
UFR ST - Besançon- L2 Info - S3 - Année 2015/16

Programmation par Objets

TP 2 - Références, toString(), tableau d'objets

Objectifs du TP : passage par valeur de types primitifs ou de types objets; constructeurs et instantiation; méthode toString(); tableaux de types objets.

Exercice 1. Passage par valeur en Java

Recopiez, *en le comprenant*, le programme java de la figure 1. Exécutez-le et observez le résultat.

```
public class ParValeur {
    public static void swap(int a, int b) {
        int tmp = a;
        a = b;
        b = tmp;
    }

    public static void main(String[] args) {
        int a = 5, b = 8;
        Ecran.afficherln("Avant echange : a = "+a+" ; b = "+b);
        swap(a, b);
        Ecran.afficherln("Après echange : a = "+a+" ; b = "+b);
    }
}
```

FIG. 1 – Un programme java à exécuter

Question 1. Pourquoi la permutation de a et de b n'a pas eu lieu ?

Question 2. Rédigez une classe nommée `MonEntier` avec un attribut entier nommé *val*, et un constructeur à 1 paramètre entier pour initialiser la valeur de *val*.

Question 3. Réécrivez le programme de la figure 1 en utilisant le type `MonEntier` pour que cette fois-ci la permutation ait lieu. **N.B.** Il est nécessaire de modifier la fonction *swap*, mais aussi la fonction *main*.

Exercice 2. Coder une classe `Personne`

On considère une classe décrivant une personne, conformément au modèle suivant :

Personne
chaîne nom chaîne prenom entier poids entier taille
Constructeur(chaîne nom, chaîne prenom, entier poids, entier taille) Constructeur(chaîne nom, chaîne prenom)
réel calculeIMC()

On peut instancier une personne soit en fournissant son nom, prénom, taille et poids, soit en fournissant simplement son nom et son prénom : dans ce cas la taille et le poids sont nuls.

La méthode `calculeIMC()` calcule l'indice masse corporelle (IMC), en fonction du poids (en *Kg*) et de la taille (en *m*), selon la formule :

$$IMC = \frac{poids}{taille^2}$$

Attention, prévoir de renvoyer un IMC de 0 si la taille est nulle.

Question 1. Codez cette classe en Java.

Question 2. Compilez cette classe. Remarque : comme la classe `Personne` ne possède pas de méthode `main()`, vous ne pouvez pas *exécuter* la classe pour l'instant.

Exercice 3. Instancier et utiliser un objet de la classe `Personne`

Puisque la classe `Personne` ne contient pas de méthode `main()`, elle ne constitue pas un programme Java exécutable, mais elle est utile uniquement à décrire des objets du type `Personne`.

Vous allez maintenant écrire parallèlement à la classe `Personne` une seconde classe, qui elle contiendra la méthode `main()`, et constituera donc le programme à proprement parler. Ce programme a pour but d'instancier (i.e. créer) un objet du type `Personne` et de le manipuler.

Rappel. En Java, on utilise l'opérateur `new` pour instancier un objet après l'avoir déclaré. L'instruction suivante permet de déclarer *et* d'instancier un objet nommé `p1` de la classe `Personne` :

```
Personne p1 = new Personne();
```

N.B. En répondant aux questions, n'oubliez de compiler régulièrement votre code pour le corriger au fur et à mesure.

Question 1. Écrivez une classe nommée `GoPersonne` avec uniquement la méthode `main()`.

Question 2. Déclarez et instanciez un objet `p1` de type `Personne` dans le `main()` par l'instruction :

```
Personne p1 = new Personne();
```

Pourquoi est-ce que la compilation de la classe `GoPersonne` échoue ?

Question 3. Pour corriger, demandez d'abord à l'utilisateur de saisir un nom et un prénom, avant d'instancier `p1` avec ce nom et ce prénom en paramètres.

Question 4. Complétez votre programme en demandant ensuite à l'utilisateur un nom, un prénom, un poids et une taille et instanciez un objet `Personne p2` avec ces données.

Question 5. Affichez la valeur de tous les attributs de `p1`, suivis de l'IMC de `p1`.

Question 6. Même chose pour `p2`.

Exercice 4. La méthode `toString()`

Il est pénible d'avoir à afficher un à un tous les attributs d'un objet pour connaître son état. Il serait plus commode de pouvoir écrire directement

```
Ecran.afficher(p1);
```

pour pouvoir afficher en une seule instruction l'état de l'objet `p1`.

Cela est possible grâce à une méthode particulière en Java nommée `toString()`. Son en-tête est obligatoirement

```
public String toString()
```

et elle doit être ajoutée à la classe dont on souhaite visualiser l'état (ici la classe `Personne`).

Le code de la méthode `toString()` doit construire une chaîne de caractères représentant l'état de l'objet, puis la renvoyer.

Par exemple, la méthode `toString()` qui permet de représenter une personne de prénom `Dave` et de nom `Sitek` sous la forme "`[Sitek, Dave]`" est la suivante :

```
public String toString() {  
    return "[" + nom + ", " + prenom + "];  
}
```

ATTENTION. La méthode ne réalise pas elle-même l'affichage par `Ecran.afficher()`. Elle se contente de *construire* la chaîne de caractère qui pourra être affichée plus tard par un `Ecran.afficher()` (par exemple dans le programme principal). Autrement dit, la méthode `toString()` permet de définir le format d'affichage de tout objet de la classe.

N.B. Dans un contexte d’affichage, il n’est pas nécessaire d’invoquer explicitement la méthode `toString()` : Java le fait pour vous. Par exemple, avec une méthode `toString()` dans `Personne`, l’instruction `Ecran.afficher(p1);` est équivalente à `Ecran.afficher(p1.toString());`.

Question 1. Dotez la classe `Personne` d’une méthode `toString()` permettant de représenter une personne par une chaîne de la forme "`nom, prenom (ppp Kg, ttt cm) : IMC = xx.xxx`", où `nom` est remplacé par le nom de la personne, `prenom` par son prénom, `ppp Kg` par le poids de la personne en Kg, `ttt cm` par sa taille en cm, et où `xx.xxx` est l’IMC calculé.

Par exemple, la chaîne représentant une personne de prénom `Dave`, de nom `Sitek`, pesant 80 Kg et mesurant 1,85 m, sera

"`Sitek, Dave (80 Kg, 185 cm) : IMC = 23.375`".

Question 2. Testez dans le `main()` l’effet de `toString()` en demandant l’affichage de `p1` et `p2` par
 `Ecran.afficher(p1);`
 `Ecran.afficher(p2);`

Exercice 5. Un tableau de personnes

Rédigez une classe permettant de gérer un ensemble de personnes (comme par exemple les étudiants d’un groupe de TP). Les informations à gérer sont l’effectif du groupe, et les étudiants qui sont regroupés au sein d’un tableau de `Personne`. La taille maximum du groupe doit être fournie en paramètre à la construction.

Question 1. Codez cette classe en Java.

Question 2. Dotez cette classe d’une méthode `toString()` qui permette de construire et retourner une chaîne présentant l’ensemble des étudiants du groupe.

Question 3. Ajoutez le code nécessaire dans votre programme principal pour pouvoir saisir, créer et enregistrer un groupe d’au moins 5 personnes, puis qui l’affiche *via* une invocation de la méthode `toString()`.