



DECORATOR

Nelson Gomes Neto

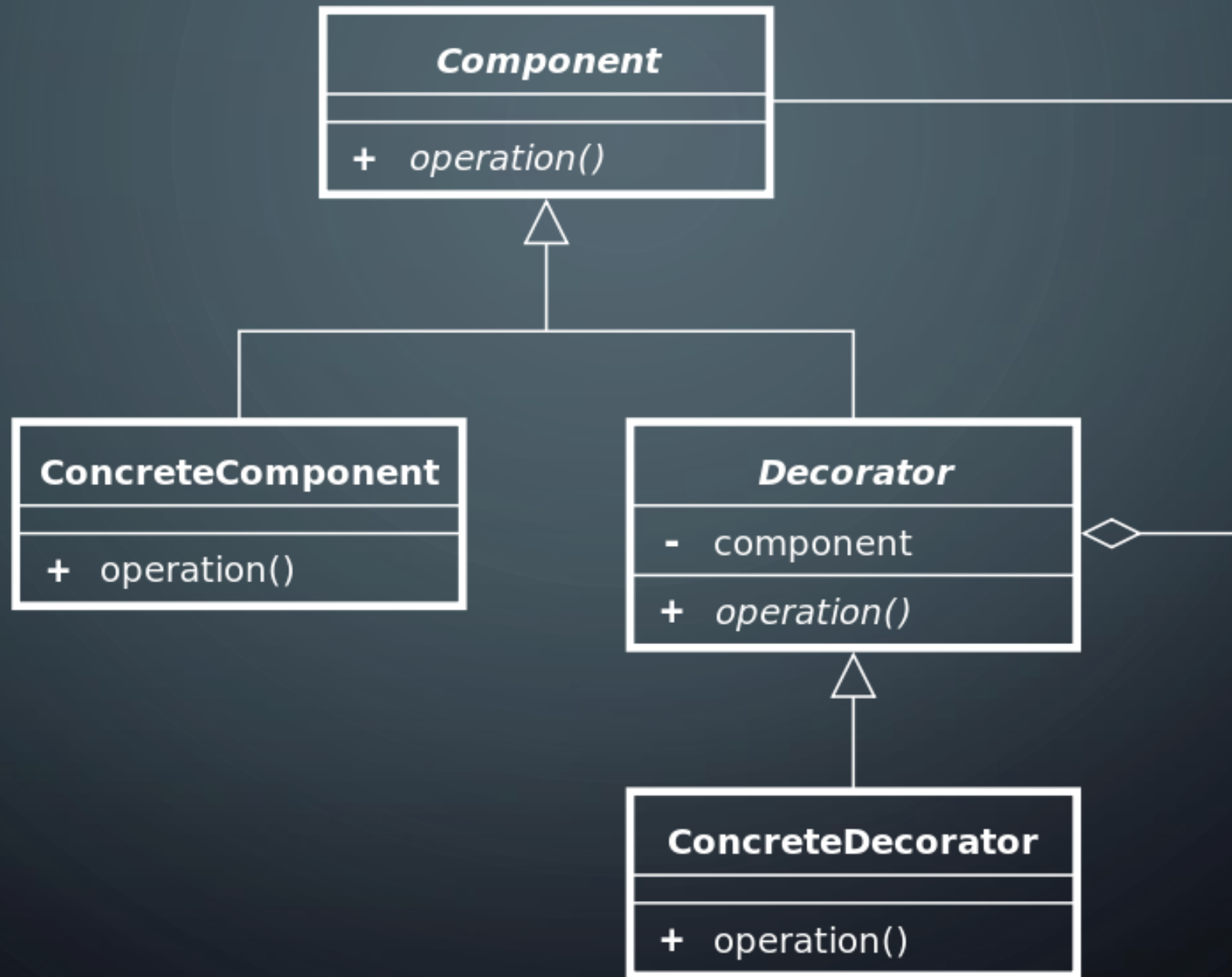
ngn@ic.ufal.br

PROBLEM

- Needs to add functionalities to objects;
- Subclasses explosion;
- Flexibility;

SOLUTION

1. Create an *Component* interface;
2. Create two classes (*ConcreteComponent* and *Decorator*) implementing *Component* interface;
3. *ConcreteComponent* implements a basic component;
4. Add a *Component* “pointer” in the *Decorator*;
5. Pass a *Component* to the *Decorator* constructor;
6. Create a subclasse *ConcreteDecorator* inheriting *Decorator*;
7. Override *Component*’s methods to modify behavior as you need.



DRAWBACKS

- Too many small classes may lead to a maintenance overhead;
- Adds complexity;
- Objects loses identity;

TARGET CODE

- <https://github.com/naryad/CodeSnippets/blob/master/src/yn/designpatterns/DecoratorPattern.java>