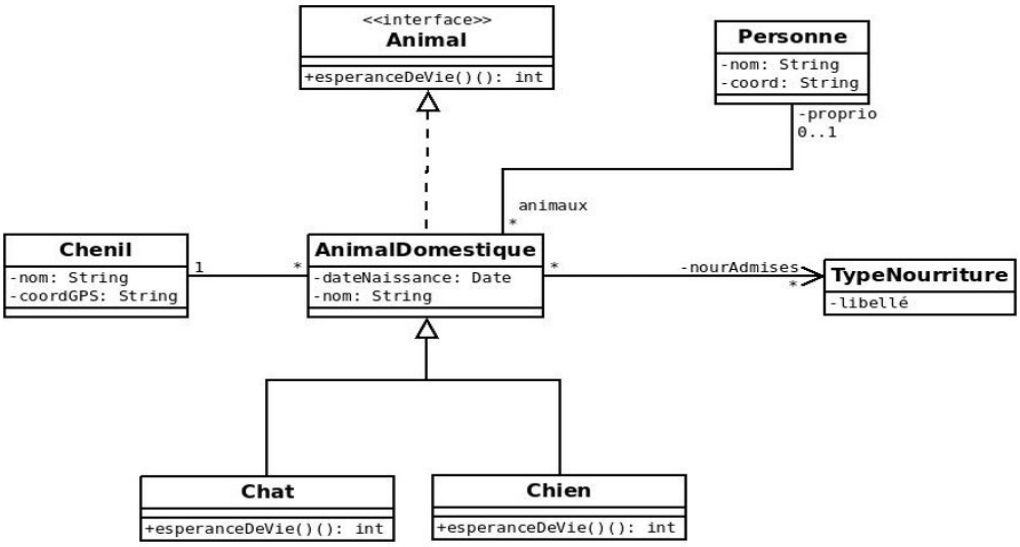
Correction-Controle-SIO-LDV



On suppose que tout objet dispose de getter/setter sur ses propres attributs

Un chenil héberge des animaux domestiques.
Règle A :Tout animal domestique est associé, au plus, à un propriétaire (nom + coordonnées), à un chenil et à une liste de nourritures admises.

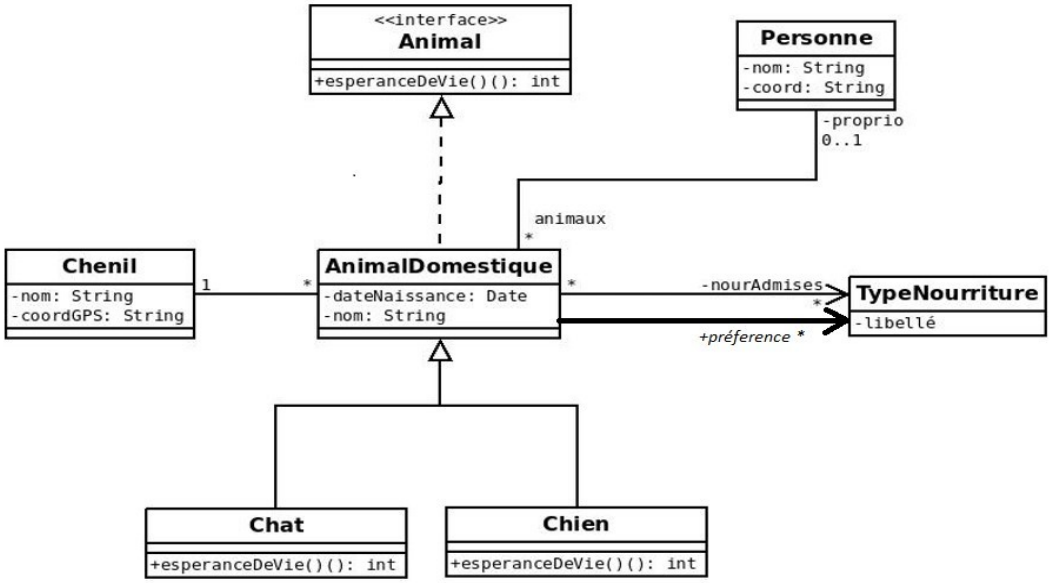
P-1 : Le diagramme est-il fidèle à la règle A ? Justifier

Oui car tout animal domestique est relié à une instance de Personne (proprio) et la multiplicité est de 0 à 1, à un chenil et une liste de types de nourriture (multiplicité *).

P-2 : On souhaite ajouter à la classe Personne la méthode isProprio() qui permettra de savoir si une personne est responsable (propriétaire) d'au moins un animal ou non. Donner le code de cette méthode

```
fonction isProprio() : boolean {
  return this.animaux.size() > 0 ;
 }
```

P-3 : On souhaite pouvoir enregistrer, pour un animal donné, ses types de nourriture préférés parmi lesquels il a droit. Faire évoluer le diagramme UML en conséquence.



On suppose que tout objet dispose de getter/setter sur ses propres attributs

Soit felix, le nom d'une variable qui référence à une instance de Chat, dans un programme qui exploite le modèce ci-dessus.

Felix-1 : Donner l'instruction qui permet de connaître le nom de (lobjet référencé par) felix.

felix.getNom();

Felix-2 : Définir une fonction nommé getNomProprio() dans AnimalDomestique, dans le but de connaître le nom de son éventuel propriétaire. Appliquez le à felix.

```
fonction getNomProprio() : String {
  if (this.proprio != null ){
     return this.proprio.getNom();
  }
  return "";
 }
```

Felix-3 :Donner l'instruction qui permet de connaître l'espérance de vie représentée par l'objet référencé par felix.

felix.esperenceDeVie()

Soit rex, le nom d'une variable qui référence une instance de Chien, dans un programme qui exploite le modèce ci-dessus.

Rex-1 :Donner l'instruction qui permet de connaître le nombre de types de nourriture admises pour (lobjet référencé par) rex

```
rex.getNourritureAdmises()
```

Rex-2 : Donner l'instruction qui affiche les types de nourriture qui lui sont permis.

```
for ( TypeNourriture n : rex.getNourritureAdmises() )
  print ( n.getLibellé() )
```

Rex-3 : Donner l'instruction qui permet de connaître son espérance de vie.

```
rex.esperenceDeVie()
```

Soit bob, le nom d'une variable qui référence une instance de Personne, dans un programme qui exploite toutjours le même modèce.

Bob-1 : Donner l'instruction qui affiche les animaux dont bob a la charge.

```
for (AnimalDomestique a : bob.getAnimaux() {
    print(a);
```

Bob-2 : Donner l'instruction qui affiche les chats de bob.

On pourra utiliser l'opérateur instanceof présent dans la plupart des langages.

Exemples :
String x = "coucou";

```
x instanceof String (rend true)
x instanceof Object (rend true, car Object est la classe mère de  String)
x instanceof Integer (rend false)
```

```
for (AnimalDomestique a : bob.getAnimaux() {
  if (a instanceof Chat)
    print (a);
```

Bonus: On souhaite pouvoir montrer aux propriétaires si leur animaux ont eu des maladies lors de leur séjour au chenil. Faire évoluer le diagramme UML.