

Interrogation n°2

La production finale devra être rendue sous la forme du code. Le plus simple est de *zipper* le répertoire du code source, de le renommer de votre nom (ex. : `yunes.zip` si votre nom de famille est `yunes`) et de l'envoyer par mail à `Jean-Baptiste.Yunes@u-paris.fr`. Le rendu final devra contenir un `main` dans une classe `Test` permettant de tester toutes les fonctionnalités implémentées....

Le but de ce sujet est d'écrire un simulateur d'aquarium. L'idée est la suivante :

- Un aquarium contient deux types d'êtres vivants : des algues et des poissons, qui peuvent vieillir et mourrir, et se reproduire
- Les poissons sont de deux types : `Herbivore`, ou `Carnivore`,
- Les carnivores mangent les herbivores qui à leur tour mangent les algues
- Le temps sera divisé en jours. A chaque jour tous les poissons vieillissent, se nourrissent si ils ont faim, se reproduisent, meurent etc...

Dans la suite, je vous suggère d'écrire des fonctions pour vos classes. Vous pouvez bien sur, et c'est recommandé, en écrire d'autres si cela vous semble pertinent.

1. La classe `EtreVivant` vous est donnée :

```
public classe EtreVivant {
    private int age; // compté en jours...
    protected int sante; // si <=0, mort!
    public int getAge() { return age; }
    public boolean isVivant() { return sante>0; }
    public int getSante() { return sante; }
    public void unJourDePlus() { age++; }
    public void gagne(int v) { sante += v; }
    public void perd(int v) { sante -= v; }
}
```

Ajoutez un constructeur qui permettra d'initialiser correctement un être vivant de sorte qu'il soit d'âge 0 à sa naissance et que sa santé soit tirée au hasard entre 10 et 20.

2. Créez une classe `Poisson` qui hérite d'`EtreVivant` sans méthode en plus.
3. Créez une classe `Algue` qui hérite d'`EtreVivant` sans méthode en plus.
4. Créez deux interfaces `Carnivore` `Herbivore`. Ses deux interfaces auront respectivement une méthode `manger(...)` avec un paramètre permettant d'indiquer que les carnivores peuvent manger n'importe quel poisson, et que les herbivores ne mangent que des algues.
5. Créez deux classes `PoissonCarnivore`, `PoissonHerbivore` qui hériteront de la bonne classe et implémenterons la bonne interface. On sait que lorsqu'un carnivore mange un autre être, le carnivore gagne 3 points de santé et l'autre être perd 5 points de santé. Lorsqu'un herbivore mange une algue, le poisson gagne 2 points de santé et l'algue en perd 3.
6. `Aquarium` Cette classe représente l'aquarium. Les champs à écrire indispensables seront :
 - un champ `int` représentant le jour (jour 0, puis jour 1, etc.),
 - une structure contenant l'ensemble des `Canivores` de l'aquarium,
 - une structure contenant l'ensemble des `Herbivores`,
 - une structure contenant les `Algues`.

Une méthode `unJourDePlus()` permettra de faire vieillir tous les poissons de l'aquarium, diminuer leur santé de 1, puis de les faire manger. Un poisson ne mange que lorsqu'il a faim, c'est-à-dire lorsque sa santé décline (≤ 5), et dans ce cas on trouve un élément mangeable adéquat en tirant au hasard (une algue pour les herbivores (`getRandomAlgue()`), un poisson quelconque pour les carnivores (`getRandomPoisson()`). À la fin de la journée on nettoie l'aquarium en supprimant les êtres morts (`nettoyage()`). Les êtres meurent lorsque leur santé est inférieure ou égale à 0, ou alors s'ils sont trop vieux (les poissons meurent en 10 jours et les algues en 25 jours). On notera que les être vivants on tous une identité propre (un numéro).

7. Créez une classe `Test`. Dans la méthode `main` créez un aquarium. Ensuite un menu permettra :
 - d'ajouter un herbivore,

- d'ajouter un carnivore,
- d'ajouter une algue,
- de faire passer 1 jour,
- d'afficher le contenu de l'aquarium,
- d'arrêter le programme.

Les affichages devront être «agréables» :

```
Aquarium (jour 10):  
Carnivore 12: (santé 7, âge 3).  
Carnivore 2: (santé 3, âge 7).  
Herbivore 5: (santé 10, âge 2).  
Algue 1: (santé 13, âge 9).
```

8. Un scientifique désire étudier la vie des poissons dans ce milieu. Pour cela il souhaite donc garder certaines informations en mémoire, comme la liste de tous les êtres morts, et pour chacun d'entre eux : la liste de ce qu'il ont mangé et quand... Comment faire? Faites...