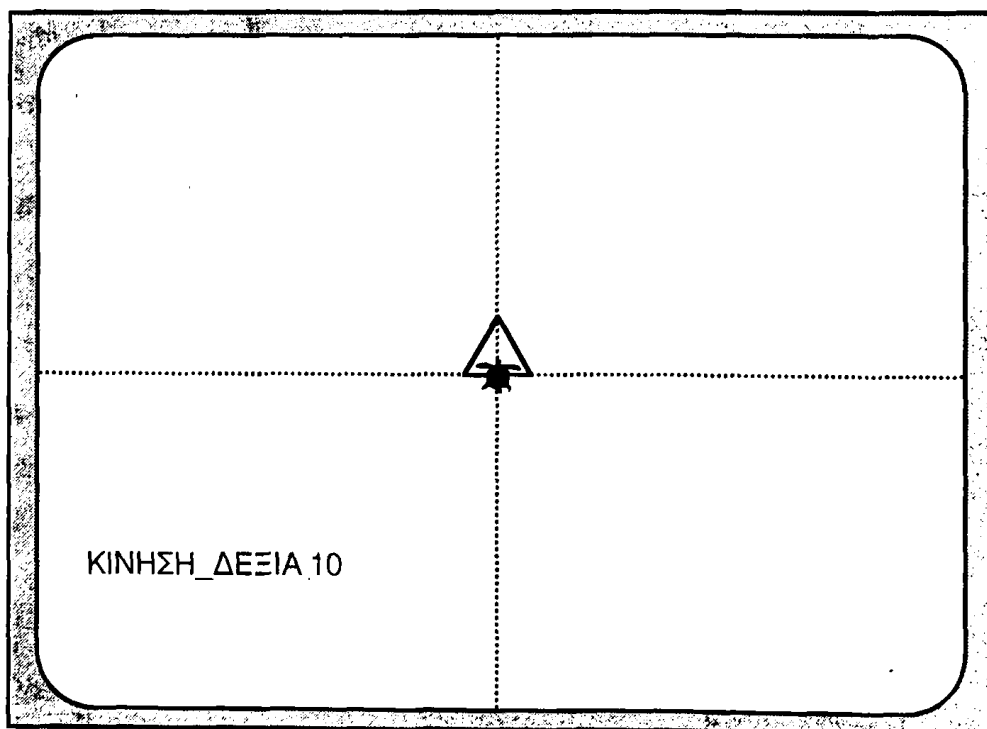
Επιστροφή στην αρχική κατάσταση

Από τα προηγούμενα διαπιστώνουμε ότι ναι μεν το κλαδί- τρίγωνο μετατοπίσθηκε στη θέση που θέλαμε, η χελώνα όμως δε βρίσκεται στο σημείο από το οποίο ξεκίνησε. Θα πρέπει λοιπόν η χελώνα να μετακινηθεί ξανά προς τα δεξιά.

Αυτό γίνεται με τη διαδικασία ΚΙΝΗΣΗ_ΔΕΞΙΑ.

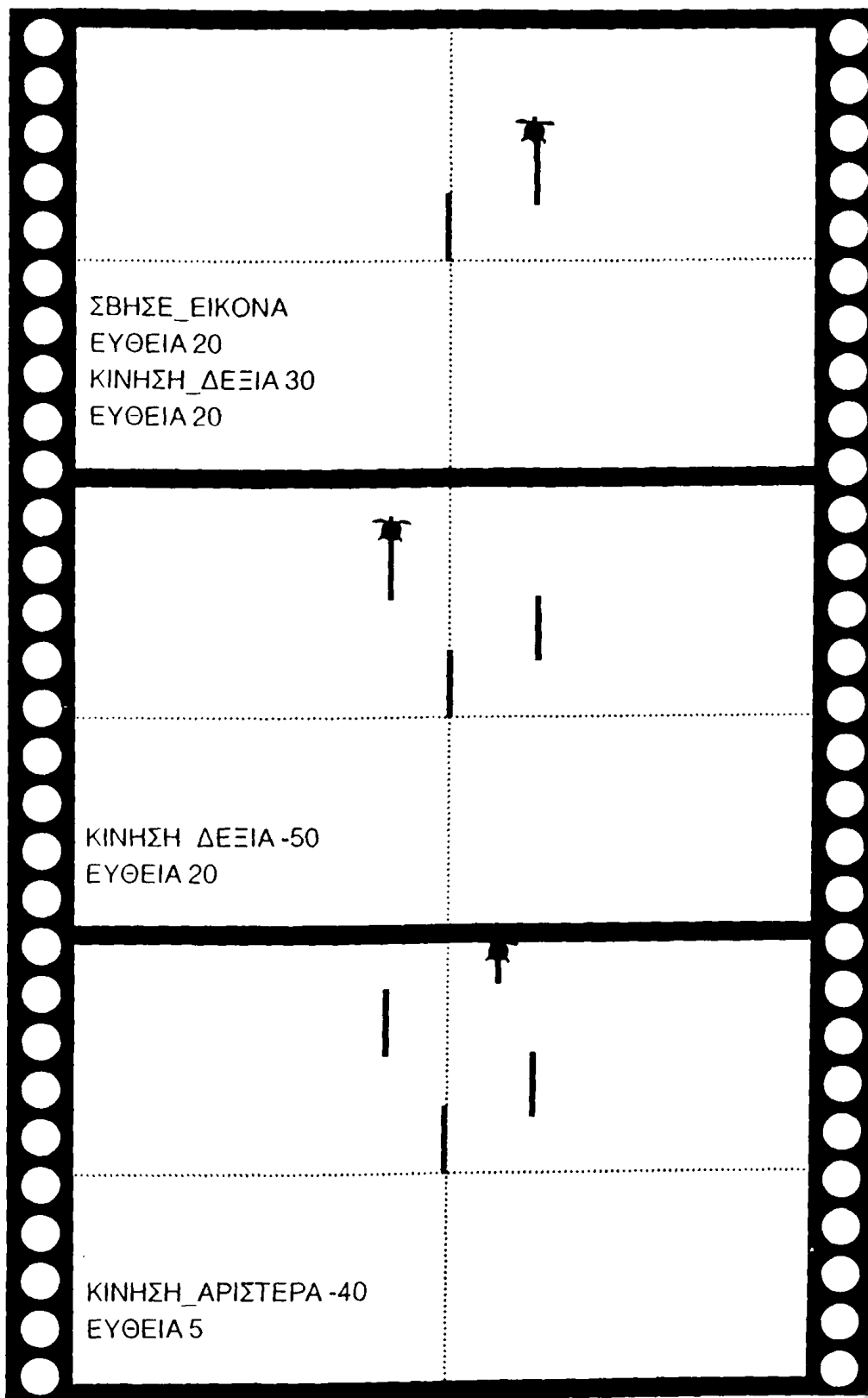
```
TO ΚΙΝΗΣΗ_ΔΕΞΙΑ :ΑΡΟΣΤΑΣΗ
ΜΗ_ΓΡΑΦΕΙΣ
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :ΑΡΟΣΤΑΣΗ
ΑΡΙΣΤΕΡΑ 90
ΓΡΑΨΕ
ΕΝΔ
```

Αφού ορίσθηκε η διαδικασία ΚΙΝΗΣΗ_ΔΕΞΙΑ ως τη χρησιμοποιήσουμε για να πετύχουμε τη μετατόπιση της χελώνας στην αφετηρία. Η μετατόπιση προς την αφετηρία είναι ίση με την αρχική απόσταση που διένυσε η χελώνα όταν μετατοπίσθηκε προς τα αριστερά.



Δεξιά μετακίνηση με αρνητική φορά

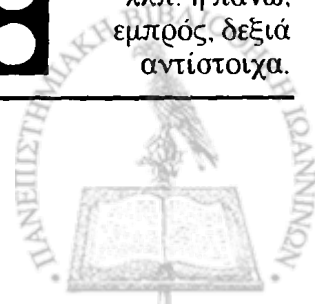
Μπορούμε με τη διαδικασία ΚΙΝΗΣΗ_ΔΕΞΙΑ να κινηθούμε προς τα αριστερά αν χρησιμοποιήσουμε αρνητική τιμή.



Οι αριθμοί που μετράνε ύψος και βάθος, εμπρός και πίσω, δεξιά και αριστερά, ακόμα και όταν είναι ίδιοι, αναφέρονται σε διαφορετικά, αντίθετα μεταξύ τους πράγματα. Τα μεγέθη δηλαδή μεταβάλλονται συνήθως κατά δύο αντίθετες φορές. Για την έκφραση τέτοιων εννοιών με αριθμούς ή άλλα σύμβολα και όχι με λέξεις, χρησιμοποιούνται τα γνωστά από τις πράξεις της πρόσθεσης και της αφαίρεσης σύμβολα συν (+) και πλην (-).

Το πλην λέγεται αρνητικό πρόσημο και το συν θετικό.

Τα πρόσημα μπαίνουν μπροστά από τους αριθμούς και συμβολίζουν τις λέξεις κάτω, πίσω, αριστερά κλπ. ή πάνω, εμπρός, δεξιά αντίστοιχα.



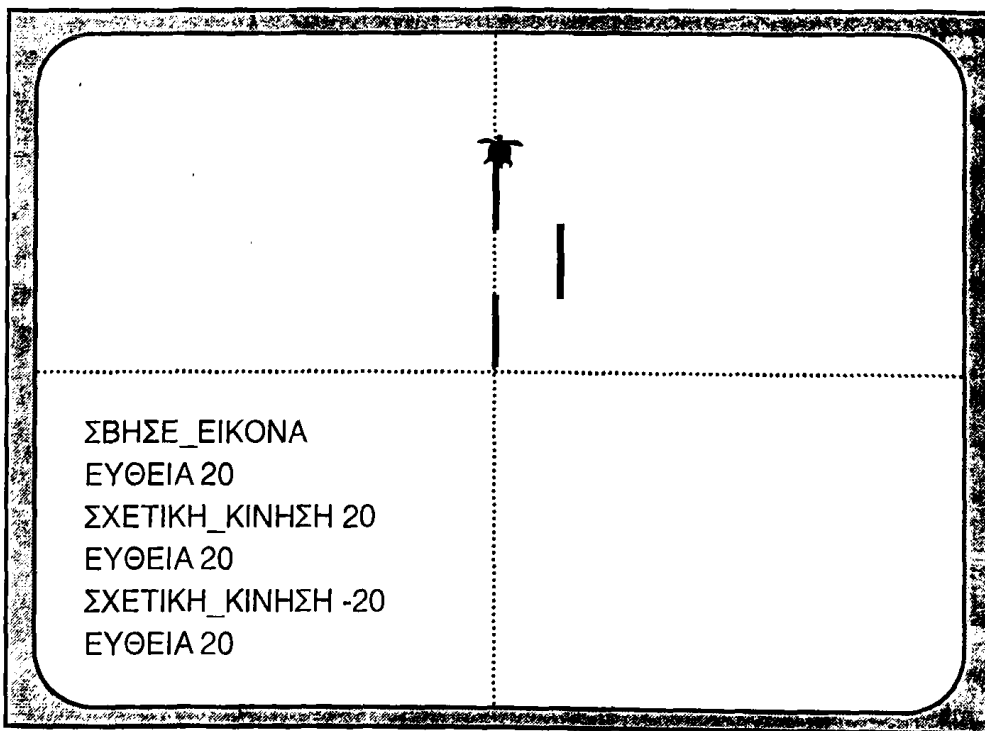
Σχετική οριζόντια μετακίνηση

Διαπιστώνουμε ότι δεν είναι απαραίτητο να έχουμε δύο διαδικασίες για αριστερή και δεξιά μετακίνηση. Αρκεί μια, η ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ, στην οποία αν η τιμή που εισάγεται στη μεταβλητή ΑΡΟΣΤΑΣΗ είναι θετική κινεί τη χελώνα προς τα δεξιά ενώ αν είναι αρνητική τη μετακινεί προς τα αριστερά.

Η διαδικασία αυτή είναι η:

```
ΤΟ ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ :ΑΡΟΣΤΑΣΗ
ΜΗ_ΓΡΑΦΕΙΣ
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :ΑΡΟΣΤΑΣΗ
ΑΡΙΣΤΕΡΑ 90
ΓΡΑΨΕ
ΕΝΔ
```

Ας πειραματιστούμε με τη διαδικασία ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ για να ελέγξουμε τη σωστή λειτουργία της.



Το κλαδί ως δομικό στοιχείο του ελάτου

Έχοντας στη διαθεσή μας τη διαδικασία ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ μπορούμε να βελτιώσουμε τη διαδικασία ΚΛΑΔΙ έτσι ώστε το τρίγωνο που θα σχηματίζει να είναι "κεντραρισμένο".

Η διαδικασία ΚΛΑΔΙ θα πρέπει

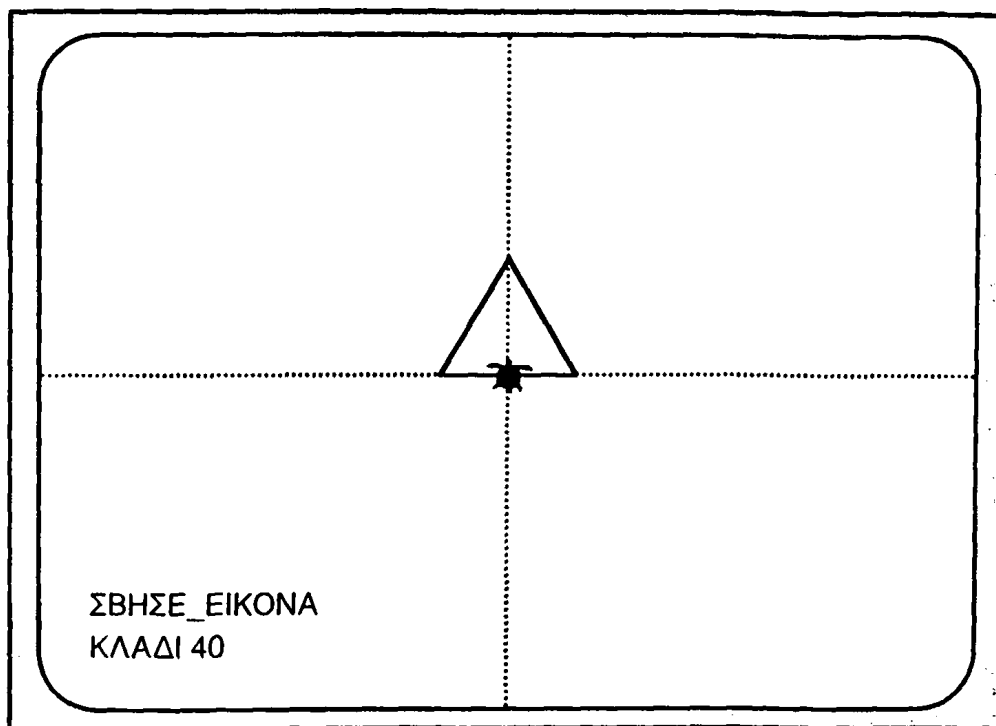
αρχικά να μετατοπίζει τη χελώνα προς τα αριστερά για απόσταση ίση με το μισό της βάσης του τριγώνου να σχηματίζει το κλαδί-τρίγωνο να επαναφέρει τη χελώνα στην αρχική της θέση (στο μέσο της βάσης του τριγώνου) μετακινώντας την προς τα δεξιά.

Φραστική
μέθοδος
περιγραφής του
αλγορίθμου

Ας δούμε πως θα διαμορφωθεί η διαδικασία ΚΛΑΔΙ.

```
ΤΟ ΚΛΑΔΙ :MEGEQOS  
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:MEGEQOS / 2)  
ΠΟΛΥΓΩΝΟ :MEGEQOS 3  
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:MEGEQOS / 2)  
END
```

Η έκφραση MINUS (:MEGEQOS/2) αντιστοιχεί στην -:MEGEQOS/2



Από που θα ξεκινήσει το επόμενο κλαδί;

Για να σχεδιάσει το παραπάνω κλαδί η χελώνα θα πρέπει να ανέβει στην επάνω κορυφή του τριγώνου. Πόσο να μετακινηθεί;

The image consists of three vertically stacked panels, each enclosed in a thick black border with a row of white circles on the left and right sides. A vertical dotted line runs through the center of all three panels. A horizontal dotted line is drawn across each panel, intersecting the vertical one. In each panel, a triangle is drawn with its base on the horizontal dotted line. A small black turtle icon is positioned at the top vertex of the triangle in the top panel, at the bottom vertex in the middle panel, and at the top vertex in the bottom panel.

ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΚΛΑΔΙ 40
ΜΗ_ΓΡΑΦΕΙΣ
ΕΥΘΕΙΑ 40

ΟΠΙΣΘΕΝ 10

ΕΥΘΕΙΑ 40
ΓΡΑΨΕ

Υπολογισμός με το Πυθαγόρειο θεώρημα

Κάναμε προηγουμένως μερικές προσπάθειες για να βρούμε πόσο διάστημα πρέπει να μετακινήσουμε τη χελώνα από τη βάση ως την κορυφή του κλαδιού. Το αποτέλεσμα όμως που βρήκαμε ισχύει μόνο για το συγκεκριμένο μέγεθος κλαδιού. Δεν είναι δυνατό να βρούμε εμπειρικά τρόπο που να προσδιορίζει το διάστημα αυτό ώστε να μπορούμε να γενικεύσουμε τον υπολογισμό.

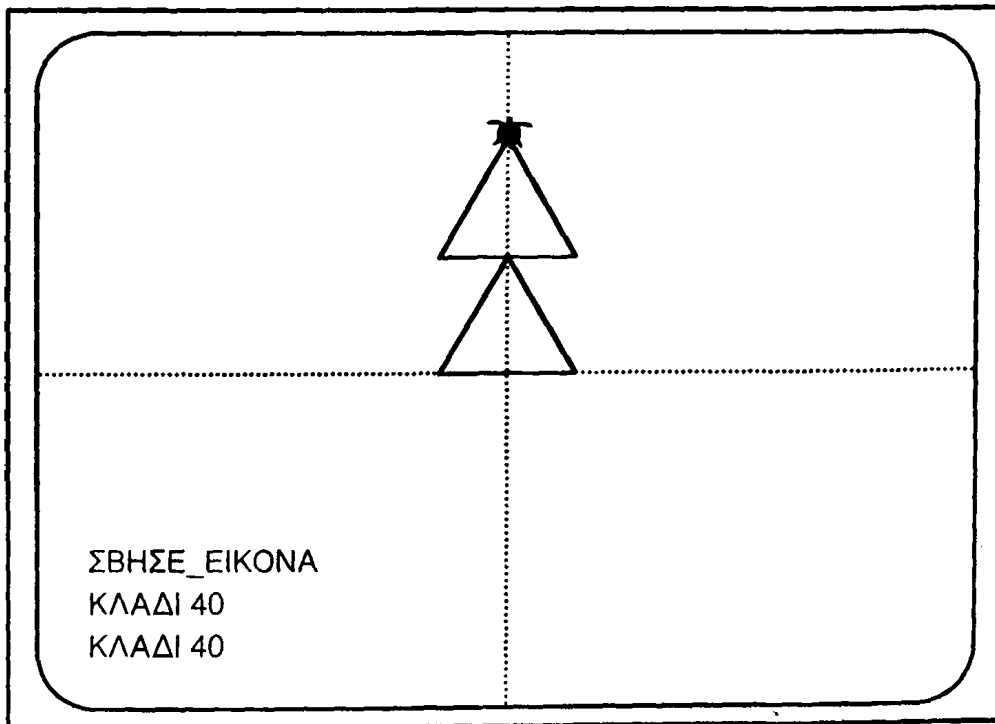
Η γενίκευση μπορεί να γίνει αν εφαρμόσουμε το πυθαγόρειο θεώρημα στο ορθογώνιο τρίγωνο που σχηματίζεται από το ύψος του ισοπλεύρου τριγώνου που φέρεται από την επάνω κορυφή. Εφαρμόζοντας το πυθαγόρειο θεώρημα, καταλήγουμε στον τύπο που υπολογίζει το ύψος του ισοπλεύρου τριγώνου:

$$\text{ΥΨΟΣ} = \frac{\sqrt{3} \times \text{ΜΗΚΟΣ ΠΛΕΥΡΑΣ}}{2}$$

Η τετραγωνική ρίζα υπολογίζεται από τη συνάρτηση SQRT.

```

ΤΟ ΚΛΑΔΙ :ΜΕΓΕΘΟΣ
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:ΜΕΓΕΘΟΣ / 2)
ΠΟΛΥΓΩΝΟ :ΜΕΓΕΘΟΣ 3
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:ΜΕΓΕΘΟΣ / 2)
ΜΗ_ΓΡΑΦΕΙΣ
ΜΑΚΕ "ΥΨΟΣ (SQRT(3) * :ΜΕΓΕΘΟΣ) / 2
ΕΥΘΕΙΑ :ΥΨΟΣ
ΓΡΑΨΕ
ΕΝΔ
    
```



Αναγκαιότητα για ακριβείς υπολογισμούς και γενίκευση του προγράμματος καθιστά απαραίτητο για το μαθητή να ξέρει να "χειρίζεται" το Πυθαγόρειο θεώρημα. Μια συνάρτηση, για όσους δεν ενδιαφέρονται για τα μαθηματικά, μπορεί να θεωρηθεί ως ένα μαύρο κουτί με δύο πόρτες. Από την πόρτα εισόδου μπαίνουν τα δεδομένα, επεξεργάζονται με τον κατάλληλο τρόπο που υπαγορεύει ο μηχανισμός της συνάρτησης και τα αποτελέσματα βγαίνουν από την πόρτα εξόδου. Π.χ. η συνάρτηση SQRT παίρνει ως είσοδο το 9 και αφού υπολογίσει την τετραγωνική του ρίζα εξάγει ως αποτέλεσμα το 3.

9-> SQRT ->3

Μια συνάρτηση είναι ένας νόμος που συνδέει δύο σύνολα X και Y και αντιστοιχίζει σε κάθε στοιχείο χ του συνόλου X ένα και μόνο ένα στοιχείο ψ του συνόλου Y.



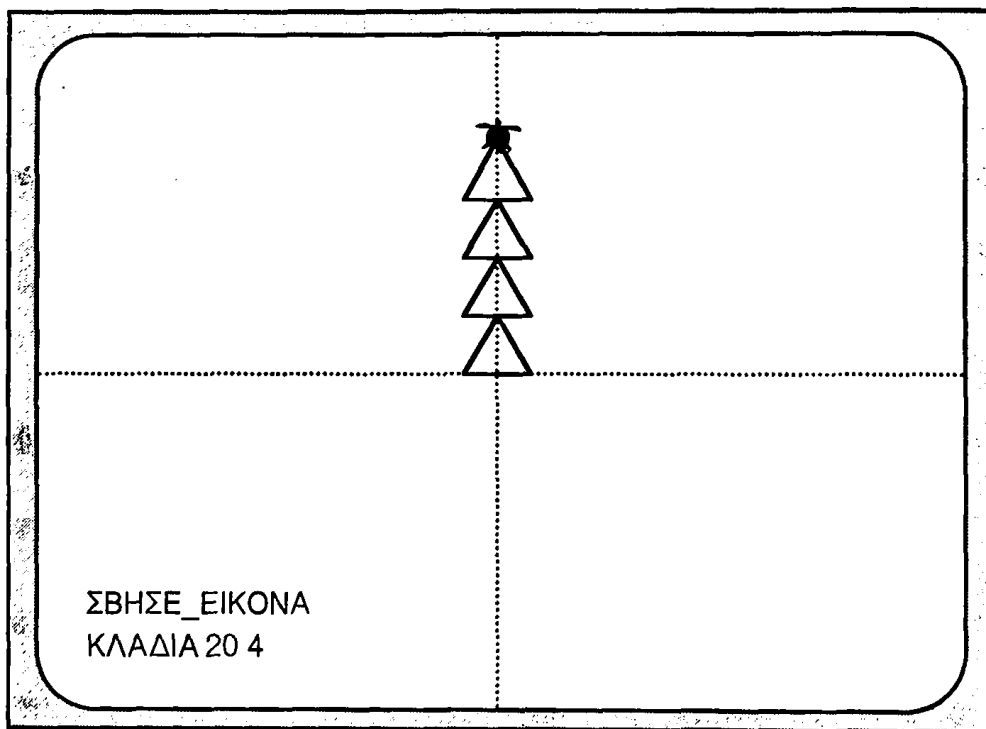
Όλα τα κλαδιά του ελάτου

Αφού μάθαμε τη χελώνα να σχεδιάζει ένα κλαδί και να πηγαίνει στην κορυφή του ώστε να είναι έτοιμη να σχεδιάσει το επόμενο πιο πάνω μπορούμε να επαναλάβουμε αυτήν τη διαδικασία ώστε να σχηματισθούν όλα τα κλαδιά του ελάτου. Αυτό γίνεται με τη διαδικασία ΚΛΑΔΙΑ.

```
TO ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ
REPEAT :ΠΛΗΘΟΣ
  [ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:ΜΕΓΕΘΟΣ / 2)
  ΠΟΛΥΓΩΝΟ :ΜΕΓΕΘΟΣ 3
  ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:ΜΕΓΕΘΟΣ / 2)
  ΜΗ_ΓΡΑΦΕΙΣ
  MAKE "ΥΡΣΟΣ (:ΜΕΓΕΘΟΣ * (SQRT(3)) / 2)
  ΕΥΘΕΙΑ :ΥΡΣΟΣ
  ΓΡΑΨΕ]
END
```

Παρατηρούμε ότι το πακέτο των εντολών που βρίσκονται μέσα στο REPEAT είναι το σύνολο των εντολών της διαδικασίας ΚΛΑΔΙ. Άρα μπορούμε να γράψουμε πιο κομπά τη διαδικασία ΚΛΑΔΙΑ χρησιμοποιώντας τη διαδικασία ΚΛΑΔΙ ως εξής:

```
TO ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ
REPEAT :ΠΛΗΘΟΣ
  [ΚΛΑΔΙ :ΜΕΓΕΘΟΣ]
END
```



Ο κορμός του ελάτου

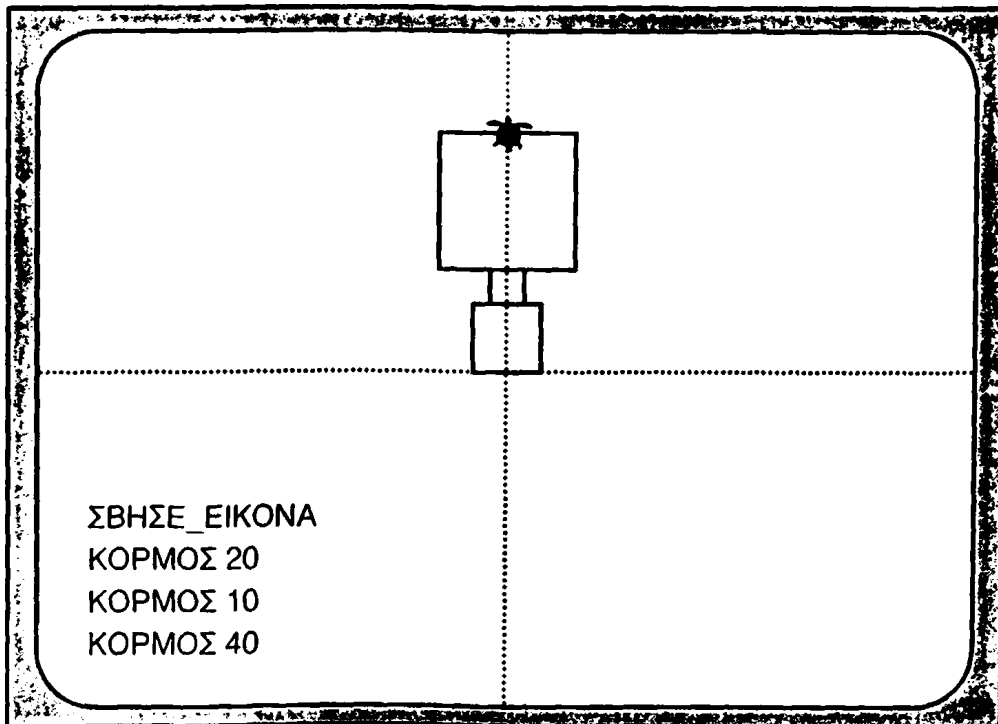
Εκτός από τα κλαδιά για να σχεδιάσουμε το έλατο χρειάζεται και ο κορμός του που θα τον φτιάξουμε με ένα τετράγωνο.

Εκείνο που έχει σημασία είναι η θέση του τετραγώνου σε σχέση με τη θέση της χελώνας για να ταιριάζει ο κορμός με τα κλαδιά.

Θα εκμεταλλευθούμε την εμπειρία που αποκτήσαμε σχεδιάζοντας το κλαδί-τρίγωνο και θα τη χρησιμοποιήσουμε για να τοποθετήσουμε σωστά τον κορμό.

```
ΤΟ ΚΟΡΜΟΣ :Y
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:Y / 2)
ΠΟΛΥΓΩΝΟ :Y 4
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:Y / 2)
ΜΗ_ΓΡΑΦΕΙΣ
ΕΥΘΕΙΑ :Y
ΓΡΑΨΕ
END
```

Συγκρίνετε τις διαδικασίες ΚΟΡΜΟΣ και ΚΛΑΔΙ, αναφέρεται τις ομοιότητες και τις διαφορές. Που οφείλονται;



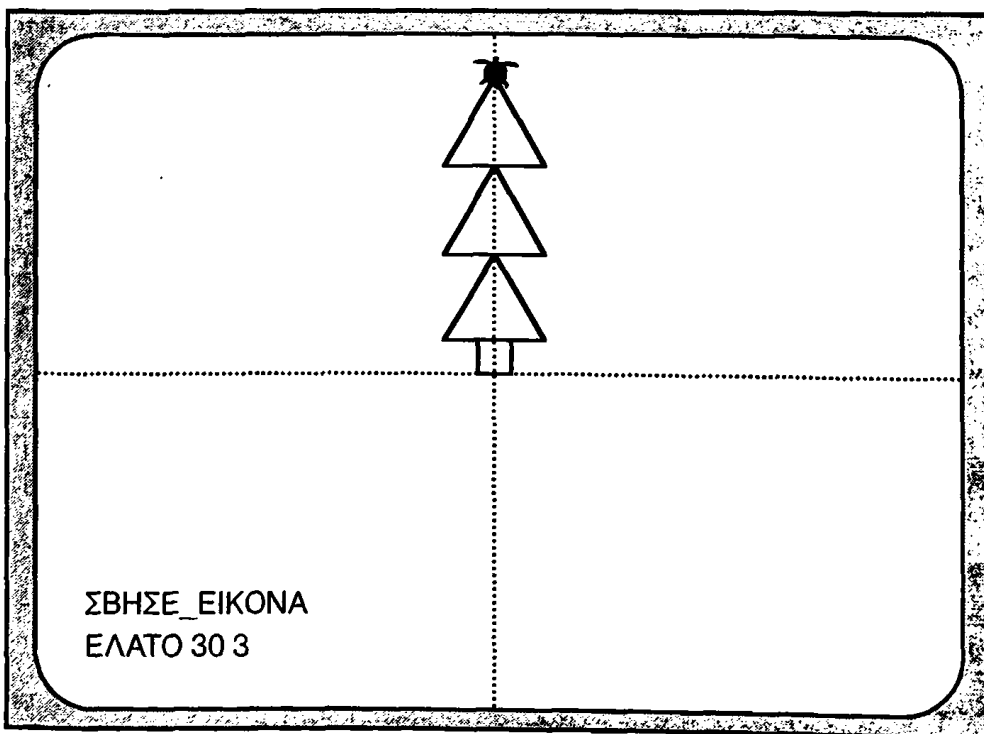
Το έλατο!

Διαθέτοντας τις διαδικασίες ΚΟΡΜΟΣ και ΚΛΑΔΙΑ η σχεδίαση του ελάτου είναι πλέον απλή υπόθεση.

Θα πρέπει μόνο να λάβουμε υπόψη μας ότι το μέγεθος του κορμού πρέπει να αναλογεί με τις διαστάσεις των κλαδιών. Κάνοντας μερικές δοκιμές καταλήγουμε ότι το έλατο θα φαίνεται φυσικότερο αν το μέγεθος του κορμού του είναι το $1/3$ του μεγέθους των πλευρών των κλαδιών-τριγώνων.

Ας δούμε τη διαδικασία ΕΛΑΤΟ:

```
ΤΟ ΕΛΑΤΟ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ  
ΚΟΡΜΟΣ (:ΜΕΓΕΘΟΣ / 3)  
ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ  
END
```



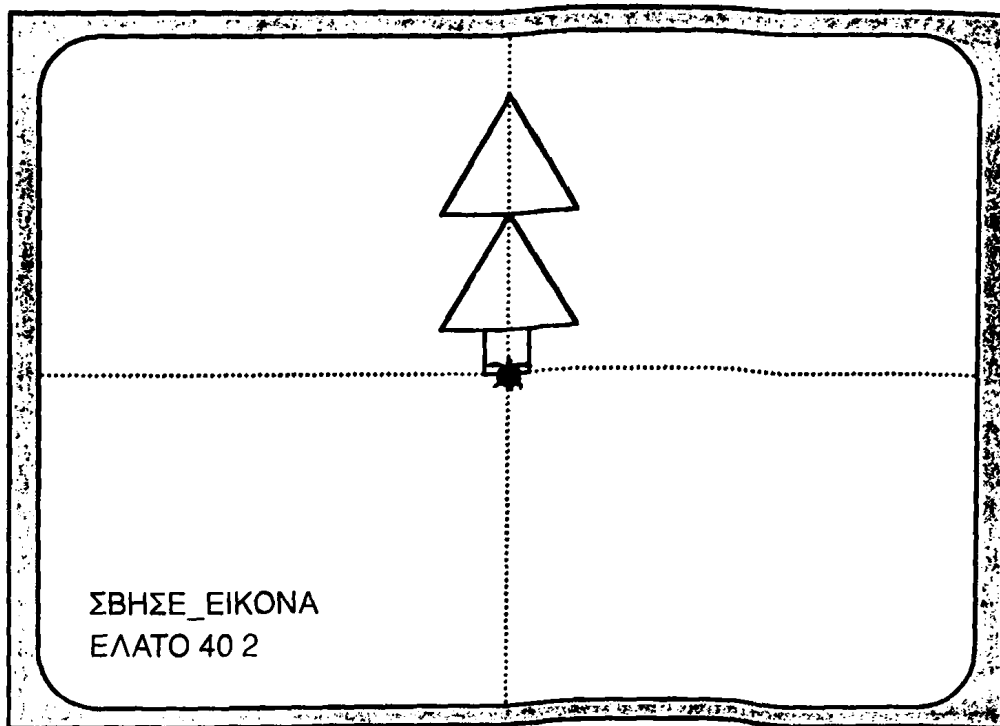
Επιστροφή στις ρίζες

Η διαδικασία ΕΛΑΤΟ όταν ολοκληρώσει τη σχεδίαση, αφήνει τη χελώνα στην κορυφή του. Είναι καλύτερα η χελώνα να επιστρέφει στη θέση από όπου ξεκίνησε.

Αυτό γίνεται βάζοντας τη χελώνα να οπισθοχωρήσει για απόσταση όση είναι το άθροισμα των υψών των κλαδιών συν το ύψος του κορμού.

Έτσι η διαδικασία ΕΛΑΤΟ βελτιώνεται ως εξής:

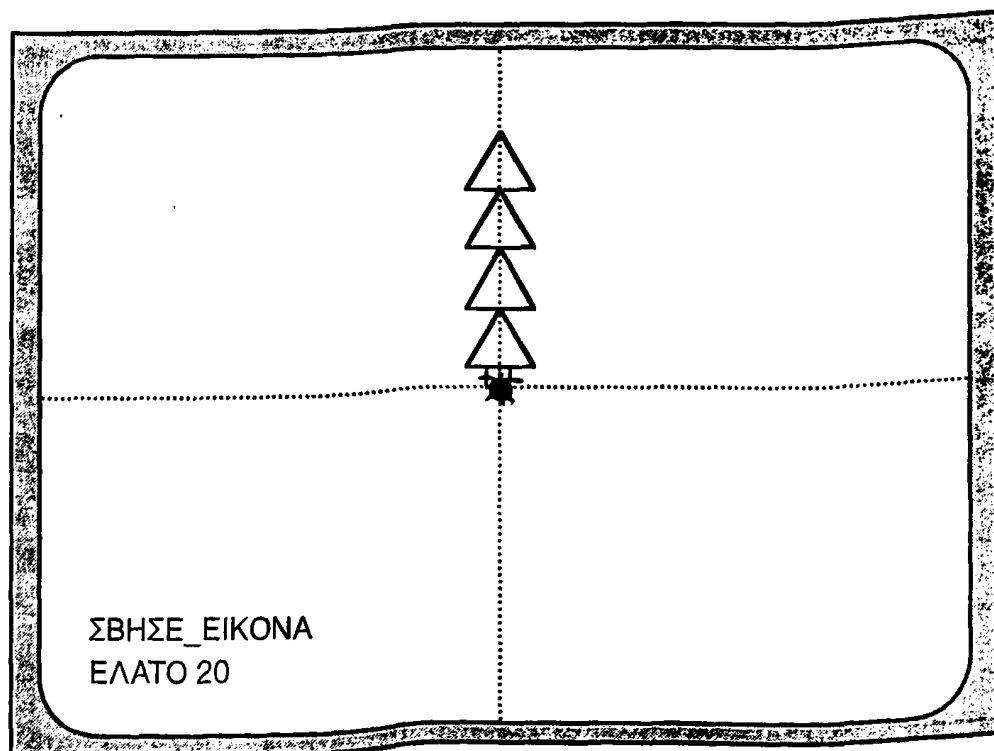
```
ΤΟ ΕΛΑΤΟ :ΜΕΓΕΘΟΣ :ΡΛΗΘΟΣ
ΚΟΡΜΟΣ (:ΜΕΓΕΘΟΣ / 3)
ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΡΛΗΘΟΣ
ΜΗ_ΓΡΑΦΕΙΣ
ΜΑΚΕ "ΥΡΣΟΣ (:ΜΕΓΕΘΟΣ * (SQRT(3)) / 2)
ΡΕΡΕΑΤ :ΡΛΗΘΟΣ
    [ΟΠΙΣΘΕΝ :ΥΡΣΟΣ]
ΟΠΙΣΘΕΝ (:ΜΕΓΕΘΟΣ / 3)
ΓΡΑΨΕ
ΕΝΔ
```



Η ηλικία του ελάτου

Το πλήθος των κλαδιών του ελάτου μας δείχνει την ηλικία του δένδρου. Θα χρησιμοποιήσουμε τη συνάρτηση RANDOM ώστε να έχουμε κάθε φορά τυχαίο πλήθος κλαδιών.

```
ΤΟ ΕΛΑΤΟ :MEGEQOS  
MAKE "PLHQOS RANDOM 5  
ΚΟΡΜΟΣ (:MEGEQOS / 3)  
ΚΛΑΔΙΑ :MEGEQOS :PLHQOS  
ΜΗ_ΓΡΑΦΕΙΣ  
MAKE "ΥΡΣΟΣ (:MEGEQOS * (SQRT(3)) / 2)  
REPEAT :PLHQOS  
  [ΟΠΙΣΘΕΝ :ΥΡΣΟΣ]  
ΟΠΙΣΘΕΝ (:MEGEQOS / 3)  
ΓΡΑΨΕ  
END
```

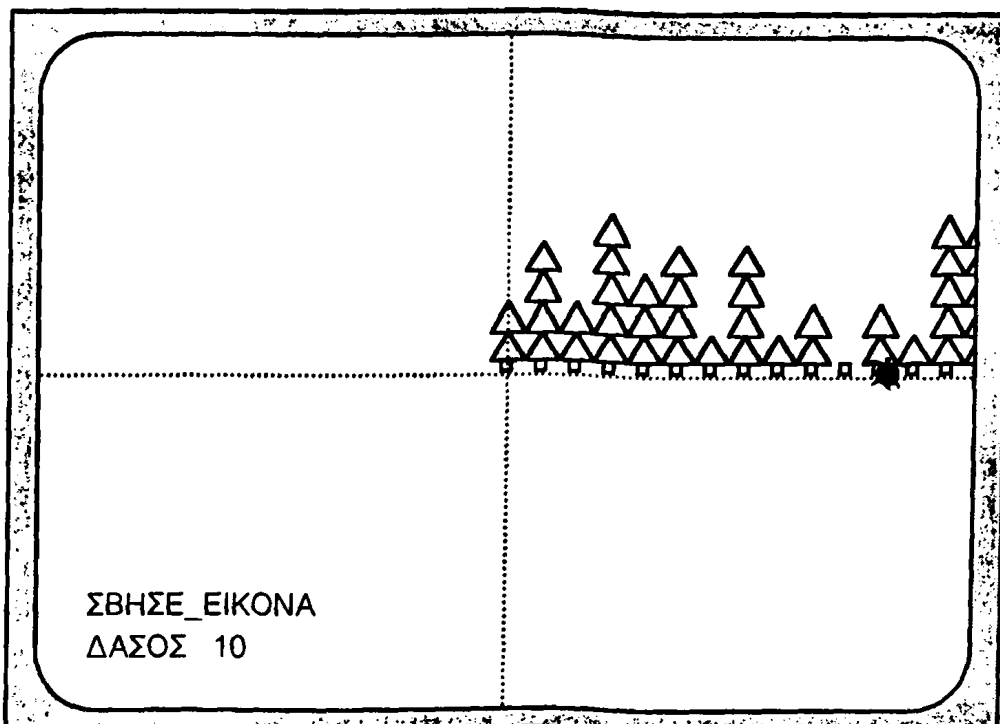


Ένα δάσος με έλατα

Πολύ εύκολα πλέον μπορούμε επαναλαμβάνοντας μερικές φορές τη διαδικασία ΕΛΑΤΟ να φτιάξουμε ένα δάσος με έλατα.

```
TO ΔΑΣΟΣ :MEGEQOS
MAKE "ΕΛΑΤΑ RANDOM 15
REPEAT :ΕΛΑΤΑ
  [ΕΛΑΤΟ :MEGEQOS
  ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ :MEGEQOS]
END
```

Σε τι χρειάζεται
η διαδικασία
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ;
Γιατί βρίσκεται
μέσα στις
αγκύλες της
εντολής
REPEAT;



Τι συνέβη στο
11ο έλατο;
Ποιός το έκοψε;
Πως έγινε αυτό;

Το έλατο που δείχνει την ηλικία του

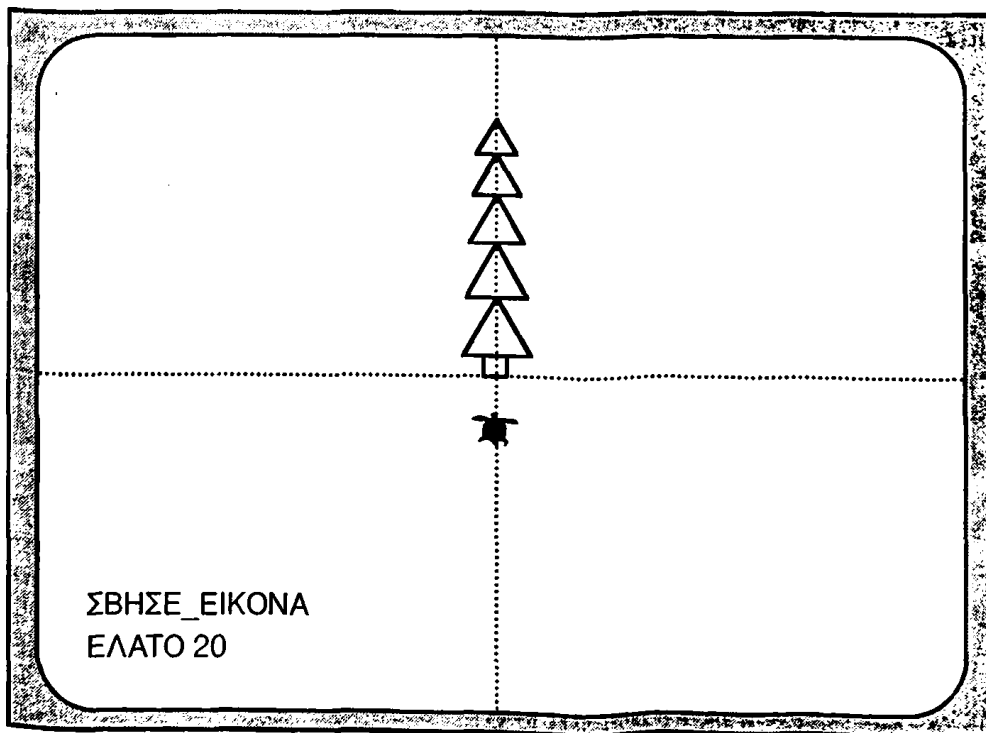
Τα κλαδιά που βρίσκονται προς τη βάση του ελάτου είναι τα πιο παλιά και συνεπώς τα πιο μεγάλα ενώ όσο ανεβαίνουμε προς την κορυφή τα κλαδιά να γίνονται μικρότερα.

Θα πρέπει λοιπόν το μέγεθος κάθε επόμενου κλαδιού να είναι μικρότερο και έστω ότι αυτή η μείωση είναι 2 βήματα χελώνας.

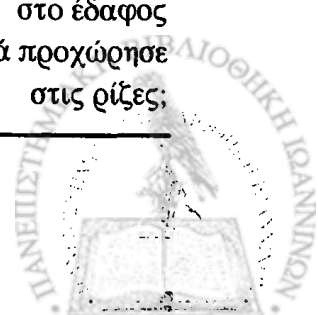
Η διαδικασία ΚΛΑΔΙΑ θα τροποποιηθεί ως εξής:

```
TO ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ
REPEAT :ΠΛΗΘΟΣ
  [ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:ΜΕΓΕΘΟΣ / 2)
  ΠΟΛΥΓΩΝΟ :ΜΕΓΕΘΟΣ 3
  ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:ΜΕΓΕΘΟΣ / 2)
  ΜΗ_ΓΡΑΦΕΙΣ
  MAKE "ΥΨΟΣ (:ΜΕΓΕΘΟΣ * (SQRT(3)) / 2)
  ΕΥΘΕΙΑ :ΥΨΟΣ
  ΓΡΑΨΕ
  MAKE "ΜΕΓΕΘΟΣ :ΜΕΓΕΘΟΣ - 2]
END
```

Τι θα συμβεί αν με τη μείωση αυτή η τιμή της μεταβλητής ΜΕΓΕΘΟΣ γίνει αρνητική; Προτείνετε εναλλακτικούς τρόπους για να γίνεται η μείωση ώστε να μην υπάρξει περίπτωση αρνητικής τιμής για τη μεταβλητή ΜΕΓΕΘΟΣ.



Γιατί η χελώνα
δεν σταμάτησε
στο έδαφος
αλλά προχώρησε
στις ρίζες;



Η χελώνα έπαψε να κυκλοφορεί υπόγεια

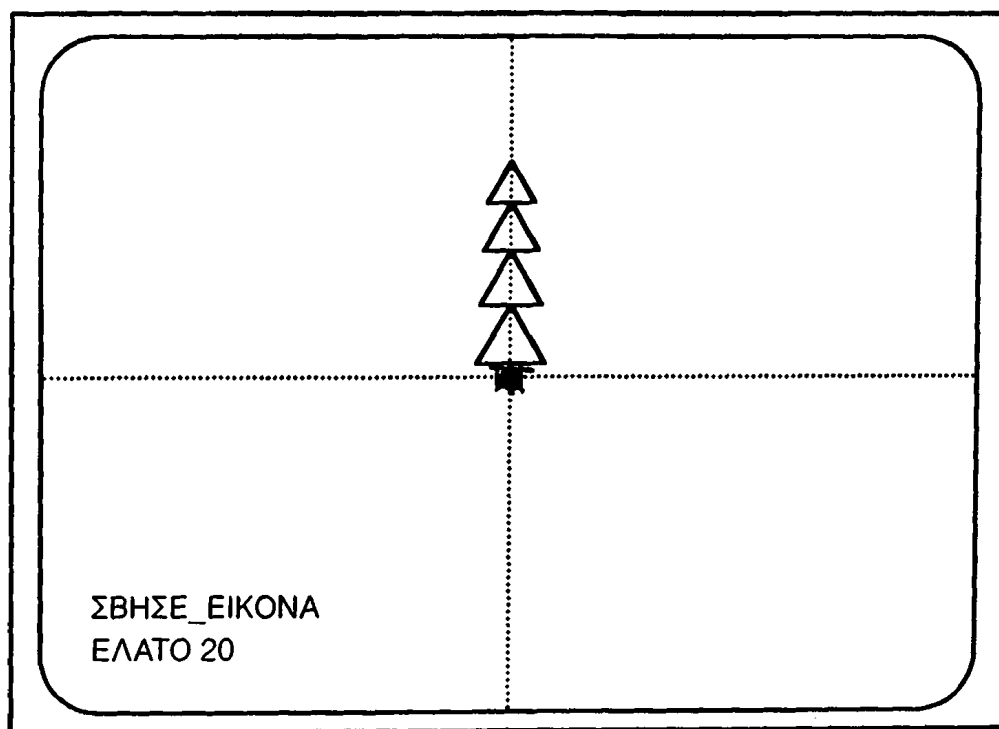
Η χελώνα προηγουμένως οπισθοχώρησε περισσότερο από όσο είχε προχωρήσει προς τα πάνω. Αυτό έγινε γιατί κατά την οπισθοχώρηση δε λάβαμε υπόψη ότι τα μεγέθη των κλαδιών μεταβάλλονταν.

Αν λοιπόν κατά την υποχώρηση μειώνουμε το μέγεθος όσο το αυξάναμε, τότε ισοφαρίζουμε τις δύο αποστάσεις.

```
ΤΟ ΕΛΑΤΟ :MEGEQOS
MAKE "PLHQOS RANDOM 6
ΚΟΡΜΟΣ (:MEGEQOS / 3)
MAKE "M1 :MEGEQOS
MAKE "M2 :MEGEQOS
ΚΛΑΔΙΑ :MEGEQOS :PLHQOS

ΜΗ_ΓΡΑΦΕΙΣ
MAKE "MEGEQOS (:MEGEQOS - 2)
REPEAT :PLHQOS
  (MAKE "YPSOS (:M1 * (SQRT(3)) / 2)
  ΟΠΙΣΘΕΝ :YPSOS
  MAKE "M1 (:M1 - 2))
ΟΠΙΣΘΕΝ (:M2 / 3)
ΓΡΑΨΕ
ΕΝΔ
```

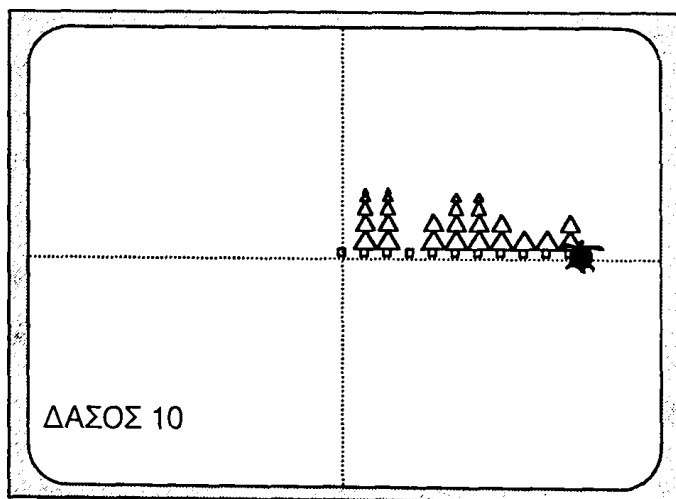
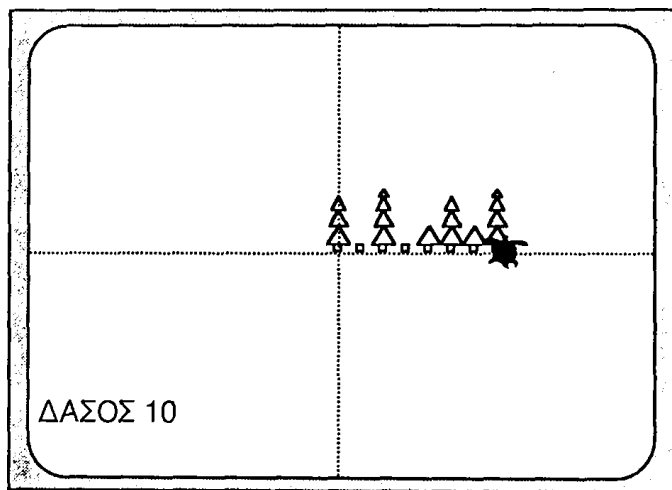
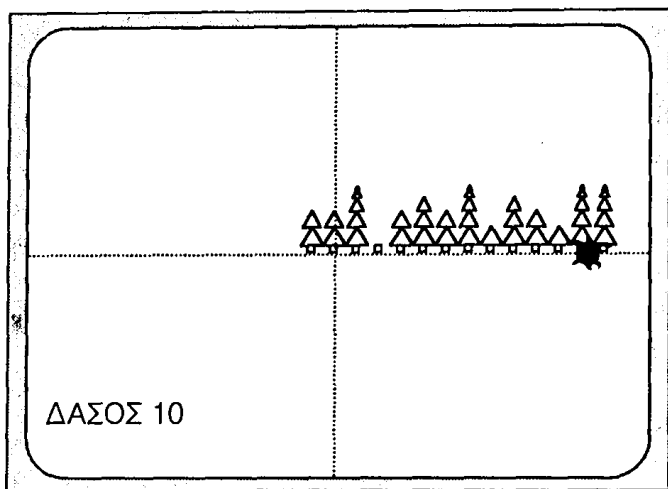
Ποιά
σκοπιμότητα
εξυπηρετούν οι
εντολές της 4ης
και της 5ης
γραμμής της
διαδικασίας
ΕΛΑΤΟ;

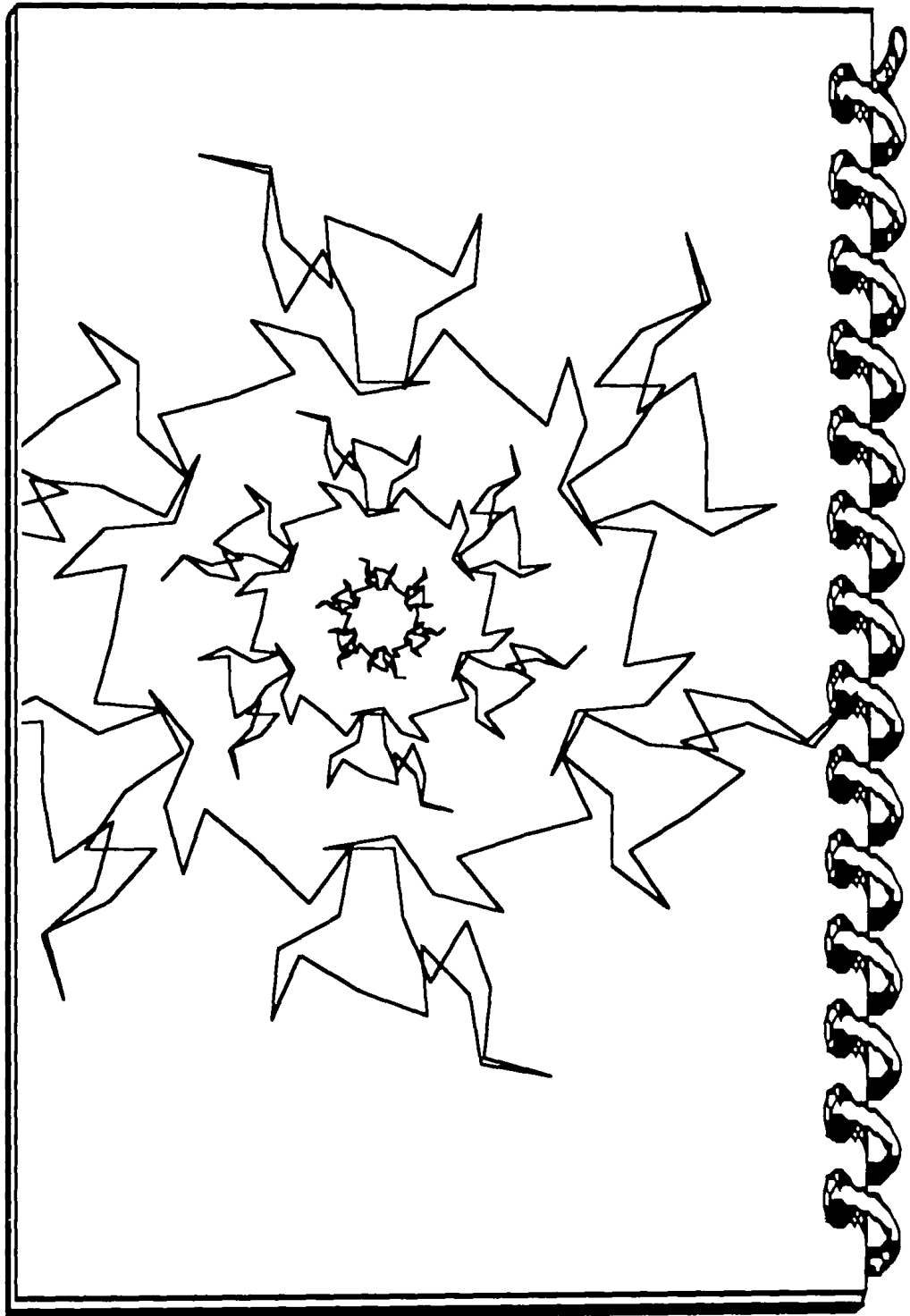


Εκδρομή στην Ελάτη*

Ας βάλουμε τη χελώνα να σχεδιάσει μερικά δάση με έλατα.

* Ελάτη, χωριό των Ζαγοροχωρίων του Ν. Ιωαννίνων που δικαιολογεί πλήρως το όνομά του γιατί είναι κατάφυτο με έλατα.





10 Το μηδέν και το άπειρο

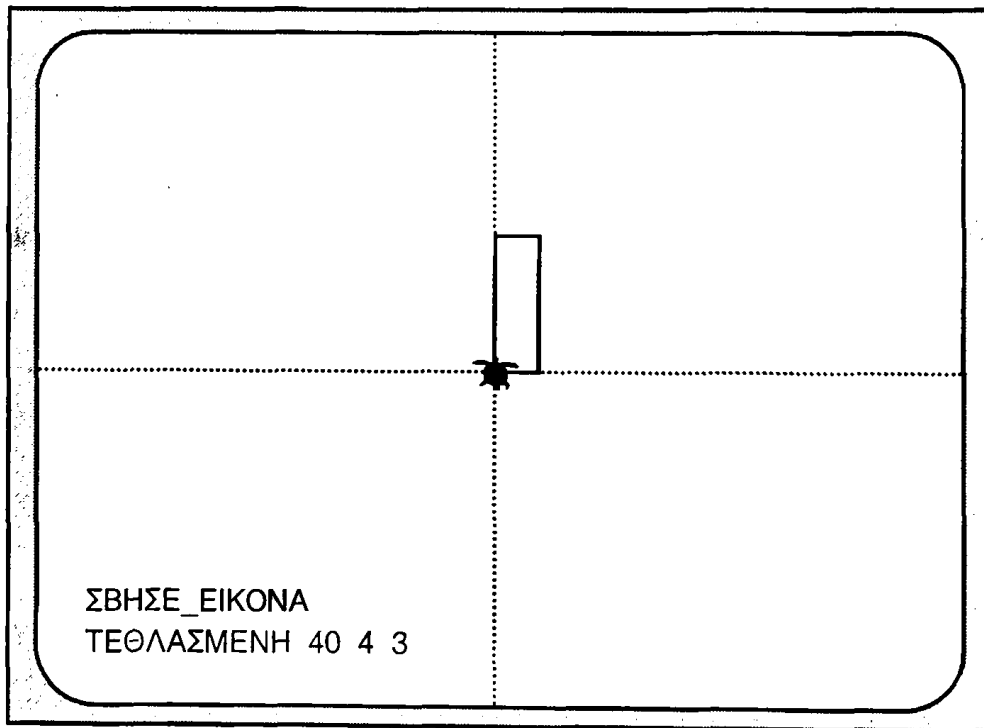
Τεθλασμένες γραμμές

Αναλύοντας ένα ορθογώνιο παραλληλόγραμμο μπορούμε να πούμε ότι αποτελείται από δύο ίδια κομμάτια. Το καθένα αποτελείται από ένα ευθύγραμμο τμήμα, μια στροφή 90 μοιρών και ένα μικρότερο ευθύγραμμο τμήμα.

Η δομή του σχήματος είναι ανάλογη με αυτήν του πολυγώνου και εκμεταλλευόμενοι την εμπειρία από τα πολύγωνα γράφουμε τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ η οποία για την τιμή 4 στη μεταβλητή PLHQOS, έχει ως αποτέλεσμα ένα ορθογώνιο παραλληλόγραμμο.

```
TO ΤΕΘΛΑΣΜΕΝΗ :PLEYRA :PLHQOS :LOGOS
ΑΡΙΣΤΕΡΑ 90
MAKE "ΜΟΙΡΕΣ (360 / :PLHQOS)
REPEAT (:PLHQOS / 2)
  [ ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA
    ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS]
ΔΕΞΙΑ 90
END
```

Η μεταβλητή LOGOS προσδιορίζει την αναλογία μεταξύ του μήκους της μεγάλης και της μικρής πλευράς.



«Για να διευκολυνθούμε, κατατάσσω τους συλλογισμούς με λόγους σε δύο επίπεδα ή στάδια:

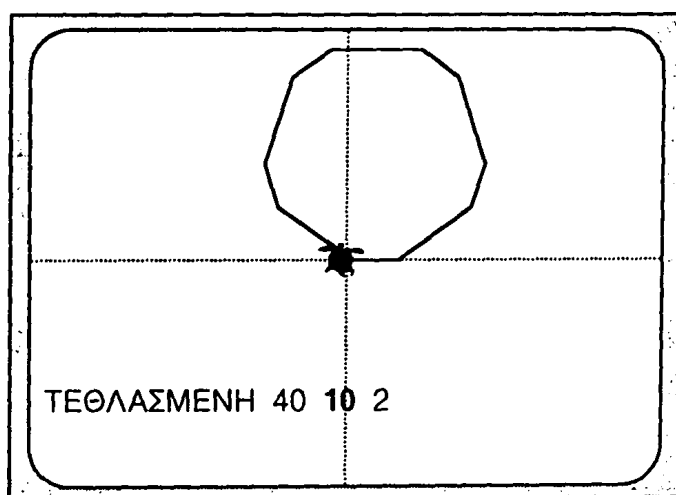
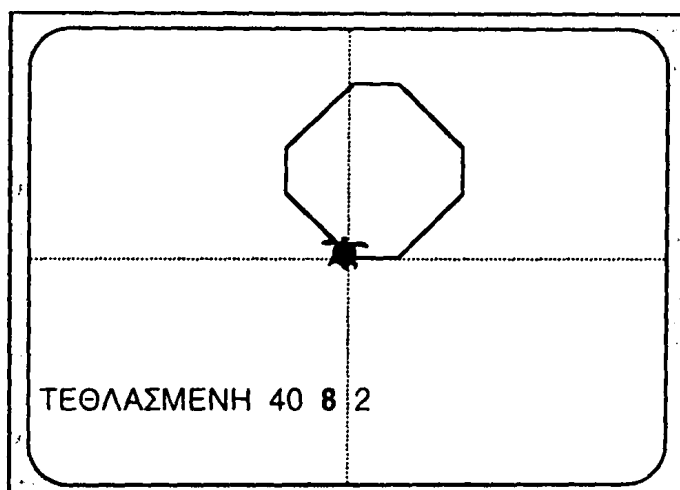
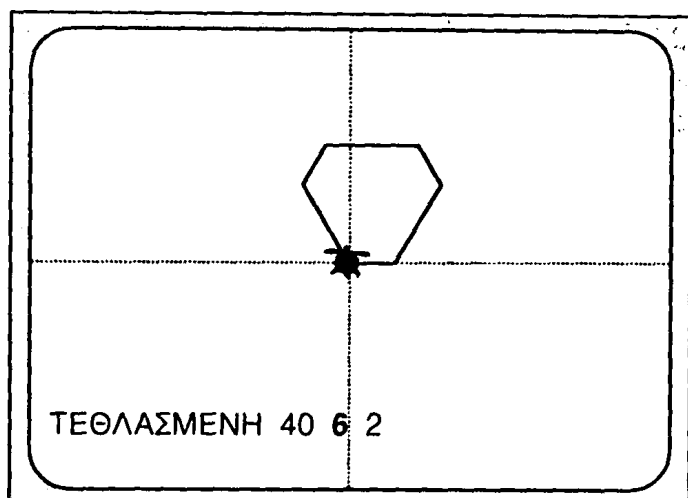
- (1) στο επίπεδο όπου ερμηνεύουμε περιφραστικά το αποτέλεσμα της διαίρεσης δύο αριθμών
- (2) στο επίπεδο όπου χρησιμοποιούμε την προηγούμενη ερμηνεία για να υπολογίσουμε κάποια άλλη ποσότητα»

A. ΑΓΩΝ



Κλειστές τεθλασμένες γραμμές

Στη διαδικασία ΤΕΘΛΑΣΜΕΝΗ διατηρούμε σταθερή την τιμή της μεταβλητής LOGOS και μεταβάλλουμε την τιμή της μεταβλητής ΡΛΗQΟΣ.



Ορθογώνιο παραλληλόγραμμο

Επειδή το ορθογώνιο παραλληλόγραμμο είναι ένα σχήμα που χρησιμοποιείται συχνά, το ορίζουμε ξεχωριστά.

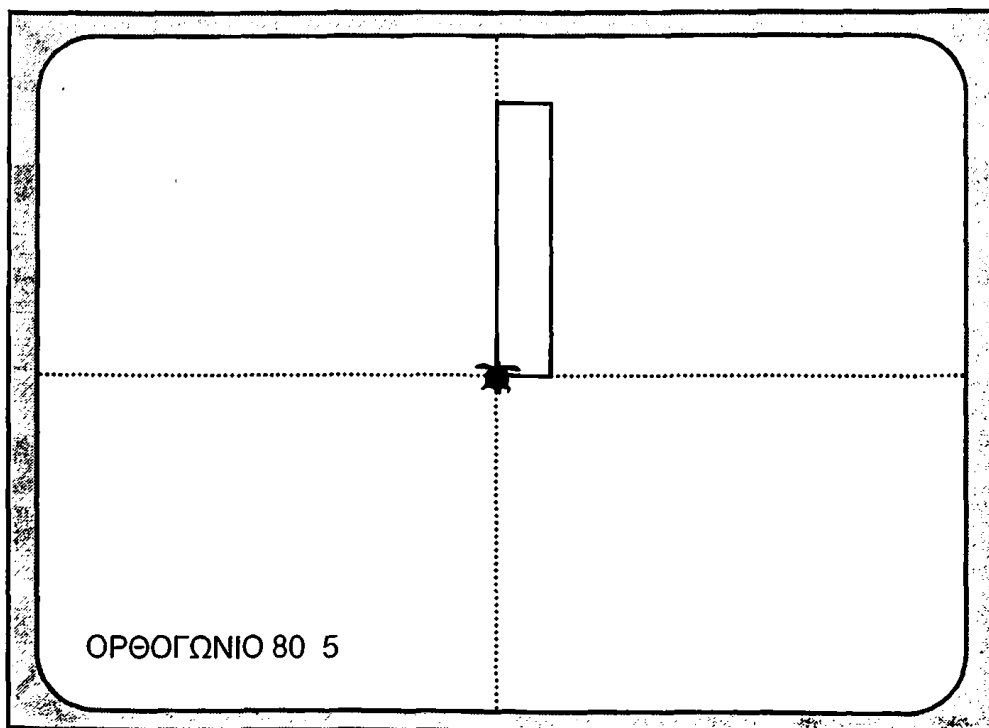
Για τον ορισμό μπορούμε ή να χρησιμοποιήσουμε τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ

```
ΤΟ ΟΡΘΟΓΩΝΙΟ :PLEYRA :LOGOS
ΤΕΘΛΑΣΜΕΝΗ :PLEYRA 4 :LOGOS
END
```

ή να ορίσουμε αναλυτικά ξανά όλα τα βήματα.

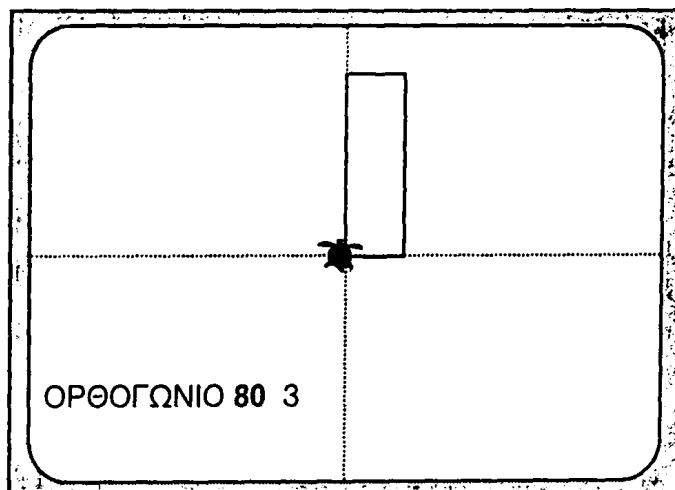
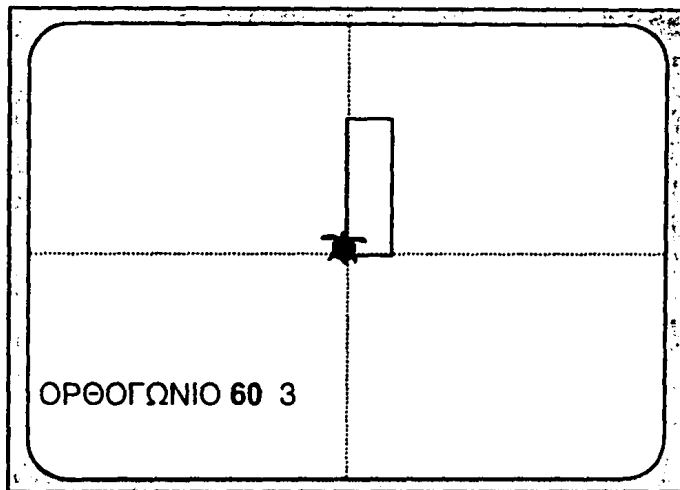
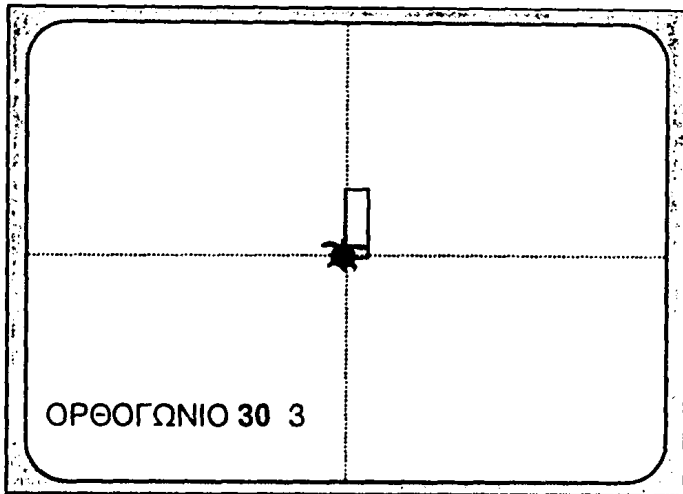
```
ΤΟ ΟΡΘΟΓΩΝΙΟ :PLEYRA :LOGOS
ΑΡΙΣΤΕΡΑ 90
ΜΑΚΕ "ΜΟΙΡΕΣ (360 / 4)
ΡΕΡΕΑΤ 2
    [ ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA
      ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS ]
ΔΕΞΙΑ 90
END
```

Με τη δεύτερη λύση η διαδικασία ΟΡΘΟΓΩΝΙΟ γίνεται ανεξάρτητη από τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ.



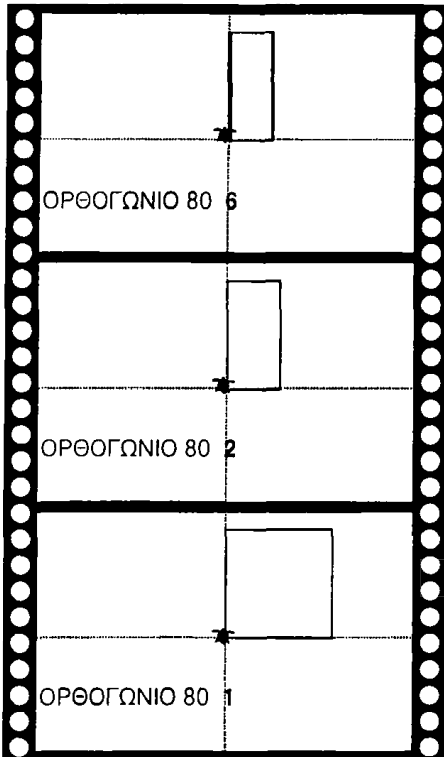
Σταθερή αναλογία πλευρών

Ας σχεδιάσουμε μερικά ορθογώνια παραλληλόγραμμα με διαφορετικά μεγέθη αλλά με σταθερό λόγο μεταξύ των πλευρών τους.



Διάφορες τιμές του λόγου των πλευρών

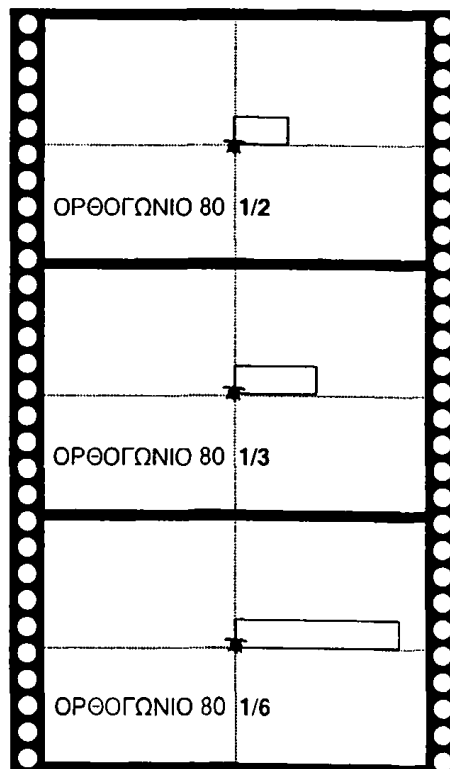
Ας σχεδιάσουμε ορθογώνια παραλληλόγραμμα με σταθερό μέγεθος πλευράς αλλάζοντας όμως το λόγο μεταξύ των πλευρών τους.



Λόγος 1 σημαίνει ισότητα

Το ορθογώνιο γίνεται τετράγωνο.

Διαίρεση δια $(1/2)$ σημαίνει πολλαπλασιασμός επί 2.



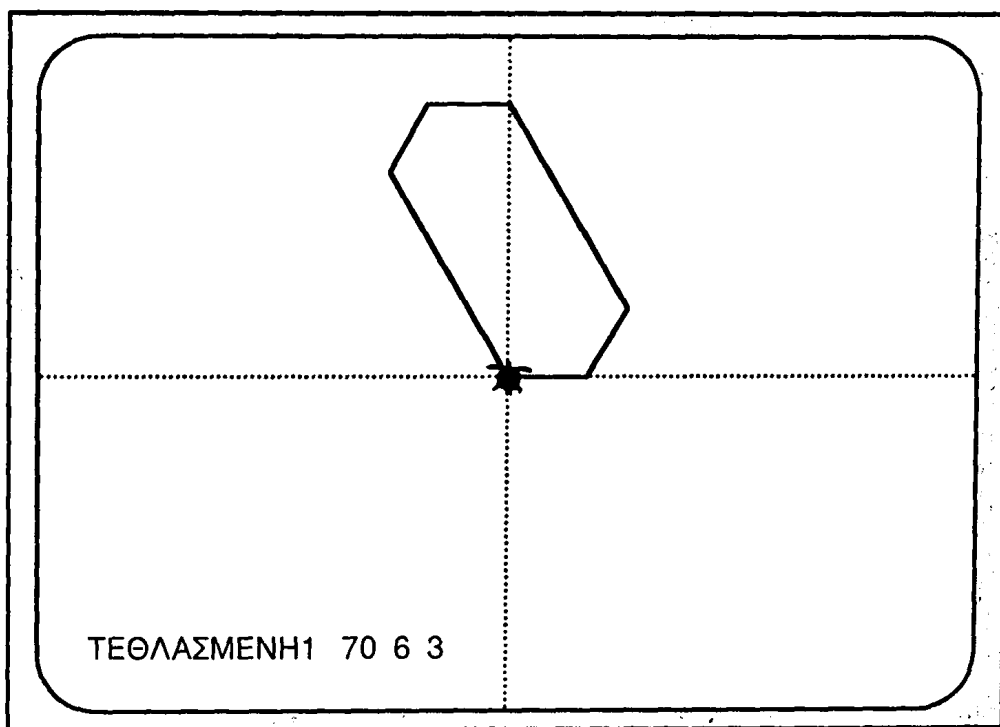
«...μελετάμε το λόγο δύο ίδιων φυσικών μεγεθών...
Στη συγκεκριμένη περίπτωση η αριθμητική τιμή του λόγου χρησιμεύει για σύγκριση: δείχνει πόσες φορές το ένα μήκος είναι μεγαλύτερο (ή μικρότερο) από το άλλο».
Α. Αγοπς

Μια πιο εξελιγμένη τεθλασμένη

Μπορούμε να εξελίξουμε τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ έτσι ώστε να είναι τρεις οι πλευρές που θα επαναλαμβάνονται.

