

ΚΕΦΑΛΑΙΟ 2

Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΗΝ ΠΡΑΞΗ
[Προσαρμογή στο KTurtle]

Σχολείο:

Σχ. έτος:

.....

.....

Ο Προγραμματισμός στην Πράξη



Εισαγωγή

Για να υλοποιήσουμε αλγορίθμους στον υπολογιστή, θα χρησιμοποιήσουμε σε αυτό το κεφάλαιο τη γλώσσα προγραμματισμού Logo. Η γλώσσα Logo έχει πάρει το όνομά της από την ελληνική λέξη «λόγος». Συγκεκριμένα το περιβάλλον προγραμματισμού στο οποίο θα αναπτύξουμε προγράμματα είναι το Kturtle του KDE. Αν δεν έχετε στη διάθεσή σας το περιβάλλον αυτό, μπορείτε να χρησιμοποιήσετε κάποια άλλη έκδοση της Logo που διανέμεται δωρεάν στο Διαδίκτυο. Αν και το Kturtle περιέχει τις εντολές στα ελληνικά, μπορούμε και με τις άλλες εκδόσεις της Logo να φτιάχνουμε αντίστοιχα προγράμματα, αρκεί να μάθουμε τις βασικές αρχές του προγραμματισμού της Logo και να βρούμε τις αντίστοιχες εντολές που χρησιμοποιεί η κάθε έκδοση.

- ✓ Πώς μπορούμε να δημιουργούμε γεωμετρικά σχέδια με τη Logo;
- ✓ Μπορούμε να κατασκευάσουμε ένα πρόγραμμα αριθμομηχανής;
- ✓ Είναι εύκολο να προγραμματίσουμε παιχνίδια;

Στο Κεφάλαιο που ακολουθεί θα μάθουμε να δίνουμε εντολές και να φτιάχνουμε μικρά προγράμματα στη γλώσσα Logo.

Το πρόγραμμα Kturtle είναι διαθέσιμο για όλες τις μεγάλες διανομές Linux καθώς και για Windows: <http://windows.kde.org/> Μπορείτε να το κατεβάσετε και να το εγκαταστήσετε στον υπολογιστή σας εντελώς δωρεάν



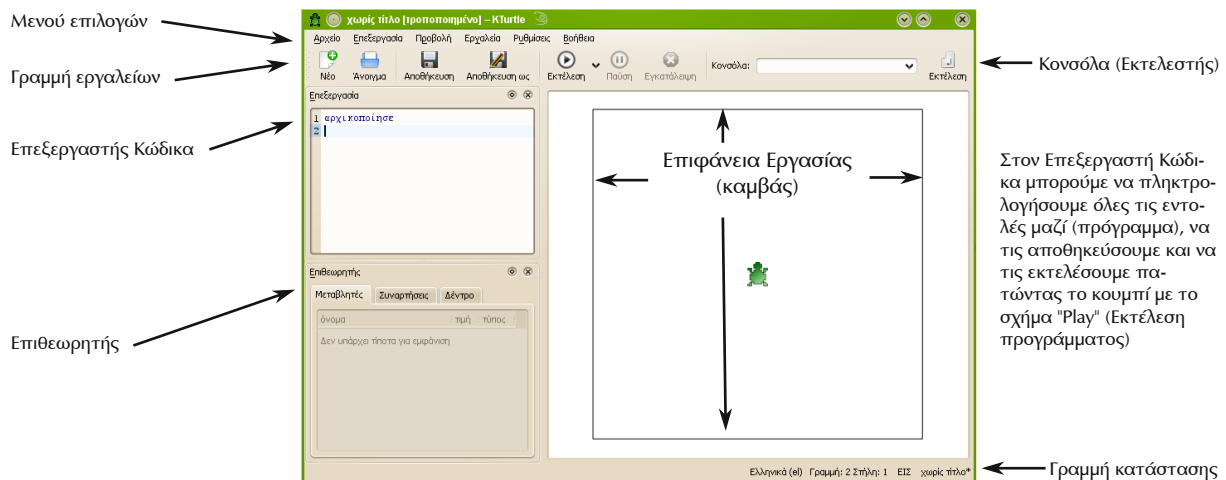
Λέξεις Κλειδιά

Περιβάλλον Προγραμματισμού,
Γλώσσα Logo,
Εντολή Εισόδου,
Εντολή Εξόδου,
Χελώνα,
Μεταβλητή,
Διαδικασία,
Δομή Επανάληψης,
Δομή Επιλογής

2.1 Το περιβάλλον προγραμματισμού Kturtle

Την πρώτη φορά που παρατηρούμε το περιβάλλον προγραμματισμού του **Kturtle** βλέπουμε ότι η οθόνη χωρίζεται σε τρεις περιοχές: *Επιφάνεια εργασίας (καμβάς)*, *Επεξεργαστής Κώδικα* και *Επιθεωρητής* (Εικόνα 2.1).

Στην *Κονσόλα* (Εκτελεστής) μπορούμε να πληκτρολογούμε εντολές στη γλώσσα προγραμματισμού Logo. Με την πληκτρολόγηση μιας εντολής (οδηγίας) και την επιλογή του πλήκτρου «Enter», η εντολή μεταφράζεται από το διερμηνέα σε γλώσσα μηχανής, ώστε να την εκτελέσει ο υπολογιστής. Το αποτέλεσμα της επε-



Εικόνα 2.1. Το περιβάλλον εκπαιδευτικού προγραμματισμού Kturtle

ξεργασίας των εντολών εμφανίζεται στην *Επιφάνεια εργασίας* (δηλαδή τον Καμβά) του περιβάλλοντος. Αν δεν συντάξουμε σωστά την εντολή που θέλουμε, τότε στην Κονσόλα αλλάζει το χρώμα φόντου ενώ στον Επεξεργαστή Κώδικα το χρώμα της εντολής και μας ειδοποιεί ότι έχουμε κάνει κάποιο λάθος.

Σημαντικός σύμβολός μας είναι η «Βοήθεια» που μας προσφέρει το προγραμματιστικό περιβάλλον. Χρησιμοποιήστε τη «Βοήθεια», για να αντλήσετε χρήσιμες πληροφορίες και παραδείγματα για τον προγραμματισμό με τη Logo, το λεξιλόγιο που χρησιμοποιεί και τον τρόπο σύνταξης των εντολών.

2.2 Οι πρώτες εντολές

Εντολή εμφάνισης (εξόδου) και αριθμητικές πράξεις

Μια βασική βοήθεια που μπορεί να μας προσφέρει ο υπολογιστής είναι η εκτέλεση σύνθετων αριθμητικών πράξεων. Αν ανατρέξουμε στην ιστορία των υπολογιστών, θα διαπιστώσουμε ότι οι πρώτοι ηλεκτρονικοί υπολογιστές στη δεκαετία του '40 είχαν κατασκευαστεί, για να βοηθήσουν στην εκτέλεση διάφορων υπολογισμών. Όλες λοιπόν οι διαδομένες γλώσσες προγραμματισμού έχουν σχεδιαστεί, ώστε να μπορούμε να εκτελούμε αριθμητικές πράξεις.

Όπως αναφέρθηκε στο πρόβλημα της εκδρομής (1ο Κεφάλαιο της Α' Γυμνασίου), τα δύο παιδιά αφού κατανόησαν το πρόβλημα της συγκέντρωσης χρημάτων για την εκπαιδευτική εκδρομή, συγκέντρωσαν τα απαραίτητα δεδομένα και βρήκαν ως λύση ότι έπρεπε να διαιρέσουν το κόστος ενοικίασης του λεωφορείου με το πλήθος των μαθητών που επρόκειτο να συμμετάσχουν στην εκδρομή. Συγκεκριμένα, έπρεπε να κάνουν τη διαίρεση $200 : 25 = ;$. Μπορούμε να δώσουμε μια εντολή με τη γλώσσα προγραμματισμού Logo και να μας εμφανίσει το αποτέλεσμα της διαίρεσης;

Η κατάλληλη εντολή είναι η: «**τύπωσε 200/25**». Η εντολή αυτή εκτελεί την πράξη $200 : 25$ και εμφανίζει το αποτέλεσμα στον καμβά. Η εντολή «τύπωσε» είναι μια εντολή **εξόδου**, καθώς έχει ως αποτέλεσμα την εμφάνιση ενός αριθμού ή μιας λέξης στην οθόνη του υπολογιστή.

Ο υπολογιστής μπορεί να κάνει όλες τις αριθμητικές πράξεις. Για τα σύμβολα των πράξεων χρησιμοποιούμε τα σύμβολα που υπάρχουν στο αριθμητικό πληκτρολόγιο (στα δεξιά του πληκτρολογίου): « + » για πρόσθεση, « - » για αφαίρεση, « * » για πολλαπλασιασμό και « / » για διαίρεση.

