ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT
TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA

Filière d'ingénieur :

Ingénierie Informatique, Big Data et Cloud Computing (II-BDCC)

Département Mathématiques et Informatique

Compte rendu TP2

DE LA CONCEPTION ET LA PROGRAMMATION ORIENTE OBJET EN C++

Par: BOUZINE Ahmed houssam (II-BDCC1)

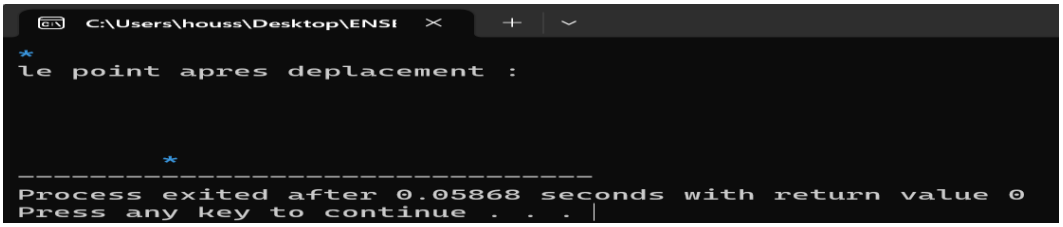
Encadré par : M. K. MANSOURI

Année Universitaire : 2022/2023

Exercice 1 :

```
using namespace std;

class Point
{
private:
    int x, y,couleur;

public:
    void initialiser(int _x = 0, int _y = 0, int _couleur = 3)
    {
        this->x = _x;
        this->y = _y;
        this->couleur = _couleur;
    }

    void afficher()
    {
        static HANDLE handle = NULL;
        COORD coords;
        coords.X = x;
        coords.Y = y;
        if (!handle)
        handle = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleCursorPosition(handle, coords);
        SetConsoleTextAttribute(handle, couleur);
        cout << "*";
        SetConsoleTextAttribute(handle, 7);
    }

    void effacer()
    {
        x = 0;
        y = 0;
        couleur=0;
    }
    void deplacer(int dx, int dy)
    {
        effacer();
        x = dx;
        y = dy;
        couleur=3;
    }

};

int main()
{
    Point point;
    point.initialiser(0, 0);
    point.afficher();
    point.deplacer(8, 6);
    cout << "\nle point apres deplacement :" << endl;
    point.afficher();
}
```

Exécution :



Exercice 2 :

```
using namespace std;
class Point
{
private:
    int x, y,couleur;
public:
    void initialiser(int _x = 0, int _y = 0, int _couleur = 3)
    {
        this->x = _x;
        this->y = _y;
        this->couleur = _couleur;
    }
    void afficher()
    {
        static HANDLE handle = NULL;
        COORD coords;
        coords.X = x;
        coords.Y = y;
        if (!handle)
        handle = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleCursorPosition(handle, coords);
        SetConsoleTextAttribute(handle, couleur);
        cout << "*";
        SetConsoleTextAttribute(handle, 7);
    }
    void effacer()
    {
        x = 0;
        y = 0;
        couleur=0;
    }
    void deplacer(int dx, int dy)
    {
        effacer();
        x = dx;
        y = dy;
        couleur=3;
    }

};
void scene()
{
    Point u;
    u.initialiser(0, 0);
    u.afficher();
    cout << "\nle point apres deplacement :" << endl;
    u.deplacer(8, 9);
    u.afficher();
}
int main()
{
   scene();
}
```

```
*
le point apres deplacement :



        *
--------------------------------
Process exited after 0.0671 seconds with return value 0
Press any key to continue . . .
```

