

Strukture

predavač: Nadežda Jakšić

Programiranje programski jezik C

Strukture

složeni tip podatka; kolekcija promenljivih različitog tipa koje su sakupljene radi lakše manipulacije

sintaksa:

```
struct [oznakaStrukture]
{
    definicija člana strukture;
    definicija člana strukture;
    ... } a,b,c;
```

//a,b,c su promenljive koje su
tipa strukture oznakaStrukture

identifikatori su vidljivi samo unutar strukture kojoj pripadaju, mogu imati ista imena u različitim strukturama

oznakaStrukture može,
ali ne mora da se
navede
to je naziv tipa podatka

```
struct Student
{
    char ime[50];
    char prezime[50];
    int brojPoena;
} s1;
```

Strukture

- najčešće se definišu iznad funkcije **main**
- svaka struktura je poseban tip podataka; mogu se deklarirati promenljive tipa strukture (strukturne promenljive) i u okviru tih promenljivih može se pristupati pojedinačnim elementima strukture, mogu im se dodeljivati, čitati i menjati vrednosti i mogu se koristiti u izrazima

```
struct Student s1 = {"Pera", "Peric", 30};
```

deklariše se nova promenljiva **s1** koja je tipa strukture Student i dodeljuju joj se vrednosti za svaki element strukture

- pojedinačnim elementima strukture pristupa se korišćenjem tačke **nazivPromenljive.nazivElementa**
- primer

```
s1.brojPoena = 35;  
gets (s1.ime);  
printf ("%s %s %d", s1.ime, s1.prezime, s1.brojPoena);
```

Struktura u strukturi

- elementi strukture mogu biti nove strukture (na primer ako hoćemo da studentu dodamo adresu koja će imati poseban podatak o ulici, broju i mestu)

```
struct Adresa {  
    char ulica[200];  
    char broj[10];  
    char mesto[100];  
};
```

```
struct Student {  
    char ime[50];  
    char prezime[50];  
    int brojPoena;  
    struct Adresa adresa; };s1
```

pristup elementima adrese (unutrašnje strukture) implementira se sa **s1.adresa.broj**

Inicijalizacija strukture

- u okviru strukture

```
struct datum  
{int dan;  
char mesec[15];  
int godina;}  
danUGodini={1,"septembar",2018};
```

- ili

```
struct datum danUGodini = {1,"septembar",2018};
```

- ili

```
struct datum  
{int dan;  
char mesec[15];  
int godina;} danUGodini;  
danUGodini.dan=1;  
danUGodini.mesec= "septembar";  
danUGodini.godina=2018;
```

Primer

```
int main()
{
    struct Ucenik
    {
        int rbr;
        char prezime [20];
        char ime [20];
        char datumRodjenja [20];
    };
    struct Ucenik ucenik1;
    printf ("Unesite podatke za ucenika:\n");
    printf ("Redni broj ucenika:\n");
    scanf ("%d",&ucenik1.rbr);
    printf ("Prezime:\n");
    scanf ("%s",&ucenik1.prezime);
```

```
printf ("Ime:\n");
scanf ("%s",&ucenik1.ime);
printf ("Datum rodjenja:\n");
scanf ("%s",&ucenik1.datumRodjenja);
printf ("Ispis podataka:\n\n");
printf ("Redni broj: %d\n", ucenik1.rbr);
printf ("Prezime: %s\n", ucenik1.prezime);
printf ("Ime: %s\n", ucenik1.ime);
printf ("Datum rodjenja:
%s\n",ucenik1.datumRodjenja);
return 0;}
```

Primer

//unose se koordinate tačaka, računa se površina i obim trougla

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

```
#include <math.h>
```

```
struct tacka
```

```
{ int x;
```

```
  int y;};
```

```
float razdaljina (struct tacka a, struct tacka b)
```

```
{return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y));}
```

```
float obimTrougla (struct tacka a, struct tacka b, struct tacka c)
```

```
{return razdaljina (a, b) + razdaljina (a, c) + razdaljina (b, c);}
```

```
float površinaTrougla (struct tacka a, struct tacka b, struct tacka c)
```

```
{float poluobim, str1, str2, str3;
```

```
  poluobim = obimTrougla (a, b, c)/2;
```

```
  str1 = razdaljina (a, b);
```

```
  str2 = razdaljina (a, c);
```

```
  str3 = razdaljina (b, c);
```

```
  return sqrt(poluobim * (poluobim - str1) * (poluobim - str2) * (poluobim - str3));}
```

