

```

/*****
Cercle.h
*****/
#pragma once
#include <math.h>
#include "Point2D.h"
#include "Figure.h"

#define PI 3.14159265358979323846

class Point2D;
class Figure;

class Cercle : public Figure
{
private :
    double rayon;
    Point2D centre; //Composition forte

public:
    Cercle(Point2D leCentre, double leRayon);
    double getPerimetre();
    double getSurface();
};

/*****
Commande.h
*****/
#pragma once
#include <vector>
#include <string>
#include "Figure.h"

using namespace std;

class Figure;

class Commande
{
private:
    bool commandeTerminee;
    double prixMetreDecoupe , prixMetreCarreMatiere ;
    string idCommande;

public:
    Commande(string identifiantCommande , double lePrixMetreDecoupe , double lePrixMetreCarreMatiere);
    string getIdCommande() { return idCommande; }
    void ajouterNouvelleFigure(Figure*);
    void cloturerCommande();
    vector<Figure*> LesFigures;

    double getPrix() ;
};

/*****
Figure.h
*****/
#pragma once

class Figure
{
public:
    virtual double getPerimetre()=0;
    virtual double getSurface()=0;
};

/*****
Point2D.h
*****/
// Cette classe n'est pas à modifier
#pragma once

class Point2D
{
private:
    double x , y ;

public:
    Point2D(double x=0 , double y=0);
    double getX();

```

1 point

0,5 point

06 déc. 21 17:29

KOUADIO.cpp

Page 2/4

```

        double getY();
        void setX(double newX);
        void setY(double newY);
};

/*****
Polygone.h
*****/
#pragma once

#include <vector>
#include "Figure.h"
#include "Point2D.h"

using namespace std;

class Point2D;
class Figure;

#define abs(x) ( (x) >=0 ? (x) : -(x) )

class Polygone : public Figure
{
protected:
    vector<Point2D *> lesSommets;
    bool estFerme;

public:
    Polygone(void);
    static double distance(Point2D P1 , Point2D P2 );
    void insereUnNouveauSommet(Point2D*, int position = -1);
    void fermeLePolygone();
    double getPerimetre();
    double getSurface();
};

/*****
Cercle.cpp
*****/
#include "Cercle.h"

Cercle::Cercle(Point2D leCentre, double leRayon)
{
    this->centre = leCentre;
    this->rayon = leRayon;
}

double Cercle::getPerimetre()
{
    return 2*PI*rayon;
};

double Cercle::getSurface()
{
    return (rayon * rayon) * PI;
}

/*****
Commande.cpp
*****/
#include "Commande.h"

Commande::Commande(string identifiantCommande , double lePrixMetreDecoupe , double lePrixMetreCarreMatiere)
{
    .....
}

void Commande::ajouterNouvelleFigure(.....)
{
    .....
}

void Commande::cloturerCommande()
{
    .....
}

```

1 point

1 point

```

double Commande::getPrix()
{
    .....
}

/*****
main.cpp
*****/
#include <iostream>
#include <conio.h>
// #include "Polygone.h"
#include "Cercle.h"
// #include "Commande.h"

using namespace std ;                // espace de nommage standard

int main()
{
    // Testez la classe Cercle
    Cercle Deis(Point2D(2,8), 12.4);
    cout << "le cercler est" << Deis.getPerimetre() << endl;
    cout << "Sa surface est" << Deis.getSurface() << endl;

    // Testez la classe Polygone avec la figure de test du sujet
    double Coordonnees[6][2]={ { 1 , 1 } , { 3 , 5 } , { 5 , 7 } , { 5 , 1 } , { 3 , 3 } , { 3 , 1 } };

    // Sapin de Noel et boules
    double CoordonneesSapin[15][2]={ { 2 , 2 } , { 5 , 4 } , { 3 , 4 } , { 5 , 6 } , { 4 , 6 } , { 6 , 8 } ,
    { 8 , 6 } , { 7 , 6 } , { 9 , 4 } , { 7 , 4 } , { 10 , 2 } , { 6
    .5 , 2 } , { 6.5 , 1 } , { 5.5 , 1 } , { 5.5 , 2 } };

    double CoordonneesCentreCercles[6][2]={ { 2.5 , 3.5 } , { 3.5 , 5.5 } , { 4.5 , 7.5 } , { 7.5 , 7.5 }
    , { 8.5 , 5.5 } , { 9.5 , 3.5 } };
    int i;

    // Création du polygone sapin
    ...

    cout << "superficie du sapin = " << ... << " " ;
    cout << "Perimetre du sapin = " << ... << endl;

    // Création des 6 cercles
    ...
    cout << "superficie du cercle " << i << " = " << ... << " " ;
    cout << "Perimetre du cercle " << i << " = " << ... << endl;
    ...

    // Création de la commande du Père Noel
    ...

    // Ajout des figures (le sapin et les 6 cercles) à la commande
    ...

    // Affichage du prix de cette commande
    cout << "\n Cout de la commande : " << ... << " = " << ... << " euros" << endl;

    _getch();        // on attend l'appui sur une touche
    return 0 ;       // fin du programme
}

/*****
Point2D.cpp
*****/
// Cette classe n'est pas à modifier
#include "Point2D.h"

```

Test de la classe
Cercle : 0,5 point

```

Point2D::Point2D(double x , double y)
{
    this->x = x;
    this->y = y;
}

double Point2D:: getX()
{ return x ;}

double Point2D::getY()
{
    return y;
}

void Point2D::setX(double newX)
{
    x = newX;
}

void Point2D::setY(double newY)
{
    y = newY;
}

/*****
Polygone.cpp
*****/
#include <math.h>
#include "Polygone.h"

Polygone::Polygone(void)
{
}

double Polygone::distance(Point2D P1, Point2D P2)
{
    return sqrt(distance(P2.getX() - P1.getX(), P2.getY() - P1.getY()));
};

void Polygone::insereUnNouveauSommet(Point2D*, int position = -1)
{
    .....
};

void Polygone::fermeLePolygone()
{
    .....
}

double Polygone::getPerimetre()
{
    return
};

double Polygone::getSurface()
{
    .....
}

```