



Πανεπιστήμιο Αιγαίου

Νέες Τεχνολογίες στην Εκπαίδευση

Ενότητα 4: Μικρόκοσμοι και η γλώσσα Logo
(Ενδεικτικές απαντήσεις στις ασκήσεις)

Τσαγγάρης Χρήστος
Τμήμα Μαθηματικών
Σάμος, Δεκέμβριος 2014



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Μικρόκοσμοι και η γλώσσα

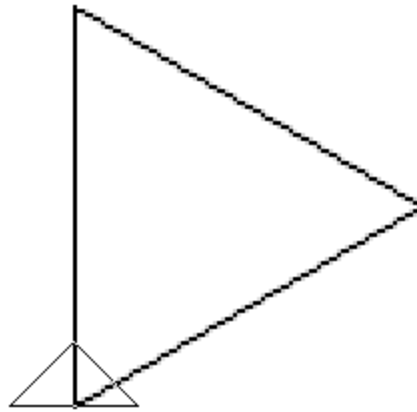
Logo

Ενδεικτικές απαντήσεις στις
ασκήσεις

Η γεωμετρία της χελώνας

Άσκηση

Σχεδιάστε με τη βοήθεια της χελώνας ένα ισόπλευρο τρίγωνο πλευράς 100 σαν αυτό της εικόνας.



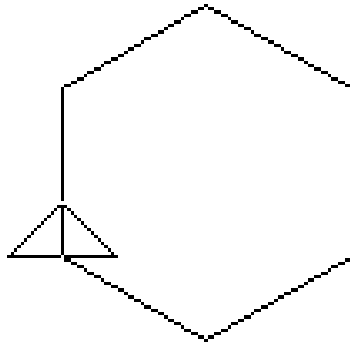
Απάντηση

- Ορισμός διαδικασίας
to triangle
 repeat 3[fd 100 rt 120]
end

- Κλήση διαδικασίας
triangle

Άσκηση

Δημιουργήστε τη διαδικασία `hexagon` η οποία να σχεδιάζει κανονικά εξάγωνα με μήκος πλευράς που θα ορίζεται σαν παράμετρος.



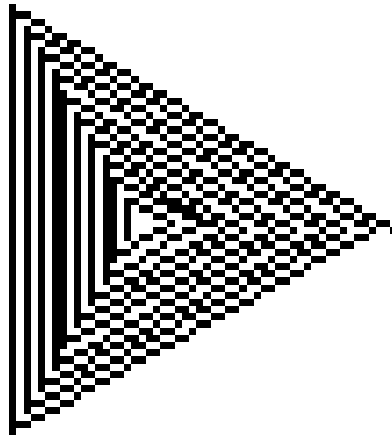
Απάντηση

- Ορισμός διαδικασίας
to hexagon :length
 repeat 6 [fd :length rt 60]
end

- Κλήση διαδικασίας
hexagon 50

Άσκηση

Να υλοποιηθεί η διαδικασία trigspiral η οποία θα σχεδιάζει σπείρα με τρίγωνα αντί για τετράγωνα.



