PoP 10g - Animals

Carl Dybdahl, Patrick Hartvigsen, Emil Søderblom

December 22, 2016

We have written a model of animals in F♯. Figure 1 shows the UML diagram we designed for this task. We have also written tests for our classes and documented their behavior.

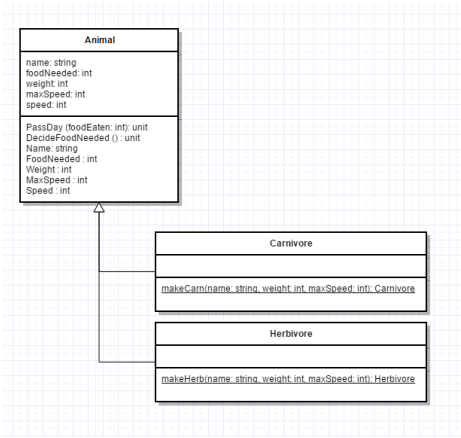

Figure 1: The UML diagram for the classes in this task.

1 Animal, Carnivore, Herbivore

We have a base class `Animal` which implements most of the behavior of animals. It has five attributes, `Name`, `Weight`, `FoodNeeded`, `MaxSpeed` and `Speed`. In addition, it has two methods, `PassDay` and `DecideFoodNeeded`. We have written XML-comments to describe the exact details of these methods and attributes, but they can roughly be described as follows:

- `Name` and `Speed` do not directly affect anything else.
- `PassDay` is given an amount of food eaten the current day and sets the `Speed` to a fraction of `MaxSpeed` based on how big a fraction of `FoodNeeded` is eaten.
- `DecideFoodNeeded` sets the food needed to a fraction of the animal's `Weight`.

A `Carnivore` is an `Animal` that needs less food, whereas a `Herbivore` is an animal that needs more food. Semantically, `Carnivore`s represent carnivorous animals, whereas `Herbivore`s represent herbivorous animals.

We made the constructor of `Carnivore` and `Herbivore` private so we could make sure `DecideFoodNeeded` was called on every animal as soon as it is constructed. To create an instance of these classes, one must use the static members `makeCarn` and `makeHerb`.

2 Race

We have written the code to do a race between `cheetah`, a `Carnivore` weighing 50 kg and having a max speed of 114 km/hour; `antelope`, a `Herbivore` weighing 50 kg and having a max speed of 95 km/hour; and `wildebeest`, a `Herbivore` weighing 200 kg and having a max speed of 80 km/hour.

The winner is declared to be the one who is the wins the most time in a test of three races on separate days, where each day the animals eat a random fraction of their `FoodNeeded`. If there is a draw, the race is retried.

3 Documentation

The code was documented according to the standard. In order to keep the documentation comments from cluttering up the code, we placed the documentation in the signature file `10g.fsi`.

4 Tests

We have tested all the methods and have gotten positive test results.