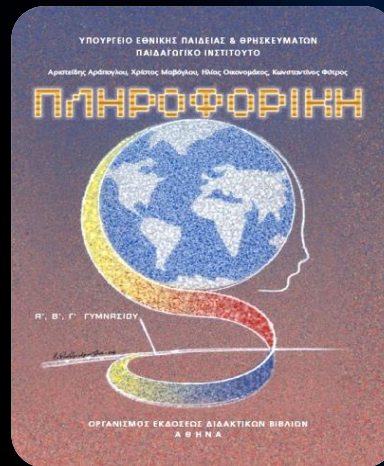




ΟΙ ΔΙΑΔΙΚΑΣΙΕΣ ΤΗΣ LOGO

ΚΕΦ. 7



ΕΥΑΓΓΕΛΟΣ Χ. ΖΙΟΥΛΑΣ (ΚΑΘΗΓΗΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ)

ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

- Στην πληροφορική ο **διαδικαστικός προγραμματισμός** (procedural programming) είναι μία φιλοσοφία δημιουργίας προγραμμάτων ως ένα σύνολο από επιμέρους ανεξάρτητα τμήματα προγραμμάτων.



- Ο προγραμματισμός αυτός βασίζεται στην αρχή του **διαίρει και βασίλευε**, καθώς διασπά το βασικό πρόβλημα σε μικρότερα **υπο-προβλήματα** (γνωστά επίσης και ως **εργασίες**).
- Κάθε εργασία με πολύπλοκη περιγραφή διαιρείται σε μικρότερες, έως ότου οι εργασίες να είναι αρκετά μικρές, περιεκτικές και εύκολες προς κατανόηση.

Ανάλυση

ΠΡΟΒΛΗΜΑ 1

ΥΠΟΠΡΟΒΛΗΜΑ 1

ΥΠΟΠΡΟΒΛΗΜΑ 2

ΥΠΟΠΡΟΒΛΗΜΑ
1.1

ΥΠΟΠΡΟΒΛΗΜΑ
1.2

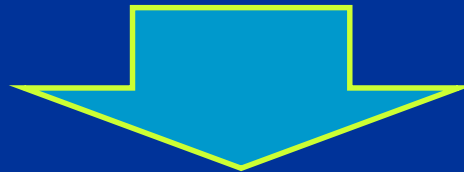
ΥΠΟΠΡΟΒΛΗΜΑ
2.1

ΥΠΟΠΡΟΒΛΗΜΑ
2.2

Επίλυση

ΔΙΑΔΙΚΑΣΙΑ

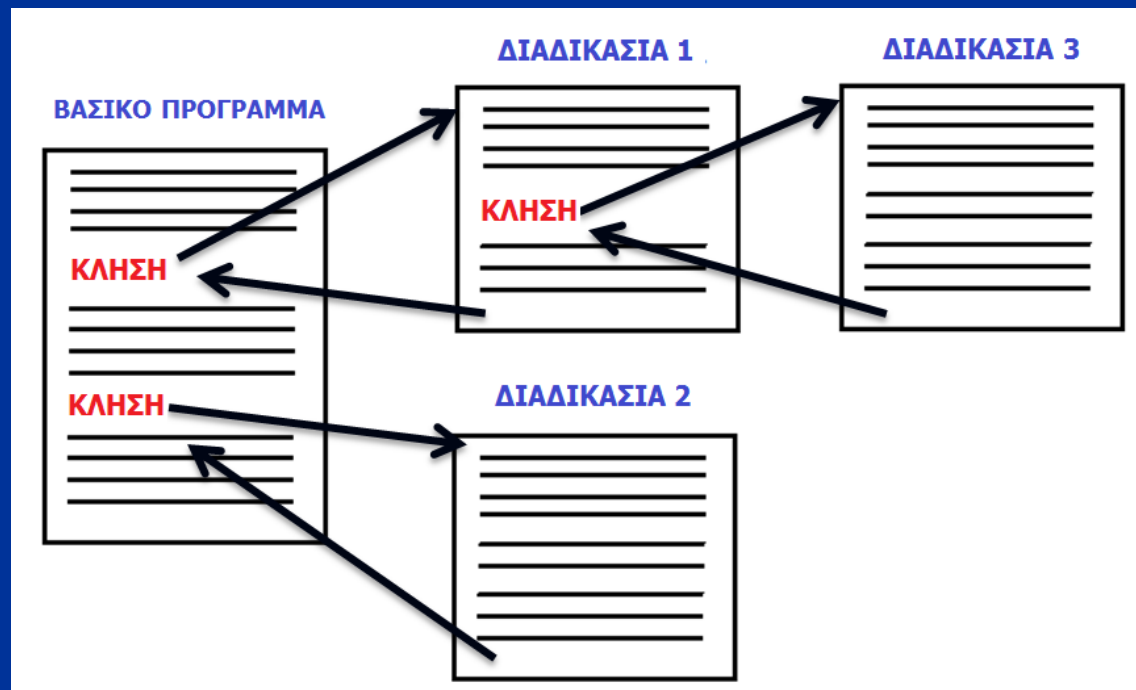
Η διαδικασία, γνωστή επίσης και ως **ρουτίνα**, **υπορουτίνα**, **μέθοδος** ή **συνάρτηση**, είναι απλά ένα αυτοτελές σύνολο εντολών προς εκτέλεση, που επιλύουν μια συγκεκριμένη εργασία.



