

Joppes Animal family v2

Project-idea

To write a program that:

- Shows the relations of classes and inheritance, we will do that with Joppes animal family v2.
- Creates a pet owner Joppe with age and one or more pets (cats and dogs)
- Joppe will have the ability to :
 - List the pets
 - play fetch with them with a ball
 - check the ball tearing
 - feed them
- All different pets created will inherit from the Animal class
- Joppe can play fetch with the pets with a ball
- Add class for Norwegian forest cat that inherit from the Cat class that inherits from Animal class. Norwegian forest cat but has his own behavior.

Challenges :

I believe that the biggest challenge is to get for example the Kittie class to work properly, as it inherit from call Cat that inherits from Animal, and do the overrides.

I believe I need to get a proper understanding about inheritance, overrides, and virtual/abstract modifiers.

Strategy:

I will start to make the PetOwner class with the fields and one or two methods.

Do the same for the Animal class, Just make it as simple I can to get it to work, then write on pet, probably the Dog class, and make everything work.

Afterward I will implement the rest.

Planning

- ✓ Classes
- ✓ Methods
- ✓ Class diagram
- ✓ Implementation
- ✓ Testing and debugging
- ✓ Review

Classes

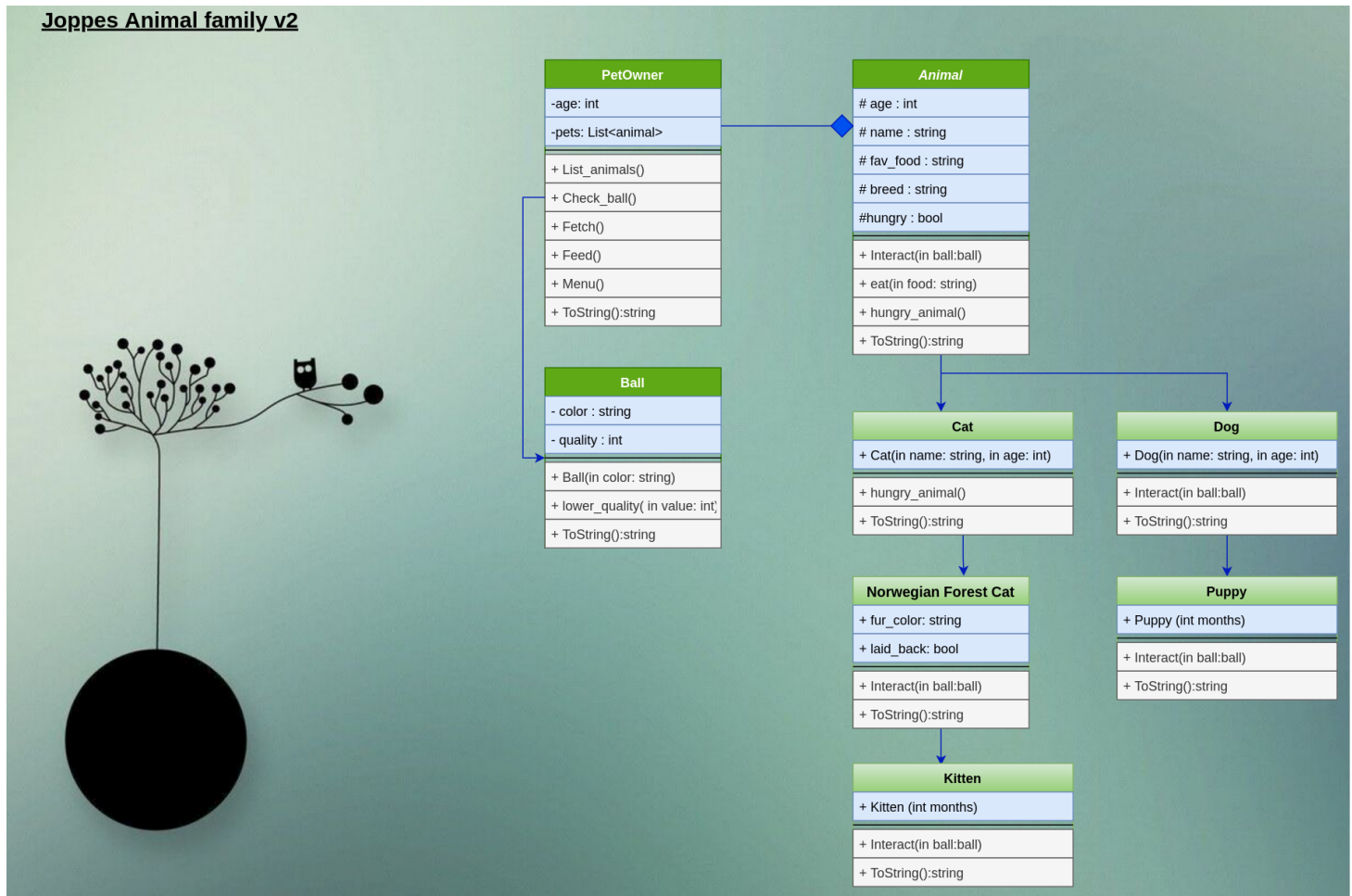
PetOwner
Animal
Cat
NorwegianForestCat
Kittie
Puppy

