

Corrigé des exercices sur les classes

Exercice 1 : utilisation d'une classe

Voici le texte d'une classe représentant de façon sommaire un compte bancaire et les opérations bancaires courantes.

```
class Compte{
    private int solde = 0;
    public void deposer(int montant){
        solde = solde + montant;
    }
    public void retirer(int montant){
        solde = solde -montant;
    }
    public void virerVers(int montant, Compte destination){
        this.retirer(montant);
        destination.deposer(montant);
    }
    public String toString(){
        return "solde:_" + solde;
    }
}
```

Question 1

Comment fonctionne la méthode `virerVers` ? Combien de comptes fait-elle intervenir ?

Réponse : la méthode virement fait intervenir deux objets de type `Compte` : `this`, l'objet sur lequel la méthode est appelée et `destination`, le paramètre de la méthode. Le virement s'effectue de `this` vers le paramètre de la méthode. L'argent est retiré d'un compte et déposé sur l'autre.

Question 2

Créez deux comptes que vous affecterez à deux variables. Ecrivez le code correspondant aux opérations suivantes :

- dépôt de 500 euros sur le premier compte.
- dépôt de 1000 euros sur le second compte.
- retrait de 10 euros sur le second compte.
- virement de 75 euros du premier compte vers le second.
- affichage des soldes des deux comptes.

Vous mettrez le code java correspondant à cette question dans la méthode main d'une nouvelle classe appelée TesteCompte. Vous compilerez et testerez ce programme.

```
public class TesteCompte {
    public static void main(String[] args) {
        Compte martin, jean;
        martin = new Compte();
        jean = new Compte();
        // dépôt de 500 euros sur le premier compte.
        martin.deposer(500);
        // dépôt de 1000 euros sur le second compte.
        jean.deposer(1000);
        // retrait de 10 euros sur le second compte.
        jean.retirer(10);
        // virement de 75 euros du premier compte vers le second.
        martin.virerVers(75, jean);
        // affichage des soldes des deux comptes.
        System.out.print("Compte_de_martin,");
        System.out.println(martin);
        System.out.print("Compte_de_jean,");
        System.out.println(jean);
    }
}
```

Question 3

Créez un tableau de dix comptes (type `Compte []`). Pour cela, notez bien qu'il faut d'abord créer le tableau puis créer successivement les dix comptes à mettre dans les dix cases de ce tableau. Dans chaque case, faites un dépôt de 200 euros plus une somme égale à 100 fois l'indice du compte dans le tableau. Ensuite, vous ferez un virement de 20 euros de chaque compte vers chacun des comptes qui le suivent dans le tableau (par exemple, du compte d'indice 5, il faut faire des virements vers les comptes d'indice 6, 7, 8 et 9). Enfin, vous afficherez les soldes de tous les comptes. Ici encore, vous testerez et compilerez le code proposé

```
public class TesteCompte {
    public static void main(String[] args) {
        Compte[] table = new Compte[10];
        for (int i=0; i<table.length; i++){
            table[i] = new Compte();
            table[i].deposer(200 + i*100);
        }
        for (int i=0; i<table.length; i++){
            for (int j=i+1; j<table.length; j++){
                table[i].virerVers(20, table[j]);
            }
        }
        for (int i=0; i<table.length; i++){
            System.out.print("Compte_numero_" + i + ",");
            System.out.println(table[i]);
        }
    }
}
```

Exercice 2 : personnes

Question 1

Écrivez une classe permettant de représenter des personnes. Les informations à représenter sont le prénom et le nom de la personne, son numéro INSEE (numéro de sécu) et son numéro de téléphone.

La classe devra comporter un constructeur pour initialiser les instances ainsi que des méthodes pour accéder aux différentes informations.

Pour ce qui est des modifications de données, on suppose que les noms, prénoms et numéros INSEE ne changent jamais, mais qu'une personne peut changer de numéro de téléphone. Il faut donc prévoir une méthode pour cela.

La classe comportera aussi une méthode `toString` renvoyant une chaîne de caractère.

Question 2

Créez plusieurs objets représentant des personnes et changez leurs numéros de téléphone.

Question 3

Ajoutez à la classe une méthode permettant de tester si deux personnes ont le même nom de famille. Cette méthode ne doit pas faire d'affichage mais elle doit renvoyer un résultat booléen.

Ajoutez au programme une invocation de méthode utilisant cette nouvelle méthode.

```
public class Personne{
    private String nom, prenom, insee, tel;
    public Personne(String n, String p, String i){
        nom=n;
        prenom=p;
        insee=i;
    }
    public String getNom(){
        return nom;
    }
    public String getPrenom(){
        return prenom;
    }
    public String getINSEE(){
        return insee;
    }
    public String getTel(){
        return tel;
    }
    public void setTel(String t){
        tel = t;
    }
    public String toString(){
        return nom + "_" + prenom + "_" + insee + "_tel:" + tel;
    }
    public boolean memeNomDeFamille(Personne autre){
        return nom == autre.getNom();
    }
    public static void main(String[] args){
```

```
Personne hugo, zola, jhugo;
hugo = new Personne("Hugo","Victor","1020225056356");
jhugo = new Personne("Hugo","Joseph","1024788304123");
hugo.setTel("0601010101");
zola = new Personne("Zola","Emile","1044075102952");
zola.setTel("0701010101");
System.out.println(hugo);
System.out.println(zola);
hugo.setTel("0602020202");
System.out.println(hugo);
System.out.println("Hugo_et_Zola_ont-ils_le_même_nom?" +
    hugo.memeNomDeFamille(zola));
System.out.println("Les_Hugo_ont-ils_le_même_nom?" +
    hugo.memeNomDeFamille(jhugo));
}
}
```