Διαδικασία ονομάζουμε ένα πλήθος εντολών – οδηγιών που ομαδοποιούνται κάτω από ένα κοινό όνομα. Οι εντολές αυτές εκτελούνται με τη σειρά κάθε φορά που γράφουμε το όνομα της διαδικασίας.

ΤΟ ΠΛΕΟΝΕΚΤΗΜΑ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ

- Το πλεονέκτημα μιας διαδικασίας είναι ότι μπορούμε **να την καλέσουμε** με το **όνομά** της όποτε χρειαστεί χωρίς να χρειάζεται να πληκτρολογούμε ξανά όλες τις εντολές που περιέχει.



ΔΗΜΟΣΙΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

- Η δημιουργία μιας **δημόσιας διαδικασίας** γίνεται στη **καρτέλα Procedures** και η μορφή της ακολουθεί μια συγκεκριμένη δομή.



Procedures

- Μια διαδικασία ξεκινά με την εντολή **to** και ακολουθεί το όνομα της διαδικασίας.
- Η διαδικασία μετά τις εντολές τελειώνει πάντα με τη λέξη **end** σε ξεχωριστή γραμμή.

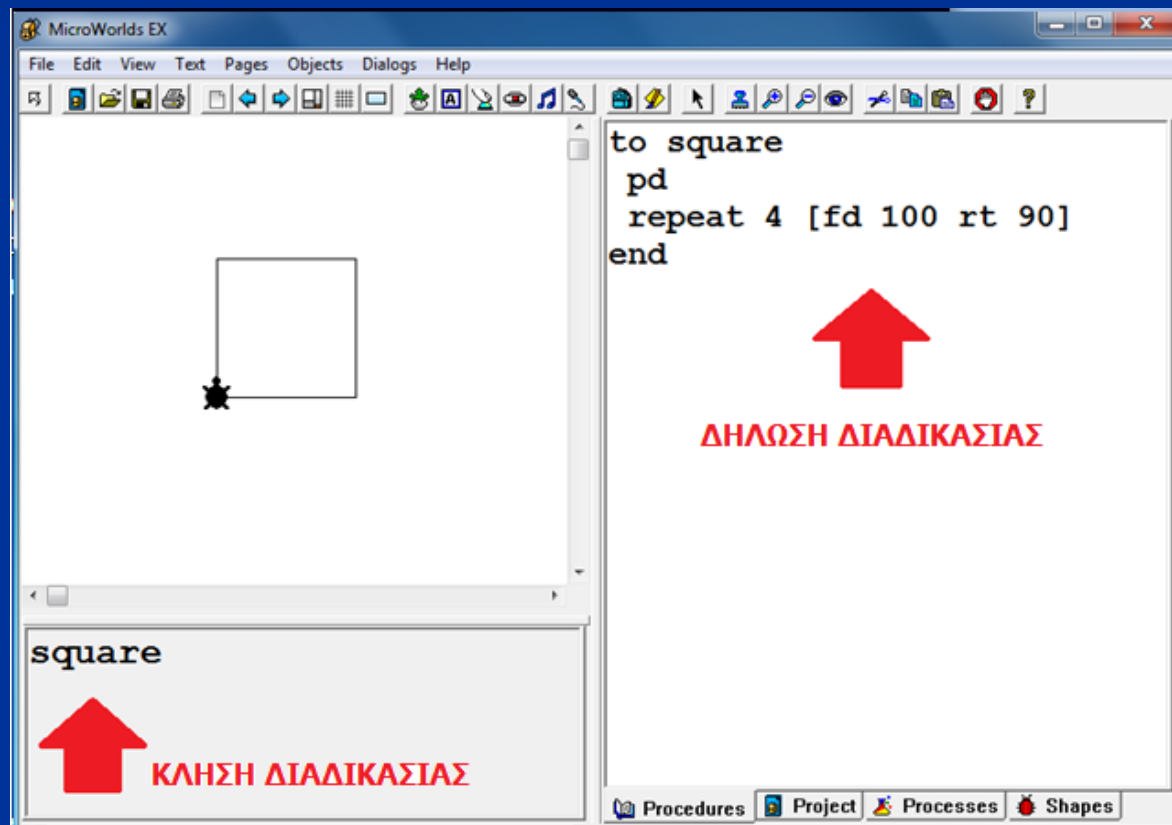
ΔΗΛΩΣΗ ΔΗΜΟΣΙΑΣ ΔΙΑΔΙΚΑΣΙΑΣ

```
to όνομα_διαδικασίας  
    εντολή1  
    εντολή 2  
    .....  
end
```

Το όνομα της διαδικασίας **δεν πρέπει** να περιέχει **κενά** διαστήματα.

ΚΛΗΣΗ ΔΙΑΔΙΚΑΣΙΑΣ

Για να καλέσουμε μια διαδικασία, αρκεί να γράψουμε το όνομα της στο **Κέντρο Εντολών** ή να την χρησιμοποιήσουμε ως τμήμα μιας εντολής μέσα στο σάκο της χελώνας ή σε ένα κουμπί.

