

Programiranje u fizici

14. Rad sa strukturama

Prirodno-matematički fakultet u Nišu
Departman za fiziku

Strukture koristimo kada opisujemo složene pojmove, npr. osobu za koji navodimo dosta podataka poput imena, prezimena, godišta, telefona i drugih.

Podaci mogu biti standardni tipovi podataka programskog jezika C ili nove strukture sa svojim podacima.

```
struct Osoba {  
    char ime[20];  
    char prezime [20];  
    int godiste;  
    char telefon [10];  
};
```

```
struct Osoba pera, mika, zika;
```

```
struct Osoba {  
    char ime[20];  
    char prezime [20];  
    int godiste;  
    char telefon [10];  
};
```

```
struct Osoba pera, mika, zika;
```

Ovako definisane osobe pera, mika i zika imaju svoja imena, prezimena i ostale podatke koji su definisani u strukturi osoba.

Za pristup pojedinačnim podacima koristise tačka, npr. *mika.ime*, *pera.telefon* ili *zika.godiste*.

- Nepraktično je svaki put pisati *struct* Osoba kada treba definisati neku novu promenljivu tipa Osoba.
- Postoji elegantniji način da se to zaobiđe pomoću *typedef*.

```
typedef struct {  
    char ime[20];  
    char prezime [20];  
    int godiste;  
    char telefon [10];  
} Osoba;
```

```
Osoba pera, mika, zika;
```

```
#include <stdio.h>

typedef struct {
    char ime[20];
    char prezime[20];
    int god;
} osoba;

int main()
{
    osoba o1,o2,m;

    printf("Unesite ime i prezime prve osobe: ");
    scanf("%s %s", &o1.ime, &o1.prezime);

    printf("Unesite godiste prve osobe: ");
    scanf("%d", &o1.god);

    printf("Unesite ime i prezime druge osobe: ");
    scanf("%s %s", &o2.ime, &o2.prezime);

    printf("Unesite godiste druge osobe: ");
    scanf("%d", &o2.god);

    m=o1.god > o2.god ? o1 : o2;

    printf("Ime i prezime: %s %s God. %d\n", m.ime, m.prezime, m.god);

    return 0;
}
```

Moguće je definisati pokazivač na strukturu kao što definišemo pokazivač na bilo koju drugu promenljivu u programu:

```
struct Books *struct_pointer;
```

Sada je moguće dodeliti adresu promenljive koja je tipa strukture u ovako definisan pokazivač

```
struct_pointer = &Book1;
```

Za pristup članovima strukture korišćenjem pokazivača na strukturu koristi se operator ->

```
struct_pointer->title
```

```
#include <stdio.h>
#include <string.h>
struct knjiga
{
    char naslov[50];
    char autor[50];
    char tema[100];
    int knjiga_id;
};
void printKnjiga (struct knjiga *knjiga);
int main ( )
{
    struct knjiga knjiga1; /* prom. knjiga1 tipa knjiga */
    struct knjiga knjiga2; /* prom. knjiga2 tipa knjiga */

    strcpy (knjiga1.naslov, "Zbirka zadataka iz C");
    strcpy (knjiga1.autor, "Mika Mikic");
    strcpy (knjiga1.tema, "Zbirka resenih zadataka iz C");
    knjiga1.knjiga_id = 6495407;

    strcpy (knjiga2.naslov, "Knjiga o Londonu");
    strcpy (knjiga2.autor, "Milos Crnjanski");
    strcpy (knjiga2.tema, "Roman");
    knjiga2.knjiga_id = 6495700;

    printKnjiga (&knjiga1 );
    printKnjiga (&knjiga2);
    return 0;
}
```

```
void printKnjiga (struct knjiga *knjiga)
{
    printf ( "Knjiga naslov : %s\n", knjiga->naslov);
    printf ( "Knjiga autor  : %s\n", knjiga->autor);
    printf ( "Knjiga tema   : %s\n", knjiga->tema);
    printf ( "Knjiga id    : %d\n", knjiga->knjiga_id);
}
```

```
Knjiga naslov : Zbirka zadataka iz C
Knjiga autor  : Mika Mikic
Knjiga tema   : Zbirka resenih zadataka iz C
Knjiga id    : 6495407
Knjiga naslov : Knjiga o Londonu
Knjiga autor  : Milos Crnjanski
Knjiga tema   : Roman
Knjiga id    : 6495700
```