TP2 : Classes JAVA

Exercice 1 :

Créez une classe LandTract qui a deux champs : un pour la longueur et un pour la largeur. La classe doit avoir une méthode qui renvoie l'aire du tract, ainsi qu'une méthode equals et une méthode toString. Testez la classe à l'aide d'un programme qui demande à l'utilisateur d'entrer les dimensions de deux Land Tract. Le programme doit afficher l'aire de chaque Land Tract et indiquer si elles sont de taille égale.

Exercice 2 :

Vous avez écrit une classe Employee. Ajoutez ce qui suit à la classe :

- Un constructeur qui accepte les valeurs suivantes comme arguments et les affecte aux champs appropriés : nom de l'employé, numéro d'identification de l'employé, département et position.
- Un constructeur qui accepte les valeurs suivantes comme arguments et les affecte aux champs appropriés : nom de l'employé et numéro d'identification. Les champs département et position doivent être affectés d'une chaîne vide ("").
- Un constructeur sans argument qui attribue des chaînes vides ("") aux champs name, department et position, et 0 au champ idNumber.

Ecrivez un programme qui teste et démontre ces constructeurs.

Exercice 3 :

Dans un programme, écrivez une méthode qui accepte deux arguments : un tableau et un nombre n. Supposons que le tableau contienne des entiers. La méthode doit afficher tous les nombres du tableau qui sont supérieurs au nombre n.

Exercice 4 :

Écrivez une classe nommée PhoneBookEntry qui contient des champs pour le nom et le numéro de téléphone d'une personne. La classe doit avoir un constructeur et des méthodes getter et setter appropriées. Ensuite, écrivez un programme qui crée au moins cinq objets PhoneBookEntry et les stocke dans un ArrayList. Utilisez une boucle pour afficher le contenu de chaque objet dans ArrayList.

Exercice 5 :

Ecrivez une classe nommée Personne avec des champs pour contenir le nom, l'adresse et le numéro de téléphone d'une personne. Écrivez un ou plusieurs constructeurs et les méthodes getter et setter appropriées pour les champs de la classe. Ensuite, écrivez une classe nommée Customer, qui hérite de la classe Person. La classe Customer doit avoir un champ pour un numéro de client et un champ booléen indiquant si le client souhaite être sur une liste de diffusion. Écrivez un ou plusieurs constructeurs et les méthodes getter et setter appropriées pour les champs de la classe.
Test : un objet de la classe Customer dans un programme simple.

Exercice 6 :

Un magasin de détail a un plan client préféré où les clients peuvent gagner des remises sur tous leurs achats. Le montant de la remise d'un client est déterminé par le montant des achats cumulés du client en magasin, comme suit :

- Lorsqu'un client privilégié dépense 500 $, il bénéficie d'une remise de 5 % sur tous ses achats futurs.
- Lorsqu'un client privilégié dépense 1 000 $, il bénéficie d'une remise de 6 % sur tous ses achats futurs.
- Lorsqu'un client privilégié dépense 1 500 $, il bénéficie d'une remise de 7 % sur tous ses achats futurs.
- Lorsqu'un client privilégié dépense 2 000 $ ou plus, il bénéficie d'une remise de 10 % sur tous ses achats futurs.

Ecrivez une classe nommée PreferredCustomer, qui hérite de la classe Customer que vous avez créée dans l'exercice précédent. La classe PreferredCustomer doit avoir des champs pour le montant des achats du client et le niveau de remise du client. Écrivez un ou plusieurs constructeurs et les méthodes getter et setter appropriées pour les champs de la classe. Démontrer la classe dans un programme simple.

Exercice 7 :

Écrivez une classe nommée TestScores. Le constructeur de classe doit accepter un tableau de résultats de test comme argument. La classe doit avoir une méthode qui renvoie la moyenne des résultats des tests. Si un résultat de test dans le tableau est négatif ou supérieur à 100, la classe doit lever une IllegalArgumentException. Démontrer la classe dans un programme.