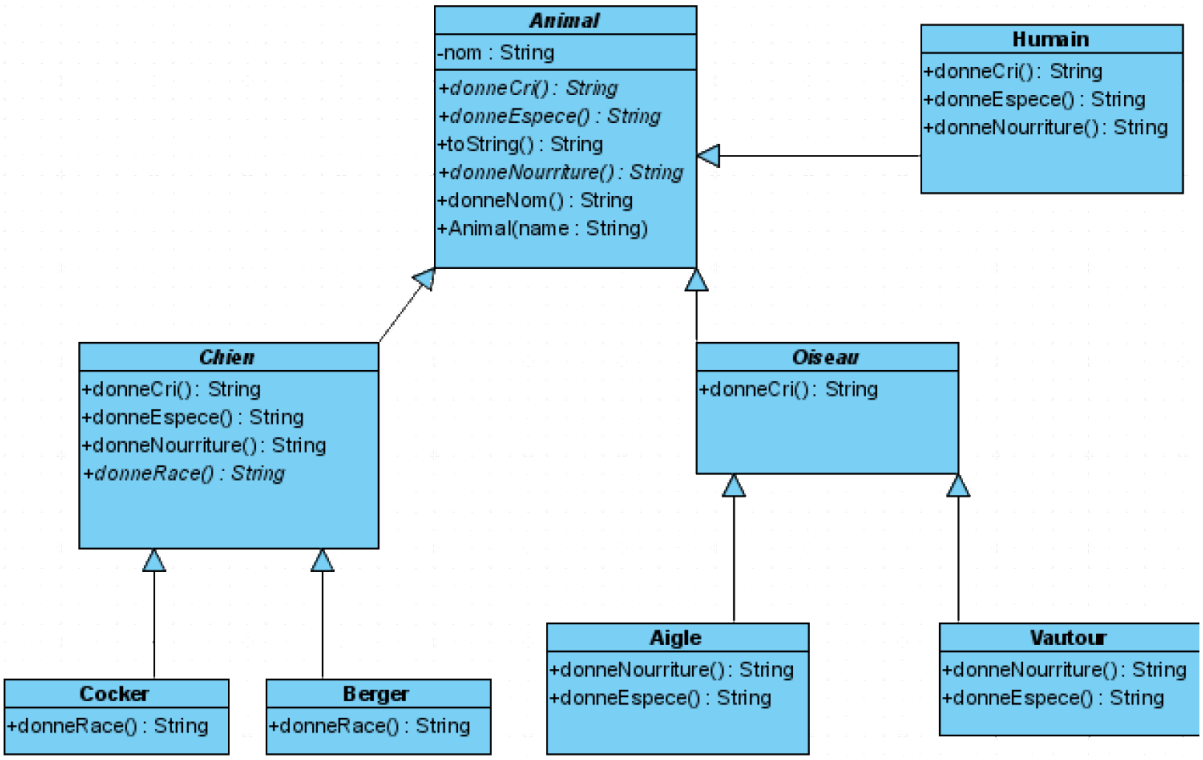
TD 08 : ANIMAUX

Sans aucun respect pour les vraies sciences naturelles, nous allons créer un jeu de classes pour représenter des animaux. La hiérarchie d'héritage est en effet particulièrement adaptée pour représenter une classification des espèces. Enfin, l'abstraction sera nécessaire à la mise en place de ces classes pour permettre l'exploitation du polymorphisme tout en empêchant l'instanciation des classes non terminales dans notre hiérarchie. Le but de cet exercice est principalement de bien réfléchir au moment où une méthode abstraite devient concrète dans la hiérarchie des classes.



Codez ces classes en suivant la hiérarchie d'héritage, en partant des classes mères vers les classes filles. La majorité des constructeurs ont volontairement été omis, à vous de voir si ils sont nécessaires.

Explication concernant les méthodes :

- donneCri() renvoie une chaine donnant le cri de l'animal (par exemple « ouaf »)
- donneEspece() renvoie une chaine donnant l'espèce (complétée le cas échéant par la race par exemple « chien berger »)
- donneNourriture() renvoie une chaine représentant la nourriture de base de l'animal (par exemple « croquettes »)
- donneNom() renvoie le nom de l'animal
- toString () renvoie une chaine indiquant l'identité complète de l'animal sous la forme :

```
Ouaf ! Je m'appelle Médor, je suis : chien cocker.
Je mange principalement : croquettes.
```

Créez un programme principal pour créer toutes ces classes.

Créez ensuite une classe Menagerie qui possède un nom et une liste d'animal. On doit pouvoir ajouter ou retirer un animal d'une ménagerie et la méthode toString de la ménagerie donne son nom ainsi que la liste des animaux qu'elle contient. **Avant de coder cette classe, ajoutez-la sur le diagramme des classes actuel !**

Modifiez le programme principal pour tester votre classe Menagerie.