ΜΙΑ ΙΔΕΑ ΑΠΟ ΤΗΝ Μ2000

Κάθε πρόγραμμα σε Μ2000 περιλαμβάνει ένα κείμενο, όπου με κατάλληλη σύνταξη περιλαμβάνονται γνωστά αναγνωριστικά (εντολές, εσωτερικές συναρτήσεις, μεταβλητές μόνο για ανάγνωση, σταθερές) και εκείνα που δημιουργούνται από το πρόγραμμα, τα αναγνωριστικά χρήστη όπως τμήματα, συναρτήσεις, σταθερές, μεταβλητές, ετικέτες. Επίσης περιλαμβάνονται και σταθερές τιμές όπως το 100 ή το "αλφαριθμητικό1234"

Με μπλε γράμματα είναι τα γνωστά αναγνωριστικά. Με μαύρα γράμματα είναι τα αναγνωριστικά χρήστη. Με πορτοκαλί χρώμα είναι οι αριθμοί, οι τελεστές όπως το * και σύμβολα όπως οι αγκύλες. Τα αλφαριθμητικά είναι με μωβ χρώμα. Οι συναρτήσεις όπως και οι πίνακες γράφονται με παρενθέσεις. Στο παρακάτω πρόγραμμα η Υ είναι σταθερή, Χ είναι μεταβλητή. Η ΔΙΠΛΑΣΙΟ() είναι συνάρτηση (με ορισμός μιας γραμμής), όπως και η ΑΘΡΟΙΣΜΑ είναι συνάρτηση (για να την χρησιμοποιήσουμε πρέπει να δώσουμε τις παρενθέσεις)

```
ΤΜΗΜΑ ΑΛΦΑ {
      ΣΤΑΘΕΡΗ Υ=5
      Χ=10

      ΤΥΠΩΣΕ Χ*Υ

      ΕΙΣΑΓΩΓΗ "Δώσε νέα Χ=", Χ
      ΑΝ Χ>0 ΤΟΤΕ ΤΥΠΩΣΕ ΡΙΖΑ(Χ)

      ΚΑΝΕ ΔΙΠΛΑΣΙΟ(Α)=2*Α
      ΤΥΠΩΣΕ @(ΠΛΑΤΟΣ/2), ΔΙΠΛΑΣΙΟ(Χ)

      ΣΥΝΑΡΤΗΣΗ ΑΘΡΟΙΣΜΑ {
            ΣΟΥΜΑ=0
            ΕΝΩ ΕΙΝΑΡ {
                  ΣΟΥΜΑ+=ΑΡΙΘΜΟΣ
            }
            =ΣΟΥΜΑ
      }
      ΤΥΠΩΣΕ ΑΘΡΟΙΣΜΑ(1,2,3,4,Υ,6,7,8,9,10)=55
}
ΑΛΦΑ
```

Κατά την εκτέλεση:

1. Θα εκτελεστεί η εντολή ΤΜΗΜΑ και θα φτιάξει το τμήμα ΑΛΦΑ στη λίστα τμημάτων. Αμέσως μετά ακολουθεί η εντολή ΑΛΦΑ που καλεί το τμήμα ΑΛΦΑ. Δηλαδή τα τμήματα καλούνται με το όνομά τους.
2. Αφού εκτελεστεί το ΑΛΦΑ θα συνεχίσει η εκτέλεση μετά τη κλήση του ΑΛΦΑ.
3. Εντός του ΑΛΦΑ η πρώτη εντολή δημιουργεί τη σταθερή Υ με τιμή το 5 (οι αριθμοί αν δεν καθορίζονται αλλιώς είναι τύπου διπλός - πραγματικός διπλού μεγέθους).
4. Η δεύτερη εντολή είναι εντολή εκχώρησης τιμής. Οι μεταβλητές μπορούν να οριστούν με εκχώρηση τιμής.
5. Ακολουθεί η τρίτη εντολή, η ΤΥΠΩΣΕ που δίνει την τιμή της αριθμητικής έκφρασης Χ*Υ