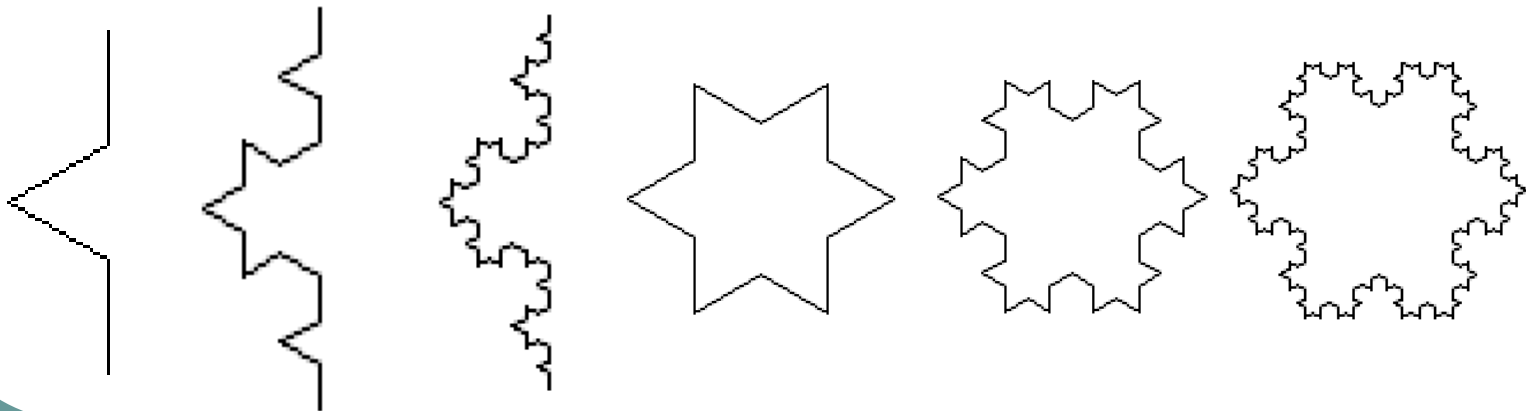
Απάντηση

- Ορισμός διαδικασίας
to trigspiral :distance
 fd :distance
 rt 120
 if :distance < 60 [trigspiral
 :distance + 2]
end

- Κλήση διαδικασίας
trigspiral 5

Άσκηση

Προσπαθήστε να φτιάξετε χρησιμοποιώντας αναδρομή την νιφάδα χιονιού του Koch (Koch Snowflake). Μπορείτε αρχικά να φτιάξετε την μία πλευρά. Η νιφάδα απλά θα καλεί κατάλληλα τρεις φορές την πλευρά. Σαν είσοδο οι διαδικασίες θα δέχονται το μήκος και το βάθος της νιφάδας. Η συνθήκη τερματισμού της πλευράς θα είναι κάτι τέτοιο: `if :depth = 0 [fd :length stop]`. Ουσιαστικά μόνο τότε η διαδικασία θα ζωγραφίζει πλευρά.



Απάντηση 1/3

- Αρχικά θα φτιάξουμε μία διαδικασία η οποία θα δημιουργεί μία απλή πλευρά (με μία μύτη):

```
to simple.side :length
```

```
  fd :length / 3
```

```
  lt 60
```

```
  fd :length / 3
```

```
  rt 120
```

```
  fd :length / 3
```

```
  lt 60
```

```
  fd :length / 3
```

```
end
```

Απάντηση 2/3

- Στη συνέχεια θα φτιάξουμε μία διαδικασία η οποία θα δημιουργεί μία απλή πλευρά με βάθος:

```
to side :length :depth
  if :depth = 0 [fd :length stop]
  side :length / 3 :depth - 1
  lt 60
  side :length / 3 :depth - 1
  rt 120
  side :length / 3 :depth - 1
  lt 60
  side :length / 3 :depth - 1
end
```

Απάντηση 3/3

- Τέλος θα φτιάξουμε μία διαδικασία η οποία θα δημιουργεί το τελικό σχήμα καλώντας 3 φορές την διαδικασία side:

```
to snowflake :length :depth  
  repeat 3 [ side :length :depth rt 120 ]  
end
```

Κλήση διαδικασίας
snowflake 100 3

Προγραμματισμός

Άσκηση

Ορίστε την συνάρτηση που ακολουθεί χρησιμοποιώντας διαδικασία της Logo.

$$f(x) = \begin{cases} 5x + 2 & x \leq 2 \\ x^3 + 3x - 2 & x > 2 \end{cases}$$

Απάντηση

- Ορισμός διαδικασίας

```
to f :x
```

```
  ifelse :x<=2 [
```

```
    output 5*:x+2][
```

```
    output (power :x 3) + 3 * :x - 2]
```

```
end
```

- Κλήση διαδικασίας

```
show f 4
```

Άσκηση

Να δημιουργηθεί διαδικασία η οποία δέχεται έναν ακέραιο αριθμό και επιστρέφει «μονός» ή «ζυγός» αν είναι άρτιος ή περιττός αντίστοιχα.

Απάντηση

- Ορισμός διαδικασίας
to evenorodd :a
 if (modulo :a 2)=0 [output
 "ζυγός]
 output "μονός
end

- Κλήση διαδικασίας
show evenorodd 3

Άσκηση

Να υλοποιηθεί διαδικασία η οποία να δέχεται σαν ορίσματα τους συντελεστές a και b μίας εξίσωσης $ax + b = 0$ και επιστρέφει την λύση της εξίσωσης, εάν υπάρχει, ή κατάλληλο μήνυμα.

