

Π. Σταθοπούλου

psth@ece.upatras.gr ή
psth@upatras.gr

Ομάδα Α' (Φοιτητές με μονό αριθμό Μητρώου)

- Διδασκαλία : Παρασκευή 11πμ-13μμ ΗΛ7
- Φροντιστήριο : Δευτέρα 11πμ-12πμ ΗΛ4

Προηγούμενη Διάλεξη

Προετοιμασία για το 4^ο εργαστήριο

Γιατί κάνουμε το 4^ο εργαστήριο ?

ποιες γνώσεις θα αξιοποιήσουμε

Συναρτήσεις, Πίνακες, Δομές

Σήμερα

Βασικές έννοιες Δομών

- * Απλές δομές & Σύνθετες δομές
- * Πίνακες δομών & Δείκτες σε δομές

Διαχείριση Αρχείων Δίσκου

- * Άνοιγμα αρχείου
- * Κλείσιμο αρχείου
- * Ανάγνωση & εγγραφή Δεδομένων

Βασικές έννοιες Δομών

Απλές δομές

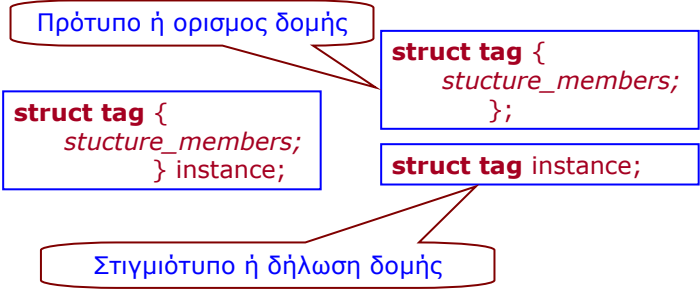
Δομή είναι μια συλλογή μιας ή περισσότερων μεταβλητών ομαδοποιημένων με ένα όνομα με βασικό στόχο τον εύκολο χειρισμό.

- * Οι **μεταβλητές** σε μια δομή, σε αντίθεση με εκείνες σε ένα πίνακα μπορούν να είναι **διαφορετικών τύπων δεδομένων**.
- * Τύποι δεδομένων μιας δομής μπορεί να είναι **πίνακες & άλλες δομές**.
- * Κάθε **μεταβλητή** σε μια δομή ονομάζεται **μέλος** της δομής.

Βασικές έννοιες Δομών

Ορισμός & δήλωση Απλών δομών

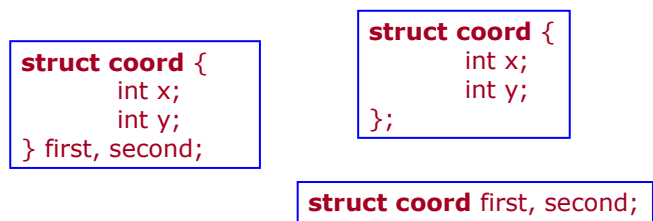
Η **Λέξη-Κλειδί struct** χρησιμοποιείται για να δηλώσει δομές.



Βασικές έννοιες Δομών

Ορισμός & δήλωση Απλών δομών

Παραδείγματα:



Βασικές έννοιες Δομών

Διαχείριση των μελών απλής δομής

Τα **μέλη δομής** μπορούμε να τα διαχειριστούμε όπως τις μεταβλητές.

Ο τρόπος προσπέλασης πραγματοποιείται με τον **τελεστή τελείας (.)** ανάμεσα στο όνομα της δομής και το όνομα του μέλους.

```
first.x=50;
second.y=100;
```

```
Printf("%d,%d", second.x, second.y);
```

```
first=second;
```

```
first.x=second.x;
first.y=second.y;
```

Βασικές έννοιες Δομών

```
#include <stdio.h>
int length, width;
long area;
struct coord{
    int x;
    int y;
} myPoint;
```

Ορισμός & δήλωση Δομής;

The coordinates are: (12, 14)

```
int main( void )
{
    /* set values into the coordinates */
    myPoint.x = 12;
    myPoint.y = 14;
    printf("\nThe coordinates are: (%d, %d).",
        myPoint.x, myPoint.y);
    return 0;
}
```

Ανάθεση τιμών

Βασικές έννοιες Δομών

Δήλωση πρότυπου δομής με όνομα SSN

```
struct SSN {  
    int first_three;  
    char dash1;  
    int second_two;  
    char dash2;  
    int last_four;  
};
```

