

Licence MASS – INF F3 – TP n° 5

2013–2014

Ce TP reprend le principe du TP précédent en introduisant la notion d'héritage. Au travers d'un aquarium, vous allez voir évoluer différentes espèces d'êtres vivants qui partagent un certain nombre de caractéristiques.

1 Présentation de l'aquarium

On ne s'intéresse, dans l'aquarium, qu'à la faune et la flore. Même s'il est plein d'eau, l'aquarium est donc initialement vide. On va pouvoir ajouter à l'intérieur des algues et plusieurs espèces de poissons. Et il sera possible de le regarder évoluer au cours du temps. Le temps sera représenté par des tours et, à chaque tour, il sera possible de voir des poissons ou des algues naître et mourir. On pourra même assister aux repas des poissons qui ont faim.

Chaque être vivant est donc susceptible de vieillir et de se faire manger (au moins en partie). À chaque tour, tous les êtres vivants deviennent plus vieux et ceux qui sont trop vieux meurent. Les poissons, quand à eux, doivent se nourrir quand ils ont faim. Certains d'entre eux sont herbivores et se dirigeront donc sur les plantes pour ça, les autres, carnivores, vont préférer se nourrir sur leurs congénères.

En plus de cela, les êtres vivants peuvent se reproduire. Les algues ont juste besoin d'assez de vivacité pour se séparer en deux alors que les poissons doivent se trouver un ou une partenaire du sexe opposé.

2 D'un point de vue informatique

L'aquarium et les différents types d'être vivant vont être représentés par des classes. Afin d'éviter d'écrire les mêmes traitements sur des aspects similaires de la vie de l'aquarium, une hiérarchie de classe sera mise en place et l'héritage sera utilisé. Les classes décrivant les êtres vivants vont être écrites avant la classe décrivant l'aquarium qui, lui, ne servira qu'à contenir un ensemble d'êtres vivants.

2.1 Les êtres vivants

Écrire une classe **EtreVivant** qui regroupe les capacités communes aux algues et aux poissons. Tous les êtres vivants peuvent vieillir et se faire manger. Utilisez deux attributs pour représenter l'âge et la résistance des êtres vivants. Un être vivant meurt à un âge de 20 ou à une résistance de 0. Quand un être vivant est mis dans l'aquarium, il possède déjà un certain âge et sa résistance est de 10.

Ajoutez une méthode `vieillir` qui va augmenter l'âge de l'être vivant et une méthode `seFaireManger` qui va diminuer de 5 la résistance de l'être vivant. Pour préparer la suite, la méthode `vieillir` devra renvoyer un être vivant (pour gérer les naissances). Pour l'instant, elle retourne `null`.

Ajoutez une méthode `estMort` qui renvoie `true` si l'être vivant ne remplit plus les conditions pour être vivant.

Surchargez la méthode `toString` de la classe `Object` pour retourner le nom, l'âge et la résistance des êtres vivants. À ce stade, les êtres vivants n'ont pas encore de nom, écrivez une méthode intermédiaire que vous appellerez dans `toString` qui s'occupera de retourner le nom de l'être vivant. Comment faire en sorte que cette méthode fonctionne mais ne soit implémentée que lorsqu'on connaîtra le nom des êtres vivants ? Qu'est-ce que cela implique sur la classe `EtreVivant` ?

2.2 Les algues

Écrire une classe `Algue` qui hérite de `EtreVivant`. L'algue possède les mêmes caractéristiques qu'un être vivant de base à cette exception près que quand elle vieillit, elle pousse et peut se dupliquer.

Surchargez la méthode `vieillir` pour que, en plus du comportement normal d'un être vivant, l'algue gagne 1 point de résistance par tour. Si la résistance de l'algue est de 11 ou plus, elle se sépare en deux : elle garde la moitié de sa résistance et l'autre moitié est utilisée pour créer une nouvelle algue. Cette nouvelle algue est retournée par la méthode.

Une algue n'a pas de nom particulier : la méthode dédiée aux noms des êtres vivants devra renvoyer la chaîne de caractères `Algue`.

2.3 Les poissons

Écrire une classe `Poisson` qui hérite de `EtreVivant`. Les poissons sont un peu plus complexe que les algues car ils vont pouvoir manger et se reproduire. Ajoutez les attributs nécessaires pour stocker la faim et le sexe d'un poisson.

De même, pour égayer l'aquarium, chaque poisson porte un nom. Ajoutez les attributs et modifiez les méthodes nécessaires pour représenter cela.

Les poissons bougent à l'intérieur de l'aquarium. Ils se retrouvent donc à côté d'autres êtres vivants. Écrire une méthode `aCoteDe` qui prend un `EtreVivant` en paramètre et qui va stocker cet être vivant comme étant à côté du poisson pour le tour.

