

2019

# Αλγόριθμος

ΚΕΦΑΛΑΙΟ 10

Algorithmos Computer  
Science Laboratory



How To Become  
**BETTER  
PROGRAMMER**



## [ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ]

[Επιλεγμένα θέματα θεωρίας σύμφωνα με την ύλη 2018-2019 στην ενότητα του  
Τμηματικού Προγραμματισμού]

## Πίνακας περιεχομένων

1. Τι ονομάζουμε τμηματικό προγραμματισμό .....	3
2. Ποια είναι η σχέση μεταξύ τμηματικού & δομημένου προγραμματισμού.....	3
3. Τι είναι το υποπρόγραμμα;.....	3
4. Ποια είναι η βασική προϋπόθεση ώστε ένα σύνθετο πρόγραμμα να χωριστεί σε υποπρογράμματα;.....	3
5. Χαρακτηριστικά (ιδιότητες) των υποπρογραμμάτων. ....	3
6. Πλεονεκτήματα της χρήσης του τμηματικού προγραμματισμού;.....	4
7. Τι ονομάζουμε κλήση υποπρογράμματος ; .....	5
8. Τι ονομάζουμε παράμετρο υποπρογράμματος ; .....	5
9. Ποια τα είδη των υποπρογραμμάτων και ποιες οι λειτουργίες επιτελούν;.....	5
10. Να δώσετε τον ορισμό της διαδικασίας και της συνάρτησης. ....	6
11. Ποια είναι η δομή μιας συνάρτησης;.....	6
12. Πώς Ενεργοποιείται (καλείται) μια συνάρτηση;.....	6
13. Ποια είναι η δομή μιας διαδικασίας;.....	6
14. Πώς Ενεργοποιείται (καλείται) μια διαδικασία; .....	7
15. Τι είναι κύριο πρόγραμμα;.....	7
16. Ποιες είναι οι διαφορές μεταξύ διαδικασίας και συνάρτησης; .....	7
17. Ποιες παράμετροι χαρακτηρίζονται ως πραγματικές και ποιες ως τυπικές; .....	7
18. Κανόνες που πρέπει να ακολουθούν οι λίστες των παραμέτρων .....	7
20. χρήση των μεταβλητών που δηλώνονται στα διάφορα τμήματα προγραμμάτων; .....	8
21. Ποια είναι η ισχύς κάθε μεταβλητής στη ΓΛΩΣΣΑ; .....	8
22. Τι ονομάζεται εμβέλεια σε προγραμματιστικό περιβάλλον; .....	8
23. Απεριόριστη εμβέλεια τα μειονεκτήματα της χρήσης της .....	8
24. Τι γνωρίζετε για την περιορισμένη εμβέλεια & ποια τα πλεονεκτήματα της χρήσης της .	9
26. Ποια είναι η ισχύς κάθε μεταβλητής στη ΓΛΩΣΣΑ; .....	9
27. Πότε επιλέγουμε την χρήση συναρτήσεων; .....	10
28. Πότε επιλέγουμε την χρήση διαδικασιών; .....	10
29. Μπορεί ένα υποπρόγραμμα να κληθεί από ένα άλλο υποπρόγραμμα ;.....	10
30. Πώς χρησιμοποιείται η στοίβα στην κλήση υποπρογραμμάτων; .....	10

### 1. Τι ονομάζουμε τμηματικό προγραμματισμό ; (Εξετάσεις 2011)

Τμηματικό προγραμματισμό ονομάζουμε την τεχνική σχεδίασης και ανάπτυξης υποπρογραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων. Είναι από τα βασικότερα συστατικά του δομημένου προγραμματισμού και σε μεγάλο βαθμό εξασφαλίζει επιτυχή και εύκολη δημιουργία σωστών προγραμμάτων. Υλοποιεί τη φιλοσοφία της ιεραρχικής σχεδίασης.

