

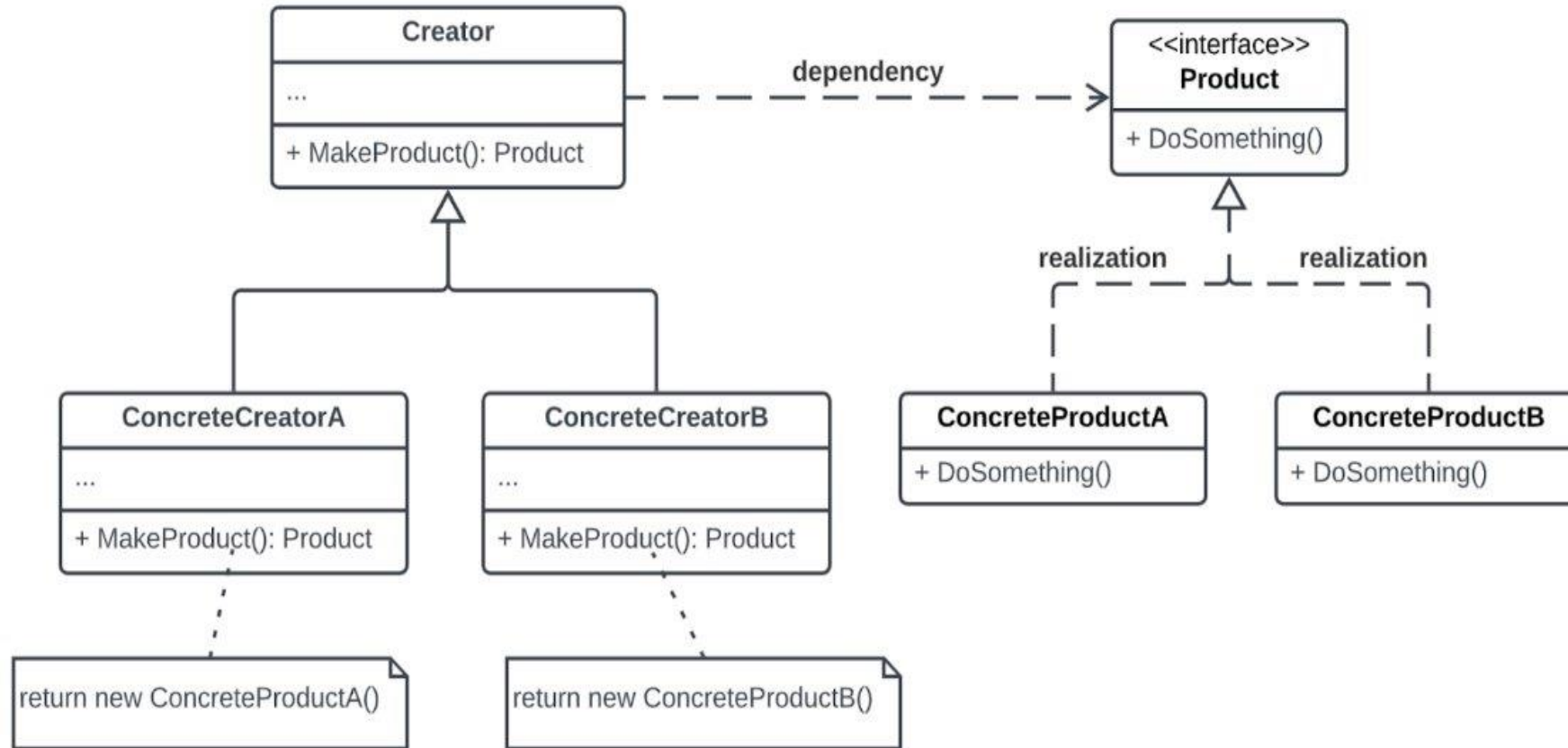
Factory

Factory Method is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created.

The Factory Method pattern suggests that you replace direct object construction calls (using the *new* operator) with calls to a special *factory* method.



