ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ
ΣΕ
ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ
ΠΕΡΙΒΑΛΛΟΝ
(ΑΕΠΠ)

Κρυμμένα Θέματα Θεωρίας

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ (ΑΕΠΠ)

Κρυμμένα Θέματα Θεωρίας

Δημιουργία - Συγγραφή

Costas Chatzinikolas
www.CostasChatzinikolas.gr
info@CostasChatzinikolas.gr

Τελευταία Ενημέρωση: 11 Απριλίου 2016

Οδηγίες

Τα θέματα θεωρίας που περιέχει το παρόν έγγραφο, αποτελούν επιπλέον θεωρία που ανήκει στην εξεταστέα ύλη των πανελληνίων εξετάσεων, όπως αυτή είχε δοθεί από το Υπουργείο Παιδείας για το σχολικό έτος 2015 - 2016.

Πρόκειται για θέματα που πολλές φορές εντοπίζονται δύσκολα στο σχολικό βιβλίο και είναι εύκολο να ξεφύγουν της προσοχής κάποιου μαθητή. Χρησιμοποιείστε τα θέματα αυτά για μία πιο άρτια προετοιμασία στη θεωρία.

Σημειώστε τα θέματα στα οποία αντιμετωπίσατε κάποια δυσκολία και επικοινωνείστε με το συγγραφέα (www.CostasChatzinikolas.gr) για διευκρινίσεις.

ΚΑΛΗ ΠΡΟΕΤΟΙΜΑΣΙΑ

Κεφάλαιο 2
Βασικές έννοιες αλγορίθμων

Από ποια σύμβολα αποτελείται ένα διαγράμμα ροής:
Ένα διάγραμμα ροής αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια ή λειτουργία. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών. Τα κυριότερα χρησιμοποιούμενα γεωμετρικά σχήματα είναι:

- Έλλειψη, που δηλώνει την αρχή και το τέλος του αλγορίθμου.
- Ρόμβος, που δηλώνει μία ερώτηση με δύο ή περισσότερες εξόδους προς απαντηση.
- Ορθογώνιο, που δηλώνει την εκτέλεση μιας ή περισσοτέρων πράξεων.
- Πλάγιο παραλληλόγραμμο, που δηλώνει είσοδο ή έξοδο στοιχείων.

Ποια η διαφορά ανάμεσα σε μία εκτελεστέα και μια δηλωτική εντολή:
Δηλωτική εντολή είναι αυτή που χρησιμοποιείται για να δηλώσει κάτι, χωρίς να περιγράφει καμία ενέργεια πάνω σε οποιοδήποτε στοιχείο. Είναι οι εντολές που χρησιμοποιούνται συνήθως στο τμήμα δηλώσεων ενός προγράμματος για να δηλώσουν π.χ.

- Το όνομα του προγράμματος (Πρόγραμμα).Κρυμμένα
- Διάφορες σταθερές, ή μεταβλητές (Σταθερές, Μεταβλητές, Ακέραιες: κτλπ).
- Την αρχή ή και το τέλος ενός προγράμματος (Αρχή, Τέλος_Προγράμματος κτλπ).

Εκτελεστέα εντολή είναι αυτή που περιγράφει κάποια ενέργεια επάνω σε κάποιο στοιχείο. Όταν εκτελεστεί επιφέρει αλλαγές στην τιμή κάποιων μεγεθών (μεταβλητών) ή επιτρέπει τη είσοδο δεδομένων ή την έξοδο αποτελεσμάτων όπως π.χ.

