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

Konstantna inicijalizacija

- Na pimer, *enumeratorske klase*, tj. klase sa konstantnim skupom instanci predstavljaju uopštenje pojma enumeracije, jer su tipovi čiji skupovi instanci ne mogu da se menjaju tokom izvršavanja programa, ali su, za razliku od enumeracija čije su instance proste, skalarne simboličke vrednosti, instance ovakvih klasa objekti sa svim svojim svojstvima
- Na primer, želimo da napravimo predefinisan, prekonfigurisan skup korisničkih uloga u programu koje se inicijalizuju statički, u toku prevođenja, pri čemu su te uloge predstavljene objektima koji imaju odgovarajuće usluge (npr. proveru da li korisnik sa tom ulogom može da izvrši datu komandu):

```
class UserRole {
public:
  constexpr UserRole (const char* name, …);
  const char* name = nullptr;

  bool isAuthorizedFor (Command* cmd) const;
  …
};

constexpr UserRole ordinary("Ordinary user", …);
constexpr UserRole privileged("Privileged user", …);
constexpr UserRole admin("Administrator", …);
```