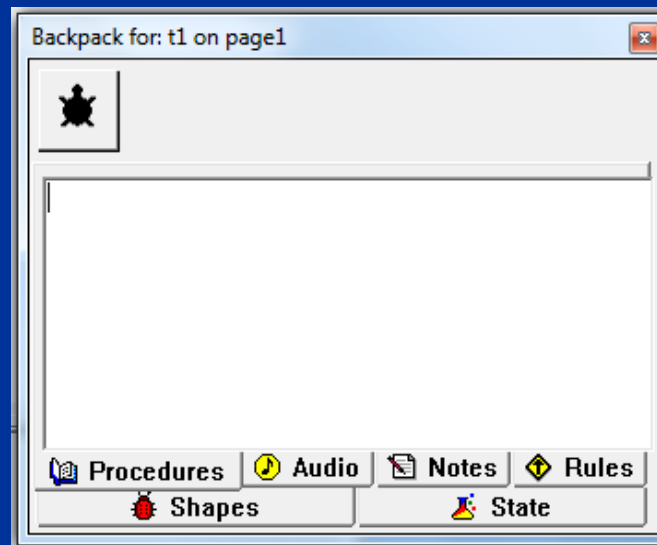


ΠΑΡΑΤΗΡΗΣΕΙΣ

- Εάν μια διαδικασία στο σάκο μιας χελώνας (**Ιδιωτική Διαδικασία**) έχει το ίδιο όνομα με μια διαδικασία στην καρτέλα Διαδικασίες (**Δημόσια Διαδικασία**), τότε η συγκεκριμένη χελώνα χρησιμοποιεί την Ιδιωτική διαδικασία και ποτέ τη Δημόσια.
- Εάν καλέσουμε στο κέντρο εντολών μια διαδικασία που δεν υπάρχει, το MicroWorlds αδυνατώντας να την εντοπίσει επιστρέφει μήνυμα λάθους: **I don't know how to ...**
- Όταν ορίσουμε μια διαδικασία, αυτή γίνεται **μέρος του λεξιλογίου** του MicroWorlds **μόνο για το συγκεκριμένο αρχείο**. Αν ανοίξουμε κάποιο άλλο αρχείο ή δημιουργήσουμε ένα νέο αρχείο, η συγκεκριμένη διαδικασία δεν είναι πια διαθέσιμη.

ΙΔΙΩΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

- Στην καρτέλα **Procedures** του σάκου της χελώνας μπορούμε να ορίσουμε **Ιδιωτικές Διαδικασίες**, οι οποίες είναι γνωστές και μπορούν να κληθούν μόνο από τη χελώνα που τις διαθέτει στο σάκο της.



Για να εμφανίσουμε τις ιδιωτικές διαδικασίες μιας χελώνας, ανοίγουμε το σάκο της χελώνας και επιλέγουμε την καρτέλα **Procedures**.

ΔΙΑΔΙΚΑΣΙΑ **START-UP**

- Αν θέλουμε κάποιες ενέργειες να εκτελούνται αυτόματα κάθε φορά που ανοίγουμε ένα αρχείο, πρέπει να τοποθετούμε τις αντίστοιχες εντολές μέσα σε μια δημόσια διαδικασία με όνομα **startup**, στην καρτέλα **Procedures** του αρχείου.

```
to startup
```

```
  announce [Welcome to my World!]
```

```
  announce [This is MicroWorlds EX]
```

```
  page1
```

```
end
```

- Η διαδικασία startup είναι επίσης ένας καλός τρόπος για να δίνουμε **αρχικές τιμές** σε **μεταβλητές** και άλλα αντικείμενα του αρχείου.
- Η διαδικασία startup λειτουργεί ως **απλή διαδικασία** και δεν μπορεί να δεχθεί παραμέτρους ως είσοδο.

to startup

t1, setc "blue

t2, setsize 100

t2, setpensize 30

setbg 55

end

ΤΥΠΟΙ ΔΙΑΔΙΚΑΣΙΩΝ

- Στο περιβάλλον MicroWorlds EX μπορούμε να δημιουργήσουμε τρία είδη διαδικασιών:

A) Απλές Διαδικασίες

B) Παραμετρικές Διαδικασίες

C) Υπερ-διαδικασίες

ΑΠΛΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

- Ονομάζονται οι διαδικασίες που δεν χρησιμοποιούν παραμέτρους (μεταβλητές).

```
to square  
  repeat 4 [fd 100 rt 90]  
end
```

κλήση: **square**

```
to rectangular  
  repeat 2 [fd 100 rt 90 fd 50 rt 90]  
end
```

κλήση: **rectangular**

ΠΑΡΑΜΕΤΡΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

- Με μια απλή διαδικασία, μπορούμε να σχεδιάσουμε το ίδιο ακριβώς σχήμα κάθε φορά που εκτελείται.
- Οι παραμετρικές είναι διαδικασίες που χρησιμοποιούν μια λίστα από **παραμέτρους (μεταβλητές)** σαν είσοδο.
- Με τη χρήση παραμέτρων, μια διαδικασία μπορεί να σχεδιάσει σχήματα με διαφορετικά μήκη.

