

# Compte rendu de TP

---

Sujet : TP BDD

Nom : MEDJAHED

Enseignant : Mme Beauvallet

## **Sommaire**

|  |          |
|--|----------|
| <b>Sommaire.....</b>                             | <b>2</b> |
| <b>Introduction.....</b>                         | <b>3</b> |
| <b>Exercice 1 : Classe Livre.....</b>            | <b>3</b> |
| <b>Exercice 2 : Classe Client et Compte.....</b> | <b>3</b> |
| <b>Conclusion.....</b>                           | <b>3</b> |

## Introduction

Dans ce travail, j'ai mis en pratique la Programmation Orientée Objet en Java à travers deux exercices progressifs.

L'objectif était de découvrir comment créer et manipuler des classes, des objets, des attributs, ainsi que des méthodes pour gérer et afficher des informations.

J'ai commencé par une première classe simple (*Livre*), puis je suis passé à quelque chose de plus complet avec les classes *Client* et *Compte*, afin de simuler des opérations bancaires.

Ce projet m'a permis de mieux comprendre comment fonctionnent les objets et comment ils interagissent entre eux.

## Exercice 1 : Classe Livre

Dans ce premier exercice, j'ai dû créer une classe Livre contenant trois attributs :

Titre, Auteur et Prix.

Ensuite, j'ai ajouté un constructeur permettant de saisir les informations d'un livre à partir de l'utilisateur, puis des méthodes d'accès (getters) ainsi qu'une méthode *afficher()* pour afficher clairement les données du livre.

```
1 package poo;
2 import java.util.Scanner;
3
4 public class Livre {
5
6     private String titre;
7     private String auteur;
8     private double prix;
9     private Scanner sc;
10
11     // Constructeur avec saisie utilisateur
12     public Livre(int num) {
13         sc = new Scanner(System.in);
14
15         System.out.println("\nLivre " + num + ":");
16         System.out.print("Donner le titre: ");
17         this.titre = sc.nextLine();
18         System.out.print("Donner l'auteur: ");
19         this.auteur = sc.nextLine();
20         System.out.print("Donner le prix ");
21         this.prix = sc.nextDouble();
22         sc.nextLine();
23     }
24
25     // Getters
26     public String getTitre() { return titre; }
27     public String getAuteur() { return auteur; }
28     public double getPrix() { return prix; }
29
30     // Affichage strictement au format demandé
31     public void afficher() {
32         System.out.println("Le titre est " + titre);
33         System.out.println("L'auteur est " + auteur);
34         System.out.println("Le prix est " + prix);
35         System.out.println("Titre: " + titre + ", Auteur: " + auteur + ",Prix: " + prix);
36     }
37 }
```

```
package poo;

public class TestLivre {
    public static void main(String[] args) {

        Livre livre1 = new Livre(1);
        livre1.afficher();

        Livre livre2 = new Livre(2);
        livre2.afficher();
    }
}
```

## Résultat

```
Livre 1:  
Donner le titre: Programmer en C  
Donner l'auteur: Claude Delannoy  
Donner le prix: 350  
Le titre est Programmer en C  
L'auteur est Claude Delannoy  
Le prix est 350.0  
Titre: Programmer en C, Auteur: Claude Delannoy,Prix: 350.0

Livre 2:  
Donner le titre: Programmer en Java  
Donner l'auteur: Claude Delannoy  
Donner le prix: 450  
Le titre est Programmer en Java  
L'auteur est Claude Delannoy  
Le prix est 450.0  
Titre: Programmer en Java, Auteur: Claude Delannoy,Prix: 450.0
```

## Exercice 2 : Classe Client et Compte

### Partie information sur le client

```
package poo;

public class Client {

    private String CIN;
    private String nom;
    private String prenom;
    private String tel;

    // 3) Constructeur complet (tous les attributs)
    public Client(String CIN, String nom, String prenom, String tel) {
        this.CIN = CIN;
        this.nom = nom;
        this.prenom = prenom;
        this.tel = tel;
    }

    // 4) Constructeur partiel (sans téléphone)
    public Client(String CIN, String nom, String prenom) {
        this.CIN = CIN;
        this.nom = nom;
        this.prenom = prenom;
        this.tel = "Non renseigné";
    }

    // 2) Méthodes d'accès (getters et setters)
    public String getCIN() { return CIN; }
    public String getNom() { return nom; }
    public String getPrenom() { return prenom; }
    public String getTel() { return tel; }

    public void setTel(String tel) { this.tel = tel; }

    // 5) Méthode Afficher()
    public void afficher() {
        System.out.println("\n--- Informations Client ---");
        System.out.println("CIN      : " + CIN);
        System.out.println("Nom      : " + nom);
        System.out.println("Prénom   : " + prenom);
        System.out.println("Téléphone : " + tel);
    }
}
```

## Les informations du compte

```
private static int compteur = 0;
private int code;
private double solde;
private Client proprietaire;

// 8) Constructeur avec propriétaire
public Compte(Client proprietaire) {
    this.code = ++compteur; // 6) Code auto-incrémenté
    this.proprietaire = proprietaire;
    this.solde = 0; // solde initial à 0
}

// 7) Accès (lecture seule pour solde et code)
public int getCode() { return code; }
public double getSolde() { return solde; }
public Client getProprietaire() { return proprietaire; }

// 9) Méthodes demandées

// Crédit simple
public void crediter(double montant) {
    solde += montant;
}

// Crédit + débit compte paramètre
public void crediter(double montant, Compte autre) {
    solde += montant;
    autre.solde -= montant;
}

// Débit simple
public void debiter(double montant) {
    solde -= montant;
}

// Débit + crédit compte paramètre
public void debiter(double montant, Compte autre) {
    solde -= montant;
    autre.solde += montant;
}

// Résumé
public void afficher() {
    System.out.println("\n----- Compte N°" + code + " -----");
    System.out.println("Propriétaire : " + proprietaire.getNom() + " " + proprietaire.getPrenom());
    System.out.println("Solde actuel : " + solde + " €");
}
```

## Les données clients

```
package poo;

public class TestCompte {
    public static void main(String[] args) {

        Client c1 = new Client("CD152", "Ahmed", "Nadir", "0612345678");
        Client c2 = new Client("AB995", "Youssef", "Hakim");

        Compte compte1 = new Compte(c1);
        Compte compte2 = new Compte(c2);

        compte1.crediter(600);
        compte2.crediter(300);
        compte1.debiter(150);
        compte1.crediter(100, compte2); // transfert compte2 -> compte1

        compte1.afficher();
        compte2.afficher();
    }
}
```

## Conclusion

Ce projet m'a permis de mieux comprendre la programmation orientée objet.

J'ai appris à créer des classes, à utiliser des attributs privés, des constructeurs, des méthodes d'accès et d'affichage.

La classe *Livre* m'a servi de base, puis la partie *Client / Compte* m'a permis de voir comment plusieurs objets peuvent travailler ensemble.

J'ai aussi compris comment gérer un solde bancaire, comment le modifier, et même comment transférer de l'argent d'un compte à un autre.

Cette activité m'a vraiment aidé à progresser en Java, et me donne une meilleure maîtrise de la POO pour les prochains projets.