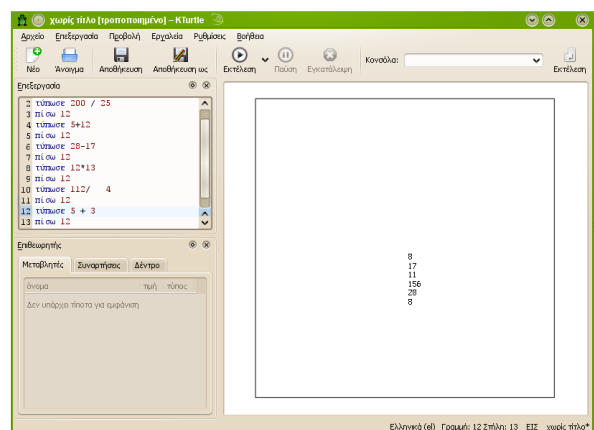
Παρατήρηση: Την εντολή «τύπωσε», όπως και τις υπόλοιπες εντολές, πρέπει να τις γράφουμε πάντα με μικρά γράμματα, χωρίς να ξεχνούμε τον τόνο. Το περιβάλλον του Kturtle ΔΕΝ θα αναγνωρίσει τις εντολές εάν δεν είναι ορθογραφικά σωστές. Το «Τύπωσε» και το «τυπωσε» π.χ. δεν είναι αποδεκτά.

Σημείωση: Όταν κάνουμε πράξεις, δεν είναι υποχρεωτικό να αφήνουμε κενό διάστημα πριν και μετά το σύμβολο της πράξης που χρησιμοποιούμε. Η εντολή «τύπωσε 5+3» θεωρείται σωστή για την Kturtle όπως και η «τύπωσε 5 + 3». Επίσης σωστή είναι και η «τύπωσε 5+ 3».

Ας δοκιμάσουμε τώρα λίγο πιο σύνθετες πράξεις. Δώστε στον υπολογιστή τις εντολές:

Εισαγωγική Δραστηριότητα
Δοκιμάστε τις παρακάτω εντολές και συμπληρώστε τα αποτελέσματα στον πίνακα. Στη συνέχεια προσπαθήστε να κάνετε διάφορους υπολογισμούς δοκιμάζοντας διάφορα νούμερα.

Εντολή	Αποτέλεσμα
τύπωσε 5+12	<input type="text"/>
τύπωσε 28-17	<input type="text"/>
τύπωσε 12*13	<input type="text"/>
τύπωσε 112/4	<input type="text"/>
τύπωσε ρίζα 4	<input type="text"/>



Εικόνα 2.2. Η χρήση της εντολής «τύπωσε»

- α. τύπωσε $12 / 2 * 3$
 β. τύπωσε $(12 / 2) * 3$
 γ. τύπωσε $12 / (2 * 3)$

1. Ποιο είναι το αποτέλεσμα στις περιπτώσεις α) ____, β) ____ και γ) ____;
2. Με ποια σειρά εκτελέστηκαν οι πράξεις στις τρεις αυτές εντολές;

Η Εμφάνιση Μηνυμάτων

Η εντολή «**τύπωσε**» επιτρέπει, εκτός από αριθμούς, να εμφανίζεται στην Επιφάνεια εργασίας (Καμβά) και κάποια λέξη. Αν, για παράδειγμα, θέλουμε να εμφανίσουμε το όνομά μας τότε μπορούμε να γράψουμε «**τύπωσε "Νίκος"**».

- Δοκιμάστε να εμφανίσετε και το δικό σας όνομα. Συμπληρώστε στο κενό την εντολή που θα δώσετε στον υπολογιστή:
- Ποιο πιστεύετε ότι θα είναι το αποτέλεσμα της εντολής **τύπωσε "2+3"**:

Αν μετά την εντολή «**τύπωσε**» βάλουμε εισαγωγικά, τότε η εκτέλεση της εντολής θα έχει ως αποτέλεσμα την εμφάνιση της λέξης που ακολουθεί μετά τα εισαγωγικά. Η εντολή «**τύπωσε "2+3"**» εμφανίζει το «2+3» και όχι το αποτέλεσμα της πράξης, γιατί ο υπολογιστής εκλαμβάνει το 2+3 ως μία λέξη και όχι ως αριθμούς με τους οποίους πρέπει να κάνει πρόσθεση.

Μέχρι τώρα είδαμε πώς μπορούμε να εμφανίζουμε αριθμούς ή λέξεις στην οθόνη του υπολογιστή. Πώς μπορούμε, όμως, να εμφανίσουμε ένα ολόκληρο κείμενο; Αν θέλουμε να εμφανίσουμε το μήνυμα «Το όνομά μου είναι Πελαγία», τότε πρέπει να γράψουμε «**τύπωσε "Το όνομά μου είναι Πελαγία"**». Ο υπολογιστής θα εμφανίσει όλες τις λέξεις που περικλείονται μεταξύ των εισαγωγικών "...". Οι λέξεις που βρίσκονται μεταξύ εισαγωγικών αποτελούν ένα σύνολο λέξεων (μία λίστα). Πειραματιστείτε εμφανίζοντας τα δικά σας μηνύματα στον υπολογιστή.

Πώς μπορούμε, όμως, να εμφανίζουμε μηνύματα μαζί με τα αποτελέσματα αριθμητικών πράξεων; Για παράδειγμα, ποια εντολή θα δίνουμε, για να εμφανιστεί στον Κωστή και στη Χρύσα το μήνυμα: «Το κόστος της εκδρομής ανά μαθητή είναι 8 €», όπου το 8 είναι το αποτέλεσμα της πράξης $200 / 25$; Αν θέλουμε να ενώσουμε δύο μηνύματα μεταξύ τους, τότε πρέπει να χρησιμοποιήσουμε το σύμβολο «+». Δοκιμάστε την εντολή «**τύπωσε "Το κόστος της εκδρομής ανά μαθητή είναι " + (200/25) + " ευρώ."**». Τι εμφανίζεται στην οθόνη; Μπορείτε να βρείτε περισσότερα για την εντολή «**τύπωσε**» στη Βοήθεια του προγράμματος και να πειραματιστείτε μ' αυτήν.

2.3 Συνομιλία με τον υπολογιστή.

Περισσότερα για τις εντολές εισόδου-εξόδου

Στις προηγούμενες παραγράφους είχαμε την ευκαιρία να γνωρίσουμε την εντολή εξόδου «**τύπωσε**» με την οποία εμφανίζουμε δεδομένα στην οθόνη του υπολογιστή. Το περιβάλλον εκπαιδευτικού προγραμματισμού K Turtle μας δίνει τη δυνατότητα να εμφανίζουμε ανακοινώσεις κειμένων με πιο εντυπωσιακό τρόπο.



Ας δούμε πώς μπορούμε να γράψουμε εντολές που δημιουργούν μια απλή εικονική συνομιλία. Η εντολή εξόδου «**μήνυμα "κείμενο"**» εμφανίζει ένα μήνυμα σε ένα παράθυρο στην οθόνη. Στη θέση «**κείμενο**» μπορούμε να προσθέσουμε όποια φράση θέλουμε. Το αποτέλεσμα της εντολής «**μήνυμα "Γειά σου!"**» φαίνεται στη διπλανή οθόνη.

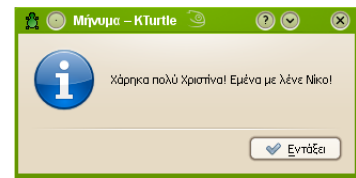
Ωστόσο ένας ενδιαφέρον διάλογος περιέχει και ερωτήσεις. Ερωτήσεις μπορούμε να κάνουμε με την εντολή «**ερώτηση "κείμενο"**» και στην κενή περιοχή που εμφανίζεται μπορούμε να δώσουμε μίαν απάντηση. Το παράθυρο της ερώτησης

« $\$x$ = ερώτηση "Πώς σε λένε;"» φαίνεται στο διπλανό παράθυρο. Το πλαίσιο χρησιμεύει, για να πληκτρολογήσουμε την απάντησή μας.

Η εντολή «ερώτηση» είναι μια εντολή εισόδου, γιατί μας επιτρέπει να δώσουμε μία τιμή (μία λέξη, ένα σύνολο λέξεων ή έναν αριθμό) στον υπολογιστή, ώστε στη συνέχεια να την επεξεργαστεί ή να την εμφανίσει στην οθόνη. Αν θέλουμε να χρησιμοποιήσουμε ξανά την τιμή που δίνουμε στο πλαίσιο της ερώτησης, αυτή αποθηκεύεται προσωρινά σε μια μεταβλητή (την $\$x$ στο παραπάνω παράδειγμα) και μπορούμε να την ανακτήσουμε χρησιμοποιώντας τη **μεταβλητή αυτή**, όπως στο επόμενο παράδειγμα.

μήνυμα "Χάρηκα πολύ " + $\$x$ + "! Εμένα με λένε Νίκο!"

