

JAVA-TP07: banque

I. Introduction	1
II. Exercice 1	2
III. Exercice 2	4
IV. Conclusion	8

I. Introduction

Dans ce TP, nous allons voir comment créer un système de création d'une "bibliothèque" ainsi qu'un compte en banque en Java et en utilisant des attributs privés

II. Exercice 1

1)

```
private String titre; // attributs privée
private String auteur; // attributs privée
private int prix; // attributs privée
```

```
public String getTitre() {
    return titre;
}

public String getAuteur() {
    return auteur;
}

public int getPrix() {
    return prix;
}

public void setTitre(String titre) {
    this.titre = titre;
}

public void setAuteur(String auteur) {
    this.auteur = auteur;
}

public void setPrix(int prix) {
    if (prix >= 0) { //si prix supérieur ou égal à 0 pour forcer a avoir un nombre positif
        this.prix = prix;
    } else { //sinon
        System.out.println("Le prix ne peut pas être négatif."); //...
    }
}
```

2)

```
public livre(String titre, String auteur, int prix) {
    this.titre = titre; //constructeur
    this.auteur = auteur; //constructeur
    this.prix = prix; //constructeur
}
```

3)

```
System.out.print(s: "Donner le titre : ");
String titreSaisie = scanner.nextLine(); // rentrer la valeur dans la variable
System.out.print(s: "Donner l'auteur : ");
String auteurSaisie = scanner.nextLine(); // rentrer la valeur dans la variable
System.out.print(s: "Donner le prix : ");
int prixSaisie = scanner.nextInt(); // rentrer la valeur dans la variable
scanner.nextLine();

livre monLivre = new livre(titreSaisie, auteurSaisie, prixSaisie); // intégrer monLivre dans livre
// titre = titreSaisie, auteur = auteurSaisie...
```

4)

```
public void afficher() {
    System.out.println("Titre : " + titre + ", Auteur : " + auteur + ", Prix : " + prix + " euros");
}
```

5)

```
public static void main(String[] args) {
    String choix = "o";
    int livre = 1;

    Scanner scanner = new Scanner(System.in);

    do { // système pour rajouter un livre
        System.out.print(s: "Donner le titre : ");
        String titreSaisie = scanner.nextLine(); // rentrer la valeur dans la variable
        System.out.print(s: "Donner l'auteur : ");
        String auteurSaisie = scanner.nextLine(); // rentrer la valeur dans la variable
        System.out.print(s: "Donner le prix : ");
        int prixSaisie = scanner.nextInt(); // rentrer la valeur dans la variable
        scanner.nextLine();

        livre monLivre = new livre(titreSaisie, auteurSaisie, prixSaisie); // intégrer monLivre dans livre
        // titre = titreSaisie, auteur = auteurSaisie...

        System.out.println("livre " + livre + " :");
        monLivre.afficher(); // afficher monLivre (donc afficher livre)

        System.out.print(s: "Voulez-vous enregistrer un deuxième livre ? (n = non / o = oui) ");
        choix = scanner.nextLine();
        livre = livre + 1;
    } while (choix.equals(anObject: "o"));
}
```

résultat:

```
Donner le titre : harry
Donner l'auteur : jk
Donner le prix : 19
livre 1 :
Titre : harry, Auteur : jk, Prix : 19 euros
Voulez-vous enregistrer un deuxième livre ? (n = non / o = Oui) o
Donner le titre : hunger
Donner l'auteur : ekfsokdof
Donner le prix : 15
livre 2 :
Titre : hunger, Auteur : ekfsokdof, Prix : 15 euros
Voulez-vous enregistrer un deuxième livre ? (n = non / o = Oui) n
PS C:\Users\eleve\Desktop\TP-POO> █
```

III. Exercice 2

1 et 2:

```
public static class Client {

    private String CIN;
    private String Nom;
    private String Prenom;
    private int tel;

    // Constructeur de la classe Client
    public Client(String CIN, String Nom, String Prenom, int tel) {
        this.CIN = CIN;
        this.Nom = Nom;
        this.Prenom = Prenom;
        this.tel = tel;
    }

    // Getters et Setters pour chaque attribut
    public String getCIN() {
        return CIN;
    }

    public void setCIN(String CIN) {
        this.CIN = CIN;
    }

    public String getNom() {
        return Nom;
    }

    public void setNom(String Nom) {
        this.Nom = Nom;
    }

    public String getPrenom() {
```

