# Java S2 – EI1

Instructions générales : attentions au normes (indentation, casse, fermeture des fichiers…) ; rendre le TP au plus tard 5 min après la fin du créneau par mail à [louis.moreau.lm@gmail.com](mailto:louis.moreau.lm@gmail.com) (1 point par minute de retard) ; les fichiers à rendre sont tous les .java et les .txt ou .bin créés.

1. Créer trois classes **Animal**, **Dauphin**, **Lion**. Animal est abstraite, et Dauphin et Lion héritent de Animal.
2. Répartir les méthodes et attributs suivants de manière logique parmi les trois classes :
   1. void deplacement()
   2. void sauverUnNageur()
   3. int tailleNageoire
   4. String typeDeBouche
   5. String regimeAlimentaire
   6. void manger()
   7. String habitat
   8. int epaisseurCrinière
   9. void attaquerUnVillageois()
3. Attention à :
   1. bien choisir entre public, private et protected ;
   2. bien utiliser l’héritage
   3. Les méthodes doivent effectuer un syso qui utilise les attributs liés (par exemple deplacement() utilise au moins habitat)
4. Ajouter les **constructeurs** en fonction des classes précédentes.
5. Créer en dur (pas de JOptionPane) une ArrayList contenant plusieurs Dauphins et Lions.
6. Créer une méthode saveTexte(ArrayList a, File f) qui sauvegarde la liste dans un fichier texte.
7. Créer une méthode convertToBin(File texte, File binaire) qui lit un fichier texte, et range son contenu **en tant qu’objets** dans un fichier binaire (possible de faire lectureTexte(File f) dans une ArrayList, puis ecritureBinaire(ArrayList a, File f) pour se simplifier)
8. Créer une méthode lectureBinaire(File f) qui lit le fichier binaire précédemment créé.
9. Dans le main final, on a donc l’enchainement suivant :
   1. création de liste
   2. sauvegarde de la liste dans un texte
   3. lecture du texte et sauvegarde dans un binaire **en tant qu’objets**
   4. lecture du binaire
   5. affichage de la liste lue depuis le binaire