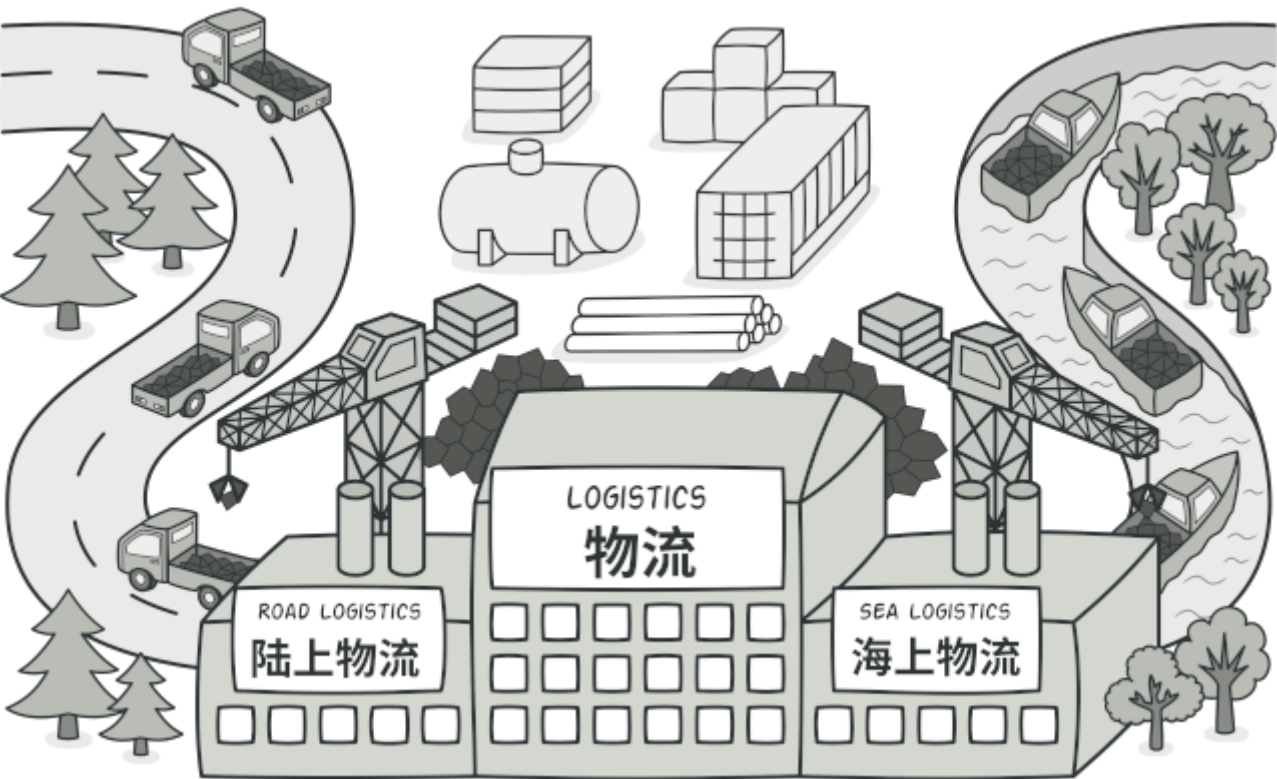
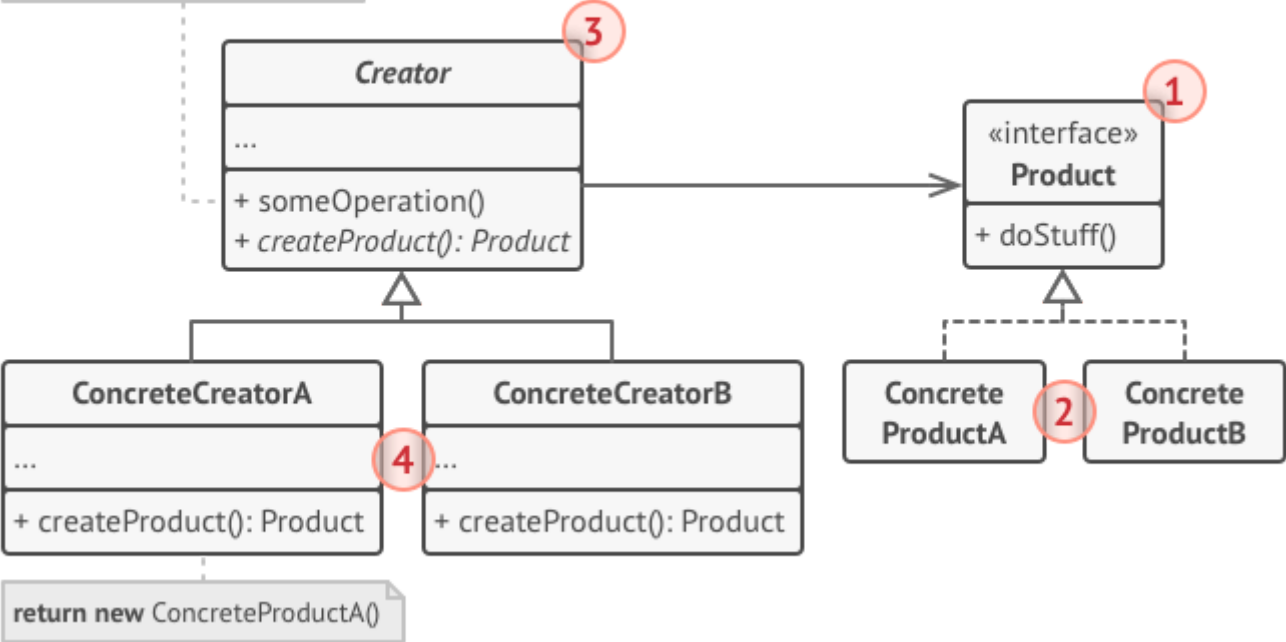


1. Creational Patterns

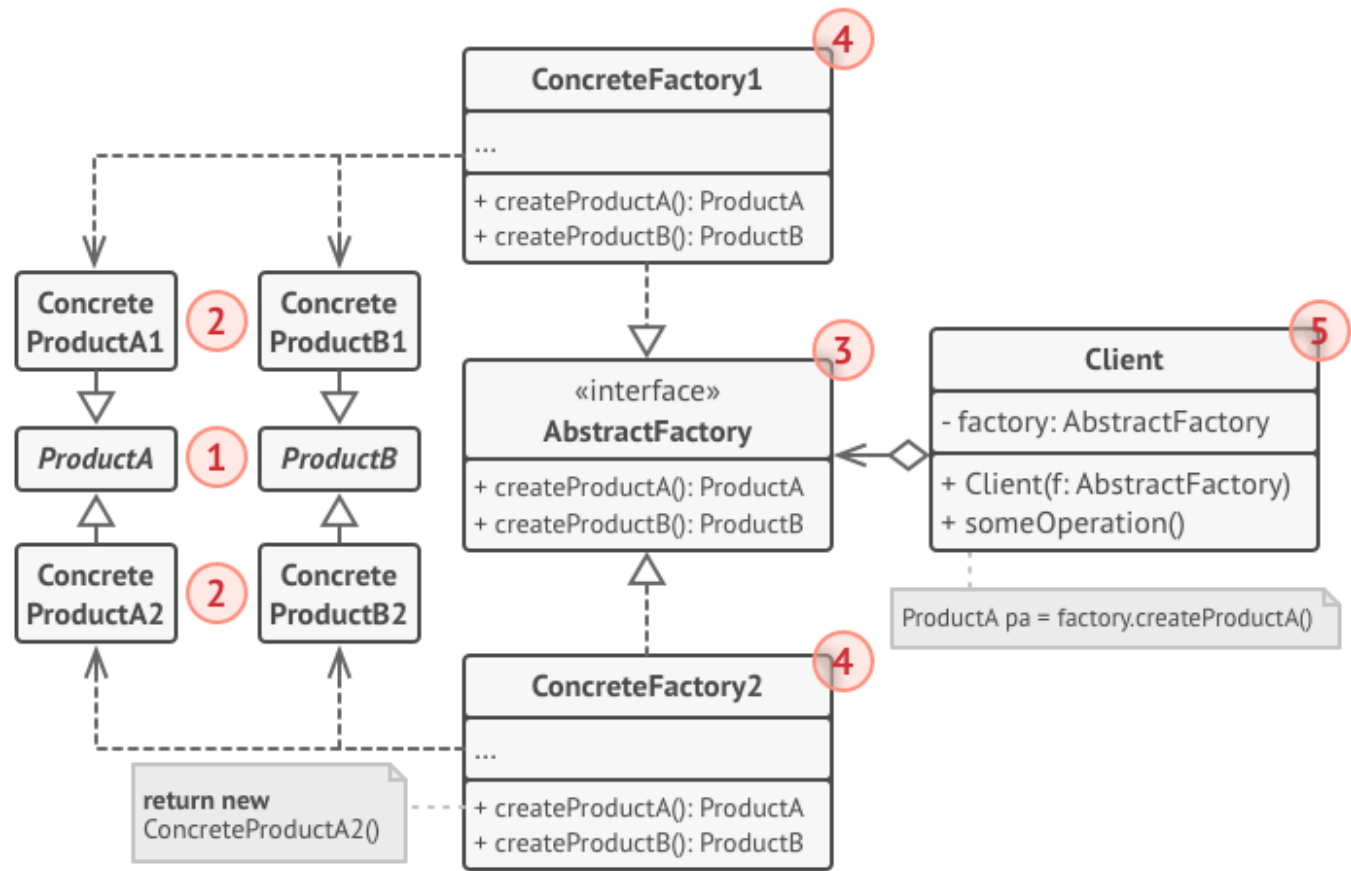
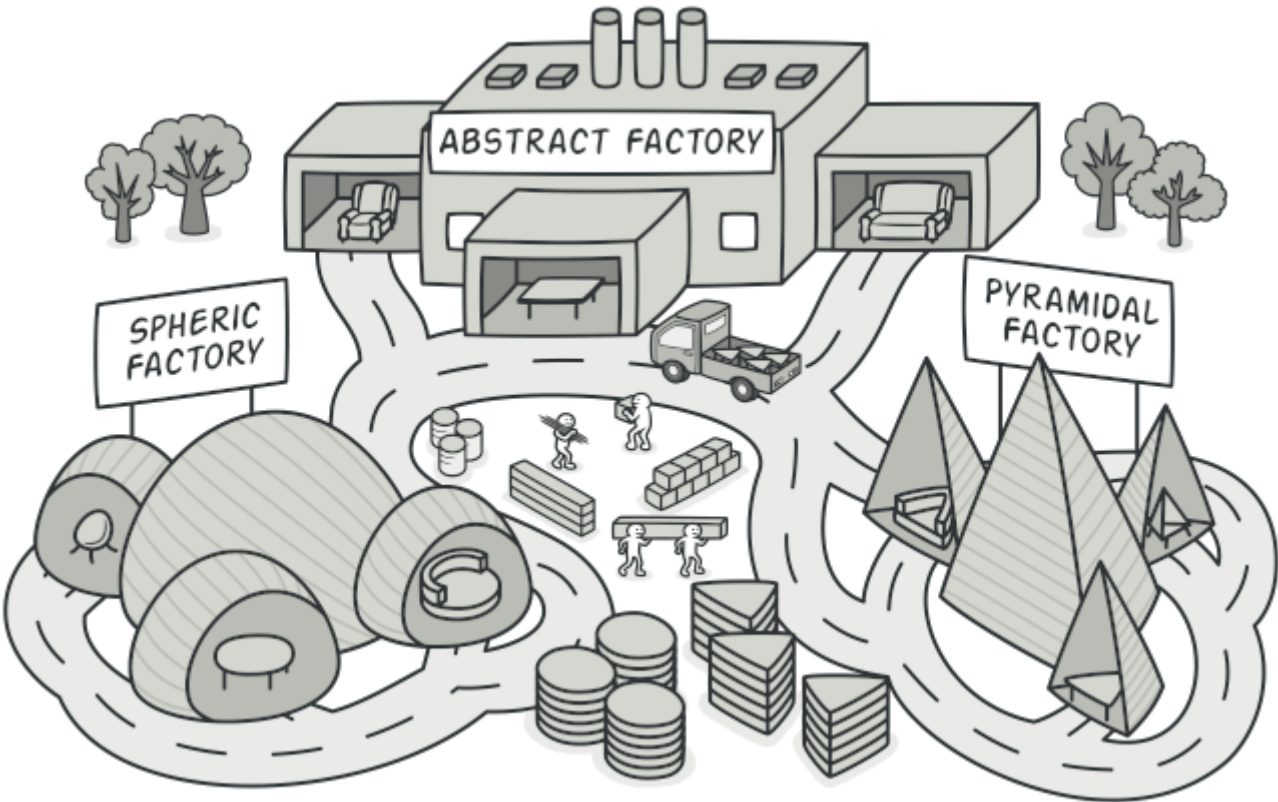
1.1 Factory Method



