Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κεντρικής Μακεδονίας - Σέρρες
Τμήμα Μηχανικών Πληροφορικής

Προγραμματισμός ΙΙ (Θ)

Δρ. Δημήτρης Βαρσάμης
Επίκουρος Καθηγητής

Μάρτιος 2017

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΙΙ (Θ)

Συναρτήσεις - Εισαγωγή I

Ο βασικός τρόπος για να αναπτύξουμε και να συντηρήσουμε ένα πρόγραμμα, είναι να το σχεδιάσουμε και στην συνέχεια να το υλοποιήσουμε σε υποπρογράμματα.

Αυτά τα υποπρογράμματα στη γλώσσα C υλοποιούνται με συναρτήσεις (functions).

Σε αυτήν την ενότητα θα δούμε πως σχεδιάζουμε τις συναρτήσεις, ποια είδη συναντάμε πιο συχνά και πως αυτό μας ωφελεί σε μεγάλα και πολύπλοκα προγράμματα.

Στην ανάπτυξη προγραμμάτων χρησιμοποιούμε συναρτήσεις γιατί θέλουμε ο κώδικας μας να είναι απλός και κατανοητός και εν συνεχεία να μπορούμε να τον συντηρήσουμε και να τον εξελίξουμε με μεγαλύτερη ευκολία.

Συναρτήσεις - Εισαγωγή II

Οι συναρτήσεις επιτρέπουν τον προγραμματιστή να χωρίσει ένα πρόγραμμα σε υπομονάδες και σε κάθε συνάρτηση θα πρέπει να αναπτύξουμε **μια** και μόνο **λειτουργία** του προγράμματος μας.

Κάτι σημαντικό που πρέπει να προσέξουμε και να δώσουμε ιδιαίτερη βάση είναι ότι οι μεταβλητές που χρησιμοποιούμε στις συναρτήσεις είναι με **τοπική εμβέλεια**. Δηλαδή, έχουν οντότητα μόνο μέσα στη συνάρτηση.

Συναρτήσεις - Εισαγωγή I

Υπάρχει όμως κάποιος συγκεκριμένος τρόπος, να προσδιορίσουμε από την αρχή τις απαραίτητες βασικές λειτουργίες?

Προφανώς Όχι! Δεν υπάρχει κάποια συγκεκριμένη μεθοδολογία.

Εάν κάνουμε σωστή μελέτη και καταγράψουμε τις απαιτήσεις μας πριν ξεκινήσουμε να γράφουμε τον κώδικα θα διευκολυνθούμε αρκετά στην δημιουργία συναρτήσεων.

Αυτός ο τρόπος μας δίνει την δυνατότητα να κατανοήσουμε καλύτερα την δομή του προγράμματος και έτσι επιτυγχάνουμε τον διαχωρισμό του σε κομμάτια.

Αυτά τα κομμάτια θα είναι οι βασικές λειτουργίες τις οποίες θα γράψουμε σε ξεχωριστές συναρτήσεις με το κατάλληλο όνομα.

Συναρτήσεις - Εισαγωγή II

Χρησιμοποιώντας λοιπόν τις συναρτήσεις έχει αποδειχθεί ότι πρώτον μειώνονται τα λάθη μας και δεύτερον όταν προκύπτουν καινούργιες απαιτήσεις, μπορούμε να τις ενσωματώσουμε πάρα πολύ εύκολα δημιουργώντας απλά μια καινούργια συνάρτηση ή ενημερώνοντας μια προηγούμενη εφόσον αυτό είναι εφικτό.

Φυσικά πέρα από τις συναρτήσεις που θα δημιουργήσουμε εμείς, η C μας παρέχει από μόνη της πολλές έτοιμες συναρτήσεις, με τις οποίες καλύπτουμε κάποιες βασικές ανάγκες. Για παράδειγμα, τέτοιες συναρτήσεις είναι οι:

- `strcat()` που ενώνει δύο αλφαριθμητικά,
- η `memcpy()` που αντιγράφει την τοποθεσία της μνήμης σε άλλη,
- η `find()` που κάνει αναζήτηση,
- η `sort()` που κάνει ταξινόμηση κ.ο.κ

Παράδειγμα 1

Να δημιουργηθεί πρόγραμμα σε C το οποίο
να δέχεται 10 ακέραιους αριθμούς
να υπολογίζει και να εμφανίζει