### 2. Ποια είναι η σχέση μεταξύ τμηματικού & δομημένου προγραμματισμού;

Η τεχνική του τμηματικού προγραμματισμού αποτελεί ένα από τα βασικότερα συστατικά του δομημένου προγραμματισμού, ο οποίος εξασφαλίζει σε μεγάλο βαθμό την επιτυχή και εύκολη δημιουργία σωστών προγραμμάτων.

### 3. Τι είναι το υποπρόγραμμα; Δώστε παραδείγματα. (Εξετάσεις 2011)

Το υποπρόγραμμα αποτελεί αυτόνομο και ανεξάρτητο τμήμα προγράμματος. Ωστόσο, το υποπρόγραμμα πρέπει συχνά να επικοινωνεί είτε με το υπόλοιπο πρόγραμμα, που ονομάζουμε κύριο πρόγραμμα, είτε με άλλα υποπρογράμματα μέσω μιας λειτουργίας που ονομάζεται κλήση υποπρογράμματος. Για παράδειγμα, η είσοδος των δεδομένων ενός προβλήματος. Ο υπολογισμός του μέσου όρου αριθμών, η εκτύπωση κάποιου δεδομένου εξόδου.

### 4. Ποια είναι η βασική προϋπόθεση ώστε ένα σύνθετο πρόγραμμα να χωριστεί σε υποπρογράμματα;

Ο χωρισμός ενός προγράμματος σε υποπρογράμματα προϋποθέτει την ανάλυση του αρχικού προβλήματος σε μικρότερα, απλούστερα υποπροβλήματα, τα οποία να μπορούν να αντιμετωπιστούν ανεξάρτητα το ένα από το άλλο. Αν και δεν υπάρχουν συγκεκριμένοι κανόνες για την επιτυχή εφαρμογή του τμηματικού προγραμματισμού, αυτή τελικά επιτυγχάνεται με προσεκτική μελέτη στην ανάλυση του προβλήματος, εμπειρία στον προγραμματισμό, ταλέντο και φυσικά γνώσεις.

### 5. Να αναφέρετε τρία χαρακτηριστικά (ιδιότητες) των υποπρογραμμάτων. (Εξετάσεις 2003, 2005 και 2009)

- **Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο.**  
Στην πραγματικότητα κάθε υποπρόγραμμα ενεργοποιείται με την είσοδο σε αυτό που γίνεται πάντοτε από την αρχή του, εκτελεί ορισμένες ενέργειες και απενεργοποιείται με την έξοδο από αυτό, που γίνεται πάντοτε από το τέλος του.
- **Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα.**  
Αυτό σημαίνει ότι κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα χωρίς να επηρεαστούν άλλα υποπρογράμματα. Στη πράξη βέβαια η απόλυτη ανεξαρτησία είναι δύσκολο να επιτευχθεί.

- **Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο.**

Η έννοια του μεγάλου υποπρογράμματος είναι υποκειμενική, αλλά πρέπει κάθε υποπρόγραμμα να είναι τόσο ώστε να είναι εύκολα κατανοητό, για να μπορεί να ελέγχεται. Γενικά κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία. Αν εκτελεί περισσότερες λειτουργίες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.

## 6. Ποια τα πλεονεκτήματα της χρήσης του τμηματικού προγραμματισμού;

Ο σωστός χωρισμός ενός σύνθετου προγράμματος σε υποπρογράμματα εξασφαλίζει τέσσερα βασικά χαρακτηριστικά του σωστού προγραμματισμού:

- **Διευκολύνει την ανάπτυξη του αλγόριθμου και του αντίστοιχου προγράμματος.**  
 Επιτρέπει την εξέταση και την επίλυση απλών προβλημάτων και όχι την αντιμετώπιση του συνολικού προβλήματος. Με τη σταδιακή επίλυση των υποπροβλημάτων και τη δημιουργία των αντίστοιχων υποπρογραμμάτων τελικά επιλύεται το συνολικό πρόβλημα.
- **Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.**  
 Ο χωρισμός του προγράμματος σε μικρότερα αυτοτελή τμήματα επιτρέπει την γρήγορη διόρθωση ενός συγκεκριμένου τμήματος του χωρίς οι αλλαγές να επηρεάσουν όλο το υπόλοιπο πρόγραμμα. Επίσης διευκολύνει οποιονδήποτε χρειαστεί να διαβάσει και να κατανοήσει τον τρόπο που λειτουργεί το πρόγραμμα. Όπως έχει πολλές φορές τονιστεί, αυτό είναι πού σημαντικό χαρακτηριστικό του σωστού προγραμματισμού, αφού ένα μεγάλο πρόγραμμα στον κύκλο της ζωής του χρειάζεται να συντηρηθεί από διαφορετικούς προγραμματιστές.
- **Απαιτεί λιγότερο χρόνο και προσπάθεια στην συγγραφή του προγράμματος.**  
 Πολύ συχνά χρειάζεται η ίδια λειτουργία σε διαφορετικά σημεία ενός προγράμματος. Από τη στιγμή που ένα υποπρόγραμμα έχει γραφεί, μπορεί το ίδιο να καλείται από πολλά σημεία του προγράμματος. Έτσι μειώνονται το μέγεθος του προγράμματος, ο χρόνος που απαιτείται για τη συγγραφή του και οι πιθανότητες λάθους, ενώ ταυτόχρονα το πρόγραμμα γίνεται πιο εύληπτο και κατανοητό.
- **Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.**  
 Ένα υποπρόγραμμα που έχει γραφεί μπορεί να χρησιμοποιηθεί πού εύκολα και σε άλλα υποπρογράμματα. Από τη στιγμή που έχει δημιουργηθεί, η χρήση του δεν διαφέρει από τη χρήση των ενσωματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού, όπως για τον υπολογισμό του ημίτονου ή του συνημίτονου ή την εντολή με την οποία εκτελεί μία συγκεκριμένη διαδικασία, για παράδειγμα γράφει στην οθόνη (εντολή γράψε). Αν λοιπόν, χρειάζεται συχνά κάποια λειτουργία που δεν υποστηρίζεται απ' ευθείας από τη γλώσσα, όπως για παράδειγμα η εύρεση του μικρότερου δύο αριθμών, τότε μπορεί να γραφεί το αντίστοιχο υποπρόγραμμα. Η συγγραφή πολλών υποπρογραμμάτων και η δημιουργία βιβλιοθηκών με αυτά ουσιαστικά επεκτείνει την ίδια τη γλώσσα προγραμματισμού.

*I. Να αναφέρετε τέσσερα πλεονεκτήματα του τμηματικού προγραμματισμού.*

*II. Να αναπτύξετε δύο από τα παραπάνω πλεονεκτήματα του τμηματικού προγραμματισμού. (Εξετάσεις 2004)*

- I, II. α) Διευκολύνει την ανάπτυξη αλγορίθμου και του αντίστοιχου προγράμματος,  
 β) διευκολύνει την κατανόηση και τη διόρθωση του προγράμματος,  
 γ) απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος και  
 δ) επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.

III. *Να αιτιολογήσετε γιατί ένα από τα πλεονεκτήματα του τμηματικού προγραμματισμού είναι το ότι απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.* (Εξετάσεις 2015)

III. Επαναχρησιμοποίηση κώδικα με την επανάκληση υποπρογραμμάτων. Μειώνεται το μέγεθος του υποπρογράμματος και τα λάθη, ενώ το πρόγραμμα καθίσταται πιο κατανοητό.

### 7. Τι ονομάζουμε κλήση υποπρογράμματος ;

Κλήση υποπρογράμματος ονομάζουμε τη διαδικασία με την οποία ένα υποπρόγραμμα ενεργοποιείται είτε μέσα από ένα άλλο υποπρόγραμμα είτε μέσα από το κύριο πρόγραμμα.