Για να καταλάβετε καλύτερα τη χρήση των εντολών «μήνυμα» και «ερώτηση», φτιάξτε τις δικές σας συνομιλίες.



Δραστηριότητα: Ας πειραματιστούμε λίγο και με τους αριθμούς

1. Τι ακριβώς κάνουν οι δύο παρακάτω εντολές:

$\$x$ = ερώτηση "Δώσε μου τον αριθμό που θέλεις να ψύσεις στο τετράγωνο:"
μήνυμα $\$x$ * $\$x$

2. Ποιο είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών, αν δώσουμε την τιμή 3456. Δοκιμάστε το στον υπολογιστή και στη συνέχεια δώστε και άλλες τιμές πατώντας κάθε φορά το κουμπί «Εκτέλεση» για να εκτελέσετε πάλι το πρόγραμμα.



3. Πώς μπορούν να τροποποιηθούν οι παραπάνω εντολές, ώστε να υπολογίζουμε τον κύβο ενός αριθμού;

2.4 Η Logo και ο σχεδιασμός γεωμετρικών σχημάτων

Κάνοντας τις πρώτες δοκιμές με τη χελώνα

Το εργαλείο χελώνα είναι ίσως το πιο βασικό χαρακτηριστικό της γλώσσας Logo. Για να δημιουργήσουμε ένα νέο πρόγραμμα στο περιβάλλον του K-Turtle, χρησιμοποιούμε το εικονίδιο «Νέο»:



Πολλές εντολές στη γλώσσα Logo μετακινούν και χειρίζονται τη χελώνα στην Επιφάνεια εργασίας (καμβά). Το σχήμα της χελώνας που εμφανίζεται είναι:



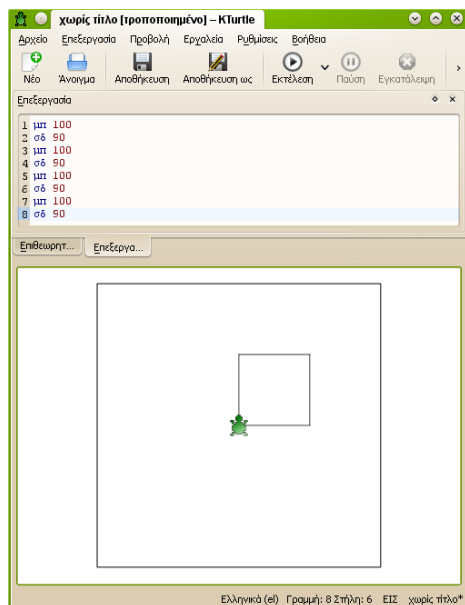
Το ίχνος που αφήνει η χελώνα, με την κατάλληλη μετακίνησή της, μας επιτρέπει να δημιουργήσουμε διάφορα σχέδια και γεωμετρικά σχήματα. Οι βασικές εντολές που μπορούμε να δώσουμε στη χελώνα, ώστε να την κατευθύνουμε, είναι:

- **μπροστά (μπ) αριθμός εικονοστοιχείων:** Με την εκτέλεση της εντολής αυτής η χελώνα προχωράει μπροστά τόσα εικονοστοιχεία όσα έχουμε ορίσει.
- **πίσω (πι) αριθμός εικονοστοιχείων:** Με την εκτέλεση της εντολής αυτής η χελώνα προχωράει προς τα πίσω τόσα εικονοστοιχεία όσα έχουμε ορίσει.
- **στρίψεδεξιά (σδ) μοίρες:** Η χελώνα στρίβει προς τα δεξιά τόσες μοίρες όσες έχουμε ορίσει.
- **στρίψεαριστερά (σα) μοίρες:** Η χελώνα στρίβει προς τα αριστερά τόσες μοίρες όσες έχουμε ορίσει.
- **στυλόκάτω (σκ):** Δίνει εντολή στην χελώνα να αφήνει ίχνος από κάθε σημείο της οθόνης που περνάει. Όταν ξεκινά η εκτέλεση ενός προγράμματος το στυλό της χελώνας βρίσκεται πάντα κάτω.
- **στυλόπάνω (σπ):** Δίνει εντολή στη χελώνα να σταματήσει να αφήνει ίχνος καθώς προχωράει.
- **αρχικοποίηση:** Σβήνει τα σχέδια που έχουμε δημιουργήσει από την επιφάνεια εργασίας (καμβά) και μεταφέρει τη χελώνα στο κέντρο της επιφάνειας εργασίας με κατεύθυνση προς τα πάνω.

Εισαγωγική Δραστηριότητα

Δημιουργείστε ένα νέο πρόγραμμα κάνοντας κλικ πάνω στο κουμπί «Νέο». Πειραματιστείτε μετακινώντας τη χελώνα στον καμβά και δοκιμάστε τις διπλανές επτά εντολές για να φτιάξετε τα δικές σας προγράμματα. Στη συνέχεια προσπαθείστε να δημιουργήσετε ένα ευθύγραμμο τμήμα μήκους 100 εικονοστοιχείων.

Ας θυμηθούμε λίγο τον αλγόριθμο του τετραγώνου που παρουσιάσαμε στο κεφάλαιο των αλγορίθμων. Ο αλγόριθμος αυτός περιέγραφε τα βήματα που πρέπει να ακολουθήσει ένα μικρό παιδί, ώστε να φτιάξει ένα τετράγωνο στην άμμο. Με μία μικρή παραλλαγή μπορούμε να υλοποιήσουμε τον αλγόριθμο αυτό, για να κατασκευάσουμε ένα τετράγωνο με μήκος πλευράς 100 εικονοστοιχεία, δίνοντας εντολές στη χελώνα.



Εικόνα 2.3. Δημιουργία ενός τετραγώνου με τη βοήθεια της χελώνας

Όπως βλέπουμε και στη διπλανή εικόνα ο συνδυασμός των εντολών:

```
μπ 100
σδ 90
μπ 100
σδ 90
μπ 100
σδ 90
μπ 100
σδ 90
```

δημιουργεί ένα τετράγωνο στην οθόνη μας (η τελευταία εντολή απλά επαναφέρει τη χελώνα στην αρχική κατεύθυνση).

Δομή επανάληψης

Αν μελετήσουμε καλύτερα το παραπάνω πρόγραμμα του τετραγώνου, παρατηρούμε ότι οι εντολές «μπ 100» και «σδ 90» επαναλήφθηκαν **τέσσερις φορές** με την ίδια σειρά. Θα μπορούσαμε να έχουμε το ίδιο αποτέλεσμα ομαδοποιώντας τις δύο εντολές και δίνοντας μια εντολή που να τις επαναλαμβάνει τέσσερις φορές. Η εντολή αυτή είναι: «**επανάλαβε** αριθμός_επαναλήψεων {λίστα οδηγιών}».

Με την εντολή αυτή μπορούμε να κατασκευάσουμε το ίδιο τετράγωνο ως εξής:

```
αρχικοποίηση
επανάλαβε 4 {μπ 100 σδ 90}
```

Η **δομή της επανάληψης** είναι πολύ χρήσιμη στον προγραμματισμό. Χρησιμοποιώντας τις δομές επανάληψης ο υπολογιστής μπορεί να εκτελεί τις επαναλαμβανόμενες ενέργειες (υπολογισμούς, εμφανίσεις στην οθόνη κλπ.) και μάλιστα πολύ πιο γρήγορα από εμάς.



Ερώτηση

Ποιο αποτέλεσμα θα προέκυπτε, αν δε γράφαμε την εντολή «αρχικοποίηση» στην αρχή;

Δραστηριότητες:

1. Να γράψετε την κατάλληλη εντολή, ώστε να εμφανιστεί το όνομά σας 200 φορές στην οθόνη του υπολογιστή:

.....

2. Στην εντολή που χρησιμοποιήσαμε, για να σχεδιάσουμε ένα τετράγωνο:

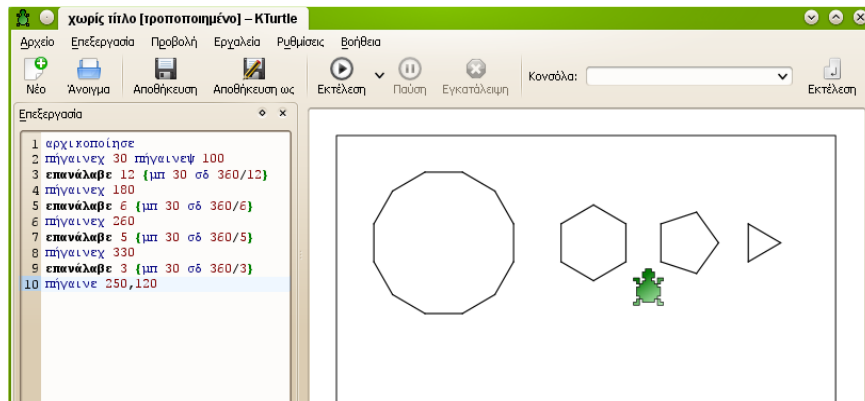

```
επανάλαβε 4 {μπ 100 σδ 90}
```

συνολικά στο σχήμα μας κάναμε στροφή 360 μοιρών σε 4 βήματα. Δηλαδή σε κάθε βήμα στρίψαμε $360:4=90$ μοίρες.

