

Klasa kao realizacija apstraktnog tipa podataka

- ❖ Osim uobičajenog načina poziva preko identifikatora, ovakve *operatorske funkcije* (*operator functions*) mogu se pozivati i implicitno, iz izraza, korišćenjem notacije operatora ugrađenih u jezik:

...c3+c4-c5...

Poziv negde u izrazu. Poziva se:

`operator-(operator+(c3,c4),c5)`

- ❖ Prevodilac raščlanjuje (parsira, *parse*) izraz na isti način kao što je definisano pravilima jezika i to se ne može promeniti:
 - broj operanada svakog od operatora (*n*-arnost)
 - prioriteti i redosled izračunavanja
 - asocijativnost (način grupisanja - sleva nadesno ili obratno)
- ❖ Međutim, ako je neki od operanada instanca korisničkog tipa, poziva se odgovarajuća operatorska funkcija definisana za taj tip operanda, umesto da se generiše uobičajeni mašinski kod za operacije nad ugrađenim tipovima (ponovo polimorfizam!)
- ❖ Većina operatora ugrađenih u jezik može se definisati za korisničke tipove (ali ne svi)
- ❖ Za mnoge operatore postoji podrazumevano značenje koje se može promeniti (redefinisati)
- ❖ Ne mogu se definisati novi operatori (samo redefinisati značenje onih koji su već ugrađeni u jezik)
- ❖ Ne može se promeniti (redefinisati) značenje za operatore koji rade (samo) nad ugrađenim tipovima (bar jedan operand operatorskih funkcija mora biti korisničkog tipa)
- ❖ Mnogo opštih i posebnih pravila za neke posebne operatore - detalji kasnije

Klasa kao realizacija apstraktnog tipa podataka

- ❖ Ostaje još operacija dodele vrednosti:

```
c3 = c5;
```

- ❖ Operacija dodele vrednosti je *različita* od *inicijalizacije*:

```
complex c3 = c5; // Initialization  
c3 = c5; // Assignment operation
```

- inicijalizacija se vrši kada se objekat kreira; operacija dodele se vrši kao operacija u izrazu
 - inicijalizacija se vrši nad objektom koji tek nastaje i do tada nije postojao; operacija dodele se vrši nad objektom koji već postoji
 - kod inicijalizacije se poziva konstruktor; kod dodele vrednosti se poziva operator dodele kao operatorska funkcija
- ❖ U jeziku C nije bilo posebne potrebe da se razlikuju ove dve operacije, jer postoje samo ugrađeni tipovi, za koje je semantika ove dve operacije uvek ista - prosto kopiranje vrednosti
 - ❖ Operator dodele je nestatička operatorska funkcija klase:

```
complex complex::operator= (complex other) {  
    this->re = other.re; this->im = other.im;  
    return *this;  
}
```

- ❖ Zašto vraćamo **this*? Da bi se redefinisana operatorska funkcija ponašala što sličnije onoj ugrađenoj, a ona vraća vrednost, i to baš onu dodeljenu, kako bi se moglo pisati:

```
c1 = c2 = c3; // Computed as: c1 = (c2=c3)
```

- ❖ Svaka klasa ima podrazumevani operator dodele, koji vrši podrazumevanu dodelu član po član, ali se on može i eksplicitno redefinisati