6. Η τέταρτη εντολή εμφανίζει στην κονσόλα το Δώσε νέα Χ= και περιμένει να εισάγουμε έναν αριθμό (μας επιτρέπει να βάλουμε μόνο αριθμό).
7. Η πέμπτη εντολή είναι μια εντολή ελέγχου όπου ελέγχουμε μια συνθήκη, εδώ το Χ>0 και αν είναι αληθής τότε εκτελούμε αυτό που γράφεται μετά την ΤΟΤΕ. Εδώ αν ο Χ είναι θετικός θα μας τυπώσει στην κονσόλα τη τετραγωνική ρίζα του αριθμού με χρήση της ενσωματωμένης συνάρτησης ΡΙΖΑ().
8. Η έκτη εντολή φτιάχνει μια συνάρτηση μιας γραμμής. Όπου στο όνομα υποχρεωτικά υπάρχουν οι παρενθέσεις. Εντός των παρενθέσεων έχουμε μια τοπική μεταβλητή την Α, τη λεγόμενη και τυπική παράμετρο. Αυτή η συνάρτηση δίνει τιμή το 2*Α δηλαδή τη διπλάσια τιμή της τιμής που θα δώσουμε ως παράμετρο στη κλήση της ΔΙΠΛΑΣΙΟ().
9. Η έβδομη εντολή περιέχει την ΤΥΠΩΣΕ με δυο πράγματα, μια συνάρτηση ειδική της ΤΥΠΩΣΕ που μετακινεί το δρομέα από τα αριστερά (πρώτος χαρακτήρας το 0) και εδώ δίνουμε ως τιμή την έκφραση ΠΛΑΤΟΣ/2 όπου το ΠΛΑΤΟΣ είναι μεταβλητή μόνο για ανάγνωση, και δίνει το πλάτος της κονσόλας σε χαρακτήρες, έτσι θέλουμε πριν τυπώσουμε την επόμενη έκφραση στη λίστα τιμών της ΤΥΠΩΣΕ να μεταφέρουμε το δρομέα στο μέσον της γραμμής της κονσόλας (εισόδου-εξόδου).
10. Ακολουθεί ο ορισμός μιας συνάρτησης, ο οποίος γράφει τη συνάρτηση ως τοπική στο τμήμα, και αμέσως μετά την χρησιμοποιούμε, με δέκα παραμέτρους. Δείτε ότι δεν γράφουμε την κάθε τιμή που περάσαμε στην συνάρτηση σε μεταβλητή, αλλά την διαβάζουμε με την ΑΡΙΘΜΟΣ, η οποία σηκώνει μια αριθμητική τιμή από το σωρό τιμών, εκεί που μπήκαν οι δέκα παράμετροι στην κλήση. Επειδή δεν ξέρουμε κάθε φορά πόσους αριθμούς θα στείλουμε πρέπει να κοιτάμε αν ο σωρός έχει αριθμό στη κορυφή με την μεταβλητή μόνο για ανάγνωση την ΕΙΝΑΡ (είναι αριθμός), μέσα από μια επαναληπτική κλήση ενός μπλοκ κώδικα με την ΕΝΩ. Κάθε φορά ο ΑΡΙΘΜΟΣ αυξάνει την ΣΟΥΜΑ.
11. Στο τέλος η τιμή της ΣΟΥΜΑ γίνεται τιμή συνάρτησης με το = ως εντολή και όχι ως τελεστής (έχει μπλε χρώμα).
12. Όταν τελειώσει η εκτέλεση της ΑΘΡΟΙΣΜΑ() η μεταβλητή και ο σωρός της συνάρτησης θα διαγραφούν.
13. Όταν τελειώσει η ΑΛΦΑ, ό,τι φτιάχτηκε θα διαγραφεί, ακόμα και οι δυο συναρτήσεις. Ο σωρός τιμών όμως δεν διαγράφεται. Τα τμήματα τον επιστρέφουν.
14. Κατά τη κλήση της ΑΛΦΑ πέρασε ο τρέχον σωρός στο τμήμα ΑΛΦΑ. Εδώ όμως δεν μπήκε στη κορυφή κάποια τιμή. Ότι υπήρχε στο σωρό τιμών θα έχει μείνει εκεί.
15. Αν όλο το πλαίσιο αυτό, ο κώδικας είναι σε ένα τμήμα Α τότε θα εκτελεστεί και στην επιστροφή από το Α δεν θα υπάρχει τίποτα, ούτε μεταβλητή, ούτε τμήματα και συναρτήσεις που φτιάχτηκαν σε αυτήν.
16. Δεν υπάρχουν περιορισμοί στο τι εντολές θα μπουν σε τμήματα και τι σε συναρτήσεις. Και τα δύο λειτουργούν το ίδιο. Η διαφορά είναι ότι η συνάρτηση καλείται σε έκφραση (μπορεί να κληθεί και ως τμήμα με την ΚΑΛΕΣΕ, αλλά μια μη μηδενική τιμή λαμβάνεται ως λάθος και επιστρέφει στο διερμηνευτή ως τιμή λάθους).
17. Στην συνάρτηση ΑΘΡΟΙΣΜΑ δεν μπορούμε να καλέσουμε την αδελφή συνάρτηση ΔΙΠΛΑΣΙΟ(). Πρέπει να αντιγράψουμε τον ορισμό της εντός της ΑΘΡΟΙΣΜΑ, να εκτελεστεί ο ορισμός της και μετά να την χρησιμοποιήσουμε. Επίσης δεν μπορούμε να δούμε τα Χ και Υ στις δυο συναρτήσεις, επειδή είναι τοπικές μόνο στο ΑΛΦΑ και όχι στα παιδιά του ΑΛΦΑ, τις δυο συναρτήσεις.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΟΜΑΔΕΣ

Το πρόγραμμα παραπάνω μπορούμε να το γράψουμε με χρήση ομάδας. Η ομάδα βάζει ένα επίπεδο θέασης επιπλέον. Μια ομάδα περιέχεται ως τοπική σε ένα τμήμα ή μια συνάρτηση. Εδώ είναι σε ένα τμήμα έστω Α. Στον ορισμό της μπορούμε να δώσουμε ΤΕΛΙΚΗ (αντί ΣΤΑΘΕΡΗ) ένα αναγνωριστικό που θα έχει μια τιμή που δεν θα αλλάζει. Επίσης μπορούμε να δώσουμε τμήματα και συναρτήσεις. Σε ένα τμήμα της ομάδας μπορούμε να βλέπουμε στοιχεία της ομάδας. Αυτά θα έχουν εντός του τμήματος την τελεία στην αρχή. Θα μπορούσαμε αντί για .Χ να είχαμε το ΑΥΤΟ.Χ απλά είναι πιο εύκολο να παραλείψουμε το ΑΥΤΟ.

Όταν εκτελεστή η εντολή ΟΜΑΔΑ θα φτιαχτούν τέσσερα πράγματα: το ΑΛΦΑ.Υ, το ΑΛΦΑ.Χ, το τμήμα ΑΛΦΑ.ΠΡΩΤΟ και η συνάρτηση ΑΛΦΑ.ΑΘΡΟΙΣΜΑ(). Ενώ πριν το τμήμα ΑΛΦΑ έφτιαχνε ότι χρειάζονταν αφού κληθεί, η ομάδα φτιάχνει με την εκτέλεση του ορισμού της τα μέλη της. Όμως ο κώδικας εντός του τμήματος και της συνάρτησης (των δυο από τα τέσσερα μέλη) δεν θα εκτελεστούν, άρα ότι συμβαίνει εκεί θα είναι το ίδιο με ότι συμβαίνει σε κάθε τμήμα και συνάρτηση, εκτός από μια διαφορά. Η ΑΥΤΟ θα αναφέρεται στο αντικείμενο ΑΛΦΑ (οι ομάδες είναι αντικείμενα) και ότι ξεκινάει με ΑΥΤΟ και τελεία ή σκέτο με μια τελεία θα πρέπει να είναι μέλος της ΑΛΦΑ (αλλιώς βγαίνει λάθος). Ένα τμήμα εντός του ΑΛΦΑ.ΠΡΩΤΟ δεν είναι μέλος του ΑΛΦΑ, δεν θα έχει θέαση των μελών του ΑΛΦΑ. Η ΔΙΠΛΑΣΙΟ() εντός της ΑΛΦΑ.ΠΡΩΤΟ δεν θα μπορούσε να είχε την .Χ αλλά αν την καλέσουμε ως ΔΙΠΛΑΣΙΟ(.Χ) εντός της ΑΛΦΑ.ΠΡΩΤΟ θα διαβάσει το ΑΥΤΟ.Χ δηλαδή το ΑΛΦΑ.Χ και θα το περάσει σαν τιμή για την τυπική παράμετρο Α.

