Statički životni vek

- Varijable sa statičkim životnim vekom su sledeće:
  - varijable deklarisane u oblasti prostora imena (*namespace*), uključujući i globalni prostor
  - ostale varijable (deklarisane u oblasti bloka ili klase) koje su deklarisane kao *static* ili *extern*

osim ako imaju specifikator *thread_local*

```
namespace N {
  int i;

  struct S {
    static int i;
  };

  void f () {
    static int i;
  }
}
```

Statički objekat u oblasti prostora imena

Statički objekat u oblasti klase (statički podatak član)

Statički lokalni objekat

- Za svaku definiciju ovakvog objekta postoji jedna instanca za vreme izvršavanja; zato prevodilac može (i po pravilu to i radi) alocirati prostor za takve objekte u prevedenom fajlu koji predstavlja memorijsku mapu programa prilikom pokretanja. U svakom slučaju, memorijski prostor za ovakve objekte alocira se na početku izvršavanja programa i dealocira na kraju izvršavanja programa

Statički životni vek

- Inicijalizacija statičkih varijabli, odnosno početak njihovog životnog veka razlikuje se za lokalne i ostale statičke varijable
- Statičke varijable koje nisu lokalne (tj. one iz prostora imena ili statički podaci članovi) inicijalizuju se na sledeći način:
  - najpre se vrši *statička inicijalizacija* (*static initialization*), što znači sledeće:
    - *konstantna inicijalizacija* (*constant initialization*) onih varijabli koje su deklarisane kao *constexpr* ili koje su inicijalizovane konstantnim izrazom; po pravilu, ovo prevodilac vrši još za vreme prevođenja (u prevedeni kod upiše vrednosti tih varijabli u statički alociran prostor za njih); čak i ako to ne uradi za vreme prevođenja, mora da obezbedi da se to radi za vreme izvršavanja pre svega ostalog
    - *inicijalizacija nulom* (*zero initialization*): za sve ostale situacije, objekti se inicijalizuju tako da dobiju binarnu vrednost 0, sa značenjem očekivano konvertovane vrednosti (npr. pokazivač će imati *null* vrednost čak i ako se ona ne implementira binarnim nulama)
  - *dinamička inicijalizacija* (*dynamic initialization*) podrazumeva izračunavanje inicijalizatora (kao izraza) za vreme izvršavanja programa, kao i poziv konstruktora objekta klase koji se inicijalizuje, odnosno upis izračunate vrednosti u varijablu koja nije klasnog tipa; ukoliko može, prevodilac dinamičku inicijalizaciju takođe može uraditi za vreme prevođenja