Exercice 3 :

```
using namespace std;
class Point
{
private:
    int x, y,couleur;
public:
    Point(int _x = 0, int _y = 0, int _couleur = 3)
    {
        this->x = _x;
        this->y = _y;
        this->couleur = _couleur;
    }
    void afficher()
    {
        static HANDLE handle = NULL;
        COORD coords;
        coords.X = x;
        coords.Y = y;
        if (!handle)
        handle = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleCursorPosition(handle, coords);
        SetConsoleTextAttribute(handle, couleur);
        cout << "*";
        SetConsoleTextAttribute(handle, 7);
    }
    void effacer()
    {
        x = 0;
        y = 0;
        couleur=0;
    }
    void deplacer(int dx, int dy)
    {
        effacer();
        x = dx;
        y = dy;
        couleur=3;
    }

};
void scene()
{
    Point u(0.0);
    u.afficher();
    cout << "\nle point apres deplacement :" << endl;
    u.deplacer(8, 9);
    u.afficher();
}
int main()
{
   scene();
}
```

Exercice 4 :

```
using namespace std;
class Point
{
private:
    int x, y;
public:
    Point(int _x = 0, int _y = 0)
    {
        this->x = _x;
        this->y = _y;
    }
    void afficher()
    {
        static HANDLE handle = NULL;
        COORD coords;
        coords.X = x;
        coords.Y = y;
        if (!handle)
        handle = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleCursorPosition(handle, coords);
        SetConsoleTextAttribute(handle, 3);
        cout << "*";
        SetConsoleTextAttribute(handle, 7);
    }
    void effacer()
    {
        x = 0;
        y = 0;
    }
    void deplacer(int dx, int dy)
    {
        effacer();
        x = dx;
        y = dy;
    }

};
void scene()
{
    Point u(0.0);
    u.afficher();
    cout << "\nle point apres deplacement :" << endl;
    u.deplacer(8, 9);
    u.afficher();
}
int main()
{
   scene();
}
```

Exécution :

```
*
le point apres deplacement :




        *
--------------------------------
Process exited after 0.0671 seconds with return value 0
Press any key to continue . . .
```

Exercice 5 :

```
using namespace std;
class Point
{
private:
    int x, y;
public:
    Point(int _x = 0, int _y = 0)
    {
        this->x = _x;
        this->y = _y;
    }
    ~Point()
    {
    cout <<"\ndernier point (" << x << ", " << y << ")" <<endl;
    }
    void afficher()
    {
        static HANDLE handle = NULL;
        COORD coords;
        coords.X = x;
        coords.Y = y;
        if (!handle)
        handle = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleCursorPosition(handle, coords);
        SetConsoleTextAttribute(handle, 3);
        cout << "*";
        SetConsoleTextAttribute(handle, 7);
    }
    void effacer()
    {
        x = 0;
        y = 0;
    }
    void deplacer(int dx, int dy)
    {
        effacer();
        x = dx;
        y = dy;
    }

};
void scene()
{
    Point u(0.0);
    u.afficher();
    cout << "\nle point apres deplacement :" << endl;
    u.deplacer(8, 9);
    u.afficher();
}
int main()
{
   scene();
}
```

```
*
le point apres deplacement :




          *
dernier point (8, 9)
```

Exercice 6 :

```cpp
#include <iostream>
using namespace std;

class SuiteAr
{
private:
    int nbr_de_termes;
    int *Val;

public:
SuiteAr(int nb, int Nul)
    {
        nbr_de_termes = nb;
        Val = new int[nbr_de_termes];

        for (int i = 0; i < nbr_de_termes; i++)
            Val[i] = i * Nul;
    }

~SuiteAr()
    {
        delete[] Val;
    }

    void afficher()
    {
        cout << "Les " << nbr_de_termes << " premiers termes de la suite sont : ";
        for (int i = 0; i < nbr_de_termes; i++)
        {
            cout << Val[i] << "\t";
        }
        cout << endl;
    }
};

int main()
{
    int nbr_de_termes, raison;

    cout << "Saisir la raison : ";
    cin >> raison;
    cout << "Saisir nbr de termes : ";
    cin >> nbr_de_termes;
    SuiteAr suite(nbr_de_termes, raison);
    suite.afficher();
}
```

```
Saisir la raison : 6
Saisir nbr de termes : 7
Les 7 premiers termes de la suite sont : 0      6       12      18      24      30      36

--------------------------------
Process exited after 2.457 seconds with return value 0
```

