

02/04/2022

Compte rendu

Des travaux pratiques de la
chapitre 2

**Programmation orientée
objet : Notion de classe**

PRÉPARÉE PAR
TAFFAH ACHRAF

ENCADRÉ PAR
PR . KHALIFA MANSOURI

Filières d'ingénieur : Génie du Logiciel et des Systèmes Informatiques Distribués



TABLE DES MATIERES

Introduction	3
Cadre général	3
Partie pratique	4
1. Exemples de cour :	4
REMARQUE :	4
EXEMPLE 1 :	4
2. Exercices :	5
Exercice 1 :	5
Exercice 2 :	6
Exercice 3 :	7
Exercice 4 :	8
Exercice 5 :	9
Exercice 6 :	10
Exercice 7 :	11
Exercice 8 :	12

INTRODUCTION

IL EST IMPORTANT D'APPRENDRE LA PROGRAMMATION PROCEDURALE AVANT D'APPRENDRE LA PROGRAMMATION ORIENTEE OBJET PARCE QUE LA PROGRAMMATION MODULAIRE QUI EST GENERALEMENT MELANGEE ET CONFONDUE AVEC LA PROGRAMMATION PROCEDURALE, PEUT ETRE APPLIQUEE A LA POO. PAR CONSEQUENT, EN PLUS LA PLUPART DES LANGAGES DE PROGRAMMATION SONT MULTIPARADIGME, A UN CERTAIN NIVEAU, MEME SI LEURS CONCEPTEURS OU DEVELOPPEURS COMMUNS DISENT LE CONTRAIRE, ET AUSSI PARCE QUE LA POO EST PLUS COMPLEXE QUE LA PROGRAMMATION PROCEDURALE, IL EST DONC PREFERABLE D'APPRENDRE LA PROGRAMMATION PROCEDURALE EN PREMIER.

CADRE GENERAL

Sujet de notre travaux pratique	Programmation orientée objet : notion de classe
La date de réalisation	02/04/2022
Date de début	28/03/2022
Date de fin	02/04/2022
Destinataire	Les étudiants de la première année Génie du Logiciel et des Systèmes Informatiques Distribuées
Encadrant de travaux pratique	Pr. KHALIFA Mansouri

PARTIE PRATIQUE

1. Exemples de cour :

REMARQUE :

J'ai remarqué que tous les exemples de cour sont les mêmes exercices sauf exemple de "Compteur", et pour cela je n'ai fait que les exercices et l'exemple ladite.

EXEMPLE 1 :

```
#include<iostream>
#include<conio.h>
using namespace std;
class compteur {
    static int ctr = 0 ;
public :
    compteur();
    ~compteur();
};

compteur :: compteur(){
    cout<< "Un objet vient de se detruire : " << "\n";
    cout<< "Il reste maintenant : " << ++ctr<< "Objets";
    getch();
}

compteur :: ~compteur(){
    cout<< "Un objet vient de se detruire : " << "\n";
    cout<< "Il reste maintenant : " << --ctr<< "Objets";
    getch();
}

int main(){
    void Essai();
    compteur a ;
    Essai();
    compteur b;
}

void Essai(){
    compteur u,v;
}
```

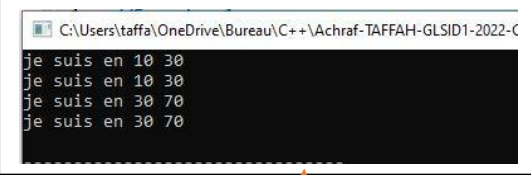
L'affichage après la compilation de ce programme

```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE2\Cour\exemple.exe
un nouvel objet vient de sa creer
il y a maintenant : 1 objets  un nouvel objet vient de sa creer
il y a maintenant : 2 objets  un objet vient de se detruire
il reste maintenant : 1 objets  un objet vient de se detruire
il reste maintenant : 0 objets  un nouvel objet vient de sa creer
il y a maintenant : 1 objets  un nouvel objet vient de sa creer
il y a maintenant : 2 objets  un nouvel objet vient de sa creer
il y a maintenant : 3 objets  un objet vient de se detruire
il reste maintenant : 2 objets  un objet vient de se detruire
il reste maintenant : 1 objets  un nouvel objet vient de sa creer
il y a maintenant : 2 objets  un objet vient de se detruire
il reste maintenant : 1 objets  un objet vient de se detruire
il reste maintenant : 0 objets
```