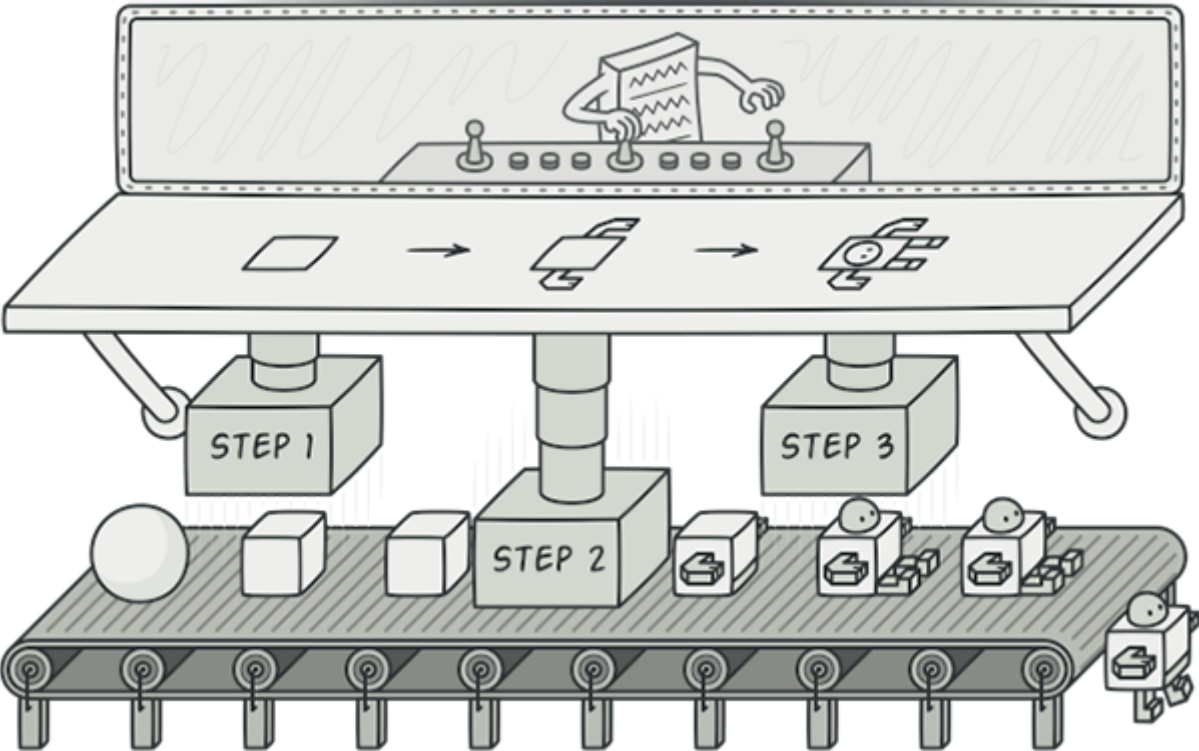
Product p = createProduct()
p.doStuff()



1.2 Abstract Factory

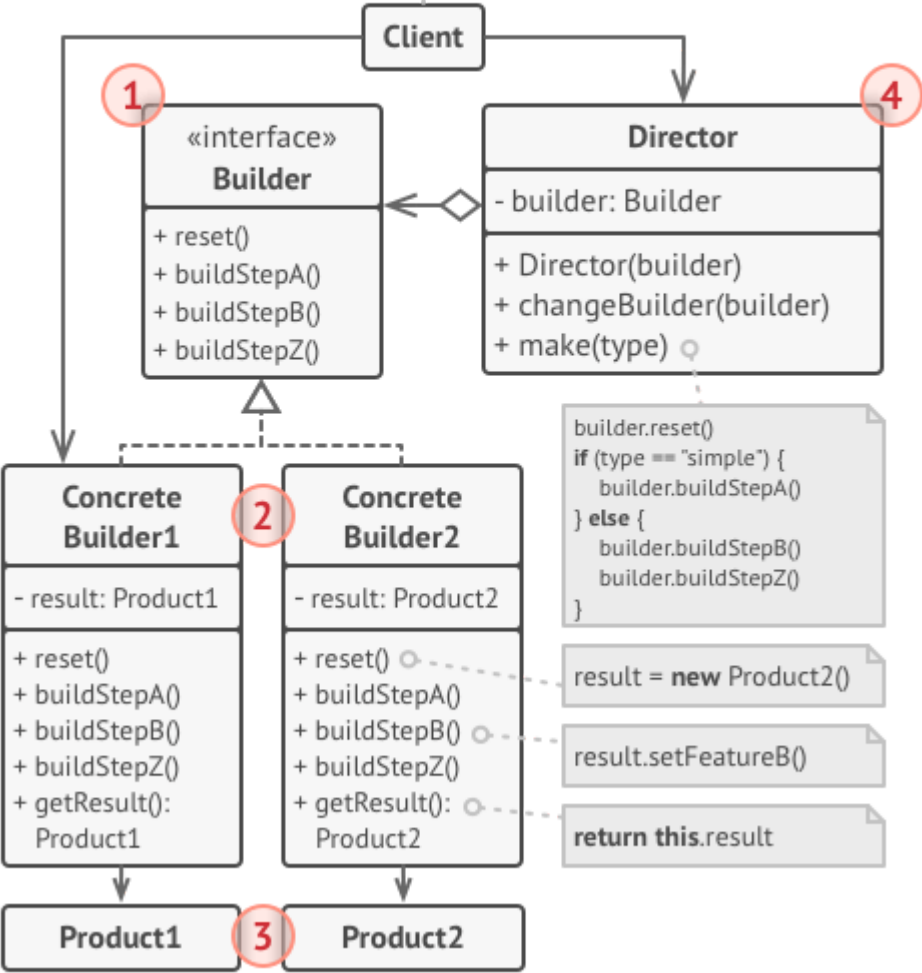


1.3 Builder

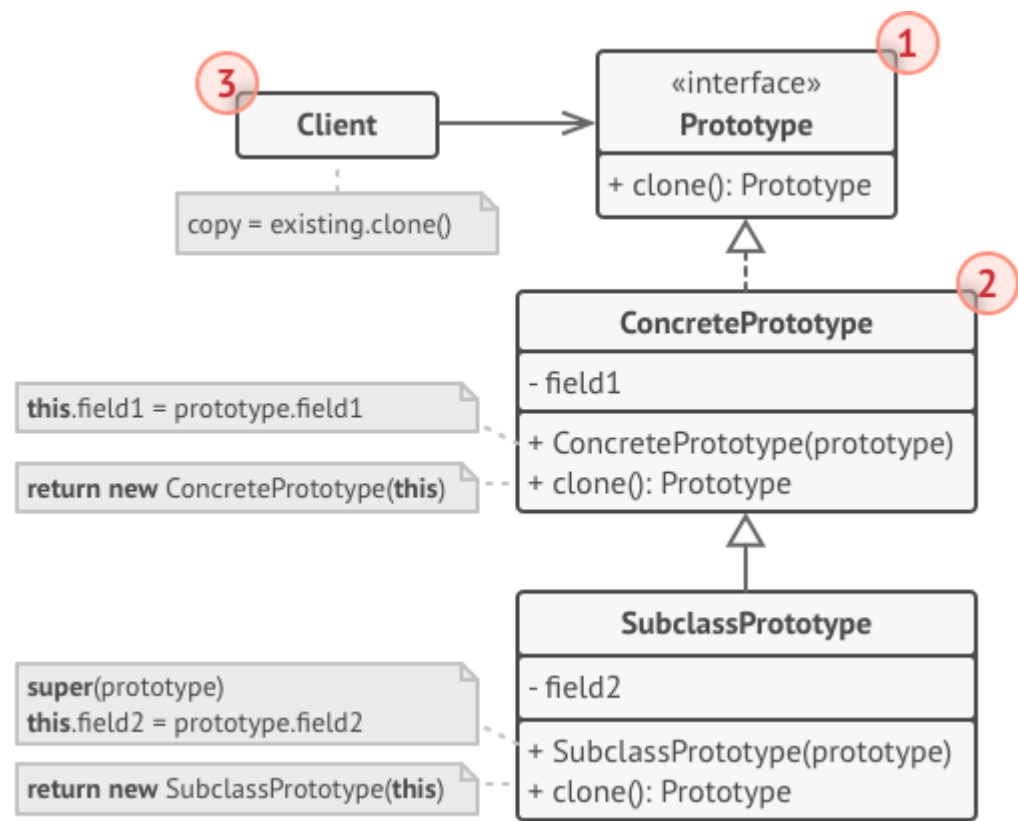
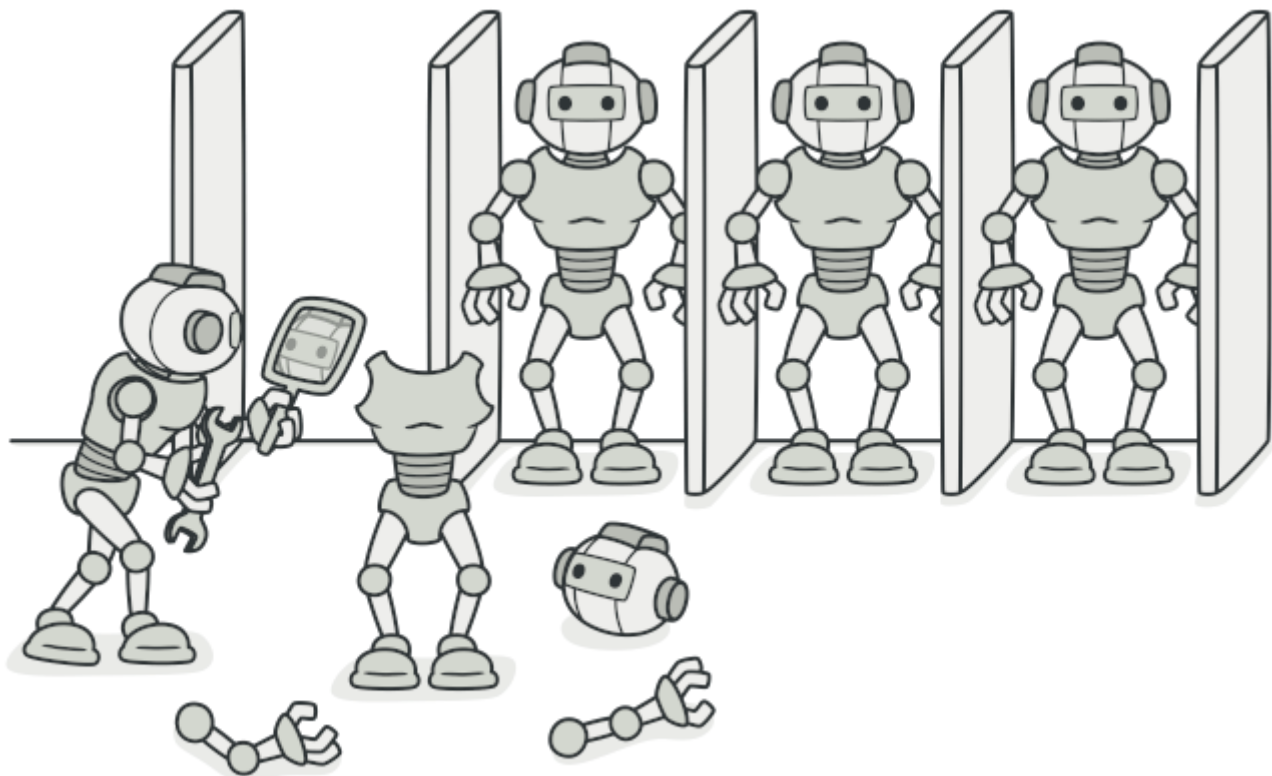


```
b = new ConcreteBuilder1()
d = new Director(b)
d.make()
Product1 p = b.getResult()
```

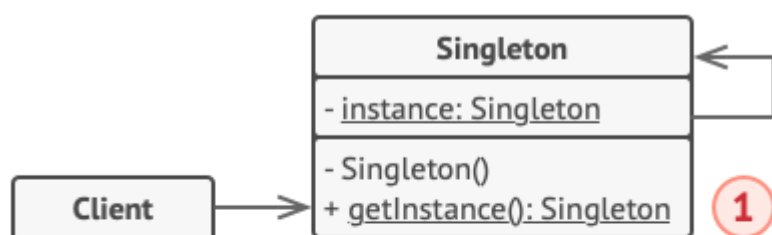
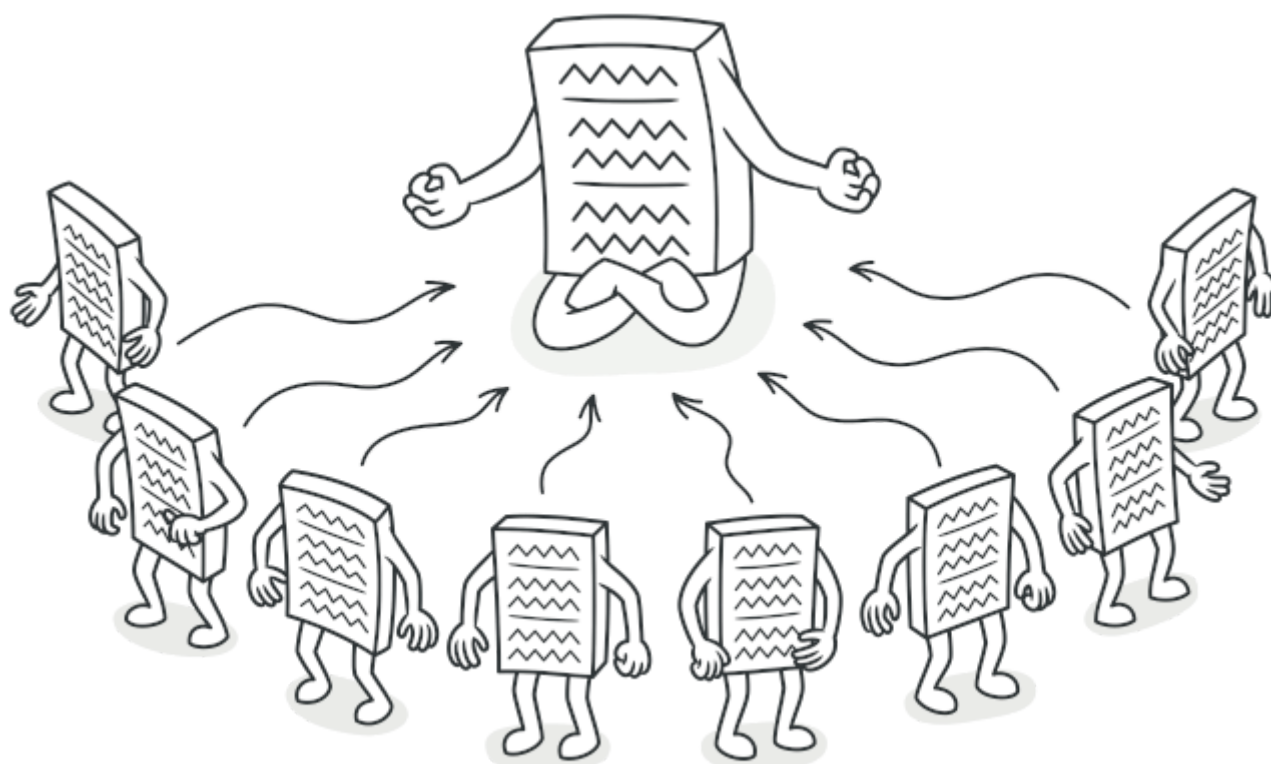
5



