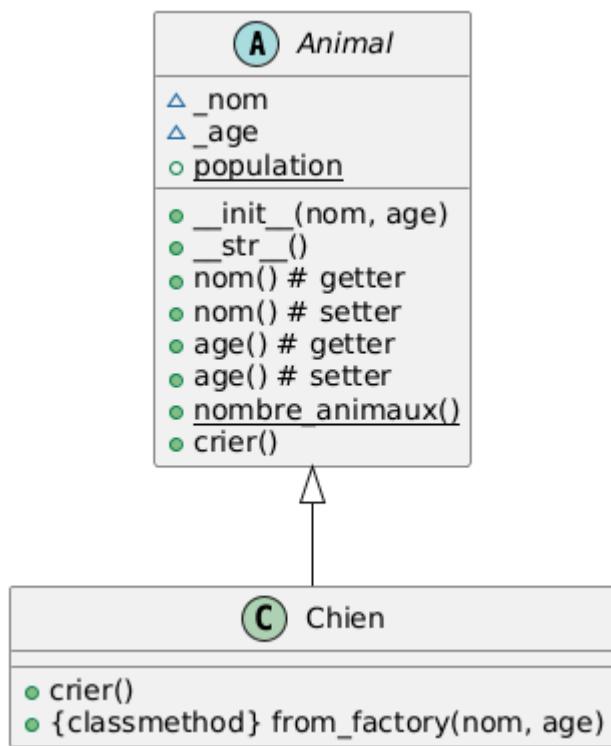


Objectif

L'objectif de cet exercice est de créer une classe `Animal` en Python et d'explorer divers concepts de la programmation orientée objet.



Étapes

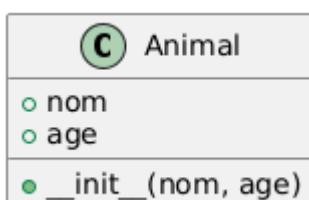
Étape 1 - Création de la classe

- Créez une classe `Animal`.



Étape 2 - Le constructeur

- Ajoutez un constructeur (`__init__`) à la classe `Animal`. Le constructeur doit prendre deux attributs : `nom` et `age`.



Étape 3 - Méthode

- Ajoutez une méthode appelée `crier` à la classe `Animal`. Cette méthode peut rester vide pour le moment.

C	Animal
•	nom
•	age
•	crier()
•	<u>__init__(nom, age)</u>

Étape 4 - Propriété statique

- joutez une propriété statique `population` à la classe `Animal`. Cette propriété doit être initialisée à zéro.

C	Animal
•	nom
•	age
•	<u>population</u>
•	<u>__init__(nom, age)</u>
•	crier()

Étape 5 - Méthode statique

- Ajoutez une méthode statique appelée `nombre_animaux` à la classe `Animal`. Cette méthode doit renvoyer le nombre total d'animaux créés.

C	Animal
•	nom
•	age
•	<u>population</u>
•	<u>__init__(nom, age)</u>
•	<u>nombre_animaux()</u>
•	crier()

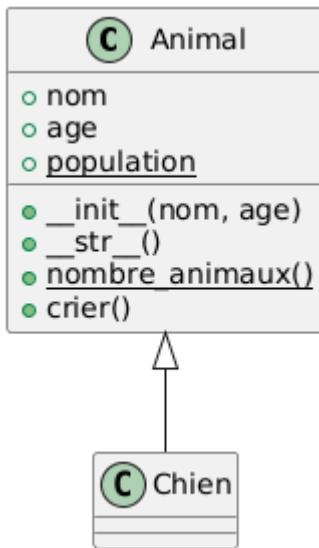
Étape 6 - Méthode magique

- Ajoutez la méthode magique `__str__` à la classe `Animal`. Cette méthode doit renvoyer une chaîne de caractères décrivant l'animal, par exemple, "Nom (Âge ans)".

C	Animal
•	nom
•	age
•	<u>population</u>
•	<u>__init__(nom, age)</u>
•	<u>str()</u>
•	<u>nombre_animaux()</u>
•	crier()

Étape 7 - Héritage

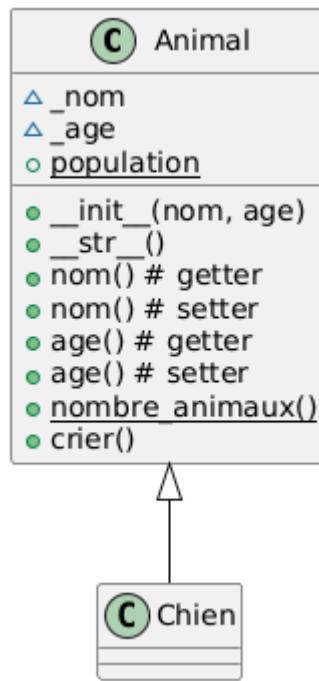
- Créez une sous-classe `Chien` de la classe `Animal`. La sous-classe doit hériter de la classe `Animal`.



Étape 8 - Encapsulation

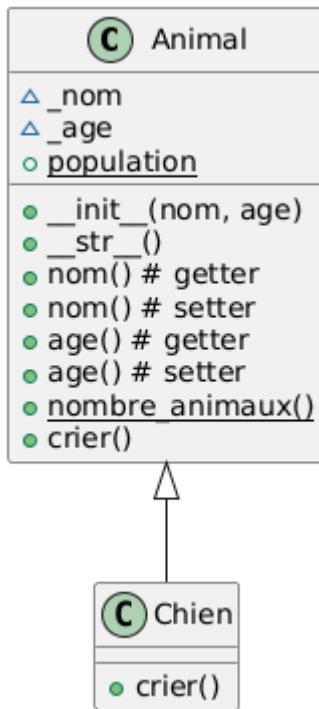
- Dans la classe **Animal**, modifiez un attribut en protected (commençant par `_`) pour le nom (`_nom`) et l'âge (`_age`).

Les attributs protected ne doivent pas être accessibles directement en dehors de la classe, mais accessible aux enfants de la classe. Donc implémentez des getters/setters.



Étape 9 - Surcharge

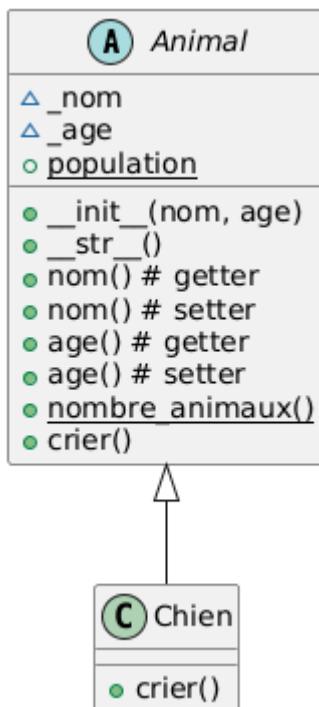
- Dans la classe **Chien**, surchargez la méthode `crier`. La méthode `crier` de la classe **Chien** doit renvoyer "Woof!".



Étape 10 - Abstraction

- La classe **Animal** est doit être abstraite, car la méthode **crier** est définie comme **pass**. Ajoutez les éléments pour le permettre.

Maintenant, Les sous-classes doivent la redéfinir pour devenir des classes concrètes.



Étape 11 - Méthode de classe

- Ajoutez une méthode de classe appelée **from_factory** à la classe **Chien** avec son nom qui a une longueur supérieure ou égal à 3, et son age supérieur à 0.

