

Inicijalizacija vrednošću

❖ Zato je počev od verzije C++11 u jezik uvedena *inicijalizacija vrednošću* (*value initialization*), čija je ideja da vrši podrazumevanu inicijalizaciju za objekte klase, a inicijalizaciju nulom za objekte neklasnih tipova, i to u svim kontekstima na uniforman način:

- kada se inicijalizuje bezimeni privremeni objekat sa praznim zagradama (običnim ili velikim):

`T()`

`T{}`

- kada se inicijalizuje dinamički objekat u izrazu *new* sa praznim zagradama (običnim ili velikim):

`new T()`

`new T{}`

- kada se inicijalizuje nestatički podatak član ili osnovna klasa u inicijalizatoru članova u konstruktoru klase sa praznim zagradama (običnim ili velikim):

`X::X (...) : T(), t() {...}`

`X::X (...) : T{}, t{} {...}`

- kada se inicijalizuje imenovana varijabla sa praznim velikim zagradama:

`T t{};`

Inicijalizacija vrednošću

❖ Pritom:

- u svim navedenim slučajevima kada se koriste velike zagrade {}, ako je T agregatni tip (niz ili klasa pod nekim uslovima, tipično struktura), vrši se agregatna inicijalizacija umesto inicijalizacije vrednošću
- ako je T klasa koja nema podrazumevani konstruktor, ali ima konstruktor koji prihvata argument tipa `std::initializer_list`, vrši se inicijalizacija listom umesto inicijalizacije vrednošću

❖ Inicijalizacija vrednošću vrši sledeće:

- Ako je T klasa bez podrazumevanog konstruktora, ili sa eksplicitno korisnički definisanim ili obrisanim podrazumevanim konstruktorom, objekat se inicijalizuje podrazumevanom inicijalizacijom (koja nije dozvoljena ako podrazumevani konstruktor ne postoji ili je obrisani)
- Ako je T klasa sa podrazumevanim konstruktorom koji nije ni eksplicitno korisnički definisan niti obrisani, objekat se najpre inicijalizuje nulom, a potom podrazumevanom inicijalizacijom ako ima netrivialan podrazumevani konstruktor
- Ako je T niz, svaki element se inicijalizuje vrednošću
- Inače, objekat se inicijalizuje nulom

❖ Uprošćeno, ako je korisnik sam napravio podrazumevani konstruktor, onda ova inicijalizacija poziva taj konstruktor i radi podrazumevanu inicijalizaciju definisanu u njemu; u suprotnom, inicijalizuje objekat nulom, a onda poziva podrazumevane konstruktore podobjekata. Na primer:

```
struct X {  
    int i;  
    X () {}  
};  
  
struct Y {  
    int j;  
    X x;  
};  
  
cout<<X().i<<endl<<Y().j<<endl<<Y().x.i;
```