

## ΚΕΦΑΛΑΙΟ 2

# Ο Προγραμματισμός στην Πράξη

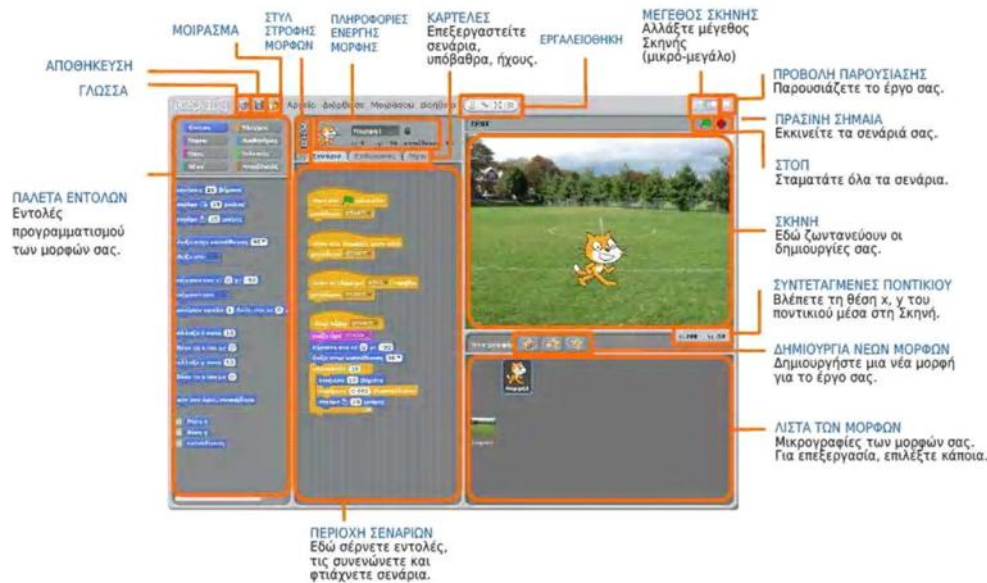
### Εισαγωγή

Το Scratch είναι μια νέα γλώσσα προγραμματισμού που μας δίνει τη δυνατότητα να δημιουργήσουμε διαδραστικές ιστορίες, παιχνίδια και κινούμενα σχέδια, καθώς και να τις μοιραστούμε με άλλους στο διαδίκτυο. Στον ιστοχώρο του Scratch υπάρχουν και πολλές άλλες πηγές μάθησης και υποστήριξης όπως: εκπαιδευτικά βίντεο, κάρτες Scratch και συχνές ερωτήσεις (<http://info.scratch.mit.edu/Support/>). Ο παρών Οδηγός Χρήσης αφορά την έκδοση 1.4 (Ιούλιος 2009) του Scratch. Νεότερες εκδόσεις διατίθενται στη διεύθυνση: <http://info.scratch.mit.edu/Support/>.

Τα έργα (project) στο Scratch οικοδομούνται από αντικείμενα που λέγονται **μορφές** (sprite). Μπορούμε να αλλάξουμε την εμφάνιση μιας μορφής δίνοντάς της μια διαφορετική **ενδυμασία** (costume). Μπορούμε να την κάνουμε να μοιάζει με άνθρωπο, με τρένο, με πεταλούδα ή με οτιδήποτε άλλο. Για ενδυμασία μπορούμε να χρησιμοποιήσουμε οποιαδήποτε εικόνα: μπορούμε να δημιουργήσουμε μία εικόνα στον Επεξεργαστή Ζωγραφικής, να εισάγουμε μια από τον σκληρό μας δίσκο ή μία από το διαδίκτυο. Μπορούμε να δώσουμε οδηγίες σε μια μορφή ώστε να κινηθεί, να παίξει μουσική ή να αλληλεπιδράσει με άλλες μορφές. Για να πούμε στη μορφή τι να κάνει, συνενώνουμε εικονικές **εντολές** (που μοιάζουν με τουβλάκια - block) μεταξύ τους σε στήλες που ονομάζονται **σενάρια ενεργειών** (κώδικες, script). Όταν κάνουμε κλικ σε ένα σενάριο, το Scratch «τρέχει» τις εντολές από την κορυφή του σεναρίου έως τον πάτο.

## 2.1 Το περιβάλλον προγραμματισμού Scratch

Την πρώτη φορά που παρατηρούμε το περιβάλλον προγραμματισμού του Scratch βλέπουμε ότι η οθόνη χωρίζεται σε τέσσερις περιοχές: *Σκηνή*, *Περιοχή σεναρίων*, *Παλέτα εντολών* και *Λίστα των μορφών* (Εικόνα 2.1).



Εικόνα 2.1. Το περιβάλλον του Scratch

Η *Σκηνή* είναι ο χώρος όπου ζωντανεύουν οι ιστορίες μας, τα παιχνίδια και τα κινούμενα σχέδια. Οι μορφές κινούνται και αλληλεπιδρούν μεταξύ τους επάνω στη *Σκηνή*. Η *Σκηνή* έχει μήκος 480 μονάδες και ύψος 360 μονάδες. Είναι χωρισμένη σε άξονες  $x$  και  $y$ . Το κέντρο της *Σκηνής* έχει συντεταγμένες  $x:0$  και  $y:0$ .

Η *Λίστα των μορφών* παρουσιάζει μικρογραφίες όλων των μορφών του έργου. Το όνομα της κάθε μορφής εμφανίζεται κάτω από τη μικρογραφία της. Για να δούμε ή να επεξεργαστούμε τα σενάρια, τις ενδυμασίες και τους ήχους μιας μορφής, κάνουμε κλικ πάνω στη μικρογραφία της στη *Λίστα των μορφών* ή κάνουμε διπλό κλικ πάνω στην ίδια τη μορφή μέσα στη *Σκηνή*. (Η επιλεγμένη μορφή είναι μαρκαρισμένη με μπλε περίγραμμα μέσα στη *Λίστα των μορφών*)



Για να προγραμματίσουμε μια μορφή, σέρνουμε εντολές (τουβλάκια) από την *Παλέτα εντολών* προς την *Περιοχή σεναρίων*. Για να εκτελεστεί μια εντολή, κάνουμε κλικ επάνω της. Δημιουργούμε σενάρια ενεργειών (προγράμματα) συνενώνοντας εντολές μεταξύ τους σε στήλες και κάνουμε κλικ οπουδήποτε στη στήλη για να εκτελεστεί ολόκληρο το σενάριο, από την αρχή ως το τέλος. Για να καταλάβουμε τι ακριβώς κάνει μια εντολή, κάνουμε δεξί κλικ πάνω της και επιλέγουμε τη βοήθεια από το αναδυόμενο μενού. Όταν σέρνουμε μία εντολή μέσα στη *Περιοχή σεναρίων*, μία λευκή υπογράμμιση υποδεικνύει πού μπορούμε να την αφήσουμε ώστε να δημιουργήσει μια σωστή ένωση με άλλη εντολή.






## 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 διαδεδομένες γλώσσες προγραμματισμού έχουν σχεδιαστεί, ώστε να μπορούμε να εκτελούμε αριθμητικές πράξεις.

Όπως αναφέρθηκε στο πρόβλημα της εκδρομής (1<sup>ο</sup> 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 Scratch και να μας εμφανίσει το αποτέλεσμα της διαίρεσης;


Η κατάλληλη εντολή είναι η . Η εντολή αυτή εκτελεί την πράξη  $200 : 25$  και εμφανίζει το αποτέλεσμα στη Σκηνή. Η εντολή  είναι μία εντολή εξόδου, καθώς έχει ως αποτέλεσμα την εμφάνιση ενός αριθμού ή ενός μηνύματος στην οθόνη του υπολογιστή.

