

# Programming Languages 2

## Lesson 6

1. Create a new NetBeans Java project. The name of the project has to be "AnimalsProject". The name of the package has to be animals. Name the Main class of our project "Main".
2. Create a new class named `Animal` in the project. This class has the following attributes: name, numberOfLegs, eatingHabit (an enum typed variable that may take one of the following values: HERBIVORE, OMNIVORE, CARNIVORE – you have to add this enum to your project). All these properties have to be **hidden** from the outside.
3. Create getter and setter methods for the above attributes. For the name add both getter and setter methods, but for the other two provide **only getters**.
4. Overload the `setName(String name)` method by adding a new parameterless `setName()` method that sets the name of the `Animal` to "unknown".
5. Add a constructor **without parameters** to the class. This does not do anything else, but instantiates a new `Animal` object. **Add another constructor**. This one has 3 parameters with which the user can set all three properties of the newly instantiated `Animal` object.
6. Use a custom "NameTooLongException" to prohibit names longer than 100 characters to be set. (Both in the constructor and in the setter method).
7. Override the `toString` method of the class. If it is called, it returns a `String` like the following (for an animal named "Wolf", that has of course 4 legs: "Wolf (4 legs) – CARNIVORE!", or for one named "Chicken", (that has 2 legs): "Chicken (2 legs) – OMNIVORE"). Take care to return a `String` **exactly** in this format. (The "" characters are not printed out.)
8. Create a child class of the `Animal` class named `Fish`. `Fish` objects are special, because the value of `numberOfLegs` must be 0. Add only one constructor to this class that lets the user to set the name of the fish and the eating habit, but sets the number of legs to 0. There must not be any other constructors in the class. **Override the toString** method of this class as well so that it prints out only the name of the fish and the eating habit. (Like "Tuna – CARNIVORE"). Do not print out the number of legs.
9. In the `Main` class, in the main method instantiate a new `Animal` object with the following properties: name: "Chicken", 2 legs, OMNIVORE. Print out the object.
10. In the `Main` class, in the main method instantiate a new `Fish` object with the following properties: name: "Tuna", CARNIVORE. Print out the object.
11. Make the `Animal` class implement the `Comparable` interface. Add an `ArrayList` of `Animals` to the main method and add the two animals to the list.
12. Print out the animals sorted by their names using the `sort` method of the `ArrayList`.
13. In the `Main` class create a **static method** that gets two `Animal` objects as parameters and returns true if their names are the same. Else it returns false.
14. Instantiate a new `Animal` object in the main method and **use the above method** on the `Chicken` and this new object from the main method. If their names match print out "Same names" else print out "Different names";