Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
Master Internet des objets : Logiciel et Analytique (MIOLA) 2020/2021

TP 1 Java
Les bases de la POO

Exercice1

Partie I : Manipulation d'un point dans le plan

Il s'agit de modéliser un point du plan dont les abscisses (x et y) sont des réels. On souhaite définir les opérations de base permettant de manipuler un point.

1. Définir la classe Point, assurer l'encapsulation de ses attributs et définir les getters et les setters.
2. Définir trois constructeurs pour la classe Point:
    a. Un constructeur par défaut initialisant les coordonnées par 0.
    b. Un constructeur recevant les valeurs avec lesquelles les cordonnées seront initialisées.
    c. Un constricteur par copie.
    d. Définir le destructeur (finalize())de la classe Point affichant un message de destruction.
3. Définir la fonction toString() permettant d'afficher l'état d'un point.
4. Définir une fonction translaterV(int d) permettant de translater verticalement le point courant avec une distance d.
5. Définir une fonction translaterH(int d) permettant de translater horizontalement le point courant avec une distance d.
6. Définir une fonction milieu(Point p) permettant récupérer le milieu des deux point représentés par l'objet courant et l'objet p.
7. Définir une fonction distance(Point p) permettant calculer la distance entre l'objet courant et l'objet p.

Partie II : Manipulation d'un segment dans le plan

un segment peut etre modelisé par deux points a et b. Les opérations que l'on souhaite faire sur ce segment sont :

- calculer sa longueur ;
- savoir si un point donné se trouve sur le segment.

1. Définir la classe segment, assurer l'encapsulation de ses attributs et définir les getters et les setters.
2. Définir un constructeur de ce segment recevant en arguments les deux extrémités du segment que l'on veut construire ;
3. Définir une méthode publique retournant la longueur du segment ;
4. Définir une méthode dont le prototype est : **public boolean** appartient(**Point** p); indiquant si le point de coordonnée x appartient ou non au segment ;
5. Définir la fonction toString() permettant d'afficher l'état d'un segment.
6. Créez une autre classe, que vous appellerez Test par exemple, muni d'une méthode main (public static void main(String[] args){ ... } qui permet de tester les différentes fonctions de la classe segment.

Exercice2

Partie 1 - Création d'une classe représentant un nombre complexe

Un nombre complexe peut être modélisé par une classe composée de deux attributs de type réel représentants sa partie réelle et sa partie imaginaire.

1. Créez une classe Complexe qui aura deux attributs privés : un réel **a** et un réel **b**, représentant la partie réelle et la partie imaginaire de ce complexe.
2. Dotez la classe Complexe d'un constructeur : celui-ci prend en argument d'entrée deux réels et les attribue aux attributs **a** et **b** de la classe.
3. Créez une méthode affiche() qui permet d'afficher à l'écran le complexe que l'on manipule sous la forme "a+ib".
4. Créez une autre classe, que vous appellerez Test par exemple, muni d'une méthode main (public static void main(String[] args){ ... }). Dans cette méthode, vous allez déclarer et instancier un complexe z, puis demander à la méthode affiche() d'afficher ce complexe à l'écran.

Partie 2 - Les méthodes de l'objet Complexe

1. Créez une méthode qui retourne le module du complexe manipulé.
2. Créez une méthode qui retourne le conjugué du complexe manipulé.
3. Créez une méthode qui permet d'additionner un autre nombre complexe et qui retourne la somme de ces deux complexes.
4. Créez une méthode qui permet de multiplier le complexe manipulé par un autre nombre complexe et qui retourne le résultat.
5. Créez une méthode qui retourne l'argument (utilisez la méthode atan2 de la classe abstraite Math : consultez la documentation en ligne, par exemple celle de sun ! (http ://java.sun.com/j2se/1.4.2/docs/api/)).
6. Créez une méthode qui affiche le complexe en notation exponentielle.

Projet Médiathèque

Une Médiathèque universitaire contient différents types de documents électroniques (articles, livres, magazines…). Mais quelque soit le document, celui-ci possède un ISBN, un titre, un tableau d'auteurs (au maximum 5 éléments de type String), un éditeur, une année d'édition et un URL.

La médiathèque distribue des Kindles à ses clients pour accéder en ligne aux différentes ressources de la médiathèque. Seuls les clients abonnés sont autorisés à accéder aux ressources de la médiathèque. Également, les Kindles ne peuvent pas être utilisés que dans la zone géographique du campus universitaire et pour une durée de 3 heures maximum après la connexion de l'utilisateur tout en respectant l'horaire d'ouverture de la médiathèque. L'utilisateur ne peut accéder qu'à un nombre limité de documents par jour, ce nombre dépond du type du client.

Les étudiants possèdent une réduction de 50% sur le prix d'abonnement annuel et peuvent accéder à tous les articles scientifiques sans restriction. Tandis que les professeurs ont une réduction de 30% et peuvent accéder à tous les documents sans restriction.

1. Définissez la classe Document avec son constructeur public.
2. Assurez le principe d'encapsulation des attributs et définissez les accesseurs et les modificateurs.
3. Quand un document est créé, il y aura des informations qui ne doivent pas être changées. Quelles sont ces informations. Quels attributs devez-vous rajouter pour interdire leur modification?
4. On veut attribuer un numéro d'enregistrement unique dès que l'on crée un objet Document : le premier document créé doit avoir le numéro 0, puis ce numéro s'incrémente de 1 à chaque création de document. Ajoutez les champs nécessaires à la classe Document, modifiez le constructeur puis ajoutez une méthode getNumero renvoyant le numéro d'enregistrement du Document. Quel(s) attribut(s) faut-il rajouter pour s'assurer que le numéro d'enregistrement restera unique et non-modifiable ?
5. Définissez la méthode toString renvoyant la chaîne décrivant l'état d'un objet document.
6. Définissez le destructeur d'un document permettant de rappeler l'état de celui-ci et affichant un message qui informe l 'utilisateur de la destruction de l'objet.
7. Définir une fonction principale main de test.