

TD 2 : Classes et encapsulation

1. Disque

Un disque est caractérisé par un diamètre

- a. Ecrire une classe Disque qui permet de calculer et d'afficher les caractéristiques d'un disque de diamètre D et qui est définie par le diagramme de classe suivant :

Disque
-diametre : réel
+ Disque() + Disque(réel) + Disque(Disque) +getDiametre() : réel + setDiametre(réel) +perimetre() : réel + surface() : réel + rayon() : réel +afficher()

Cette classe possède :

- ✓ Une variable membre privé diametre
 - ✓ Des constructeurs
 - ✓ Une méthode modificatrice de diametre
 - ✓ Une méthode sélectrice pour retourner le diamètre
 - ✓ Des méthodes pour calculer le périmètre, la surface et le rayon
 - ✓ Une méthode pour afficher les caractéristiques du disque
- b. Ecrire une classe de test qui permet de :
- ✓ Créer 2 objets d1 et d2 de type Disque de diamètres d1(2.5), d2(5.6).
 - ✓ D'afficher les caractéristiques des 2 disques.

2. Classe CPoint3D :

Un point dans l'espace est définie par les coordonnées (x,y,z)

- a. Ecrire une classe CPoint3D qui permet de d'utiliser des points dans l'espace et qui est définie par le diagramme de classe suivant :

CPoint3D
-x, y, z : réel
+ CPoint3D() + CPoint3D(réel, réel, réel) + CPoint3D(CPoint3D) +getX() : réel +getY() : réel +getZ() : réel + setPoint(réel, réel, réel) +deplacer(réel, réel,réel) + distance() : réel + distance(CPoint3D) : réel + egal(CPoint3D) : boolean +afficher()

- ✓ La méthode distance permet de calculer la distance du point par rapport à l'origine O(0,0,0)
- ✓ La méthode distance(CPoint3D) permet de calculer la distance du point par rapport à un autre point reçu comme paramètre.
- ✓ La méthode egal permet de retourner true si le point est égal au point reçu comme paramètre.

Formule :

La distance euclidienne entre deux points P1(x1, y1, z1) et P2(x2, y2, z2) est définie par l'expression $d(P1, P2) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2}$

b. Ecrire une classe de test qui permet de :

- ✓ Créer 2 objets p1 et p2 de type CPoint3D de coordonnées p1(2, 5, 7), p2(5,9,16).
- ✓ D'afficher les caractéristiques des 2 points, ainsi que la distance qui les sépare.

3. Classe CSphere

a. Ecrire une classe CSphere qui permet de d'afficher les caractéristique d'une sphère et qui est définie par le diagramme de classe suivant :

CSphere
- centre : CPoint3D - rayon : réel
+ CSphere() + CSphere(réel, réel, réel, réel) + CSphere(CPoint3D, réel) + getRayon() : réel + getCentre() : CPoint3D + setSphere(CPoint3D, réel) + surface() : réel + volume() : réel + afficher()

$$S = 4\pi R^2$$

$$V = \frac{4}{3}\pi R^3$$

b. Ecrire une classe de test qui permet de :

- ✓ Créer un objet de type CSphere de rayon égale à 10 et de coordonnées pour le centre (2, 5, 7)
- ✓ D'afficher ses caractéristiques

4. Segment

On dispose de la classe CPoint3D pour manipuler les points dans l'espace :

a- Ecrire une classe Segment3D permettant de manipuler des segments dans l'espace. Un segment sera caractérisé par deux points : un point d'origine et un point d'extrémité.

Le diagramme de classe est le suivant :

Segment3D
- origine, extremite : CPoint3D
+ Segment(réel, réel, réel, réel, réel, réel) + Segment(CPoint3D, CPoint3D) + longueur() : réel + getOrigine() : CPoint3D + getExtremite() : CPoint3D + setSegment(CPoint3D, CPoint3D) + afficher()

b- Ecrire une classe de test qui permet de :

- ✓ Créer un segment s1 de caractéristiques : origine(2.5, 4, 3), extremite(13, 5.6, 2).

- ✓ D'afficher la longueur du segment.

5. Package MesUtilitaires

On suppose avoir à notre disposition une classe **LireClavier**, faisant partie d'un package MesUtilitaires. Cette classe possède une méthode **litChaine(message)** qui prend en argument une chaîne de caractères message et l'affiche à l'écran, puis elle lit une chaîne de caractères rentrée au clavier par l'utilisateur et la renvoie.

Pour nous aider voici la classe LireClavier :

```
Package MesUtilitaires;
import java.io.*;
public class LireClavier{
public static String litChaine(String message){
String ligneEntree="";
System.out.print(message + " ");
try{
    BufferedReader fluxEntree = new BufferedReader(new
InputStreamReader(System.in));
    ligneEntree = fluxEntree.readLine();
    if(ligneEntree.length()==0) return null;
}
catch (IOException e){
    System.out.println("IOException "+e);
}
return ligneEntree;
}
}
```

- Compiler la classe LireClavier du package MesUtilitaires
- En important cette classe, écrire un programme qui calcule la moyenne d'un ensemble de notes rentrées au clavier. Le programme demandera d'abord le nombre de notes à lire, puis demandera et récupérera les différentes notes rentrées par l'utilisateur. Le programme affichera la moyenne des notes ainsi que la proportion de notes supérieures à la moyenne.
- Ajouter la méthode statique LireFloat à la classe LireClavier, et modifier le programme de calcul de moyen.