Απάντηση

- Ορισμός διαδικασίας
to firstgrade :a :b
 if :a=0 [
 ifelse :b=0 [output
 "αόριστη][output "αδύνατη]
]
 output -:b/:a
end

- Κλήση διαδικασίας
show firstgrade -2 4

Άσκηση

Αλλάξτε τη διαδικασία μετρήματα ώστε

- Να μετρά ως το 20
- Να μετρά ανά 2
- Να ξεκινά το μέτρημα από το 5
- Να μετρά ανάποδα από το 20 έως το 1

Απάντηση

- Ορισμός διαδικασίας (για ανάποδο μέτρημα)

```
to countdown
```

```
  make "i 20
```

```
  while [:i>=1][
```

```
    (print "i: :i)
```

```
    make "i :i-1
```

```
  ]
```

```
end
```

- Κλήση διαδικασίας
countdown

Άσκηση

Να υλοποιηθεί διαδικασία η οποία δέχεται σαν όρισμα το n και υπολογίζει και επιστρέφει το επόμενο άθροισμα

$$S = 1^2 + 3^2 + \dots + (2n - 1)^2$$

Απάντηση

- Ορισμός διαδικασίας

```
to S2 :n
```

```
  make "sum 0
```

```
  make "i 1
```

```
  while [:i<=2 * :n - 1][
```

```
    make "sum :sum + :i * :i
```

```
    make "i :i + 2
```

```
  ]
```

```
  output :sum
```

```
end
```

- Κλήση διαδικασίας

```
show S2 5
```

Άσκηση

Να υλοποιηθεί η διαδικασία fact η οποία επιστρέφει το παραγοντικό ενός αριθμού n .

Απάντηση

- Ορισμός διαδικασίας

```
to fact :n
```

```
  if :n = 0 [output 1]
```

```
  output :n * fact :n - 1
```

```
end
```

- Κλήση διαδικασίας

```
show fact 5
```

Λέξεις και Λίστες

Άσκηση

Να δημιουργηθεί η διαδικασία greetings η οποία να δέχεται σαν είσοδο μία λίστα με ένα ονοματεπώνυμο και να επιστρέφει μία λίστα σαν αυτή της επόμενης εξόδου

show greetings [Eddie Vedder]

[Hello Eddie . So, your last name is Vedder]

Απάντηση

- Ορισμός διαδικασίας
to greetings :name
output (sentence [Hello] first
:name [. So, your last name
is] last :name)
end

- Κλήση διαδικασίας
show greetings [Eddie Vedder]

Άσκηση 1/2

- Θα δημιουργηθεί μία διαδικασία η οποία θα παράγει τυχαία κουτσομπολιά.
- Για ένα καλό κουτσομπολιό, απαραίτητο είναι το πρόσωπο που θα κουτσομπολέψουμε. Μπορούμε να φτιάξουμε μία διαδικασία που θα επιστρέφει στην τύχη έναν από πιθανούς «στόχους».

```
to target.person
```

```
  output pick [Μαρία Γιώργος Κώστας Βαγγέλης Κατερίνα]
```

```
end
```

```
show target.person
```

```
Μαρία
```

```
show target.person
```

```
Γιώργος
```


Άσκηση 2/2

- Δημιουργήστε τη διαδικασία `target.gossip` με πιθανές πράξεις-κουτσομπολιά από τα οποία θα επιλέγεται τυχαία ένα.
- Στη συνέχεια δημιουργήστε τη διαδικασία `gossip` η οποία θα ενώνει σε μία πρόταση, λίστα ένα πρόσωπο με το αντίστοιχο κουτσομπολιό όπως φαίνεται στο επόμενο παράδειγμα.
 `show gossip`
 [Μαρία , συνέχεια αντιγράφει στις εξετάσεις]
- Τέλος, με κάποια επαναληπτική δομή παράξτε 10 κουτσομπολιά.

Απάντηση

- Ορισμός διαδικασίας
to target.gossip
 output pick [[, όλο λέει Ψέματα]
 [, δεν σταματά να μιλάει][,
 συνέχεια αντιγράφει στις
 εξετάσεις]]
end
to target.person
 output pick [Μαρία Γιώργος
 Κώστας Βαγγέλης Κατερίνα]
end
to gossip
 output sentence target.person
 target.gossip
end

- Κλήση διαδικασίας
repeat 10[show gossip]

Άσκηση

Να δημιουργηθούν οι διαδικασίες `oddp` (ή `odd?`) και `evenp` (ή `even?`) οι οποίες δέχονται σαν είσοδο έναν αριθμό και επιστρέφουν `true` αν είναι άρτιος και `false` αν είναι περιττός (η `oddp`), και αντίστροφα (η `evenp`).

Υπενθυμίζεται η διαδικασία `modulo` που επιστρέφει υπόλοιπο διαίρεσης.

