

TP n°2 – Figures géométriques

L'objectif est de coder en Javascript les classes **Point**, **Figure**, **Cercle**, **Polygone** et **Losange**, puis de manipuler une file de figures géométriques.

Une **figure** est composée de 1 ou de plusieurs **points**. La dimension d'une figure est 2 (on se situe dans le plan). Parmi les figures possibles se trouve le **cercle** (défini par son point central et son rayon) et le **polygone** (défini par un tableau de points). Le **losange** est un **polygone** particulier qui à quatre côtés de même longueur.

1 – Codage et test de la classe Point

Coder la classe **Point** avec 2 attributs pour les coordonnées, un constructeur à deux paramètres (par défaut c'est le point (0,0)) et les *getters*.

Ajouter la méthode **distance()** qui retourne la distance qui sépare le point courant d'un autre point passé en paramètre. La distance entre le point (x1, y1) et le point (x2, y2) se calcule avec $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$

Tester :

```
let p1 = new Point(1, 2)
let p2 = new Point(2, 3)
// distance entre deux points
console.log("distance p1p2 = " + p1.distance(p2))
```

2 – Codage et test de la classe Figure

Coder la classe **Figure** avec un attribut (tableau de points). Si le tableau n'est pas passé en paramètre, le tableau de points est vide. Ajoutez la méthode **ajouterPoint()** qui ajoute un point en fin de tableau. Ajouter la méthode **dimension()** qui retourne 2 (on se situe par défaut dans le plan).

3 – Codage et test des classes Polygone et Cercle

Coder la classe **Polygone** : ne mettre que les attributs et les méthodes constructeurs et éventuels *getters* et *setters*.

Ajoutez dans **Polygone** la méthode qui retourne le nombre de côtés.

Tester :

```
// creation d'un polygone à 5 sommets
let p0= new Point(0, 0), p1= new Point(1, 1),
p2= new Point(1, 2), p3= new Point(0, 3),
p4= new Point(-1, 2);

let penta = new Polygone(new Array(p0,p1,p2,p3,p4));

// perimetre du polygone
console.log ("perimetre de penta (doit être 7,47...) = " + penta.perimetre());
//dimension
console.log("dimension de penta = " + penta.dimension());

//nb cotes
console.log("nb de cotes de penta = " + penta.nbCotes());
```

Coder la classe `Cercle` ainsi que ainsi que `Losange` (par exemple le polygone ayant les points (0,0), (-1,2), (0,4) et (1,2) est un losange).

4 – Méthode `toString()`

Redéfinir la méthode `toString()` pour afficher une représentation symbolique de chaque figure. Par exemple, pour :

- Un polygone « je suis un polygone à ... côtés »
- Un losange : « je suis un polygone à ... côtés – je suis un parallélogramme – je suis un losange »
- ...

5 – Collection de figures géométriques

Coder la classe `FileFigure` avec, comme attribut, un tableau contenant des figures. Ajoutez les méthodes :

- la méthode `ajouter()` qui ajoute un figure en fin de tableau
- la méthode `sortirCercle()` qui enlève de la file le premier cercle rencontré, en commençant par le début du tableau, la méthode retourne le cercle enlevé,
- une méthode qui affiche dans la console la composition de la file en commençant par le début du tableau. Pour chaque élément, on affiche sa position et ce qu'il est c'est-à-dire soit « je suis un polygone à ... côtés », soit « je suis un polygone à ... côtés – je suis aussi un losange », soit « je suis un cercle de centre le point (...,...) »

Écrire un script qui :

- Crée une file de figures avec en tête un cercle, suivi d'un losange, d'un polygone à 5 côtés et d'un autre cercle.
- Affiche la composition de la file du premier entré au dernier entré,
- Enlève de la file le premier cercle,
- Affiche à nouveau la composition de la file.