Podrazumevana inicijalizacija

- Prema tome, automatski i dinamički objekti neklasnih tipova imaju nedefinisane podrazumevane vrednosti; statički i oni vezani za nit se inicijalizuju nulom. Reference se ne mogu podrazumevano inicijalizovati
- Na primer:

```
struct X {
  int m;
  X () {}
};
int i;
int main () {
  int j;
  X x;
}
```

m nije pomenut u listi inicijalizatora članova, pa se podrazumevano inicijalizuje (nema akcije jer je tipa *int*)

Inicijalizuje se najpre nulama, a onda podrazumevano (ništa), pa ostaje nula

Podrazumevana inicijalizacija bez efekta, ima neodređenu vrednost

Podrazumevana inicijalizacija, poziva se podrazumevani konstruktor, *x.m* ima neodređenu vrednost

- Motivacija za ovakvo ponašanje je efikasnost: u nekim situacijama nije neophodna nikakva određena inicijalna vrednost, pa nema potrebe gubiti vreme na inicijalizaciju (cilj je da prevedeni kod bude jednako efikasan kao da je pisan):

```
int m;
cin>>m;
```


Copyright 2018-2024 by Dragan Milićev

Inicijalizacija vrednošću

- Kao što je pokazano, podrazumevana inizijalizacija u nekim situacijama poziva podrazumevane konstruktore, dok za automatske i dinamičke objekte neklasnih tipova ne radi nikakvu inicijalizaciju, ostavljajući ih sa nedefinisanim vrednostima. U mnogim situacijama potrebno je da takvi objekti budu definisani određenom vrednošću, odnosno inicijalizovani nulom
- U verziji jezika C++03, u kontekstima gde je to bilo moguće, ovo se postizalo inicijalizatorom sa praznim zagradama:

```
C::C () : t() {}
T* p = new T();
```

- Međutim, u mnogim kontekstima to nije moguće, jer takva notacija znači deklaraciju funkcije bez parametara koja vraća tip *T*, a ne objekta tipa *T:*

```
T t();
```

- U takvim situacijama morala se koristiti drugačija inicijalizacija, ali ona ima i drugu semantiku (inicijalizacije kopiranjem ili agregatne inicijalizacije):

```
T t = T();
T t = {};
```

- Ili nešto drugo, u zavisnosti od tipa, ali problem ostaje kada je tip nepoznat, kao što je slučaj sa šablonima:

```
int i = 0;
Clock clk;
int a[10]{}
```


Copyright 2018-2024 by Dragan Milićev