



FERIT

Programiranje 2

Laboratorijske vježbe

LV3 Strukture

**Fakultet elektrotehnike računarstva i
informacijskih tehnologija Osijek**

Kneza Trpimira 2b

www.ferit.unios.hr

Uvod

Za pohranjivanje jedne vrijednosti određenog tipa podatka koristi se varijabla. Kada je potreba za pohranjivanjem više vrijednosti istoga tipa podatka koristilo se polje. Vrlo je često praktična potreba za grupiranjem različitih tipova srodnih podataka pod jednim imenom i za to se koristi struktura. Struktura je korisnički definiran tip podatka koji omogućava grupiranje različitih tipova srodnih podataka pod jednim imenom i kreiranje novog složenog tipa podatka. Kreiranjem varijable strukture omogućava se praktičan način dohvaćanja grupe različitih srodnih podataka korištenjem samo jednog imena. Ovim načinom se program čini modularnijim, jer se pojednostavljuje pristup vrijednostima. Strukturom se omogućava opis objekta iz stvarnog svijeta. Strukture su jako korisne u situacijama gdje je potrebno grupirati veliki broj različitih podataka, npr. za pohranjivanje vrijednosti u bazu podataka, informacije o kontaktima, adresama itd.

Definicija strukture

Struktura je korisnički kreirani složeni tip podatka koji se prvo mora definirati kako bi se mogao koristiti.

Opći oblik definicije strukture:

```
struct ime_strukture {  
    tip_podatka_1 ime_varijable_1;  
    tip_podatka_2 ime_varijable_2;  
    ... ...;  
    tip_podatka_n ime_varijable_n;  
}[jedna ili više varijabli strukture, kao i polje strukture];
```

gdje je:

- `struct` je ključna riječ kojom se definira struktura kao tip podatka
- `ime_strukture` je naziv koji se daje strukturi kako bi se imenovao tip podatka
- `{}` predstavljaju tijelo definicije strukture
- `tip_podatka_1 ime_varijable_1, ..., tip_podatka_n ime_varijable_n` su članovi strukture
- na samom kraju definicije strukture, može se navesti jednu ili više varijabli strukture ili polje strukture
- definicija strukture mora završiti sa znakom `(;)`.

Definicija strukture može se nalaziti na:

- početku programa izvan svih funkcija (globalna deklaracija)
- unutar funkcije (lokalna deklaracija)
- unutar nekog bloka naredbi.

Član strukture može biti:

- bilo koji ugrađeni (primitivni) tip podatka, npr. *char*, *short*, *unsigned short*, *int*, *unsigned int*, *long*, *unsigned long*, *float*, *double*
- bilo koji složeni tip podatka, npr. struktura kao član strukture
- jednodimenzionalna i višedimenzionalna polja kreirana različitim tipom podatka, npr. ugrađenim ili složeni tipom podatka
- pokazivači na različite tipove podataka, npr. ugrađene ili složene tipove podataka.

Primjer 1: Primjer definicije strukture.

```
struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};
```

Primjer 2: Primjer definicije strukture korištenjem koncepta struktura unutar strukture.

```
struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
};
```

Primjer 3: Primjer definicije strukture izvan svih funkcija .

```
struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

int main(void) {
    return 0;
}
```

Primjer 4: Primjer definicije strukture unutar funkcije.

```
int main(void) {  
    struct student {  
        char ime[20];  
        char prezime[20];  
        char indeks[10];  
        float prosjek;  
        unsigned short dan;  
        unsigned short mjesec;  
        unsigned short godina;  
    };  
  
    return 0;  
}
```

Primjer 5: Primjer definicije strukture unutar bloka.

```
int main(void) {  
    {  
        struct student {  
            char ime[20];  
            char prezime[20];  
            char indeks[10];  
            float prosjek;  
            unsigned short dan;  
            unsigned short mjesec;  
            unsigned short godina;  
        };  
    }  
  
    return 0;  
}
```

Deklariranje varijable strukture i inicijaliziranje

Definicija strukture samo opisuje složeni tip podatka i ne zauzima memorijski prostor. Tek nakon što je struktura definirana može se deklarirati varijabla strukture za koju se zauzima memorijski prostor. Definicija strukture predstavlja samo nacrt, opis složenog tipa podatka, a varijabla strukture predstavlja konkretnu realizaciju strukture. Treba napomenuti kako su članovi unutar definicije strukture povezani s varijablom strukture, ne postoje ako se ne kreira varijabla strukture. Iz jedne definicije strukture može se kreirati mnogo varijabli strukture koje individualno sadrže svoje članove strukture, ne dijeleći ih između pojedine varijable strukture.

