

# Exercices dirigés

## séance n°8

### Exercice 1 : saisie robuste

On veut écrire la fonction `saisieCorrecte` qui permet de saisir correctement un entier. Si l'utilisateur saisit une donnée dont le format n'est pas celui d'un entier, le programme lève l'exception `InputMismatchException`.

#### Question 1

La fonction devra traiter cette erreur en fournissant une solution alternative. Un message d'erreur sera affiché avec la proposition d'effectuer une nouvelle saisie.

Note: la classe `InputMismatchException` appartient au package `java.util`.

#### Question 2

L'entier saisi doit être impérativement supérieur à 10.

On demande donc de créer une classe d'exception adaptée à cette erreur, puis de modifier le programme afin de traiter ce cas d'erreur.

Note:

On aurait pu utiliser l'exception prédéfinie `IllegalArgumentException` pour vérifier que l'entier saisi est supérieur à 10.

## Exercice 2 : cercles

On veut créer une classe définissant des cercles.

### Question 1

Les cercles sont caractérisés par leur rayon. Définir une classe `Cercle` comportant les deux accesseurs `getRayon` et `setRayon` ainsi qu'une méthode de calcul de la surface. On définira la valeur de  $\pi$  comme une constante de la classe.

### Question 2

Ecrire la classe `TestCercle` qui créera 2 instances de cercles et affichera leur surface.

### Question 3

A la création d'une instance de cercle, il faut vérifier que la valeur de son rayon est positive.

Dans le cas contraire, on souhaite que le programme de test retourne un message d'erreur. Comment prendre en compte cette contrainte ?

### Question 4

Plutôt que d'envoyer un simple message d'erreur, il serait préférable que le programme ne s'interrompe pas mais demande une nouvelle saisie du rayon. Modifier le programme en conséquence.

### Question 5

On souhaite encore modifier le programme pour lui permettre d'afficher le nombre d'instances de la classe `Cercle` créées.