### 8. Τι ονομάζουμε παράμετρο υποπρογράμματος ;

Τα υποπρογράμματα συνήθως δέχονται τιμές από το τμήμα του προγράμματος που τα καλεί και μετά την εκτέλεση επιστρέφει σε αυτό νέες τιμές, αποτελέσματα. Οι τιμές αυτές που περνούν από το ένα υποπρόγραμμα στο άλλο λέγονται παράμετροι. Οι παράμετροι λοιπόν είναι σαν τις κοινές μεταβλητές ενός προγράμματος , με μία ουσιώδη διαφορά : χρησιμοποιούνται για να περνούν τιμές στα υποπρογράμματα .

**Συνοπτικά**, μία παράμετρος είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.

### 9. Ποια τα είδη των υποπρογραμμάτων και ποιες οι λειτουργίες που μπορούν να εκτελέσουν ;

Υπάρχουν δύο είδη υποπρογραμμάτων και το είδος του κάθε υποπρογράμματος καθορίζεται από το είδος της λειτουργίας που καλείται να επιτελέσει . Συγκεκριμένα, τα είδη υποπρογραμμάτων είναι τα ακόλουθα,

**Οι διαδικασίες**, που μπορούν να εκτελέσουν οποιαδήποτε λειτουργία από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα , δηλαδή:

- Να εισάγουν δεδομένα
- Να εκτελέσουν υπολογισμούς
- Να μεταβάλλουν τις τιμές των μεταβλητών
- Να τυπώσουν αποτελέσματα .

Επιπλέον με τη χρήση των παραμέτρων μπορούν να μεταφέρουν τις τιμές των μεταβλητών τους σε άλλα υποπρογράμματα ή στο κύριο πρόγραμμα .

**Οι συναρτήσεις**. Η λειτουργία των συναρτήσεων είναι πιο περιορισμένη από αυτή των διαδικασιών. Συγκεκριμένα, οι συναρτήσεις υπολογίζουν μόνο μια τιμή, αριθμητική , λογική ή χαρακτήρα και μόνο αυτήν επιστρέφουν στο υποπρόγραμμα – κύριο πρόγραμμα που τις κάλεσε. Οι συναρτήσεις μοιάζουν με τις συναρτήσεις των Μαθηματικών και η χρήση τους είναι όμοια με τη χρήση των ενσωματωμένων συναρτήσεων που υποστηρίζει η γλώσσα προγραμματισμού.

## 10. Να δώσετε τον ορισμό της διαδικασίας και της συνάρτησης.

- Η συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομα της (όπως οι μαθηματικές συναρτήσεις).
- Η διαδικασία είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελέσει όλες τις λειτουργίες ενός προγράμματος.

## 11. Ποια είναι η δομή μιας συνάρτησης;

Η δομή μιας συνάρτησης είναι αντίστοιχη με εκείνη ενός προγράμματος: αρχικά χρησιμοποιείται η δεσμευμένη λέξη ΣΥΝΑΡΤΗΣΗ και ένα έγκυρο όνομα σε ΓΛΩΣΣΑ. Στη συνέχεια παρατίθενται οι παράμετροι εισόδου και δηλώνεται ο τύπος της συνάρτησης, δηλαδή ο τύπος της τιμής την οποία επιστρέφει (οι αποδεκτές μορφές δεδομένων της ΓΛΩΣΣΑΣ). Ακολουθεί η δήλωση των σταθερών (αν υπάρχουν) και των μεταβλητών, όπως και σε ένα πρόγραμμα.

```

ΣΥΝΑΡΤΗΣΗ
Όνομα(παράμετροι): τύπος
ΣΤΑΘΕΡΕΣ
.....
ΜΕΤΑΒΛΗΤΕΣ
.....
ΑΡΧΗ
    Εντολές σε ΓΛΩΣΣΑ
.....
    Όνομα <- έκφραση
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
    
```

Στη συνέχεια τοποθετείται η δεσμευμένη λέξη με ολοκληρώνεται με τη δεσμευμένη λέξη ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ. Στο τμήμα εντολών της συνάρτησης πρέπει υποχρεωτικά να υπάρχει μια εντολή εκχώρησης της τιμής που θα επιστραφεί στο όνομα της συνάρτησης.

