Curs 6 - POO

Funcțiile Friend sunt independente (nu fac parte din clasă)

```
class idClasă
  {
      - - - - - -
      friend tip_returnat nume funcție (<list_args>);
  };

~~idClasa::~~ tip_returnat numefuncție (<list_args>) {......}
```

OBS 2 | Funcția friend nu primește argumentul this -> Funcția friend primește ca argument referința obiectului

---

2) Supraîncarcare operatori C++

tip_date (int, double, Persoana) ⟶ Setul de date (int -> $[-2^{32}, 2^{32}-1]$)
↳ Setul de operații
↳ (int ← +, -, /, ·, %)
↳ (Persoană ← afișare, get/set, calcul Salariu)

Exemplu: Complex $z_1(1,2)$, $z_2(3,4)$;
Complex z;
$z = z_1 \boxed{+} z_2$;

OBS | Pentru tipurile de date definite de programator, nu se pot utiliza operatorii definiți de limbaj.
Soluție 1 | Implementare unei metode membre a operației în sine

```
class Complex
    {. - - - -
      public:
  Complx adunare (Complex z2)
          { Complex suma;
              suma.re = this -> re + z2.re;
              suma.im = this -> im + z2.im;
              return suma;
          }
    };

int main(){
    Complex z1(1,2), z2(2,3), z;
    z = z1.adunare(z2);        // z=z1+z2;
```

ARITATEA | Numarul operanzilor folosiți de operator

Supraincarcarea unui operator ⟶ FUNCȚIE OPERATOR

Există 2 modalități prin care un operator poate fi supraincarcat:
1) Prin metode operator membri clasei (this)
2) Prin funcții independente clasei de tip friend (fara this)

---

1) Supraincarcarea operatorilor prin metode membre

SINTAXA:
tip_returnat operator # (<list_arg>)

#| Simbolul operatorului
<list_args> | Conține un numar de argumente = aritate - 1

Exemplu:
"+" op binar -> operator + combină doar un argument

OBS | Argumentul din lista_argumente este operandul din dreapta!

$z = \boxed{z_1} + \boxed{z_2}$ → argumentul
↳ This

OBS| Argumentul din stânga este mereu pointerul THIS (obiect curent)

2) Supraîncarcarea functiilor prin functii independente FRIEND

Sintaxa:

friend tip_returnat operator# (<list_args>)

<list_args> | nr. args = aritatea operatorului

Op. binar → un argument din stânga
↳ un argument din dreapta

$z = \boxed{z_1} + \boxed{z_2};$ ↳ al doilea argument
↳ prim argument

Exp: $z = 2 \cdot z;$ → (double v, complex z)
$z = z \cdot 2;$ → (Complex z, double v)

OBS| Se pot supraîncarca funcțiile operator

OBS 2| Sunt operatori care se pot supra-încarca prin ambele metode, dar sunt operatori care se pot supraîncarca restrictiv, fie prin metode membre, fie prin funcții friend.

Ex. "<<" supra-încarcare prin funcție friend!