

# ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ

Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: [belk@cs.ucy.ac.cy](mailto:belk@cs.ucy.ac.cy)





# Τύποι Δεδομένων και Εγγραφές

# Εγγραφές – Struct(ures)

2

- typedef struct επιτρέπει τον **ορισμό σύνθετων τύπων** δεδομένων

```
typedef struct {  
    int    id;  
    float  vathmoi[3];  
} student_t;
```

← structure members  
μέλη εγγραφής

# Σύνταξη: 1<sup>ος</sup> τρόπος

3

```
typedef struct {  
    μέλος;  
    μέλος;  
    ...  
} <όνομα νέου τύπου> ;
```

μέλος: δήλωση οποιασδήποτε μεταβλητής  
(μπορεί να είναι struct)

# Σημασία

4

- Το **typedef struct** επιτρέπει τον **ορισμό ενός καινούργιου σύνθετου τύπου**
- Το typedef struct **δεν** είναι μεταβλητή

# Σύνταξη: 2<sup>ος</sup> τρόπος

5

```
struct <όνομα νέου τύπου>{  
    μέλος;  
    μέλος;  
    ...  
} <μια ή περισσότερες structure μεταβλητές> ;
```

μέλος: δήλωση οποιαδήποτε μεταβλητής  
(μπορεί να είναι struct)

# struct: Ορισμός Μεταβλητής

6

```
typedef struct {  
    int    id;  
    float  vathmoi[3];  
} student_t;
```

**student\_t student;** // δήλωση μεταβλητής student

**Σημασία:** δεσμεύει χώρο για **ΟΛΑ** τα μέλη της δομής  
ασχέτως εάν ο χρήστης τα χρησιμοποιήσει.

*Στο παράδειγμα πόσα bytes δεσμεύονται για το student;*

# struct: Ορισμός Μεταβλητής και Πίνακα

7

Ορίζω δύο μεταβλητές τύπου `student_t`:

```
student_t student_a, student_b;
```

Ορίζω έναν πίνακα τύπου `student_t`.

□ Στην κάθε θέση του πίνακα υπάρχει μια δομή!

```
student_t student_table[STUDENT_NUM];
```

*Εάν το `STUDENT_NUM` είναι 100, πόσα bytes δεσμεύονται για το `student_table`;*



# struct: Αναφορά σε Μέλη (1)

8

```
typedef struct {  
    int    id;  
    float  vathmoi[3];  
} student_t;
```

```
student_t student;
```

```
student.id = 123456;
```

```
student.vathmoi[0] = 8;
```

```
student.vathmoi[1] = 5;
```

```
student.vathmoi[2] = 9.5;
```

• τελεστής επιλογής (μέλους)

*Τελεστής με την πιο ψηλή προτεραιότητα...  
Επίσης [] και f(...)*

## struct: Αναφορά σε Μέλη (2)

9

```
printf(“%6d %5.2f %5.2f %5.2f\n”, student.id,  
student.vathmoi[0], student.vathmoi[1], student.vathmoi[2]);
```

```
scanf(“%d%f%f%f”, &student.id, &student.vathmoi[0],  
&student.vathmoi[1], &student.vathmoi[2]);
```

# Αναφορά σε Πίνακα με structs (1)

10

```
#include <stdio.h>
#define STUDENT_NUM 3

int main ()
{
    int i;

        typedef struct {
            int    id;
            float  vathmoi[3];
        } student_t;

    student_t student_table[STUDENT_NUM];
```

# Αναφορά σε Πίνακα με structs (2)

11

```
for (i=0; i<STUDENT_NUM; i++){
    printf("Parakalo dose ta stoixeia tou foititi %d:", i+1);
    scanf("%d%f%f%f", &student_table[i].id, &student_table[i].vathmoi[0],
&student_table[i].vathmoi[1], &student_table[i].vathmoi[2]);
}

printf("Ta stoixeia ton foititon einai:\n");
    for (i=0; i<STUDENT_NUM; i++){
        printf("%6d %5.2f %5.2f %5.2f\n", student_table[i].id,
student_table[i].vathmoi[0], student_table[i].vathmoi[1], student_table[i].vathmoi[2]);
    }

    return 0;
}
```

```
Parakalo dose ta stoixeia tou foititi 1:1 9.5 7.4 3.5
Parakalo dose ta stoixeia tou foititi 2:2 7.5 6.0 5.5
Parakalo dose ta stoixeia tou foititi 3:3 5.5 8.5 8.0
Ta stoixeia ton foititon einai:
    1  9.50  7.40  3.50
    2  7.50  6.00  5.50
    3  5.50  8.50  8.00
```

# struct: Σαν Παράμετρος

12

```
void display_student(student_t s){  
    printf(“%6d %5.2f %5.2f %5.2f\n”,  
        s.id, s.vathmoi[0],  
        s.vathmoi[1], s.vathmoi[2]);  
}
```

