Inheritance, Overriding and Polymorphism

GRA-4152 OOP with Python
Autumn 2023

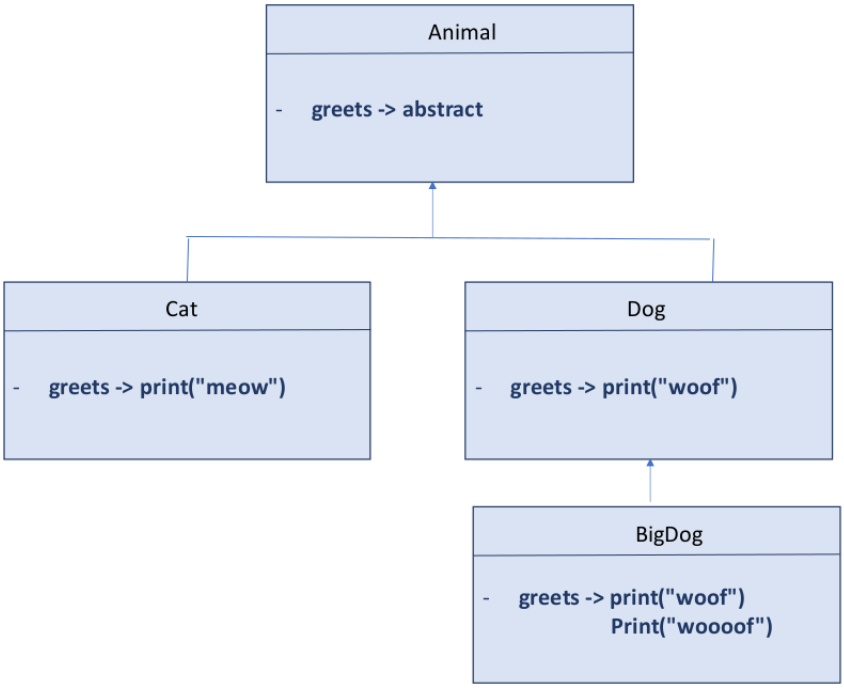
Honor Code: By answering this exercise, I confirm that I will not give or receive any help, including the use of AI technology. Any suspicion of cheating will be reported immediately to the exam administration. Students may also be called for an oral consultation as an additional verification.

Instructions

Create a new folder called `second_programming_ex`, save your Python and text files there, and push the folder by **November 10 at 09:00** to your GitHub repository. You are not allowed to use notebooks (colab, jupyter, etc.).

1 Animal Hierarchy

Code the following hierarchy of animals:



You are free to include more methods in addition to the `greets` method, which contains only the `print` functions shown in the diagram above. Make sure to include the following:

1. `docstring` for the superclass and all subclasses
2. An explanation of how you use the following concepts in your answer: `inheritance`, `overriding`, and `polymorphisim`.

Programming Exercise: Business P10.24

Read the exercise carefully. Note that this exercise starts with the *Business P10.22* exercise, in which you are asked to create a superclass `Appointment` and three subclasses `Daily`, `Monthly`, and `Onetime`. The exercise then continues with the *Business P10.23* exercise where you are asked to improve your previous code. However, I cannot see that the book asks for new methods, so you do not need to worry about exercise 23. Apart from the instructions in the book, make sure to follow these steps:

1. Provide a `docstring` for the superclass and all subclasses
2. Use `inheritance` in the `save` method
3. Use `overriding` in the `occursOn` method. Explain how you are further developing the method from the superclass in each of the subclasses. Recall to use the appropriate syntax for method calls from a superclass.
4. Use `overriding` in the in-build `__rep__` method for each of the subclasses in such a way that the string representation of an object is something like this:

```
# pseudo-code
#
'One time appointment {} {} {} {}'.format(day,month,year,
    description)
# or
'Daily time appointment starting {} {} {} {}'.format(day,month,year
    ,description)
# or
'Monthly time appointment starting {} {} {} {}'.format(day,month,
    year,description)
```

5. Explain how could you code a polymorphic approach for the `save` method and how such an approach differs from the approach in the above question number 2.