

# Strukture Klasse i OOP

Strukture u C++-u

# Podatkovni tip - Strukture

- **Derivirani - izvedeni podatkovni tip** – stvara ga programer
- grupa logički povezanih podataka **različitih** podatkovnih tipova (npr. `char`, `int`, `double`) sa zajedničkim imenom
- podaci (članovi strukture) su međusobno povezani identifikatorom – imenom strukture; deklaracijom članova strukture određuje se ime i tip pojedinih članova strukture
- ako je definiran izvan tijela funkcije `main()` – sve funkcije programa mogu koristiti taj tip
- između imena strukturne varijable i imena člana te strukture stavlja se operator `'.'`



# Strukture i njihovo korištenje

- struktura je korisnički definiran složeni tip
- sastavljen je od polja ili članova koji mogu biti različitih tipova
- u C++, struktura je ista kao i klasa osim što su njeni članovi podrazumijevano javni
- varijable se kod definicije strukturnog tipa može deklarirati nizanjem jednog ili više imena varijabli odvojenih zarezima između zatvorene zagrade i ;

# Podatkovni tip stuktore

- **Derivirani - izvedeni podatkovni tip** – stvara ga programer
- grupa logički povezanih podataka **različitih** podatkovnih tipova (npr. `char`, `int`, `double`) sa zajedničkim imenom
- podaci (članovi strukture) su međusobno povezani identifikatorom – imenom strukture; deklaracijom članova strukture određuje se ime i tip pojedinih članova strukture
- ako je definiran izvan tijela funkcije `main()` – sve funkcije programa mogu koristiti taj tip
- između imena strukturne varijable i imena člana te strukture stavlja se operator `'.'`



# Strukture i njihovo korištenje

- struktura je korisnički definiran složeni tip
- sastavljen je od polja ili članova koji mogu biti različitih tipova
- u C++, struktura je ista kao i klasa osim što su njeni članovi podrazumijevano javni
- varijable se kod definicije strukturnog tipa može deklarirati nizanjem jednog ili više imena varijabli odvojenih zarezima između zatvorene zagrade i ;

# Srukture - Primjeri

```
struct OSOBA                // deklaracija strukture OSOBA
{int starost;              // deklaracije tipova članova
float tezina;
char ime[25]; } clan_obitelji; // definira objekt tipa OSOBA

OSOBA brat;
OSOBA sestra;              // C++ stil deklaracije strukture
sestra.starost = 13;
brat.starost = 7;          // pridružuje vrijednosti članovima

struct TOCKA                // deklaracija strukture TOCKA
{ int x;                   // definira članove x i y
  int y; } mjesto = { 20, 40 }; // varijabla mjesto ima vrijednosti x = 20, y = 40

TOCKA tamo;                // varijabla tamo je tipa TOCKA
```

# Vježba

- Deklarirati strukturu “sportski\_klub” koja će moći pamtit i podatke za neki sportski klub, i to:
  - Naziv kluba (polje znakova ili string)
  - Grad iz kojeg je klub
  - Ime i prezime trenera
  - Broj igrača
  - Broj postignutih pogodaka
  - Broj primljenih pogodaka
  - Da li je član prve lige
  - Iznos godišnjeg “proračuna”
- Deklarirati dvije varijable tog tipa i pridružiti im vrijednosti



# Rješenje

```
struct KLUB
{
    string NAZIV_KLUBA;
    string GRAD;
    char ime[100];
    int BROJ_IGRACA;
    int BROJ_POSTIGNUTIH_GOLOVA;
    int BROJ_PRIMLJENIH_GOLOVA;
    bool PRVA_LIGA;
    float GODISNJI_PRORACUN;
} klub1, klub2;

KLUB klubx;

klub1.NAZIV_KLUBA="PIPO IPC";
klub1.GRAD="ČAKOVEC";
klub1.PRVA_LIGA=true;

klub2.NAZIV_KLUBA="CROATIA ZAGREB";
klub2.GRAD="ZAGREB";
```

