

# Preklopljeni operatori

Izraz	Kao članica	Kao nečlanica	Primer
@a	(a).operator@()	operator@(a)	!std::cin poziva std::cin.operator!()
a@b	(a).operator@(b)	operator@(a,b)	std::cout << 42 poziva std::cout.operator<<(42)
a = b	(a).operator=(b)	Ne može	std::string s; s = "abc"; poziva s.operator=("abc")
a(b...)	(a).operator() (b...)	Ne može	std::random_device r; auto n = r(); poziva r.operator>()()
a[b]	(a).operator[] (b)	Ne može	std::map<int,int> m; m[1] = 2; poziva m.operator[](1)
a->	(a).operator->()	Ne može	auto p = std::make_unique<S>(); p->bar() poziva p.operator->()
a@	(a).operator@()	operator@(a,0)	std::vector<int>::iterator i =...; i++ poziva i.operator++()

# Preporučeni načini preklapanja operatora

- ❖ Osim navedenih ograničenja, pravila jezika ne nameću bilo koja druga, pa stoga:
  - preklopljeni operator može imati bilo kakve efekte i bilo kakvo značenje; na primer, operator dodele `=` uopšte ne mora raditi dodelu kopiranjem, ili operator `+` uopšte ne mora raditi sabiranje (šta god to značilo za dati tip)
  - preklapanje jednog operatora ne znači implicitno i preklapanje nekog drugog, blisko povezanog operatora; na primer, ako je za klasu preklopljen operator `==`, nije implicitno preklopljen i operator `!=`, ili, ako je za klasu preklopljen operator `<`, nije implicitno preklopljen i operator `>` i slično
  - veze koje postoje za ugrađene operatore ne moraju biti takve i za preklopljene operatore; na primer, za ugrađene operatore `+`, `=` i `+=` važi odgovarajuća sprega, koja uopšte ne mora da važi za preklopljene operatore (da  $a+=b$  ima efekat kao  $a=a+b$ )
  - preklopljeni operatori mogu da vraćaju bilo koji tip, pa i tip *void*, iako ugrađeni operatori vraćaju odgovarajuće tipove; na primer, ugrađeni operator dodele vraća vrednost koja upućuje na levi operand, dok preklopljeni operator dodele može vratiti bilo koji tip i bilo kakvu vrednost