- το μέσο όρο σύμφωνα με τον τύπο

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- το εύρος σύμφωνα με τον τύπο

$$R = \max - \min$$

- τη διακύμανση σύμφωνα με τον τύπο

$$s^2 = \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\right)$$

Παράδειγμα 1 I

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int i,a,sum=0,sum2=0,min,max;
    for(i=1;i<=10;i=i+1) {
        printf("Give a number :");
        scanf("%d",&a);
        sum=sum+a;
        sum2=sum2+a*a;
        if((a>max)||(i==1)){
            max=a;
        }
        if((a<min)||(i==1)){
            min=a;
        }
    }
    printf("Mean = %f\n",(float)sum/10);
```

Παράδειγμα 1 II

```
    printf("Range = %d\n",max-min);
    printf("S2= %f\n",((float)(sum2-(sum*sum)/10)/10));
    system("PAUSE");
    return 0;
}
```


Παράδειγμα 2

Να δημιουργηθεί πρόγραμμα σε C το οποίο

- να δέχεται 10 ακέραιους βαθμούς $[1, 10]$
- να υπολογίζει και να εμφανίζει το μεγαλύτερο βαθμό
- να υπολογίζει και να εμφανίζει το μέσο όρο

Παράδειγμα 2 I

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int i,a,sum=0,max;
    for(i=1;i<=10;i=i+1){
        do{
            printf("Give a grade :");
            scanf("%d",&a);
        }while((a<1)||(a>10));
        sum=sum+a;
        if((a>max)||(i==1)){
            max=a;
        }
    }
    printf("Max = %d\n",max);
    printf("Mean = %f\n",(float)sum/10);
    system("PAUSE");
```


Παράδειγμα 2 II

```
    return 0;
}
```

Παράδειγμα 3

Να δημιουργηθεί πρόγραμμα σε C το οποίο

- να δέχεται ακέραιους αριθμούς και να σταματά όταν δοθεί ως βαθμός η τιμή μηδέν
- να υπολογίζει και να εμφανίζει το μεγαλύτερο βαθμό
- να υπολογίζει και να εμφανίζει το μέσο όρο

Παράδειγμα 3 I

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int i,a,sum=0,max;
    printf("Give a grade :");
    scanf("%d",&a);
    max=a;
    while(a!=0){
        i=i+1;
        sum=sum+a;
        if(a>max){
            max=a;
        }
        printf("Give a grade :");
        scanf("%d",&a);
    }
    printf("Max = %d\n",max);
```

Παράδειγμα 3 II

