

## SEQUENCE 3 – HERITAGE

### SEANCES 3 ET 4 : ZOO – ABSTRACTION

#### OBJECTIFS

- Comprendre l'abstraction et la généricité.
- Savoir programmer des classes abstraites et interfaces, en déduire les obligations qui en découlent et savoir les programmer

#### CONSIGNES

Au sein de votre répertoire « **Sequence\_3\_Heritage** ». Créez un nouveau **projet** « **GestionZoo** » dans une **solution** « **GestionZoo** »

#### PARTIE 1 :

1. Définissez la classe **abstraite** **Animal** : en suivant les consignes énoncées ci-dessous :

**Animal**  
Abstract Classe

▲ Champs

- 🔒 anneeNaissance : int
- 🔒 nom : string
- 🔒 poids : double
- 🔒 sexe : SexeAnimal
- 🔒 taille : double
- 🔒 UM\_AGE : string
- 🔒 UM\_POIDS : string
- 🔒 UM\_TAILLE : string

▲ Propriétés

- 🔑 Age { get; } : int
- 🔑 AnneeNaissance { get; set; } : int
- 🔑 Nom { get; set; } : string
- 🔑 Poids { get; set; } : double
- 🔑 Sexe { get; set; } : SexeAnimal
- 🔑 Taille { get; set; } : double

▲ Méthodes

- 🌟 Animal(SexeAnimal sexe, string nom, int anneeNaissance, double poids, double taille)
- 🔒 CompareTo(object obj) : int
- 🔒 Cri() : string
- 🔒 Equals(object obj) : bool
- 🔒 EstAdulte() : bool
- 🔒 ToString() : string

**SexeAnimal**  
Enum

- Feminin
- Masculin
- Hermaphrodite

Ajoutez en dehors de la classe (au-dessus), l'énum suivante :

```
public enum SexeAnimal { Feminin = 'F', Masculin = 'M' , Hermaphrodite = 'H' };
```

Ajoutez les 3 constantes suivantes :

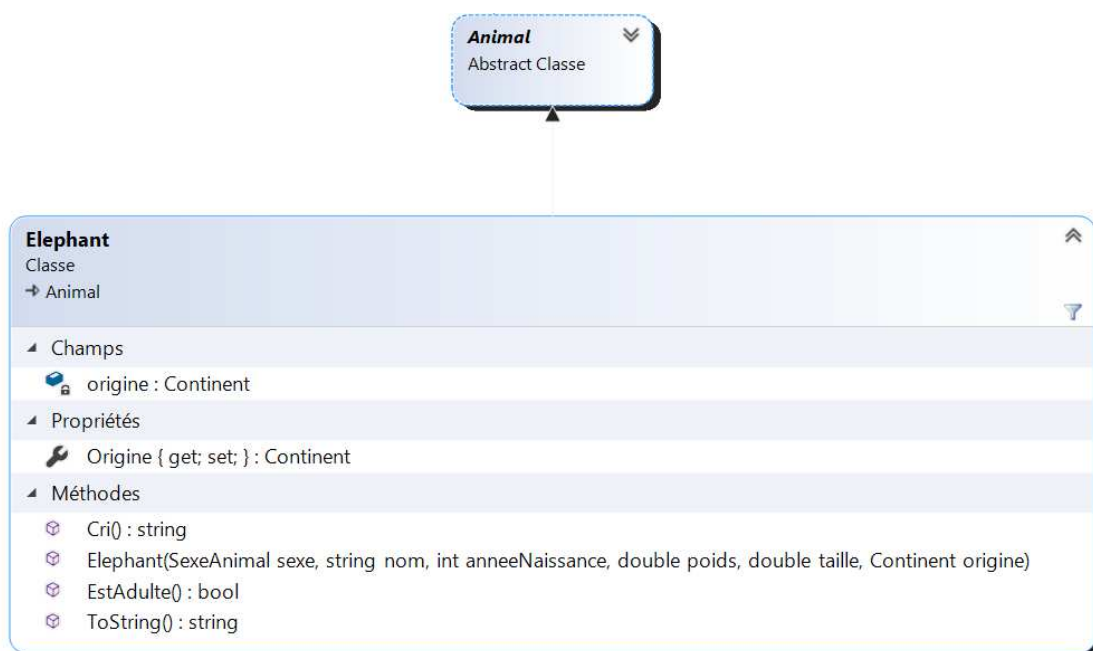
```
public const String UM_TAILLE = "m";
public const String UM_POIDS = "kg";
public const String UM_AGE = "an(s)";
```

