

## Supraîncărcarea operatorului +

În cazul în care este definit în cadrul unei clase, el va avea efect între o variabilă de tipul clasei respective și altceva.

Pe lângă în cadrul clasei vom defini operatorul.

sem tip- rezultat  
tip-clasa operator + (parametri)  
{ ... }

Îată cum putem defini acest lucru pt.  
clasa noastră  $\mathbb{C}^{\text{plx}}$ :

```

class cplx
{
private:
    double re, im;
public:
    // ...
    cplx operator + (cplx z)
    {
        cplx rez;
        rez.re = re + z.re;
        rez.im = im + z.im;
        return rez;
    }
}

```

// funcția de mai sus va permite expresii  
 // de tipul  $\text{cplx} + \text{cplx} \rightarrow$  suma a două  
 nr. complexe

Vom supraîncărca operatorul  $+$  a.î. să știe să lucreze  
și cu  $\text{cplx} + \text{int}$   
 $+ \text{double}$

```
cplx operator+(double pr)
```

```
{ cplx rez;  
  rez.re = re + pr;  
  rez.im = im;  
}
```

pe această funcție, dacă avem  
 $\text{cplx } z;$

ni folosim expresia  $z + 1 \rightarrow$  funcționare

În schimb, expresia 1 + z NU merge.

→ adunarea NU se mai face între un repres.  
al clasei și alt parametru, ci între un double  
și un cplx.

Pentru asta trebuie supraîncărcat +  
ca funcție extensă clasei, prețenă acesteia.

O funcție prețenă ARE dreptul să  
aceseze datele private ale clasei.

Întâ codul:

friend cplx operator+(double x, cplx z)

{ cplx rez;

rez.re = z.re + x;

rez.im = z.im;

return rez;

}