1.4 Prototype

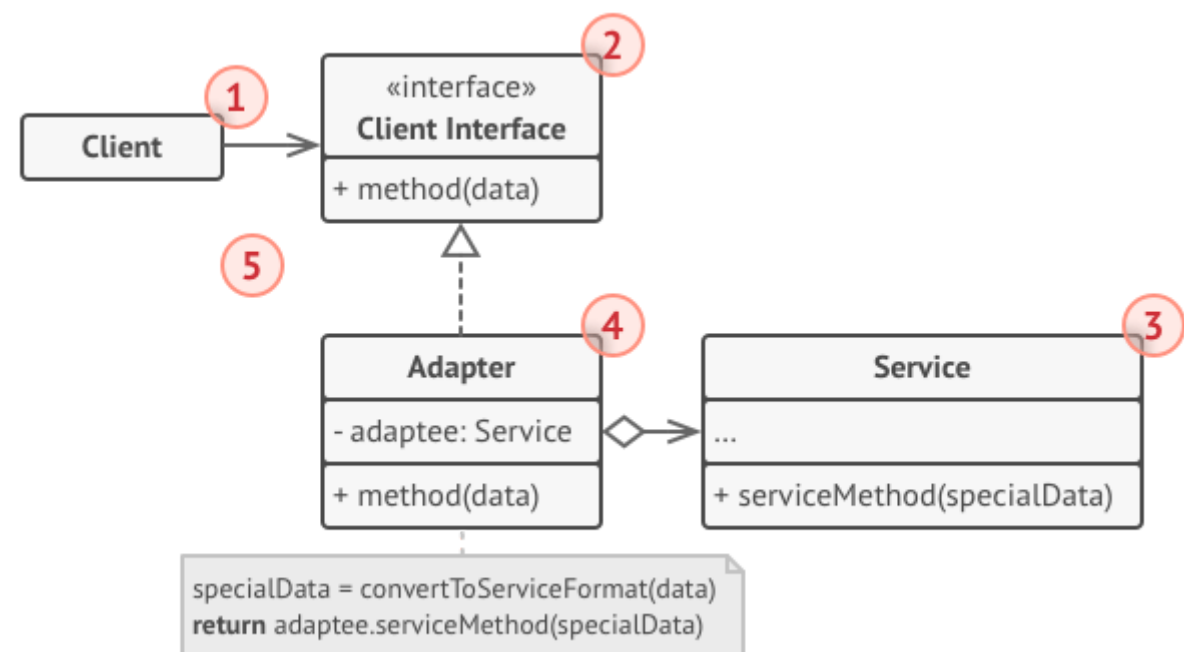


1.5 Singleton

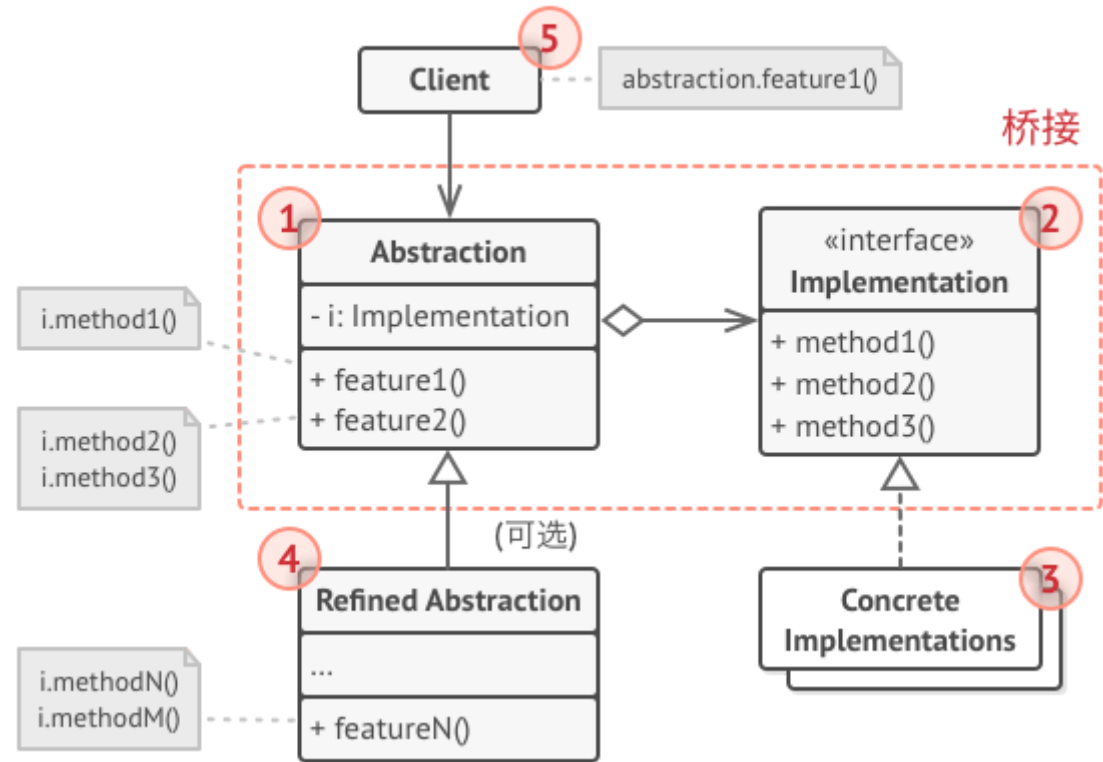
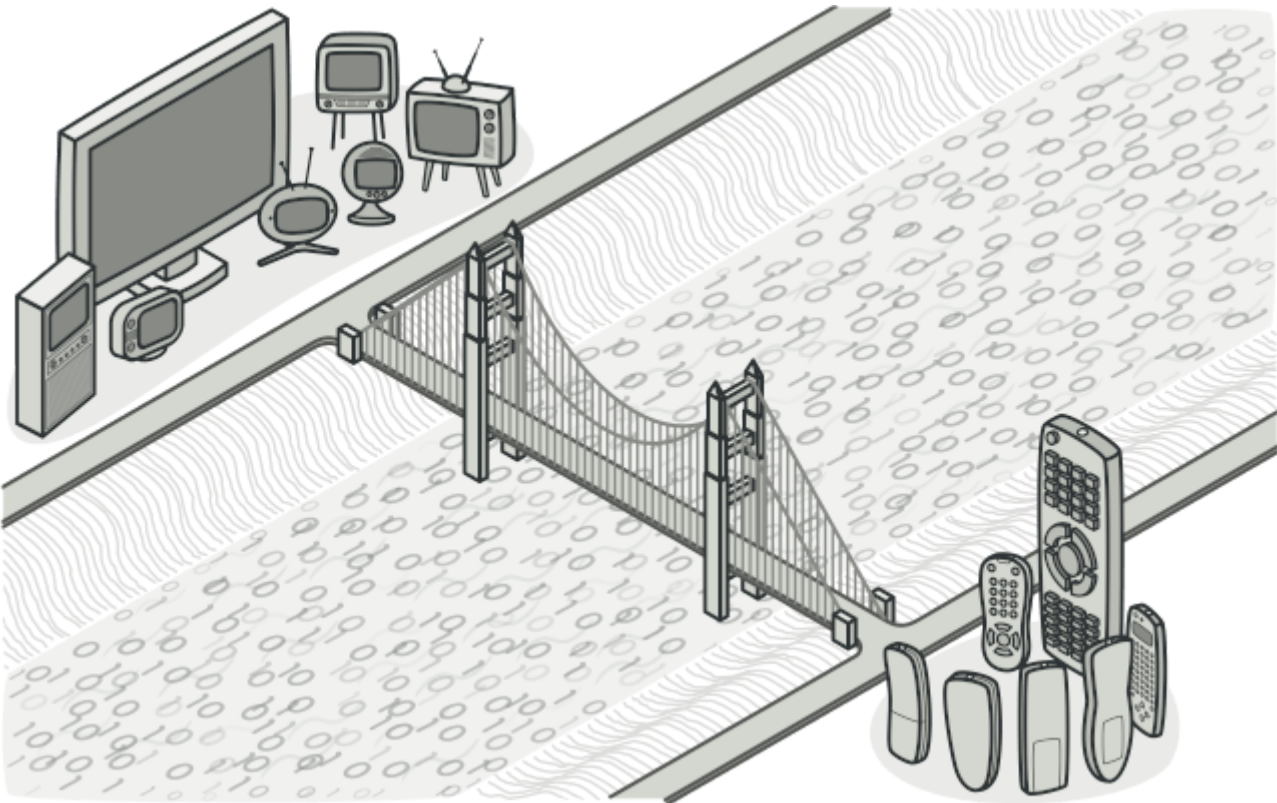


```
if (instance == null) {  
    // 注意：如果程序需要支持多线程，  
    // 你必须在此放置线程锁。  
    instance = new Singleton()  
}  
return instance
```

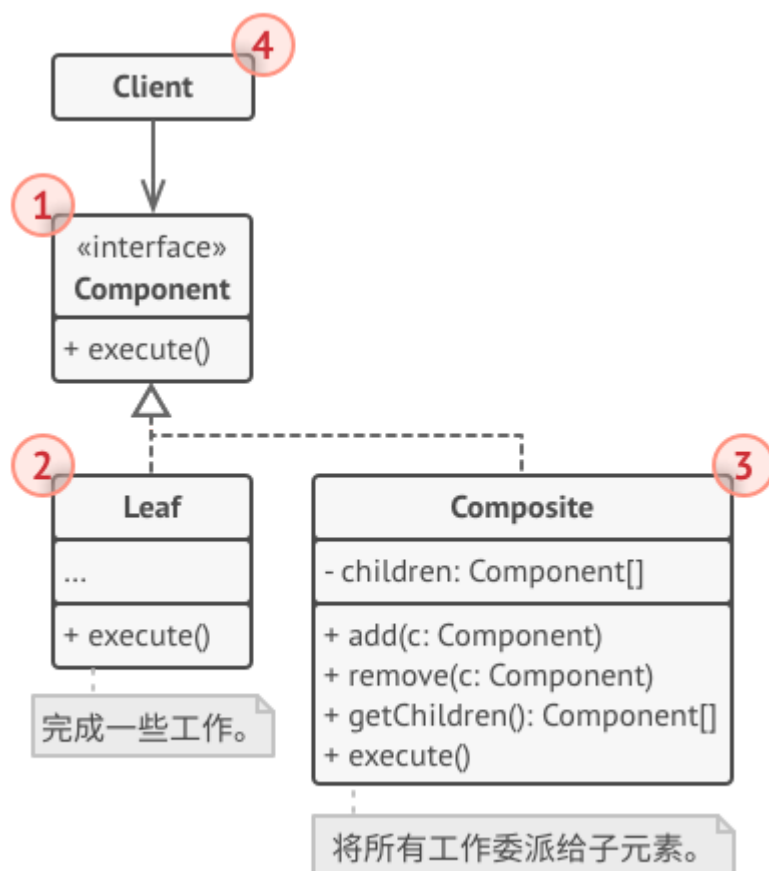
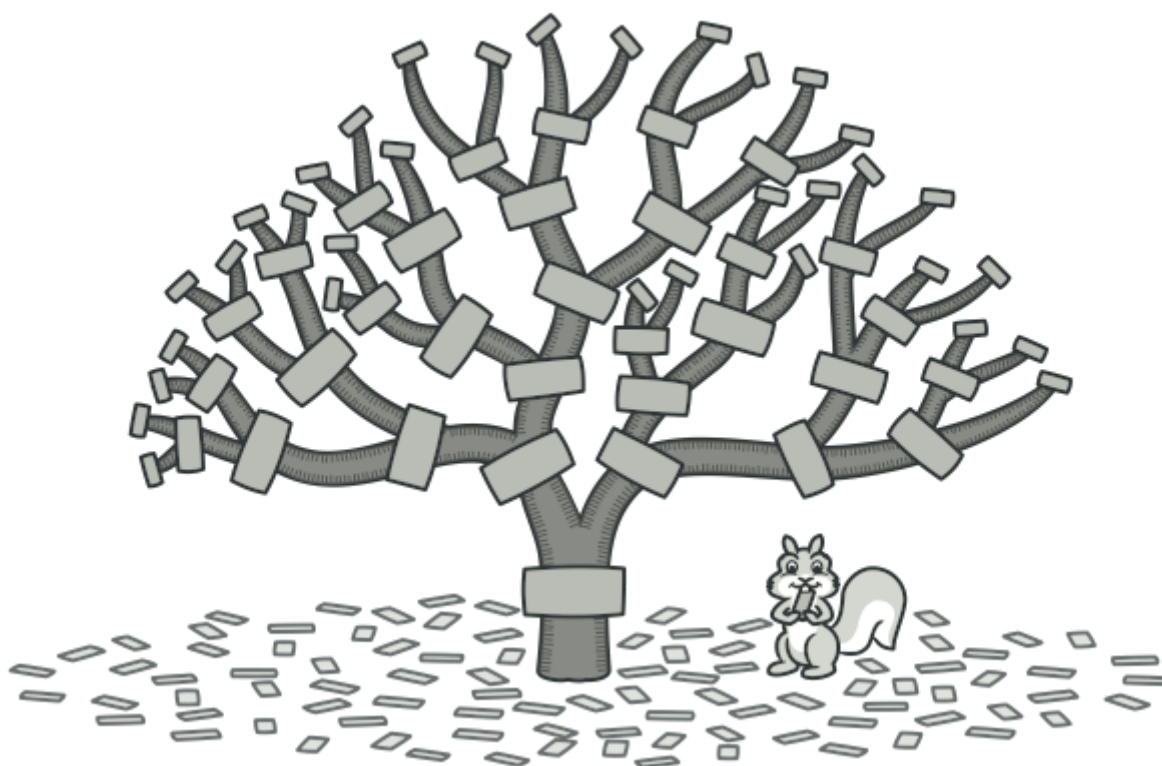

