

Travaux dirigés

Les classes et les instances en JAVA

1 OBJECTIF

Définir une classe, ses caractéristiques, ses comportements

2 TRAVAIL

Nous allons définir puis tester une classe en java. Ce premier TD va mettre en oeuvre en la complétant la classe Point présentée en cours.

2.1 Analyse (**Attention rien à coder sur machine ici**)

Actuellement la classe Point vue en cours est capable de créer un point. Toutefois la plupart du temps, le point à créer est fourni lors de sa construction :

- Avec des paramètres contenant les coordonnées du point à créer (abscisse et ordonnée).
- Avec un autre point fourni en paramètre

1. Proposez une la classe point pour fabriquer un objet point selon la description ci-dessus. Combien y a-t-il alors de constructeurs ?

Définir le moyen de créer des objets géométriques ayant les caractéristiques suivantes ou à définir

- des carrés, on doit pouvoir calculer l'aire de ces derniers
- des cercles, on doit pouvoir calculer l'aire de ces derniers
- des lignes constituées de traits continus, pointillés avec une longueur calculable, positionnée sur un plan, d'une épaisseur connue

2. Définissez les classes correspondantes avec leurs attributs et au moins un constructeur

3. Créer des objets des types précédents dans une fonction principale d'une classe nommée Geometrique (cette classe ne contiendra qu'un "main").

2.2 Codage de la classe Point

Utiliser Netbeans pour tester votre travail.

- Créez un projet de nom **TDGeometrie**
- Créez un paquetage **figGeometrique** dans lequel vous placerez les classes associées à ce type d'objets (classes **Point**, **Geometrique**)
- Définissez dans ce paquetage la classe **Point**. Elle sera conforme à ce qui suit :
 - 2 attributs x et y privés de type double
 - 3 constructeurs (sans paramètres, avec paramètres, avec un autre point)
 - Une méthode **translator** qui déplacera le point en abscisse et en ordonnée selon les offsets en paramètre.
 - Une méthode **calculerDistance** qui calculera la distance du point par rapport à l'origine (voici le code permettant de faire le calcul en JAVA : `Math.sqrt(x*x+y*y)`)
 - Une méthode **calculerDistancePoint** prenant en paramètre un Point et permettant de calculer la distance entre le point courant et le point en paramètre.

On rappelle la formule: $\|\vec{AB}\| = \sqrt{(xb - xa)^2 + (yb - ya)^2}$

- une méthode **afficher** qui va afficher à l'écran l'abscisse et l'ordonnée du point.
- Créez les points p1, p2, p3 et p4, comme indiqué ci-dessous, dans la méthode **main** de la classe **Geometrique**

```
Point p1, p2, p3, p4 ;
p1 = new Point() ;
p2 = new Point(1.0, 2.1) ;
p3 = new Point(p2) ;
p4 = p1 ;
p4.x = 1 ;
```

- Placez un point d'arrêt sur la première ligne de main et exécutez en pas à pas le programme

4. Combien d'objets ont été créés ? Justifiez votre réponse.

- Ajouter une méthode **afficher** qui affichera les coordonnées du point (à votre disposition les méthodes **print**, **println** et **formats** associés à l'objet standard de sortie **out**).
- Affichez les coordonnées de chacun des points après leur création.
- Déplacez le point p2 de {5, 5} puis afficher les coordonnées des trois points.

5. Que peut-on en conclure quant à l'instanciation d'un objet.

- On désire maintenant créer un nouveau point p5 de même caractéristique que p3.
- On affichera ses coordonnées puis on le déplacera de {5, 5}.

6. Même question. Comment avez-vous procédé pour démontrer la différence de comportement avec la question 5 ?

2.3 Codage des figures géométriques

- Définissez, toujours dans le même paquetage, les classes vues en analyse.
 - **Trait**. Un trait peut être continu ou pointillé, avec une longueur définie, positionné sur un plan et il a épaisseur connue
 - **Carre**
 - **Cercle**
- Définissez également une classe **TestFigure** qui devra :
 - Tester chacune des méthodes des classes Carre, Cercle et Ligne
 - Créer un objet carre, un objet cercle et un objet ligne
 - Afficher l'aire du carré
 - Afficher l'aire du cercle
 - Afficher la longueur du trait.

7. Quel élément permet de savoir qu'une classe se situe dans un paquetage ?

8. Peut-on à partir de la classe *TestFigure* accéder directement aux attributs de la classe *Point* ? Et pourquoi ?