Να δώσετε τις κατάλληλες εντολές στη χελώνα, ώστε να σχεδιάσει ένα ισόπλευρο τρίγωνο, ένα πεντάγωνο, ένα εξαγώνο ή ένα δωδεκάγωνο, όπως τα σχήματα της Εικόνας 2.4.

3. Με τι μοιάζει το σχήμα που δημιουργεί η επόμενη εντολή:


```
επανάλαβε 360 {μπ 1 σδ 1}
```



Εικόνα 2.4. Δημιουργία γεωμετρικών σχημάτων με τη χελώνα του K-Turtle

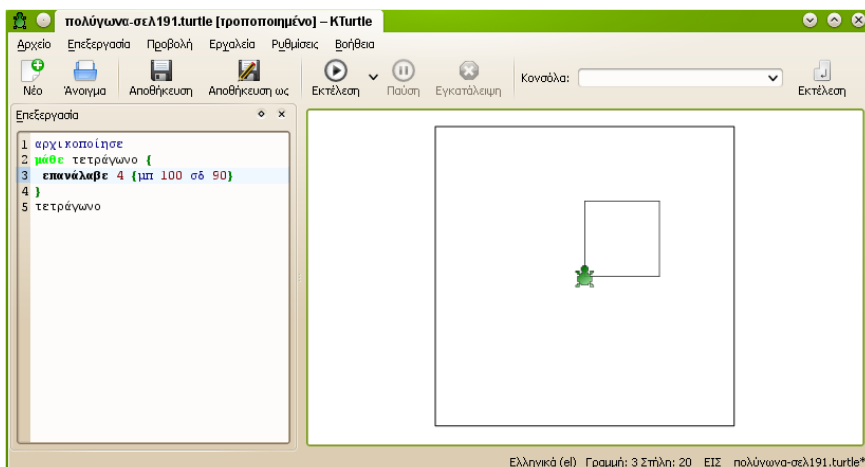
2.5 Δημιουργώντας νέες λέξεις - Διαδικασίες

Εντολές που επιθυμούμε να εκτελεστούν με τη σειρά μπορούν να ομαδοποιηθούν σε ένα νέο όνομα. Η ομαδοποίηση αυτή των εντολών καλείται **Διαδικασία**. Η εκτέλεση μίας διαδικασίας μπορεί να χρησιμεύσει στον υπολογισμό ενός μαθηματικού τύπου (συνάρτηση), στη δημιουργία ενός σχήματος ή στην εμφάνιση μιας συνομιλίας. Το πλεονέκτημα της διαδικασίας είναι ότι μπορούμε να την καλέσουμε με το όνομά της όποτε τη χρειαστούμε, χωρίς να είναι απαραίτητο να πληκτρολογήσουμε ξανά όλες τις εντολές που περιέχει.

Για τη δημιουργία μιας διαδικασίας πληκτρολογούμε στο πλαίσιο επεξεργασίας εντολών την ομάδα των εντολών με τη μορφή:

```
μάθε όνομαδιαδικασίας {
    εντολή 1
    εντολή 2
    ...
}
```

Στο επόμενο παράδειγμα (Εικόνα 2.5) έχουμε γράψει ένα παράδειγμα μιας διαδικασίας, με την οποία σχεδιάζουμε ένα τετράγωνο πλευράς 100.



Εικόνα 2.5. Η διαδικασία τετράγωνο έχει δημιουργήσει μια νέα εντολή «τετράγωνο»

Κάθε φορά που, μέσα στο πρόγραμμα, γράφουμε το όνομα της διαδικασίας «τετράγωνο», σχηματίζεται ένα τετράγωνο. Ουσιαστικά με αυτόν τον τρόπο η Logo μας επιτρέπει να δημιουργούμε τις δικές μας λέξεις-εντολές.

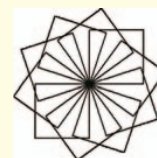
1η Δραστηριότητα

Δημιουργήστε δύο διαδικασίες: μία με το όνομα τετράγωνο, που θα σχεδιάζει ένα τετράγωνο και μία δεύτερη με το όνομα τρίγωνο, που θα σχεδιάζει ένα τρίγωνο. Χρησιμοποιώντας τις νέες λέξεις-εντολές «τετράγωνο» και «τρίγωνο» που μόλις δημιουργήσατε, προσπαθήστε να δημιουργήσετε μια διαδικασία με το όνομα σπίτι, που θα σχεδιάζει ένα σπιτάκι, όπως στο διπλανό σχήμα. Στη συνέχεια δημιουργήστε μια διαδικασία με το όνομα χωριό, που θα χρησιμοποιεί τη λέξη «σπίτι» και θα σχεδιάζει πολλά σπιτάρια το ένα δίπλα στο άλλο.



2η Δραστηριότητα

Προσπαθήστε να περιγράψετε την πορεία της χελώνας ακολουθώντας την εντολή:
επανάλαβε 10 {τετράγωνο σδ 36}
όπου «τετράγωνο» είναι το όνομα της διαδικασίας που δημιουργήσαμε στην Εικόνα 2.5. Το αποτέλεσμα της εντολής φαίνεται στη διπλανή εικόνα. Αφού έχετε κατανοήσει τη σημασία των αριθμών 10 και 36 στην εντολή, προσπαθήστε να τους αλλάξετε δημιουργώντας τα δικά σας σχήματα.



2.6 Μεταβλητές

Σύμφωνα με την παραπάνω διαδικασία «τετράγωνο», όταν θέλουμε να δημιουργήσουμε τετράγωνα με διαφορετικό μήκος πλευράς, πρέπει να επεμβαίνουμε κάθε φορά στην εντολή **μπροστά** και να αλλάζουμε το μήκος της πλευράς. Έτσι, αν θέλουμε να σχεδιάσουμε ένα τετράγωνο πλευράς 50 εικονοστοιχείων, θα δώσουμε την εντολή:

επανάλαβε 4 {μπ 50 σδ 90}

ενώ, αν θέλουμε ένα τετράγωνο πλευράς 150, θα δώσουμε την εντολή:

επανάλαβε 4 {μπ 150 σδ 90}

Μπορούμε να χρησιμοποιούμε την ίδια πάντα διαδικασία για το σχεδιασμό τετραγώνων διαφορετικών πλευρών; Η απάντηση είναι καταφατική. Τη διαδικασία «τετράγωνο» μπορούμε να την τροποποιήσουμε ως εξής:

```
αρχικοποίηση
μάθε νέοτετράγωνο $μήκος {
  επανάλαβε 4 {μπ $μήκος σδ 90}
}
```

Αφού έχουμε γράψει την παραπάνω διαδικασία, μπορούμε να σχεδιάσουμε ένα τετράγωνο πλευράς 50 δίνοντας την εντολή:

νέοτετράγωνο 50

Δηλαδή, μαζί με το όνομα της διαδικασίας δίνουμε και το επιθυμητό μήκος της πλευράς του τετραγώνου. Η τιμή 50 αποθηκεύεται προσωρινά στο \$μήκος και η εντολή «μπ \$μήκος» μετακινεί τη χελώνα μπροστά κατά 50 εικονοστοιχεία. Κάθε φορά που «καλούμε» τη διαδικασία νέοτετράγωνο στο «\$μήκος», αποθηκεύεται προσωρινά μια διαφορετική τιμή. Το «\$μήκος» ονομάζεται **μεταβλητή**.



Ερώτηση

Πώς θα ενεργοποιήσουμε τη διαδικασία νέοτετράγωνο, ώστε να σχεδιάσει ένα τετράγωνο με πλευρά μήκους 80;

Το περιεχόμενο μιας **μεταβλητής** μπορεί να μεταβάλλεται κατά την εκτέλεση ενός προγράμματος. Μια μεταβλητή αντιστοιχεί σε μία θέση της μνήμης του υπολογιστή και γίνεται αναφορά σε αυτή με το όνομα που της δίνουμε εμείς. Μία θέση μνήμης μπορεί να έχει **μόνο μία τιμή** κάθε φορά, αλλά μπορούμε να την αλλάζουμε, όποτε είναι απαραίτητο, με μία άλλη τιμή.

Φανταστείτε τη μεταβλητή σα μια φωλιά, η οποία χωράει μόνο ένα αυγό. Όπως μπορούμε να αντικαθιστούμε το αυγό στη φωλιά με ένα άλλο, έτσι μπορούμε να αντικαθιστούμε την τιμή μιας μεταβλητής με μία άλλη τιμή.