Δήλωση στιγμιότυπου τύπου SSN

```
struct SSN customer_ssn;
```

Βασικές έννοιες Δομών

Δήλωση δομής & στιγμιότυπου

```
struct date {  
    char month[2];  
    char day[2];  
    char year[4];  
} current_date;
```

Δήλωση δομής & στιγμιότυπου με αρχική τιμή

```
struct time {  
    int hours;  
    int minutes;  
    int second;  
} time_of_birth={8, 45,0};
```

Βασικές έννοιες Δομών

Σύνθετες δομές

- * Δομές σε άλλες δομές
- * Δομές που περιέχουν πίνακες
- * Πίνακες δομών

Δομές σε άλλες δομές

Από τον ορισμό των απλών δομών γνωρίζουμε ότι τα μέλη μιας δομής μπορούν να είναι μεταβλητές οποιουδήποτε τύπου ακόμη και **δομές** ή **πίνακες**.

Βασικές έννοιες Δομών

Δομές σε άλλες δομές

Παραδείγματα:

```
struct coord {  
    int x;  
    int y;  
} topleft, bottomrigh;
```

```
struct rectangle {  
    struct cord topleft;  
    struct cord bottomrigh;  
};
```

```
struct rectangle mybox;
```

```
struct rectangle {  
    struct cord topleft;  
    struct cord bottomrigh;  
}mybox;
```

Βασικές έννοιες Δομών

Διαχείριση μελών σύνθετων Δομών

Η προσπέλαση στα πραγματικά δεδομένα των μελών (μέλη τύπου `int`) των σύνθετων δομών πραγματοποιείται με την εφαρμογή του *τελεστή της τελείας* (`.`) δύο φορές.

Έτσι η έκφραση

```
mybox.topleft.x
```

αναφέρεται στο μέλος `x` του μέλους `topleft` της δομής τύπου `rectangle` με όνομα `mybox`

Βασικές έννοιες Δομών

Διαχείριση μελών σύνθετων Δομών

Για να ορίσουμε ένα ορθογώνιο με συντεταγμένες $(0,10),(100,200)$ πρέπει να γράψουμε:

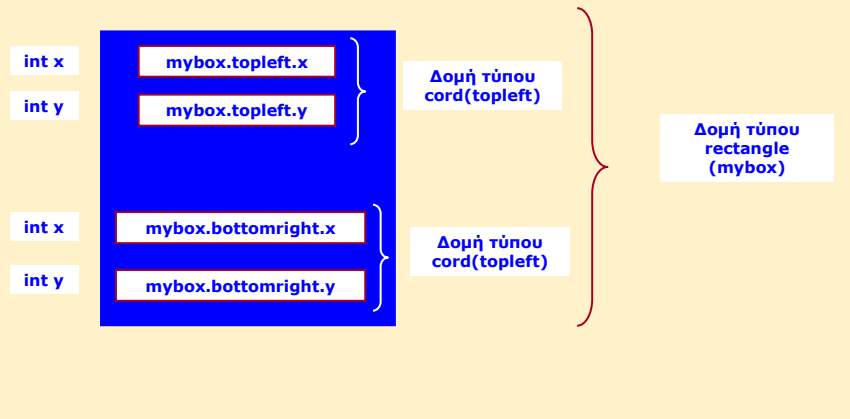
```
mybox.topleft.x=0;
mybox.topleft.y=10;
mybox.bottomright.x=100;
mybox.bottomright.y=200;
```

```
struct coord {
    int x;
    int y;
} topleft, bottomright;
```

```
struct rectangle {
    struct coord topleft;
    struct coord bottomright;
} mybox;
```

Βασικές έννοιες Δομών

Απεικόνιση σύνθετων Δομών



Βασικές έννοιες Δομών

Πρόγραμμα – Παράδειγμα:

```
#include <stdio.h>
int length, width;
long area;

struct coord{
    int x;
    int y;
} topleft, bottomrt;

struct rectangle{
    struct coord topleft;
    struct coord bottomrt;
} mybox;
```

Ορισμός & δήλωση απλής δομής

Στιγμιότυπα
`topleft, bottomrt, mybox`

Ορισμός & δήλωση σύνθετης δομής

Βασικές έννοιες Δομών

```
int main( void )
{
    /* Input the coordinates */
    printf("\nEnter the top left x coordinate: ");
    scanf("%d", &mybox.topleft.x);

    printf("\nEnter the top left y coordinate: ");
    scanf("%d", &mybox.topleft.y);

    printf("\nEnter the bottom right x coordinate: ");
    scanf("%d", &mybox.bottomrt.x);

    printf("\nEnter the bottom right y coordinate: ");
    scanf("%d", &mybox.bottomrt.y);
}
```

