

## Codage de Huffman

### Récupération des fichiers

Les fichiers utilisés dans ce TP sont disponibles dans le répertoire `Info Mias-SM\Java\TP10` sur `jupiter`. Ce répertoire contient les fichiers suivants :

Nom du fichier	Contenu du fichier
<code>InterfaceHuffman.java</code>	la classe définissant l'interface graphique du codage
<code>CodeHuffman.java</code>	la classe effectuant le codage et le décodage
<code>ArbreBinaire.java</code>	la super-classe de l'arbre de codage

Ce répertoire contient de plus un répertoire `fichiers`, dans lequel sont regroupés sous la forme d'un paquet les fichiers correspondant aux classes et interfaces utilisées pour lire et écrire des fichiers :

Nom du fichier	Contenu du fichier
<code>ListeEnum.java</code>	l'interface française pour les énumérations
<code>FichierES.java</code>	la partie commune aux fichiers d'entrée/sortie
<code>LecteurCaracteres.java</code>	la classe permettant la lecture d'un fichier de caractères
<code>FichierBits.java</code>	la partie commune aux fichiers de bits
<code>LecteurBits.java</code>	la classe permettant la lecture d'un fichier de bits
<code>EnregistreurBits.java</code>	la classe permettant l'écriture d'un fichier de bits

Ce répertoire contient également un répertoire `doc`, contenant des pages HTML décrivant la documentation de chacune des classes.

Avant de commencer à travailler, il vous faut copier les fichiers `.java` dans un répertoire qui vous est propre. Pour cela, effectuez les opérations suivantes :

1. créez un nouveau répertoire `TP10` dans le répertoire `Java` de votre répertoire personnel (`Z:`) ;
2. copiez les trois fichiers `.java` et le répertoire `fichiers` de `jupiter` dans votre répertoire `TP10` (utilisez les raccourcis `Ctrl-C` et `Ctrl-V`, ou cliquez sur le bouton droit et sélectionnez les libellés `copier` et `coller`) ;

La documentation n'a pas besoin d'être copiée. Pour y accéder plus facilement, vous pouvez créer dans votre répertoire `TP10` un *lien* (appelé *raccourci* sous Windows) vers le répertoire `TP10\doc` de `jupiter`. Ce lien peut être créé par le menu contextuel (obtenu d'un clic sur le bouton droit de la souris) de votre répertoire `TP10`.

Cela étant fait, vous pouvez lancer `Emacs`.

### Objectifs de l'application

L'application visée est un « codeur » de Huffman qui effectuerait les opérations de calcul du code optimal (cette partie est entièrement fournie), le codage du texte initial dans un fichier binaire et le décodage du fichier binaire.

Une interface est proposée. Son fonctionnement est le suivant :

- le texte doit être contenu dans un fichier, dont le nom apparaît dans le champ `Fichier à coder`,
- lorsqu'on définit ce fichier, le codage correspondant est calculé ,
- lorsqu'un codage a été calculé, on peut obtenir des informations sur chaque caractère (sa fréquence et son code), en utilisant le premier bouton,
- pour coder un texte, il faut indiquer le nom du fichier qui contiendra le texte codé dans le champ `Fichier codé` (attention, si le fichier existe, son contenu sera effacé par cette opération),
- pour décoder un texte, il faut indiquer le nom du fichier codé et activer le bouton ad-hoc, (attention, le décodage se fait avec le dernier code calculé).

### **Les opérations à implanter**

Votre travail est de définir les méthodes de codage et de décodage d'un texte. Ces deux méthodes sont définies dans les classes `InterfaceHuffman` et `CodeHuffman`, les méthodes de la première appelant celles de la seconde. Les profils ainsi que quelques opérations techniques sont déjà écrites dans la classe `CodeHuffman`, vous n'avez qu'à compléter le cœur de ces fonctions dans cette classe.

### **Ce qu'il faudrait encore faire**

L'application actuelle est incomplète. Pour une application réelle, il faut aussi conserver le code de chaque caractère pour pouvoir décoder le fichier binaire. Comment peut-on imaginer de conserver cet arbre? Quelle serait la taille de cet arbre pour un texte utilisant 40 caractères?