
FICHE DE TP N°3

Exercice 1

L'objectif de cet exercice est d'écrire un programme Java pour gérer les comptes appartenant aux clients d'une banque.

- Un client est caractérisé par un identifiant, un nom et un numéro de téléphone.
 - Un compte bancaire est caractérisé par un numéro (créé et incrémenté automatiquement) et un solde.
 - Le numéro et le solde d'un compte sont accessibles en lecture seule.
 - A sa création, un compte bancaire a un solde nul.
 - Il est aussi possible de créer un compte en spécifiant son solde initial.
 - L'utilisateur d'un compte peut y faire des dépôts et des retraits en indiquant le montant de l'opération.
 - L'utilisateur peut aussi consulter les informations sur son compte avec une méthode `toString()`.
 - Un compte épargne possède en plus un attribut `taux d'intérêt` et une méthode `calculInteret()` qui permet de mettre à jour le solde en tenant compte des intérêts.
 - Pour un compte courant, chaque opération de retrait et de dépôt est payante (attribut `coût opération`, par défaut 500 F CFA).
1. Définir les classes `Client`, `Compte`, `CompteEpargne` et `CompteCourant`.
 2. Créer un programme pour tester ces classes :
 - créer trois instances de la classe `Client` ;
 - créer une instance de chacune des trois classes `Compte`, `CompteEpargne` et `CompteCourant` ;
 - utiliser les méthodes `deposer` et `retirer` sur chaque compte ;
 - utiliser la méthode `calculInteret` de `CompteEpargne` ;
 - consulter les trois comptes en utilisant la méthode `toString`.

Exercice 2

On veut développer un programme Java permettant de gérer une bibliothèque. La bibliothèque contient des exemplaires d'œuvres écrites par des auteurs. L'objectif est de modéliser chacun de ces éléments.

1. Un auteur est caractérisé par un son nom ainsi qu'une indication permettant de savoir s'il a été primé. La classe `Auteur` doit fournir les méthodes suivantes :
 - a. un constructeur permettant d'initialiser les attributs ;
 - b. les méthodes d'accès aux attributs (getters et setters) ;
2. Une œuvre est caractérisée son titre, l'auteur qui l'a rédigée, l'année de publication de l'œuvre et la langue dans laquelle elle a été rédigée. La classe `Oeuvre` doit définir les méthodes suivantes :
 - a. un constructeur avec comme arguments le titre, l'auteur et la date de publication. La langue par défaut est le "Français" ;

- b. un constructeur permettant d'initialiser tous les attributs ;
 - c. les accesseurs (getters) ;
 - d. une méthode permettant d'afficher les caractéristiques d'une œuvre en respectant strictement le format suivant :
 <titre>, <nom de l'auteur>, en <langue>.
3. Un exemplaire d'une œuvre est caractérisée par un numéro et une référence à l'œuvre. La classe **Exemplaire** comporte les méthodes suivantes :
- a. un constructeur prenant en arguments un numéro et une référence à une œuvre ;
 - b. une méthode **getOeuvre** retournant l'œuvre ;
 - c. une méthode permettant d'afficher une description de l'exemplaire respectant strictement le format suivant :
 Exemplaire <numero> de <titre>, <nom de l'auteur>, en <langue>
4. Une bibliothèque a un nom et contient une liste (tableau) d'exemplaires. Elle possède deux attributs supplémentaire indiquant le nombre effectif et le nombre maximal d'exemplaires. La classe **Bibliotheque** comporte les méthodes suivantes :
- a. un constructeur pour initialiser tous les attributs ;
 - b. une méthode **getNom** retournant le nom de la bibliothèque ;
 - c. une méthode **getNbExemplaires** retournant le nombre d'exemplaires ;
 - d. une méthode **ajouterExemplaire** permettant d'ajouter un exemplaire d'une œuvre dans la bibliothèque ;
 - e. une méthode **listerExemplaires** qui affiche tous les exemplaires d'une œuvre écrite dans une langue donnée ;
 - f. une méthode **compterExemplaires** retournant le nombre d'exemplaires d'une œuvre donnée en paramètre ;
 - g. une méthode **afficherAuteur** avec un paramètre de type booléen, qui affiche les noms des auteurs primés si l'argument vaut true, les auteurs non primés sinon.
5. Créer un programme (classe avec une méthode **main**) pour tester ces différentes méthodes. Exemples de déroulement

```
1 public class TestBibliotheque{
2     public static void main (String args[]){
3         Auteur auteur1 = new Auteur("Camara Laye", true);
4         Auteur auteur2 = new Auteur("Birago Diop", true);
5         Oeuvre o1 = new Oeuvre("L'Enfant noir", auteur1, 1953, "FR");
6         Oeuvre o2 = new Oeuvre("Le Maitre de la parole", auteur1, 1978);
7         Oeuvre o3 = new Oeuvre("Les Contes d'Amadou Koumba", auteur2, 1947);
8         Oeuvre o4 = new Oeuvre("L'Os de Mor Lam", auteur2, 1979);
9         Exemplaire e1 = new Exemplaire(100, o1);
10        Exemplaire e2 = new Exemplaire(101, o1);
11        Exemplaire e3 = new Exemplaire(102, o1);
12        Exemplaire e4 = new Exemplaire(103, o2);
13        Exemplaire e5 = new Exemplaire(104, o3);
14        Exemplaire e6 = new Exemplaire(105, o4);
15        Exemplaire e7 = new Exemplaire(106, o4);
```

```
16 Exemple tabExemp [];  
17 Bibliotheque biblio = new Bibliotheque("BU UASZ", tabExemp, 1000);  
18 biblio.ajouterExemplaire(e1);  
19 biblio.ajouterExemplaire(e2);  
20 biblio.ajouterExemplaire(e3);  
21 biblio.ajouterExemplaire(e4);  
22 biblio.ajouterExemplaire(e5);  
23 biblio.ajouterExemplaire(e6);  
24 biblio.ajouterExemplaire(e7);  
25 biblio.listerExemplaires(o1);  
26 System.out.println(biblio.compterExemplaires(o1)); //3  
27 biblio.afficherAuteur(true);  
28 }  
29 }
```

