ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

Προγραμματισμός Υπολογιστών & Υπολογιστική Φυσική

Ενότητα 7: Συναρτήσεις

Νικόλαος Στεργιούλας
Τμήμα Φυσικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

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 Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την 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.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ

Μέρος 7ο

ΝΙΚΟΛΑΟΣ ΣΤΕΡΓΙΟΥΛΑΣ

ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΥΝΑΡΤΗΣΕΙΣ

Μία συνάρτηση είναι ένα ανεξάρτητο τμήμα κώδικα, που εκτελεί μία ορισμένη εργασία (και προαιρετικά επιστρέφει μία τιμή).

Ο τρόπος συγγραφής προγραμμάτων με χρήση συναρτήσεων που εκτελούν ανεξάρτητες εργασίες ονομάζεται δομημένος προγραμματισμός.

Το βασικό πλεονέκτημα είναι πως με τη χρήση συναρτήσεων ένα πρόγραμμα χωρίζεται σε μικρότερα τμήματα, άρα ο κώδικας διαβάζεται, τροποποιείται και ελέγχεται πιο εύκολα.

Όταν γραφεί μια συνάρτηση, μπορεί να χρησιμοποιηθεί ξανά σε οποιοδήποτε άλλο πρόγραμμα.

ΟΡΙΣΜΟΣ ΣΥΝΑΡΤΗΣΗΣ

Ο ορισμός μιας συνάρτησης γίνεται ως εξής:

```
τύπος_επιστροφής όνομα_συνάρτησης(τύπος όνομα1,
                    τύπος όνομα2, τύπος όνομα3, … )
{

 … /* εντολές */

}
```

Παράδειγμα:

```
int sum(int i, int j)
{ int result;

  result = i+j;

  return result;
}
```

ΟΡΙΣΜΟΣ ΣΥΝΑΡΤΗΣΗΣ

Ο τύπος επιστροφής μιας συνάρτησης μπορεί να είναι οποιοσδήποτε τύπος δεδομένων της C, εκτός από πίνακες.

Ο τύπος επιστροφής είναι void, όταν η συνάρτηση δεν επιστρέφει κάποια τιμή.

Το όνομα της συνάρτησης ακολουθείται από μια λίστα παραμέτρων.

Εάν η συνάρτηση δεν δέχεται παραμέτρους, τότε η λίστα είναι void.

Εάν ολόκληρο το πρόγραμμα που γράφουμε περιέχεται σε ένα αρχείο μόνο, τότε οι συναρτήσεις είναι καλό να ορίζονται μετά από την main().

ΔΗΛΩΣΗ ΣΥΝΑΡΤΗΣΗΣ

Εκτός από τον ορισμό της συνάρτησης μετά την main(), απαιτείται και μια δήλωση της συνάρτησης, πριν τη main().

Η δήλωση της συνάρτησης πρέπει να είναι ταυτόσημη με την πρώτη γραμμή του ορισμού της συνάρτησης, με ένα επιπλέον ; στο τέλος.

```
τύπος_επιστροφής όνομα_συνάρτησης(τύπος όνομα1,
                                   τύπος όνομα2, τύπος όνομα3, … );
```

ΠΡΟΣΟΧΗ: Όποτε αλλάζουμε κάτι στον ορισμό της συνάρτησης, πρέπει να κάνουμε την ίδια αλλαγή και στη δήλωσή τους, ώστε να υπάρχει πάντα συμφωνία.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int sum(int i, int j);
```

(ΔΗΛΩΣΗ της συνάρτησης sum)

```c
int main(void)
{
    int result;

    result = sum(1,2);
    printf("%d\n", result);

    return 0;
}
```

(ΚΥΡΙΟ μέρος του προγράμματος)

```c
int sum(int i, int j)
{
    int result;

    result = i+j;

    return result;
}
```

(ΟΡΙΣΜΟΣ της συνάρτησης sum)

ΟΡΙΣΜΑΤΑ - ΠΑΡΑΜΕΤΡΟΙ

```c
#include <stdio.h>

int sum(int i, int j);

int main(void)
{
    int result;

    result = sum(1,2);
    printf("%d\n", result);

    return 0;
}

int sum(int i, int j)
{
    int result;

    result = i+j;