Ο υπολογιστής μπορεί να κάνει όλες τις αριθμητικές πράξεις. Για τα σύμβολα των πράξεων χρησιμοποιούμε τα σύμβολα που υπάρχουν στην Παλέτα εντολών  (στα αριστερά της Επιφάνειας εργασίας του Scratch):  για πρόσθεση,  για αφαίρεση,  για πολλαπλασιασμό και  για διαίρεση.

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
πες 	<input type="text"/>
πες 	<input type="text"/>
πες 	<input type="text"/>
πες 	<input type="text"/>

α. πες  $(12 / 2) * 3$







β. πες  $12 / (2 * 3)$



1. Ποιο είναι το αποτέλεσμα στις περιπτώσεις α) \_\_\_\_\_ και β) \_\_\_\_\_ ;
2. Με ποια σειρά εκτελέστηκαν οι πράξεις στις δύο αυτές εντολές;

### Η Εμφάνιση Μηνυμάτων

Η εντολή  επιτρέπει, εκτός από αριθμούς, να εμφανίζεται στη Σκηνή και κάποιο μήνυμα. Αν, για παράδειγμα, θέλουμε να εμφανίσουμε το όνομα μας, τότε μπορούμε να γράψουμε .

Αν μετά την εντολή  γράψουμε την αριθμητική έκφραση χωρίς να χρησιμοποιήσουμε τελεστές, τότε η εκτέλεση της εντολής θα έχει ως αποτέλεσμα την εμφάνιση της έκφρασης όπως εμφανίζεται μέσα στο πλαίσιο. Η εντολή  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. Αν θέλουμε να εμφανίσουμε το μήνυ-

μα «Το όνομα μου είναι Στέλιος», τότε πρέπει να γράψουμε `πες Το όνομά μου είναι Στέλιος`. Ο υπολογιστής θα εμφανίσει όλες τις λέξεις που περιλαμβάνονται στο πλαίσιο. 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 δυο μηνύματα μεταξύ τους, τότε πρέπει να χρησιμοποιήσουμε την εντολή `ένωσε`.

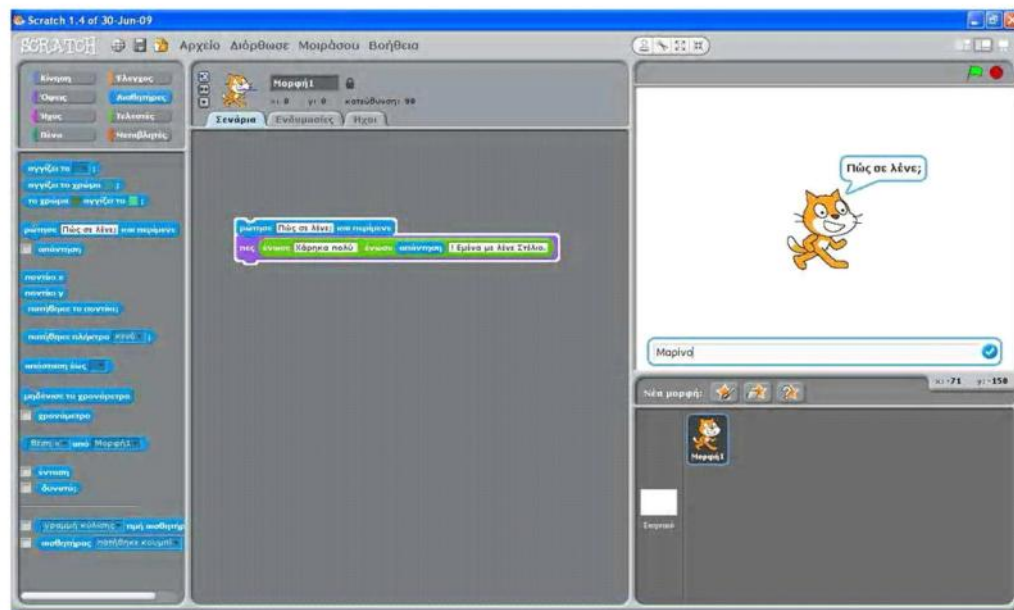
Δοκιμάστε την εντολή `πες ένωσε Το κόστος της εκδρομής ανά μαθητή είναι ένωσε 200 / 25 ευρώ.`. Τι εμφανίζεται στην οθόνη;

### 2.3 Συνομιλία με τον υπολογιστή. Περισσότερα για τις εντολές εισόδου-εξόδου

Στις προηγούμενες παραγράφους είχαμε την ευκαιρία να γνωρίσουμε την εντολή εξόδου `πες`, με την οποία εμφανίζουμε δεδομένα στη Σκηνή της Επιφάνειας εργασίας του Scratch. Το περιβάλλον προγραμματισμού Scratch μας δίνει τη δυνατότητα να πραγματοποιούμε διαλόγους χρησιμοποιώντας ερωταποκρίσεις.

Ερωτήσεις μπορούμε να κάνουμε με την εντολή `ρώτησε` και `και περίμενε` από την Παλέτα εντολών `Αισθητήρες` και στην κενή περιοχή που εμφανίζεται μπορούμε να δώσουμε μίαν απάντηση. Το παράθυρο της ερώτησης `ρώτησε Πώς σε λένε; και περίμενε` φαίνεται στην παρακάτω

εικόνα. Το πλαίσιο  χρησιμοποιείται, για να πληκτρολογήσουμε την απάντηση μας.



Η εντολή **ρώτησε** και **περίμενε** είναι μια εντολή εισόδου, γιατί μας επιτρέπει να δώσουμε μία τιμή (για μία λέξη, ένα σύνολο λέξεων, δηλαδή μια λίστα, 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 και μπορούμε να την ανακτήσουμε χρησιμοποιώντας τη λέξη **απάντηση**, όπως στο επόμενο παράδειγμα.

**πες** **ένωσε** **Κάρηκα πολύ** **ένωσε** **απάντηση** **! Εμένα με λένε Στέλιο.**

Για να καταλάβετε καλύτερα τη χρήση των εντολών **ρώτησε** και **περίμενε** και **απάντηση**, φτιάξτε τις δικές σας συνομιλίες.

#### **Δραστηριότητα:** Ας πειραματιστούμε λίγο και με τους αριθμούς

1. Τι ακριβώς κάνουν οι δυο παρακάτω εντολές:

**ρώτησε** **Δώσε μου τον αριθμό που θέλεις να υψώσεις στο τετράγωνο.** και **περίμενε**

**πες** **απάντηση** \* **απάντηση**


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 Enter μετά την πληκτρολόγηση του αριθμού.
3. Πώς μπορούν να τροποποιηθούν οι παραπάνω εντολές, ώστε να υπολογίζουμε τον κύβο ενός αριθμού;

## 2.4 Η γλώσσα Scratch και ο σχεδιασμός γεωμετρικών σχημάτων

Κάνοντας τις πρώτες δοκιμές με τη γάτα

Οι μορφές είναι ίσως το πιο βασικό χαρακτηριστικό της γλώσσας Scratch. Για να δημιουργήσουμε μια μορφή στη **Σκηνή**, χρησιμοποιούμε τα εικονίδια με την ετικέτα «Νέα Μορφή»:



Πολλές εντολές στη γλώσσα Scratch μετακινούν και χειρίζονται τις μορφές στη **Σκηνή**. Η μορφή της γάτας που εμφανίζεται είναι αυτή που εικονίζεται στο πλάι: 

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

- **κινήσου 10 βήματα**: Με την εκτέλεση της εντολής αυτής η γάτα προχωράει μπροστά τόσα βήματα όσα έχουμε ορίσει (στο συγκεκριμένο παράδειγμα 10). Για να κινηθεί η γάτα προς τα πίσω αρκεί να εισάγουμε έναν αρνητικό αριθμό, π.χ. **κινήσου -50 βήματα**
- **στρίψε 15 μοίρες**: Η γάτα στρίβει προς τα δεξιά τόσες μοίρες όσες έχουμε ορίσει (στο παράδειγμα 15).
- **στρίψε 60 μοίρες**: Η γάτα στρίβει προς τα αριστερά τόσες μοίρες όσες έχουμε ορίσει (στην περίπτωση μας 60).
- **κατέβασε πένα**: Δίνει εντολή στη γάτα να αφήνει ίχνος από κάθε σημείο της οθόνης που περνάει. Αν δεν έχουμε δώσει στην αρχή αυτή την εντολή, η γάτα μετακινείται με τις κα-

#### **Εισαγωγική Δραστηριότητα**

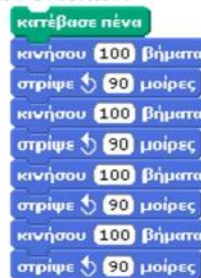
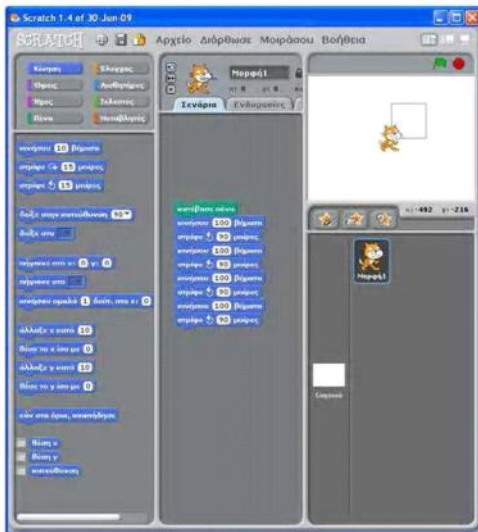
Τοποθετήστε ένα ψάρι στην **Σκηνή** επιλέγοντας το εικονίδιο «**Διάλεξε νέα μορφή από αρχείο**». Πειραματιστείτε μετακινώντας το ψάρι στη **Σκηνή** και δοκιμάστε τις διπλανές εντολές στην **Περιοχή Σεναρίων**. Στη συνέχεια προσπαθήστε να δημιουργήσετε ένα ευθύγραμμο τμήμα μήκους 100 βημάτων.

τάλληλες εντολές στην οθόνη, χωρίς να σχεδιάζει τίποτε.

- **σηκώσε πένα**: Δίνει εντολή στη γάτα να σταματήσει να αφήνει ίχνος καθώς προχωράει.
- **καθάρισε**: Σβήνει τα σχέδια που έχουμε δημιουργήσει στη σκηνή
- **πήγαινε στο x: 0 y: 0**: Μεταφέρει τη γάτα στο κέντρο της σκηνής
- **δείξε στην κατεύθυνση 90**: Στρέφει τη γάτα προς την αρχική κατεύθυνση (προς τα δεξιά).

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 βήματα, δίνοντας εντολές στη γάτα.

Όπως βλέπουμε και στη διπλανή εικόνα ο συνδυασμός των εντολών:



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

### Δομή Επανάληψης

Αν μελετήσουμε καλύτερα το παραπάνω πρόγραμμα του τετραγώνου, παρατηρούμε ότι

οι εντολές **κινήσου 100 βήματα** και **στρίψε 90 μοίρες** επαναλήφθηκαν τέσσερις

φορές με την ίδια σειρά. Θα μπορούσαμε να έχουμε το ίδιο αποτέλεσμα ομαδοποιώντας τις δυο εντολές και δίνοντας μια εντολή που να τις επαναλαμβάνει τέσσερις φορές. Η εντολή



αυτή είναι:

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

Εικόνα 2.3. Δημιουργία ενός τετραγώνου με τη βοήθεια της γάτας.

**Ερώτηση**  
? Ποιο αποτέλεσμα θα προέκυπτε, αν δε γράφαμε την εντολή **κατέβασε πένα** στην αρχή;

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

### Δραστηριότητες:

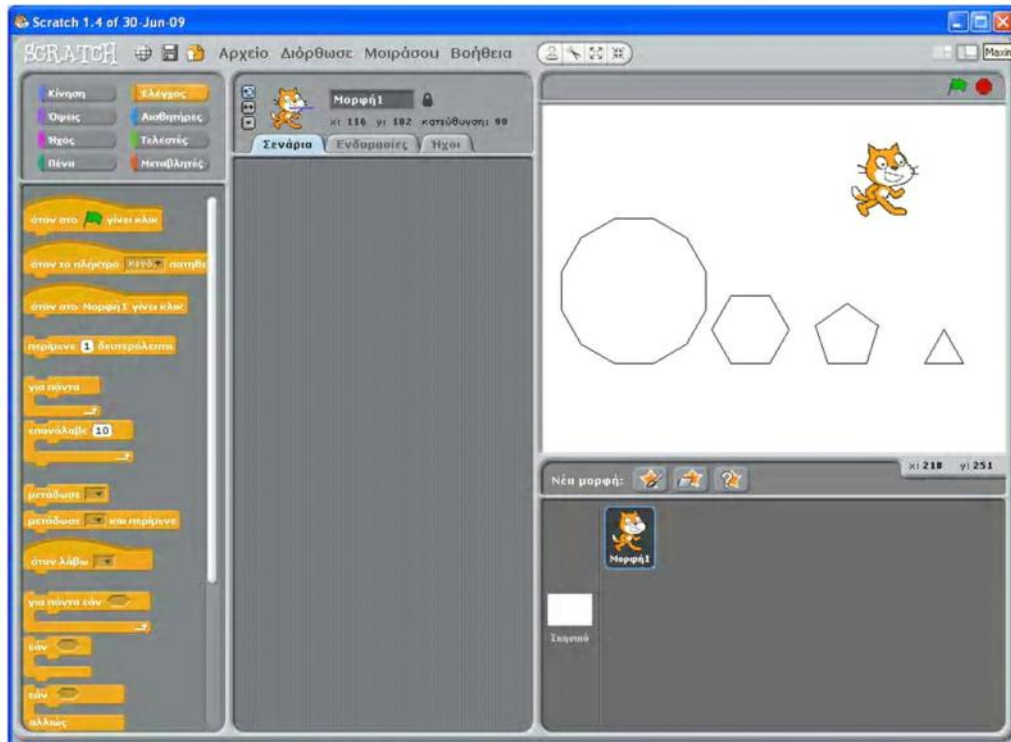
1. Να γράψετε την κατάλληλη εντολή, ώστε να εμφανιστεί το όνομά σας 200 φορές στην οθόνη του υπολογιστή:
2. Στην εντολή που χρησιμοποιήσαμε, για να σχεδιάσουμε ένα τετράγωνο:



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 γάτα, ώστε να σχεδιάσει ένα ισόπλευρο τρίγωνο, ένα πεντάγωνο, ένα εξάγωνο ή ένα δωδεκάγωνο, όπως τα σχήματα της Εικόνας 2.4.