Ας δούμε πως θα είναι μια τέτοια διαδικασία.

```
TO ΤΕΘΛΑΣΜΕΝΗ1 :PLEYRA :PLHQOS :LOGOS
ΑΡΙΣΤΕΡΑ 90
MAKE "ΜΟΙΡΕΣ (360 / :PLHQOS)
REPEAT (:PLHQOS / 3)
  [ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA
  ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS
  ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS ]
ΔΕΞΙΑ 90
END
```



Πλάγιο παραλληλόγραμμο

Το άθροισμα των γωνιών κάθε κανονικού πολυγώνου είναι 360° .

Στο ορθογώνιο παραλληλόγραμμο όλες οι γωνίες είναι ορθές και κατά συνέπεια ίσες μεταξύ τους.

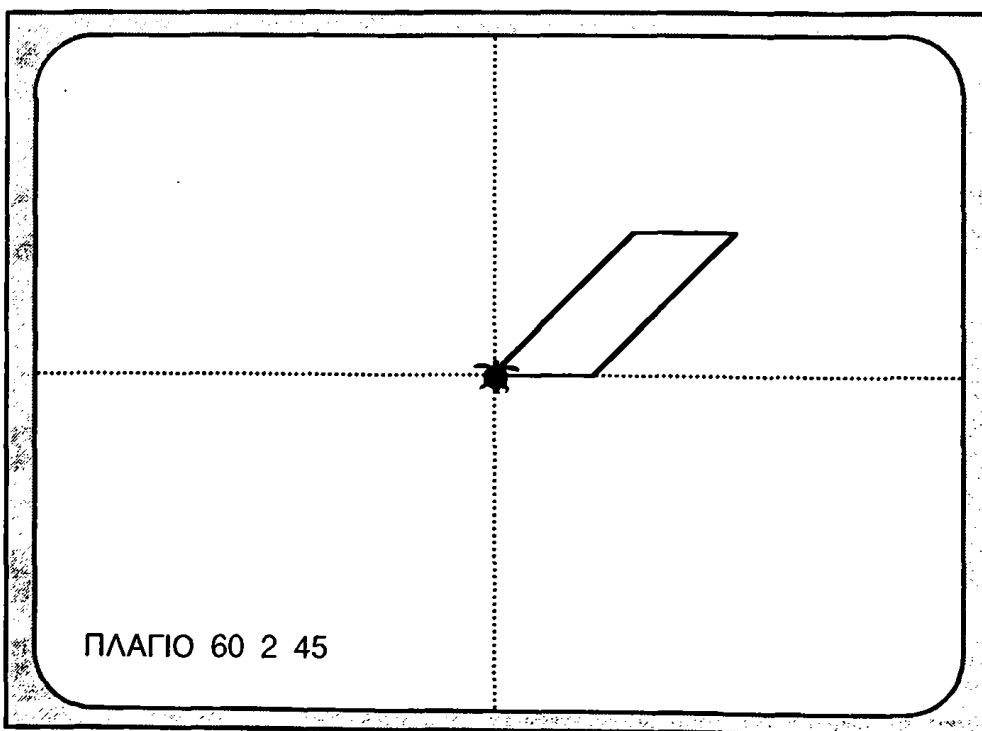
Τι θα συνέβαινε άραγε αν αυξάναμε δύο απέναντι γωνίες ενός ορθογωνίου παραλληλογράμμου και ταυτόχρονα μειώναμε τις άλλες δύο κατά ίση ποσότητα; Αυτό μπορεί να γίνει ως εξής:

Η γωνία του ορθογωνίου παραλληλογράμμου προκύπτει από τη διαίρεση $360/4$. Ας ονομάσουμε τις (απέναντι) μεγάλες γωνίες MOIRESA και τις (απέναντι) μικρές γωνίες MOIRESB.

Στις μεγάλες γωνίες προσθέτουμε την ποσότητα D δηλαδή έχουν τιμή $(360/4) + D$ ενώ στις μικρές την αφαιρούμε $(360/4) - D$.

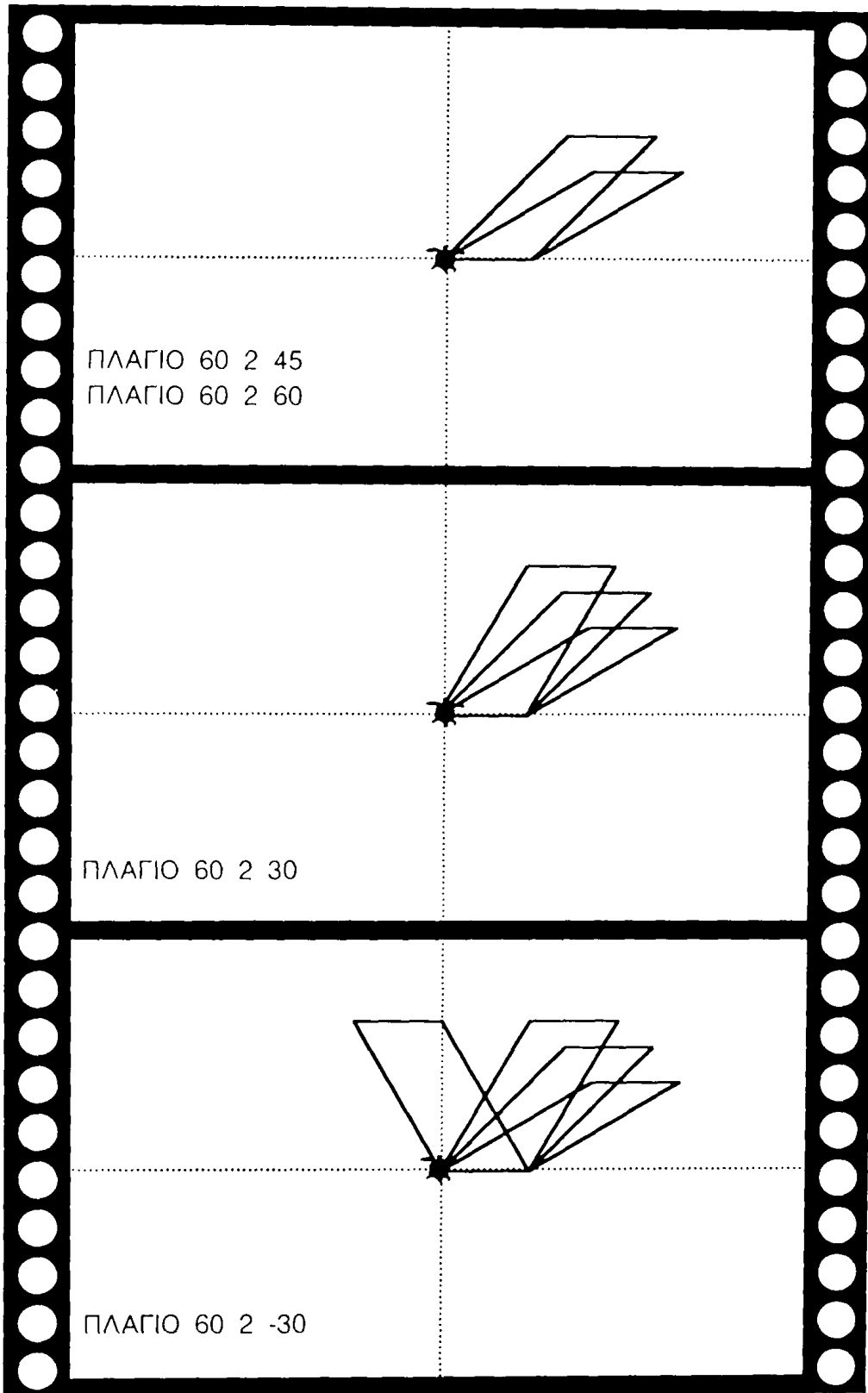
Ας δούμε πως διαμορφώνεται μια τέτοια διαδικασία, η ΠΛΑΓΙΟ:

```
TO ΠΛΑΓΙΟ :PLEYRA :LOGOS :D
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRESA (360 / 4) + :D
MAKE "MOIRESB (360 / 4) - :D
REPEAT 2
  [ΓΡΑΜΜΗ :MOIRESA :PLEYRA
  ΓΡΑΜΜΗ :MOIRESB :PLEYRA / :LOGOS]
ΔΕΞΙΑ 90
END
```



Πλάγια παραλληλόγραμμα

Για να καταλάβουμε καλύτερα το μηχανισμό της διαδικασίας ΠΛΑΓΙΟ ας τη βάλουμε να «τρέξει» μερικές φορές.



Η αίσθηση ότι η επιφάνεια πέφτει

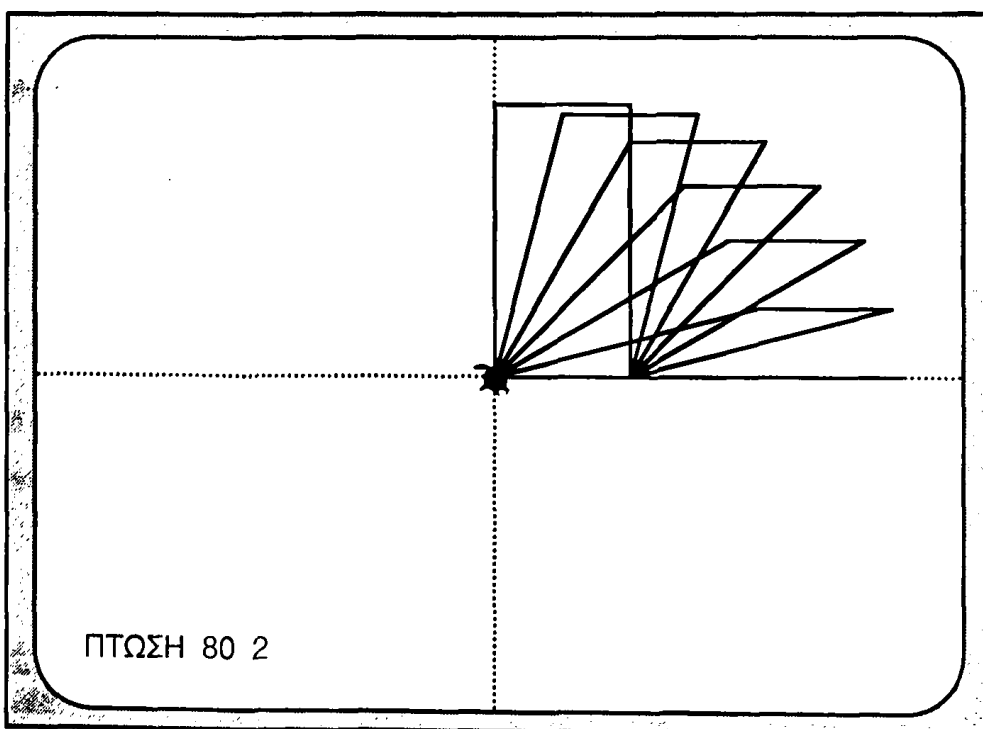
Οι εικόνες της προηγούμενης σελίδας δημιουργούν την αίσθηση ότι το παραλληλόγραμμο πέφτει.

Η αίσθηση δημιουργείται από τα διαφορετικά πλάγια παραλληλόγραμμο που σχεδιάζονται καθώς αυξάνει η τιμή της γωνίας D . Θα εκμεταλλευτούμε αυτήν την παρατήρηση για να φτιάξουμε μια διαδικασία που να σχεδιάζει διαδοχικές φάσεις της πτώσης του παραλληλογράμμου. Στην αρχή η D θα είναι μηδέν και στη συνέχεια θα αυξάνει κατά ένα σταθερό βήμα.

Ας δούμε τη διαδικασία ΠΤΩΣΗ:

```
TO ΠΤΩΣΗ :PLEYRA :LOGOS
MAKE "D 0
REPEAT 7
  [ ΠΛΑΓΙΟ :PLEYRA :LOGOS :D
  MAKE "D (:D + 15) ]
END
```

Παρατηρείστε ότι όταν η D είναι 0 σχηματίζεται ορθογώνιο παραλληλόγραμμο, και όταν γίνει 90° το πλάγιο παραλληλόγραμμο εκφυλίζεται σε μια (οριζόντια) γραμμή.

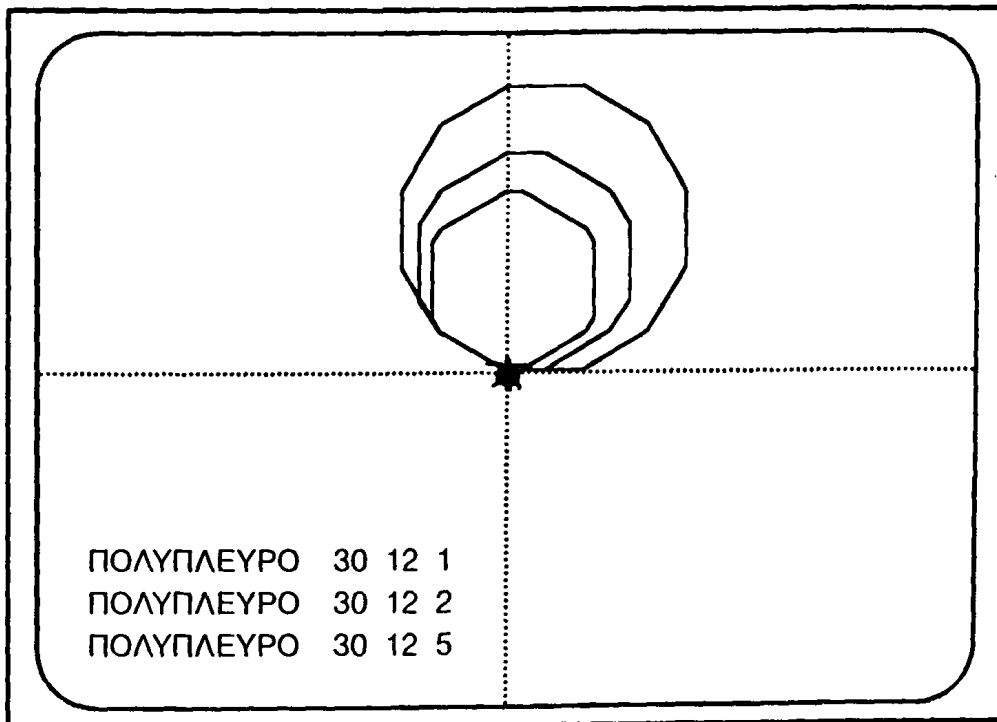


Ένα πολύπλευρο

Συνδυάζοντας την εμπειρία από τις διαδικασίες ΤΕΘΛΑΣΜΕΝΗ και ΠΟΛΥΓΩΝΟ μπορούμε να φτιάξουμε τη διαδικασία ΠΟΛΥΠΛΕΥΡΟ. Το πολύπλευρο είναι ένα σχήμα με σταθερό λόγο των μηκών δύο διαδοχικών πλευρών. Αυτός ο συνδυασμός των δύο πλευρών επαναλαμβάνεται μέχρι να σχηματισθεί μια κλειστή τεθλασμένη γραμμή. Στη διαδικασία θα πρέπει να δίνονται ως παράμετροι το πλήθος των πλευρών, το μέγεθος της μιας βασικής πλευράς και ο λόγος της δευτερεύουσας πλευράς ως προς τη βασική. Η γωνία μεταξύ των πλευρών θα υπολογίζεται ως το πηλίκο των 360 μοιρών δια το πλήθος των πλευρών.

Έτσι η διαδικασία ΠΟΛΥΠΛΕΥΡΟ θα είναι:

```
ΤΟ ΠΟΛΥΠΛΕΥΡΟ :PLEYRA :PLHQOS :LOGOS
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS / 2)
  [ΓΡΑΜΜΗ :MOIRES :PLEYRA
  ΓΡΑΜΜΗ :MOIRES :PLEYRA / :LOGOS ]
ΔΕΞΙΑ 90
END
```



Φωλιασμένες επαναλήψεις

Στον κώδικα της προηγούμενης διαδικασίας υπάρχουν δύο σχεδόν ίδιες εντολές που σχεδιάζουν την κύρια και τη δευτερεύουσα πλευρά του πολύπλευρου, οι

```
GRAMMH :MOIRES :PLEYRA
GRAMMH :MOIRES :PLEYRA/:LOGOS
```

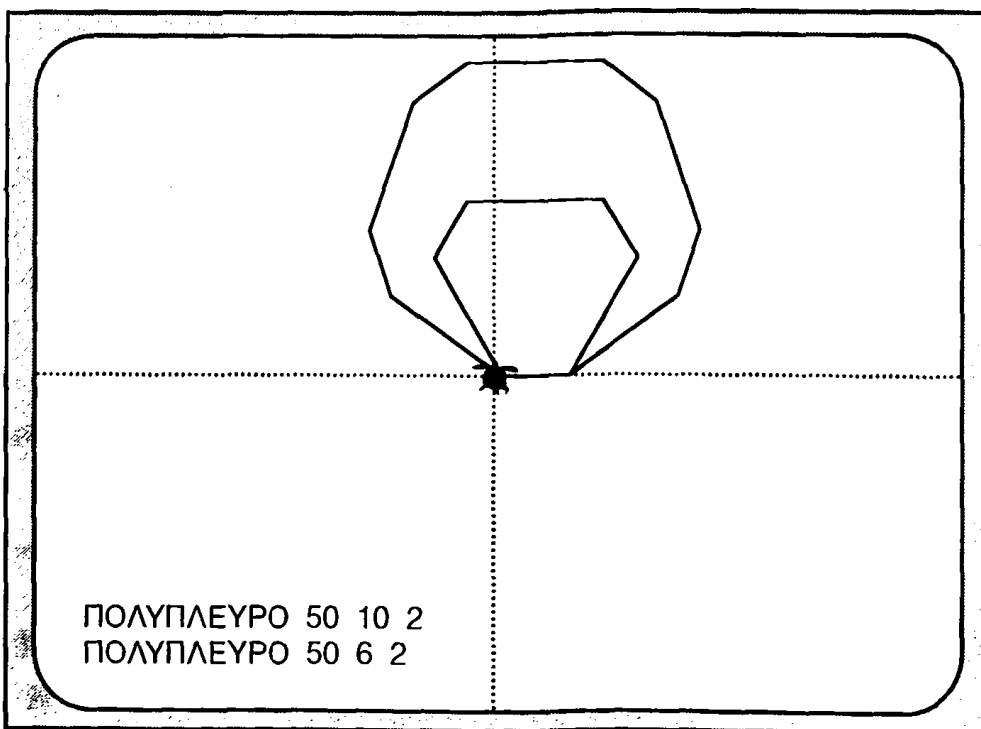
Μπορούμε και τις δύο να τις αντικαταστήσουμε με μια εντολή που επαναλαμβάνεται δύο φορές. Αυτό γίνεται αν χρησιμοποιήσουμε τη μεταβλητή ΒΗΜΑ_ΛΟΓΟΥ που θα αλλάζει κάθε φορά το λόγο της δευτερεύουσας πλευράς ως προς την κύρια:

```
REPEAT 2
_ [GRAMMH :MOIRES :PLEYRA/:LOGOS
_ MAKE "LOGOS :LOGOS*:ΒΗΜΑ_ΛΟΓΟΥ ]
```

Με αυτές τις τροποποιήσεις η διαδικασία ΠΟΛΥΠΛΕΥΡΟ γίνεται:

```
ΤΟ ΠΟΛΥΠΛΕΥΡΟ :PLEYRA :PLHQOS :ΒΗΜΑ_ΛΟΓΟΥ
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS / 2)
  [ MAKE "LOGOS 1
  REPEAT 2
    [ GRAMMH :MOIRES :PLEYRA / :LOGOS
    MAKE "LOGOS :LOGOS * :ΒΗΜΑ_ΛΟΓΟΥ
  ]
]
ΔΕΞΙΑ 90
END
```

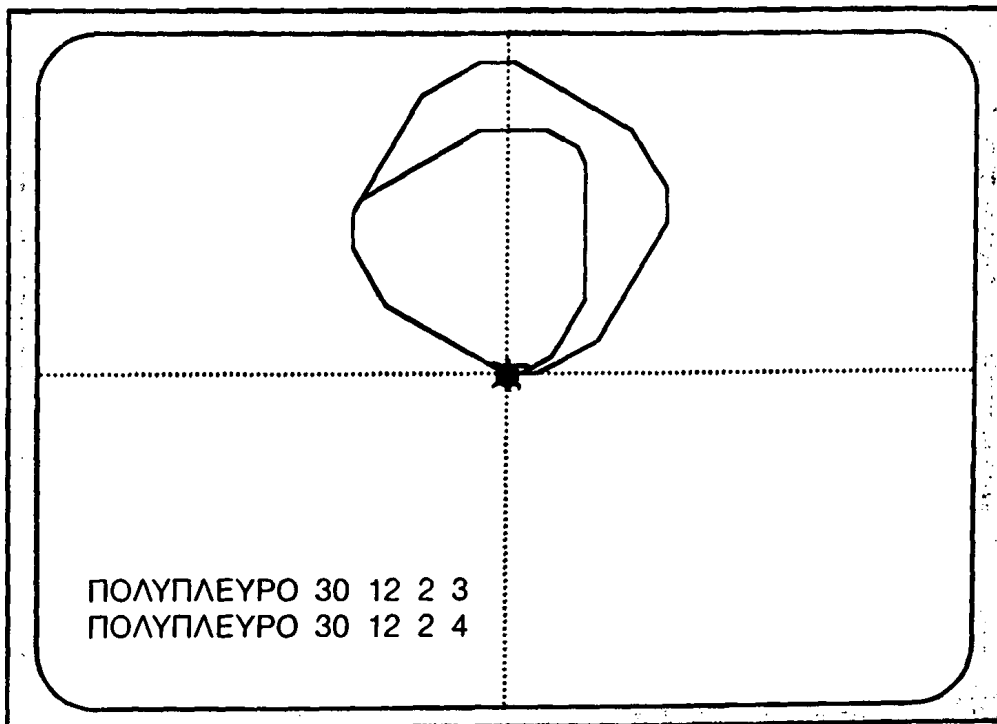
Παρατηρείστε την ύπαρξη μιας δομής επανάληψης μέσα σε μια άλλη. Είναι αξιοσημείωτο ότι η μια είναι φωλιασμένη στο εσωτερικό της άλλης και δεν υπάρχει περίπτωση διασταύρωσης των δύο αυτών δομών.



Ένα πιο ευέλικτο πολύπλευρο

Η προηγούμενη διαδικασία σχεδιάζει πολύπλευρα που έχουν προκύψει από την επανάληψη δύο πλευρών. Μπορούμε να βελτιώσουμε τη διαδικασία έτσι ώστε να δέχεται ως παράμετρο το πλήθος των διαδοχικών βασικών πλευρών που θα επαναλαμβάνονται. Αρκεί να χρησιμοποιήσουμε τη μεταβλητή DIADOXH που θα δίνει το πλήθος των διαδοχικών πλευρών που επαναλαμβάνονται και να κάνουμε τις σχετικές μικρές τροποποιήσεις στον κώδικα της προηγούμενης διαδικασίας.

```
ΤΟ ΠΟΛΥΠΛΕΥΡΟ :PLEYRA :PLHQOS :BHMA_LOGOY :DIADOXH
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS / :DIADOXH)
  [ MAKE "LOGOS 1
    REPEAT :DIADOXH
      [ ΓΡΑΜΜΗ :MOIRES :PLEYRA / :LOGOS
        MAKE "LOGOS :LOGOS " :BHMA_LOGOY
      ]
    ]
]
ΔΕΞΙΑ 90
END
```



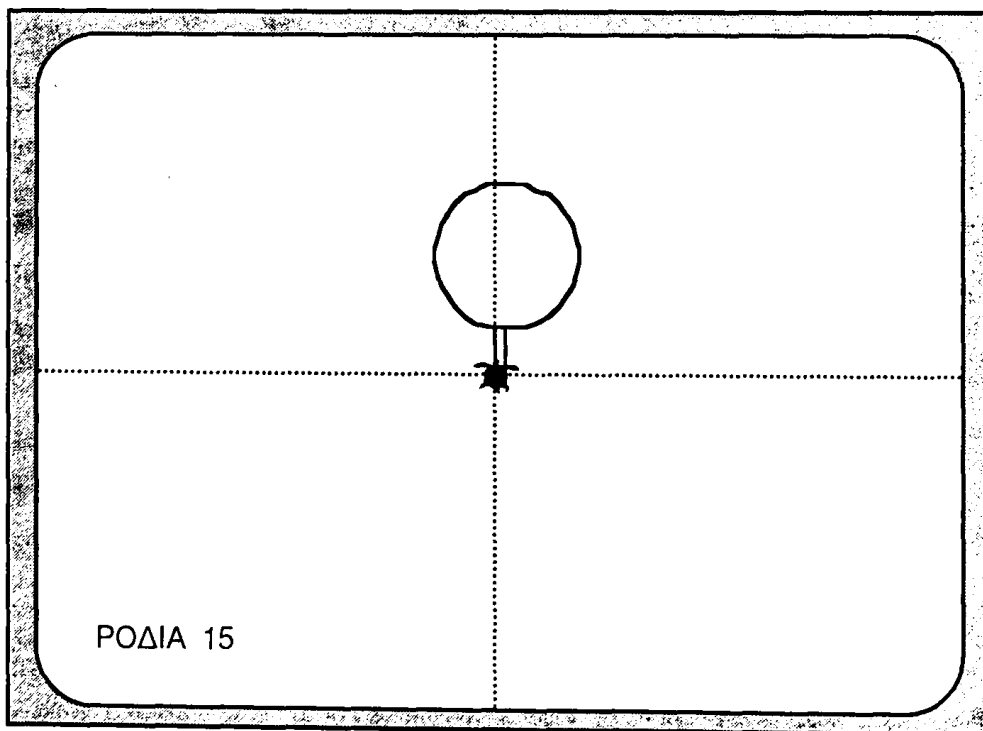
Σχεδιάζοντας τη ροδιά

Μια ροδιά μπορεί να σχεδιασθεί από τον κορμό και το φύλλωμα. Το φύλλωμα παριστάνεται ως κύκλος και ο κορμός σχεδιάζεται με τη βοήθεια του ορθογωνίου.

```
ΤΟ ΚΟΡΜΟΣ :YPS_KORMOY  
ΜΑΚΕ "LOGOS 4  
ΚΙΝΗΣΗ_ΔΙΠΛΑ MINUS (:YPS_KORMOY / (2*:LOGOS))  
ΟΡΘΟΓΩΝΙΟ :YPS_KORMOY :LOGOS  
ΚΙΝΗΣΗ_ΔΙΠΛΑ (:YPS_KORMOY / (2*:LOGOS))  
ΜΗ_ΓΡΑΦΕΙΣ  
ΕΥΘΕΙΑ :YPS_KORMOY  
ΓΡΑΨΕ  
ΕΝΔ
```

Η διαδικασία ΡΟΔΙΑ είναι:

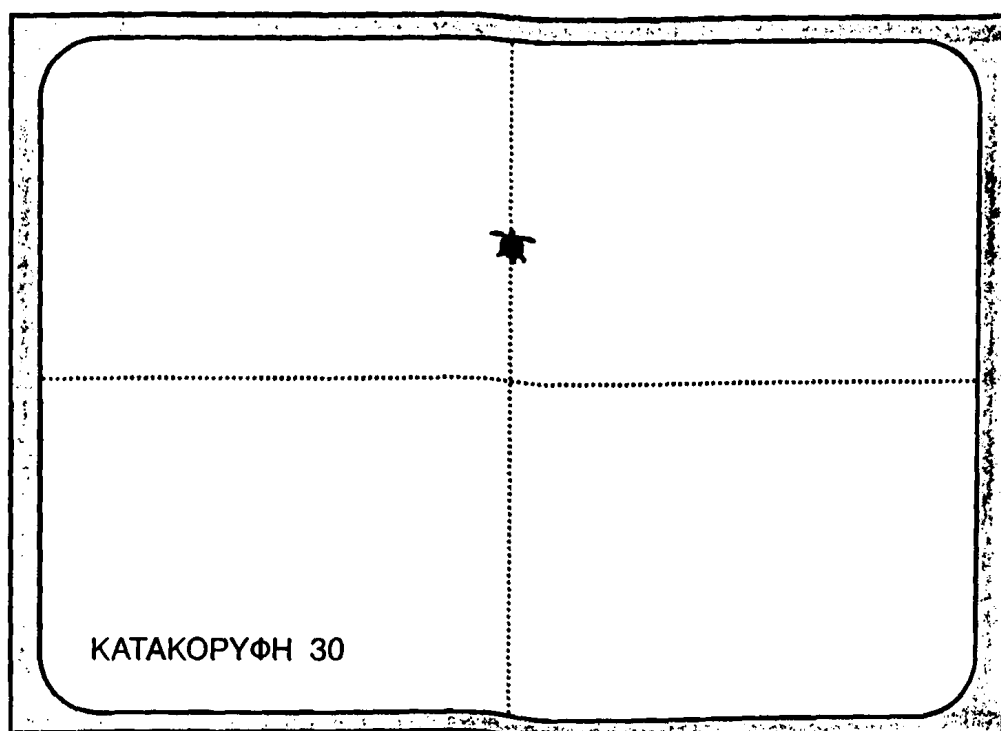
```
ΤΟ ΡΟΔΙΑ :ΜΕΓΕΘΟΣ  
ΚΟΡΜΟΣ :ΜΕΓΕΘΟΣ  
ΚΥΚΛΟΣ (:ΜΕΓΕΘΟΣ / 4)  
ΜΗ_ΓΡΑΦΕΙΣ  
ΟΠΙΣΘΕΝ :ΜΕΓΕΘΟΣ  
ΓΡΑΨΕ  
ΕΝΔ
```



Κατακόρυφη μετατόπιση

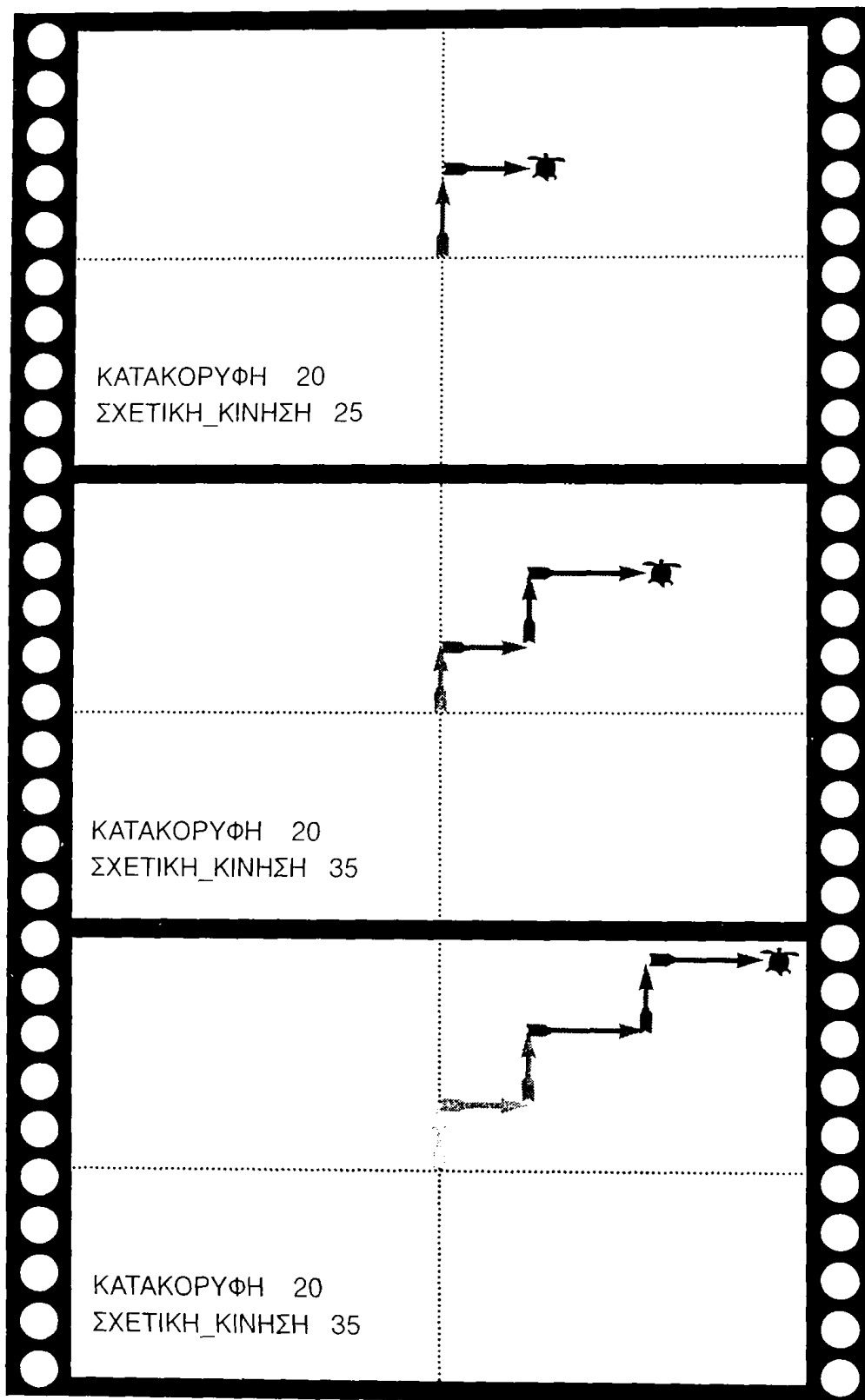
Εκτός από την οριζόντια μετατόπιση θα χρειαστεί και κατακόρυφη μετατόπιση της χελώνας. Η κατακόρυφη μετατόπιση είναι σχετικά εύκολη γιατί έχουμε συνηθίσει να αφήνουμε τη χελώνα με κατεύθυνση προς τα πάνω. Έτσι το μόνο που χρειάζεται είναι να πούμε στη χελώνα να μη γράφει, να κινηθεί ευθύγραμμα και τέλος να έρθει σε θέση για να γράψει πάλι.

```
ΤΟ ΚΑΤΑΚΟΡΥΦΗ :APOSTASH  
ΜΗ_ΓΡΑΦΕΙΣ  
ΕΥΘΕΙΑ :APOSTASH  
ΓΡΑΨΕ  
END
```



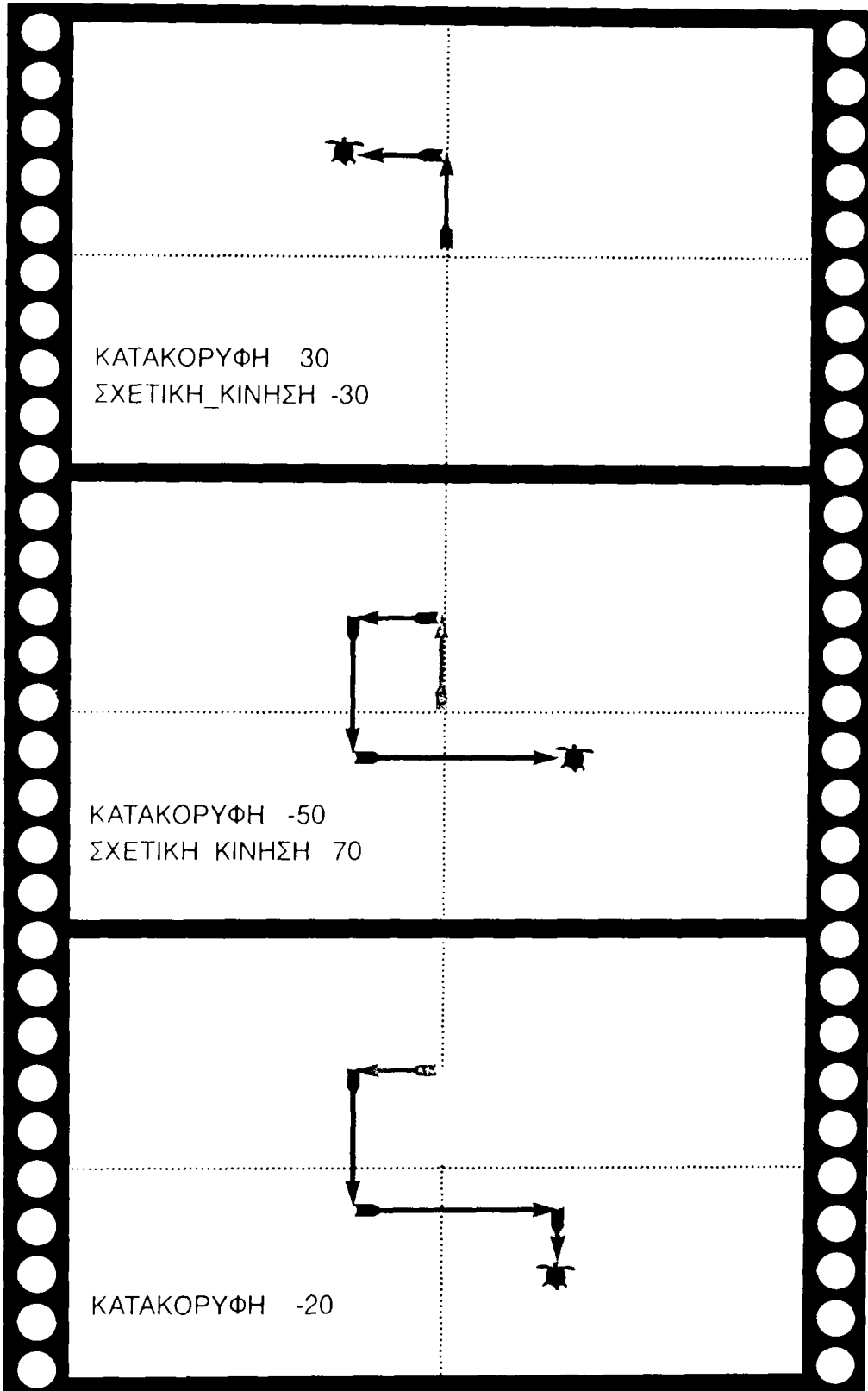
Η χελώνα μετατοπίζεται στο επίπεδο

Χρησιμοποιώντας τις ΚΑΤΑΚΟΡΥΦΗ και ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ μετακινούμε τη χελώνα σε όποιο σημείο της οθόνης θέλουμε.



Αρνητικές μετατοπίσεις

Δίνοντας αρνητικές τιμές στις προηγούμενες διαδικασίες μπορούμε να μετατοπίσουμε τη χελώνα προς όλες τις κατευθύνσεις.



Μια διαδικασία μετατόπισης

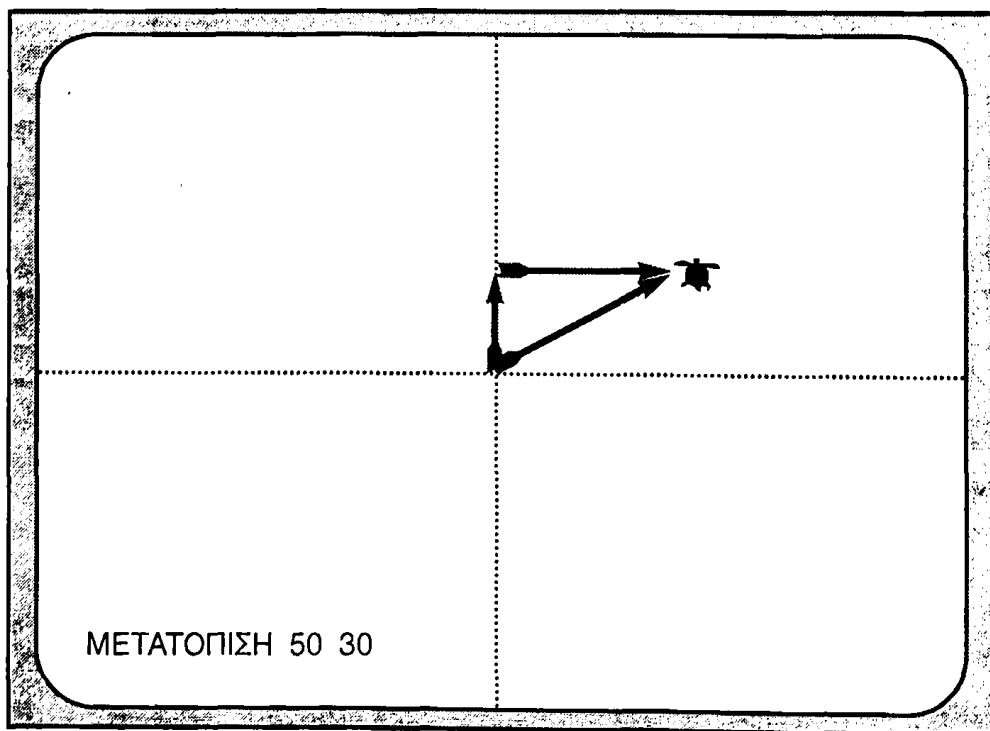
Για την κίνηση στο επίπεδο οι δύο διαδικασίες, για την οριζόντια και για την κατακόρυφη μετατόπιση, μπορούν να αντικατασταθούν από μια. Αυτή η διαδικασία που θα την ονομάσουμε ΜΕΤΑΤΟΠΙΣΗ θα έχει δύο μεταβλητές που θα καθορίζουν την οριζόντια και την κάθετη συνιστώσα της μετατόπισης.

```
ΤΟ ΜΕΤΑΤΟΠΙΣΗ :X :Y
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ :X
ΚΑΤΑΚΟΡΥΦΗ :Y
END
```

Η τελική θέση της χελώνας στο σύστημα αναφοράς με αρχή το κέντρο της οθόνης, δίνεται από το διάνυσμα θέσης που σχεδιάζεται από το κέντρο του συστήματος μέχρι τη χελώνα.

Η πλάγια κίνηση θεωρείται ως σύνθεση μιας οριζόντιας και μιας κατακόρυφης.

Το διάνυσμα από μαθηματική άποψη, είναι ένα διατεταγμένο ζεύγος αριθμών.



Η διδιάστατη χελώνα κινείται σε ένα σύστημα καρτεσιανών συντεταγμένων.



Η χελώνα σκαφαλώνει σε μια σκάλα

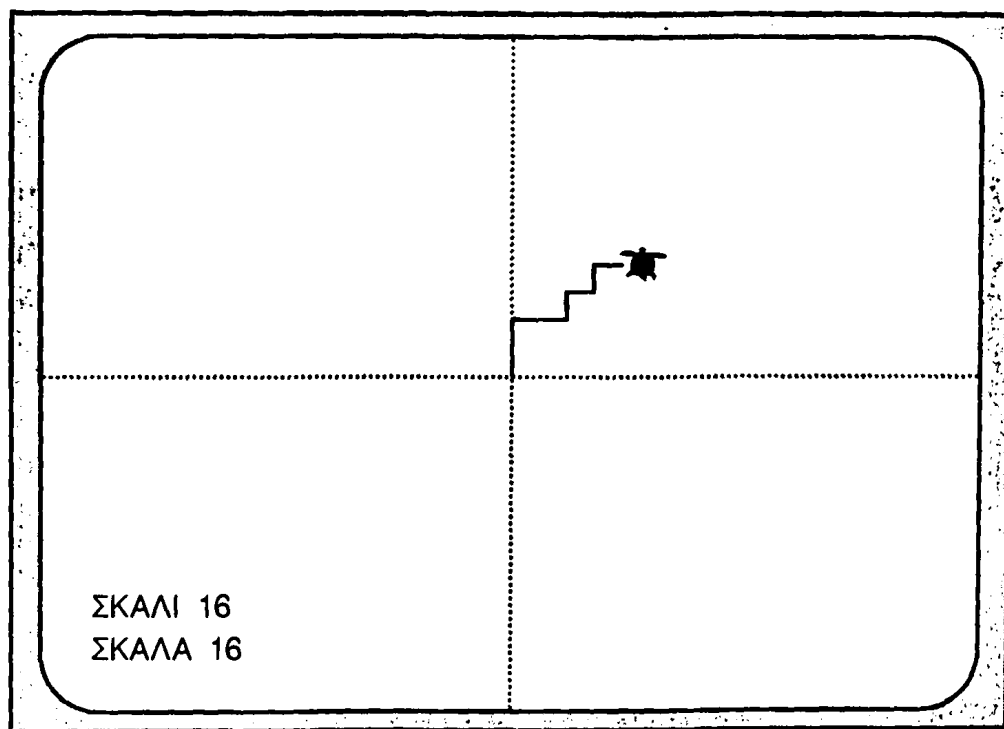
Χρησιμοποιώντας την εμπειρία από τις μετατοπίσεις, μπορούμε να φτιάξουμε μια διαδικασία που να διατάζει τη χελώνα να ακολουθήσει μια διαδρομή που να μοιάζει με σκάλα.

Το μόνο που χρειάζεται είναι η οριζόντια (ή η κατακόρυφη) συνιστώσα της απόστασης που πρέπει να διανύσει η χελώνα.

```
TO ΣΚΑΛΙ :APOSTASH
MAKE "YPSOS :APOSTASH
ΕΥΘΕΙΑ :YPSOS
MAKE "PLATOS :APOSTASH
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :PLATOS
ΑΡΙΣΤΕΡΑ 90
END
```

Έστω ότι θέλουμε να ανέβουμε στο ίδιο ύψος αλλά το σκαλί που σχεδιάσαμε προηγουμένως είναι πολύ ψηλό για τη χελώνα. Ας σχεδιάσουμε μια σκάλα που να αποτελείται από δύο σκαλιά που να έχουν το μισό ύψος.

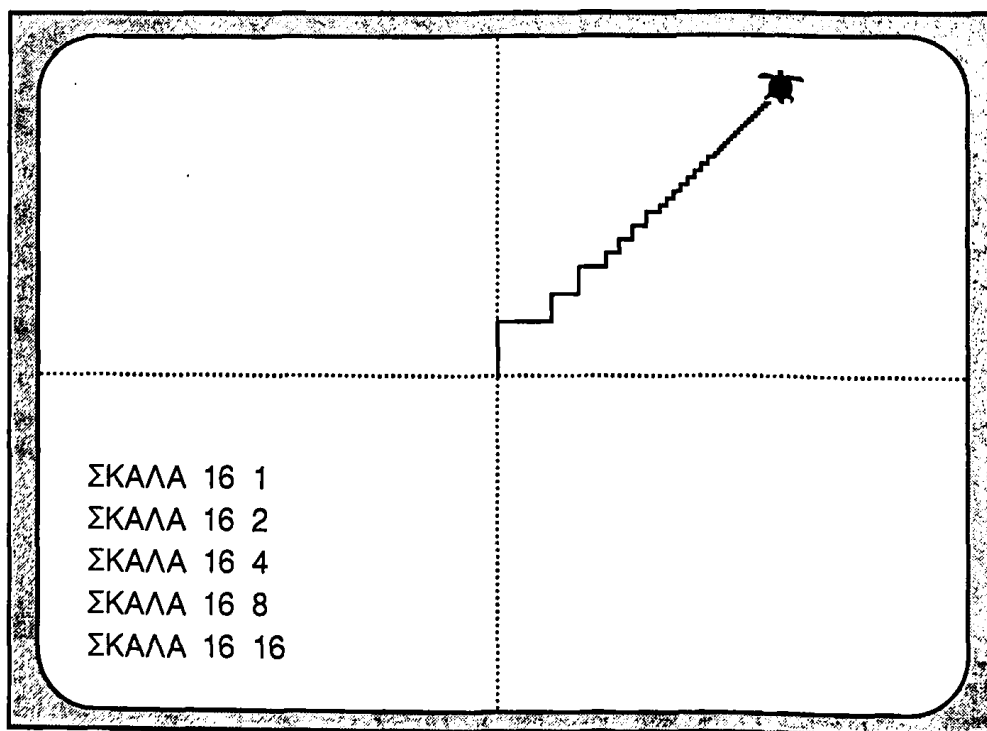
```
TO ΣΚΑΛΑ :APOSTASH
REPEAT 2
  [ ΣΚΑΛΙ (:APOSTASH / 2) ]
END
```



Επιλέγοντας το πλήθος των σκαλιών

Το επόμενο βήμα είναι μια σκάλα που θα καλύπτει την ίδια κατακόρυφη και οριζόντια απόσταση αλλά με περισσότερα σκαλιά το πλήθος των οποίων θα δίνεται ως παράμετρος της διαδικασίας.

```
TO ΣΚΑΛΑ :APOSTASH :PLHQOS
MAKE "YPSOS (:APOSTASH / PLHQOS)
MAKE "PLATOS (:APOSTASH / PLHQOS)
REPEAT :PLHQOS
  [ ΕΥΘΕΙΑ :YPSOS
    ΔΕΞΙΑ 90
    ΕΥΘΕΙΑ :PLATOS
    ΑΡΙΣΤΕΡΑ 90 ]
END
```



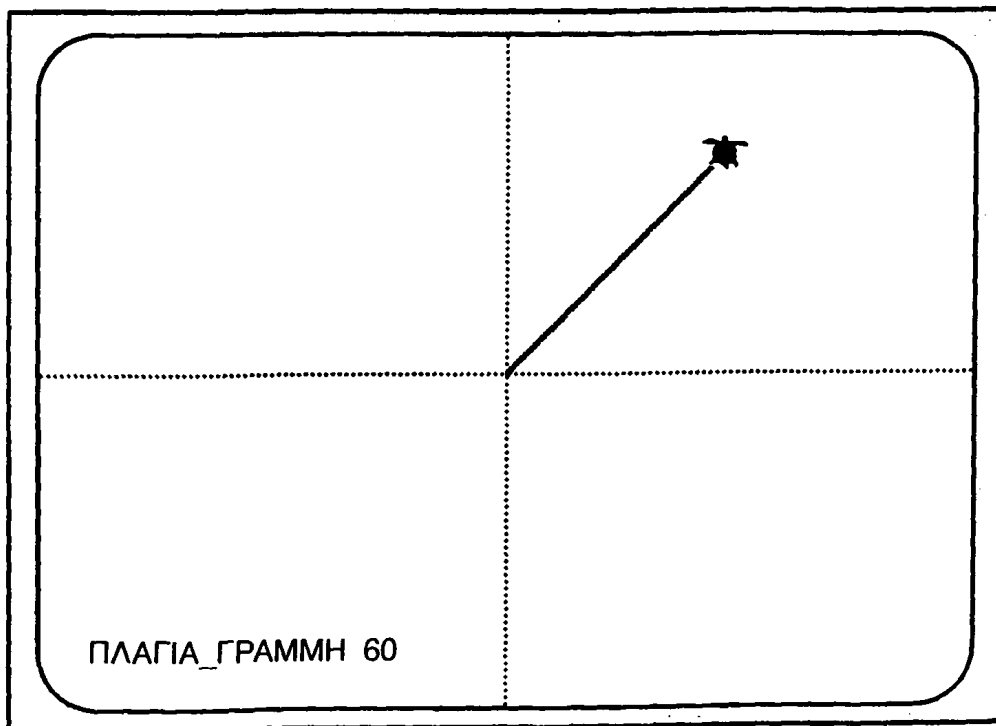
Πλάγια γραμμή

Η πραγματική απόσταση που διανύει σε ευθεία γραμμή η χελώνα από το σημείο που ξεκινάει μέχρι εκεί που καταλήγει, υπολογίζεται από το Πυθαγόρειο θεώρημα ως η υποτείνουσα του ορθογώνιου τριγώνου με κάθετες πλευρές την κατακόρυφη και την οριζόντια απόσταση.

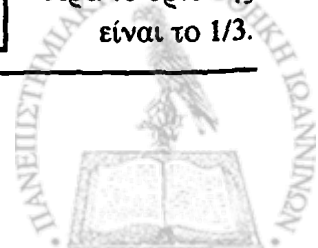
Αν το πλήθος των σκαλιών είναι ίσο με τα βήματα που πρέπει να γίνουν για να καλύψει η χελώνα την πραγματική απόσταση τότε το μέγεθος των σκαλιών-βημάτων θα προκύπτει από τη διαίρεση της υποτείνουσας δια το πλήθος των σκαλιών.

Έτσι μπορούμε να γράψουμε μια διαδικασία που να χαράζει μια πλάγια γραμμή.

```
TO ΠΛΑΓΙΑ_ΓΡΑΜΜΗ :ΑPOSTASH
MAKE "Y :ΑPOSTASH
MAKE "X :ΑPOSTASH
MAKE "ΥΡΟΤΕΙΝ SQRT((:Y * :Y) + (:X * :X))
MAKE "PLHQOS (:ΥΡΟΤΕΙΝ / 1)
MAKE "ΥPSOS (:Y / :PLHQOS)
MAKE "PLATOS (:X / :PLHQOS)
REPEAT :PLHQOS
  [ ΕΥΘΕΙΑ :ΥPSOS
    ΔΕΞΙΑ 90
    ΕΥΘΕΙΑ :PLATOS
    ΑΡΙΣΤΕΡΑ 90 ]
END
```



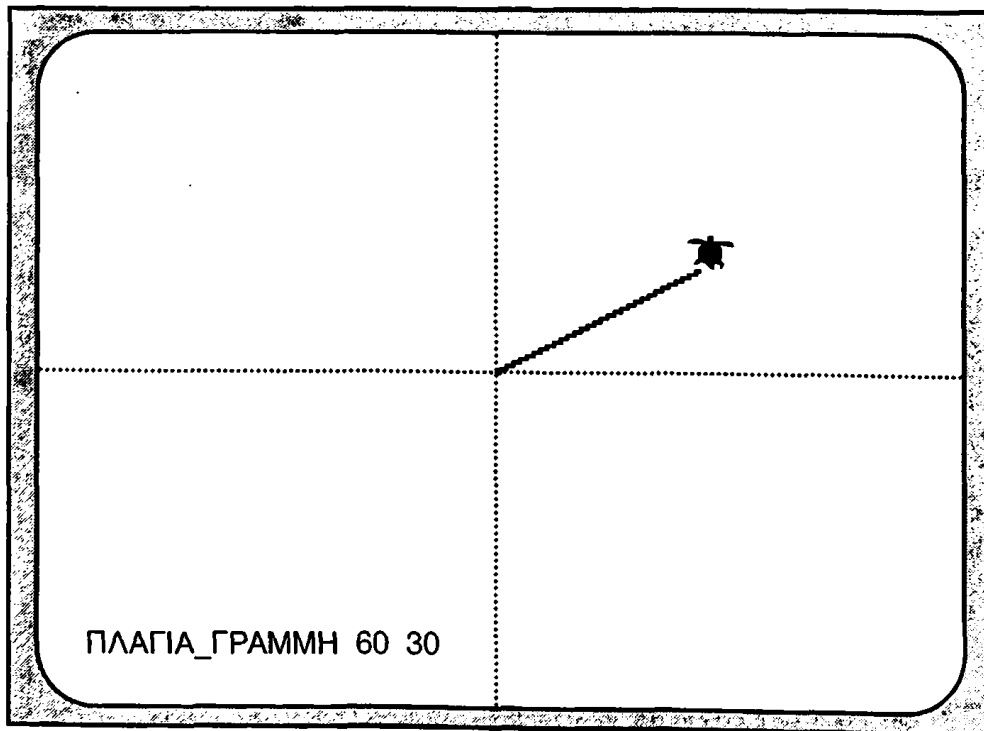
Διαισθητικός ορισμός του ορίου: Αν υπάρχει μια άπειρη ακολουθία διαδοχικών προσεγγίσεων ενός αριθμού, τότε ονομάζουμε αυτόν τον αριθμό όριο της ακολουθίας. Παράδειγμα: Η ακολουθία 0,3 0,33 0,333 0,3333... είναι μια ακολουθία διαδοχικών προσεγγίσεων του αριθμού $1/3$. Άρα το όριο της είναι το $1/3$.



Επιλογή της κλίσης της πλάγιας γραμμής

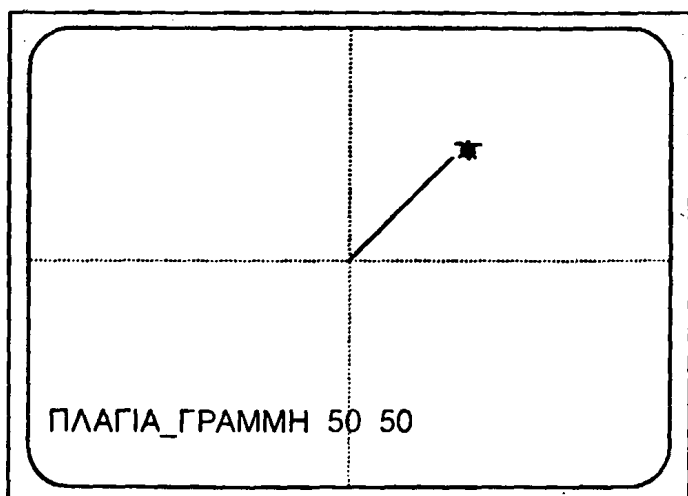
Αν η οριζόντια και η κατακόρυφη απόσταση δεν είναι ίδιες, τότε η κλίση της γωνίας δε θα είναι πάντοτε η ίδια. Έτσι αν τροποποιήσουμε τη διαδικασία ΠΛΑΓΙΑ_ΓΡΑΜΜΗ έτσι ώστε να δέχεται ως παραμέτρους την οριζόντια και την κατακόρυφη απόσταση τότε θα μπορούμε να αλλάζουμε και την κλίση της πλάγιας γραμμής.

```
TO ΠΛΑΓΙΑ_ΓΡΑΜΜΗ :X :Y
MAKE "ΥΡΟΤΕΙΝ SQRT((:Y * :Y) + (:X * :X))
MAKE "PLHQOS (:ΥΡΟΤΕΙΝ / 1)
MAKE "ΥΡΣΟΣ (:Y / :PLHQOS)
MAKE "PLATOS (:X / :PLHQOS)
REPEAT :PLHQOS
  [ ΕΥΘΕΙΑ :ΥΡΣΟΣ
    ΔΕΞΙΑ 90
    ΕΥΘΕΙΑ :PLATOS
    ΑΡΙΣΤΕΡΑ 90 ]
END
```

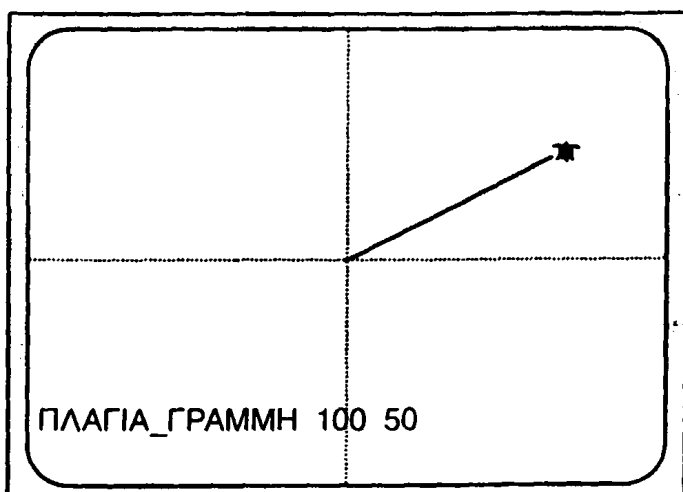


Πλάγιες γραμμές με διαφορετικές κλίσεις

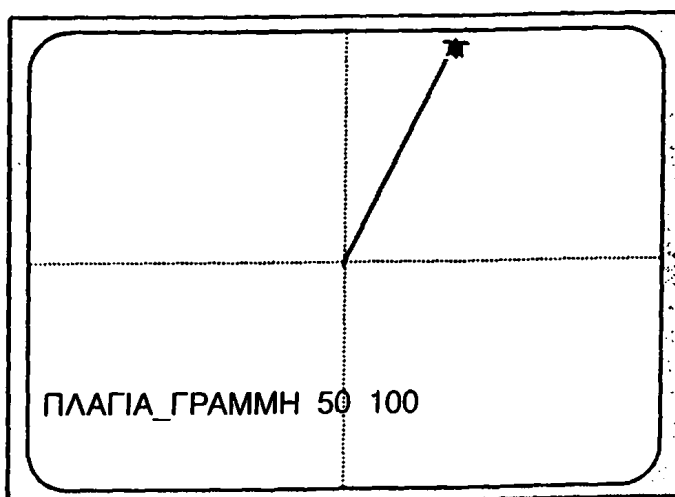
Ας πειραματιστούμε σχεδιάζοντας μερικές πλάγιες γραμμές που να έχουν μια ποικιλία οριζόντιων και κατακόρυφων αποστάσεων.



Η πλάγια ευθεία που έχει ίση την κατακόρυφη και την οριζόντια απόσταση έχει κλίση ίση με μισή ορθή γωνία.



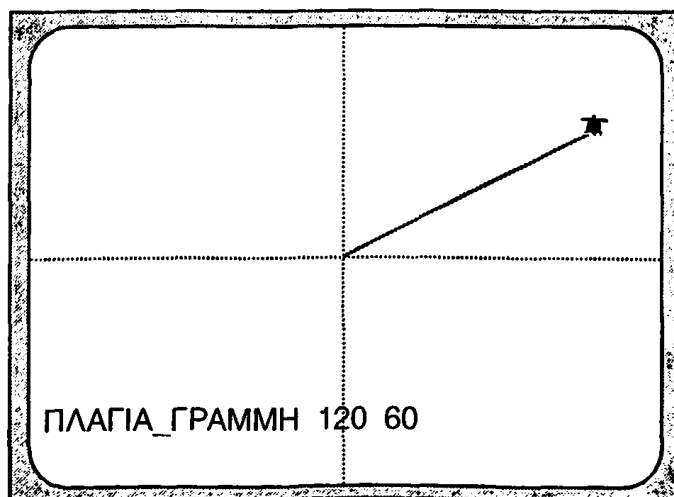
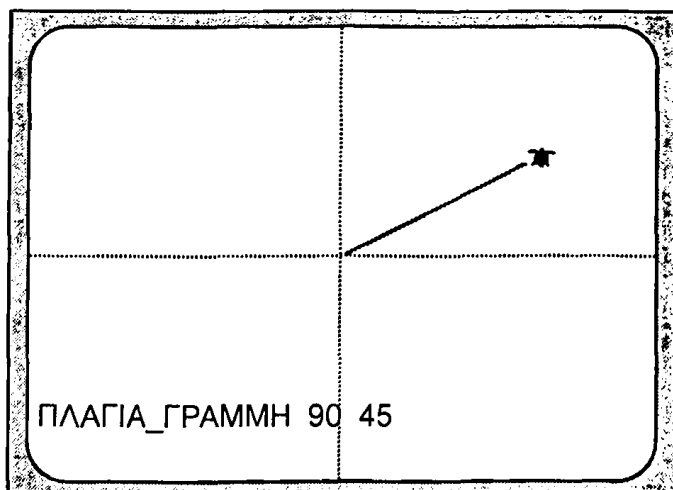
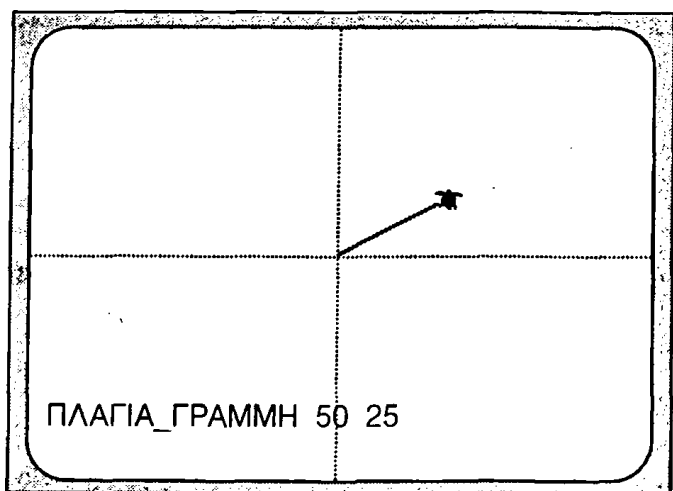
Οι πλάγιες ευθείες με παραμέτρους (100, 50) και (50, 100) ενώ έχουν το ίδιο μήκος, έχουν διαφορετική κλίση.



Γιατί αυτές οι δύο ευθείες έχουν το ίδιο μήκος;

Η αναλογία των αποστάσεων και η κλίση

Ας σχεδιάσουμε τρεις πλάγιες γραμμές με οριζόντιες και κατακόρυφες αποστάσεις (50, 25), (90, 45) και (120, 60) αντίστοιχα.

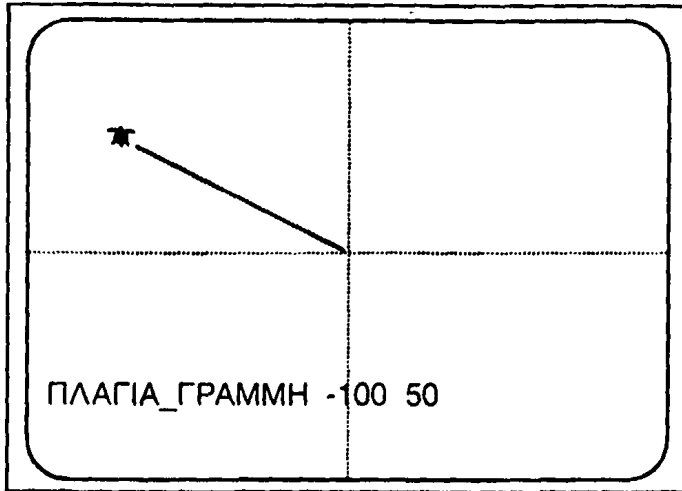


Οι τρεις πλάγιες γραμμές αν και έχουν διαφορετικά μήκη, έχουν την ίδια κλίση. Πρέπει να αναζητήσουμε τι είναι αυτό από τις παραμέτρους εισόδου που μένει σταθερό και στις τρεις περιπτώσεις. Παρατηρούμε ότι οι λόγοι $25/50$, $45/90$ και $60/120$ είναι ίσοι. Άρα η κλίση μας πλάγιας γραμμής εξαρτάται από το λόγο της κατακόρυφης ως προς την οριζόντια απόσταση. Αυτός ο λόγος εκφράζεται από τον τριγωνομετρικό αριθμό που λέγεται εφαπτομένη της γωνίας που σχηματίζει η πλάγια ευθεία με την οριζόντια. Η γωνία μπορεί να μετρηθεί (εκτός από μοίρες) και σε μια ακόμα μονάδα, το ακτίνιο. Η γωνία της συγκεκριμένης πλάγιας ευθείας είναι 0,5 ακτίνια.

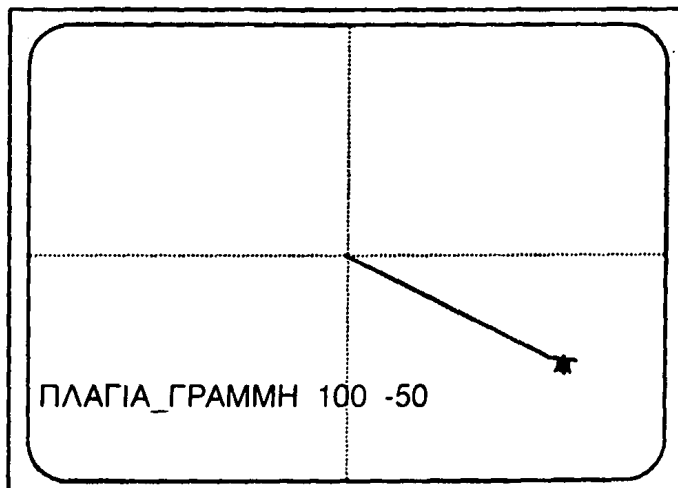


Μετρήσεις της γωνίας σε ακτίνια

Θα χαράξουμε τρεις ακόμα πλάγιες ευθείες και θα υπολογίσουμε τη γωνία της κλίσης τους σε ακτίνια.

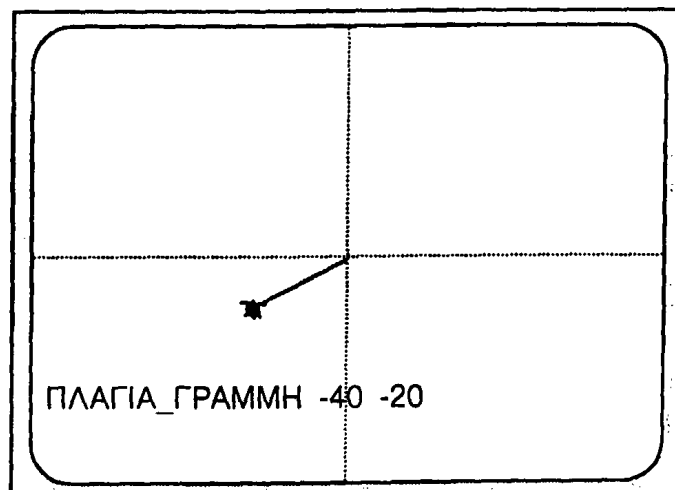


Η γωνία της κλίσης υπολογίζεται από το λόγο της κατακόρυφης προς την οριζόντια απόσταση. Έτσι η κλίση της πλάγιας ευθείας με παραμέτρους -100, 50 θα είναι $50/(-100) = -0,5$



Η κλίση της πλάγιας ευθείας με παραμέτρους 100, -50 είναι $(-50)/100 = -0,5$

Οι δύο προηγούμενες πλάγιες ευθείες έχουν την ίδια κλίση.



Η κλίση της πλάγιας ευθείας με παραμέτρους -40, -20 είναι $(-20)/(-40) = 0,5$

Μισό πολύγωνο

Η διαδικασία

```
ΤΟ ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
```

```
ΑΡΙΣΤΕΡΑ 90
```

```
MAKE "MOIRES ( 360 / :PLHQOS )
```

```
REPEAT :PLHQOS
```

```
  [ ΓΡΑΜΜΗ :MOIRES :PLEYRA ]
```

```
ΔΕΞΙΑ 90
```

```
END
```

σχεδιάζει ένα πολύγωνο.

Για να σχηματισθεί μόνο το μισό πολύγωνο θα πρέπει να σχεδιάσουμε μόνο το μισό πλήθος των πλευρών του. Δηλαδή αντί η επανάληψη να γίνει όσες φορές είναι το πλήθος των πλευρών θα πρέπει να γίνει τις μισές.

Ας δούμε πως θα γράψουμε τη διαδικασία που σχεδιάζει μισό πολύγωνο.

```
ΤΟ ΜΙΣΟ_ΠΟΛΥ ΓΩΝΟ :PLEYRA :PLHQOS
```

```
ΑΡΙΣΤΕΡΑ 90
```

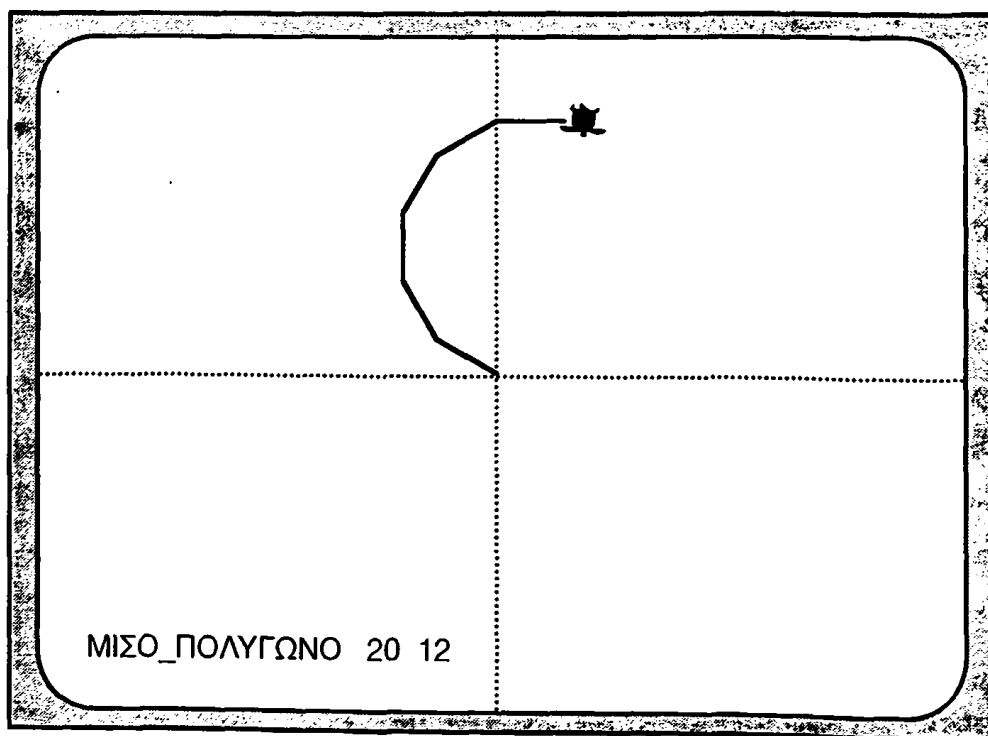
```
MAKE "MOIRES (360 / :PLHQOS)
```

```
REPEAT (:PLHQOS * (1 / 2))
```

```
  [ ΓΡΑΜΜΗ :MOIRES :PLEYRA ]
```

```
ΔΕΞΙΑ 90
```

```
END
```

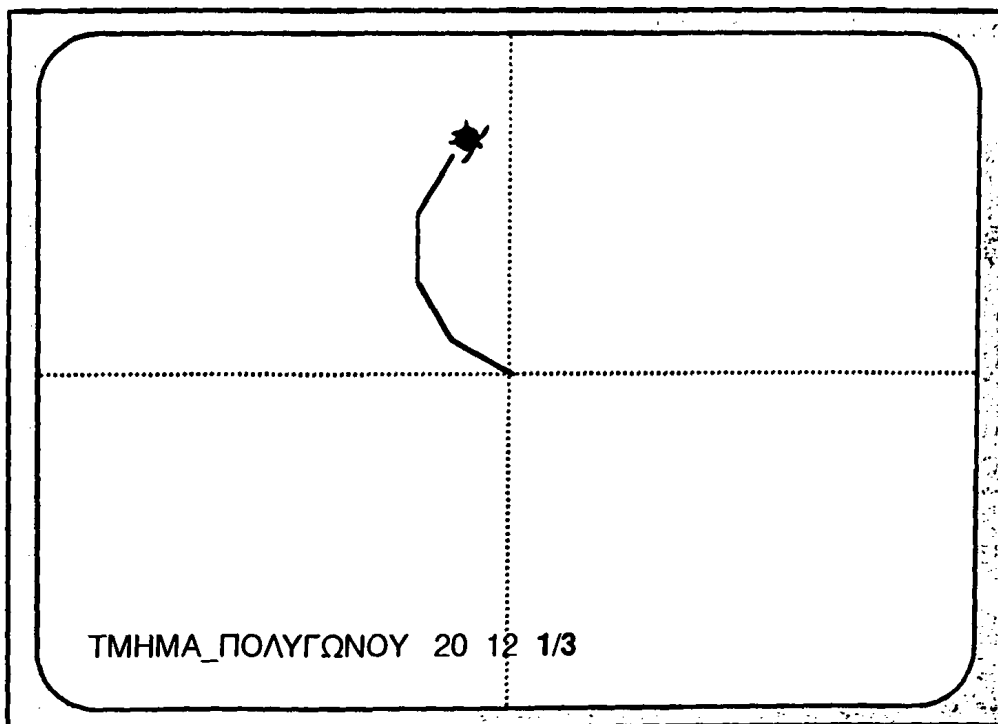


Ένα κομμάτι από ολόκληρο το πολύγωνο

Μπορούμε να σχεδιάσουμε κάποιο κλάσμα του πολυγώνου προχωρώντας στη γενίκευση με μια διαδικασία η οποία χρησιμοποιώντας μια παράμετρο επιπλέον, σχεδιάζει ένα τμήμα του πολυγώνου. Η επιπλέον αυτή παράμετρος θα είναι η μεταβλητή ΚΛΑΣΜΑ που θα αντιπροσωπεύει το κλάσμα του πολυγώνου που θέλουμε να σχεδιάσουμε.

```
ΤΟ ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ :PLEYRA :PLHQOS :KLASMA
ΑΡΙΣΤΕΡΑ 90
ΜΑΚΕ "ΜΟΙΡΕΣ (360 / :PLHQOS)
ΡΕΡΕΑΤ (:PLHQOS * :KLASMA)
    [ ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA ]
ΔΕΞΙΑ 90
ΕΝΔ
```

Να μια μέθοδος αναπαράστασης "γραμματικής" μορφής για τη διδασκαλία της έννοιας κλάσμα.