Primjer 6: Primjer deklariranja varijable strukture.

```
struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
}/*najStudent*/;

int main(void) {

    struct student najStudent;

    return 0;
}
```

Na temelju složenog tipa podatka `struct student` kreirana je varijabla strukture `najStudent` za koju se zauzima memorijski prostor. Deklariranje varijable strukture može se kreirati na samom kraju definicije strukture poslije zatvorene vitičaste zagrade ili naknadno unutar kôda. Minimalni broj bajtova koji zauzima varijabla strukture `najStudent` dobije se sumiranjem ukupnog broja bajtova svakog člana strukture i za varijablu strukture `najStudent` iznosi minimalno 60 bajta, odnosno 64 bajta zbog poravnanja podataka i pakiranje strukture u memoriji.

Kada se kreira varijabla strukture, njezini članovi su ne inicijalizirani, i ne može im se pristupiti, u slučaju pristupanja članovima strukture koji nisu inicijalizirani program će se srušiti. Stoga, članove deklarirane strukture moguće je inicijalizirati.

Primjer 7: Primjer inicijalizacije varijable strukture.

```
struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

int main(void) {

    struct student najStudent = {"Ivan", "Ivic", "10879", 4.56f, 5, 5, 1998};
    //struct student najStudent = { 0 };

    return 0;
}
```

Primjer 8: Primjer inicijalizacije varijable strukture.

```
struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
};

int main(void) {

    struct student najStudent = {"Ivan", "Ivic", "10879", 4.56f, {5, 5, 1998}};
    //struct student najStudent = { 0 };

    return 0;
}
```

Pristup članovima strukture

Kako ne postoji format za zapis ili ispis varijable strukture, potrebno je individualno pristupiti svakom članu varijable strukture korištenjem operator točka (.) koji se navodi između imena varijable strukture i člana strukture koji se želi dohvatiti.

U slučaju da je varijabla strukture ne inicijalizira, potrebno je prvo svakom članu varijable strukture pristupiti zasebno i pridružiti određenu vrijednost.

Primjer 9: Pristup elementima strukture radi zapisivanja i dohvaćanja sadržaja.

```
#include<stdio.h>

struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};
struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
};

int main(void) {
    struct student najStudent;

    printf("Unesite ime\n");
    scanf("%19s", najStudent.ime);
    printf("Unesite prezime\n");
    scanf("%19s", najStudent.prezime);
    printf("Unesite indeks studenta\n");
    scanf("%9s", najStudent.indeks);
    printf("Unesite prosjek studenta\n");
    scanf("%f", &najStudent.prosjek);
    printf("Unesite dan rodjendan studenta\n");
    scanf("%hu", &najStudent.datumRodjenja.dan);
    printf("Unesite mjesec rodjendan studenta\n");
    scanf("%hu", &najStudent.datumRodjenja.mjesec);
    printf("Unesite godinu rodjendan studenta\n");
    scanf("%hu", &najStudent.datumRodjenja.godina);

    printf("Ime: %s\nPrezime: %s\nIndeks: %s\nProsjek: %.2f\nDatum rodenja:\n%2hu.%2hu.%4hu.\n", najStudent.ime, najStudent.prezime, najStudent.indeks, najStudent.prosjek, najStudent.datumRodjenja.dan, najStudent.datumRodjenja.mjesec, najStudent.datumRodjenja.godina);

    return 0;
}
```

Na primjeru koncepta struktura kao član strukture, članovima podstrukture pristupa se operatorom (.) i to tako da se spustiti se na razinu člana podstrukture.

Važno je za zapamtiti kako se aritmetičke operacije i uspoređivanje ne mogu izvršavati nad varijablom strukture, već samo nad članovima strukture.