Η προηγούμενη τιμή της μεταβλητής, όμως, χάνεται και δεν μπορούμε να τη χρησιμοποιήσουμε ξανά. Ωστόσο, μπορούμε να χρησιμοποιήσουμε περισσότερες μεταβλητές, για να αποθηκεύσουμε διαφορετικές τιμές.

Στο πρόγραμμα K Turtle, για να αναφερθούμε στην τιμή της μεταβλητής, χρησιμοποιούμε το όνομά της. Το ίδιο κάνουμε και στην περίπτωση που θέλουμε να αποθηκεύσουμε μια τιμή στη μεταβλητή. Η διαδικασία αυτή λέγεται εκχώρηση τιμής.

Για να δώσουμε (εκχωρήσουμε) τιμή σε μία μεταβλητή, μπορούμε να χρησιμοποιήσουμε το σύμβολο του ίσον «=» ως εξής: «**\$μεταβλητή = ΤιμήΜεταβλητής**». Αν θέλουμε, για παράδειγμα, να δώσουμε στη μεταβλητή με όνομα \$X την τιμή 2, γράφουμε:

$$\$X = 2$$

ενώ, αν θέλουμε να δώσουμε την τιμή Γάτα γράφουμε:

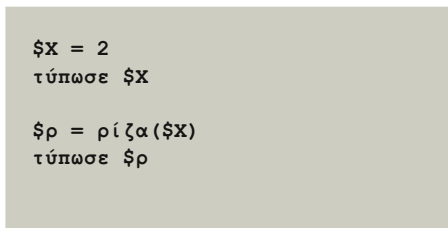
$$\$X = \text{"Γάτα"}$$

Πολλές φορές κάνουμε το λάθος και λέμε ότι η τιμή του X είναι ίση με 2.

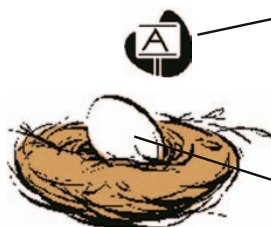
Ποια εντολή θα χρησιμοποιήσουμε, για να εμφανίσουμε την τιμή που περιέχει η μεταβλητή X: _____;

Πώς μπορούμε να υπολογίσουμε στη συνέχεια την τετραγωνική ρίζα του X και να εμφανιστεί στην οθόνη;

Η απάντηση φαίνεται στην παρακάτω Εικόνα 2.6.



Εικόνα 2.6. Εκχώρηση τιμής σε μεταβλητή και εμφάνισή της στην οθόνη



\$A: το όνομα της μεταβλητής

Δίνοντας τιμή σε μία μεταβλητή είναι σαν να τοποθετούμε στη φωλιά ένα αυγό

Η τιμή της μεταβλητής

Το σύμβολο ίσον «=» χρησιμοποιείται για εκχώρηση τιμής. Σαν σύμβολο ισότητας χρησιμοποιείται το «=».



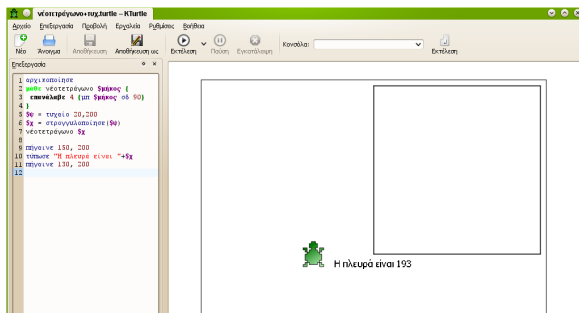
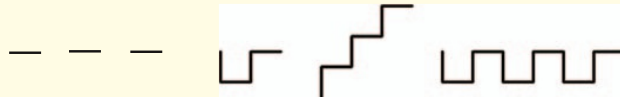
Η μεταβλητή στον προγραμματισμό δεν έχει την ίδια έννοια που έχει η μεταβλητή στα Μαθηματικά. Στον προγραμματισμό σε μία μεταβλητή X τοποθετούμε (εκχωρούμε) μία τιμή, δηλαδή, στη θέση μνήμης που αντιστοιχεί στη μεταβλητή X αποθηκεύουμε προσωρινά μία τιμή.

Δραστηριότητες

1. α) Προσπαθήστε να δώσετε το όνομά σας σε μία μεταβλητή \$ONOMA και στη συνέχεια εμφανίστε το στον καμβά, το χώρο που ζωγραφίζει η χελώνα.
 β) Προσπαθήστε να εμφανίσετε στην οθόνη το όνομά σας, χωρίς να το ξαναγράψετε με το συνοδευτικό μήνυμα «Το όνομά μου είναι.....».
2. Γράψτε δίπλα από τις εντολές εξόδου τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των εντολών;
 - A. \$ζώο = "λιοντάρι"
 τύπωσε \$ζώο
 τύπωσε "λιοντάρι"
 τύπωσε "ζώο"
 \$ζώο = "σκύλο"
 τύπωσε "έχω ένα " + \$ζώο
 - B. \$χ = 3
 τύπωσε 12 + 5 * \$χ
 τύπωσε 2 * 5 - \$χ * 4
 \$χ = 8
 τύπωσε 14 + 2 + \$χ / 2
3. Γράψτε και εκτελέστε τις παρακάτω εντολές:
 \$α = 1
 τύπωσε \$α
 επανάλαβε 9 { \$α = \$α + 1 τύπωσε \$α }

Ποιο είναι το αποτέλεσμα της εκτέλεσης της διαδικασίας;
 Βρείτε ποια είναι η λειτουργία της εντολής «\$α = \$α + 1», ώστε να μπορείτε να τη χρησιμοποιήσετε και στις επόμενες ασκήσεις.

4. Να δημιουργήσετε μία διαδικασία που να κατασκευάζει ένα ορθογώνιο παραλληλόγραμμο δίνοντάς του τα εκάστοτε μήκη των πλευρών.
5. Να καταγραφεί η διαδικασία που να υπολογίζει το εμβαδόν ενός τριγώνου πλευράς α και ύψους υ.
6. Δημιουργείστε τα 4 διπλανά σχήματα



Εικόνα 2.7. Η διαδικασία νέοτετράγωνο παίρνει τιμές από τη γεννήτρια τυχαίων αριθμών (εντολή «τυχ»).

Μια παραλλαγή της διαδικασίας τετράγωνο ή παίζοντας με τη γεννήτρια τυχαίων αριθμών

Αντί να δίνουμε κάθε φορά το μήκος της πλευράς στη διαδικασία «τετράγωνο», θα μπορούσαμε να χρησιμοποιήσουμε τη γεννήτρια τυχαίων αριθμών (Εικόνα 2.7). Με τη χρήση της εντολής «τυχαίο» (ή «τυχ» συντομότερα) μπορούμε να βάλουμε σε μια μεταβλητή έναν τυχαίο αριθμό. Φυσικά, καθορίζουμε το πάνω και το κάτω όριο τιμών. Τη χρησιμοποιούμε για να δημιουργήσουμε τυχαία μεγέθη σχημάτων, τυχαία χρώματα κοκ. Στο επόμενο παράδειγμα φαίνεται ο τρόπος σύνταξης της εντολής:

$$\$χ = \text{τυχ } 1,10$$

(βάζει στην \$χ ένα τυχαίο αριθμό ανάμεσα στο 1 και το 10)

Η χελώνα του Kturtle καταλαβαίνει και χρησιμοποιεί το χρωματικό μοντέλο RGB (Κόκκινο, Πράσινο, Μπλε) για τη διαχείριση των χρωμάτων. Για καθένα από τα τρία βασικά χρώματα καθορίζεται ένας αριθμός από το 0 έως το 255. Για παράδειγμα, 0,0,0 είναι το μαύρο, 255,255,255 το άσπρο, 255,0,0 το κόκκινο κλπ. Δίνοντας τυχαίους αριθμούς μπορούμε να έχουμε τυχαία χρώματα. Η Εικόνα 2.8 δείχνει τον τρόπο. Με την εντολή «χρώμα-στυλό» («χσ» σύντομα) αλλάζουμε το χρώμα του στυλό της χελώνας. Στην Εικόνα 2.9 δίνεται ένα παράδειγμα δημιουργίας τυχαίων χρωμάτων.

```

3  $κοκκ = τυχ 0,255
4  $πρασ = τυχ 0,255
5  $μπλε = τυχ 0,255
6  χσ $κοκκ, $πρασ, $μπλε

```

Εικόνα 2.8. Ο τρόπος δημιουργίας τυχαίων χρωμάτων με τις εντολές «τυχ» και «χσ»

2.7 Επιλέγοντας...

Αρκετές φορές η περιγραφή της λύσης ενός προβλήματος δεν είναι μία ακολουθία βημάτων που πρέπει να εκτελεστούν όλα σε σειρά το ένα μετά το άλλο. Υπάρχουν προβλήματα που, για να λυθούν, πρέπει να επιλέγουμε ποια βήματα θα εκτελεστούν.