- Μια διαδικασία μπορεί να τροποποιηθεί έτσι ώστε αντί να σχεδιάζει πάντα ένα πολύγωνο με πλευρά 100 βήματα, να μπορεί να σχεδιάζει πολύγωνα με διαφορετικά μήκη πλευράς.
- Κάθε φορά που θα εκτελούμε τη διαδικασία, πρέπει να προσδιορίζουμε το μήκος της πλευράς του πολυγώνου που θέλουμε να σχεδιάσουμε.
- Η τιμή αυτή αποτελεί την **είσοδο** (**παράμετρος** ή **μεταβλητή**) της διαδικασίας.

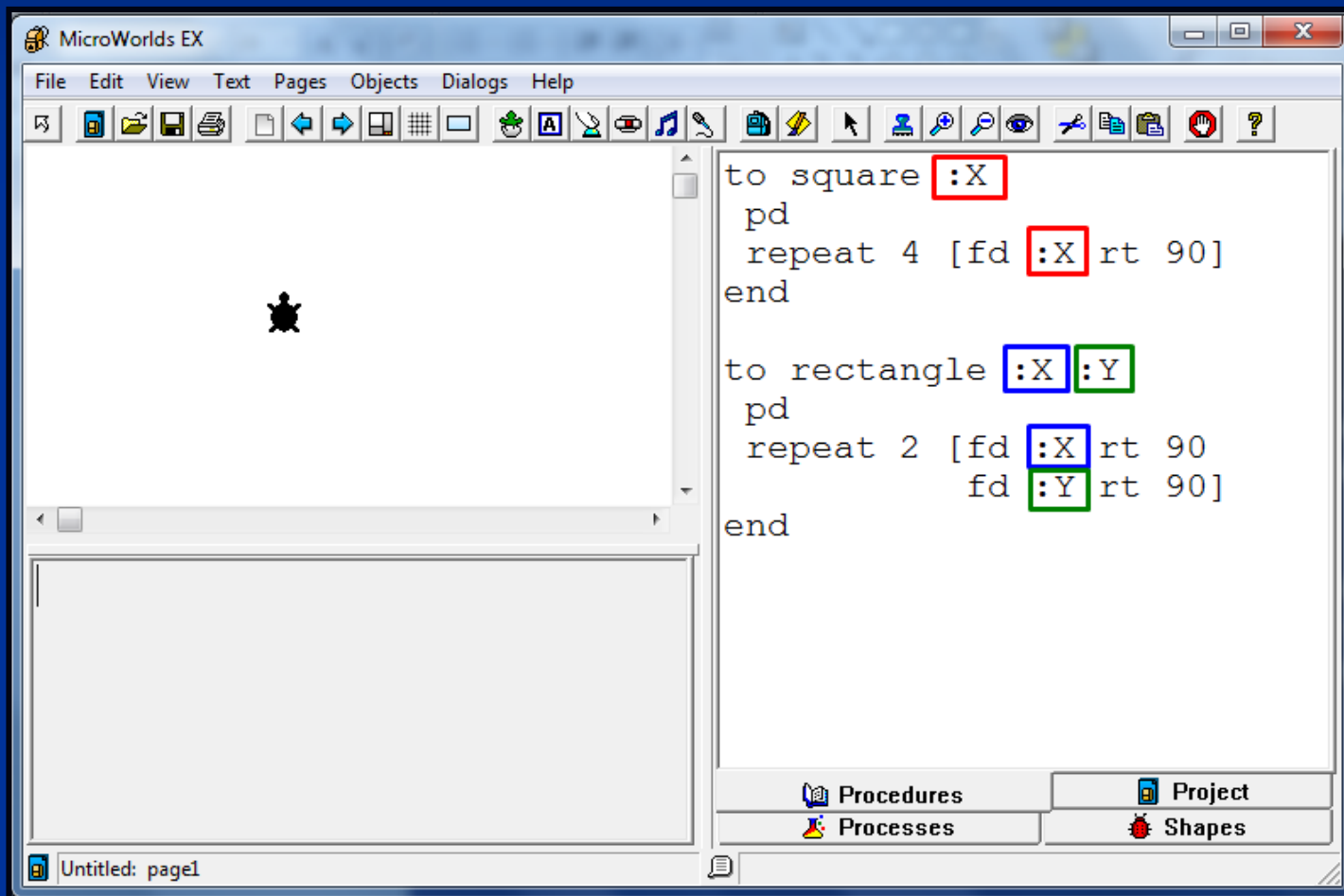
```
to square :X  
  repeat 4 [fd :X rt 90]  
end
```