## 12. Πώς ενεργοποιείται (καλείται) μια συνάρτηση;

Μια συνάρτηση μπορεί να κληθεί από οποιοδήποτε τμήμα προγράμματος: το κύριο πρόγραμμα, κάποια διαδικασία ή κάποια άλλη συνάρτηση. Η κλήση (ενεργοποίησή) της πραγματοποιείται με την εμφάνιση του ονόματός της (ακολουθούμενου από τις παραμέτρους της) σε μια οποιαδήποτε έκφραση. Η έκφραση αυτή μπορεί να είναι λογική (συνθήκες) ή αριθμητική (σε μια εντολή εκχώρησης ή σε μια εντολή εξόδου ΓΡΑΨΕ).

## 13. Ποια είναι η δομή μιας διαδικασίας;

Η δομή μιας διαδικασίας είναι αντίστοιχη με εκείνη ενός προγράμματος: Αρχικά χρησιμοποιείται η δεσμευμένη λέξη ΔΙΑΔΙΚΑΣΙΑ και ένα έγκυρο όνομα σε ΓΛΩΣΣΑ. Στη συνέχεια παρατίθενται οι παράμετροι (εισόδου και εξόδου). Ακολουθεί η δήλωση των σταθερών (αν υπάρχουν) και των μεταβλητών, όπως και σε ένα πρόγραμμα. Στη συνέχεια τοποθετείται η δεσμευμένη λέξη ΑΡΧΗ και το τμήμα εντολών. Το υποπρόγραμμα ολοκληρώνεται με τη δεσμευμένη λέξη ΤΕΛΟΣ ΔΙΑΔΙΚΑΣΙΑΣ.

```

ΔΙΑΔΙΚΑΣΙΑ Όνομα (παράμετροι)
ΣΤΑΘΕΡΕΣ
.....
ΜΕΤΑΒΛΗΤΕΣ
.....
ΑΡΧΗ
    .....
    Εντολές σε ΓΛΩΣΣΑ
    .....
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

#### 14. Πώς Ενεργοποιείται (καλείται) μια διαδικασία;

Μια διαδικασία μπορεί να κληθεί από οποιοδήποτε τμήμα προγράμματος: το κύριο πρόγραμμα ή κάποιο άλλο υποπρόγραμμα. Η κλήση (ενεργοποίηση) μιας διαδικασίας πραγματοποιείται με την εντολή ΚΑΛΕΣΕ, που ακολουθείται από το όνομα της διαδικασίας και τη λίστα των παραμέτρων μέσα σε παρενθέσεις.

#### 15. Τι είναι κύριο πρόγραμμα;

Σύμφωνα με τη φιλοσοφία του τμηματικού προγραμματισμού, κάθε ανεξάρτητη λειτουργία, ενότητα ενός αλγορίθμου αναπτύσσεται σε ξεχωριστό τμήμα προγράμματος (υποπρόγραμμα): διαδικασία ή συνάρτηση. Το πρόγραμμα πλέον έχει τον ρόλο του συντονισμού και της ενεργοποίησης - κλήσης των υποπρογραμμάτων με σκοπό την επίλυση του προβλήματος (αποκαλείται και κύριο πρόγραμμα).

#### 16. Ποιες είναι οι διαφορές μεταξύ διαδικασίας και συνάρτησης;

Η συνάρτηση υπολογίζει και επιστρέφει μία τιμή με το όνομά της, ενώ η διαδικασία μπορεί να επιστρέφει πολλές τιμές (για παράδειγμα, έναν πίνακα). Ακόμη, απαιτούν διαφορετικό τρόπο κλήσης: η συνάρτηση συμμετέχει με το όνομά της σε κάποια αλγοριθμική έκφραση, ενώ η διαδικασία ενεργοποιείται με τη δεσμευμένη λέξη ΚΑΛΕΣΕ. Η διαδικασία έχει τη δυνατότητα αλληλεπίδρασης με τον χρήστη (εντολές εισόδου / εξόδου), ενώ η φιλοσοφία των συναρτήσεων δεν είναι αυτή. Μια συνάρτηση δέχεται τιμές και επιστρέφει μία τιμή, όπως οι συναρτήσεις της ΓΛΩΣΣΑΣ.

