Εισαγωγή στον δομημένο προγραμματισμό

Ενότητα 10η: Δομές

Αν. καθηγητής Στεργίου Κώστας
e-mail: kstergiou@uowm.gr

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

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 Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη 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.

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

Στόχοι της διάλεξης

- Ανάλυση της έννοιας της δομής ως μεθόδου ορισμού νέων τύπων από τον προγραμματιστή.
- Ανάλυση των βασικών χαρακτηριστικών των δομών.

Περιεχόμενα

- Συναθροιστικοί τύποι δεδομένων.
- Δομές.
  - Ορισμός και αρχικοποίηση.
  - Αναφορά στα μέλη.
  - Συναρτήσεις και δομές.
  - Πίνακες δομών.
  - Δείκτες και δομές.
- Σταθερές απαρίθμησης και η εντολή typedef.

Ομαδοποίηση και Αφαιρετικότητα

- Πολύ συχνά έχουμε πληροφορίες διαφορετικού τύπου οι οποίες όμως είναι άμεσα σχετιζόμενες.
  - Π.χ. θέλουμε να κρατήσουμε πληροφορίες για ένα σύνολο φοιτητών όπου ο κάθε φοιτητής έχει:
    - *Όνομα.*
    - *Αριθμό μητρώου.*
    - *Ηλικία .*

```
struct student {
int AM;
char name[40];
int age;
};
```

- Η C δίνει την δυνατότητα ομαδοποίησης σχετιζόμενων δεδομένων μέσω της **δομής** (structure -> **struct**).
  - Με τις συναρτήσεις επιτυγχάνουμε αφαιρετικότητα στις διεργασίες.
  - Η αφαιρετικότητα στα δεδομένα επιτυγχάνεται με τις δομές **.**

Συναθροιστικοί Τύποι Δεδομένων

- Οι τύποι δεδομένων που έχουμε χρησιμοποιήσει καλύπτουν απλές περιπτώσεις.
  - μετρητής: ακέραιος.
  - θερμοκρασία: πραγματικός, απλής ακρίβειας.
- Για την αναπαράσταση πιο περίπλοκων δεδομένων χρησιμοποιούμε συναθροιστικούς τύπους δεδομένων.
  - Ταχυδρομική διεύθυνση: αλφαριθμητικά και ακέραιοι:
    - Μεγ. Αλεξάνδρου 10, Κοζάνη, 50100.
  - Συντεταγμένες σημείου: 2 πραγματικοί αριθμοί:
    - (3.2, 4.5)

Δομές

- Μας επιτρέπουν τον χειρισμό ομάδων δεδομένων που έχουν κάποια σχέση μεταξύ τους.
- Σε αντίθεση με τους πίνακες, τα δεδομένα των δομών μπορεί να έχουν διαφορετικό τύπο.
- Σε αντίθεση με τους πίνακες, κάθε δεδομένο μέλος της δομής μπορεί να έχει το δικό του όνομα.
- Τα μέλη μιας δομής μπορεί να ανήκουν στους βασικούς τύπους (int, char, float, double), να είναι πίνακες ή ακόμη και άλλες δομές.

```
struct student {
int AM;

char name[40];

int age;
};
```

Ορισμός Δομής

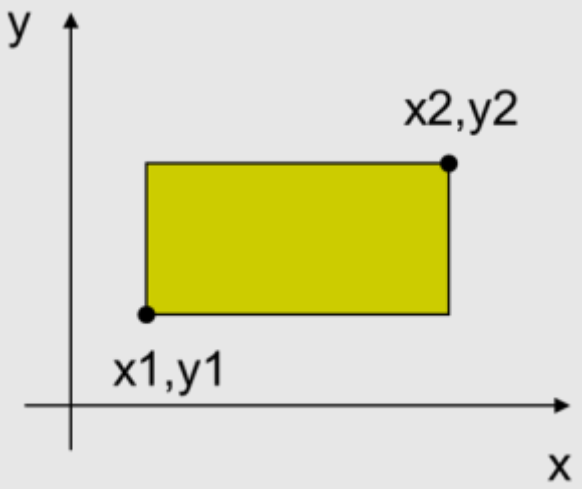
```
struct <όνομα δομής> {
    <τύπος 1ου μέλους> <όνομα 1ου μέλους>;
    <τύπος 2ου μέλους> <όνομα 2ου μέλους>;
    ...
    <τύπος Νου μέλους> <όνομα Νου μέλους>;
};
```

