Ugrađeni objekti

- Objekti ugrađeni u druge objekte složenih tipova nazivaju se *podobjektima*; to su sledeći:
  - podobjekat osnovne klase unutar objekta izvedene klase
  - objekat koji je manifestacija nestatičkog podatka člana, tzv. objekat član (*member object*), ugrađen u objekat klase čiji je taj podatak član
  - element niza
- Objekti koji nisu podobjekti drugih objekata nazivaju se *kompletnim objektima* (*complete object*)
- Kompletni objekti, objekti članovi i objekti koji su elementi niza nazivaju se i *najizvedenijim objektima* (*most-derived object*), da bi se razlikovali od podobjekata osnovnih klasa ugrađene u objekte izvedenih klasa
- Životni vek ugrađenih objekata određen je životnim vekom objekta u koji su ugrađeni i vezan je za njega: ugrađeni objekti nastaju zajedno sa okružujućim objektom (inicijalizuju se tokom inicijalizacije okružujućeg objekta) i nestaju zajedno sa njim (uništavaju se tokom uništavanja okružujućeg objekta)
- Redosled inicijalizacije je uvek određen za objekte ugrađene u isti objekat. Redosled njihovog uništavanja je takođe određen i uvek je suprotan redosledu njihove inicijalizacije
- Elementi niza inicijalizuju se po redosledu njihovih indeksa (počev od elementa na poziciji 0 ka rastućim pozicijama); uništavaju se obrnutim redosledom

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;
};

Whole::Whole () : Entity(), part(this) {
  …part…
}

Whole::~Whole () {…}
```

- 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 konstatntog 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) {}
```