

# Travaux Dirigés de Java (2<sup>ème</sup> Partie)

Joseph NDONG

## Exercice 1 : Thread : utilisation de la classe ReentrantLock

L'interface **Lock** est nouvellement introduite en java **1.5** pour gérer la synchronisation des **threads**. Etudier la et reprendre les exemples de Thread synchronisés donnés dans le module Threads du cours de Java et utiliser convenablement cette interface pour gérer les processus. On peut utiliser pour cela la classe **ReentrantLock**.

## Exercice 2a: String

Ecrire une classe contenant une méthode de classe prenant en argument une chaîne **ch** et renvoyant sous forme d'un tableau d'entiers les indices (positions) des éventuelles voyelles qui sont à des index impairs dans la chaîne **ch**.

## Exercice 2b: String

Ecrire une classe contenant une méthode d'instance prenant en argument un tableau de chaînes **tabCh** et renvoyant l'index de l'élément qui a la plus grande taille. S'il existe plusieurs éléments qui correspondent à ce qui est cherché, on renvoie le premier élément trouvé.

## Exercice 3b: String

Ecrire une méthode de classe qui prend une chaîne en argument et renvoie le nombre de caractères **distincts** (qui ne sont pas répétés dans la chaîne) que celle-ci contient.

Ecrire une méthode de classe qui prend une chaîne en argument et renvoie le nombre de caractères **présents au moins deux (2) fois dans celle-ci**.

## Exercice 3: Entrées-Sorties

Ecrire une classe contenant une méthode de classe prenant en argument un fichier texte et qui affiche sur la console le nombre d'occurrences de **chaque mot** trouvé dans le fichier.

## Exercice 4: Entrées-Sorties | Swing

Ecrire un programme qui permet de sélectionner un fichier audio **MP3** depuis le poste de travail et qui permet de l'exécuter (lire) dans une fenêtre graphique Swing. L'utilisateur doit cliquer sur un bouton pour ouvrir le poste de travail, ensuite sélectionner le fichier qui sera lu dans une fenêtre.

Créer une méthode similaire pouvant exécuter des fichiers video **MP4**.