Ανάθεση τιμών από
το πληκτρολόγιο

Βασικές έννοιες Δομών

```
/* Calculate the length and width */
width = mybox.bottomrt.x - mybox.topleft.x;
length = mybox.bottomrt.y - mybox.topleft.y;

/* Calculate and display the area */
area = width * length;
printf("\nThe area is %ld units.\n", area);

return 0;
}
```

Μαθηματικές πράξεις σε
δομές

Βασικές έννοιες Δομών

Δομές που περιέχουν πίνακες

Παραδείγματα:

```
struct data {  
    int x[4];  
    char y[10];  
};  
  
struct data record;
```

```
struct data {  
    int x[4];  
    char y[10];  
} record;
```

Βασικές έννοιες Δομών

Διαχείριση δομών με πίνακες

```
record.x[2]=100;
```

```
record.y[5]='f';
```

Η έκφραση `record.y` είναι ένας δείκτης

η πρόταση `puts(record.y)` προβάλλει τα περιεχόμενα του πίνακα `y[]`

Βασικές έννοιες Δομών

Πρόγραμμα – Παράδειγμα:

```
/* Demonstrates a structure that has array members. */  
  
#include <stdio.h>  
  
/* Define and declare a structure to hold the data. */  
/* It contains one float variable and two char arrays. */  
  
struct data{  
    float amount;  
    char fname[30];  
    char lname[30];  
} rec;
```

Ορισμός & δήλωση
δομής πινάκων

στιγμιότυπο

Βασικές έννοιες Δομών

```
int main(int argc, char *argv[])  
{  
    /* Input the data from the keyboard. */  
  
    printf("Enter the donor's first and last names,\n");  
  
    printf("separated by a space: ");  
    scanf("%s %s", rec.fname, rec.lname);  
  
    printf("\nEnter the donation amount: ");  
    scanf("%f", &rec.amount);  
}
```

Εκχώρηση τιμών
μεταβλητών δομών

Βασικές έννοιες Δομών

```
/* Display the information. */  
/* Note: %.2f specifies a floating-point value */  
/* to be displayed with two digits to the right of the decimal point. */  
  
/* Display the data on the screen. */  
  
printf("\nDonor %s %s gave $%.2f.\n", rec.fname, rec.lname,  
      rec.amount);  
  
system("PAUSE");  
return 0;  
}
```

Εκτύπωση τιμών
μεταβλητών δομών

Βασικές έννοιες Δομών

Πίνακες Δομών

```
struct entry {  
    char fname[10];  
    char lname[12];  
    char phone[8];  
};  
  
struct entry list[1000];
```

Πίνακας 1000 στοιχείων
Κάθε στοιχείο του πίνακα
μία δομή τύπου **entry**

Βασικές έννοιες Δομών

Διαχείριση Πινάκων Δομών

list[1]=list[5]; Εκχώρηση των δεδομένων ενός στοιχείου σε ένα άλλο στοιχείο.

strcpy(list[1].phone, list[5].phone);
αντιγραφή μιας ακολουθίας χαρακτήρων

list[5].phone[1]=list[2].phone[3];
μεταφορά χαρακτήρα 4^ο χαρακτήρα του 3^{ου} στοιχείου, σε 2^ο χαρακτήρα του 6^{ου} στοιχείου

Βασικές έννοιες Δομών

Πρόγραμμα – Παράδειγμα:

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

/* Demonstrates using arrays of structures. */
/* Define a structure to hold entries. */

struct entry {
    char fname[20];
    char lname[20];
    char phone[10];
};

struct entry list[4]; /* Declare an array of structures. */
int i;
```

Ορισμός δομής
πινάκων

Δήλωση πίνακα
δομής

Βασικές έννοιες Δομών

```
int main(int argc, char *argv[])
{
    /* Loop to input data for four people. */
    for (i = 0; i < 4; i++)
    {
        printf("\nEnter first name: ");
        scanf("%s", list[i].fname);
        printf("Enter last name: ");
        scanf("%s", list[i].lname);
        printf("Enter phone in 123-4567 format: ");
        scanf("%s", list[i].phone);
    }
}
```

Εκχώρηση τιμών
σε πίνακα δομής

Βασικές έννοιες Δομών

