

Ugrađeni objekti

- ❖ Inicijalizatori za podobjekte osnovnih klasa i za objekte članove navode se u zaglavlju konstruktora (izvedene) klase; svako pominjanje podatka člana u telu konstruktora klase više nije inicijalizacija, nego neka druga operacija nad već inicijalizovanim objektom članom:

```
class Whole : public Entity {  
public:  
    Whole ();  
    ~Whole ();  
private:  
    Part part;  
};
```

Destruktor

Ovo je inicijalizacija
podobjekta osnovne klase

Ovo je inicijalizacija
podobjekta člana

```
Whole::Whole () : Entity(), part(this) {  
    ...part...  
}
```

Ovde više ne može da se izvede inicijalizacija podobjekata

```
Whole::~~Whole () {...}
```

Ovde se implicitno pozivaju destruktori člana *part* i podobjekta osnovne klase

- ❖ Dakle, u telu konstruktora klase ne može se inicijalizovati objekat član, već mu se eventualno može dodeliti vrednost operatorom dodele, što je potpuno druga operacija od inicijalizacije (iako za neklasne tipove ima isti efekat - prosto kopiranje vrednosti); naravno, ako je nestatički podatak član tipa reference ili konstatnog tipa, on se mora inicijalizovati, svakako mu se ne može dodeliti vrednost:

```
struct X {  
    X (int&);  
    int& r;  
    const int c;  
};
```

```
X::X (int& i) : r(i), c(i) {}
```

Ugrađeni objekti

- ❖ Kada se inicijalizuje objekat klase, redosled inicijalizacije je sledeći:
 - ako se radi o najizvedenijem objektu (tj. ovo je “prva” inicijalizacija objekta u hijerarhiji nasleđivanja), inicijalizuju se podobjekti virtuelnih osnovnih klasa, po redosledu obilaska grafa nasleđivanja po dubini, sleva nadesno (po redosledu kako su te osnovne klase navedene u deklaracijama izvedenih klasa)
 - inicijalizuju se podobjekti direktnih osnovnih klasa (iz koje je ova klasa neposredno izvedena), po redosledu njihovog navođenja u deklaraciji izvedene klase
 - inicijalizuju se objekti članovi
 - izvršava se telo konstruktora klase čiji je objekat direktna instanca
- ❖ Treba primetiti da svako pominjanje “inicijalizacije (pod)objekta” u navedenim pravilima znači rekurzivnu inicijalizaciju podobjekata osnovnih klasa, objekata članova i izvršavanje tela konstruktora za taj (pod)objekat
- ❖ Redosled uništavanja (destrukcije) objekta uvek je tačno obrnut:
 - izvršava se telo destruktora klase čiji je objekat direktna instanca
 - uništavaju se objekti članovi, po redosledu obrnutom od redosleda inicijalizacije
 - uništavaju se podobjekti direktnih osnovnih klasa, po redosledu obrnutom od redosleda inicijalizacije
 - ako se radi o najizvedenijem objektu, uništavaju se podobjekti virtuelnih osnovnih klasa, po redosledu obrnutom od redosleda inicijalizacije