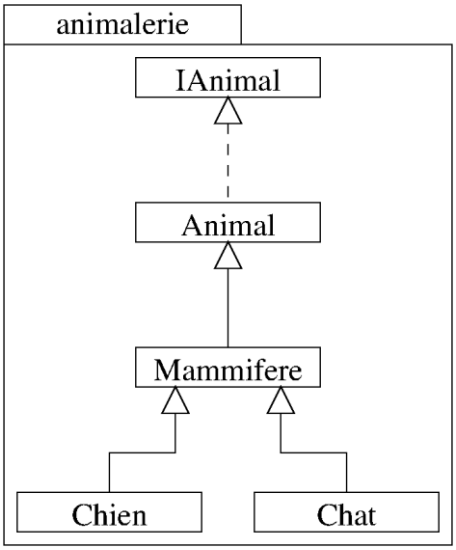
TP 6 : héritage

I. Le package animalerie

Nous souhaitons créer le package `animalerie` tel qu'il est montré sur la figure ci-dessous :



L'interface `IAnimal` est la suivante :

```
package animalerie;

/**
 * Interface IAnimal devant etre implementee par la classe Animal
 * @author Cyril Rabat
 * @version 03/03/2011
 */
public interface IAnimal {

    /**
     * Fait crier l'animal
     */
    public void crier();

}
```

Un animal dispose d'un nom (qui correspond à son prénom tel que Médor, Azraël ou LeChat) et d'un nombre de pattes.

1. À votre avis, la classe doit-elle posséder un constructeur par défaut ?
2. Et un constructeur par initialisation ?
3. Et un constructeur par copie ?
4. Et des *getters* pour chaque attribut ?
5. Et des *setters* pour chaque attribut ?
6. Quels attributs sont nécessaires ?
7. Écrivez la classe `Animal` qui, en plus de ce que vous avez indiqué, contiendra une méthode `toString` et une méthode `afficher`. Lorsqu'un animal est affiché à l'écran, on obtient : `Bidule, animal à 4 pattes` (si l'animal s'appelle Bidule et s'il possède 4 pattes...). Pour un animal, la méthode `crier` se contente d'afficher `hum, hum` à l'écran.

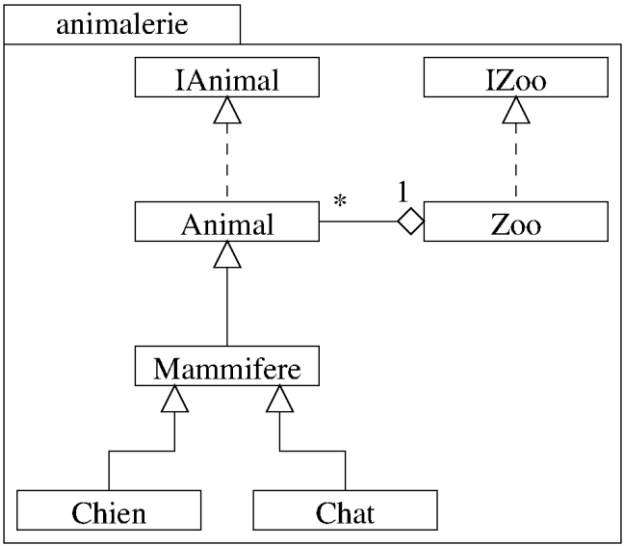
Nous souhaitons maintenant écrire la classe `Mammifere`. Un mammifère possède obligatoirement 4 pattes à sa création (tant pis pour les chauves-souris, les baleines et autres...). Lorsqu'un mammifère est affiché à l'écran, on obtient : `Bob, mammifere a 4 patte(s)` (s'il s'appelle Bob et qu'il possède 4 pattes). Tout comme un animal, le cri du mammifère est aussi `hum, hum...`.

8. Quel(s) attribut(s) doit-on spécifier dans la classe Mammifere ?
9. Doit-on réécrire les *getters* correspondant au nom et au nombre de pattes ?
10. Doit-on réécrire la méthode crier ?
11. Expliquez si la méthode toString peut réutiliser ou non, la méthode toString de la classe Animal.
12. Écrivez la classe Mammifere.
13. Écrivez maintenant les classes Chien et Chat. Un chat fait miaou, miaou et le chien ouaf, ouaf. Un chien s'affichera Chien Medor, mammifere a 4 patte(s) (inutile de préciser pour son nom et son nombre de pattes !?!) et un chat Chat Azrael, mammifere a 4 patte(s).
14. Écrivez la classe `TestAnimalerie` qui permet de créer des `Animal`, `Mammifere`, `Chien` et `Chat`. Vérifiez que vos classes sont correctes.

La classe `TestAnimalerie` doit être dans le répertoire du TP 6 et non pas dans le répertoire du package `animalerie` (qui lui, doit être dans le répertoire packages, du répertoire `Info0201`).

15. En quelques instructions, illustrez le principe du polymorphisme

Nous souhaitons créer un Zoo contenant des animaux (pour le moment, des chiens et des chats, ce qui en fait un zoo d'exception). Il est caractérisé par un certain nombre de cages, chacune pouvant être vide (par défaut) ou pleine. Pour cela, nous choisissons d'utiliser un tableau. Le package `animalerie` est maintenant le suivant :



16. Quels sont les attributs nécessaires dans la classe Zoo (précisez leur type) ?

Nous supposons qu'un Zoo créé par défaut contient un nombre de cages aléatoire entre 10 et 20 (ces nombres minimum et maximum doivent être déclarés comme des constantes). Le constructeur par initialisation prend un nombre de cages qui doit être compris entre le minimum et le maximum (gérez le cas où un nombre invalide est spécifié).

L'interface `IZoo` est la suivante :

```
package animalerie;

/**
 * Interface IZoo devant etre implementee par la classe Zoo
 * @author Cyril Rabat
 * @version 03/03/2011
 */
public interface IZoo {

    /**
     * Ajoute un animal dans une cage du zoo
     * @param a l'animal a ajouter
```

```
     * @param i le numero de la cage
     */
    public void ajouterAnimal(Animal a, int i);

    /**
     * Recupere un animal dans une des cages du zoo
     * @param i le numero de la cage
     * @return l'animal present dans la cage
     */
    public Animal getAnimal(int i);

    /**
     * Supprime l'animal situe dans une cage
     * @param i le numero de la cage
     */
    public void supprimerAnimal(int i);

    /**
     * Fait crier tous les animaux du zoo
     */
    public void faireCrier();

    /**
     * Convertit tout le zoo en chaine de caracteres
     * @return une chaine de caracteres
     */
    public String toString();

}
```

17. Écrivez la classe `Zoo`.
18. Créez une classe `TestZoo` permettant de tester le Zoo. L'utilisateur peut choisir d'afficher le Zoo, d'ajouter un animal de son choix (avec le nom de son choix), de supprimer un animal ou de faire crier tous les animaux.