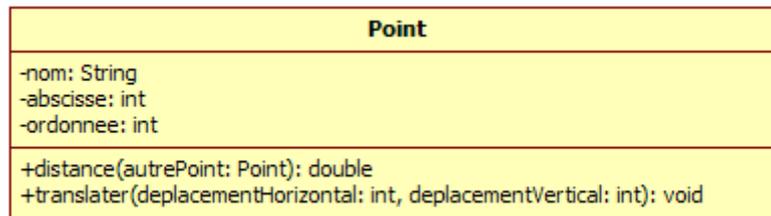


## TD2 – Première Classe Java : Point

### Introduction.

Le but de ce TD est d'écrire votre première classes Java à partir d'un diagramme de classes UML et de traduire les notions de classe, attributs et méthodes.

La classe Point de ce premier TD permettra de décrire par la suite des figures sur un écran avec des coordonnées entières :



Pour réaliser des calculs de distance, vous pouvez utiliser la classe Math qui offre un grand nombre de constantes et méthodes mathématiques. La plupart sont des champs ou méthodes de classes qui doivent être préfixées par le nom de la classe pour être appelée.

Ex : Math.PI pour accéder à la constante mathématique  $\Pi$ .

Pour pouvoir convertir des données numériques entre types primitifs (entier vers flottant, flottant double précision vers flottant simple précision), vous devez utiliser le mécanisme de cast :

```
Ex :   int a ;
       double b ;
       ....
       b = (double) a ;           // convertit l'entier a en double
```

### Exercice 1. Un Point c'est tout.

- Ecrire la classe **Point** qui décrit un point dans un repère en 2 dimensions avec des coordonnées entières (abscisse et ordonnée) et un nom.  
  
Les informations du point doivent être encapsulées et en lecture seule (**getter** uniquement).
- Ajouter un **constructeur** complet qui prend en paramètres le nom et les coordonnées du point.
- Implémenter le code des deux méthodes de la classe Point :
  - distance** : renvoie la distance de l'instance courante à celle d'un autre point passé en paramètre,
  - translater** : translate l'instance courante d'un vecteur de déplacement passé en paramètre.
- Ajouter une méthode **String toString()** permettant de représenter le Point par une chaîne de caractère de la forme **nom(abscisse, ordonnee)**.

### Exercice 2. Une figure avec 1 point.

- Créer une classe **AppliFigures** pour tester votre classe Point : le programme crée un point A de coordonnées (3, 5) et l'affiche sur la console.
- Translater le point A d'un vecteur (4, -3) et l'afficher à nouveau.
- Créer un deuxième point B de coordonnées (3, 5), calculer sa distance avec le point A et l'afficher.

### Exercice 3. Ant

Reprendre le fichier de configuration de projet Ant du TD précédent :

- a. Ajouter l'attribut nom de projet
- b. Une règle par défaut à votre convenance (compilation ou exécution)
- c. Définir l'architecture de votre projet dans des propriétés :
  - i. nom du répertoire contenant le code source Java (src)
  - ii. nom du répertoire contenant tout ce qui est produit (build)
  - iii. nom du répertoire contenant le bytecode produit (build/classes)
  - iv. nom de la classe à exécuter

Vous pourrez ajouter par la suite des propriétés pour référencer des répertoires contenant des bibliothèques utilisées ou produites, la documentation, l'archivage de votre projet, etc ...