Εντός των μελών της ΑΛΦΑ δεν καλούνται τα μέλη με το ΑΛΦΑ μπροστά. Μόνο με τελεία ή με το ΑΥΤΟ και την τελεία.

Δείτε μια σημαντική διαφορά με το τμήμα ΑΛΦΑ. Όσες φορές και να καλέσουμε το τμήμα ΑΛΦΑ του πρώτου παραδείγματος η απόκρισή του θα είναι ίδια (αν δώσουμε και την ίδια νέα Χ). Δεν συμβαίνει αυτό στην ομάδα ΑΛΦΑ. Με την νέα Χ θα αλλάξει η Χ της ομάδας, και θα παραμείνει με την νέα τιμή μέχρι να πάρει άλλη. Αυτό συμβαίνει γιατί εκτελέσαμε το ΑΛΦΑ.ΠΡΩΤΟ και σε αυτό τα .Χ και .Υ δεν ήταν τοπικές μεταβλητές, υπήρχαν και θα παραμείνουν μέχρι να τερματίσει το τμήμα που δημιούργησε την ομάδα ΑΛΦΑ./

Επίσης εδώ όλα τα μέλη είναι δημόσια. Θα μπορούσαμε πριν την Υ να γράψουμε ΙΔΙΩΤΙΚΟ: και μετά την Χ να γράψουμε ΔΗΜΟΣΙΟ: ώστε εντός του τμήματος που φτιάξαμε την ΑΛΦΑ να μην υπάρχουν τα ΑΛΦΑ.Χ και ΑΛΦΑ.Υ, αλλά να υπάρχουν μόνο για την ομάδα.

```
ΟΜΑΔΑ ΑΛΦΑ {
    ΤΕΛΙΚΗ Υ=5
    Χ=10

    ΤΜΗΜΑ ΠΡΩΤΟ {
        ΤΥΠΩΣΕ .Χ*.Υ

        ΕΙΣΑΓΩΓΗ "Δώσε νέα Χ=", .Χ
        ΑΝ .Χ>0 ΤΟΤΕ ΤΥΠΩΣΕ ΡΙΖΑ(.Χ)

        ΚΑΝΕ ΔΙΠΛΑΣΙΟ(Α)=2*Α
        ΤΥΠΩΣΕ @(ΠΛΑΤΟΣ/2), ΔΙΠΛΑΣΙΟ(.Χ)
        ΤΥΠΩΣΕ .ΑΘΡΟΙΣΜΑ(1,2,3,4,.Υ,6,7,8,9,10)=55
    }

    ΣΥΝΑΡΤΗΣΗ ΑΘΡΟΙΣΜΑ {
        ΣΟΥΜΑ=0
        ΕΝΩ ΕΙΝΑΡ {
            ΣΟΥΜΑ+=ΑΡΙΘΜΟΣ
        }
        =ΣΟΥΜΑ
    }

}
ΑΛΦΑ.ΠΡΩΤΟ
```

Εδώ είναι η παραλλαγή με τα ιδιωτικά μέλη, με τη διαφορά ότι τώρα η ομάδα ΑΛΦΑ είναι σε μια συνάρτηση και γυρίζει ως επιστροφή της. Η ομάδα είναι σαν μεταβλητή (αλλά όχι ακριβώς). Εδώ αντί μετά τον ορισμό της να εκτελέσουμε ένα τμήμα της ή μια συνάρτησή της την δίνουμε ως αντικείμενο στην επιστροφή τιμής.

Μετά την Συνάρτηση που απλά βάζει το Κ() στη λίστα συναρτήσεων, κάνουμε εκχώρηση τιμής στην ΒΗΤΑ το αποτέλεσμα της Κ(), το οποίο είναι το αντικείμενο της ομάδας ΑΛΦΑ, το οποίο έχει πια διαγραφεί αφού είναι τοπικό στην Κ() αλλά έχει επιστραφεί ένα αντίγραφό της, ανώνυμο, το οποίο παίρνει το όνομα ΒΗΤΑ.

Με μια εντολή δώσαμε στο ΒΗΤΑ τέσσερα μέλη. Επίσης αν το δούμε αλλιώς, περάσαμε ως αποτέλεσμα κάτι που μεταφέρει κώδικα, σε δυο μέλη του, ένα τμήμα και μια συνάρτηση.