Primjer 10: Pridruživanje jedne varijable strukture drugoj, uspoređivanje članova strukture.

```
#include<stdio.h>

struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
};

int main(void) {
    struct student najStudent;
    struct student tempStudent;

    printf("Unesite prosjek studenta\n");
    scanf("%f", &najStudent.prosjek);

    tempStudent = najStudent;
    //tempStudent == najStudent; // ilegalna radnja

    printf("Prosjek: %.2f\n", /*najStudent.prosjek*/tempStudent.prosjek);
    printf("%d\n", tempStudent.prosjek == najStudent.prosjek);

    return 0;
}
```

Polje struktura

Polje struktura predstavlja polje složenog tipa podatka, gdje je svaki element polja struktura istog složenog tipa podatka.

Primjer 11: Primjer deklariranja polja struktura tipa `struct student`.

```
struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
}/*studenti[3]*/;

int main(void) {

    struct student studenti[3];

    return 0;
}
```

Primjer 12: Primjer inicijalizacije polja struktura primjenom inicijalizatora.

```
struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};

struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
};

int main(void) {

    struct student studenti[] = {"Ivan", "Ivic", "10879", 4.56f, {5, 5, 1998},
                                "Marko", "Markic", "10880", 4.0f, {2, 3, 1998},
                                "Petar", "Peric", "10881", 3.5f, {3, 4, 1997}};
    //struct student studenti[3] = { 0 };

    return 0;
}
```

Primjer 13: Pristup elementima polja strukture radi zapisivanja i dohvaćanja sadržaja.

```
#include<stdio.h>

struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
};
struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    struct datum datumRodjenja;
};

int main(void) {
    struct student studenti[3];

    for (int i = 0; i < 3; i++)
    {
        printf("Unesite ime %d. studenta\n", i + 1);
        scanf("%19s", studenti[i].ime);
        printf("Unesite prezime %d. studenta\n", i + 1);
        scanf("%19s", studenti[i].prezime);
        printf("Unesite indeks %d. studenta\n", i + 1);
        scanf("%9s", studenti[i].indeks);
        printf("Unesite prosjek %d. studenta\n", i + 1);
        scanf("%f", &studenti[i].prosjek);
        printf("Unesite dan rođendan %d. studenta\n", i + 1);
        scanf("%hu", &studenti[i].datumRodjenja.dan);
        printf("Unesite mjesec rođendan %d. studenta\n", i + 1);
        scanf("%hu", &studenti[i].datumRodjenja.mjesec);
        printf("Unesite godinu rođendan %d. studenta\n", i + 1);
        scanf("%hu", &studenti[i].datumRodjenja.godina);

        printf("Ime %d. studenta: %s\nPrezime %d. studenta: %s\nIndeks %d.\n"
            studenta: %s\nProsjek %d. studenta: %.2f\nDatum rođenja %d. studenta:\n"
            %2hu.%2hu.%4hu.\n", i + 1, studenti[i].ime, i + 1,
            studenti[i].prezime, i + 1, studenti[i].indeks, i + 1,
            studenti[i].prosjek, i + 1, studenti[i].datumRodjenja.dan,
            studenti[i].datumRodjenja.mjesec, studenti[i].datumRodjenja.godina);

    }

    return 0;
}
```

Ključna riječ *typedef*

C programski jezik omogućuje primjenu ključne riječi *typedef* za promjenu imena određenom ugrađenom ili složenom tipu podatka. Ključna riječi *typedef* postaje vrlo praktična prilikom primjene sa strukturama jer se strukturi daje novo simboličko ime kao tipu podatka i time sintaksa postaje jasnije prilikom deklariranja varijable strukture.

Primjer 14: Primjer primjene ključne riječi *typedef*.

```
typedef struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
}DATUM;

typedef struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    DATUM datumRodjenja;
}STUDENT;
```

Strukturama se promijenilo ime iz `struct datum` u `DATUM` i iz `struct student` u `STUDENT`. Prema konvencija kada se s *typedef* promijeni ime određenom tipu podatka, savjetuje se pisanje novog imena tipa podatka velikim slovima kako bi se lakše razlikovalo od ugrađenih tipova podataka.

