

Δομές

Ιωάννης Γ. Τσούλος

2020

1 Απλές 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. Ύψος (δεκαδικός αριθμός)

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 μιας δομής μοιάζει αρκετά με την δήλωση συναρτήσεων, χωρίς όμως τον εκτελέσιμο κώδικα. 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`. Ένα πλήρες παράδειγμα χρήσης δομής παρουσιάζεται στον αλγόριθμο 1. 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.

Algorithm 1 Εισαγωγή και εμφάνιση στοιχείων προσώπου.

```
1 # include <string.h>
2 # include <stdio.h>
3 struct Person
4 {
5     char name[100];
6     char lastname[100];
7     int age;
8     double height;
9 };
10 int main()
11 {
12     struct Person myPerson;
13     printf("Doste_ta_stoixeia_tou_prosopou\n");
14     printf("Doste_onoma\n");
15     scanf("%s", myPerson.name);
16     printf("Doste_epitheto\n");
17     scanf("%s", myPerson.lastname);
18     printf("Doste_ilikia\n");
19     scanf("%d",&myPerson.age);
20     printf("Doste_ypsos\n");
21     scanf("%lf",&myPerson.height);
22     printf("Emfanisi_stoixeion\n");
23     printf("%s_%s\n", myPerson.name, myPerson.lastname);
24     printf("%d_%lf\n", myPerson.age, myPerson.height);
25     if(myPerson.age >= 18)
26         printf("To_prosopo_einai_eniliko\n");
27     else
28         printf("To_prosopo_einai_aniliko\n");
29     return 0;
30 }
```

Algorithm 2 Δήλωση τύπου αλφαριθμητικού.

```
1 # include <string.h>
2 # include <stdio.h>
3
4 typedef char String[100];
5
6 int main()
7 {
8     String myname;
9     printf("Poio_einai_to_onoma_sas_\n");
10    scanf("%s",myname);
11    printf("To_onoma_sas_einai_%s\n",myname);
12    return 0;
13 }
```

2 Χρήση του typedef

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 ψευδώνυμο 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, όπως στο παράδειγμα 2, όπου δηλώνεται ένας νέος τύπος για αλφαριθμητικά σαν ψευδώνυμο σε πίνακα γραμμάτων.

Σε πολλές περιπτώσεις δεν είναι βολικό ο χρήστης να πρέπει να δηλώσει μια μεταβλητή δομής με την χρήση του

```
struct Person myPerson;
```

Θα ήταν προτιμότερο η παραπάνω δήλωση να αντικατασταθεί με

```
Person myPerson;
```

Για να γίνει αυτό η δήλωση της δομής μπορεί να γίνει με την βοήθεια της εντολής typedef που παρουσιάστηκε πριν. Ένα ενδεικτικό παράδειγμα χρήσης παρουσιάζεται στον αλγόριθμο 3.

3 Αρχικοποίηση δομής

Μια μεταβλητή δομής μπορεί να αρχικοποιηθεί με αναθέσεις και ανάγνωση τιμών αλλά μπορεί να αρχικοποιηθεί και κατά την δήλωση, όπως και στην περίπτωση των

Algorithm 3 Ο τύπος δομής Person με χρήση typedef.

```
1 # include <string.h>
2 # include <stdio.h>
3 typedef struct
4 {
5     char name[100];
6     char lastname[100];
7     int age;
8     double height;
9 }Person;
10
11 int main()
12 {
13     Person myPerson;
14     printf("Doste_ta_stoixeia_tou_prosopou_\n");
15     printf("Doste_onoma_\n");
16     scanf("%s", myPerson.name);
17     printf("Doste_epitheto_\n");
18     scanf("%s", myPerson.lastname);
19     printf("Doste_ilikia_\n");
20     scanf("%d",&myPerson.age);
21     printf("Doste_ypsos_\n");
22     scanf("%lf",&myPerson.height);
23     printf("Emfanisi_stoixeion_\n");
24     printf("%s_%s_\n", myPerson.name, myPerson.lastname);
25     printf("%d_%lf_\n", myPerson.age, myPerson.height);
26     if (myPerson.age >= 18)
27         printf("To_prosopo_einai_eniliko_\n");
28     else
29         printf("To_prosopo_einai_aniliko_\n");
30     return 0;
31 }
```

