Klasa kao realizacija apstraktnog tipa podataka

- Prenos objekata kao argumenata poziva funkcija po vrednosti, kopiranjem:

```
complex complex::add (complex ca, complex cb);
```

...`complex::add(c3,c4)`...

- Formalni argument, kao lokalni automatski objekat, inicijalizuje se stvarnim argumentom, na isti način - konstruktorom kopije:

```
complex::complex (complex other);
```

  U trenutku poziva funkcije:

```
complex::add(c3,c4)
```

  na mestu ulaska u funkciju, kreira se lokalni, automatski objekat - formalni argument *ca* (odnosno *cb*) i inicijalizuje stvarnim argumentom *c3* (odnosno *c4*); semantika te inicijalizacije ista je kao i semantika bilo koje druge inicijalizacije, kao da je izvedeno:

```
complex ca = c3;
```

- A kako se vrši ova inicijalizacija? Pozivom konstruktora kopije za objekat *ca* koji se inicijalizuje objektom *c3*:

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

- Međutim, i konstruktor kopije ima formalni argument *other*, koji se opet formira kao automatski objekat i inicijalizuje stvarnim argumentom (*c3*) u trenutku ovog poziva. Kako? Pozivom istog tog konstruktora kopije…
- Problem - beskonačna rekurzija.

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 *alijas* (*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;
}
```