

TD 4 - Types de classe, objets, constructeurs, méthodes, attributs-méthodes de classes

Une géométrie variable

Nous voulons réaliser un programme qui permet de représenter des dessins géométriques composés que de rectangles. Le programme doit être capable, en plus de représenter des dessins, d'opérer des translations p/r à un vecteur (couple de réels) ou encore par symétrie par rapport à l'origine. L'origine est un point, par défaut (0,0), mais le programme doit permettre de changer d'origine (attention l'origine doit être le même pour tout les points d'un dessin).

Point : un point est défini par ses deux coordonnées (abscisse et ordonnée), son nom ainsi que le point origine. Toutes les instances du type point doivent avoir la même origine. La classe **Point** offre un seul constructeur à trois paramètres ; abscisse, ordonné et le nom. Elle offre également :

- Quatre méthodes **getAbs**, **getOrd**, **setAbs** et **setOrd**. Ces méthodes permettent d'obtenir les valeurs de l'abscisse et de l'ordonnée, modifier l'abscisse et l'ordonnée.
- Deux méthodes **getX** et **getY** qui permettent d'obtenir les valeurs de l'abscisse et de l'ordonnée p/r au point origine.
- Deux méthodes **pegX** et **pegY** qui permettent de tester si l'abscisse (resp. ordonnée) du point est plus petit ou égale à l'abscisse (resp. ordonnée) d'un autre point donné en paramètre.
- Méthode **translate** qui prend deux réels et provoque la translation du point.
- Méthode **symétrie** qui permet de traduire le point à son symétrique p/r à l'origine.
- Méthode **info** qui retourne une chaîne de caractère qui résume les informations sur le point sous le format <nom-point>(<abs>,<ord>). Par exemple la méthode **info** sur la variable qui code le point **a** de la figure ci-après retournera la chaîne **a(3,4)**

Question 1> Écrivez la classe Point

Rectangle : un rectangle est défini par deux points qui correspondent à ses deux extrémités diagonaux et son nom. La classe **Rectangle** offre un constructeur qui prend seulement deux points. Le nom du rectangle est la concaténation des noms des points extrémités. La classe **Rectangle** possède également ;

- Les méthodes **getPPAbs**, **getPGAbs**, **getPPOrd** et **getPGOrd** renvoient respectivement le plus petit abscisse, le plus grand abscisse, le plus petit ordonnée, le plus grand ordonnée des points extrémités.
- La méthode **translate** qui permet de traduire le rectangle p/r à un vecteur (deux réels).
- La méthode **symetrie** qui translate le rectangle à son symétrique p/r à l'origine.
- La méthode **surface** qui calcule la surface du rectangle.
- La méthode **info** qui retourne une chaîne de caractère qui résume les informations sur le rectangle sous le format <nom-rectangle>[<extrémité1> , <extrémité 2>]. Par exemple, l'application de **info** sur la variable du rectangle de la figure, formé par les point **a** et **b**, retournera la chaîne **ab[a(3,4),b(4,2)]**

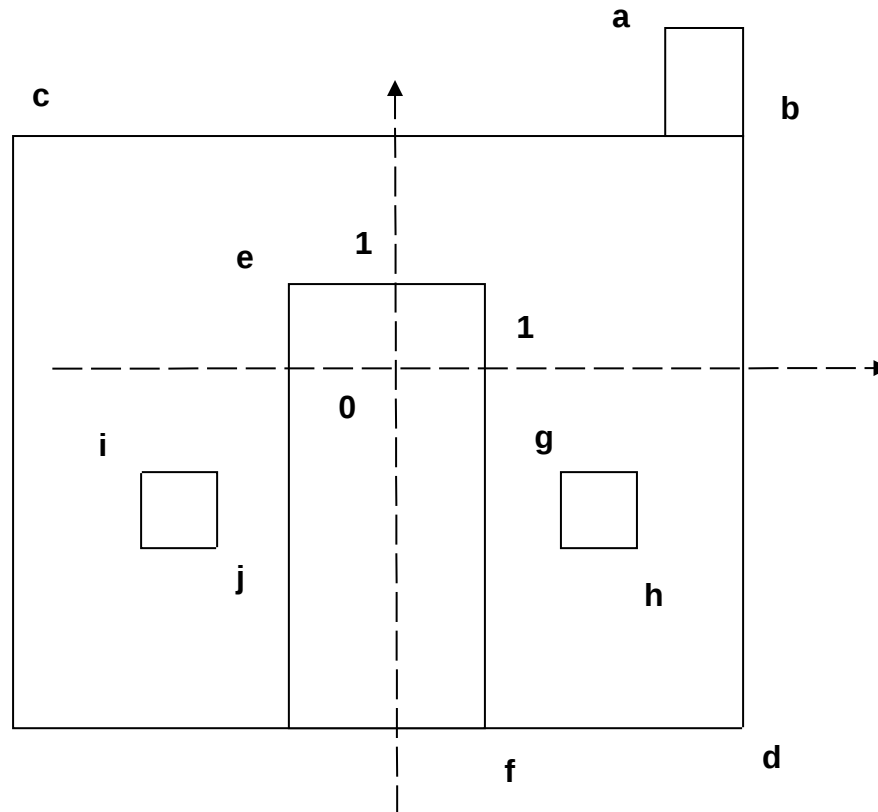
Question 2> Écrivez la classe Rectangle.

Dessin : La classe dessin est composé d'un tableau de rectangle.

Elle propose les méthodes suivantes :

- Méthode **ajout** qui permet d'ajouter un rectangle
- Méthode **translate** permet la translation de tout le dessin selon un couple de réels
- Méthode **symetrie** transforme le dessin à sont image p/r à l'origine
- Méthode **surface** qui calcule la surface du plus petit rectangle qui couvre le dessin.
- Méthode **affiche** qui liste les infos de chaque rectangle.

Question3 > Écrivez la classe Dessin.



Question4> Dans une classe Main, écrivez le programme qui définit l'objet de type Dessin qui code la figure. Affichez le dessin et calculez la surface du rectangle englobant. Ensuite, faites en sorte que **d** devienne le point origine, déplacez les rectangle (**a,b**) et (**g,h**) par rapport à l'origine affichez le nouveau dessin et recalculez la surface du rectangle englobant.

Le résultat attendu est le suivant :

```
ab[a(3.0,4.0),b(4.0,2.0)]
cd[c(-4.0,2.0),d(4.0,-4.0)]
ef[e(-1.0,1.0),f(1.0,-4.0)]
gh[g(2.0,-1.0),h(3.0,-2.0)]
ij[i(-3.0,-1.0),j(-2.0,-2.0)]
[(-4.0,4.0),(4.0,-4.0)]
64.0
ab[a(5.0,-12.0),b(4.0,-10.0)]
cd[c(-4.0,2.0),d(4.0,-4.0)]
ef[e(-1.0,1.0),f(1.0,-4.0)]
gh[g(6.0,-7.0),h(5.0,-6.0)]
ij[i(-3.0,-1.0),j(-2.0,-2.0)]
[(-4.0,2.0),(6.0,-12.0)]
140.0
```