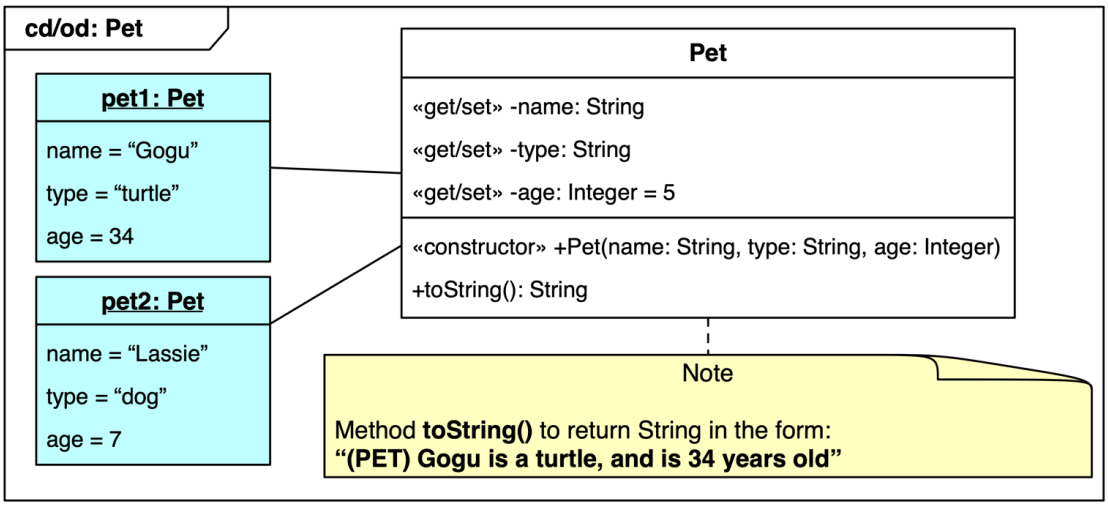
OO Analysis and Design

Lab 02

OOAD - Week 2 Java recap
Exercises

Exercise 1- (7.5 Book)
class with constructor, accessor methods and `toString()` method



Class-object (cd/od) diagram for class `Pet`

Class `Pet`

Declare a class called `Pet` that encapsulates the following 3 attributes: `name`, `type`, and `age`.

Ensure that you apply the principle of **data hiding** (`private` attributes accessed through `public` getters/setters).

- set the default value of `age` to 5
- declare a constructor that initialises the 3 fields to appropriate custom values.
- declare setter methods for each of the 3 fields.
- declare getter methods for each of the 3 fields.
- In the above class also override the method `toString()` to return the string representation of a `Pet` instance in the form:

(PET) <name> is a <type>, and is <age> years old.

Class `Main`

Declare a second class called `Main`, and in it add a `main()` method to do the following:

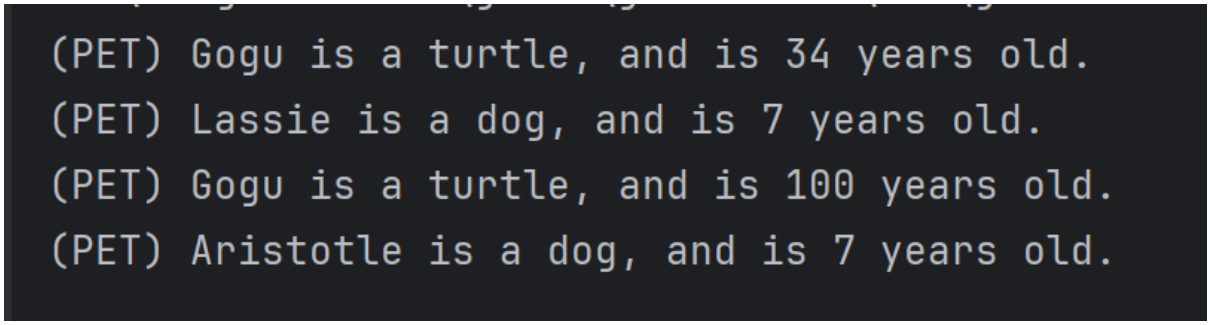
- Create 2 instances of `Pet` using the following values:
    - `Gogu`, `turtle`, `34`
    - `Lassie`, `dog`, `7`

-Print out the details of each Pet object

`- making use of its `toString()` method without having to write `toString()`...`

- Use the appropriate set methods to change
    - the value of age for the first Pet instance to 100
    - the value of name for the second Pet instance to Aristotle
- Then print out these changed values inside the two objects, but this time using the appropriate `get` methods.

OUTPUT



Exercise 2 (7.6 Book) - setters with validation logic

Using the principles of encapsulation and information hiding, create a class Book that contains the following:

- 2 instance variables corresponding to
    - title (e.g. Casino Royale, I am Pilgrim, Java in 1 semester, etc.)
    - price (e.g., 23.45, 12.3, 7.50, etc.).
- A constructor that allows the 2 instance variables to be initialised to custom values
    - If an empty title is passed, the title instance variable must be defaulted to a value of your choice, eg “no title set”
    - if a negative price is passed, the price instance variable must be defaulted to 7.7
    - otherwise, the instance variables should be initialised to the values of the corresponding parameters.
- 2 set methods that allow each of the instance variables to be modified to a custom value.
    - In the case of the title setter, if an empty title is passed, the title instance variable must be defaulted to a value of your choice, otherwise, it should allow that variable to be changed
    - In the case of the price setter, if a negative price is passed, the price instance variable must be defaulted to 7.7, otherwise, it should allow that variable to be changed.
- 2 get methods that allow you to access each of the instance variables.
- Override the method toString() to return the string representation of a Book object

Declare a second class called Main, and in it add a main method to do the following:

- Create 2 instances of Book
    - one provides valid initial values for the object:
        - title: The Restaurant at the end of the Universe
        - price: 9.99
    - another provides invalid values (so the default values should be set)
        - title: “”
        - price -2.00
- Print out the details of each object
    - making use of its `toString()` method without having to write `toString()`…)

Output

```
Book{title='The restaurant at the end of the universe', price=9.99}
Book{title='(no title set)', price=7.7}
```

Exercise 3 - create a class-object diagram for Pet exercise (Book 7.7)

A class/object (cd/od) diagram was provided for the above Pet exercise. Try to draw your own class/object diagram for the Book exercise (7.6) above.