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

COMPTE RENDU DE CHAPITRE III + TPS

Réalisé par :
Abderrahmane ETTOUNANI
II-BDCC

Encadré par :
Monsieur K.MANSOURI

Ecole Normale Supérieure de l'Enseignement Technique
BP 159 Bd Hassan II Mohammedia
Tél. : 05 23 32 22 20 - 05 23 32 35 30
Fax : 05 23 32 25 46

Année universitaire : 2021 - 2022

exercices et exemples de cours

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

You, 7 hours ago | 1 author (You)
class Point
{
    int x, y;
      You, 7 hours ago • add exempl
public:
    Point();
    Point(int);
    Point(int, int);
    void affiche();
    void affiche(char *);
};
Point ::Point()
{
    x = 0;
    y = 0;
}

Point ::Point(int abs)
{
    x = abs;
    y = abs;
}

Point ::Point(int abs, int ord)
{
    x = abs;
    y = ord;
}
void Point ::affiche()
{
    cout << " je suis en " << x << " " << y;
}
```

```cpp
void Point ::affiche(char *message)
{
    cout << message << endl;
}

int main()
{
    Point a;
    a.affiche();
    Point b(5);
    b.affiche("point b");
    Point c(3, 12);
    c.affiche("point c");

    return 0;
}
```

```
je suis en 0 0
point b
je suis en 5 5
point c
je suis en 3 12
```

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

You, 3 minutes ago | 1 author (You)
class Point
{
    int x, y;

public:
    Point()
    {
        x = 0;
        y = 0;
    }
    Point(int abs)
    {
        x = abs;
        y = abs;
    }
    Point(int abs, int ord)
    {
        x = abs;
        y = ord;
    }
    void affiche()
    {
        cout << " je suis en " << x << " " << y << endl;
    }       You, 7 hours ago • add exemple2 …
    void affiche(char *message)
    {
        cout << message << endl;
        cout << " je suis en " << x << " " << y << endl;
    }
};
```

"ex1.o" selected (4.7 kB) < "ex2.o" selected (6.0 kB)

```cpp
#include <iostream>      You, 7 hours ago • add exemple3 …
using namespace std;
You, 7 hours ago | 1 author (You)
class point
{
    int x, y;

public:
    point(int abs = 0, int ord = 2)
    {
        x = abs;
        y = ord;
    }                                          // constructeur
    void affiche(char * = "Position du point"); // argument par défaut
};
void point::affiche(char *message)
{
    cout << message;
    cout << "le point est en " << x << " " << y << "\n";
}
int main()
{
    point a, b(40);
    a.affiche();
    b.affiche("Point b");
    char texte[10] = "Bonjour";
    point c(3, 12);
    c.affiche(texte);
    return 0;
}
```

```
Position du point
le point est en 0 2
Point b
le point est en 40 2
Bonjour
le point est en 3 12
```

```cpp
#include <iostream>
using namespace std;
You, 7 hours ago | 1 author (You)
class point
{
    int x, y;

public:
    point(int abs = 0, int ord = 2)
    {
        x = abs;
        y = ord;
    } // constructeur
    int coincide(point);
};
int point::coincide(point pt)
{
    if ((pt.x = x) && (pt.y = y))
        return 1;
    else
        return 0;
}
int main()
{
    int test1, test2;      You, 7 hours ago • add exemple4 …
    point a, b(1), c(0, 2);
    test1 = a.coincide(b);
    test2 = b.coincide(a);
    cout << "a et b:" << test1 << " ou " << test2 << "\n";
    test1 = a.coincide(c);
    test2 = c.coincide(a);
    cout << "a et c: " << test1 << " ou " << test2 << "\n";
    return 0;
}
```

```
a et b : 0 ou 1
a et c : 0 ou 0
```

```cpp
class vecteur
{
    float x, y;
public:
    vecteur(float, float);
    void homotethie(float);
    void affiche();
    float det(vecteur);
};      You, 7 hours ago • add exemple 5 part-1 …
vecteur::vecteur(float abs = 0, float ord = 0)
{
    x = abs;

    y = ord;
}
void vecteur ::homotethie(float val)
{
    x = x * val;
    y = y * val;
}
void vecteur::affiche()
{
    cout << "x = " << x << " y = " << y << "\n";
}
float vecteur::det(vecteur v)
{
    float res;
    res = x * v.y - y * v.x;
    return res;
}
int main()
{
    vecteur v1(2.3, 5.4), v2(1.3, 2.2);
    v1.affiche();
    v2.affiche();
    cout << "determinant est : " << v1.det(v2);
    return 0;
}
```

result :

```
x = 2.3 y = 5.4
x = 1.3 y = 2.2
determinant est : -1.96
```

passage par adresse

```cpp
float det(vecteur *);
```

```cpp
float vecteur::det(vecteur *v)
{
    float res;
    res = x * v->y - y * v->x;
    return res;
}
```

```cpp
v1.det(&v2);
```