- Εάν δύο ή περισσότερα μέλη έχουν τον ίδιο τύπο η αναφορά μπορεί να γίνει πιο απλοποιημένα:

  <τύπος μέλους> <όνομα 1ου>, <όνομα 2ου>;

Παράδειγμα: Δομή Ορθογωνίου

```
struct rectangle {
float x1, y1;     /* Συντ. γωνίας 1 */
float x2, y2;     /* Συντ. γωνίας 2 */
int line_color;
int fill_color;
};
```



Δήλωση Μεταβλητών

struct <όνομα δομής> <όνομα μεταβλητής>;

```
struct rectangle r1;
```

struct <όνομα δομής> <λίστα μεταβλητών>;

```
struct rectangle r1, r2, r3;
```

Δήλωση δομής και λίστας μεταβλητών;

```
struct rectangle {
…
} r1, r2, r3;
```

Απόδοση Αρχικών Τιμών

```
struct rectangle r1 = {10.1,10.5,40,40,3,5};

struct rectangle r1 = {10.1,10.5,40,40,3,5},
                 r2 = {5.0,5.0,50,50,3,6};
```

Αναφορά στα Μέλη

<όνομα μεταβλητής>.<όνομα μέλους>

```
r1.x1 = 10.0;
/* το μέλος x1 της μεταβλητής r1*/
r2.line_color = 2;
/* το μέλος line_color της μεταβλητής r2 */
```

- Δείκτες δομής.

  <όνομα δείκτη>-><όνομα μέλους>

```
struct rectangle *pr;
pr->x1
/* το μέλος x1 του δείκτη pr */
```

Παράδειγμα (1/2)

- Υποθέτουμε ότι θέλουμε να καταγράψουμε τα στοιχεία των βιβλίων μιας βιβλιοθήκης.

```
/* Δήλωση δομής */
struct book {
char *title;        /* Τίτλος */
char *writer;       /* Συγγραφέας */
 int isbn;          /* Αριθμός ISBN */
 int code;          /* Κωδικός βιβλιοθήκης */
};
/* Δήλωση μεταβλητών */
struct book book1,book2,book3;
/* Αρχικοποίηση μεταβλητής */
struct bookbook1 = {"Προγραμματισμός σε C",
"N. Παπαδόπουλος", 9613450739, 1555};
```

Παράδειγμα (2/2)

```
/* Λάθος Αρχικοποίηση */
struct bookbook1 = {“Ν. Παπαδόπουλος”,
   “Προγραμματισμός σε C”, 9613450739, 1555};

/* ΛΑΘΟΣ τύπος */
struct bookbook1 = {“Προγραμματισμός σε C”,
   9613450739, “Ν. Παπαδόπουλος”, 1555};

/* Εναλλακτικός τρόπος αρχικοποίησης */
book1.title =     “Προγραμματισμός σε C”;
book1.writer = “Ν. Παπαδόπουλος”;
book1.isbn = 9613450739;
book1.code = 1555;
```

Ένθεση Δομών

- Μία δομή μπορεί να περιλαμβάνει μέλη που είναι τύπου άλλης δομής.
- Δεν υπάρχει περιορισμός στο βάθος ένθεσης.

```
struct address { … };
struct day { … };

struct person {
    struct address addr;
    struct day birthday;
};

struct person emp;
emp.addr.city = "Kozani";
```

Παράδειγμα: Ένθετες Δομές

```
struct point {
   float x,y;        /* Συντεταγμένες σημείου */
};
struct circle {
   struct point p;   /* Κέντρο του κύκλου */
   float r;          /* Ακτίνα του κύκλου */
};
struct circle read_circle(struct circle c1)
{
   printf("Δώσε συντεταγμένες κέντρου:");
   scanf("%f %f", &c1.p.x,&c1.p.y);
   scanf("%f", &c1.r);
   return c1;
}
x = read_circle(c);/* Κλήση της συνάρτησης */
```

Δομές και Συναρτήσεις (1/2)

```
struct point { /* Δήλωση δομής σημείου */
  float x;
  float y;
};
/* Δημιουργία σημείου από τις συντεταγμένες του */
struct point makepoint(int x, int y)
{
  struct point temp;
  temp.x=x;
  temp.y=y;
  return temp;
}
```

