

# Konstruktor kopije

- ❖ Konstruktor klase  $X$  čiji je prvi parametar tipa  $X\&$ ,  $\text{const } X\&$ ,  $\text{volatile } X\&$  ili  $\text{const volatile } X\&$ , a koji ili nema druge parametre, ili svi ostali parametri imaju podrazumevane vrednosti, naziva se *konstruktor kopije* (*copy constructor*)
- ❖ Ovaj konstruktor poziva se kada se objekat klase  $X$  inicijalizuje objektom istog tipa (osim ako postupak odabira konstruktora ne odabere neki drugi konstruktor koji više odgovara pozivu), kao što su sledeće situacije:
  - inicijalizacija objekta:  $X\ x1 = x2$  ili  $X\ x1(x2)$ , gde je  $x2$  objekat klase  $X$  ili iz nje izvedene klase
  - prenos argumenta pri pozivu funkcije  $f(x)$  koja ima taj parametar tipa  $X$ , gde je  $x$  objekat klase  $X$  ili iz nje izvedene klase
  - povratak vrednosti iz funkcije  $f$  koja ima povratni tip  $X$  naredbom  $\text{return } x$ , gde je  $x$  objekat klase  $X$  ili iz nje izvedene klase
- ❖ Ako se objekat izvedene klase inicijalizuje objektom osnovne klase, i ako se odabere ovaj konstruktor, kao i na svim drugim mestima, referenca na osnovnu kalsu inicijalizuje se referencom na podobjekat osnovne klase unutar objekta izvedene klase:

```
struct Base {  
    Base (const Base&);  
    ...  
};
```

Konstruktor kopije

```
struct Derived : Base {  
    Derived ();  
    ...  
};
```

```
Derived d;  
Base b(d);
```

Ovde se poziva konstruktor kopije osnovne klase  $\text{Base::Base}(\text{const Base}\&)$ , čiji se parametar - referenca vezuje za podobjekat osnovne klase unutar objekta  $d$

# Konstruktor kopije

- ❖ Ako klasa *X* nema nijedan eksplicitno deklarisan konstruktor kopije (tj. korisnički deklarisan konstruktor kopije), prevodilac će implicitno deklarirati konstruktor kopije koji je *javan*, *inline*, nije *explicit*, i za koji važi:
  - ako svaka direktna i virtuelna osnovna klasa *B* ima konstruktor kopije koji ima parametar tipa *const B&* ili *const volatile B&*, i ako svaka klasa *M* podatka člana ima konstruktor kopije koji prima parametar tipa *const M&* ili *const volatile M&*, onda i ovaj implicitno generisani konstruktor kopije ima parametar tipa *const X&*
  - u suprotnom, ovaj konstruktor kopije ima parametar tipa *X&*
- ❖ Klasa može imati i više konstruktora kopije, recimo onaj koji prima *X&* i onaj koji prima *const X&*
- ❖ Ako klasa ima neki konstruktor kopije (pa prevodilac ne generiše implicitni konstruktor kopije), programer ipak može forsirati automatsko deklarisanje konstruktora koji bi prevodilac implicitno deklarirao specifikatorom *=default*:

```
struct X {  
    X (X&);  
    X (const X&) = default;  
};
```

- ❖ Ako podobjekti osnovnih klasa i objekti članovi ne mogu da se kopiraju, recimo zato što su objekti klasa koje nemaju dostupne konstruktore kopije, ili ako klasa ima korisnički, eksplicitno deklarisan konstruktor premeštanja (*move constructor*) ili operator dodele premeštanjem (*move assignment operator*), onda će ovaj implicitno deklarirani konstruktor kopije biti obrisani (smatraće se da njegov poziv nije dozvoljen, iako je on deklarisan)
- ❖ U suprotnom, ako ovaj implicitno deklarisan ili podrazumevani konstruktor kopije nije obrisani, on će biti definisan i vršiće podrazumevano kopiranje podobjekata osnovnih klasa i objekata članova, po istom redosledu kao i u inicijalizaciji; ako su ti podobjekti objekti nekih klasa, pozivaju se njihovi konstruktori kopije, u suprotnom, vrši se prosto kopiranje vrednosti