- Διάβασε x
- x <— x + 1
- Εκτύπωσε x

Τι είναι οι σταθερές:
Με τον όρο σταθερές αναφερόμαστε σε προκαθορισμένες τιμές που παραμένουν αμετάβλητες σε όλη τη διάρκεια εκτέλεσης ενός προγράμματος. Οι σταθερές διακρίνονται σε:

- Αριθμητικές
- Αλφαριθμητικές
- Λογικές

Τι είναι οι μεταβλητές:
Μια μεταβλητή είναι ένα γλωσσικό αντικείμενο, που χρησιμοποιείται για να αναπαραστήσει ένα στοιχείο δεδομένου. Στη μεταβλητή εκχωρείται μία τιμή, η οποία μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του προγράμματος. Ανάλογα με το είδος της τιμής που μπορούν να λάβουν οι μεταβλητές διακρίνονται σε:

- Αριθμητικές
- Αλφαριθμητικές
- Λογικές

Τι είναι οι τελεστές:
Πρόκειται για γνωστικά σύμβολα που χρησιμοποιούνται στις διάφορες πράξεις. Οι τελεστές διακρίνονται σε:

- Αριθμητικούς
- Λογικούς
- Συγκριτικούς

Τι είναι οι εκφράσεις:
Οι εκφράσεις διαμορφώνονται από τους τελεστέους, που είναι σταθερές και μεταβλητές και από τους τελεστές. Η διεργασία αποτίμησης μιας έκφρασης συνίστατια στην απόδοση τιμών στις μεταβλητές και στην εκτέλεση των πράξεων. Η τελική τιμή μιας έκφρασης εξαρτάται από την ιεραρχία των πράξεων και τη χρήση παρενθέσεων. Μια έκφραση μπορεί να αποτελείται από μόνο μία μεταβλητή ή σταθερά μέχρι μια πολύπλοκη μαθηματική σχέση.

Τι είναι ο βρόχος:
Ο βρόχος είναι το τμήμα του αλγορίθμου που επαναλαμβάνεται, δηλαδή αυτό που βρίσκεται μέσα σε μία δομή επανάληψης.

Τι είναι ο ατέρμων βρόχος:
Ατέρμων βρόχος είναι αυτός ο βρόχος, ο οποίος επαναλαμβάνεται επ' άπειρον, λόγω της τιμής της συνθήκης στη δομή επανάληψης. Ένας ατέρμων βρόχος όταν προκύπτει, αποτελεί λογικό λάθος.

Τι είναι ο πολλαπλασιασμός αλά ρωσικά:
Πρόκειται για τη μέθοδο που χρησιμοποιείται από τον Η/Υ για πολλαπλασιασμό 2 ακεραίων αριθμών. Πιο συγκεκριμένα, οι 2 αριθμοί γράφονται δίπλα – δίπλα και ο πρώτος διπλασιάζεται, ενώ ο δεύτερος υποδιπλασιάζεται με ακέραια διαίρεση - div - (αγνοώντας δηλαδή το δεκαδικό μέρος στο πηλίκο). Τελικά το γινόμενο ισούται με το άθροισμα των αριθμών που προκύπτουν από το διπλασιασμό του πρώτου, όταν ο αντίστοιχος αριθμός που προκύπτει από τον υποδιπλασιασμό του δευτέρου είναι περιττός.

Έστω Μ1, Μ2 οι δύο αριθμοί που θέλουμε να πολλαπλασιάσουμε. Ο αλγόριθμος είναι ο ακόλουθος:

```
απ <— 0
Όσο (Μ2 > 0) επανέλαβε
    Αν (Μ2 mod 2 = 1) τότε
        απ <— απ + Μ1
    Τέλος_Αν
    Μ1 <— Μ1 * 2
    Μ2 <— Μ2 div 2
Τέλος_Επανάληψης
Γράψε απ
```

Χωρίς βλάβη της γενικότητας θεωρούμε ότι οι δύο ακέραιοι είναι θετικοί, αλλά η μέθοδος μπορεί εύκολα να μετατραπεί ώστε να περιγράψει και την περίπτωση των αρνητικών ακεραίων.