Algorithm 4 Αρχικοποίηση δομής κατά την δήλωση.

```
1 # include <string.h>
2 # include <stdio.h>
3 struct Person
4 {
5     char name[100];
6     char lastname[100];
7     int age;
8     double height;
9 };
10 int main()
11 {
12     struct Person myPerson={"Giannis", "Pappas", 37, 171};
13     printf("Emfanisi_stoixeion_\n");
14     printf("%s_%s\n", myPerson.name, myPerson.lastname);
15     printf("%d_%lf\n", myPerson.age, myPerson.height);
16     if(myPerson.age >= 18)
17         printf("To_prosopo_einai_eniliko_\n");
18     else
19         printf("To_prosopo_einai_aniliko_\n");
20     return 0;
21 }
```

πινάκων. Ένα απλό παράδειγμα αρχικοποίησης της δομής προσώπου παρουσιάζεται στον αλγόριθμο 4.

4 Συναρτήσεις και δομές

Οι δομές μπορούν να χρησιμοποιηθούν σαν ορίσματα σε συναρτήσεις αλλά και σαν τιμές επιστροφής. Στο παράδειγμα του αλγορίθμου 5 παρουσιάζεται μια εφαρμογή που διαβάζει την ώρα σε μορφή δομής και την εμφανίζει σε 12ωρη ή σε 24ωρη μορφή.

5 Σύγκριση δομών

Οι δομές είναι σύνθετοι τύποι δεδομένων και σαν τέτοιοι δεν μπορούν να συγκριθούν άμεσα. Το παράδειγμα

```
Time t1, t2;
if (t1==t2)
{
}
```

Algorithm 5 Εφαρμογή ανάγνωσης και εμφάνισης της ώρας.

```
1 # include <stdio.h>
2
3 typedef struct
4 {
5     int hour;
6     int minute;
7     int second;
8 }Time;
9
10 Time readTime()
11 {
12     Time t;
13     printf("<<"Doste_ora_");
14     scanf("%d",t.hour);
15     printf("Doste_lepta_");
16     scanf("%d",&t.minute);
17     printf("Doste_deyterolepta_");
18     scanf("%d",&t.second);
19     return t;
20 }
21
22 void printTime(Time t,int flag24)
23 {
24     if(flag24==1)
25         printf("%d:%d:%d\n",
26             t.hour,t.minute,t.second);
27     else
28         printf("%d:%d:%d\n",
29             t.hour-12,t.minute,t.second);
30 }
31
32 int main()
33 {
34     Time mytime;
35     mytime=readTime();
36     printTime(mytime,0);
37     printTime(mytime,1);
38     return 0;
39 }
```

δεν είναι σωστό και θα πρέπει να γίνει σύγκριση των πεδίων των δύο δομών ένα προς ένα προκειμένου να γίνει έλεγχος, όπως παρουσιάζεται και στον αλγόριθμο 6, όπου δύο χρονικές στιγμές συγκρίνονται από την συνάρτηση compareTime(). Η συνάρτηση επιστρέφει 1 αν η πρώτη χρονική στιγμή είναι πιο μετά στον χρόνο, 0 για ισότητα και -1 αν η δεύτερη χρονική στιγμή είναι πιο μετά.

6 Δομές που έχουν δομές σαν πεδία

Οι δομές μπορούν να βρίσκονται και σαν πεδία άλλων δομών όταν αυτό απαιτείται. Για παράδειγμα η δομή ΑΝΘΡΩΠΟΣ μπορεί να περιλαμβάνει την δομή ΗΜΕΡΟΜΗΝΙΑ σαν πεδίο του, πχ σαν το πεδίο ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ. Το θέμα αυτό παρουσιάζεται στον αλγόριθμο 7.

7 Πίνακες δομών

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. Στο παράδειγμα του αλγορίθμου 8 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.

8 Δομές με πίνακες σαν πεδία

Ένα ακόμα παράδειγμα χρήσης δομών είναι η χρήση πινάκων σαν πεδία δομών. Για παράδειγμα ένας μαθητής μπορεί να έχει τον αριθμό μητρώου του και έναν πίνακα με τις βαθμολογίες στα μαθήματα του εξαμήνου του. Στο παράδειγμα του αλγορίθμου 9 δημιουργείται ένας πίνακας μαθητών με 5 μαθήματα ο καθένας στο εξάμηνο. Η εφαρμογή εμφανίζει για κάθε μαθητή τον μέσο όρο στο εξάμηνό του.