Methods

- PetOwner
 - List_animal()
 - Check_ball()
 - Fetch()
 - Feed()
 - Menu()
 - toString()
- Animal
 - Interact(in ball)
 - eat(in food)
 - hungry_animal()
 - ToString()
- Cat
 - hungry_animal()
 - toString()
- NorwegianForestCat
 - Interact(in ball)
 - ToString()
- Kitten
 - Interact(in boll)
 - ToString()
- Dog
 - Interac(in ball)
 - ToString()
 - Puppy

- Interact(in boll)
- ToString()

Class Diagram



Implementation

See the cs code file

Review

What went well

To write the code was actually easier that I have anticipated, because of the planning phase. I had already a schema for what kind of classes, methods, fields, variables etc that was needed. I just had to look at them and code accordingly.

What went less well:

To think about the classes and make the UML. I did find it in the beginning like a waste of time. I felt that I spend most of my time thinking about how to write the UML, and plan how it was going to work

together. I have of course almost all the UML given by the assignment. But to write it again and then think about the class for get score A.

I just wanted to start code right away, and find the solutions while I wrote. And if I encounter any problems I would fix them in the spot.

All this felt like overkill. Until it didn't. When I started to write the code following the UML diagram, it was absolutely fantastic. It went very fast and with the overview of the diagram I had total control of what that i was doing.

How well does the program compare to the initial plan

I find that the program fill all the functionality that was planned. I had some additional like getBreed() and animalName() in the animal class.

I added to the petowner class the ability to add new pets.

I have added also to the Check_ball() class a little more variance of the deterioration.

Program defects:

For one, the user has to follow the programs instructions. In other words I did not write error checking if the user decided to do something that is not meant to do.

I did this in the dart program too. Is not that I have forgotten. I know this is lacking in my program, and it will be in my to do list to fix. I believe I want to do a crash course on only error handling to be able to write in a more standard way.

How to scale the program and make it better:

To begging with get a great error handling, then have more functionality, the ability to add other kinds of animals, like fish, snake, rabbits etc. Add more personality to each individual pet. Add maybe functionality to get weaker if not get food after some time pass, have ability to train your pet to do tricks, and the more it trains then get better at it, in form of levels gain by practice.

Testing and debugging

problem: 1:

To test my program I wrote the names of the pets in lower letters, the program did nothing. If I wrote the names in capital letters then the program did nothing.

solution:

Make sure that if the user write in capital or lower letters, it will always convert the string to lowercase and the name for compare.

problem 2:

when I called the interact(ball) for any pet i always got the same output.

solution:

I had to remove the body from the animal class and make it abstract.

problem 3:

When i made a new Kitten, i had to always write name, age(in years), fur color and months.
solution:

I passed the values for age to the base

```
public Kitten(string name, int months, string Fur_color) : base(name, 0, Fur_color)
{
    this.months = months;
    this.breed = "Norwegian Forest kittie cat";
}
```

Does the program meed the expected behavior?

Yes it does. The program does exactly what is requested plus some additions. and some defects/vulnerabilities. Please see above explanations.

Expect for the error handling that was not requested in the assignment. I am content with my project. Is it ready for deployment? of course not, I would never release a program that does not have error handling.