3. Με τι μοιάζει το σχήμα που δημιουργεί η επόμενη ομάδα εντολών;



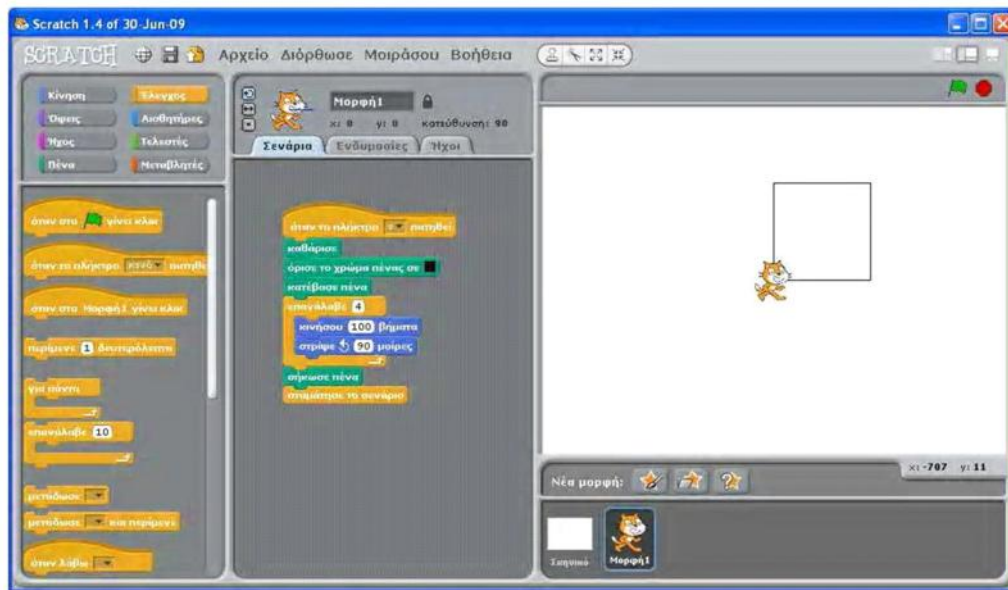
Εικόνα 2.4. Δημιουργία γεωμετρικών σχημάτων με τη γάτα της γλώσσας Scratch.

## 2.5 Δημιουργώντας Σενάρια

Εντολές που επιθυμούμε να εκτελεστούν με το πάτημα ενός πλήκτρου μπορούν να ομαδοποιηθούν. Η ομαδοποίηση αυτή των εντολών καλείται **Σενάριο**. Η εκτέλεση ενός σεναρίου μπορεί να χρησιμεύσει στον υλοποίηση μίας πιο σύνθετης εφαρμογής, στη βελτίωση της επικοινωνίας με το χρήστη ή στη διαδραστικότητα μεταξύ δύο ή περισσότερων μορφών. Το πλεονέκτημα του σεναρίου είναι ότι μπορούμε να το εκτελέσουμε με το πάτημα ενός πλήκτρου όποτε το χρειαστούμε.

Για τη δημιουργία ενός σεναρίου εισάγουμε ως πρώτη την εντολή **όταν το πλήκτρο ΚΕΝΟ πατηθεί** από την ομάδα **Έλεγχος** επιλέγοντας το πλήκτρο που μας εξυπηρετεί:

Στο επόμενο παράδειγμα (Εικόνα 2.5) έχουμε γράψει ένα παράδειγμα ενός σεναρίου, με το οποίο κάθε φορά που πιέζεται το πλήκτρο **s**, η γάτα σχεδιάζει ένα τετράγωνο πλευράς 100.



Εικόνα 2.5. Το σενάριο **s** έχει δημιουργήσει ένα τετράγωνο

Κάθε φορά που πιέζουμε το γράμμα **s** στο πληκτρολόγιο, θα σχηματίζεται ένα τετράγωνο. Ουσιαστικά με αυτόν τον τρόπο η γλώσσα Scratch μας επιτρέπει να δημιουργούμε τα δικά μας σενάρια-εντολές.

### 1<sup>η</sup> Δραστηριότητα

Δημιουργήστε δύο σενάρια: ένα που θα σχεδιάζει ένα τρίγωνο με το πάτημα του πλήκτρου **t** και ένα που θα σχεδιάζει ένα τετράγωνο με το πάτημα του πλήκτρου **s**. Χρησιμοποιώντας τα νέα σενάρια που μόλις δημιουργήσατε, προσπαθήστε να δημιουργήσετε ένα σπίτι, όπως το διπλανό σχήμα.

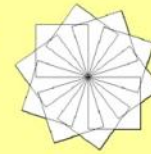
Στη συνέχεια δημιουργήστε ένα σενάριο που θα σχεδιάζει με το πάτημα του πλήκτρου **v** πολλά σπίτια σε τυχαίες θέσεις.





## 2<sup>η</sup> Δραστηριότητα

Προσπαθήστε να περιγράψετε την πορεία της γάτας ακολουθώντας τις παρακάτω εντολές. Το αποτέλεσμα των εντολών φαίνεται στην παρακάτω εικόνα. Αφού έχετε κατανοήσει τη σημασία των αριθμών 10 και 36 στις εντολές, προσπαθήστε να τους αλλάξετε δημιουργώντας τα δικά σας σχήματα.



### 2.6 Μεταβλητές

Σύμφωνα με το παραπάνω σενάριο (Εικόνα 2.5), όταν θέλουμε να δημιουργούμε τετράγωνο με διαφορετικό μήκος πλευράς, πρέπει να επεμβαίνουμε κάθε φορά στην εντολή **κινήσου 50 βήματα** και να αλλάζουμε το μήκος της πλευράς. Έτσι, αν θέλουμε να σχεδιάσουμε ένα τετράγωνο πλευράς 50 βημάτων, θα δώσουμε τις εντολές:



ενώ, αν θέλουμε ένα τετράγωνο πλευράς 150, θα δώσουμε τις εντολές:



Μπορούμε να χρησιμοποιούμε το ίδιο πάντα σενάριο για το σχεδιασμό τετραγώνων διαφορετικών πλευρών; Η απάντηση είναι καταφατική. Το σενάριο μπορούμε να το τροποποιήσουμε ως εξής:



Αφού έχουμε γράψει το παραπάνω σενάριο, μπορούμε να σχεδιάσουμε ένα τετράγωνο πλευράς 50 δίνοντας, στο ίδιο ή σε κάποιο άλλο σενάριο, την εντολή **όρισε το μήκος σε 50**

Δηλαδή, πριν την εκτέλεση του σεναρίου *s* ορίζουμε το επιθυμητό μήκος της πλευράς του τετραγώνου. Η τιμή 50 αποθηκεύεται προσωρινά στο **μήκος** και η εντολή **κινήσου μήκος βήματα** μετακινεί τη γάτα μπροστά κατά 50 βήματα. Κάθε φορά που εκτελούμε το σενάριο *s*, στο **μήκος** αποθηκεύεται προσωρινά μία διαφορετική τιμή. Το **μήκος** ονομάζεται **μεταβλητή**.



### Ερώτηση

Πώς θα ενεργοποιήσουμε το σενάριο s, 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.