κλήση: **square 50**

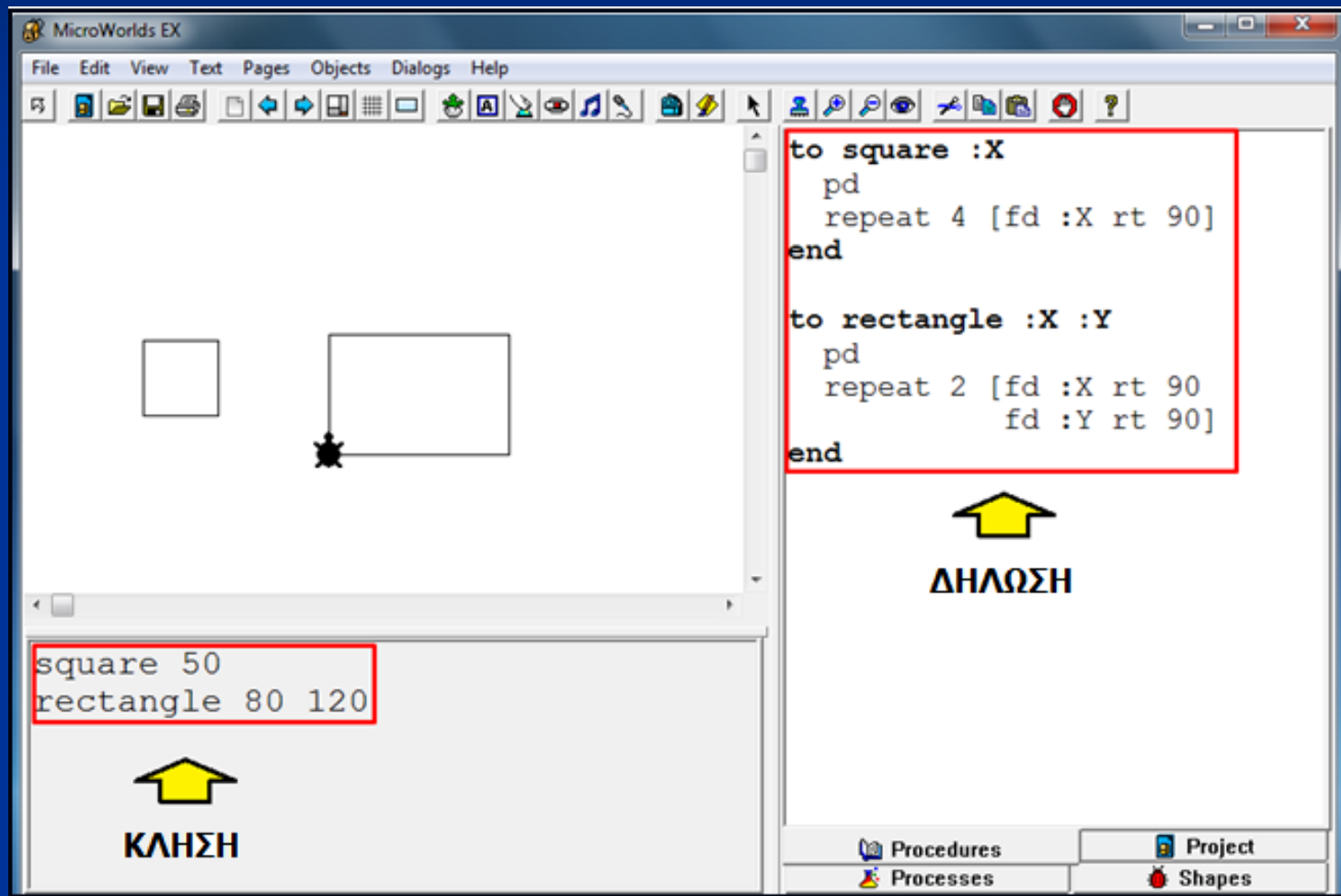
```
to rectangular :X :Y  
  repeat 2 [fd :X rt 90 fd :Y rt 90]  
end
```

κλήση: **rectangular 200 100**

- Το όνομα της παραμέτρου γράφεται στην γραμμή τίτλου της διαδικασίας μετά το όνομα της. Μπορούμε να επιλέγουμε ότι όνομα θέλουμε σαν παράμετρο, αλλά πάντα πρέπει να τοποθετούμε μπροστά από αυτό το **σύμβολο άνω-κάτω τελεία :** e.g. :X,:Y

