

TD2 : Java et premiers pas en programmation orientée objet

V2.0.0



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Document en ligne : www.mickaël-martin-nevot.com

1 Généralités

Écrivez les applications ci-dessous en Java et en respectant la norme de programmation donnée en cours ; puis testez-les.

2 Géométrie

2.1 Point

2.1.1 Classe Point

La classe `Point` permet de manipuler un point. Cette classe contient :

- les constantes de classe correspondant aux coordonnées par défaut : `DFL_X` et `DFL_Y` ;
- les variables d'instance correspondant aux coordonnées : `x` et `y` ;
- les accesseurs et mutateurs associés à ces variables d'instance ;
- plusieurs constructeurs (avec ou sans coordonnées par défaut) ;
- la méthode `move(double x, double y)` qui effectue une translation ;
- la méthode `toString()` qui retourne une chaîne de caractères au format : "`(x,y)`".

2.1.2 Classe TestPoint

La classe `TestPoint` contient une méthode principale qui crée plusieurs objets de type `Point` (avec ou sans coordonnées par défaut), les déplace et les affiche (avant et après les déplacements).

2.2 Cercle

2.2.1 Classe Circle

La classe `Circle` permet de représenter un cercle. Un `Circle` est caractérisé par son centre (un `Point`) et son rayon. Outre les accesseurs et mutateurs associés à ses variables d'instance, un `Circle` doit avoir des méthodes permettant de :

- calculer sa surface ;
- calculer son périmètre ;
- tester sa superposition avec un autre `Circle` ;

- tester l'appartenance d'un `Point` à l'objet ;
- renvoyer une chaîne de caractère représentant le `Circle` de manière textuelle.

2.2.2 Classe `TestCircle`

La classe `TestCircle` doit réaliser les instructions suivantes (dans l'ordre) :

- création d'un `Circle` de centre (5,5) et de rayon 2 ;
- affichage de son périmètre ;
- affichage de sa surface ;
- multiplication par trois de son rayon ;
- nouvel affichage de son périmètre
- nouvel affichage de sa surface ;
- translation du centre du `Circle` à l'origine ;
- vérification que le `Point` (1,1) est bien à l'intérieur du `Circle` ;
- vérification que le `Point` (9,9) est bien à l'extérieur du `Circle` ;
- test de superposition du `Circle` avec un autre `Circle` de centre (0,0) et de rayon 2 ;
- test de superposition du `Circle` avec un autre `Circle` de centre (0,0) et de rayon 6.