

Object référence, égalité, String, tableau

Exercice 1 - Assignment, égalité, référence

- 1 Qu'affiche le code suivant :

```
String s1="toto";
String s2=s1;
String s3=new String(s1);

System.out.println(s1==s2);
System.out.println(s1==s3);
```

Expliquer

- 2 Quelle est la méthode à utiliser si l'on veut tester le contenu des chaînes de caractère ?
3 Qu'affiche le code suivant :

```
private static boolean test(String u,String v) {
    return u==v;
}
public static void main(String[] args) {
    String s1="toto";
    String s2=s1;
    System.out.println(test(s1,s2));
}
```

Expliquer

- 4 Qu'affiche le code suivant :

```
String s1="toto";
String s2="toto";

System.out.println(s1==s2);
```

Expliquer.

- 5 Qu'affiche le code suivant :

```
String s="hello";
s.toUpperCase();
System.out.println(s);
```

Expliquer

Exercice 2 - Conversion de String en entier

On souhaite écrire un programme affichant la somme d'entiers pris en paramètre sur la ligne de commande.

Voici un exemple d'exécution :

```
$ java sum 15 5 231
entiers: 15 5 231
sum: 251
```

Ce programme est décomposé en plusieurs fonctions :

- 1 Écrire une méthode qui prend un tableau de chaînes de caractères en argument et

renvoie un tableau d'entiers de même taille contenant les entiers issus des chaînes de caractères.

La méthode statique `parseInt(String s)` de la classe `java.lang.Integer` permet de récupérer la valeur d'un entier stockée dans une chaîne de caractères.

- 2 Que se passe-t-il lorsqu'un mot pris en argument n'est pas un nombre ?
- 3 Écrire une méthode qui prend un tableau d'entiers en argument et renvoie la somme de ceux-ci.
- 4 Écrire la méthode `main` qui utilise les deux méthodes précédentes pour afficher le tableau d'entiers ainsi que sa somme.
Il y a un petit piège pour afficher les tableaux vous pouvez vous aider de la classe `java.util.Arrays`.

Exercice 3 - Comment créer un livre ?

Le but de cet exercice est de définir des objets `Book`. Nous allons pour cela créer la classe `Book`.

- son titre (une chaîne de caractères) ;
- son auteur ;
- son numéro ISBN (une chaîne de caractères) ;
- son prix en euros (un nombre réel).

Écrire une classe `Book` dont les attributs sont tout ce qui caractérise un livre. Dans la classe `Book`, mettre un constructeur `public Book(String title, String author, String isbn, double price)` servant à initialiser les attributs des instances de la classe `Book` (les arguments du constructeur sont respectivement le titre, les auteurs, l'ISBN et le prix).

Comment faire pour construire un objet livre (une instance de la classe `Book`) ?

Exercice 4 - Comment afficher un livre ?

On aimerait pouvoir afficher les caractéristiques d'un livre, par le code Java suivant :

```
Book book=...
System.out.println(book);
```

Java sait faire cela, à condition de mettre dans la classe `Book` une méthode `public String toString()` retournant une chaîne de caractères, qu'on construit typiquement à partir des attributs de l'objet.

Écrire cette méthode, pour obtenir par exemple l'affichage suivant :

```
Title : Concepts fondamentaux de l'informatique
Author : Alfred Aho
ISBN : 2-7117-8689-7
Price : 40.0
```

Ajouter l'annotation `@Override` sur la méthode `toString`. Que signifie cette annotation ?

Exercice 5 - Tri à bulle [à la maison]

- 1 Écrire une méthode `swap` qui échange les valeurs de deux cases d'un tableau. `void`

```
swap(int[] array,int index1,int index2)
```

- 2 Écrire une méthode `indexOfMin` qui renvoie l'index de la valeur minimale d'un tableau.
- 3 Modifier la méthode `indexOfMin` en ajoutant deux index indiquant que l'on cherche l'index du minimum non pas sur tout le tableau mais sur la partie de tableau entre les deux index (le premier inclus, le deuxième exclus).
- 4 Écrire la méthode `sort` qui prend un tableau d'entier en paramètre et qui trie celui-ci en utilisant pour cela les méthodes `indexOfMin` et `swap`.