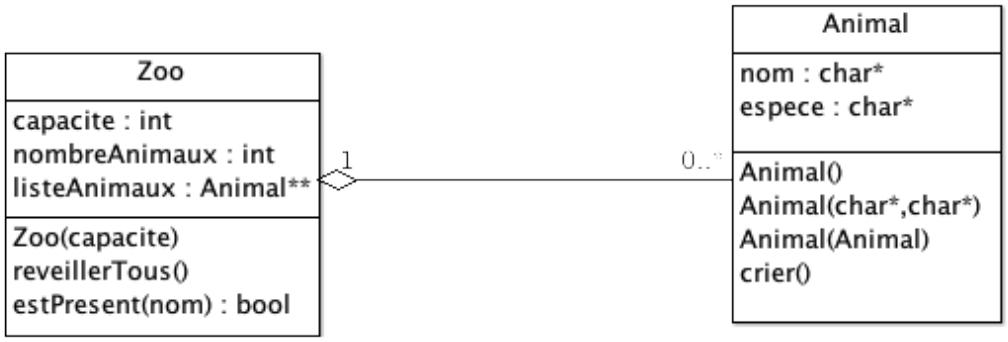
TP 6

Partie 1

1 Introduction

Le but de ce TP est de savoir écrire, compiler et exécuter un programme simple en C++ utilisant quelques classes.
Nous allons représenter un zoo rempli d'animaux. Pour cela, nous allons d'abord créer une classe **Animal**, puis une classe **Zoo**, qui contiendra un tableau d'Animal, et des méthodes d'ajout, d'affichage et de test d'appartenance. Le programme est résumé par le diagramme UML suivant.



Exercice 1

- écrire les fichiers Animal.h et Animal.cpp permettant de définir la classe Animal.
- écrire le fichier main.cpp où l'on créera un objet Animal, et le faire crier (on pourra par exemple afficher un texte "L'animal Toto (tigre) crie").

compilez puis exécutez.

Aide

- Vous disposez déjà du fichier main.cpp, Animal.h, et de l'ossature du fichier Animal.cpp. Vous n'avez qu'à remplir les fonctions du fichier Animal.cpp

- Pour copier un char* dans un autre char*, utilisez la fonction **strcpy** de la bibliothèque **string.h** (pour l'utilisation, référez vous au manuel : **man strcpy** sur linux, et dans tous les cas "C strcpy" sur n'importe quel moteur de recherche)

Exercice 2

- écrire les fichiers Zoo.h et Zoo.cpp permettant de définir la classe Zoo.
- créer un Zoo de capacité 100 et tester les méthodes de la classe :
  - ajouter l'animal créée précédemment
  - ajouter un second animal
  - afficher les animaux du zoo
  - tester la méthode d'appartenance *estPresent()*

Aide

- **listeAnimaux** est un tableau de pointeurs de Animal. Dans le constructeur, on peut donc allouer dynamiquement un tableau de pointeurs de Animal de taille **capacite**.
- ne pas oublier d'initialiser le compteur (variable **nombreAnimaux**)
- dans le destructeur, on ne fait rien car les objets utilisés peuvent continuer à vivre à l'extérieur de la classe
- dans la fonction **ajouter(Animal*)**, on ajoute le nouvel animal à la dernière case du tableau (ne pas oublier d'incrémenter le compteur...)
- pour la méthode **estPresent(char*)** :
  - on recherche un animal à partir de son **nom**, on retourne vrai s'il est présent, et faux sinon.
  - on peut par exemple définir une méthode privée qui prend en entrée un nom et retourne le numéro de la case du tableau qui contient cet animal (ou -1 s'il n'existe pas).
  - il suffit de parcourir le tableau **listeAnimaux** (boucle pour)

  - pour comparer deux char*, on peut utiliser la fonction **strcmp** (pour l'utilisation, se référer au manuel : **man strcmp**)

- dans la fonction **afficherTout()**, on peut utiliser la fonction **afficher()** de la classe Animal

Bonus

- Plutôt que d'utiliser les fonctions **afficher** et **afficherTout**, redéfinir l'opérateur « pour les classes Animal et Zoo.

Partie 2

Exercice 3

- Créer deux classes Cheval et Tigre (ou autre selon votre sensibilité...) **héritant** toutes les deux de Animal. Ces classes ont un ou plusieurs attributs en plus (à vous de les imaginer...)

- **Redéfinir** la méthode de cri et d'affichage dans ces nouvelles classes

- modifier le main pour créer des objets de ces nouvelles classes, et tester la méthode d'affichage pour vérifier que les bonnes méthodes sont appelées.