

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 - De C vers Java

Cet exemple a pour but de montrer les différence de performance en un programme en C et le même en Java.

```
#include <stdio.h>
#include <stdlib.h>

int pascal (int nBut, int pBut){
    int * tab;
    unsigned int n, i;

    tab = (int *)malloc ((nBut+1)*sizeof(int));
    if(tab==NULL){
        fprintf(stderr, "Pas assez de place\n");
        exit(0);
    }

    tab[0] = 1;

    for(n=1; n<=nBut; n++){
        tab[n] = 1;

        for(i=n-1; i>0; i--){
            tab[i] = tab[i-1] + tab[i];
        }
    }

    return tab[pBut];
}

int main(int argc, char * argv[]) {
    printf(" Cn, p = %d\n", pascal (30000, 250));
    return 0;
}
```

- 1 Compiler (gcc pascal.c) et exécuter le programme a.out en demandant au système le temps d'exécution du programme. (time a.out).
- 2 Ecrire le programme (Pascal.java) équivalent en Java. Pour une fois, servez vous du copier/coller. Compiler le programme. Exécuter en mesurant le temps (toujours avec time).

Comment peut-on expliquer la différence de vitesse ?

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

Exercice 5 - 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 6 - 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 7 - 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.