# struct – definiranje, deklariranje, variable

```
struct Tekucina
{
    public:                                (ili private)
        char oznaka;
        double temp, gustoca;
};

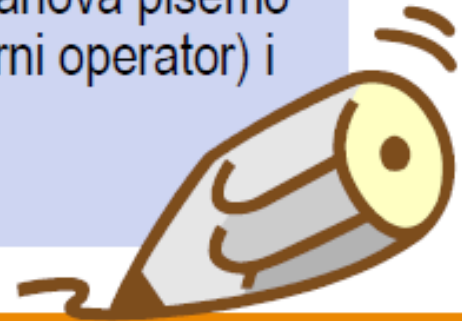
int main()
{
    Tekucina voda, ulje, mineralna;
    voda.oznaka = 'V';
    voda.temp = 28.6;
    voda.gustoca = 9.81;

    mineralna = voda;

    cout << "voda: " << voda.oznaka << endl; ...}
```

# Deklariranja struct tipa i struct varijabli

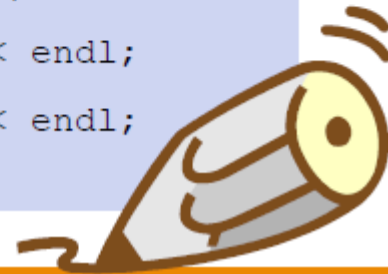
- deklaracijom određujemo koji će se tipovi informacija moći spremiti u memoriju rezerviranu za **strukturnu varijablu** tog tipa (za sve podatkovne članove te strukture)
- strukturna varijabla je složena varijabla koja sadrži niz vrijednosti istog ili različitog tipa
- deklaracijom **strukture** određuje se **oblik strukture** i ne zauzima se memorijski prostor; deklaracijom **strukturne varijable** rezervira se memorijski prostor
- za spremanje vrijednosti podatkovnih (public) članova pišemo ime `struct` varijable (objekta), točku (strukturni operator) i ime podatkovnog člana



# Pridruživanje i ispis vrijedosti podatkovnih članova

- vrijednosti **svih članova** jedne `struct` varijable možemo pridružiti članovima druge istovrsne `struct` varijable jednom naredbom pridruživanja
- za ispis vrijednosti **pojedinih** podatkovnih članova `struct` varijable (kao i za sve ostale operacije nad strukturama) moramo navoditi **svaki član pojedinačno** upotrebom operatora `(.)`, npr.:

- ```
cout << "voda: " << voda.oznaka << endl;
```
- ```
cout << "temperatura: " << voda.temp << endl;
```
- ```
cout << "gustoca: " << voda.gustoca << endl;
```



# Usporedba struktura i razreda

- u C++ strukture imaju ista svojstva kao razred
- način deklaracije je isti s tom razlikom što se koristi ključna riječ `struct`
- struktura može sadržavati podatkovne i funkcijske članove, konstruktore, destruktore i ključne riječi za dodjelu prava pristupa
- razlika:
  - ako se ne navede pravo pristupa, članovi imaju podrazumijevani javni pristup
  - strukture ne omogućuju nasljeđivanje
- nakon što se usvoji koncept razreda, strukture se obično izbacuju iz upotrebe



# Radni zadatak

- Deklarirati strukturu Razred koja će pamtit i podatke o razredu. Podatke koje vi mislite da treba pamtit i
- Sličan primjer - Struktura Klub
- Napišite kod, pokrenite ga, upišite i ispišite podatke.

# Dodatno

- **Napravite program koji će iz datoteke učitavati imena i prezimena učenika te njihove ocjene iz pismenog ispita. Program treba ispisati koliko učenika je dobilo odličan, koliko vrlo dobar, dobar, dovoljan i nedovoljan te kolika je prosječna ocjena razreda iz pismenog. Ispisati imena i prezimena učenika koji su dobili odličan.**