Primer - nastavak

//unose se koordinate tačka, računa se površina i obim trougla

```
int main()
{
    struct tacka a, b, c;
    float p, o;
    printf ("Unesite koordinate temena A: ");
    scanf ("%d %d", &a.x, &a.y);
    printf ("Unesite koordinate temena B: ");
    scanf ("%d %d", &b.x, &b.y);
    printf ("Unesite koordinate temena C: ");
    scanf ("%d %d", &c.x, &c.y);
    p = površinaTrougla(a, b, c);
    o = obimTrougla(a, b, c);
    printf ("Površina = %.2f, Obim = %.2f\n", p, o);
    return 0;}

```


Definicija tipa - typedef

- strukture se često koriste zajedno sa oznakom za definiciju tipa **typedef**
- **typedef** se koristi za definisanje novih tipova na osnovu postojećih i preimenovanje tipa
typedef int ceoBroj; //uvodi novo ime ceoBroj za tip podataka int
sada promenljiva čiji je tip **int** može da se deklariše ovako
ceoBroj br;
- u strukturama **typedef** uvodi novi tip podataka strukture
typedef struct
{
char ime[50];
char prezime[50];
int brojPoena; }Student;
posle ovoga deklaracija promenljivih tipa student se navodi bez reči **struct**, npr. **Student s1,s2;**

Typedef - primer

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
typedef struct
{
    char naslov[50];
    char autor[50];
    char tema [100];
    int knjigald;
} Knjiga;
int main( ) {
```

```
    Knjiga knjiga; //promenljiva knjiga je tipa strukture Knjiga
    strcpy( knjiga.naslov, "C Programming");
    strcpy( knjiga.autor, "Dennis MacAlistair Ritchie");
    strcpy( knjiga.tema, "C Programming Tutorial");
    knjiga.knjigald = 6495407;
```

```
    printf ("Naslov knjige: %s\n", knjiga.naslov);
    printf ("Autor: %s\n", knjiga.autor);
    printf ("Tema: %s\n", knjiga.tema);
    printf ("Identifikacioni broj knjige: %d\n", knjiga.knjigald);
    getch ();
    return 0;
}
```

Programi

- unosi se ime, prezime, godina studija i prosek za određeni broj studenata (**n**); napraviti strukturu **Student** i niz **studenti[n]** koji je tipa strukture **Student**; uneti sve podatke za **n** studenata i prikazati studenta sa najboljim prosekom kao i sve njegove podatke
- uneti podatke za model, godište i cenu određenog broja automobila ispisati podatke o automobilima čija je cena preko 10 000 evra
- unose se podaci za ime, prezime, matični broj, prosek i datum rođenja za sve učenike jednog razreda; nakon unosa svih podataka prikazuje se učenik sa najboljim prosekom i na kraju se ispisuju njegovi podaci; napraviti strukturu **Datum**, strukturu **Ucenik** i niz **razred[30]** koji je tipa strukture **Ucenik**; na početku programa korisnik unosi broj učenika, ako unese broj koji je veći od 30 prikazuje se poruka "Broj učenika ne može da bude veći od 30" i izlazi se iz programa

Programi

- unose se podaci za ime, prezime i telefonski broj određenog broja pretplatnika - sortirati imenik po prezimenima (napisati funkciju za sortiranje kao i funkciju koja zamenjuje mesta), prebrojati pretplatnike i za uneto prezime ispisati podatke
funkcija **strcmpi()** je ista kao **strcmp()** - poredi stringove, ali nije **case sensitive**, ne vodi računa o tome da li su slova mala ili velika, **a** je isto što i **A**