2.1 Adapter



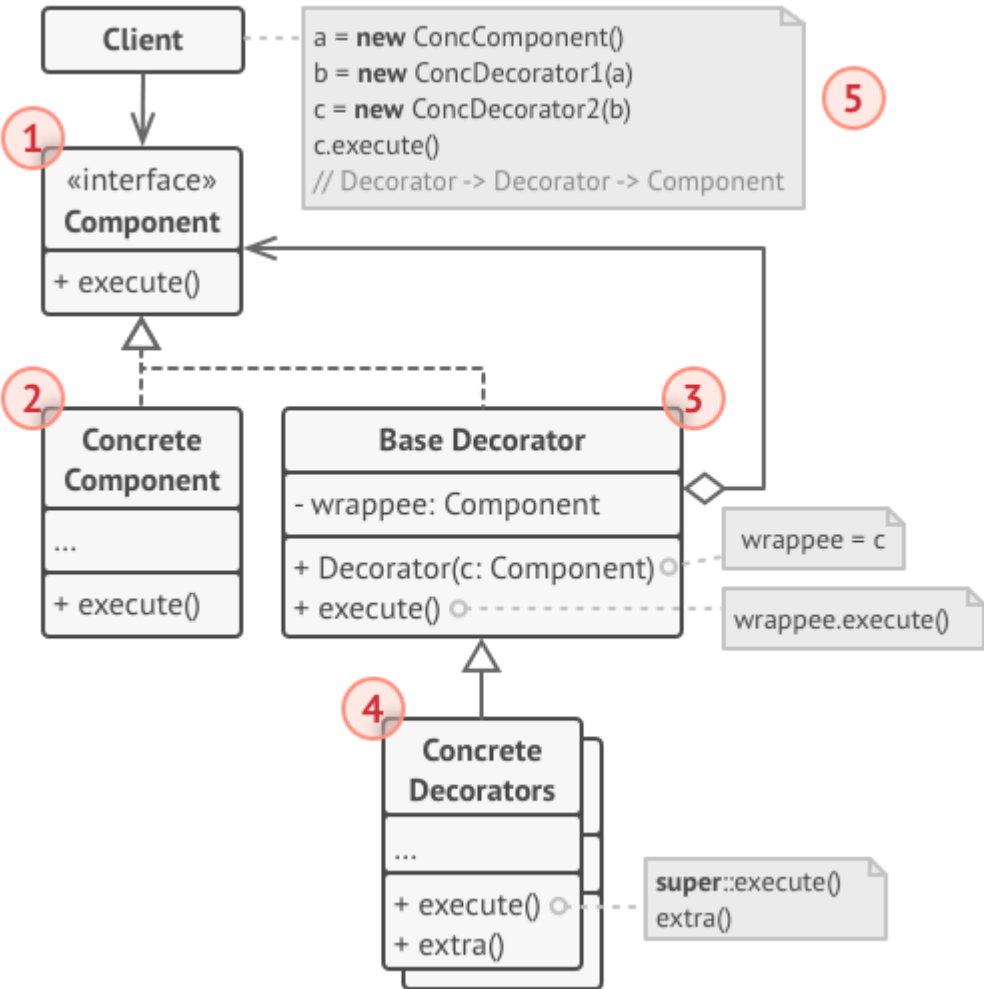
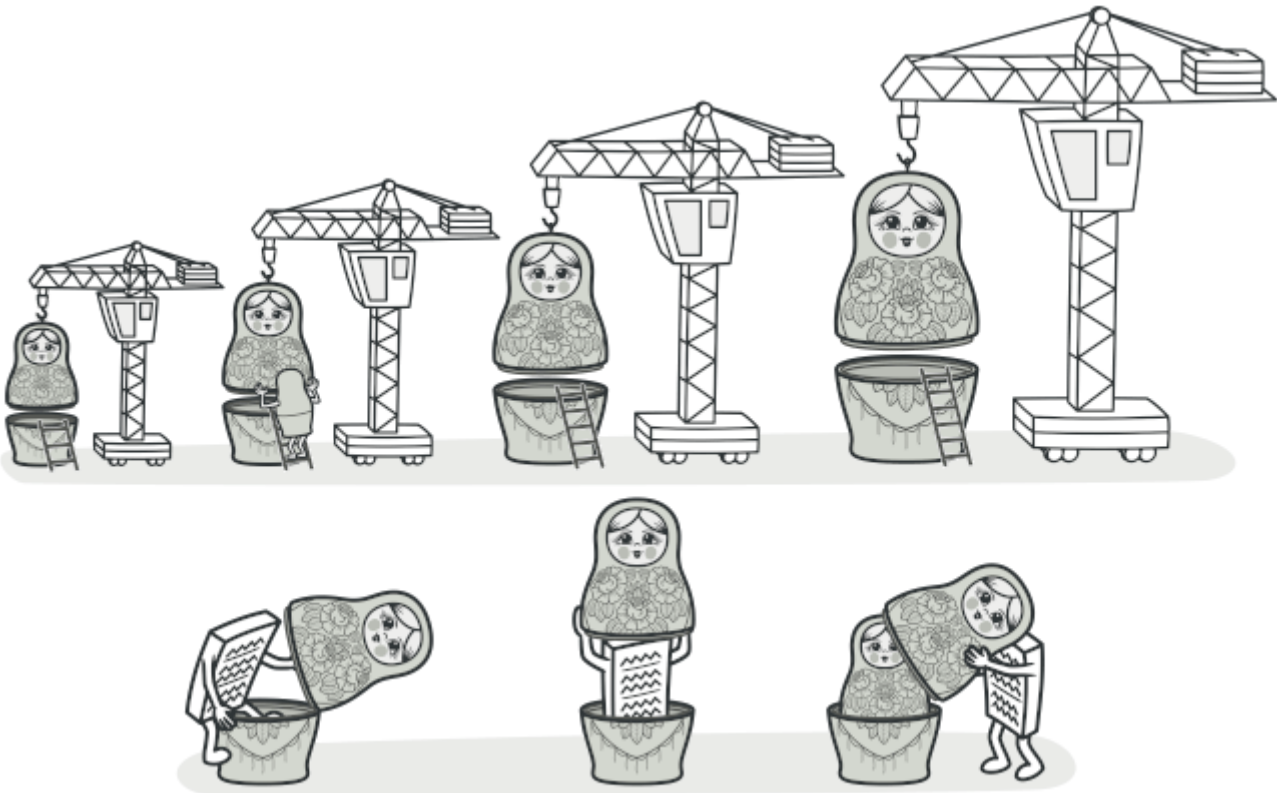
2.2 Bridge



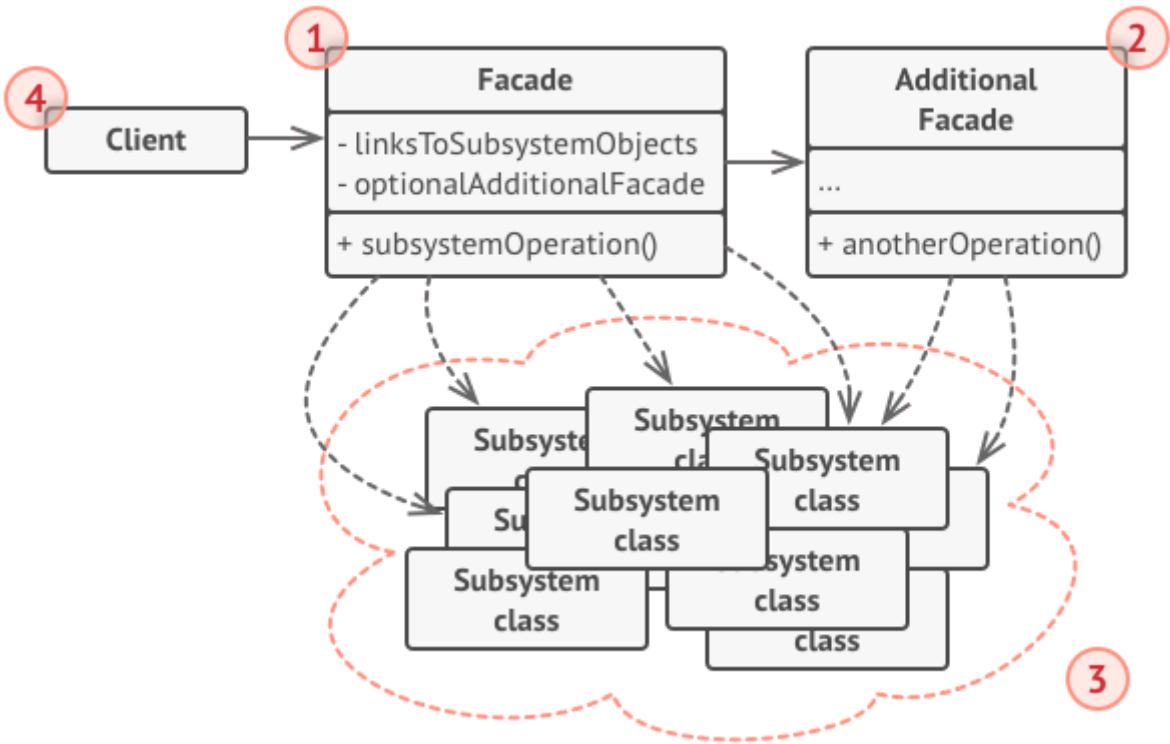
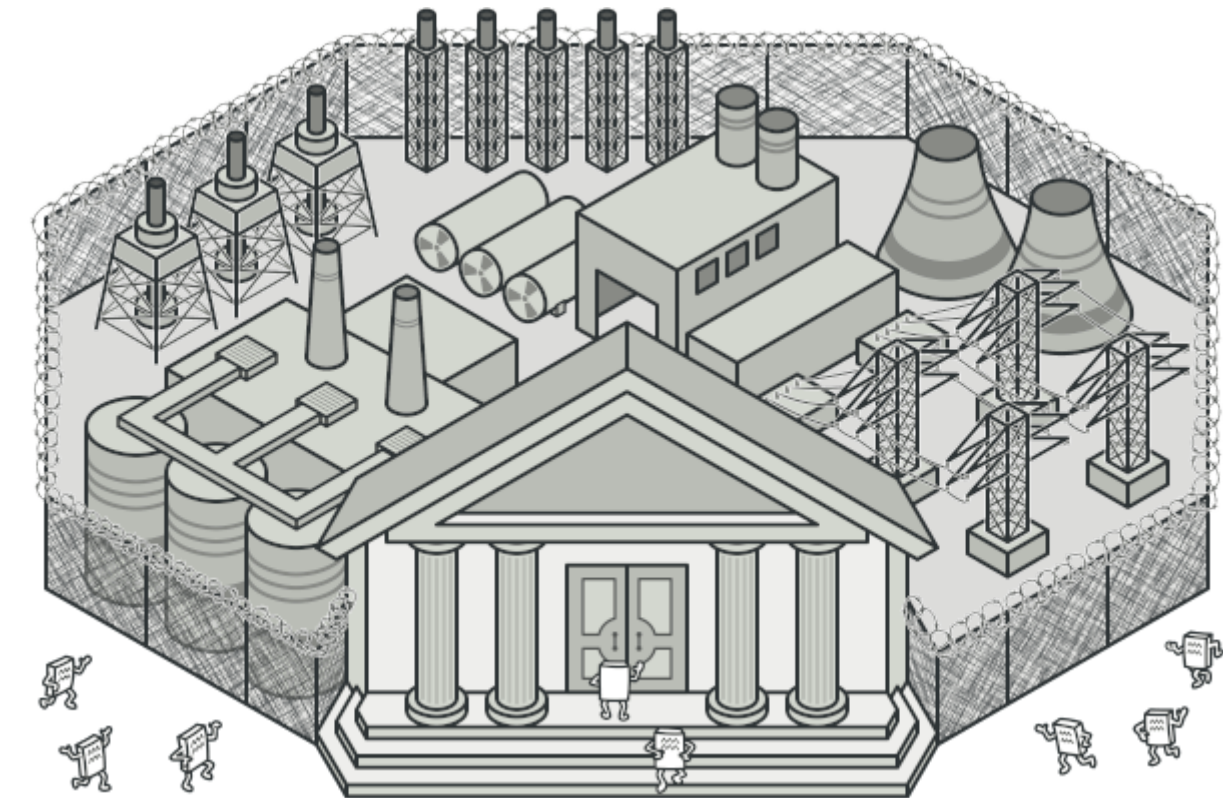
2.3 Composite