Primjer 15: Primjer kreiranja varijable strukture s novim imenom složenog tipa podatka.

```
typedef struct datum {
    unsigned short dan;
    unsigned short mjesec;
    unsigned short godina;
}DATUM;

typedef struct student {
    char ime[20];
    char prezime[20];
    char indeks[10];
    float prosjek;
    DATUM datumRodjenja;
}STUDENT;

int main(void) {
    STUDENT najStudent;

    return 0;
}
```

Zadaci

1. Napisati C program u kojemu treba napraviti novi složeni tip podatka *struct kompleksni* pod nazivom *KOMPLEKSNI* i kreirati strukturne varijable *br1*, *br2*, *rez*. Struktura *KOMPLEKSNI* treba sadržavati sljedeće realne članove: *re* i *im*. Omogućiti unos dva kompleksna broja. Izračunati i ispisati zbroj, razliku, produkt, količnik dva kompleksna broja u obliku *re + im*i*, pripaziti na predznak ako je imaginarna komponenta negativna, te izračunati modul svakog kompleksnog broja.

Dodatak:

Opći oblik kompleksnog broja:

$$z = a + b \cdot i$$

Zbrajanje i oduzimanje:

$$(a + bi) \pm (c + di) = (a \pm c) + (b \pm d) \cdot i$$

Množenje:

$$(a + bi) \cdot (c + di) = (a \cdot c - b \cdot d) + (a \cdot d + b \cdot c) \cdot i$$

Dijeljenje:

$$\frac{a + bi}{c + di} = \frac{a + bi}{c + di} \cdot \frac{c - di}{c - di} = \frac{(ac + bd) + (bc - ad) \cdot i}{c^2 + d^2}$$

Modul:

$$z = \sqrt{a^2 + b^2}$$

2. Napisati C program koji omogućuje unos podataka o programerima jedne IT tvrtke, kao što je ime, prezime, OIB, plaća i datum rođenja, od ukupno 15 programera. Tražiti od korisnika unos broj *n* ($1 \leq n < 16$), ponavljati sve dok unos ne odgovara traženom intervalu. Popuniti informacije o onoliko programera ovisno o broju *n*, pronaći programera koji ima najbolju plaću, te ispisati sve informacije o tome programeru. S obzirom na navedene podatke o programeru, formirati strukturu koristeći ključnu riječ *typedef* i pripaziti na ispis formata za datum kako bi bio u obliku DD.MM.GGGG.
3. Napisati program u C jeziku u kojem se treba kreirati novi složeni tip podatka *struct cijeli_broj* pod nazivom *CIJELI_BROJ*. Struktura *CIJELI_BROJ* treba sadržavati dva cijela člana tipa *long* za *uneseni_broj* i *int* za *broj_znamenaka*. U član *uneseni_broj* potrebno je upisati generirani P-S broj iz intervala $[-2147483647, 2147483647]$, a u član *broj_znamenaka* potrebno je upisati koliko generirani broj ima znamenaka. Za generiranje P-S vrijednosti potrebno je koristiti funkciju koja ne prima argumente, a vraća P-S generiranu vrijednost, a za izračunavanje broja znamenaka potrebno je također koristiti funkciju koja prima P-S generirani broj te izračunava i vraća broj znamenaka. U glavnom dijelu programa potrebno je kreirati polje struktura *CIJELI_BROJ* od 40 elemenata, te pomoću ranije spomenutih funkcija popuniti članove pojedine strukture iz polja struktura. Pronaći i ispisati sve članove strukture koja ima najveći, kao i strukturu koja ima najmanji broj znamenaka generiranog broja.
4. Deklarirati strukturu stan koja ima članove: površina, cijena, adresa, energetska razred, klima uređaj [1/0]. Omogućiti korisniku unos podataka za željeni broj stanova (max. 20) u polje struktura. Ponavljati unos dokle god je željeni broj manji od 2, a veći od 20. Ispisati podatke o svim unesenim stanovima kod kojih je cijena po kvadratu manja od 900 eura i sve stanove kojima je cijena veća od 1100 eura po kvadratu, a nemaju klima uređaj. Ako nema takvih stanova ispisati tekst: „Niti jedan stan ne odgovara kriterijima.“, a ako ima takvih stanova ispisati ih u formatu „adresa, cijena“ za svaki takav stan u novi red.