

Preporučeni načini preklapanja operatora

- ❖ Binarni, infiksni relacioni operatori se obično preklapaju kao funkcije nečlanice, jer su im operandi ravnopravni
- ❖ Preklapaju se tako što se implementira operator `==` i, ako je potrebno, operator `<`, a onda ostali svode na njih:

```
bool operator== (const X& x1, const X& x2) {  
    ...  
}
```

```
inline bool operator!= (const X& x1, const X& x2)  
    { return !(x1==x2); }
```

```
bool operator< (const X& x1, const X& x2) {  
    ...  
}
```

```
inline bool operator> (const X& x1, const X& x2)  
    { return x2<x1; }
```

```
inline bool operator<= (const X& x1, const X& x2)  
    { return (x1<x2)|| (x1==x2); }
```

```
inline bool operator>= (const X& x1, const X& x2)  
    { return x2<=x1; }
```

Preporučeni načini preklapanja operatora

- ❖ Od verzije C++20 postoji operator `<=>` koji se može preklopiti, a kad je on preklopljen, implicitno su definisani i svi drugi relacioni operatori u prirodnom obliku:

```
int operator<=> (const string& s1, const string& s2) {  
    return strcmp(s1.str,s2.str);  
}
```

- ❖ Kad je ovaj operator preklopljen, implicitno je definisano i svih šest ostalih relacionih operatora u prirodnom obliku:

```
inline bool operator== (const string& s1, const string& s2) { return (s1<=>s2)==0; }  
inline bool operator!= (const string& s1, const string& s2) { return !(s1==s2); }  
inline bool operator< (const string& s1, const string& s2) { return (s1<=>s2) < 0; }  
inline bool operator> (const string& s1, const string& s2) { return s2<s1; }  
inline bool operator<= (const string& s1, const string& s2)  
    { return (s1<s2)|| (s1==s2); }  
inline bool operator>= (const string& s1, const string& s2) { return (s2<=s1); }
```

- ❖ Može se tražiti i generisanje podrazumevane implementacije ovog operatora, koja se svodi na isti operator primenjen redom na podobjekte, u poretku njihovog deklarisanja:

```
bool operator<=> (const X&, const X&) = default;
```