Τι είναι η ολίσθηση (Shift):
Στα κυκλώματα του υπολογιστή τα δεδομένα αποθηκεύονται σε δυαδική μορφή, δηλαδή 0 και 1, ανεξάρτητα από το πώς τα ορίζει ο προγραμματιστής (όπως ακέραια, πραγματικά, χαρακτήρες ή λογικά δεδομένα). Ο αριθμός 17 στο δεκαδικό σύστημα αντιστοιχεί για παράδειγμα στην ακολουθία 00010001 του δυαδικού συστήματος. Δηλαδή:

(17)10 = (00010001)2

Αν μετακινήσουμε τα ψηφία αυτά κατά μία θέση προς τα αριστερά (shift left), δηλαδή αν προσθέσουμε ένα 0 στο τέλος του αριθμού και αγνοήσουμε το αρχικό 0, τότε θα προκύψει ο αριθμός 00100010 ο οποίος αντιστοιχεί στον αριθμό 34 του δεκαδικού συστήματος

(00100010)2 = (34)10 (πολλαπλασιάσαμε το 17 επί 2)

Αν μετακινήσουμε τα ψηφία αυτά κατά μία θέση προς τα δεξιά (shift right), δηλαδή αν κόψουμε το τελευταίο 1 και εισάγουμε ένα 0 στην αρχή, τότε θα προκύψει ο αριθμός 00001000 ο οποίος αντιστοιχεί στον αριθμό 8 του δεκαδικού συστήματος

(00001000)2 = (8)10 (διαιρέσαμε το 17 δια 2, με ακέραια διαίρεση, δηλαδή div)

Επομένως, στο δυαδικό σύστημα:

- η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί 2, και
- η ολίσθηση προς τα δεξιά ισοδυναμεί με ακέραια διαίρεση (div) δια 2

Ποια η σχέση πολλαπλασιασμού αλά ρωσικά και ολίσθησης:
Η μέθοδος πολλαπλασιασμού αλά ρωσικά χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ότι ο γνωστός μας χειρωνακτικός τρόπος πολλαπλασιασμού. Πιο συγκεκριμένα απαιτεί πολλαπλασιασμό επί δύο, διαίρεση δια δύο και πρόσθεση. Σε αντίθεση η γνωστή μας διαδικασία πολλαπλασιασμού απαιτεί πολλαπλασιασμό με οποιοδήποτε ακέραιο και πρόσθεση. Σε επίπεδο, λοιπόν, κυκλωμάτων υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση δια δύο μπορούν να υλοποιηθούν ταχύτατα με μια εντολή ολίσθησης (shift), σε αντίθεση με τον πολλαπλασιασμό με οποιοδήποτε ακέραιο που θεωρείται πιο χρονοβόρα διαδικασία.

Κεφάλαιο 3
Δομές δεδομένων και αλγόριθμοι

Γιατί υπάρχουν αρκετές διαφορετικές δομές δεδομένων:
Συνηθέστατα παρατηρείται το φαινόμενο μία δομή δεδομένων να είναι αποδοτικότερη από μία άλλη δομή με κριτήριο κάποια λειτουργία, για παράδειγμα την αναζήτηση, αλλά λιγότερο αποδοτική για κάποια άλλη λειτουργία, για παράδειγμα την εισαγωγή. Αυτές οι παρατηρήσεις εξηγούν αφ' ενός την ύπαρξη διαφερετικών δομών, και αφ' ετέρου τη σπουδαιότητα επιλογής της κατάλληλης δομής κάθε φορά.

Με ποιο κριτήριο επιλέγουμε μέθοδο αναζήτησης:
Υπάρχουν αρκετές μέθοδοι αναζήτησης σε πίνακα που εξαρτώνται κυρίως από το, αν ο πίνακας είναι ταξινομημένος ή όχι. Μια άλλη παράμετρος είναι, αν ο πίνακας περιέχει στοιχεία που είναι όλα διάφορα μεταξύ τους ή όχι.