    return result;
}
```

(ΟΡΙΣΜΑΤΑ της συνάρτησης sum)

(ΠΑΡΑΜΕΤΡΟΙ της συνάρτησης sum)

ΕΠΙΣΤΡΟΦΗ ΤΙΜΗΣ ΚΑΙ ΤΕΡΜΑΤΙΣΜΟΣ

Όταν μια συνάρτηση επιστρέφει κάποια τιμή με τύπο διαφορετικό από void (π.χ. τύπου float), θα πρέπει όλα τα δυνατά «μονοπάτια» της να επιστρέφουν κάποια τιμή αυτού του τύπου με μια εντολή return.

Αν η τιμή που δίνουμε στην return δεν έχει τον αναμενόμενο τύπο, τότε γίνεται αυτόματη μετατροπή σε αυτόν.

Η συνάρτηση τερματίζει στο **πρώτο** return που θα συναντήσει κατά την εκτέλεσή της!

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

float absolutediff(float a, float b);

int main(int argc, char **argv)
{
    float number1, number2;

    number1 = 4.1;
    number2 = 2.3;

    printf("%4.3e\n", absolutediff(number1,number2));

    return 0;
}

float absolutediff(float a, float b)
{
    if(a>b)
        return a-b;
    else if (a<b)
        return b-a;

    return 0;
}
```

(αν a>b, θα τερματίσει εδώ)

(αν a<b, θα τερματίσει εδώ)

(αν a=b, θα τερματίσει εδώ, ενώ θα γίνει αυτόματη μετατροπή του 0 σε 0.0)

ΚΛΗΣΗ ΣΥΝΑΡΤΗΣΗΣ ΜΕΣΩ ΑΝΑΦΟΡΑΣ

Όταν επιθυμούμε μια συνάρτηση να μπορεί να αλλάξει τις τιμές των ορισμάτων που δέχεται, τότε τα ορίσματα πρέπει να δίνονται στη συνάρτηση μέσω αναφοράς.

Δηλαδή δίνουμε στη συνάρτηση ως ορίσματα δείκτες στις διευθύνσεις μνήμης των ορισμάτων και όχι τις τιμές των ορισμάτων (η δεύτερη περίπτωση ονομάζεται κλήση μέσω τιμής).

Όταν δίνουμε έναν δείκτη ως όρισμα, η συνάρτηση έχει πλέον πρόσβαση στη διεύθυνση μνήμης που αντιστοιχεί στο όρισμα και μπορεί έτσι να αλλάξει την τιμή του ορίσματος.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

void increase(int *i_ptr);

int main(void)
{
    int i;

    i = 20;

    printf("Αρχική τιμή ορίσματος i = %d\n", i);

    increase(&i);

    printf("Τελική τιμή ορίσματος i = %d\n", i);

    return 0;
}

void increase(int *i_ptr)
{
    (*i_ptr) += 1;
}
```

το όρισμα είναι η διεύθυνση του i

η παράμετρος είναι δείκτης

αυξάνεται η τιμή του i

