

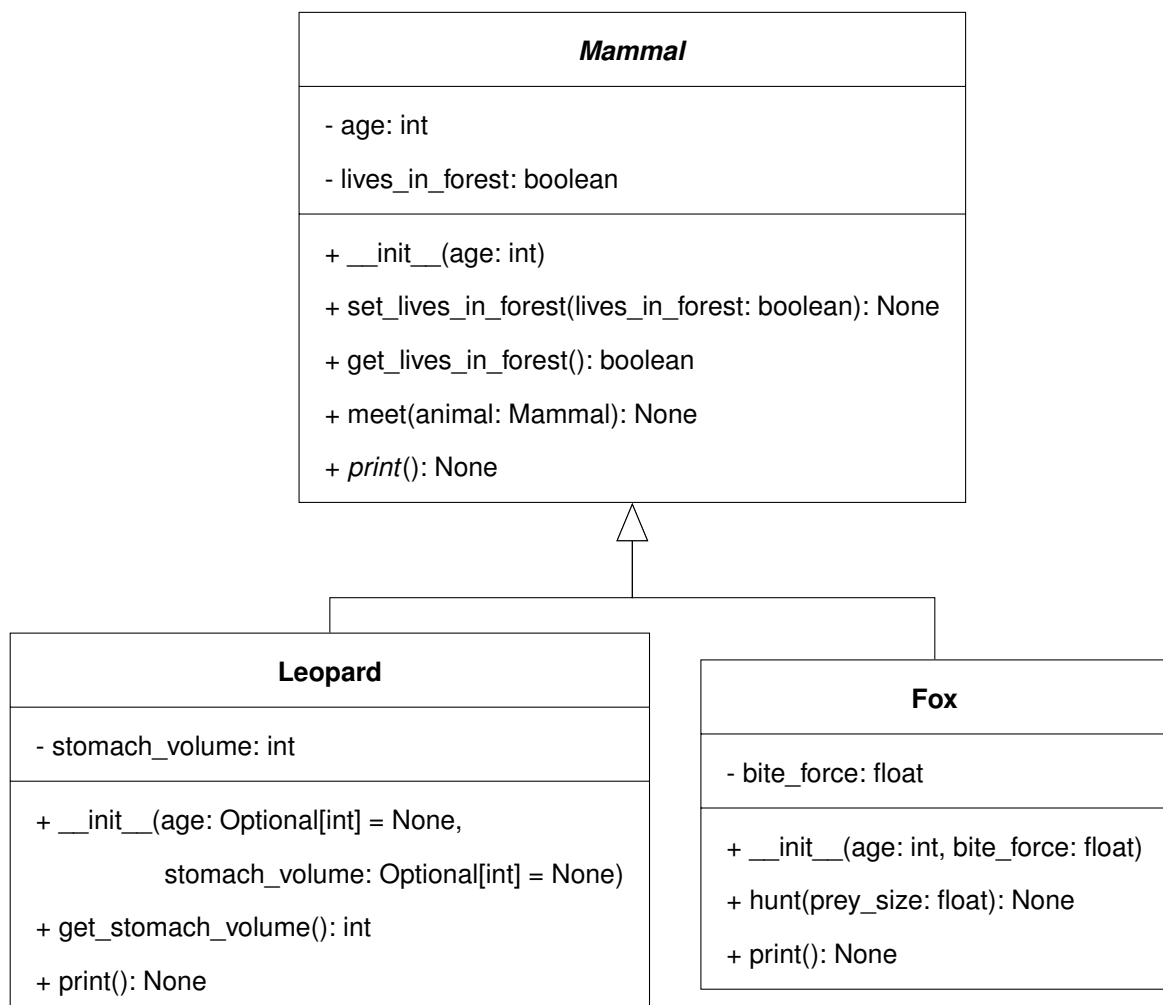
## Assignment for y0111166:

### Object Oriented Programming - Mammal

In this first assignment animals shall be modelled with the help of object-oriented programming. For this purpose you have to implement a superclass `Mammal` and two subclasses `Leopard` and `Fox`. The class structure is shown in the given UML diagram. Further information about the methods can be found within the respective tasks.

All values and properties are only used to adapt the task and do not have a realistic reference.

However, you are free to use units for output on the console.



**Abstract Superclass: Mammal**

Implement the superclass `Mammal` exactly as it is shown in the UML diagram. Use the following guidelines for orientation.

1. The constructor should assign the passed value to the attribute `age`.  
`lives_in_forest` should be set to `true` by default.
2. Create a get and set method for the attribute `lives_in_forest`.
3. The method `meet` should check whether two animals can meet each other. That is decided by comparing the attribute `lives_in_forest` of the different animals. Think about two useful text outputs for the cases that the animals can or cannot meet. The output should be a complete sentence on the console.

**Subclass: Leopard**

Implement the subclass exactly as it is shown in the UML diagram. Use the following instructions for orientation.

1. The constructor uses two optional arguments. They are used to set the attributes of the animal accordingly. In case they are not passed, a newborn leopard should be created. This is accomplished by passing the age 0 to the constructor of the superclass and by default, the stomach volume of a newborn should be set to 49.
2. For the class attribute only a `get()` method has to be created.
3. The implementation of the abstract `print()` method is intended to combine all attributes available for this animal in a console output formulated as a complete sentence.

**Note:** Make sure that you declare all data types correctly and consistent so a check with *mypy* won't return any erros. This is mandatory to pass this assignment.

**Subclass: Rat**

Implement the subclass exactly as it is shown in the UML diagram. Use the following guidelines for orientation.

1. There is only one standard constructor. The implementation is analogous to the one of the first animal. The same applies to the `print()` method.
2. The method `hunt` should inform on the console in one sentence if the fox can hunt a given prey. There are three possible cases:
  - a) Twice the prey is larger than the bite force. The hunt is not successful.
  - b) Twice the prey is smaller than the bite force. The hunt is successful and the fox gets something to eat.
  - c) If the prey is smaller than 5 it is too small and a hunt is not worthwhile for the fox.

**Playground**

Implement the following commands in a new python script where you import the previous created classes.

1. Create two objects of the class `Leopard`. The different constructors respectively. Pass the age 49, and for the class-specific attribute the value 357.
2. Instantiate an object of the class `Fox`, use the first two digits of your matriculation number for the age and the last two digits for the class-specific attribute.
3. Change the `lives_in_forest` attribute of one of the three animals using a set method.
4. Call the activity method with the object of the `Fox` class. The passed value may be freely chosen.
5. Place all three animals in one array. Use a loop to call the `print()` method of each animal.
6. Let the animal objects make two encounters by calling the `meet()` method.

**Note:** Make sure that you declare all data types correctly and consistent so a check with *mypy* won't return any erros. This is mandatory to pass this assignment.