Καλούμε το ΒΗΤΑ.ΠΡΩΤΟ και έχουμε την ίδια απόκριση με το ΑΛΦΑ.ΠΡΩΤΟ

Αυτό που δεν φαίνεται εδώ είναι ότι η επιστροφή της Κ δίνει στην ΒΗΤΑ την τρέχουσα κατάσταση της ΑΛΦΑ. Δηλαδή όπως είδαμε πιο πριν αν τρέξουμε δυο ή τρεις φορές την ΑΛΦΑ.ΠΡΩΤΟ το Χ θα κρατάει τη νέα κατάσταση, έτσι και στην αντιγραφή του ΑΛΦΑ στο ΒΗΤΑ δίνουμε την νέα κατάσταση του ΑΛΦΑ (εδώ απλά δεν μεσολαβεί κάποια εντολή που να αλλάξει την κατάσταση της ΑΛΦΑ, εντός της Κ() και πριν την εκχώρηση ως τιμή συνάρτησης).

Στις συναρτήσεις δεν δηλώνουμε το τι θα επιστρέφει. Αν όμως θέλουμε να επιστρέφει αλφαριθμητικό τότε πρέπει το όνομα να έχει το $ πριν την παρένθεση. Μια συνάρτηση Μ$() γυρίζει αλφαριθμητικό ή κάποιο αντικείμενο που γράφεται με όνομα αλφαριθμητικού.

```
ΣΥΝΑΡΤΗΣΗ Κ {
    ΟΜΑΔΑ ΑΛΦΑ {
    ΙΔΙΩΤΙΚΟ:
        ΤΕΛΙΚΗ Υ=5
        Χ=10

    ΔΗΜΟΣΙΟ:
        ΤΜΗΜΑ ΠΡΩΤΟ {
            ΤΥΠΩΣΕ .Χ*.Υ

            ΕΙΣΑΓΩΓΗ "Δώσε νέα Χ=", .Χ
            ΑΝ .Χ>0 ΤΟΤΕ ΤΥΠΩΣΕ ΡΙΖΑ(.Χ)

            ΚΑΝΕ ΔΙΠΛΑΣΙΟ(Α)=2*Α
            ΤΥΠΩΣΕ @(ΠΛΑΤΟΣ/2), ΔΙΠΛΑΣΙΟ(.Χ)
            ΤΥΠΩΣΕ .ΑΘΡΟΙΣΜΑ(1,2,3,4,.Υ,6,7,8,9,10)=55
        }

        ΣΥΝΑΡΤΗΣΗ ΑΘΡΟΙΣΜΑ {
            ΣΟΥΜΑ=0
            ΕΝΩ ΕΙΝΑΡ {
                ΣΟΥΜΑ+=ΑΡΙΘΜΟΣ
            }
            =ΣΟΥΜΑ
        }

    }
    =ΑΛΦΑ
}
ΒΗΤΑ=Κ()
ΒΗΤΑ.ΠΡΩΤΟ
```

Για ευκολία αντί να δώσουμε μια συνάρτηση να επιστρέφει μια ομάδα, μπορούμε να δώσουμε τον ίδιο κώδικα σε μια ΚΛΑΣΗ που είναι μια συνάρτηση που έχει τον ορισμό μιας ομάδας.

```
ΚΛΑΣΗ Κ {
ΙΔΙΩΤΙΚΟ:
    ΤΕΛΙΚΗ Υ=5
    Χ=10

ΔΗΜΟΣΙΟ:
    ΤΜΗΜΑ ΠΡΩΤΟ {
        ΤΥΠΩΣΕ .Χ*.Υ

        ΕΙΣΑΓΩΓΗ "Δώσε νέα Χ=", .Χ
        ΑΝ .Χ>0 ΤΟΤΕ ΤΥΠΩΣΕ ΡΙΖΑ(.Χ)

        ΚΑΝΕ ΔΙΠΛΑΣΙΟ(Α)=2*Α
        ΤΥΠΩΣΕ @(ΠΛΑΤΟΣ/2), ΔΙΠΛΑΣΙΟ(.Χ)
        ΤΥΠΩΣΕ .ΑΘΡΟΙΣΜΑ(1,2,3,4,.Υ,6,7,8,9,10)=55
    }

    ΣΥΝΑΡΤΗΣΗ ΑΘΡΟΙΣΜΑ {
        ΣΟΥΜΑ=0
        ΕΝΩ ΕΙΝΑΡ {
            ΣΟΥΜΑ+=ΑΡΙΘΜΟΣ
        }
        =ΣΟΥΜΑ
    }
}
ΒΗΤΑ=Κ()
ΒΗΤΑ.ΠΡΩΤΟ
```