Που χρησιμοποιείται η σειριακή αναζήτηση:
Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερο αποτελεσματική μέθοδος αναζήτησης. Έτσι, δικαιολογείται η χρήση της μόνο σε περιπτώσεις όπου:

- ο πίνακας είναι μη ταξινομημένος
- ο πίνακας είναι μικρού μεγέθους (π.χ. n <= 20)
- η αναζήτηση σε ένα συγκεκριμένο πίνακα γίνεται σπάνια

Τι είναι οι δομές δεδομένων δευτερεύουσας μνήμης¨
Σε μεγάλες πρακτικές εμπορικές/επιστημονικές εφαρμογές, το μέγεθος της κύριας μνήμης δεν επαρκεί για την αποθήκευση των δεδομένων. Στην περίπτωση αυτή χρησιμοποιούνται ειδικές δομές για την αποθήκευση των δεδομένων στη δευτερεύουσα μνήμη, δηλαδή κυρίως στο μαγνητικό δίσκο. Οι ειδικές αυτές δομές ονομάζονται αρχεία (files).

Μια σημαντική διαφορά μεταξύ κύριας μνήμης και μαγνητικού δίσκου είναι ότι στην περίπτωση του δίσκου, τα δεδομένα δεν χάνονται, αν διακοπεί η ηλεκτρική παροχή. Έτσι, τα δεδομένα των αρχείων διατηρούνται ακόμη και μετά τον τερματισμό ενός προγράμματος, κάτι που δεν συμβαίνει στην περίπτωση των δομών της κύριας μνήμης, όπως είναι οι πίνακες, όπου τα δεδομένα χάνονται όταν τελειώσει το πρόγραμμα.

Τα στοιχεία ενός αρχείου ονομάζονται εγγραφές (records), όπου κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία (fields), που ταυτοποιούν την εγγραφή, και από άλλα πεδία που περιγράφουν διάφορα χαρακτηριστικά της εγγραφής. Για παράδειγμα, έστω η εγγραφή ενός μαθητή με πεδία:

- Αριθμός Μητρώου
- Ονοματεπώνυμο
- Έτος Γέννησης
- Τάξη
- Τμήμα

Το πεδίο Αριθμός Μητρώου ταυτοποιεί την εγγραφή και ονομάζεται πρωτεύον κλειδί (primary key) ή απλά κλειδί. Το πεδίο Ονοματεπώνυμο επίσης ταυτοποιεί την εγγραφή και γι' αυτό αποκαλείται δευτερεύον κλειδί (secondary key), αν υπάρχει πρωτεύον κλειδί.

Το πρόβλημα της αναζήτησης (searching) μιας εγγραφής με βάση την τιμή του πρωτεύοντος ή ενός δευτερεύοντος κλειδιού σε αρχεία είναι ιδιαίτερα ενδιαφέρον, αν ληφθεί υπ' όψη η μεγάλη ποικιλία των χαρακτηριστικών τόσο της δομής (για παράδειγμα, στατική ή δυναμική, τρόπος οργάνωσης, μέσο αποθήκευσης κτλπ), του τύπου των δεδομένων (για παράδειγμα, ακέραιοι, κείμενο, χαρτογραφικά δεδομένα, χρονοσειρές κτλπ), όσο και της αναζήτησης (δηλαδή, με βάση το πρωτεύον ή το δευτερεύον κλειδί κτλπ).

Υπάρχουν άλλες μέθοδοι ταξινόμησης πέραν της φυσσαλίδας:
Για την ταξινόμηση δεδομένων, έχουν εκπονηθεί πάρα πολλοί αλγόριθμοι. Άλλοι σχετικά απλοί αλγόριθμοι είναι η ταξινόμηση με επιλογή και η ταξινόμηση με παρεμβολή. Ο πιο γρήγορος αλγόριθμος ταξινόμησης είναι η "γρήγορη ταξινόμηση" (quicksort). Η ταξινόμηση φυσσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο αργός αλγόριθμος ταξινόμησης.