Απόδοση Τιμής μέσω Συνάρτησης

```
struct rectangle {
   struct point pt1, pt2;
   int line_color, fill_color;
};
struct rectangle screen;
struct point middle;
struct point makepoint(int, int);

screen.pt1 = makepoint(0,0);
screen.pt2 = makepoint(XMAX, YMAX);
middle = makepoint((screen.pt1.x + screen.pt2.x)/2,
                   (screen.pt1.y + screen.pt2.y)/2);
```

Δομές και Συναρτήσεις (2/2)

```
/* Πρόσθεση δύο σημείων */
struct point addpoint(struct point p1, struct point
   p2)
{
   p1.x += p2.x;
   p1.y += p2.y;
   return p1;
}

/* Επιστρέφει 1 αν το p είναι μέσα στο r, αλλιώς 0 */
int ptinrect(struct point p, struct rect r)
{
   return (p.x >= r.pt1.x && p.x <= r.pt2.x &&
           p.y >= r.pt1.y && p.y <= r.pt2.y);
}
```

Πίνακες Δομών

- Η δήλωσή τους ακολουθεί το πρότυπο των άλλων πινάκων.

```
struct point pt[10];
/* Δήλωση πίνακα pt που αποτελείται από 10
   στοιχεία τύπου δομής point */
```

- Αρχικοποίηση.

```
struct point pt[10] = {
                          {2.3, 3.3},
                          {2.1, 2.5},
                          {4.4, 3.4},
                          {3.2, 4,1},…
                        };

p1=pt[0].x;
```

Παράδειγμα

- Εμφάνιση στοιχείων πίνακα δομών.

```
for (i=0;i<10;i++)
    printf("%f %f",pt[i].x,pt[i].y);
```

- Αποθήκευση στοιχείων σε πίνακα δομών.

```
for (i=0;i<10;i++)
    scanf("%f %f", &pt[i].x, &pt[i].y);
```

- Αναζήτηση σε πίνακα δομών.

```
scanf("%f",&timi);
for (i=0;i<10;i++)
   if (pt[i].x == timi)
      printf("Βρέθηκε στη θέση %d\n",i);
```

Παράδειγμα: Στοιχεία Βιβλίων

```
struct book books[50];
int i=0;
while (i<50) {
   printf("Δώσε τίτλο βιβλίου:\n");
   scanf(" %s", books[i].title);
   printf("Δώσε συγγραφέα:\n");
   scanf(" %s", books[i].writer);
   printf("Δώσε αριθμό ISBN:\n ");
   scanf(" %d", &books[i].isbn);
   printf("Δώσε κωδικό ταξινόμησης:\n");
   scanf(" %d", &books[i].code);
   i++;
}

/* ΛΑΘΟΣ: Δεν επιτρέπεται η αρχικοποίηση όλων των
   μελών ενός στοιχείου του πίνακα ταυτόχρονα */
books[0] ={"Προγραμματισμός σε C","Ν. Παπαδόπουλος",
   9613450739, 1555};
```

Δομή με Μέλος Πίνακα

```
/* Επιτρέπει την εισαγωγή μέχρι 5 συγγραφέων */

struct book {
   char *title;
   char *writer[5];
   int isbn;
   int code;
};
```

Δείκτες σε Δομές

```
struct point pt;
/* η μεταβλητή pt είναι τύπου δομής point*/

struct point *ptr;
/* ο δείκτης ptr δείχνει σε αντικείμενα τύπου δομής
   point */

ptr = &pt;
/* ο δείκτης ptr δείχνει στη διεύθυνση της
   μεταβλητής pt */

ptr->x = 3.0;
/* το πεδίο x της δομής που δείχνει ο ptr γίνεται
   3.0 */
```

Παράδειγμα: Δείκτης σε Δομή

```
#include <stdio.h>
#define MAX 4

struct part {
    int number;
    char name[10];
} data[MAX] = {{1, "Smith"}, {2, "Jones"}, {3, "Adams"},
   {4,"Wilson"}};

main()
{  struct part *p_part;
   int count;
   p_part = data;
   for (count = 0; count < MAX; count++)
   {   printf("At address %d: %d %s\n", p_part, p_part->number,
                p_part->name);
       p_part++;   }
 }
```

