

Similarity

Intent

Both decouple abstraction from its implementation.

We can vary implementation independently. So it can not affect to client.

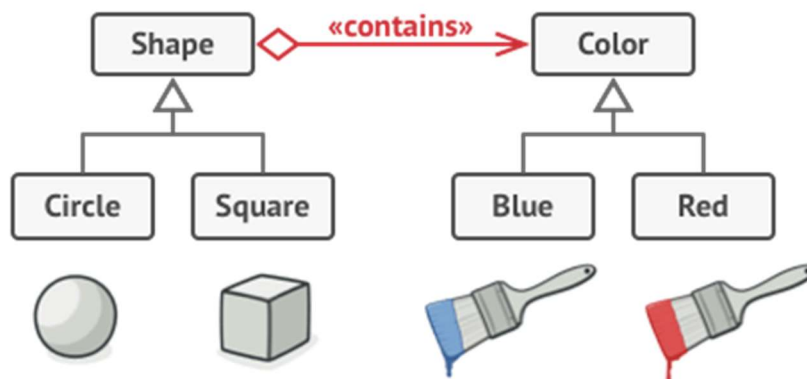
UML

Both compose interface of implementations.

In bridge, abstraction composes implementor.

In strategy, context composes strategy.

- a. Bridge pattern decouples implementation class and abstract class by providing a bridge structure between them. Concrete classes and interface implementor classes can be altered structurally without affecting each other.
In Strategy pattern, we create objects which represent various strategies and a context object whose behavior varies as per its strategy object.
- b. In bridge pattern, if we want to add one or more features to an specific object this pattern is right choice.



The Strategy pattern suggests that you take a class that does something specific in a lot of different ways and extract all of these algorithms into separate classes called strategies.