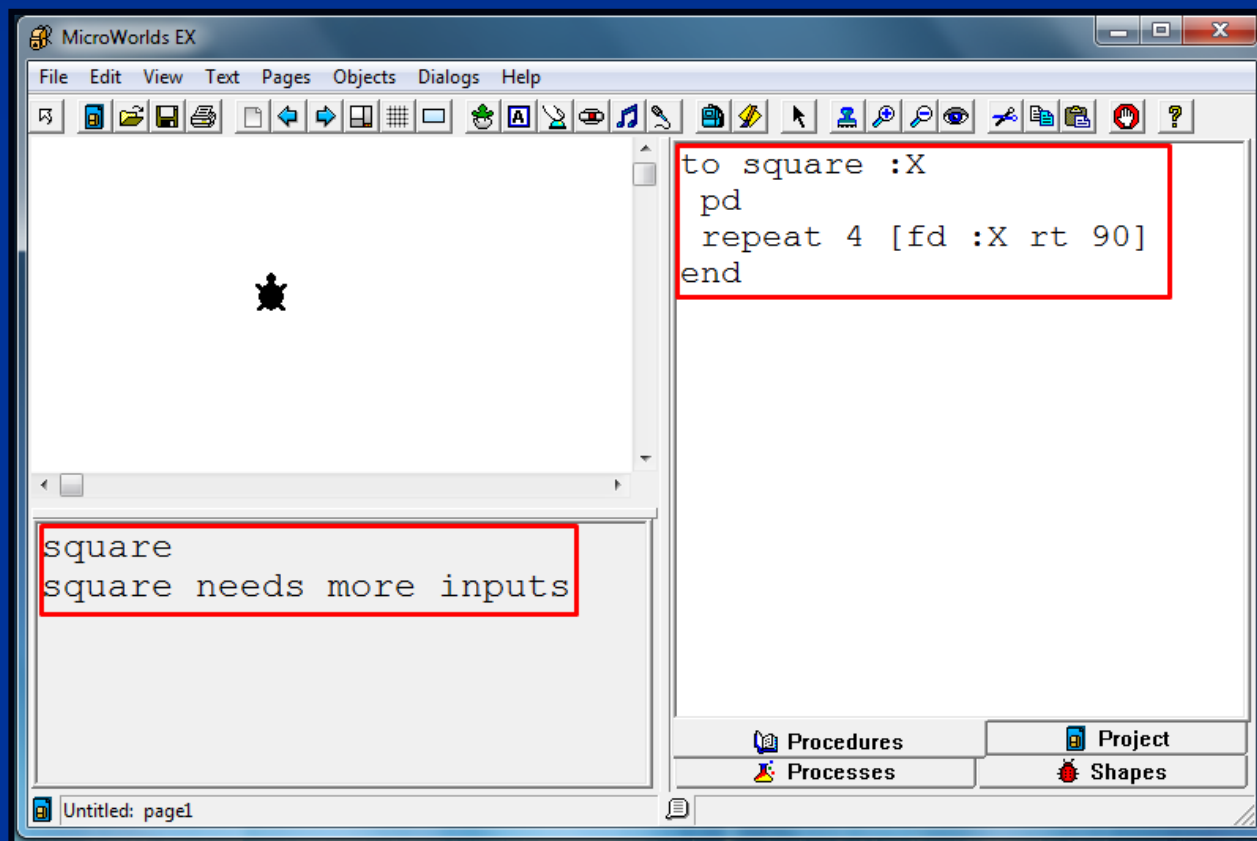


Κάθε φορά που χρησιμοποιούμε την παράμετρο μέσα στη διαδικασία οφείλουμε πάντα να τοποθετούμε μπροστά της το χαρακτήρα της **άνω-κάτω τελείας :**



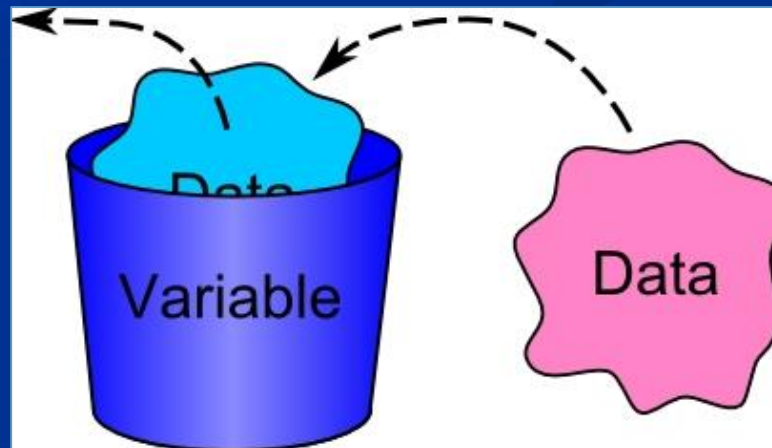
ΠΑΡΑΔΕΙΓΜΑ ΕΣΦΑΛΜΕΝΗΣ ΚΛΗΣΗΣ

- Αν το MicroWorlds δεν εντοπίσει κάποια τιμή ως είσοδο κατά την κλήση μιας παραμετρικής διαδικασίας, εμφανίζει μήνυμα λάθους.



ΜΕΤΑΒΛΗΤΕΣ

- Μια μεταβλητή είναι ένα **συμβολικό όνομα** που δίνουμε σε μια θέση της μνήμης του υπολογιστή.
- Στην μεταβλητή **αποθηκεύουμε προσωρινά** μια τιμή.
- Το **περιεχόμενο** μιας μεταβλητής μπορεί να **μεταβάλλεται** κατά την εκτέλεση του προγράμματος. Αυτό σημαίνει ότι η προηγούμενη τιμής της χάνεται και δεν ξαναχρησιμοποιείται.



ΜΕΤΑΒΛΗΤΕΣ

- Μια μεταβλητή είναι χώρος στη μνήμη που μπορεί να **αποθηκεύει μία μόνο τιμή** κάθε φορά.
- Στη γλώσσα Logo **για να αναφερθούμε** στην τιμή της μεταβλητής βάζουμε μπροστά από το όνομά της το **σύμβολο :**
- Για να της **εκχωρήσουμε μία νέα τιμή**, τότε βάζουμε μπροστά από το όνομά της το **σύμβολο "**

Η ΕΝΤΟΛΗ MAKE

ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ

- Είναι εντολή που **εκχωρεί** μία **τιμή σε** μία **μεταβλητή**.
- Η τιμή που εκχωρείται στην μεταβλητή μπορεί να είναι αριθμός, λέξη ή λίστα.
- Οι μεταβλητές που δημιουργούνται με την εντολή **make** διατηρούν τις τιμές τους μέχρι να διαγραφούν ή να γίνει έξοδος από το MicroWorlds EX. Στην ουσία δεν αποθηκεύονται στο αρχείο.

```
make "A 5
show :A
5
clearname "A
show :A
A has no value
```