Αν δεν θέλουμε η συνάρτηση ΑΘΡΟΙΣΜΑ να είναι μέλος της ομάδας, μπορούμε να την βάλουμε ως τοπική στο τμήμα ΠΡΩΤΟ, όπως το παράδειγμα παρακάτω:

```
ΣΥΝΑΡΤΗΣΗ Κ {
    ΟΜΑΔΑ ΑΛΦΑ {
    ΙΔΙΩΤΙΚΟ:
        ΤΕΛΙΚΗ Υ=5
        Χ=10

    ΔΗΜΟΣΙΟ:
        ΤΜΗΜΑ ΠΡΩΤΟ {
            ΤΥΠΩΣΕ .Χ*.Υ

            ΕΙΣΑΓΩΓΗ "Δώσε νέα Χ=", .Χ
            ΑΝ .Χ>0 ΤΟΤΕ ΤΥΠΩΣΕ ΡΙΖΑ(.Χ)

            ΚΑΝΕ ΔΙΠΛΑΣΙΟ(Α)=2*Α
            ΤΥΠΩΣΕ @(ΠΛΑΤΟΣ/2), ΔΙΠΛΑΣΙΟ(.Χ)

            ΣΥΝΑΡΤΗΣΗ ΑΘΡΟΙΣΜΑ {
                ΣΟΥΜΑ=0
                ΕΝΩ ΕΙΝΑΡ {
                    ΣΟΥΜΑ+=ΑΡΙΘΜΟΣ
                }
                =ΣΟΥΜΑ
            }
            ΤΥΠΩΣΕ ΑΘΡΟΙΣΜΑ(1,2,3,4,.Υ,6,7,8,9,10)=55
        }
    }
    =ΑΛΦΑ
}
ΒΗΤΑ=Κ()
ΒΗΤΑ.ΠΡΩΤΟ
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΛΑΜΔΑ ΣΥΝΑΡΤΗΣΕΙΣ

Επειδή η ομάδα του παραδείγματος έχει μόνο ένα τμήμα φανερό (δημόσιο) θα μπορούσαμε να την αντικαταστήσουμε με μια λάμδα συνάρτηση. Η λάμδα συνάρτηση δεν έχει μέλη, αλλά έχει τα λεγόμενα κλεισίματα, δηλαδή αντίγραφα τα οποία φαίνονται ως τοπικά στο σώμα της συνάρτησης αλλά παραμένουν κλεισμένα σε αυτήν.

Μπορούμε την ΔΙΠΛΑΣΙΟ και ΑΘΡΟΙΣΜΑ να τις κάνουμε επίσης λάμδα συναρτήσεις.

Μια λάμδα συνάρτηση είναι μια μεταβλητή με δυο πράγματα, μια λίστα κλεισιμάτων και ένα μπλοκ κώδικα (συνάρτησης). Πριν το βέλος -> μπορούμε να δώσουμε μια λίστα κλεισιμάτων και μια λίστα παραμέτρων (η οποία θα γίνει ΔΙΑΒΑΣΕ εσωτερικά της συνάρτησης ως πρώτη εντολή). Η ΔΙΠΛΑΣΙΟ έχει μόνο τη λίστα παραμέτρων, ενώ η ΑΘΡΟΙΣΜΑ δεν έχει τίποτα από τα δύο (αλλά θα διαβάσει τις δέκα τιμές που θα δώσουμε), και η επιστροφή της Κ() έχει τιμή μια ανώνυμη λάμδα, με τέσσερα κλεισίματα.

Οι τιμές των κλεισιμάτων είναι ιδιωτικές για τις Λάμδα, και μόνο μέσα από αυτές αλλάζουν, με την εξαίρεση των τιμών που είναι δείκτες σε αντικείμενα, οι οποίοι μπορεί να είναι πολλαπλοί για το ίδιο αντικείμενο.

Κάθε φορά που κατασκευάζεται μια λάμδα με κλεισίματα, μπαίνουν αντίγραφα ως κλεισίματα. Θα μπορούσαμε με επαναληπτική διαδικασία να στήσουμε μια λάμδα με την προηγούμενη τιμή της ως κλείσιμο, με μεγάλο βάθος.

Εδώ δεν έχουμε το τμήμα ΠΡΩΤΟ γιατί έγινε το σώμα της λάμδα, και το καλούμε σαν τμήμα με την ΚΑΛΕΣΕ.

Επειδή αλλάζουμε τιμή στη Χ, θα έχουμε και εδώ κράτηση κατάστασης. Γενικά στο Λάμδα λογισμό δεν έχουμε κράτηση κατάστασης, αλλά εδώ είναι στο χέρι μας αν το θέλουμε να κρατάμε ή όχι κατάσταση.

Με το κράτημα κατάστασης φτιάχνουμε αυτό που λέγεται γεννήτρια. Κάθε νέα κλήση στη λάμδα δίνει την επόμενη τιμή ενώ αν αντιγράψουμε σε κάποια στιγμή τη λάμδα σε άλλη μεταβλητή (γιατί είναι μεταβλητές) τότε θα αντιγραφεί και η κατάστασή της, έτσι μπορούμε να κρατάμε τη γεννήτρια σε διαφορετικές καταστάσεις σε διαφορετικές μεταβλητές.

