
Preklopljeni operatori

❖ Postoje sledeća ograničenja:

- Ne mogu da se preklope sledeći operatori: `::`, `.`, `.*` i `?:`, kao ni operatori koji nisu simboli, već ključne reči (*sizeof*, *const_cast*, *static_cast*, *dynamic_cast*, *reinterpret_cast* itd.)
- Ne mogu da se definišu novi operatori, npr. `**` ili `<>`
- Preklopljeni operatori `&&` i `||` gube svojstvo izračunavanja prečicom (*short-circuit evaluation*)
- Preklopljeni operatori `=`, `()`, `[]`, i `->` moraju biti nestatičke funkcije članice
- Preklopljeni operator `->` mora da vrati ili običan pokazivač, ili objekat (po vrednosti ili referenci) za koji je ponovo definisan preklopljeni operator `->`
- Ne može da se promeni broj operanada, način grupisanja i prioritet operatora; zapravo se ne može promeniti način tumačenja (parsiranja) izraza i pridruživanja operanada operatorima, jer to određuju pravila jezika; ali kada pridruži operande operatorima, prevodilac traži odgovarajuću preklopljenu funkciju za operande koji su tipa klase ili enumeracije

❖ Povratni tip preklopljenog operatora može biti bilo koji, pa i *void*

Preklopljeni operatori

- ❖ Preklopljeni operatori mogu biti deklarirani ili kao nestatičke funkcije članice ili kao funkcije nečlanice (često prijateljske); neki operatori mogu biti preklopljeni na bilo koji od ta dva načina, a neki samo kao članice
- ❖ Preklopljeni operatori mogu biti unarni (imaju jedan operand) ili binarni (imaju dva operanda)
- ❖ Unarni operator @ ima jedan operand, pa može biti preklopljen kao:

- nestatička funkcija članica klase X bez parametara:

T X::operator@ ();

tada se operacija @x tumači kao

x.operator@ ();

- jedini operand je objekat čija se funkcija članica poziva

- funkcija nečlanica klase X sa jednim parametrom koji prihvata argument tipa X:

T operator@ (X);

tada se operacija @x tumači kao

operator@ (x)

- jedini operand se prenosi kao argument funkcije