Κεφάλαιο 7 & 8
Βασικά στοιχεία προγραμματισμού - Επιλογή και επανάληψη

Επιλογή γλώσσας προγραμματισμού:
Η επιλογή της κατάλληλης γλώσσας δεν είναι εύκολη και εξαρτάται από το είδος του προγράμματος, το διαθέσιμο εξοπλισμό και σαφώς τις γνώσεις και τις ιδιαίτερες προτιμήσεις του προγραμματιστή. Συχνά το ίδιο πρόβλημα μπορεί να λυθεί εξίσου ικανοποιητικά με πολλές διαφορετικές γλώσσες προγραμματισμό.

Τι πρέπει να έχουμε υπόψη μας κατά την επιλογή γλώσσας:
- Κάθε γλώσσα προγραμματισμού σχεδιάζεται για συκγεκριμένο σκοπό, δίνοντας έμφαση σε ορισμένα χαρακτηριστικά σε βάρος κάποιων άλλων. Δεν υπάρχει καλύτερη γλώσσα προγραμματισμού, απλά υπάρχει γλώσσα καταλληλότερη για την ανάπτυξη συγκεκριμένου τύπου εφαρμογών.
- Καθώς η γλώσσα εξελίσσεται, κάποια χαρακτηριστικά της αλλάζουν αρκετά συχνά και εξαρτώνται σε μεγάλο βαθμό από τον εξοπλισμό και το λειτουργικό σύστημα. Οι νεότερες γλώσσες διαθέτουν πλουσιότερο ρεπερτόριο εντολών και δυνατοτήτων, χωρίς όμως να προσθέτουν τίποτα στην εκμάθηση της δημιουργίας σωστών προγραμμάτων.
- Σχεδόν όλες οι γλώσσες έχουν κοινά χαρακτηριστικά, επεξεργάζονται κατά κανόνα τους ίδιους τύπους δεδομένων, υποστηρίζουν τις ίδιες βασικές δομές και έχουν παρόμοιες εντολές.

Πώς είναι σχεδιασμένη η ΓΛΩΣΣΑ:
Η ΓΛΩΣΣΑ είναι σχεδιασμένη έτσι ώστε να αποτελέσει ένα εργαλείο προγραμματισμού κατάλληλο για εκπαιδευτικούς σκοπούς. Περιέχει τα χαρακτηριστικά, τις δομές και τις εντολές που περιέχονται σε διάφορες σύγχρονες γλώσσες προγραμματισμού όπως η Pascal, Visual Basic, C, C++, Java και άλλες, χωρίς όμως να ασχολείται με τις τεχνικές λεπτομέρειες αυτών.

Πώς καταχωρούνται τα δεδομένα στη μνήμη του υπολογιστή:
Στην πραγματικότητα τα δεδομένα καταχωρούνται στη μνήμη του υπολογιστή καταλαμβάνοντας συγκεκριμένο αριθμό θέσεων (bytes). Ανάλογα με τον τύπο του δεδομένου και το διατιθέμενο αριθμό bytes ποικίλει και το εύρος τιμών που μπορούν να λάβουν. Έτσι στον υπολογιστή διαθέτουμε ένα υποσύνολο ακεραίων ή πραγματικών αριθμών. Συνήθεις τύποι δεδομένων σε διάφορες γλώσσες προγραμματισμού είναι ο ακέραιος (integer) σε 1, 2 ή 4 bytes και ο πραγματικός (real) σε 4 ή 8 bytes.

Πώς χρησιμοποιούνται οι παρενθέσεις:
Πάντα πρέπει να χρησιμοποιούνται ζεύγη παρενθέσεων. Διαφορετικός αριθμός αριστερών από δεξιές παρενθέσεις στην ίδια έκφραση είναι ένα από τα πιο συνηθισμένα (συντακτικά) λάθη.