*Πόσος χώρος δεσμεύεται για το s;*

```
for(i=0;i<STUDENT_NUM;++i)  
    display_student(student_table[i]);
```

# struct: Ανάθεση

13

- Επιτρέπεται ανάθεση μεταξύ μεταβλητών τύπου structs
  - ▣ `student_a = student_b;`
  - ▣ `student_table[i] = student_table[j];`
  - ▣ αντιγράφει την τιμή κάθε μέλους
- Πχ.
  - ▣ `student_a = {1234, 5.0, 3.5, 6.0}; /* συντακτικά ορθό*/`
  - ▣ `student_b = {7654, 5.5, 7.5, 9.0};`
  - ▣ `student_a = student_b;`

*Τι περιέχουν τα δύο structs;*
- **Δεν επιτρέπεται σύγκριση struct (αν χρειάζεται, τότε πρέπει να γίνει ανά μέλος)**

# Ανακεφαλαίωση

14

- **typedef struct** επιτρέπει τον ορισμό σύνθετων τύπων δεδομένων
- Δύο βήματα για ορισμό μεταβλητής

## (A) Ορίζουμε τον νέο τύπο

```
typedef struct {  
    int    id;  
    float  vathmoi[3];  
} student_t;
```

## (B) Μπορούμε να δηλώσουμε μεταβλητές του τύπου

```
student_t student;
```

**Σημασία:** δεσμεύει χώρο για ΟΛΑ τα μέλη της δομής

# Παράδειγμα (1)

15

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

typedef struct {
    char title[50];
    char author[50];
    char subject[100];
    int book_id;
} Books;

int main( )
{
    int i;
    Books Book1={"C Programming", "Nuha Ali", "C Programming Tutorial",
6495407};          /* Declare Book1 of type Book */
    Books Book2;          /* Declare Book2 of type Book */
```

Συνέχεια στην επόμενη διαφάνεια...



# Παράδειγμα (2)

16

```
/* book 2 specification */  
strcpy( Book2.title, "Telecom Billing");  
strcpy( Book2.author, "Zara Ali");  
strcpy( Book2.subject, "Telecom Billing Tutorial");  
Book2.book_id = 6495700;
```

```
/* print Book1 info */  
printf( "Book 1 title : %s\n", Book1.title);  
printf( "Book 1 author : %s\n", Book1.author);  
printf( "Book 1 subject : %s\n", Book1.subject);  
printf( "Book 1 book_id : %d\n", Book1.book_id);  
printf("\n");
```

Συνέχεια στην επόμενη διαφάνεια...

# Παράδειγμα (3)

17

```
/* print Book2 info */
printf( "Book 2 title : %s\n", Book2.title);
printf( "Book 2 author : %s\n", Book2.author);
printf( "Book 2 subject : %s\n", Book2.subject);
printf( "Book 2 book_id : %d\n", Book2.book_id);
printf("\n");

return 0;
}
```

```
Book 1 title : C Programming
Book 1 author : Nuha Ali
Book 1 subject : C Programming Tutorial
Book 1 book_id : 6495407

Book 2 title : Telecom Billing
Book 2 author : Zara Ali
Book 2 subject : Telecom Billing Tutorial
Book 2 book_id : 6495700
```

# Παραδείγματα structs για αυτοεξέταση

18

- Σελίδα 597 του βιβλίου
  - ▣ Ερωτήματα 1 – 8

1. What is the primary difference between a structure and an array? Which would you use to store the catalog description of a course? To store the names of students in the course?
2. How do you access a component of a structure type variable?

Exercises 3–8 refer to the following type `student_t` and to variables `stu1` and `stu2`.

```
typedef struct {
    char fst_name[20],
        last_name[20];
    int  score;
    char grade;
} student_t;
. . .
student_t stu1, stu2;
```

3. Identify the following statements as possibly valid or definitely invalid. If invalid, explain why.

- a. `student_t stulist[30];`
- b. `printf("%s", stu1);`
- c. `printf("%d %c", stu1.score, stu1.grade);`
- d. `stu2 = stu1;`
- e. `if (stu2.score == stu1.score)`  
    `printf("Equal");`
- f. `if (stu2 == stu1)`  
    `printf("Equal structures");`

# Παραδείγματα structs για αυτοεξέταση

20

g. `scan_student(&stu1);`

h. `stu2.last_name = "Martin";`

4. Write a statement that displays the initials of `stu1` (with periods).
5. How many components does variable `stu2` have?
6. Write functions `scan_student` and `print_student` for type `student_t` variables.
7. Declare an array of 40 `student_t` structures, and write a code segment that displays on separate lines the names (*last name, first name*) of all the students in the list.
8. Identify the type of each of the following references:
  - a. `stu1`
  - b. `stu2.score`
  - c. `stu2.fst_name[3]`
  - d. `stu1.grade`

# Λύση παραδείγματος 6 (1)

21

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

