

TP n°2
Interfaces et *Javadoc*

1 Première interface

On fournit le fichier `IPoint.java` qui contient l'interface `IPoint` :

1. Téléchargez le fichier et placez-le dans le répertoire du TP.
2. Créez le fichier `PointCartesien.java` qui contiendra la classe `PointCartesien` qui implémente cette interface (n'oubliez pas le mot-clef `implements` suivi du nom de l'interface après le nom de la classe).
3. Spécifiez les attributs ainsi que les trois constructeurs classiques.
4. Que se passe-t-il lorsqu'on compile la classe (une fois toutes les erreurs de syntaxe éventuelles corrigées) ? Corrigez le problème avant de continuer (au besoin n'hésitez pas à solliciter votre intervenant de TP !)
5. Ajoutez les *setters* et les *getters*.
6. Ajoutez les méthodes `toString` et `afficher`.
7. Si ce n'est déjà fait, implémentez les méthodes issues de l'interface `IPoint`.
8. Écrivez la classe de test `TestPoint` qui teste toutes les méthodes de la classe `PointCartesien`.

2 Multiples implémentations d'une interface

Nous souhaitons maintenant écrire une classe `PointPolaire` qui implémente elle-aussi, l'interface `IPoint`. Pour rappel, la figure suivante présente la différence entre un point en coordonnées cartésiennes (à gauche) et en coordonnées polaires (à droite) :

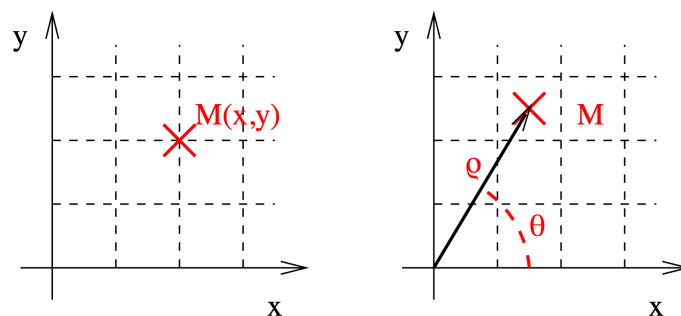


FIGURE 1 – Un point en coordonnées cartésiennes (à gauche) et en coordonnées polaires (à droite)

1. Créez le fichier `PointPolaire.java` qui contient la classe `PointPolaire` qui implémente l'interface `IPoint`.
2. Ajoutez les attributs, les constructeurs, les *getters* et *setters*, la méthode `toString` et la méthode `afficher`.
3. Implémentez les méthodes de l'interface.

4. Modifiez la classe de test `TestPoint` pour tester toutes les méthodes de la classe `PointPolaire`.
5. Modifiez à nouveau la classe de test `TestPoint` pour permettre à l'utilisateur de créer cinq points. Pour chaque point, il peut choisir de fournir les coordonnées cartésiennes ou polaires ; les coordonnées sont alors saisies au clavier. Affichez ensuite les 5 points ainsi que leur distance à l'origine. Indications : pour cela vous utiliserez un tableau de `IPoint`.

3 La *Javadoc*

Le but de cet exercice est de reprendre les classes développées dans les exercices précédents afin de générer une documentation à l'aide de l'outil `Javadoc`.

1. Générez la `Javadoc` de l'interface `IPoint` à l'aide de la commande suivante :
`javadoc IPoint.java -d docs`
2. Ouvrez ensuite le fichier `index.html` (à l'aide d'un navigateur Web) situé dans le répertoire `docs`.
3. Générez la `Javadoc` de l'ensemble de vos classes et interfaces à l'aide de la commande suivante :
`javadoc *.java -d docs`
4. Vérifiez le résultat obtenu.
5. En vous aidant de l'exemple de l'interface, complétez vos classes pour afficher une documentation complète pour chaque méthode (et chaque paramètre et valeur de retour).
6. Pour afficher l'auteur dans la `Javadoc` ajoutez l'option `-author`, et `-version` pour la version :
`javadoc *.java -d docs -author -version`
7. Faites une archive de l'ensemble des dossiers/fichiers du TP2 et déposez-la sur le Moodle du cours dans le dépôt TP2 de votre groupe.



S'il vous reste du temps, complétez la `Javadoc` et générez la documentation pour les classes du premier TP.