TP 4. CONCEPTS OBJET : PACKAGES MULTIPLES ET COMPOSITION

EXERCICE 1 VISIBILITE

- Dans `TestCompteCourant.main()`, modifier les valeurs des soldes des 3 comptes courants précédemment créés.
- Modifier la visibilité des attributs de CompteCourant pour qu'ils soient tous privés. Que se passe-t-il ?
- Pour résoudre ce problème, créer dans CompteCourant les getters / setters pour les attributs. Attention, il faut que cela ait un sens au niveau de l'application… il y a au moins deux attributs pour lesquels un setter n'est pas judicieux :
  - `nbComptesCourants` (car il doit être incrémenté seulement par le constructeur)
  - `solde` car un compte ne change pas de solde, il est le support d'opérations (cf méthodes crediter et debiter déjà définies)
- Modifier `TestCompteCourant.main()` en conséquence. Afficher le nombre de comptes courants à la fin de la méthode `main(...)`.


Tester.

EXERCICE 2 PACKAGES MULTIPLES

- Créer un package `com.bankonet.test`. Déplacer la classe `TestCompteCourant` dans `com.bankonet.test`. Que se passe-t-il ?
- Effectuer les modifications nécessaires.

Situation actuelle

CompteCourant
private String numero private String intitule private double solde private double montantDecouvertAutorise private static int nbComptesCourants
public CompteCourant() public CompteCourant(String, String, double, double) public String toString() public void debiter(double) public void crediter(double)

NB : les méthodes getters et setters (client.getNom()…) n'apparaîtront plus sur les diagrammes de classes. Il faut bien entendu les créer.

EXERCICE 3 COMPOSITION

- ⇨ Dans la classe `CompteCourant`, supprimer l'attribut `nbComptesCourants` et modifier les méthodes qui la référencent.
- ⇨ Ajouter dans le projet la classe `CompteEpargne` . Un compte épargne a un taux d'intérêts, mais pas de découvert autorisé.

Nom	Type	static
Numero	String	non
Intitule	String	non
Solde	double	non
tauxInteret *(exprimé en %)*	double	non

⇨ Définir une classe `Client` avec les attributs privés suivants :

Nom	Type	static
Identifiant	String	Non
Nom	String	Non
Prenom	String	Non
compteCourant	CompteCourant	Non
compteEpargne	CompteEpargne	Non

⇨ Définir un constructeur `Client(...)`. Ses 5 arguments correspondant aux attributs de la classe `Client`.

⇨ Définir une méthode publique `calculerAvoirGlobal()` qui retourne le montant des soldes cumulés des différents comptes du client (pour l'instant le compte courant et le compte épargne). Ne pas oublier de tester si le client possède bien un compte courant et/ou un compte épargne (la non-existence étant représentée par null pour les attributs).

⇨ Définir une méthode `toString()` qui permet d'avoir un rendu textuel du client, à savoir le nom et le prénom d'un client (utiliser la concaténation de chaînes de caractères).

⇨ Dans le package `com.bankonet.test`, Définir une classe `TestClient`. Sa méthode main(…) crée `clientTab`, un tableau de clients de taille 3 :

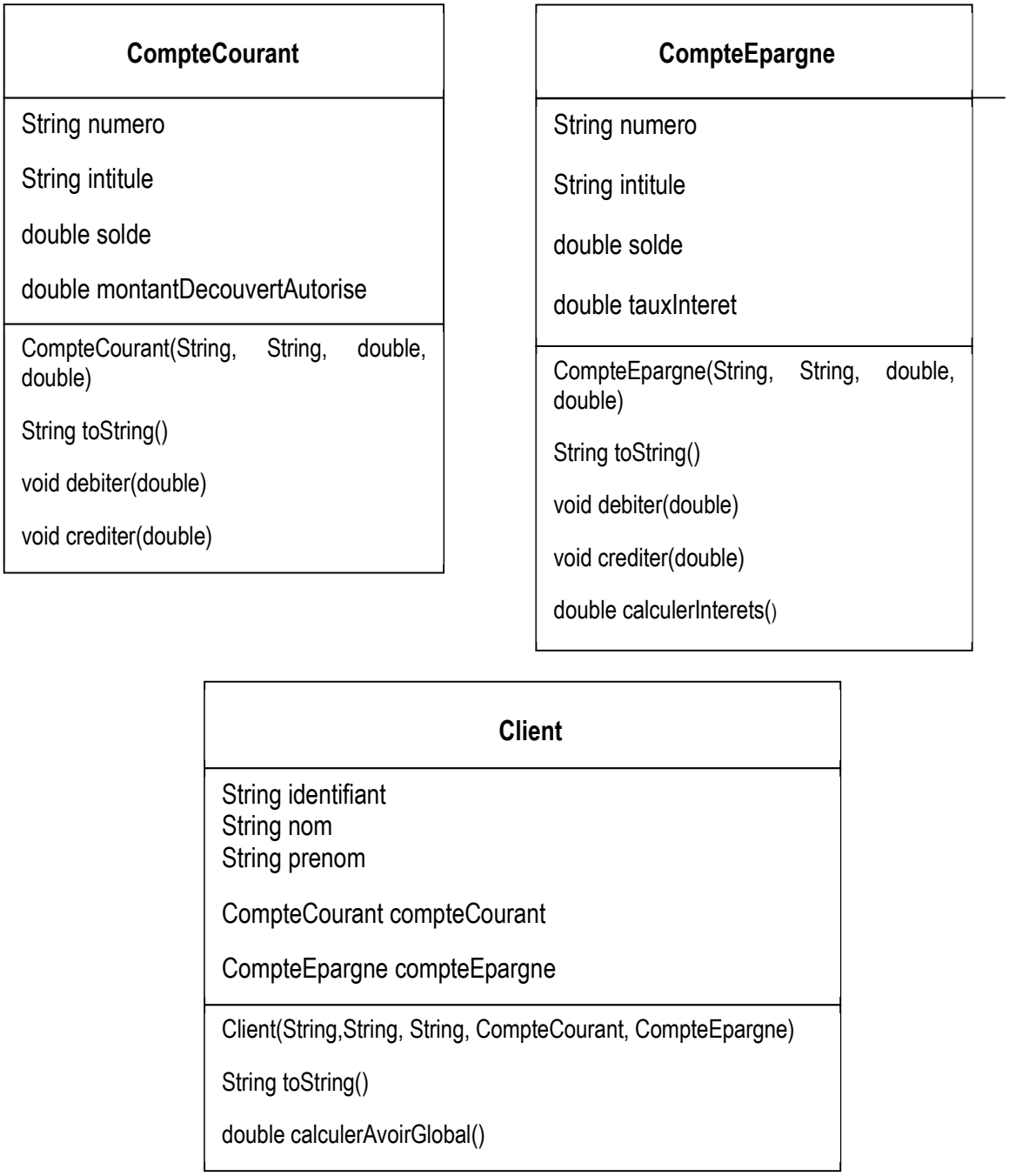
- `clientTab[0]` possède 1 compte courant et 1 compte épargne.
- `clientTab[1]` possède 1 compte courant et 1 compte épargne.
- `clientTab[2]` possède juste 1 compte courant.

⇨ Parcourir `clientTab` à l'aide d'une boucle for. Pour chacun des clients :

- Afficher son compte courant et son compte épargne s'ils existent.
- Afficher son avoir global, par appel à la méthode à `calculerAvoirGlobal()`.

 Tester.

Situation actuelle



Note : Sauf cas particuliers, les modificateurs de visibilité n'apparaissent plus dans les schémas (on considère notamment que les attributs sont tous privés).