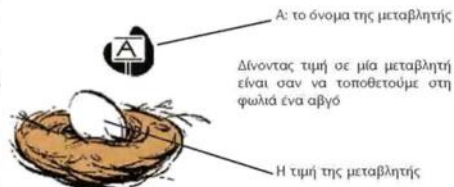
Στη γλώσσα Scratch, για να αναφερθούμε στην τιμή της μεταβλητής, χρησιμοποιούμε το «τουβλάκι» με το όνομα της μεταβλητής, π.χ. **μήκος**. Για να δημιουργήσουμε μία νέα μεταβλητή, κάνουμε κλικ στην εντολή **Δημιούργησε μια μεταβλητή** από την ομάδα **Μεταβλητές**. Για να δώσουμε (εκχωρήσουμε) τιμή σε μία μεταβλητή, μπορούμε να χρησιμοποιήσουμε την εντολή **όρισε το μήκος σε 50**, όπως αναφέρθηκε και στο παραπάνω παράδειγμα, επιλέγοντας με το βελάκι το όνομα της μεταβλητής που μας ενδιαφέρει και συμπληρώνοντας στο λευκό πλαίσιο την τιμή που θέλουμε να αποθηκευτεί. Αν θέλουμε, για παράδειγμα, να δώσουμε στη μεταβλητή με όνομα **x** την τιμή 2 γράφουμε **όρισε το x σε 2**, ενώ αν θέλουμε να δώσουμε την τιμή Γάτα γράφουμε **όρισε το x σε Γάτα**.

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

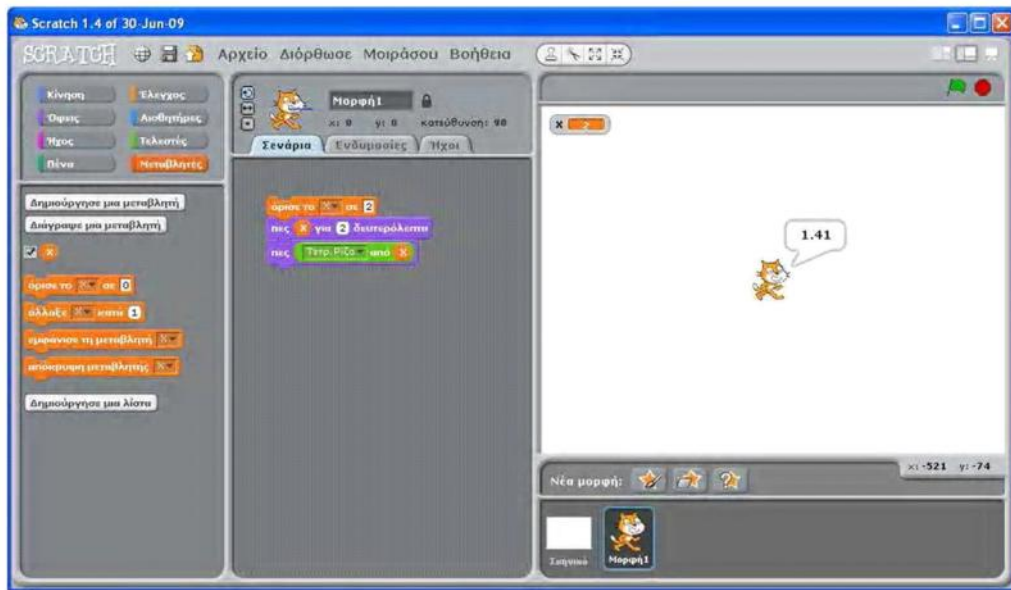
Ποια εντολή θα χρησιμοποιήσουμε, για να εμφανίσουμε την τιμή που περιέχει η μεταβλητή **x**;

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

Η απάντηση φαίνεται στην παρακάτω Εικόνα 2.6.



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.



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

### Δραστηριότητες

1. α) Προσπαθήστε να δώσετε το όνομά σας σε μία μεταβλητή ΟΝΟΜΑ και στη συνέχεια εμφανίστε το στη Σκηνή.  
 β) Προσπαθήστε να εμφανίσετε στην οθόνη το όνομά σας, χωρίς να το ξαναγράψετε με το συνοδευτικό μήνυμα «Το όνομα μου είναι ...».
2. Γράψτε δίπλα από τις εντολές εξόδου τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των εντολών;

A. **όρισε το ζώο σε λιοντάρι**

```

πες ζώο
πες λιοντάρι
πες ζώο
όρισε το ζώο σε σκύλο
πες ένωσε έχω ένα ζώο

```

B.

```

όρισε το X σε 3
πες 12 + 5 * X
πες 2 * 5 - X * 4
όρισε το X σε 8
πες 14 + 2 + X / 2

```

3. Γράψτε και εκτελέστε τις παρακάτω εντολές:

```

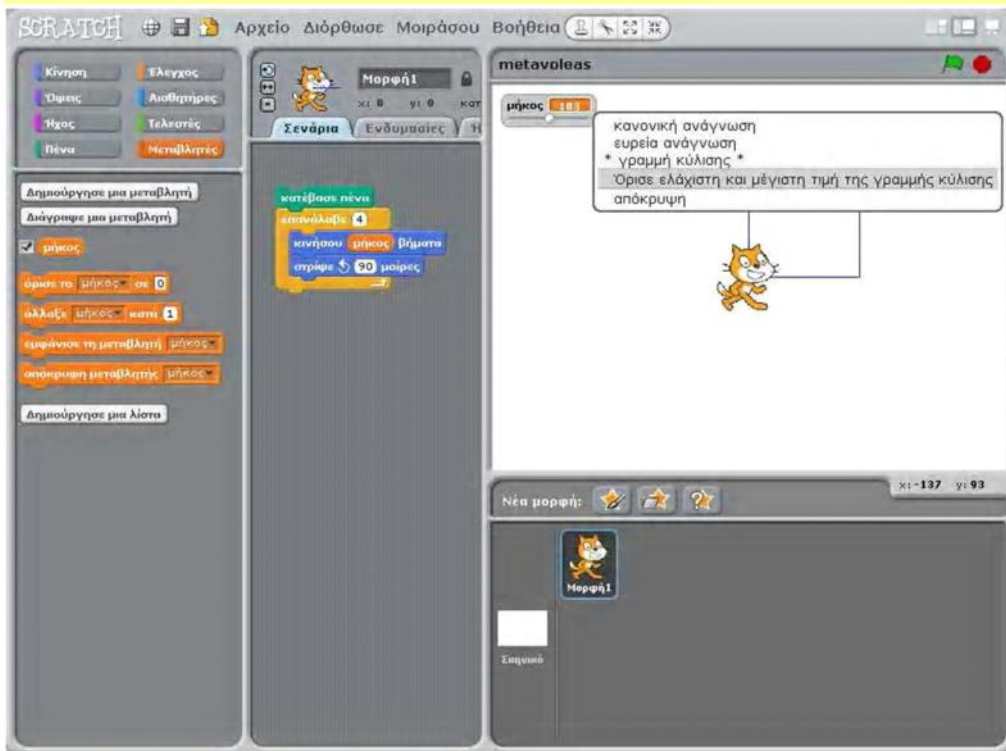
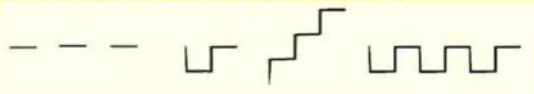
όρισε το α σε 1
πες α για 1 δευτερόλεπτα
επανάλαβε 9
  άλλαξε α κατά 1
  πες α για 1 δευτερόλεπτα

```

Ποιο είναι το αποτέλεσμα της εκτέλεσης της διαδικασίας;

Βρείτε ποια είναι η λειτουργία της εντολής `άλλαξε α κατά 1`, ώστε να μπορείτε να τη χρησιμοποιήσετε και στις επόμενες ασκήσεις.

