

ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE
DE MOHAMMEDIA

➤ UNIVERSITÉ HASSAN II DE CASABLANCA

TP5 : programmation orientée objet en c++

ETUDIANT DE FILIER GLSID 1
ENSET

➤ ENCADRÉ PAR:
M.K.MANSOURI

➤ REALISER PAR :
ZAKARIA EL MOURTAZAK

PARTIE 1

LE CODE

```
h partie1.h > ...
1  #include <iostream>
2  class coordonne
3  {
4  private:
5      int x, y;
6  public:
7      coordonne(int,int);
8      void deplace(int, int);
9      void affiche();
10 };
11 coordonne::coordonne(int a = 0, int b = 0)
12 {
13     x = a;
14     y = b;
15 }
16 void coordonne::deplace(int a, int b)
17 {
18     x += a;
19     y += b;
20 }
21 }
22 void coordonne::affiche()
23 {
24     std::cout << "x= " << x << " y = " << y << std::endl;
25 }
26 class forme
27 {
28 protected:
29     short couleur;
30 public:
31     forme(short);
32     forme(forme &);
33     void affiche();
34     forme operator=(forme &);
35 };
36 forme::forme(short couleur = 1)
37 {
38     this->couleur = couleur;
39 }
40 forme::forme(forme &other)
41 {
42     this->couleur = other.couleur;
43 }
44 void forme::affiche()
45 {
46     std::cout << "couleur = " << couleur << std::endl;
47 }
48
49 forme forme::operator=(forme &other)
50 {
51     forme f(other.couleur);
52     return f;
53 }
```

PARTIE 2

LE CODE

```
1  #include "partiel.h"
2  class cercle : public forme
3  {
4  protected:
5      short rayon;
6      coordonne centre;
7
8  public:
9      cercle() = default;
10     cercle(int, int, short, short);
11     cercle(cercle &);
12     cercle operator=(cercle &);
13     void affiche();
14     void deplace(int, int);
15     int surface;
16     int perimetre;
17 };
18
19 cercle::cercle(int x, int y, short couleur, short rayon) : forme(couleur)
20 {
21     coordonne Centre(x, y);
22     centre = Centre;
23     this->rayon = rayon;
24     surface = 0;
25     perimetre = 0;
26 }
27
28 cercle::cercle(cercle& Cercle) : forme(Cercle.couleur)
29 {
30     centre = Cercle.centre;
31     rayon = Cercle.rayon;
32     surface = 0;
33     perimetre = 0;
34 }
35
36 cercle cercle::operator=(cercle &Cercle)
37 {
38     cercle cercleCrr(Cercle);
39     return Cercle;
40 }
```

```

41
42 void cercle::affiche()
43 {
44     std::cout << "rayon = " << rayon << std::endl;
45     centre.affiche();
46     forme::affiche();
47     std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;
48 }
49
50 void cercle::deplace(int x, int y)
51 {
52     centre.deplace(x,y);
53 }
54
55 int main(int argc, char const *argv[])
56 {
57     cercle cl(10, 20, 5, 12);
58     cl.surface = 2;
59     cl.perimetre = 10;
60     cl.affiche();
61     std::cout << "-----" << std::endl;
62     cl.deplace(2, 4);
63     cl.affiche();
64     std::cout << "-----" << std::endl;
65     getchar();
66
67     return 0;
68 }

```

L'EXECUTION

```

rayon = 12
x= 10 y = 20
couleur = 5
surface = 2 perimetre = 10
-----

