# Java Avancé L3 Miage 2018-2019

# Session 2 (Aout)

# Examen (durée 2h)

**Consignes :**

Supports de cours et utilisation d'internet autorisés.
**Communication physique, par mail, messagerie instantanée formellement interdite**.
Dans la mesure du possible, déclarer des attributs private et utiliser des getter et des setter.
Les chemins vers les fichiers et les dossiers doivent être **relatifs**.
Tous les exercices sont à réaliser dans des packages différents.

## Partie 1 : Stream et collections

### Exercice 1

1. Télécharger sur Arche le fichier metal_bands_2017.csv et le mettre dans un dossier data au même niveau que le dossier src de votre code. Ouvrir le fichier avec le tableur de votre choix pour prendre connaissance des données.
2. Créer une classe *CsvReader* possédant deux attributs de type String *inputFolderPath* et *outputFolderPath*.
3. Créer un constructeur prenant en paramètre l'url **relative** du fichier (pas de C://).
4. Créer une méthode *existFile* qui retourne un booléen permettant de vérifier si le fichier que vous aurez téléchargé et mis dans le dossier correspondant à *inputFolderPath* existe bien.
5. Créer une classe MetalBand ayant pour attribut :
    a. Un identifiant
    b. Un nom
    c. Un nombre de fans
    d. Une date de formation
    e. Un pays d'origine
    f. Une date de séparation (-1 si le groupe ne s'est pas séparé)
    g. Les genres (**une liste** de style, pas des boolean pour chaque genre)
6. La classe MetalBand possède un constructeur qui initialise tous les attributs.
7. Créer une méthode permettant de lire le fichier ligne par ligne et de découper chaque ligne selon la chaine de caractère passé en argument de cette méthode. Le caractère de séparation de chaque colonne est le ''**;**''.
8. Parser le fichier afin de créer une collection contenant l'ensemble des groupes du fichier. Cette collection sera un attribut de la classe CsvReader et sera une Hashmap ayant pour clé l'id d'un groupe et pour valeur le groupe correspondant.
9. Dans la classe CsvReader, créer les méthodes permettant de :
    a. Déterminer le nombre de pays d'origine différents (en utilisant un set).
    b. Calculer le minimum, le maximum et la moyenne du nombre de fans.
    c. Déterminer pour chacune des années présentes dans le fichier, le nombre de groupes s'étant formés (En utilisant une Hashmap).

d. Même question sur les genres de musique sauf qu'au lieu de stocker pour chaque genre le nombre de groupes, stocker la liste des noms de ces groupes.
e. Trier et afficher les groupes selon l'ordre alphabétique des noms (cette méthode prendra un paramètre indiquant si l'on souhaite trier par ordre croissant ou décroissant). Les données devront être stockées dans un premier temps dans une ArrayList.

10. Créer une méthode permettant de sauvegarder la collection des groupes. Cette méthode prend un argument (String) indiquant le type de sauvegarde à effectuer et que vous devez implémenter :
    a. Sérialisation.
    b. Écriture dans un fichier csv.
11. Ecrire également les méthodes inverses, c'est-à-dire permettant de récupérer les données sauvegardées et ce, pour les deux types de sauvegarde. La méthode permettant de lire les données sous format CSV correspond à votre Parser, vous n'avez donc pas à la réécrire.
12. Quelles sont les avantages et les inconvénients de ces deux méthodes de sauvegarde ?
13. Quels sont les rôles des méthodes Hashcode et Equals ? Comment fait-on pour les redéfinir de manière optimale ? Quels sont leurs rôles respectifs dans le cas des HashMap ?

# Partie 2 : Interfaces graphiques

## Exercice 2

1. Créer une Jframe contenant un Jpanel. La Jframe possède une taille que vous choisirez et le Jpanel occupe toute la taille de la Jframe.
2. Par défaut, le Jpanel possède une couleur de fond orange.
3. Sur ce Jpanel se trouve une zone de texte de couleur blanche avec le texte ''pas cliqué''.
4. Créer un listenner permettant d'afficher les coordonnées d'un clic (gauche) souris dans la console lorsqu'un clic est effectué dans le jPanel. Pour la suite, vous laisserez ces coordonnées s'afficher dans la console.
5. Lors d'un clic, si le texte était ''pas cliqué'' ce dernier se transformer en ''cliqué'' et réciproquement.
6. En fonction des coordonnées de clics et de l'état du clic, la couleur de fond du Jpanel doit changer. Pour ce faire, les coordonnées du pointeur seront passées à une méthode qui différenciera 4 cas :
    - En haut à gauche : vert lorsque la souris est pressée, bleu lorsqu'elle est **relâchée**
    - En haut à droite : jaune lorsque la souris est pressée, rouge lorsqu'elle est **relâchée**
    - En bas à gauche : blanc lorsque la souris est pressée, noir lorsqu'elle est **relâchée**
    - En bas à droite : violet lorsque la souris est pressée, rose lorsqu'elle est **relâchée**

Souris pressée = clic gauche ou clic droit

Les cas gauche, droite, haut et bas sont obtenus relativement par rapport à la taille du jpanel, vous ne devez pas créer plusieurs Jpanel. Par exemple, si le Jpanel fait 200px par 200px, tous les clics où x<100 et y<100 donneront du vert ou du bleu… N'utilisez pas des valeurs absolues mais relatives.

7. Ajouter une interface de connexion permettant d'arriver sur ce panel. Cette interface contient deux champs : un pour le pseudo et un pour le mot de passe (qui ne doit pas être visible lorsque l'utilisateur le tape) et un bouton valider. Le pseudo et les mots de passe associés sont stockés

dans une Hashmap<pseudo, password>. En cas d'authentification réussie, l'utilisateur arrive sur le panel permettant de changer de couleur, sinon, une fenêtre d'erreur apparait lui indiquant soit que son pseudo n'est pas connu, soit que son mdp est erroné. Si son pseudo n'est pas connu, un bouton ''s'inscrire'' apparait et doit permettre l'inscription. L'inscription va alors modifier la HashMap et ajouter l'identifiant et le mot de passe précédemment renseignés.