1ο Παράδειγμα

Αν θέλουμε να περάσουμε το δρόμο, επιλέγουμε τι θα κάνουμε ανάλογα με το τι δείχνει το φανάρι. Αν το φανάρι για τους πεζούς είναι πράσινο, περνάμε το δρόμο. Αν είναι κόκκινο, σταματάμε και περιμένουμε. Όμοια, για να ξέρουμε, αν πρέπει να πάρουμε μαζί μας ομπρέλα ή γυαλιά ηλίου, σφειλούμε να κάνουμε τους παρακάτω ελέγχους:

- Αν βρέχει, τότε θα πάρουμε μαζί μας ομπρέλα.
- Αν ο ήλιος είναι δυνατός, τότε πρέπει να φορέσουμε τα γυαλιά ηλίου.

Στην παράγραφο αυτή θα έχουμε την ευκαιρία να μάθουμε πώς να δίνουμε εντολές στον υπολογιστή, ώστε να επιλέγει, ανάλογα με τις συνθήκες που ισχύουν, ποια βήματα πρέπει να εκτελέσει.

Αν θέλουμε να γράψουμε έναν αλγόριθμο που να τον εκτελεί ένα μικρό παιδί, ώστε να διασχίσει με ασφάλεια το δρόμο, πρέπει να συμπεριλάβουμε τον έλεγχο του φαναριού (Εικόνα 2.10). αλγόριθμος μπορεί να περιγραφεί με βήματα ως εξής:

1. Περπάτησε μέχρι την άκρη του πεζοδρομίου.
2. Έλεγξε το σηματοδότη για τους πεζούς.
3. Αν ο σηματοδότης είναι πράσινος, **τότε** πέρασε προσεκτικά το δρόμο, (δηλ. αν είναι κόκκινος), περίμενε στην άκρη του πεζοδρομίου μέχρι το φανάρι να γίνει πράσινο.

2ο Παράδειγμα

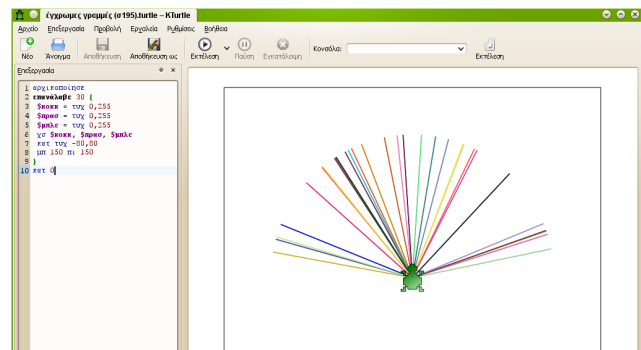
Να γραφεί ένας αλγόριθμος που θα μας δίνει την απόλυτη τιμή ενός αριθμού.

Αν θυμηθούμε λίγο τα Μαθηματικά, η απόλυτη τιμή ενός αριθμού x ισούται με:

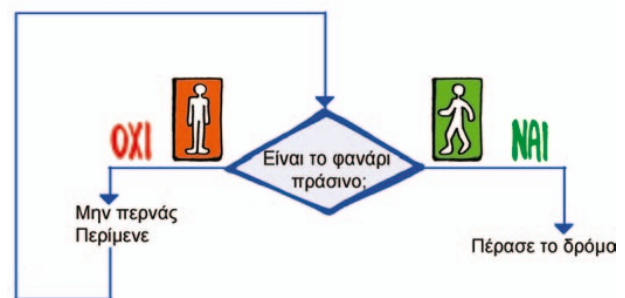
- x , αν $x > 0$,
- 0 , αν $x = 0$ και
- $-x$ αν $x < 0$.

Επομένως, ο αλγόριθμος για την εύρεση της απόλυτης τιμής ενός αριθμού, με μια μικρή μετατροπή, μπορεί να διαμορφωθεί ως εξής:

- Μάθε την τιμή του x .



Εικόνα 2.9. Παράδειγμα δημιουργίας τυχαίων χρωμάτων με τη χρήση των εντολών «τυχ» και «χσ»



Εικόνα 2.10. Σχηματική αναπαράσταση του αλγορίθμου έλεγχος φαναριού πεζών

- Αν το x είναι μικρότερο από το 0 τότε υπολόγισε την τιμή $-x$ (δηλαδή $-1*x$) και εμφάνισέ την
- διαφορετικά εμφάνισε το x .

Η εντολή της Kturtle που χρησιμεύει για την εκτέλεση του παραπάνω αλγορίθμου από τον υπολογιστή είναι η:

αν συνθήκη {εντολή 1} αλλιώς {εντολή 2}

Με την εντολή αυτή ο υπολογιστής ελέγχει αρχικά, αν ισχύει η συνθήκη. Στη συνέχεια ανάλογα με το αν ισχύει (είναι αληθής), εκτελεί την πρώτη εντολή. Διαφορετικά εκτελεί τη δεύτερη.

Η συνθήκη είναι μια λογική πρόταση. Χρησιμοποιεί συνήθως σύμβολα:

Σύμβολο	Σημασία	Παράδειγμα
==	ισότητα	$\$a == 5$
>	μεγαλύτερο	$\$a > \b
<	μικρότερο	$5 < \$a$
>=	μεγαλύτερο ή ίσο	$\$a >= 0$
<=	μικρότερο ή ίσο	$\$a <= \$b + 1$
!=	όχι ίσο (ανισότητα)	$\$a != 0$

Από τον αλγόριθμο της εύρεσης της απόλυτης τιμής ενός αριθμού μπορεί να προκύψει η εξής διαδικασία:

```

μάθε απόλυτη $χ {
  αν $χ < 0 {
    μήνυμα "η απόλυτη τιμή του χ είναι: " + (-1 * $χ)
  } αλλιώς {
    μήνυμα "η απόλυτη τιμή του χ είναι: " + $χ
  }
}

```

Στη διαδικασία «απόλυτη» η συνθήκη ελέγχου είναι η $\$χ < 0$ και χρησιμοποιείται για να ελέγξει, αν η τιμή της μεταβλητής « χ » είναι μικρότερη από το μηδέν. Αν είναι, τότε εκτελείται η πρώτη εντολή μήνυμα "η απόλυτη τιμή του χ είναι: " + $(-1 * \$\chi)$, η οποία ανακοινώνει στην οθόνη την τιμή του χ με το συνοδευτικό μήνυμα «το χ είναι:». Σε διαφορετική περίπτωση, εκτελείται η δεύτερη εντολή, που εμφανίζει το ίδιο μήνυμα και στη συνέχεια την αρχική τιμή του χ .

Δραστηριότητες

1. Φτιάξτε τα δικά σας παιχνίδια γνώσεων.

Χρησιμοποιώντας τις εντολές επιλογής μπορούμε να φτιάξουμε τα δικά μας παιχνίδια γνώσεων και να παίζουμε με τους φίλους μας

```

$απ = ερώτηση "Πώς ονομάζεται στα αγγλικά η μνήμη του
υπολογιστή, όπου αποθηκεύονται προσωρινά δεδομένα
και εντολές;"
αν $απ == "RAM" ή $απ == "R.A.M." {
  μήνυμα "ΜΠΡΑΒΟ!"
} αλλιώς {
  μήνυμα "Θα πρέπει να μελετήσεις ξανά το κεφάλαιο με
το υλικό του υπολογιστή"
}

```

α. Δοκιμάστε να αλλάξετε τις ερωτήσεις και τις απαντήσεις του παιχνιδιού γνώσεων.

β. Τι ελέγχει η συνθήκη: « $\$απ == "RAM"$ ή $\$απ == "R.A.M."$ »;

2. Δημιουργείστε μια διαδικασία που να υπολογίζει την τιμή του x στη συνάρτηση $\alpha \cdot x + \beta = 0$.