Exercice 7 :

1.

```
using namespace std;

class hasard{
     private:
          int val[10];
     public:
     hasard(int _max){
     srand(time(NULL));
     for(int i = 0; i < 10; i++)
          val[i] = rand()%_max;
}

void afficher(){
     cout <<" Les valeurs sont: ";
     for(int i = 0; i < 10; i++){
          cout <<val[i] <<"\t";
     }
     cout <<endl;
}
};


int main(){
     int _max;
     cout << "Saisir max =  ";
     cin >> _max;
     hasard hasard(_max);
     hasard.afficher();
}
```

2.

```
using namespace std;

class hasard
{
private:
     int *val;
     int size;

public:
     hasard(int _max, int _size)
     {
          this->size = _size;
          val = new int[_size];
          srand(time(NULL));
          for (int i = 0; i < 10; i++)
               val[i] = rand() % _max;
     }

     void afficher()
     {
          cout << " Les valeurs sont: ";
          for (int i = 0; i < size; i++)
          {
               cout << val[i] << "\t";
          }
          cout << endl;
     }
};

int main()
{
     int _max, _size;
     cout << "Saisir max =  ";
     cin >> _max;
     cout << "Saisir size =  ";
     cin >> _size;
     hasard hasard(_max, _size);
     hasard.afficher();
}
```

3 .

```
using namespace std;

class hasard
{
private:
    int *val;
    int size;

public:
    hasard(int _max, int _size)
    {
        this->size = _size;
        val = new int[_size];
        srand(time(NULL));
        for (int i = 0; i < 10; i++)
            val[i] = rand() % _max;
    }
    ~hasard(){
        delete[] val;
    }

    void afficher()
    {
        cout << " Les valeurs sont: ";
        for (int i = 0; i < size; i++)
        {
            cout << val[i] << "\t";
        }
        cout << endl;
    }
};

int main()
{
    int _max, _size;
    cout << "Saisir max =  ";
    cin >> _max;
    cout << "Saisir size =  ";
    cin >> _size;
    hasard hasard(_max, _size);
    hasard.afficher();
}
```

Exercice 8 :

```
using namespace std;

class Complexe
{
private:
    double pr, pi;

public:

void initialiser(double _pr, double _pi)
{
    this->pr = _pr;
    this->pi = _pi;
}

double calculerModule()
{
    return sqrt(pr * pr + pi * pi);
}

void afficher()
{
    cout << "\tLe nombre : " << pr << " + " << "i" << pi  << endl;
}
Complexe(double _pr, double _pi)
{
    this->pr = _pr;
    this->pi = _pi;
}

Complexe()
{
}

Complexe(double _pr)
{
    this->pr = _pr;
    this->pi = 0;
}
};
```

```cpp
int main()
{

    Complexe nbr1;
    Complexe nbr2(3.4);
    Complexe nbr3(1.3, 2.3);

    cout << "1ere nombre complexe : " << endl;
    nbr1.initialiser(2, 3);
    double module1 = nbr1.calculerModule();

    cout << "2eme nombre complexe : " << endl;
    double module2 = nbr2.calculerModule();

    cout << "3eme nombre complexe : " << endl;
    double module3 = nbr3.calculerModule();

    cout << "\nnbr1 =" ;
    nbr1.afficher() ;
    cout << " son module = " << module1 << endl;
    cout << "\nnbr2 =" ;
    nbr2.afficher();
    cout  << " son module = " << module2 << endl;
    cout << "\nnbr3 =" ;
    nbr3.afficher() ;
    cout << " son module = " << module3 << endl;


}
```