Exercises 14

Java classes

Exercise 1.14 - Using a class

Here is the text of a class that represents a bank account and banking transactions.

```
class Compte{
    int solde = 0;
    void deposer(int montant){
        solde = solde + montant;
    }
    void retirer(int montant){
        solde = solde - montant;
    }
    void virerVers(int montant, Compte destination){
        autre.retirer(montant);
        this.deposer(montant);
    }
    void afficher(){
        Terminal.ecrireString("solde: "+ solde);
    }
}
```

Question 14.1.1

How does the method `virerVers`()? How many accounts does it involve?

Question 14.1.2

Create two accounts that you assign to two variables. Enter the code corresponding to the following:

- **deposit** 500 euros on the first account.
- **deposit** 1000 euros on the second account.
- **withdraw** 10 euros on the second account.
- **transfer** 75 euros the first account to the second.
- **display** balances of the two accounts.

Put the corresponding Java code into the `main`() method of a new class called `TesteCompte`. Test and compile the program.

Question 14.1.3

Create an array of ten accounts. For this, note that we must first create the array and then create successively ten accounts put in ten cells of this array.

In each case, make a deposit of 200 euros plus an amount equal to 100 times the index of the account in the array.

Then you make a deposit of 20 euros at each account to each of the accounts that follow in the array (for example, the account under index 5 must make transfers to accounts under index 6, 7, 8 and 9).

Finally, display the balances of all accounts.

Exercice 14.2 constructeurs

Cet exercice reprend la classe `Compte` de l'exercice précédent.

Question 14.2.1

Complétez la classe `Compte` avec une information supplémentaire : le nom du titulaire du compte (type `String`). Vous modifierez la méthode d'affichage pour qu'elle affiche cette information.

Question 14.2.2

Créez un constructeur pour la classe `Compte`. Ce constructeur doit prendre en paramètre le nom du titulaire du compte.
Donnez le code de création d'un compte qui appelle ce constructeur.

Question 14.2.3

Faut-il prévoir des méthodes permettant de changer le nom du titulaire du compte ?

Exercice 14.3 méthodes statiques ou non

Parmi les méthodes de la classe suivante, lesquelles peuvent être **statiques** et lesquelles ne peuvent en aucun cas être statiques ?

```
class Exo12_3{
    int x, y;
    String nom;
    void afficher(){
        Terminal.ecrireString(nom + " " + x + "  " + y);
    }
    void ajouter(Exo12_3 obj){
        x = x + obj.x;
        y = y + obj.y;
        nom = nom + obj.nom;
    }
    Exo12_3 nouveau(int n){
        Exo12_3 res = new Exo12_3();
        res.x = n;
        res.y = n*2;
        res.nom = "Auto_"+n;
        return res;
    }
    boolean plusGrand(Exo12_3 obj){
        if (obj.x == x){
            return y>obj.y;
        }else{
            return x>obj.x;
        }
    }
    boolean compare(Exo12_3 obj1, Exo12_3 obj2){
        if (obj1.x == obj2.x){
            return obj1.y>obj2.y;
        }else{
            return obj1.x>obj2.x;
        }
    }
}
```

Exercice 14.4 : égalité d'objets

Essayez de prédire le résultat de l'exécution de ce programme. Testez le programme. Que peut-on en déduire sur la notion d'égalité d'objets en java ?
Notez que l'expression `c1.incremente().getValeur()` comprend deux appels de méthodes successifs. D'abord la méthode incremente est appelée sur `c1`. Cette méthode renvoie un objet de type `Compte` qui est celui sur lequel la méthode est appelée (`this`). Sur cet objet, la méthode `getValeur`() est appelée. Cela revient à appeler successivement les deux méthodes sur le même objet.

```
class Exo12_4{
    public static void main(String[] argv){
        Compteur c1, c2, c3;
        c1 = new Compteur(0);
        c1.incremente();
        c2 = new Compteur(1);
        c3 = c1;
        if (c1 == c3){
            Terminal.ecrireStringln("c1 et c3 sont égaux");
        }else{
            Terminal.ecrireStringln("c1 et c3 ne sont pas égaux");
        }
        if (c1.getValeur() == c2.getValeur()){
            Terminal.ecrireStringln("c1 et c2 ont même valeur");
        }else{
            Terminal.ecrireStringln("c1 et c2 n'ont pas la même valeur");
        }
        if (c1 == c2){
            Terminal.ecrireStringln("c1 et c2 sont égaux");
        }else{
            Terminal.ecrireStringln("c1 et c2 ne sont pas égaux");
        }
        if (c1.getValeur() == c1.incremente().getValeur()){
            Terminal.ecrireStringln("c1 et c1 incremente ont même valeur");
        }else{
            Terminal.ecrireStringln("c1 et c1 incremente n'ont pas la même
valeur");
        }
        if (c1 == c1.incremente()){
            Terminal.ecrireStringln("c1 et c1 incremente sont égaux");
        }else{
            Terminal.ecrireStringln("c1 et c1 incremente ne sont pas égaux");
        }
    }
```

Exercice 14.5 conception

Cet exercice a pour but de réfléchir sur la conception d'un programme, sa structuration en classes. Il ne s'agit pas pour le moment de réaliser ce programme, mais juste de concevoir son architecture.
On fait des cocktails avec différents liquides (alcools, sodas, jus de fruits). On a un bar avec des bouteilles qui peuvent être pleines ou à moitié vides. On a des shakers qui ont une contenance donnée. Il y a des recettes de cocktails qui indiquent seulement les proportions. Ces recettes peuvent s'appliquer à des quantités plus ou moins grandes selon les besoins du moment.

Les cocktails se font en déversant une partie du contenu des bouteilles dans des shakers. Après, il faut secouer. Les shakers sont ensuite vidés (dans les verres, mais on ne tiendra pas compte des verres dans cette application). Il faut les laver après usage.

Questions :
Quelles classes faut-il créer ?
Quelles informations faut-il dans chaque classe ?
Quelles méthodes faut-il écrire, et dans quelle classe les mettre ?
Pour chaque méthode, précisez le type des paramètres et de la valeur de retour.