2/3, 3/3 και 4/3

Ας σχεδιάσουμε μερικά τμήματα πολυγώνων που αντιστοιχούν στα κλάσματα 2/3, 3/3 και 4/3.

ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ 20 12 2/3

ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ 20 12 3/3

ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ 20 12 4/3

Τα 6/6 είναι
μεγαλύτερα
από τα 2/2;

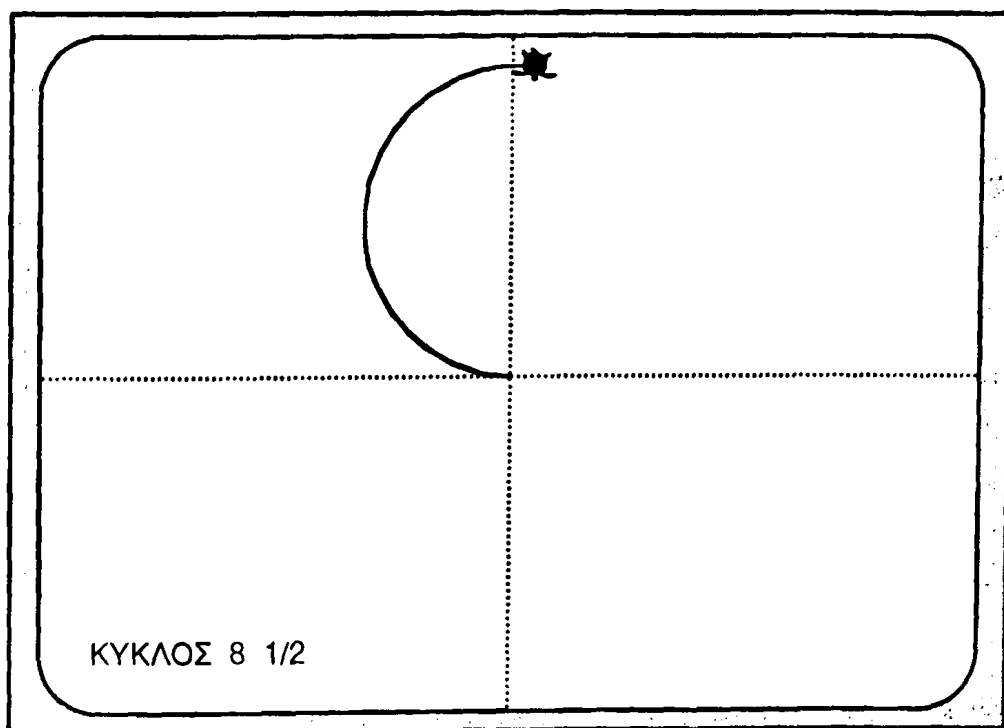
Ένα κομμάτι της περιφέρειας του κύκλου

Είχαμε δεχθεί ότι ο κύκλος θα σχεδιάζεται ως ένα πολύγωνο με 36 πλευρές. Αφού λοιπόν ο κύκλος "είναι" πολύγωνο, μπορούμε να εφαρμόσουμε και σ' αυτόν τα σχετικά με τα τμήματα των πολυγώνων. Έτσι θα μπορούμε να σχεδιάσουμε ημικύκλια, τεταρτοκύκλια και άλλα τμήματα του κύκλου.

Η τροποποίηση που πρέπει να γίνει στη διαδικασία ΚΥΚΛΟΣ
ΤΟ ΚΥΚΛΟΣ :PLEYRA
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360/36)
REPEAT 36
[ΓΡΑΜΜΗ :MOIRES :PLEYRA]
ΔΕΞΙΑ 90
END

είναι να συμπληρωθεί με τη μεταβλητή ΚΛΑΣΜΑ και να χρησιμοποιηθεί στο σημείο που καθορίζεται το πλήθος των επαναλήψεων.

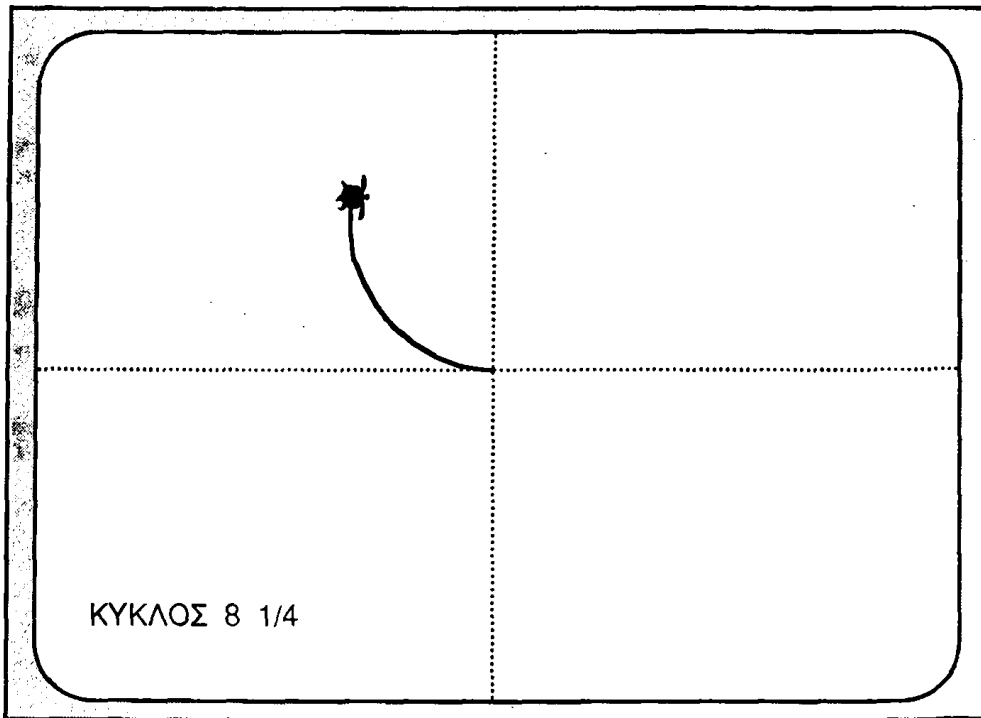
ΤΟ ΚΥΚΛΟΣ :PLEYRA:ΚΛΑΣΜΑ
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / 36)
REPEAT (36 * :ΚΛΑΣΜΑ)
[ΓΡΑΜΜΗ :MOIRES :PLEYRA]
ΔΕΞΙΑ 90
END



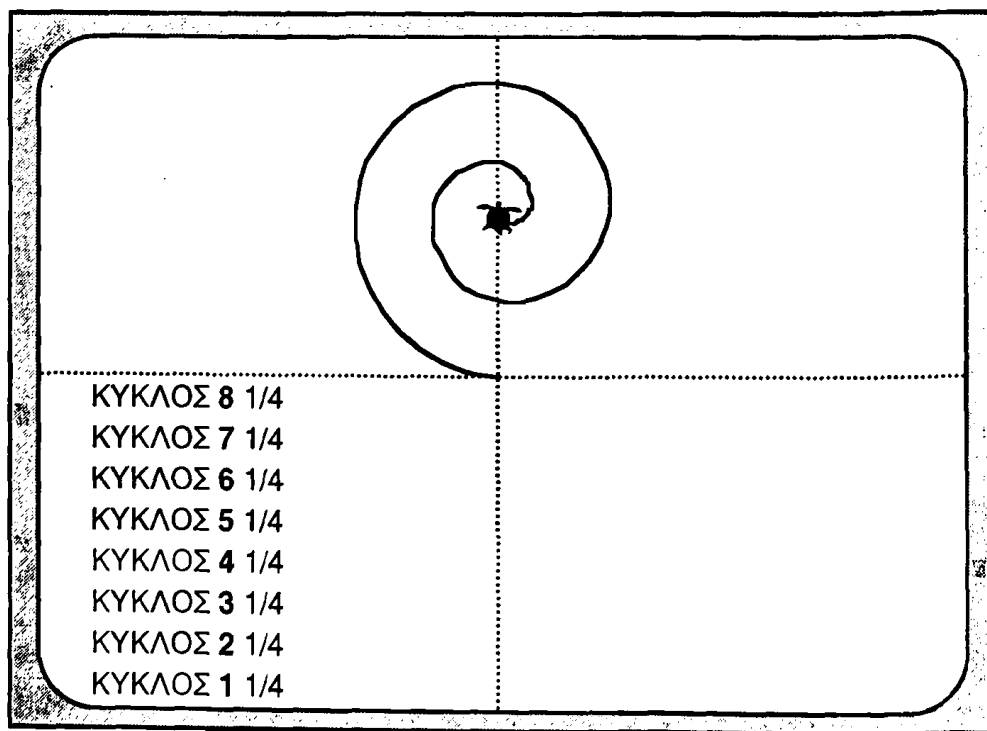
ημικύκλιο

Τεταρτοκύκλιο και κοχλίας

Χρησιμοποιώντας τη νέα έκδοση της διαδικασίας ΚΥΚΛΟΣ, μπορούμε να σχεδιάσουμε ένα τεταρτοκύκλιο δίνοντας απλά ως τιμή στην παράμετρο ΚΛΑΣΜΑ το $1/4$.

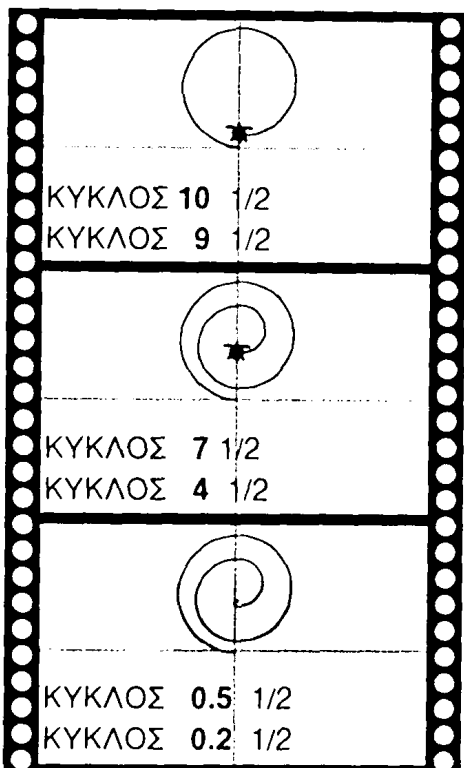


Κρατώντας σταθερό το κλάσμα και μειώνοντας την τιμή της μεταβλητής PLEΥΡΑ μπορούμε με διαδοχικές κλήσεις της διαδικασίας ΚΥΚΛΟΣ να σχηματίσουμε έναν κοχλία.

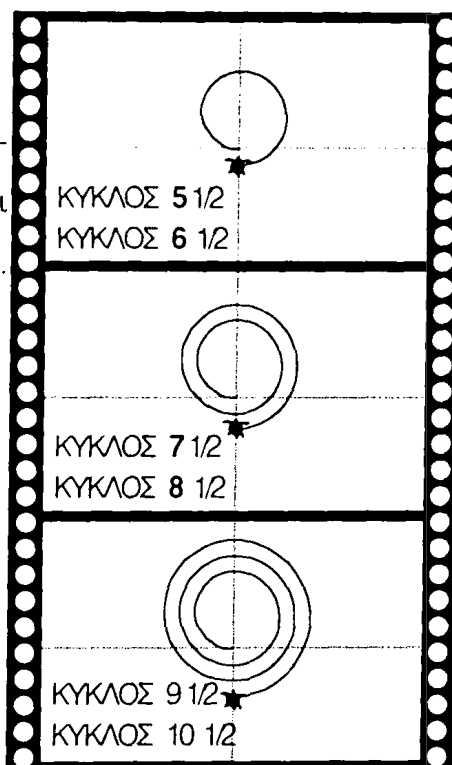


Κοχλίας από ημικύκλια

Ας χρησιμοποιήσουμε ημικύκλια αντί για τεταρτοκύκλια για τη δημιουργία ενός κοχλία.



Με αύξηση του μεγέθους της πλευράς, ο "εσωστρεφής" κοχλίας γίνεται "εξωστρεφής".



Το πρόγραμμα του κοχλίας

Ας γράψουμε μια διαδικασία που να σχεδιάζει έναν κοχλία. Ως παραμέτρους θα δέχεται το μέγεθος, το πλήθος των στροφών, το αν θα είναι εσωστρεφής ή εξωστρεφής και το κλάσμα του κύκλου που θα χρησιμοποιηθεί.

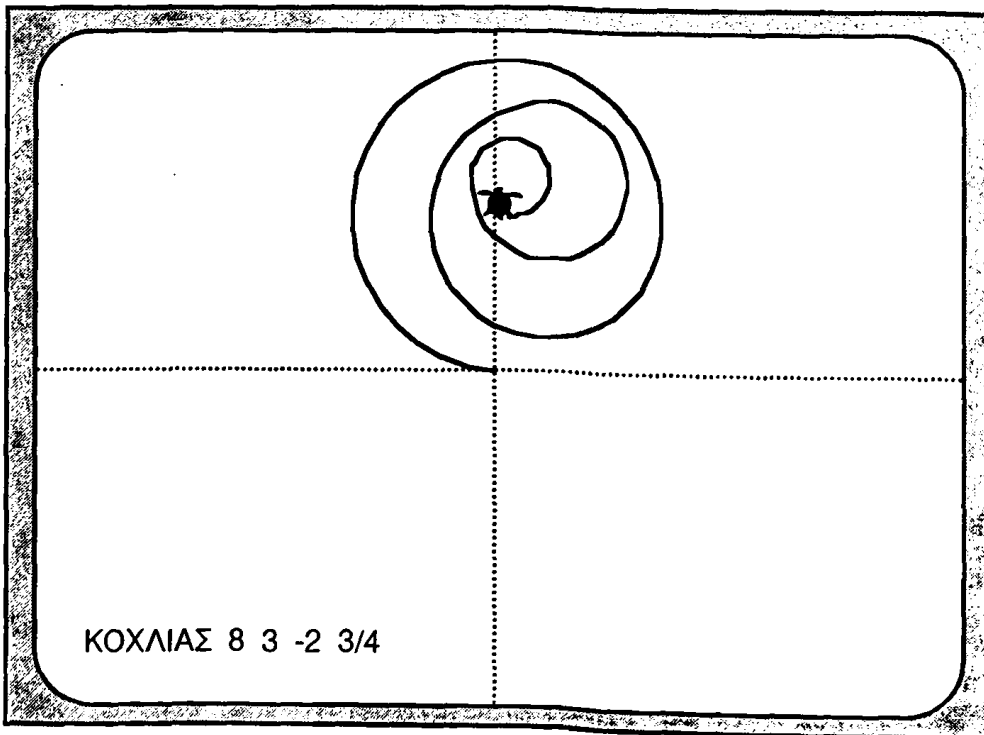
```
TO KOXLIAS :MEG :STROFH :ESEX :KLASMA
MAKE "FORES (:STROFH / :KLASMA)
REPEAT :FORES
  [ KYKLOS :MEG :KLASMA
    MAKE "MEG :MEG + :ESEX ]
END
```

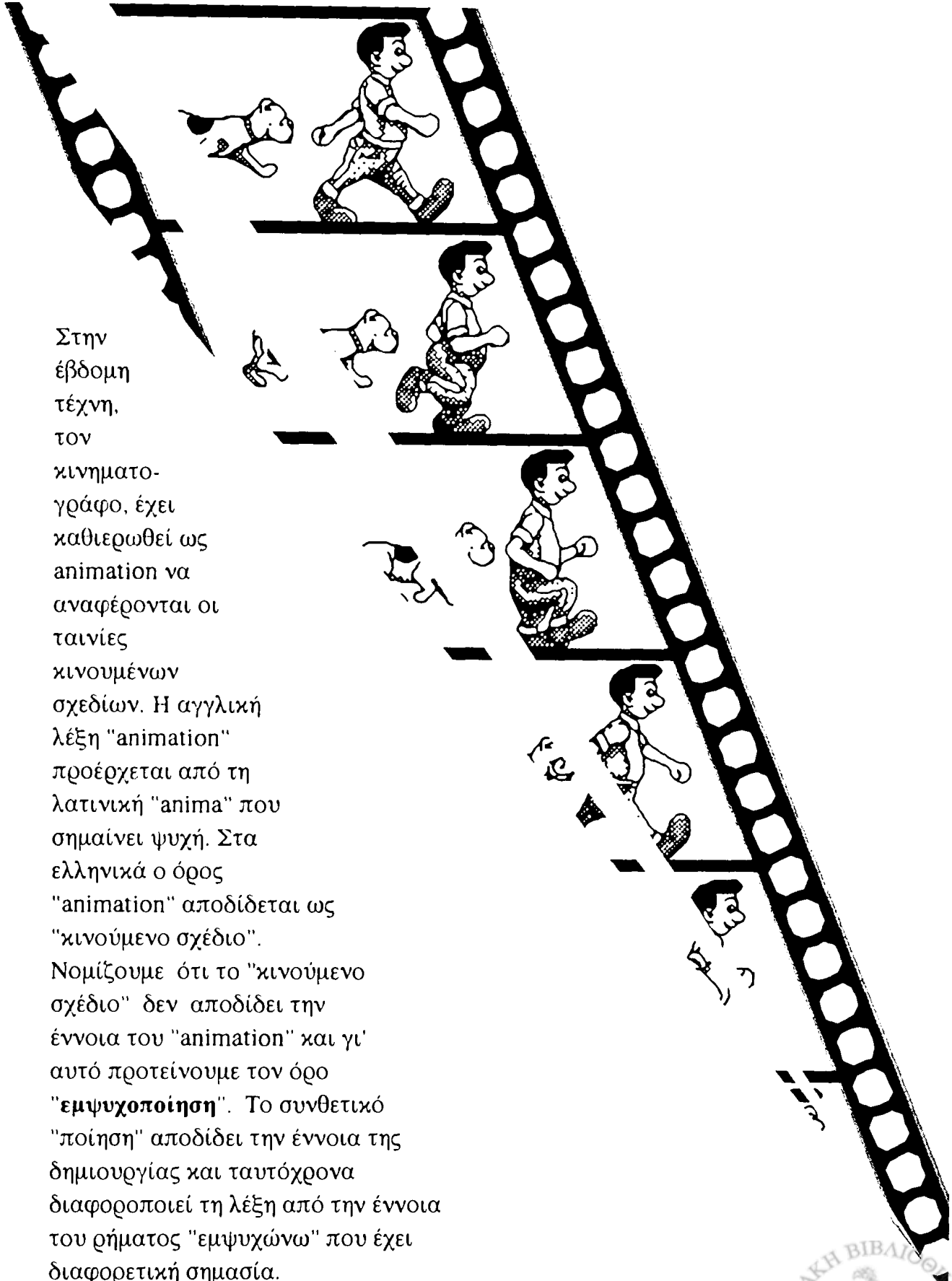
Μπορείτε να μαντέψετε τι σχήματα θα προκύψουν από τις εξής κλήσεις της διαδικασίας ΚΟΧΛΙΑΣ;

ΚΟΧΛΙΑΣ 5 4 -1 1/2

ΚΟΧΛΙΑΣ -3 3 2 1/2

ΚΟΧΛΙΑΣ 3 3 2 -1/2





Στην
έβδομη
τέχνη,
τον
κινηματο-
γράφο, έχει
καθιερωθεί ως
animation να
αναφέρονται οι
ταινίες
κινουμένων
σχεδίων. Η αγγλική
λέξη "animation"
προέρχεται από τη
λατινική "anima" που
σημαίνει ψυχή. Στα
ελληνικά ο όρος
"animation" αποδίδεται ως
"κινούμενο σχέδιο".
Νομίζουμε ότι το "κινούμενο
σχέδιο" δεν αποδίδει την
έννοια του "animation" και γι'
αυτό προτείνουμε τον όρο
"εμπυχοποίηση". Το συνθετικό
"ποίηση" αποδίδει την έννοια της
δημιουργίας και ταυτόχρονα
διαφοροποιεί τη λέξη από την έννοια
του ρήματος "εμπυχώνω" που έχει
διαφορετική σημασία.

11 Animation / Εμ-ψυχο-ποίηση *

Η αίσθηση ότι το τετράγωνο ολισθαίνει

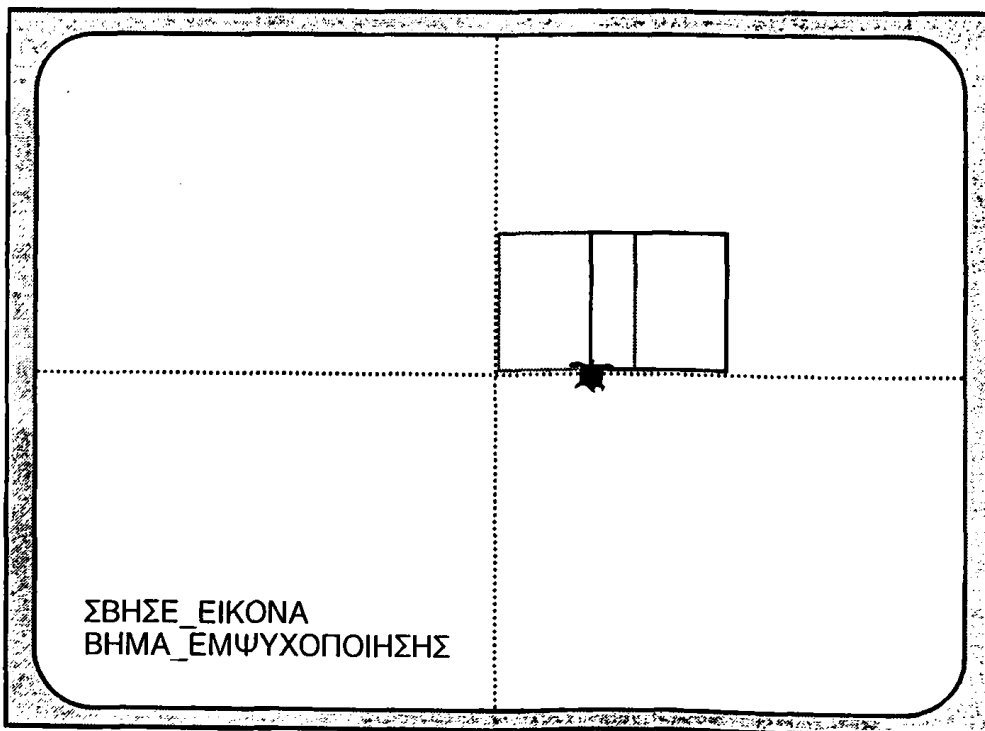
Ένα σχήμα μπορούμε να το σβήσουμε αν διατάξουμε τη χελώνα να το διατρέξει αλλά αν αντί να γράφει, να σβήνει. Η κατάσταση με την οποία η χελώνα σβήνει αντί να γράφει, υλοποιείται με την εντολή Pen Erase (που σημαίνει η πένα σβήνει) ή σε συντομογραφία PE. Μια "εξελληνισμένη" διαδικασία που θα πραγματοποιεί το ίδιο έργο με την εντολή Pen Erase είναι η διαδικασία ΣΒΗΣΕ.

```
ΤΟ ΣΒΗΣΕ  
PE  
END
```

Μπορούμε να δημιουργήσουμε την αίσθηση ότι ένα σχήμα που σχεδιάστηκε στην οθόνη κινείται. Αυτό το πετυχαίνουμε αν αφού σχεδιάσουμε το σχήμα το σβήσουμε, και στη συνέχεια το ξανασχεδιάσουμε μετατοπισμένο.

Αυτό κάνει η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ:

```
ΤΟ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ  
ΠΟΛΥΓΩΝΟ 40 4  
ΣΒΗΣΕ  
ΠΟΛΥΓΩΝΟ 40 4  
ΜΕΤΑΤΟΠΙΣΗ 10 0  
ΓΡΑΨΕ  
ΠΟΛΥΓΩΝΟ 40 4  
END
```



* Στην έβδομη τέχνη, τον κινηματογράφο, έχει καθιερωθεί ως animation να αναφέρονται οι ταινίες κινουμένων σχεδίων. Η αγγλική λέξη "animation" προέρχεται από τη λατινική "anima" που σημαίνει ψυχή. Στα ελληνικά ο όρος "animation" αποδίδεται ως "κινούμενο σχέδιο". Νομίζουμε ότι το "κινούμενο σχέδιο" δεν αποδίδει την έννοια του "animation" και γι' αυτό προτείνουμε τον όρο "εμψυχοποίηση". Το συνθετικό "ποίηση" αποδίδει την έννοια της δημιουργίας και ταυτόχρονα διαφοροποιεί τη λέξη από την έννοια του ρήματος "εμψυχώνω" που έχει διαφορετική σημασία.



Κίνηση τετραγώνου προς κάθε κατεύθυνση

Βελτιώνουμε τη διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ βάζοντας της ως παραμέτρους την οριζόντια και την κατακόρυφη απόσταση που θα μετατοπισθεί το τετράγωνο.

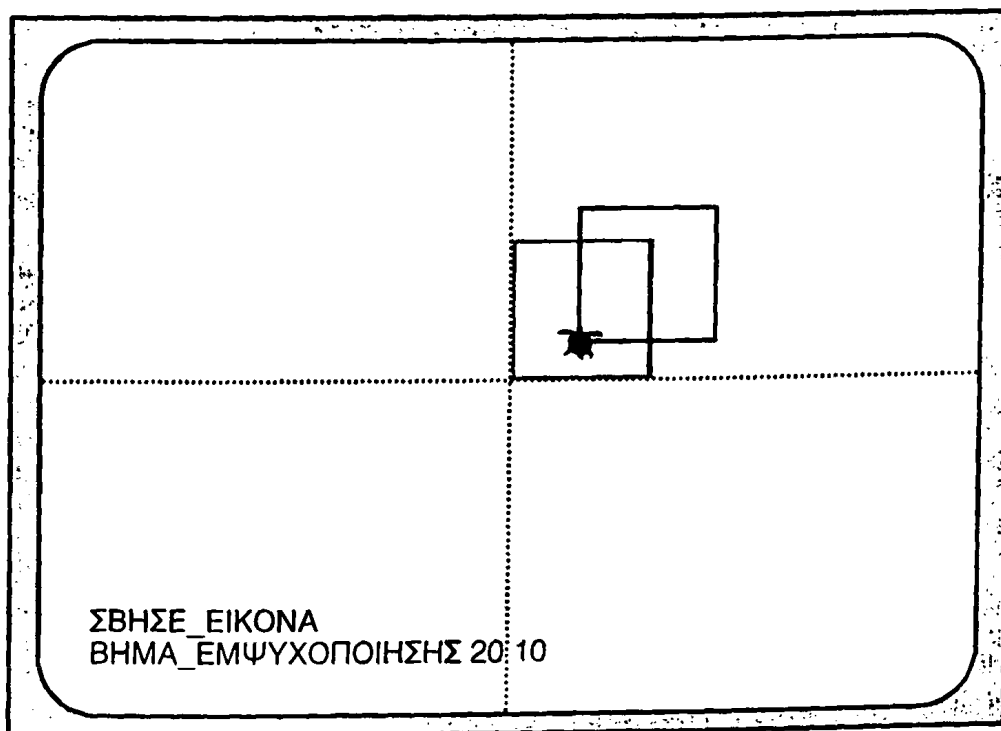
Η μεταβλητή που θα αντιπροσωπεύει την οριζόντια συνιστώσα της μετατόπισης ονομάζεται DX και η αντίστοιχη που θα αντιπροσωπεύει την κατακόρυφη DY .

Έτσι η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ γίνεται:

```
ΤΟ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY  
ΠΟΛΥΓΩΝΟ 40 4
```

```
ΣΒΗΣΕ  
ΠΟΛΥΓΩΝΟ 40 4
```

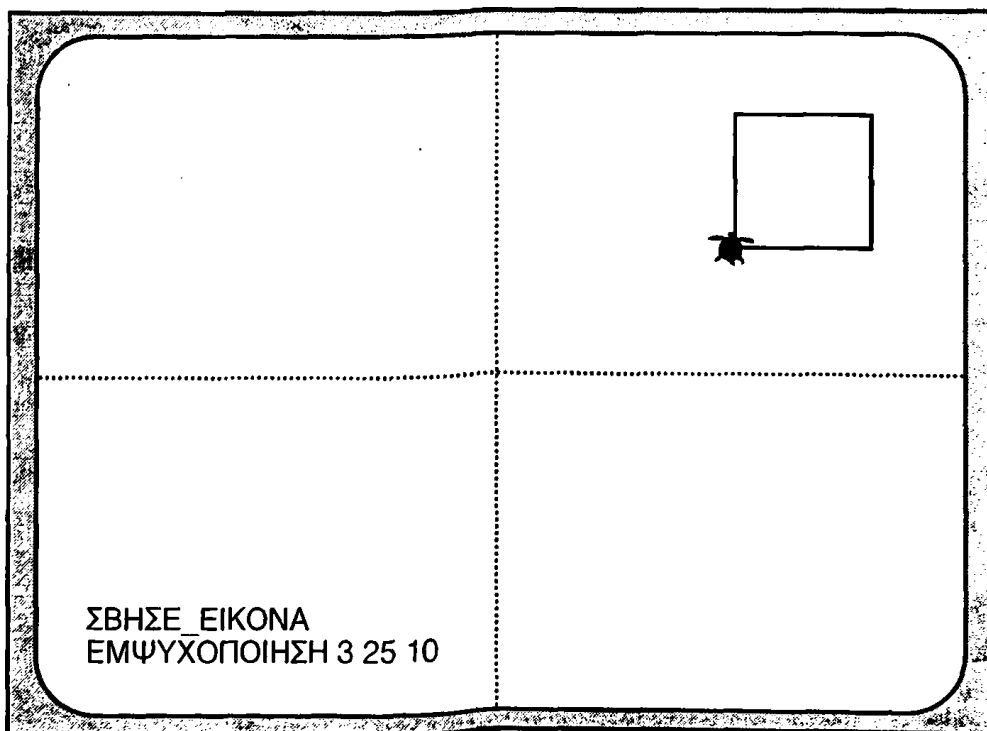
```
ΜΕΤΑΤΟΠΙΣΗ :DX :DY  
ΓΡΑΨΕ  
ΠΟΛΥΓΩΝΟ 40 4  
ΕΝΔ
```



Ολοκληρωμένη κίνηση τετραγώνου

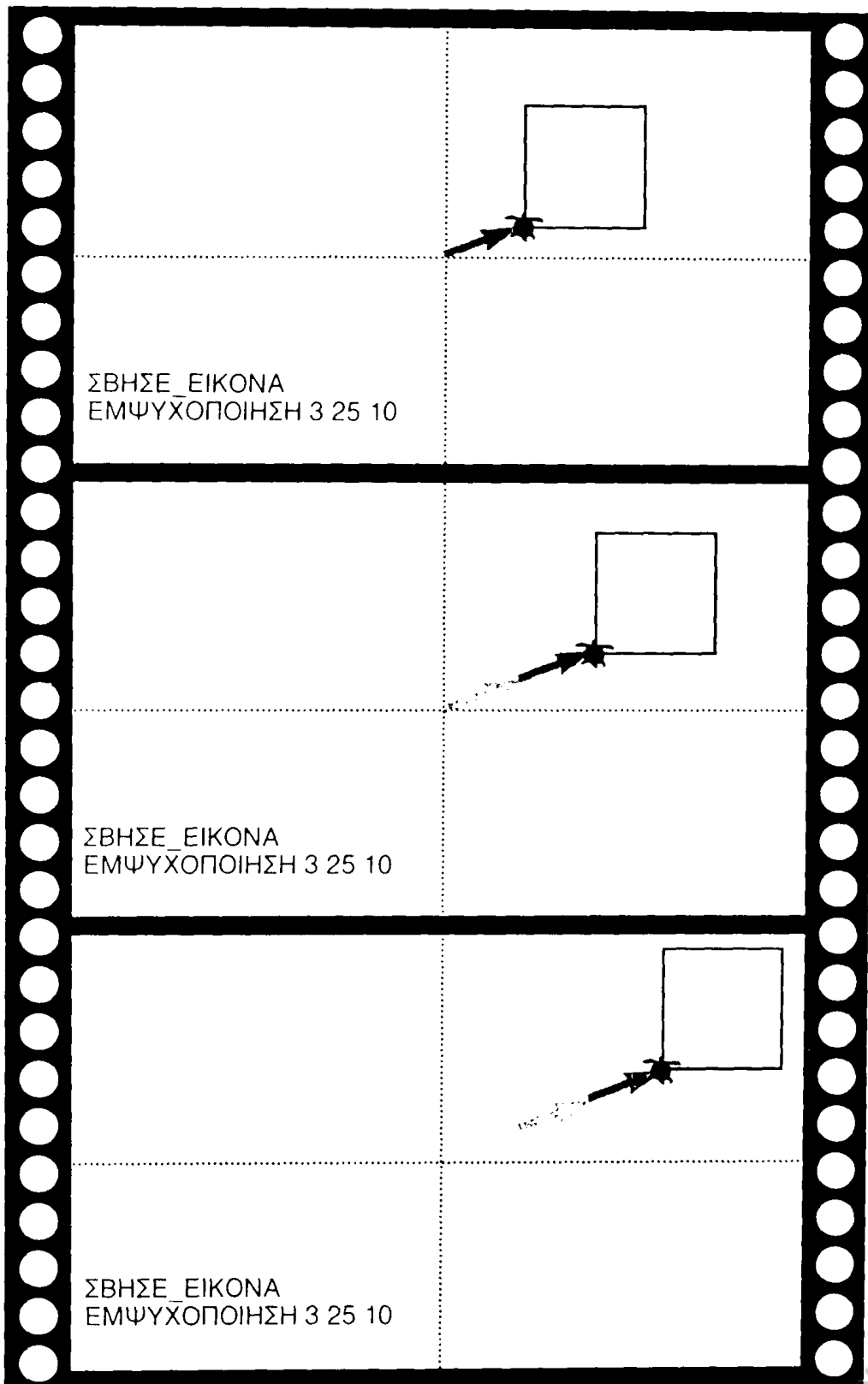
Η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ πραγματοποιεί μόνο ένα βήμα. Μια διαδικασία που επαναλαμβάνει πολλές φορές τη διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ θα δημιουργεί την αίσθηση μιας ολοκληρωμένης κίνησης. Την αίσθηση αυτή τη δημιουργεί η επόμενη διαδικασία, ΕΜΨΥΧΟΠΟΙΗΣΗ. Η μεταβλητή ΒΗΜΑΤΑ καθορίζει το πόσες φορές θα επαναληφθεί η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ.

```
TO ΕΜΨΥΧΟΠΟΙΗΣΗ :ΒΗΜΑΤΑ :DX :DY
  REPEAT :ΒΗΜΑΤΑ
    [ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY ]
  END
```



Τα διαδοχικά βήματα της κίνησης

Ας παρακολουθήσουμε τα τρία διαδοχικά βήματα που θα πραγματοποιήσει η κλήση της διαδικασίας ΕΜΨΥΧΟΠΟΙΗΣΗ 3 25 10.



Μια ταχύτερη κίνηση

Η εμφάνιση της χελώνας στην οθόνη καθώς κινείται για να σχηματίσει ή να σβήσει το τετράγωνο, κάνει την κίνηση αργή. Μπορούμε με την εντολή **Hide Turtle** (κρύψε τη χελώνα) ή σε συντομογραφία **HT** να εξαφανίσουμε την εικόνα της χελώνας από την οθόνη. Μια "εξελληνισμένη" διαδικασία που θα πραγματοποιεί το ίδιο έργο με την εντολή **Hide Turtle**, είναι η διαδικασία **ΚΡΥΨΟΥ** που ορίζεται ως:

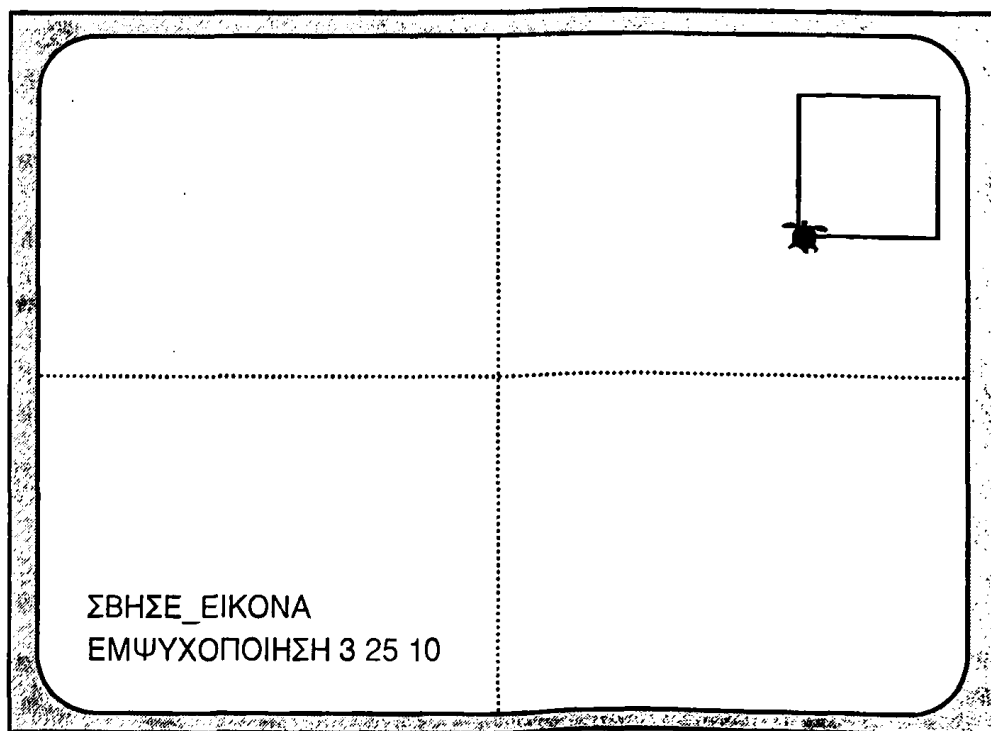
```
TO ΚΡΥΨΟΥ  
HT  
END
```

Η εντολή που επανεμφανίζει τη χελώνα στην οθόνη είναι η **Show Turtle** (εμφάνισε τη χελώνα) ή σε συντομογραφία **ST**. Μια "εξελληνισμένη" διαδικασία για το ίδιο έργο είναι η **ΒΓΕΣ**.

```
TO ΒΓΕΣ  
ST  
END
```

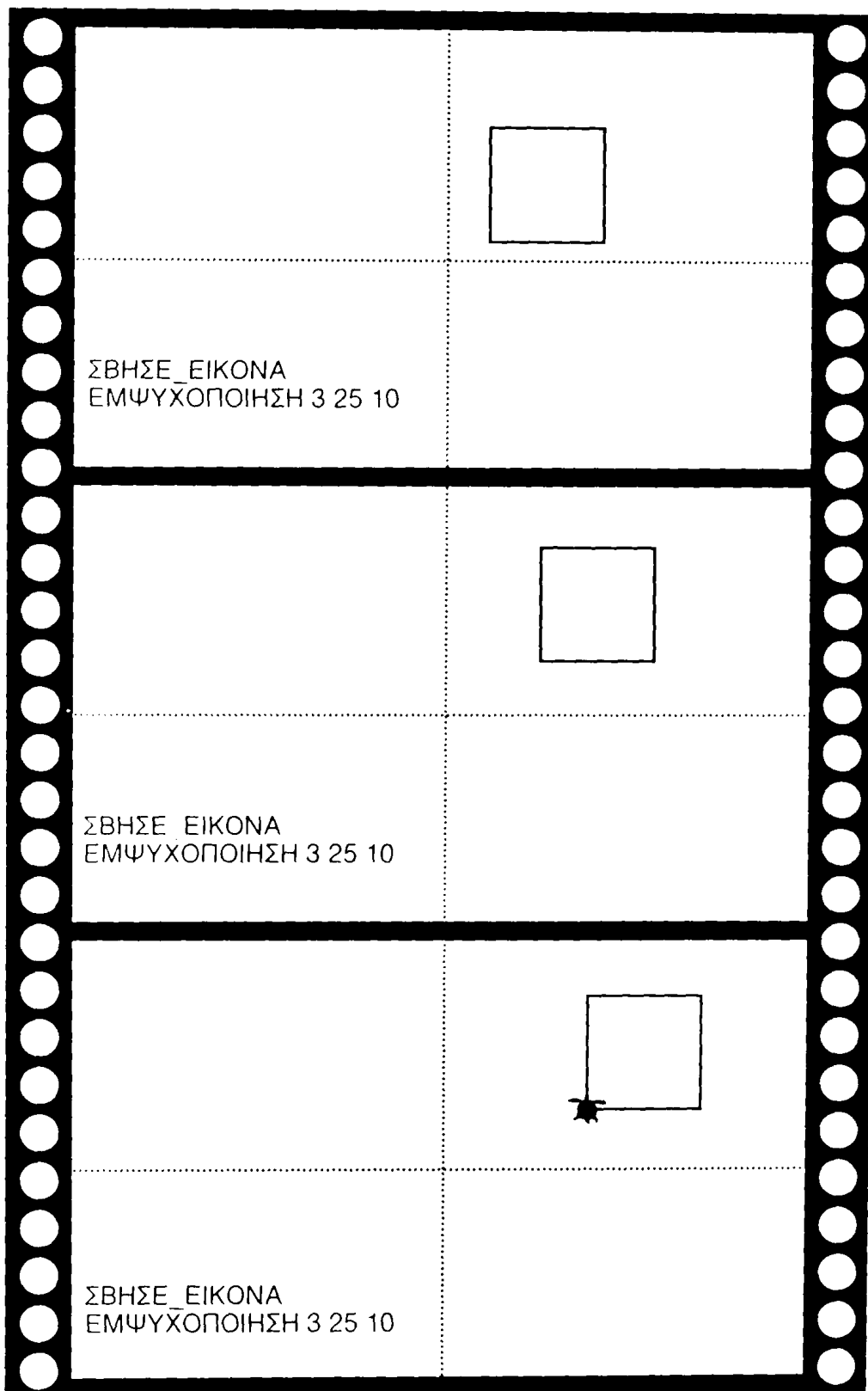
Με τη βοήθεια της διαδικασίας **ΚΡΥΨΟΥ**, η διαδικασία **ΕΜΨΥΧΟΠΟΙΗΣΗ** τροποποιείται και γίνεται ταχύτερη.

```
TO ΕΜΨΥΧΟΠΟΙΗΣΗ :ΒΗΜΑΤΑ :DX :DY  
ΚΡΥΨΟΥ  
REPEAT :ΒΗΜΑΤΑ  
  [ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY]  
ΒΓΕΣ  
END
```



Διαδοχικά βήματα της ταχύτερης κίνησης

Η γελώνα εμφανίζεται μετά την ολοκλήρωση της κίνησης και έτσι η προσομοίωση της κίνησης είναι ταχύτερη και πιο ρεαλιστική.

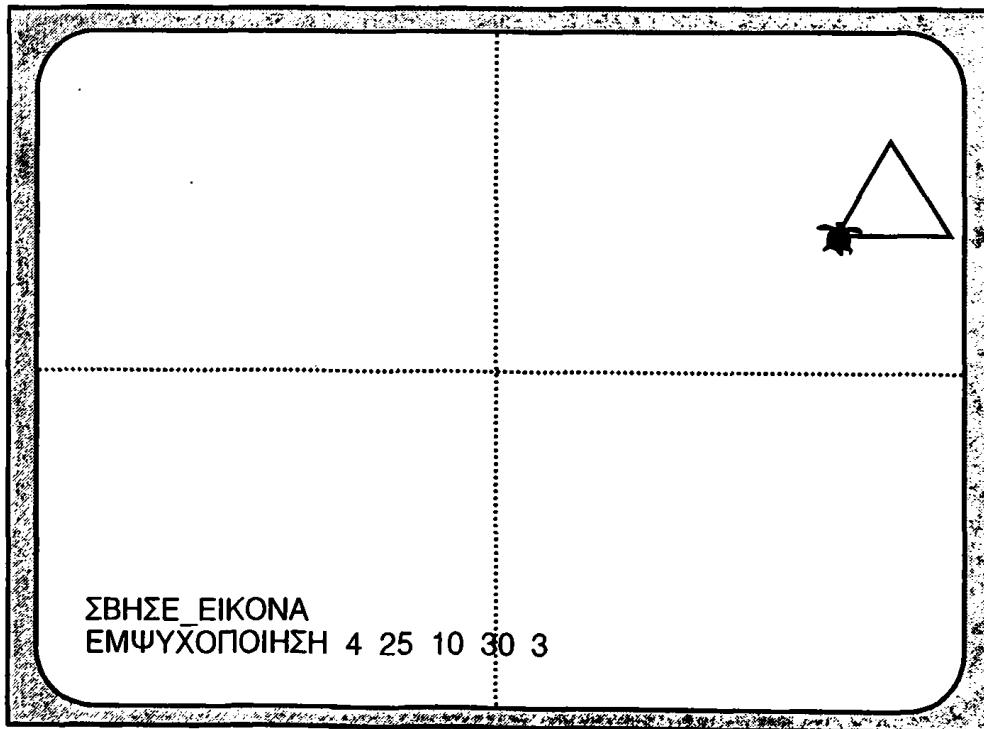


Ας επιλέξουμε το σχήμα που θα κινήσουμε

Μέχρι τώρα, το σχήμα που κινούσε η διαδικασία ΕΜΨΥΧΟ-ΠΟΙΗΣΗ ήταν ένα τετράγωνο με πλευρά 40. Μπορούμε να προσθέσουμε δυο παραμέτρους στις διαδικασίες ΒΗΜΑ_ΕΜΨΥΧΟ-ΠΟΙΗΣΗΣ και ΕΜΨΥΧΟΠΟΙΗΣΗ, που θα καθορίζουν το πλήθος των πλευρών του πολυγώνου (PLHQOS / PLHQ) καθώς και το μήκος της πλευράς του (PLEYRA / PLEYR).

```
TO ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY :PLEYRA :PLHQOS
ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
ΣΒΗΣΕ
ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
ΜΕΤΑΤΟΠΙΣΗ :DX :DY
ΓΡΑΨΕ
ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
END
```

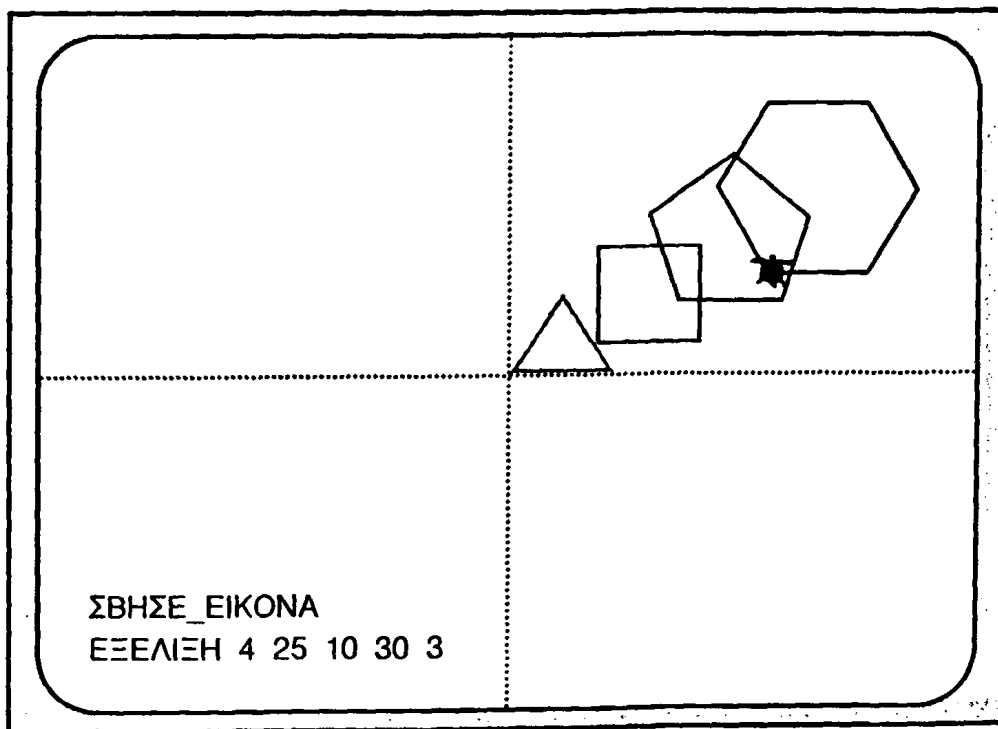
```
TO ΕΜΨΥΧΟΠΟΙΗΣΗ :ΒΗΜΑΤΑ :DX :DY :PLEYRA :PLHQOS
ΚΡΥΨΟΥ
REPEAT :ΒΗΜΑΤΑ
  [ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY :PLEYRA :PLHQOS]
ΒΓΕΣ
END
```



Οι γωνίες αυξάνονται

Μπορούμε να χρησιμοποιήσουμε τη δομή της διαδικασίας ΕΜΨΥΧΟΠΟΙΗΣΗ για να παρουσιάσουμε σε κάθε διαδοχικό βήμα και ένα πολύγωνο που να διαθέτει μια επιπλέον πλευρά από αυτές που είχε στο προηγούμενο βήμα. Αυτό γίνεται στην επόμενη διαδικασία ΕΞΕΛΙΞΗ όπου μέσα στην επανάληψη η τιμή της μεταβλητής PLHQ που αντιπροσωπεύει το πλήθος των πλευρών του πολυγώνου αυξάνει κατά μια μονάδα.

```
TO ΕΞΕΛΙΞΗ :ΒΗΜΑΤΑ :DX :DY :PLEYRA :PLHQOS
ΚΡΥΨΟΥ
REPEAT :ΒΗΜΑΤΑ
  [ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY :PLEYRA :PLHQOS
  MAKE "PLHQOS :PLHQOS + 1]
ΒΓΕΣ
END
```



Το διαστημικό πρόγραμμα της logo

Με την τεχνογνωσία που έχουμε αποκτήσει, μπορούμε να αναπτύξουμε το δικό μας διαστημικό πρόγραμμα. Πρώτα ας κατασκευάσουμε τον πύραυλο.

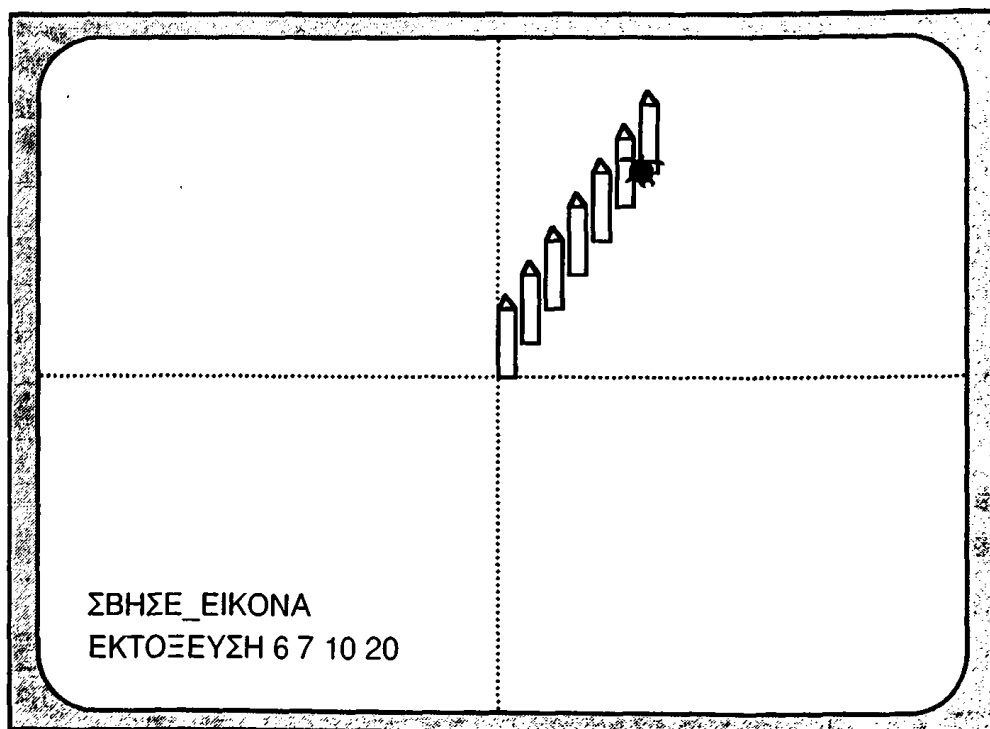
```
ΤΟ ΠΥΡΑΥΛΟΣ :MEGEQOS
ΟΡΘΟΓΩΝΙΟ :MEGEQOS 4
ΕΥΘΕΙΑ :MEGEQOS
ΠΟΛΥΓΩΝΟ :MEGEQOS / 4 3
ΟΠΙΣΘΕΝ :MEGEQOS
END
```

Ας δοκιμάσουμε να μετακινήσουμε τον πύραυλο ένα βήμα.

```
ΤΟ ΒΗΜΑ_ΠΥΡΑΥΛΟΥ :DX :DY :MEGEQOS
ΠΥΡΑΥΛΟΣ :MEGEQOS
ΣΒΗΣΕ
ΠΥΡΑΥΛΟΣ :MEGEQOS
ΜΕΤΑΤΟΠΙΣΗ :DX :DY
ΓΡΑΨΕ
ΠΥΡΑΥΛΟΣ :MEGEQOS
END
```

Τέλος, ας προγραμματίσουμε την εκτόξευση του πυραύλου μας.

```
ΤΟ ΕΚΤΟΞΕΥΣΗ :ΒΗΜΑΤΑ :DX :DY :MEGEQOS
ΚΡΥΨΟΥ
REPEAT :ΒΗΜΑΤΑ [ΒΗΜΑ_ΠΥΡΑΥΛΟΥ :DX :DY :MEGEQOS]
ΒΓΕΣ
END
```



Ο χώρος του διαστημικού κέντρου

Ας σχεδιάσουμε τώρα το χώρο από όπου θα γίνεται η εκτόξευση.
Ας ξεκινήσουμε από το έδαφος με την ομώνυμη διαδικασία.

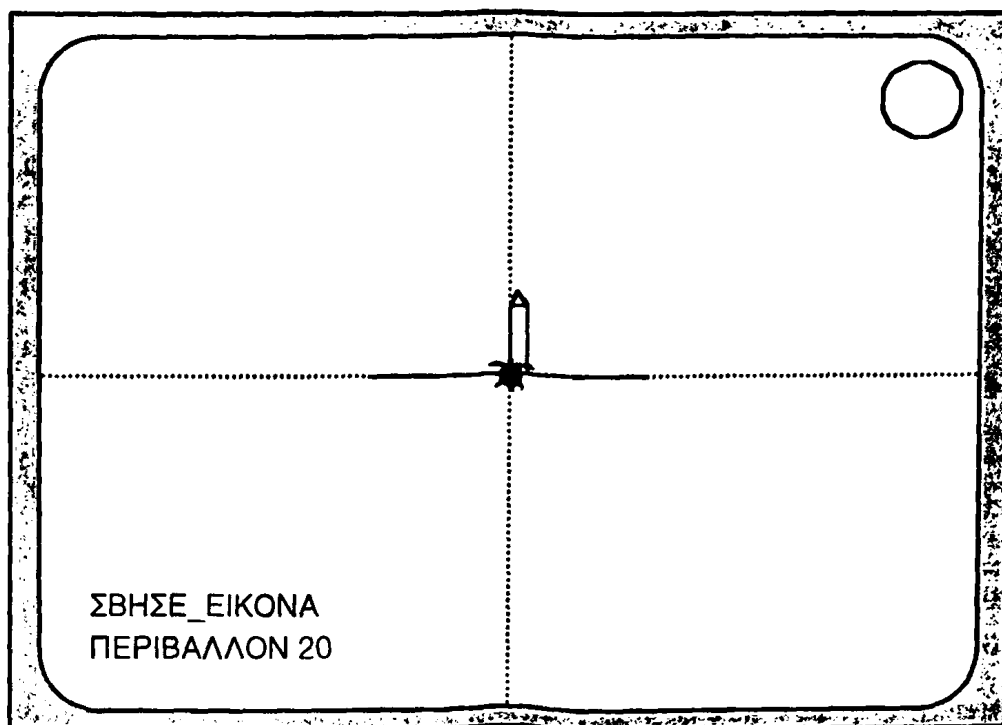
```
ΤΟ ΕΔΑΦΟΣ  
ΜΕΤΑΤΟΠΙΣΗ MINUS 40 0  
ΓΡΑΜΜΗ 90 80  
ΑΡΙΣΤΕΡΑ 90  
ΜΕΤΑΤΟΠΙΣΗ MINUS 40 0  
END
```

Ας σχεδιάσουμε τώρα τη σελήνη προς την οποία θα κατευθύνεται
ο πύραυλος.

```
ΤΟ ΣΕΛΗΝΗ  
ΜΕΤΑΤΟΠΙΣΗ 120 70  
ΚΥΚΛΟΣ 2 1  
ΜΕΤΑΤΟΠΙΣΗ MINUS 120 MINUS 70  
END
```

Έχοντας ορίσει τις διαδικασίες ΕΔΑΦΟΣ και ΣΕΛΗΝΗ, τις
συνδυάζουμε και σχεδιάζουμε το περιβάλλον όπου θα λάβει μέρος το
διαστημικό πρόγραμμα.

```
ΤΟ ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ  
ΕΔΑΦΟΣ  
ΣΕΛΗΝΗ  
ΠΥΡΑΥΛΟΣ :ΜΕΓΕΘΟΣ  
END
```

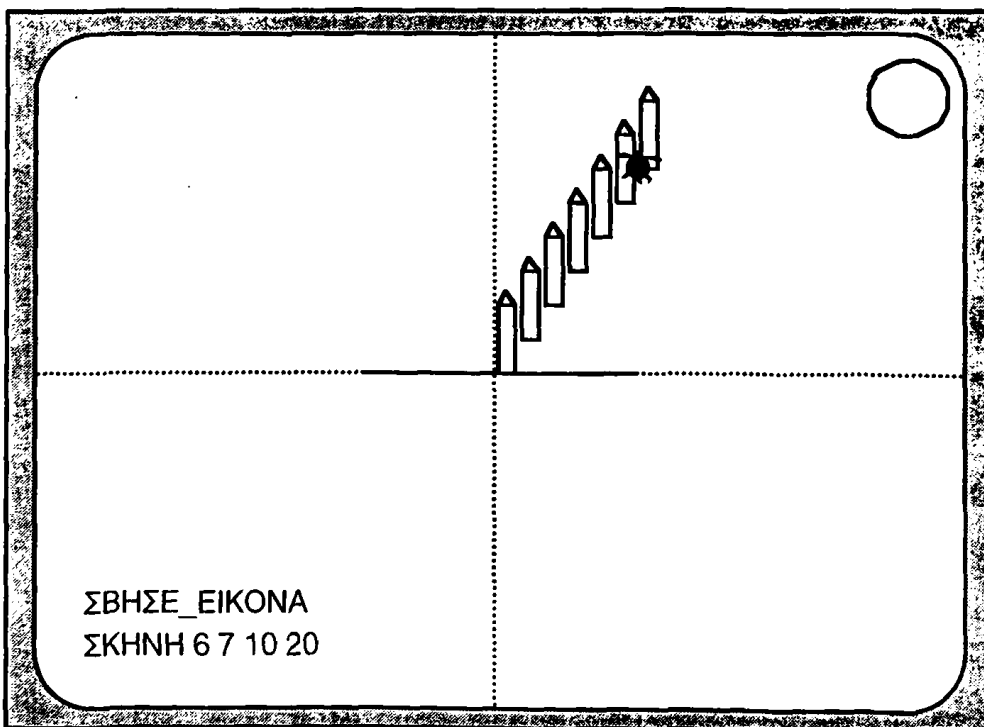


Δοκιμές της εκτόξευσης

Αφού έχουμε τη διαδικασία που περιγράφει την εκτόξευση του πυραύλου και αυτήν που σχεδιάζει το περιβάλλον από όπου θα γίνει η εκτόξευση, ας γράψουμε τη διαδικασία που θα παρουσιάζει τη σκηνή της εκτόξευσης.

```
ΤΟ ΣΚΗΝΗ :ΒΗΜΑΤΑ :DX :DY :ΜΕΓΕΘΟΣ  
ΚΡΥΨΟΥ  
ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ  
ΕΚΤΟΞΕΥΣΗ :ΒΗΜΑΤΑ :DX :DY :ΜΕΓΕΘΟΣ  
ΒΓΕΣ  
ΕΝΔ
```

Η μεταβλητή ΜΕΓΕΘΟΣ καθορίζει το μέγεθος του πυραύλου, οι μεταβλητές DX και DY καθορίζουν την οριζόντια και την κατακόρυφη μετατόπιση του πυραύλου σε κάθε του βήμα και η μεταβλητή ΒΗΜΑΤΑ το πλήθος των βημάτων (καρέ) που θα κάνει ο πύραυλος μέχρι να εξαντληθούν τα καύσιμα του και να σταματήσει.



Ένα ρεαλιστικότερο περιβάλλον

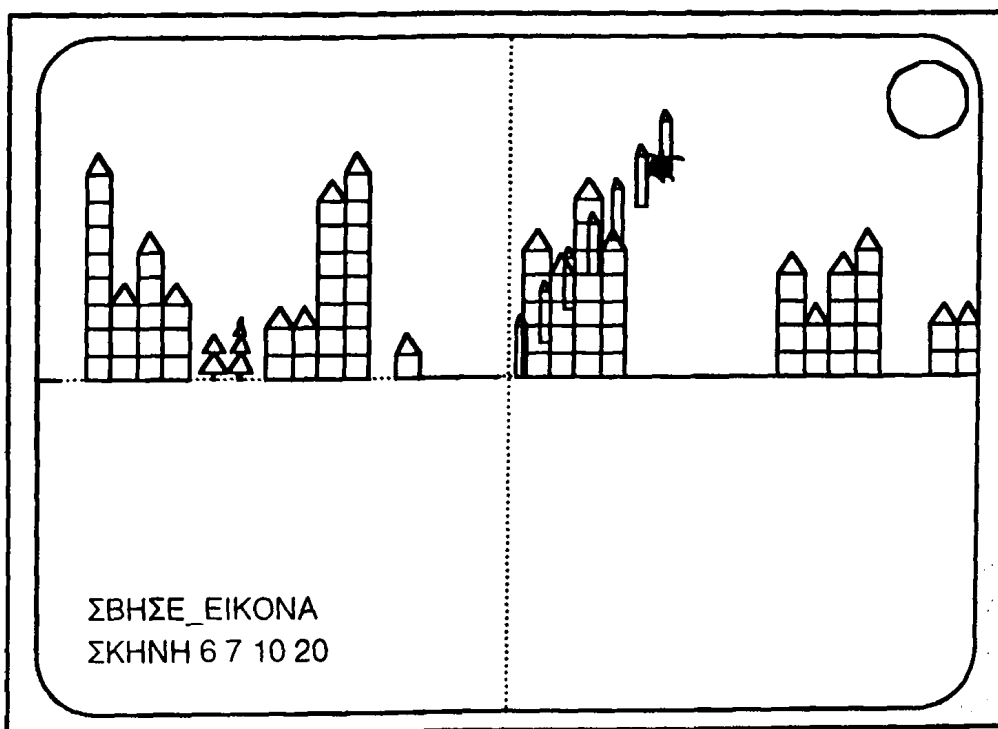
Ας σχεδιάσουμε ένα περιβάλλον με κτίρια και δένδρα.

```
ΤΟ ΕΔΑΦΟΣ :MEGEQOS
ΜΕΤΑΤΟΠΙΣΗ MINUS 200 0
REPEAT 5
  [MAKE "ΕΡΙΛΟΓΗ RANDOM 3
  IFELSE :ΕΡΙΛΟΓΗ = 0
    [MAKE "ΣΠΙΤΙΑ 4 MAKE "ΟΡΙΟ 4
    ΠΟΛΗ :MEGEQOS / 2 :ΣΠΙΤΙΑ :ΟΡΙΟ]
    [ΔΑΣΟΣ :MEGEQOS / 2]
  ]
ΑΦΕΤΗΡΙΑ
END
```

Μετά το σχεδιασμό του περιβάλλοντος η χελώνα θα πρέπει να επιστρέφει στην αφετηρία της. Αυτό γίνεται με την ΑΦΕΤΗΡΙΑ που χρησιμοποιεί την εντολή HOME (σπίτι). Η εντολή αυτή επιστρέφει τη χελώνα στο σημείο αφετηρίας, στο κέντρο της οθόνης.

```
ΤΟ ΑΦΕΤΗΡΙΑ
ΜΗ_ΓΡΑΦΕΙΣ
HOME
ΓΡΑΨΕ
END
```

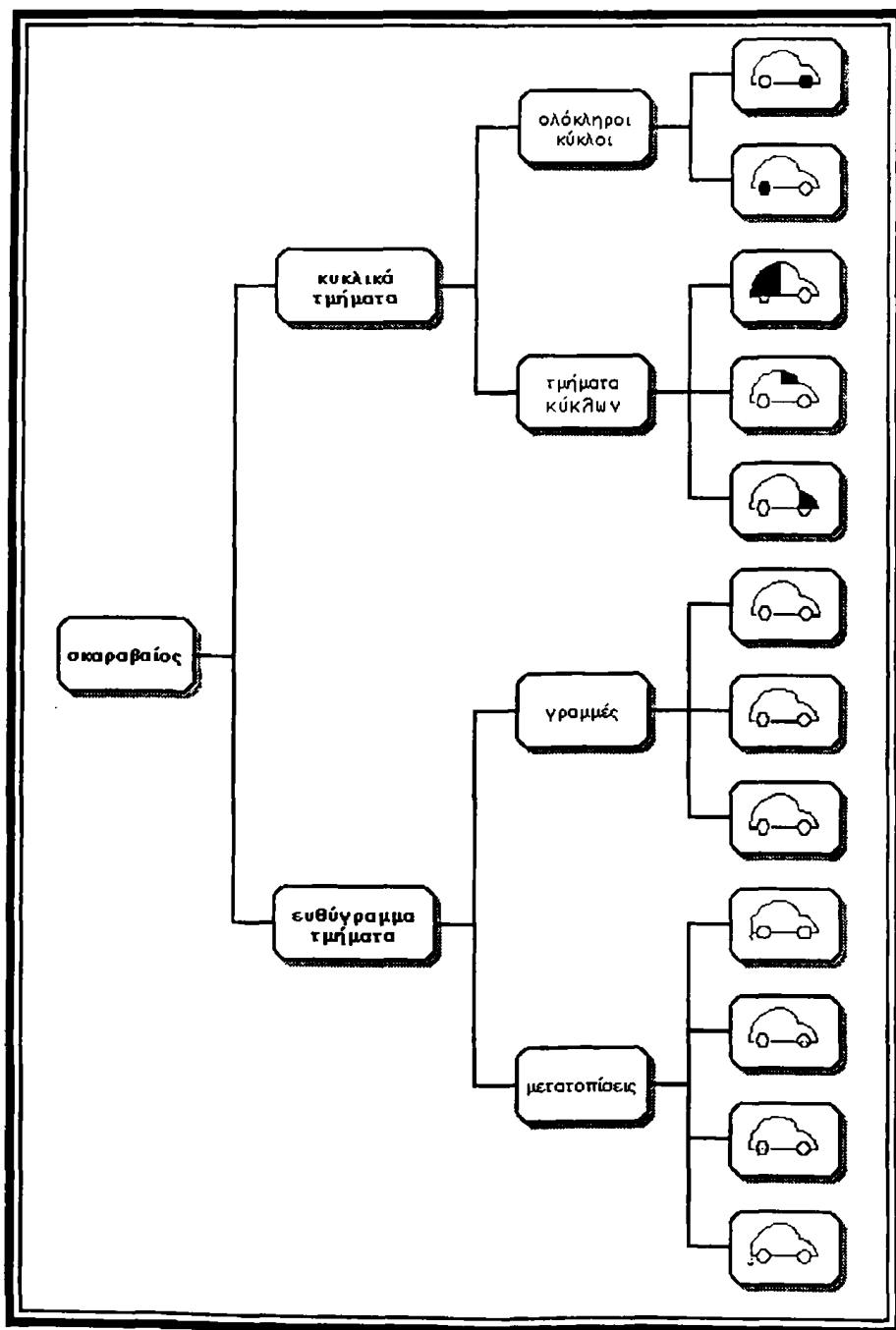
```
ΤΟ ΠΕΡΙΒΑΛΛΟΝ :MEGEQOS
ΕΔΑΦΟΣ :MEGEQOS
ΣΕΛΗΝΗ
ΠΥΡΑΥΛΟΣ :MEGEQOS
END
```



12 Ιεραρχικός σχεδιασμός

Οπτικός πίνακας περιεχομένων (Ο.Πι.Πε.)

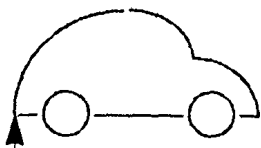
Σε έναν Ο.Πι.Πε. προσπαθούμε να απεικονίσουμε ομαδοποιημένα τα στοιχειώδη δομικά υλικά από τα οποία αποτελείται το σχήμα που θέλουμε να υλοποιήσουμε. Η σχεδίαση ενός Ο.Πι.Πε. είναι μια καθαρά αναλυτική νοητική διεργασία.



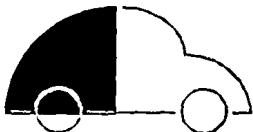
Στο διπλανό Ο.Πι.Πε. απεικονίζονται τα δομικά υλικά από τα οποία αποτελείται ο σκαρβαβίος. Αναλύοντας το σχήμα του σκαρβαβίου σε ένα πρώτο επίπεδο, διαπιστώνουμε ότι αποτελείται από κυκλικά και ευθύγραμμα τμήματα. Έτσι ξέροντας να φτιάχνουμε κύκλους και ευθύγραμμα τμήματα, έχουμε τα απαραίτητα υλικά για να δομήσουμε το σκαρβαβίο. Προχωρώντας σε δεύτερο επίπεδο, παρατηρούμε ότι τα κυκλικά τμήματα είναι είτε ολόκληροι κύκλοι είτε τεταρτοκύκλια. Ανάλογα, τα ευθύγραμμα τμήματα είναι είτε γραμμές είτε μετατοπίσεις. Άρα "παίζοντας" με τις παραμέτρους του κλάσματος του κύκλου και με το γράψε-μη γράφεις μπορούμε να έχουμε τα επιθυμητά σχήματα. Σε ένα τρίτο επίπεδο π.χ. μεταξύ των τεταρτοκυκλίων, διαπιστώνουμε ότι οι διαφορές εντοπίζονται στο μέγεθος και τον προσανατολισμό τους.



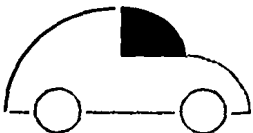
Η σωστή σειρά των πραγμάτων



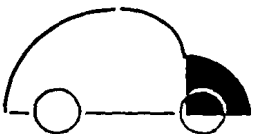
Είδαμε στον Ο.Πι.Πε. τα τμήματα από τα οποία αποτελείται ο σκαραβαίος. Για να σχεδιαστεί όμως θα πρέπει αυτά να γίνουν με μια συγκεκριμένη σειρά. Ας ξεκινήσουμε με τη ΜΕΤΑΤΟΠΙΣΗ 0 5.



Αφού η χελώνα έφτασε στο κάτω αριστερά σημείο θα πρέπει να σχεδιάσουμε το πίσω τεταρτοκύκλιο. Αυτό γίνεται με τις εντολές ΔΕΞΙΑ 90 και ΚΥΚΛΟΣ 10 1/4.

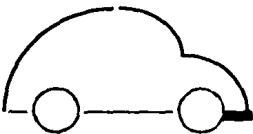


Στη συνέχεια πρέπει να σχεδιαστεί το τεταρτοκύκλιο που αντιπροσωπεύει το μπροστινό μέρος της σκελής του σκαραβαίου. Αυτό θα γίνει με την εντολή

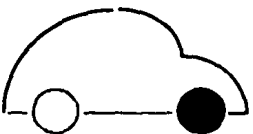


ΚΥΚΛΟΣ 4 1/4.

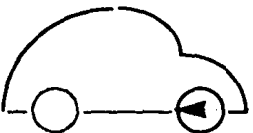
Το εμπρός μέρος του αυτοκινήτου είναι και αυτό ένα τεταρτοκύκλιο που θα σχεδιαστεί αφού προσανατολιστεί η χελώνα με τις εντολές ΑΡΙΣΤΕΡΑ 90 και ΚΥΚΛΟΣ 4 1/4.



Το επόμενο βήμα είναι να σχεδιάσουμε το ευθύγραμμο κάτω μέρος μέχρι τη μπροστινή πόρτα. Αυτό θα γίνει με την εντολή ΕΥΘΕΙΑ 15.



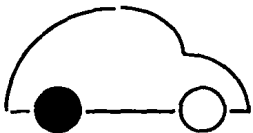
Η μπροστινή ρόδα είναι ένας πλήρης κύκλος και η δημιουργία της είναι εύκολη με την εντολή ΚΥΚΛΟΣ 1.5 1



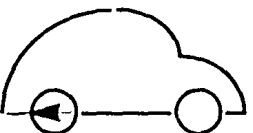
Με τον πλήρη κύκλο η χελώνα έχει επιστρέψει στο δεξιό σημείο από όπου ξεκίνησε. Θα πρέπει να μετατοπισθεί παρακάμπτοντας τη ρόδα. Γίνεται με την εντολή ΜΕΤΑΤΟΠΙΣΗ 0 18.



Το κάτω μέρος του σκαραβαίου μεταξύ των δύο τροχών γίνεται με την εντολή ΕΥΘΕΙΑ 42.



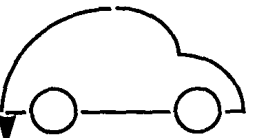
Ο πίσω τροχός είναι ένας πλήρης κύκλος και η δημιουργία του είναι εύκολη με την εντολή ΚΥΚΛΟΣ 1.5 1.



Με τον πλήρη κύκλο η χελώνα έχει επιστρέψει στο δεξί σημείο από όπου ξεκίνησε. Θα πρέπει να μετατοπισθεί παρακάμπτοντας τη ρόδα. Γίνεται με την εντολή ΜΕΤΑΤΟΠΙΣΗ 0 18.



Απομένει το κάτω μέρος του αυτοκινήτου, από τον τροχό μέχρι το πίσω μέρος. Αυτό γίνεται με την εντολή ΕΥΘΕΙΑ 10.



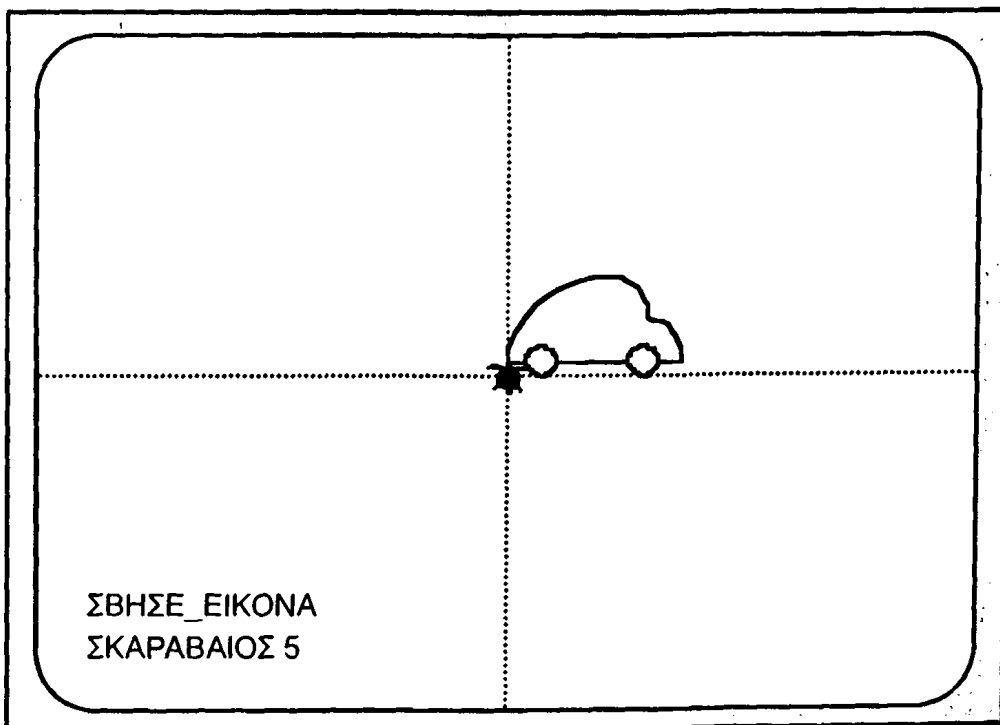
Τέλος πρέπει η χελώνα αφού προσανατολιστεί με την εντολή ΔΕΞΙΑ 90 να επιστρέψει στην αφετηρία της με τη ΜΕΤΑΤΟΠΙΣΗ 0 -7.

