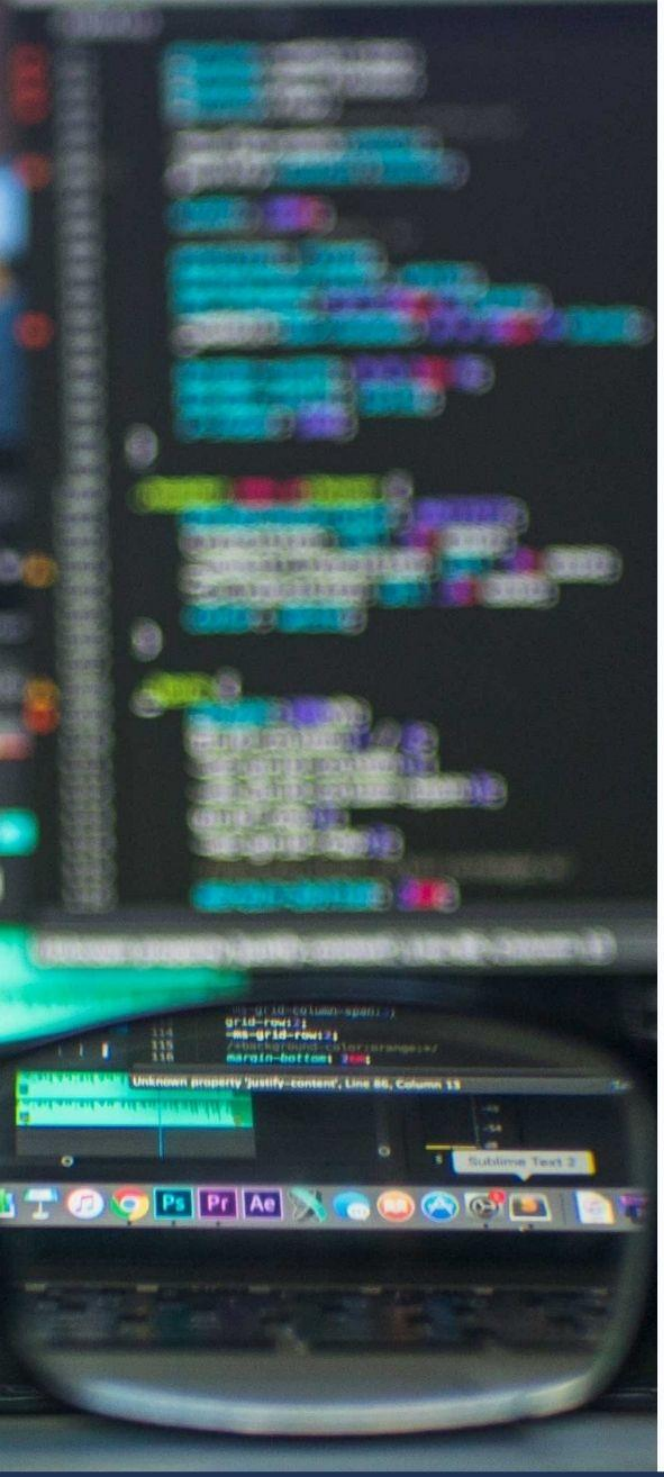


ECOLE NORMALE SUPÉRIEURE DE
L'ENSEIGNEMENT TECHNIQUE MOHAMMEDIA
DÉPARTEMENT MATHÉMATIQUES ET
INFORMATIQUE

24/04/2022

Compte rendu

Des travaux de Synthèse

La surdéfinition des opérateurs (cas des nombres complexe)

PRÉPARÉE POUR
TAFFAH ACHRAF

ENCADRÉ PAR
PR. KHALIFA MANSOURI

Filière d'ingénieur : Génie du Logiciel et des Systèmes Informatiques

TABLE DES MATIERES

PARTIE PRATIQUE

1. La surdéfinition des opérateurs (cas des nombres complexe) :

LA DECLARATION DE LA CLASSE COMPLEXE DANS LE FICHER HEADER :

```
#ifndef _COMPLEXE
    #define _COMPLEXE_
        #include<string>
        using namespace std;
        class complexe{
            double real,img;
        public:
            complexe(double,double);
            double get_real() const;
            double get_img() const;
            void set_real(double);
            void set_img(double);
            complexe conj();
            double norm();
            double arg();
            complexe polar(double,double);
            complexe operator+(complexe const &);
            complexe operator-(complexe const &);
            complexe operator*(complexe const &);
            complexe operator/(complexe const &);
            bool operator==(complexe const &);
            bool operator!=(complexe const &);
            complexe operator-();
            string to_string() const;

            friend complexe operator+(complexe const &,double);
            friend complexe operator+(double,complexe const &);

            friend complexe operator-(complexe const &,double);
            friend complexe operator-(double,complexe const &);

            friend complexe operator*(complexe const &,double);
            friend complexe operator*(double,complexe const &);

            friend complexe operator/(complexe const &,double);
            friend complexe operator/(double,complexe const &);

            friend ostream & operator << (ostream &, complexe const &);
            friend istream & operator >> (istream &, complexe &);
        };
        #endif
```

LA DEFINITION DES METHODES DE LA CLASSE COMPLEXE :

