

Indications : Il vous faut respecter la présentation des listes faites en cours. C'est à dire que la modélisation doit se faire en utilisant 3 classes :

- une pour le contenu,
- une pour les cellules, où se trouve l'essentiel de la gestion du chaînage,
- et une principale, qui modélise la liste dans sa globalité, et qui implémente les méthodes classiques de l'interface de liste.

Dans ces exercices nous nommons les classes **Contenu**, **Cellule**, **Liste** d'une autre façon, mais vous les reconnaîtrez sans difficultés.

Exercice 1

On souhaite réaliser une variante à une dimension du jeu de la vie. On utilisera une liste chaînée afin de modéliser une ligne d'animaux. On définira donc une classe **Animal** avec un attribut **regime** de type **String** indiquant si l'animal est herbivore ou carnivore et cet animal sera inclus dans une **Cellule** dont un attribut **suivant** sera lié à une autre **Cellule** ainsi qu'une classe **Enclos** contenant un attribut **debut** de type **Cellule**. Ces attributs seront privés.

1. Écrivez les déclarations de ces trois classes et ajoutez :
 - dans la classe **Enclos** un constructeur donnant un enclos vide,
 - dans la classe **Cellule** un constructeur recevant un argument de type **String**, et permettant d'intégrer un nouvel animal ayant ce régime.
2. Écrivez une méthode **ajouter** qui prend en argument un régime alimentaire et qui ajoute un **Animal** au début d'un enclos. (Cela doit être clair pour vous où il faut écrire cette méthode)
3. Écrivez le code permettant de créer un enclos contenant deux herbivores puis un carnivore, et un autre enclos contenant trois herbivores puis un carnivore.
4. Écrivez une méthode pour compter le nombre d'animaux
5. Écrivez une méthode transformant tous les animaux carnivores en herbivore et réciproquement.
6. Écrivez une méthode affichant le régime de tous les animaux de la liste (dans l'ordre qui vous convient).
7. (**à faire à la maison**) Écrivez une méthode simplifiant cet affichage en l'exprimant sous la forme "<premier régime>" associé au nombre d'animaux successifs correspondant, puis "<l'autre régime>" suivi du nombre d'animaux successifs ayant ce régime etc...

Exercice 2

Ces enclos servent à présent de support au jeu de la vie : à chaque phase les deux étapes suivantes ont lieu dans cet ordre :

- tout herbivore ayant commencé le tour à côté d'un carnivore est mangé (et retiré de la liste) ;
- une fois la règle précédente appliquée, si deux herbivores sont côte à côte, un nouvel herbivore est placé entre les deux, disponible pour le tour suivant.

1. Donnez l'évolution de l'enclos lors des trois premiers tours de jeu sur les deux exemples de la question 3.

2. Écrivez une méthode réalisant la deuxième étape d'un tour de jeu, *i.e.* la reproduction des herbivores. (La solution la plus simple consiste à créer une nouvelle liste en conservant dans un premier temps la liste précédente en l'état, cela permet de bien distinguer les états "avant-après" d'une phase)
3. Écrivez une méthode réalisant la première étape d'un tour de jeu, *i.e.* le repas des carnivores.
4. Écrivez et testez une méthode `jouer` recevant un argument `nbTours` de type `int` et réalisant les `nbTours` premiers tours de jeu.