OOP homework

Homework Assignment: Creating a Zoo Management System

Instructions:

You are tasked with building a simple Zoo Management System using Object-Oriented Programming in Python. The system should be able to manage different types of animals in the zoo. You need to create classes for the following entities:

1. `Animal`: The base class for all animals. It should have the following attributes:
   - `name` (str): The name of the animal.
   - `species` (str): The species of the animal.
2. `Mammal`: A subclass of `Animal` for mammals. It should have an additional attribute:
   - `fur_color` (str): The color of the mammal's fur.
3. `Bird`: A subclass of `Animal` for birds. It should have an additional attribute:
   - `wing_span` (float): The wingspan of the bird.
4. `Reptile`: A subclass of `Animal` for reptiles. It should have an additional attribute:
   - `scale_type` (str): The type of scales the reptile has.
5. `Zoo`: A class to manage the animals in the zoo. It should have the following methods:
   - `add_animal(animal)`: Add an animal to the zoo.
   - `list_animals()`: List all the animals in the zoo.
   - `get_animals_by_species(species)`: List animals of a specific species in the zoo.
   - `remove_animal(name)`: Remove an animal from the zoo by its name.
   - `count_animals()`: Return the total number of animals in the zoo.

Requirements:

1. Create the necessary classes (`Animal`, `Mammal`, `Bird`, `Reptile`, and `Zoo`) with their attributes and methods.

2. Implement proper encapsulation by making attributes private and providing appropriate getters and setters.
3. Write a `main` program that demonstrates the functionality of the Zoo Management System. Create some animals of different types and perform operations such as adding, listing, and removing animals from the zoo.
4. Ensure that your code is well-documented, and you include comments explaining the purpose of classes, methods, and any important code blocks.

Bonus Challenge (Optional):

Create a method `feed_animals()` in the `Zoo` class that simulates feeding all the animals in the zoo. Each type of animal may have a different feeding mechanism, and you can implement this method to showcase polymorphism.

Grading:

- Correct implementation of classes and methods.
- Proper encapsulation and use of getter and setter methods.
- Demonstrated functionality in the main program.
- Code quality and documentation.