

Preklopljeni operatori

- ❖ Operatori ++ i -- imaju prefiksni i postfiksni oblik koji se mogu razlikovati tako što se preklopljeni postfiksni operatori definišu sa parametrom koji prihvata drugi operand tipa *int*; pri pozivu preklopljene operatorske funkcije, ovaj argument ima vrednost 0:

```
complex& complex::operator++ () {...}
```

Prefiksni operator, vraća lrvrednost

```
complex complex::operator++ (int) {...}
```

Postfiksni operator, vraća čdvrednost

```
complex c(3.,4.);
```

```
c++;
```

Tumači se kao *c.operator++(0)*

```
++c;
```

Tumači se kao *c.operator++()*

- ❖ Preklopljeni operator -> mora biti nestatička funkcija članica koja nema parametre i koja mora da vrati ili običan pokazivač, ili (referencu na) objekat za koji je takođe preklopljen operator ->:

```
template<typename T>
```

```
T* smart_ptr<T>::operator-> () {...}
```

```
smart_ptr<Clock> p = &clk;
```

```
p->tick();
```

Tumači se kao *(p.operator->())->tick();*

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++()