Απάντηση

- Ορισμός διαδικασίας

```
to evenp :n
```

```
  output (modulo :n 2) = 0
```

```
end
```

```
to oddp :n
```

```
  output (modulo :n 2) = 1
```

```
end
```

- Κλήση διαδικασίας

```
show evenp 3
```

```
show oddp 3
```

Άσκηση

- Να δημιουργηθεί η διαδικασία `dist` η οποία θα δέχεται σαν είσοδο τις συντεταγμένες ενός σημείου του καρτεσιανού επιπέδου και θα επιστρέφει την απόστασή του από την αρχή των αξόνων.
 - `show dist 300 400`
 - `500`
- Στη συνέχεια να δημιουργηθεί η διαδικασία `dist.list` η οποία θα δέχεται δύο λίστες συντεταγμένων στη μορφή $[x_0 x_1 x_2 \dots]$ και $[y_0 y_1 y_2 \dots]$, και θα επιστρέφει μία λίστα με τις αποστάσεις των σημείων από την αρχή των αξόνων.
 - `show dist.list [300 3 100] [400 4 200]`
 - `[500 5 223.606797749979]`

Απάντηση

- Ορισμός διαδικασίας

```
to dist :x :y
```

```
  output (sqrt (power :x 2) +  
          (power :y 2))
```

```
end
```

```
to dist.list :x :y
```

```
  output (map "dist :x :y)
```

```
end
```

- Κλήση διαδικασίας

```
show dist 300 400
```

```
show dist.list [300 3 100] [400 4  
200]
```



Εφαρμογές

Άσκηση

Μορφοποιήστε τη διαδικασία `simple.random.walk` ώστε η στροφή της χελώνας να μην είναι μέχρι 30 μοίρες δεξιά αλλά από 15 μοίρες αριστερά μέχρι 15 μοίρες δεξιά.

Απάντηση

- Ορισμός διαδικασίας
to better.random.walk
 repeat 100 [fd (random 100) +
 1 (rt (random 31) - 15)]
end

- Κλήση διαδικασίας
better.random.walk

Άσκηση

Μορφοποιήστε τις διαδικασίες ώστε η χελώνα να παίρνει έναν πόντο κάθε φορά που βρίσκει ένα φαγητό και αντί να εμφανίζεται η λέξη `yummy` κάθε φορά να εμφανίζονται οι συνολικοί πόντοι.

```
find.by.smell.list.points [200 0 -100] [300 -100 200] 0
```

```
points: 1
```

```
points: 2
```

```
points: 3
```

Απάντηση

- Ορισμός διαδικασίας

```
to find.by.smell.points :x :y :points
  forever [
    make "old.distance distance (list
      :x :y)
    fd 1
    if (distance (list :x :y))<=10 [
      make "points :points + 1
      output :points
    ]
    if :old.distance <distance (list :x
      :y)[
      rt 20
    ]
  ]
end
```

```
to find.by.smell.list.points :x :y :points
  if empty? :x [stop]
  make "points find.by.smell.points
    first :x first :y :points
  (show "points: :points)
  find.by.smell.list.points
    butfirst :x butfirst :y :points
end
```

- Κλήση διαδικασίας

```
find.by.smell.list.points [200 0 -
  100] [300 -100 200] 0
```

Άσκηση

Κατασκευάστε τη διαδικασία union η οποία θα δέχεται δύο λίστες - σύνολα και θα επιστρέφει μία λίστα που αναπαριστά την ένωσή τους.

Απάντηση

- Ορισμός διαδικασίας
to union :s1 :s2
 if empty? :s1 [output :s2]
 ifelse (not member? (first :s1)
 :s2)[
 output sentence (first :s1)
 union (butfirst :s1) :s2][
 output union (butfirst :s1) :s2]
end

- Κλήση διαδικασίας
show union [1 2 3 4 5] [a 3 b 4 g]

Άσκηση

Δημιουργήστε τη διαδικασία `encode.list` που χρησιμοποιώντας τις προηγούμενες διαδικασίες κρυπτογραφεί προτάσεις όπως φαίνεται και στο επόμενο παράδειγμα.

```
show encode.list [επιθεση την αυγη] "αβγδεζηθικλμνξοπρστυφχψω  
"ερτυθιοπασδφγηξκλζχψωβνμ  
[θκαπθζο χογ εψτο]
```

Απάντηση

- Ορισμός διαδικασίας
to encode.list :list :alphabet
:code
output map [encode.word ?
:alphabet :code] :list
end

- Κλήση διαδικασίας
show encode.list [επιθεση την
αυγη]
"αβγδεζηθικλμνξοπρστυφχψω
"ερτυθιοπασδφγηξκλζχψωβνμ

Ευχαριστίες

Ευχαριστούμε τον Αντώνη Κοντογιάννη για την ουσιαστική του συνεισφορά στη συγγραφή των διαφανειών.