



Consignes :

- Vous travaillerez dans un répertoire evaluations/int03.
- Vous pouvez utiliser toutes les notes que vous voulez ainsi que les programmes qui se trouvent dans votre répertoire.
- Cette interrogation est à faire exclusivement sous Linux1.
- Vous disposez de **2 heures** exactement.
- Lors de la correction de l'interrogation, nous serons très attentifs au style (indentation, choix des noms des variables, etc. ...).
- L'interrogation est individuelle, toute fraude ou tentative de fraude entraîne des sanctions disciplinaires (**Paragraphe 8.6 du R.E.E**).

Objectif du programme:

L'objectif est d'écrire une petite application qui simule la gestion d'une pseudo agence bancaire.

1 – La classe Client (6 points)

Ecrivez la classe Client dont les attributs privés sont :

- numCli : une chaîne de caractères représentant le numéro d'identification du client ;
- nom : une chaîne de caractères ;
- prénom : une chaîne de caractères ;
- dateInscrit : qui est la date d'inscription du client à l'agence bancaire ;

et les méthodes :

- un constructeur à 4 paramètres ;
- les assesseurs ;
- la méthode déposer(double unMontant, Compte monCompte) qui ne retourne rien et met à jour les avoirs du compte ;
- la méthode retirer(double unMontant, Compte monCompte) qui ne retourne rien et met à jour les avoirs du compte ; le retrait est possible si le solde est suffisant ;
- la méthode String toString() : vous afficherez le numCli, le nom, le prénom et la date d'inscription.

2 – La classe Compte (3 points)

Ecrivez la classe Compte dont les attributs privés sont :

- numero: une chaîne de caractères qui représente le numéro de compte ;
- type : une chaîne de caractères qui représente le type de compte ;
- solde : de type double qui représente les avoirs du compte ;
- propriétaire : de type Client qui représente le propriétaire du compte ;

et les méthodes :

- un constructeur à 4 paramètres ;
- les assesseurs ;
- la méthode String toString() : Vous affichez le numéro de compte, le solde, le nom du propriétaire et le numéro d'identification du client.

3 – La classe AgBancaire (6 points)

Ecrivez la classe AgBancaire dont l'unique attribut privé est :

- la collection des comptes bancaires ;

et les méthodes :

- un constructeur ;
- la méthode ajouterUnCompte(Compte unCompte) qui ne retourne rien et permet d'ajouter un compte donné à la collection ;
- la méthode supprimerUnCompte(Compte unCompte) qui ne retourne rien et permet de supprimer un compte donné ;
- la méthode connaitreLesAvoirs() qui retourne un double représentant le total des avoirs de l'agence bancaire (la somme de tous les soldes des comptes) ;
- la méthode publierLesComptes() qui ne retourne rien et permet la publication de la liste de tous les comptes de l'agence bancaire

Contraintes : Pour l'implémentation de cette classe, nous exigeons l'utilisation d'une ArrayList. Tout parcours de la collection se fera en utilisant un Iterator.

4 – Test unitaire (JUnit) (3 points)

Ecrivez une classe qui permet de réaliser un test unitaire de la classe AgBancaire. Testez uniquement :

- la méthode ajouterUnCompte(...) ;
- la méthode supprimerUnCompte(...) ;
- la méthode connaitreLesAvoirs() ;

5 – La Javadoc (2 points)

Ajouter une documentation digne de ce nom à votre classe Client. On se limitera à ce qui suit :

- constructeur ;
- méthode déposer() ;
- méthode retirer() ;