

# Konstantna inicijalizacija

- ❖ Konstantna inicijalizacija se vrši za varijable sa statičkim trajanjem skladištenja i onima vezanim za niti (*thread local*), samo pod uslovom da su inicijalizovani konstantnim izrazom
- ❖ *Konstantan izraz (constant expression)* je izraz koji se može izračunati za vreme prevođenja i može se koristiti gde god je potreban konstantan izraz (npr. za dimenzije nizova ili inicijalizaciju konstanti)
- ❖ Ovakvi su izrazi koji uključuju npr. samo literale, konstante primitivnih tipova i operatore koji se mogu izračunati za vreme prevođenja
- ❖ Međutim, od verzije C++11, u konstantnim izrazima mogu učestvovati i varijable, uključujući i objekte klasa, pa čak i pozivi korisničkih funkcija, uključujući i pozive funkcija članica klasa, pod uslovom da su te funkcije deklarisanе kao *constexpr*, i da se te funkcije, uključujući i konstruktore, mogu izvršiti za vreme prevođenja, što znači da su im argumenti konstantni izrazi, da su inicijalizatori svih podobjekata konstantni izrazi, da su svi podaci članovi inicijalizovani i slično
- ❖ Ovakve klase i funkcije moraju zadovoljiti niz definisanih uslova, pri čemu su se ti uslovi menjali kroz verzije jezika, i dalje se menjaju u novijim verzijama jezika u pravcu relaksacije ograničenja; na primer, od verzije C++20, ove funkcije mogu da budu čak i virtuelne
- ❖ Jedno od osnovnih ograničenja jeste to da se te funkcije ne mogu oslanjati na pozive funkcija koje nisu označene kao *constexpr* ili vrednosti varijabli koje nisu označene kao *constexpr* ili podataka članova koji nemaju inicijalizatore
- ❖ Funkcije koje su označene kao *constexpr*, uključujući i konstruktore, mogu se pozivati i van konstantnih izraza, odnosno za vreme izvršavanja
- ❖ Objekti deklarisanі kao *constexpr* su implicitno konstantni i moraju biti inicijalizovani konstantnim izrazima; funkcije koje su označene kao *constexpr* su implicitno *inline*

---

# Konstantna inicijalizacija

---

- ❖ Ideja ove tehnike je zapravo to da se čitav deo programa, uključujući definicije klasa, kreiranje njihovih objekata i pozive funkcija, a ne samo izračunavanje prostih izraza, izvršava za vreme prevođenja, kako bi se proizveo odgovarajući željeni rezultat
- ❖ Ova tehnika može se koristiti na primer za:
  - prekonfigurisanje, odnosno inicijalizaciju složenijih struktura podataka ili objekata i njihovih veza koja se može izvršiti za vreme prevođenja, a koje se onda koriste u izvršavanju programa
  - složenija izračunavanja parametara programa

koja se mogu izvršiti za vreme prevođenja, kako se ne bi trošilo vreme i zauzimao memorijski prostor prilikom svakog izvršavanja programa, tipično prilikom pokretanja programa

- ❖ Ovo može da bude značajno na primer za sistemski, ugrađeni (*embedded*) softver, posebno za onaj za koji se zahteva mali memorijski otisak (*memory footprint*), brzo pokretanje i izvršavanje, ili što manja potrošnja energije pri izvršavanju