

Preporučeni načini preklapanja operatora

- ❖ Međutim, preklopljene operatore treba praviti tako da imaju očekivano značenje i efekte, kao i da njihova upotreba što više liči na upotrebu ugrađenih operatora, jer to i jeste osnovna motivacija i namera celog koncepta preklapanja operatora - da se korisnički definitisani tipovi mogu upotrebljavati na isti način kao i ugrađeni tipovi, pa i u izrazima
- ❖ Prvo, preklopljeni operatori treba da imaju očekivane, logične efekte; na primer, operator dodele = treba da radi dodelu kopiranjem ili premeštanjem, a operator + (ako je definisan) treba da radi sabiranje (šta god to značilo za dati tip)

Preporučeni načini preklapanja operatora

- ❖ Skup preklopljenih operatora treba da bude “zatvoren”, u smislu da su preklopljeni svi operatori koji se očekuju zajedno; to nikako ne znači da za klasu treba preklapati sve moguće operatore, već samo da skup operatora treba kompletirati na one očekivane; na primer:
 - ako je preklopljen operator `==`, treba preklopiti i operator `!=`, i to baš tako da vraća negaciju operatora `==` za iste operande, ali ne obavezno i operatore `<` i `>` (jer za klasu možda nema smisla takva relacija uređenja)
 - ako je za klasu preklopljen operator `==` i `<`, treba preklopiti i operatore `!=`, `>`, `<=` i `>=`; od verzije jezika C++20, u jeziku postoji operator opšteg poređenja `<=>` koji vraća vrednost manju od 0 ako je prvi operand manji od drugog, veću od 0 ako je prvi veći od drugog, a nulu ako su operandi jednaki; dovoljno je preklopiti samo ovaj operator i svi drugi vezani operatori poređenja na jednakost i nejednakost biće implicitno definisani na odgovarajući način
 - ako je za klasu preklopljen operator `+` i postoji operator `=`, treba preklopiti i operator `+=` i to sa očekivanim spregnutim značenjem, jer će se i on očekivati,

i tako dalje