Shapes and Figures

This exercice focuses on structural design pattern training

Preamble

A *shape* can be **drawn at a given position**.

Circle and **Rectangle** are shapes, with their own properties :

- a circle is defined by its *radius*
- a rectangle is defined by its *width and height*

Produce a **relevant ULM class diagram**, **implement it** and **write an application that creates several shapes and draws them**. *To simplify, consider that shape drawing only results in logging in System.out the shape being drawn (and at which position), as well as its properties.*

Decorator

Shapes can have several enhancements, such as:

- a **border**, defined by its thickness
- a **shadow**, defined by its angle and intensity
- a **solid fill**, defined by its color,
- ... (enhancements are not all enumerated)

Use the *decorator design pattern* to **attach enhancements to shapes**. **Write an application that creates some shapes, with some enhancements**, and draws them.

Builder

Use the *builder design pattern* to simplify the way clients (here the application) **create enhanced shapes**, and particularly avoiding them to explicitly handle decorators.

Composite

A *figure* can be either:

- a **simple figure**, defined by the association of a shape (with or without enhancements) and the position where it is supposed to be placed
- a **complex figure**, defined by the combination of two figures with a constructive geometry operator (union, intersection, substraction)

Figures, as shapes, can be drawn.

Use the *composite design pattern* to implement figures, a **write an application that creates a (very) complex figure and draws it**.