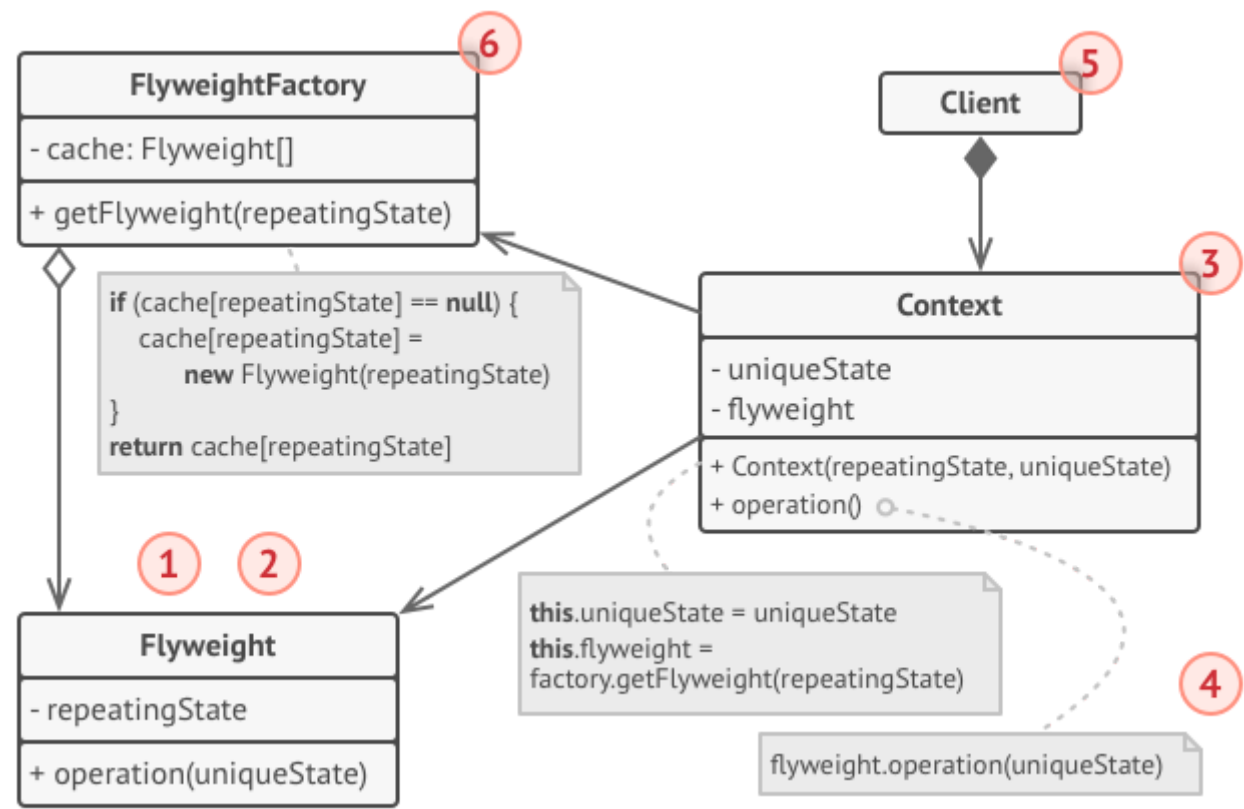
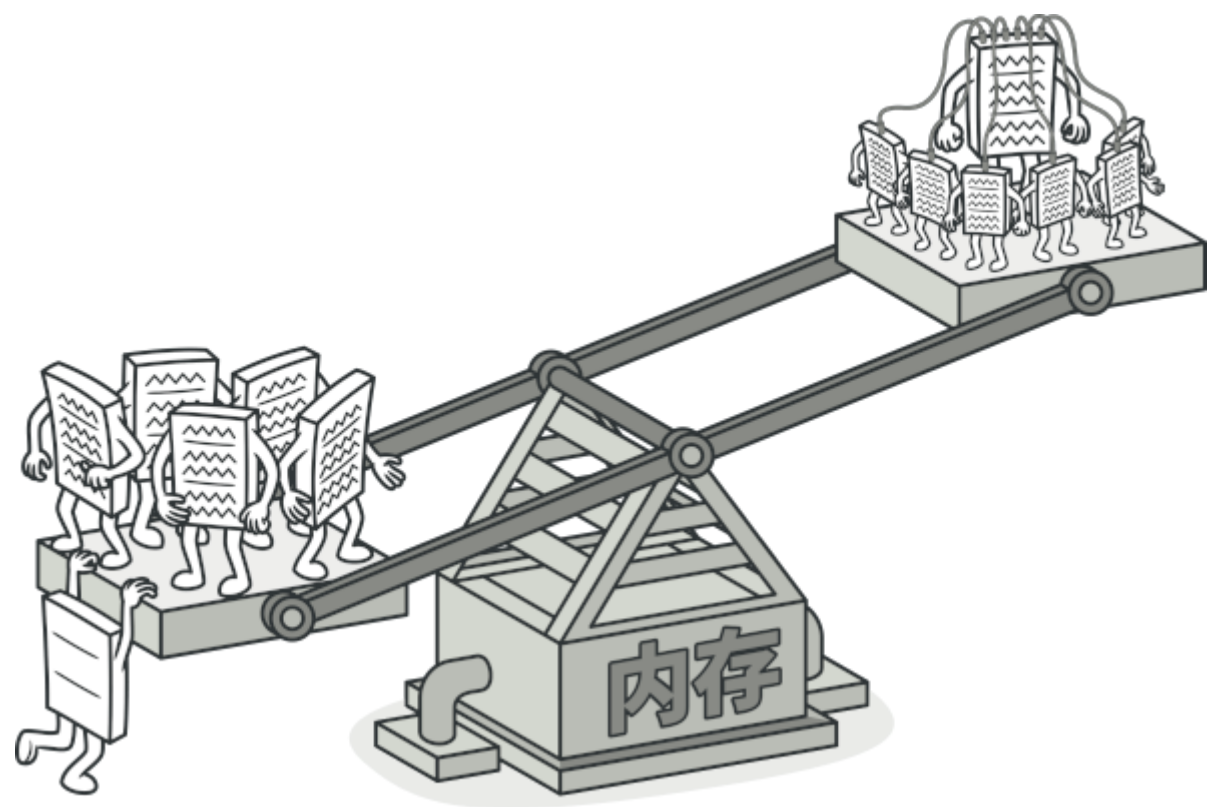
2.4 Decorator



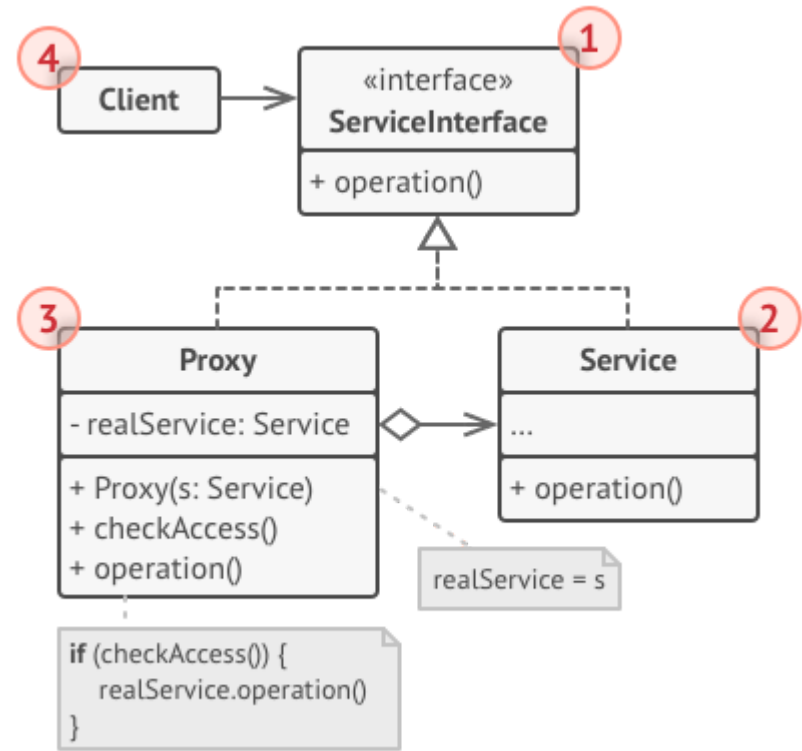
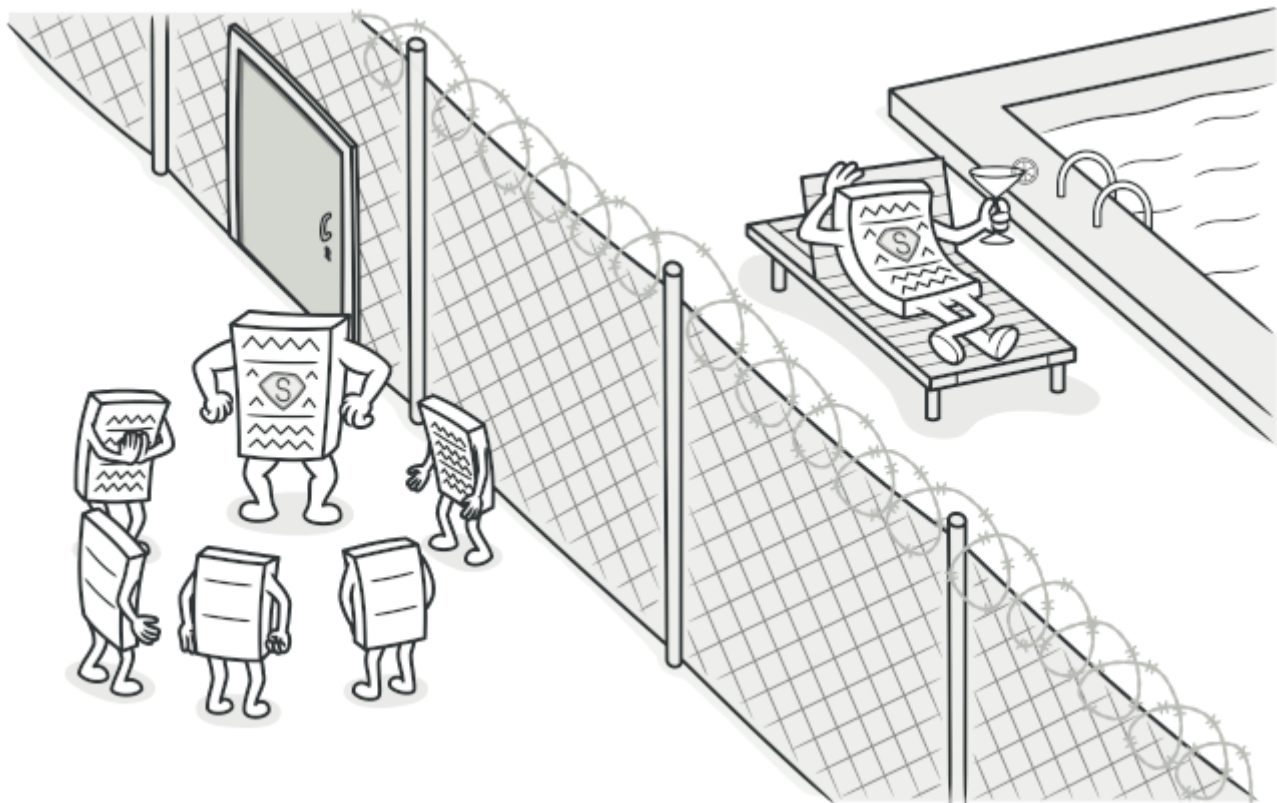
2.5 Facade



2.6 Flyweight

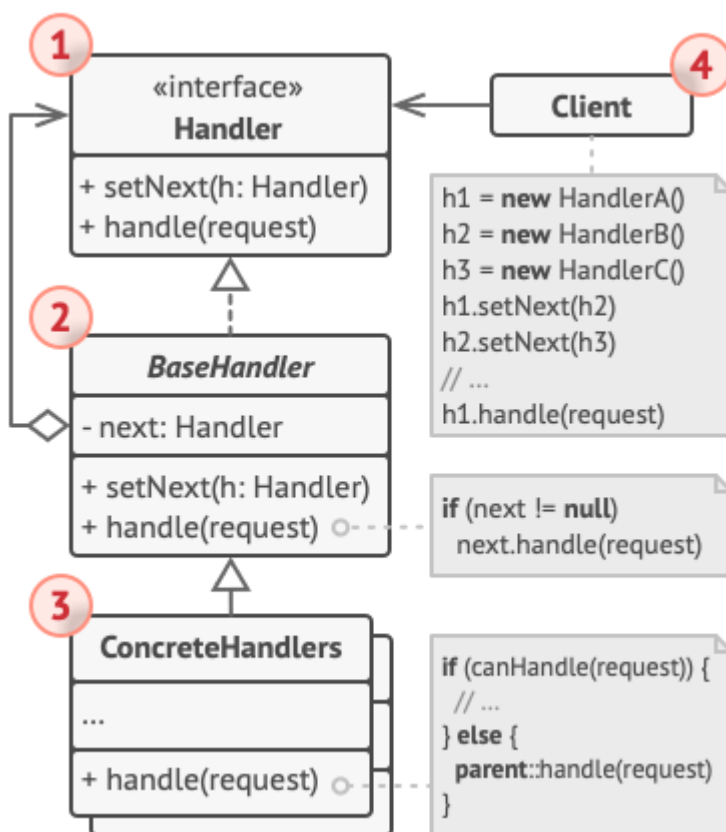
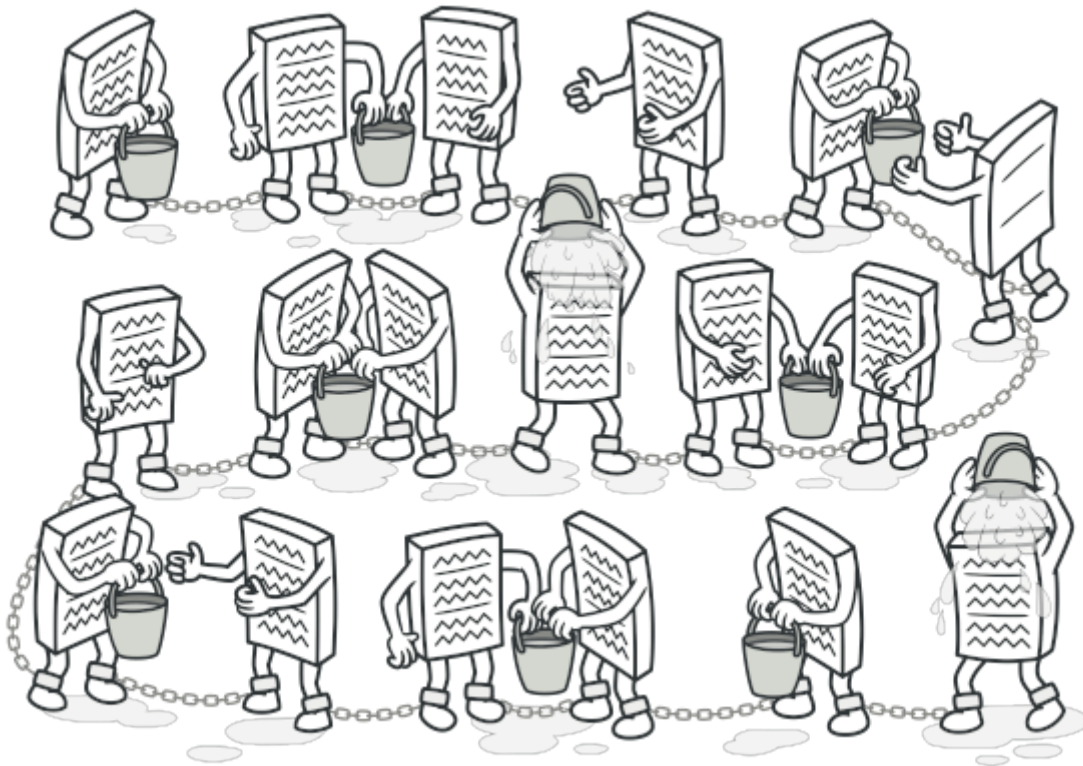