- L'Age est une propriété calculée qui ne dispose que du get.
- L'année de naissance ne doit pas être supérieure à l'année actuelle.
- Le nom ne doit pas être nul, ne doit pas être vide et doit être mis en majuscule.
- Le poids et la taille ne doivent pas être négatifs.
- La méthode Equals : elle ne compare que le nom, car le nom identifie de manière unique l'animal.
- Les méthodes Cri() et EstAdulte() sont **abstraites**.
- La méthode ToString indique, en plus des valeurs propres à l'animal, s'il s'agit d'un adulte pour cela, elle s'appuie sur la méthode **EstAdulte**, même si elle est abstraite. Rem : vous ne pourrez tester le ToString qu'après avoir fait la classe instanciable Elephant ci-dessous !

```
Nom : DJUMBO
Sexe : Masculin
Annee de naissance : 1988
Age : 35 an(s)
Adulte : True
Poids : 200 kg
Taille : 2,54 m
```

## CLASSE ELEPHANT

2. Définissez la classe Elephant en suivant les consignes énoncées ci-dessous. Elle est **fil**le de la classe Animal. Elle n'est pas abstraite.



Ajoutez l'énum suivante :

```
public enum Continent { Asie , Afrique };
```

- ToString : elle doit faire appel au ToString hérité puis en plus indiquer :
  - o qu'il s'agit d'un éléphant
  - o son origine
  - o la taille des oreilles de l'éléphant et la présence ou non de défenses, sachant qu'un éléphant d'Afrique a de très grandes oreilles alors qu'un éléphant d'Asie a de petites oreilles. Les éléphantess (sexe féminin) d'Asie n'ont pas de défense.
- EstAdulte : retourne vrai s'il a plus de 13 ans (faites en une constante AGE\_ADULTE =13)
- Cri : retourne une chaîne correspondant au barrissement de l'éléphant, celle-ci doit donner la sensation d'un barrissement plus fort s'il s'agit d'un éléphant adulte.

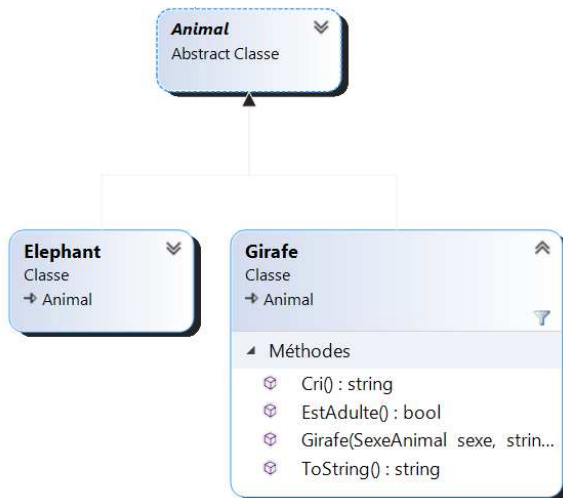
3. Au sein de la classe program, instanciez les 2 éléphants ci-dessous : 1 adulte mâle d'Afrique et 1 jeune femelle d'Asie, ajoutez les à une liste : List <Animal> : Affichez-les ainsi que leurs cris à l'aide d'une boucle.

```
Nom : DJUMBO
Sexe : Masculin
Annee de naissance : 1988
Age : 35 an(s)
Adulte : True
Poids : 200 kg
Taille : 2,54 m
Cri : HUUUUMMMMM
Eléphant d'Afrique avec de grandes oreilles et défenses
-----

Nom : ASIII
Sexe : Feminin
Annee de naissance : 2018
Age : 5 an(s)
Adulte : False
Poids : 150 kg
Taille : 1,7 m
Cri : huummmm
Eléphant d'Asie avec de petites oreilles sans défense
-----
```

## CLASSE GIRAFE

### 4. Définissez la classe Girafe :

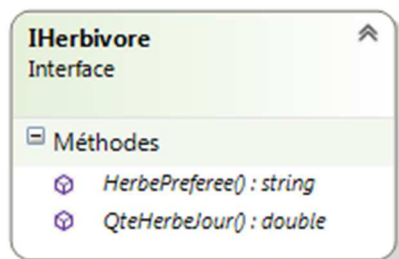


- ToString : elle doit en plus indiquer qu'il s'agit d'une girafe
- EstAdulte : une girafe est adulte à partir de 4 ans.
- Crie : retourne une chaîne correspondant au mugissement de la girafe, celle-ci doit donner la sensation d'un mugissement plus fort s'il s'agit d'une girafe non adulte. En effet, les adultes font moins de bruit.