Exercice 3

L'objectif de cet exercice est d'écrire un programme Java permettant de gérer les employés d'une entreprise d'informatique. Un employé (classe **Employe**) est caractérisé par un numéro, un nom, un revenu mensuel, un taux d'occupation (pourcentage de temps travaillé par mois, exemple : emploi à 80%) et une prime. On distingue trois types d'employés :

- les managers (classe **Manager**), caractérisés en plus par un nombre de jours voyagés et un nombre de nouveaux clients apportés;
- les testeurs (classe **Testeur**), caractérisés par un nombre d'erreurs corrigées;
- les programmeurs (classe **Programmeur**), caractérisés par un nombre de projets réalisés.

1. Définir les différentes classes du programme avec chacune les méthodes suivantes :

- a. un constructeur permettant d'initialiser toutes les valeurs des attributs de la classe. Le taux d'occupation est de 100% par défaut. De plus, si un constructeur reçoit un taux d'occupation inférieur à 10%, le taux de 10% est retenu. De même, si le taux d'occupation dépasse 100%, il devra être limité à 100%. La prime est initialisée à zéro.

La construction d'un employé est accompagné du message :

Un nouvel employé : <...> (voir exemples).

- b. une méthode **double revenuAnnuel()** qui calcule et retourne le salaire annuel (avec 3 chiffres après la virgule) comme suit :

- tout employé a un salaire de base qui vaut la somme de 12 fois son salaire mensuel multiplié par son taux d'occupation et sa prime;
- un manager a, en plus de son salaire de base, un bonus de 50000 F pour chaque client apporté, et de 10000 F pour les dépenses de chaque jour voyagé;
- un testeur a, en plus de son salaire de base, un bonus de 10000F F pour chaque erreur corrigée;
- un programmeur a, en plus de son salaire de base, un bonus de 20000 F pour chaque projet achevé.

NB : pour les bonus, utiliser des constantes pour fixer leurs valeurs.

- c. une méthode `String toString()` permettant de représenter les informations d'un employé sous forme d'une chaîne de caractères (voir exemples). La prime est prise en compte seulement si elle n'est pas nulle.
- d. une méthode `void demandePrime()` qui :
- demande à l'employé de saisir le montant de la prime souhaitée ;
 - redemande le montant tant que la donnée saisie est trop grande (l'employé ne peut demander plus de 2% de son salaire annuel) ou que la donnée saisie est non numérique.
- L'employé n'a droit qu'à 5 tentatives de saisie. Si après 5 tentatives le montant de la prime est incorrect, la prime de l'employé reste à zéro. Sinon la prime de l'employé vaudra le montant saisi.
2. Créer un programme (classe avec une méthode `main`) pour tester ces différentes méthodes. Exemples de déroulement :

```
Un nouvel employé : 1213, Modou Sylla, c'est un manager.
Un nouvel employé : 1214, Fatou Diop, c'est un programmeur.
Un nouvel employé : 1215, Sény Diatta, c'est un testeur.
Affichage des employés :
  1213 Modou Sylla :
    Taux d'occupation : 100%. Salaire annuel : 4500000.00 F CFA.
    A voyagé 30 jours et a apporté 4 nouveaux clients.
  1214 Fatou Diop :
    Taux d'occupation : 80%. Salaire annuel : 3580000.00 F CFA.
    A réalisé 3 projets.
  1215 Sény Diatta :
    Taux d'occupation : 60%. Salaire annuel : 2450000.00 F CFA.
    A corrigé 124 erreurs.
Montant de la prime souhaitée par Modou Sylla ?
300000
Trop cher!
Montant de la prime souhaitée par Modou Sylla ?
250000
Trop cher!
Montant de la prime souhaitée par Modou Sylla ?
60000
Affichage après demande de prime :
  1213 Modou Sylla :
    Taux d'occupation : 100%. Salaire annuel : 4560000.00 F CFA,
    Prime : 60000.00.
    A voyagé 30 jours et apporté 4 nouveaux clients.
```