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕΤΑΒΛΗΤΩΝ

```
make "X 25
```

```
show :X
```

```
25
```

```
show sqrt :X
```

```
5
```

```
make "Y "Hello
```

```
show :Y
```

```
Hello
```

```
show (st :Y "students)
```

```
Hello students
```

Εκχωρεί την τιμή 25 στη μεταβλητή X

Εμφανίζει την τιμή της μεταβλητής X

Εμφανίζει την τετραγωνική ρίζα της μεταβλητής X

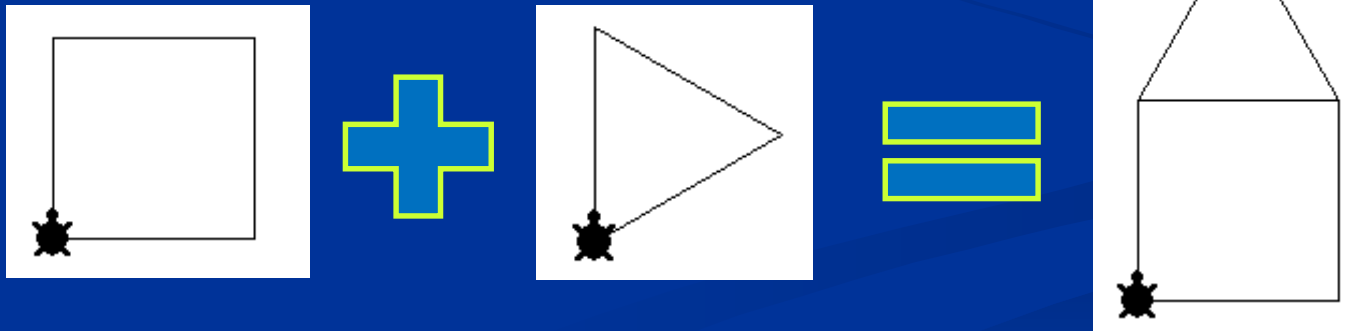
Εκχωρεί την τιμή "Hello στη μεταβλητή Y

Εμφανίζει την τιμή της μεταβλητής Y

Εμφανίζει τη φράση *Hello students*

ΥΠΕΡ-ΔΙΑΔΙΚΑΣΙΕΣ

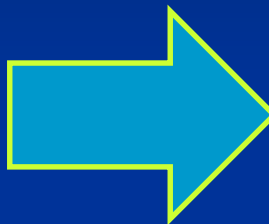
- Είναι διαδικασίες που **περιλαμβάνουν** (καλούν) **άλλες διαδικασίες** μέσα στις εντολές τους.
- Αν δημιουργήσουμε μια διαδικασία square και μία διαδικασία triangle, τότε είναι εφικτό να δημιουργήσουμε μια διαδικασία house η οποία στις εντολές της θα καλεί τις δύο πρώτες.



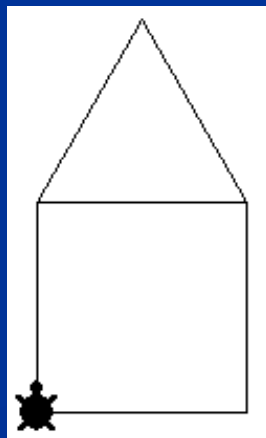
ΠΑΡΑΔΕΙΓΜΑ

```
to square  
  repeat 4 [fd 100 rt 90]  
end
```

```
to triangle  
  repeat 3 [fd 100 rt 120]  
end
```



```
to house  
  pd  
  square  
  fd 100 rt 30  
  triangle  
  lt 30 bk 100  
end
```

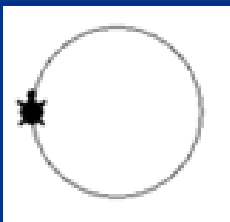


κλήση: **house**

ΠΑΡΑΔΕΙΓΜΑ

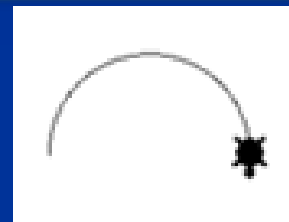
to circle

```
repeat 360 [fd 1 rt 1]  
end
```



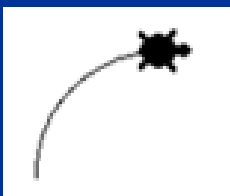
to semicircle

```
repeat 180 [fd 1 rt 1]  
end
```



to quadrant

```
repeat 90 [fd 1 rt 1]  
end
```



to petal

```
repeat 2 [quadrant rt 90]  
end
```

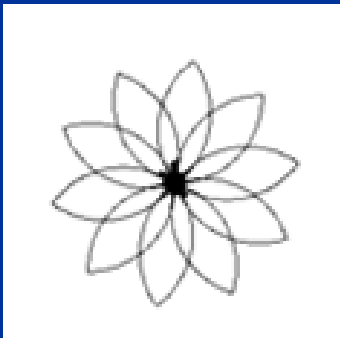


ΠΑΡΑΔΕΙΓΜΑ

```
to flower
```

```
  repeat 10 [petal rt 36]
```

```
end
```



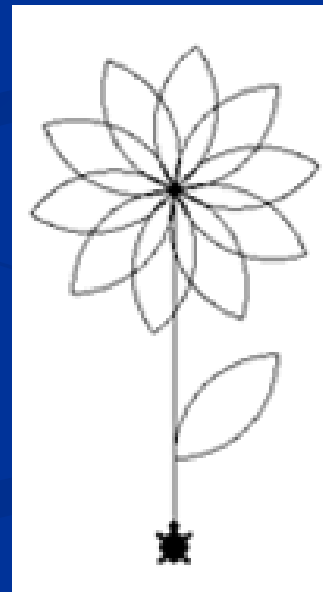
κλήση: **flower**

```
to plant
```

```
  flower
```

```
  bk 150 petal bk 50
```

