Operator dodele premeštanjem

- *Operator dodele premeštanjem* (*move assignment operator*) klase *X* je nestatička operatorska funkcija članica te klase sa imenom *operator*= čiji je jedini parametar tipa *X&&*, *const X&&*, *volatile X&&* ili *const volatile X&&*
- Ova funkcija poziva se kada se objekat klase *X* nalazi kao levi operand operatora dodele = i kada se baš ona odabere u postupku odabira preklopljene operatorske funkcije za desni operand tog operatora, odnosno kada je desni operand dvrednost ili se može konvertovati u dvrednost
- I za operator dodele premeštanjem važe slična pravila kao i za konstruktor premeštanja, odnosno za operator dodele kopiranjem: ako klasa nema nijedan eksplicitno deklarisan operator dodele premeštanjem (tj. korisnički deklarisan operator dodele premeštanjem) i pod uslovom da ta klasa nema korisnički definisan konstruktor kopije, konstruktor premeštanja, operator dodele kopiranjem i destruktor, prevodilac će implicitno deklarisati konstruktor premeštanja koji je javan, *inline* i nije *explicit*
- Ovaj implicitno deklarisani operator dodele premeštanja može se smatrati i obrisanim pod odgovarajućim uslovima, a ako nije obrisan ili je podrazumevan, on vrši dodelu premeštanjem podobjekata osnovnih klasa i objekata članova, odnosno dodelu kopiranjem ako premeštanje nije definisano

Semantika vrednosti od C++17

- Do verzije C++17, optimizacija izostavljanja kopiranja bila je neobavezna i važilo je *as-if* pravilo: čak i kada se kopiranje izbegava, odgovarajući konstruktor kopije ili premeštanja morao je da bude definisan za navedene situacije inicijalizacije privremenim objektima, čak i kada je sigurno da se ti konstruktori nikada ne pozivaju (jer prevodilac uvek vrši ovu optimizaciju i programer to može da zna)
- Zato su morali da se prave prestupi i definišu ovakvi konstruktori i za klase koje prirodno ne treba da budu takve da se njihovi objekti mogu kopirati ili premeštati, ili su pravljena drugačija zaobilazna rešenja, ako se želi prenos po vrednosti
- Počev od verzije C++17, optimizacija izbegavanja kopiranja (*copy elision*) obavezna je (a ne samo opciona) na sledećim mestima:
  - u naredbi *return*, kada je operand ove naredbe čdvrednost (*prvalue*) istog klasnog tipa (ignorišući cv-kvalifikacije) kao i povratni tip funkcije (RVO):

```
X f () {
  return X();
}

f();
```

  - pri inicijalizaciji varijable, kada je inicijalizator čdvrednost (*prvalue*) istog klasnog tipa (ignorišući cv-kvalifikacije) kao i varijabla:

```
X x = X(X(f()));
```

- Odgovarajući konstruktori kopije i premeštanja čak više ne moraju ni da postoje niti da budu dostupni, jer semantika jezika garantuje da se oni ne pozivaju na ovim mestima
- Optimizacija NRVO je i dalje opciona: ako se ona ne vrši, poziva se odgovarajući konstruktor premeštanja ili kopije prilikom inicijalizacije povratne vrednosti ako je operand naredbe *return* imenovana automatska varijabla (ali ne parametar)