

Premier pas en Java, référence, objet et classe

Exercice 1 - Hello Groland

On rappelle qu'il est préférable en Java que chaque classe soit définie dans un fichier qui lui est propre. Le nom du fichier doit être le nom de la classe qu'il contient, auquel on ajoute le suffixe `.java`. Les noms des classes doivent être constitués de mots accolés dont la première lettre est une majuscule.

Dans un premier temps nous allons écrire des petits programmes permettant de se familiariser avec le compilateur, la machine virtuelle et les méthodes.

Écrire un programme qui affiche `Hello Groland`.

Exercice 2 - En morse. Stop.

Écrire une classe `Morse` qui permet lors de son exécution d'afficher les chaînes de caractères présent en argument séparé par des `"Stop."`.

```
$ java Morse ceci est drôle
ceci Stop. est Stop. drôle Stop.
```

Les arguments de la lignes de commandes sont stockées dans le tableau de chaînes de caractères passé en argument à la méthode `public static main(String[] args)`.

Utiliser dans un premier temps, l'opérateur `+` permet la concaténation de deux chaînes de caractères.

Utiliser dans un second temps la classe `StringBuffer`. Quelle est l'avantage par rapport à la solution précédente ?

Exercice 3 - Conversion de String en entier

Écrire un programme qui met les nombres entiers qu'on lui passe en argument sur sa ligne de commandes dans un tableau d'entiers et qui affiche ce tableau suivi de la somme de ses éléments. Voici un exemple d'exécution :

```
$ java Sum 15 5 231
15 5 231
Somme: 251
```

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

Note 1: Il est possible de faire plusieurs méthodes en Java.

Note 2: Que se passe-t'il lorsqu'une lettre est prise en argument au lieu d'un nombre ?

Exercice 4 - 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) ;
- ses auteurs (un tableau de chaînes de caractères contenant le nom de l'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[] authors, 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 5 - 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
  Authors : Alfred Aho, Jeffrey Ullman
  ISBN : 2-7117-8689-7
  Price : 40.0
```

Exercice 6 - Une bibliothèque

Une bibliothèque contient des livres. Nous représenterons les bibliothèques par des objets instances de la classe `Library` qu'il nous faudra écrire. On trouvera en attribut de la classe un tableau de références sur des instances de `Book`.

On définira dans la classe `Library` :

- 1 un constructeur, qui prend en paramètre le nombre maximum de livres que peut contenir la bibliothèque (sa capacité) ;
- 2 une méthode `int capacity()` retournant la capacité de la bibliothèque ;
- 3 une méthode `void add(Book b)` qui ajoute un livre dans la bibliothèque ;
- 4 une méthode `int size()` retournant le nombre de livres contenus dans la bibliothèque ;
- 5 une méthode `public String toString()`, qui affiche l'ensemble des livres contenus dans la bibliothèque.

Exercice 7 - Bibliothèque (2) le retour

On aimerait attribuer à chaque livre un entier `id` qui référence de façon unique un livre dans la bibliothèque.

Ceci permet d'ajouter à la classe `Library` la méthode `Book find(int id)` qui recherche dans la bibliothèque une instance de `Book` ayant l'id fourni en argument, et retourne une référence sur ce livre s'il est dans la bibliothèque, la référence `null` sinon.

Afin que la recherche soit effectuée le plus rapidement possible, l'attribut tableau de livres

contenu dans chaque instance de `Library` sera maintenu trié par ordre croissant des `id` des livres.