Εντολή typedef

- Διευκολύνει τον χρήστη να ορίσει δικά του ονόματα για τους τύπους των δεδομένων.

```
typedef int Length;
/* Το Length είναι συνώνυμο του int */
Length len, maxlen;
typedef char * String;
/* Το String είναι συνώνυμο του char * */
String p, a[10];
int strcmp(String, String);
```

Παράδειγμα (4)

```
struct book {
    char *title; /* Τίτλος */
    char *writer;/* Συγγραφέας */
    int isbn;    /* Αριθμός ISBN */
    int code;    /* Κωδικός βιβλιοθήκης */
  };

typedef struct book BOOK;
BOOK book1,book2,book3;
BOOK book1 = {"Προγραμματισμός σε C", Γ.
  Παπαδόπουλος", 9613450739, 1555};
```

Σταθερές Απαρίθμησης

- Για τη δήλωση σταθερών απαρίθμησης στη C χρησιμοποιείται η εντολή `enum` (enumerate = απαριθμώ).

- **`enum`** <ετικέτα τύπου> {λίστα σταθερών απαρίθμησης} [λίστα μεταβλητών];

- Ένα κλασσικό παράδειγμα χρήσης της enum είναι ο ορισμός Boolean μεταβλητών (που παίρνουν λογικές τιμές: αληθής και ψευδής).

  - `enum boolean {FALSE, TRUE};`
  - `enum boolean found, flag;`

Παράδειγμα: Γραμμική Αναζήτηση

```
/* Δεδομένου πίνακα a[N] */

enum boolean {FALSE, TRUE} found;
int i=0, timi;
scanf("%d", &timi);
found = FALSE;
while ((found == FALSE) && (i<N))
   if (a[i] == timi)
        found = TRUE;
   else i++;
if (found == FALSE)
  printf("ΒΡΕΘΗΚΕ ΤΟ %d ΣΤΗ ΘΕΣΗ %d\n", a[i],i);
else
  printf("ΔΕΝ ΒΡΕΘΗΚΕ\n");
```

Σταθερές Απαρίθμησης (1/2)

- Εξ' ορισμού το πρώτο στοιχείο στη λίστα παίρνει την τιμή 0, το δεύτερο 1, και η απαρίθμηση συνεχίζει μέχρι να υπάρξει διαφορετικός ορισμός.
- Στο προηγούμενο παράδειγμα το FALSE αντιστοιχεί με 0 και το TRUE με 1.
- Θα μπορούσε, όμως, εξίσου καλά να δηλωθεί ως εξής:
  - `enum boolean { FALSE='F', TRUE='T'} book_found, code_found;`
- Σε αυτή την περίπτωση το FALSE αντιστοιχεί με 'F' και το TRUE με 'T'.

Σταθερές Απαρίθμησης (2/2)

- Δεν είναι απαραίτητο να δίνονται τιμές απαρίθμησης σε όλες τις σταθερές, ούτε είναι απαραίτητο να δίνονται διαφορετικές μεταξύ τους τιμές.

- `enum books {MATHEMATICS, ALGEBRA=0, PHYSICS, LITERATURE, GEOMETRY=0, HISTORY=3};`
- `MATHEMATICS=ALGEBRA=GEOMETRY=0`
- `PHYSICS=1`
- `LITERATURE=2`
- `HISTORY=3`

Άσκηση 1η (1/2)

- Για τους υπαλλήλους μιας εταιρείας θέλουμε να καταγράψουμε τα εξής στοιχεία:
  - επίθετο,
  - όνομα,
  - ηλικία,
  - φύλο,
  - κιν. τηλέφωνο και
  - μισθός των τελευταίων 3 ετών.

- Να προτείνετε μία δομή (struct) C με το όνομα emp που να επιτρέπει την ομαδοποίηση αυτών των στοιχείων.

Άσκηση 1η (2/2)

- Να γράψετε ένα πρόγραμμα C για την επεξεργασία των στοιχείων 50 υπαλλήλων.
  - Εισαγωγή από το χρήστη των πιο πάνω στοιχείων για τον κάθε υπάλληλο.
  - Για τον κάθε υπάλληλο υπολογισμός του μέσου όρου των μισθών των 3 τελευταίων ετών.
  - Για το τελευταίο έτος, ο μέσος όρος των μισθών των υπαλλήλων.
  - Τα στοιχεία του υπαλλήλου με τον μεγαλύτερο μισθό στο τελευταίο έτος.

