TD Programmation avancée

Exercice 1 :
Proposer une implémentation de la classe **Point** caractérisé par

- les attributs : x et y
- un constructeur avec argument
- une méthode ***afficher*** permettant d'afficher les coordonnés d'un point
- une méthode ***calculerDistance*** permettant de calculer la distance avec un autre point

Exercice 2 :
Proposer une implémentation de la classe **Droite** passant par deux points. La classe Droite prend les attributs suivants :

- **point1** correspondant au premier point de la droite. Il est de type Point.
- **point2** correspondant au second point de la droite. Il est de type Point.

On note également les méthodes suivantes

- ***afficher*** permettant d'afficher les deux points par lesquels passent la droite
- ***equals*** permettant de vérifier si deux droites sont identique
-

Exercice 3

On souhaite développer un programme de gestion des menus. Le programme comporte deux classes : ***Ingredient*** et ***Menu***.

La classe ***Ingredient*** est caractérisée par les attributs suivants

- **libelle** de type String : il définit le nom de l'ingrédient ;
- **quantite** de type double : il définit la quantité de l'ingrédient utilisée dans un menu ;
- **prix** de type double : il définit le prix.

La classe **Ingredient** est caractérisée par les méthodes suivantes :

- un constructeur par défaut ;
- un constructeur avec arguments ;
- des accesseurs et des modificateurs ;
- Une méthode ***afficher*** qui permet d'afficher les information sur un ingrédient ;
- une méthode ***calculerCout*** qui permet de calculer le coût d'un ingrédient en fonction du prix et de la quantité et retourne la valeur. Le calcul du coût se fait comme suit :
    - **cout = prix*quantite ;**

La classe Menu est caractérisée par les attributs suivants :

- **nom** de type String : il définit le nom du menu ;
- **prix** de type double : il définit le prix de vente d'un plat du menu ;
- **cout** de type double : il définit le coût du menu (il dépend des ingrédients utilisés pour faire le menu) ;
- **vente** de type double : il définit le montant total vendu ;
- **quantite** de type int : il définit le nombre de plats disponibles ;
- **ingredients** de type **ArrayList**<**Ingredient**> : il définit la liste des ingrédients utilisés pour constituer le menu.

La classe Menu est caractérisée par les méthodes suivantes :

- un constructeur par défaut ;
- un constructeur avec arguments ;
- une méthode ***afficher*** qui permet d'afficher les information du menu ;

- une méthode ***ajouterIngredient*** qui prend en paramètre un ingrédient et l'ajoute dans la liste des ingrédients du menu ;
- une méthode ***calculerCout*** qui calcule le coût d'un menu en fonction des ingrédients utilisés.
- Une méthode ***ajouterVente*** qui prend en argument un paramètre nbrePlatAchete et calcule le montant total vendu et diminue le nombre de plats disponible comme suit :
    - vente = vente + nbrePlatsAchte*prix
    - quantite = quantite - nbrePlatAchete

1. Donner une implémentation des classes **Ingredient** et **Menu**.
2. Implémenter une méthode main qui permet de :
    - créer un menu quelconque ;
    - créer deux ingrédients quelconques;
    - ajouter les deux ingrédients créés dans le menu précédemment créé ;
    - vendre deux plats
    - afficher les informations du menu

Exercice 4 (héritage)
On souhaite développer un programme permettant de gérer les comptes bancaires. Un compte est caractérisé par son numéro, sa date de création et son solde. Sur un compte on peut déposer une certaine somme, retirer une certaine somme.
On distingue deux types de compte : compte courant et compte d'épargne. Un compte courant est un compte avec un agios (frais de gestion). Un compte d'épargne est caractérisé par un taux d'intérêt.
Le compte doit être implémenté dans une classe Compte qui dispose :

- d'un constructeur par défaut qui initialise le solde à 0 ;
- d'un constructeur avec arguments;
- d'une méthode **deposer** pour déposer une certaine somme passée en argument;
- d'une méthode **retirer** pour retirer une certaine somme passée en argument. Noter qu'on ne peut pas retirer une somme supérieure au solde disponible.

Le compte courant doit être implémenté dans une classe **CompteCourant** qui hérite de la classe Compte et qui dispose :

- d'un attribut **fraisGestion** qui correspond aux frais de gestion mensuels ;
- d'un constructeur par défaut
- d'un constructeur avec arguments
- d'une méthode **calculerFraisGestion** qui permet de calculer les frais de gestion selon un certain nombre de mois passé en paramètre et de les retirer sur le solde.
- d'une méthode **afficher** qui permet d'afficher les informations relatives au Compte courant.

Le compte d'épargne doit être implémenté dans une classe CompteEpargne qui hérite de la classe Compte et qui dispose

- d'un attribut **tauxInteret** qui correspond au taux d'intérêt appliqué
- d'un constructeur par défaut
- d'un constructeur avec arguments
- d'une méthode **ajouterInteret** qui permet de calculer l'intérêt selon le taux d'intérêt appliqué et de l'ajouter au solde. solde = solde *(1+taux) ;
- d'une méthode afficher qui permet d'afficher les informations relatives au Compte d'épargne.

Proposer une implémentation da la classe Compte, CompteEpargne et CompteCourant.

Exercice 5 : classe abstraite
En informatique, une **Pile** est une structure de données fondée sur le principe dernier arrivé, premier sorti. En informatique, une **Pile** est une structure de données fondée sur le principe premier arrivé, premier sorti. On

souhaite implémenter une classe Pile destinée à la gestion des piles d'entier et une classe File destinée à la gestion des files d'entier. On fait l'hypothèse qu'une Pile et une File sont des sorte de Sac. La classe Sac est caractérisée par :

- une liste d'éléments qui le compose
- un constructeur par défaut
- un méthode ***empiler*** qui prend en paramètre un entier et qui l'empile;
- d'une méthode abstraite ***depiler*** qui retourne l'élément dépilé
- d'une méthode ***estVide*** qui permet de vérifier si un sac est vide et qui retourne un booléen
- d'une méthode ***affiche*** qui affiche les éléments d'un Sac.

1. Donner une implémentation de la classe **Sac**.
2. Donner une implémentation de la classe **Pile** et de la classe **File** sachant qu'elles héritent de la classe **Sac**.

Exercice 6 (héritage)

On souhaite implémenter en java une carte spatiale. Une carte est caractérisée par un ensemble d'entités spatiales. Une entité spatiale est un point, un polygone ou une ligne. Un point est caractérisé par son abscisse et son ordonné. Une ligne a un point de départ, un point d'arrivée et une longueur. On a plusieurs types de polygone : le rectangle, le losange. Un rectangle est caractérisé par sa longueur et sa largeur qui sont des lignes. Un losange est caractérisé par ses deux diagonales qui sont des lignes. On souhaiterait connaître la surface et le périmètre d'un polygône pour effectuer des calculs géométriques sur une carte.

1. Faire une implémentation java des classes permettant de représenter une carte.

Exercice 7 Classe Interface (4 points)

On souhaite implémenter en java la gestion des véhicules. On considère qu'un véhicule peut être motorisé ou non. Un véhicule est caractérisé par son type, sa marque et sa vitesse. Un véhicule motorisé demarre, consomme du carburant par heure, fait le plein et roule. On considère que les véhicules motorisés n'ont pas exactement le même comportement. Ils ne demarrent pas de la même manière, consomment différemment, roulent différemment et font le plein différemment.

On souhaite intégrer une classe Voiture, une classe Avion et une classe Velo.

Proposer une implémentation java de la solution.