Τι πρέπει να ισχύει σε μια εντολή εκχώρησης:
Σε μια εντολή εκχώρησης η μεταβλητή και η έκφραση πρέπει να είναι του ιδίου τύπου δεδομένων.

Πώς μπορεί να διασπαστεί μία εντολή σε πάνω από μία γραμμές:
Αν μια εντολή πρέπει να συνεχιστεί και στην επόμενη γραμμή, τότε ο πρώτος χαρακτήρας αυτής της γραμμής πρέπει να είναι ο χαρακτήρας &.

Πώς αποτιμάται μια έκφραση όπου συνδυάζονται διάφοροι τελεστές:
Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μια έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες. Ακόμη, οι λογικοί τελεστές έχουν χαμηλότερη προτεραιότητα από τους συγκριτικούς.

Ποιοι κανόνες πρέπει να ακολουθούνται στη χρήση εμφωλευμένων βρόχων:

- Ο εσωτερικός βρόγχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόγχος που ξεκινάει τελευταίος, πρέπει να ολοκληρώνεται πρώτος.
- Η είσοδος σε κάθε βρόγχο υποχρεωτικά γίνεται από την αρχή του.
- Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόγχων που ο ένας βρίσκεται στο εσωτερικό του άλλου (βλέπε ταξινόμηση με φυσαλίδα).

Τι είναι η τιμή φρουρός:
Εϊναι μία τιμή που χρησιμοποιείται για τον τερματισμό μιας επαναληπτικής διαδικασίας. Συνήθως πρόκειται για αυθαίρετη επιλογή του προγραμματιστή και αποτελεί σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή είναι τέτοια ώστε να μην είναι λογικά σωστή για το πρόβλημα.

Για παράδειγμα στο πρόβλημα:
Να γραφεί πρόγραμμα το οποίο να διαβάζει από το πληκτρολόγιο συνεχώς αριθμούς, μη μηδενικούς, και να υπολογίζει και να τυπώνει το άθροισμα και το μέσο όρο τους. Ως τέλος της διαδικασίας εισαγωγής στοιχείων να χρησιμοποιηθεί η τιμή 0.

Η τιμή μηδέν είναι η τιμή φρουρός. Η τιμή που θα σημάνει τη λήξη της επανάληψης και τον τερματισμό του προγράμματος.

Κεφάλαιο 9
Πίνακες

Τι ονόματα δίνουμε στους δείκτες ενός πίνακα:
Ο δείκτης ενός πίνακα είναι μια μεταβλητή που μπορεί να έχει οποιοδήποτε αποδεκτό όνομα. Είναι σύνηθες όμως στον προγραμματισμό ως δείκτες να χρησιμοποιούνται οι μεταβλητές i, j, k.

Τι συμβαίνει με τους πίνακες με διάσταση μεγαλύτερη του δύο:
Εκτός από μονοδιάστατους και δισδιάστατους πίνακες υπάρχουν πίνακες με περισσότερες διαστάσεις τρισδιάστατοι, τετραδιάστατοι και γενικά πολυδιάστατοι, ανάλογα με τον αριθμό των δεικτών που χρησιμοποιούνται για τον καθορισμό των στοιχείων. Ωστόσο τα περισσότερα προβλήματα αντιμετωπίζονται με τη χρήση πινάκων μονοδιάστατων ή δισδιάστατων.

Τι είναι η δυαδική αναζήτηση:
Η δυαδική αναζήτηση χρησιμοποιείται μόνο σε ταξινομημένους πίνακες και είναι σαφώς αποδοτικότερη από τη σειριακή μέθοδο.

Πόσο σημαντικές είναι οι τυπικές επεξεργασίες σε πίνακες:
Για τις τυπικές επεξεργασίες πινάκων έχουν ανατυχθεί αρκετοί αλγόριθμοι και η μελέτη τους αποτελεί έναν από τους σημαντικότερους τομείς της αλγοριθμικής.

