

## Travaux Dirigés : Programmation Orienté Objet

**Exercice 1** : Créer la classe Client correspondant au schéma UML suivant

<b>Client</b>
nom : String prenom : String numeroTel : long
<b>Client</b> (nom : String, prenom: String, numero: long) <b>getNom</b> (): String <b>getPrenom</b> (): String <b>getNumeroTel</b> (): String <b>setNumeroTel</b> (numero :long): void <b>saisir</b> (): void <b>afficher</b> (): void

Travail à faire :

- Déclarer des attributs nom, prenom et numeroTel comme dans le tableau ci- dessus. Ces attributs sont private pour assurer une bonne encapsulation.
- Définir un constructeur prenant en paramètres nom, prenom et numeroTel.
- Définir des accesseurs getNom(), getPrenom() et getNumeroTel() qui retournent respectivement les valeurs des attributs nom, prenom et numeroTel.
- Définir un mutateur (manipulateur) setNumeroTel() qui permet de modifier la valeur d'attribut numeroTel.
- Ajouter la méthode saisir() qui permet de saisir les informations d'un client.
- Ajouter la méthode afficher() qui permet d'afficher les informations d'un client.

**Exercice 2** : le panier d'achat

1. Écrire une classe ShoppingCart modélisant un panier d'articles dans lequel on peut:

- Ajouter un article avec addItem();
- Retirer un article avec removeItem(), qui devra renvoyer false si l'item que l'on essaye de supprimer n'existe pas;
- Connaître le nombre d'articles avec itemCount();
- Calculer le prix total du panier avec totalPrice() (vous écrirez en commentaire au début de la méthode quel est l'ordre de grandeur de la complexité de cette méthode);
- Indication: utilisez une collection de java.util pour stocker les articles du panier.

Écrire une classe Main permettant de tester la création d'un panier de plusieurs articles.

2. On souhaite que chaque panier d'achat créé puisse disposer automatiquement à sa création d'un numéro de série unique (qui commence à 1 et qui est incrémenté de 1 à chaque nouveau panier créé), et qui soit connu comme l'identifiant (id) de ce panier. Ajoutez une méthode getId() à la classe ShoppingCart qui retourne cet entier de type int, et tout ce dont vous avez besoin pour l'implémenter.

3. Ajouter à la classe ShoppingCart une méthode toString() qui retourne une représentation du contenu du panier, commençant par l'identifiant unique du panier et le nombre d'articles contenus, puis affichant tous les articles du panier, un article par ligne.

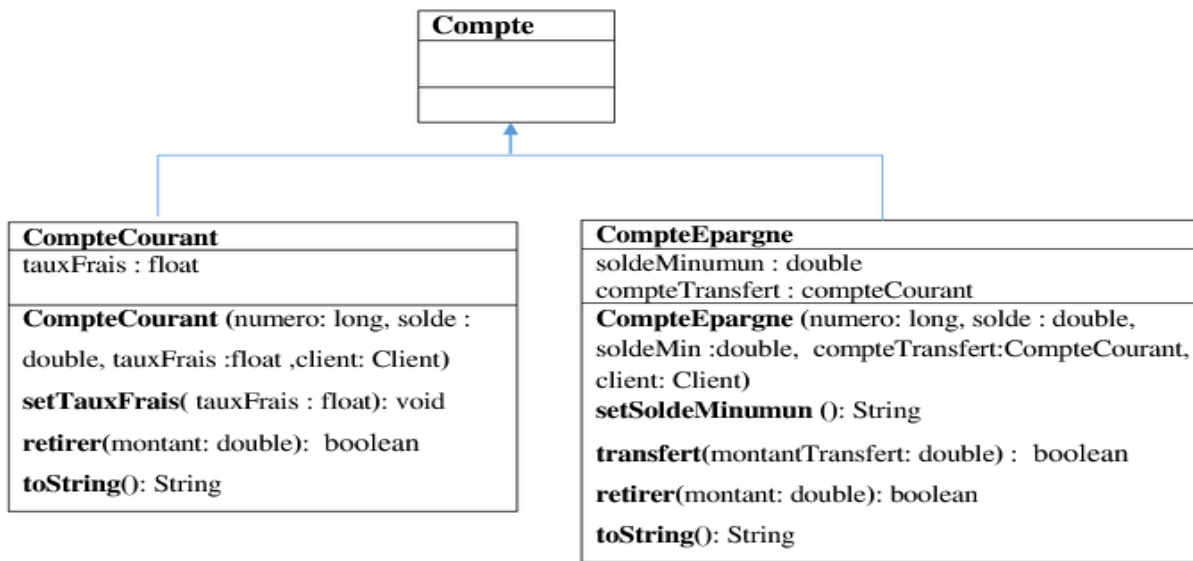
### **Exercice 3 :**

Dans la classe compte, nous redéfinissons la méthode toString() héritée de la classe Object (la classe Object la racine de l'arborescence d'héritage). La méthode toString() retourne « Compte numéro 12121 avec solde 120000 UM ». En fait, nous pouvons distinguer 2 types de comptes spécialisés :

- Compte courant : un compte destiné aux transactions quotidiennes.
- Compte d'épargne : un compte de dépôt rémunérés qui permet des transactions limitées.

Créer les sous-classes CompteCourant et CompteEpargne héritées de la super-classe

Compte. Utiliser le schéma UML ci-dessous pour créer les sous-classes :



#### 1- Compte courant

- Créer une classe CompteCourant qui hérite de la classe Compte.
- Ajouter un attribut tauxFrais: int qui représente le taux de frais qui doit être soustraits du solde à chaque opération de retrait.
- Redéfinir la méthode retirer tel que le frais soustrait du solde.
- Redéfinir la méthode toString() tel que l'affichage devient: « Compte courant numéro ... avec solde ... ».

#### 2- Compte d'épargne

- Créer une classe CompteEpargne qui hérite de la classe Compte.
- Ajouter un attribut soldeMinumun: double qui représente le solde minimum requis pour maintenir un compte d'épargne.
- Définir un attribut compteCourant permettant de faire référence au compte courant à partir duquel les transferts seront effectués.
- Définir une méthode transfert() permettant de retirer du compte courant en utilisant le montant transféré.
- Redéfinir la méthode retirer telle que le solde d'un compte d'épargne ne soit pas inférieur à soldeMinumun.
- Redéfinir la méthode toString() telle que l'affichage devient:« Compte d'épargne numéro ... avec solde ... ».