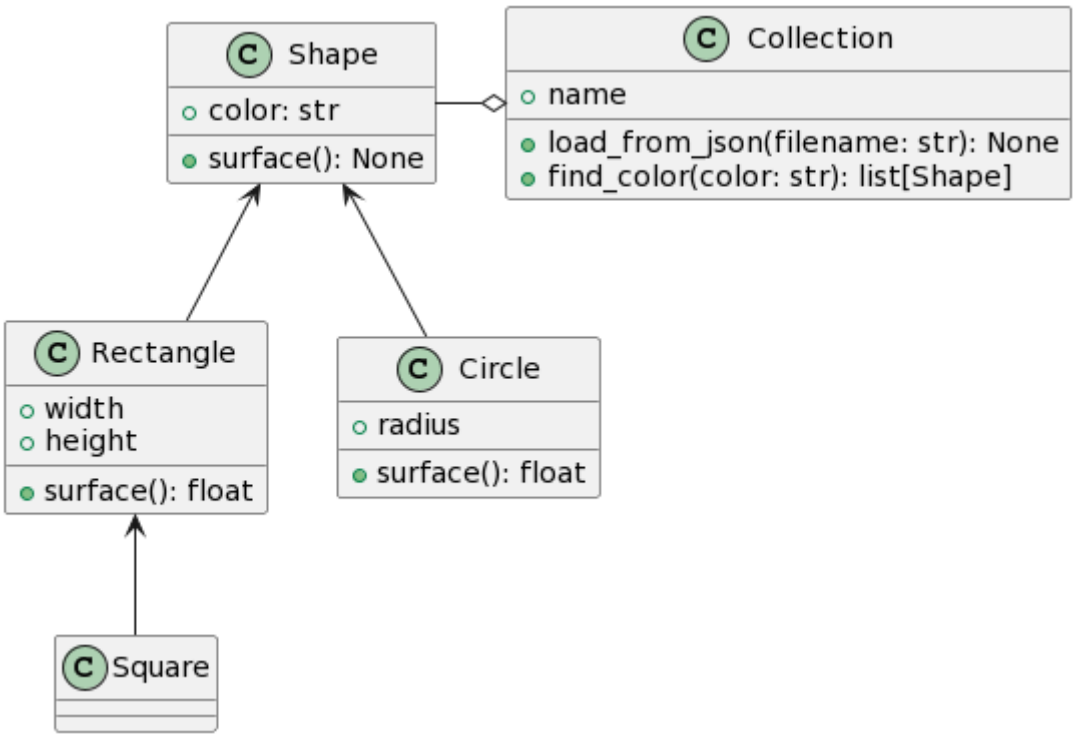
Midterm exam -- ACIT2515

Inheritance and advanced OOP

In this assignment, you will create 5 classes.



> It is strongly recommended to get the shape classes done and tested before implementing the `Collection`. Run `pytest test_shapes.py` to only run tests for these classes.

Shapes

- Read the tests in the `test_shapes.py` file.
- You have to create 4 classes.
- The `Shape` class takes a color (= a string) as argument
  - If the color provided is not a string, raise a `ValueError`
- the `Circle` class takes a radius (= a number) and a color
- The `Rectangle` class takes 2 numbers (width and height) and a color
- The `Square` class takes 1 number (= size) and a color

Hints:

- No parameter validation of the shape dimensions is required for `Circle`, `Rectangle`, `Square`.
- You should only define `__str__` once.
- You can get the surface of a shape by calling the `surface()` method.
  - The surface of a rectangle is `width * height`

- The surface of a circle is `π * radius²` (π is available as `math.pi` in Python)
- A generic `Shape` does not have a surface, and the method must raise an exception
- All surfaces are returned as `float`, with 2 decimals precision (use the Python builtin function `round`)

READ THE TESTS!

9 tests = 9 marks

Collection

A collection contains shapes.

- Its constructor takes an argument: the name of the collection.
  - the name must be a string, otherwise raise a `TypeError` exception
  - the name is available as the instance attribute `name`
- The collection maintains a list of all its shapes in the `shapes` attribute.

`load_from_json`

A collection can load shapes from a JSON file: `instance.load_from_json("example.json")` will load the information located in the `example.json` file. The JSON contains a list of shapes, each with the following format:

```
{
    "type": "rectangle",
    "dimensions": [2, 100],
    "color": "red"
}
```

This defines a `Rectangle` shape, with width == 2, height == 100 and color == "red". Your code must work with the shape types "rectangle", "square" and "circle". If another shape type is found in the JSON, ignore it.

> Look at the structure of the JSON file provided (and/or the tests) to implement this method!

`surface`

This method calculates the total surface occupied by all the shapes in the collection.

`find_color`

This method takes a string as argument, and **returns** a list of shapes that match the given color. `instance.find_color("blue")` will return a list of all the *blue* shapes in the `instance` collection. The list of shapes must be sorted by surface descending: the largest shape must be first in the list (and the smallest the last).

READ THE TESTS!

8 marks = 7 tests (1 mark per test) + 1 mark for usage of unpacking

Submission

Submit your files to D2L.

Total marks: 17