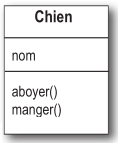
**Atelier 3 : Programmation Orientée Objet Java (1)**

Entrainement à la POO Java avec des mini exercices.

**Contexte :**

**Exercice 1 :**

L’objectif de cet exercice est contrôlé votre chien. Créer la classe suivante :

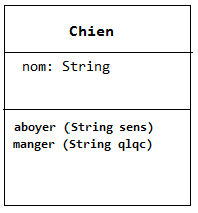


Créer une classe de teste qui permet de créer deux chiens.

* Le premier s’appelle « Rex ». À l’appel de ses méthodes « aboyer » et « manger » on obtient le Message (**Rex dit wwaaaarf. Je mange de la viande)**
* Le deuxième s’appelle « Max ». À l’appel de ses méthodes « aboyer » et « manger » on obtient le Message (**Max dit wwaaaarf. Je mange de la viande)**

**Exercice 2 :**

Soit la classe suivante :

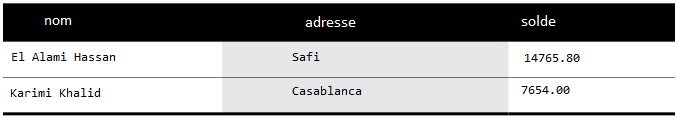


Créer une classe de teste qui permet de créer deux chiens.

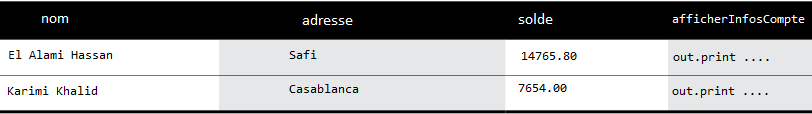
* Le premier s’appelle "Rex". À l’appel de ses méthodes "aboyer" et "manger" on obtient le Message [**Je suis en colère. Ghhhhhhhhhh !!! WWWWOARF !! WWWWOARF !! - Je mange de la viande**]
* Le deuxième s’appelle "Max". À l’appel de ses méthodes "aboyer" et "manger" on obtient le Message [**Je ne suis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



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 à complément changé. Avec la programmation OO chaque objet – de type Account – a un nom, une adresse et un solde, mais possède aussi ses propres comportements.



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 message).

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

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

double calculerInterer(double tauxInteret)

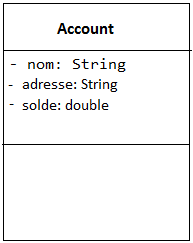
Écrire une classe Account possédant les propriétés : nom, adresse, solde et ayant deux méthodes, la première « afficherInfosSolde » permettant d’afficher les informations du solde et la deuxième « calculerInterer » permettant de calculer le solde suivant un taux.

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



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) {

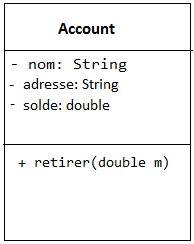
**if** (!n.equals("")) {

nom = n;

}

}

Soit la classe suivante :



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.

**Modalités pédagogiques :**

* Equipe de 2 apprenants, faut respecter respecter les mesures sanitaires (distance, bavette ...)
* Durée 2 jours

Voici les concepts que vous devez appréhender :

* Les Classes
* Les Objets
* Les attributs
* Les modificateurs d’accès
* Les méthodes
* Les mutateurs et les successeurs
* L’encapsulation

**Livrable :**

Un dépôt Git dans lequel on retrouve chaque exercice dans un fichier séparé

**Critères de performance**

Le formateur peut exécuter le code sur sa propre machine.

Le code est livré avant la date et l’heure limite.

**Modalités d'évaluation**

Votre participation dans le groupe sera appréciée.

Une présentation de la démonstration.

**Compétences visées :**

Développer la partie back-end d’une interface utilisateur web