

# PCLP2

## Laboratorul 6

Supraîncărcarea operatorilor este o metodă prin care au diferite implementări care depind de operanzii pe care îi folosesc.

1. **Supraîncărcarea operatorului binar + ca funcție membră a clasei `Complex`.** Testați programul de mai jos. Clasa `Complex` dispune de două date membre de tip `real`, un constructor cu parametri care au valori implicite, funcții getter și setter pentru datele membre și un operator de adunare.

### **complex.h**

```
#ifndef COMPLEX_H
#define COMPLEX_H
class Complex
{
public:
    Complex(double=0, double=0);
    void setReal(double);
    double getReal();
    void setImaginar(double);
    double getImaginar();
    Complex operator+(Complex);
private:
    double real;
    double imaginar;
};
#endif
```

### **complex.cpp**

```
#include "complex.h"
Complex::Complex(double r, double i)
{
    real = r;
    imaginar = i;
}
void Complex::setReal(double r)
{
    real = r;
}
double Complex::getReal()
{
    return real;
}
void Complex::setImaginar(double i)
{
    imaginar = i;
}
double Complex::getImaginar()
{
    return imaginar;
}
Complex Complex::operator+(Complex c)
```

```
{
    Complex nc;
    nc.setReal(real+c.real);
    nc.setImaginar(imaginar+c.imaginar);
    return nc;
}
```

### **test\_complex.cpp**

```
#include <iostream>
using std::cout;
using std::endl;

#include "complex.h"

int main()
{
    Complex n1(2,4);
    Complex n2(1,-4);
    n2 = n1+n2;
    cout << n2.getReal() << ";" << n2.getImaginar() << endl;
    return 0;
}
```

Refolosiți programul de mai sus și urmați exemplele din curs pentru exercițiile următoare. Apelați în funcția `main` fiecare dintre cei trei operatori de mai jos pentru a ilustra funcționarea lor corectă.

2. Implementați operatorul binar `*` de înmulțire a două numere complexe ca funcție nemembră a clasei `Complex`.

3. Implementați operatorul unar `!` ca funcție membră în clasa `Complex`. Acest operator schimbă semnul părții imaginare.

4. Implementați operatorul `<<` pentru clasa `Complex`.