```
ΣΥΝΑΡΤΗΣΗ Κ {
      ΣΤΑΘΕΡΗ Υ=5
      Χ=10
      ΔΙΠΛΑΣΙΟ=ΛΑΜΔΑ (Α)->2*Α
      ΑΘΡΟΙΣΜΑ= ΛΑΜΔΑ ->{
                  ΣΟΥΜΑ=0
                  ΕΝΩ ΕΙΝΑΡ {
                        ΣΟΥΜΑ+=ΑΡΙΘΜΟΣ
                  }
                  =ΣΟΥΜΑ
            }
      =ΛΑΜΔΑ Χ, Υ, ΔΙΠΛΑΣΙΟ, ΑΘΡΟΙΣΜΑ -> {

            ΤΥΠΩΣΕ Χ*Υ

            ΕΙΣΑΓΩΓΗ "Δώσε νέα Χ=", Χ
            ΑΝ .Χ>0 ΤΟΤΕ ΤΥΠΩΣΕ ΡΙΖΑ(Χ)

            ΤΥΠΩΣΕ @(ΠΛΑΤΟΣ/2), ΔΙΠΛΑΣΙΟ(Χ)
            ΤΥΠΩΣΕ ΑΘΡΟΙΣΜΑ(1,2,3,4,.Υ,6,7,8,9,10)=55
      }
}
ΒΗΤΑ=Κ()
ΚΑΛΕΣΕ ΒΗΤΑ()
```

Μπορούμε να κρατάμε λάμδα συναρτήσεις και σε πίνακες. Δείτε το παράδειγμα παρακάτω (ακολουθεί τον ορισμό της Κ από το προηγούμενο παράδειγμα Με & περνάμε κάτι με αναφορά (η αναφορά γράφεται στο σωρό τιμών, και διαβάζεται με μια εντολή ΔΙΑΒΑΣΕ &Κ η οποία θα μπει αν πρώτη εντολή στο τμήμα ΕΚΤΕΛΕΣΕ. Περνάμε τη Α(3) λάμδα με αναφορά για να κρατήσουμε την κατάσταση.

Θα μπορούσαμε να χρησιμοποιήσουμε την Α(3)() που εκτελεί την λάμδα στην Α(3) αλλά μόνο σε αριθμητική έκφραση, οπότε βάζουμε μια Ν για να εκχωρήσουμε τιμή με τύπο Empty (χωρίς τιμή).

Οι διαφορές με την ΚΑΛΕΣΕ είναι:

1. Μια μη μηδενική τιμή σημαίνει λάθος, σταμάτημα του διερμηνευτή εκτός και αν εκτελείται σε μια ΔΕΣ {} η οποία πετάει το λάθος. Μπορούμε να καλέσουμε με ΚΑΛΕΣΕ ΚΕΝΗ ώστε να πεταχτεί το αποτέλεσμα πριν σημάνει λάθος. Με κλήση της λάμδα σε έκφραση δεν έχουμε τον έλεγχο λάθους

2. Με κλήση της λάμδα σε έκφραση ο σωρός τιμών είναι νέος (με ότι έχουμε για τιμές για παραμέτρους), ενώ η ΚΑΛΕΣΕ καλεί σαν τμήμα, δηλαδή περνάει τον τρέχον σωρό τιμών (με τις τιμές για παραμέτρους στην κορυφή, με πρώτο στη κορυφή την πρώτη τιμή), που σημαίνει ότι αν αφήσουμε τιμές σκουπίδια στο σωρό, θα μείνουν μετά τη κλήση. Αυτό δεν συμβαίνει με τη κλήση ως συνάρτηση σε έκφραση.

