

# M1 Programmation avancée et répartie en Java : TP 6

Jean Fortin et Frédéric Gava  
gava@u-pec.fr ou jean.fortin@gmail.com

## 1 Exercice 1 : Ménagerie

- Écrire les classes suivantes :
  - Animal avec la méthode vide `void manger()` et un champ private entier `id` avec assesseur et un constructeur.
  - Oiseau qui hérite d'Animal avec la méthode `voler`
  - Poisson qui hérite d'Animal avec la méthode `nager`
- Écrivez un code pour construire un Poisson sans le mot clés `new`
- Essayez de modifier l'id d'un animal sans utiliser un assesseur. Essayez d'empêcher cela!
- Écrivez un code pour construire un `Rapace` qui hérite de `Oiseau` mais avec en plus la méthode `chasser`. Sans `extends` et `new`.
- Écrivez un classloader pour les animaux qui affiche quelles classes sont chargées

## 2 Exercice 2 : La calculatrice

Reprenez le code de l'exercice 3 du TP2. Modifiez le code afin que toutes les méthodes statiques de `java.lang.maths` (qui retournent un float à partir d'un float comme p. ex. `cos`, `tan`, etc.) soient acceptées et possiblement exécutables. Bien évidemment, votre code doit pouvoir fonctionner si à l'avenir on ajoute ou supprime des méthodes à cette classe.

## 3 Exercice 3 : La calculatrice parallèle

Complétez l'exercice 2, de telle sorte que le calcul soit dans un thread séparé et que plusieurs questions soient possiblement demandées par l'utilisateur. Modifiez l'affichage pour qu'il donne la question suivie de sa réponse sans interférence d'autres questions-réponses. L'idée est que potentiellement, le temps de calcul de la réponse peut être gros comme sur certains serveurs.