passage par reference

```cpp
float det(vecteur &);
```

```cpp
float vecteur::det(vecteur &v)
{
    float res;
    res = x * v.y - y * v.x;
    return res;
}
```

```cpp
#include <iostream>      You, 7 hours ago • add exemple
using namespace std;

// La valeur de retour d'une fonction est un objet
// Transmission par valeur
You, 7 hours ago | 1 author (You)
class point
{
    int x, y;

public:
    point(int abs = 0, int ord = 0)
    {
        x = abs;
        y = ord;
    } // constructeur
    point symetrique();
    void affiche();
};
point point ::symetrique()
{
    point res;
    res.x = -x;
    res.y = -y;
    return res;
}
void point ::affiche()
{
    cout << "Le point est en " << x << " et "
         << " " << y << "\n";
}
void main()
{
    point a, b(1, 6);
    a = b.symetrique();
    a.affiche();
    b.affiche();
}
```

```
Le point est en -1 et  -6
Le point est en 1 et  6
```

par adresse

```cpp
point *point ::symetrique()
{
    point *res;
    res = new point;
    res->x = -x;
    res->y = -y;
    return res;
}
```

par refference

```cpp
point &point ::symetrique()
{
    static point res; // static est obligatoir
    res.x = -x;
    res.y = -y;
    return res;
}
```

```cpp
class compte_objet
{
    static int ctr;
public:
    compte_objet();
    ~compte_objet();
    static void compte();
};
compte_objet ::compte_objet()
{
    cout << "++construction : il y a m
         << ++ctr
         << " objets\n ";
}
compte_objet ::~compte_objet()
{
    cout << " --desruction : il y a maintenant " << --ctr << " objets\n ";
}
void fonction()
{
    compte_objet u, v;
}
void main()
{
    void fonction();
    compte_objet ::compte();
    compte_objet a;
    compte_objet ::compte();
    fonction();
    compte_objet ::compte();
    compte_objet b;
    compte_objet ::compte();
}
```

```
0
++construction : il y a maintenant 1 objets
 1
++construction : il y a maintenant 2 objets
 ++construction : il y a maintenant 3 objets
 --desruction : il y a maintenant 2 objets
 --desruction : il y a maintenant 1 objets
 1
++construction : il y a maintenant 2 objets
 2
--desruction : il y a maintenant 1 objets
 --desruction : il y a maintenant 0 objets
```

```cpp
#include <iostream>      You, 8 hours ago • add exemple 10 …
using namespace std;

You, 8 hours ago | 1 author (You)
class point
{
    int x, y;

public:
    point(int abs = 0, int ord = 0)
    {
        x = abs;
        y = ord;
    } // constructeur en ligne
    void affiche();
    int coincide(point *);
};
void point ::affiche()
{
    cout << "Adresse : " << this << " - Coordonnees " << y << x << endl;
}
int point::coincide(point *adpt)
{
    if ((this->x == adpt->x) && (this->y == adpt->y))
        return 1;
    else
        return 0;
}
int main()
{
    point a(5), b(3, 15);
    a.affiche();
    b.affiche();
    return 0;
}
```

```
Adresse : 0x7ffe8553cdd8 - Coordonnees 05
Adresse : 0x7ffe8553cde0 - Coordonnees 153
```

```cpp
#include <iostream>      You, 8 hours ago • add exemple
using namespace std;

You, 8 hours ago | 1 author (You)
class vecteur
{
    float x, y;

public:
    vecteur(float, float);
    void homotethie(float);
    void affiche();
    float det(vecteur);
    float prod_scal(vecteur);
    vecteur somme(vecteur);
};

vecteur::vecteur(float abs = 0, float ord = 0)
{
    x = abs;

    y = ord;
}
void vecteur ::homotethie(float val)
{
    x = x * val;
    y = y * val;
}
void vecteur::affiche()
{
    cout << "x = " << x << " y = " << y << "\n";
}
float vecteur::det(vecteur v)
{
    float res;
    res = x * v.y - y * v.x;
    return res;
}
```

```cpp
float vecteur::prod_scal(vecteur v)
{
    float res;
    res = x * v.x + y * v.y;
    return res;
}
vecteur vecteur::somme(vecteur v)
{
    vecteur res;
    res.x = x + v.x;
    res.y = y + v.y;
    return res;
}

int main()
{
    vecteur v1(2.3, 5.4), v2(1.3, 2.2);
    v1.affiche();
    v2.affiche();
    cout << "determinant est : " << v1.det(v2);
    cout << "produit scalaire est : " << v1.prod_scal(v2);
    vecteur v3;
    v3 = v1.somme(v2);
    v3.affiche();
    return 0;
}
```

```
x = 2.3 y = 5.4
x = 1.3 y = 2.2
determinant est : -1.96
produit scalaire est : 14.87
x = 3.6 y = 7.6
```