#### 17. α) Ποιες παράμετροι χαρακτηρίζονται ως πραγματικές και ποιες ως τυπικές; β) Πώς γίνεται η ανταλλαγή τιμών ανάμεσα στις τυπικές και στις πραγματικές παραμέτρους; Δώστε ένα παράδειγμα.

**α.** Οι τυπικές παράμετροι χρησιμοποιούνται κατά τη δήλωση του υποπρογράμματος. Οι πραγματικές παράμετροι χρησιμοποιούνται κατά την κλήση του υποπρογράμματος.

**β.** Αντιστοιχίζονται μία προς μία, κάθε πραγματική παράμετρος με την αντίστοιχη τυπική. Επιπρόσθετα:

- Σε μια διαδικασία κατά την κλήση της περνούν οι τιμές από τις πραγματικές παραμέτρους στις αντίστοιχες τυπικές, ενώ, όταν ολοκληρωθεί η εκτέλεση της, όλες οι τυπικές παράμετροι επιστρέφουν τις τιμές τους στις αντίστοιχες πραγματικές.

- Σε μια συνάρτηση όλες οι πραγματικές παράμετροι δίνουν τις τιμές τους στις αντίστοιχες τυπικές. Όταν ολοκληρωθεί η εκτέλεση της συνάρτησης, επιστρέφει μια τιμή με το όνομα της.

#### 18. Να αναφέρετε τους κανόνες που πρέπει να ακολουθούν οι λίστες των παραμέτρων κατά την κλήση ενός υποπρογράμματος. (Εξετάσεις 2006, 2010)

α. Το πλήθος των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι το ίδιο,

β. Κάθε πραγματική αντιστοιχίζεται στην τυπική που βρίσκεται στην ίδια θέση,

γ. Η πραγματική και η αντίστοιχη τυπική παράμετρος πρέπει να έχουν τον ίδιο τύπο.



### 19. Ποια η διαφορά μεταξύ:

- α. μεταβλητών και παραμέτρων;
- β. τυπικών και πραγματικών παραμέτρων; (Εξετάσεις 2007)

**α.** Οι παράμετροι είναι μεταβλητές, αλλά χρησιμοποιούνται για το πέρασμα τιμών από ένα τμήμα προγράμματος σε ένα άλλο. Οι υπόλοιπες μεταβλητές ενός υποπρογράμματος δεν έχουν αυτή την ιδιότητα, αλλά χρησιμοποιούνται για την αποθήκευση και την επεξεργασία δεδομένων, **β.** Κατά την κλήση ενός υποπρογράμματος οι τυπικές παράμετροι είναι μεταβλητές του υποπρογράμματος που καλείται, ενώ οι πραγματικές παράμετροι είναι μεταβλητές του τμήματος προγράμματος που καλεί.

### 20. Ποια είναι η χρήση των μεταβλητών που δηλώνονται στα διάφορα τμήματα προγραμμάτων;

- Σε ένα πρόγραμμα οι μεταβλητές χρησιμοποιούνται για την αποθήκευση στοιχείων. Κάποιες από αυτές χρησιμοποιούνται για την κλήση υποπρογραμμάτων ως πραγματικές παράμετροι.
- Σε κάποιο υποπρόγραμμα κάποιες μεταβλητές χρησιμοποιούνται ως τυπικές παράμετροι για τη δήλωση του υποπρογράμματος, ενώ αξιοποιούνται στο τμήμα εντολών του υποπρογράμματος. Ωστόσο στο σώμα εντολών του υποπρογράμματος μπορεί να καλείται κάποιο άλλο υποπρόγραμμα, συνεπώς κάποιες μεταβλητές του μπορεί να χρησιμοποιούνται ως πραγματικές παράμετροι. Οι υπόλοιπες μεταβλητές του υποπρογράμματος χρησιμοποιούνται για τις ανάγκες της κωδικοποίησης.

### 21. Ποια είναι η ισχύς κάθε μεταβλητής στη ΓΛΩΣΣΑ;

Κάθε μεταβλητή στη ΓΛΩΣΣΑ είναι γνωστή, έχει ισχύ, όπως λέγεται, μόνο στο τμήμα προγράμματος που έχει δηλωθεί, ισχύει δηλαδή τοπικά για το συγκεκριμένο υποπρόγραμμα ή για το κύριο πρόγραμμα. Όλες οι μεταβλητές (και οι σταθερές) είναι τοπικές στο συγκεκριμένο τμήμα προγράμματος.