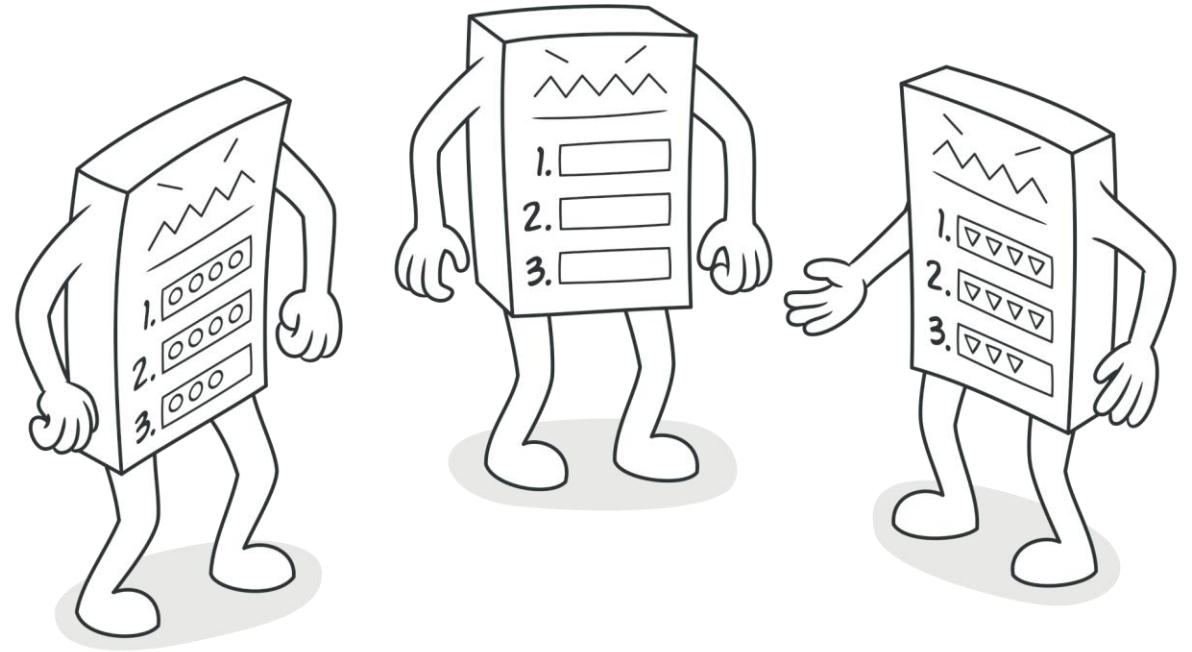
Factory



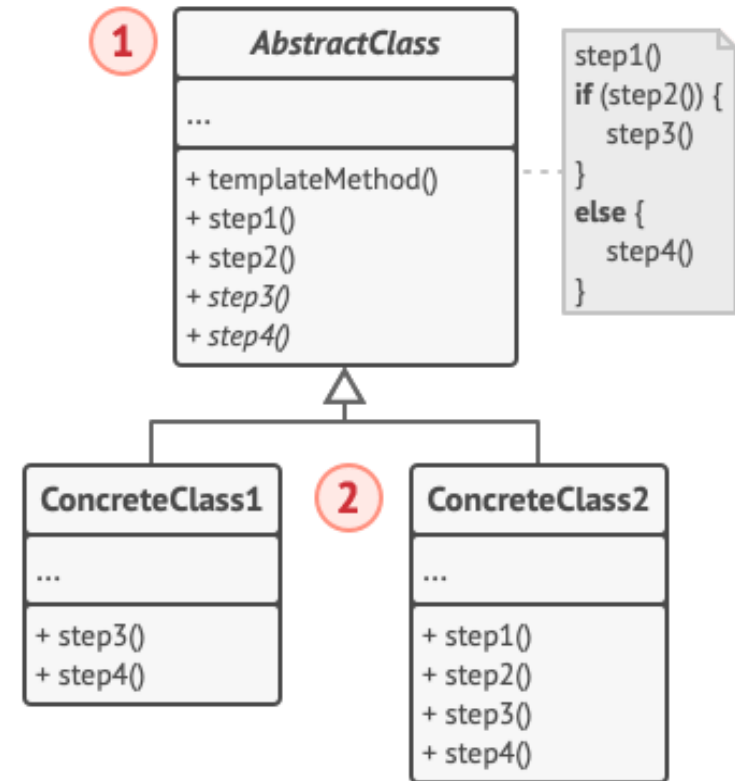
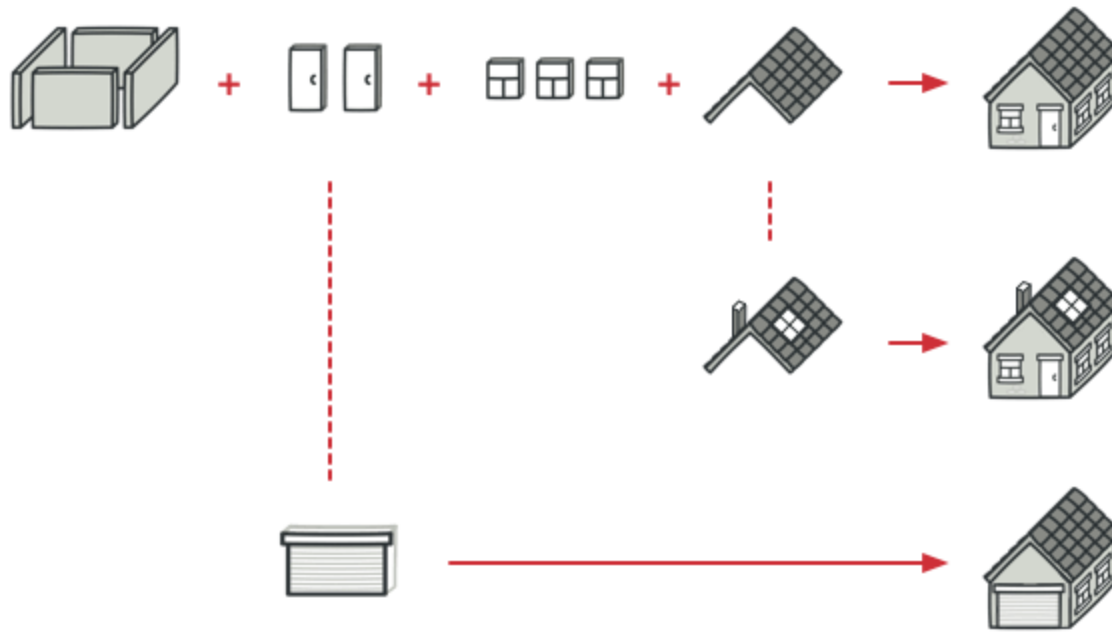
Template

Template Method is a behavioral design pattern that defines the skeleton of an algorithm in the superclass but lets subclasses override specific steps of the algorithm without changing its structure.

The Template Method pattern suggests that you break down an algorithm into a series of steps, turn these steps into methods, and put a series of calls to these methods inside a single *template method*. The steps may either be abstract, or have some default implementation.



