Klasa kao realizacija apstraktnog tipa podataka

- Želimo da realizujemo *apstraktni tip podataka* (*abstract data type*), kompleksan broj: strukturu sa pridruženim operacijama koja predstavlja *korisnički definisani tip* (*user-defined type*) - tip koji ne postoji ugrađen u jezik (*built-in type*), već ga definiše programer (kao korisnik jezika)
- Možemo kao i ranije, na jeziku C:

```
struct _complex {
  double re, im;
};
typedef struct _complex complex;

void complex_init (complex* this, double real, double imag);
complex complex_add (complex c1, complex c2);
complex complex_sub (complex c1, complex c2);
…

void complex_init (complex* this, double real, double imag) {
  this->re = real; this->im = imag;
}

complex complex_add (complex c1, complex c2) {
  complex result;
  result.re = c1.re + c2.re;
  result.im = c1.im + c2.im;
  return result;
}
…
```

- I onda to da koristimo:

```
complex c1, c2, c3, c4;
complex_init(&c1, -2.5, 6.8);
complex_init(&c2, 46.5, -34.45);
c3 = complex_add(c1,c2);
c4 = complex_sub(c1,c2);
complex c5 = c3;
c3 = c4;
```

Klasa kao realizacija apstraktnog tipa podataka

- Primetimo sledeće: vrednosti tipa *complex* su za jezik C proste strukture i *imaju semantiku kreiranja i kopiranja po vrednosti* (*value semantics, copy semantics*):
  - Kreiraju se instance (primerci, promenljive) svih vrsta životnog veka (automatski, statički, privremeni itd.):

```
complex result;
complex c1, c2, c3, c4;
```

  - Inicijalizuju se drugim instancama istog tipa, ponovo po vrednosti, prostim kopiranjem:

```
complex c5 = c3;
```

  - Prenose se kao argumenti poziva funkcija po vrednosti, prostim kopiranjem:

```
...complex_add(c1,c2)...
```

  - Vraćaju se kao rezultati poziva funkcija po vrednosti, prostim kopiranjem:

```
return result;
...
c3 = complex_add(c1,c2);
```

  - Dodeljuju se drugim instancama istog tipa, ponovo po vrednosti, prostim kopiranjem:

```
c4 = complex_sub(c1,c2);
c3 = c4;
```

- Sve ovo je zato što je na jeziku C ovako definisana semantika za strukture, na potpuno isti način kao i za sve ostale ugrađene tipove (celobrojne, racionalne, pokazivače itd.)
- Zašto nam ovo ovako odgovara: zato što različite instance ovog tipa, sa istim vrednostima (svih svojih atributa) predstavljaju *isti konceptualni entitet* - isti kompleksan broj; zato se mogu slobodno kopirati