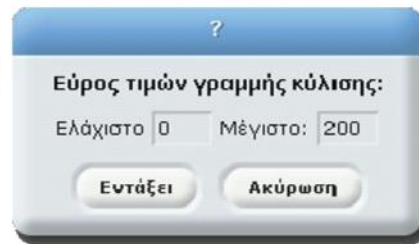
4. Να δημιουργήσετε μία διαδικασία που να κατασκευάζει ένα ορθογώνιο παραλληλόγραμμο δίνοντάς του τα εκάστοτε μήκη των πλευρών.
5. Να καταγραφεί η διαδικασία που να υπολογίζει το εμβαδόν ενός τριγώνου πλευράς  $a$  και ύψους  $u$ .
6. Δημιουργήστε τα 4 παρακάτω σχήματα



Εικόνα 2.7. Το σενάριο τετράγωνο παίρνει τιμές από το μεταβολέα «μήκος».

### Μία παραλλαγή του σεναρίου τετράγωνο ή παίζοντας με τη γραμμή κύλισης μεταβλητών

Αντί να δίνουμε κάθε φορά το μήκος της πλευράς στο σενάριο  $s$  («τετράγωνο»), θα μπορούσαμε να χρησιμοποιήσουμε τη γραμμή κύλισης μίας μεταβλητής (Εικόνα 2.7). Κάνουμε δεξί κλικ στην ένδειξη της μεταβλητής επάνω στη Σκηνή και επιλέγουμε *γραμμή κύλισης* για να εμφανιστεί ο μεταβολέας. Για να αλλάξουμε τα όρια των τιμών που παίρνει ο μεταβολέας, ξανακάνουμε δεξί κλικ, επιλέγουμε *Όρισε ελάχιστη και μέγιστη τιμή της γραμμής κύλισης* και εμφανίζεται στην οθόνη μας η Εικόνα 2.8.



Εικόνα 2.8. Παράθυρο καθορισμού τιμών του μεταβολέα.

Αν εκτελέσουμε το σενάριο «τετράγωνο», η μεταβλητή **μήκος** παίρνει αυτόματα τιμή από το μεταβολέα (Εικόνα 2.7).

## 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

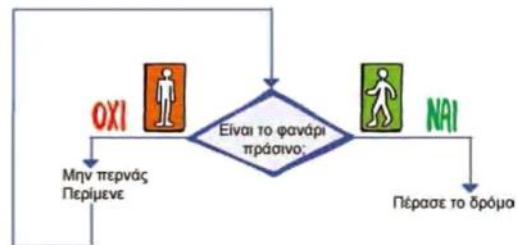
Αν θέλουμε να περάσουμε το δρόμο, επιλέγουμε τι θα κάνουμε ανάλογα με το τι δείχνει το φανάρι. Αν το φανάρι για τους πεζούς είναι πράσινο, περνάμε το δρόμο. Αν είναι κόκκινο, σταματάμε και περιμένουμε. Όμοια, για να ξέρουμε, αν πρέπει να πάρουμε μαζί μας ομπρέλα ή γυαλιά ηλίου, οφείλουμε να κάνουμε τους 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). Ο αλγόριθμος μπορεί να περιγραφεί με βήματα ως εξής:

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



Εικόνα 2.10. Σχηματική αναπαράσταση του αλγορίθμου ελέγχου φαναριού πεζών

### 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  $x$ .
  - Αν το  $x$  είναι μικρότερο από το 0 τότε υπολόγισε την τιμή  $-x$  (δηλαδή  $-1 * x$ ) και εμφάνισε την
  - διαφορετικά εμφάνισε το  $x$ .
- Η εντολή της γλώσσας Scratch που χρησιμεύει για την εκτέλεση του παραπάνω αλγορίθμου από τον υπολογιστή είναι η:



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 ή ομάδα εντολών, διαφορετικά εκτελεί τη δεύτερη.

Η συνθήκη είναι μία λογική πρόταση και εικονίζεται πάντα ως «κουτάκι» με γωνίες. Χρησιμοποιεί συνήθως τα παρακάτω σύμβολα, που ονομάζονται και λογικοί τελεστές.

Σύμβολο	Σημασία	Παράδειγμα
	ισότητα	
	μεγαλύτερο	
	μικρότερο	
	αν μία τιμή περιέχεται σε μία λίστα τιμών	

Από τον αλγόριθμο της εύρεσης της απόλυτης τιμής ενός αριθμού μπορεί να προκύψει το εξής σενάριο:



Στο σενάριο του παραδείγματος η συνθήκη ελέγχου είναι η  $x < 0$  και χρησιμοποιείται για να ελέγξει αν η τιμή της μεταβλητής  $x$  είναι μικρότερη από το μηδέν. Αν είναι, τότε ε-

κτελείται η πρώτη εντολή , η οποία ανακοινώνει στην οθόνη την τιμή του  $x$  με το συνοδευτικό μήνυμα «το  $|x|$  είναι ». Σε διαφορετική περίπτωση, εκτελείται η δεύτερη εντολή, που εμφανίζει το ίδιο μήνυμα και στη συνέχεια την αρχική τιμή του  $x$ .

#### 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

```

όταν το πλήκτρο q πατηθεί
  ρώτησε Πώς ονομάζεται στα αγγλικά η μνήμη του υπολογιστή, όπου αποθηκεύουμε προσωρινά δεδομένα και εντολές και περίμενε
  εάν Μνήμες περιέχει απάντηση
    πες ΜΠΡΑΒΟ!
  αλλιώς
    πες Θα πρέπει να μελετήσεις ξανά το κεφάλαιο με το υλικό του υπολογιστή.
  σταμάτησε το σενάριο

```

Για να μπορέσουμε, βέβαια, να παίξουμε το συγκεκριμένο παιχνίδι, θα πρέπει να εκτελεστεί προηγουμένως το παρακάτω σενάριο, προκειμένου να εισαχθούν οι σωστές απαντήσεις στη λίστα Μνήμες.

```

όταν το πλήκτρο I πατηθεί
  πρόσθεσε RAM στο Μνήμες
  πρόσθεσε R.A.M. στο Μνήμες
  σταμάτησε το σενάριο

```

- α. Δοκιμάστε να αλλάξετε τις ερωτήσεις και τις απαντήσεις του παιχνιδιού γνώσεων.
  - β. Τι ελέγχει η συνθήκη Μνήμες περιέχει απάντηση;
2. Μετρήστε πόσο χρόνο χρειαστήκατε, για να υπολογίσετε σωστά το γινόμενο δυο τυχαίων αριθμών μεταξύ του 1 και του 100.