2. Exercices :

EXERCICE 1 :

```
//Exercice 1
#include <iostream>
using namespace std;
class Point{
    int x, y, couleur;
public:
    void initialiser(int, int, int);
    void deplacer(int, int);
    void afficher();
    void effacer();
};
void Point::initialiser(int abs, int ord, int c){
    x = abs;
    y = ord;
    couleur = c;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 1;
    afficher();
    couleur = aux;
}
int main(){
    Point p1;
    p1.initialiser(10, 30, 1);
    p1.afficher();
    p1.deplacer(20, 40);
    p1.effacer();
    return 0;
}
```

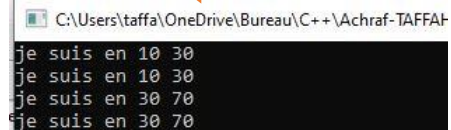


L'affichage après la compilation de ce programme

EXERCICE 2 :

```
//Exercice 2
#include <iostream>
using namespace std;
class Point{
    int x, y, couleur;
public:
    void initialiser(int, int, int);
    void deplacer(int, int);
    void afficher();
    void effacer();
};
void Point::initialiser(int abs, int ord, int c){
    x = abs;
    y = ord;
    couleur = c;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}
void scene(){
    Point u;
    u.initialiser(10, 30, 1);
    u.afficher();
    u.deplacer(20, 40);
    u.effacer();
}
int main(){
    scene();
    return 0;
}
```

L'affichage après la
compilation de ce
programme

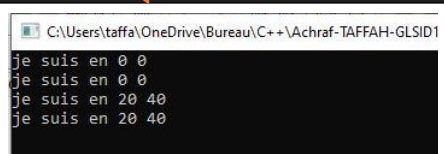


```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFA\
je suis en 10 30
je suis en 10 30
je suis en 30 70
je suis en 30 70
```


EXERCICE 3 :

```
//Exercice 3
#include <iostream>
using namespace std;
class Point{
    int x, y, couleur;
public:
    Point();
    void deplacer(int, int);
    void afficher();
    void effacer();
};
Point::Point(){
    x = 0;
    y = 0;
    couleur = 0;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}
void scene(){
    Point u;
    u.afficher();
    u.deplacer(20, 40);
    u.effacer();
}
int main(){
    scene();
    return 0;
}
```

L'affichage après la
compilation de ce
programme



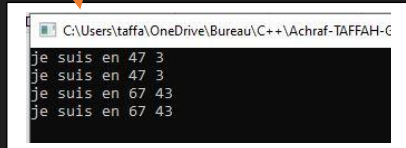
```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1
je suis en 0 0
je suis en 0 0
je suis en 20 40
je suis en 20 40
```

EXERCICE 4 :

```
//Exercice 4
#include <iostream>
using namespace std;
class Point{
    int x, y, couleur;

public:
    Point(int, int, int);
    void deplacer(int, int);
    void afficher();
    void effacer();
};
Point::Point(int abs, int ord, int c){
    x = abs;
    y = ord;
    couleur = c;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}
void scene(){
    Point u(47, 3, 1);
    u.afficher();
    u.deplacer(20, 40);
    u.effacer();
}
int main(){
    scene();
    return 0;
}
```

L'affichage après la compilation de ce programme



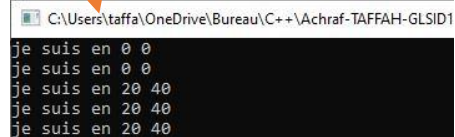
```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-C
je suis en 47 3
je suis en 47 3
je suis en 67 43
je suis en 67 43
```


