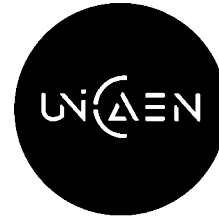


## **TP 1**

### **REVISIONS SUR LES CLASSES ET LES INSTANCES**



**UNIVERSITÉ  
CAEN  
NORMANDIE**

## **1. Classe Point**

### **1. Attributs, constructeurs, accesseurs, toString**

Créer la classe Point permettant de créer des points sur un plan (attributs double x et y), avec deux constructeurs : l'un qui prend un x et un y, l'autre sans argument qui crée un point positionné en (0,0). Eviter la redondance de code dans les constructeurs.

Ajouter les accesseurs et mutateurs.

Redéfinir toString() afin de décrire le point sous la forme suivante : "(3,2)" pour un point situé en x=3 et y=2.

### **2. Attributs et méthodes de classe (static)**

On souhaite compter le nombre d'instances créées par cette classe.

Définir un attribut static int nInstances permettant de les comptabiliser.

Ajouter un accesseur (static) sur cet attribut

Ajouter le code nécessaire au niveau des constructeurs.

Dans un main(), créer 3 instances de Point, et afficher ce que renvoie l'accesseur sur nInstances.

### **3. Translation et homothétie**

Définir les 3 méthodes suivantes :

- public void translation(double dx, double dy)
- public void homothetie(Point centre, double k)
- public void homothetie(double centreX, double centreY, double k)

Eviter la redondance de code entre les deux signatures d'homothétie en utilisant la délégation.

Testez la translation avec un point situé en (5,2) et se déplaçant de  $dx=1$  et  $dy=1$ , et qui doit donc se retrouver en (6,3).

Testez l'homothétie avec un point situé en (2,2) subissant une homothétie de centre (1,1) et de rapport 3, qui doit se retrouver en (4,4).

## 2. Classe Cercle

Un cercle possède un centre et un rayon. On utilisera donc la classe Point pour son centre. Faut-il utiliser l'héritage ou l'association ? Pourquoi ?

Définir dans cette classe aussi les 3 méthodes précédentes (translation et homothéties). Ici encore, éviter la redondance de code. Indication : l'homothétie d'un cercle revient à faire l'homothétie de son centre et à multiplier son rayon par  $k$ .

Redéfinir son `toString()` de sorte que l'on obtienne par exemple "centre=(3,2), rayon=4".

Testez notamment l'homothétie avec un cercle de centre (2,2) et de rayon 4 subissant l'homothétie de centre (1,1) et de rapport 3, qui doit donc se retrouver centré en (4,4) et avec un rayon 12.

## 3. Question supplémentaire : Classe Polygone

S'il vous reste du temps, commencer à définir la classe Polygone, avec les mêmes méthodes que dans Point et Cercle.

Vous pouvez éventuellement définir une méthode de calcul de l'aire du polygone. Elle pourra utiliser la somme algébrique de tous les trapèzes induits entre l'axe des  $x$  et les côtés du polygone. Selon que l'on tourne dans un sens ou dans l'autre, on obtient l'aire ou son opposé, donc on retiendra la valeur absolue.