Životni vek vezan za nit

- Varijable koje bi inače bile statičke po životnom veku (lokalne, u prostoru imena ili u oblasti klase) mogu biti deklarisane kao *thread_local*; tada imaju životni vek vezan za nit
- Ovakve varijable nastaju pri kreiranju svake nove niti i nestaju kada se nit završi
- Svaka nit ima svoju instancu ove varijable, nezavisnu od ostalih niti; svako obraćanje toj varijabli odnosi se na onu instancu koja pripada niti u čijem kontekstu se izvršava taj pristup. Na primer:

```
thread_local int i=0;
void f (int ii) {
  i = ii;
}

void tf (int id) {
  f(id);
  std::cout<<++i<<std::endl;
}

int main () {
  i = 10;
  std::thread t1(tf,1);
  std::thread t2(tf,2);
  std::thread t3(tf,3);

  std::cout<<i<<std::endl;

}
```

Životni vek vezan za nit

Odnosi se na instancu *i* koja pripada niti u čijem kontekstu se ovo izvršava

Odnosi se na instancu *i* koja pripada niti u čijem kontekstu se ovo izvršava

Funkcija *main* se izvršava u kontekstu početne, "glavne" niti programa

Odnosi se na instancu *i* koja pripada niti u čijem kontekstu se ovo izvršava

Privremeni objekti

- Privremeni objekti su anonimni (bezimeni) objekti klasa čiji je životni vek najčešće kratak; oni nastaju implicitno, u određenim situacijama, i nestaju implicitno, pod kontrolom prevodioca
- Osnovna potreba za privremenim objektom jeste u tome da se rezultat nekog izraza, uključujući i poziv funkcije (i operatorske funkcije) može dalje iskoristiti kao operand neke druge operacije (ili argument funkcije)
- Bez obzira na to kada se tačno privremeni objekat inicijalizuje i uništava, uvek se pri inicijalizaciji poziva njegov konstruktor, a pri uništavanju njegov destruktor
- Pre verzije jezika C++17, rezultat poziva funkcije (uključujući i operatorske) koja ne vraća referencu (na lvrednost), odnosno čiji je povratni tip klasa (a ne referenca na nju), bio je uvek privremeni objekat: u trenutku povratka iz funkcije, na mestu poziva funkcije, pravi se privremeni objekat koji se inicijalizuje izrazom iza naredbe *return*:

```
class X {...};

X f (X x) {
  return x;
}

int main () {
  X x1;

  ...f(x1)...

  ...f( f(x1) )...
}
```