Konstruktor

- U definiciji konstruktora klase, pre njegovog tela, a iza liste parametara i znaka :, može se navesti *lista inicijalizatora članova* (*member initializer list*), u kojoj se navode inicijalizacije objekata članova i podobjekata osnovnih klasa razdvojene zarezima. Ove inicijalizacije završavaju se pre ulaska u telo konstruktora klase
- Ako neki podobjekat nema podrazumevanu inicijalizaciju, ovde se mora navesti njegova inicijalizacija, u suprotnom, prevodilac će prijaviti grešku. Na primer:

```
struct Base {
  Base (int, int);
};
```

Klasa *Base* nema podrazumevanu inicijalizaciju

```
struct Derived : Base {
  Base b;
  int& r;
  Derived ();
};

Derived::Derived () {}
```

Greška u prevođenju: podobjekat osnovne klase *Base*, objekat član *b*, kao i referenca *r* nemaju podrazumevanu inicijalizaciju

- Ako nestatički podatak član ima podrazumevani inicijalizator u definiciji klase, a naveden je u ovoj listi, biće inicijalizovan kao što piše u toj listi u konstruktoru, a podrazumevana inicijalizacija biće ignorisana:

```
struct X {
  int i = 1;
  X ();
};

X::X () : i(2) {}
```

Podrazumevana inicijalizacija podatka člana

Objekat član *i* imaće vrednost 2

- Izuzeci tokom ove incijalizacije mogu se hvatati *try-catch* konstruktom na nivou cele funkcije

Copyright 2018-2024 by Dragan Milićev

Konstruktor

- Lista inicijalizatora članova u konstruktoru klase može imati i samo jednu inicijalizaciju (i ništa više osim nje), koja navodi ime te iste klase i inicijalizator (listu argumenata u zagradi). Tada se radi o *delegiranju* inicijalizacije: posmatrani konstruktor je *delegirajući* (*delegating constructor*), a onaj određen inicijalizacijom u listi je *ciljni konstruktor* (*target constructor*)
- Na primer:

```
class Matrix {
public:

  Matrix (int m, int n);
  Matrix (int m, int n, long copyFrom[]);
  …

private:

  long (*mat)[N][N];
  int m, n;

};

Matrix::Matrix (int mm, int nn) : m(mm), n(nn) {
  if (m>N || n>N) throw MatrixTooLarge();
  mat = new long[m][N];
}

Matrix::Matrix (int m, int n, long a[]) : Matrix(m,n) {
  for (int i=0; i<m; i++)
    …
}
```

- Prilikom inicijalizacije, najpre se izvršava ciljni konstruktor, određen na osnovu uobičajenih pravila odabira konstruktora prema stvarnim argumentima, a onda se izvršava telo delegirajućeg konstruktora
- Na ovaj način se pravilna algoritamska dekompozicija može implementirati nešto kompaktnije, jer se zajednički deo dva konstruktora ne mora izdvajati u posebnu pomoćnu operaciju koja se poziva iz oba konstruktora, već se poziv onog prostijeg može “ugraditi” u onaj složeniji

Copyright 2018-2024 by Dragan Milićev