

Preporučeni načini preklapanja operatora

- ❖ Unarni logički operator negacije `!` se preklapa za klase čiji se objekti koriste u kontekstu gde se očekuju Bulovi izrazi
- ❖ Zbog toga ove klase najčešće imaju i operator konverzije u tip `bool` koji je *explicit*, a operator `!` svodi se na negaciju te konvertovane vrednosti:

```
X::operator bool () {  
    ...  
}
```

```
inline bool operator! (const X& x) {  
    return !bool(x);  
}
```

Preporučeni načini preklapanja operatora

- ❖ Operator `[]` se mora preklopiti kao nestatička funkcija članica; ona može imati parametar bilo kog tipa; taj parametar prima vrednost izraza između uglastih zagrada (drugog operanda):

`x[expression]`

tumači se kao:

`x.operator[](expression)`

- ❖ Ovaj operator se tipično preklapa tako da postoje varijante koje omogućavaju i čitanje iz konstantnih i nekonstantnih objekata, kao i upis u nekonstantne objekte, odnosno varijante koje su konstantna i nekonstantna funkcija članica i koje vraćaju odgovarajuće lrvrednosti (reference na konstantu i nekonstantu, respektivno):

```
class X {  
    ...  
    T& operator[] (size_t);  
    const T& operator[] (size_t) const;  
    ...  
};
```