2.7 Proxy

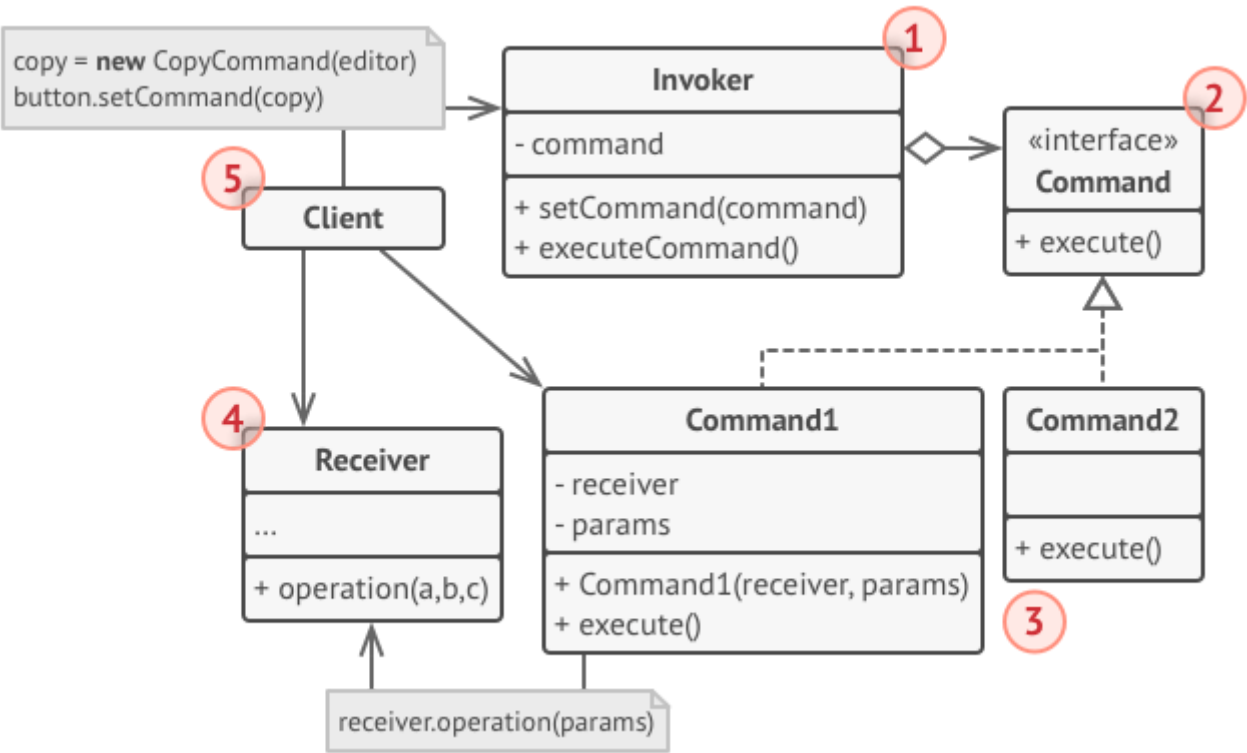
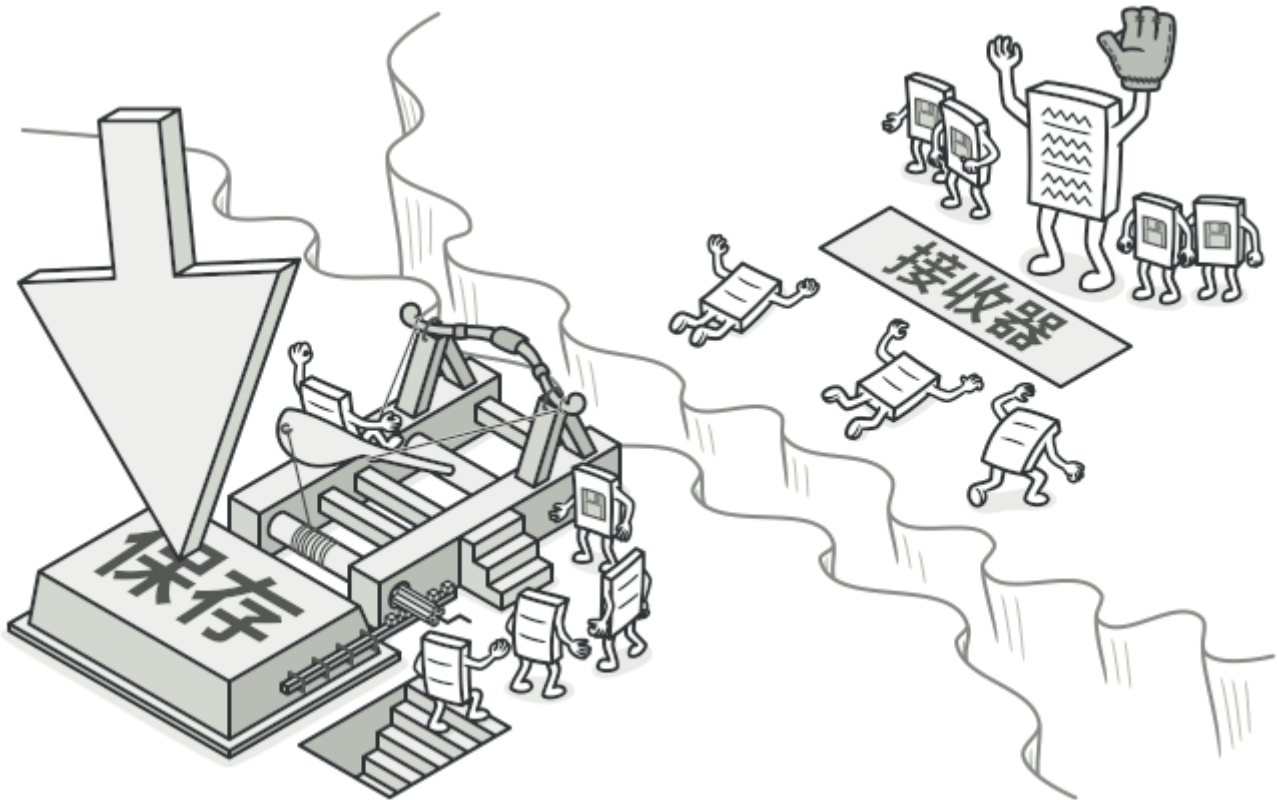


3. Behavioral Patterns

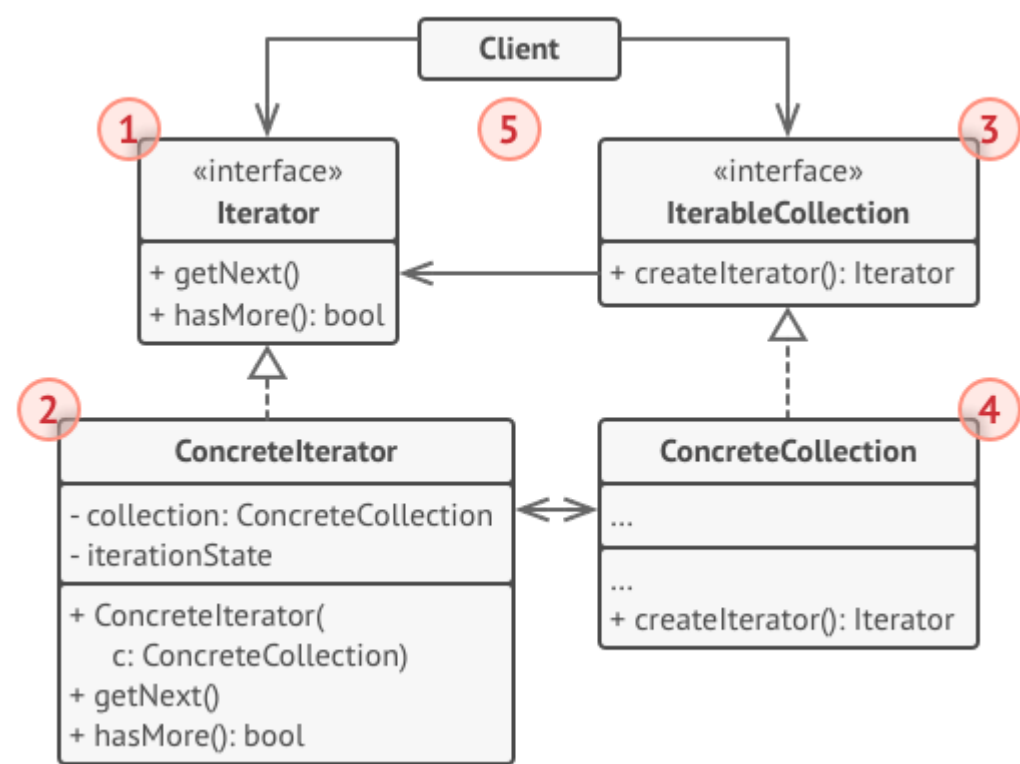
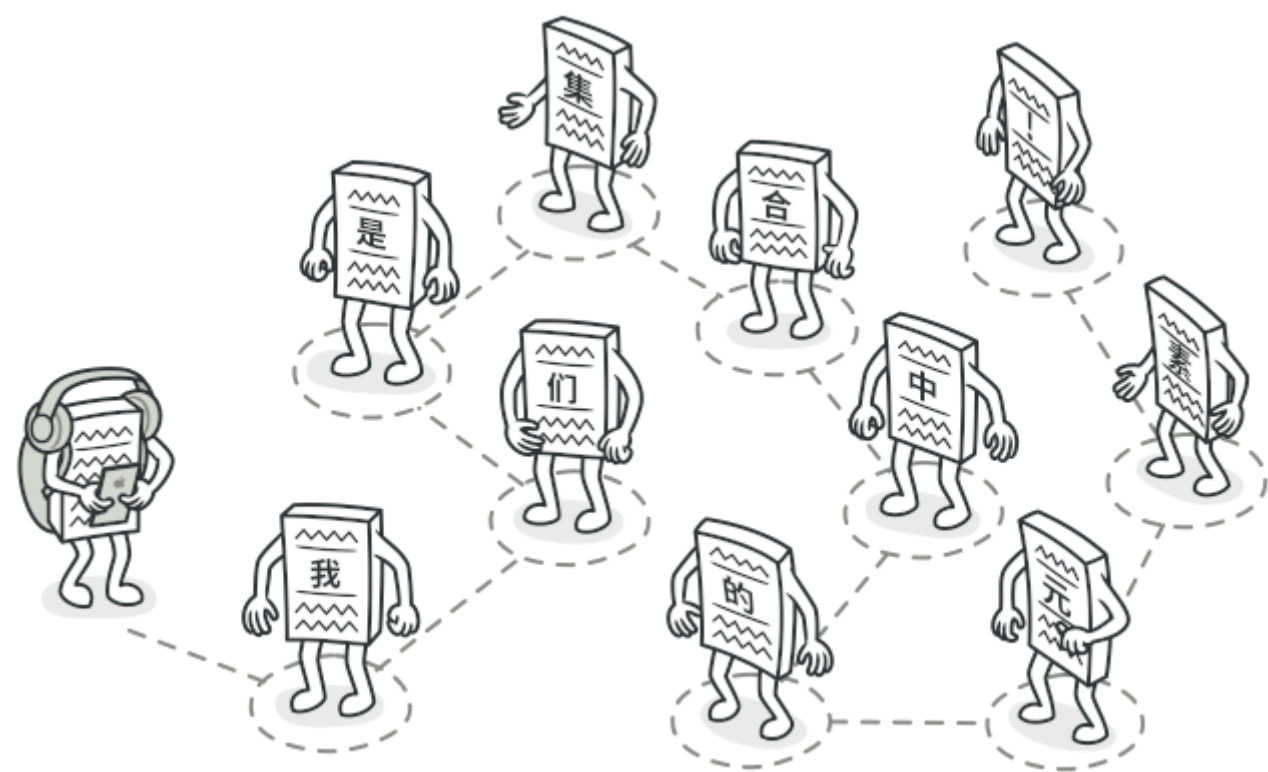
3.1 Chain of Responsibility



