

Προγραμματισμός σε C - Αρχεία

Ιωάννης Γ. Τσούλος

Τμήμα Πληροφορικής και τηλεπικοινωνιών
Πανεπιστήμιο Ιωαννίνων

2023

Περίληψη

- 1 Απλές δομές
- 2 Συναρτήσεις και δομές
- 3 Πίνακες και δομές

- Η δομή χρησιμοποιείται σαν συλλογή μεταβλητών διαφορετικού τύπου προκειμένου να περιγράψει συνολικά μια οντότητα. Για παράδειγμα η οντότητα ΑΝΘΡΩΠΟΣ αποτελείται από τα πεδία:
 - 1 Όνομα (αλφαριθμητικό)
 - 2 Επίθετο (αλφαριθμητικό)
 - 3 Ηλικία (ακέραιος αριθμός)
 - 4 Έτος (δεκαδικός αριθμός)

Δήλωση δομών

- Για την δήλωση των δομών χρησιμοποιείται ο προσδιοριστής `struct`, όπως στην συνέχεια:

```
struct Person
```

```
{  
char name[100];  
char lastname[100];  
int age;  
double height;  
};
```

Η δήλωση δομής δεν σημαίνει όμως και δήλωση μεταβλητής.

Αυτό πρέπει να γίνει μέσα σε κάποια συνάρτηση, όπως πχ στην `main()` όπως

```
struct Person myPerson;
```

Αυτή η δήλωση δημιουργεί μια μεταβλητή τύπου `Person` με το όνομα `myPerson`.

Εισαγωγή και εμφάνιση στοιχείων προσώπου

```
#include <stdio.h>
#include <stdio.h>
struct Person
{
    char name[100];
    char lastname[100];
    int age;
    double height;
};
int main()
{
    struct Person myPerson;
    printf("Doste_tou_stoixeia_tou_prosopou\n");
    printf("Doste_onoma\n");
    scanf("%s", myPerson.name);
    printf("Doste_epitheto\n");
    scanf("%s", myPerson.lastname);
    printf("Doste_ilikia\n");
    scanf("%d",&myPerson.age);
    printf("Doste_ypsos\n");
    scanf("%lf",&myPerson.height);
    printf("Emfanisi_stoixeion\n");
    printf("%s%s\n", myPerson.name, myPerson.lastname);
    printf("%d%lf\n", myPerson.age, myPerson.height);
    if (myPerson.age >= 18)
        printf("To_prosopo_einai_eniliko\n");
    else
        printf("To_prosopo_einai_aniliko\n");
    return 0;
}
```

Δομή Person

- Αυτή η δήλωση δημιουργεί μια μεταβλητή τύπου Person με το όνομα myPerson.
- Για την αναφορά στα πεδία της δομής χρησιμοποιείται ο τελεστής συμμετοχής (.)
- Κάθε πεδίο της μεταβλητής myPerson μπορεί να χρησιμοποιηθεί και σαν διαφορετική μεταβλητή, δηλαδή για παράδειγμα η μεταβλητή myPerson.age αντιμετωπίζεται σαν μια ξεχωριστή ακέραια μεταβλητή που στην περίπτωση του συγκεκριμένου προβλήματος αναπαριστά ηλικίες.

- Η εντολή typedef χρησιμοποιείται για να ορίσει νέους τύπους δεδομένων σαν ψευδώνυμα άλλων τύπων, προκειμένου για παράδειγμα να ορίσουν ονόματα που απομνημονεύονται ευκολότερα.
- Για παράδειγμα η δήλωση
`typedef double BigNumber;` ορίζει πως το όνομα `BigNumber` μπορεί να χρησιμοποιηθεί και σαν ψευδώνυμα του ενσωματωμένου τύπου δεδομένων `double`. Συνεπώς η δήλωση `double x` και η δήλωση `BigNumber x` είναι ακριβώς οι ίδιες. Επιπλέον η εντολή typedef μπορεί να χρησιμοποιηθεί για να αντικαταστήσει πίνακες με τύπους δεδομένων που ορίζονται από τον χρήστη

Δήλωση αλφαριθμητικού με typedef

- **# include** <stdio.h>
include <stdio.h>

```
typedef char String[100];  
int main()  
{  
    String myname;  
    printf("Ποιο είναι το όνομα σας\n");  
    scanf("%s", myname);  
    printf("Το όνομα σας είναι %s\n", myname);  
    return 0;  
}
```

Χρήση typedef σε αλφαριθμητικό

- Έγινε δήλωση ενός νέου τύπος για αλφαριθμητικά σαν ψευδώνυμο σε πίνακα γραμμάτων.
- Με αυτόν τον τρόπο ο χρήστης μπορεί να θέσει ονόματα που θυμάται εύκολα για τύπους μεταβλητών.