Παραμετροποιημένος σκαραβαίος

Η προηγούμενη διαδικασία σχεδιάζει ένα σκαραβαίο σταθερού μεγέθους. Μπορούμε χρησιμοποιώντας την παράμετρο YPSOS να μεταβάλουμε το μέγεθος του σκαραβαίου.

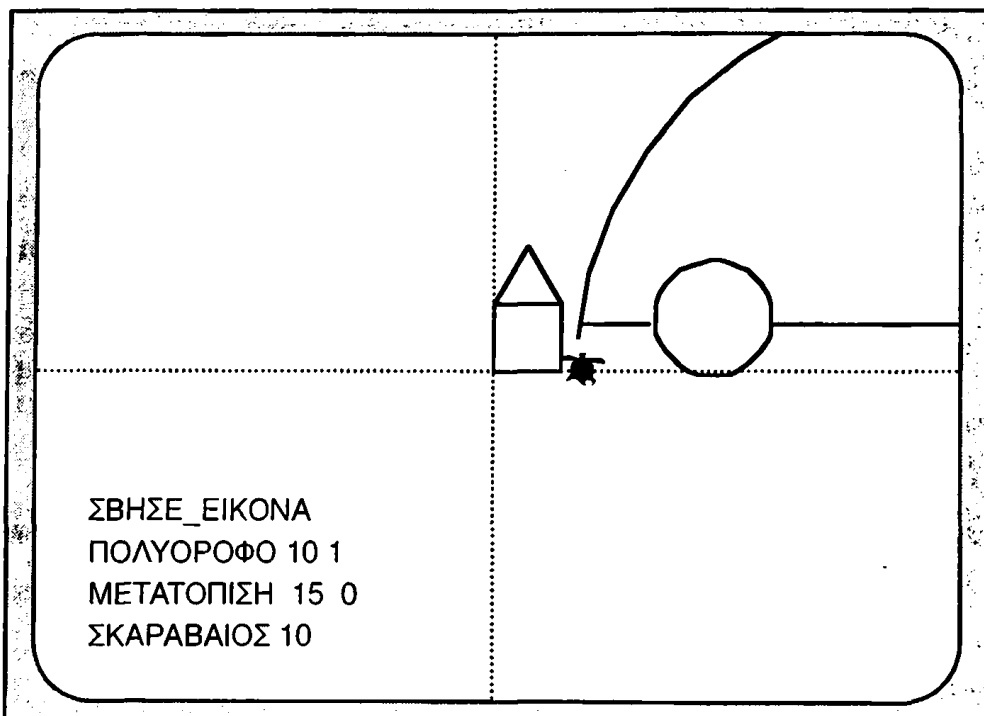
Προϋπόθεση είναι να διατηρήσουμε στη νέα διαδικασία τις αναλογίες μεταξύ των τιμών με τις οποίες καλούνται να εκτελεσθούν όλες οι διαδικασίες που χρησιμοποιούνται από την κύρια διαδικασία.

```
ΤΟ ΣΚΑΡΑΒΑΙΟΣ :YPSOS
ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 0.5
ΔΕΞΙΑ 90
ΚΥΚΛΟΣ :YPSOS 1 / 4
ΚΥΚΛΟΣ :YPSOS * 0.4 1 / 4
ΑΡΙΣΤΕΡΑ 90
ΚΥΚΛΟΣ :YPSOS * 0.4 1 / 4
ΕΥΘΕΙΑ :YPSOS * 1.5
ΚΥΚΛΟΣ :YPSOS * 0.15 1
ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 1.8
ΕΥΘΕΙΑ :YPSOS * 4.2
ΚΥΚΛΟΣ :YPSOS * 0.15 1
ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 1.8
ΕΥΘΕΙΑ :YPSOS * 1.0
ΔΕΞΙΑ 90
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:YPSOS * 0.7)
END
```

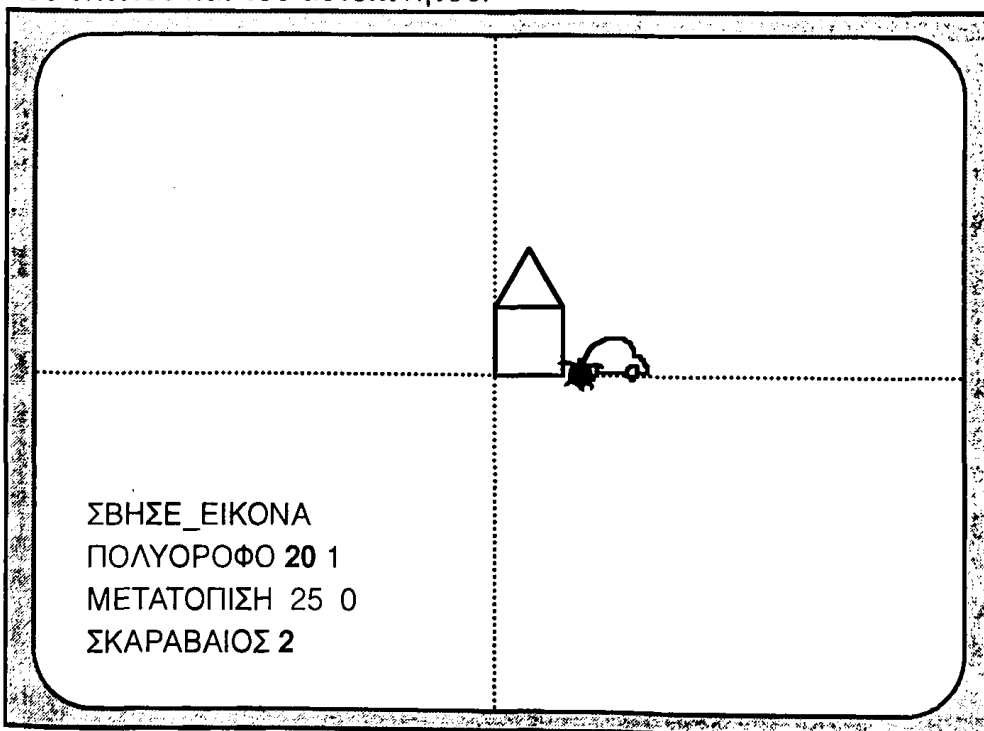


Ένας σκαραβαίος γίγαντας

Σχεδιάζοντας ένα σπίτι και ένα σκαραβαίο δίνοντας την ίδια τιμή στην παράμετρο που καθορίζει το μέγεθός τους, θα πάρουμε το παρακάτω σχήμα. Προφανώς το σκαραβαίο τον οδηγεί κάποιος γίγαντας.



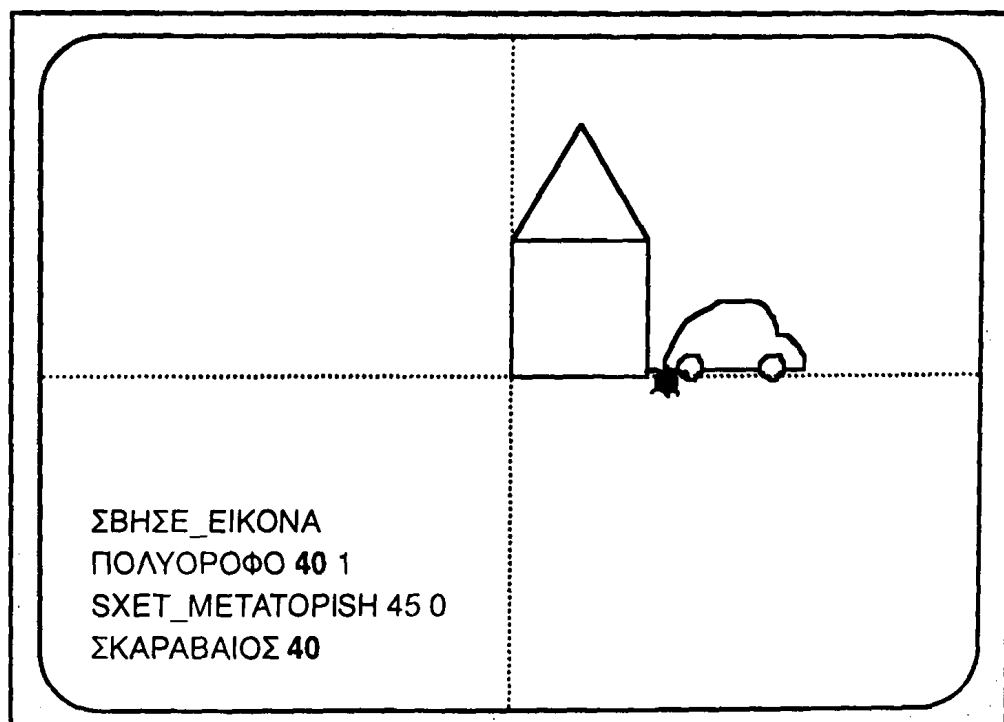
Μπορούμε με μερικές προσπάθειες κρατώντας σταθερό το μέγεθος του ενός σχήματος και μεταβάλλοντας την τιμή του μεγέθους του άλλου, να πετύχουμε μια παραδεκτή αναλογία μεγέθους μεταξύ του σπιτιού και του αυτοκινήτου.



“Κανονικοποίηση” του σκαραβαίου

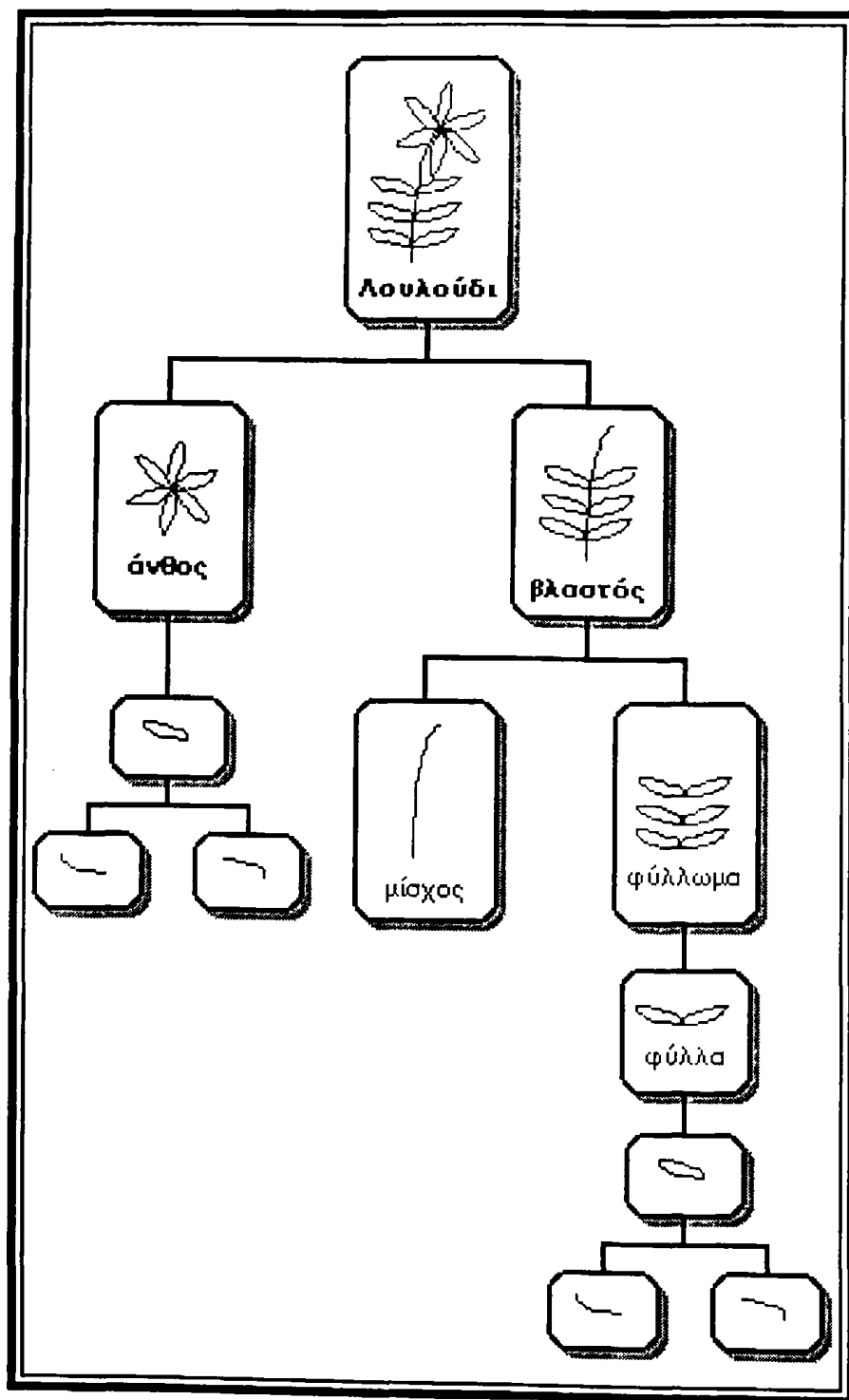
Θα πρέπει την αναλογία μεταξύ σπιτιού-περιβάλλοντος και αυτοκινήτου να την ενσωματώσουμε στον κώδικα της διαδικασίας του σκαραβαίου. Είδαμε ότι το μέγεθος του σκαραβαίου αντιστοιχεί στο 1/10 της τιμής του μεγέθους του σπιτιού. Αυτήν την τιμή που κανονικοποιεί τις δύο διαδικασίες θα εισάγουμε στον κώδικα της διαδικασίας ΣΚΑΡΑΒΑΙΟΣ.

```
ΤΟ ΣΚΑΡΑΒΑΙΟΣ :ΜΕΓΕΘΟΣ
ΜΑΚΕ *ΥΡΣΟΣ (:ΜΕΓΕΘΟΣ / 10)
ΜΕΤΑΤΟΠΙΣΗ 0 :ΥΡΣΟΣ*0.5
ΔΕΞΙΑ 90
ΚΥΚΛΟΣ :ΥΡΣΟΣ 1 / 4
ΚΥΚΛΟΣ :ΥΡΣΟΣ * 0.4 1 / 4
ΑΡΙΣΤΕΡΑ 90
ΚΥΚΛΟΣ :ΥΡΣΟΣ * 0.4 1 / 4
ΕΥΘΕΙΑ :ΥΡΣΟΣ * 1.5
ΚΥΚΛΟΣ :ΥΡΣΟΣ * 0.15 1
ΜΕΤΑΤΟΠΙΣΗ 0 :ΥΡΣΟΣ * 1.8
ΕΥΘΕΙΑ :ΥΡΣΟΣ * 4.2
ΚΥΚΛΟΣ :ΥΡΣΟΣ * 0.15 1
ΜΕΤΑΤΟΠΙΣΗ 0 :ΥΡΣΟΣ * 1.8
ΕΥΘΕΙΑ :ΥΡΣΟΣ * 1.0
ΔΕΞΙΑ 90
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:ΥΡΣΟΣ * 0.7)
END
```



Αναλύοντας τα μέρη ενός λουλουδιού

Για να σχεδιάσουμε το λουλούδι θα ακολουθήσουμε την ίδια μέθοδο που ακολουθήσαμε και στο σκαραβαίο. Θα αναλύσουμε το λουλούδι στα βασικά του τμήματα και στη συνέχεια θα τα αναλύσουμε περαιτέρω. Αυτό θα συνεχιστεί μέχρι να φτάσουμε σε στοιχειώδη μέρη που μπορούν να σχεδιαστούν εύκολα.



Στο διπλανό σχήμα φαίνεται ο Οπτικός Πίνακας Περιεχομένων όπου απεικονίζονται τα δομικά υλικά από τα οποία αποτελείται το λουλούδι.

Σε ένα πρώτο επίπεδο το λουλούδι αποτελείται από το άνθος και το βλαστό.

Αναλύοντας το βλαστό σε ένα δεύτερο επίπεδο τον χωρίζουμε στο μίσχο και το φύλλωμα. Ο μίσχος είναι ένα τόξο (τμήμα περιφέρειας κύκλου).

Σε τρίτο επίπεδο το φύλλωμα αποτελείται από τα φύλλα.

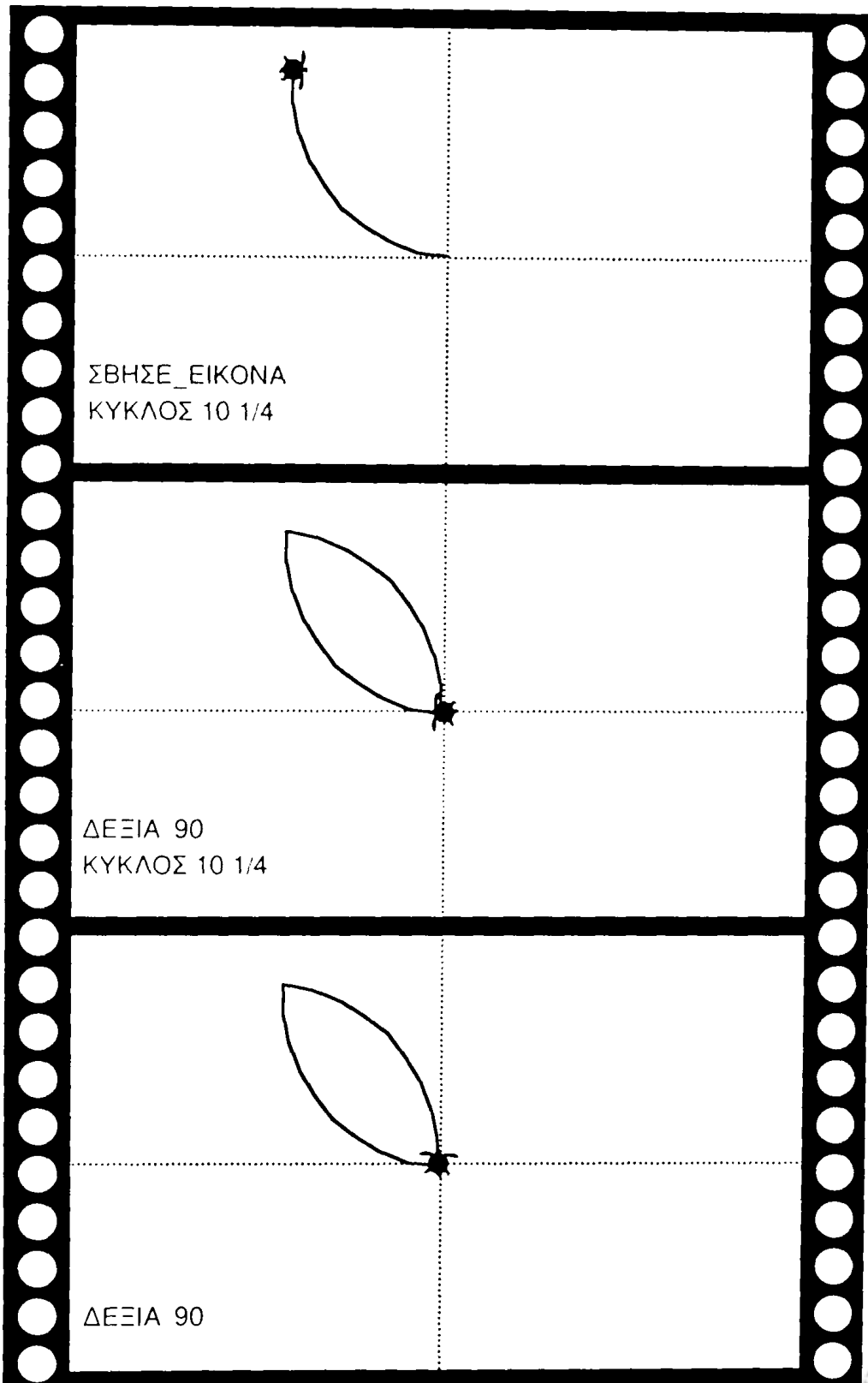
Στο προτελευταίο επίπεδο τα φύλλα αλλά και το άνθος αποτελούνται από το "φύλλο" ή "πέταλο".

Τέλος το "φύλλο" ή "πέταλο" σχηματίζεται από δύο τμήματα κύκλου.



Δημιουργώντας το φύλλο ή πέταλο

Σχεδιάζουμε το φύλλο, ένα από τα στοιχειώδη σχήματα του λουλουδιού.



Σχεδιάζουμε ένα τεταρτοκύκλιο.

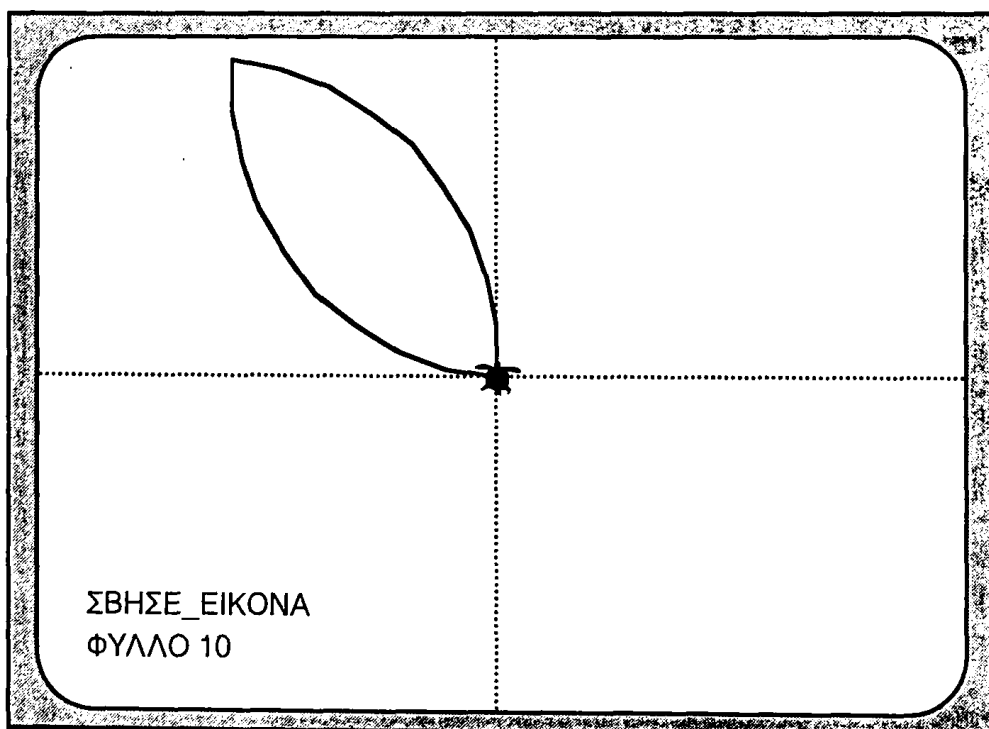
Στη συνέχεια στρίβοντας τη χελώνα 90 μοίρες δεξιά σχεδιάζουμε ένα ακόμα τεταρτοκύκλιο. Η χελώνα έχει επιστρέψει στην αφετηρία.

Τέλος, επαναφέρουμε τη χελώνα στην αρχική της κατάσταση.

Το φύλλο ως διαδικασία

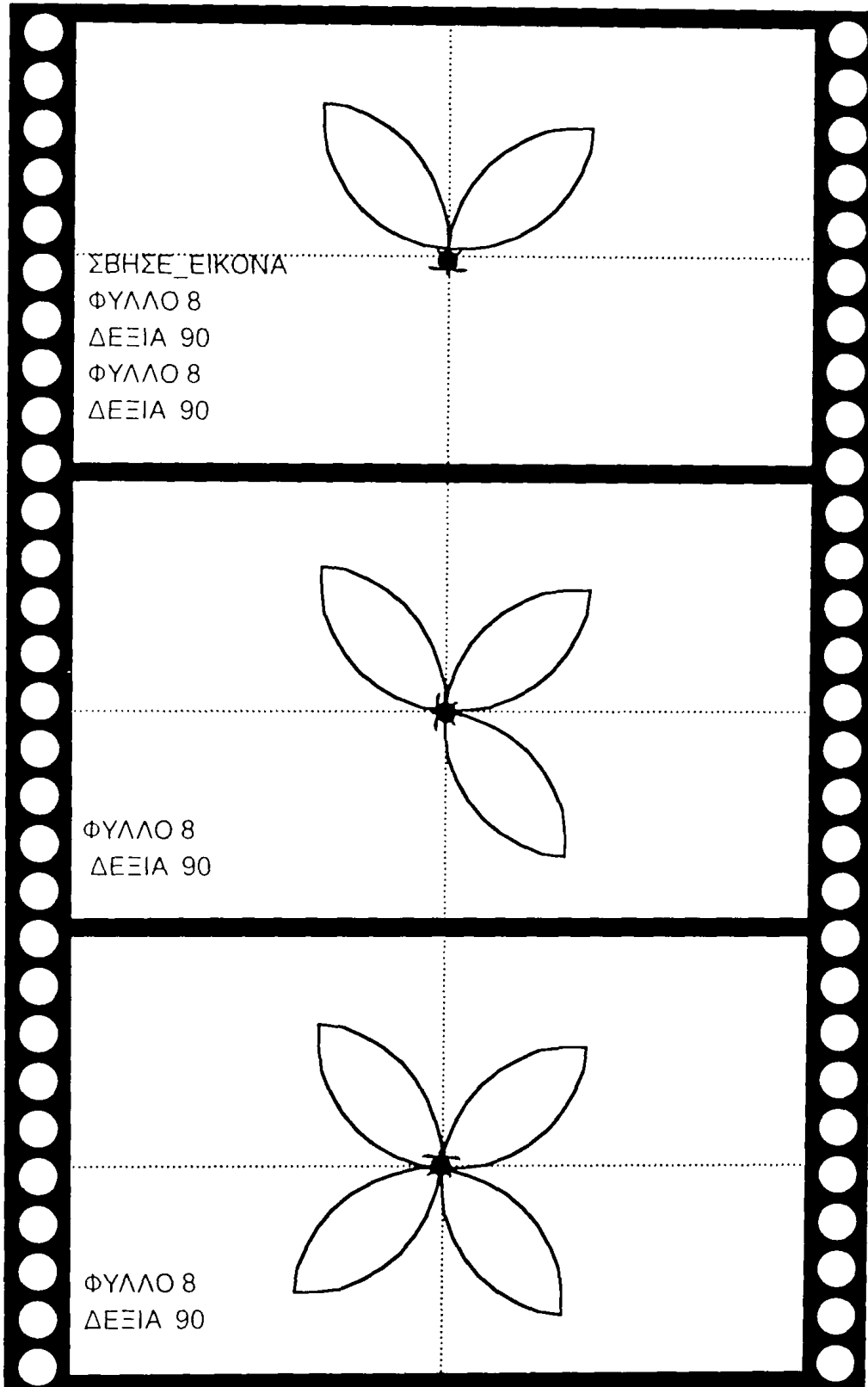
Οι προηγούμενες εντολές ολοκληρώνονται στη διαδικασία ΦΥΛΛΟ που θα σχεδιάζει το φύλλο του βλαστού και το πέταλο του άνθους του λουλουδιού.

```
ΤΟ ΦΥΛΛΟ :ΜΕΓΕΘΟΣ  
ΚΥΚΛΟΣ :ΜΕΓΕΘΟΣ 1 / 4  
ΔΕΞΙΑ 90  
ΚΥΚΛΟΣ :ΜΕΓΕΘΟΣ 1 / 4  
ΔΕΞΙΑ 90  
END
```



Ένα άνθος με τέσσερα πέταλα

Περιστρέφοντας το φύλλο/πέταλο 90 μοίρες κάθε φορά, σχηματίζουμε ένα άνθος.



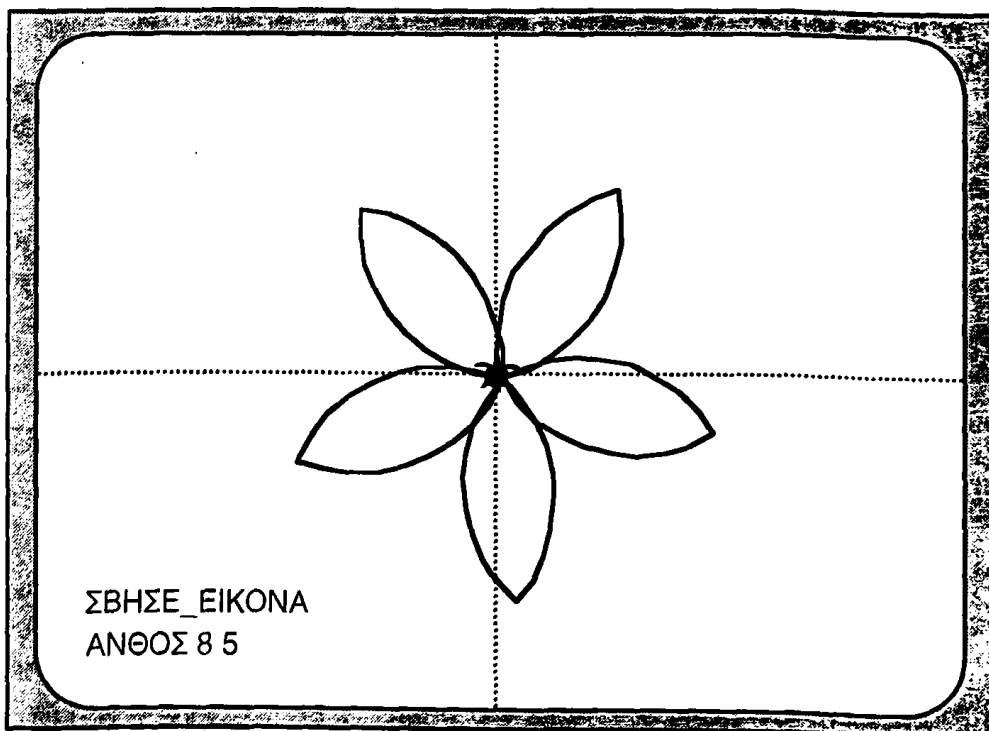
Σχεδιάζοντας το άνθος

Η διαδικασία ΑΝΘΟΣ ολοκληρώνεται ως εξής:

```
TO ΑΝΘΟΣ :MEGEQOS  
REPEAT 4  
  [ΦΥΛΛΟ :MEGEQOS ΔΕΞΙΑ 90]  
END
```

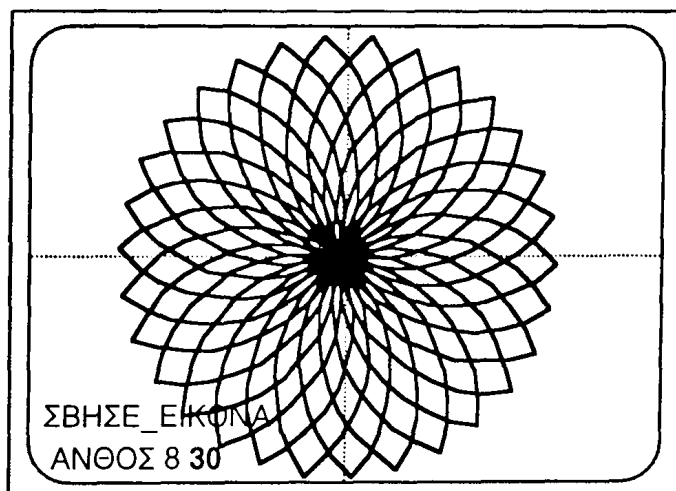
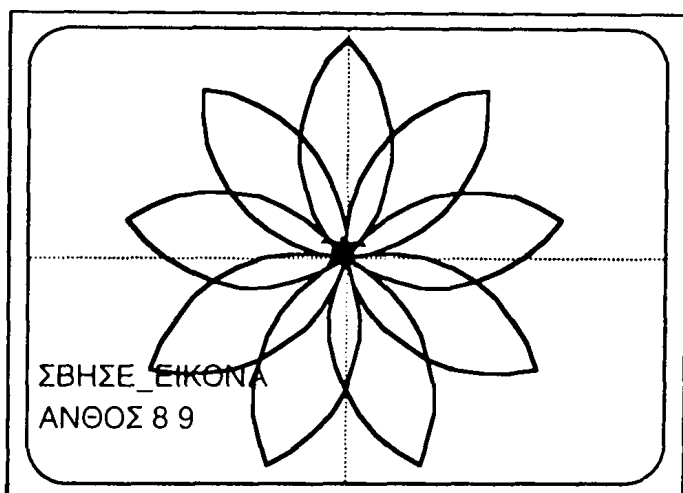
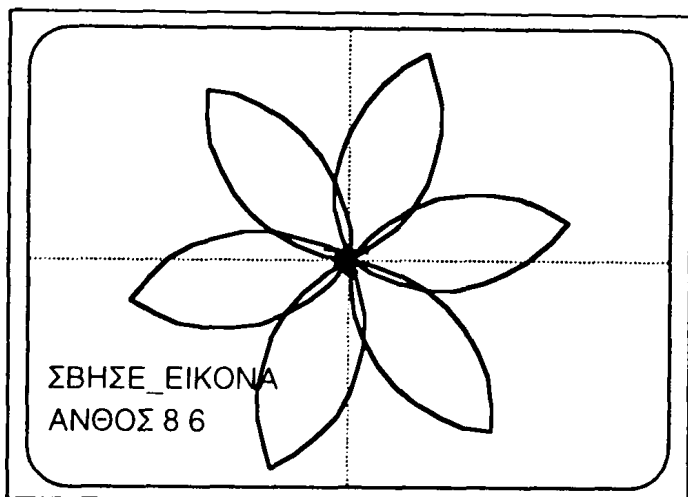
Μπορούμε να γενικεύσουμε τη διαδικασία ΑΝΘΟΣ έτσι ώστε να καθορίζεται έξω από αυτήν το πλήθος των πετάλων που θα έχει το άνθος. Η στροφή που θα κάνει το φύλλο κάθε φορά δε θα είναι πλέον 90 μοίρες αλλά υπολογίζεται από τη διαίρεση του 360 διά το πλήθος των πετάλων.

```
TO ΑΝΘΟΣ :MEGEQOS :PETALA  
MAKE "GWNIA (360 / :PETALA)  
REPEAT :PETALA  
  [ΦΥΛΛΟ :MEGEQOS  
  ΔΕΞΙΑ :GWNIA]  
END
```



Σχεδιάζοντας άνθη

Ας βάλουμε τη χελώνα να σχεδιάσει μερικά άνθη με διαφορετικό πλήθος πετάλων.



Ένα άνθος με 30
ροδοπέταλα θα
μπορούσαμε να
το ονομάζουμε
τριαντάφυλλο.

Τμήμα μίσχου του βλαστού του λουλουδιού

Θα μπορούσαμε να σχεδιάσουμε ολόκληρο το μίσχο αλλά μετά θα έπρεπε να οπισθοχωρούμε και να "κολλάμε" σε διάφορα σημεία του τα φύλλα. Γι' αυτόν το λόγο θα ακολουθήσουμε μια διαφορετική προσέγγιση. Κάθε φορά, σχεδιάζουμε ένα τμήμα του μίσχου μέχρι το σημείο στο οποίο θα πρέπει να προστεθούν τα φύλλα.

Ας γράψουμε τη διαδικασία που θα σχεδιάζει το τμήμα του μίσχου. Βασικό στοιχείο του θα είναι το τόξο της περιφέρειας κύκλου που θα αντιστοιχεί σε μια μικρή γωνία και θα πρέπει να είναι σωστά προσανατολισμένο.

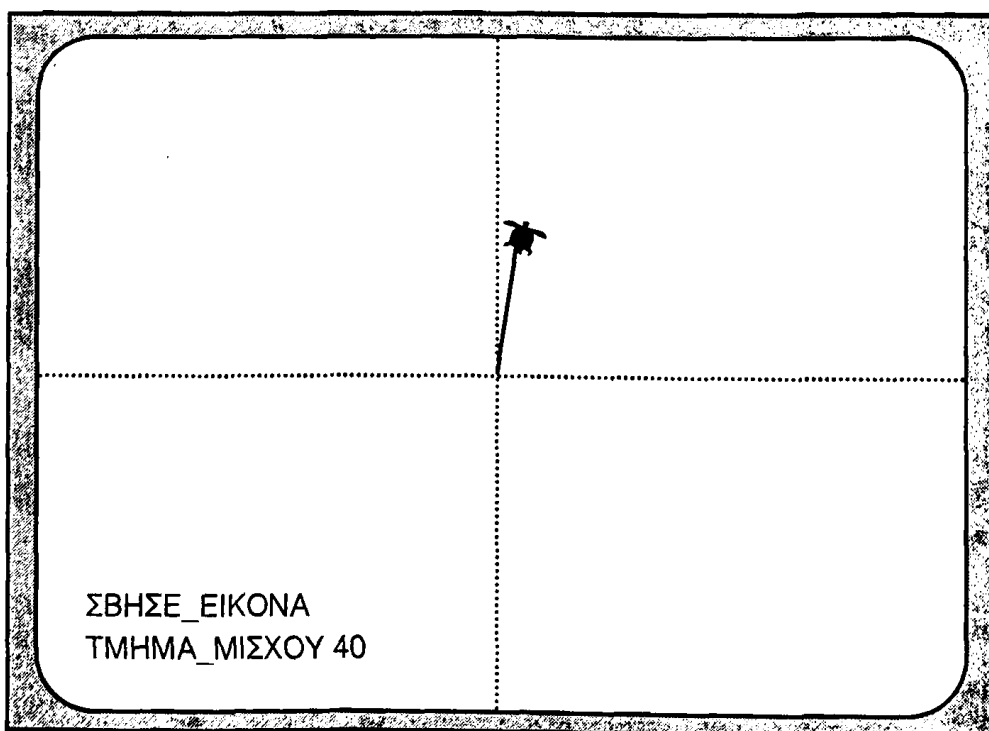
ΤΟ ΤΜΗΜΑ_ΜΙΣΧΟΥ :MEGEQOS

ΔΕΞΙΑ 90

ΚΥΚΛΟΣ :MEGEQOS 1 / 24

ΑΡΙΣΤΕΡΑ 90

END



Τα φύλλα του βλαστού του λουλουδιού

Μετά το τμήμα του μίσχου προσθέτουμε τα δύο φύλλα. Ας γράψουμε τη διαδικασία που θα σχεδιάζει αυτά τα δύο φύλλα.

ΤΟ ΦΥΛΛΑ :ΜΕΓΕΘΟΣ

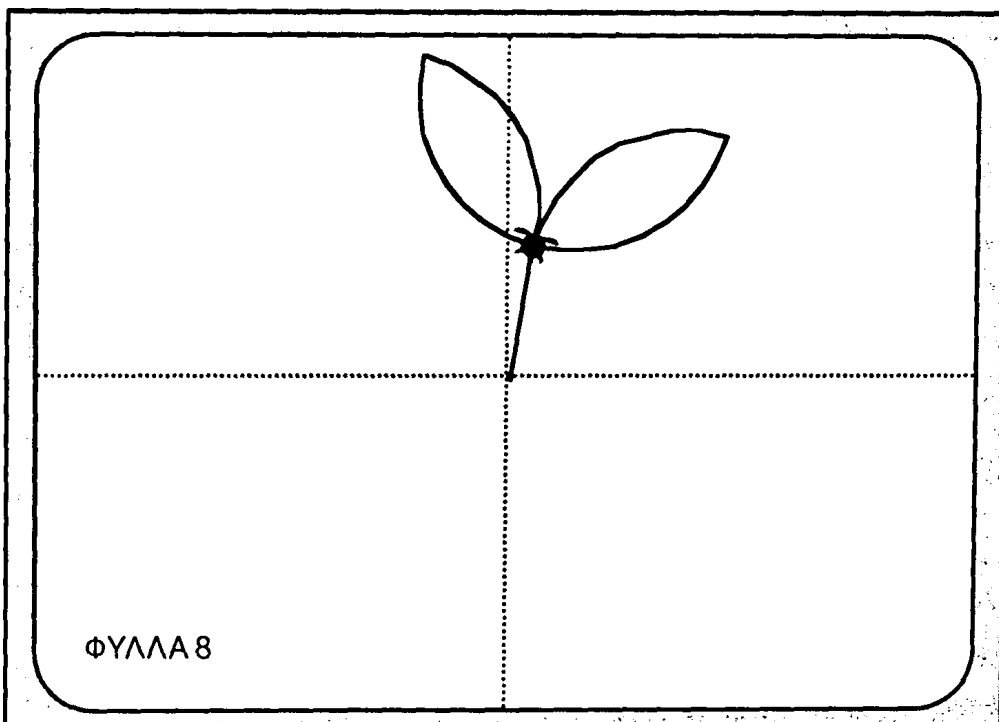
ΦΥΛΛΟ :ΜΕΓΕΘΟΣ

ΔΕΞΙΑ 90

ΦΥΛΛΟ :ΜΕΓΕΘΟΣ

ΑΡΙΣΤΕΡΑ 90

END



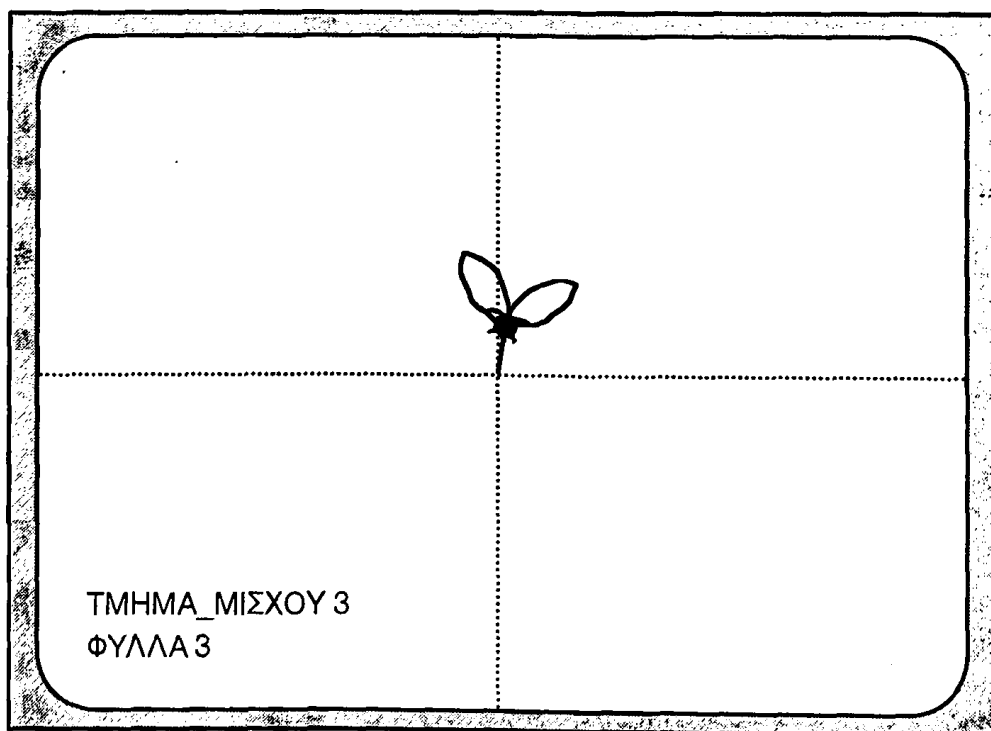
“Κανονικοποίηση” του τμήματος του μίσχου

Τα φύλλα σχεδιάζονται με τη διαδικασία ΦΥΛΛΑ 8, ενώ το τμήμα του μίσχου με τη ΤΜΗΜΑ_ΜΙΣΧΟΥ 40.

Θα πρέπει το μέγεθος του μίσχου να προσαρμοστεί έτσι ώστε να ταιριάζει με το μέγεθος των φύλλων.

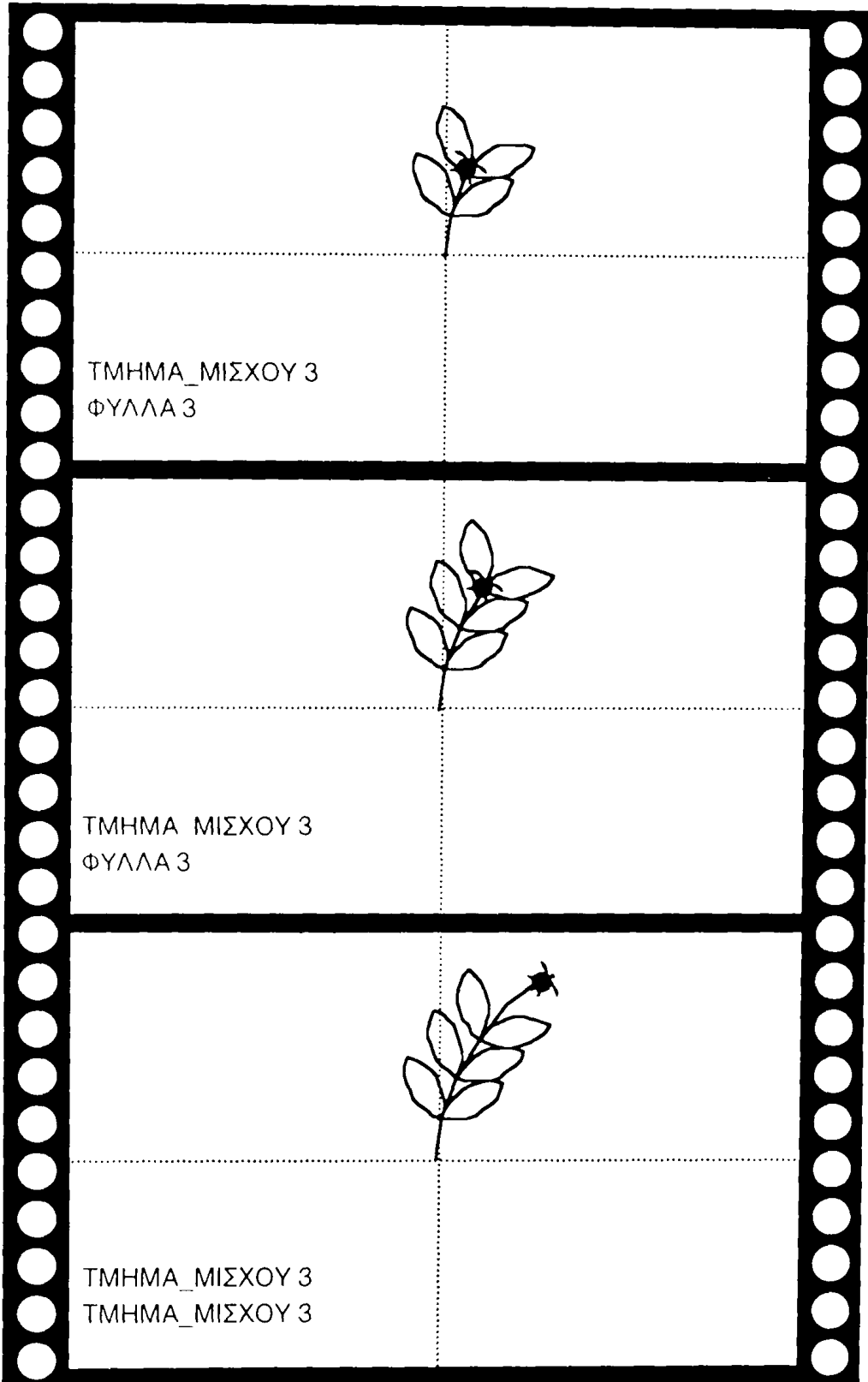
Συγκρίνοντας τα αντίστοιχα μεγέθη (8 και 40) διαπιστώνουμε ότι είτε θα πρέπει να πενταπλασιάσουμε το μέγεθος του τμήματος του μίσχου ή να υποπενταπλασιάσουμε το μέγεθος των φύλλων. Ας διαλέξουμε τη πρώτη επιλογή και ας την ενσωματώσουμε στη διαδικασία ΤΜΗΜΑ_ΜΙΣΧΟΥ.

```
TO ΤΜΗΜΑ_ΜΙΣΧΟΥ :ΜΕΓΕΘΟΣ  
MAKE "ΜΕΓΕΘΟΣ (:ΜΕΓΕΘΟΣ * 5)  
ΔΕΞΙΑ 90  
ΚΥΚΛΟΣ :ΜΕΓΕΘΟΣ 1 / 24  
ΑΡΙΣΤΕΡΑ 90  
END
```



Σχηματίζοντας το βλαστό

Σχεδιάζοντας βήμα προς βήμα τα τμήματα του μίσχου και τα φύλλα, σχηματίζεται ο βλαστός.



Η διαδικασία που σχεδιάζει το βλαστό

Η διαδικασία ΒΛΑΣΤΟΣ ενσωματώνει όλες τις προηγούμενες ενέργειες για να σχεδιασθεί ο βλαστός του λουλουδιού.

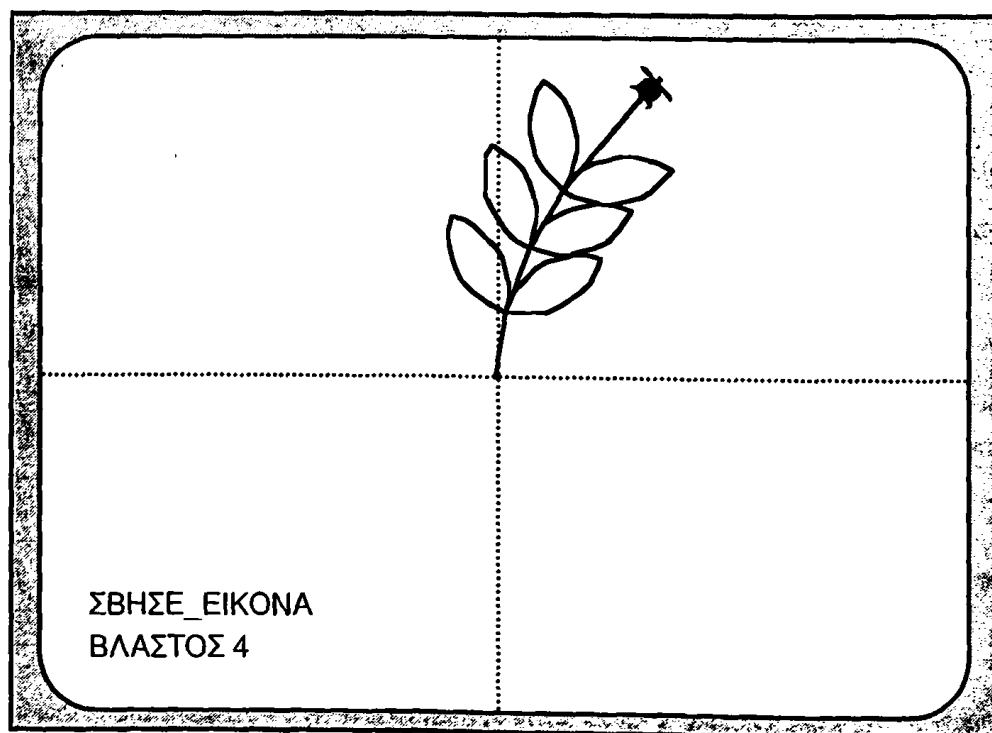
Στο επάνω μέρος του μίσχου βάλαμε δύο επιπλέον τμήματα μίσχου ώστε στο τέλος του να υπάρχει χώρος για να σχεδιαστεί το άνθος.

ΤΟ ΒΛΑΣΤΟΣ :MEGEQOS

REPEAT 3 [ΤΜΗΜΑ_ΜΙΣΧΟΥ :MEGEQOS ΦΥΛΛΑ :MEGEQOS]

ΤΜΗΜΑ_ΜΙΣΧΟΥ (:MEGEQOS + :MEGEQOS)

END



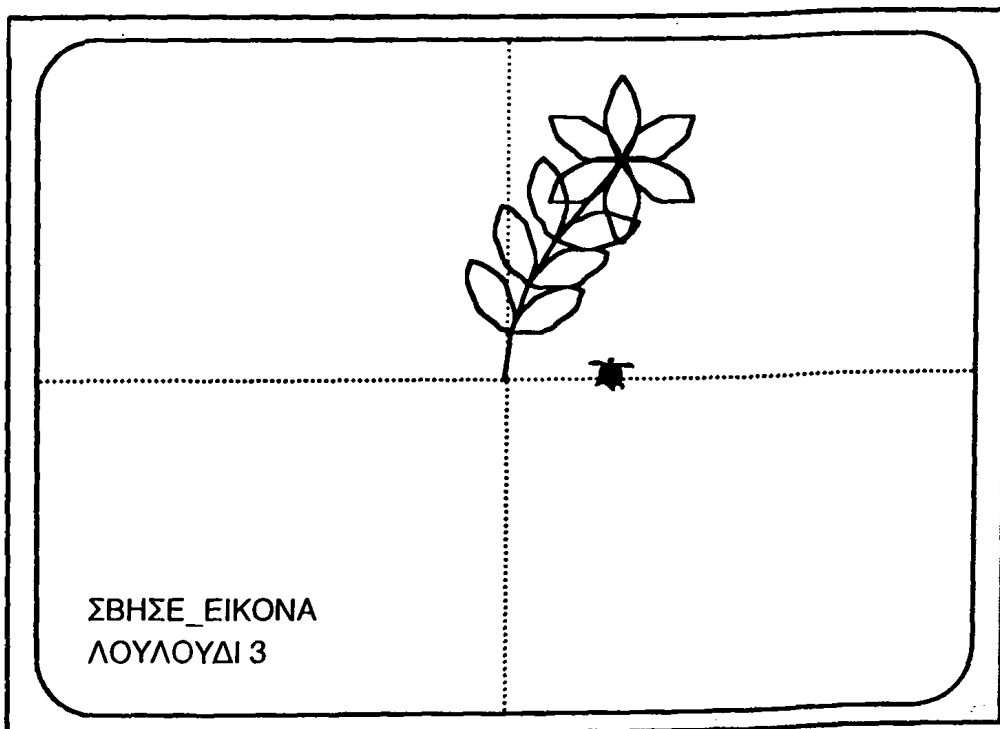
Ολόκληρο το λουλούδι

Αφού έχουμε δημιουργήσει ξεχωριστά όλες τις επιμέρους διαδικασίες από τις οποίες αποτελείται το λουλούδι, τις συναρμολογούμε σε μια που το σχεδιάζει.

Το πλήθος των πετάλων του άνθους θα παίρνει τυχαία τιμή κάθε φορά, με τη βοήθεια της συνάρτησης RANDOM. Εμείς όμως δε θέλουμε ένα μαδημένο άνθος με κανένα ή ένα-δύο πέταλα. Γι' αυτό θα προσθέτουμε τον αριθμό 3 στην τιμή που θα προκύπτει από τη συνάρτηση RANDOM έτσι ώστε η τιμή του πλήθους των πετάλων να μη μπορεί να είναι μικρότερη του 3 (περίπτωση που η συνάρτηση RANDOM επιστρέψει την τιμή 0).

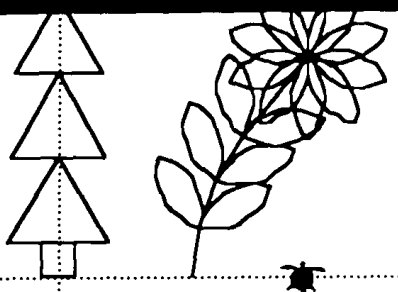
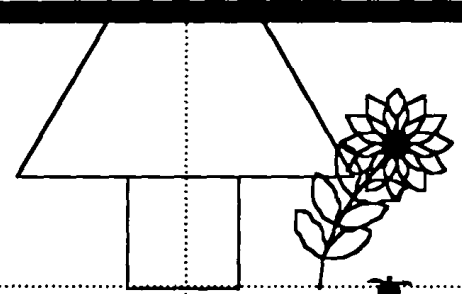
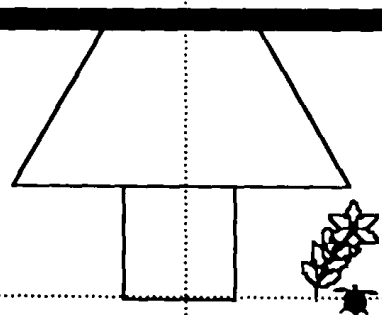
Θα πρέπει να λάβουμε πρόνοια ώστε η χελώνα μετά τη σχεδίαση του λουλουδιού, να παίρνει θέση ώστε να μπορεί στη συνέχεια να σχεδιαστεί κάτι άλλο στην κατάλληλη θέση.

```
ΤΟ ΛΟΥΛΟΥΔΙ :MEGEQOS
MAKE *PETALA (3 + RANDOM 10)
ΒΛΑΣΤΟΣ :MEGEQOS
ΑΝΘΟΣ :MEGEQOS :PETALA
ΜΗ_ΓΡΑΦΕΙΣ
ΑΡΙΣΤΕΡΑ 40
ΟΠΙΣΘΕΝ (21.5 * :MEGEQOS)
ΓΡΑΨΕ
END
```



“Κανονικοποίηση” του λουλουδιού

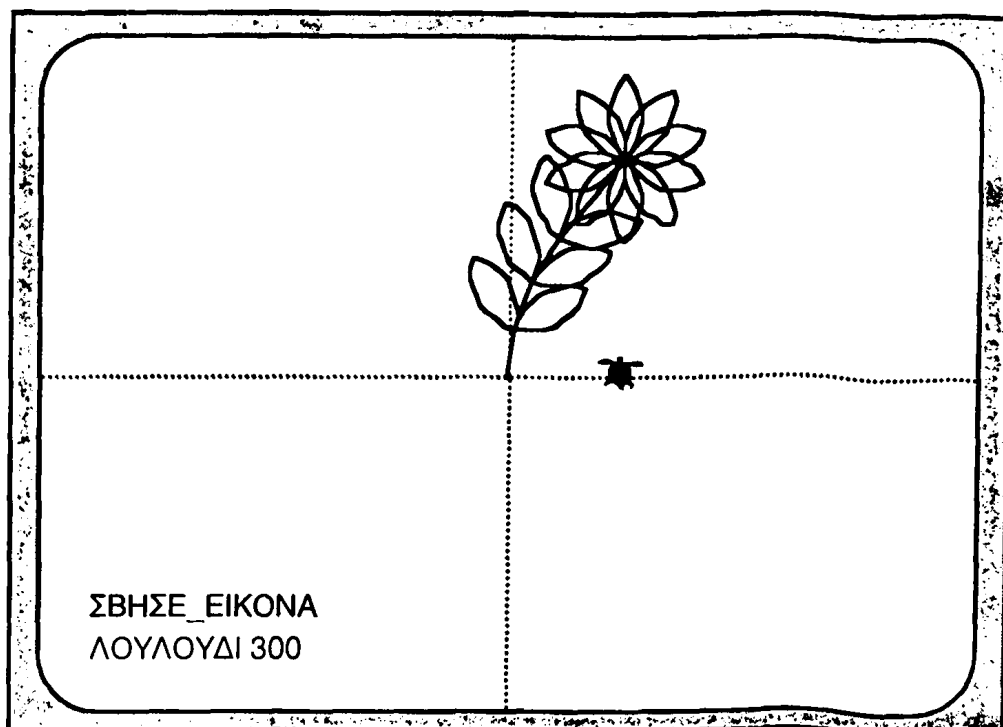
Θα προσπαθήσουμε να βρούμε τη σωστή αναλογία μεταξύ του μεγέθους του λουλουδιού και ενός δένδρου.


ΣΒΗΣΕ_ΕΙΚΟΝΑ ΕΛΑΤΟ 30 ΜΕΤΑΤΟΠΙΣΗ 40 0 ΛΟΥΛΟΥΔΙ 3

ΣΒΗΣΕ_ΕΙΚΟΝΑ ΕΛΑΤΟ 100 ΜΕΤΑΤΟΠΙΣΗ 40 0 ΛΟΥΛΟΥΔΙ 2

ΣΒΗΣΕ_ΕΙΚΟΝΑ ΕΛΑΤΟ 100 ΜΕΤΑΤΟΠΙΣΗ 40 0 ΛΟΥΛΟΥΔΙ 1

Η διαδικασία που σχεδιάζει το λουλούδι

Πειραματιζόμενοι διαπιστώνουμε ότι το μέγεθος του λουλουδιού θα πρέπει να είναι το 1/100 του μεγέθους του δένδρου. Αυτή τη διαπίστωση θα την ενσωματώσουμε στη διαδικασία ΛΟΥΛΟΥΔΙ.

```
ΤΟ ΛΟΥΛΟΥΔΙ :ΜΕΓΕΘΟΣ
ΜΑΚΕ "ΜΕΓΕΘΟΣ (:ΜΕΓΕΘΟΣ / 100)
ΜΑΚΕ "ΡΕΤΑΛΑ (3 + RANDOM 10)
ΒΛΑΣΤΟΣ :ΜΕΓΕΘΟΣ
ΑΝΘΟΣ :ΜΕΓΕΘΟΣ :ΡΕΤΑΛΑ
ΜΗ_ΓΡΑΦΕΙΣ
ΑΡΙΣΤΕΡΑ 40
ΟΠΙΣΘΕΝ (21.5 * :ΜΕΓΕΘΟΣ)
ΓΡΑΨΕ
ΕΝΔ
```



Σχετική και απόλυτη θέση στην οθόνη

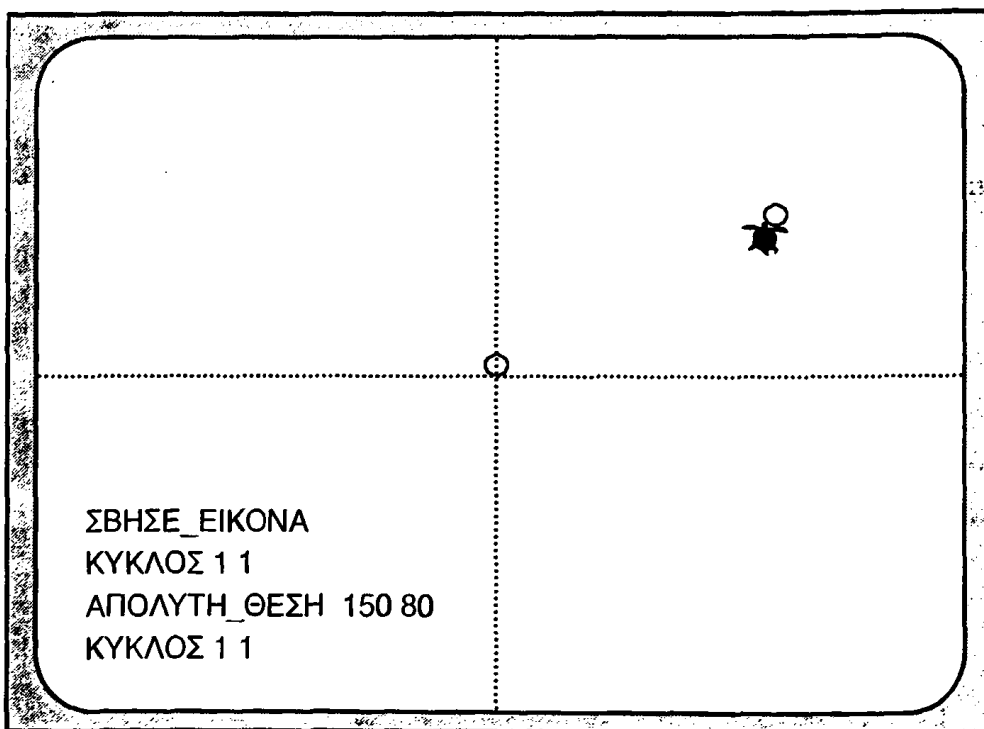
Για να μετακινηθεί η χελώνα από το σημείο που βρίσκεται σε κάποιο άλλο, πρέπει να ξέρουμε την απόσταση των δύο σημείων. Το σημείο που θα φτάσει η χελώνα καθορίζεται με βάση το σημείο από το οποίο ξεκινάει, δηλαδή η θέση του είναι σχετική προς το αρχικό σημείο.

Μπορούμε να θεωρήσουμε αυθαίρετα κάποιο συγκεκριμένο σημείο της οθόνης ως απόλυτο αρχικό σημείο και όλα τα σημεία να καθορίζονται με βάση αυτό. Έτσι το σημείο που βρίσκεται κάποια στιγμή η χελώνα αλλά και το σημείο που θα βρεθεί μια επόμενη στιγμή, εξαρτώνται από το απόλυτο αρχικό σημείο.

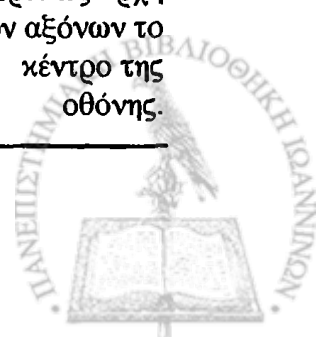
Με τη διαδικασία ΑΠΟΛΥΤΗ_ΘΕΣΗ μετατοπίζουμε κατευθείαν τη χελώνα στο σημείο που θέλουμε δηλώνοντας μόνο τις συντεταγμένες της θέσης του.

```
ΤΟ ΑΠΟΛΥΤΗ_ΘΕΣΗ :Χ :Υ
ΜΗ_ΓΡΑΦΕΙΣ
SETX :Χ
SETY :Υ
ΓΡΑΦΕ
ΕΝΔ
```

Η διαδικασία ΑΠΟΛΥΤΗ_ΘΕΣΗ χρησιμοποιεί τις εντολές SETX και SETY και μετακινεί τη χελώνα στη θέση που έχει ως συντεταγμένες αυτούς τους αριθμούς.

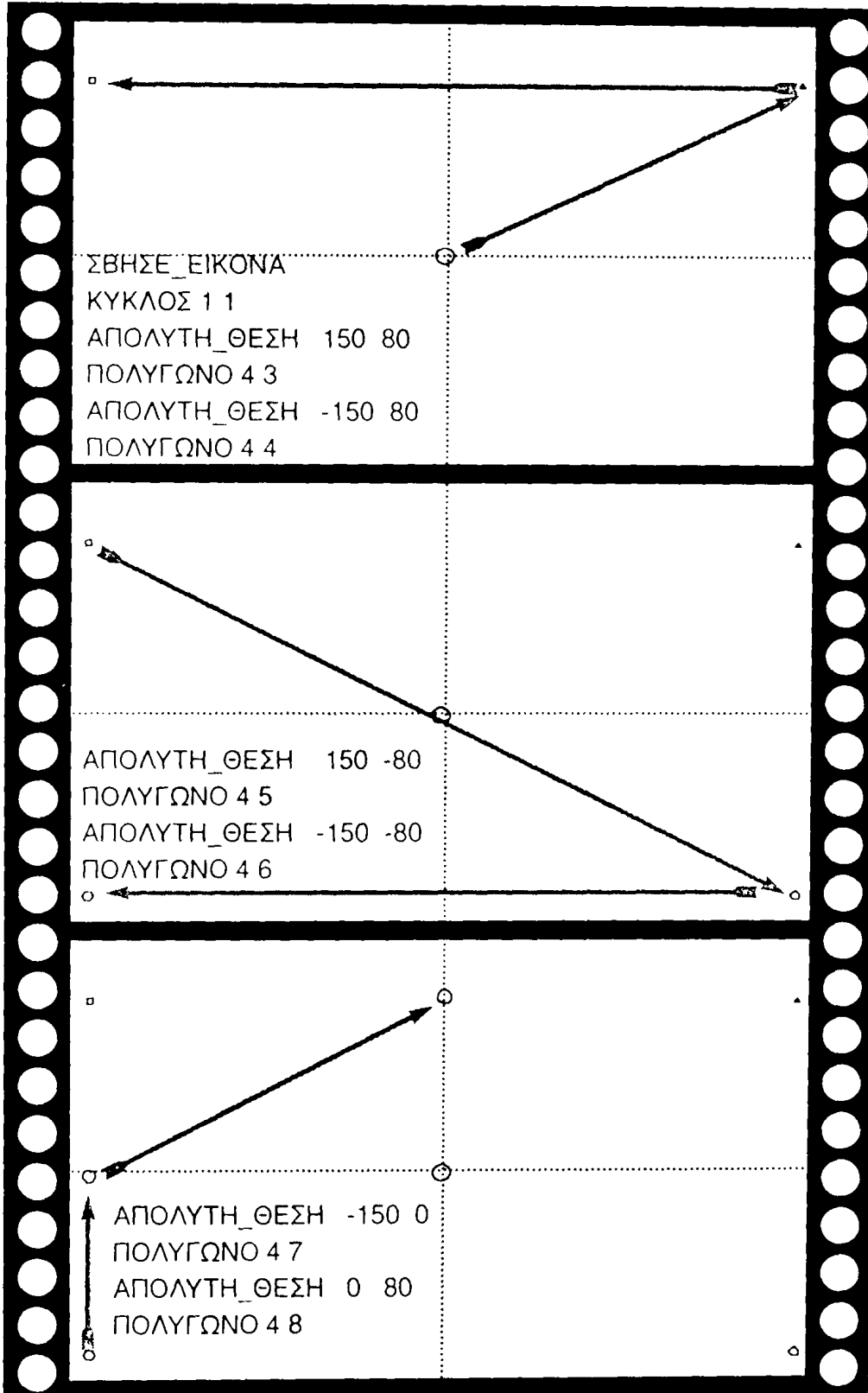


Η LCSI- logo
θεωρεί ως αρχή
των αξόνων το
κέντρο της
οθόνης.



Εξερευνώντας τα όρια της οθόνης

Με περαματισμούς όπως αυτοί που ακολουθούν μπορούμε να εξερευνήσουμε τα όρια της οθόνης του υπολογιστή για τη χελώνα.



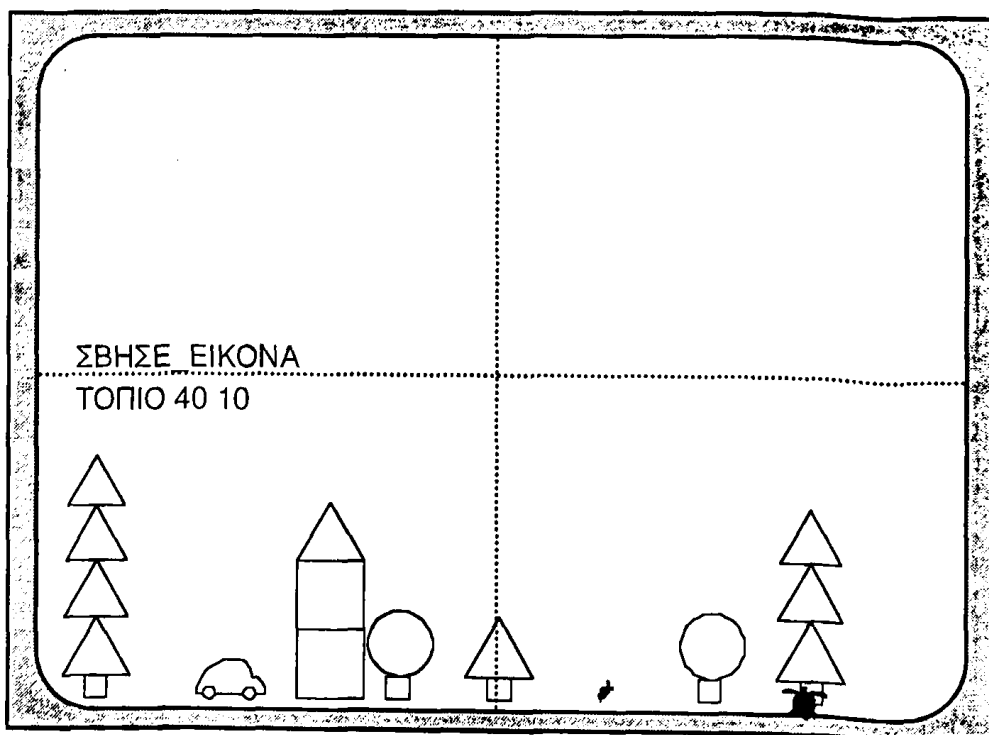
Πολλαπλές επιλογές

Ας φτιάξουμε μια διαδικασία που να σχεδιάζει ένα τοπίο με αυτοκίνητα, κτίρια, λουλούδια, ροδιές και έλατα επιλέγοντας στην τύχη τον εκάστοτε συνδυασμό τους. Η δομή IFELSE μας επιτρέπει να επιλέξουμε μεταξύ δύο δυνατοτήτων. Αν και η δεύτερη είναι μια εντολή IFELSE, τότε συνολικά οι επιλογές μας γίνονται τρεις. Αυτό μπορεί να συνεχιστεί και σε επόμενα στάδια.

```

TO ΤΟΠΙΟ :MEGEQOS :FORES
MAKE "ARXHΧ MINUS 150
MAKE "ARXHΥ MINUS 80
ΑΠΟΛΥΤΗ_ΘΕΣΗ :ARXHΧ :ARXHΥ
REPEAT :FORES
  [ ΜΕΤΑΤΟΠΙΣΗ (:MEGEQOS * 1.5) 0
  MAKE "MEG :MEGEQOS
  MAKE "ΕΡΙΛΟΓΗ RANDOM 6
  IFELSE :ΕΡΙΛΟΓΗ = 0
    [ ΣΚΑΡΑΒΑΙΟΣ :MEG ]
    [ IFELSE :ΕΡΙΛΟΓΗ = 1
      [ MAKE "ΟΡΟΦΟΙ RANDOM 3
        ΠΟΛΥΟΡΟΦΟ :MEG :ΟΡΟΦΟΙ + 1 ]
      [ IFELSE :ΕΡΙΛΟΓΗ = 2
        [ ΛΟΥΛΟΥΔΙ :MEG ]
        [ IFELSE :ΕΡΙΛΟΓΗ = 3
          [ ΡΟΔΙΑ :MEG / 3 ]
          [ ΕΛΑΤΟ :MEG ]
        ]
      ]
    ]
  ]
END
  
```

Η μεταβλητή ΕΡΙΛΟΓΗ παίρνει τυχαία μια τιμή από 0 μέχρι 5. Αν η τιμή είναι 0 τότε σχεδιάζεται ένα αυτοκίνητο, αλλιώς αν η τιμή είναι 1 τότε σχεδιάζεται ένα κτίριο, αλλιώς αν η τιμή είναι 2 τότε σχεδιάζεται ένα λουλούδι, αλλιώς αν η τιμή είναι 3 τότε σχεδιάζεται μια ροδιά αλλιώς (τιμή 4 ή 5) σχεδιάζεται ένα έλατο.



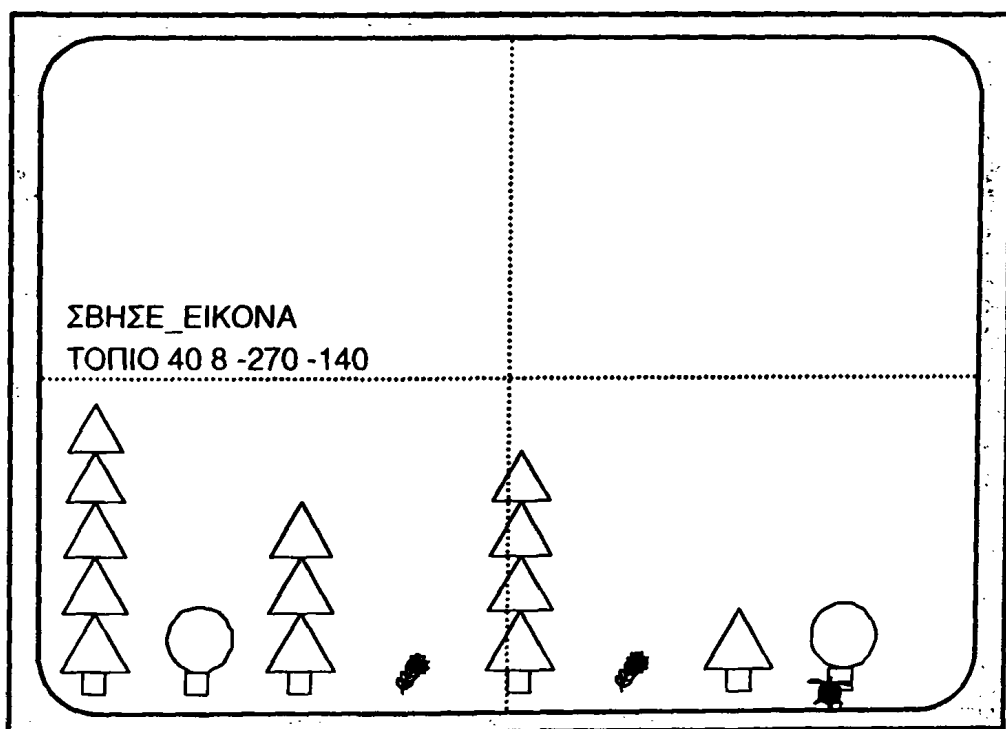
Βάλε το τοπίο όπου θέλεις

Μπορούμε να παραμετροποιήσουμε την προηγούμενη διαδικασία δίνοντας κατά την κλήση της τις απόλυτες συντεταγμένες από τις οποίες θα αρχίσει να σχεδιάζεται το τοπίο.

```

ΤΟ ΤΟΠΙΟ :ΜΕΓΕΘΟΣ :ΦΟΡΕΣ :ΑΡΧΗΧ :ΑΡΧΗΥ
ΑΠΟΛΥΤΗ_ΘΕΣΗ :ΑΡΧΗΧ :ΑΡΧΗΥ
ΡΕΡΕΑΤ :ΦΟΡΕΣ
  [ ΜΕΤΑΤΟΠΙΣΗ (:ΜΕΓΕΘΟΣ * 1.5) 0
    MAKE "ΜΕΓ :ΜΕΓ
    MAKE "ΕΡΙΛΟΓΗ RANDOM 8
    IFELSE :ΕΡΙΛΟΓΗ = 0
      [ ΣΚΑΡΑΒΑΙΟΣ :ΜΕΓ * 1.2 ]
      [ IFELSE :ΕΡΙΛΟΓΗ = 1
        [ MAKE "ΟΡΟΦΟΙ RANDOM 3
          ΠΟΛΥΟΡΟΦΟ :ΜΕΓ :ΟΡΟΦΟΙ + 1 ]
        [ IFELSE :ΕΡΙΛΟΓΗ = 2
          [ ΛΟΥΛΟΥΔΙ :ΜΕΓ * 2 ]
          [ IFELSE :ΕΡΙΛΟΓΗ = 3
            [ ΡΟΔΙΑ :ΜΕΓ / 3 ]
            [ ΕΛΑΤΟ :ΜΕΓ ]
          ]
        ]
      ]
    ]
  ]
END
    
```

Ο κώδικας του προγράμματος πρέπει να είναι ευανάγνωστος για να διευκολύνει τον προγραμματιστή να το κατανοεί εύκολα. Για το λόγο αυτό δε βλέπουμε να σπαταλώνται μερικές γραμμές (όπως γίνεται με τις αγκύλες που κλείνουν στη διπλανή διαδικασία) προκειμένου ο κώδικας του προγράμματος να είναι περισσότερο ευανάγνωστος.