```
typedef struct {
```

```
    int    id;
```

```
    float  vathmoi[3];
```

```
} student_t;
```

```
student_t scan_student () {
```

```
    student_t s;
```

```
    printf("Parakalo dose ta stoixeia tou foititi:\n");
```

```
    scanf("%d%f%f%f", &s.id, &s.vathmoi[0], &s.vathmoi[1], &s.vathmoi[2]);
```

```
    return s;
```

```
}
```

# Λύση παραδείγματος 6 (2)

22

```
void print_student(student_t s){
    printf("Ta stoixeia tou foititi einai:\n");
    printf("%6d %5.2f %5.2f %5.2f\n", s.id, s.vathmoi[0], s.vathmoi[1], s.vathmoi[2]);
}

int main ()
{
    student_t stu2;
    stu2= scan_student();
    print_student(stu2);

    return 0;
}
```

# Άσκηση

23

Σας δίνεται η πιο κάτω συνάρτηση

```
float apostasi (float xa, float ya, float xb, float yb)
{
    return (xa-xb)*(ya-yb);
}
```

Ξαναγράψετε την συνάρτηση συμπληρώνοντας τα πιο κάτω αλλά αυτή την φορά μόνο με δύο παραμέτρους. Οι δύο παράμετροι πρέπει να είναι του ιδίου τύπου και να είναι εγγραφές (structs).

```
typedef struct {
    } point_t;

float apostasi (
    ,
)
{
}
}
```



# Παράδειγμα

24

Υλοποιήστε ένα πρόγραμμα που ορίζει μια δομή δεδομένων (structure) η οποία περιέχει τον αριθμό λογαριασμού, το όνομα και το επίθετο του ιδιοκτήτη του λογαριασμού, καθώς και το διαθέσιμο υπόλοιπο. Το πρόγραμμα θα διαβάζει από το πληκτρολόγιο 10 λογαριασμούς. Στη συνέχεια θα εκτυπώνει στην οθόνη:

- a. το συνολικό διαθέσιμο υπόλοιπο
- b. τον μέσο όρο των διαθέσιμων υπολοίπων όλων των λογαριασμών
- c. όλα τα στοιχεία των λογαριασμών, οι οποίοι έχουν διαθέσιμο υπόλοιπο πέραν των €500.

Σημείωση: Για την εκτύπωση του εκάστοτε λογαριασμού να οριστεί και να χρησιμοποιηθεί η συνάρτηση `print_account()`, η οποία δέχεται σαν παράμετρο μια δομή δεδομένων.

# Λύση παραδείγματος (1)

25

```
#include <stdio.h>
#define SIZE 10

typedef struct {
    int account_no;
    char name[15];
    char surname[15];
    float balance;
} accounts_s;

void print_account(accounts_s s){
    printf("Ta stoixeia pelati einai:\n");
    printf("%6d %s %s %5.2f\n", s.account_no, s.name, s.surname, s.balance);
}
```

# Λύση παραδείγματος (2)

26

```
int main(){

    accounts_s account_array[SIZE];

    int i;
    float sum=0.0;

    for (i=0; i<SIZE; i++){
        printf("Dose ta stoixeia tou logariasmou %d: ", i+1);
        scanf("%d %s %s %f", &account_array[i].account_no, &account_array[i].name,
&account_array[i].surname, &account_array[i].balance);
    }

    // Evresi athrsoismatos
    for (i=0; i<SIZE; i++){
        sum += account_array[i].balance;
    }
}
```

# Λύση παραδείγματος (3)

27

```
printf("To synoliko diathesimo ypoloipo einai %.2f kai o mesos oros einai %.2f\n", sum,  
sum/SIZE);
```

```
//Ektyposi stoixeion logariasmon me ypoloipo >500 Euro
```

```
for (i=0; i<SIZE; i++){
```

```
    if (account_array[i].balance > 500.0)
```

```
        print_account(account_array[i]);
```

```
}
```

```
return 0;
```

```
}
```

# Χρήση struct

28

- Πίνακες με struct αντί παράλληλοι πίνακες
  
- Πιο αφαιρετική προσέγγιση – ομαδοποίηση συγγενικών μεταβλητών
  - ▣ Πχ. Πληροφορίες μιας ταινίας DVD
    - Κωδικός DVD
    - Τίτλος ταινίας
    - Κατηγορία ταινίας
    - Είδος ταινίας
    - Διάρκεια ταινίας
    - Ονοματεπώνυμα πρωταγωνιστών
    - ...

# Περίληψη

29

- Εγγραφές (Structs)
  - ▣ Σύνταξη και σημασιολογία
  - ▣ Ορισμός μεταβλητών σύνθετων τύπων δεδομένων
  - ▣ Αναφορά σε μέλη εγγραφών
  - ▣ Εγγραφές και συναρτήσεις
  - ▣ Εγγραφές και πίνακες
  - ▣ Χρήση εγγραφών