

## Programmation Java

### DM - Éléphants et sardines en essaim

1. Questions préparatoires :

- (a) Créer une hiérarchie de classes qui représente partiellement le règne animal.
  - (b) À chacune des espèces animales, associer une forme géométrique (d'un type similaire au type `Shape` discuté en cours) agrémentée de propriétés supplémentaires comme la couleur ou l'épaisseur du bord (une extension/modification du type `Shape` est donc ici nécessaire).
  - (c) Faire en sorte de pouvoir représenter un groupe d'animaux dans une fenêtre graphique, chacun par sa forme géométrique correspondante (utilisation *élémentaire* de `java.awt` ou `javax.swing` requise).
2. À chacune des espèces animales, associer un comportement de déplacement, et en faire la représentation dans une fenêtre graphique (on placera un ou plusieurs animaux dans la fenêtre graphique et on les laissera se déplacer pendant un laps temps fixé).
3. À chacune des espèces animales, associer un comportement de déplacement lorsqu'ils se trouvent en groupe (pas nécessairement corrélé avec celui de la question 2) : le comportement de chacun des animaux doit pouvoir être directement influencé par leurs voisins directs (afin de les imiter, de réduire la distance avec eux si celle-ci est trop grande, tout en tentant de préserver une distance minimale afin d'éviter les heurts et les chevauchements). En d'autres termes, il s'agit d'obtenir un comportement de bancs, essaims, vols, meutes, etc.

Représenter dans une fenêtre graphique le déplacement de tels groupes d'animaux (avec des conditions initiales et des vitesses de simulation différentes).

NB: Ne pas ajouter d'interaction graphique à votre simulateur. Il ne vous est nullement demandé ici de créer une interface graphique, seulement une *représentation graphique*.

4. Présenter l'ensemble de votre programme par un diagramme UML de classes.

*Directives :*

- Ce devoir est à rendre pour le 7 janvier à 16h. M'envoyer un \*.tgz qui contiendra les sources Java, le diagramme UML en pdf ou jpg, et un petit texte d'explication en pdf de 4 pages au maximum – à narbel@labri.fr.
- Ce devoir peut s'effectuer par groupe de deux.
- Les points principaux pris en compte pour l'évaluation de ce devoir:
  1. La qualité d'écriture du code Java (structuration, indentation, lisibilité);
  2. La qualité de la conception objet (séparation des rôles entre les différentes classes, la définition adéquate de leurs liens);
  3. La qualité du diagramme UML (disposition adéquate des classes et de leurs liens);
  4. L'exhaustivité de la réalisation,
  5. La qualité/robustesse du fonctionnement du programme.
- Attention au plagiat de code : si vous empruntez ou si vous vous inspirez d'un code trouvé sur internet, vous devez impérativement en indiquer la provenance et en justifier la raison.