```

```

rayon = 12
x= 12 y = 24
couleur = 5
surface = 2 perimetre = 10
-----

```

PARTIE 3

LE CODE

```
1  #include "partie1.h"
2
3  class triangle : public forme
4  {
5  protected:
6      coordonne a, b, c;
7
8  public:
9      triangle(int, int, int, int, int, int, short);
10     triangle(triangle &);
11     triangle operator=(triangle &);
12     void affiche();
13     void deplace(int, int);
14     int surface;
15     int perimetre;
16 };
17
18 triangle::triangle(int x1, int y1, int x2, int y2, int x3, int y3, short couleur) : forme(couleur)
19 {
20     a = coordonne(x1, y1);
21     b = coordonne(x2, y2);
22     c = coordonne(x3, y3);
23     surface = 0;
24     perimetre = 0;
25 }
26
27 triangle::triangle(triangle &Triangle)
28 {
29     a = Triangle.a;
30     b = Triangle.b;
31     c = Triangle.c;
32     surface = 0;
33     perimetre = 0;
34 }
35
36 triangle triangle::operator=(triangle &Triangle)
37 {
38     triangle triangleCrr(Triangle);
39     return triangleCrr;
40 }
41
42 void triangle::affiche()
```

```
triangle triangle::operator=(triangle &Triangle)
{
    triangle triangleCrr(Triangle);
    return triangleCrr;
}

void triangle::affiche()
{
    std::cout << "les coordonne de a : " << std::endl;
    std::cout << "-----" << std::endl;
    a.affiche();
    forme::affiche();
    std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;

    std::cout << "-----" << std::endl;

    std::cout << "les coordonne de b : " << std::endl;
    std::cout << "-----" << std::endl;
    b.affiche();
    forme::affiche();

    std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;

    std::cout << "-----" << std::endl;

    std::cout << "les coordonne de c : " << std::endl;
    std::cout << "-----" << std::endl;
    c.affiche();
    forme::affiche();

    std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;

    std::cout << "-----" << std::endl;
}

void triangle::deplace(int x, int y)
{
    a.deplace(x, y);
    b.deplace(x, y);
    c.deplace(x, y);
}

int main(int argc, char const *argv[])
{
    triangle T(10, 20, 30, 40, 50, 50, 11);
    T.affiche();
    getchar();
    T.deplace(5, 4);
    T.affiche();
    return 0;
}
```

L'EXECUTION

les coordonne de a :

x= 10 y = 20

couleur = 11

surface = 0 perimetre = 0

les coordonne de b :

x= 30 y = 40

couleur = 11

surface = 0 perimetre = 0

les coordonne de c :

x= 50 y = 50

couleur = 11

surface = 0 perimetre = 0



PARTIE 4

LE CODE

```
1  #include "partiel.h"
2
3
4  class rectangle : public forme
5  {
6  protected:
7      coordonne a, b, c, d;
8
9  public:
10     rectangle(int, int, int, int, int, int, int, int, short);
11     rectangle(rectangle &);
12     rectangle operator=(rectangle &);
13     void affiche();
14     void deplace(int, int);
15     int surface;
16     int perimetre;
17 };
18
19 rectangle::rectangle(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4, short couleur) : forme(couleur)
20 {
21     a = coordonne(x1, y1);
22     b = coordonne(x2, y2);
23     c = coordonne(x3, y3);
24     d = coordonne(x4, y4);
25     surface = 0;
26     perimetre = 0;
27 }
28
29 rectangle::rectangle(rectangle &Rectangle)
30 {
31     a = Rectangle.a;
32     b = Rectangle.b;
33     c = Rectangle.c;
34     d = Rectangle.d;
35     surface = 0;
36     perimetre = 0;
37 }
```



```

39 void rectangle::affiche()
40 {
41     std::cout << "les coordonne de a :" << std::endl;
42     std::cout << "-----" << std::endl;
43     a.affiche();
44     forme::affiche();
45     std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;
46
47     std::cout << "-----" << std::endl;
48
49     std::cout << "les coordonne de b :" << std::endl;
50     std::cout << "-----" << std::endl;
51     b.affiche();
52     forme::affiche();
53
54     std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;
55
56     std::cout << "-----" << std::endl;
57
58     std::cout << "les coordonne de c :" << std::endl;
59     std::cout << "-----" << std::endl;
60     c.affiche();
61     forme::affiche();
62
63     std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;
64
65     std::cout << "-----" << std::endl;
66     std::cout << "les coordonne de d :" << std::endl;
67     std::cout << "-----" << std::endl;
68     d.affiche();
69     forme::affiche();
70
71     std::cout << "surface = " << surface << " perimetre = " << perimetre << std::endl;
72
73     std::cout << "-----" << std::endl;
74 }
75
76 void rectangle::deplace(int x, int y)
77 {
78     a.deplace(x, y);
79     b.deplace(x, y);
80     c.deplace(x, y);
81     d.deplace(x, y);
82 }
83
84 rectangle rectangle::operator=(rectangle &Rectangle)
85 {
86     rectangle rectangleCrr(Rectangle);
87     return rectangleCrr;
88 }
89