2.8 Δημιουργώντας πιο σύνθετες εφαρμογές με τη γλώσσα Logo Δημιουργία του παιχνιδιού «Μάντεψε τον αριθμό»

```

αρχικοποίηση
$σκορ = 0
$π = ερώτηση "Παίζουμε ΜΑΝΤΕΨΕ ΤΟΝ ΑΡΙΘΜΟ; (N/O)"
όσο $π == "N" ή $π == "ν" {
  $προσπάθειες = 7
  $βρέθηκε = "ΟΧΙ"
  $αριθμός = στρογγυλοποίηση(τυχ 1,100)
  $μνμ = "Μάντεψε τον αριθμό που σκέφτηκα! (από το 1 ως το 100) Έχεις "
  $μνμ = $μνμ + $προσπάθειες + " προσπάθειες."
  μήνυμα $μνμ
  όσο ($προσπάθειες>0) και ($βρέθηκε=="ΟΧΙ") {
    $απ = ερώτηση "Δώσε αριθμό:"
    αν $απ != $αριθμός {
      $προσπάθειες = $προσπάθειες - 1
      αν $προσπάθειες > 0 {
        αν $απ > $αριθμός {
          $μνμ = $απ + ": Μεγάλος... Έχεις " + $προσπάθειες
          $μνμ = $μνμ + " προσπάθειες ακόμα"
          μήνυμα $μνμ
        } αλλιώς {
          $μνμ = $απ + ": Μικρός... Έχεις " + $προσπάθειες
          $μνμ = $μνμ + " προσπάθειες ακόμα"
          μήνυμα $μνμ
        }
      }
    } αλλιώς {
      $βρέθηκε = "ΝΑΙ"
      $σκορ = $σκορ + 1000 + (στρογγυλοποίηση(τυχ 1,10))*10
    }
  }
  αν $βρέθηκε == "ΟΧΙ" {
    $μνμ = "Λυπάμαι, έχασες. Ο αριθμός ήταν ο " + $αριθμός
  } αλλιώς {
    $μνμ = "ΜΠΡΑΒΟ! Ο αριθμός είναι το " + $αριθμός
  }
  $μνμ = $μνμ + "! Το σκορ σου είναι " + $σκορ
  $μνμ = $μνμ + ". Θέλεις να ξαναπαίξεις; (N/O)"
  $π=ερώτηση $μνμ
}
αν $σκορ == 0 {
  μήνυμα "OK... Γεια σου!"
} αλλιώς {
  μήνυμα "Το σκορ σου έφτασε τους " + $σκορ + " βαθμούς! Ευχαριστώ που παίξαμε!"
}

```

Απλή αριθμομηχανή

```

αρχικοποίησε
μάθε πρόσθεση $χ, $ψ {
    $απ = $χ + $ψ
    μήνυμα "Το άθροισμα "+$χ+"+"+$ψ+" είναι "+$απ
}
μάθε αφαίρεση $χ, $ψ {
    $απ = $χ - $ψ
    μήνυμα "Η διαφορά "+$χ+"-"+$ψ+" είναι "+$απ
}
μάθε πολλαπλασιασμός $χ, $ψ {
    $απ = $χ * $ψ
    μήνυμα "Το γινόμενο "+$χ+"*"+$ψ+" είναι "+$απ
}
μάθε διαίρεση $χ, $ψ {
    αν $ψ != 0 {
        $απ = $χ / $ψ
        μήνυμα "Το ηλίκο "+$χ+"/"+"+$ψ+" είναι "+$απ
    } αλλιώς {
        μήνυμα "Δεν είναι δυνατή η διαίρεση με το 0"
    }
}
$α = ερώτηση "Δώσε τον αριθμό α"
$β = ερώτηση "Δώσε τον αριθμό β"
$πράξη = ερώτηση "Δώσε την πράξη (+-*/)"
αν $πράξη=="+" {
    πρόσθεση $α, $β
} αλλιώς αν $πράξη=="-" {
    αφαίρεση $α, $β
} αλλιώς αν $πράξη=="*" {
    πολλαπλασιασμός $α, $β
} αλλιώς αν $πράξη=="/" {
    διαίρεση $α, $β
} αλλιώς {
    μήνυμα "Δέν έδωσες σωστά την πράξη."
}

```

Το πρόγραμμα αυτό χρησιμοποιεί τέσσερις διαδικασίες, μια για κάθε πράξη, πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση. Αρχικά ζητούνται οι δύο αριθμοί πάνω στους οποίους θα εφαρμοστεί η πράξη, την οποία, καθορίζει ο χρήστης αμέσως μετά.

Στην περίπτωση της διαίρεσης, υπάρχει έλεγχος ούτως ώστε ο δεύτερος αριθμός να μην είναι μηδέν.

Κίνηση, μια απλή εφαρμογή

```

αρχικοποίησε κχ
πήγαινε 10,200
κατεύθυνση 90
μέγεθοςστυλό 20
επανάλαβε 180 {
    καθάρισε
    μπ 2
    περίμενε 0.1
}

```

Η ψευδαίσθηση της κίνησης δημιουργείται μέσω της γρήγορης εναλλαγής εικόνων. Το διπλανό μικρό πρόγραμμα παρουσιάζει ακριβώς αυτό το μηχανισμό: ζωγραφίζει ένα τετράγωνο διαδοχικά σε διπλανές θέσεις, καθαρίζοντας όμως πριν το προηγούμενο. Το τελικό αποτέλεσμα είναι ένα μαύρο τετράγωνο που κινείται οριζόντια πάνω στον καμβά.

Άσκηση:

Τροποποιείστε το πρόγραμμα ούτως ώστε το τετράγωνο να επιστρέφει στην αρχική θέση. Δοκιμάστε να ζωγραφίσετε δύο τοίχους μεταξύ των οποίων θα αναπηδά το τετράγωνο.

Δοκιμάστε την υλοποίηση διαγώνιας κίνησης.

Προχωρημένο παράδειγμα με κίνηση

```

1  αρχικοποίησε κχ
2  πήγαινε 30,200
3  μέγεθοςστυλό 20
4  κατ 90
5  $α = 0
6  $τ = 2
7  όσο 0<1 {
8     $χ = πάρεχ
9     $ψ = πάρεψ
10    αν $χ<=10 {κατ τυχ 20,160 $α=$α+1 $τ=τυχ 1,5}
11    αν $χ>=390 {κατ τυχ 200,340 $α=$α+1 $τ=τυχ 1,5}
12    αν $ψ<=10 {κατ τυχ 110,250 $α=$α+1 $τ=τυχ 1,5}
13    αν $ψ>=390 {κατ τυχ -70,70 $α=$α+1 $τ=τυχ 1,5}
14    καθάρισε μπ $τ περίμενε 0.005
15
16    αν $α == 10 {έξοδος}
17 }

```

σης στο αντίθετο ημιεπίπεδο. Επίσης, καταμετράται ο αριθμός προσκρούσεων ($\alpha = \alpha + 1$). Στη γραμμή 14 ορίζεται μια μικρή καθυστέρηση στην εκτέλεση με την εντολή "περίμενε". Αυτό γίνεται για να μην είναι η κίνηση του τετραγώνου τόσο γρήγορη που να μην προλαβαίνουμε να την αντιληφθούμε.

Παρατηρείστε πως, η συνθήκη "0<1" της εντολής "όσο" στη γραμμή 7 είναι πάντα αληθής. Η έξοδος

Στο παράδειγμα αυτό, ένα μαύρο τετράγωνο 20 εικονοστοιχείων αναπηδά πάνω στα όρια του καμβά, αλλάζοντας συνεχώς κατεύθυνση. Στη μεταβλητή \$α αποθηκεύεται ο αριθμός των αναπηδήσεων στους "τοιίχους" του καμβά. Στη μεταβλητή \$τ ορίζεται η ταχύτητα της κίνησης, η οποία διαφοροποιείται σε κάθε πρόσκρουση, όπως φαίνεται στις γραμμές 10 ως 13, με τυχαίο τρόπο. Αυτές οι γραμμές κώδικα ελέγχουν την πρόσκρουση του τετραγώνου στα όρια του καμβά και ορίζουν μια τυχαία αντίθετη κατεύθυνση κίνησης.

από την επανάληψη ελέγχεται στη γραμμή 16.

Σημείωση: Η παρακολούθηση της εκτέλεσης του προγράμματος μέσα από τον επεξεργαστή εντολών καθυστερεί σε σημαντικό βαθμό την εκτέλεση του προγράμματος. Για να επιτύχετε πλήρη ταχύτητα εκτέλεσης και καλύτερο οπτικό αποτέλεσμα, απενεργοποιήστε από το μενού προβολή τον επεξεργαστή εντολών και τα υπόλοιπα στοιχεία.

Άσκηση: Τροποποιήστε το παραπάνω πρόγραμμα ούτως ώστε το τετράγωνο που αναπηδά πάνω σε μια πλευρά του καμβά να απομακρύνεται με την ίδια γωνία με την οποία προσκρούει, αντί για μια τυχαία, όπως συμβαίνει στην παραπάνω υλοποίηση.

Δοκιμάστε να κάνετε το τετράγωνο να αλλάζει σε τυχαία κατεύθυνση την κίνησή του κάθε τέσσερις προσκρούσεις.

ΑΝΑΚΕΦΑΛΑΙΩΣΗ



Στη ζωή μας πολλές φορές καλούμαστε να λύσουμε πολλά και ποικίλα προβλήματα. Μερικά από αυτά τα προβλήματα μπορούμε να τα λύσουμε και με τη βοήθεια υπολογιστή. Η περιγραφή της λύσης ενός προβλήματος με λογικά βήματα ονομάζεται «**αλγόριθμος**». Για να υλοποιήσουμε έναν αλγόριθμο στον υπολογιστή, πρέπει να τον γράψουμε με μια γλώσσα προγραμματισμού. Η συγγραφή ενός προγράμματος σε μια γλώσσα προγραμματισμού γίνεται συνή-

θως σε ειδικά περιβάλλοντα που μας διευκολύνουν στη σύνταξή του. Επίσης μετατρέπουν τον **κώδικα** με τον οποίο γράφουμε σε μορφή κατάλληλη, ώστε να τον εκτελέσει ο υπολογιστής.

