

TD n° 1

Conception de classes – 1

N'oubliez pas de vous inscrire sur le DidEL de code POO ! Les groupes sont créés, pensez par conséquent à vous inscrire dans *VOTRE* groupe. Si vous n'avez pas le même groupe de TD et TP, inscrivez vous dans les deux.

Lisez attentivement le sujet, un exercice est toujours plus facile à faire quand on a *bien* lu l'énoncé de l'exercice en entier.

Il est probable que l'exercice 3 ne soit pas traité en TD, vous pouvez y réfléchir chez vous avant le TP.

Conceptions

Dans ces exercices, on ne vous demande pas de code ! Ils seront programmés en TP.

On s'efforcera de ne pas mettre dans la même méthode du code qui **modifie** l'objet et du code qui interagit avec l'utilisateur (écriture, lecture).

Exercice 1 [Tamagotchis]

Présentation et but du programme Le Tamagotchi est un animal de compagnie virtuel japonais. Ce nom est un mot-valise créé à partir des mots tamago ("œuf") et de l'abréviation de watchi qui vient du mot anglais watch ("montre"). Le jeu consiste à simuler l'éducation d'un animal à l'aide d'une petite console miniature, de la taille d'une montre, dotée d'un programme informatique.

Un tamagotchi va donc être un objet qui vit (perd de l'énergie) et qui doit être alimenté pour survivre (augmenter son énergie).

Conception de la classe Tamagotchi Pour être plus précis, on veut qu'un Tamagotchi naisse avec une certaine énergie, qu'à chaque "tour", il perde un peu d'énergie à cause des dépenses énergétiques normales et aussi qu'il vieillisse (on compte son âge en "tour"). Il peut regagner de l'énergie si on le nourrit, mais sans dépasser une valeur limite qui est propre à chaque Tamagotchi et est fixée à la naissance de manière aléatoire. Les quantités qu'il perd ou qu'il gagne sont elles aussi tirées au hasard dans un intervalle fixé, le même pour tous les Tamagotchis.

Toutes les valeurs tirées aléatoirement le sont dans des intervalles fixés communs à tous les Tamagotchi.

Dans cette classe, on ne fera aucun affichage (c'est-à-dire pas de `print`).

1. Quelles méthodes seront nécessaires pour pouvoir interagir avec un Tamagotchi ?

2. Quels sont les attributs d'objet dont nous aurons besoin ?
3. Quels sont les attributs de classe ?
4. Quel(s) constructeur(s) faut-il prévoir ?
5. Quels tests feriez vous pour vérifier que vous n'avez pas fait d'erreur dans votre code ?

Programme principal Dans cette classe, on fera un main et éventuellement des méthodes statiques.

On n'écrira pas le code, mais on écrira ce qu'il faut faire et on réfléchira aux méthodes de Tamagotchi qui serviront pour ça. Si vous vous apercevez qu'il manque des méthodes ou des attributs dans la classe Tamagotchi, ajoutez-les.

Exemple d'exécution du main :

Quel nom pour le nouveau tamagotchi : XYZ

-----Cycle no 1 -----

XYZ : je suis affamé !

Voulez-vous nourrir XYZ ? 0

-----Cycle no 2 -----

XYZ : tout va bien !

Voulez-vous nourrir XYZ ? 0

XYZ: Je n'ai pas faim!

-----Cycle no 3 -----

XYZ : tout va bien !

Voulez-vous nourrir XYZ ? N

-----Cycle no 4 -----

XYZ : je suis affamé !

Voulez-vous nourrir XYZ ? N

-----Cycle no 5 -----

XYZ : je suis affamé !

Voulez-vous nourrir XYZ ? N

-----Cycle no 6 -----

XYZ : je meurs... Arrrggh !

Exercice 2 [Tamagotchis bis]

Dans cet exercice, on veut faire des variantes de Tamagotchis. Les classes correspondantes doivent avoir les mêmes en-têtes de méthodes, mais pas forcément le même code pour ces méthodes, ni les mêmes attributs, ni bien sûr les mêmes constructeurs.

On peut, par exemple, imaginer des Tamagotchis éternels (qui ne meurent jamais), des Tamagotchis qui ne gagnent aucune énergie quand on les nourrit, . . .

Inventez quelques variantes et indiquez de quels attributs vous aurez besoin pour ça.

S'il vous reste du temps . . .

Exercice 3 [Rationnels]

Dans cet exercice, on veut concevoir une classe qui permettra de manipuler des nombres rationnels.

Un nombre rationnel est représenté par deux entiers, le numérateur et le dénominateur. Communément il est noté $\frac{\textit{numérateur}}{\textit{dénominateur}}$. De plus le dénominateur est toujours strictement positif.

Quelles opérations ou tests voulez-vous pouvoir faire sur un rationnel ?

1. Quels sont les attributs d'objet dont nous aurons besoin ?
2. Quels sont les attributs de classe ?
3. Quelles méthodes seront nécessaires ?
4. Quels tests feriez-vous pour vérifier que vous n'avez pas fait d'erreurs dans votre code ?