Για να μετρήσουμε το χρόνο, θα χρησιμοποιήσουμε την εντολή μηδένισε το χρονόμετρο ώστε να μηδενίζεται ο χρόνος στην αρχή του σεναρίου. Με την ενεργοποίηση της παραπάνω εντολής ο χρόνος μετρείται σε δευτερόλεπτα και αποθηκεύεται στη μεταβλητή με το όνομα χρονόμετρο. Στη δραστηριότητα θα εμφανίζεται ο χρόνος, μόνο αν το αποτέλεσμα είναι σωστό, διαφορετικά θα εμφανίζεται ένα μήνυμα λάθους. Το σενάριο μπορεί να γραφεί ως εξής:

```

όταν το πλήκτρο t πατηθεί
  μηδένισε το χρονόμετρο
  όρισε το a1 σε τυχαία επιλογή από 1 μέχρι 100
  όρισε το a2 σε τυχαία επιλογή από 1 μέχρι 100
  ρώτησε ένωση Πόσο κάνει ένωση ένωση a1 * a2 και περίμενε
  εάν απάντηση = a1 * a2
    πες ένωση Μπράβο! Ο χρόνος σου ήταν ένωση χρονόμετρο δευτερόλεπτα
  αλλιώς
    πες Λάθος απάντηση!
  σταμάτησε το σενάριο

```

- α. Τροποποιήστε την παραπάνω δραστηριότητα, ώστε να παίζουν δυο παίκτες. Με την τροποποίηση που θα κάνετε πρέπει το νέο πρόγραμμα να συγκρίνει τους δυο χρόνους και να εμφανίζει σε μήνυμα ποιος νίκησε.
  - β. Τροποποιήστε τη δραστηριότητα, ώστε το πρόγραμμα να συγκρίνει πέντε διαφορετικούς χρόνους και να εμφανίζει τον παίκτη που νίκησε τις περισσότερες φορές.
3. Δημιουργήστε ένα σενάριο που να υπολογίζει την τιμή του x στη συνάρτηση  $\alpha \cdot x + \beta = 0$ .

## 2.8 Δημιουργώντας πιο σύνθετες εφαρμογές με τη γλώσσα Scratch


### Δημιουργία μιας αριθμομηχανής

#### 1η Δραστηριότητα

Στη δραστηριότητα αυτή θα κατασκευάσουμε μια αριθμομηχανή που να κάνει τις τέσσερις βασικές αριθμητικές πράξεις.

**Βήμα 1:** Στην Παλέτα εντολών δημιουργούμε δυο μεταβλητές με τα ονόματα **α** και **β**, και τις εμφανίζουμε στη Σκηνή. Ορίζουμε τις μεταβλητές έτσι ώστε να εμφανίζονται οι αντίστοιχες γραμμές κύλισης και να παίρνουν τιμές από 1 μέχρι και 100. Για να το πετύχουμε αυτό κάνουμε δεξί κλικ σε κάθε μία από τις μεταβλητές στη Σκηνή και από το μενού που εμφανίζεται επιλέγουμε *γραμμή κύλισης*. Στη συνέχεια ξανακάνουμε δεξί κλικ και επιλέγουμε *Όρισε ελάχιστη και μέγιστη τιμή της γραμμής κύλισης* για να καθορίσουμε τα όρια των τιμών που θα παίρνουν οι μεταβλητές.

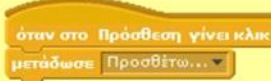



**Βήμα 2:** Δημιουργούμε τέσσερα κουμπιά, ένα για την κάθε πράξη, με τα ονόματα «Πρόσθεση», «Αφαίρεση», «Πολλαπλασιασμός» και «Διαίρεση». Τα κουμπιά εισάγονται ως νέες μορφές με το εικονί-

διο  από τις έτοιμες που υπάρχουν στο φάκελο Things. Στη συνέχεια επιλέγοντας **Διόρθωσε** από την καρτέλα *Ενδυμασίες* της κάθε μορφής μπορούμε να προσθέσουμε το κατάλληλο σύμβολο.

**Βήμα 3:** Κάθε κουμπί στη Σκηνή δημιουργείται για κάποιο σκοπό· συνήθως με την ενεργοποίησή του εκτελείται ένα σενάριο ή ομάδα εντολών. Κάνοντας κλικ στο κάθε κουμπί θέλουμε να ενεργοποιείται η γάτα και να εμφανίζει το αποτέλεσμα της αντίστοιχης πράξης. Για το λόγο αυτό θα χρειαστεί να μεταδοθεί ένα διαφορετικό μήνυμα για κάθε κουμπί στο οποίο θα κάνουμε κλικ. Το μήνυμα αυτό θα το λάβει η γάτα και θα εκτελέσει το αντίστοιχο σενάριο υπολογισμού του αποτελέσματος.

**Βήμα 4:** Δημιουργούμε ένα σενάριο για κάθε πράξη που θα εκτελεί η γάτα με το οποίο θα εμφανίζεται και το αποτέλεσμα.

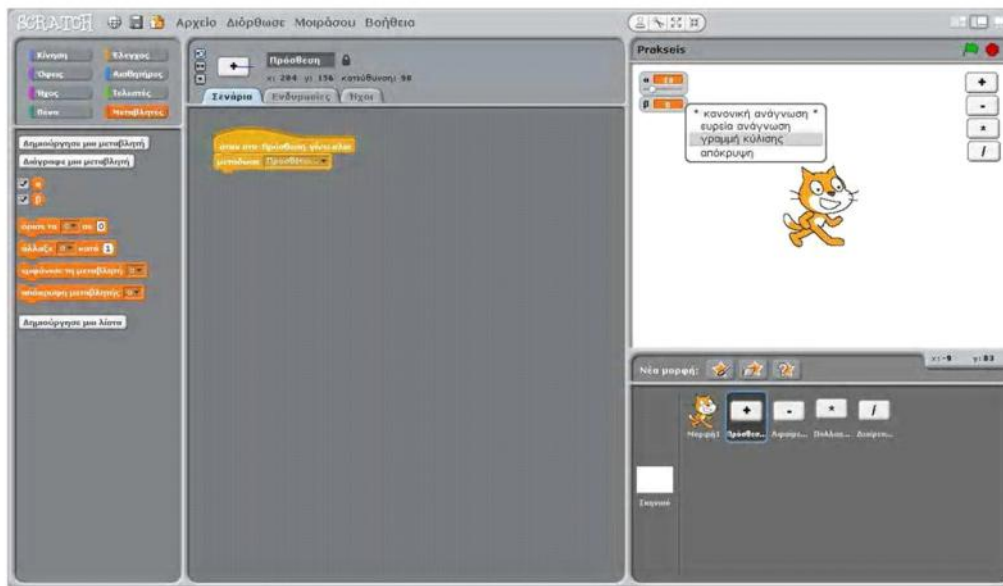
Το πρώτο ζεύγος σεναρίων που αντιστοιχεί στο κουμπί της Πρόσθεσης είναι

 και  για τις μορφές  και  αντίστοιχα.

Στην παρακάτω εικόνα φαίνεται τόσο το σενάριο της μορφής , όσο και το μενού που εμφανίζεται κάνοντας δεξί κλικ στη μεταβλητή **β**.

**Άσκηση:** Να δημιουργήσετε τα κατάλληλα σενάρια, ώστε να λειτουργούν και τα τέσσερα κουμπιά στην αριθμομηχανή μας.





## Κυνηγητό σκύλου γάτας

### 2η Δραστηριότητα

Μία πιο σύνθετη δραστηριότητα περιλαμβάνει περισσότερες κινούμενες μορφές. Επίσης οι μορφές μπορούν να αλλάζουν ενδυμασίες, ώστε να δημιουργούμε πιο ελκυστικά σενάρια.

Στην παρακάτω εικόνα έχουμε βάλει δυο μορφές, της γάτας και του σκύλου. Η εναλλαγή στην ενδυμασία μίας μορφής γίνεται ως εξής:

**Βήμα 1:** Επιλέγουμε την μορφή που θέλουμε από τη λίστα των Μορφών.

**Βήμα 2:** Κάνουμε κλικ στην καρτέλα Ενδυμασίες και με το κουμπί **Εισαγωγή** εμφανίζεται ένα παράθυρο από όπου μπορούμε να επιλέξουμε την ενδυμασία που μας εξυπηρετεί.

