

## Cours Java Avancé – DIC2

**NB** : En plus du rendu, la qualité du code sera prise en compte dans la correction de ce travail. De même, ce projet devra être fait par groupe d'au plus trois étudiants et devra être envoyé au plus tard le Lundi 12 Février 2018 à 23h59mn59s à l'adresse [envoip@gmail.com](mailto:envoip@gmail.com). L'objet du mail devra être `MP1_JAVA_numeroGroupe_DIC2`. Le non-respect de ces contraintes rendra le travail irrecevable.

### **Mini-projet 2** : Compresseur de fichiers

L'objectif de ce MP est de mettre sur place un logiciel d'archivage et de compression de fichiers. L'utilisation des bibliothèques de manipulation de fichiers archivés (zip, etc.) est interdite.

Pour rappel, ***l'archivage*** est l'utilisation d'un seul fichier pour stocker plusieurs fichiers et la ***compression*** de données est la diminution de l'espace occupé sur le support numérique, sans perte de qualité. On peut donc le comparer à la combinaison de ***tar*** (archivage) et ***gzip*** (compression) dans le cadre d'une archive compressée ***.tgz***.<sup>1</sup>

Le travail peut donc être découpé en deux parties :

- Archivage des données : Ce module de votre programme devra permettre de regrouper plusieurs fichiers en un seul et de pouvoir aussi, à partir du fichier regroupant le tout, récupérer les fichiers d'origine.
- Compression des données : Ce module aura pour objectif de compresser les données afin de réduire la taille du fichier d'archive. Il s'agira ici de faire de sorte que la taille du fichier regroupant le tout soit le plus petit possible. A ce niveau, vous allez utiliser un algorithme de compression sans perte bien déterminé. L'un des plus célèbres est DEFLATE (qui est d'ailleurs utilisé par le format Zip) mais rien ne vous oblige à l'utiliser, il en existe d'autres après tout.

La classe principale d'utilisation de votre programme portera le nom ***SenFileCompressor*** et devra donner à l'utilisateur les options d'utilisation suivantes :

- `java SenFileCompressor -h` devra fournir en sortie l'aide du programme qui donne les instructions d'utilisation du programme. Cette aide devrait aussi être affichée à chaque fois que les autres options fournies ne sont pas correctes.

---

<sup>1</sup> Source : [https://fr.wikipedia.org/wiki/ZIP\\_\(format\\_de\\_fichier\)](https://fr.wikipedia.org/wiki/ZIP_(format_de_fichier))

- `java SenFileCompressor -c <liste fichier à compresser>` <sup>2</sup> devra fournir en sortie un fichier d'extension « .sfc » qui regroupe, sous forme compressé, les différents fichiers fournis en paramètre.
- `java SenFileCompressor -d fichierADecompresser.sfc` devra fournir en sortie l'intégralité des fichiers contenus dans l'archive donné en paramètre

**NB :** Les options « -c », « -d » et « -h » sont mutuellement exclusives. Les deux premières peuvent être combinées avec des paramètres optionnels tels que :

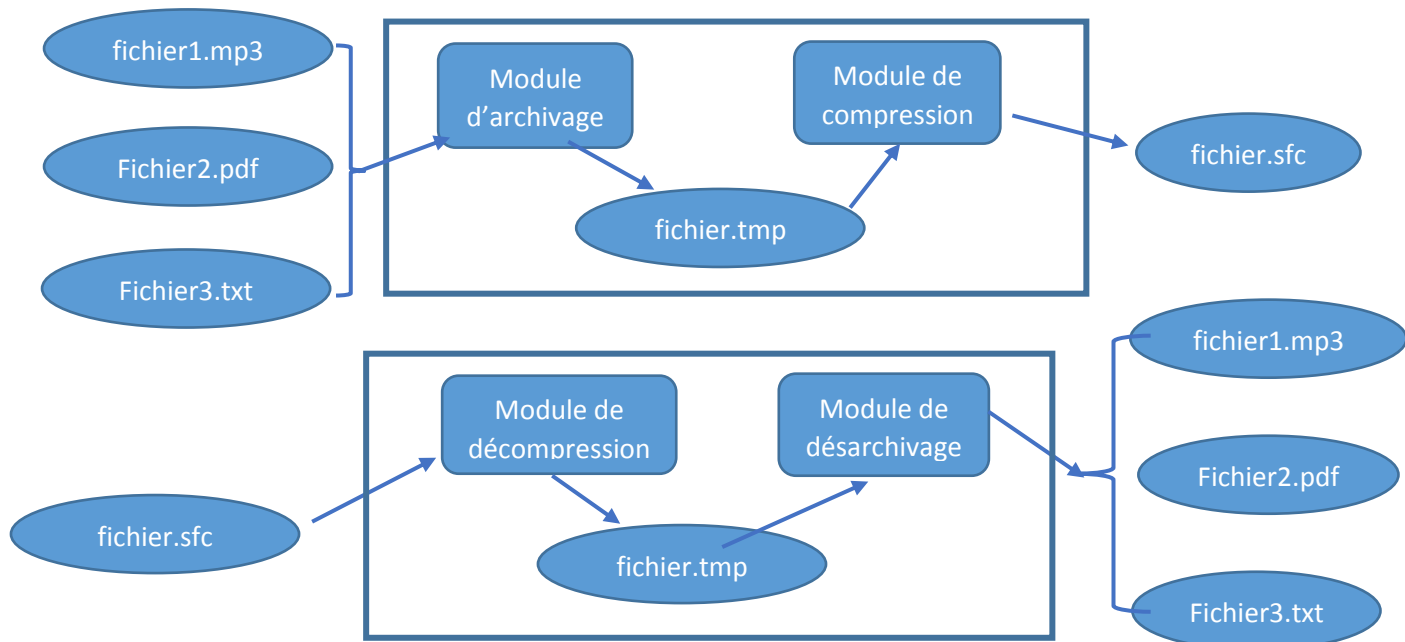
- « -r » qui permet de spécifier le chemin (absolu ou relatif) vers le répertoire où seront stockés les fichiers (ou le fichier) produits par le programme. Un message d'erreur est affiché si le chemin n'existe pas, à moins qu'il ait spécifié l'option « -f », auquel cas, le chemin sera automatiquement créé. En ce sens, l'option « -f » ne doit donc être donnée que si l'option « -r » est précisée.
- « -v » concerne la verbosité du programme. Le programme donnera des détails sur son exécution en fonction de la présence ou non de ce paramètre.

Java SenFileCompressor -c fic1 fic2 fic3... [-r cheminVersRepertoire [-f] ] [-v]

| -d fichier.sfc [-r cheminVersRepertoire [-f] ] [-v]

| -h

Un schéma d'illustration du travail demandé :



<sup>2</sup> Les noms des fichiers seront séparés par des espaces