3)

```
public compte(String CIN, String Nom, String Prenom, int tel) {  
    this.CIN = CIN; // Initialisation de l'attribut CIN  
    this.Nom = Nom; // Initialisation de l'attribut Nom  
    this.Prenom = Prenom; // Initialisation de l'attribut Prenom  
    this.tel = tel; // Initialisation de l'attribut tel  
}
```

4)

```
public compte(String CIN, String Nom, String Prenom) {  
    this.CIN = CIN;  
    this.Nom = Nom;  
    this.Prenom = Prenom;  
    this.tel = 0; // Valeur par défaut pour tel  
}
```

5)

```
public void afficher() {  
    System.out.println("CIN : " + CIN + "Nom : " + Nom + "Prenom : " + Prenom + "téléphone" + tel);  
}
```

6)

```
// Constructeur avec solde  
public compte(String CIN, String Nom, String Prenom, double solde) {  
    this(CIN, Nom, Prenom, tel: 0, solde); // Appel au constructeur avec téléphone par défaut à 0  
}  
  
// Constructeur sans solde (initialisé à 0)  
public compte(String CIN, String Nom, String Prenom) {  
    this(CIN, Nom, Prenom, tel: 0, solde: 0); // Appel au constructeur avec téléphone et solde par défaut à 0  
}
```

7)

```
private final int nbCompte; // Numéro unique pour chaque compte (en lecture seule)
private double solde; // Solde du compte (en lecture seule)

// Getter pour Solde (en lecture seule)
public double getSolde() {
    return solde;
}

// Getter pour Numéro de compte (en lecture seule)
public int getNbCompte() {
    return nbCompte;
}
```

8)

```
// Constructeur sans solde (initialisé à 0)
public Compte(String CIN, String Nom, String Prenom) {
    this(CIN, Nom, Prenom, tel: 0, solde: 0); // Appel au constructeur avec téléphone et solde par défaut à 0
}

public static void main(String[] args) {
    // Crédit d'un compte avec CIN, Nom, Prénom, Téléphone, et Solde
    Compte compteUtilisateur = new Compte(CIN: "EE111222", Nom: "Salim", Prenom: "Omar", tel: 61111111, solde: 0.0);
}
```

9)

```
// Méthode pour créditer le compte
public void crediter(double somme) {
    solde += somme;
    System.out.println("*****");
    System.out.println("Donner le montant à déposer : " + somme);
    System.out.println("Opération bien effectuée\n");
    afficherDetails();
}

// Méthode pour débiter le compte
public void debiter(double somme) {
    solde -= somme;
    System.out.println("*****");
    System.out.println("Donner le montant à retirer : " + somme);
    System.out.println("Opération bien effectuée\n");
    afficherDetails();
}

// Méthode pour afficher les détails du compte
public void afficherDetails() {
    System.out.println("Détails du compte :");
    System.out.println("*****");
    System.out.println("Numéro de compte : " + nbCompte);
    System.out.println("Solde de compte : " + solde);
    client.afficherClient(); // Affiche les détails du client
}
```

10)

```
// Méthode main pour tester les fonctionnalités
Run | Debug
public static void main(String[] args) {

    Client client1 = new Client(CIN: "EE111222", Nom: "Salim", Prenom: "Omar", tel: 611111111); // Crédit d'un client

    Compte compte1 = new Compte(client1, solde: 0.0); // Crédit d'un compte pour ce client avec un solde initial de 0

    compte1.afficherDetails(); // affichage détail du compte

    compte1.crediter(somme: 5000); // Crédit de 5000
    compte1.debiter(somme: 1000); // Débiter 1000
}
```

IV. Conclusion

j'ai trouvé ce TP relativement compliqué car la partie des constructeurs est un peu compliquée à comprendre. Les parties fonction comme "afficherDetails" ont été plus facile à créer, mais l'ensemble a quand même été relativement long à faire.