Κεφάλαιο 10
Υποπρογράμματα

Η λίστα παραμέτρων σε ένα υποπρόγραμμα είναι υποχρεωτική:
Η λίστα παραμέτρων δεν είναι υποχρεωτική. Μία διαδικασία θα μπορούσε να διαβάζει τα δεδομένα της, ενώ μία συνάρτηση θα μπορούσε να δημιουργεί δεδομένα μόνη της π.χ. με τη χρήση γεννήτριας τυχαίων αριθμών.

Ποια η χρήση των πραγματικών και τυπικών παραμέτρων:
Η λίστα των τυπικών παραμέτρων (formal parameter list) καθορίζει τις παραμέτρους στη δήλωση του προγράμματος.
Η λίστα των πραγματικών παραμέτρων (actual parameter list) καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.

Ποια η εμβέλεια (ισχύς) των μεταβλητών στα προγράμματά μας σε ΓΛΩΣΣΑ:
Όλες οι μεταβλητές ενός υποπρογράμματος ή προγράμματος έχουν εμβέλεια μόνο εντός του προγράμματος ή υποπρογράμματος ή αλλιώς, όπως λέγεται, είναι τοπικές μεταβλητές. Γι' αυτό και το υποπρόγραμμα έχει ξεχωριστό τμήμα δηλώσεων.

Αυτό πρακτικά σημαίνει ότι η τιμή μιας μεταβλητής του υποπρογράμματος (ακόμη κι αν υπάρχει μεταβλητή με το ίδιο όνομα στο κύριο πρόγραμμα) είναι γνωστή μόνο μέσα στο υποπρόγραμμα.

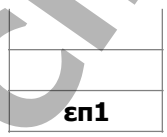
Ορίσματα (!):
Ναι, μερικές γλώσσες προγραμματισμού ονομάζουν ορίσματα τις τυπικές παραμέτρους και απλά παραμέτρους τις πραγματικές παραμέτρους.

Ποια η χρήση της στοίβας στην κλήση διαδικασιών:
Η έννοια της στοίβας είναι πολύ χρήσιμη στο λογισμικό των γλωσσών προγραμματισμού.
Όταν στο κυρίως πρόγραμμα μας έχουμε κλήση μιας διαδικασίας ή συνάρτησης, ο έλεγχος όπως λέμε περνάει στο υποπρόγραμμα. Εκείνη τη στιγμή και για όσο χρονικό διάστημα το υποπρόγραμμα λειτουργεί (τρέχει), το κυρίως πρόγραμμα δεν εκτελείται. Όταν όμως το υποπρόγραμμα τερματίσει, τότε ο έλεγχος περνάει ξανά στο κυρίως πρόγραμμα (εκτός αν εν τω μεταξύ έχουν προηγηθεί κλήσεις άλλων υποπρογραμμάτων), και η εκτέλεση του κυρίως προγράμματος πρέπει να συνεχίσει από την αμέσως επόμενη εντολή από αυτήν που περιείχε την κλήση στο υποπρόγραμμα.
Αυτή η εντολή (ή η διεύθυνσή της στη μνήμη καλύτερα) ονομάζεται διεύθυνση επιστροφής και αποθηκεύεται (**ωθείται**) από το μεταφραστή σε μια στοίβα που ονομάζεται **στοίβα χρόνου εκτέλεσης** τη στιγμή που γίνεται η κλήση στο υποπρόγραμμα.
Όταν τελειώσει η εκτέλεση του υποπρογράμματος, η διεύθυνση επιστροφής λαμβάνεται (**απωθείται**) από τη στοίβα και έτσι ο έλεγχος μεταφέρεται πάλι στο κυρίως πρόγραμμα.
Αυτή η τεχνική εφαρμόζεται πάντα όταν το κυρίως πρόγραμμα καλεί ένα υποπρόγραμμα, ή όταν ένα υποπρόγραμμα καλεί άλλο υποπρόγραμμα κοκ.

