

SERIE DE TP N° 3

Matière : Programmation O. O. Java

Enseignante: Hanen Karamti

Section : LCS 2

Exercice 1 :

Ecrire la classe **Cercle** qui possède deux attributs centre (de type **Point**) et rayon de type réel. Créer un constructeur qui prend deux paramètres (un point et un réel) pour initialiser le centre et le rayon, et un constructeur sans paramètres qui initialise le centre au point (0, 0) et le rayon à 1. Créer la méthode **afficher** permettant d'afficher les informations d'un cercle.

Définir les méthodes **getSurface()** et **getPerimetre()** permettant respectivement de retourner la surface et le périmètre d'un cercle.

Tester la classe Cercle.

Exercice 2 :

Ecrire un programme qui simule le mouvement d'un robot simple. Un tel robot occupe une certaine position (X, Y), il a une orientation parmi (Nord, Est, Sud, Ouest), il est initialisé à une position et une orientation donnée, il peut tourner à droite ou bien à gauche et il peut avancer d'un pas.

La classe **Robot** doit avoir les attributs :

- x, y de type int
- orientation qui peut prendre une valeur constante entre 1 et 4 : 1 pour Nord ; 2 Est pour, 3 pour Est et 4 pour Ouest.

Et les méthodes :

- **tourneraDroite, tourneraGauche** qui modifient l'orientation du robot sans changer sa position.
- **avancer(int d)** qui modifie la position du robot selon son orientation actuelle.
- **afficher** qui affiche la position et l'orientation du robot.
- d'accès aux attributs (get et set).

Ecrire aussi un programme qui teste la classe *Robot*.

Exercice 3 :

Ecrire la classe *Rectangle* qui possède trois attributs :

- sommet (qui est un point et qui représente le point en haut et à gauche)
 - deux réels qui représentent la longueur et la largeur du rectangle.
1. Ajouter les constructeurs que vous jugez nécessaires.
 2. Ajouter les méthodes d'accès aux attributs.
 3. Ajouter les méthodes *surface* et *perimetre* qui permettent respectivement de retourner la surface et le périmètre d'un rectangle.
 4. Ajouter la méthode *afficher* qui permet d'afficher les informations d'un rectangle.
 5. Ajouter la méthode *appartient(Point p)* qui retourne un booléen indiquant si le point p passé comme argument se trouve dans le rectangle ou non.
 6. Créer un programme principal dans lequel vous testez la classe Rectangle et ses différentes méthodes.