```
Αρχική τιμή ορίσματος i = 20
Τελική τιμή ορίσματος i = 21
```

ΕΜΒΕΛΕΙΑ ΜΕΤΑΒΛΗΤΗΣ

Εμβέλεια μιας μεταβλητής καλείται το τμήμα του προγράμματος στο οποία η μεταβλητή είναι προσβάσιμη.

Η εμβέλεια μιας μεταβλητής εξαρτάται από το σημείο δήλωσής της μέσα στο πρόγραμμα.

Τα διαφορετικά είδη μεταβλητών βάσει της εμβέλειάς των είναι:

- Οι τοπικές μεταβλητές (local variables)
- Οι καθολικές μεταβλητές (global variables)

ΤΟΠΙΚΕΣ ΜΕΤΑΒΛΗΤΕΣ

Μια μεταβλητή που δηλώνεται στο σώμα μιας συνάρτησης είναι τοπική μεταβλητή.

Η εμβέλεια μιας τοπικής μεταβλητής περιορίζεται στη συνάρτηση όπου δηλώνεται, που σημαίνει πως άλλες συναρτήσεις του προγράμματος δεν έχουν πρόσβαση σε αυτή τη μεταβλητή.

Επειδή μια τοπική μεταβλητή δεν είναι ορατή έξω από τη συνάρτηση στην οποία δηλώνεται, έχουμε την ευχέρεια να δηλώσουμε μεταβλητές με το ίδιο όνομα και σε άλλες συναρτήσεις.

ΜΝΗΜΗ ΤΟΠΙΚΩΝ ΜΕΤΑΒΛΗΤΩΝ

Όταν εκτελείται το τμήμα ενός προγράμματος που καλεί μια συνάρτηση, δεσμεύεται χώρος στη μνήμη για τις τοπικές μεταβλητές που αυτή περιέχει. Όταν τερματίζεται η συνάρτηση, αποδεσμεύεται η μνήμη αυτών των τοπικών μεταβλητών.

Εάν ξανακαλέσουμε την ίδια συνάρτηση, η τοπικές μεταβλητές θα δεσμεύσουν και πάλι κάποιο χώρο στη μνήμη.

Άρα, οι τοπικές μεταβλητές δε διατηρούν την τιμή τους ανάμεσα σε διαδοχικές κλήσης της ίδιας συνάρτησης.

ΣΤΑΤΙΚΕΣ ΜΕΤΑΒΛΗΤΕΣ

Εάν θέλουμε οπωσδήποτε κάποια τοπική μεταβλητή να διατηρεί την τιμή της ανάμεσα στις επανειλημμένες κλήσεις μιας συνάρτησης, τότε αυτή μπορεί να οριστεί ως static (στατική μεταβλητή). Για να λειτουργήσει ως static, πρέπει υποχρεωτικά να γίνεται η αρχικοποίηση της τιμής μαζί με τη δήλωση.

Προσοχή: Η αρχικοποίηση εφαρμόζεται μόνο κατά την πρώτη κλήση αυτής της συνάρτησης. Σε όλες τις επόμενες κλήσεις, η στατική μεταβλητή δεν αρχικοποιείται ξανά, αλλά διατηρεί την τιμή που είχε από την προηγούμενη κλήση.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

void increase();

int main(void)
{
    increase();
    increase();
    increase();

    return 0;
}
```

```c
void increase()
{
    int i = 0;
    static int j = 0;

    i++;
    j++;

    printf("i = %d  j = %d\n", i, j);
}
```

(αρχικοποιείται κάθε φορά)

(μόνο την πρώτη φορά)

```
i = 1  j = 1
i = 1  j = 2
i = 1  j = 3
```

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

void increase();

int main(void)
{
    increase();
    increase();
    increase();

    return 0;
}

void increase()
{
    int i = 0;
    static int j;

    j = 0;

    i++;
    j++;

    printf("i = %d  j = %d\n", i, j);
}
```

(αρχικοποίηση μετά τη δήλωση)

```
i = 1  j = 1
i = 1  j = 1
i = 1  j = 1
```

ΚΑΘΟΛΙΚΕΣ (GLOBAL) ΜΕΤΑΒΛΗΤΕΣ

Εάν μια μεταβλητή δηλώνεται έξω από οποιαδήποτε συνάρτηση (ακόμη και από την main), τότε είναι global (καθολική μεταβλητή).

Η εμβέλεια μιας καθολικής μεταβλητής εκτείνεται από το σημείο της δήλωσής της, μέχρι το τέλος του αρχείου στο οποίο δηλώνεται.

Άρα, μόνο οι συναρτήσεις που ορίζονται μετά από τη δήλωση μιας καθολικής μεταβλητής έχουν πρόσβαση σε αυτή.

Μια τοπική μεταβλητή μπορεί να έχει το ίδιο όνομα με μια καθολική μεταβλητή.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

void add_one();
void subtract_one();

int globalvariable = 10;

int main(void)
{
    add_one();
    printf("Η τιμή της καθολικής μεταβλητής είναι %d\n", globalvariable);

    subtract_one();
    printf("Η τιμή της καθολικής μεταβλητής είναι %d\n", globalvariable);

    return 0;
}

void add_one()
{
    globalvariable++;

}

void subtract_one()
{
    globalvariable--;

}
```

```
Η τιμή της καθολικής μεταβλητής είναι 11
Η τιμή της καθολικής μεταβλητής είναι 10
```

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

void function1();

int a = 10;

int main(void)
{
    printf("Η τιμή της a είναι %d\n", a);

    function1();

    printf("Η τιμή της a είναι %d\n", a);

    return 0;
}

void function1()
{
    int a = 20;

    printf("Η τιμή της a είναι %d\n", a);
}
```

(εντός της συνάρτησης υπερισχύει ο ορισμός της τοπικής μεταβλητής)