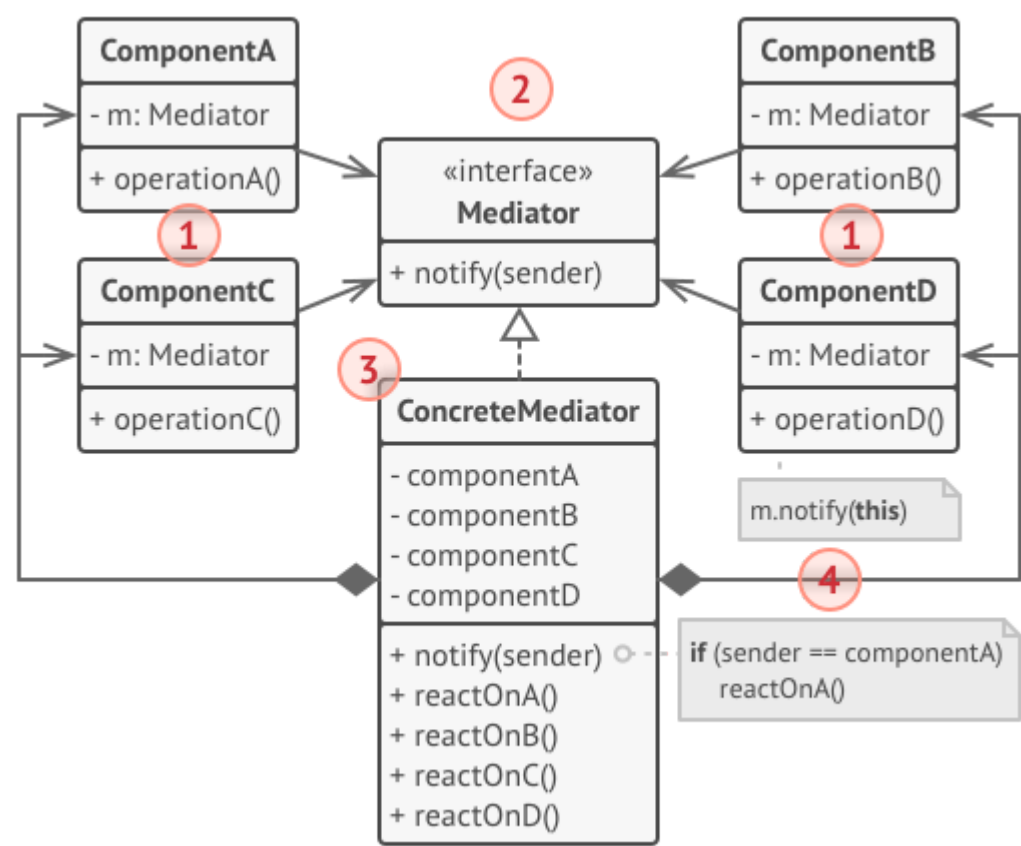
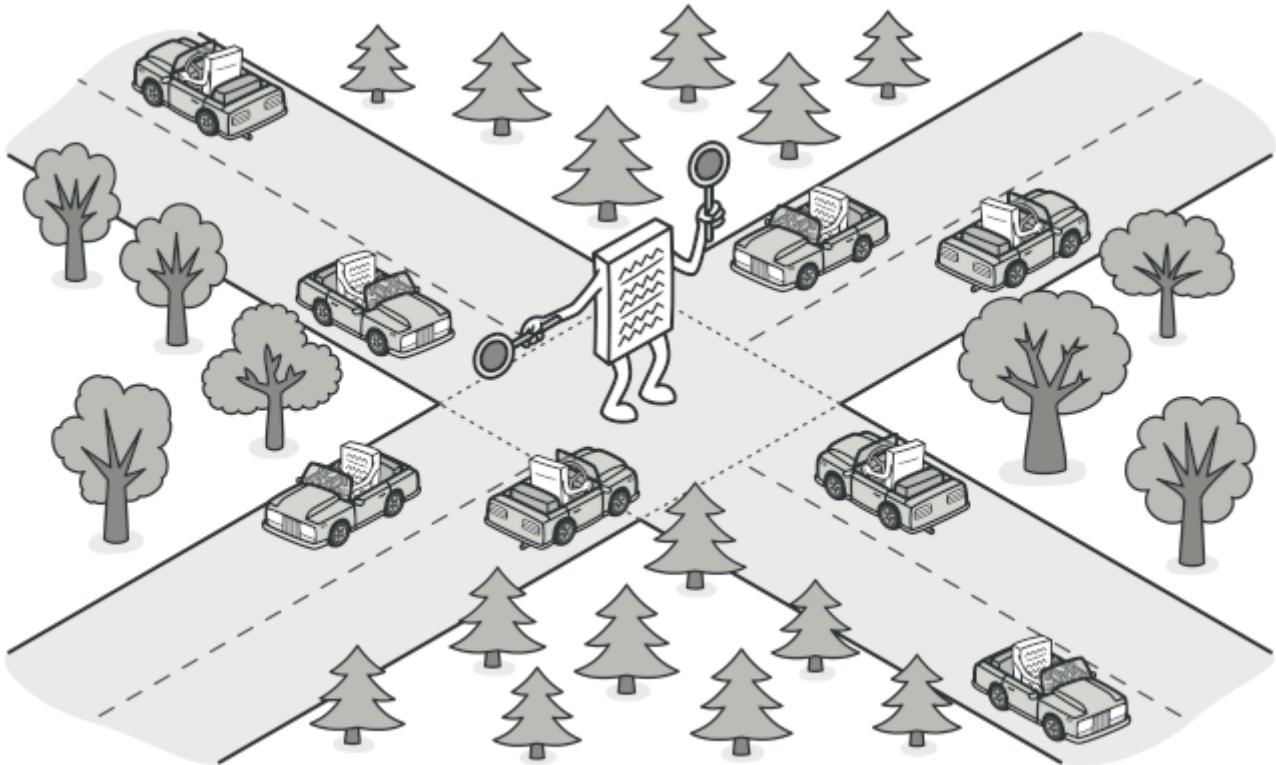
3.2 Command



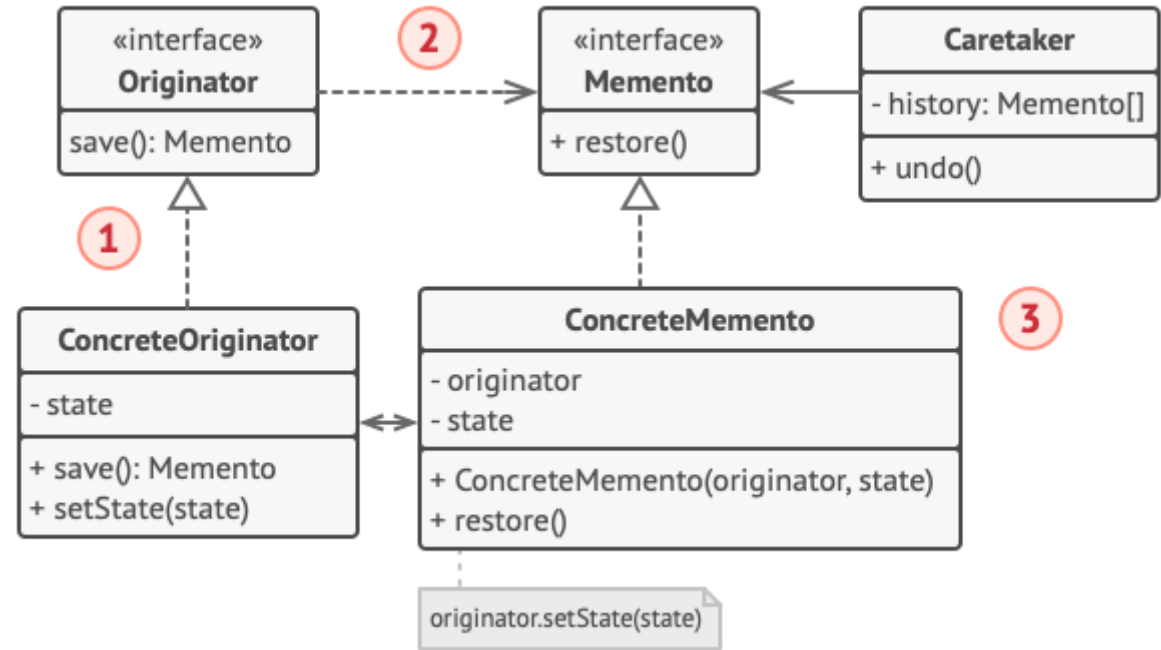
3.3 Iterator



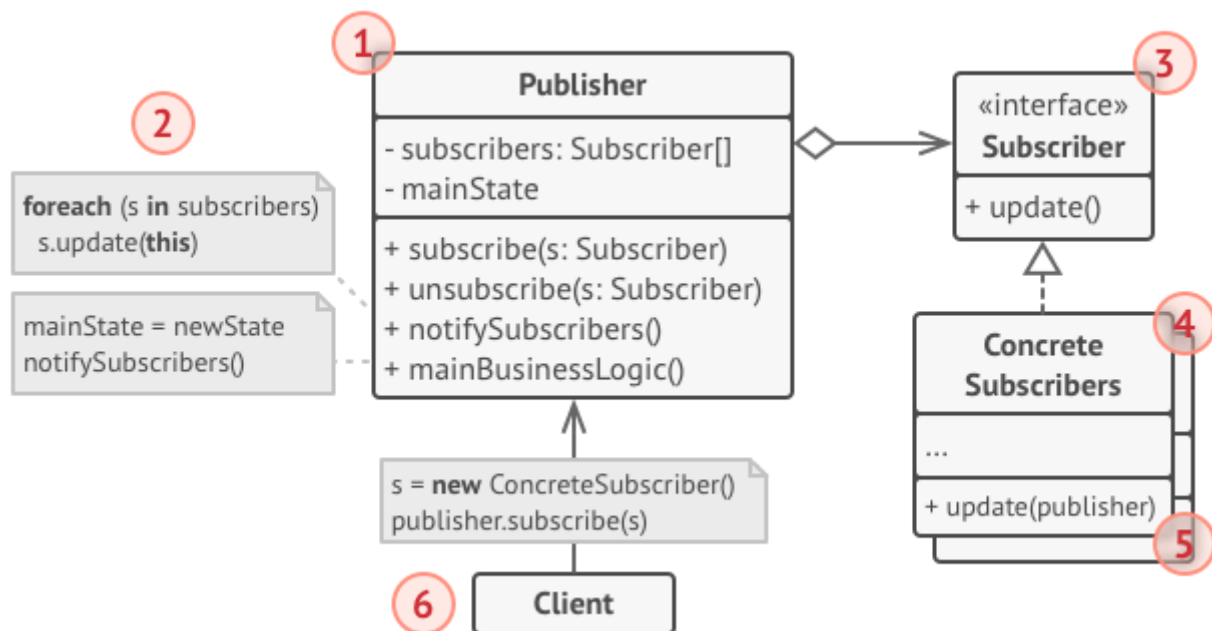
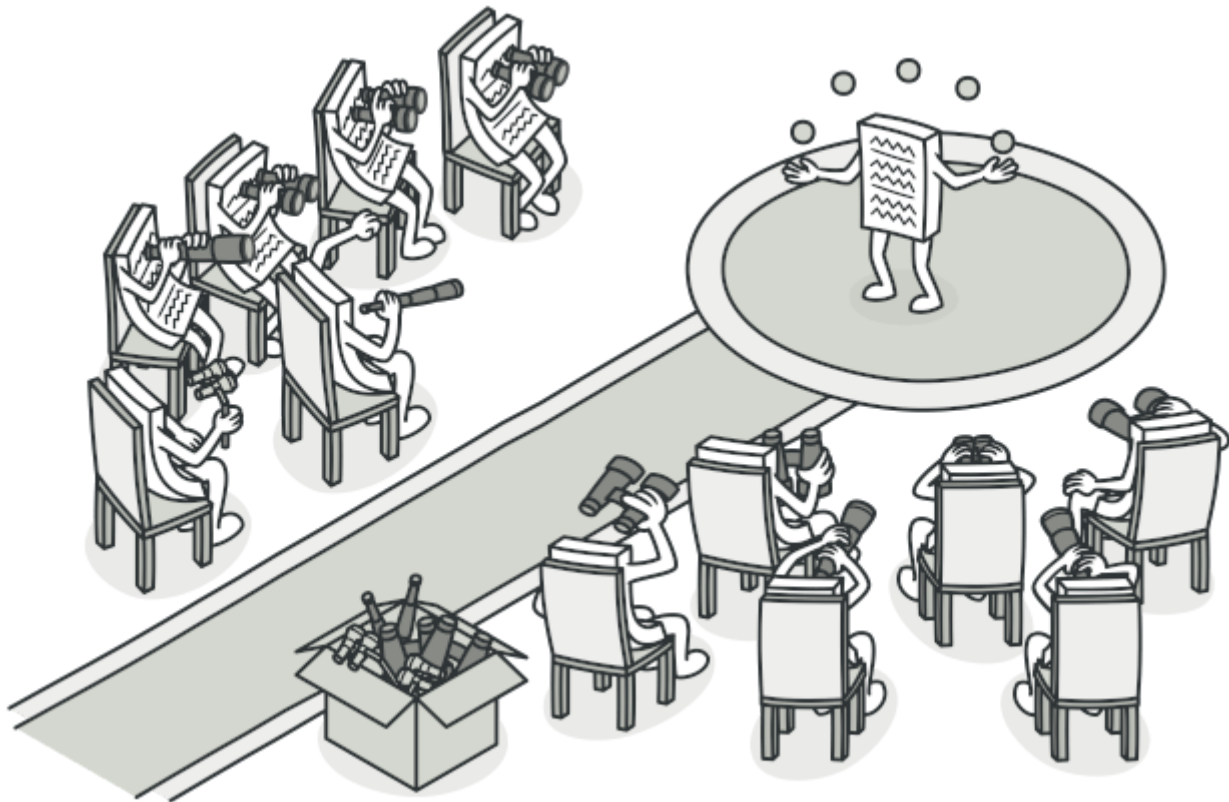
3.4 Mediator