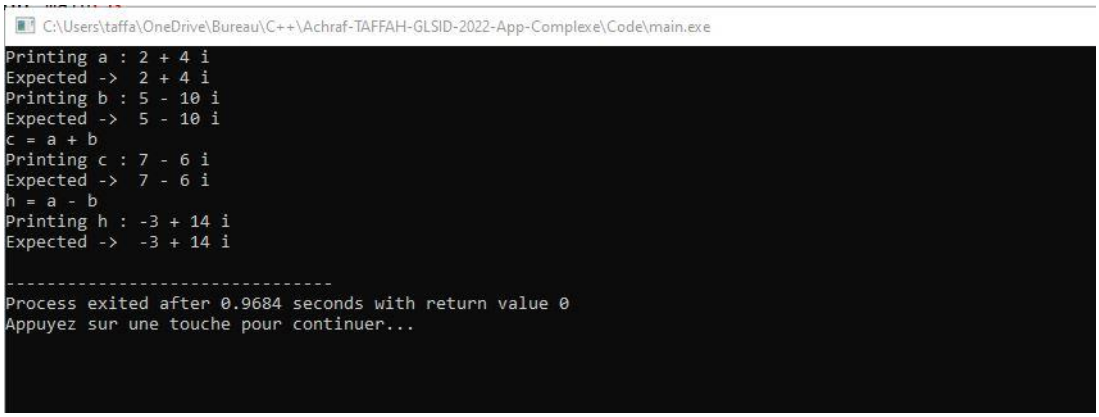
```cpp
/*Partie 1 --> Complexe.cpp */
#include "complexe.hpp"
    #include <iostream>
    #include <sstream>
    #include <math.h>
    #include<string.h>
    using namespace std;
    complexe::complexe(double r,double im):real(r),img(im){}

    double abs(double a){
        return a > 0 ? a : -a;
    }
    double complexe::get_real() const{
        return this->real;
    }
    double complexe::get_img() const{
        return this->img;
    }
    void complexe::set_real(double r){this->real = r;}
    void complexe::set_img(double im){this->img = im;}
    complexe complexe::polar(double adj,double deg){
        return complexe(adj * cos(deg) , adj * sin(deg));
    }

    complexe complexe::conj(){
        return complexe(this->real,-this->img);
    }
    double complexe::norm(){
        return sqrt((this->real*this->real)+(this->img*this->img));
    }
    double complexe::arg(){
        return atan(this->real/this->img);
    }

    complexe complexe::operator+(complexe const & comp){
        return complexe(this->real+comp.real,this->img+comp.img);
    }
    complexe complexe::operator-(complexe const & comp){
        return complexe(this->real-comp.real,this->img-comp.img);
    }
    complexe complexe::operator*(complexe const & comp){
        return complexe(this->real*comp.real,this->img*comp.img);
    }
    complexe complexe::operator/(complexe const & comp){
        double r,im;
        (comp.real == 0) ? r = 0 : r = this->real / comp.real;
        (comp.img == 0) ? im = 0 : im= this->img / comp.img;
        return complexe(r,im);
    }
    bool complexe::operator==(complexe const & comp){
        return (this->real == comp.real) && (this->img == comp.img);
    }
```

```cpp
/*Partie 1 --> Complexe.cpp */
    bool complexe::operator!=(complexe const & comp){
        return (this->real != comp.real) || (this->img != comp.img);
    }
    complexe complexe::operator-(){
        return complexe(-this->real,-this->img);
    }
    string complexe::to_string() const{
        stringstream ss;
        ss<<this->real<<(this->img > 0 ? " + " : " - ")<<abs(this->img)<<" i";
        return ss.str();
    }

    complexe operator+(complexe const & comp,double value){
        return complexe(comp.real,comp.img+value);
    }
    complexe operator+(double value,complexe const & comp){
        return complexe(comp.real+value,comp.img);
    }

    complexe operator-(complexe const & comp,double value){
        return complexe(comp.real,comp.img-value);
    }
    complexe operator-(double value,complexe const & comp){
        return complexe(comp.real-value,comp.img);
    }

    complexe operator*(complexe const & comp,double value){
        return complexe(comp.real,comp.img*value);
    }
    complexe operator*(double value,complexe const & comp){
        return complexe(comp.real*value,comp.img);
    }

    complexe operator/(complexe const & comp,double value){
        double im;
        if(value == 0)
            im=0;
        else
            im=comp.img/value;
        return complexe(comp.real,im);
    }
    complexe operator/(double value,complexe const & comp){
        double r;
        (value == 0) ? r = 0 : r = comp.real/value;
        return complexe(r,comp.img);
    }

    ostream & operator << (ostream & out, complexe const & comp){
        out << comp.to_string();
        return out;
    }
    istream & operator >> (istream & in, complexe & comp){
        in >> comp.real >> comp.img;
        return in;
    }
```

LA FONCTION MAIN POUR TESTER LE PROGRAMME :

```cpp
/*La fonction main */
    #include "complexe.cpp"
int main(){
  complexe a(2,4);
  complexe b(5,-10);
  cout <<"Printing a : "<< a << endl;
  cout << "Expected ->  2 + 4 i"<<endl;
  cout <<"Printing b : "<< b << endl;
  cout << "Expected ->  5 - 10 i"<<endl;

  complexe c = a + b;
  cout <<"c = a + b" << endl;
  cout <<"Printing c : "<< c << endl;
  cout << "Expected ->  7 - 6 i"<<endl;

  complexe h = a - b;
  cout <<"h = a - b" << endl;
  cout <<"Printing h : "<< h << endl;
  cout << "Expected ->  -3 + 14 i"<<endl;



  return 0;
}
```

RESULTAT :


C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID-2022-App-Complexe\Code\main.exe
Printing a : 2 + 4 i
Expected -> 2 + 4 i
Printing b : 5 - 10 i
Expected -> 5 - 10 i
c = a + b
Printing c : 7 - 6 i
Expected -> 7 - 6 i
h = a - b
Printing h : -3 + 14 i
Expected -> -3 + 14 i

Process exited after 0.9684 seconds with return value 0
Appuyez sur une touche pour continuer...