

B2 : Atelier 1

Exercice 1:

Ecrivez un programme JAVA qui modélise un point de coordonnées x, y

1. Créer une classe Point avec deux variables d'instance x et y et une méthode publique qui calcul la distance entre deux points
2. Créez une classe TestPoint qui contient la méthode main() qui permet la création de deux objets Point et affiche le résultat.

```
Point p1 = new Point(1, 4);
```

Exercice 2:

Soit la classe suivante :

Chien
nom: String
aboyer (String sens) manger (String qlqc)

Créer une classe de teste qui permet de créer deux chiens.

- Le premier s'appelle "Rex". À l'appel de ses méthodes "aboyer" et "manger" on obtient le Message [**Je suis en colère. Ghhhhhhhhhh !!! WWWWOARF !! WWWWOARF !! - Je mange de la viande**]
- Le deuxième s'appelle "Max". À l'appel de ses méthodes "aboyer" et "manger" on obtient le Message [**Je ne suis pas du tout en colère. Iwiw !! awaw !! - Je mange du poisson**]

Exercice 3:

On a une base de données qui permet de gérer les comptes clients. Cette base de données contient une table qui permet de stocker les informations suivantes :

nom	adresse	solde
El Alami Hassan	Safi	14765.80
Karimi Khalid	Casablanca	7654.00

Avec la programmation procédurale, chaque compte possède des propriétés (nom, adresse et solde). Mais avec l'arrivée de la programmation OO, la manière de présenter les choses a complètement changé. Avec la programmation OO chaque objet – de type Account – a un nom, une adresse et un solde, mais possède aussi ses propres comportements.

nom	adresse	solde	afficherInfosCompte
El Alami Hassan	Safi	14765.80	out.print
Karimi Khalid	Casablanca	7654.00	out.print

Chaque objet « account » a 4 éléments un nom, une adresse, un solde et une façon de présenter les informations du compte

Créer cette classe ayant les 3 propriétés (nom, adresse et solde) et une méthode permettant à chaque compte d'afficher ses informations du solde (permettant d'avoir le même résultat qu'avant).

El Alami Hassan habite à (Safi) - infos solde 14765.8 DH

Karimi Khalid habite à (Casablanca) - infos solde 7654.0 DH

Exercice 4 :

En java, un objet peut avoir une ou plusieurs méthodes avec ou sans paramètres.

Dans la vie courante, lorsqu'on veut acheter une chose (du lait) d'un supermarché, il est possible qu'on demande à une personne d'aller acheter du lait. On envoie cette même personne pour faire une autre chose. Imaginez donc un objet comme une personne qui peut vous servir (répondre à vos messages).

On veut que notre objet, soit capable de calculer l'intérêt.

Donc notre objet account peut (nous servir) et exposer une méthode :

double calculerInteret(double tauxInteret)

Écrire une classe Account possédant les propriétés : nom, adresse, solde et ayant deux méthodes, la première « afficherInfosSolde » permettant d'afficher les informations du solde et la deuxième « calculerInteret » permettant de calculer le solde suivant un taux. Pour chaque client, on veut afficher les informations du solde en plus de ses intérêts.

Données : tauxInteret = 7.0%

Exercice 5:

Reprenons le modèle de notre classe

Account
- nom: String - adresse: String - solde: double

Pour chacune des variables de notre classe « Account » créer les méthodes getters et setters ensuite afficher les informations des comptes.

NB : remarquez qu'on a modifié les modificateurs d'accès de nos propriétés et qu'on a rendu nos variables d'instances inaccessibles.

Exercice 6 :

Application de règles avec des méthodes d'accès.

Même si on a rendu nos propriétés (nom par exemple) inaccessibles, rien n'empêche que l'utilisateur de notre code saisi des valeurs non permises (""). Pour cela on peut envisager d'ajouter des traitements à nos méthodes d'accès pour palier à ce genre de problèmes.

Exemple: **public void** setName(String n) {
 if (!n.equals("")) {
 nom = n;
 }
 }

Soit la classe suivante :

Account
- nom: String - adresse: String - solde: double
+ retirer(double m)

Ajouter une méthode permettant à l'utilisateur de retirer un montant de son solde.

La méthode ne doit pas permettre à l'utilisateur de retirer plus que le montant de son solde.

Exercice 7 : « accesseurs & mutateurs » :

Écrivez un programme JAVA qui affiche les caractéristiques d'un Diplôme

- private String titre;
- private String mention;
- private int anobtenu;

1. Créer deux constructeurs « avec et sans paramètres »
2. Créer les getters et setters nécessaires pour pouvoir modifier les valeurs des attributs