EXERCICE 5 :

```
//Exercice 5
#include <iostream>
using namespace std;
class Point{
    int x, y, couleur;
public:
    Point();
    void deplacer(int, int);
    void afficher();
    void effacer();
    ~Point();
};
Point::Point(){
    x = 0;
    y = 0;
    couleur = 0;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}
Point::~~Point(){
    afficher();
}
}
void scene(){
    Point u;
    fficher(); u
    deplacer(20, 40); u
    effacer(); }

int main(){
    scene();
    return 0;
}
```

L'affichage après la
compilation de ce
programme



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1
je suis en 0 0
je suis en 0 0
je suis en 20 40
je suis en 20 40
je suis en 20 40
```

EXERCICE 6 :

```
//Exercice 6
#include <iostream>
#include <stdlib.h>
using namespace std;

class SuitAr
{
    int nbval, *val;

public:
    SuitAr(int, int);
    ~SuitAr();
    void afficher();
};

SuitAr::SuitAr(int nb, int mul)
{
    int i;
    nbval = nb;
    val = new int[nbval];
}

SuitAr::~SuitAr()
{
    cout << "Destructeur..."<<endl;
}

int main()
{
    SuitAr ar(2, 4);

    return 0;
}
```

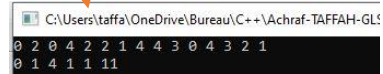
L'affichage après la
compilation de ce
programme

C:\Users\taffa\OneDrive\Bureau\C++\Achraf-1
Destructeur...

EXERCICE 7 :

```
//Exercice 7
#include <iostream>
#include <stdlib.h>
using namespace std;
class Hasard{
    int nbval;
    int *val;
public:
    Hasard(int, int);
    ~Hasard();
    void affiche();
};
Hasard::Hasard(int nb, int max){
    int i;
    val = new int[nbval = nb];
    for (i = 0; i < nb; i++)
        val[i] = double(rand()) / RAND_MAX * max;
}
Hasard::~Hasard(){
    delete val;
}
void Hasard::affiche(){
    int i;
    for (i = 0; i < nbval; i++)
        cout << val[i] << " ";
    cout << "\n";
}
int main(){
    Hasard suite1(15, 5);
    suite1.affiche();
    Hasard suite2(6, 12);
    suite2.affiche();
    return 0;
}
```

L'affichage après la
compilation de ce
programme



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLT
0 2 0 4 2 2 1 4 4 3 0 4 3 2 1
0 1 4 1 1 11
```

EXERCICE 8 :

```
//Exercice 8
#include <iostream>
#include <stdlib.h>
#include <math.h>
using namespace std;
class Complexe{
private:
    double x, y;
public:
    Complexe();
    Complexe(double, double);
    Complexe(double a);
    void set(double, double);
    double module();
    void affiche();
    double real() const;
    double img() const;
};
void Complexe::set(double a, double b){
    x = a;
    y = b;
}
double Complexe::module(){
    double res;
    res= sqrt(x * x + y * y);
    return res; }

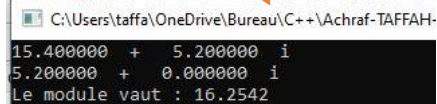
void Complexe::affiche(){
    printf("%f + %f i\n", x, y);
}
Complexe::Complexe(double a, double b){
    x = a;
    y = b;
}
Complexe::Complexe(double a){
    x = a;
    y = 0;
}

Complexe::Complexe()
{
    x = 0;
    y = 0;
}
int main()
{
    Complexe c1(15.4, 5.2);
    c1.affiche();

    Complexe c2(5.2);
    c2.affiche();

    double res = c1.module();
    cout << "Le module vaut : " << res << endl;
    return 0;
}
```

L'affichage après la
compilation de ce
programme



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-
15.400000 + 5.200000 i
5.200000 + 0.000000 i
Le module vaut : 16.2542
```