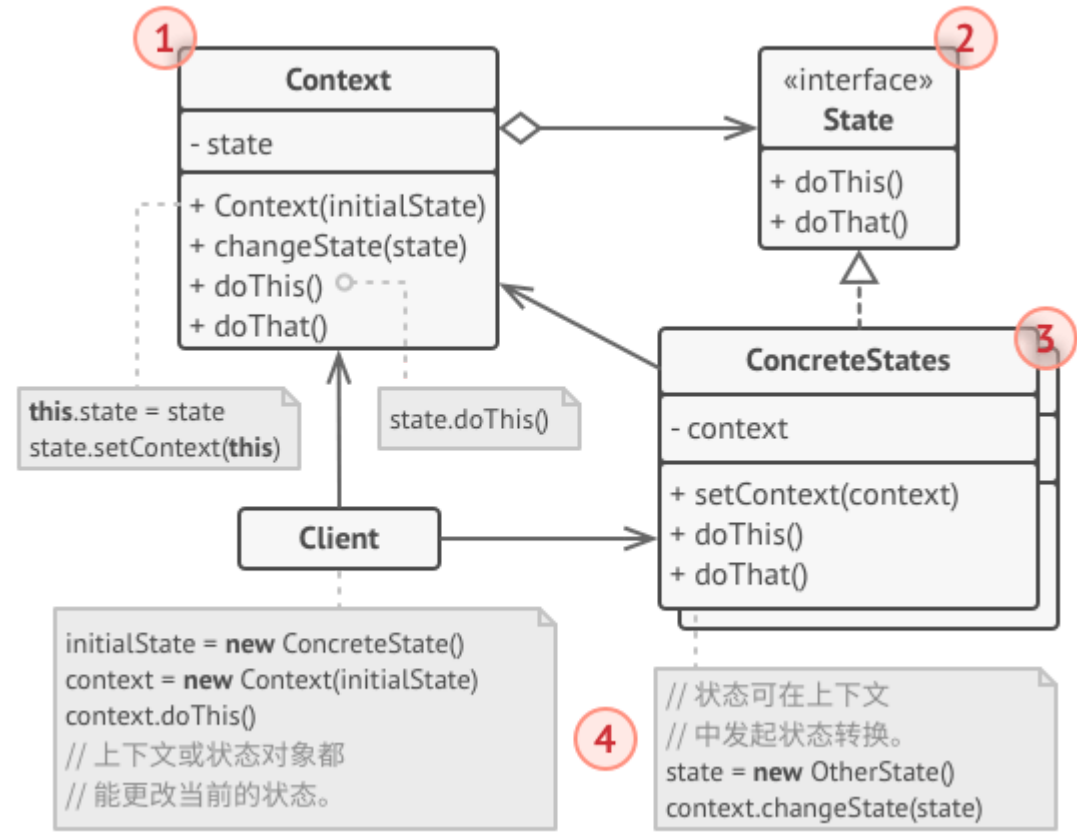
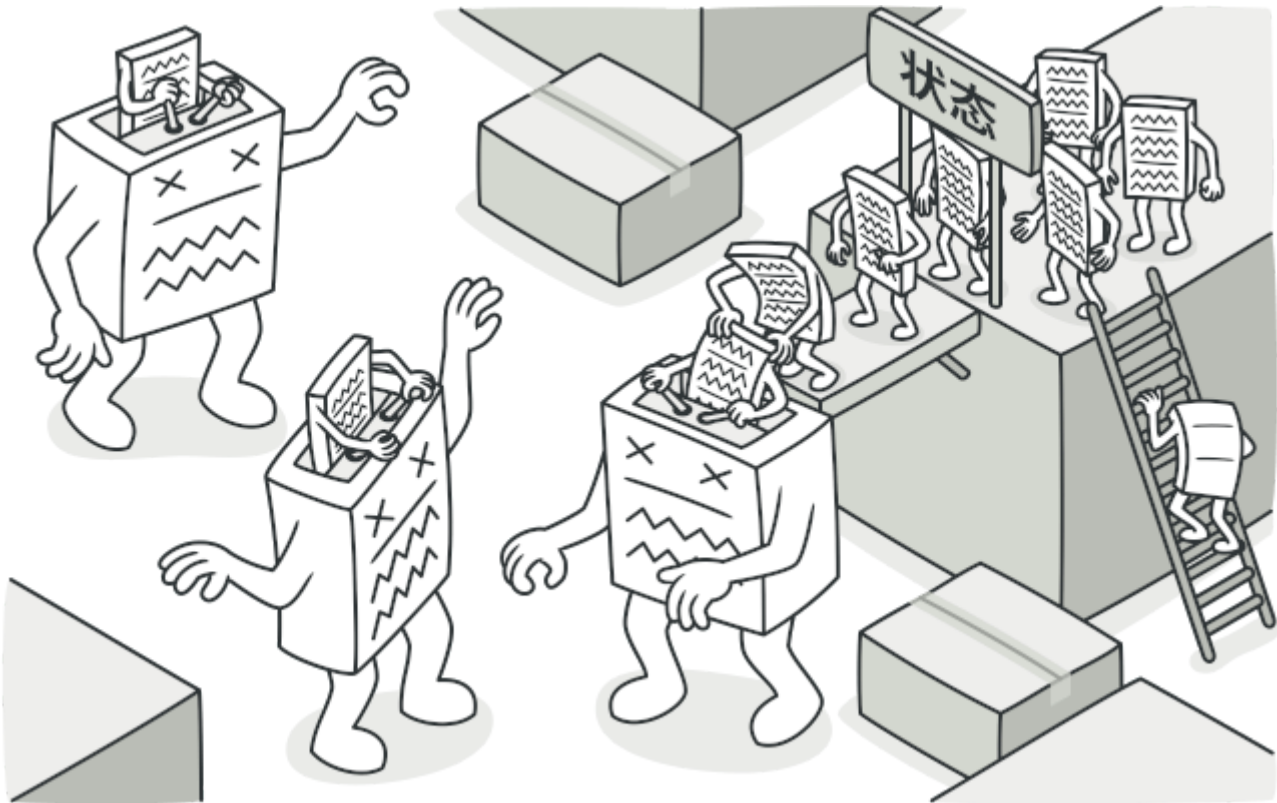
3.5 Memento



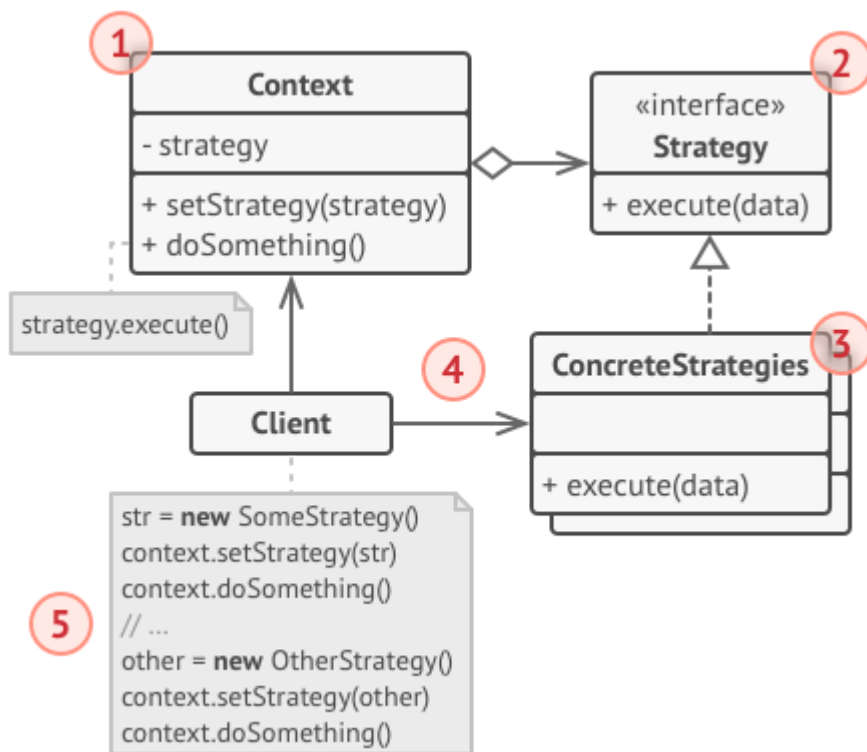
3.6 Observer



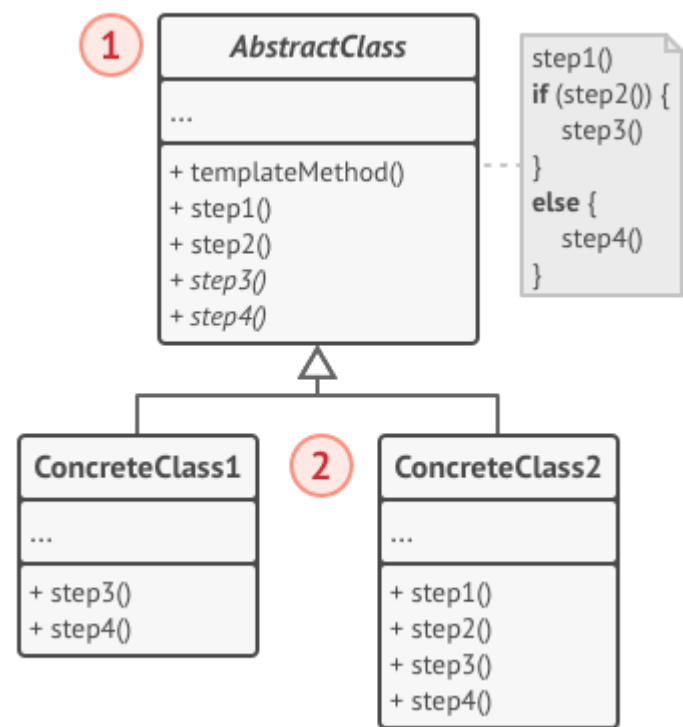
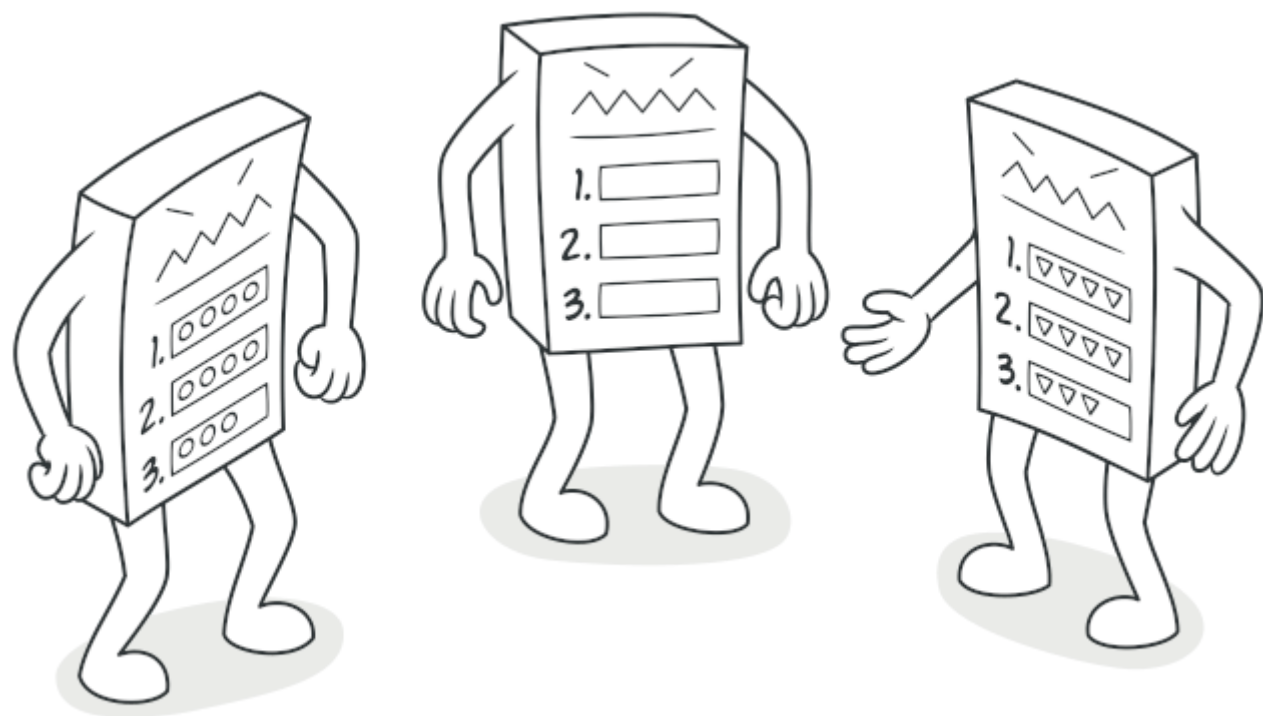
3.7 State



3.8 Strategy



3.9 Template Method



3.10 Vistor