**Βήμα 3:** Στην περίπτωση που θέλουμε οι ενδυμασίες της μορφής να εναλλάσσονται, ώστε να δημιουργείται η ψευδαίσθηση της κίνησης, εισάγουμε σε μία εντολή επανάληψης την εντολή **επόμενη ενδυμασία** από την ομάδα **Όψεις**.

Οι μορφές αρχικά παίρνουν όνομα τύπου Μορφή1, Μορφή2, Μορφή3 ... Μπορούμε, όμως, να αλλάζουμε τα ονόματά τους με επιλογή πάνω στη μορφή και κλικ στο πλαίσιο πάνω από τις καρτέλες της Περιοχής σεναρίων.

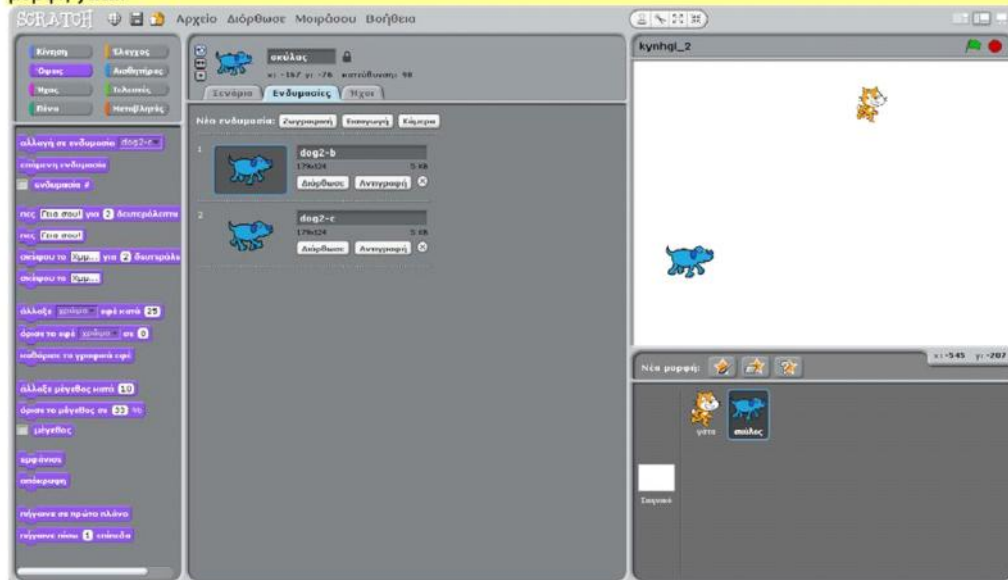
Στη συνέχεια μπορούμε να καλούμε τη μορφή με το νέο όνομα, όπως π.χ.: γάτα, σκύλος

Η δημιουργία των σεναρίων έχει ως αποτέλεσμα ένα «κυνηγητό» μεταξύ γάτας και σκύλου, μέχρι ο σκύλος να ακουμπήσει τη γάτα. Τότε τα σενάρια σταματούν και ο σκύλος εμφανίζει το μήνυμα «Σ' έπιασα», ενώ η γάτα ζητά απεγνωσμένα βοήθεια. Παρακάτω φαίνονται τα δύο σενάρια, για τη γάτα και το σκύλο αντίστοιχα.



Η εντολή **επανάλαβε ώσπου** χρησιμοποιεί μια λογική πρόταση για να ελέγξει την επανάληψη(την εκτέλεση)/τον τερματισμό της εκτέλεσης των εντολών στο εσωτερικό της. Η λογική πρόταση είναι η: **αγγίζει το γάτο** για το σενάριο που εκτελεί ο σκύλος, και **αγγίζει το σκύλος** για το σενάριο της γάτας. Αυτές οι λογικές προτάσεις είναι ΑΛΗΘΕΙΣ όταν οι δύο μορφές σκύλος και γάτα έρχονται σε επαφή.

Η εντολή **δείξε στο γάτο** αλλάζει συνεχώς την κατεύθυνση της μορφής **σκύλος**, ώστε να κοιτάει προς τη μορφή **γάτα**.



**Άσκηση:** Τροποποιήστε την παραπάνω δραστηριότητα, ώστε η γάτα να κινείται σε μία διεύθυνση μόνο. Ποια εντολή πρέπει να διαγράψετε;

Στη συνέχεια δημιουργήστε δυο σενάρια που το καθένα απ' αυτά θα ενεργοποιείται από τα βελάκια «←» και «→» στο πληκτρολόγιο. Το κουμπί «→» θα αλλάζει τη κατεύθυνση της μορφής με το όνομα γάτα προς τα δεξιά κατά 90 μοίρες. Αντίστοιχα το κουμπί «←» θα αλλάζει τη κατεύθυνση της μορφής με το όνομα γάτα προς τα αριστερά κατά 90 μοίρες.

## 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 συνήθως 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.

Σε αυτές τις σημειώσεις παρουσιάζεται το περιβάλλον προγραμματισμού Scratch.



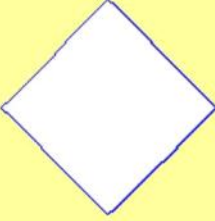
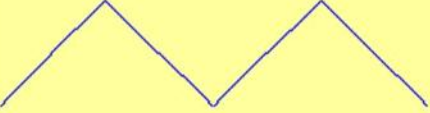
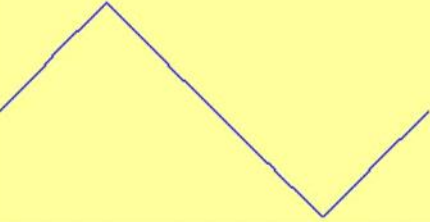
## ΑΣΚΗΣΕΙΣ ΑΥΤΟ-ΑΞΙΟΛΟΓΗΣΗΣ

1. 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 πρόταση.

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 ένας υπολογιστής είναι η γλώσσα μηχανής.	

2. Σχεδίασε το αποτέλεσμα που προκύπτει από τις παρακάτω εντολές στο περιβάλλον της γλώσσας Scratch. (Σημείωση: στο περιβάλλον έχουμε τοποθετήσει μία μορφή, ώστε να μπορούν να εκτελεστούν οι εντολές που ακολουθούν.)

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

<p><b>κατέβασε πένα</b></p> <p>στρίψε ↺ 45 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↻ 90 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↺ 90 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↻ 90 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↺ 90 μοίρες</p>	<p>1.</p> 
<p><b>κατέβασε πένα</b></p> <p>στρίψε ↺ 45 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↻ 90 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↺ 90 μοίρες</p> <p>κινήσου 100 βήματα</p>	<p>2.</p> 
<p><b>κατέβασε πένα</b></p> <p>στρίψε ↺ 45 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↻ 90 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↻ 90 μοίρες</p> <p>κινήσου 100 βήματα</p> <p>στρίψε ↻ 90 μοίρες</p> <p>κινήσου 100 βήματα</p>	<p>3.</p> 



## 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 περάσανε 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, πώς 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;



### 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 νομίζετε ότι υπάρχουν πολλές γλώσσες προγραμματισμού; Θα μπορούσαν να αντικατασταθούν όλες οι γλώσσες προγραμματισμού με μία;

### Βιβλιογραφία

- Αράπογλου Α. κ.α., (2007). Πληροφορική, Ο.Ε.Δ.Β., Αθήνα.
- Lifelong Kinderkarten Group, (2009). Οδηγός χρήσης Scratch - έκδοση 1.4, MIT Media Lab, <http://info.scratch.mit.edu/Languages> & <http://scratch.mit.edu> (η.τ.ε. 07/05/2010).