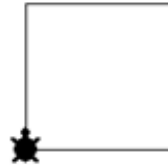
```
end
```



κλήση: **plant**

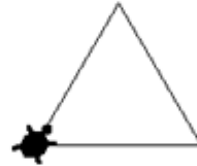
ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΑΡΑΜΕΤΡΙΚΩΝ ΥΠΕΡΔΙΑΔΙΚΑΣΙΩΝ

```
to square :size  
  repeat 4 [fd :size rt 90]  
end
```



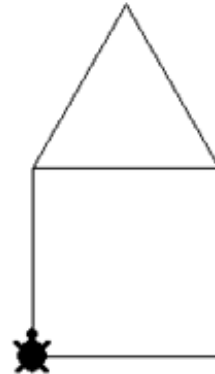
κλήση: square 150

```
to triangle :size  
  repeat 3 [fd :size rt 120]  
end
```



κλήση: triangle 150

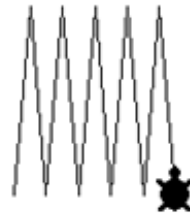
```
to house :size  
  square :size  
  fd :size  
  triangle :size  
  lt 30  
  bk :size  
end
```



κλήση: triangle 150

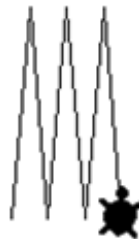
ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΑΡΑΜΕΤΡΙΚΩΝ ΥΠΕΡΔΙΑΔΙΚΑΣΙΩΝ

```
to zigzag :size
  rt 5
  repeat 5
    [fd :size rt 170
     fd :size lt 170]
end
```



κλήση: zigzag 100

```
to zigzag :num :size :angle
  rt 5
  repeat :num
    [fd :size rt :angle
     fd :size lt :angle]
end
```



κλήση: zigzag 3 100 170

ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΑΡΑΜΕΤΡΙΚΩΝ ΥΠΕΡΔΙΑΔΙΚΑΣΙΩΝ

```
to staircase :steps
  repeat :steps
    [fd 30 rt 90
     fd 20 lt 90]
end
```



κλήση: staircase 5

```
to staircase :steps :h :w
  repeat :steps
    [fd :h rt 90
     fd :w lt 90]
end
```



κλήση: staircase 10 20 15

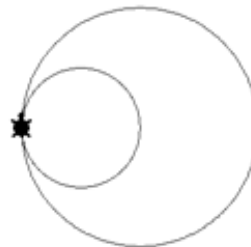
ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΑΡΑΜΕΤΡΙΚΩΝ ΥΠΕΡΔΙΑΔΙΚΑΣΙΩΝ

```
to arc :size  
  repeat :size [fd 1 rt 1]  
end
```



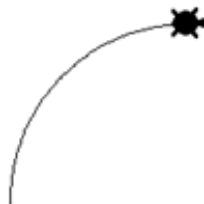
κλήση: arc 270

```
to circle :X  
  repeat 360 [fd :X rt 1]  
end
```



κλήση: circle 1
κλήση: circle 2

```
to quadrant :X  
  repeat 90 [fd :X rt 1]  
end
```



κλήση: quadrant 2

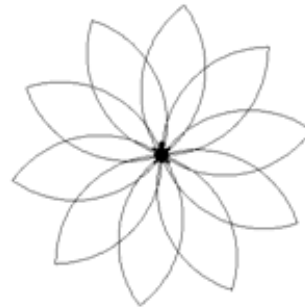
ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΑΡΑΜΕΤΡΙΚΩΝ ΥΠΕΡΔΙΑΔΙΚΑΣΙΩΝ

```
to petal :X  
  repeat 2 [quadrant :X  
    rt 90]  
end
```



κλήση: petal 1
κλήση: petal 2
κλήση: petal 3

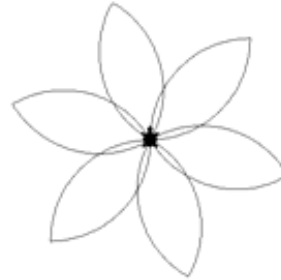
```
to flower :X  
  repeat 10 [petal :X  
    rt 36]  
end
```



κλήση: flower 2

ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΑΡΑΜΕΤΡΙΚΩΝ ΥΠΕΡΔΙΑΔΙΚΑΣΙΩΝ

```
to flower :num :size
  repeat :num
    [petal :size
     rt 360 / :petals]
end
```



κλήση: flower 6 2

```
to plant :num :size
  flower :num :size
  bk 50
  petal :size
  bk 50
end
```



κλήση: plant 15 1