Η δομή Person με χρήση typedef

```

● # include <string.h>
  # include <stdio.h>
  typedef struct
  {
      char name[100];
      char lastname[100];
      int age;
      double height;
  } Person;
  int main(){
      Person myPerson;
      printf("Doste_onoma\n"); scanf("%s", myPerson.name);
      printf("Doste_epitheto\n");
      scanf("%s", myPerson.lastname);
      printf("Doste_ilikia\n");
      scanf("%d",&myPerson.age);
      printf("Doste_ypsos\n");
      scanf("%lf",&myPerson.height);
      printf("%s_%s\n", myPerson.name, myPerson.lastname);
      printf("%d_%lf\n", myPerson.age, myPerson.height);
      if (myPerson.age >= 18)
          printf("To_prosopo_einai_eniliko\n");
      else
          printf("To_prosopo_einai_aniliko\n");
      return 0;}

```

- Τα στοιχεία μιας δομής μπορούν να αρχικοποιηθούν με ξεχωριστές αναθέσεις.
- Μια δομή μπορεί να αρχικοποιηθεί και κατά την δήλωσή της

Παράδειγμα αρχικοποίησης δομής κατά την δήλωση

```
#include <string.h>
#include <stdio.h>
struct Person
{
    char name[100];
    char lastname[100];
    int age;
    double height;
};
int main()
{
    struct Person myPerson={"Giannis","Pappas",37,171};
    printf("Emfanisi stoixeion\n");
    printf("%s%s\n",myPerson.name,myPerson.lastname);
    printf("%d%lf\n",myPerson.age,myPerson.height);
    if(myPerson.age>=18)
        printf("To prosopo einai eniliko\n");
    else
        printf("To prosopo einai aniliko\n");
    return 0;
}
```

Γενικά στοιχεία

- Μια συνάρτηση μπορεί να δεχτεί δομή σαν όρισμα
- Μια συνάρτηση μπορεί να επιστρέψει δομή με την χρήση του return

Εμφάνιση δομής με συνάρτηση

- **typedef struct**

```
{  
    int hour;  
    int minute;  
    int second;  
} Time;  
  
void printTime(Time t, int flag24)  
{  
    if (flag24==1)  
        printf ("%d:%d:%d\n",  
            t.hour, t.minute, t.second);  
    else  
        printf ("%d:%d:%d\n",  
            t.hour-12, t.minute, t.second);  
}
```

Εμφάνιση δομής με συνάρτηση

- Η συνάρτηση `printTime()` δέχεται δύο ορίσματα:
 - Μια μεταβλητή τύπου `Time` με το όνομα `t`. Αυτή η μεταβλητή αναπαριστά την χρονική στιγμή που θα εμφανιστεί στην οθόνη.
 - Μια ακέραια μεταβλητή με το όνομα `flag24`. Αν αυτή η τιμή είναι 1 (αληθής), τότε η χρονική στιγμή θα εμφανιστεί στην οθόνη με 24ωρη μορφή αλλιώς με 12ώρη μορφή.

Ανάγνωση δομής από το πληκτρολόγιο

- Time readTime()
{
 Time t;
 printf("Doste_ora_");
 scanf("%d",&t.hour);
 printf("Doste_lepta_");
 scanf("%d",&t.minute);
 printf("Doste_deyterolepta_");
 scanf("%d",&t.second);
 return t;
}

- Η συνάρτηση `readTime` κατασκευάζει τοπικά μια δομή
- Η συνάρτηση `readTime` διαβάζει την τοπική δομή
- Η συνάρτηση `readTime` επιστρέφει την τοπική δομή

Σύγκριση δομών

```
• int      Time2Seconds(Time t)
{
    return t.hour*24*60*60+
           t.minute*60+t.second;
}
int      compareTime(Time t1,Time t2)
{
    int t1seconds=Time2Seconds(t1);
    int t2seconds=Time2Seconds(t2);
    if(t1seconds>t2seconds) return 1;
    else
    if(t1seconds<t2seconds) return -1;
    else return 0;
}
```

- Οι δομές είναι σύνθετοι τύποι δεδομένων και σαν τέτοιοι δεν μπορούν να συγκριθούν άμεσα.
- Η συνάρτηση `compareTime()` πρώτα μετατρέπει τις χρονικές στιγμές σε δευτερόλεπτα με την χρήση της βοηθητικής συνάρτησης `Time2Seconds()` και στην συνέχεια συγκρίνει τα δευτερόλεπτα που προκύπτουν για κάθε χρονική στιγμή.
- Η συνάρτηση `compareTime` επιστρέφει
 - 1 αν το πρώτο όρισμα είναι πιο μετά από το δεύτερο όρισμα στον χρόνο.
 - 0, αν οι δύο χρονικές στιγμές είναι ίδιες
 - -1, αν η δεύτερη χρονική στιγμή είναι πιο μετά στον χρόνο από την πρώτη. `\end{itemize}`

- Όπως ένας πίνακας μπορεί να περιέχει απλά στοιχεία έτσι μπορεί να περιέχει και δομές.
- Για παράδειγμα οι μαθητές μιας τάξης θα μπορούσαν να αποτελέσουν έναν πίνακα από δομές όπως και οι υπάλληλοι μιας επιχείρησης.

Πίνακας υπαλλήλων

```
#include <stdio.h>
typedef struct
{
    char lastname[100];
    int depid;
    double salary;
}Employee;
Employee readEmployee()
{
    Employee p;
    printf("Doste ta stoixeia toy ypallilou \n");
    scanf("%s",p.lastname);
    scanf("%d",&p.depid);
    scanf("%lf",&p.salary);
    return p;
}
int main(){
    Employee table[5];
    int i;
    int total=0;
    for(i=0;i<5;i++)
    {
        table[i]=readEmployee();
        if(table[i].salary<1000) total++;
    }
    printf("Synolika ypάρχουν %d ypalliloi\n",total);
    return 0;
}
```

- Στο παράδειγμα του προηγούμενου κώδικα δημιουργείται ένας πίνακας από υπαλλήλους με πεδία: Επίθετο, Αριθμό τμήματος και Μισθό.
- Στην συνέχεια εμφανίζεται στην οθόνη το πλήθος των υπαλλήλων που λαμβάνουν λιγότερα από 1000 ευρώ τον μήνα.

Σύνοψη

- Απλές δομές
- Συναρτήσεις και δομές
- Πίνακες δομών