### 5. Au sein de la classe program, ajoutez deux girafes à votre liste: Une femelle adulte et un mâle non adulte.

## INTERFACE IHERBIVORE

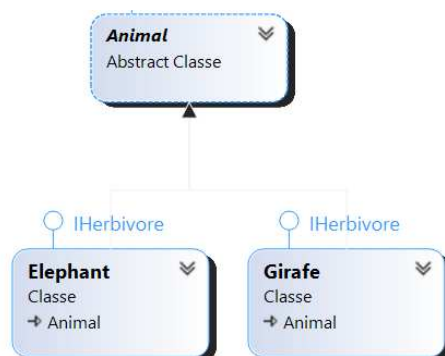
### 6. Définissez l'interface : IHerbivore . Elle contient uniquement la signature des méthodes.



```

public interface IHerbivore
{
    double QteHerbeJour();
    String HerbePreferee();
}
  
```

### 7. Implémentez l'interface : Indiquez que les classes Elephant et Girafe sont herbivores.



codez les méthodes :

- Un éléphant mange 25 % de son poids (un éléphant d'une tonne mange à peu près 250 kg d'herbes par jour). Les éléphants africains se nourrissent de feuillages et d'arbustes alors que les éléphants asiatiques préfèrent les herbes. Jusqu'à 5 ans, il est alimenté par le lait maternel.
- La girafe se nourrit essentiellement d'acacias, elle mange 5% de son poids. Jusqu'à 2 ans, elle est alimentée par le lait maternel.

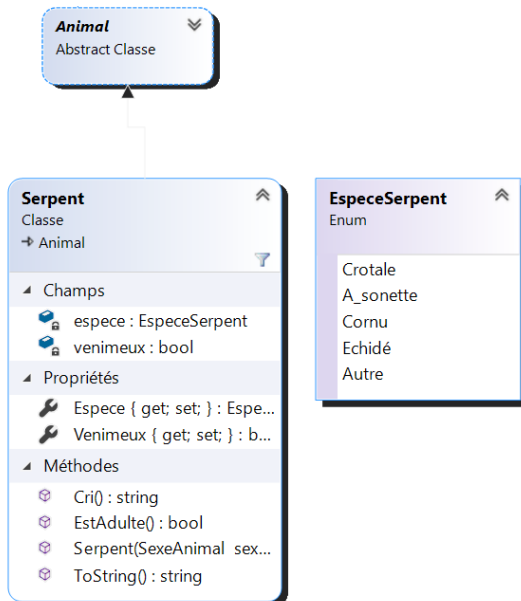
8. Transformez votre liste d « Animal » en liste d « lherbivore », de manière à afficher le repas journalier de chaque animal contenu dans la liste, pensez à indiquer son nom en face du repas.

## CLASSE SERPENT

9. Définissez la classe Serpent :

- enum :

```
public enum EspeceSerpent { Crotale, A_sonette, Cornu, Echidé, Autre }
```



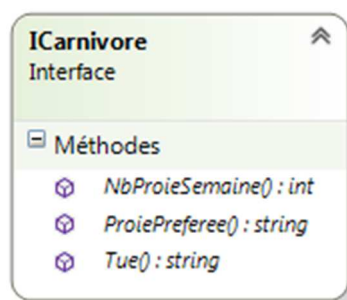
- ToString doit en plus indiquer qu'il s'agit d'un serpent et s'il est venimeux.

- EstAdulte : retourne vrai s'il a plus de 2 ans

- Cri : retourne une chaîne correspondant au bruit émis par le serpent. De nombreux serpents sifflent, mais certains produisent des sons plus insolites : comme le bruit de grelot que font les crotales ou serpents à sonnette. Le céraste cornu et les échidés ont sur les flancs des écailles denticulées, comme des morceaux de papier de verre, qu'ils frottent les uns contre les autres pour produire un crissement d'avertissement

## INTERFACE ICARNIVORE

10. Définissez l'interface : ICarnivore . Elle contient uniquement la signature des méthodes.



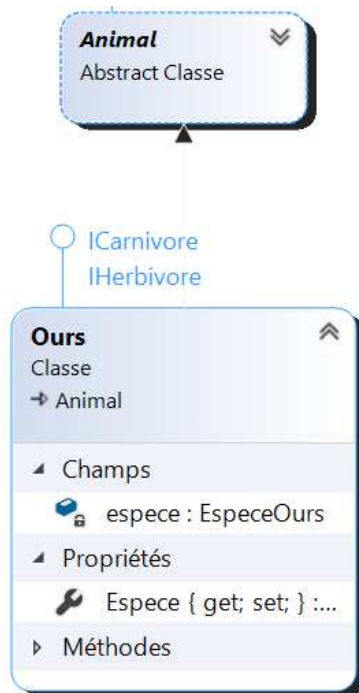
11. Implémentez l'interface : Indiquez que la classe serpent est Carnivore. Puis codez les méthodes :

- La nourriture des serpents dépend de leur âge :
  - À moins d'un an : 2 souris ou équivalents par semaine
  - À partir d'un an : 2 souris ou équivalents par semaine
  - À partir de 2 ans : 1 cochon d'inde ou équivalent par semaine
- Tue renvoie une chaîne décrivant le procédé employé par le carnivore pour tuer ses proies

**CLASSE OURS : IHERBIVORE ET ICARNIVORE**

12. Définissez la classe Ours : Un ours appartient à une espèce : Polaire, Brun, Noir.  
Vous définirez une énum.

Un ours est omnivore, il est à la fois carnivore et herbivore !



