université PARIS DIDEROT PARIS 7

INITIATION À LA PROGRAMMATION 2 (IP2)

TD1 : Objets, attributs et constructeurs

Préambule

Inscrivez-vous au cours Moodle *Introduction à la programmation java Semestre 2*. Vous y retrouverez tous les énoncés de TD et TP, et des documents importants. Il vous permettra de communiquer avec vos enseignants, ainsi qu'à déposer les travaux qui seront notés en Contrôle Continu.

Il vous faut vous inscrire à la fois au groupe de TD, et au groupe de TP qui **correspondent à votre inscription pédagogique**. Les modules d'inscriptions seront disponible une quinzaine de jours, ils sont déclaratifs et permettrons de commencer à communiquer en attendant d'avoir les listes officielles définitives. (Nous les croiserons à ce moment là)

1 Des Animaux

On veut définir une classe `Animal` qui se caractérise par :

- un attribut `nom` qui sera une chaîne de caractères,
- un `sexe` de type `char` qui vaudra `'m'` pour un mâle, `'f'` pour une femelle,
- un `age` qui sera un entier donnant l'âge de l'animal en jours,
- un `poids` qui sera un entier donnant le poids de l'animal en kilogrammes,
- un `espece` qui sera une chaîne de caractères.

Remarques pour la suite :

- pour simplifier les années feront exactement 365 jours ;
- toutes les méthodes seront statiques ici ; vous apprendrez rapidement que ce n'est pas forcément la meilleure manière de procéder.
- on ne se pose pas encore de questions sur la visibilité, donc le mot clé public (ou l'absence de précision) conviennent très bien pour ce TD.

1. Écrivez une classe `Animal` permettant de modéliser nos animaux selon la description faite.
2. Écrivez un constructeur prenant des arguments (dans l'ordre décrit plus haut) permettant d'initialiser les attributs au moment de la création de l'objet.
3. Que doit-on écrire pour créer un zèbre mâle de 5 ans, pesant 300 kg s'appelant Marti ? un hippopotame femelle de 7 ans, d'une tonne et demie s'appelant Gloria ?
4. Que doit-on écrire pour diminuer le poids de Gloria de 50 kg ?
5. Que va afficher le code suivant ?

```
1 Animal a = new Animal("Melman", 'm', 1230, 998, "girafe");
  Animal b = a;
3 b.poids = 950;
  System.out.println(a.poids);
```

6. Écrivez une méthode statique `description` qui prend en argument un `Animal` et qui renvoie une chaîne de caractères le décrivant ("Je m'appelle Rico, je suis un manchot mâle, j'ai 1129 jours et je pèse 30 kg").
7. *(A faire chez vous)* Écrivez une méthode `descriptionBis`, similaire à la précédente, qui exprimera plus lisiblement l'âge en nombre d'années complètes et en nombre de jours restant. Pour l'exemple précédent on aurait donc : "Je m'appelle Rico, je suis un manchot male, j'ai 3 ans et 34 jours et je pese 30 kg".

8. On ajoute un attribut `faim` à cette classe pour représenter l'état de satiété de l'animal. Il sera compris entre `0` et `10` : à `0` il n'a pas faim, à `10` il est mort de faim. Modifiez le constructeur précédent pour qu'il prenne ce paramètre dans ses arguments. Vous tronquerez la valeur donnée pour qu'elle reste comprise dans l'intervalle de définition.
9. Écrivez une méthode `nourrir` qui diminue le degré de faim d'un animal de `1` unité. De plus, nourrir un animal qui n'avait pas faim le fera grossir de 10%.
10. Écrivez une méthode `lePlusGros` qui prend en argument un tableau d'animaux et qui en renvoie un de poids maximal. Testez-la sur un tableau d'au moins trois animaux.
11. Écrivez une méthode `reproduction`, prenant en argument deux animaux et qui, s'ils sont de sexes opposés, de la même espèce, et si leurs niveaux de faim sont tous deux inférieurs à 5, renvoie un nouvel animal de la même espèce et de sexe tiré aléatoirement. Le poids sera un nombre aléatoire situé entre les poids des deux parents. Le nom sera la concaténation des deux noms des parents, mis entre parenthèses. Quel est le résultat de la reproduction dans les cas où les conditions ne sont pas remplies ?
    Remarque : Pour tirer aléatoirement un nombre réel entre 0 inclus et 1 exclu, on utilisera la méthode `Math.random()`. Pour obtenir un nombre entre 2 valeurs `x` incluse et `y` exclue, on fera donc le calcul simple : `(int)(Math.random()*(y-x)+x)`.

2 Des Zoos *(À commencer en TD et à terminer chez vous. Faites au moins les méthodes de 1 à 4.)*

1. Définissez une classe `Enclos`, dont les objets sont destinés à contenir un ensemble d'animaux : vous utiliserez un attribut de type tableau d'animaux dont la capacité sera donnée en argument au constructeur et un attribut entier dont la valeur indiquera sa population actuelle.
    Indication : La capacité du tableau est le nombre d'animaux maximum qu'il peut contenir, c'est donc sa taille. Le tableau sera utilisé comme suit : au départ, toutes les cases sont à `null` et sa population est donc à 0. En cours d'utilisation, les cases à `null` seront toutes regroupées à droite du tableau. Et donc les cases contenant des animaux seront entre 0 et `population - 1`.

2. Ecrivez une méthode pour ajouter un animal à un enclos. Elle retournera vrai ou faux selon que l'ajout a été possible ou non.
3. Définissez maintenant la classe `Zoo`, qui possède comme attributs
    — une `ville` de type chaîne de caractères,
    — un `contenu` de type tableau d'`Enclos`.
    Le constructeur d'un zoo prendra en argument son nom, un entier précisant le nombre d'enclos, et un second entier précisant la taille de ses enclos, ils seront initialement vides.
4. Écrivez une méthode permettant d'ajouter un animal au zoo s'il existe un enclos dans lequel on peut l'insérer. Elle retournera vrai ou faux si l'ajout a été possible ou non.
5. Écrivez une méthode qui nourrira tous les animaux du zoo qui ont plus de 5 en faim.
6. Écrivez une méthode `unJourPasseAuZoo` qui fait que tous les animaux vieillissent un peu, voient leur faim augmenter de 2, et ceux pour qui elle atteint 10 disparaissent *(difficile)* .
7. *(difficile)* Complétez la méthode précédente pour que si une reproduction au sein d'un enclos est possible alors elle ait lieu (dans la limite de la place disponible). Une femelle ne donnera qu'un descendant par jour. Les nouveaux-nés seront féconds dès le lendemain.