

Klasa kao realizacija apstraktnog tipa podataka

- ❖ Ideja rešenja: umesto prenosa po vrednosti (*pass by value*), prenositi argumente *po referenci* (*pass by reference*) - formalni argument je *referenca na objekat* (posrednik do objekta), slično kao pokazivač:

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

- ❖ Sada je rekurzija prekinuta, jer se prilikom poziva konstruktora kopije:

```
complex::complex(c3)
```

formalni argument *other*, kao referenca, *vezuje* tako da referencira (upućuje) na stvarni argument *c3*

- ❖ Svaka upotreba reference, nakon njene inicijalizacije, odnosi se na *referencirani objekat* (bez izuzetka); zato se kaže da je referenca *alias* (*alias*) za objekat za koji je vezan
- ❖ Nema načina da se izvrši operacija nad referencom, veza reference sa objektom na koji ona upućuje ne može se raskinuti ili preusmeriti kao za pokazivače, iako se u principu implementira i ponaša slično:

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

Pristup do člana objekta na koji upućuje referenca *other* - upotreba reference u izrazu uvek se odnosi na referencirani objekat - stvarni argument funkcije u ovom slučaju

Klasa kao realizacija apstraktnog tipa podataka

- ❖ Prenos objekata kao argumenata poziva funkcija po vrednosti, kopiranjem, se rešava na isti način:

```
complex complex::add (complex ca, complex cb);  
...complex::add(c3,c4)...
```

- ❖ Ponovo se formalni argument *ca* (*cb*) inicijalizuje stvarnim argumentom *c3* (*c4*) pozivom konstruktora kopije, sa istom semantikom:

```
complex ca = c3;  
complex cb = c4;
```

- ❖ I povratna vrednost funkcije tj. rezultat poziva funkcije se vrši po vrednosti, takođe kopiranjem, na isti način
- ❖ Na mestu poziva funkcije, u trenutku povratka iz funkcije, kreira se *privremeni* (*temporary*) objekat koji se inicijalizuje izrazom iz naredbe *return*; semantika ove inicijalizacije je ista kao i svake druge inicijalizacije:

```
complex complex::add (complex ca, complex cb) {  
    ...  
    return result;  
}  
...complex::add(c1,c2)...
```

Rezultat poziva *complex::add* jeste jedan bezimeni, privremeni objekat koji se kreira na mestu poziva ove funkcije, u trenutku povratka iz nje, i inicijalizuje rezultatom izraza iza naredbe *return* - objektom *result*. Poziva se konstruktor kopije, sa sledećim značenjem:

```
complex temp = result;
```