- ToString pour indiquer qu'il s'agit d'un ours et pour avoir l'espèce
- EstAdulte : retourne vrai s'il a plus de 4 ans pour l'ours brun, sinon 3 ans pour les autres espèces.
- Crie : retourne une chaîne correspondant au grognement d'un ours

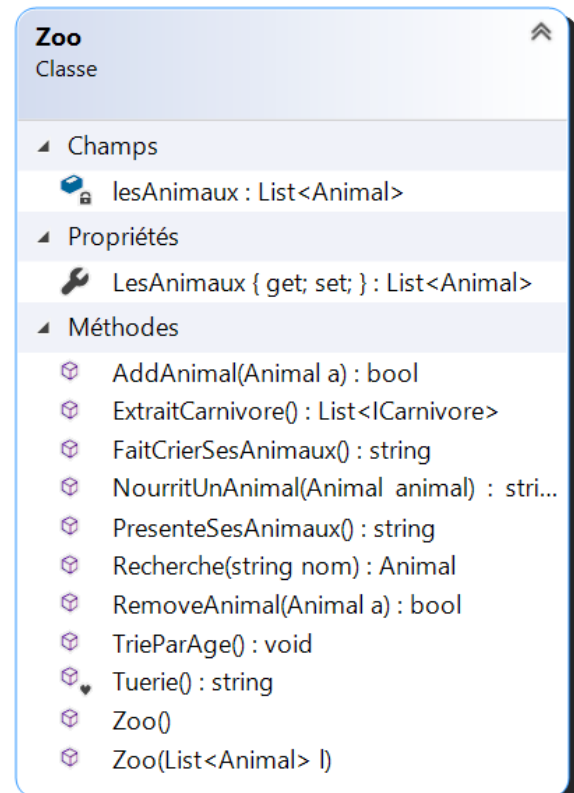
**CLASSE ZOO**

13. Codez de la classe Zoo : c'est une liste d'animaux. Faites les méthodes dans l'ordre de manière à coder les actions du menu au fur et à mesure :

```

C:\Windows\system32\cmd.exe

MENU ZOO
-----
0. Quitter
1. Presentation des animaux
2. Cri des animaux
3. Repas d'un animal
4. Tuerie de carnivore
5. Suppression d'un animal
6. Trier les animaux par age
-----
  
```



```

String choix;
do
{
    Console.WriteLine("-----");
    Console.WriteLine("MENU ZOO");
    Console.WriteLine("-----");
    Console.WriteLine("0. Quitter");
    Console.WriteLine("1. Presentation des animaux");
    Console.WriteLine("2. Cri des animaux");
    Console.WriteLine("3. Repas d'un animal");
    Console.WriteLine("4. Tuerie de carnivore");
    Console.WriteLine("5. Suppression d'un animal");
    Console.WriteLine("6. Trier les animaux par age");
    Console.WriteLine("-----");
    choix = Console.ReadLine();
    Console.WriteLine("-----");

    switch (choix)
    {
        case "0":
            Console.WriteLine("Le zoo vous souhaite une bonne journée");
            break;
        case "1":
            Console.WriteLine("LES HABITANTS DU ZOO");
            Console.WriteLine("-----");
            Console.WriteLine(monZoo.PresenteSesAnimaux());
            break;
        case "2":
            Console.WriteLine("CRIS");
            Console.WriteLine("-----");
            Console.WriteLine(monZoo.FaitCrierSesAnimaux());
            break;
        case "3":
            Console.WriteLine("REPAS");
            Console.WriteLine("-----");
            Console.WriteLine("Quel animal ?");
            String nom = Console.ReadLine();
            Animal a = monZoo.Recherche(nom);
            if (a == null)
                Console.WriteLine(nom + " n'existe pas.");
            break;
        case "4":
            Console.WriteLine("TUERIE DE CARNIVORE");
            Console.WriteLine("-----");
            break;
        case "5":
            Console.WriteLine("SUPPRESSION");
            Console.WriteLine("-----");
            Console.WriteLine("Quel animal ?");
            nom = Console.ReadLine();
            break;
        case "6":
            Console.WriteLine("DU PLUS JEUNE AU PLUS VIEUX");
            Console.WriteLine("-----");
            break;
        default:
            Console.WriteLine("Mauvais choix !");
            break;
    }
    Console.WriteLine("appuie sur une touche");
    Console.ReadLine();
    Console.Clear();
}
while (!choix.Equals("0"));

```