Άσκηση 2η (1/2)

- Θεωρούμε ότι στα πλαίσια μιας χρηματιστηριακής εφαρμογής, τα ακόλουθα στοιχεία είναι διαθέσιμα για 100 μετοχές:
  - Το όνομα της μετοχής (το πολύ 5 εκτυπώσιμοι χαρακτήρες),
  - Οι τιμές κλεισίματος της μετοχής (ακρίβεια 2 δεκαδικών ψηφίων) τις τελευταίες 30 εργάσιμες ημέρες με αντίστροφη χρονολογική σειρά (η τρέχουσα ημέρα στην αρχή)
  - Το πλήθος των συναλλαγών της μετοχής για την τρέχουσα ημέρα.

Άσκηση 2η (2/2)

- Να υλοποιηθεί πρόγραμμα σε C που να χειρίζεται τα δεδομένα αυτά και να κάνει τα ακόλουθα:
  - Α) Να καλεί το χρήστη να εισάγει τα δεδομένα για την κάθε μετοχή.
  - Β) Να υπολογίζει και να εμφανίζει για την κάθε μετοχή το μέσο όρο των τιμών της για τις τελευταίες 10 ημέρες (εκτύπωση μίας μετοχής ανά γραμμή).
  - Γ) Να εμφανίζει την μετοχή με το μεγαλύτερο πλήθος συναλλαγών.
  - Δ) Να εμφανίζει την μετοχή η τιμή της οποίας αυξήθηκε σε μεγαλύτερο ποσοστό (%) την τρέχουσα ημέρα (σε σχέση με την προηγούμενη ημέρα).

Άσκηση 3η

- Για ένα σύνολο αυτοκινήτων υπάρχουν τα εξής στοιχεία:
  - ΑΡΙΘΜΟΣ (μόνο το αριθμητικό μέρος): π.χ. 4352
  - ΧΡΩΜΑ: π.χ. RED
  - ΚΑΤΑΣΚΕΥΑΣΤΗΣ: π.χ. ALFA-ROMEO
  - ΗΜΕΡΟΜΗΝΙΑ ΚΑΤΑΣΚΕΥΗΣ: π.χ. 2005

- Τα στοιχεία των αυτοκινήτων αποθηκεύονται στο αρχείο CARS.TXT. Σε κάθε γραμμή του αρχείου υπάρχει μία εγγραφή με τα στοιχεία ενός αυτοκινήτου ως εξής:
  - ΑΡΙΘΜΟΣ ΧΡΩΜΑ ΚΑΤΑΣΚΕΥΑΣΤΗΣ ΗΜΕΡΟΜΗΝΙΑ.

- Να γραφεί πρόγραμμα C που αφού διαβάσει τα στοιχεία των αυτοκινήτων θα υπολογίζει:
  - Τι ποσοστό των αυτοκινήτων κατασκευάστηκαν το 2006;
  - Τη διαφορά σε σχέση με το 2005 και το 2004.

Άσκηση 4η

- Να γραφεί πρόγραμμα σε γλώσσα C, το οποίο:
- Θα ζητά από τον χρήστη να εισάγει τις τιμές των δεδομένων για μια ομάδα ανθρώπων (το πολύ για μέχρι 100 άτομα) ως εξής:
  - πρώτα το επώνυμο, μετά η ημ. γέννησης (σύμφωνα με το πρότυπο HH/MM/EEEE, π.χ. 15/04/1983) και τέλος το ύψος (σε μέτρα).
  - Η διαδικασία πρέπει να τελειώνει είτε όταν συμπληρωθούν στοιχεία για 100 άτομα είτε όταν πληκτρολογηθεί μόνο το “x” στη θέση του ονόματος.
- Θα εκτυπώνει τον αριθμό των ατόμων που έχουν γεννηθεί σε κάθε μήνα.
- Θα εκτυπώνει τον μέσο όρο του ύψους ανά έτος γέννησης.

Τέλος Ενότητας

Σημείωμα Αναφοράς

- Copyright Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Στεργίου Κωνσταντίνος. «Εισαγωγή στον Δομημένο Προγραμματισμό». Έκδοση: 1.0. Κοζάνη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:

  https: //eclass.uowm.gr/courses/ICTE258/

Σημείωμα Αδειοδότησης

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.