La faim d'un poisson est notée entre 0 et 10 et débute à 0. À partir de 5 un poisson a faim et cherchera à se nourrir. Arrivé à 10, le poisson meurt de faim. Surchargez la méthode `estMort` pour tenir compte de cette condition.

La faim d'un poisson augmente d'un point à chaque tour, surchargez la méthode `vieillir` pour représenter cela. Si un poisson est affamé, il va essayer de manger, sinon il va essayer de se reproduire. Ajoutez ce comportement dans la méthode `vieillir`. Comme on ne connaît pas encore le régime alimentaire des poissons et les compatibilités entre espèces, que faut-il faire pour déléguer ce comportement dans une classe qui aura cette information ? Qu'est-ce que cela implique sur la classe `Poisson` ?

2.4 Régime alimentaire

Les poissons peuvent avoir deux régimes alimentaires distincts. Créez une classe `PoissonHerbivore` et ajoutez ou modifiez les méthodes responsables des repas. Un poisson herbivore mange une algue si il est à côté et regagne 3 points de résistance. Sa faim retombe alors à 0.

Créez une classe `PoissonCarnivore` et faites de même. Un poisson carnivore mange un autre poisson si il est à côté et regagne 5 points de résistance. Sa faim retombe alors à 0. Attention, un poisson carnivore ne mange pas de poisson de la même espèce (classe, voir 2.5) que lui.

2.5 Les différentes espèces de poissons

Chaque branche de poissons va être déclinée en 3 (pour rester simple) espèces distinctes. Pour les poissons carnivores, créez les classes `Rouget`, `Labre` et `Merou`. Pour les poissons herbivores vous pouvez utiliser `Hippocampe`, `Salarias` et `Chirurgien`. Ces catégories ne représentent pas forcément la réalité mais permettent d'introduire un peu de diversité dans notre aquarium virtuel.

Créez ou modifiez les méthodes nécessaires pour permettre à ces espèces de poisson de s'accoupler. Une règle simple est de faire en sorte que seuls un mâle et une femelle de la même espèce peuvent se reproduire entre eux et produisent un bébé de la même espèce. Quand vous instancierez le bébé, utilisez la concaténation des deux noms des parents pour celui du bébé.

Vous pouvez aussi implémenter des règles différentes en fonction des espèces. Par exemple, faire en sorte que certaines espèces puissent s'accoupler avec d'autres espèces que la leur, introduire la notion de poissons hermaphrodites qui changent de sexe avec le temps ou au besoin si le partenaire n'est pas du bon sexe.

Une fois les espèces de poissons terminées, pensez à tester votre hiérarchie de classe. Créez une méthode `main` dans n'importe quelle classe du package. Dans cette méthode, créez quelques poissons et quelques algues, placez les manuellement les uns à côté des autres et appelez les méthodes `vieillir` de chacun d'entre eux pour vérifier ce qu'il se passe.

3 L'aquarium

Une fois que le nécessaire pour peupler l'aquarium a été créé, vous pouvez vous intéresser à faire fonctionner un aquarium de manière semi-automatique.

Créez une classe visuelle qui va servir à contenir et faire évoluer un ensemble d'êtres vivants (créez les attributs nécessaires). La fenêtre devra contenir une zone pour afficher le contenu de l'aquarium et une zone d'« actions » pour ajouter des algues, des poissons et faire évoluer l'aquarium.

En fonction de la méthode utilisée pour gérer les êtres vivants, vous allez probablement devoir définir une taille limite au nombre d'être vivants pouvant peupler l'aquarium. Vous pouvez utiliser un `JSpinner` ou un `JSlider` pour laisser le choix de cette taille à l'utilisateur.

Écrivez une méthode `miseAJour` qui va permettre d'afficher dans la zone dédiée la liste des êtres vivants présents dans l'aquarium. Vous pouvez séparer cette zone en deux pour présenter d'un côté les algues et de l'autre les poissons. Utilisez cette méthode à chaque fois que vous modifiez l'aquarium (ajout ou modification de taille).

Concernant les boutons d'ajout, le principe est simple : pour les algues, une nouvelle algue est ajoutée et pour les poissons, l'espèce est demandée pour pouvoir créer le poisson adéquat avant de l'ajouter. Basez-vous sur le TP n°1 pour demander les informations nécessaires aux différents constructeurs. Vous pouvez aussi tenter de créer un **JDialog** personnalisé pour pouvoir demander plusieurs informations en même temps.

Pour finir, le bouton pour faire évoluer l'aquarium va devoir réaliser 1 tour d'action pour tous les êtres vivants. Ce tour comporte 3 étapes :

- les poissons se déplacent à côté de quelque chose ;
 - tous les êtres vivants vieillissent ;
 - les morts sont retirés de l'aquarium et le ménage est fait.
- N'oubliez pas de mettre à jour l'affichage à la fin de chaque tour.