```

```
int main(int argc, char const *argv[])
{
    rectangle R(10, 20, 30, 40, 50, 50, 11, 40, 50);
    R.affiche();
    getchar();
    R.deplace(5, 4);
    R.affiche();
    return 0;
}
```

L'EXECUTION

les coordonne de a :

x= 10 y = 20

```
couleur = 50
```

```
surface = 0  perimetre = 0
```

les coordonne de b :

$$x = 30 \quad v = 40$$

```
couleur = 50
```

```
surface = 0  perimetre = 0
```

les coordonne de c :

$x = 50 \quad v = 50$

```
couleur = 50
```

```
surface = 0  perimetre = 0
```

les coordonne de d :

x= 11 y = 40

```
couleur = 50
```

surface = 0 perimetre = 0

PARTIE 5

LE CODE

```
#include "partie4.h"

class carre : public rectangle
{
private:
    short cote;
public:
    carre(int, int, int, int, int, int, int, int, int, short, short);
    carre(carre &);
    carre operator=(carre &);
    void affiche();
    void deplace(int, int);
    int surface;
    int perimetre;
};

carre::carre(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4, short cote, short couleur) : rectangle(x1, y1, x2, y2, x3, y3, x4, y4, couleur)
{
    this->cote = cote;
    surface = 0;
    perimetre = 0;
}

carre::carre(carre &cr) : rectangle(cr)
{
    cote = cr.cote;
    surface = 0;
    perimetre = 0;
}

carre carre::operator=(carre &cr)
{
    carre crCrr(cr);
    return crCrr;
}

void carre::affiche()
{
    rectangle::affiche();
    std::cout << "cote = " << cote << std::endl;
}

void carre::deplace(int x, int y)
{
    rectangle::deplace(x, y);
}

int main(int argc, const char **argv)
{
    carre cr(12, 23, 34, 1, 2, 54, 23, 1, 23, 34);
    cr.affiche();
    cr.deplace(2, 3);
    cr.affiche();
    return 0;
}
```

L'EXECUTION

```
les coordonne de a :  
-----  
x= 12 y = 23  
couleur = 34  
surface = 0 perimetre = 0  
-----  
les coordonne de b :  
-----  
x= 34 y = 1  
couleur = 34  
surface = 0 perimetre = 0  
-----  
les coordonne de c :  
-----  
x= 2 y = 54  
couleur = 34  
surface = 0 perimetre = 0  
-----  
les coordonne de d :  
-----  
x= 23 y = 1  
couleur = 34  
surface = 0 perimetre = 0  
-----  
cote = 23  
--> apres d|@placement  
les coordonne de a :  
-----  
x= 14 y = 26  
couleur = 34  
surface = 0 perimetre = 0  
-----  
les coordonne de b :  
-----  
x= 36 y = 4  
couleur = 34  
surface = 0 perimetre = 0  
-----  
les coordonne de c :  
-----  
x= 4 y = 57  
couleur = 34  
surface = 0 perimetre = 0  
-----  
les coordonne de d :  
-----  
x= 25 y = 4  
couleur = 34  
surface = 0 perimetre = 0  
-----  
cote = 23
```

FIN.