

Automatski životni vek

- ❖ Kada izvršavanje napušta blok na bilo koji način (prolaskom kroz kraj bloka, naredbom *return* ili zbog bačenog izuzetka), propisno se uništavaju svi automatski objekti tog bloka koji su kreirani, ali i samo oni: ako neki objekat nije kreiran (recimo zato što je pre izvršavanja njegove definicije bačen izuzetak), objekat neće biti ni uništen
- ❖ Ovo važi i za napuštanje bloka zbog podignutog izuzetka: svi objekti koji su kreirani, a čiji se blokovi napuštaju do ulaska u odgovarajući *catch* blok, propisno se uništavaju (pozivom destruktora); ovo obuhvata i okružujuće blokove, odnosno blokove funkcija koje su pozvane, a nisu završene
- ❖ Analogno važi i za objekte koji su samo delimično kreirani, jer je izuzetak podignut tokom njihove inicijalizacije (poziva konstruktora): svi njihovi podobjekti osnovnih klasa i članovi koji su kreirani biće propisno uništeni, a oni koji nisu, neće
- ❖ Ovaj postupak naziva se *razmotavanje steka* (*stack unwinding*):

```
void f () {  
    try {  
        X x1;  
        g();  
    }  
    catch (...) {}  
}
```

Ulaskom u ovaj *catch* zbog izuzetka bačenog u funkciji *h* biće uništeni objekti *x3*, *x2* i *x1*

```
void g () {  
    X x2;  
    h();  
}
```

```
void h () {  
    X x3;  
    throw 0;  
}
```

- ❖ Automatski objekti se uništavaju po redusledu uvek tačno obrnutom od onog kojim su kreirani

Statički životni vek

❖ Varijable sa statičkim životnim vekom su sledeće:

- varijable deklarisanе u oblasti prostora imena (*namespace*), uključujući i globalni prostor
- ostale varijable (deklarisanе u oblasti bloka ili klase) koje su deklarisanе kao *static* ili *extern*

osim ako imaju specifikator *thread_local*

```
namespace N {  
    int i;  
  
    struct S {  
        static int i;  
    };  
  
    void f () {  
        static int i;  
    }  
}
```

❖ 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