Algorithm 6 Σύγκριση χρονικών στιγμών.

```
1 # include <stdio.h>
2
3 typedef struct
4 {
5     int hour;
6     int minute;
7     int second;
8 }Time;
9
10 Time readTime ()
11 {
12     Time t;
13     printf("<<"Doste_ora_");
14     scanf ("%d", &t.hour);
15     printf ("Doste_lepta_");
16     scanf ("%d",&t.minute);
17     printf ("Doste_deyterolepta_");
18     scanf ("%d",&t.second);
19     return t;
20 }
21
22
23 int Time2Seconds (Time t)
24 {
25     return t.hour*24*60*60+t.minute*60+t.second;
26 }
27
28 int compareTime (Time t1 ,Time t2)
29 {
30     int t1seconds=Time2Seconds (t1);
31     int t2seconds=Time2Seconds (t2);
32     if (t1seconds>t2seconds) return 1;
33     else
34     if (t1seconds<t2seconds) return -1;
35     else return 0;
36 }
37
38 int main ()
39 {
40     Time time1 ,time2;
41     int t;
42     time1=readTime ();
43     time2=readTime ();
44     t=compareTime (time1 ,time2);
45     if (t==1) printf ("I_proti_ora_einai_megalyteri_\n");
46     else
47     if (t== -1) printf ("I_deyteri_ora_einai_megalyteri_\n");
48     else printf ("Einai_oi_idies_\n");
49     return 0;
50 }
```

Algorithm 7 Δομή σαν πεδίο άλλης δομής.

```
1 # include <stdio.h>
2
3 typedef struct
4 {
5     int year;
6     int month;
7     int day;
8 }Date;
9
10 typedef struct
11 {
12     char name[100];
13     char lastname[100];
14     Date birthdate;
15 }Person;
16 Date readDate()
17 {
18     Date d;
19     printf("Doste_imerominia_");
20     scanf("%d_%d_%d",&d.day,&d.month,&d.year);
21     return d;
22 }
23 void printDate(Date d)
24 {
25     printf("%d/%d/%d\n",d.day,d.month,d.year);
26 }
27 Person readPerson()
28 {
29     Person p;
30     printf("Doste_ta_stoixeia_sas_\n");
31     scanf("%s",p.name);
32     scanf("%s",p.lastname);
33     p.birthdate=readDate();
34     return p;
35 }
36 void printPerson(Person p)
37 {
38     printf("Ta_stoixeia_sas_\n");
39     printf("%s,%s\n",p.name,p.lastname);
40     printDate(p.birthdate);
41 }
42 int main()
43 {
44     Person myPerson;
45     myPerson=readPerson();
46     printPerson(myPerson);
47     return 0;
48 }
```

Algorithm 8 Πίνακας υπαλλήλων.

```
1 # include <stdio.h>
2
3 typedef struct
4 {
5     char lastname[100];
6     int depid;
7     double salary;
8 }Employee;
9
10 Employee readEmployee()
11 {
12     Employee p;
13     printf(Doste ta stoixeia toy ypallilou \n");
14     scanf ("%s",p.lastname);
15     scanf ("%d",&p.depid);
16     scanf ("%lf",&p.salary);
17     return p;
18 }
19
20 int main()
21 {
22     Employee table[5];
23     int i;
24     int total=0;
25     for (i=0;i<5;i++)
26     {
27         table[i]=readEmployee();
28         if(table[i].salary<1000) total++;
29     }
30     printf("Synolika yparxoun %d ypalliloi
31     me <1000 euro amoibi \n",total);
32     return 0;
33 }
```

Algorithm 9 Πίνακας μαθητών.

```
1 # include <stdio.h>
2
3 typedef struct
4 {
5     int studentid;
6     double lesson[5];
7 }Student;
8
9 Student readStudent()
10 {
11     Student s;
12     int i;
13     printf("Doste_ton_kodiko_tou_mathiti_");
14     scanf("%d",&s.studentid);
15     for(i=0;i<5;i++)
16     {
17         printf("Doste_mathima_%d\n",i);
18         scanf("%lf",&s.lesson[i]);
19     }
20     return s;
21 }
22
23 double_average(Student_s)
24 {
25     double_sum=0.0;
26     int_i;
27     for(i=0;i<5;i++)
28         sum=sum+s.lesson[i];
29     return_sum/5.0;
30 }
31
32 int_main()
33 {
34     Student_myclass[5];
35     int_i;
36     double_s;
37     for(i=0;i<5;i++)
38     {
39         myclass[i]=readStudent();
40         s=average(myclass[i]);
41         printf("O mesos oros einai %lf\n",s);
42     }
43     return_0;
44 }
```
