TP 2 : Concepts POO

Exercice 1 :

On souhaite créer une application en java qui permet de gérer les salaires des ingénieurs et des managers d'une entreprise de développement informatique.

1. Créez la classe abstraite **Employe** avec les attributs nom, prenom, email, telephone, et salaire. Ajoutez les constructeurs avec et son paramètres, puis la méthode abstraite calculerSalire() qui retourne le salaire d'un employé.
2. Créez la classe **Ingénieur** avec l'attribut spécialité. Redéfinissez la méthode calculerSalire() sachant qu'on prévoit une augmentation de 15% par rapport à son salaire normal.
3. Créez la classe **Manager** avec l'attribut service. Redéfinissez la méthode calculerSalire() sachant qu'on prévoit une augmentation de 20% par rapport à son salaire normal.
4. Créez une application qui contient une méthode main() pour tester les différentes classes, dans laquelle :
   — déclarez et intentiez un ingénieur ;
   — déclarez et intentiez un manager ;
   — affichez les informations de l'ingénieur et du manager (nom, prénom, salaire, service, et spécialité).

Exercice 2 :

L'objectif de cet exercice et de concevoir et de réaliser une application JAVA qui gère les commandes des clients d'une entreprise qui vend des ordinateurs. L'application demandée doit donner la possibilité de gérer les ordinateurs, les catégories, et les commandes de l'entreprise.

- Créez une classe **Ordinateur** avec les attributs nom, marque, prix, description, et nombre en stock. Chaque ordinateur appartient à une catégorie. Ajoutez une méthode qui retourne le prix pour une quantité donnée. overriding equals of Object
- Créez une classe **Catégorie** avec les attributs nom, description et une liste d'ordinateurs. Ajoutez la méthode ajouterOrdinateur() pour ajouter un nouveau ordinateur à la liste (vous devez vérifier s'elle existe déjà avant de l'ajouter), une méthode supprimerOrdinateur() pour supprimer un ordinateur, et une méthode rechercherParPrix() qui retourne la liste des ordinateurs par un prix donné en paramètre.

- Créez une classe **Commande** avec les attributs référence, le client, la date de commande, et l'état de la commande.
- Créez une classe **LigneCommande** avec les attributs quantité, la commande et l'ordinateur commandé.
- Créez une classe **Client** avec les attributs nom, prénom, adresse, email, ville, téléphone, et une liste de commandes effectuées. Ajoutez la méthode ajouterCommande() pour ajouter une nouvelle commande à la liste (vous devez vérifier s'elle existe déjà avant de l'ajouter), et une méthode supprimerCommande() pour supprimer une commande.

Modélisez cette application à l'aide d'un diagramme de classes et implémentez toutes les classes avec leurs attributs. Ajoutez également les constructeurs avec et sans paramètres, les getters, les setters et la méthodes toString pour chaque classe.

Créez une application qui contient une méthode main() pour tester les différentes classes, dans laquelle :

— déclarez et intentiez une liste de trois ordinateurs ;
— déclarez et intentiez une catégorie ;
— déclarez et intentiez un client ;
— déclarez et instanciez une commande du client ;
— déclarez et instanciez une liste de trois lignes de commandes pour la commande et les ordinateurs créés ;
— affichez toutes les informations de la commande.