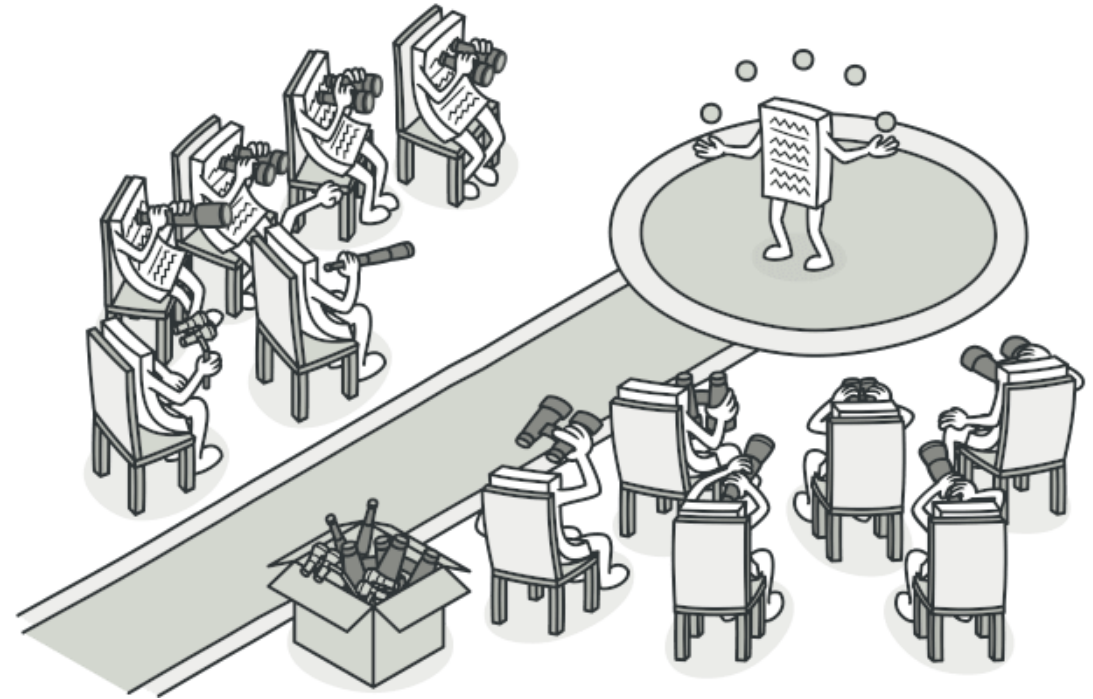
Template



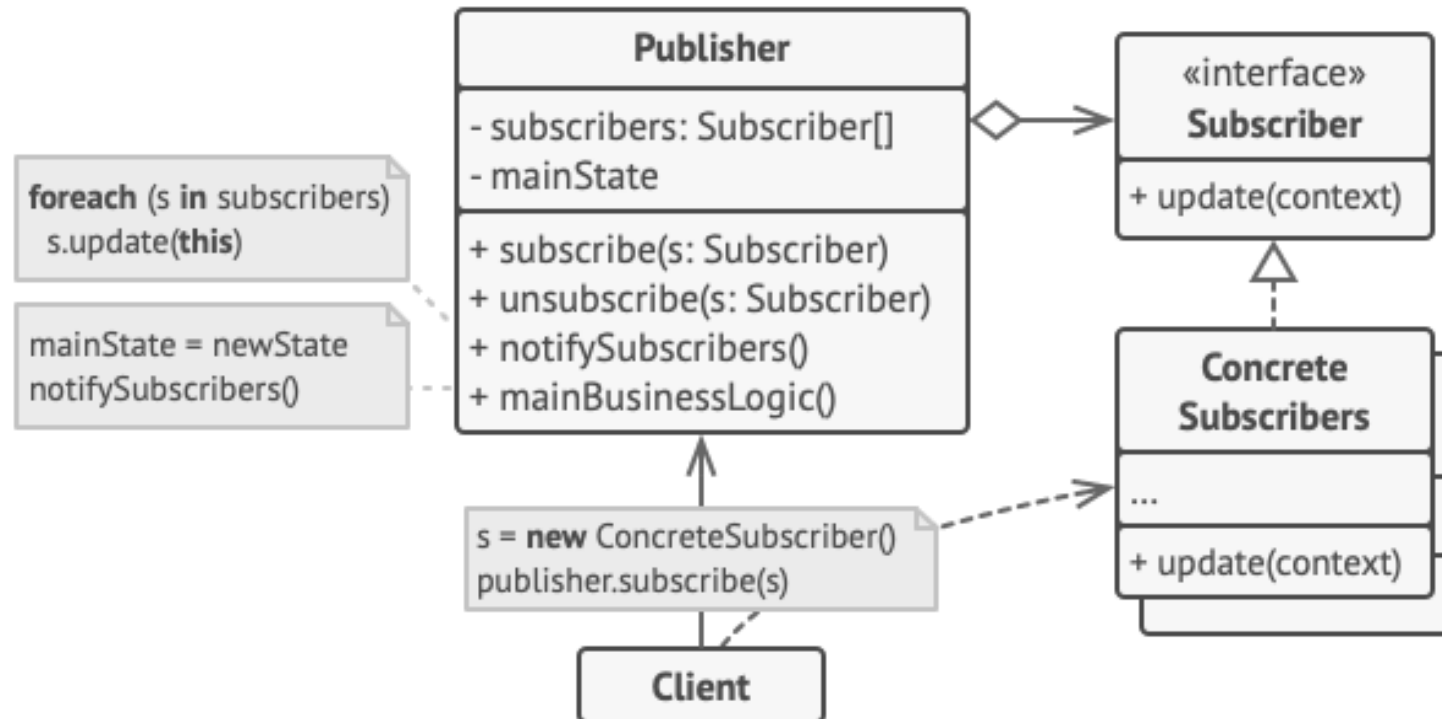
Observer

Observer is a behavioral design pattern that lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they're observing.

The object that has some interesting state and also going to notify other objects about the changes to its state, called *publisher*. All other objects that want to track changes to the publisher's state are called *subscribers*.



Observer



Observer