Μια ολοκληρωμένη ζωγραφιά

Μπορούμε να γεμίσουμε την οθόνη με τοπία, που το ένα να βρίσκεται πίσω από το άλλο και να έχει μικρότερο μέγεθος ώστε να δημιουργείται η αίσθηση του βάθους.

Η διαδικασία ΖΩΓΡΑΦΙΑ, σχεδιάζει τα τοπία με μια επανάληψη που καθορίζεται από την παράμετρο ΒΑQ. Σε κάθε επανάληψη οι συντεταγμένες από τις οποίες αρχίζει η σχεδίαση του τοπίου μεταβάλλονται, καθώς επίσης και το μέγεθος των αντικειμένων του τοπίου.

```
TO ΖΩΓΡΑΦΙΑ :MEG :FORA :ARXHΧ :ARXHΥ :BAQ
```

```
REPEAT :BAQ
```

```
  [ΤΟΠΙΟ :MEG :FORA :ARXHΧ :ARXHΥ
```

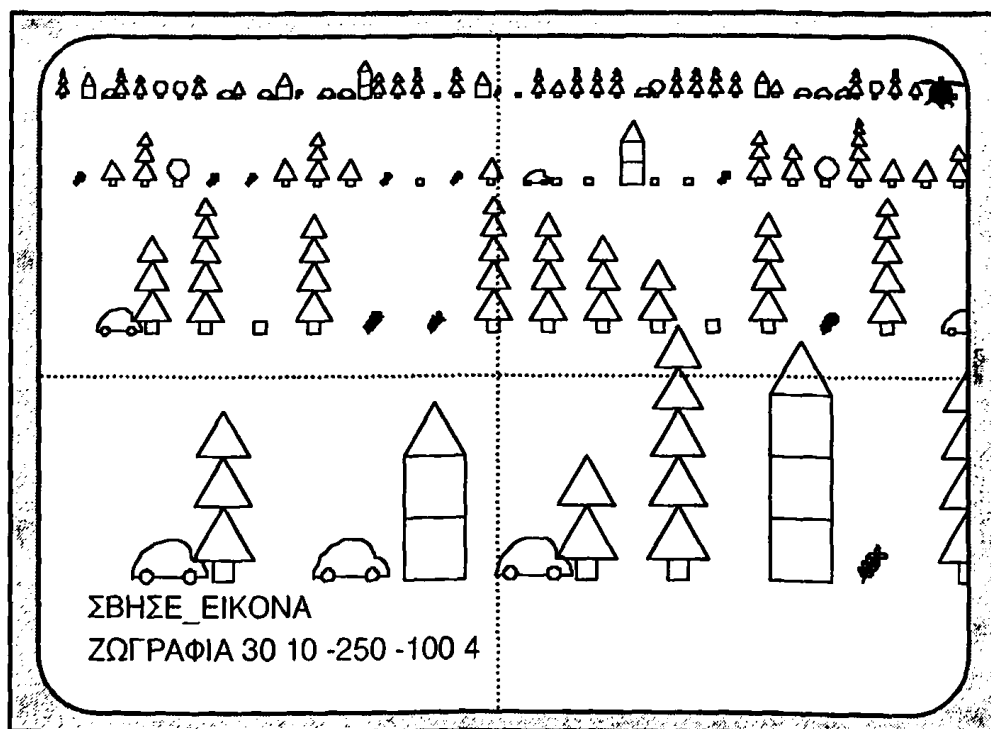
```
  MAKE "ARXHΧ :ARXHΧ
```

```
  MAKE "ARXHΥ :ARXHΥ + (4 * :MEG)
```

```
  MAKE "FORA :FORA * 1.7
```

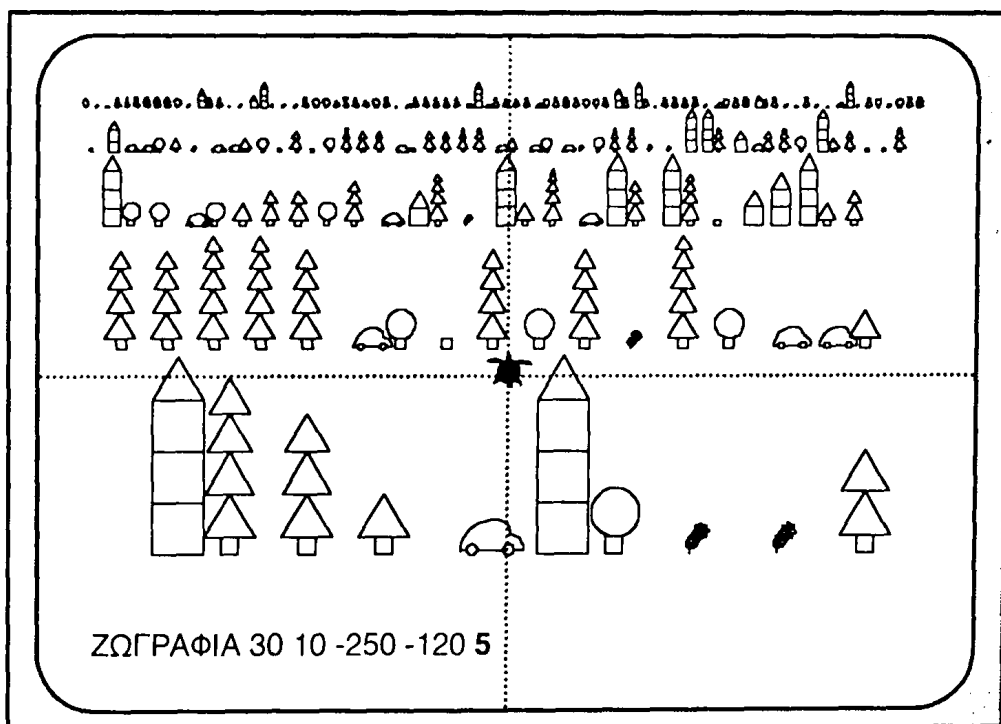
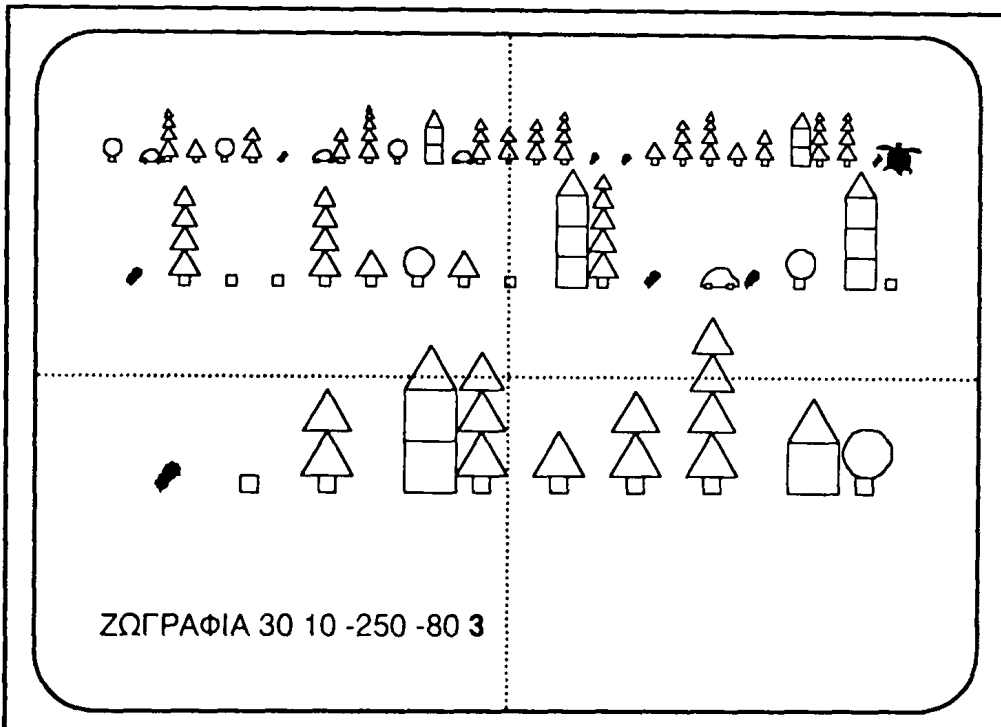
```
  MAKE "MEG :MEG * 0.6]
```

```
END
```



Η δημιουργικότητα της χελώνας

Ας αφήσουμε τη χελώνα να δημιουργήσει τη δική της πινακοθήκη, σχεδιάζοντας με τη βοήθεια της τύχης μερικές ζωγραφιές.



13 Αναδρομικές διαδικασίες

Ένα δυαδικό δένδρο

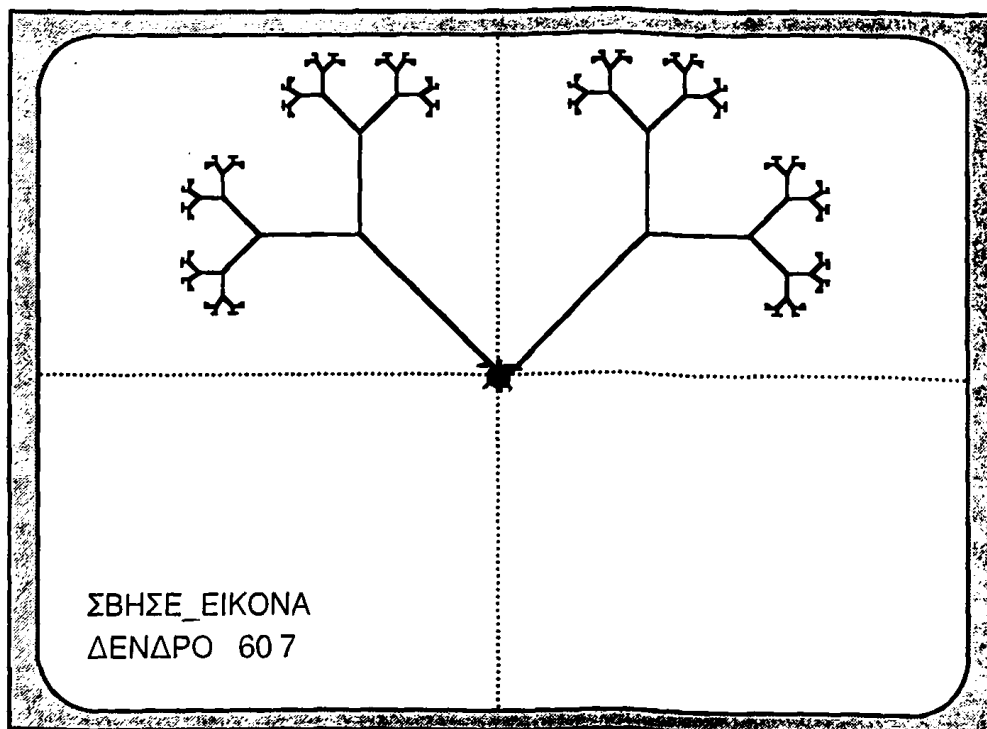
Η διαδικασία ΔΕΝΔΡΟ που περιγράφεται παρακάτω σχηματίζει την εικόνα ενός δυαδικού δένδρου.

Η διαδικασία λειτουργεί αναδρομικά καλώντας τον εαυτό της δύο φορές, μια για τα αριστερά κλαδιά και μια για τα δεξιά.

Το «βάθος» που αναπαράγονται τα κλαδιά του δένδρου καθορίζεται από τη μεταβλητή GENIA, που ελέγχει πότε θα σταματήσει η αναδρομική κλήση της διαδικασίας.

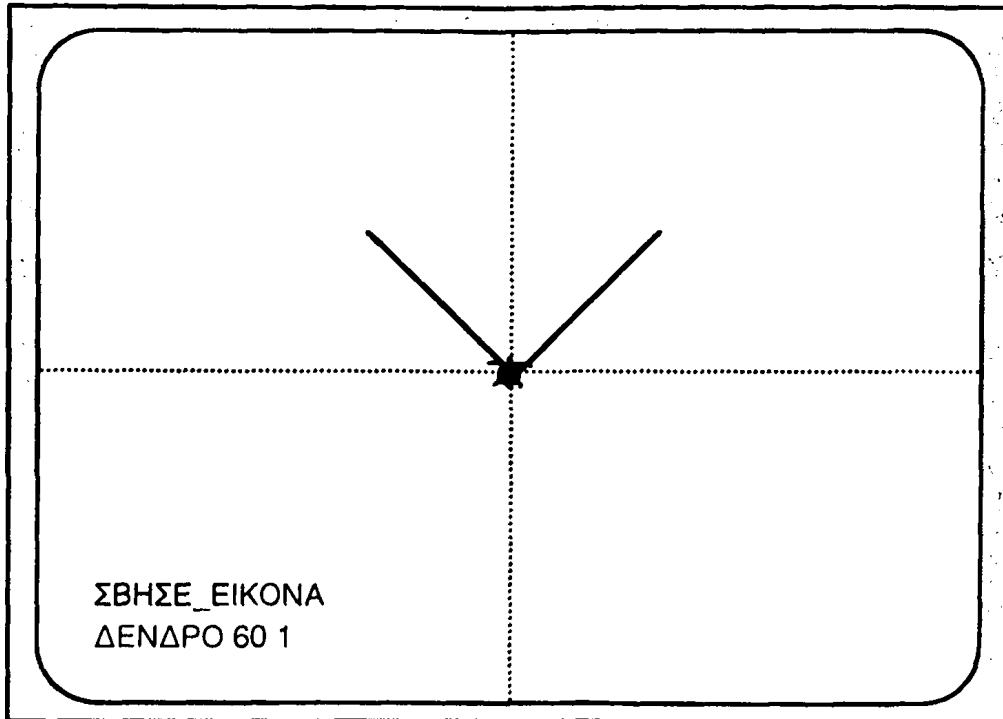
```
TO ΔΕΝΔΡΟ :KLADI :GENIA
IF :GENIA > 0
  [ ΓΡΑΜΜΗ 315 :KLADI
  ΔΕΝΔΡΟ :KLADI * (1 / 2) :GENIA - 1
  ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :KLADI
  ΓΡΑΜΜΗ 90 :KLADI
  ΔΕΝΔΡΟ :KLADI * (1 / 2) :GENIA - 1
  ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :KLADI
  ΑΡΙΣΤΕΡΑ 45 ]
END
```

Το μέγεθος του δένδρου καθορίζεται από την τιμή της μεταβλητής KLADI. Σε κάθε επόμενη κλήση της διαδικασίας ΔΕΝΔΡΟ το μήκος του κλαδιού μειώνεται στο μισό.

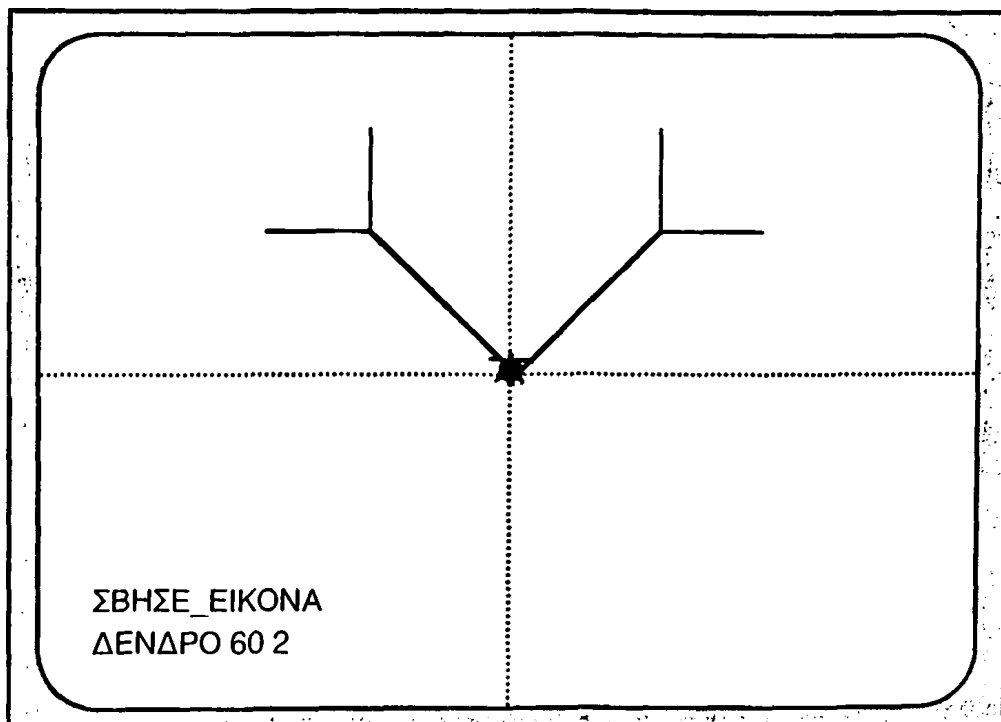


Διαφορετικές γενιές δυαδικών δένδρων

Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 1 , σχηματίζεται η πρώτη γενιά των κλαδιών.

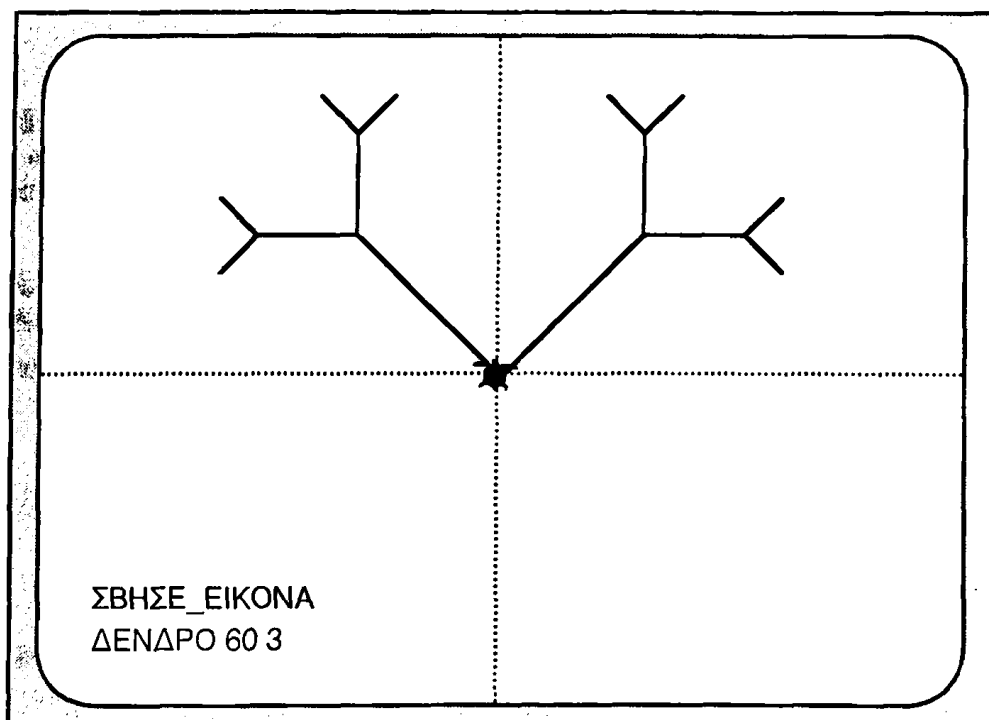


Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 2 , εκτός από την πρώτη γενιά των κλαδιών σχηματίζονται και τα κλαδιά της δεύτερης.

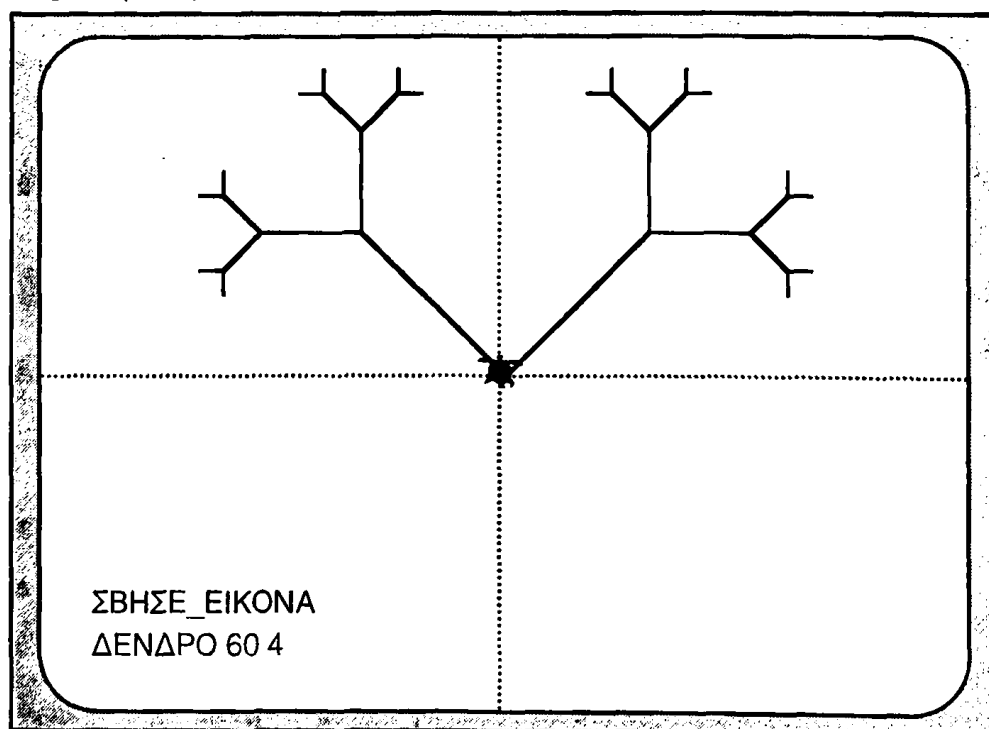


Διαφορετικές γενιές δυαδικών δένδρων

Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 3, εκτός από την πρώτη και δεύτερη γενιά των κλαδιών σχηματίζονται και τα κλαδιά της τρίτης.



Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 4, εκτός από την πρώτη, δεύτερη και τρίτη γενιά των κλαδιών, σχηματίζονται και τα κλαδιά της επόμενης.



Πως δουλεύει η αναδρομή

Μια τεχνική για να κατανοήσουμε τον τρόπο λειτουργίας ενός μηχανισμού, είναι να τον παρακολουθήσουμε βήμα προς βήμα όταν δουλεύει. Αυτό θα κάνουμε για τη διαδικασία ΔΕΝΔΡΟ. Για να μπορέσουμε να παρακολουθήσουμε την εξέλιξη του προγράμματος, θα αριθμήσουμε τις γραμμές του και θα προσθέσουμε μερικές εντολές που θα μας δείχνουν το εκάστοτε σημείο της ροής.

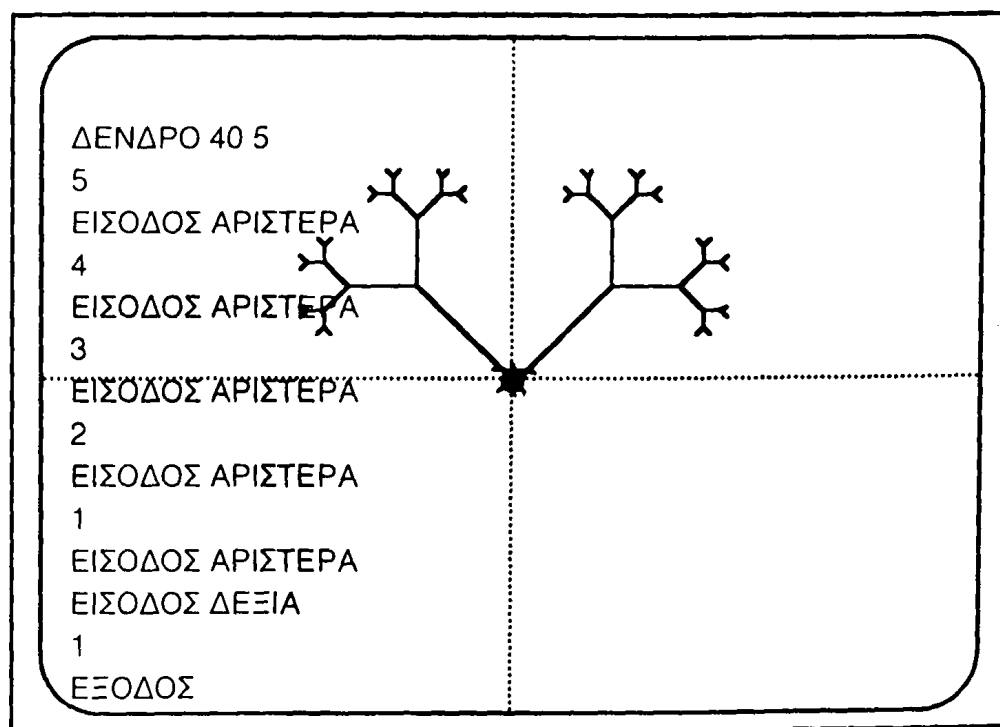
```

1  ΤΟ ΔΕΝΔΡΟ :ΚΛΑΔΙ :ΓΕΝΙΑ
2  IF :ΓΕΝΙΑ > 0
3      [PRINT :ΓΕΝΙΑ
4      PRINT "είσοδος αριστερά
5      ΓΡΑΜΜΗ 315 :ΚΛΑΔΙ
6      MAKE "Q READCHAR
7      ΔΕΝΔΡΟ :ΚΛΑΔΙ * (1 / 2) :ΓΕΝΙΑ - 1
8      ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :ΚΛΑΔΙ
9
10     PRINT "είσοδος δεξιά
11     ΓΡΑΜΜΗ 90 :ΚΛΑΔΙ
12     MAKE "Q READCHAR
13     ΔΕΝΔΡΟ :ΚΛΑΔΙ * (1 / 2) :ΓΕΝΙΑ - 1
14     ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :ΚΛΑΔΙ
15     ΑΡΙΣΤΕΡΑ 45
16     PRINT :ΓΕΝΙΑ
17     PRINT "έξοδος
18     MAKE "Q READCHAR]
19  END
    
```

Οι εντολές των γραμμών 3, 4, 10, 16, 17 εμφανίζουν στην οθόνη μηνύματα που μας βοηθούν να καταλάβουμε ποιό σημείο του προγράμματος εκτελείται την αντίστοιχη στιγμή.

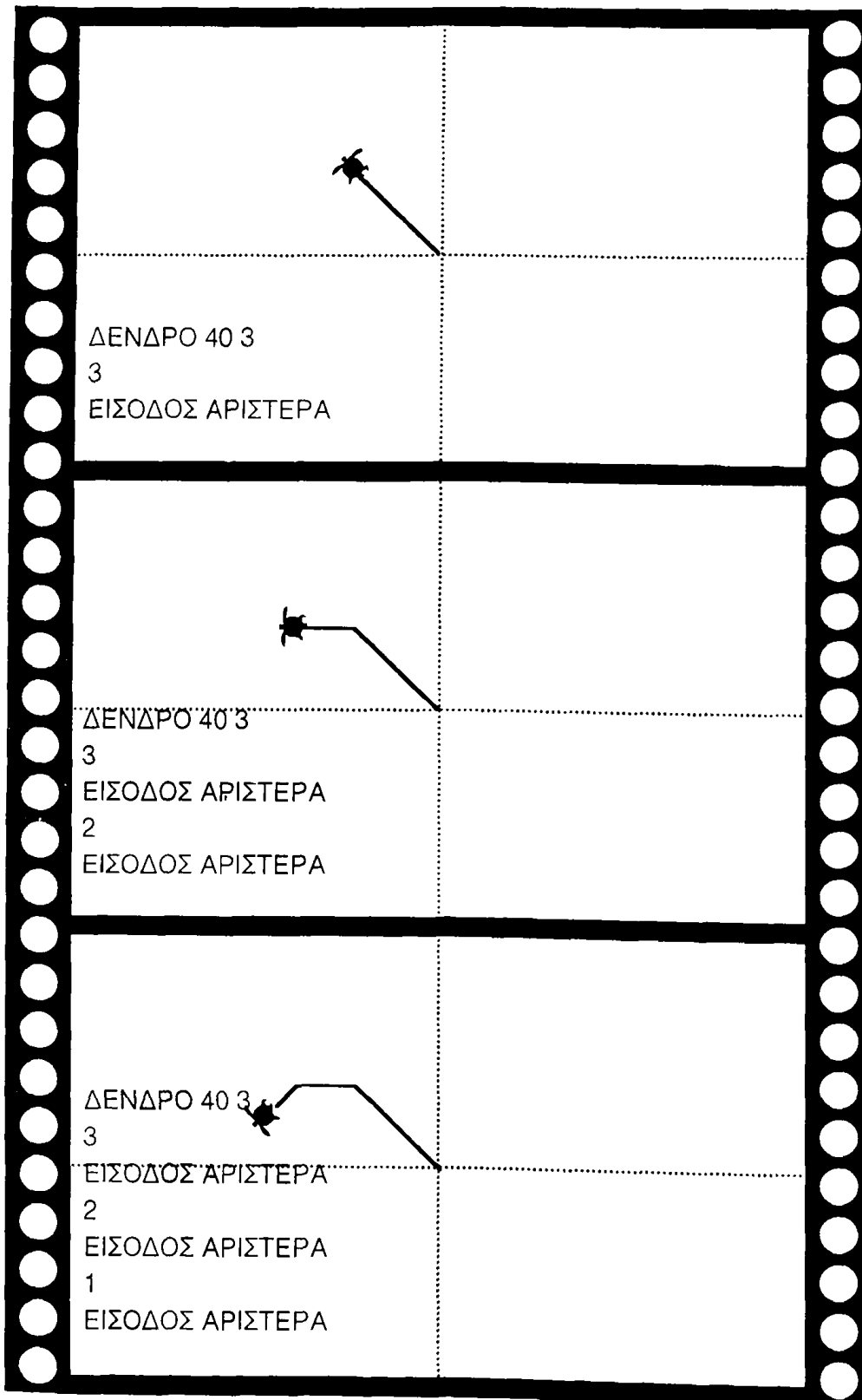
Οι εντολές των γραμμών 6, 12, 18 σταματούν τη ροή του προγράμματος μέχρι να δώσει ο χρήστης κάποιο δεδομένο από το πληκτρολόγιο.

Αυτό το τέχνασμα χρησιμοποιείται όταν θέλουμε να τρέξουμε κομμάτι-κομμάτι το πρόγραμμα για να το παρακολουθήσουμε.

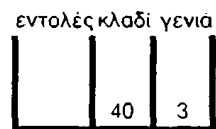


Αναρρίχηση σε δυαδικό δένδρο (1)

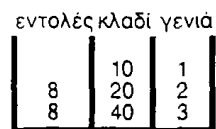
Ας παρακολουθήσουμε την εκτέλεση της διαδικασίας ΔΕΝΔΡΟ στην οθόνη (σχέδιο και μηνύματα) και στη μνήμη του υπολογιστή.



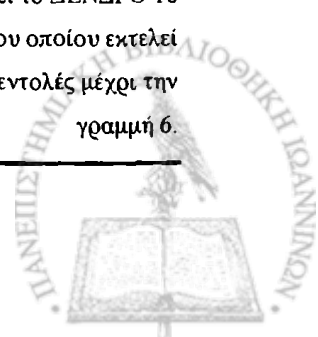
Το ΔΕΝΔΡΟ 40 3
βάζει στη μεταβλητή
ΚΛΑΔΙ το 40. στη
GENIA το 3 και
εκτελεί τις εντολές
μέχρι τη γραμμή 6.
Εκεί σταματάει την
εκτέλεση και
αναμένει πάτημα
πληκτρού από το
πληκτρολόγιο για να
συνεχίσει.



Η γραμμή 7
διακόπτει τη ροή της
ΔΕΝΔΡΟ 40 3 και
αφού φυλάξει τον
αριθμό γραμμής 8.
καλεί το
ΔΕΝΔΡΟ 20 2 του
οποίου εκτελεί τις
εντολές μέχρι τη
γραμμή 6.

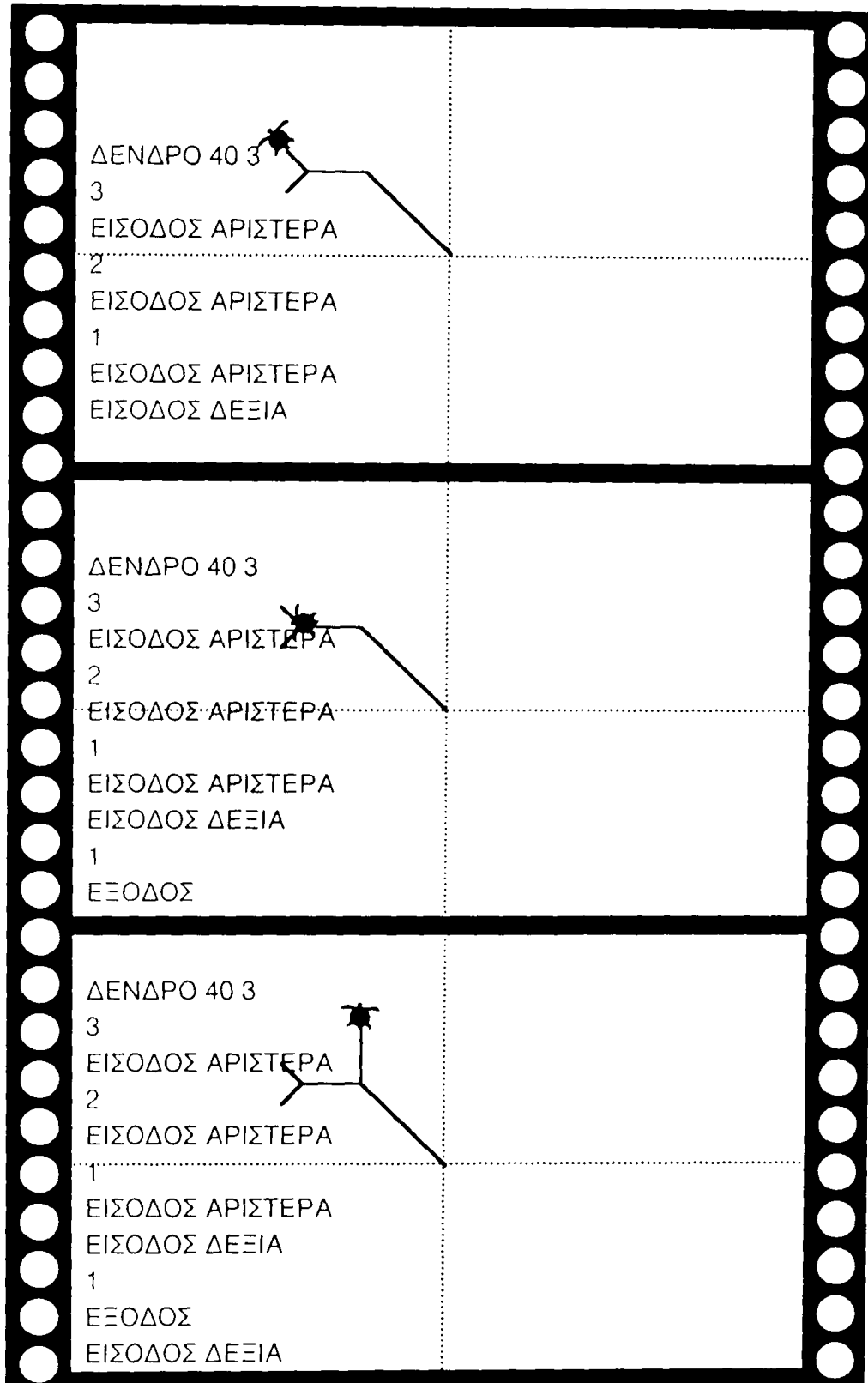


Η γραμμή 7
διακόπτει τη ροή της
ΔΕΝΔΡΟ 20 2 και
αφού φυλάξει τον
αριθμό γραμμής 8
καλεί το ΔΕΝΔΡΟ 10
1 του οποίου εκτελεί
τις εντολές μέχρι την
γραμμή 6.



Αναρρίχηση σε δυαδικό δένδρο (2)

Η γελώνα συνεχίζει τη σχεδίαση του αριστερού κλαδιού του δένδρου με όλα τα παρακλάδια του.



Όμοια καλείται η ΔΕΝΔΡΟ 5 0 και εκτελεί τις γραμμές 1. 2. 19.

εντολές κλαδι γενιά

	5	0
8	10	1
8	20	2
8	40	3

Το END πετάει τις τιμές 5. 0 και παίρνει από τη στοίβα την πιο πάνω εντολή (8).

Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 10 1.

εντολές κλαδι γενιά

	10	1
8	20	2
8	40	3

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 5 0 και αφού φυλάξει το 14 εκτελεί τις γραμμές 1. 2. 19.

εντολές κλαδι γενιά

	5	0
14	10	1
8	20	2
8	40	3

Το END πετάει τις τιμές 5. 0 και παίρνει από τη στοίβα το 14. Έτσι εκτελεί τις γραμμές 14-18 της ΔΕΝΔΡΟ 10 1.

εντολές κλαδι γενιά

	10	1
8	20	2
8	40	3

Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 10 1 πετάει τις τιμές 10. 1 και παίρνει από τη στοίβα το 8.

Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 20 2.

εντολές κλαδι γενιά

	20	2
8	40	3



Αναρρίχηση σε δυαδικό δένδρο (4)

Η γελώνα επέστρεψε στη ρίζα και από εκεί ξεκινά να σχεδιάζει το δεξί κλαδί του δένδρου.

<p>1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΞΟΔΟΣ 2 ΕΞΟΔΟΣ</p>	
<p>ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΞΟΔΟΣ 2 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p>	
<p>1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>1 ΕΞΟΔΟΣ 2 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p> <p>2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ</p>	

Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 10 1 πετάει τις τιμές 10. 1 και παίρνει από τη στοίβα το 14. Έτσι εκτελεί τις γραμμές 15-18 της ΔΕΝΔΡΟ 20 2.

εντολές κλαδι γενιά

8	20 40	2 3
---	----------	--------

Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 20 2 πετάει τις τιμές 20. 2 και παίρνει από τη στοίβα το 8. Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 40 3.

εντολές κλαδι γενιά

	40	3
--	----	---

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 20 2 και αφού φυλάξει τον αριθμό γραμμής 14 εκτελεί τις εντολές μέχρι τη γραμμή 6 του ΔΕΝΔΡΟ 20 2.

εντολές κλαδι γενιά

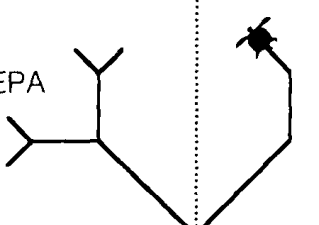
14	20 40	2 3
----	----------	--------



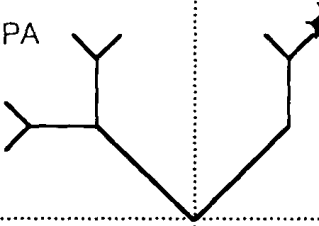
Αναρρίχηση σε δυαδικό δένδρο (5)

Η χελώνα συνεχίζει τη σχεδίαση του δεξιού κλαδιού του δένδρου με όλα τα παρακλάδια του.

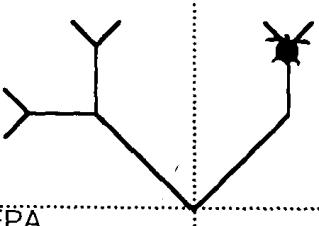
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
1
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
1
ΕΞΟΔΟΣ
2



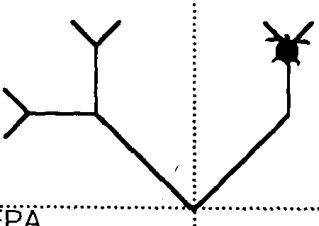
ΕΞΟΔΟΣ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
2
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
1
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ



1
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
1
ΕΞΟΔΟΣ
2
ΕΞΟΔΟΣ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
2
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
1
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ



ΕΙΣΟΔΟΣ ΔΕΞΙΑ
1
ΕΞΟΔΟΣ
2
ΕΞΟΔΟΣ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
2
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
1
ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ
ΕΙΣΟΔΟΣ ΔΕΞΙΑ
1
ΕΞΟΔΟΣ



Η γραμμή 7 διακόπτει τη ροή της ΔΕΝΔΡΟ 20 και αφού φυλάξει τον αριθμό γραμμής 8 καλεί το ΔΕΝΔΡΟ 10 1 του οποίου εκτελεί τις εντολές μέχρι τη γραμμή 6.

εντολές κλαδί γενιά		
8	10	1
14	20	2
	40	3

Όμοια καλείται η ΔΕΝΔΡΟ 5 0 και εκτελεί τις γραμμές 1. 2. 19.

εντολές κλαδί γενιά		
8	5	0
8	10	1
14	20	2
	40	3

Το END πετάει τις τιμές 5. 0 και παίρνει την πιο πάνω εντολή από τη στοίβα. Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 10 1.

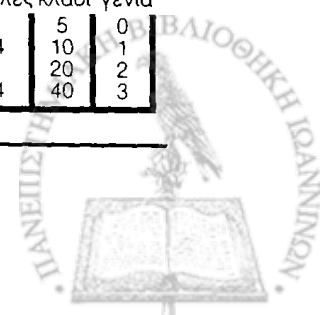
εντολές κλαδί γενιά		
8	10	1
14	20	2
	40	3

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 5 0 και αφού φυλάξει το 14 εκτελεί τις γραμμές 1. 2. 19.

εντολές κλαδί γενιά		
14	5	0
8	10	1
14	20	2
	40	3

Το END πετάει τις τιμές 5.0 και παίρνει από τη στοίβα το 14. Έτσι εκτελεί τις γραμμές 14-18 της ΔΕΝΔΡΟ 10 1.

εντολές κλαδί γενιά		
14	5	0
8	10	1
14	20	2
	40	3



Αναρρίχηση σε δυαδικό δένδρο (6)

Η γελώνα ολοκληρώνει τη σχεδίαση του δεξιού κλαδιού του δένδρου με όλα τα παρακλάδια του.

1 ΕΞΟΔΟΣ 2 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ	
2 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ	
ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ	

Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 10 1 πετάει τις τιμές 10. 1 και παίρνει από τη στοιβα το 8 Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 20 2.

εντολές κλαδι γενιά

14	20	2
	40	3

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 10 1 και αφού φυλάξει τον αριθμό γραμμής 14 εκτελεί τις εντολές μέχρι τη γραμμή 6 του ΔΕΝΔΡΟ 10 1.

εντολές κλαδι γενιά

14	10	1
14	20	2
	40	3

Η γραμμή 7 καλεί τη ΔΕΝΔΡΟ 5 0 και εκτελεί τις γραμμές 1. 2. 19.

εντολές κλαδι γενιά

8	5	0
14	10	1
14	20	2
	40	3

Το END πετάει τις τιμές 5.0 και παίρνει την πιο πάνω εντολή από τη στοιβα. Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 10 1.

εντολές κλαδι γενιά

14	10	1
14	20	2
	40	3



Ποικιλία δυαδικών δένδρων

Αφού κατανοήσαμε τον τρόπο λειτουργίας της αναδρομικής διαδικασίας ΔΕΝΔΡΟ, ας εισάγουμε δύο νέες μεταβλητές που θα την κάνουν περισσότερο ευέλικτη.

Η μεταβλητή **ΦΟΥΝΤ** που οι τιμές της θα είναι σε μοίρες, θα καθορίζει το πόσο φουντωτό θα είναι το δένδρο επηρεάζοντας τη γωνία μεταξύ δύο κλαδιών της ίδιας γενιάς.

Η άλλη παράμετρος **ΚΛΑΣΜΑ**, θα καθορίζει το ποσοστό της μεταβολής του μήκους των κλαδιών από γενιά σε γενιά.

ΤΟ ΔΕΝΔΡΟ :ΚΛΑΔΙ :ΓΕΝΙΑ :ΦΟΥΝΤΑ :ΚΛΑΣΜΑ

IF :ΓΕΝΙΑ > 0

[ΓΡΑΜΜΗ (360 - :ΦΟΥΝΤΑ) :ΚΛΑΔΙ

ΔΕΝΔΡΟ :ΚΛΑΔΙ * :ΚΛΑΣΜΑ :ΓΕΝΙΑ - 1 :ΦΟΥΝΤΑ :ΚΛΑΣΜΑ

ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :ΚΛΑΔΙ

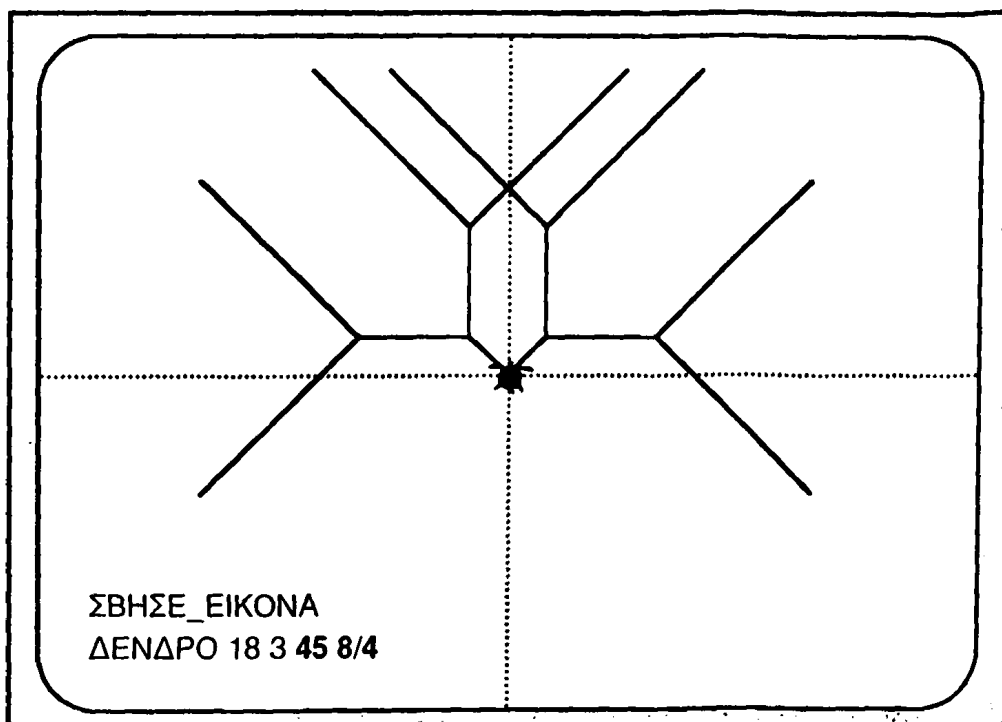
ΓΡΑΜΜΗ (2 * :ΦΟΥΝΤΑ) :ΚΛΑΔΙ

ΔΕΝΔΡΟ :ΚΛΑΔΙ * :ΚΛΑΣΜΑ :ΓΕΝΙΑ - 1 :ΦΟΥΝΤΑ :ΚΛΑΣΜΑ

ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :ΚΛΑΔΙ

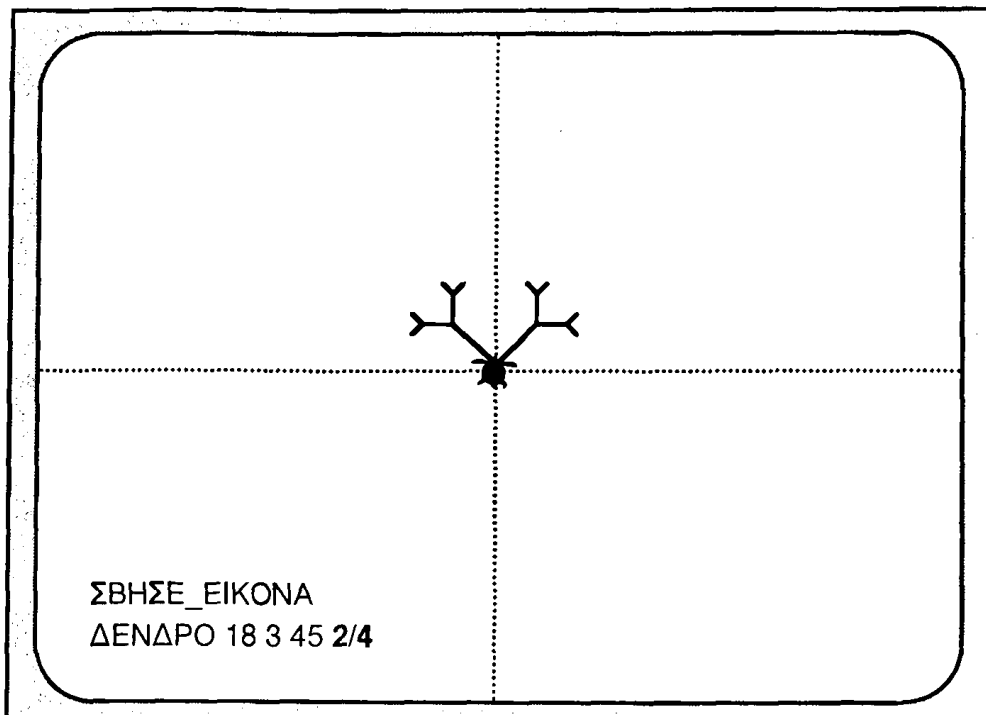
ΑΡΙΣΤΕΡΑ :ΦΟΥΝΤΑ]

END

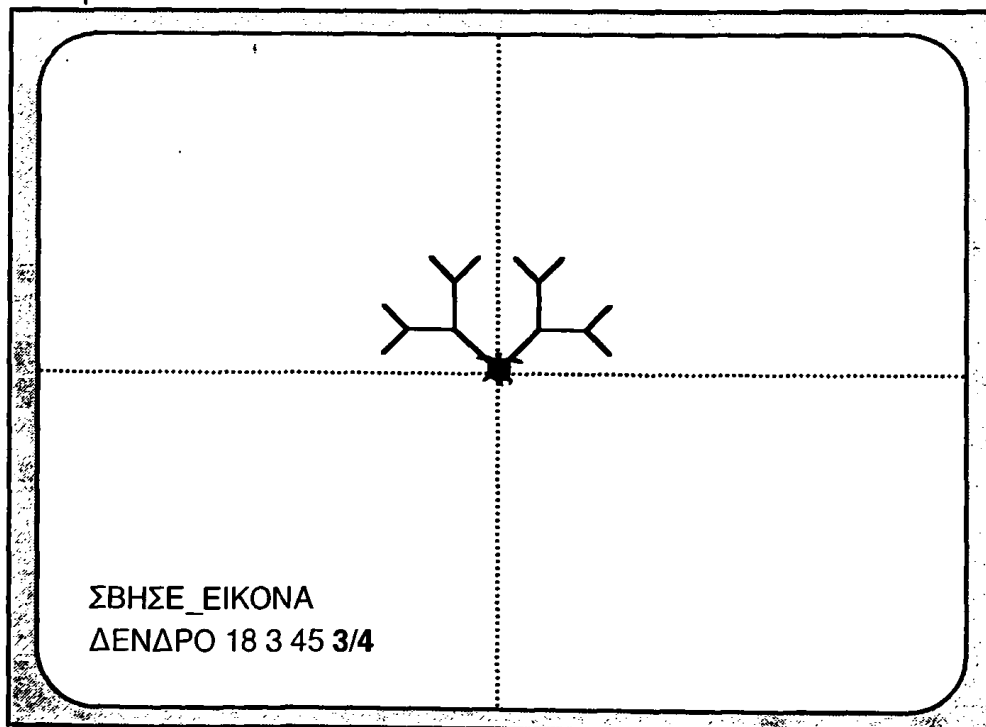


Παίζοντας με τις παραμέτρους (1)

Για να έχουμε ένα μέτρο σύγκρισης, θα ζητήσουμε από τη χελώνα να σχηματίσει ένα δένδρο που η γωνία μεταξύ των κλαδιών της ίδιας γενιάς να είναι 45 μοίρες και που το μήκος του κλαδιού κάθε επόμενης γενιάς να είναι το μισό του μήκους της προηγούμενης.

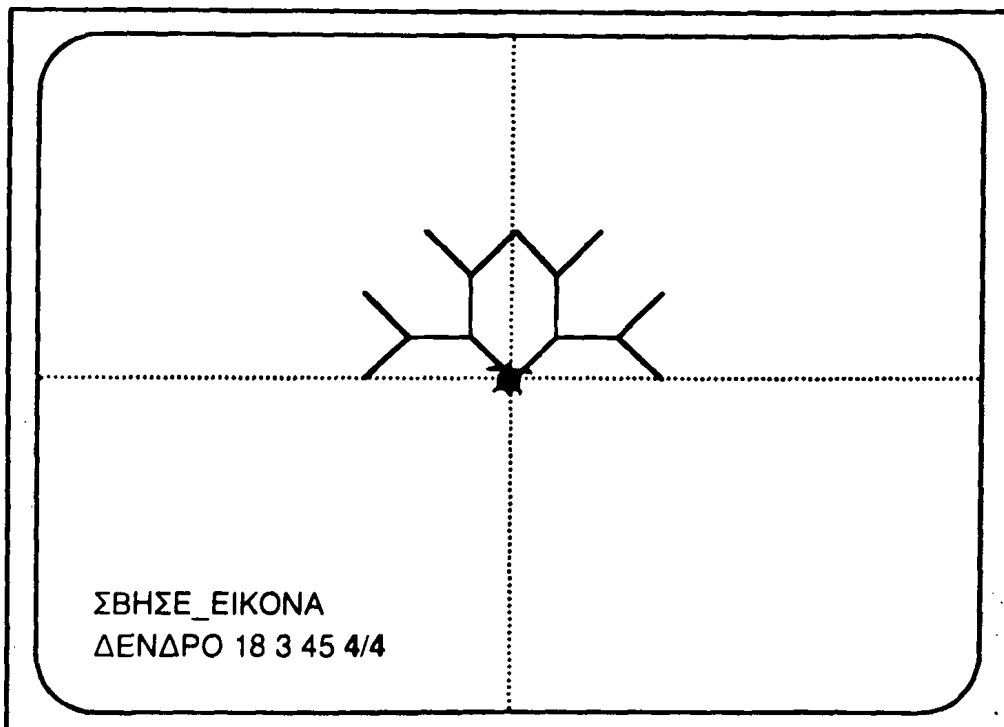


Στο επόμενο δένδρο θα διατηρήσουμε σταθερή τη γωνία μεταξύ των κλαδιών (45 μοίρες) και θα αλλάξουμε το ποσοστό της μεταβολής του μήκους του κλαδιού από γενιά σε γενιά κάνοντάς το 3/4, από 2/4 που ήταν.

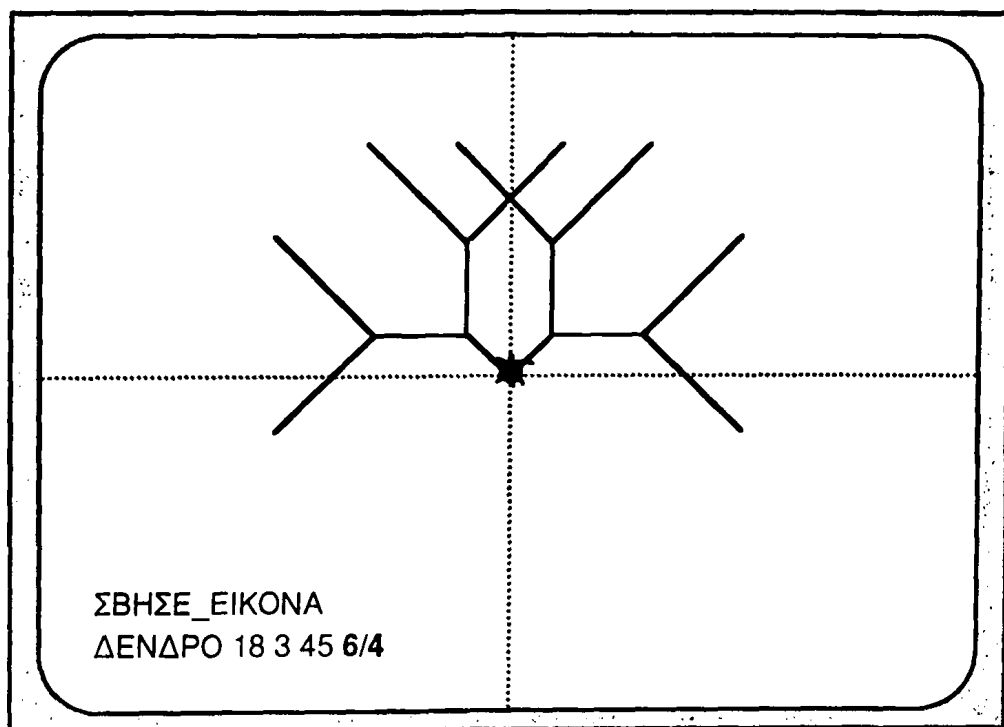


Παίζοντας με τις παραμέτρους (2)

Στο δένδρο που θα σχηματισθεί θα διατηρήσουμε σταθερό το μήκος των κλαδιών από γενιά σε γενιά, δίνοντας τα 4/4 ως τιμή στην παράμετρο ΚΛΑΣΜΑ.

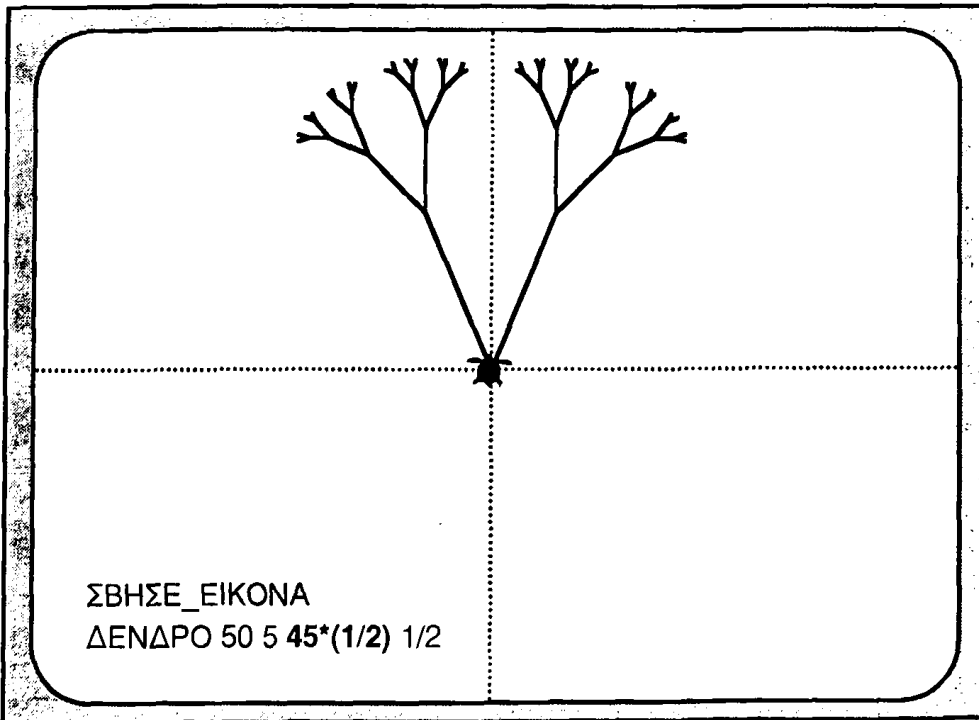


Στο επόμενο δένδρο, το μήκος των κλαδιών από γενιά σε γενιά θα αυξάνει δίνοντας τα 6/4 ως τιμή στην παράμετρο ΚΛΑΣΜΑ.

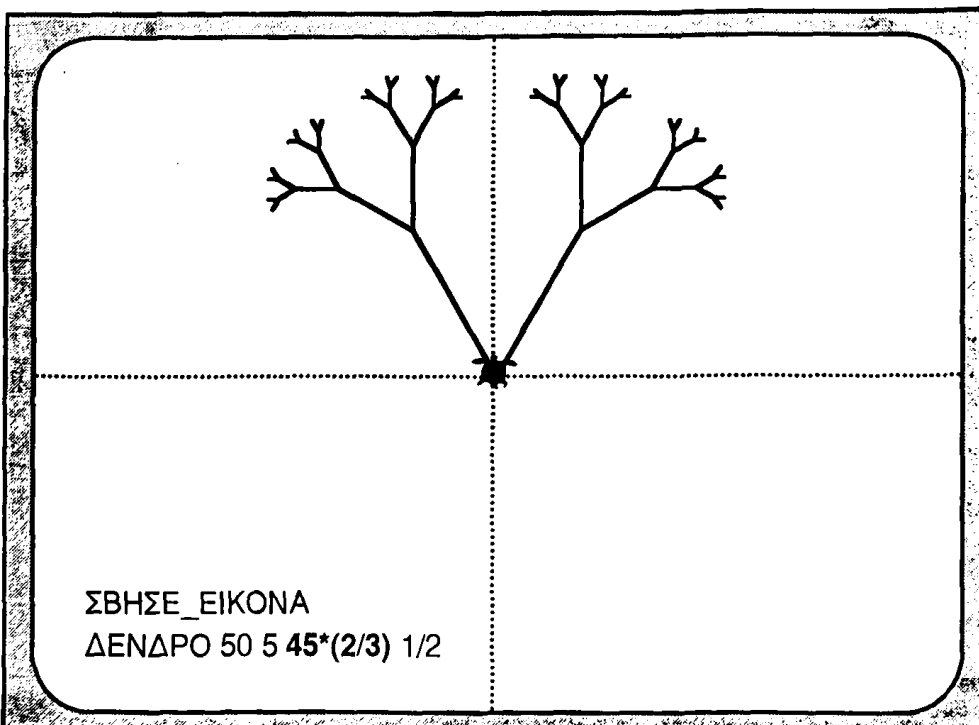


Παίζοντας με τις παραμέτρους (3)

Στη συνέχεια θα δώσουμε διάφορες τιμές στην παραμέτρο **ΦΟΥΝΤ** της διαδικασίας **ΔΕΝΔΡΟ**. Ας δούμε το δένδρο που η γωνία μεταξύ των κλαδιών θα είναι το $1/2$ των 45 μοιρών.

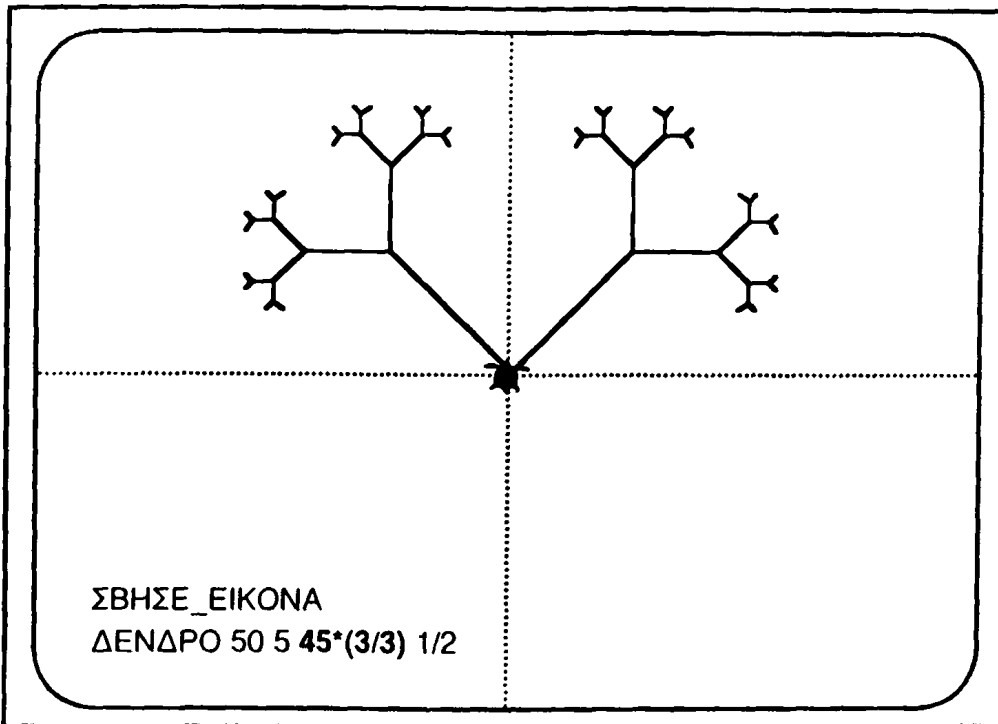


Αλλάζοντας τη γωνία μεταξύ των κλαδιών από το $1/2$ στα $2/3$ των 45 μοιρών, παίρνουμε το παρακάτω σχήμα.

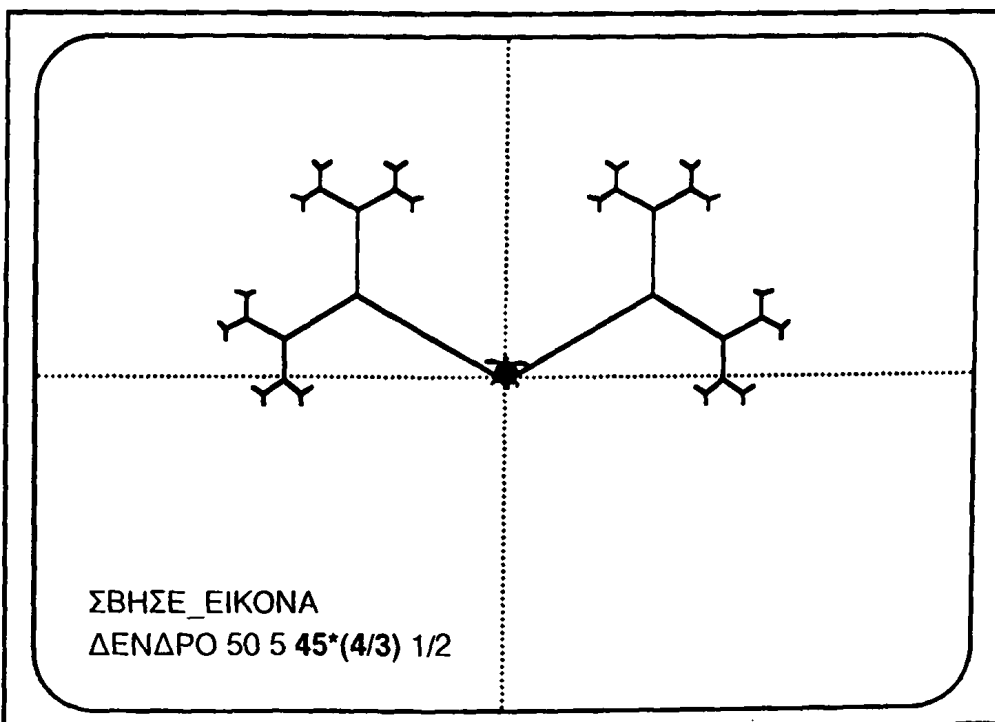


Παίζοντας με τις παραμέτρους (4)

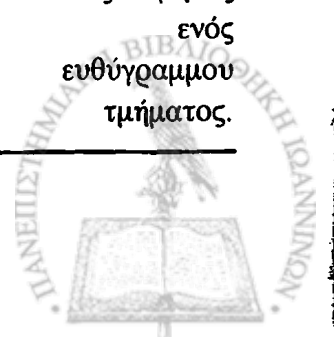
Το επόμενο σχήμα θα σχηματισθεί αν αναφέρουμε τη γωνία μεταξύ των κλαδιών στις 45 μοίρες.



Αυξάνοντας τη γωνία μεταξύ των κλαδιών στα 4/3 των 45 μοιρών, θα πάρουμε το παρακάτω σχήμα.



Τα δυαδικά δένδρα δίνουν μια άλλη οπτική στις αναπαραστάσεις των κλασμάτων. Η έννοια του κλάσματος ως τμήμα κάποιου διδιάστατου (επιπέδου) σχήματος μεταφέρεται στη μια διάσταση, ως το μήκος ενός ευθύγραμμου τμήματος.



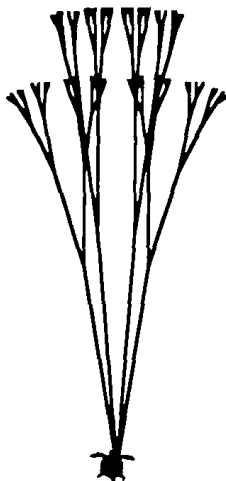
Κατ' εικόνα και ομοίωση

Δίνοντας διάφορες τιμές στις παραμέτρους, θα προσπαθήσουμε να δημιουργήσουμε σχήματα που να μοιάζουν με πραγματικά φυτά.

ΣΒΗΣΕ_ΕΙΚΟΝΑ

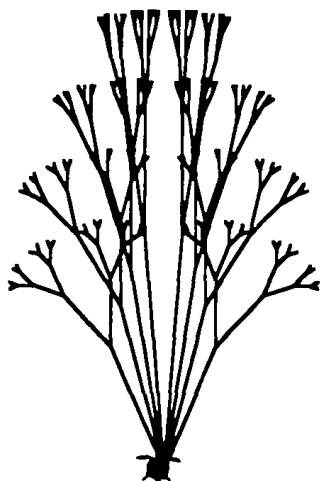
ΔΕΝΔΡΟ 70 5 (45/10) 1/2

ΔΕΝΔΡΟ 60 5 (45/5) 1/2



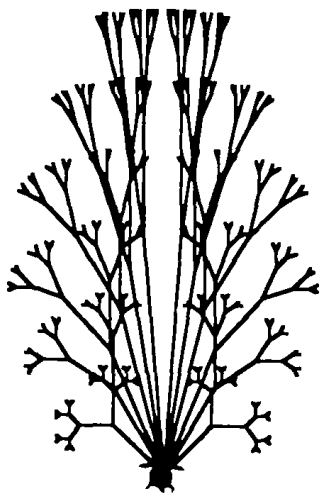
ΔΕΝΔΡΟ 50 5 (45/3) 1/2

ΔΕΝΔΡΟ 40 5 (45/2) 1/2



ΔΕΝΔΡΟ 50 5 (45/(3/2)) 1/2

ΔΕΝΔΡΟ 40 5 (45/1) 1/2



«Αυτά τα σχήματα έχουν πολλές ενδιαφέρουσες αναφορές στη βιολογία. Είναι εύκολο να δούμε πως μπορούμε με τη βοήθειά τους να ερευνήσουμε πολύ απλά τα κελύφη των σαλιγκαριών, τις κορυφές των σπόρων, τα κέρατα των ζώων και άλλα παρόμοια φαινόμενα. Η εργασία αυτή συνδέεται και με εκπαιδευτικά θέματα υψηλότερου επιπέδου...»
Boris Allan.

Η ανάπτυξη ενός φυτού

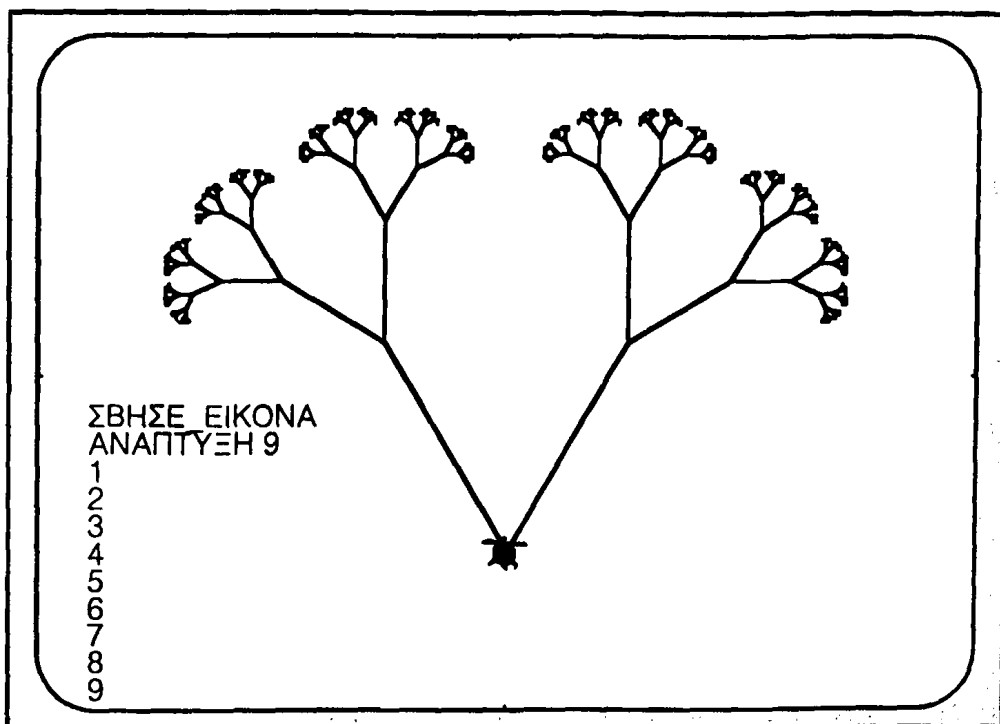
Μπορούμε να προσομοιώσουμε την ανάπτυξη ενός φυτού με μια διαδικασία που θα σχηματίζει διαδοχικά τις εικόνες του φυτού με το πέρασμα του χρόνου. Μια διαδικασία θα επαναλαμβάνει μερικές φορές (που θα αντιστοιχούν σε χρονικές στιγμές) τη διαδικασία ΔΕΝΔΡΟ και θα μεταβάλλει κάθε φορά τις τιμές των παραμέτρων της, έτσι ώστε να φαίνεται ότι το φυτό αναπτύσσεται.

Η επιτυχία της προσομοίωσης θα εξαρτηθεί από το βαθμό μεταβολής της κάθε παραμέτρου της διαδικασίας ΔΕΝΔΡΟ.

Ας δούμε μια τέτοια διαδικασία, την ΑΝΑΠΤΥΞΗ.

