

## TD JAVA

### Classes abstraites, interfaces et exceptions.

#### Exercice 1 :

Créer une classe `ObjetGeometrique2D` déclarant deux méthodes abstraites :

- `double perimetre()`
- `double surface()`

Créer deux sous-classes de `ObjetGeometrique2D` permettant de représenter un `Cercle` et un `Rectangle` (ces classes ne sont pas abstraites)

#### Exercice 2 :

Créer une interface `Zoomable` qui définit le comportement des objets pouvant être zoomés et déclarant les méthodes :

- `void agrandir(double facteur)`
- `void reduire(double facteur)`

Implémenter l'interface `Zoomable` pour que les `ObjetGeometrique2D` soit `Zoomable`.

#### Exercice 3 :

Créer une Classe `ObjetGeometrique3D` déclarant deux méthodes abstraites :

- `double surface()`
- `double volume()`

Créer deux sous-classes de `ObjetGeometrique3D` permettant de représenter un `Cylindre` et une `Sphère` (ces classes ne sont pas abstraites)

Rappel :

- la surface d'un cylindre :  $2 \cdot \text{PI} \cdot r \cdot h$
- le volume d'un cylindre :  $\text{PI} \cdot r^2 \cdot h$
- la surface d'une sphère :  $4 \cdot \text{PI} \cdot r^2$
- le volume d'une sphère :  $(4 \cdot \text{PI} \cdot r^3) / 3$

Implémenter l'interface `Zoomable` pour que les `ObjetGeometrique3D` soit `Zoomable`.

#### Exercice 4 :

Créer une classe `ObjetGeometrique2DSitue`, qui a les mêmes propriétés et comportements que `ObjetGeometrique2D` et qui dispose en plus des coordonnées de l'objet (en X et Y).

Créer deux sous-classes de `ObjetGeometrique2DSitue` permettant de représenter un `CercleSitue` et un `RectangleSitue` (ces classes ne sont pas abstraites)

#### Exercice 5 :

Créer l'interface `Mobile` qui définit le comportement des objets pouvant être déplacés et déclarant les méthodes :

- `void deplacerHorizontalement(double dep)`
- `void deplacerVerticalement(double dep)`

Implémenter l'interface `Mobile` pour que les `ObjetGeometrique2DSitue` soit mobiles.

#### Exercice 6 :

Modifier les méthodes `agrandir` et `reduire` définies par les objets implémentant l'interface `Zoomable` pour qu'elles lèvent une exception si le facteur d'agrandissement ou de réduction est inférieur ou égal à 0.

L'exception levée sera une `IllegalZoomFactorException`.

#### Exercice 7 :

Écrire un programme qui crée quelques exemplaires de chacun de ces objets et qui fait appel au moins une fois à chacune de leurs méthodes. En particulier, vous veillerez à faire des appels aux méthodes `agrandir` et `reduire` des objets `Zoomable`.