### 22. Τι ονομάζεται εμβέλεια σε προγραμματιστικό περιβάλλον; Τι εμβέλεια χρησιμοποιείται στη ΓΛΩΣΣΑ; (Εξετάσεις 2017)

Το τμήμα του προγράμματος στο οποίο ισχύουν οι μεταβλητές λέγεται εμβέλεια (scope μεταβλητών). Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι τοπικές, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν. Στη ΓΛΩΣΣΑ έχουμε περιορισμένη εμβέλεια.

### 23. Τι γνωρίζετε για την απεριόριστη εμβέλεια και ποια είναι τα μειονεκτήματά της χρήσης της ;

Σύμφωνα με αυτή την αρχή, όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος άσχετα από πού δηλώθηκαν. Όλες δηλαδή οι μεταβλητές είναι **καθολικές**.

Τα μειονεκτήματά της απεριόριστης εμβέλειας είναι τα ακόλουθα:



- Καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων.
- Δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη η χρήση της για μεγάλα προγράμματα τα οποία χρησιμοποιούν άλλα υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα .

## 24. Τι γνωρίζετε για την περιορισμένη εμβέλεια και ποια τα πλεονεκτήματα της χρήσης της ;

Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα του προγράμματος να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι **τοπικές**, ισχύουν δηλαδή για το πρόγραμμα - υποπρόγραμμα στο οποίο δηλώθηκαν. (Στη ΓΛΩΣΣΑ δηλαδή έχουμε περιορισμένη εμβέλεια.)

Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι τα ακόλουθα:

- Προσφέρει απόλυτη αυτονομία όλων των υποπρογραμμάτων .
- Δίνει τη δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα μεταβλητής/σταθεράς χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

### 25. α. Ποιες μεταβλητές ονομάζονται καθολικές;

*β. Η χρήση καθολικών μεταβλητών σε ένα πρόγραμμα καταστρατηγεί μία από τις βασικές αρχές του τμηματικού προγραμματισμού (ιδιότητες που πρέπει να διακρίνουν τα υποπρογράμματα). Να αναφέρετε ποια είναι αυτή η ιδιότητα και να εξηγήσετε γιατί καταστρατηγείται. (Εξετάσεις 2016)*

Σύμφωνα με αυτή την αρχή, όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα με το πού δηλώθηκαν. Όλες οι μεταβλητές είναι καθολικές. Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

## 26. Ποια είναι η ισχύς κάθε μεταβλητής στη ΓΛΩΣΣΑ;

### **Απεριόριστη εμβέλεια**

Σύμφωνα με αυτή την αρχή, όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα με το πού δηλώθηκαν. Όλες οι μεταβλητές είναι καθολικές. Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

