

# Travaux pratiques

## séance n°8

### Héritage et Polymorphisme

#### Exercice

**Question 1** : construire une classe représentant des cercles. Par défaut chaque instance de cercle a un rayon égal à 1. . Implanter les accesseurs de la variable d'instance (`getRayon` et `setRayon`) et une méthode de calcul de surface ( $\text{rayon} \times \text{rayon} \times \text{Math.PI}$ ).

**Question 2** : modifier la méthode `setRayon` de manière à gérer par une exception le cas où le rayon passé en paramètre est négatif.

**Question 3** : on peut considérer qu'un cylindre est un cercle dont l'épaisseur est quelconque. Construire une classe `Cylindre` qui dérive de la classe `Cercle`. Par défaut, sa hauteur vaut 1. Ajouter un constructeur qui initialise les variables d'instance, puis une méthode pour calculer son volume ( $2 \times \text{surface du cercle} + 2 \times \text{Math.PI} \times \text{hauteur}$ ). Quelle nouvelle visibilité pour la variable d'instance de `Cercle` ?

**Question 4** : écrire un programme de test qui crée une instance de cylindre, affiche sa hauteur, son rayon, son volume et la surface de son cercle.

**Question 5** : l'héritage permet la factorisation des membres d'une hiérarchie de classes. Vérifiez que vous avez utilisé toutes les possibilités de l'héritage (partage de variables, redéfinition de méthodes).

**Question 6** : la pseudo variable `this` est souvent implicite. Modifier la classe `Cylindre` de manière à la rendre explicite partout où c'est possible. On peut ainsi économiser les identificateurs.

**Question 7** : la pseudo variable `super` est souvent implicite. Modifier la classe `Cylindre` de manière à la rendre explicite.

**Question 8** : ajouter une méthode de calcul de la surface d'un cylindre ( $\text{surface du cercle} + 2 \times \text{rayon} \times \text{Math.PI} \times \text{hauteur du cylindre}$ ).

**Question 9** : compléter le programme de test suivant ( utilisation de l'opérateur `instanceof` ) :

```
Cercle c = new Cylindre();
c.setRayon(4.5);
.....
.....
System.out.println(" le volume de mon cylindre est : " +.....);
```

**Question 10** : compléter le programme de test avec une méthode qui affiche la surface d'un cercle et d'un cylindre ainsi que le volume du cylindre.

La signature de la méthode est : `static void afficher( Cercle cercle)`

**Question 11** : le programme de test créé 2 instances, l'une de `Cercle` et l'autre de `Cylindre`. Il affiche ensuite ces 2 objets en exécutant la méthode précédente.

**Question 12** : construire un tableau constitué d'instances de `Cercle` et `Cylindre`. Parcourir ce tableau afin d'afficher les surfaces respectives de chaque objets contenus dans ce tableau.