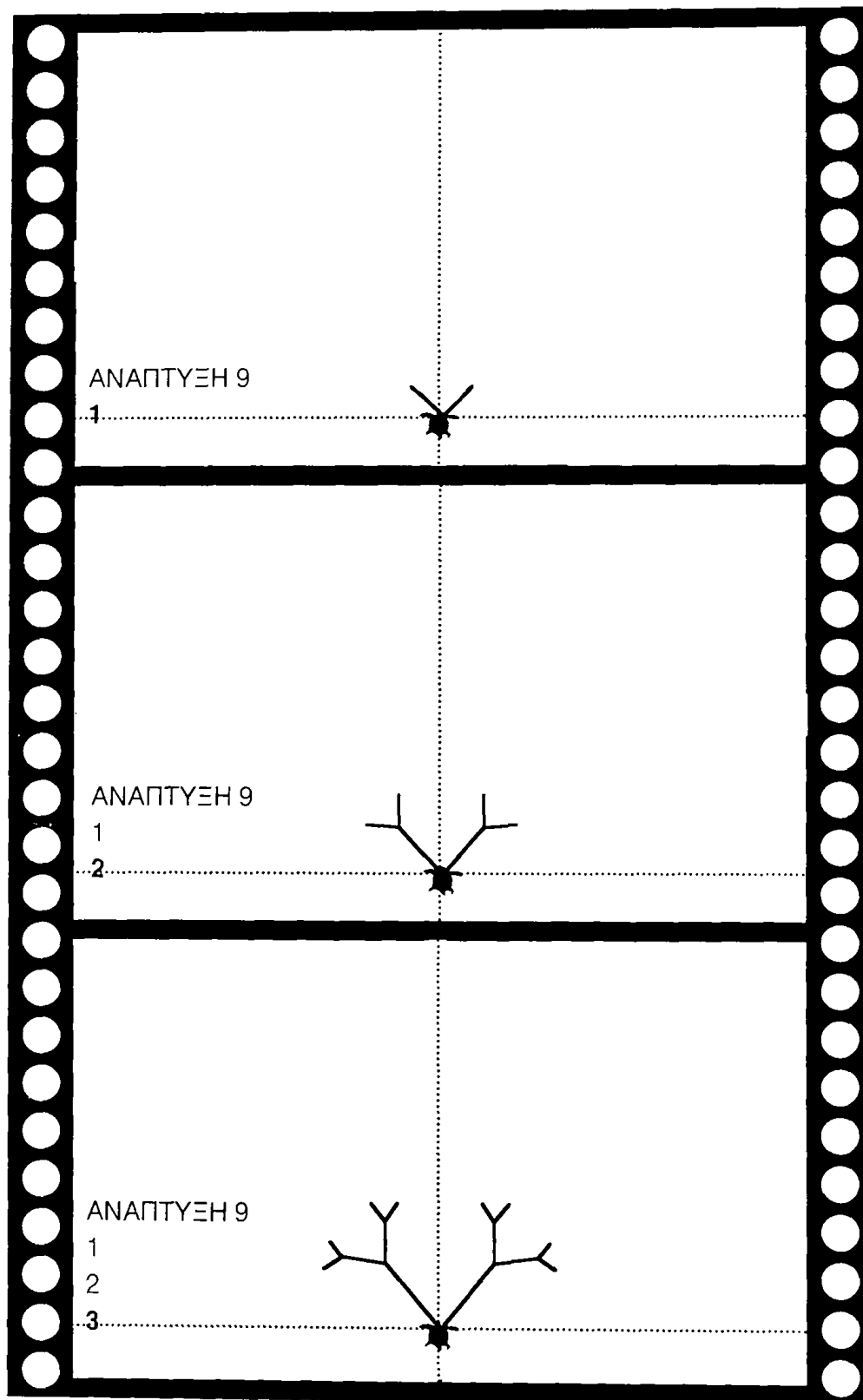
```
TO ΑΝΑΠΤΥΞΗ :ΧΡΟΝΙΑ
ΚΡΥΨΟΥ
ΜΑΚΕ "ΚΛΑΔΙ 15
ΜΑΚΕ "ΓΕΝΙΑ 1
ΜΑΚΕ "ΦΟΥΝΤΑ 45
ΜΑΚΕ "ΚΛΑΣΜΑ 1 / 2
ΡΕΡΕΑΤ :ΧΡΟΝΙΑ
[ΣΒΗΣΕ_ΕΙΚΟΝΑ
PRINT :ΓΕΝΙΑ
ΔΕΝΔΡΟ :ΚΛΑΔΙ :ΓΕΝΙΑ :ΦΟΥΝΤΑ :ΚΛΑΣΜΑ
ΜΑΚΕ "ΚΛΑΔΙ :ΚΛΑΔΙ + 7
ΜΑΚΕ "ΓΕΝΙΑ :ΓΕΝΙΑ + 1
ΜΑΚΕ "ΦΟΥΝΤΑ :ΦΟΥΝΤΑ * 0.95]
```

```
ΒΓΕΣ
ΕΝΔ
```



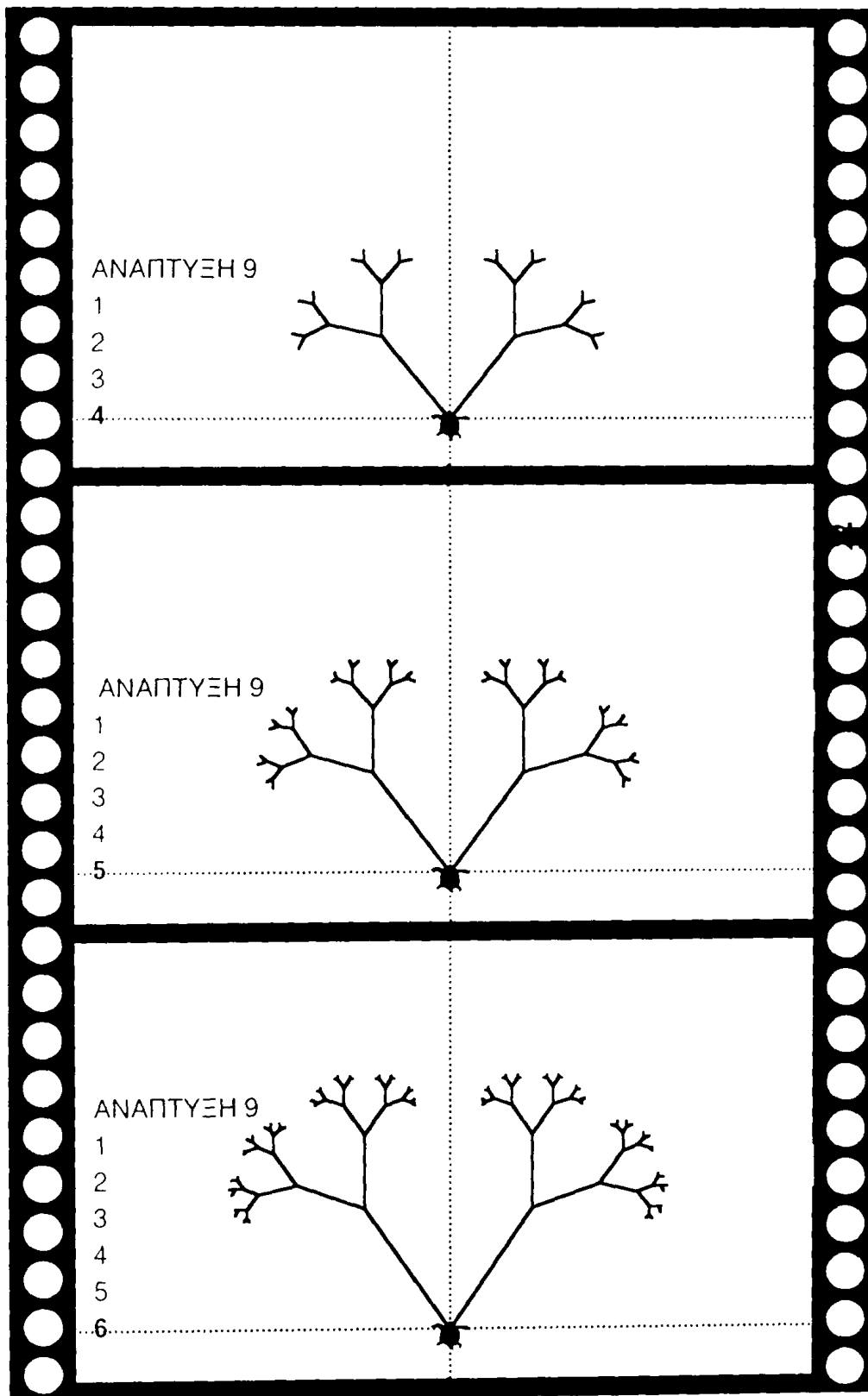
Το φυτό αναπτύσσεται (1)

Στα επόμενα σχήματα παρακολουθούμε στιγμιότυπα από την ανάπτυξη του φυτού κατά τα 3 πρώτα χρόνια-στάδια της ζωής του.



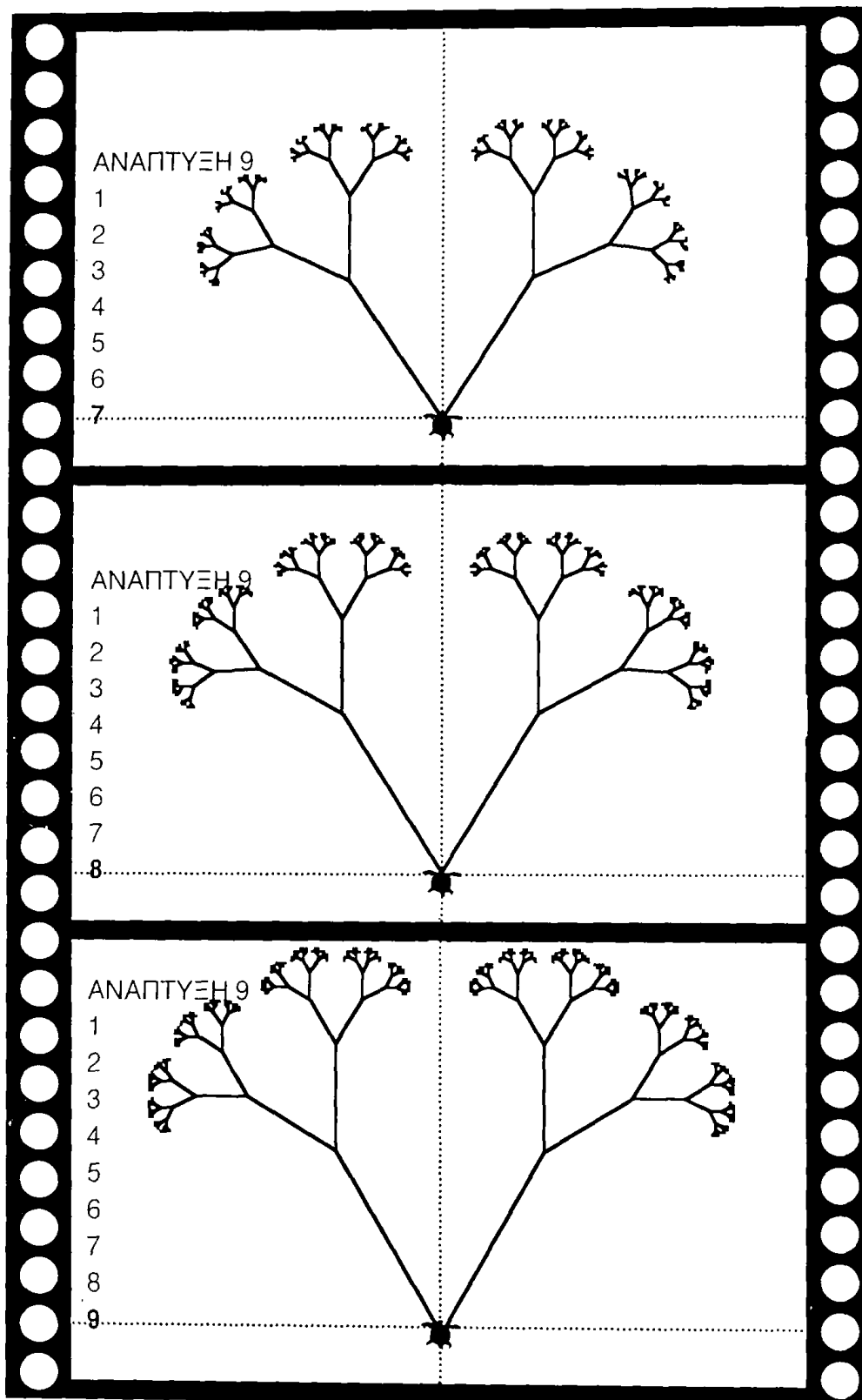
Το φυτό αναπτύσσεται (2)

Στα επόμενα στιγμιότυπα παρακολουθούμε την ανάπτυξη του φυτού από τον 4ο μέχρι τον 6ο χρόνο της ζωής του.



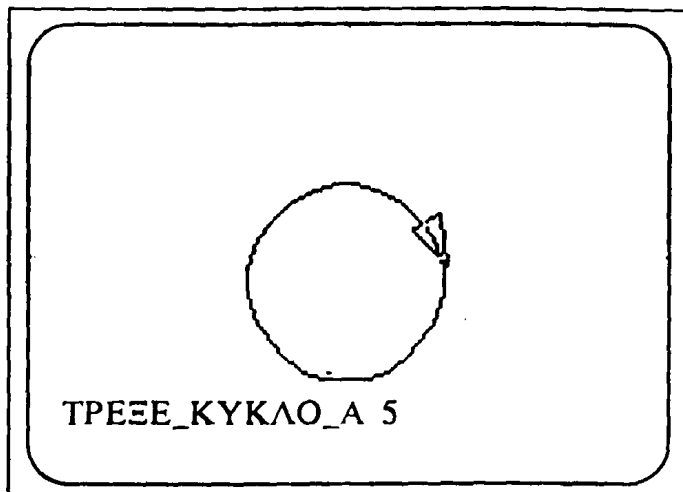
Το φυτό αναπτύσσεται (3)

Στα επόμενα στιγμιότυπα παρακολουθούμε την ανάπτυξη του φυτού από τον 7ο μέχρι τον 9ο χρόνο της ζωής του.



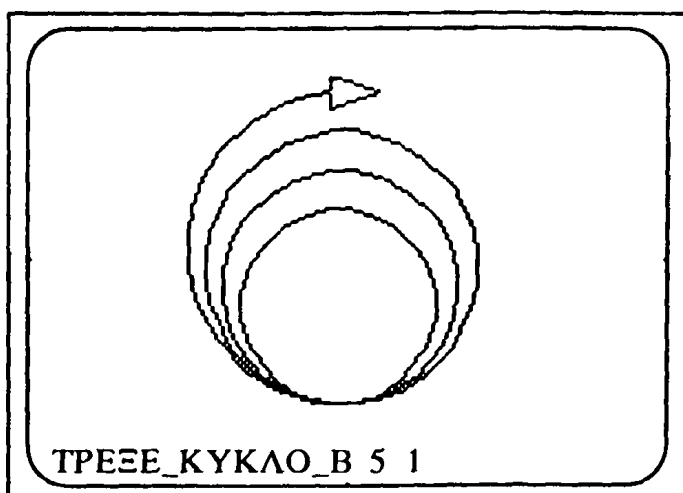
Κίνηση με άμεση αναδρομή

Ξέρουμε ήδη να σχεδιάζουμε την περιφέρεια ενός κύκλου με τη διαδικασία ΚΥΚΛΟΣ.



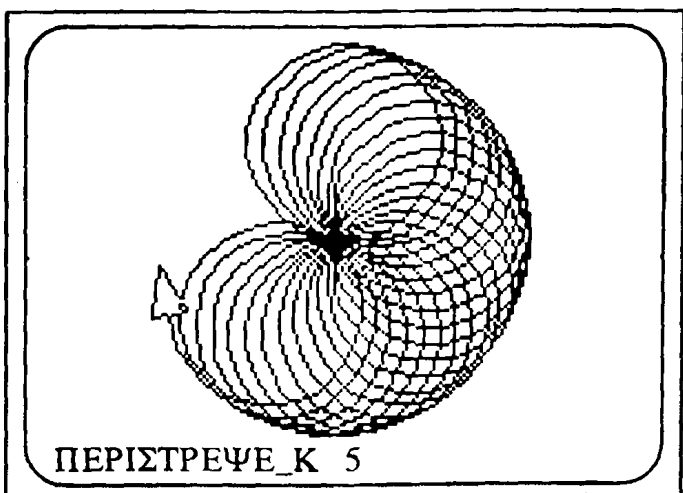
Μπορούμε με τη διαδικασία ΤΡΕΞΕ_ΚΥΚΛΟ_Α να κάνουμε τη χελώνα να τρέχει αριστερόστροφα τον κύκλο αυτό αενάως (χρησιμοποιώντας άμεση αναδρομή).

```
ΤΟ ΤΡΕΞΕ_ΚΥΚΛΟ_Α :PLEYRA  
ΚΥΚΛΟΣ :PLEYRA 1  
ΤΡΕΞΕ_ΚΥΚΛΟ_Α :PLEYRA  
END
```



Επίσης μπορούμε αυξάνοντας το μέγεθος της πλευράς του τριανταεξαγώνου-κύκλου με τη διαδικασία ΤΡΕΞΕ_ΚΥΚΛΟ_Β να έχουμε το διπλανό σχήμα.

```
ΤΟ ΤΡΕΞΕ_ΚΥΚΛΟ_Β :PL :ΒΗΜΑ  
ΚΥΚΛΟΣ :PL 1  
MAKE "PL :PL + :ΒΗΜΑ  
ΤΡΕΞΕ_ΚΥΚΛΟ_Β :PL :ΒΗΜΑ  
END
```



Ανάλογα μπορούμε με τη διαδικασία ΠΕΡΙΣΤΡΕΨΕ_Κ να περιστρέψουμε αυτούς τους κύκλους.

```
ΤΟ ΠΕΡΙΣΤΡΕΨΕ_Κ :PLEYRA  
ΚΥΚΛΟΣ :PLEYRA 1  
MAKE "MOIRES (360/36)  
ΔΕΞΙΑ :MOIRES  
ΠΕΡΙΣΤΡΕΨΕ_Κ :PLEYRA  
END
```



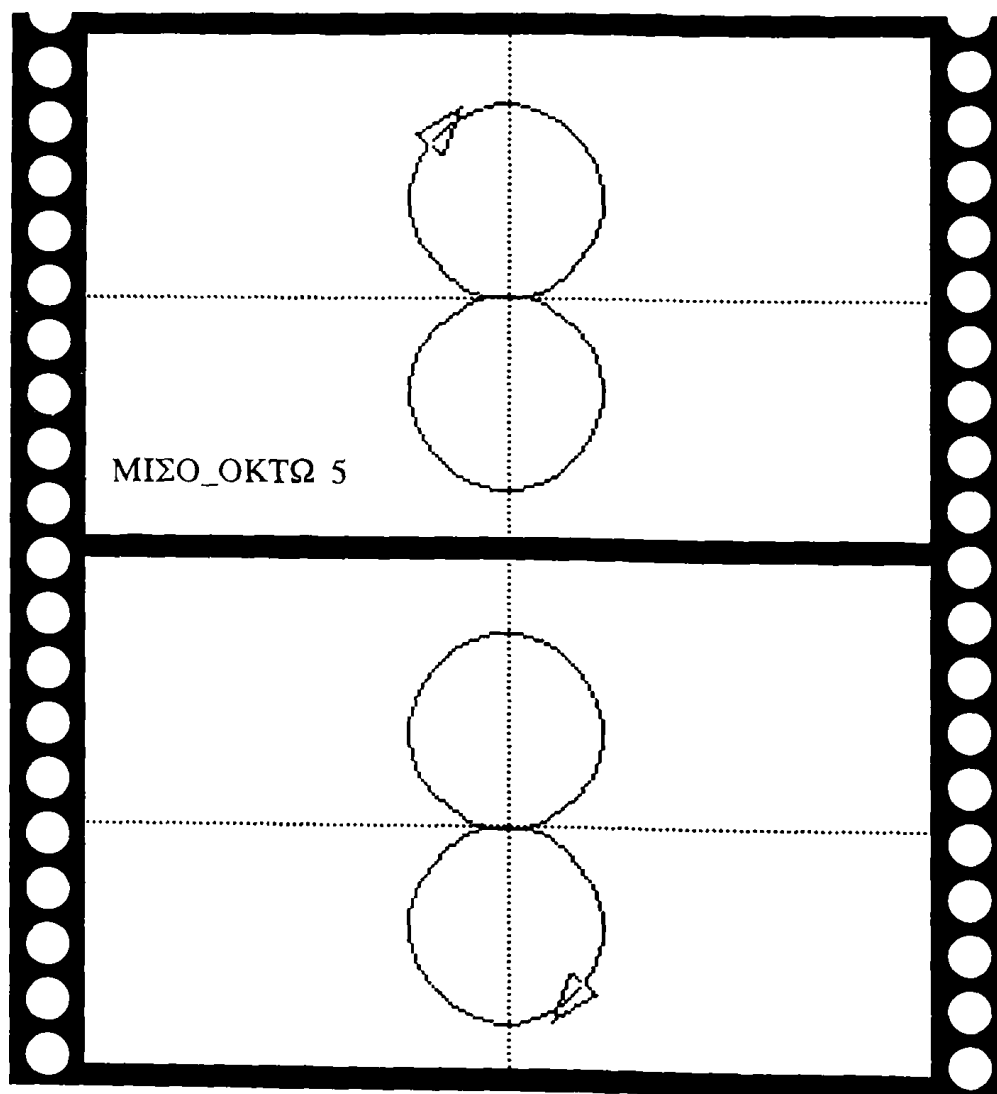
Εξετάσεις για δίπλωμα μοτοσυκλέτας

Ας γράψουμε μια διαδικασία η οποία με άμεσο αναδρομικό τρόπο κινείται αενάως σχεδιάζοντας ένα οκτάρι.

Η διαδικασία ΜΙΣΟ_ΟΚΤΩ

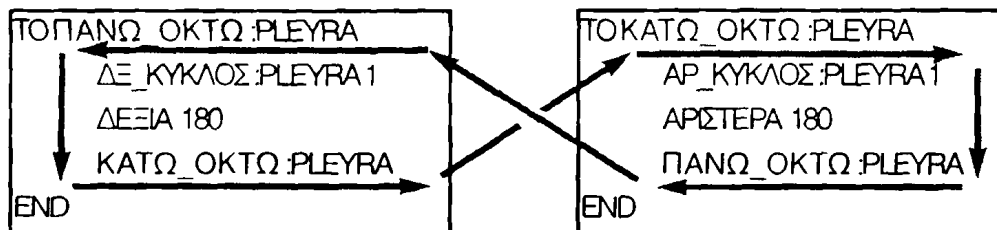
```
ΤΟ ΜΙΣΟ_ΟΚΤΩ :PLEYRA  
ΚΥΚΛΟΣ :PLEYRA 1  
ΔΕΞΙΑ 180  
ΜΙΣΟ_ΟΚΤΩ :PLEYRA  
END
```

μας δίνει τα επόμενα στιγμιότυπα όπου όμως το 8 σχηματίζεται από δύο αριστερόστροφους κύκλους.



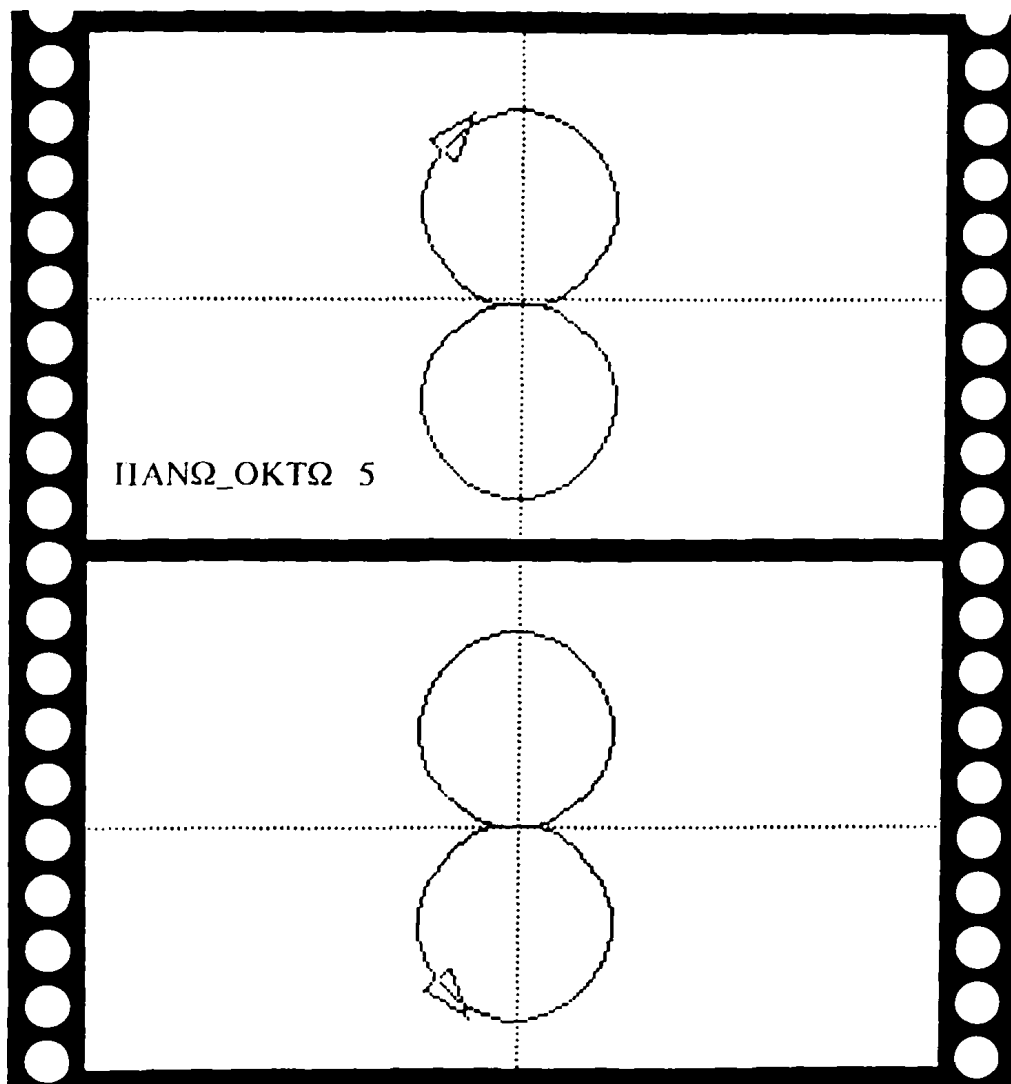
Κίνηση με έμμεσο αναδρομικό τρόπο

Για να πετύχουμε να σχηματίζεται το 8 με τον τρόπο που γράφουμε δηλαδή με έναν αριστερόστροφο πάνω κύκλο και έναν δεξιόστροφο κάτω κύκλο θα χρειαστούμε δύο διαδικασίες, την ΠΑΝΩ_ΟΚΤΩ και την ΚΑΤΩ_ΟΚΤΩ.



οι οποίες κατορθώνουν να κινούν τη χελώνα καλώντας η μία την άλλη με έμμεσο αναδρομικό τρόπο.

Οι παραπάνω διαδικασίες χρησιμοποιούν τις διαδικασίες ΔΕ_ΚΥΚΛΟΣ και ΑΡ_ΚΥΚΛΟΣ που δίνονται παρακάτω:



ΤΟ ΔΞ_ΚΥΚΛΟΣ :PL :KLASMA
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360/36)
REPEAT (:KLASMA*36) [ΔΞ_ΓΡΑΜΜΗ :MOIRES :PL]
ΔΕΞΙΑ 90
END

ΤΟ ΑΡ_ΚΥΚΛΟΣ :PL :KLASMA
ΔΕΞΙΑ 90
MAKE "MOIRES (360/36)
REPEAT (:KLASMA*36) [ΑΡ_ΓΡΑΜΜΗ :MOIRES :PL]
ΑΡΙΣΤΕΡΑ 90
END

Αυτές με τη σειρά τους χρησιμοποιούν τις διαδικασίες
DX_GRAMMH και AR_GRAMMH:

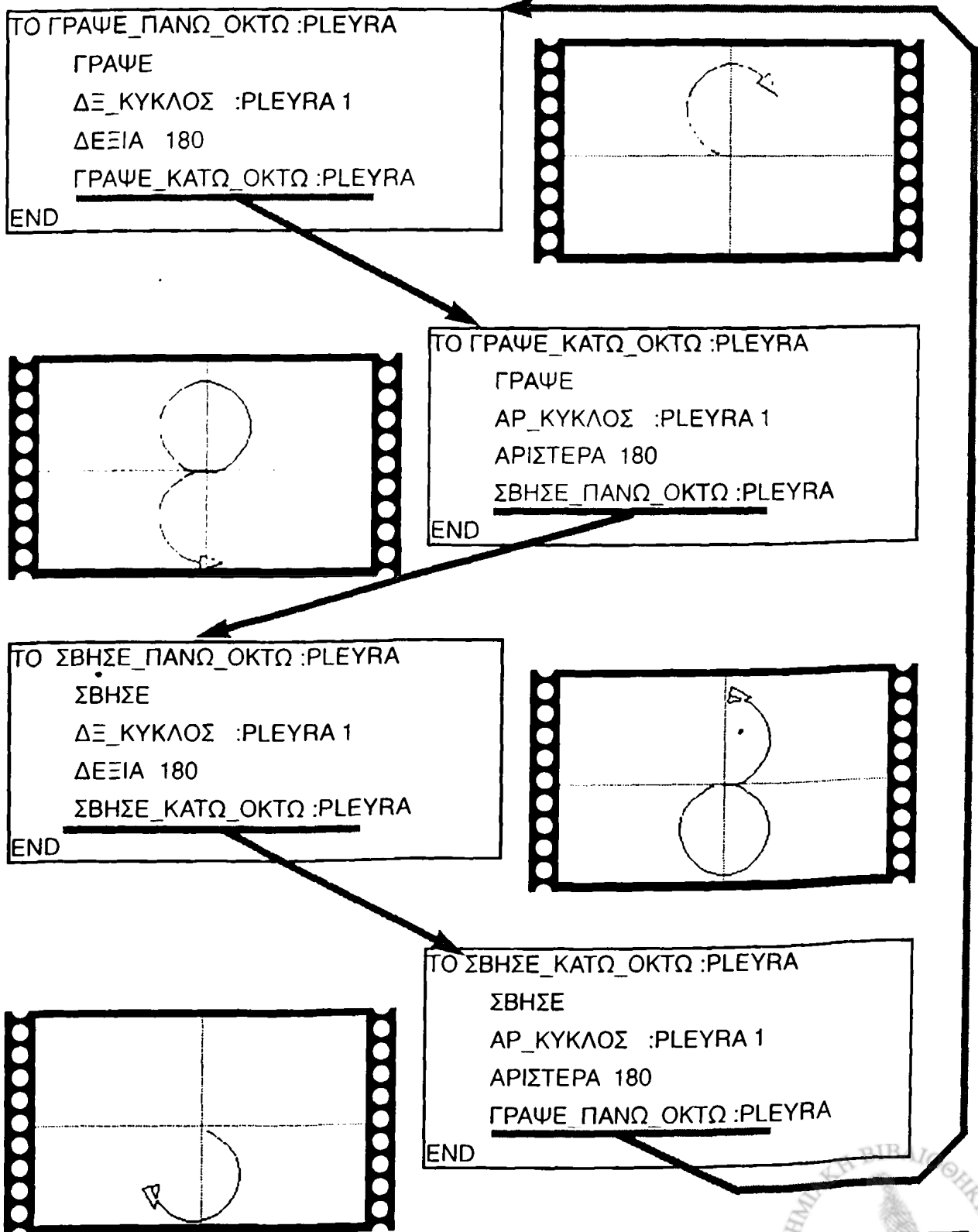
ΤΟ ΔΞ_ΓΡΑΜΜΗ :MOIRES :ΜΗΚΟΣ
ΔΕΞΙΑ :MOIRES
ΕΥΘΕΙΑ :ΜΗΚΟΣ
END

ΤΟ ΑΡ_ΓΡΑΜΜΗ :MOIRES :ΜΗΚΟΣ
ΑΡΙΣΤΕΡΑ :MOIRES
ΕΥΘΕΙΑ :ΜΗΚΟΣ
END



Τετραπλή έμμεση αναδρομή

Ας φτιάξουμε τώρα μια σειρά τεσσάρων διαδικασιών οι οποίες καλώντας η μία την άλλη με έμμεσο αναδρομικό τρόπο να σχηματίζουν ένα οκτάρι και να το σβήνουν όπως φαίνεται στα επόμενα στιγμιότυπα.



14 Fractals

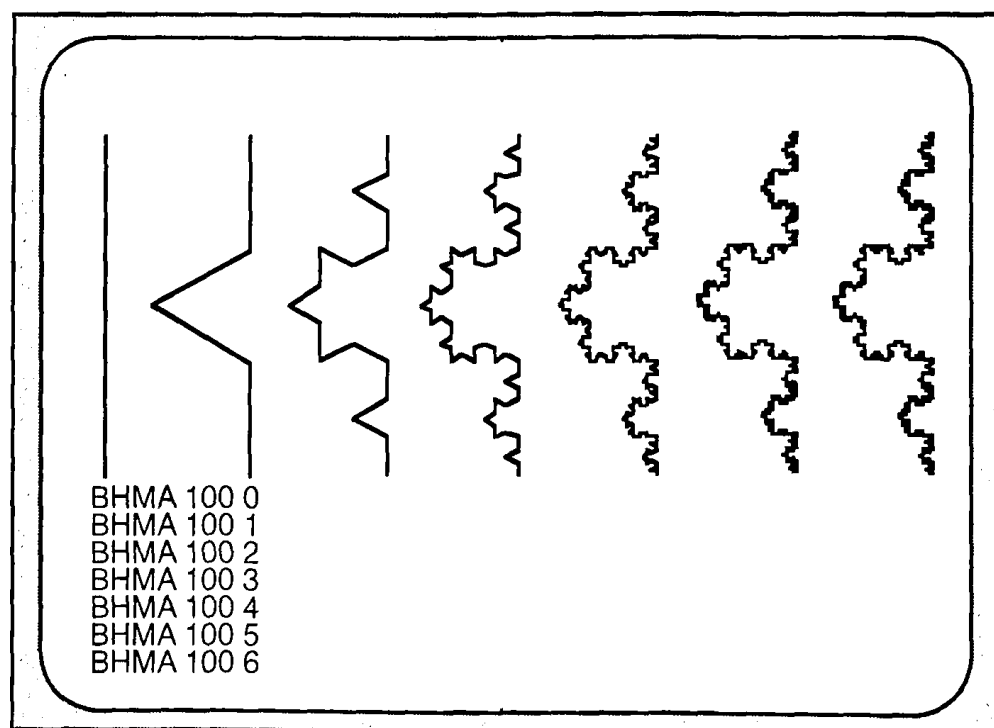
Η έννοια του μετασχηματισμού

Με απλές αναδρομικές διαδικασίες μπορούν να δημιουργηθούν εντυπωσιακές εικόνες.

Η διαδικασία ΒΗΜΑ μετασχηματίζει ένα ευθύγραμμο τμήμα (η κατακόρυφη γραμμή στα αριστερά της οθόνης) σε μια τεθλασμένη γραμμή όπως η δεύτερη από αριστερά γραμμή της οθόνης. Ο μετασχηματισμός αυτός θα γίνεται και στα επιμέρους ευθύγραμμα τμήματα της δεύτερης γραμμής με αποτέλεσμα να έχουμε το σχήμα της τρίτης γραμμής κ.ο.κ. Η αναδρομική κλίση θα σταματάει όταν η τιμή της μεταβλητής GENIA πάρει την τιμή μηδέν.

```
TO ΒΗΜΑ :MEGEQOS :GENIA
IFELSE :GENIA = 0
  [ ΕΥΘΕΙΑ :MEGEQOS ]
  [ ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 60
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 120
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 60
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1) ]
```

END



«Το σύνολο Μάντελμπροτ είναι το πιο πολύπλοκο πράγμα στα μαθηματικά. .. Η καταγραφή των διαφορετικών σχημάτων που υπάρχουν σ' αυτό ή η αριθμητική περιγραφή της περιμετρικής του γραμμής φαίνεται πως θα απαιτούσε μια απειρία πληροφοριών. Αλλά εδώ βρίσκεται το παράδοξο: για να σταλεί μια πλήρης περιγραφή του συνόλου μέσα από μια γραμμή επικοινωνίας χρειάζονται λίγες μόνο δεκάδες χαρακτηριστικών κώδικα. Ένα μικρό πρόγραμμα υπολογιστή περιέχει αρκετές πληροφορίες για να αναπαράγει ολόκληρο το σύνολο...»

James Gleick



Μετασχηματίζοντας τις πλευρές τριγώνου

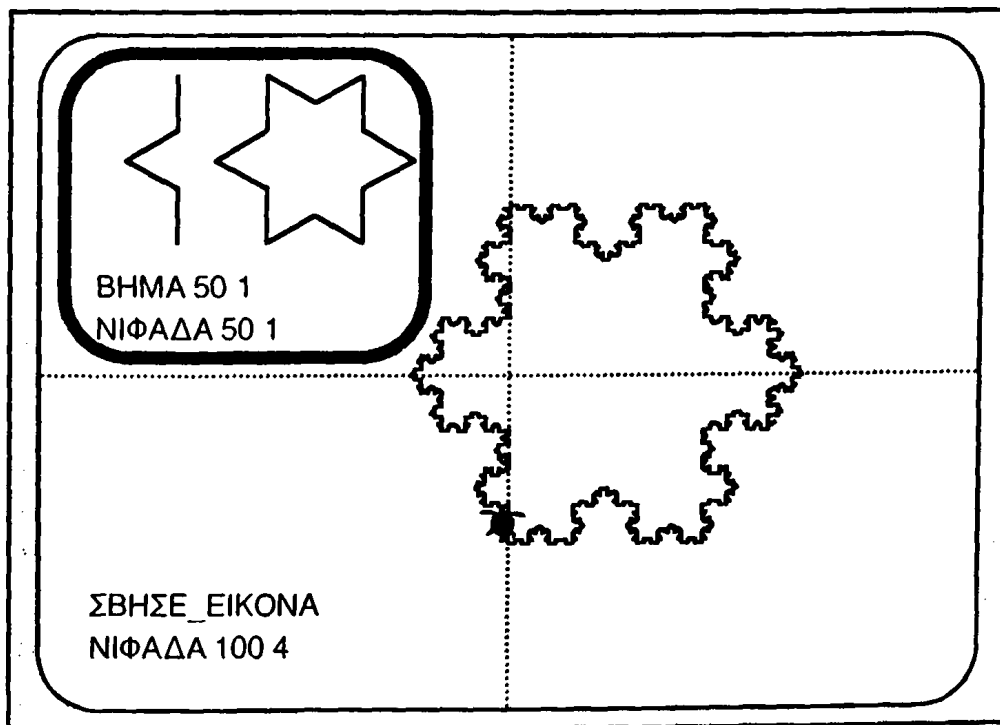
Ας θεωρήσουμε το τμήμα της τεθλασμένης γραμμής που φτιάξαμε προηγουμένως ως την πλευρά ενός ισόπλευρου τριγώνου και ας φτιάξουμε τη διαδικασία ΝΙΦΑΔΑ.

```

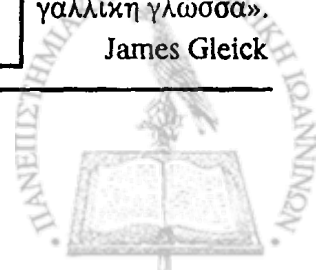
TO ΝΙΦΑΔΑ :MEGEQOS :GENIA
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:MEGEQOS / 2)
REPEAT 3
  [ΒΗΜΑ :MEGEQOS :GENIA ΔΕΞΙΑ 120 ]
END
    
```

Το σχήμα που σχηματίζεται ονομάζεται καμπύλη του Koch και είναι fractal.

Ο όρος fractal αποδίδεται στα ελληνικά ως "κλασματομορφικός" (Τάσος Μπουντής "Εισαγωγή στα Δυναμικά Συστήματα").

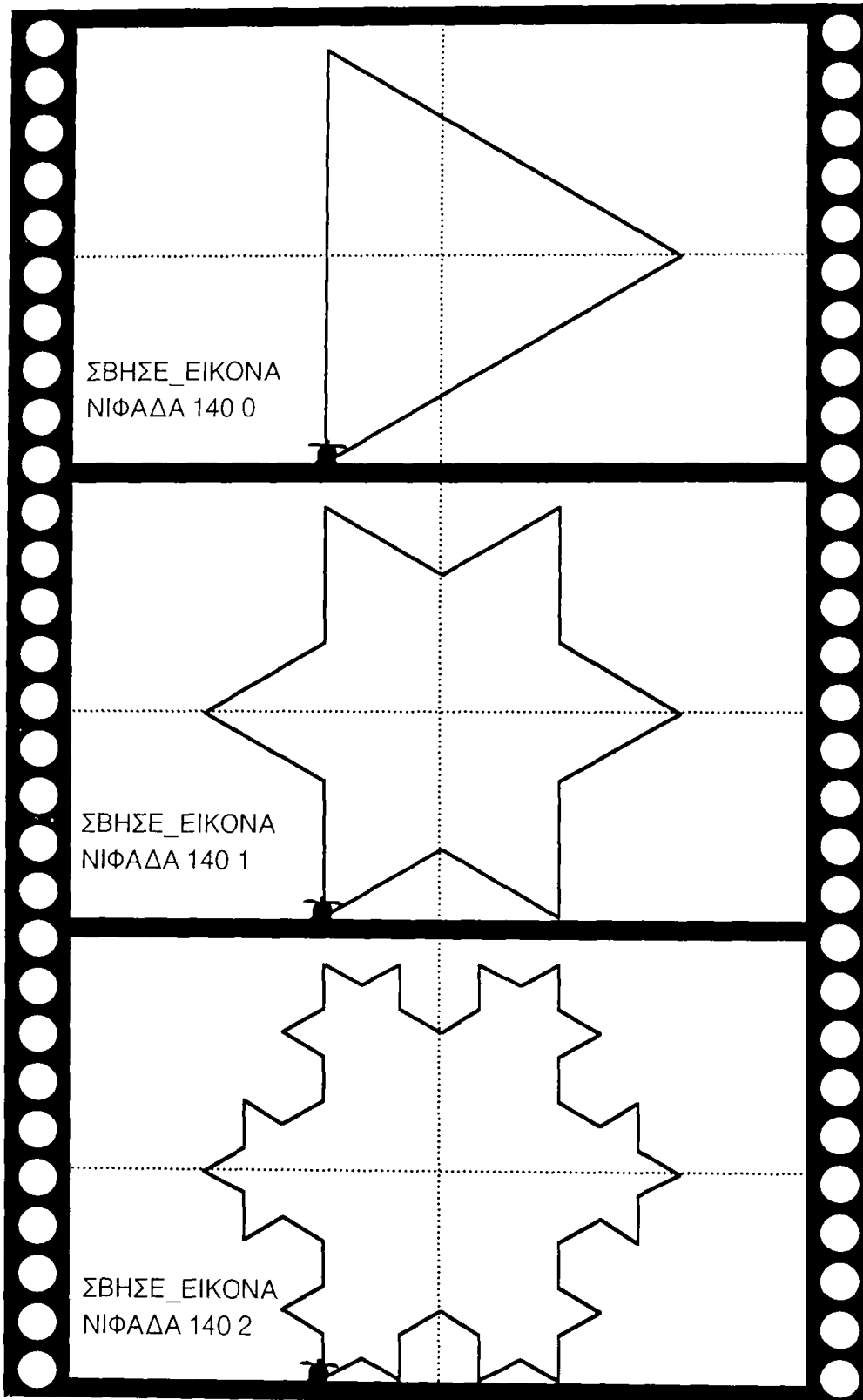


«Ένα χειμωνιάτικο απόγευμα του 1975, ο Μάντελμπροτ (Mandelbrot) κατέληξε ότι χρειαζόταν ένα όνομα για τα σχήματα, τις διαστάσεις και τη γεωμετρία του. Στο λατινικό λεξικό του γιού του συνάντησε το επίθετο fractus, από το ρήμα frangere: σπάζω. Η ακουστική ομοιότητα με τις ομόρριζες αγγλικές λέξεις fracture και fraction (σπάω και τεμάχιο/κλάσμα) φαινόταν κατάλληλη. Ο Μάντελμπροτ επινόησε τη λέξη fractal, που είναι ουσιαστικό και επίθετο στην αγγλική και τη γαλλική γλώσσα». James Gleick



Ο κ. Koch και μια ιδανική νιφάδα χιονιού (1)

Ας «τρέξουμε» μερικές φορές τη διαδικασία ΝΙΦΑΔΑ για διάφορες τιμές της παραμέτρου GENIA.



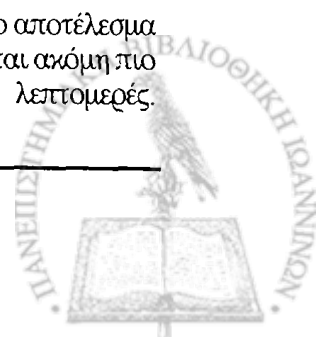
Έστω ένα ισόπλευρο τρίγωνο. Διαιρέστε μια πλευρά του στα τρία. Με το μεσαίο ένα τρίτο της πλευράς και προς το εξωτερικό του τριγώνου, σχηματίστε ένα άλλο ισόπλευρο τρίγωνο. Επαναλάβετε αυτή την εργασία και για τις άλλες δύο πλευρές του αρχικού τριγώνου.

Εφαρμόστε την παραπάνω διαδικασία σε κάθε πλευρά του σχήματος που δημιουργήθηκε.

Το αποτέλεσμα γίνεται όλο και πιο λεπτομερές.

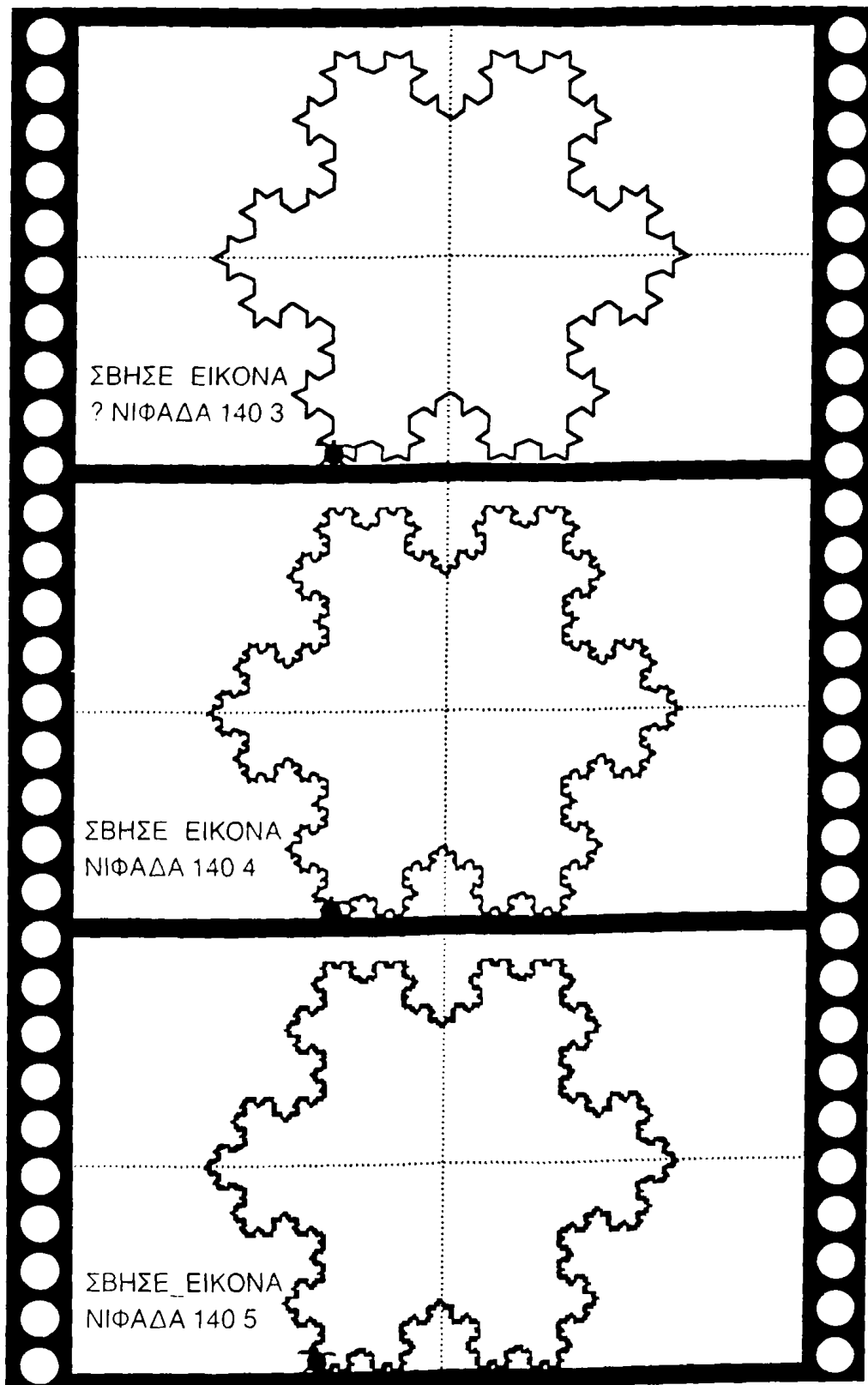
Εφαρμόστε την παραπάνω διαδικασία σε κάθε πλευρά του σχήματος που δημιουργήθηκε.

Το αποτέλεσμα γίνεται ακόμη πιο λεπτομερές.



Ο κ. Koch και μια ιδανική νιφάδα χιονιού (2)

Συνεχίζοντας το τρέξιμο της διαδικασίας ΝΙΦΑΔΑ για μεγαλύτερες τιμές της GENIAS αυξάνει η πολυπλοκότητα των σχημάτων.



Συνεχίζοντας την προηγούμενη διαδικασία μέχρι το άπειρο παίρνουμε την καμπύλη Κοχ (από τον Σουηδό μαθηματικό Helge Von Koch). Η καμπύλη Κοχ έχει μερικά ενδιαφέροντα χαρακτηριστικά. Είναι μια συνεχής κλειστή γραμμή που δεν τέμνει ποτέ τον εαυτό της. Ενώ το μήκος της με κάθε μετασχηματισμό αυξάνει τείνοντας στο άπειρο, το εμβαδόν της αυξάνει ελάχιστα, αλλά το ολικό εμβαδόν παραμένει πάντα μικρότερο από το εμβαδόν του περιγεγραμμένου κύκλου στο αογκικό τρίγωνο. Έτσι έχουμε μια άπειρη σε μήκος γραμμή να περιβάλλει ένα πεπερασμένο εμβαδόν. Είναι κάτι περισσότερο από γραμμή αλλά και κάτι λιγότερο από επίπεδο. Έχει κάτι περισσότερο από μια διάσταση και κάτι λιγότερο από δύο διαστάσεις. Έχει κλασματική διάσταση ίση με 1.2618.

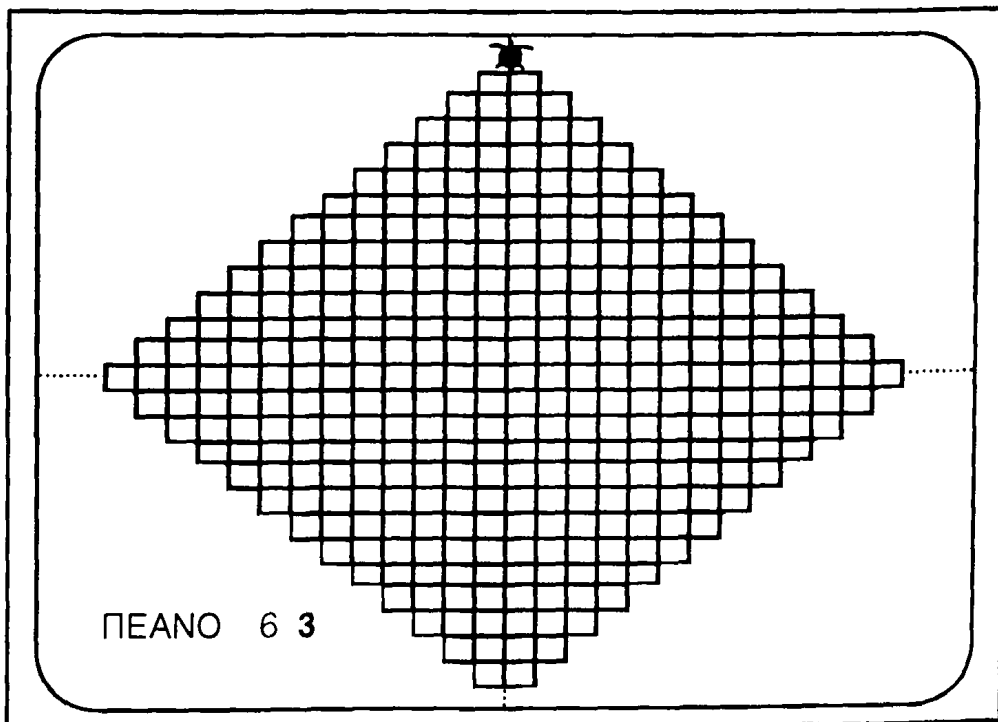
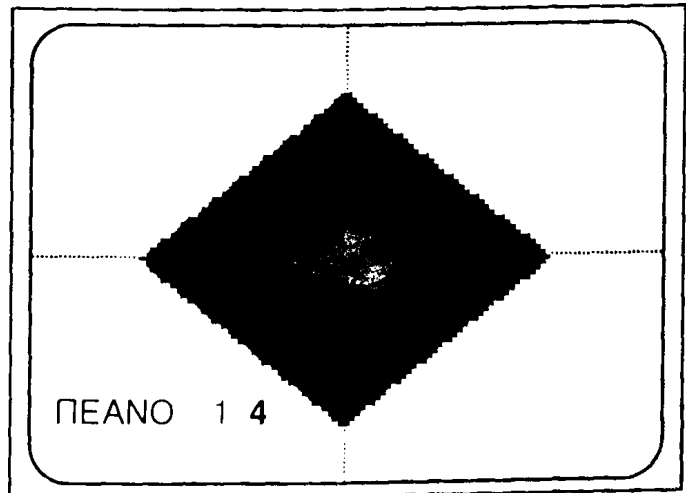


Ο κ.Ρεανο και η καμπύλη που γεμίζει το χώρο

Ας καθοδηγήσουμε τη χελώνα ώστε να σχηματίσει την καμπύλη Ρεανο χρησιμοποιώντας την τεχνική της άμεσης αναδρομής.

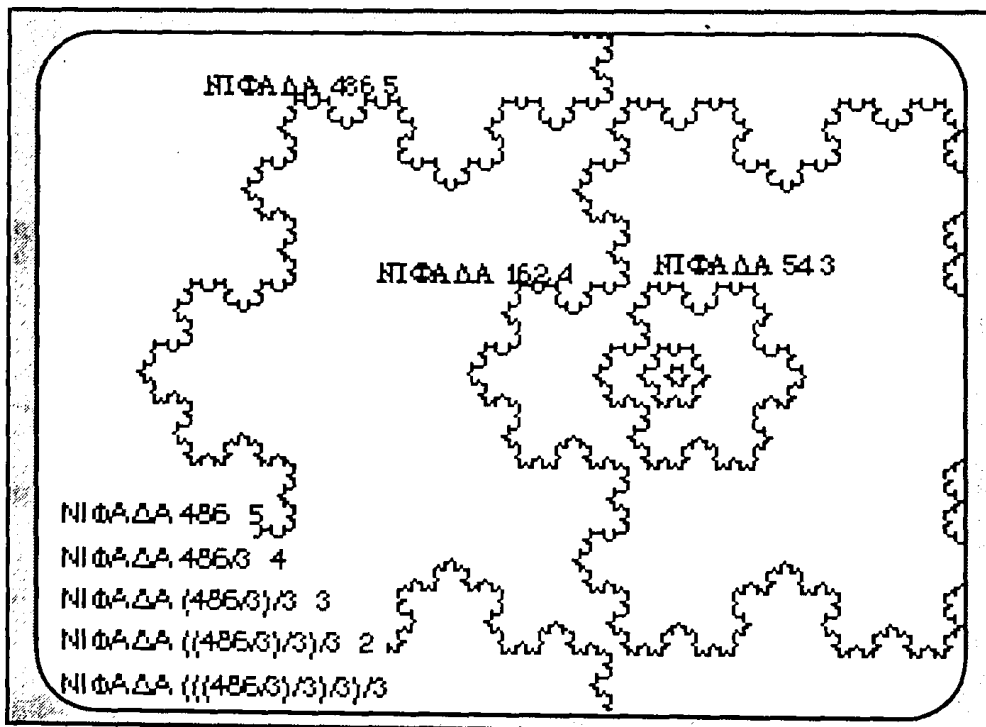
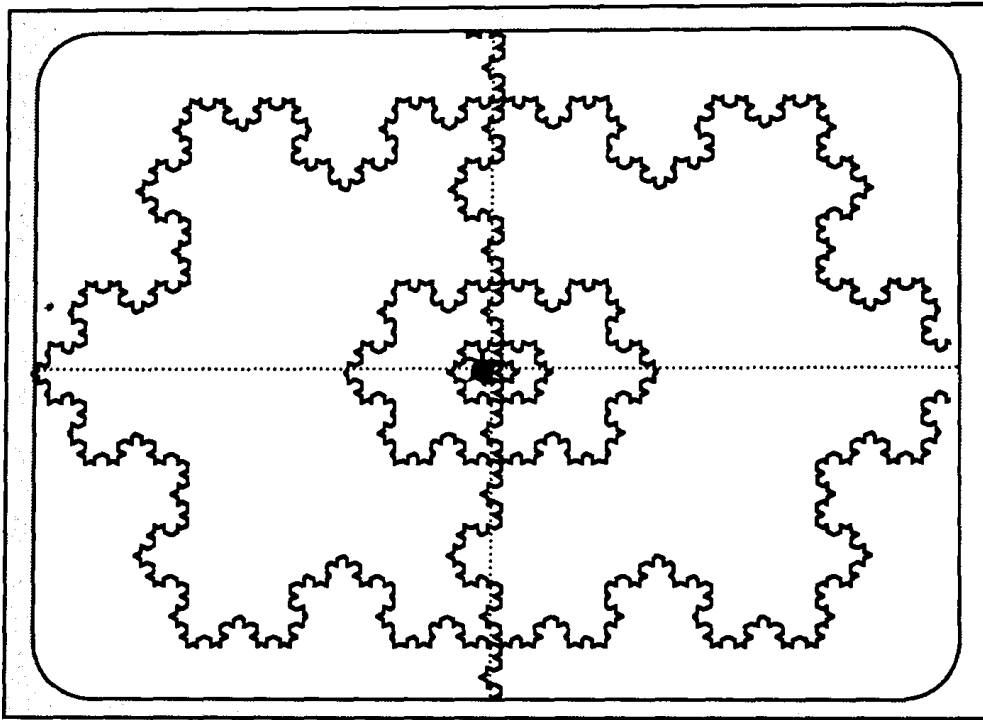
```

ΤΟ ΠΕΑΝΟ :ΒΗΜΑ :ΓΕΝΙΑ
IFELSE (:ΓΕΝΙΑ>0)
  [ ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 270:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 90:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 90:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 90:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 270:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 270:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 270:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
    ΓΡΑΜΜΗ 90:ΒΗΜΑ
    ΠΕΑΝΟ (:ΒΗΜΑ) (:ΓΕΝΙΑ-1)
  ]
[ STOP ]
END
    
```



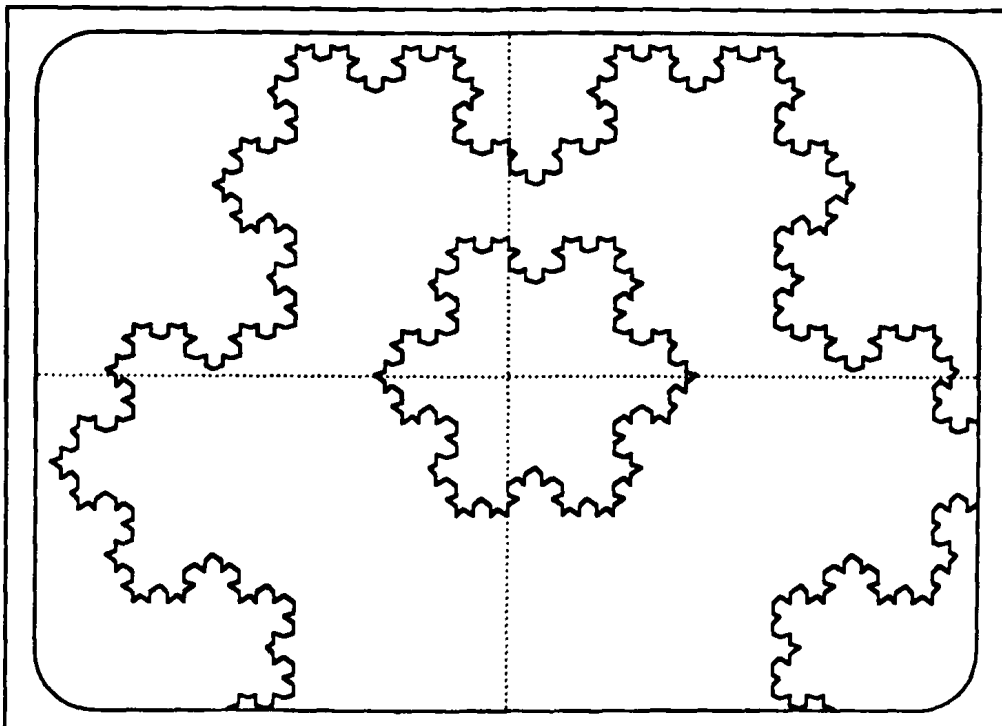
Η αυτο-ομοιότητα της καμπύλης Koch

Η ομοιότητα μεταξύ των επομένων πέντε καμπυλών Koch είναι προφανής και ανεξάρτητη της κλίμακας σχεδίασης.

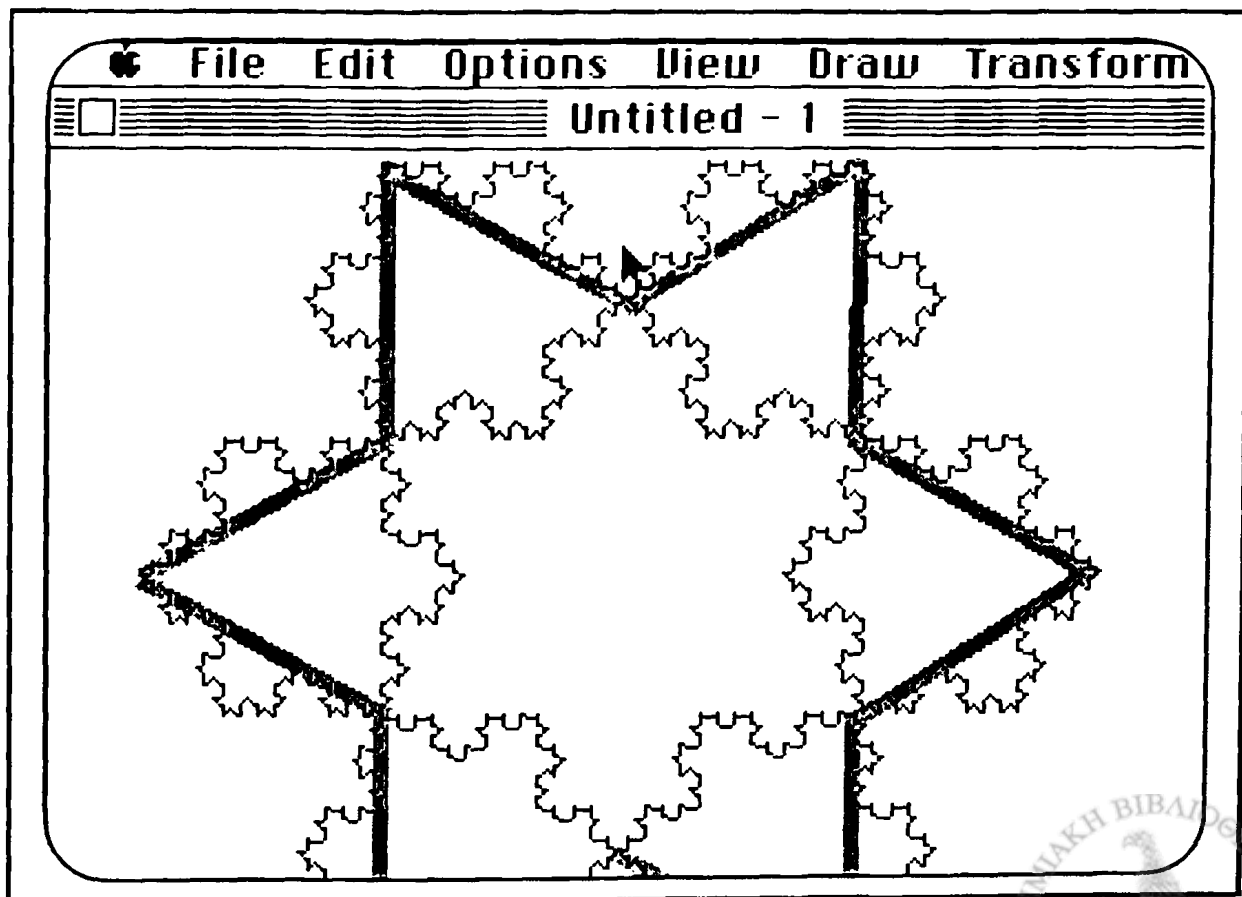


Ανακαλύπτοντας τη δομή (1)

Βασιζόμενοι στην ιδιότητα της αυτο-ομοιότητας μπορούμε να ανακαλύψουμε τους κανόνες που κρύβονται πίσω από ένα φράκταλ.

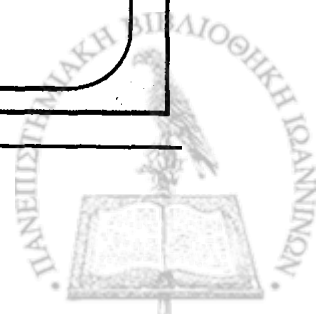
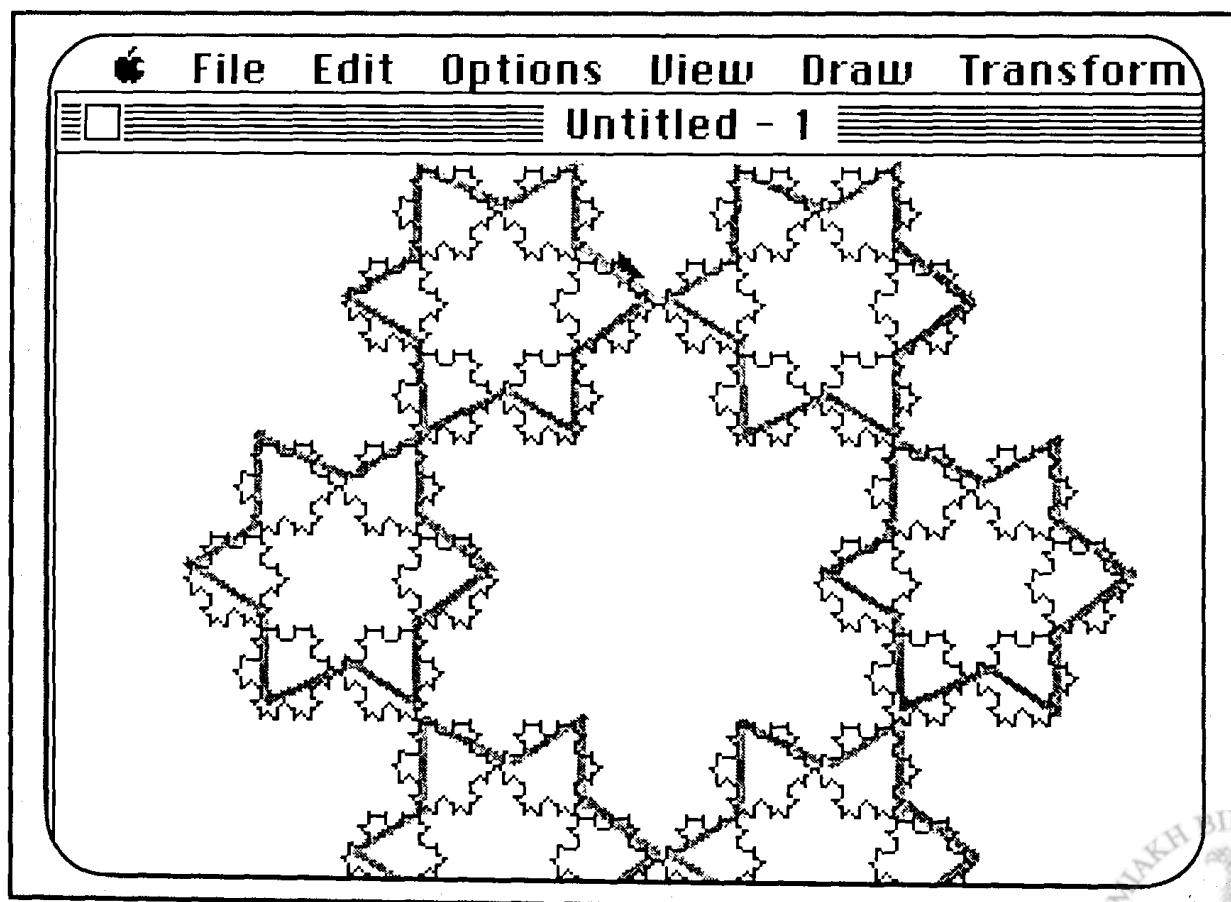
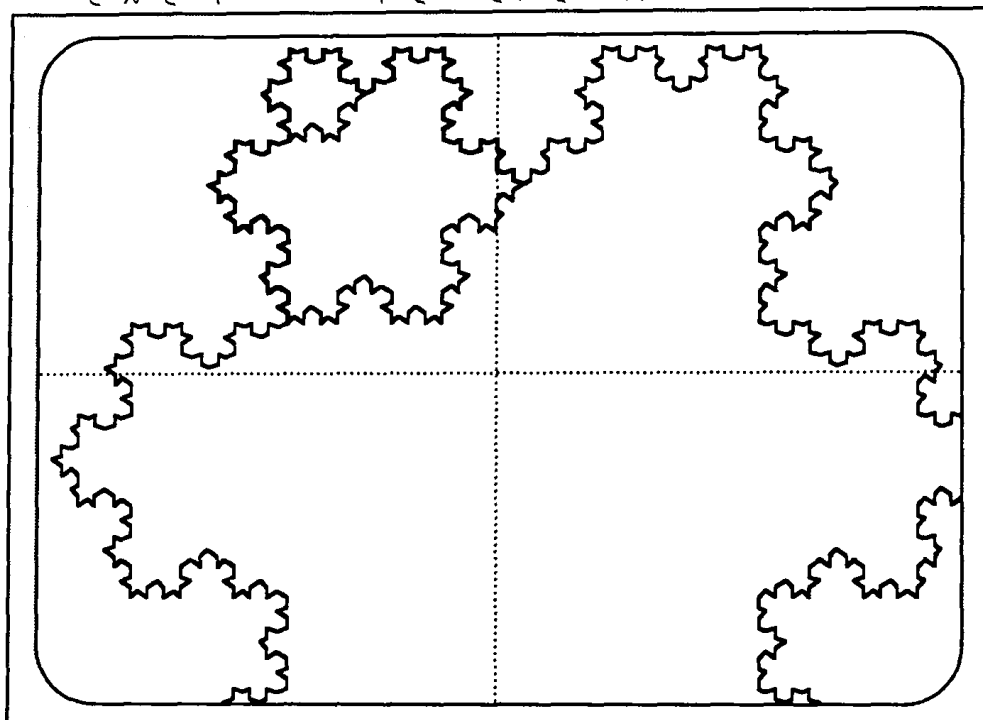


Με τη βοήθεια ενός σχεδιαστικού προγράμματος (του SuperPaint), μπορούμε να μετατοπίσουμε, αναπαράγουμε και εφαρμόσουμε τα φράκταλς που παίρνουμε από τη διαδικασία ΝΙΦΑΔΑ. Η τοποθέτηση των έξι μικρών φράκταλς στις άκρες του μεγάλου, μας βοηθούν να κατανοήσουμε τον τρόπο που δημιουργήθηκε και να ανακαλύψουμε τη δομή του.



Ανακαλύπτοντας τη δομή (2)

Λογω της αυτο-ομοιότητας η δομή του φράκταλ επιβεβαιώνεται όσο προχωρούμε σε λεπτομερέστερη προσέγγιση.



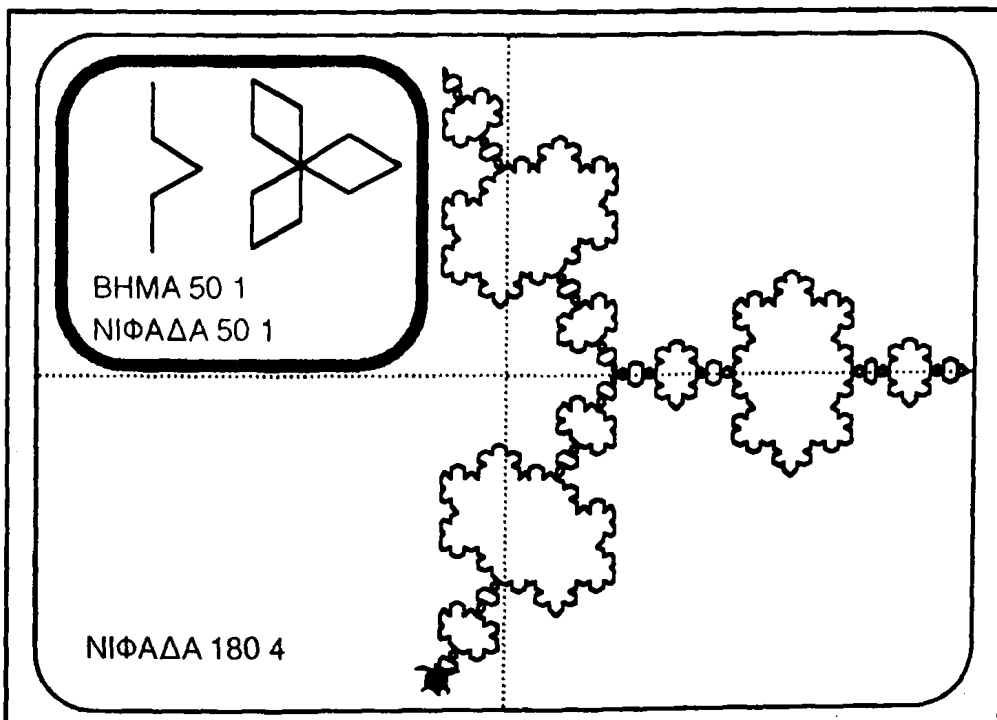
Μια εσωστρεφής καμπύλη Koch

Στη διαδικασία ΝΙΦΑΔΑ το μικρότερο τρίγωνο σχηματίζεται προς το εξωτερικό του μεγαλύτερου τριγώνου. Ας πειραματιστούμε κάνοντας το μικρότερο τρίγωνο να σχηματίζεται προς το εσωτερικό του μεγαλύτερου.

Αυτό μπορούμε να το πετύχουμε αντιμεταθέτοντας τις εντολές ΑΡΙΣΤΕΡΑ και ΔΕΞΙΑ στη διαδικασία ΒΗΜΑ, η οποία πλέον ξαναορίζεται.

```

TO ΒΗΜΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
IFELSE :ΓΕΝΙΑ = 0
  [ ΕΥΘΕΙΑ :ΜΕΓΕΘΟΣ ]
  [ ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
  ΔΕΞΙΑ 60
  ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
  ΑΡΙΣΤΕΡΑ 120
  ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
  ΔΕΞΙΑ 60
  ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
]
END
  
```



«Ανοίγοντας τις πλούσιες ευκαιρίες του παιχνιδιού με το άπειρο, η ομάδα των ιδεών που αντιπροσωπεύει αυτή η διαδικασία φέρνει το παιδί σε επαφή με κάτι από το πως είναι να είναι κανείς μαθηματικός... Το εκπληκτικό φαινόμενο που παράγεται συχνά δίνει κίνητρα για μακρινές εξερευνήσεις, στις οποίες η μαθηματική και γεωμετρική σκέψη περιπλέκεται με την αισθητική».