Υπάρχουν πολυάριθμες **γλώσσες προγραμματισμού**. Η καθεμία έχει δικά της χαρακτηριστικά (αλφάβητο, λεξιλόγιο, συντακτικό) και δυνατότητες. Η επιλογή της κατάλληλης γλώσσας προγραμματισμού εξαρτάται σε μεγάλο βαθμό από τις λειτουργίες του προγράμματος που σχεδιάζουμε.

Στις σημειώσεις αυτές παρουσιάζεται η γλώσσα προγραμματισμού Logo και χρησιμοποιείται το περιβάλλον προγραμματισμού Kturtle.



ΑΣΚΗΣΕΙΣ ΑΥΤΟ-ΑΞΙΟΛΟΓΗΣΗΣ

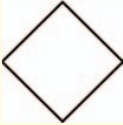

- Χαρακτηρίστε τις παρακάτω προτάσεις ως σωστές ή λάθος βάζοντας δίπλα στα αντίστοιχα κελιά Σ ή Λ. Στην περίπτωση που πιστεύετε ότι είναι λάθος σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή απάντηση.

	Προτάσεις Σωστού-Λάθους	Σ ή Λ
1	Ένα πρόβλημα μπορεί να λυθεί πάντα με μαθηματικούς υπολογισμούς.	
2	Η επίλυση ενός προβλήματος προηγείται της κατανόησής του.	
3	Πρέπει να καθορίσουμε τα ζητούμενα ενός προβλήματος, για να μπορέσουμε να το επιλύσουμε.	
4	Όλα τα προβλήματα έχουν λύση.	
5	Ένας αλγόριθμος πρέπει πάντοτε να «εξασφαλίζει» το ότι θα τερματίσει.	
6	Η εντολή «Πες ένα αστείο» είναι αυστηρά καθορισμένη.	
7	Ένα πρόγραμμα είναι η γραφή ενός αλγορίθμου σε μια γλώσσα προγραμματισμού.	
8	Υπάρχουν πολλές διαφορετικές γλώσσες, για να προγραμματίσουμε έναν υπολογιστή.	
9	Ο μεταφραστής βρίσκει τα λογικά λάθη ενός προγράμματος.	
10	Η γλώσσα που καταλαβαίνει ο υπολογιστής είναι η γλώσσα μηχανής.	

- Σχεδιάσε το αποτέλεσμα που προκύπτει από τις παρακάτω εντολές στο περιβάλλον του Kturtle.

επανάλαβε 2 {μπ 30 σδ 90} μπ 30 σα 90	
επανάλαβε 4 {επανάλαβε 2 {μπ 30 σδ 90} μπ 30 σα 90}	

- Αντιστοιχίστε τα σχήματα στα δεξιά με τα τμήματα του κώδικα στα αριστερά.

Α. σδ 45 μπ 100 σδ 90 μπ 100 σα 90 μπ 100 σδ 90 μπ 100 σα 90	1. 
Β. σδ 45 μπ 100 σδ 90 μπ 100 μπ 100 σα 90 μπ 100	2. 

Γ. σδ 45
μπ 100 σδ 90
μπ 100 σδ 90
μπ 100 σδ 90
μπ 100

3.



ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ

Κεφάλαιο 1: Εισαγωγή στην Έννοια του Αλγορίθμου και στον Προγραμματισμό

1. Σε ποια απλούστερα προβλήματα μπορεί να χωριστεί το πρόβλημα των μαθητικών εκλογών; Σε τι μας βοηθάει η ανάλυση του προβλήματος σε επιμέρους προβλήματα; Ποιος είναι ο χώρος του προβλήματος;
2. Να αναλύσετε το πρόβλημα «αγορά Η/Υ» σε απλούστερα προβλήματα.
3. Προσπαθήστε να επιλύσετε το παρακάτω πρόβλημα: Έστω ότι δεν διαθέτουμε ρολόι αλλά δύο κλεψύδρες, των 7 λεπτών η μία και των 4 λεπτών η άλλη. Πώς μπορούμε να ξέρουμε πότε ακριβώς πέρασαν 9 λεπτά;
4. Φτιάξτε ένα διάγραμμα με τα βήματα που πρέπει να ακολουθήσουμε, για να κατασκευάσουμε ένα βαρκάκι ή ένα αεροπλανάκι με ένα τετράγωνο χαρτί.
5. Να γραφεί ένας αλγόριθμος που να περιγράφει σε κάποιον που δεν ξέρει, πως μπορεί να κάνει ποδήλατο. Τι πρόβλημα μπορεί να έχει αυτός ο αλγόριθμος;
6. Να γραφεί ένας αλγόριθμος που να περιγράφει σε ένα μικρό παιδί πώς να σχηματίσει ένα ορθογώνιο παραλληλόγραμμο με τα βήματά του στην άμμο.
7. Στον παρακάτω αλγόριθμο:
Δώσε μου το έτος που έχουμε σήμερα.
Δώσε μου το έτος που γεννήθηκες.
Η ηλικία σου υπολογίζεται με το άθροισμα του έτους που γεννήθηκες και του έτους που έχουμε σήμερα.
Εμφάνιση της ηλικίας.
 Πώς θα χαρακτηρίζαμε το λάθος που υπάρχει;

Κεφάλαιο 2: Ο Προγραμματισμός στην Πράξη

8. Σχεδιάστε μια σκακιέρα, όπως το σχήμα που ακολουθεί, στο περιβάλλον MicroWorlds Pro ή σε κάποιο άλλο περιβάλλον της γλώσσας Logo.

Προαπαιτούμενες γνώσεις

Εντολές: **μπροστά** (μπ), **πίσω** (πι), **στρίψεδεξιά** (σδ), **στρίψεαριστερά** (σα), **επανάλαβε**.

Χρήση διαδικασιών. Κάλεσμα διαδικασιών μέσα σε άλλες διαδικασίες.

Νέες εντολές: χρώμαστυλό (χσ), μέγεθοςστυλό (μσ)

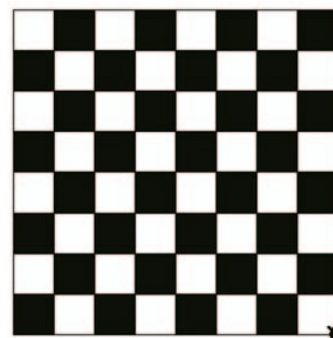
Η εντολή «χρώμαστυλό» αλλάζει το χρώμα με το οποίο γράφει η χελώνα. Η εντολή χρησιμοποιεί το μοντέλο RGB (Red, Green, Blue - Κόκκινο, Πράσινο, Μπλε) για να ορίσει το χρώμα.

Χρησιμοποιούνται τρεις αριθμοί, χωρισμένοι με κόμμα, από τους οποίους ο πρώτος αντιστοιχεί στο κόκκινο χρώμα, ο δεύτερος στο πράσινο και ο τρίτος στο μπλε. Π.χ.: χσ 255,0,0

Το μαύρο χρώμα αντιστοιχεί στο συνδυασμό 0,0,0 ενώ το άσπρο στο 255,255,255. Το 255 είναι ο μέγιστος αριθμός που μπορεί να τεθεί. Ενδιάμεσες τιμές θα δώσουν διάφορες αποχρώσεις του γκρι. Για να πάρουμε τα υπόλοιπα χρώματα δίνουμε διαφορετικές τιμές σε κάθε αριθμό της τριάδας. Π.χ.: χσ 255,0,0 - κόκκινο, χσ 0,255,0 - πράσινο, χσ 0,0,255 - μπλε, κλπ.

Η εντολή «μέγεθοςστυλό» μεταβάλλει το μέγεθος του ίχνους που αφήνει η χελώνα. Μετρίεται σε εικονοστοιχεία. Το σχήμα του αποτυπώματος που αφήνει το στυλό της χελώνας είναι πάντοτε τετράγωνο.

Π.χ.: μσ 5





ΘΕΜΑΤΑ ΓΙΑ ΣΥΖΗΤΗΣΗ

1. Να αναφέρετε, από τις εγκυκλοπαιδικές σας γνώσεις, μερικά προβλήματα που παραμένουν άλυτα ή που έχει αποδειχτεί ότι δεν έχουν λύση.
2. Γιατί ναμίζετε ότι υπάρχουν πολλές γλώσσες προγραμματισμού; Θα μπορούσαν να αντικατασταθούν όλες οι γλώσσες προγραμματισμού με μία;