

2 : Atelier 2: Programmation Orientée Objet Java

Entraînement à la POO Java avec des mini exercices.

Exercice 1 :

Écrire une nouvelle classe `Employe` possédant les propriétés : nom, prenom, adresse, anneeNaissance et anneeEncours, et ayant deux méthodes, la première "calculerAge" permettant de calculer l'âge d'un employé et la deuxième "afficherInfosClient" permettant d'afficher ses informations.

Employe
- nom : String - prenom : String - adress : String - anneeNaissance : int - anneeEncours : int
calculerAge() : int afficherInfosClient() : void constructeur()

Créer une classe de test dans le package "test" qui permet de créer trois instances :

El alami Mohammad, né le 1963 habitant à Casablanca, 30 ans

Sbai Fouad, né le 2006 habitant à Salé. 21 ans

Exercice 2:

Une classe **Article** possède deux champs privés : nom et prix, et une méthode polymorphe `afficher()` pour afficher le prix d'un article.

Une classe **ArticleEnSolde** dérive de la classe `Article`. Cette sous-classe comprend le pourcentage de réduction sur le prix d'origine et également la redéfinition de la méthode `affiche()` afin d'afficher le pourcentage de remise sur le prix d'origine.

Exercice 3 :

Au sein d'une banque, chaque adhérent pourra avoir un compte bancaire qui peut être un compte épargne ou un compte courant. La classe compte bancaire comporte les données membres protégées suivantes : numéro du compte (type long) et le solde (type double) et les méthodes publiques suivantes :

- **Ajouter()** : ajout de l'argent dans le compte
- **Retirer()** : retirer de l'argent à partir d'un compte
- **Afficher()** : afficher les caractéristiques d'un compte.

Au sein d'un compte épargne, la valeur minimale du solde est 200 MAD et contient un taux d'intérêt annuel et une méthode permettant de calculer l'intérêt annuel.

Définir les trois classes et tester par la suite ces méthodes dans un programme principal.

Exercice 4 :

Dans un établissement Scolaire, on trouve trois sortes de personnes : Secrétaires, enseignants et étudiants. Chaque personne est caractérisée par son nom et prénom, sa date de naissance, son adresse et sa ville qui sont des attributs communs.

On définit les méthodes public suivantes de la classe **Personne** :

- ✓ Le constructeur **Personne** (String nom, String prenom, String adresse, String ville, dateNaissance) : crée et initialise un objet de type Personne.
- ✓ String **toString** () : fournit une chaîne de caractères correspondant aux caractéristiques (attributs) d'une personne.
- ✓ **modifiePersonne** (String adresse, String ville) : modifie l'adresse et la ville d'une personne
- ✓ **ecrirePersonne**() : pourrait afficher les caractéristiques d'une personne. Elle est déclarée **abstraite**.
- ✓ **Secretaire** (String nom, String prenom, String adresse, String ville, String numeroBureau):
- ✓ Le constructeur **Secretaire** doit fournir les caractéristiques pour construire une Personne, plus les spécificités de la classe Secretaire (numeroBureau).
- ✓ String **toString** () : fournit une chaîne contenant les caractéristiques d'une Secretaire.

De même, un **Enseignant** est une Personne enseignant une spécialité (mathématiques, informatique, anglais, gestion, etc.). Un **Étudiant** est une Personne préparant un diplôme (diplome). Les méthodes pour Enseignant et Étudiant sont similaires à celles de Secretaire. Une variable privée static dans chaque classe compte le nombre de personnes créées dans chaque catégorie.

Questions :

1. Proposez un diagramme de classes UML dans lequel vous préciserez les relations d'héritage.
2. Réalisez le code Java