```
/* Loop to display data. */
for (i = 0; i < 4; i++) {
    printf("Name: %s %s", list[i].fname, list[i].lname);
    printf("\t\tPhone: %s\n", list[i].phone);
}

system("PAUSE");
return 0;
}
```

Εκτύπωση των τιμών
ενός πίνακα δομής

Βασικές έννοιες Δομών

Απόδοση αρχικών τιμών σε δομές

Παράδειγμα 1°

```
struct sale {
    char customer[20];
    char item[20];
    float amount;
}mysale={
    "acme Industries",
    "Left-handed widget",
    1000,00
};
```

mysale.customer
mysale.item
mysale.amount

Βασικές έννοιες Δομών

Παράδειγμα 2°

```
struct customer{
    char fir[20];
    char contact[25];
};

struct sale {
    struct customer buyer;
    char item[20];
    float amount;
}mysale={{ "Acme Industries", "George Adams"},
    "Left-handed widget",
    1000,00
};
```

Βασικές έννοιες Δομών

Παράδειγμα 3^ο

```
struct customer{  
    char fir[20];  
    char contact[25];  
} buyer;
```

```
struct sale {  
    struct customer buyer;  
    char item[20];  
    float amount;  
};
```

```
struct sale list[100] {  
    {{"Acme Industries", "George Adams"},  
     "Left-handed widget",  
     1000,00  
    }  
    {{"Wilson & Co.", "Ed Wilson"},  
     "type 12 gizmo",  
     290.00  
    }  
};
```

Απόδοση αρχικών τιμών
σε σύνθετη δομή

Βασικές έννοιες Δομών

Δείκτες σε δομές

```
struct data{  
    int *value;  
    int *rate;  
} first;
```

```
first.value = &cost;  
first.rate = &interest;
```

Διεύθυνση μνήμης

```
*first.value = τιμή της cost;  
*first.rate = τιμή της interest;
```


Βασικές έννοιες Δομών

Δείκτες σε δομές

δείκτης

```
char *p_message;  
p_message = "Teach yourself C in 21 Days";
```

διεύθυνση

```
struct msg{  
    char *p1;  
    char *p2;  
} myptrs;
```

διευθύνσεις

```
myptrs.p1 = "Teach yourself C in 21 Days";  
myptrs.p2 = "By SAMS Publishing";
```

Βασικές έννοιες Δομών

Διαχείριση Δείκτες σε δομές

```
print ("%s, %s", myptrs.p1, myptrs.p2);
```

? Ποια η διαφορά ανάμεσα (ως μέλη δομής) σε πίνακα τύπου char & δείκτη τύπου char

```
struct msg{  
    char p1[30];  
    char *p2;  
} myptrs;
```

Προσοχή!!!!
p1 χωρίς αγκύλες = δείκτης
άρα p1 & p2
χρησιμοποιούνται με τον ίδιο τρόπο

Βασικές έννοιες Δομών

Διαχείριση Δείκτες σε δομές

Προσοχή!!!!!! Πρέπει να έχει αρχική τιμή ο δείκτης

```
strcpy(myptrs.p1, "Teach yourself C in 21 Days");  
strcpy(myptrs.p2, "By SAMS Publishing");
```

```
struct msg{  
    char p1[10];  
    char p2[10];  
} myptrs;  
.....  
strcpy(myptrs.p1, "mineapolis");  
strcpy(myptrs.p2, "MN");
```

Προσοχή!!!!!!
μεγαλύτερη του πίνακα
σπατάλη χώρου

Βασικές έννοιες Δομών

Δημιουργία δεικτών προς δομές

```
struct part{  
    short number;  
    char name[10];  
};
```

struct part *p_part δείκτης τύπου δομής part
 έχει τιμή ο δείκτης? 'ΟΧΙ

γιατί ? δεν έχουν δηλωθεί τα στιγμιότυπα

Εάν δηλώσω struct part gizmo;
Τότε p_part=&gizmo

Βασικές έννοιες Δομών

Διαχείριση δεικτών προς δομές

Εκχώρηση τιμής

```
Εάν δηλώσω   struct part gizmo;  
Τότε         p_part=&gizmo  
  
gizmo.number=100;      (*p_part).number=100;  
  
Αφού        *p_part= gizmo
```

Βασικές έννοιες Δομών

Εργασία με δεικτές & πίνακες δομών

```
struct part{  
    short number;  
    char name[10];  
};  
  
struct part data[100];  
struct part *p_part;  
p_part=&data[0];      p_part=&data;  
                    πίνακα δομών part    &    δείκτη p_part  
  