```
Η τιμή της a είναι 10
Η τιμή της a είναι 20
Η τιμή της a είναι 10
```

ΣΥΝΑΡΤΗΣΗ ΜΕ ΠΑΡΑΜΕΤΡΟ ΠΙΝΑΚΑ

Εάν μια συνάρτηση έχει ως παράμετρο έναν πίνακα, τότε στη δήλωσή της δίνουμε το όνομα του πίνακα με κενές αγκύλες.

Παράδειγμα:

```
void test(int arr[]);
```

Όταν καλούμε τη συνάρτηση, τότε ως όρισμα δίνουμε μόνο το όνομα του πίνακα χωρίς αγκύλες.

Παράδειγμα:

```
test(arr);
```

ΣΥΝΑΡΤΗΣΗ ΜΕ ΠΑΡΑΜΕΤΡΟ ΠΙΝΑΚΑ

Όταν μια συνάρτηση έχει ως παράμετρο έναν πίνακα, τότε στη συνάρτηση μεταβιβάζεται μόνο η διεύθυνση του πρώτου στοιχείου του πίνακα (και όχι ολόκληρος ο πίνακας). Δηλαδή, πρόκειται για κλήση μέσω αναφοράς.

Με βάση τη διεύθυνση του πρώτου στοιχείου, η συνάρτηση μπορεί στη συνέχεια να τροποποιήσει τις τιμές όλων των στοιχείων του πίνακα, διότι αυτές καταχωρούνται σε μια συμπαγή περιοχή συνεχόμενων διευθύνσεων μνήμης. Σε αυτή την περίπτωση, χρειάζεται να δώσουμε και το μέγεθος του πίνακα ως ξεχωριστό όρισμα.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

void function1(int pinakas[], int size);

int main(void)
{
    int pinakas[101];

    function1(pinakas, 100);

    return 0;
}
```

(101, διότι είναι από 0 έως 100)

(εδώ αρκεί το χρήσιμο μέγεθος του πίνακα, δηλ. 1 έως 100)

```c
void function1(int pinakas[], int size)
{
    int i;

    for(i = 1; i<=size; i++) pinakas[i] = 2*i;

    printf("Η τιμή του στοιχείου pinakas[3] είναι %d\n", pinakas[3]);
}
```

```
Η τιμή του στοιχείου pinakas[3] είναι 6
```

ΣΥΝΑΡΤΗΣΗ ΜΕ ΠΑΡΑΜΕΤΡΟ ΣΤΑΘΕΡΟ ΠΙΝΑΚΑ

Εάν θέλουμε μια συνάρτηση να μη μπορεί να τροποποιήσει τις τιμές των στοιχείων ενός πίνακα που δέχεται ως όρισμα, τότε θα πρέπει ο πίνακας να δοθεί ως const.

Παράδειγμα:

```
void function(const int arr[]);
```

ΣΥΝΑΡΤΗΣΗ ΜΕ ΠΑΡΑΜΕΤΡΟ ΔΙΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ

Εάν μια συνάρτηση έχει ως παράμετρο έναν διδιάστατο πίνακα, τότε στη δήλωσή της δίνουμε το όνομα του πίνακα μαζί με τις διαστάσεις του σε αγκύλες.

Παράδειγμα:

```
void test(int two_arr[5][10]);
```

Εναλλακτικά, αρκεί η δεύτερη διάσταση, δηλ. αρκεί

```
void test(int two_arr[][10]);
```

Όταν καλούμε τη συνάρτηση, τότε ως όρισμα δίνουμε μόνο το όνομα του πίνακα χωρίς αγκύλες.

Παράδειγμα: `test(two_arr);`

ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

Αρκετές χρήσιμες μαθηματικές συναρτήσεις είναι ορισμένες στη βιβλιοθήκη math.h :

Mathematical Functions <math.h>

Arguments and returned values are double

trig functions	sin(x), cos(x), tan(x)
inverse trig functions	asin(x), acos(x), atan(x)
$\arctan(y/x)$	atan2(y,x)
hyperbolic trig functions	sinh(x), cosh(x), tanh(x)
exponentials & logs	exp(x), log(x), log10(x)
exponentials & logs (2 power)	ldexp(x,n), frexp(x,&e)
division & remainder	modf(x,ip), fmod(x,y)
powers	pow(x,y), sqrt(x)
rounding	ceil(x), floor(x), fabs(x)

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Τέλος ενότητας

Επεξεργασία: Νικόλαος Τρυφωνίδης
Θεσσαλονίκη, 20/09/2015

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΣΗΜΕΙΏΜΑΤΑ

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.