

Samedi, 09 novembre 2024

NOM : **TITTI TCHIEWOU**

PRENOM : **WILLY NELROD**

MATRICULE : **23103132**

GI2 FI2 (E302)

Contrôle continu pratique

A-GI311 PROGRAMMATION ORIENTEE

EXERCICE1 : banques

Pour transformer ce programme en un programme orienté objet, nous allons structurer le code en utilisant des classes pour mieux représenter les concepts comme Client et Compte. Voici comment nous pouvons procéder :

- Etudiez le fonctionnement du programme

1. Déclaration des variables :

- taux1 et taux2 : taux d'intérêt pour les comptes privés et d'épargne respectivement.
- nom1, ville1, solde1PremierClient, solde2PremierClient : informations sur le premier client.
- nom2, ville2, solde1DeuxiemeClient, solde2DeuxiemeClient : informations sur le deuxième client.

2. Affichage des informations des clients :

- La méthode afficherClient() prend en paramètres le nom, la ville, le solde du compte privé et le solde du compte d'épargne d'un client, et affiche ces informations.
- Cette méthode est appelée une fois pour le premier client, puis une fois pour le deuxième client.

3. Calcul des nouveaux soldes avec les intérêts :

- La méthode bouclierCompte() prend en paramètres le solde initial et le taux d'intérêt.
- Elle calcule les intérêts en multipliant le solde par le taux d'intérêt, puis ajoute ces intérêts au solde initial pour obtenir le nouveau solde.
- Le nouveau solde est ensuite retourné par la méthode.

4. Mise à jour des soldes :

- Les nouveaux soldes avec les intérêts sont calculés pour les comptes privés et d'épargne des deux clients, mais ne sont pas affichés dans le programme.

Programme orienté objet sous le nom **banque2.java**:

// Classe Compte

```
private double solde;
private double tauxInteret;

public Compte(double solde, double tauxInteret) {
    this.solde = solde;
    this.tauxInteret = tauxInteret;
}

public double calculerNouveauSolde() {
    double interets = solde * tauxInteret;
    return solde + interets;
}

public double getSolde() {
    return solde;
}
}
```

// Classe Client

```

private String nom;
private String ville;
private Compte comptePrive;
private Compte compteEpargne;

public Client(String nom, String ville, double soldePrive, double soldeEpargne,
double tauxInteretPrive, double tauxInteretEpargne) {
    this.nom = nom;
    this.ville = ville;
    this.comptePrive = new Compte(soldePrive, tauxInteretPrive);
    this.compteEpargne = new Compte(soldeEpargne, tauxInteretEpargne);
}

public void afficherInformations() {
    System.out.println("Client " + nom + " de " + ville);
    System.out.println("Compte privé: " + comptePrive.getSolde() + " francs");
    System.out.println("Compte d'épargne: " + compteEpargne.getSolde() + " francs");
}

public void mettreAJourSoldes() {
    double nouveauSoldePrive = comptePrive.calculerNouveauSolde();
    double nouveauSoldeEpargne = compteEpargne.calculerNouveauSolde();
    comptePrive.solde = nouveauSoldePrive;
    compteEpargne.solde = nouveauSoldeEpargne;
}
}

```

// Classe principale Banque2

```

public static void main(String[] args) {
    Client client1 = new Client("Pedro", "Geneve", 1000.0, 2000.0, 0.01, 0.02);
    Client client2 = new Client("Alexandra", "Lausanne", 3000.0, 4000.0, 0.01, 0.02);

    client1.afficherInformations();
    client2.afficherInformations();

    client1.mettreAJourSoldes();
    client2.mettreAJourSoldes();

    client1.afficherInformations();
    client2.afficherInformations();
}
}

```

// Créer les comptes et les clients

Compte comptePrive1 = new Compte(1000.0, tauxPrive);

Compte compteEpargne1 = new Compte(2000.0, tauxEpargne);

Client client1 = new Client("Pedro", "Geneve", comptePrive1, compteEpargne1);

```

Compte comptePrive2 = new Compte(3000.0, tauxPrive);
Compte compteEpargne2 = new Compte(4000.0, tauxEpargne);
Client client2 = new Client("Alexandra", "Lausanne", comptePrive2, compteEpargne2);

// Afficher les informations des clients avant le calcul des intérêts
System.out.println("Avant application des intérêts :");
client1.afficherInfo();
client2.afficherInfo();

// Appliquer les intérêts
client1.appliquerInteretAuxComptes();
client2.appliquerInteretAuxComptes();

// Afficher les informations des clients après le calcul des intérêts
System.out.println("\nAprès application des intérêts :");
client1.afficherInfo();
client2.afficherInfo();
}
}

```

Explication du code :

- **Classe Compte** : Elle contient le solde et le taux d'intérêt d'un compte. La méthode `appliquerInteret` calcule les intérêts et met à jour le solde.
- **Classe Client** : Elle regroupe les informations d'un client (nom, ville) et ses deux comptes (privé et épargne). La méthode `appliquerInteretAuxComptes` applique les intérêts aux deux comptes.
- **Classe Banque2** : C'est la classe principale qui crée les objets `Client` et `Compte`, affiche les informations avant et après l'application des intérêts.