

# 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 (primeri, 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

# Klasa kao realizacija apstraktnog tipa podataka

❖ Kako bi to izgledalo na jeziku C++ korišćenjem samo onoga što smo do sada videli:

```
class complex {
public:
    complex (double real, double imag);
    complex (complex* other); // For copy-initialization

    void copy (complex* other); // For copy-assignment

    static complex* add (complex* c1, complex* c2);
    static complex* sub (complex* c1, complex* c2);

private:
    double re, im;
};

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

complex* complex::add (complex* c1, complex* c2) {
    complex* result = new complex(c1->re+c2->re,c1->im+c2->im);
    return result;
}

...
```

❖ *Statičke funkcije članice (static member functions)* su deklarisanе uz ključnu reč *static* ispred:

- jesu članice klase, pripadaju oblasti važenja klase i mogu da pristupaju privatnim članovima te klase (bilo kog objekta)
- nemaju u sebi implicitno deklarisan pokazivač *this*
- stoga ne mogu da pristupaju članovima neposredno, bez navođenja objekta kome član pripada
- zato konceptualno “ne pripadaju” pojedinačnom objektu, već celoj klasi: pružaju uslugu koja se traži od klase, a ne pojedinačnog objekta