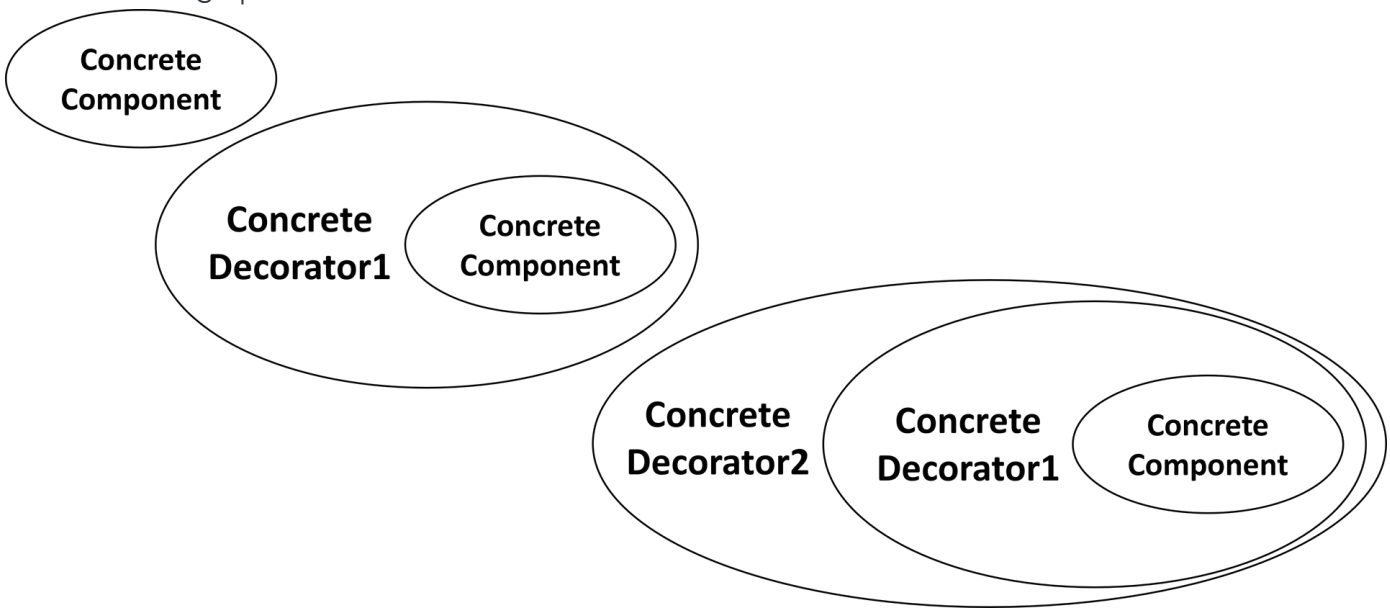
8. Decorator

Definition

Structural design patterns.



> **Definition:** Combines multiple subclasses to create multiple behaviours instead of adding a subclass for each behaviour

The Decorator Pattern attaches additional responsibilities to an object dynamically.
Decorators provide a flexible alternative to subclassing for extending functionality.

Motivation

Avoiding subclass explosion!

This breaks with SRP.

How to?

By using composition, the decorator pattern makes it possible to add any number of behaviour to a concrete component during runtime

```
Tesla Tesla = new MY();
Tesla = new color(Tesla);
Tesla = new premium(Tesla,12);
```

Pros & cons

Pros	Cons
Flexibility - low coupling	No rules
OCP & SRP	Increased complexity
Simplifies code	Performance

Solid

S Each decorator has its own resposibility.
O Easy to add new decorator subclasses without changing existing code.
L
I
D High level class 'Tesla' and low level class 'Decorator' depend on abstractions.

Comparison

Structural	Behaviour
Organisation of object structures	Interaction between objects
New behaviour / functionality added by adding an object	Using behaviour from another object
Decorator	Strategy and template

Similarities and differences

Template method pattern

- receive behaviour from subclasses
- changing an algorithm
- compile time

Strategy pattern

- runtime

- change behaviour on object