



TP2 : semaines 2 et 3

Exercice 1

- Définir la classe Point telle qu'énoncée dans le cours :
 - Ses 3 attributs privés
 - Un constructeur par défaut (sans paramètre) initialisant les attributs avec des valeurs par défaut
 - Le constructeur avec 2 coordonnées comme le cours mais qui initialise aussi le nom avec une valeur par défaut
 - Le constructeur avec le nom en paramètre mais qui initialise les 2 coordonnées avec des valeurs par défaut en appelant le constructeur par défaut
 - Toutes les méthodes énoncées dans le cours, tous les getters/setters et toute autre méthode que vous jugez nécessaire : par exemple, une méthode surchargée *@Override toString()* qui retourne une expression sous la forme de la chaîne "nom=" + name + " abscisse=" + x + " ordonnee=" + y. La méthode *affiche()* du cours sera surchargée avec une autre méthode *affiche(String chaîne)* qui affiche la chaîne en paramètre.
- Définir une classe Test avec le main :
 - Créer un objet p1 de la classe Point avec le premier constructeur par défaut
 - Saisir au clavier 2 coordonnées
 - Créer un objet p2 de la classe Point avec le second constructeur avec les 2 coordonnées saisies
 - Saisir au clavier un nom (chaîne)
 - Créer un objet p3 de la classe Point avec le troisième constructeur avec le nom saisi en paramètre
 - Modifier l'objet p2 en lui affectant le nom saisi avec le setter approprié
 - Pour les 3 objets, appeler toutes les méthodes y compris les 2 méthodes surchargées *affiche*, l'une sans paramètre, l'autre avec pour paramètre la chaîne récupérée avec la méthode *toString()*
- Définir une classe Segment :
 - 2 objets de la classe Point et la longueur réelle du segment comme attributs privés
 - Un constructeur par défaut pour initialiser les 3 attributs avec des valeurs par défaut
 - Un constructeur avec 2 points en paramètres pour initialiser ces 2 points en attributs
 - Une méthode qui calcule et retourne la longueur du segment.
 - Une méthode qui déplace les 2 points du segment avec les 2 coordonnées en paramètres, ceci en vous servant de la méthode *deplace* de la classe Point vue en cours
 - Une méthode surchargée *@Override toString()* qui retourne une expression sous forme de chaîne, en appelant la méthode *toString()* pour chacun des 2 points du segment
 - Comme pour la classe Point, surchargez la méthode *affiche* sans paramètre et avec une chaîne en paramètre pour afficher les informations des 2 points du segment et sa longueur, ceci en appelant la bonne méthode *affiche* pour chacun des 2 points du segment
- Dans le main de la classe Test :
 - Créer un objet s1 de Segment avec en paramètres les 2 points (p1, p2) et afficher la longueur de s1
 - Créer un objet s2 de Segment avec en paramètres les 2 points (p2, p3) et afficher la longueur de s2
 - Afficher les informations de s1 et s2 en appelant les 2 méthodes surchargées *affiche*
 - Déplacer les 2 points de s1 avec les coordonnées de p3 et afficher les informations de s1
- Définir une classe Polygone :
 - Définir un tableau d'objets de Segment et un entier indiquant le nombre de segments du polygone
 - 3 constructeurs : l'un sans paramètre (par défaut), le second avec le nombre de segments pour instancier le tableau, le troisième avec un tableau d'objets de Segment à copier dans le tableau en attribut
 - Toutes les méthodes nécessaires : déplacer tous les segments du polygone, afficher les informations de tous les segments du polygone par appel des méthodes de la classe Segment
- Compléter le main de Test pour créer un polygone non fermé avec un tableau de 2 segments tels que ces 2 segments partagent un point commun. Tester vos méthodes de déplacement et d'affichage



Exercice 2

- Ecrire la classe Triangle héritant de la classe Polygone :
 - Définir les attributs périmètre et surface
 - Implémenter une méthode calculant et retournant le périmètre du Triangle soit la somme des longueurs des 3 segments.
 - Implémenter une méthode calculant et retournant la surface :

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

avec $p = \text{périmètre}/2$ et a, b et c les longueurs des 3 segments

- Ecrire la classe Rectangle héritant de la classe Polygone :
 - Définir les attributs périmètre et surface
 - Implémenter une méthode calculant et retournant le périmètre du Rectangle soit la somme des longueurs des 4 segments.
 - Implémenter une méthode calculant et retournant la surface
- Dans le main de la classe Test :
 - Créer un objet de Triangle avec un tableau de 3 segments tels que 2 segments contigus aient un point commun.
 - Afficher les informations de tous les segments du Triangle
 - Afficher le périmètre et la surface du Triangle
 - Créer un objet de Rectangle de telle sorte que les 4 segments de son tableau forment un rectangle : vous saisirez les coordonnées des 4 points du rectangle et blinderez votre saisie pour que ces 4 points forment bien un rectangle
 - Afficher les informations de tous les segments du Rectangle
 - Afficher le périmètre et la surface du rectangle