### **Περιορισμένη εμβέλεια**

Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι τοπικές, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν. Στη ΓΛΩΣΣΑ έχουμε περιορισμένη εμβέλεια. Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων και η δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

### Μερικώς περιορισμένη εμβέλεια

Σύμφωνα με αυτή την αρχή, άλλες μεταβλητές είναι τοπικές και άλλες καθολικές. Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις με τις οποίες ορίζονται οι μεταβλητές ως τοπικές ή καθολικές. Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά πλεονεκτήματα στον πεπειραμένο προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.

Συμπεράσματα		
Μεταβλητές	Ισχύς	Εμβέλεια
Τοπικές	Ισχύουν μόνο εκεί που δηλώθηκαν.	Περιορισμένη
Καθολικές	Έχουν ισχύ σε όλα τα προγράμματα/υποπρογράμματα που τις χρησιμοποιούν.	Απεριόριστη

### 27. Πότε επιλέγουμε την χρήση συναρτήσεων;

Συναρτήσεις χρησιμοποιούμε στις περιπτώσεις που ζητείται υποπρόγραμμα το οποίο δεν απαιτεί την εκτέλεση εντολών γράψε ή διάβασε και συγχρόνως πρέπει να επιστρέψει αποκλειστικά μια τιμή.

### 28. Πότε επιλέγουμε την χρήση διαδικασιών;

Η χρήση διαδικασίας είναι υποχρεωτική στις περιπτώσεις που ζητείται υποπρόγραμμα στο οποίο απαιτείται διάβασμα ή εμφάνιση στοιχείων .Επίσης πρέπει να χρησιμοποιηθεί διαδικασία αν το υποπρόγραμμα πρέπει να επιστρέψει στο κύριο πρόγραμμα - υποπρόγραμμα που την κάλεσε είτε πίνακα είτε περισσότερες της μίας τιμής.

Πρέπει όμως να υπενθυμίσουμε ότι με διαδικασία μπορούν να υλοποιηθούν όλα τα υποπρογράμματα τα οποία μπορούν να υλοποιηθούν με συναρτήσεις. Παρ' όλα αυτά θα δούμε ότι συχνά η χρήση συναρτήσεων, όπου βέβαια αυτή είναι δυνατή, οδηγεί σε πιο κομψά και κατανοητά προγράμματα.

### 29. Μπορεί ένα υποπρόγραμμα να κληθεί από ένα άλλο υποπρόγραμμα ;

Κάθε διαδικασία ή συνάρτηση μπορεί να καλείται είτε από άλλη συνάρτηση ή διαδικασία είτε φυσικά από το κύριο πρόγραμμα . Παρ' όλα αυτά, στην περίπτωση που μία συνάρτηση καλεί μία διαδικασία, τότε η τελευταία δεν πρέπει να περιέχει εντολές **εισόδου ή εξόδου**.

### 30. Πώς χρησιμοποιείται η στοίβα στην κλήση υποπρογραμμάτων;

Όταν μια διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κύριου προγράμματος, που ονομάζεται διεύθυνση επιστροφής, αποθηκεύεται από τον μεταγλωττιστή σε μια στοίβα που ονομάζεται στοίβα χρόνου εκτέλεσης. Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης η διεύθυνση επιστροφής απωθείται από τη στοίβα και έτσι ο έλεγχος μεταφέρεται και πάλι στο κύριο πρόγραμμα. Αυτή η τεχνική χρησιμοποιείται σε όλες τις κλήσεις υποπρογραμμάτων. Η στοίβα είναι μια λίστα στοιχείων/πληροφοριών που λειτουργεί όπως μια στοίβα πιάτων: κάθε νεοεισερχόμενο πιάτο εισέρχεται πάντα στην κορυφή αλλά και κάθε πιάτο που φεύγει από την κορυφή.