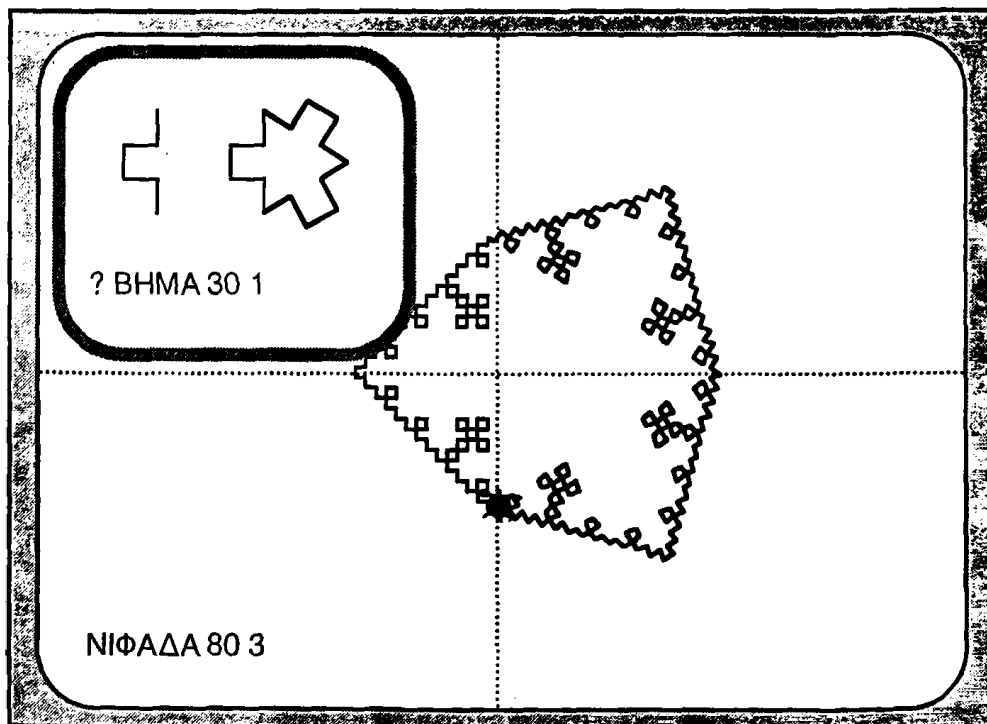
Seymour Papert



Ένας άλλος μετασχηματισμός

Αν αντί για τρίγωνο στο ένα τρίτο κάθε πλευράς σχηματίζεται ένα τετράγωνο, τότε το σχήμα που θα παραχθεί θα είναι διαφορετικό. Μερικές από τις ιδιότητες που είχαν τα προηγούμενα φράκταλς όπως η περιμετρική γραμμή τους να μην τέμνει τον εαυτό της, εδώ δεν ισχύουν.

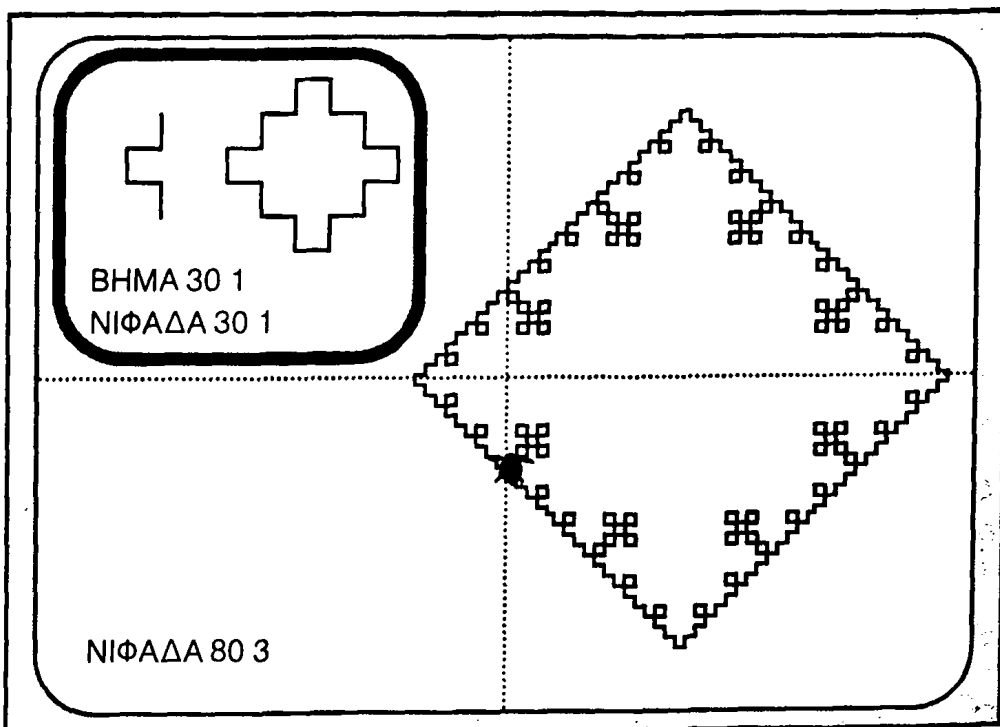
```
TO ΒΗΜΑ :MEGEQOS :GENIA
IFELSE :GENIA = 0
  [ ΕΥΘΕΙΑ :MEGEQOS ]
  [ ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ]
END
```



Εξελίσσοντας τη νιφάδα

Αν στη διαδικασία ΝΙΦΑΔΑ βάλουμε την επανάληψη να γίνεται 4 αντί για 3 φορές και αντικαταστήσουμε τη γωνία της στροφής με 90 μοίρες, τότε το φράκταλ αντί να έχει τριγωνική θα έχει τετραγωνική μορφή.

```
TO ΝΙΦΑΔΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:ΜΕΓΕΘΟΣ / 2)
REPEAT 4
  [ ΒΗΜΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
    ΔΕΞΙΑ 90 ]
END
```

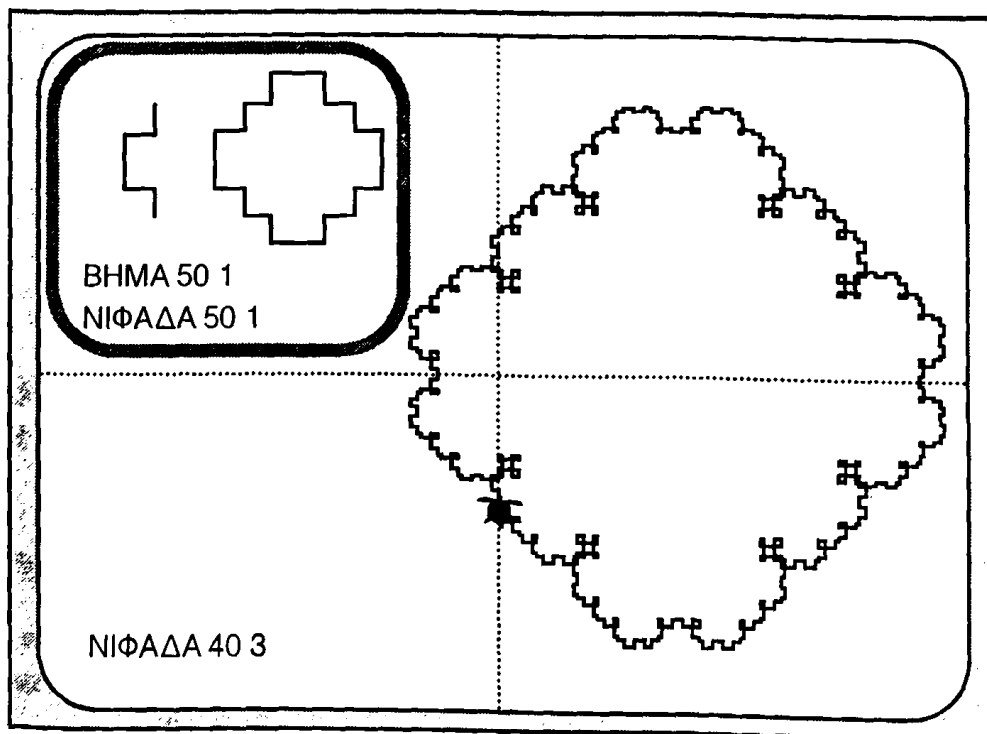


Το φράκταλ - δαντέλα

Για να μετασχηματίσουμε το αρχικό μήκος με την ακόλουθη τεθλασμένη γραμμή, θα χρειαστεί να χρησιμοποιήσουμε την παρακάτω διαδικασία. Η τεθλασμένη γραμμή ονομάζεται γεννήτορας του φράκταλ.

```

TO ΒΗΜΑ :MEGEQOS :GENIA
IFELSE :GENIA = 0
  [ ΕΥΘΕΙΑ :MEGEQOS ]
  [ ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
]
END
    
```



«Αν υπάρχει ένα μέτρο για την "αισθητική πραγματικότητα" ενός καλλιτεχνικού αντικειμένου, τότε το μέτρο αυτό εξαρτάται τόσο από την "πολυπλοκότητα" του υλικού που αναλώθηκε, όσο και από τη διάθρωσή του, δηλ. την "τάξη"... Ο Birkhoff ερμήνευσε έτσι το "αισθητικό μέτρο" της μορφοποίησης σε συνάρτηση δύο μεταβλητών, της "πολυπλοκότητας" και της "τάξης"... Αυτό που ο Birkhoff ονομάζει «πολυπλοκότητα» δεν είναι τίποτα άλλο παρά η στατιστική πληροφοριακή ποσότητα και αυτό που ονομάζει "τάξη" ανήκει στα πλεονασματικά χαρακτηριστικά που είναι απαραίτητα για να αναγνωρισθεί και κατανοηθεί μια "πληροφορία" σαν τέτοια».

Max Bense



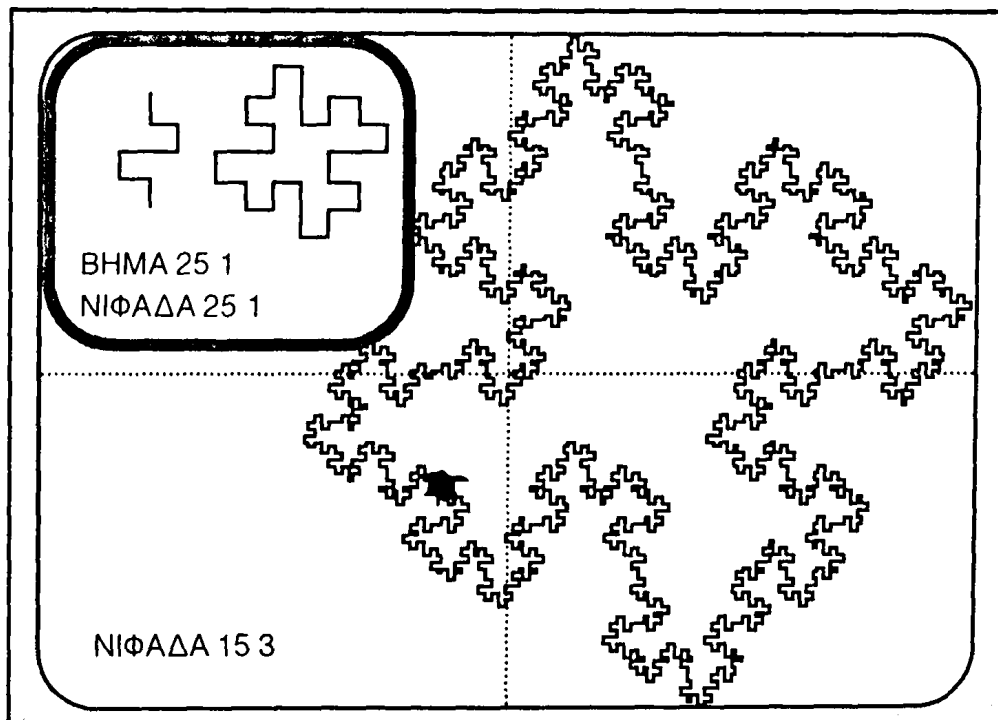
Ένα ακόμα φράκταλ

ΑΣ χρησιμοποιήσουμε έναν ακόμα γεννήτορα για να φτιάξουμε το επόμενο φράκταλ. Ο γεννήτορας που θα χρησιμοποιήσουμε παρουσιάζεται στο διπλανό σχήμα.



```

TO ΒΗΜΑ :MEGEQOS :GENIA
IFELSE :GENIA = 0
  [ ΕΥΘΕΙΑ :MEGEQOS ]
  [ ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΑΡΙΣΤΕΡΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ΔΕΞΙΑ 90
  ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
]
END
  
```



Η κότα κάνει το αυγό...

Μέχρι τώρα στις αναδρομικές διαδικασίες που χρησιμοποιήσαμε, μια διαδικασία καλεί τον εαυτό της (άμεση αναδρομή). Ας δοκιμάσουμε την τεχνική της έμμεσης αναδρομής όπου μια διαδικασία καλεί μια άλλη και αυτή με τη σειρά της καλεί αυτήν που την κάλεσε. Έτσι η διαδικασία ΒΗΜΑ1 καλεί τη διαδικασία ΒΗΜΑ2 και αυτή με τη σειρά της καλεί τη ΒΗΜΑ1.

Έστω ότι η ΒΗΜΑ1 χρησιμοποιεί ως γεννήτορα το εξωστρεφές τρίγωνο, ενώ η ΒΗΜΑ2 το εσωστρεφές.

```
TO ΒΗΜΑ1 :M :GN
```

```
IFELSE :GN = 0
```

```
  [ΕΥΘΕΙΑ :M]
```

```
  [ ΒΗΜΑ2 (:M / 3.0) (:GN - 1)
```

```
  ΑΡΙΣΤΕΡΑ 60
```

```
  ΒΗΜΑ2 (:M / 3.0) (:GN - 1)
```

```
  ΔΕΞΙΑ 120
```

```
  ΒΗΜΑ2 (:M / 3.0) (:GN - 1)
```

```
  ΑΡΙΣΤΕΡΑ 60
```

```
  ΒΗΜΑ2 (:M / 3.0) (:GN - 1)
```

```
  ]
```

```
END
```

```
TO ΒΗΜΑ2 :M :GN
```

```
IFELSE :GN = 0
```

```
  [ΕΥΘΕΙΑ :M]
```

```
  [ ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
```

```
  ΔΕΞΙΑ 60
```

```
  ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
```

```
  ΑΡΙΣΤΕΡΑ 120
```

```
  ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
```

```
  ΔΕΞΙΑ 60
```

```
  ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
```

```
  ]
```

```
END
```

```
TO ΝΙΦΑΔΑ :MEGEQOS :GENIA
```

```
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:MEGEQOS / 2)
```

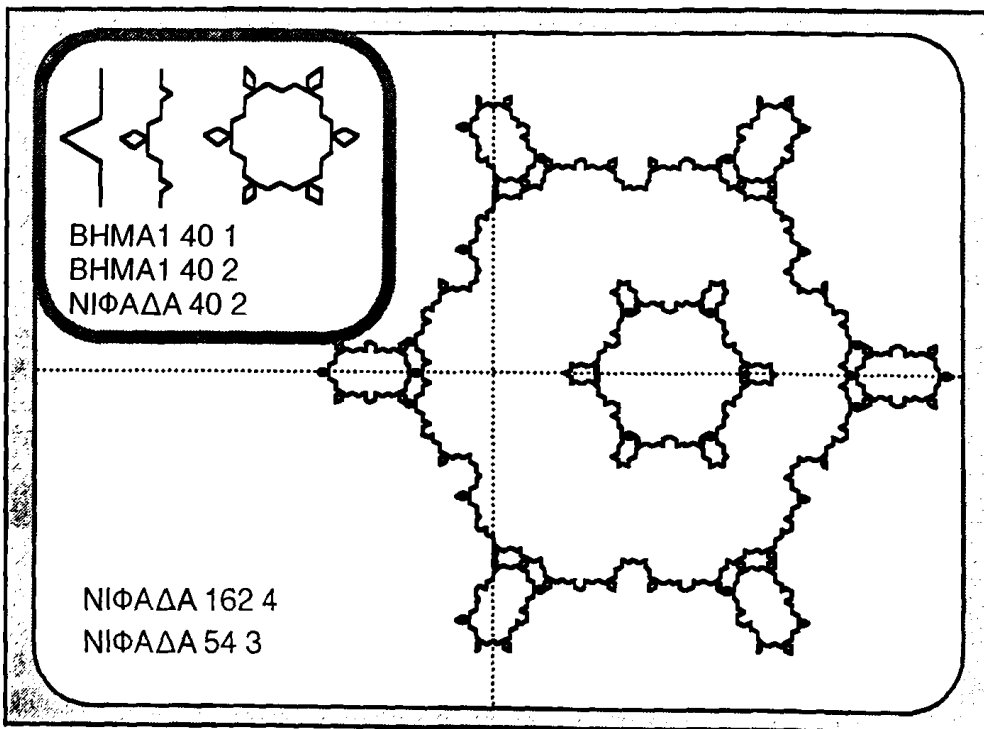
```
REPEAT 3
```

```
  [ΒΗΜΑ1 :MEGEQOS :GENIA
```

```
  ΔΕΞΙΑ 120
```

```
  ]
```

```
END
```



... ή το αυγό την κότα;

Αν η διαδικασία ΝΙΦΑΔΑ χρησιμοποιεί τη διαδικασία ΒΗΜΑ2 αντί τη διαδικασία ΒΗΜΑ1, τότε το φράκταλ που θα προκύψει θα διαφέρει σημαντικά από το προηγούμενο.

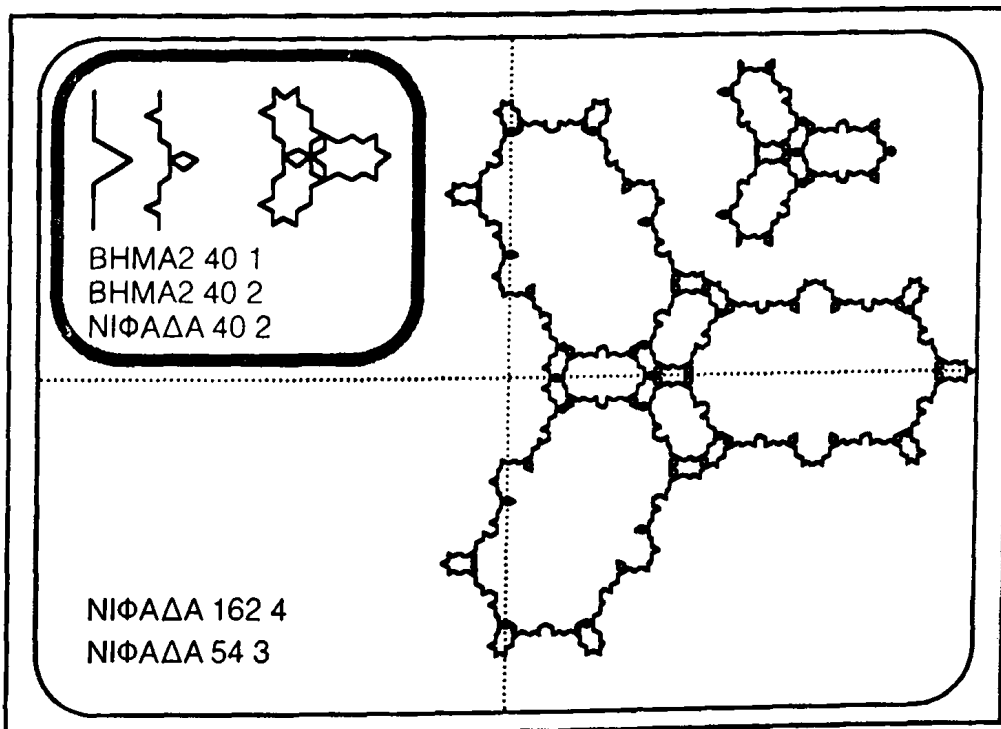
```

ΤΟ ΝΙΦΑΔΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:ΜΕΓΕΘΟΣ / 2)
ΡΕΡΕΑΤ 3
  [ ΒΗΜΑ2 :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
    ΔΕΞΙΑ 120
  ]
ΕΝΔ
    
```

Το σχήμα που προκύπτει από τη συγκεκριμένη διαδικασία είναι ένα καλό παράδειγμα για την αυτο-ομοιότητα.

«...Πάνω από όλα, φράκταλ σημαίνει κάτι όμοιο με τον εαυτό του. Η αυτο-ομοιότητα είναι συμμετρία που συναντάμε σε όλες τις κλίμακες. Σημαίνει αναδρομή, μορφή μέσα στη μορφή. ...σχήματα σαν την καμπύλη του Κοχ εμφανίζουν ομοιότητα με τον εαυτό τους γιατί φαίνονται ακριβώς ίδια, ακόμα και σε μεγάλη μεγέθυνση. Η αυτο-ομοιότητα βρίσκεται μέσα στην τεχνική της κατασκευής των καμπυλών - ο ίδιος μετασχηματισμός επαναλαμβάνεται σε όλο και μικρότερες κλίμακες...»

James Gleick



15 Παράλληλος προγραμματισμός

Ψευδο-παράλληλος προγραμματισμός

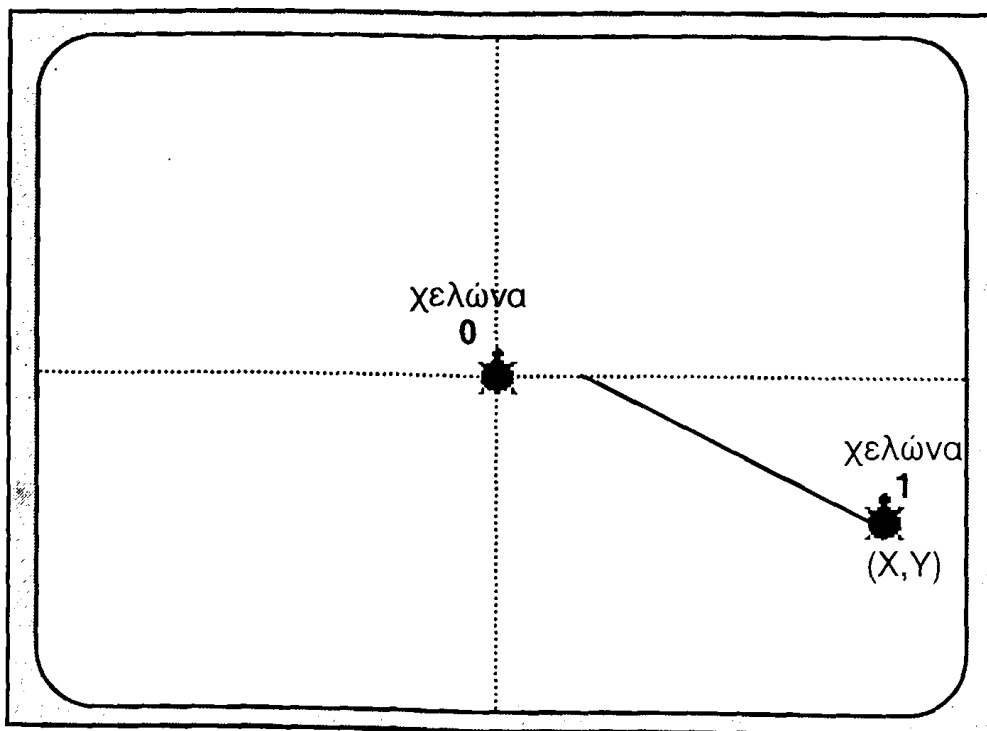
Όταν μιλάμε για παράλληλο προγραμματισμό ενός υπολογιστή, προϋποτίθεται ότι ο υπολογιστής διαθέτει περισσότερους του ενός επεξεργαστές οι οποίοι θα εκτελούν συγχρόνως (παράλληλα) ο καθένας κάποιο δικό του τμήμα προγράμματος. Ένας υπολογιστής που διαθέτει μόνο έναν επεξεργαστή είναι δυνατό να δημιουργήσει στο χρήστη την αίσθηση ότι «τρέχει» ταυτόχρονα περισσότερα του ενός προγράμματα αν εκτελεί για λίγο το πρώτο πρόγραμμα, στη συνέχεια εκτελεί το δεύτερο και μετά πάλι το πρώτο, κ.ο.κ. Η εναλλαγή μεταξύ των προγραμμάτων πρέπει να είναι τόσο γρήγορη ώστε ο χρήστης να μην την αντιλαμβάνεται. Τότε λέμε ότι έχουμε ψευδο-παράλληλο προγραμματισμό. Ένα παράδειγμα ψευδο-παράλληλου προγραμματισμού στο περιβάλλον της Logo θα δούμε παρακάτω.

Εκτός από τη βασική χελώνα είναι δυνατό να έχουμε και άλλες. Η βασική χελώνα αριθμείται ως 0 ενώ οι άλλες είναι οι 1, 2, 3 κ.λ.π. με το πλήθος τους να εξαρτάται από την έκδοση της Logo.

Με την εντολή TELL απευθυνόμαστε στη συγκεκριμένη χελώνα που θέλουμε. Για παράδειγμα αν θέλουμε να απευθυνθούμε στη χελώνα 1 και στη συνέχεια να της ζητήσουμε να πάει στις συντεταγμένες (X, Y) θα πρέπει να κάνουμε τα εξής:

```
TELL 1  
SETX :X  
SETY :Y
```

Αυτό έχει ως αποτέλεσμα να εμφανισθεί μια επιπλέον χελώνα (δίπλα στην χελώνα 0) και στη συνέχεια να μετατοπισθεί στη θέση (X, Y).



Η χελώνα σε ρόλο χαμαιλέοντα

Η χελώνα της logo (ο δρομέας της θέσης στην οθόνη) μπορεί να πάρει διάφορες μορφές εκτός από την εξ' ορισμού (που είναι η χελώνα). Αυτά τα σχήματα ποικίλουν από έκδοση σε έκδοση της logo. Στον επόμενο πίνακα παρουσιάζονται οι μορφές που μπορεί να πάρει ο δείκτης της οθόνης της logowriter.

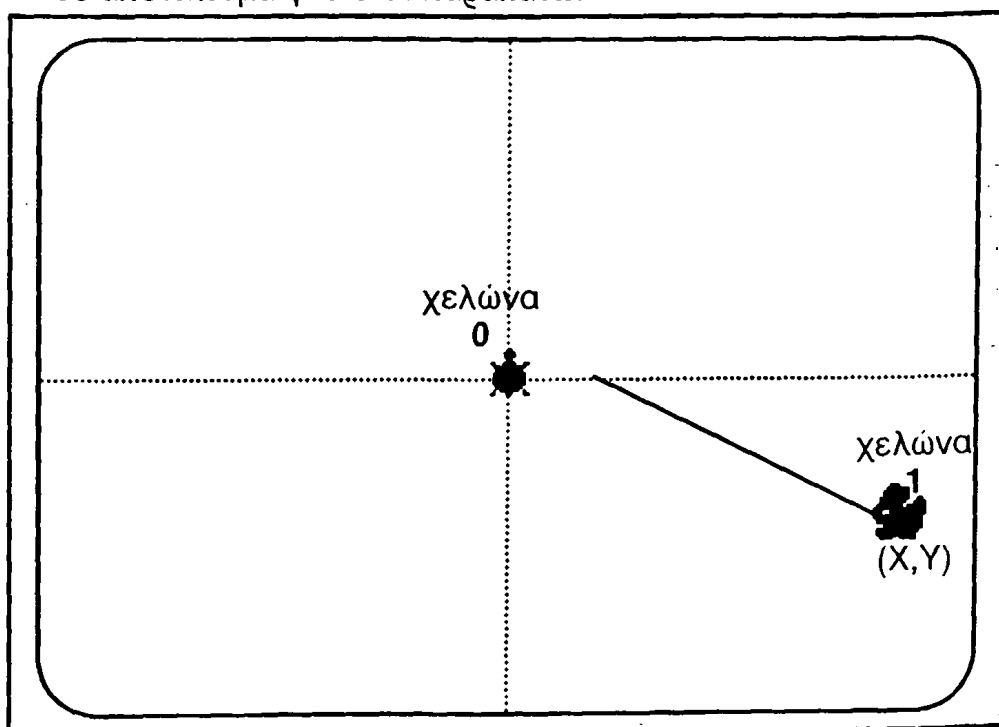
1	1	2	2	3	3	4	4	5	5	6	6
7	7	8	8	9	9	10	0	11	■	12	●
13	←	14	♥	15	😊	16	↑	17	↗	18	↘
19	↖	20	🏠	21	👨	22	👩	23	🌳	24	🎮
25	🚗	26	🚚	27	🚗	28	🚗	29	🚗	30	

Αν θέλουμε η χελώνα 1 να πάρει άλλη μορφή, θα πρέπει να χρησιμοποιήσουμε την εντολή SETSH. Για παράδειγμα αν θέλουμε να πάρει το σχήμα του λαγού θα χρησιμοποιήσουμε τις εντολές:

TELL 1

SETSH 22

Το αποτέλεσμα φαίνεται παρακάτω:



Λάβετε θέσεις, έτοιμοι...

Ας φτιάξουμε μια διαδικασία που θα βάζει το λαγό ή τη χελώνα στη θέση εκκίνησης για τον αγώνα.

Η διαδικασία ΕΚΚΙΝ :ΑΝΤΚ :Χ :Υ :SXHM

μετατοπίζει, χωρίς να αφήνει ίχνη, την υπ' αριθμό :ΑΝΤΚ χελώνα στη θέση με συντεταγμένες Χ και Υ, τη στρέφει προς τα αριστερά και της δίνει το σχήμα SXHM.

ΤΟ ΕΚΚΙΝ :ΑΝΤΚ :Χ :Υ :SXHM

TELL :ΑΝΤΚ

PU

SETX :Χ

SETY :Υ

PD

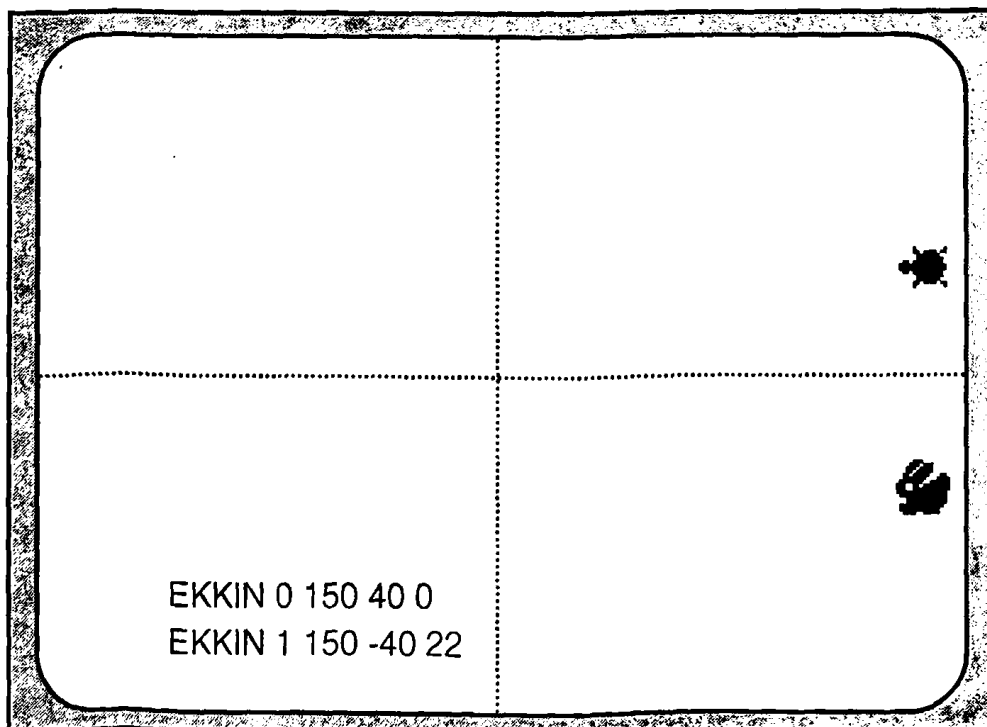
LT 90

SETSH :SXHM

ST

END

Αν την καλέσουμε δύο φορές με τις αντίστοιχες τιμές, θα βάλει το λαγό και τη χελώνα στη θέση εκκίνησης.



Φύγαμε...

Αν θέλουμε να φτιάξουμε ένα πρόγραμμα που να προσομοιώνει τον αγώνα του λαγού με τη χελώνα θα πρέπει να κινούμε το λαγό αλλά και τη χελώνα ταυτόχρονα. Όπως αναφέρθηκε παραπάνω η αίσθηση του «ταυτόχρονα» επιτυγχάνεται προγραμματίζοντας τη χελώνα να κάνει μια κίνηση, το λαγό να κάνει και αυτός μια κίνηση και επαναλαμβάνοντας αυτή τη διαδοχή των κινήσεων μέχρι κάποιος από τους δύο να φτάσει στο τέλος.

Ας δούμε λοιπόν τη διαδικασία KINSHH που κινεί διαδοχικά το λαγό και τη χελώνα.

Ας φτιάξουμε το πρόγραμμα που προσομοιώνει το τρέξιμο του λαγού και της χελώνας κάνοντάς τα να προχωρούν κάθε φορά κάποια "τυχαία" απόσταση.

ΤΟ ΚΙΝΗΣΗ

TELL 0

MAKE "BX RANDOM 4

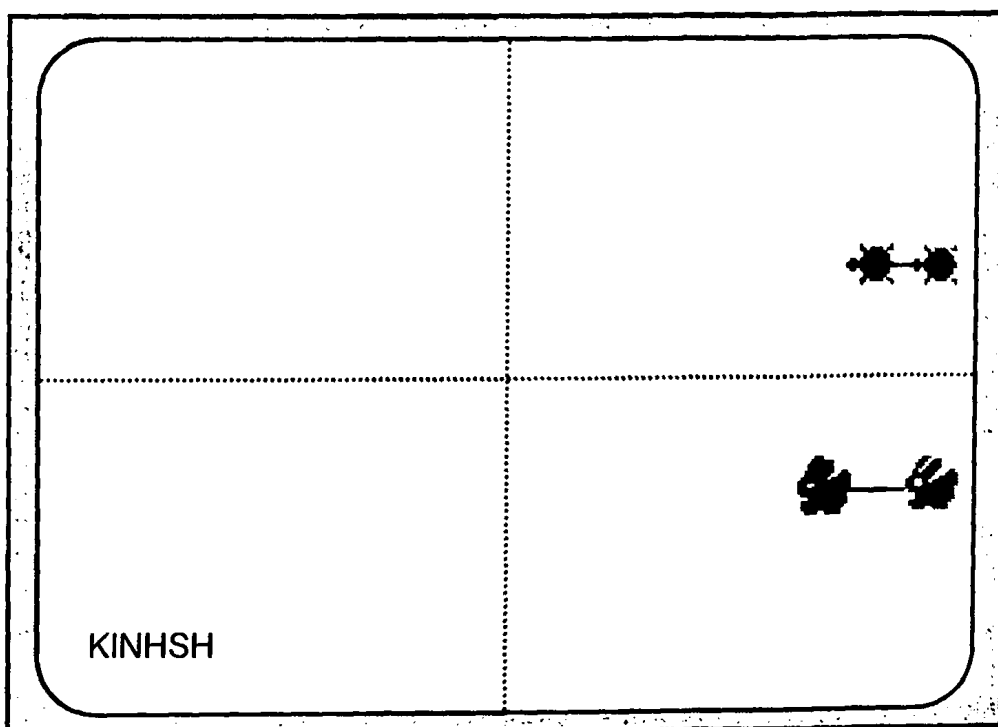
FD :BX

TELL 1

MAKE "BL RANDOM 10

FD :BL

END

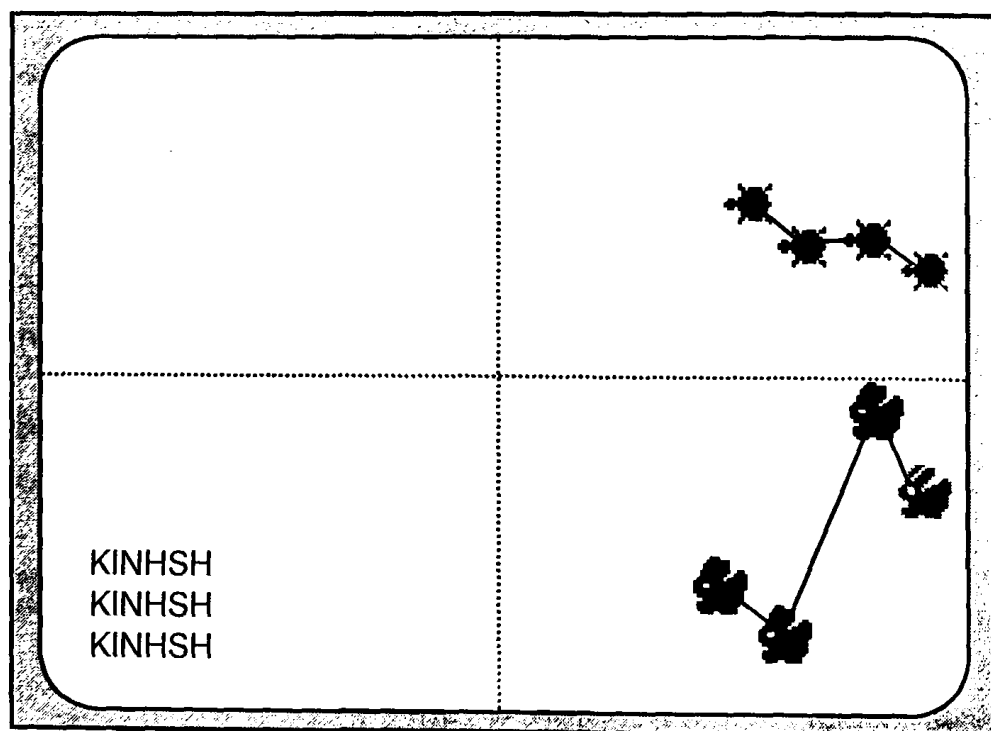


«Κίνηση Brown»

Για να γίνει πιο ενδιαφέρων ο αγώνας ας κάνουμε τον κάθε δρομέα να αποκλίνει (όπως θα έκανε ένα ζώο) κάθε φορά από τη σωστή κατεύθυνση κατά μια "τυχαία" γωνία. Η διαδικασία KINHSH γίνεται:

```
TO KINHSH
  TELL 0
  MAKE "GX RANDOM 10
  IFELSE :PKX = 1
    [RT :GX MAKE "PKX 0]
    [LT :GX MAKE "PKX 1]
  MAKE "BX RANDOM 4
  FD :BX
  TELL 1
  MAKE "GL RANDOM 30
  IFELSE :PKL = 1
    [RT :GL MAKE "PKL 0]
    [LT :GL MAKE "PKL 1]
  MAKE "BL RANDOM 10
  FD :BL
END
```

Οι μεταβλητές PKX και PKL χρησιμοποιούνται για να κάνουν τους αγωνιζόμενους τη μια φορά να αποκλίνουν προς τα δεξιά και την άλλη προς τα αριστερά.



Το τρέξιμο

Η διαδικασία KINHSH κινεί τους αγωνιζόμενους μόνο κατά ένα βήμα. Αν θέλουμε να έχουμε μια διαδικασία που να περιγράφει όλο το τρέξιμο του αγώνα θα πρέπει να τροποποιήσουμε τη διαδικασία KINHSH έτσι ώστε με αναδρομικό τρόπο να "επαναλαμβάνεται" μέχρι ο λαγός ή η χελώνα να φτάσει στο τέλος. Έτσι προκύπτει η διαδικασία TREXIMO.

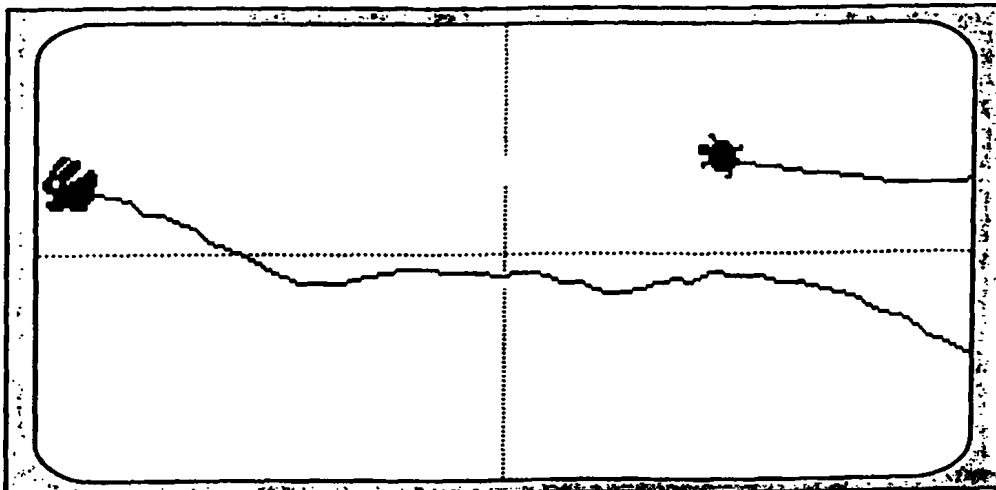
TO TREXIMO

```
TELL 0
MAKE "GX RANDOM 10
IFELSE :PKX = 1
    [RT :GX MAKE "PKX 0]
    [LT :GX MAKE "PKX 1]
MAKE "BX RANDOM 4
FD :BX
TELL 1
MAKE "GL RANDOM 30
IFELSE :PKL = 1
    [RT :GL MAKE "PKL 0]
    [LT :GL MAKE "PKL 1]
MAKE "BL RANDOM 10
FD :BL
```

KINHSH

```
TELL 0
MAKE "XXEL XCOR
TELL 1
MAKE "XLAG XCOR
MAKE "TELOS OR :XXEL < -150 :XLAG < -150
IF NOT :TELOS [TREXIMO]
```

END

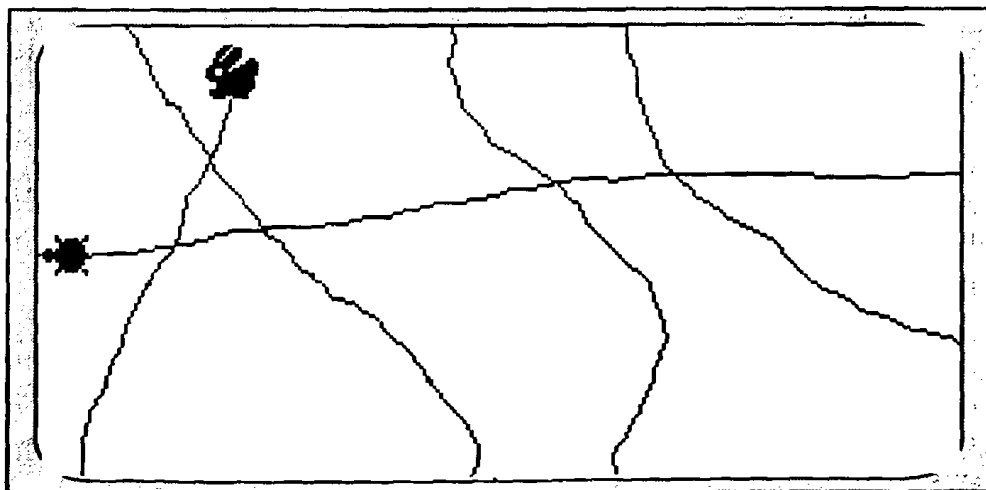


Νυν υπέρ πάντων αγών...

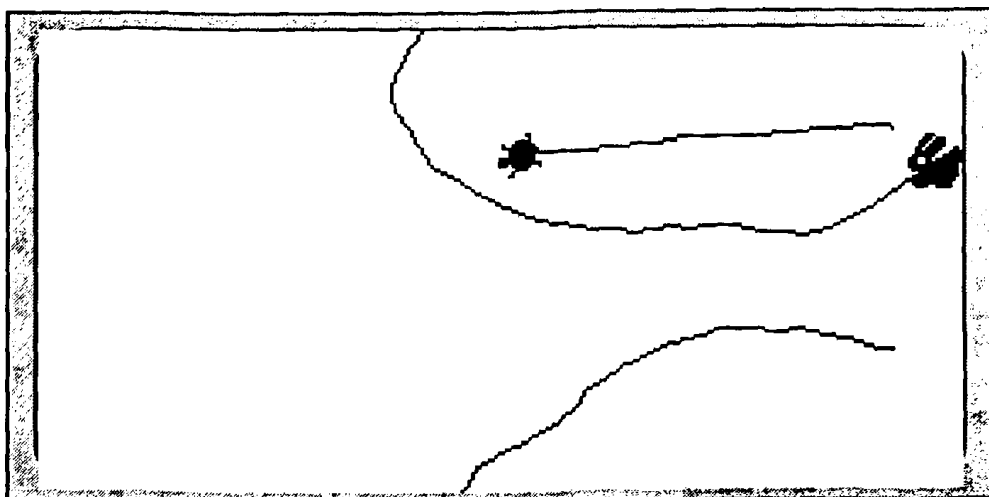
Το κύριο πρόγραμμα που θα βάζει το λαγό και τη χελώνα στην αφετηρία και θα καλεί την διαδικασία TREXIMO είναι το παρακάτω:

```
TO AGWNAS
  EKKIN 0 150 40 0
  EKKIN 1 150 -40 22
  MAKE "PKX 1
  MAKE "PKL 1
  TREXIMO
END
```

Η έκβαση του αγώνα καθορίζεται από πολλούς τυχαίους παράγοντες και έτσι άλλες φορές βλέπουμε το λαγό να νικά (όπως προηγουμένως) ...



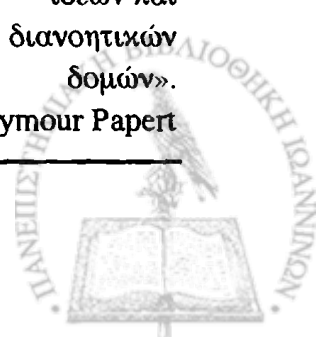
...και άλλες φορές να νικά η χελώνα, αφού ο λαγός αν και ταχύτερος αποπροσανατολίζεται δίνοντας στη χελώνα την ευκαιρία να τερματίσει ή να χάσει τον αγώνα επιστρέφοντας στην αφετηρία.



«Οι μαθητές δεν μπορούν να κάνουν κτήμα τους κάτι που δεν το γνωρίζουν. Γι' αυτό είναι προαπαιτούμενο να βρούμε τρόπους να διευκολύνουμε την προσωπική ιδιοποίηση όχι μόνο της νευτώνειας κίνησης και των νόμων που την περιγράφουν, αλλά επίσης της γενικής έννοιας των νόμων.

Αυτό το κάνουν σχεδιάζοντας μια σειρά από μικρόκοσμους. Ο σχεδιασμός του μικρόκοσμου τον κάνει "φυτώριο" ενός συγκεκριμένου είδους δυναμικών ιδεών και διανοητικών δομών».

Seymour Papert



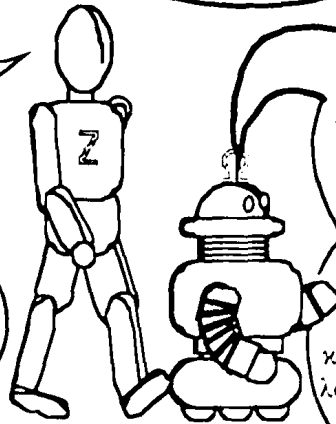
Η άποψή σας είναι σχεδόν αιρετική! Δεν είναι δυνατό να πιστεύετε ότι οι πρόγονοί μας οι υπολογιστές, ήταν ανθρώπινες κατασκευές!

Αν πρόκειται να συζητήσουμε για τη δημιουργία μας, θα πρέπει να τη δούμε σαν μέρος της ιστορίας της φύσης



Η προϊστορία μας αρχίζει πριν δισεκατομμύρια χρόνια. Το big-bang δημιούργησε μια σούπα πρωτονίων, νετρονίων, νετρίνων και φωτονίων. Μ' αυτά σχηματίστηκαν τα άτομα και τα πρώτα μόρια. Μια συσσωμάτωση τέτοιων μορίων φτιάξαν τη γη. Εκεί μετά από πολλές δοκιμές εμφανίστηκε ένα σύνθετο μόριο, το DNA, που ήταν ικανό να αναπαράγει τον εαυτό του. Αυτό είναι το θεμέλιο της ιστορίας, στην οποία εμείς μαζί με τους ανθρώπους είμαστε μια από τις πρόσκαιρες καταλήξεις της

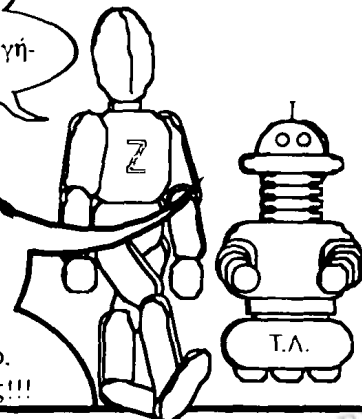
Το μόνο που μπορώ να δεχθώ είναι ότι υπάρχω γιατί σκέπτομαι. Νομίζω ότι δύσκολα θα αποδείξετε ότι οι λεπτεπίλεπτες πνευματικές μας λειτουργίες είναι προϊόντα αυτών των χονδροκομμένων υπολογιστών και όχι δημιουργήματα μιας Ανώτατης Διάνοιας!



Στην εξέλιξη της ζωής, η πραγματικότητα κάθε στιγμής καθορίζει το σύνολο των δυνατοτήτων της επόμενης στιγμής. Έτσι μέσα από μια μακρόχρονη πορεία με πρωταγωνιστές την τύχη και την αναγκαιότητα δημιουργήθηκε ο άνθρωπος. Αυτός κατασκεύασε τους πρώτους υπολογιστές, τους προγόνους μας.

Το να υποθέσω ότι είμαι απόγονος μιας ανθρώπινης κατασκευής, είναι αντίθετο σε κάθε κανόνα της λογικής γιατί κανένα πλάσμα δεν μπορεί να δημιουργήσει ένα άλλο πλάσμα ανώτερο από το ίδιο.

Κι όμως, ο άνθρωπος κατάφερε να ξεπεράσει τον εαυτό του αναπτύσσοντας την τεχνητή νοημοσύνη. Αυτή του επέτρεψε να κατασκευάσει έξυπνες μηχανές οι οποίες μπορούσαν να δημιουργούν μηχανές πιο έξυπνες από τον εαυτό τους. Έτσι δημιουργήθηκαν εμείς και μέσω της τεχνητής ευφυΐας γίναμε, μαζί με τον άνθρωπο, φορείς της συνείδησης του σύμπαντος!!!



16 Λίστες. Μια δυναμική δομή της logο

Υλοποίηση δομής στοίβας με λίστες

Το φυσικό πρόβλημα:

Θέλουμε να προγραμματίσουμε έναν υπολογιστή ώστε να αυτοματοποιεί τη διαδικασία εισόδου - εξόδου (παρκάρισμα - ξεπαρκάρισμα) σε ένα ιδιόμορφο γκαράζ. Το γκαράζ είναι στενόμακρο, με μια λωρίδα και μια πόρτα (είσοδο / έξοδο). Επίσης δίπλα από το γκαράζ υπάρχει ένα όμοιο πάρκινγκ για να διευκολύνει τις manούβρες των αυτοκινήτων. Όταν έρχεται ένα αυτοκίνητο απλά χρειάζεται να το βάλουμε στο γκαράζ πίσω από το τελευταίο αυτοκίνητο που είναι παρκαρισμένο. Όταν έρχεται κάποιος και ζητάει το αυτοκίνητό του τότε θα πρέπει να βγάζουμε ένα ένα τα αυτοκίνητα (βάζοντάς τα προσωρινά στο πάρκινγκ) μέχρι να βγάλουμε το σωστό. Αφού παραδώσουμε το ζητούμενο αυτοκίνητο πρέπει να μεταφέρουμε τα αυτοκίνητα που απέμειναν στο πάρκινγκ ξανά στο γκαράζ.

Το γκαράζ και το πάρκινγκ μπορούμε να τα προσομοιώσουμε με δομές στοίβας που θα τις υλοποιήσουμε με λίστες. Μια στοίβα (stack) είναι μια δομή δεδομένων όπου ό,τι μπαίνει τελευταίο βγαίνει πρώτο (Last In First Out, LIFO).

Κατ' αρχήν ας γράψουμε τη διαδικασία INITG που δημιουργεί μια άδεια λίστα με όνομα GARAGE και τη διαδικασία INITP που δημιουργεί μια άδεια λίστα με όνομα PARKING.

```
TO INITG                TO INITP
  MAKE "GARAGE []      MAKE "PARKING []
END                      END
```

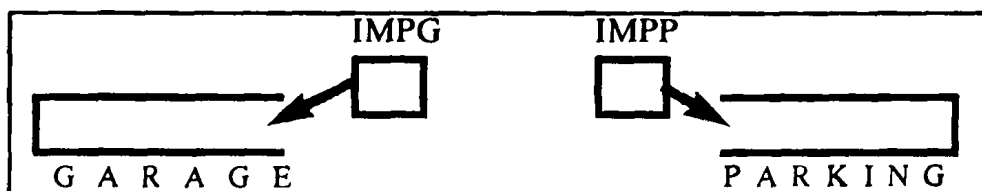


To parking

Η διαδικασία **BALEG** βάζει στη στοίβα **GARAGE** ως πρώτο στοιχείο (**FPUT = FirstPUT**) ό,τι υπάρχει στη μεταβλητή **IMPG**. Όμοια η διαδικασία **BALEP** βάζει στη στοίβα **PARKING** ως πρώτο στοιχείο ό,τι υπάρχει στη μεταβλητή **IMPP**.

```
TO BALEG :IMPG
  MAKE "GARAGE FPUT :IMPG :GARAGE
END
```

```
TO BALEP :IMPP
  MAKE "PARKING FPUT :IMPP :PARKING
END
```



Η διαδικασία **BGALEG** βγάζει από τη στοίβα **GARAGE** το στοιχείο που μπήκε τελευταίο και το φυλάει στη μεταβλητή **EXPG** αφού εξετάσει (με τη συνθήκη **EMPTY?**) αν η λίστα **GARAGE** είναι άδεια.

Η εντολή **MAKE "EXPG FIRST :GARAGE**
βάζει στη μεταβλητή **EXPG** ένα αντίγραφο του πρώτου στοιχείου που υπάρχει στη λίστα **:GARAGE**
ενώ η εντολή **MAKE "GARAGE BF :GARAGE**
βάζει στη λίστα **"GARAGE** όλα τα στοιχεία της λίστας **:GARAGE** εκτός από το πρώτο (**BF = ButFirst**).

Όμοια η διαδικασία **BGALEP** βγάζει από τη στοίβα **PARKING** το στοιχείο που μπήκε τελευταίο και το φυλάει στη μεταβλητή **EXPP**.

```
TO BGALEG
  IFELSE EMPTY? :GARAGE
    [PR [το γκαράζ είναι άδειο] ]
    [MAKE "EXPG FIRST :GARAGE
     MAKE "GARAGE BF :GARAGE ]
END
```



```

TO BGALEP
  IFELSE EMPTY? :PARKING
    [PR [ADEIO PARKING]]
    [MAKE "EXPP LAST :PARKING
    MAKE "PARKING BL :PARKING]
END

```

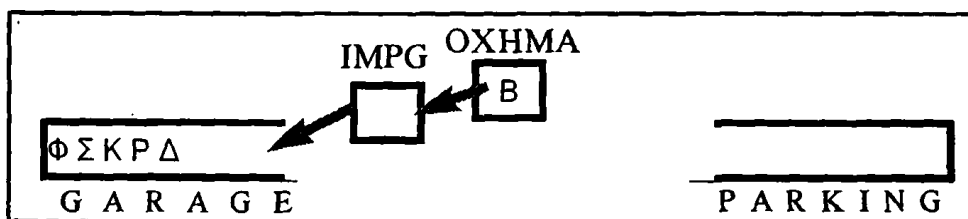


Ας δούμε πως θα φτιάξουμε τη διαδικασία EISHGAGE που θα πάρει από το πληκτρολόγιο τον κωδικό ενός αυτοκινήτου και θα τον βάλει στο γκαράζ.

```

TO EISHGAGE
  PR [Δώσε τον κωδικό του οχήματος (χαρακτήρα)]
  MAKE "OXHMA READCHAR
  BALEG :OXHMA
END

```



Η διαδικασία TYPWSEG εμφανίζει στην οθόνη όλα τα στοιχεία της λίστας GARAGE. Όμοια η διαδικασία TYPWSEP εμφανίζει όλα τα στοιχεία της λίστας PARKING.

```

TO TYPWSEG
  SHOW :GARAGE
END

```

```

TO TYPWSEP
  SHOW :PARKING
END

```



Η διαδικασία TYPWSE εμφανίζει στην οθόνη τα περιεχόμενα και των δύο λιστών

```
TO TYPWSE
  SHOW [περιεχόμενο γκαράζ]
  TYPWSEG
  SHOW [περιεχόμενο πάρκινγκ]
  TYPWSEP
END
```

Ας δούμε πως μπορούμε να βγάλουμε από το γκαράζ το αυτοκίνητο που ζητάει κάποιος. Η διαδικασία EXHGAGE αφού ζητήσει να μάθει ποιο όχημα ζητείται, ελέγχει αν αυτό υπάρχει στο γκαράζ με τη συνθήκη **MEMBER? :OXHMA :GARAGE** και αν υπάρχει τότε καλεί τη διαδικασία PARE που αναλαμβάνει να παραδώσει το όχημα.

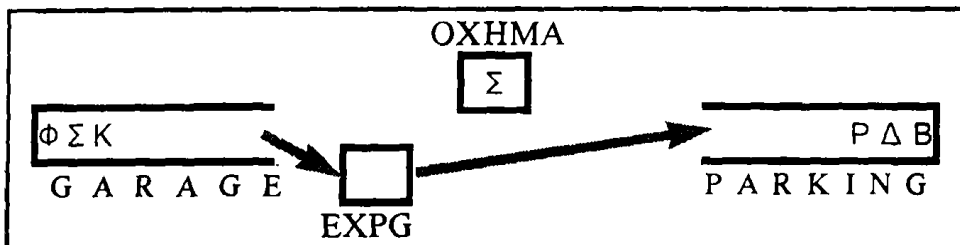
```
TO EXHGAGE
  PR [Δώσε τον κωδικό του οχήματος (χαρακτήρα)]
  MAKE "OXHMA READCHAR
  IFELSE MEMBER? :OXHMA :GARAGE
    [IF NOT EMPTY? :GARAGE
      [ PARE :OXHMA
        TYPWSE ]
    [PR [Δεν υπάρχει τέτοιο όχημα στο γκαράζ] ] ]
END
```

Ας δούμε πως η διαδικασία PARE διεκπεραιώνει το καθήκον της που είναι να βρεί το ζητούμενο όχημα και να το παραδώσει.

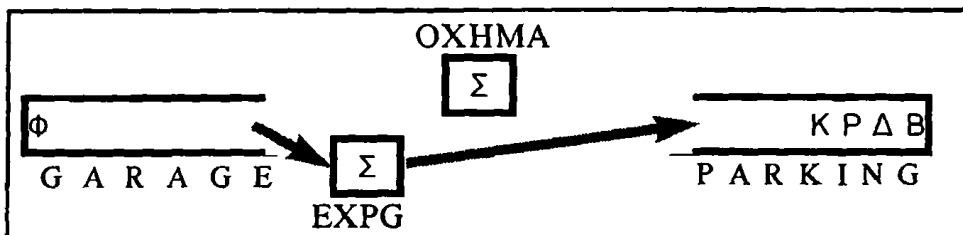
```
TO PARE :OXHMA
  BGALEG
  IFELSE :EXPG = :OXHMA
    [ PR [πάρε το όχημα ]
      ADEIASEP ]
    [ BALEP :EXPG
      PARE :OXHMA ]
END
```



Αυτό το πετυχαίνει (με αναδρομικό τρόπο) βγάζοντας ένα-ένα τα οχήματα από το γκαράζ και βάζοντας τα στο πάρκινγκ μέχρι να βρεθεί το ζητούμενο.



Όταν βρεί το ζητούμενο όχημα, το παραδίδει στον κάτοχό του



και επαναφέρει στο γκαράζ τα οχήματα που έχουν μεταφερθεί προσωρινά στο παρκινγκ με τη βοήθεια της διαδικασίας ADEIASEP.

Η διαδικασία ADEIASEP (με αναδρομικό τρόπο) βγάζει ένα-ένα τα οχήματα από το πάρκινγκ και τα βάζει στο γκαράζ μέχρι να αδειάσει το πάρκινγκ.

ΤΟ ADEIASEP

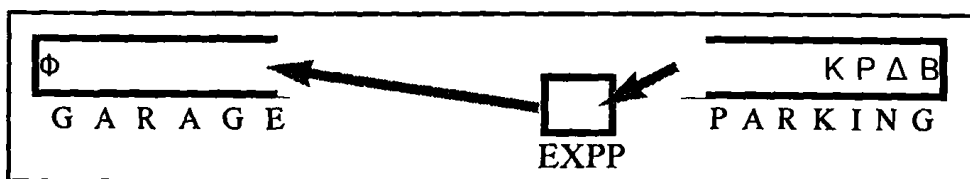
IF NOT EMPTY? :PARKING

[BGALEP

BALEG :EXPP

ADEIASEP]

END



Η διαδικασία EXYPHRETHSH επιτρέπει να επικοινωνήσει το πρόγραμμα με τον άνθρωπο που θα ζητήσει να παρκάρει ή να ξεπαρκάρει το όχημά του. Η διαδικασία EXYPHRETHSH ζητάει από τον οδηγό να την ενημερώσει για το αν θα γίνει εισαγωγή οχήματος, εξαγωγή ή τέλος του προγράμματος. Σε περίπτωση που δεν επιλέξει μία από τις τρεις περιπτώσεις η διαδικασία καλεί τον εαυτό της ζητώντας νέα επιλογή.

TO EXYPHRETHSH

PR [Να κάνω εισαγωγή, εξαγωγή ή τέλος]

MAKE "ENERGEIA READLIST

IFELSE :ENERGEIA = [εισαγωγή]

[EISHGAGE]

[IFELSE :ENERGEIA = [εξαγωγή]

[EXHGAGE]

[IF :ENERGEIA = [τέλος]

[STOPALL]

]

]

TYPWSE

EXYPHRETHSH

END

Αφού φτιάξαμε όλες τις απαραίτητες διαδικασίες που θα διαχειριστούν το πρόβλημα του ιδιόμορφου γκαράζ ας φτιάξουμε και το τελικό πρόγραμμα PARKADOROS.

TO PARKADOROS

INITG

INITP

EXYPHRETHSH

END



Μια κερκόπορτα για την τεχνητή ευφυΐα

Στη logo αντικείμενα μιας λίστας μπορούν να είναι και εντολές της. Έτσι παρέχεται η δυνατότητα στον προγραμματιστή να βάλει σε μια λίστα κάποιες εντολές

π.χ. MAKE "ENTOLES [REPEAT 4 [RT 90 FD 10]]

και στη συνέχεια με την εντολή RUN να ζητήσει να εκτελεσθούν.

RUN :ENTOLES

Ένα παράδειγμα δίνεται με το παρακάτω πρόγραμμα που με αναδρομικό τρόπο ανεβοκατεβάζει τη χελώνα μεταξύ του άνω και του κάτω ορίου της οθόνης εκμεταλλευόμενο την προαναφερθείσα δυνατότητα της logo.

TO ANWKATW

MAKE "QESH POS

MAKE "Y LAST :QESH

IFELSE AND :Y < 80 :Y > -80

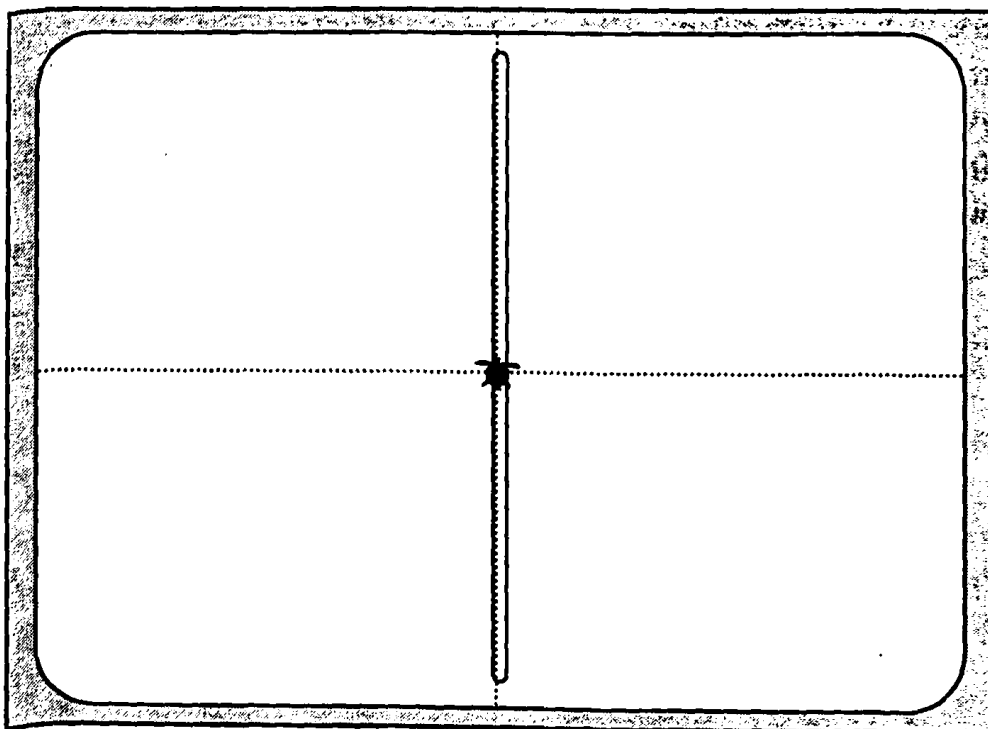
[MAKE "ENTOLH [FD 1]]

[MAKE "ENTOLH [RT 180 FD 1]]

RUN :ENTOLH

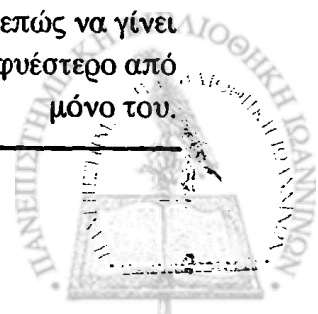
ANWKATW

END



Στη δυνατότητα της logo να αλλάξει τον κώδικα του προγράμματος ενώ αυτό εκτελείται, υπάρχει το σπέρμα της τεχνητής ευφυΐας.

Το πρόγραμμα κατά τη διάρκεια της εκτέλεσής του μπορεί από τα αποτελέσματα που προκύπτουν να αποφασίζει να αλλάξει την δομή του, να αυτοβελτιωθεί και συνεπώς να γίνει ευφυέστερο από μόνο του.



Το πρόγραμμα "αλλάζει" τον εαυτό του

Ας χρησιμοποιήσουμε τη δυνατότητα της Logo να αλλάζει το πρόγραμμα ενώ αυτό εκτελείται για να μετασχηματίσουμε το πρόγραμμα που προσομοιώνει τον αγώνα του λαγού με τη χελώνα.

Ας φτιάξουμε μια διαδικασία, την INITKNS, η οποία να δημιουργεί μια λίστα KNS που να περιέχει ως δεδομένα τις εντολές που περιγράφουν την κίνηση ενός αντικειμένου (του λαγού ή της χελώνας).

TO INITKNS

```
MAKE "KNS [TELL :ANTK  
MAKE "GA RANDOM :EGA  
RT :GA  
MAKE "PKA :PERA  
MAKE "BA RANDOM :EBA  
FD :BA ]
```

END

Η μεταβλητή ANTK μπορεί να περιέχει την τιμή 0 ή 1 που να αντιστοιχεί στη χελώνα ή το λαγό. Η μεταβλητή EGA περιέχει τη μέγιστη γωνία που μπορεί να αποκλίνει από την πορεία του το αντικείμενο, η μεταβλητή GA τη γωνία που θα αποκλίνει από την πορεία του το αντικείμενο, η μεταβλητή EBA το μέγιστο βήμα που μπορεί να κάνει το αντικείμενο, η μεταβλητή BA το βήμα που θα κάνει το αντικείμενο. Η μεταβλητή PKA χρησιμεύει για να στρίβει εναλλάξ το αντικείμενο αριστερά και δεξιά.

Η επόμενη διαδικασία INITLX καλεί τη διαδικασία INITKNS και δίνει αρχικές τιμές στις μεταβλητές ANTK, EGA, PERA, EBA.

TO INITLX

```
INITKNS  
MAKE "ANTK 0  
MAKE "EGA 10  
MAKE "PERA 1  
MAKE "EBA 5
```

END



Η διαδικασία ΕΚΚΙΝ είναι η ίδια που χρησιμοποιήθηκε στην προηγούμενη έκδοση του προγράμματος και σκοπό έχει να βάλει το αντικείμενο στη θέση εκκίνησης.

```
TO ΕΚΚΙΝ :ΑΝΤΚ :Χ :Υ :ΣΧΗΜ
  ΤΕΛΛ :ΑΝΤΚ
  ΡΥ
  ΣΕΤΧ :Χ
  ΣΕΤΥ :Υ
  ΡΔ
  ΛΤ 90
  ΣΕΤΣΗ :ΣΧΗΜ
  ΣΤ
ΕΝΔ
```

Ας δούμε τώρα τη διαδικασία ΚΙΝΛΧ που είναι αυτή που με αναδρομικό τρόπο περιγράφει την κίνηση του λαγού και της χελώνας μέχρι κάποιο από τα δύο να τερματίσει.

```
ΤΟ ΚΙΝΛΧ
  ΙΦΕΛΣΕ :ΡΕΡΑ = 1
    [ΑΛΛΑΧΕ 7 "ΛΤ
      ΚΙΝΗΣΕΛΧ
      ΜΑΚΕ "ΡΕΡΑ 0
    ]
    [ΑΛΛΑΧΕ 7 "ΡΤ
      ΚΙΝΗΣΕΛΧ
      ΜΑΚΕ "ΡΕΡΑ 1
    ]
  ΤΕΛΛ 0
  ΜΑΚΕ "ΧΧΕΛ ΧΧΟΡ
  ΤΕΛΛ 1
  ΜΑΚΕ "ΧΛΑΓ ΧΧΟΡ
  ΜΑΚΕ "ΤΕΛΟΣ ΟΡ :ΧΧΕΛ < -150 :ΧΛΑΓ < -150
  ΙΦ ΝΟΤ :ΤΕΛΟΣ [ΚΙΝΛΧ]
ΕΝΔ
```



Ανάλογα με την τιμή της μεταβλητής PERA, η διαδικασία ALLAXE (θα δούμε παρακάτω πως) αλλάζει στη λίστα KNS το 7^ο στοιχείο τη μια φορά σε LT και την άλλη φορά σε RT.

Η διαδικασία KINHSELX θέτει (με τη βοήθεια της διαδικασίας ALLAXE) τις τιμές εκείνες στη λίστα KNS που αντιστοιχούν στα δεδομένα της χελώνας και μετά με την εντολή RUN :KNS πραγματοποιεί το “στιγμιαίο” βήμα της χελώνας. Στη συνέχεια κάνει την αντίστοιχη δουλειά και κινεί στιγμιαία και το λαγό.

TO KINHSELX

ALLAXE 2 0

ALLAXE 6 15

ALLAXE 15 5

RUN :KNS

ALLAXE 2 1

ALLAXE 6 30

ALLAXE 15 10

RUN :KNS

END

Ας δούμε τώρα πως η διαδικασία ALLAXE αλλάζει το νιοστό στοιχείο μιας λίστας (το AAST) με την τιμή της μεταβλητής NEO.

TO ALLAXE :AAST :NEO

MAKE "PRS []

REPEAT :AAST [BGALEKNS BALEPRS]

MAKE "PRS BF :PRS

MAKE "KNS FPUT :NEO :KNS

MAKE "AAST :AAST - 1

REPEAT :AAST [BGALEPRS BALEKNS]

END

Με την εντολή MAKE "PRS []

φτιάχνει μια προσωρινή άδεια λίστα. Στη συνέχεια με την εντολή REPEAT βγάζει τα ν πρώτα στοιχεία της λίστας KNS (με τη διαδικασία BGALEKNS) και για να μη χαθούν τα φυλάει στη λίστα PRS (με τη διαδικασία BALEPRS). Μετά σβήνει το τελευταίο στοιχείο



που μπήκε στη λίστα PRS και βάζει στη λίστα KNS το στοιχείο που υπάρχει στη μεταβλητή NEO. Τέλος μεταφέρει ξανά όλα τα στοιχεία της λίστας PRS στη λίστα KNS.

Οι λίστες KNS και PRS δουλεύονται ως στοίβες και έτσι χρησιμοποιούνται οι διαδικασίες BALEKNS, BGALEKNS, BALEPRS και BGALEPRS που βάζουν και βγάζουν ένα στοιχείο αντίστοιχα.

```
TO BALEKNS
  MAKE "KNS FPUT :STOIXEIO :KNS
END
```

```
TO BGALEKNS
  IF NOT EMPTY? :KNS
    [MAKE "STOIXEIO FIRST :KNS
    MAKE "KNS BF :KNS]
END
```

```
TO BALEPRS
  MAKE "PRS FPUT :STOIXEIO :PRS
END
```

```
TO BGALEPRS
  IF NOT EMPTY? :PRS
    [MAKE "STOIXEIO FIRST :PRS
    MAKE "PRS BF :PRS]
END
```

Ας δούμε τώρα και το βασικό πρόγραμμα που θα φτιάξει τον αγώνα του λαγού με τη χελώνα.

```
TO AGWN
  EKKIN 0 150 40 0
  EKKIN 1 150 -40 22
  INITLX
  KINLX
END
```



Ανάπτυξη εργαλείων λογισμικού

Συχνά υπάρχει η ανάγκη να κάνουμε κάποιο χειρισμό αλλά η γλώσσα προγραμματισμού στην οποία δουλεύουμε δεν διαθέτει την αντίστοιχη εντολή. Τότε μπορούμε να φτιάξουμε μια διαδικασία που να χρησιμοποιεί τις υπάρχουσες εντολές της γλώσσας και να κάνει τη δουλειά που θέλουμε. Για παράδειγμα θέλουμε να φτιάξουμε την "εντολή-φίλτρο" ΠΑΡΕ_ΑΡΙΘΜΟ που να παίρνει από το πληκτρολόγιο έναν πολυψήφιο ακέραιο αριθμό αγνοώντας τυχόν άλλους χαρακτήρες που μπορεί να πληκτρολογηθούν κατά λάθος.

Η διαδικασία ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ χρησιμοποιεί την εντολή READCHAR και παίρνει από το πληκτρολόγιο έναν χαρακτήρα. Αν είναι χαρακτήρας που αντιστοιχεί σε αριθμητικό ψηφίο ή ο χαρακτήρας που αντιστοιχεί στο RETURN τον φυλάει στη μεταβλητή MONOCHAR αλλιώς τον αγνοεί και με αναδρομικό τρόπο παίρνει από το πληκτρολόγιο τον επόμενο χαρακτήρα.

```
TO ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ
  MAKE "ΑΡΙΘΜΟΙ [ 0 1 2 3 4 5 6 7 8 9 ]
  MAKE "ENTER CHAR 13
  MAKE "MONOCHAR READCHAR
  MAKE "ΤΕΛΟΣ :MONOCHAR = :ENTER
  IF NOT MEMBER? :MONOCHAR :ΑΡΙΘΜΟΙ
    [ IF NOT :ΤΕΛΟΣ [ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ] ]
END
```

Ο χαρακτήρας που υπάρχει στη μεταβλητή MONOCHAR μπορεί να μετασχηματισθεί στον αντίστοιχο του μονοψήφιο αριθμό PSHFIO με τις εντολές:

```
MAKE "MONONUM ASCII :MONOCHAR
MAKE "PSHFIO :MONONUM - 48
```

όπου 48 είναι ο αριθμός του κώδικα ASCII που αντιστοιχεί στον χαρακτήρα του αριθμού μηδέν.

Η διαδικασία ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ με αναδρομικό τρόπο παίρνει πολλούς χαρακτήρες και αφού τους φιλτράρει τους μετατρέπει σε πολυψήφιο ακέραιο που τον βάζει στη μεταβλητή NUM.




```

ΤΟ ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ
  ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ
  IFELSE NOT :ΤΕΛΟΣ
    [ MAKE "ΜΟΝΟΝΟΜ ASCII :ΜΟΝΟΧΑΡ
      MAKE "ΨΗΦΙΟ :ΜΟΝΟΝΟΜ - 48
      MAKE "ΝΟΜ :ΝΟΜ * 10
      MAKE "ΝΟΜ :ΝΟΜ + :ΨΗΦΙΟ
      ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ
    ]
  [STOP]
END

```

Τέλος, η διαδικασία ΠΑΡΕ_ΑΡΙΘΜΟ αφού μηδενίσει αρχικά τον αριθμό ΝΟΜ καλεί την διαδικασία ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ και πετυχαίνει το επιθυμητό αποτέλεσμα.