```
    printf("Mean = %f\n",(float)sum/i);
    system("PAUSE");
    return 0;
}
```


Συναρτήσεις - Δήλωση, Ορισμός, Κλήση I

Ας μάθουμε κάποιες βασικές λειτουργίες που ισχύουν για όλες τις συναρτήσεις ανεξαρτήτου τύπου.

Οι συναρτήσεις είναι ανεξάρτητα κομμάτια κώδικα τα οποία δεν μπορούν να λειτουργήσουν από μόνα τους εκτός και αν κληθούν από κάποια άλλη συνάρτηση.

Πρέπει όμως να προσέξουμε ότι και η `main()` είναι μια συνάρτηση της οποίας το όνομα δεν αλλάζει και από αυτήν ξεκινάει αυτόματα η εκτέλεση του κυρίως προγράμματός μας. Αν λείπει η συνάρτηση `main()` από ένα πρόγραμμα σε C, τότε το πρόγραμμα δεν μπορεί να εκτελέσει τις εντολές του άρα κατ' επέκταση δεν θα λειτουργήσει.

Οι ενσωματωμένες συναρτήσεις της C για να εκτελεστούν πρέπει να δηλωθούν και να ορισθούν, διαδικασία που επιτυγχάνεται με την οδηγία `#include <stdio.h>`

Συναρτήσεις - Δήλωση, Ορισμός, Κλήση II

- το αρχείο κεφαλίδας `stdio.h` περιέχει την δήλωση και τον ορισμό κάποιων συναρτήσεων
- επομένως, ανάλογα με τις ενσωματωμένες συναρτήσεις της C που θέλει ο προγραμματιστής να χρησιμοποιήσει θα πρέπει να "φορτώσει" και την οδηγία με το κατάλληλο αρχείο κεφαλίδας.

- να δηλωθεί η κεφαλίδα της συνάρτησης στην αρχή του προγράμματος
- και να οριστεί η συνάρτηση με άλλα λόγια να γραφεί το σώμα της στο τέλος του προγράμματος

Συναρτήσεις - Δήλωση I

Μία καινούργια συνάρτηση οριζόμενη από τον προγραμματιστή πρέπει πρώτα να την δηλώσουμε.

Η δήλωση μιας καινούργιας συνάρτησης γίνεται **πάντα** πριν από το κυρίως πρόγραμμα (δηλ. πριν από την `main()`).

Ο γενικός τρόπος **δήλωσης** μιας συνάρτησης γίνεται ως εξής:

τύπος_συνάρτησης όνομα_συνάρτησης (**τύπος_ορίσματος** όρισμα);

- Ο **τύπος_συνάρτησης** είναι ο τύπος του στοιχείου που θα επιστρέφει η συνάρτηση που δημιουργήσαμε. Θα πρέπει πάντα να υπάρχει ένας τύπος συνάρτησης, ακόμη και στις συναρτήσεις χωρίς επιστρεφόμενη τιμή ο τύπος της θα είναι **`void`**.
  Εάν δεν προσδιορίσουμε κάποιον τύπο κατά την δήλωση της συνάρτησης, τότε αυτή θα επιστρέψει έναν αριθμό είτε έναν χαρακτήρα, ο οποίος θα μεταφραστεί αυτόματα σε αριθμό.

Συναρτήσεις - Δήλωση II

Αυτό που πρέπει να προσέξουμε είναι πως οι τύποι της συνάρτησης αλλά και των ορισμάτων που θα δούμε παρακάτω, πρέπει να είναι ίδιοι με αυτούς που περιμένει να δεχτεί το κυρίως πρόγραμμα.
Σε περίπτωση ασυμφωνίας θα δημιουργηθούν πολύπλοκα και σοβαρά προβλήματα στον κώδικα μας. Το σύνηθες είναι να μην τρέξει το πρόγραμμα και ο μεταγλωττιστής να μας ενημερώσει για το λάθος.

- Το **όνομα_συνάρτησης** προσδιορίζεται από τον προγραμματιστή. Καλό είναι όπως αναφέραμε και προηγουμένως το όνομα να είναι κάτι εύστοχο που θα προσδιορίζει ακριβώς την λειτουργία που θα εκτελεί η συνάρτηση.
  Το όνομα της συνάρτησης ακολουθεί τους κανόνες της ονοματολογίας των μεταβλητών.

Συναρτήσεις - Δήλωση III

- Ο **τύπος_ορίσματος** είναι ανάλογος με το όρισμα μας. Δηλαδή, η δήλωση των ορισμάτων γίνεται όπως και η δήλωση των μεταβλητών, πρώτα προσδιορίζουμε τον τύπο και μετά το όνομα.
- Τα ονόματα από τα **ορίσματα** των συναρτήσεων είναι και πάλι επιλογή του προγραμματιστή και δεν έχουνε κάποια σύνδεση με τα ονόματα του κυρίως προγράμματος γιατί η εμβέλεια τους είναι όπως αναφέραμε τοπική.

Συναρτήσεις - Ορισμός I

Μία καινούργια συνάρτηση οριζόμενη από τον προγραμματιστή εφόσον την δηλώσουμε πρέπει να την ορίσουμε.

Η ορισμός μιας καινούργιας συνάρτησης γίνεται **πάντα** μετά από το κυρίως πρόγραμμα (δηλ. πριν από την `main()`). Ο γενικός τρόπος **ορισμού** μιας συνάρτησης γίνεται ως εξής:

τύπος_συνάρτησης όνομα_συνάρτησης (**τύπος_ορίσματος** όρισμα)
{

Σώμα Συνάρτησης

}

Συναρτήσεις - Ορισμός II

Επίσης, ίσως η πιο **βασική λειτουργία** μιας συνάρτησης παρέχεται από την εντολή **`return`**. Βρίσκεται στον κύριο κορμό της συνάρτησης και είναι ο σύνδεσμος που επικοινωνεί με το κυρίως πρόγραμμα. Οποιαδήποτε λειτουργία και αν έχουμε γράψει στην συνάρτηση το κυρίως πρόγραμμα θα λάβει την τιμή που επιστρέφει η εντολή **`return`**.

Συναρτήσεις - Κλήση I

Μία καινούργια συνάρτηση οριζόμενη από τον προγραμματιστή εφόσον την δηλώσουμε και την ορίσουμε μπορούμε να την καλέσουμε.

Η κλήση μιας καινούργιας συνάρτησης γίνεται είτε από το κυρίως πρόγραμμα είτε από άλλη συνάρτηση με τον ίδιο τρόπο με τον οποίο καλούμε τις ενσωματωμένες συναρτήσεις της γλώσσας C.

Τύποι Συναρτήσεων

Οι τύποι συναρτήσεων που θα αναπτύξουμε είναι οι παρακάτω:

Με επιστρεφόμενη τιμή.

Χωρίς επιστρεφόμενη τιμή (`void`).

Αναδρομικές συναρτήσεις.

Συνάρτηση με επιστρεφόμενη τιμή I

Οι συναρτήσεις με επιστρεφόμενη τιμή χωρίζονται σε δύο κατηγορίες:

σε αυτές με ορίσματα και

σε αυτές χωρίς ορίσματα

Ας δούμε πρώτα πώς λειτουργούν οι συναρτήσεις που **δέχονται ορίσματα**.

Στον προγραμματισμό τέτοιου είδους συναρτήσεις είναι οι πιο διαδεδομένες.

Συνήθως θέλουμε να δώσουμε κάποια στοιχεία σε μία συνάρτηση.

Αυτή με την σειρά της θα τα υποβάλει σε κάποια επεξεργασία και έπειτα θα τα επιστρέψει.

Ο ορισμός τους γίνεται ακριβώς όπως είδαμε παραπάνω.

Συνάρτηση με επιστρεφόμενη τιμή II

Συνάρτηση με επιστρεφόμενη τιμή I

Δήλωση συνάρτησης με ορίσματα και επιστρεφόμενη τιμή:

```
int max(int a , int b);
```

Ορισμός (κατασκευή) συνάρτησης με ορίσματα και επιστρεφόμενη τιμή:

```
int max(int a , int b){
    if (a>b){
        return a;
    }
    else{
        return b;
    }
}
```


Συνάρτηση με επιστρεφόμενη τιμή II

Κλήση συνάρτησης με ορίσματα και επιστρεφόμενη τιμή:

```
    int a1, a2, m;
    printf("Give a number :");
    scanf("%d", &a1);
    printf("Give a number :");
    scanf("%d" ,&a2);
    m=max(a1, a2);
    printf("Max = %d\n", m);
```

Συνάρτηση με επιστρεφόμενη τιμή I

Το πρόγραμμα θα γίνει

```
#include <stdio.h>
#include <stdlib.h>

int max(int a , int b);

int main(){
    int a1, a2, m;
    printf("Give a number :");
    scanf("%d", &a1);
    printf("Give a number :");
    scanf("%d", &a2);
    m=max(a1, a2);
    printf("Max = %d\n", m);
    system("PAUSE");
    return 0;
}
```


Συνάρτηση με επιστρεφόμενη τιμή II

```

int max(int a , int b){
    if (a>b){
        return a;
    }
    else{
        return b;
    }
}
```

Συνάρτηση με επιστρεφόμενη τιμή I

Καλό είναι η κατασκευή της συνάρτησης να γίνεται μετά το κυρίως πρόγραμμα. Δεν είναι λάθος να γίνει πιο πριν αλλά σε μεγάλα προβλήματα με πολλές συναρτήσεις το πιο πιθανό είναι να δημιουργηθούν προβλήματα πολυπλοκότητας.

Παρατηρούμε επίσης ότι κατά την δήλωση της συνάρτησης ακολουθεί το "**;**" ενώ στον ορισμό της όχι. Επομένως προσέχουμε να μην ξεχάσουμε ποτέ το ερωτηματικό στην δήλωση της συνάρτησης διότι όπως ακριβώς και στην δήλωση οποιαδήποτε μεταβλητής έτσι και εδώ θα δημιουργηθούν προβλήματα και δεν θα τρέξει το πρόγραμμα μας.

Συνάρτηση με επιστρεφόμενη τιμή II

Εφόσον έχουμε δηλώσει και συγγράψει επιτυχώς την συνάρτηση που θα χρησιμοποιήσουμε ήρθε η ώρα να την **καλέσουμε** στο κυρίως πρόγραμμα μας. Αυτή η διαδικασία είναι πολύ απλή, το μόνο που πρέπει να κάνουμε είναι να δημιουργήσουμε μια καινούργια μεταβλητή ιδίου τύπου με την επιστρεφόμενη τιμή της συνάρτησης, και έπειτα να εκχωρήσουμε για τιμή το όνομα της συνάρτησης μαζί με τα ορίσματα.

Συνάρτηση με επιστρεφόμενη τιμή I

Ας δούμε τώρα και την δεύτερη περίπτωση, τι συμβαίνει όταν έχουμε κάποια επιστρεφόμενη τιμή αλλά **δεν έχουμε ορίσματα**;

Συνήθως αυτό συμβαίνει όταν χρησιμοποιούμε συναρτήσεις από την πρότυπη βιβλιοθήκη[1] της C.

Ας κατασκευάσουμε όμως ένα παράδειγμα, το οποίο θα εμφανίζει τυχαίους αριθμούς απο ένα συγκεκριμένο εύρος αριθμών 1 έως 100 κάνοντας χρήση μίας πρότυπης βιβλιοθήκης.

[1] Η πρότυπη βιβλιοθήκη C είναι μια πρότυπη συλλογή από αρχεία, επικεφαλίδες και συναρτήσεις βιβλιοθήκης που υλοποιούν κάποιες κοινές λειτουργίες. Σε αντίθεση με άλλες γλώσσες η C δεν περιλαμβάνει ενσωματωμένες δεσμευμένες λέξεις για αυτές τις λειτουργίες, και για αυτόν τον λόγο σχεδόν όλα τα προγράμματα γραμμένα στην C βασίζονται στην πρότυπη βιβλιοθήκη για να λειτουργήσουν.

Συνάρτηση με επιστρεφόμενη τιμή I

Δήλωση συνάρτησης με επιστρεφόμενη τιμή αλλά χωρίς ορίσματα:

```
int randomNumber(void);
```

Ορισμος (κατασκευή) συνάρτησης με επιστρεφόμενη τιμή αλλά χωρίς ορίσματα:

```
int randomNumber(void){
    srand(time(NULL));
    return(rand() % 100 + 1);
}
```

Η **κλήση** της συνάρτησης γίνεται με τον ίδιο τρόπο που είδαμε στο προηγούμενο παράδειγμα με την μόνη διαφορά ότι στην παρένθεση δεν θα βάλουμε ορίσματα.

```
int rNumber = randomNumber();
```

Χωρίς επιστρεφόμενη τιμή (void)

Οι συναρτήσεις χωρίς επιστρεφόμενη τιμή ή αλλιώς κενού τύπου, έχουνε τύπο επιστρεφόμενης τιμής **void**, για αυτόν τον λόγο τις αποκαλούμε και τύπου **void**.

Αυτό σημαίνει ότι δεν επιστρέφουν τίποτα και για αυτόν τον λόγο μέσα στο κορμό τους **δεν έχουν** την **return**.

Συνήθως οι τύπου **void** συναρτήσεις κάνουν διάφορους υπολογισμούς και δείχνουν τα αποτελέσματα τους με την συνάρτηση printf.

Χωρίς επιστρεφόμενη τιμή (void)

Παρόλα αυτά όπως και στις συναρτήσεις με επιστρεφόμενη τιμή έτσι και εδώ έχουμε δύο κατηγορίες:

με ορίσματα και

χωρίς ορίσματα

Οι συναρτήσεις **με ορίσματα** είναι αυτές που αναφέραμε πιο πάνω. Είναι αυτές που δέχονται κάποιες παραμέτρους, κάνουν τους απαραίτητους υπολογισμούς και έπειτα εμφανίζουν το αποτέλεσμα στην οθόνη με την εντολή συνάρτηση printf.

Χωρίς επιστρεφόμενη τιμή (void) I

Για παράδειγμα ας υποθέσουμε ότι δεχόμαστε δύο τιμές και θέλουμε να βρούμε τον μέσο όρο:

Δήλωση συνάρτησης με ορίσματα αλλά χωρίς επιστρεφόμενη τιμή:

```
void average(int x, int y);
```

Ορισμός (κατασκευή) συνάρτησης με ορίσματα αλλά χωρίς επιστρεφόμενη τιμή:

```
void average(int x, int y){
    float mesosOros;
    mesosOros = (float)( x + y ) / 2 ;
    printf("Mean of %d and %d is %.2f\n",x,y,mesosOros);
}
```

Κλήση συνάρτησης με ορίσματα και χωρίς επιστρεφόμενη τιμή:

```
average(x, y);
```


Χωρίς επιστρεφόμενη τιμή (void) I

Οι συναρτήσεις **χωρίς ορίσματα** είναι οι πιο απλές συναρτήσεις που θα συναντήσουμε.

Ούτε δέχονται ούτε εμφανίζουν τιμές. Μπορούν να χρησιμοποιηθούν για να εμφανίσουν κάποιο μήνυμα ή για να εμφανίσουν πληροφορίες κ.ο.κ.

Ας εξετάσουμε ένα παράδειγμα που θα εμφανίζει κάποιο μήνυμα στην οθόνη:

Δήλωση συνάρτησης χωρίς ορίσματα και χωρίς επιστρεφόμενη τιμή:

```
void showMessage(void);
```

Ορισμός (κατασκευή) συνάρτησης χωρίς ορίσματα και χωρίς επιστρεφόμενη τιμή:

```
void showMessage(void){
    printf("This is a simple message!!!");
}
```

Χωρίς επιστρεφόμενη τιμή (void) II

```
showMessage
```