Προβολή περιεχομένων  
  
printf("%d %s",(*p_part).number, (*p_part).name);
```

Βασικές έννοιες Δομών

Δομές σε ορίσματα συναρτήσεων

```
/* Demonstrates passing a structure to a function. */  
  
#include <stdio.h>  
/* Declare and define a structure to hold the data. */  
struct data {  
    float amount;  
    char fname[30];  
    char lname[30];  
} rec;  
  
/* The function prototype. The function has no return value, */  
/* and it takes a structure of type data as its one argument. */  
void print_rec(struct data displayRec);
```

Όρισμα δομή

Δήλωση συνάρτησης

Βασικές έννοιες Δομών

Δομές σε ορίσματα συναρτήσεων

```
int main( void )  
{  
    /* Input the data from the keyboard. */  
    printf("Enter the donor's first and last names,\n");  
    printf("separated by a space: ");  
    scanf("%s %s", rec.fname, rec.lname);  
    printf("\nEnter the donation amount: ");  
    scanf("%f", &rec.amount);  
    /* Call the display function. */  
    print_rec( rec );  
    return 0;  
}  
  
void print_rec(struct data displayRec)  
{  
    printf("\nDonor %s %s gave $%.2f.\n", displayRec.fname,  
        displayRec.lname, displayRec.amount);  
}
```

Κλήση συνάρτησης

σώμα συνάρτησης

Διαχείριση Αρχείων Δίσκου

Διαχείριση Αρχείων Δίσκου

- * Άνοιγμα αρχείου
- * Κλείσιμο αρχείου
- * Ανάγνωση & εγγραφή Δεδομένων

Άνοιγμα αρχείου Δίσκου

Συνάρτηση **fopen()** της τυπικής βιβλιοθήκης **stdio.h**
`FILE *fopen(const char *filename, const char *mode);`

Διαχείριση Αρχείων Δίσκου

Άνοιγμα αρχείου Δίσκου

Συνάρτηση **fopen()** της τυπικής βιβλιοθήκης **stdio.h**
`FILE *fopen(const char *filename, const char *mode);`

* Επιστρεφόμενη τιμή δείκτης τύπου **FILE**
Δομή δηλωμένη στην stdio.h

Άρα όταν πρόκειται να ανοίξουμε ένα αρχείο πρέπει να δηλώνεται ένας δείκτης.

Όταν καλείται η **fopen()** δημιουργείται ένα στιγμιότυπο δομής **FILE** και επιστρέφει ένα δείκτη στην δομή αυτή

Διαχείριση Αρχείων Δίσκου

Ο δείκτης αυτός χρησιμοποιείται σε όλες τις επόμενες λειτουργίες του αρχείου.

Εάν η συνάρτηση **fopen()** αποτύχει επιστρέφει **NULL**.

Ορίσματα:

filename όνομα αρχείο μαζί με το path

mode καθορίζει τον τρόπο ανοίγματος του αρχείου

mode={r, w, a, r+, w+, a+}

Διαχείριση Αρχείων Δίσκου

Ανάγνωση & εγγραφή Δεδομένων αρχείων δίσκου

mode={r, w, a, r+, w+, a+}

r ανάγνωση

w εγγραφή

a προσάρτηση

r+ ανάγνωση & εγγραφή

w+ ανάγνωση & εγγραφή

a+ ανάγνωση & προσάρτηση

Εάν δεν υπάρχουν δημιουργούνται

Υπερεγγραφή

Διαχείριση Αρχείων Δίσκου

Ανάγνωση & εγγραφή Δεδομένων αρχείων δίσκου

- * **Μορφοποιημένη έξοδο (αρχεία κειμένου)**
- * **Έξοδο χαρακτήρα (κυρίως αρχεία κειμένου)**
- * **Άμεση έξοδο (μνήμης σε αρχείο)**

Διαχείριση Αρχείων Δίσκου

Συναρτήσεις διαχείρισης των περιεχομένων των αρχείων δίσκου

- * **Μορφοποιημένη είσοδο/έξοδο**
fprintf() **fscanf()**
- * **είσοδο/έξοδο χαρακτήρα**
gets() **fgetc()** **fgets()**
fputc() **puts()** **fputs()**
- * **Είσοδο/έξοδο δυαδικών αρχείων**
fread() **fwrite()**

Διαχείριση Αρχείων Δίσκου

Κλείσιμο αρχείων

```
int fclose(FILE *fp);
```

το όρισμα fp είναι ο δείκτης FILE

fclose() επιστρέφει 0 σε επιτυχία και -1 σε σφάλμα