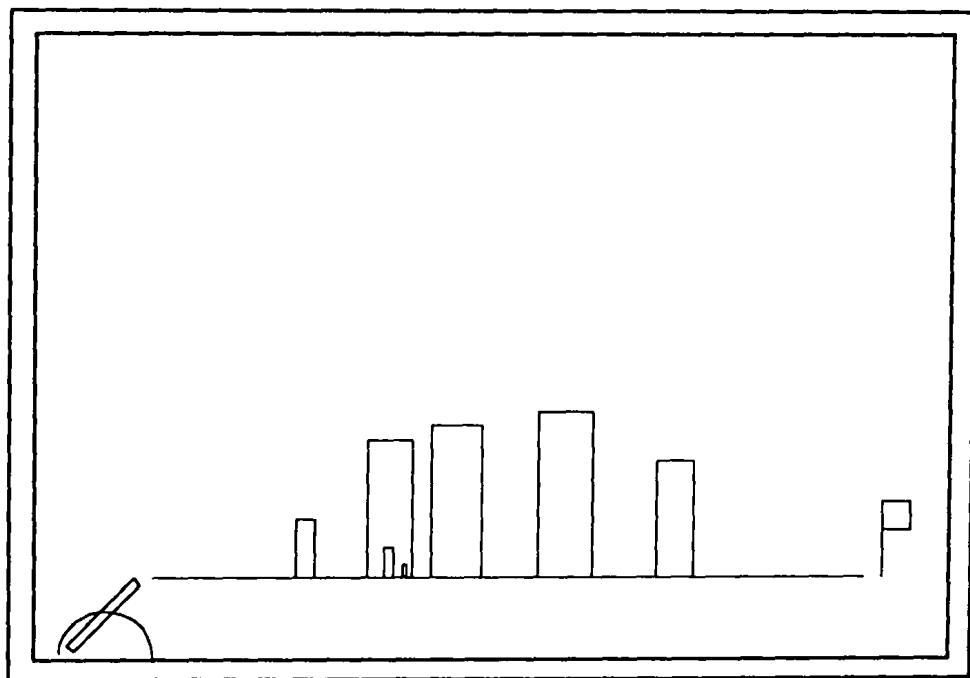
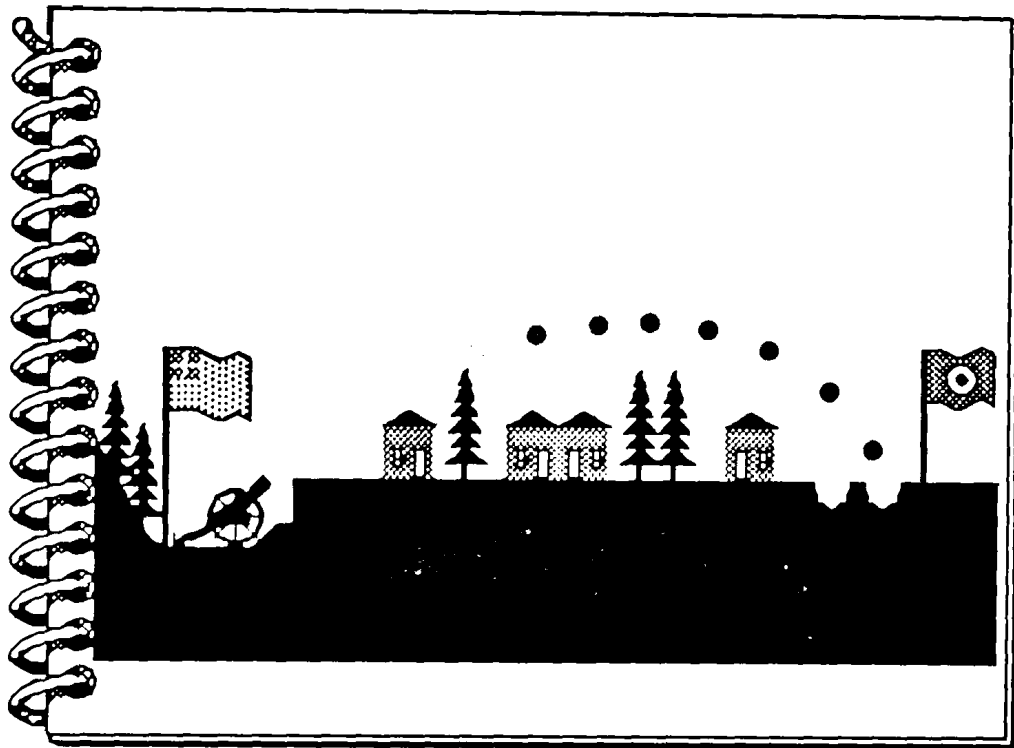
```

ΤΟ ΠΑΡΕ_ΑΡΙΘΜΟ
  MAKE "ΝΟΜ 0
  ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ
END

```

Για να εμφανισθεί στην οθόνη ο αριθμός, πρέπει να συμπεριληφθεί στη διαδικασία ως τελευταία εντολή η PR :ΝΟΜ.

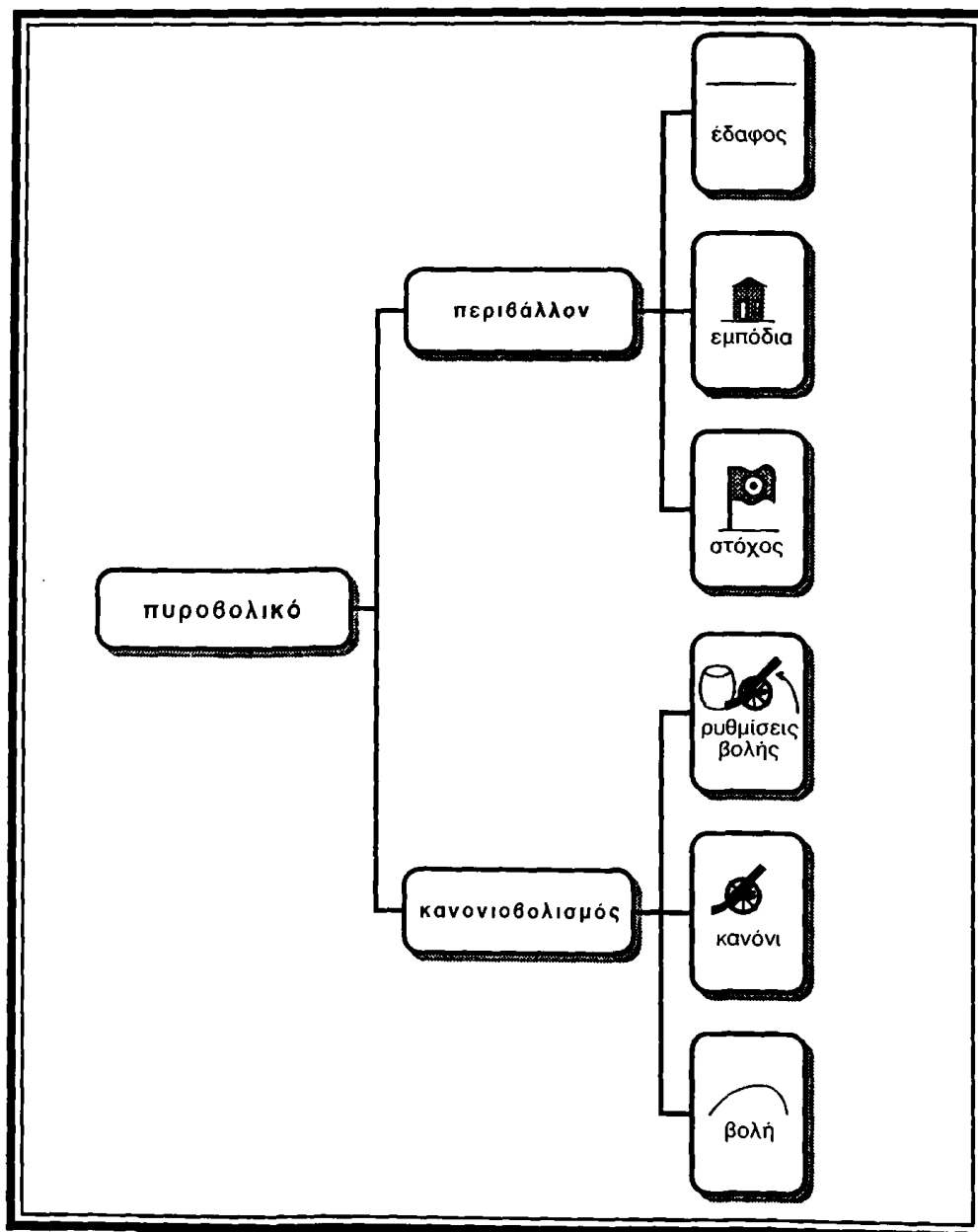
Σε περίπτωση που πληκτρολογηθεί αλφαριθμητικός χαρακτήρας (όχι αριθμός), στην οθόνη εμφανίζεται το 0.



17 Μικρόκοσμοι

Ο οπτικός πίνακας περιεχομένων

Το «Cannon Fodder» είναι ένα παιχνίδι για υπολογιστή που προσο-
μοώνει μια μάχη πυροβολικού. Θα προσπαθήσουμε να το υλοποιήσουμε με
logo. Στο πρώτο επίπεδο του οπτικού πίνακα περιεχομένων, οι εργασίες που
πρέπει να γίνουν είναι να ζωγραφιστεί το περιβάλλον και να ρυθμιστεί,
υπολογιστεί και σχεδιαστεί η διαδικασία του κανονιοβολισμού. Στις επό-
μενες σελίδες θα αναπτύξουμε την κάθε μια διαδικασία ξεχωριστά.



Το έδαφος

Ξεκινώντας να σχεδιάζουμε το περιβάλλον, ας αρχίσουμε με το έδαφος. Για να δημιουργήσουμε την αίσθηση του συμπαγούς που πρέπει να έχει το έδαφος, σχεδιάζουμε πολλές παράλληλες γραμμές από την επιφάνεια του εδάφους μέχρι κάποιο βάθος. Για να μη σχηματισθεί μια μαύρη επιφάνεια, μεταξύ δύο διαδοχικών γραμμών αφήνονται δύο κενές.

ΤΟ ΕΔΑΦΟΣ

MAKE "X MINUS 150

MAKE "Y MINUS 80

REPEAT 5

[ΜΗ_ΓΡΑΦΕΙΣ

SETX :X

SETY :Y

ΓΡΑΨΕ

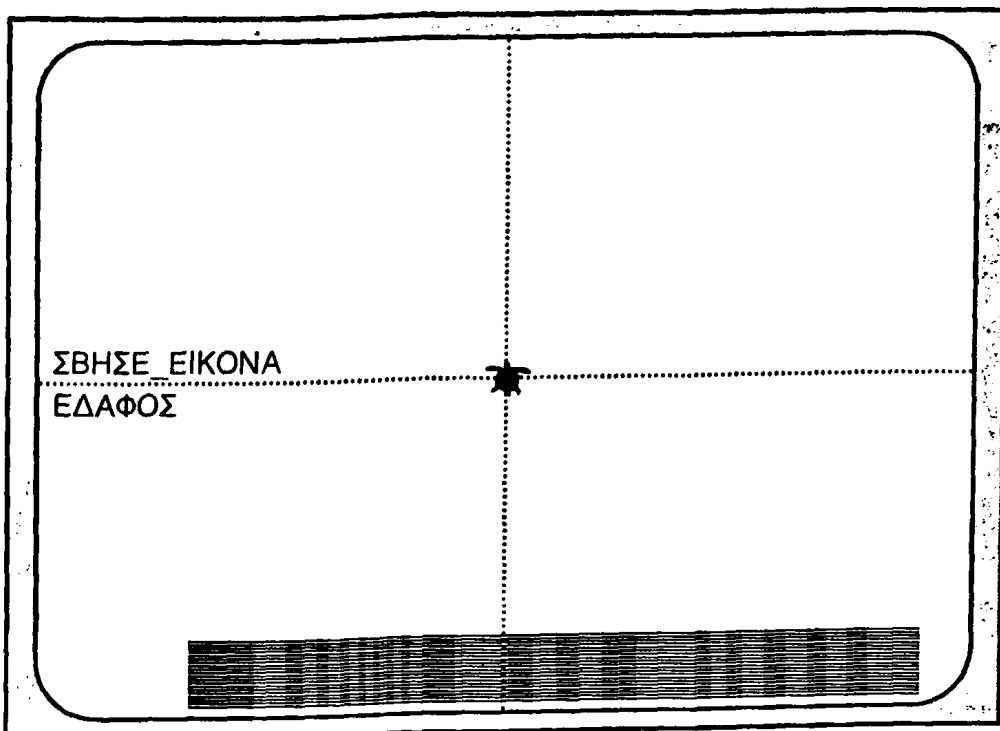
ΓΡΑΜΜΗ 90 300

MAKE "X MINUS 150

MAKE "Y :Y + 2

ΑΦΕΤΗΡΙΑ]

END



Τα εμπόδια

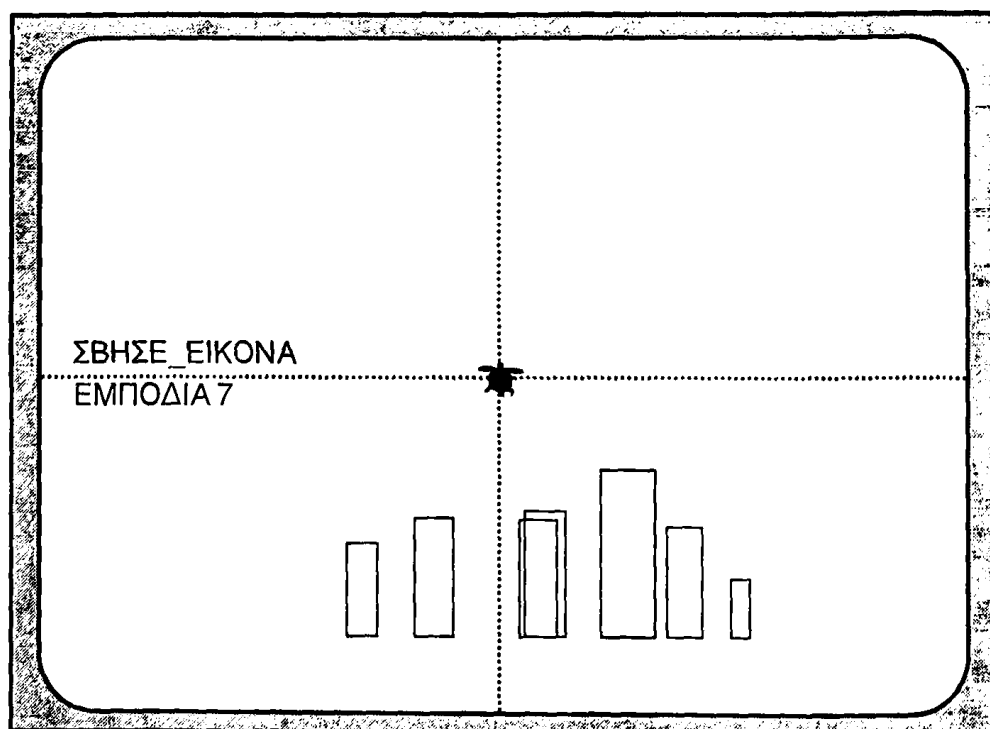
Στο παιχνίδι μεταξύ του κανονιού και του στόχου παρεμβάλλονται μερικά κτίρια που εμποδίζουν το κανόνι να βάλει σε ευθεία βολή κατά του στόχου.

Στο πρόγραμμά μας, προσομοιάζουμε τα κτίρια με ορθογώνια παραλληλόγραμμα. Η θέση των κτιρίων στο έδαφος βρίσκεται μεταξύ -125 και 124 και προκύπτει τυχαία. Ακόμα, το ύψος του κτιρίου κυμαίνεται από 0 μέχρι 99 βήματα χελώνας.

```
ΤΟ ΕΜΠΟΔΙΟ
ΜΗ_ΓΡΑΦΕΙΣ
ΜΑΚΕ "X (RANDOM 250) - 125
ΜΑΚΕ "Y MINUS 70
SETX :X
SETY :Y
ΓΡΑΨΕ
ΜΑΚΕ "ΥPSOS RANDOM 100
ΟΡΘΟΓΩΝΙΟ :ΥPSOS 3
ΑΦΕΤΗΡΙΑ
ΕΝΔ
```

Τα πολλά εμπόδια που χρειάζονται, δημιουργούνται από τη διαδικασία ΕΜΠΟΔΙΑ και το πλήθος τους καθορίζεται από την τιμή της μεταβλητής POSA.

```
ΤΟ ΕΜΠΟΔΙΑ :POSA
ΡΕΡΕΑΤ :POSA
  [ΕΜΠΟΔΙΟ]
ΕΝΔ
```



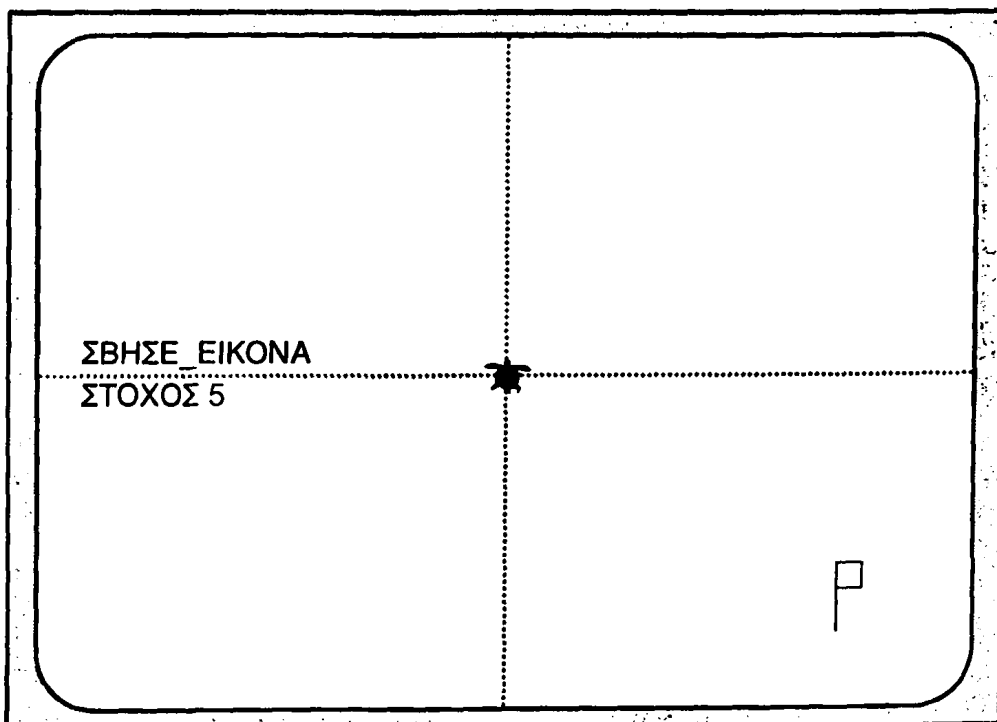
Ο στόχος

Ο στόχος εναντίον του οποίου θα βάλλει το κανόνι, είναι μια σημαία. Η θέση της σημαίας είναι επί του εδάφους, στο δεξιό άκρο της οθόνης στο σημείο (140, -70).

Η σημαία αποτελείται από τον ιστό και το πανί της. Ο ιστός θα σχεδιαστεί ως μια ευθεία γραμμή και το πανί ως ένα τετράγωνο.

Τέλος για να μη χάσει η χελώνα τον προσανατολισμό της, ώστε να μπορεί να χρησιμοποιηθεί από επόμενη διαδικασία, την επαναφέρουμε στο σημείο αφετηρίας της.

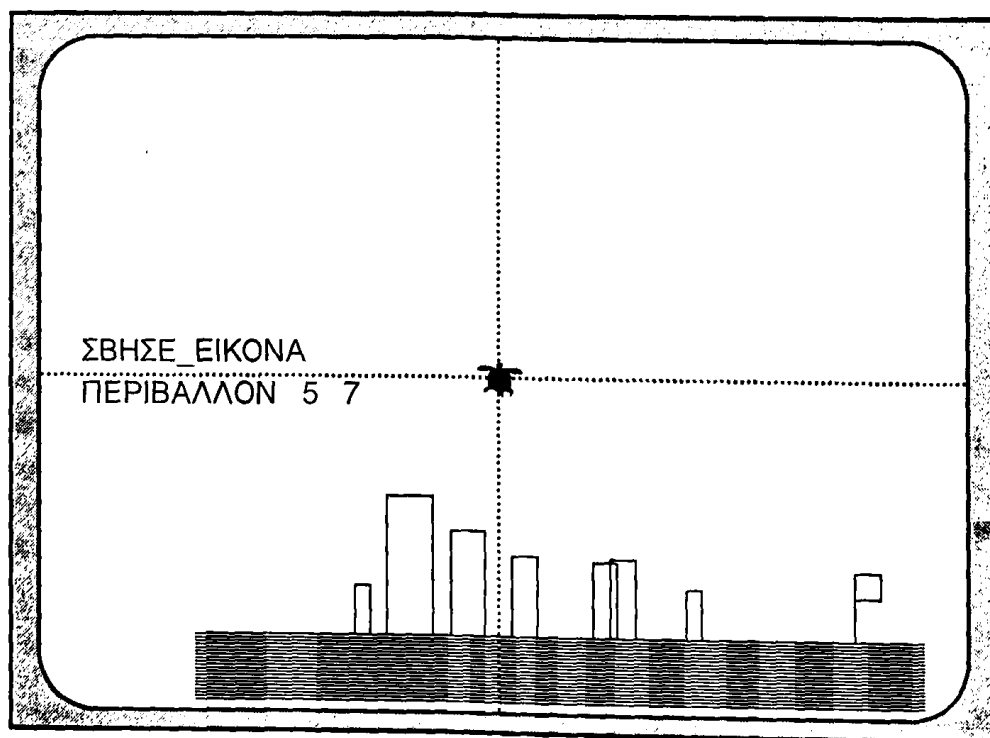
```
ΤΟ ΣΤΟΧΟΣ :MEGEQOS  
ΜΗ_ΓΡΑΦΕΙΣ  
MAKE "X 140  
MAKE "Y MINUS 70  
SETX :X  
SETY :Y  
ΓΡΑΨΕ  
ΕΥΘΕΙΑ :MEGEQOS * 5  
ΠΟΛΥΓΩΝΟ :MEGEQOS * 3 4  
ΑΦΕΤΗΡΙΑ  
END
```



Το περιβάλλον

Τα τμήματα του περιβάλλοντος στο οποίο θα πραγματοποιηθεί ο κανονιοβολισμός είναι πλέον έτοιμα. Απομένει να συνδέσουμε τις διαδικασίες που σχεδιάζουν το έδαφος, τα εμπόδια και το στόχο. Αυτό γίνεται με τη διαδικασία ΠΕΡΙΒΑΛΛΟΝ.

```
ΤΟ ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ :ΡΟΣΑ  
ΕΔΑΦΟΣ  
ΕΜΠΟΔΙΑ :ΡΟΣΑ  
ΣΤΟΧΟΣ :ΜΕΓΕΘΟΣ  
ΕΝΔ
```



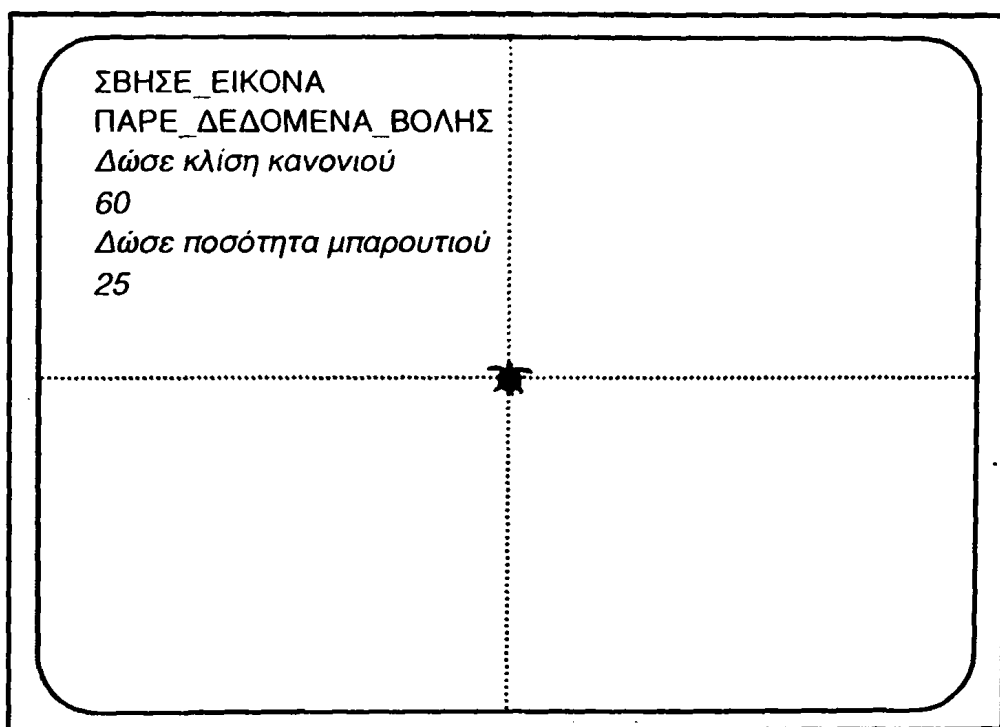
Οι ρυθμίσεις της βολής

Στο παιχνίδι δύο παράμετροι καθορίζουν την τροχιά του βλήματος. Η γωνία κλίσης της κάνης του πυροβόλου (συμβολίζεται ως ϕ ή F στο πρόγραμμα) και η ποσότητα του μπαρουτιού που υπάρχει στο κανόνι. Αυτές οι δύο παράμετροι θα ζητούνται από το χρήστη κατά την εκτέλεση του προγράμματος.

Για λόγους απλούστευσης του προβλήματος, ταυτίζουμε την ποσότητα του μπαρουτιού με την αρχική ταχύτητα του βλήματος, που συμβολίζεται ως u_0 ή U_0 στο πρόγραμμα.

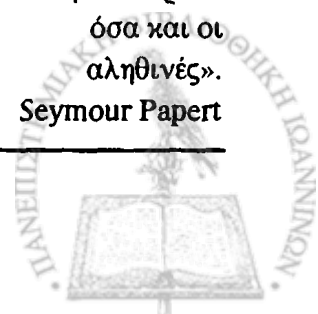
```
ΤΟ ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ
ΡΡ "Δώσε κλίση κανονιού και ENTER
ΠΑΡΕ_ΑΡΙΘΜΟ
ΜΑΚΕ "F :NUM
ΡΡ "Δώσε ποσότητα μπαρουτιού και ENTER
ΠΑΡΕ_ΑΡΙΘΜΟ
ΜΑΚΕ "U0 :NUM
ΕΝΔ
```

Μπορούμε να θέσουμε και άλλες παραμέτρους από τις οποίες να εξαρτάται η βολή. Τη φορά και την ένταση του πνέοντος ανέμου, από το αν η βολή εκτελείται στον Ισημερινό ή στους πόλους, στην επιφάνεια της θάλασσας ή σε βουνό, στη Σελήνη ή στον Άρη (μεταβάλλεται η τιμή της επιτάχυνσης της βαρύτητας που συμβολίζεται με g ή G στο πρόγραμμα).



«Όλοι μαθαίνουμε κατασκευάζοντας, εξερευνώντας και κτίζοντας θεωρίες, αλλά το μεγαλύτερο μέρος του κτισίματος που μας ταλαιπώρησε κατέληξε σε θεωρίες τις οποίες θα χρειαστεί αργότερα να παρατήσουμε... Τα παιδιά δεν ακολουθούν ένα μονοπάτι μάθησης που οδηγείται από μια "αληθινή θέση" σε άλλη, πιο προχωρημένη "αληθινή θέση". Τα φυσικά τους μονοπάτια μάθησης περιέχουν "ψεύτικες θεωρίες" που διδάσκουν σχετικά με το κτίσιμο θεωριών όσα και οι αληθινές».

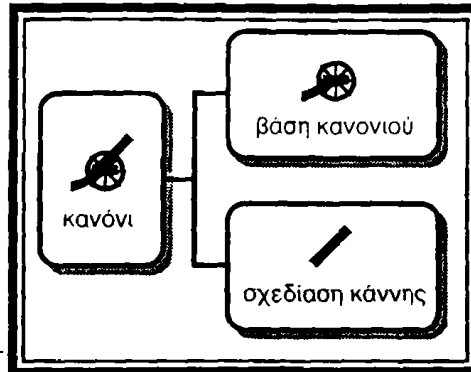
Seymour Papert



Η βάση και η κάννη του κανονιού

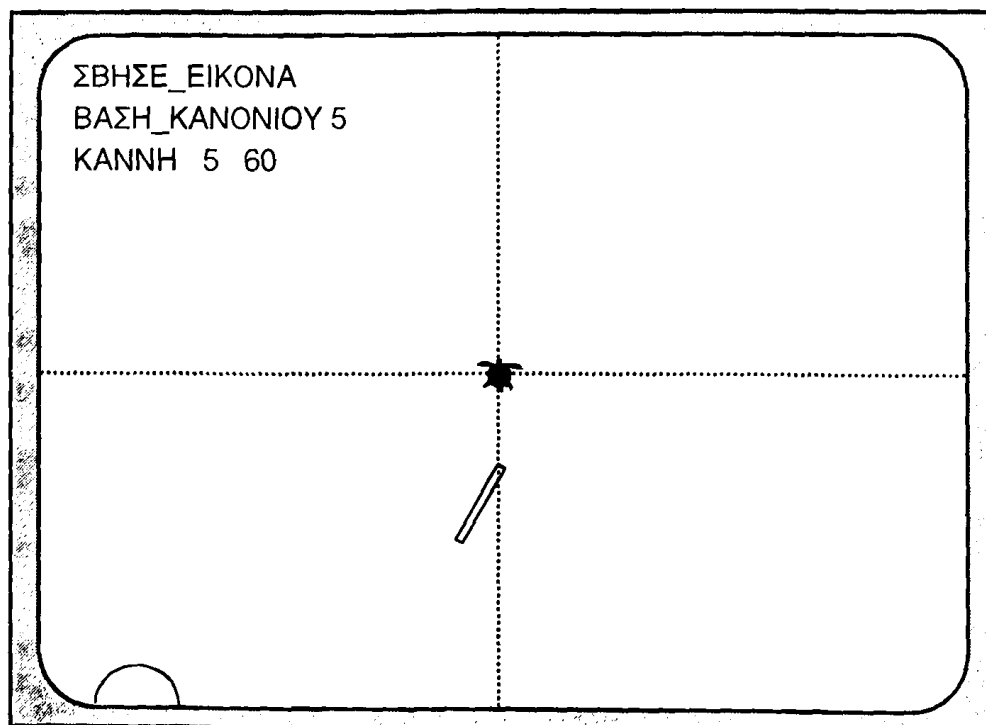
Το κανόνι αποτελείται από δύο μέρη, τη βάση και την κάννη. Είναι απαραίτητο να σχεδιασθούν αυτά τα μέρη ξεχωριστά, γιατί ενώ η βάση παραμένει ακίνητη, η κάννη πρέπει να στρέφεται ανάλογα με την κλίση. Η βάση θα σχεδιασθεί ως ένα ημικύκλιο.

```
ΤΟ ΒΑΣΗ_ΚΑΝΟΝΙΟΥ :MG
ΜΗ_ΓΡΑΦΕΙΣ
ΜΑΚΕ "X MINUS 150
ΜΑΚΕ "Y MINUS 80
SETX :X
SETY :Y
ΓΡΑΨΕ
ΔΕΞΙΑ 90
ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ :MG 32 1 / 2
ΑΦΕΤΗΡΙΑ
ΕΝΔ
```



ενώ η κάννη ως ένα ορθογώνιο παραλληλόγραμμο. Η στροφή της κάννης θα γίνεται ως προς το άνω δεξιό άκρο που θα παραμένει ακίνητο και από το οποίο θα αρχίζει η κίνηση του βλήματος.

```
ΤΟ ΚΑΝΗ :MEGEQOS :KLISH
ΑΡΙΣΤΕΡΑ :KLISH + 180
ΜΑΚΕ "LOGOS 1 / 10
ΟΡΘΟΓΩΝΙΟ :MEGEQOS :LOGOS
ΔΕΞΙΑ :KLISH + 180
ΕΝΔ
```



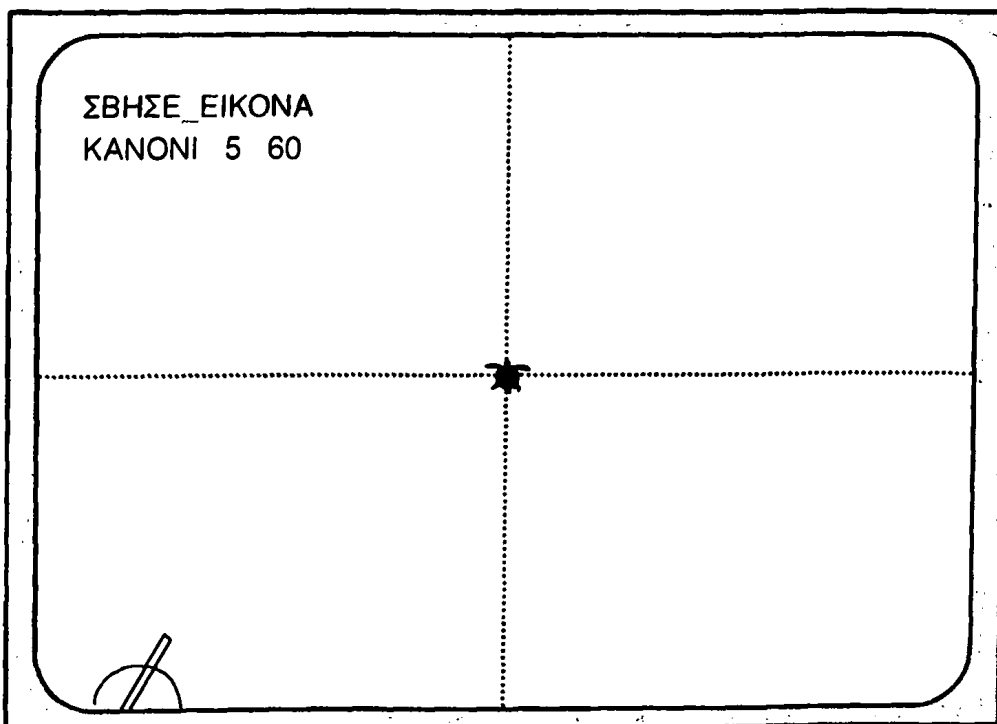
Το κανόνι

Η κάννη πρέπει να τοποθετηθεί στο σωστό σημείο και όχι στο κέντρο της οθόνης. Αυτό γίνεται με τη διαδικασία ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ που μεταφέρει τη χελώνα στο σημείο (-120, -60) που είναι η θέση το πυροβόλου.

```
ΤΟ ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ
ΜΗ_ΓΡΑΦΕΙΣ
ΜΑΚΕ "X MINUS 120
ΜΑΚΕ "Y MINUS 60
SETX :X
SETY :Y
ΓΡΑΨΕ
ΕΝΔ
```

Η διαδικασία που σχεδιάζει το κανόνι, δεν έχει παρά να συνδέσει τις διαδικασίες ΒΑΣΗ_ΚΑΝΟΝΙΟΥ, ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ, ΚΑΝΝΗ και ΑΦΕΤΗΡΙΑ.

```
ΤΟ ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :ΚΛΙΣΗ
ΒΑΣΗ_ΚΑΝΟΝΙΟΥ :ΜΕΓΕΘΟΣ
ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ
ΚΑΝΝΗ :ΜΕΓΕΘΟΣ :ΚΛΙΣΗ
ΑΦΕΤΗΡΙΑ
ΕΝΔ
```



Υπολογισμοί της τροχιάς του βλήματος

Η βολή προϋποθέτει κάποιους υπολογισμούς, τη σχεδίαση του βλήματος και τη διαδικασία του "animation" που θα σχεδιάσει την τροχιά του βλήματος.

Ο καθορισμός της τροχιάς που θα ακολουθήσει το βλήμα, γίνεται με τη βοήθεια της φυσικής. Δε θα μας απασχολήσει ο τρόπος που θα καταλήξουμε στους απαραίτητους τύπους. Απλά θα δανειστούμε από τη φυσική τους εξής τύπους που σχετίζονται με την πλάγια βολή:

$x = u_0 \cos \varphi t$ x , είναι η οριζόντια απόσταση του βλήματος από την αρχή κάποια χρονική στιγμή t .

$y = x \tan \varphi - \frac{g t^2}{2 \cos^2 \varphi}$ y , είναι η κατακόρυφη απόσταση του βλήματος από την αρχή τη χρονική στιγμή t .

$t_{ολ} = \frac{2 u_0}{g} \eta \mu \varphi$ $t_{ολ}$, ο χρόνος που διαρκεί η βολή.

Κάποια τμήματα αυτών των υπολογισμών είναι ανεξάρτητα του χρόνου και έτσι μας συμφέρει (από άποψη ταχύτητας εκτέλεσης του προγράμματος) να υπολογιστούν στην αρχή. Αυτά είναι τα εξής:

$x_{χωρίς_t} = u_0 \cos \varphi$ $\text{παρονομαστής} = 2 u_0 \cos \varphi$

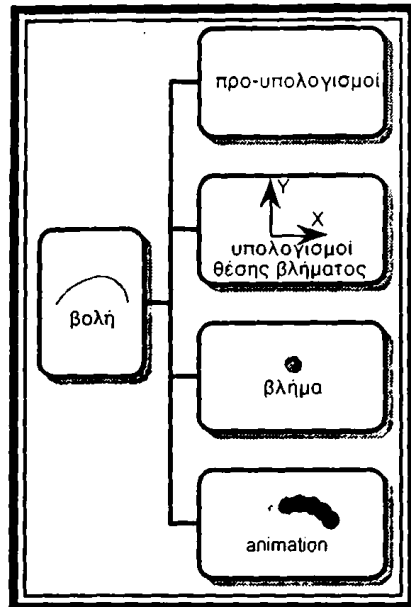
καθώς και ο υπολογισμός της συνάρτησης της εφφ .

Επίσης θα πρέπει να δοθεί η σταθερή τιμή στην επιτάχυνση της βαρύτητας g , καθώς και οι συντεταγμένες $X_0 = -195, Y_0 = -100$ από όπου θα ξεκινάει το βλήμα.

Έτσι έχουμε τη διαδικασία ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ.

```

ΤΟ ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ :F :U0
MAKE "G 9.81
MAKE "X0 MINUS 195
MAKE "Y0 MINUS 100
MAKE "U0 :U0 * 4
MAKE "TOL (2 * :U0 / :G) * SIN(:F)
MAKE "T 0
MAKE "X_XWRIS_T (:U0 * COS(:F))
MAKE "EFF (SIN(:F) / COS(:F))
MAKE "PAR (2 * :U0 * COS(:F))
END
    
```



«... μαθητές οι οποίοι προσπαθούν να αναπτύξουν νευτώνεια σκέψη σχετικά με την κίνηση αντιμετωπίζουν προβλήματα... Πρέπει πρώτα να μάθουν πως να δουλεύουν με τις εξισώσεις πριν τις χρησιμοποιήσουν για να σχεδιάσουν ένα νευτώνειο κόσμο... Σε ένα προσομοιωμένο κόσμο, όπου οι μαθητές θα έχουν άμεση πρόσβαση στη νευτώνεια κίνηση δε χρειάζεται να κάνει καλά τις εξισώσεις. Ακριβώς το αντίθετο: αντί να κάνουμε τους μαθητές να περιμένουν τις εξισώσεις, μπορούμε να προκαλέσουμε και να διευκολύνουμε την απόκτηση εξισωτικών δεξιοτήτων παρέχοντας ένα διαισθητικά πολύ καλά κατανοητό πλαίσιο για τη χρήση τους».
Seymour Papert

Η θέση του βλήματος κάποια στιγμή

Με βάση τους αρχικούς υπολογισμούς, οι εξισώσεις που καθορίζουν τις συντεταγμένες X, Y κάποια χρονική στιγμή είναι:

$$x = x_{\text{χωρίς}_t} \cdot t$$

$$y = x \text{ εφφ} - \frac{g t}{\text{παρονομαστής}} x$$

Έτσι, υπολογίζεται η θέση του βλήματος με τη διαδικασία:

```
TO ΘΕΣΗ_ΒΛΗΜΑΤΟΣ
MAKE "X :X_XWRIS_T* :T
MAKE "Y (:X* :EFF) - (:G* :T* :X / :PAR)
END
```

Το βλήμα και η κίνησή του, περιγράφονται από τις ακόλουθες διαδικασίες:

```
TO ΒΛΗΜΑ :MEGEQOS
ΠΟΛΥΓΩΝΟ :MEGEQOS / 2 6
END
```

```
TO ΚΙΝΗΣΗ_ΒΛΗΜΑΤΟΣ :MEGEQOS
ΜΗ_ΓΡΑΦΕΙΣ
SETX (:X0 + :X / 10)
SETY (:Y0 + :Y / 10)
ΓΡΑΨΕ
ΒΛΗΜΑ :MEGEQOS
ΣΒΗΣΕ
ΒΛΗΜΑ :MEGEQOS
END
```

Για λόγους που έχουν σχέση με την ταχύτητα εκτέλεσης του προγράμματος το σχήμα του βλήματος δεν έγινε κυκλικό (36-πλευρο) αλλά κανονικό εξάγωνο.

Για να ταιριάζει με το μέγεθος της οθόνης, το βήμα X και Y διαιρείται δια 10.

```
ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ 60 50
MAKE "T 20
ΘΕΣΗ_ΒΛΗΜΑΤΟΣ
ΚΙΝΗΣΗ_ΒΛΗΜΑΤΟΣ 5
ΚΙΝΗΣΗ_ΒΛΗΜΑΤΟΣ 5
```



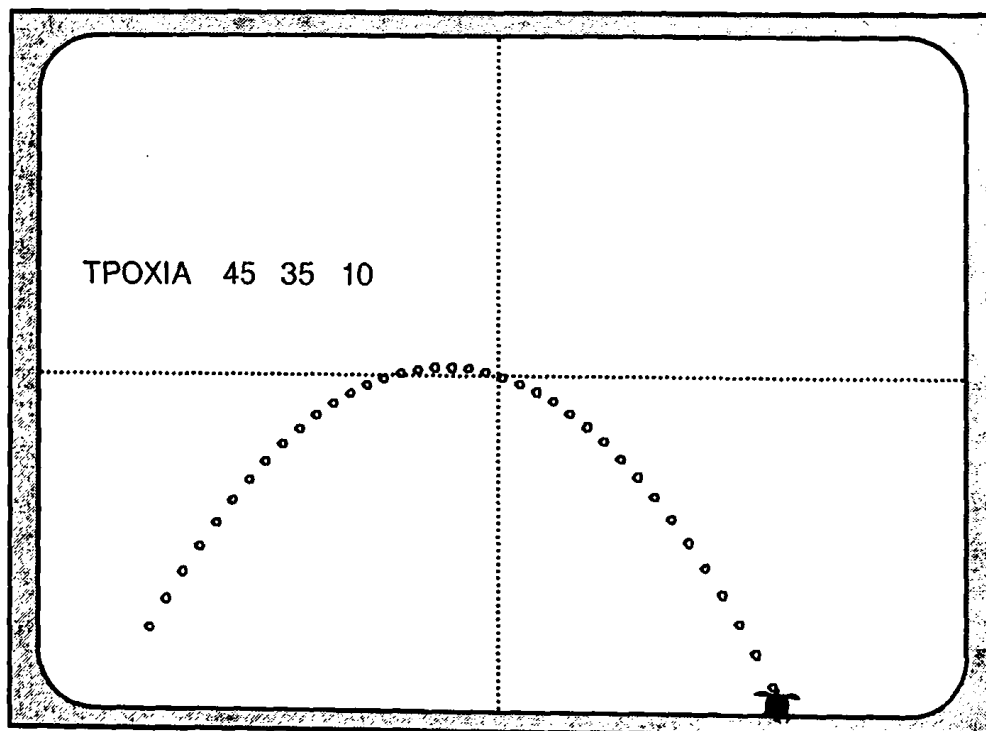
Η τροχιά του βλήματος

Η χάραξη της τροχιάς του βλήματος είναι πλέον απλή υπόθεση. Η διαδικασία που θα την υλοποιεί αφού κάνει τους απαραίτητους αρχικούς υπολογισμούς, θα επαναλάβει τη διαδικασία ANIMATION_BΛΗΜΑΤΟΣ τόσες φορές όση είναι η τιμή του χρόνου t_{ol} που διαρκεί η βολή.

Σε κάθε επανάληψη θα πρέπει να αυξάνει η τιμή της μεταβλητής T που εκφράζει το χρόνο που ρέει και με βάση την οποία γίνονται κάθε φορά οι υπολογισμοί που καθορίζουν τις συντεταγμένες X, Y του βλήματος.

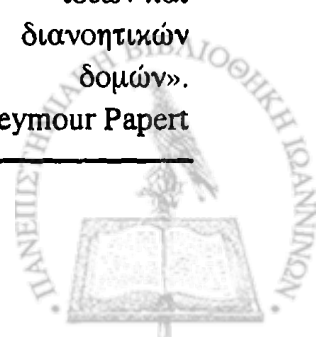
```

ΤΟ ΤΡΟΧΙΑ :F :U0 :MEGEQOS
ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ :F :U0
ΡΕΡΕΑΤ :TOL
  [ΘΕΣΗ_ΒΛΗΜΑΤΟΣ
  ΚΙΝΗΣΗ_ΒΛΗΜΑΤΟΣ :MEGEQOS
  MAKE "T :T + 1
  ΓΡΑΨΕ ]
END
    
```



«Οι μαθητές δεν μπορούν να κάνουν κτήμα τους κάτι που δεν το γνωρίζουν. Γι' αυτό είναι προαπαιτούμενο να βρούμε τρόπους να διευκολύνουμε την προσωπική ιδιοποίηση όχι μόνο της νευτώνειας κίνησης και των νόμων που την περιγράφουν, αλλά επίσης της γενικής έννοιας των νόμων. Αυτό το κάνουν σχεδιάζοντας μια σειρά από μικρόκοσμους. Ο σχεδιασμός του μικρόκοσμου τον κάνει "φυτώριο" ενός συγκεκριμένου είδους δυναμικών ιδεών και διανοητικών δομών».

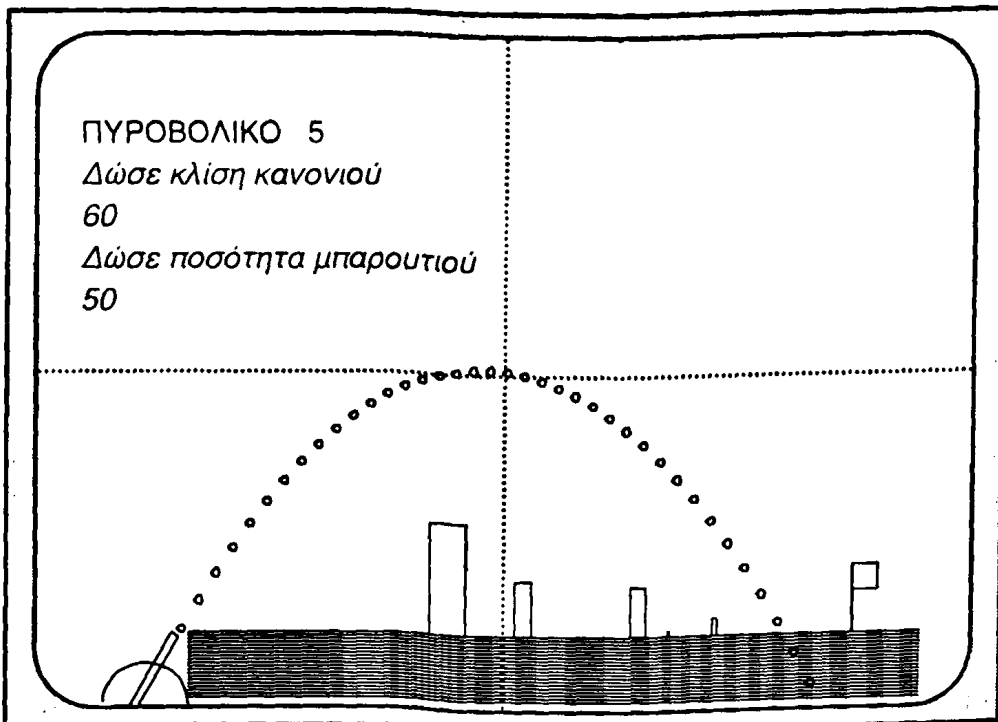
Seymour Papert



Το κανόνι βάλλει κατά του στόχου

Ας συνοψίσουμε σε μια διαδικασία, την ΠΥΡΟΒΟΛΙΚΟ, τη σχεδίαση του περιβάλλοντος, την εισαγωγή των δεδομένων της ρύθμισης της βολής, τη σωστή τοποθέτηση του κανονιού και τέλος τη χάραξη της τροχιάς του βλήματος.

```
ΤΟ ΠΥΡΟΒΟΛΙΚΟ :MEGEQOS
ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΚΡΥΨΟΥ
ΠΕΡΙΒΑΛΛΟΝ :MEGEQOS 5
ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ
ΚΑΝΟΝΙ :MEGEQOS :F
ΤΡΟΧΙΑ :F :U0 :MEGEQOS
ΒΓΕΣ
END
```



«Σ' ένα καλό μάθημα τέχνης, το παιδί αποκτά τεχνική γνώση ως μέσο για να φτάσει στο δημιουργικό και προσωπικά καθορισμένο τέλος. Θα υπάρξει προϊόν. Ο δάσκαλος όπως και το παιδί μπορεί να ενθουσιαστεί γνήσια απ' αυτό. Στο περιβάλλον της logo αυτό συμβαίνει συχνά. Ο γνήσιος ενθουσιασμός του δασκάλου για το προϊόν μεταδίδεται στα παιδιά τα οποία γνωρίζουν ότι κάνουν κάτι σημαντικό. Διαφορετικά από το μάθημα αριθμητικής, όπου γνωρίζουν ότι οι αθροίσεις που κάνουν είναι μόνον ασκήσεις, εδώ αντιλαμβάνονται σοβαρά τη δουλειά τους».

Seymour Papert

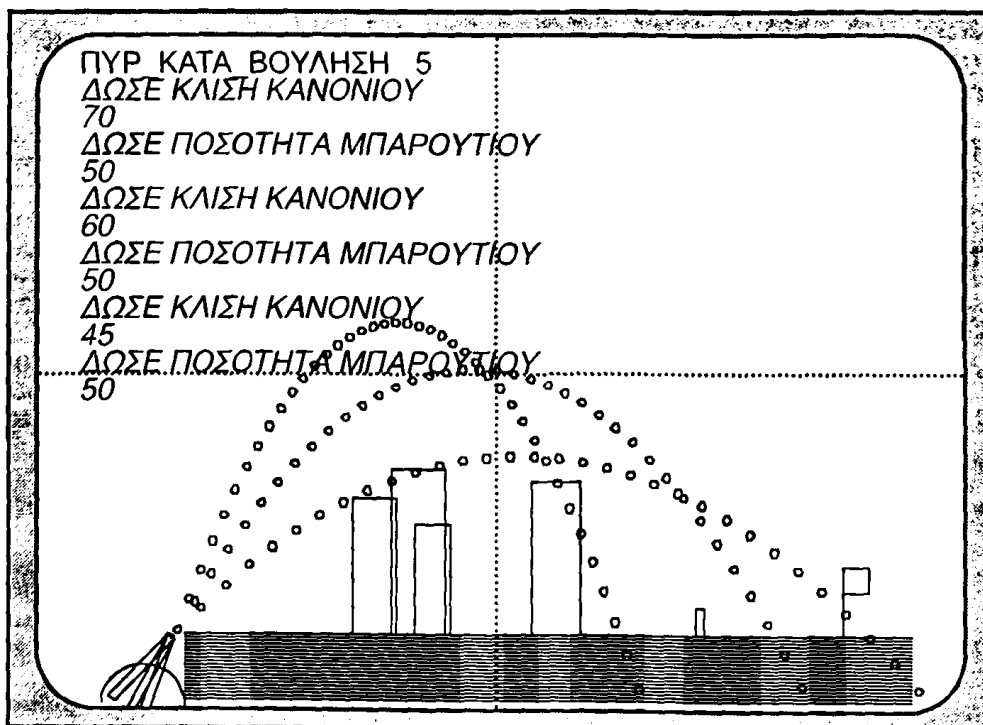
Δικαίωμα για επαναληπτικές βολές

Μπορούμε να επαναλαμβάνουμε τη βολή όσες φορές θέλουμε αλλά κάθε φορά θα αλλάζει και το τοπίο με τα εμπόδια. Για να αποφύγουμε αυτή τη συνεχή αλλαγή, θα βάλουμε μέσα σε μια εντολή repeat το τμήμα της διαδικασίας ΠΥΡΟΒΟΛΙΚΟ που υπάρχει μετά τη γραμμή που σχεδιάζει το περιβάλλον.

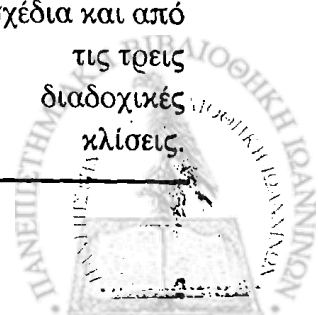
Καθορίζουμε σε 3 το δικαίωμα των βολών που δικαιούται κάποιος για να πετύχει το στόχο.

Έτσι έχουμε τη διαδικασία ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ.

```
TO ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ :ΜΕΓΕΘΟΣ
ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΚΡΥΨΟΥ
ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ 5
REPEAT 3
  [ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ
  ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :F
  ΤΡΟΧΙΑ :F :U0 :ΜΕΓΕΘΟΣ
  MAKE "Q READCHAR ]
ΒΓΕΣ
END
```



Παρατηρούμε ότι σε κάθε βολή, η κάνη του κανονιού στρέφεται για να πάρει τη νέα γωνία. Στην οθόνη παραμένουν τα σχέδια και από τις τρεις διαδοχικές κλίσεις.



Κάλυψη και απόκρυψη του κανονιού

Διαπιστώνουμε την ανάγκη να σβήνει η κάννη του κανονιού μετά από κάθε βολή. Έτσι φτιάχνεται η διαδικασία

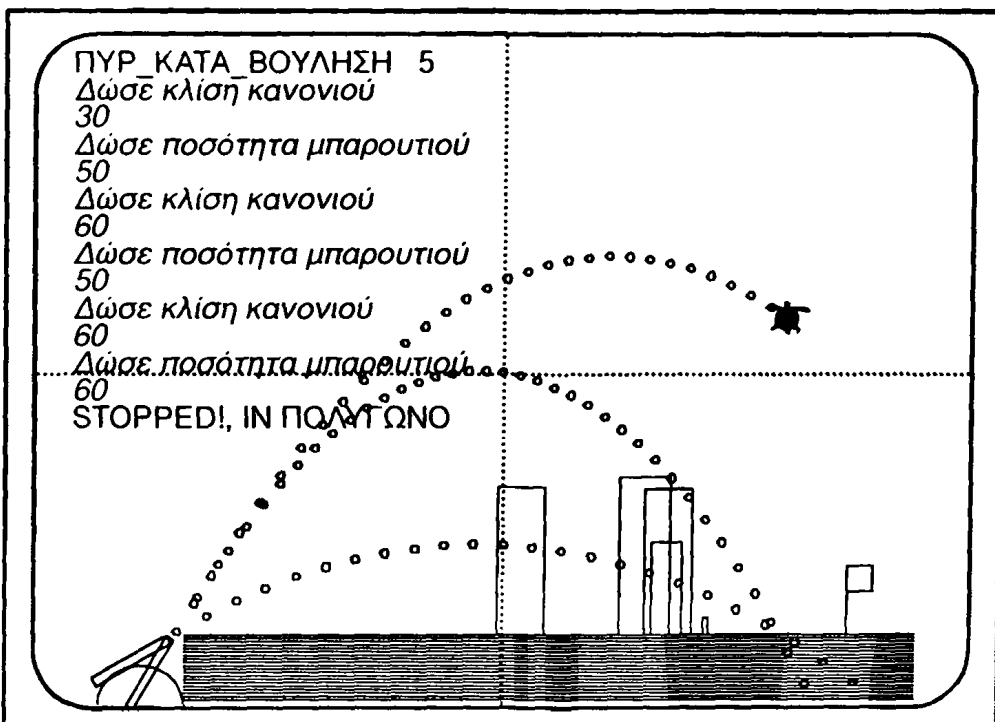
```

ΤΟ ΣΒΗΣΕ_ΚΑΝΟΝΙ :MEGEQOS :F
ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ
ΑΡΙΣΤΕΡΑ :F + 180
MAKE "LOGOS 1 / 10
ΣΒΗΣΕ
ΟΡΘΟΓΩΝΙΟ :MEGEQOS :LOGOS
ΓΡΑΨΕ
ΔΕΞΙΑ :F + 180
ΑΦΕΤΗΡΙΑ
END
    
```

Η διαδικασία ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ τροποποιείται:

```

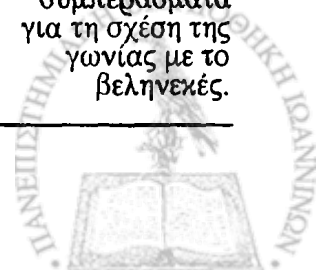
ΤΟ ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ :MEGEQOS
ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΚΡΥΨΟΥ
ΠΕΡΙΒΑΛΛΟΝ :MEGEQOS 5
ΡΕΡΕΑΤ 3
[ ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ
ΚΑΝΟΝΙ :MEGEQOS :F
ΤΡΟΧΙΑ :F :UO :MEGEQOS
MAKE "Q READCHAR
ΣΒΗΣΕ_ΚΑΝΟΝΙ :MEGEQOS :F ]
ΒΓΕΣ
END
    
```



«Ο Piaget έδειξε ότι τα παιδιά μαθαίνουν θεμελιώδεις μαθηματικές ιδέες, κτίζοντας πρώτα τα δικά τους, πολύ διαφορετικά μαθηματικά. Και τα παιδιά μαθαίνουν μια γλώσσα μαθαίνοντας πρώτα τις δικές τους ("παιδιάστικες") διαλέκτους. Έτσι, όταν σκεπτόμαστε τους μικρόκοσμους ως εκκολαπτήρια δυναμικών ιδεών, προσπαθούμε να σχεδιάσουμε πάνω σ' αυτή την αποτελεσματική στρατηγική: επιτρέπουμε στους μαθητές να μάθουν την "επίσημη" φυσική, δίνοντάς τους την ελευθερία να εφεύρουν πολλά που μπορούν να λειτουργήσουν σε ισάριθμους κόσμους που θα επινοήσουν».

Seymour Papert

Πειραματιζόμενοι με διάφορες γωνίες και κρατώντας σταθερό το μπαρούτι μπορούμε να βγάλουμε χρήσιμα συμπεράσματα για τη σχέση της γωνίας με το βεληνεκές.



Αναφορές

- Agons A. B., "Οδηγός Διδασκαλίας της Φυσικής", Τροχαλία 1992
- Bense M., "Συνοπτική θεμελίωση της σύγχρονης αισθητικής", Δευκαλίων 2/1970
- Boris A., "LOGO για αρχαρίους", Κλειδάριθμος
- Briggs J. & Peat F. D., "Ο ταραγμένος καθρέπτης", Κάτοπτρο 1991
- Gleick J., "Χάος. Μια νέα επιστήμη", Κάτοπτρο 1990
- Papert S., "Νοητικές Θύελλες. Παιδιά, Ηλεκτρονικοί Υπολογιστές και Δυναμικές Ιδέες. Τα πάντα γύρω από τη Logo", Οδυσσέας 1991



ΠΑΡΑΡΤΗΜΑ Α

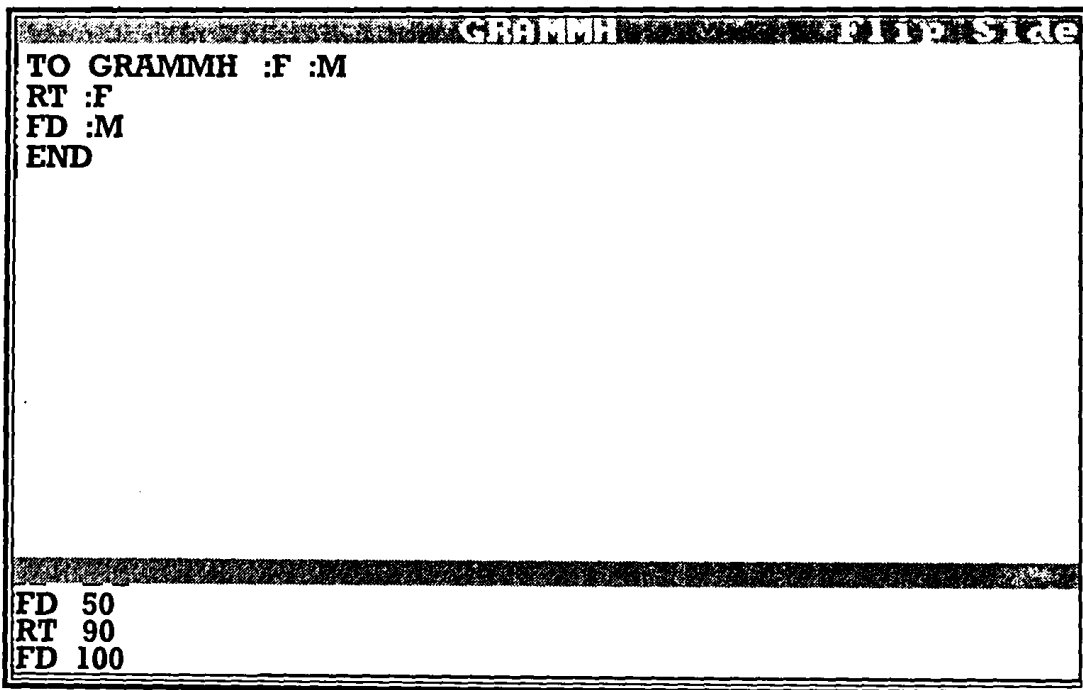
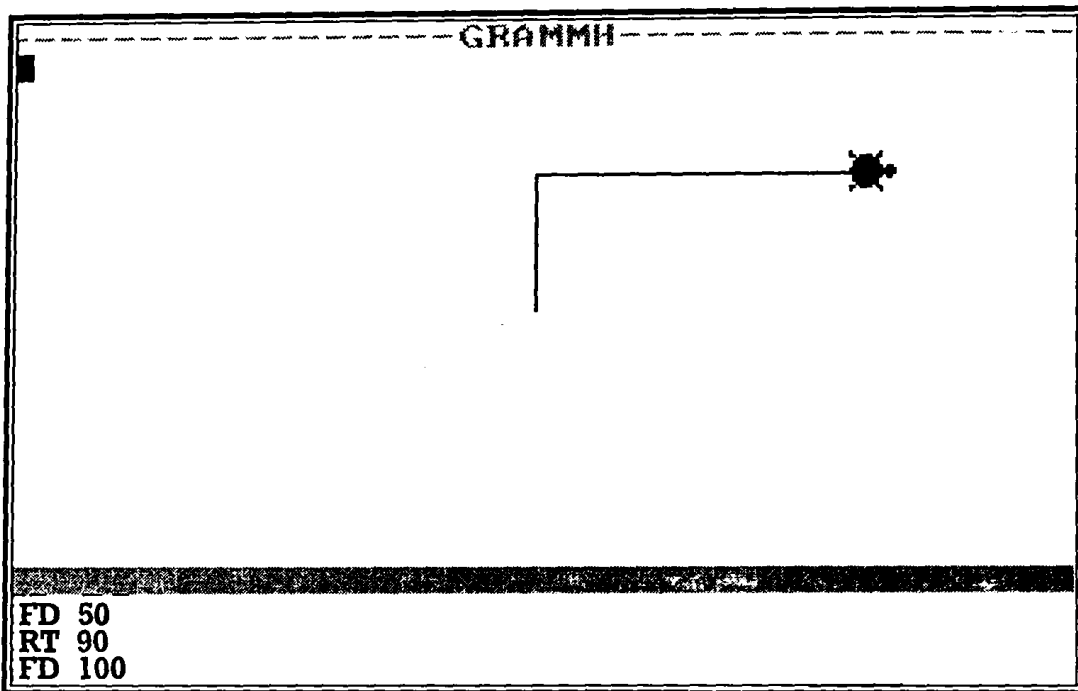
A.1 Η LogoWriter 2.0 της LCSΙ

Η logo στην οποία είναι γραμμένες οι διαδικασίες του βιβλίου, είναι η LogoWriter 2.0 της LCSΙ. Είναι λογισμικό για το λειτουργικό σύστημα DOS, με μικρές απαιτήσεις υπολογιστικής ισχύος. Παρά το λιτό για τα σημερινά δεδομένα περιβάλλον εργασίας της, προσφέρει στο χρήστη τα περιθώρια να αφοσιωθεί στο έργο του και είναι συμβατή με τις επόμενες εξελιγμένες εκδόσεις για περιβάλλοντα windows και macintosh.

Ο τρόπος αλληλεπίδρασης του χρήστη με τη LogoWriter χρησιμοποιεί τη μεταφορά του βιβλίου ή ενός τετραδίου. Κάθε φύλλο του (page) περιέχει ένα πρόγραμμα - διαδικασία, και αποτελεί ένα αρχείο με επέκταση LWR. Ομάδες από αρχεία μπορεί να αποτελούν κεφάλαια που υλοποιούνται από τους καταλόγους (Directories) του DOS. Η πρώτη σελίδα του τετραδίου είναι αυτή των περιεχομένων (Contents), που εμφανίζει με αλφαβητική σειρά τις διαδικασίες και τα κεφάλαια του τετραδίου. Η επιλογή κάποιας διαδικασίας γίνεται με τα βέλη του πληκτρολογίου και η μετάβαση σ' αυτήν με το πλήκτρο "enter".

Το κάθε φύλλο έχει δύο πλευρές, τις σελίδες του. Στη μπροστά πλευρά (front side) κυριαρχεί ο χώρος για τα γραφικά, ένα καρτεσιανό σύστημα συντεταγμένων με τη χελώνα στο κέντρο του. Ένα μικρό μέρος της οθόνης διατίθεται για την άμεση εκτέλεση των εντολών και των διαδικασιών. Στην πίσω πλευρά του φύλλου (flip side), είναι ο χώρος για τον ορισμό των προγραμμάτων - διαδικασιών. Η εναλλαγή από τη μια στην άλλη πλευρά του φύλλου γίνεται με το ταυτόχρονο πάτημα των πλήκτρων "CONTROL - F". Ένας τρόπος για την αποθήκευση των πλέον πρόσφατων αλλαγών της εργασίας σε κάθε φύλλο, είναι με το πλήκτρο "escape". Το σχήμα δείχνει το περιβάλλον εργασίας της LCSΙ logo





Το περιβάλλον εργασίας της logo



A.2 Πρωτογενείς διαδικασίες - εντολές της LogoWriter

AND αληθής/ψευδής1 αληθής/ψευδής2

Επιστρέφει αληθή τιμή αν όλες οι εισοδοι της είναι αληθείς

ARCTAN αριθμός

Επιστρέφει το τόξο εφαπτομένης του αριθμού

BACK αριθμός ή BK αριθμός

Μετακινεί τη χελώνα προς τα πίσω σε απόσταση ίση με τον αριθμό (των βημάτων)

CC

Σβήνει τις εντολές από το χώρο άμεσης εκτέλεσης

CG

Σβήνει τα γραφικά και επιστρέφει τη χελώνα στο κέντρο των αξόνων, με κατεύθυνση προς τα πάνω

CONTENTS

Αποθηκεύει την τρέχουσα εργασία (σελίδα) και εμφανίζει τη σελίδα περιεχομένων

COS μοίρες

Επιστρέφει το συνημίτονο του αριθμού εισόδου σε μοίρες

CT

Σβήνει το κείμενο της τρέχουσας σελίδας



DOS

Αποθηκεύει την εργασία, σταματά τη λειτουργία της Ιογο και επιστρέφει στο λειτουργικό σύστημα

EMPTY? λέξη/λίστα

Αναφέρει αληθές αν η είσοδος της είναι μια άδεια λέξη ή λίστα

END

Δηλώνει το τέλος μιας διαδικασίας στην πίσω πλευρά ενός φύλλου. Διαχωρίζει διαδικασίες στην πίσω πλευρά της ίδιας σελίδας

FIRST λέξη/λίστα

Επιστρέφει το πρώτο στοιχείο μιας λέξης ή λίστας

FORWARD αριθμός ή FD αριθμός

Μετακινεί τη χελώνα εμπρός σε απόσταση ίση με τον αριθμό των βημάτων

FPUT λέξη/λίστα λίστα (First PUT)

Δημιουργεί μια λίστα με πρώτο στοιχείο τη λέξη ή τη λίστα

GETTOOLS όνομα φύλλου

Φορτώνει στη μνήμη τις διαδικασίες που βρίσκονται στο αρχείο με το όνομα φύλλου. Χρησιμοποιούνται ως εργαλεία - βοηθητικές διαδικασίες και λειτουργούν όπως οι εντολές. Οι διαδικασίες - εργαλεία δεν εμφανίζονται στην πίσω πλευρά της τρέχουσας σελίδας και είναι διαθέσιμες μέχρι την παύση της Ιογο ή την τροποποίησή τους.

HOME

Μετακινεί τη χελώνα στην αρχική της θέση με κατεύθυνση προς τα πάνω



HT

Κρύβει τη χελώνα. Τα γραφικά σχεδιάζονται ταχύτερα.

IF συνθήκη [διαδικασίες ή εντολές]

Αν ισχύει η συνθήκη, εκτελείται ό,τι υπάρχει στις αγκύλες, αλλιώς αγνοείται και συνεχίζει παρακάτω

IFELSE συνθήκη [διαδικασίες ή εντολές] [διαδικασίες ή εντολές]

Αν ισχύει η συνθήκη, εκτελείται ό,τι υπάρχει στις πρώτες αγκύλες.

Αν δεν ισχύει, εκτελείται ό,τι υπάρχει στις δεύτερες.

LAST λέξη/λίστα

Επιστρέφει το τελευταίο στοιχείο μιας λέξης ή λίστας

LEAVEPAGE

Επιστροφή στη σελίδα των περιεχομένων, χωρίς αποθήκευση της τρέχουσας εργασίας

LEFT μοίρες ή LT μοίρες

Στρίβει τη χελώνα αριστερά γωνία ίση με μοίρες

MAKE “μεταβλητή τιμή/λίστα τιμή

Δημιουργεί μια μεταβλητή και της δίνει την τιμή “τιμή/λίστα”. Οι μεταβλητές που δημιουργούνται με τη make είναι καθολικές και παραμένουν στη μνήμη και μετά την αλλαγή της σελίδας. Είναι τοπικές όταν η make ορίζει μεταβλητή που εμφανίζεται στον τίτλο της διαδικασίας

MEMBER? λέξη/λίστα1 λέξη/λίστα2

Αναφέρει αληθές αν η λέξη/λίστα1 περιέχεται στη λέξη/λίστα2



MINUS αριθμός

Αναφέρεται σε αρνητικό αριθμό (μέσα σε διαδικασία)

NAMEPAGE όνομα φύλλου ή NP όνομα φύλλου

Οναμάζει και αποθηκεύει την τρέχουσα εργασία/φύλλο

NOT αληθής/ψευδής

Επιστρέφει το λογικό αντίστροφο της εισόδου

PD

Κατεβάζει την πένα της χελώνας. Εμφανίζεται το ίχνος της χελώνας κατά την κίνησή της

PE

Κατεβάζει τη σβήστρα της χελώνας. Σβήνεται κάθε γραμμή που συναντά η χελώνα

PRINT ή PR αριθμός/λέξη/μεταβλητή/λίστα

Τυπώνει το αριθμός/λέξη/μεταβλητή/λίστα και ακολουθεί αλλαγή γραμμής

PU

Ανεβάζει την πένα της χελώνας. Δεν εμφανίζεται το ίχνος της χελώνας κατά την κίνησή της

RANDOM αριθμός

Επιστρέφει έναν τυχαίο αριθμό από το 0 μέχρι τον αριθμό εισόδου μείον 1

READCHAR

Διαβάζει ένα χαρακτήρα που πληκτρολογείται



READLIST ή RL

Διαβάζει κείμενο που πληκτρολογείται (μέχρι να πατηθεί "enter") και το επιστρέφει ως λίστα

REPEAT αριθμός [λίστα εντολών]

Επαναλαμβάνει τη λίστα εντολών όσες φορές καθορίζει ο αριθμός

RG

Σβήνει τα γραφικά και αποκαθιστά χρώματα και χελώνες στην αρχική τους θέση (εμφανίζεται μόνο η χελώνα 0).

RIGHT μοίρες ή RT μοίρες

Στρίβει τη χελώνα δεξιά κατά γωνία ίση με μοίρες

RUN λίστα εντολών

Εκτελεί τη λίστα εντολών

SETSH αριθμός

Θέτει το σχήμα της χελώνας που καθορίζεται από τον αριθμό

SETX αριθμός

Θέτει την οριζόντια (x) συντεταγμένη της χελώνας στη θέση που δηλώνει ο αριθμός

SETY αριθμός

Θέτει την κατακόρυφη (y) συντεταγμένη της χελώνας στη θέση που δηλώνει ο αριθμός

SHOW λέξη/λίστα

Τυπώνει τη λέξη ή τη λίστα στο χώρο εντολών. Μια λίστα τυπώνεται μαζί με τις αγκύλες της



SIN μοίρες

Επιστρέφει το ημίτονο του αριθμού εισόδου σε μοίρες

SQRT αριθμός

Επιστρέφει την τετραγωνική ρίζα του αριθμού

ST

Εμφανίζει τη χελώνα

STARTUP

Ένα ειδικό όνομα διαδικασίας ή φύλλου. Η διαδικασία startup εκτελείται αμέσως με την κλήση του φύλλου που την περιέχει. Το φύλλο με όνομα startup εμφανίζεται αυτόματα με την είσοδο στο περιβάλλον της Logo

STOP

Σταματά τη διαδικασία που εκτελείται

STOPALL

Σταματά όλες τις διαδικασίες που εκτελούνται

TELL χελώνα / λίστα χελωνών

Καθορίζει την ή τις χελώνες που θα ακολουθούν τις εντολές

TO

Δηλώνει την αρχή του ορισμού μιας διαδικασίας. Χρησιμοποιείται μόνο στην πίσω πλευρά του φύλλου



A.3 Πλήκτρα ειδικών λειτουργιών - Συνδυασμοί πλήκτρων

F1 Επιλογή (σε συνδυασμό με κίνηση του δρομέα)

F2 Κοπή επιλεγμένου κειμένου

F3 Αντιγραφή επιλεγμένου κειμένου

F4 Κόληση επιλεγμένου κειμένου

F6 Σβήσιμο αρχείου

Ctrl - Break ή Ctrl - S Διακοπή εκτέλεσης διαδικασίας

Ctrl - F Αναστροφή πλευράς φύλλου

Ctrl - D Μετακίνηση δρομέα κάτω

Ctrl - U Μετακίνηση δρομέα πάνω

Ctrl ← Μετακίνηση δρομέα στην αρχή γραμμής

Ctrl → Μετακίνηση δρομέα στο τέλος γραμμής

PgDn Επόμενη οθόνη

PgUp Προηγούμενη οθόνη

Home Αρχή σελίδας

End Τέλος σελίδας



A.4 Προειδοποιητικά μηνύματα

Can't divide by zero

Δε μπορώ να διαιρέσω δια του μηδέν

Can't use to as a command

Δεν μπορώ να χρησιμοποιήσω το TO ως εντολή

Disk full

Ο δίσκος είναι γεμάτος

I don't know how to ...

Δεν υπάρχει εντολή ή διαδικασία με αυτό το όνομα στο φύλλο

I don't know what to do with ...

Η γραμμή δε ξεκινά με εντολή

Missing] in ... ή Missing [in ...

Λείπει] ή [

... didn't report anything to ...

Μια εντολή ή διαδικασία που απαιτεί κάποια είσοδο, δεν τη βρίσκει

... has no value

Η μεταβλητή δεν έχει τιμή

... needs more inputs

Η μεταβλητή ή η διαδικασία απαιτεί περισσότερες εισόδους

Please name this page

Παρακαλώ δώστε όνομα στο φύλλο

Stopped!!

Έχουν πατηθεί τα πλήκτρα για σταμάτημα εκτέλεσης της διαδικασίας



- 1) Πρωτογενής διαδίκτυα - Βαβυλκή ρυθμίση
- 2) Άμεση διδακτορική
- 3) Δευτερογενής διαδίκτυα - εξειδικευμένη
- 4) Άμεση αειδρική
- 5) Διδακτορική με βόλτη αλληλεπίδραση
- 6) Έλεγχος προόδου
- 7) Συνεργασία.
- 8) Απομνημόνευση
- 9) Μεταφορά γνώσεων
- 10) Τύποι ρόλων
- 11) Ενίσχυση-πρωτογενής-δράση
- 12) Έκδοση αντιλήψεων



ΤΥΠΟΓΡΑΦΙΚΕΣ ΕΡΓΑΣΙΕΣ
Γ. ΔΟΥΒΑΛΗΣ - Ε. ΑΠΟΣΤΟΛΟΥ Ο.Ε.

ΙΩΑΝΝΙΝΑ, 28ης ΟΚΤΩΒΡΙΟΥ 41 (Στοά Καμπέρη) ΤΗΛ.: (0651) 27.845 - 20.544 FAX: 24672



