

## **PARTIEL 2**

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant (et celui-ci seulement) et les notes manuscrites.

### Exercice 1.- (Serveur web rudimentaire)

Écrire un serveur web en Java qui ne reconnaît que la commande GET. Lorsqu'un client se connecte à votre serveur et envoie une commande du type :

```
GET filename HTTP/1.0
```

le serveur renvoie l'en-tête :

```
HTTP/1.1 200 OK
```

suivi d'une ligne blanche et de toutes les lignes du fichier.

Si la syntaxe de la commande n'est pas correcte ou si le fichier n'existe pas, il renvoie une ligne blanche suivie de :

```
404 Not Found.
```

[ Le serveur doit évidemment pouvoir accepter plusieurs clients à la fois. Son numéro de port est 80.

Pour tester ce serveur chez vous, après avoir lancé le serveur, utiliser un navigateur et indiquer en entrée, par exemple :

```
http://127.0.0.1/applis/wollic/index.html
```

si `/applis/wollic/index.html` est un nom de fichier valide sous Linux. Vous verrez apparaître votre page web de façon habituelle. Dans la fenêtre serveur, le source sera affiché. Dans le cas d'un fichier texte (en fait non HTML), vous verrez également le texte affiché dans la fenêtre du navigateur. ]

### Exercice 2.- (Tri par insertion)

- 1<sup>o</sup>) Écrire une application Java qui demande un entier naturel, qui génère un tableau d'entiers naturels (compris entre 0 et 99 999), qui affiche les cent premiers éléments de ce tableau, qui trie ce tableau par tri insertion puis qui affiche les cent premiers éléments du tableau trié.

[ La génération du tableau et le tri seront effectués par des méthodes.

Rappelons que le tri insertion consiste à déterminer le plus grand élément du tableau, à le placer à la fin, puis le plus grand élément du tableau moins le dernier et à le placer comme avant-dernier et ainsi de suite. ]

- 2<sup>o</sup>) Quel est le coût du tri insertion, l'opération élémentaire étant la comparaison et la taille le nombre d'éléments du tableau ?

- 3<sup>o</sup>) Que vaut-il mieux utiliser entre tri insertion, tri à bulle et tri fusion ? Justifier votre réponse en précisant éventuellement les circonstances d'utilisation.

# DOCUMENTATION

## 1 Nombre au hasard

La méthode statique :

```
double random()
```

de la classe `Math` permet d'obtenir un nombre réel, obtenu de façon aléatoire, appartenant à l'intervalle  $[0, 1[$ .

## 2 Tokenisation

Lorsqu'on veut analyser une commande obtenue sous forme d'une phrase, c'est-à-dire d'une chaîne de caractères, chaque mot représente en général un opérateur ou un opérande de cette commande. On veut donc décomposer cette phrase en mots, ou plus exactement en **tokens** (terme introduit par Peirce à la fin du XIX<sup>e</sup> siècle).

La classe `StringTokenizer` du paquetage `java.util` nous aide pour cela. Elle comprend un constructeur dont le paramètre est un objet de la classe `String`. La méthode `nextToken()` de cette classe donne le token suivant, qui est un `String`.

## 3 Ouverture de fichier en lecture

Rappelons que le nom logique d'un fichier en lecture est un objet de la classe `FileInputStream`. Lorsqu'on essaie d'instantier un tel fichier, s'il n'existe pas alors une exception du type :

```
FileNotFoundException
```

est levée.