```
ΠΙΝΑΚΑΣ Α(10)
Α(3)=Κ()
\\ δεν μπορούμε να δώσουμε το ΚΑΛΕΣΕ Α(3)()
\\ μπορούμε να το περάσουμε με αναφορά σε τμήμα
ΤΜΗΜΑ ΕΚΤΕΛΕΣΕ (&Κ) {
      ΚΑΛΕΣΕ Κ()
}
ΕΚΤΕΛΕΣΕ &Α(3)
ΕΚΤΕΛΕΣΕ &Α(3)
\\ ή μπορούμε να το δώσουμε έτσι
Ν=Α(3)()
```

Αν θέλουμε να κάνουμε αναδρομή σε συνάρτηση ενώ δεν ξέρουμε ποιο θα είναι το όνομά της (αφού μπορεί να έχει περαστεί ως αναφορά ή απλά να είναι λάμδα συνάρτηση), χρησιμοποιούμε το ΛΑΜΔΑ() που καλεί τη τρέχουσα συνάρτηση (θα βγει λάθος αν το χρησιμοποιήσουμε μέσα σε τμήμα).

Υπάρχει τρόπος να έχουμε αναδρομή με χρήση Υ συνυαστή. Βέβαια απλά εδώ δείχνουμε πώς γίνεται. Πρακτικά δεν έχει άξια, αφού η γλώσσα έχει αναδρομή.

Ο συνδυαστής είναι η συνάρτηση Λ (g, x)= g(g,x) όπου το g είναι μια άλλη λάμδα συνάρτηση, με δυο παραμέτρους g και n. Εδώ καλούμε τη κάθε συνάρτηση χωρίς να την έχουμε περάσει σε όνομα, με την προσθήκη παρενθέσεων μετά τον ορισμό (για το λόγο αυτό έχουμε τον κώδικα της Λ σε αγκύλες, οπότε βάζουμε και το = την εντολή επιστροφής τιμής). Μετά στην πρώτη παράμετρο έχουμε μια λάμδα (εδώ χωρίς αγκύλες ο κώδικας), ο οποίος έχει μια ΑΝ() η οποία παίρνει μια συνθήκη και ανάλογα εκτελεί την πρώτη ή την δεύτερη έκφραση μετά το βέλος. Όλα τα περάσματα είναι με τιμή, οπότε η λάμδα που δίνουμε, πχ για το παραγοντικό θα γίνει g και θα αντιγραφεί σε κάθε κλήση. Αντί να καλούμε τον εαυτό της καλούμε το αντίγραφό της. Τερματίζει σε κάθε περίπτωση μόλις γίνει αληθής η συνθήκη. Οι τιμές μαζεύονται όταν πιάσουμε τη συνθήκη. Πχ στο Φιμπονάτσι θα φτάσουμε στο 1 συνολικά 55 φορές, γιατί μόνο το 1 έχουμε ως αριθμό που κάπου προστίθεται! Αντίθετα στο Παραγοντικό θα φτάσουμε μόνο μια φορά στο 1 γιατί όλα τα προηγούμενα n θα είναι σε αναμονή εκτέλεσης του πολλαπλασιασμού!

```
ΤΜΗΜΑ Υ_συνδυαστής {
      \\ Παραγοντικό
      ΤΥΠΩΣΕ ΛΑΜΔΑ (g, x)->{=g(g, x)}(ΛΑΜΔΑ (g, n)->Αν(n=0 ->1, n*g(g, n-1)), 10)=3628800
      \\ Φιμπονάτσι
      ΤΥΠΩΣΕ ΛΑΜΔΑ (g, x)->{=g(g, x)}(ΛΑΜΔΑ (g, n)->Αν(n<=1 ->n, g(g, n-1)+g(g, n-2)), 10)=55
}
Υ_συνδυαστής
```