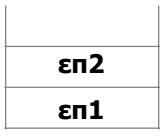
Παράδειγμα:
Το κυρίως πρόγραμμα καλεί μια διαδικασία Α, η οποία με τη σειρά της καλεί μία άλλη διαδικασία Β.

```
        Πρόγραμμα Τεστ                Διαδικασία Διαδ_Α          Διαδικασία Διαδ_Β
        .............                 .............              .............
        ΚΑΛΕΣΕ Διαδ_Α                 .............              .............
επ1     .............                 ΚΑΛΕΣΕ Διαδ_Β              .............
        .............          επ2    .............              .............
        Τέλος_Προγράμματος            Τέλος_Διαδικασίας          Τέλος_Διαδικασίας
```

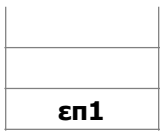
Όταν το κυρίως πρόγραμμα καλέσει τη Διαδ_Α, τότε η διεύθυνση της επόμενης εντολής επ1, ωθείται στη στοίβα χρόνου εκτέλεσης η οποία θα έχει τη μορφή:



Καθώς εκτελείται η Διαδ_Α και τη στιγμή που θα καλέσει τη Διαδ_Β, η διεύθυνση της επόμενης εντολής της, δηλ. επ2 ωθείται και αυτή στη στοίβα χρόνου εκτέλεσης η οποία έχει πλέον τη μορφή:



Μετά το τέλος εκτέλεσης της Διαδ_Β, από τη στοίβα απωθείται πρώτο το επ2, οπότε ο έλεγχος επιστρέφει στη Διαδ_Α και η εκτέλεση της συνεχίζεται. Η μορφή που έχει η στοίβα χρόνου εκτέλεσης θα είναι:



Αφού τελειώσει και η Διαδ_Α, απωθείται από τη στοίβα το επ1, οπότε ο έλεγχος επιστρέφει στο κυρίως πρόγραμμα και παραμένει εκεί μέχρι να τελειώσει η εκτέλεσή του. Η στοίβα πλέον είναι άδεια.

Ποιους κανόνες πρέπει να ακολουθούν οι λίστες παραμέτρων:
Οι λίστες των παραμέτρων πρέπει να ακολουθούν τους εξής κανόνες:

- Ο αριθμός των πραγματικών και τυπικών παραμέτρων πρέπει να είναι ο ίδιος.
- Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση. Για παράδειγμα η πρώτη της λίστας των τυπικών παραμέτρων στην πρώτη της ίστας των πραγματικών παραμέτρων κοκ.
- Η τυπική παράμετρος και η αντίστοιχη της πραγματική πρέπει να είναι του ιδίου τύπου.

Ποιες οι διαφορές Διαδικασιών – Συναρτήσεων:

- Τρόπος κλήσης: Οι διαδικασίες καλούνται με τη χρήση της δεσμευμένης λέξης ΚΑΛΕΣΕ, οι συναρτήσεις με χρήση του ονόματός τους.
- Πλήθος επιστρεφόμενων τιμών: Οι διαδικασίες μπορούν να επιστρέφουν καμία, μία ή περισσότερες τιμές, οι συναρτήσεις επιστρέφουν ακριβώς μία τιμή.
- Τρόπος επιστροφής τιμών: Οι διαδικασίες επιστρέφουν τιμές (εφόσον επιστρέφουν) μέσω των παραμέτρων τους, οι συναρτήσεις επιστρέφουν μία τιμή μέσω του ονόματος τους. Άρα το όνομα μιας συνάρτησης είναι ουσιαστικά η παράμετρος εξόδου.
- Δυνατότητες: Οι διαδικασίες μπορούν να εκτελέσουν όλες τις λειτουργίες που μπορεί να εκτελέσει ένα πρόγραμμα, ενώ οι συναρτήσεις υπολογίζουν και επιστρέφουν μόνο μία τιμή.