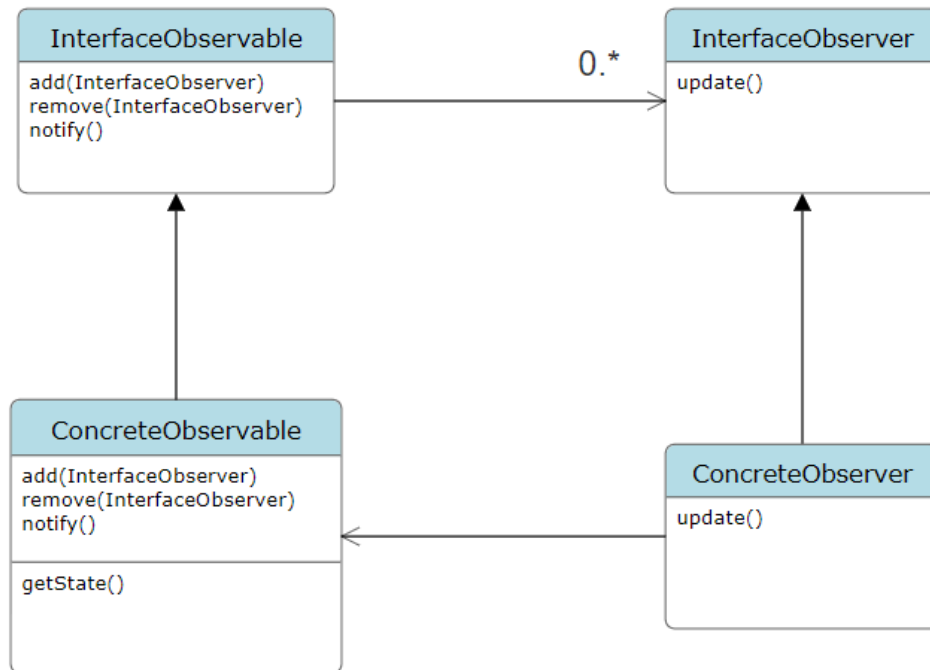


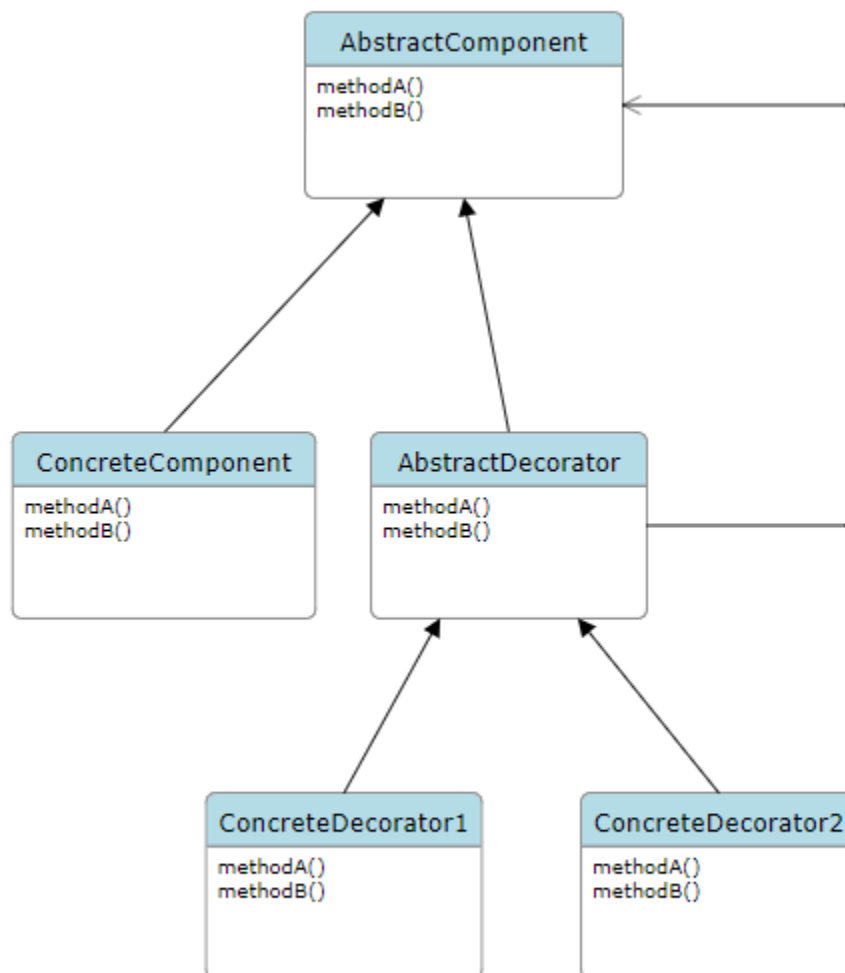
## Observer Pattern

→ The observer pattern defines a one-to-many dependency between objects so that when one object changes state, all dependencies are notified and updated automatically.



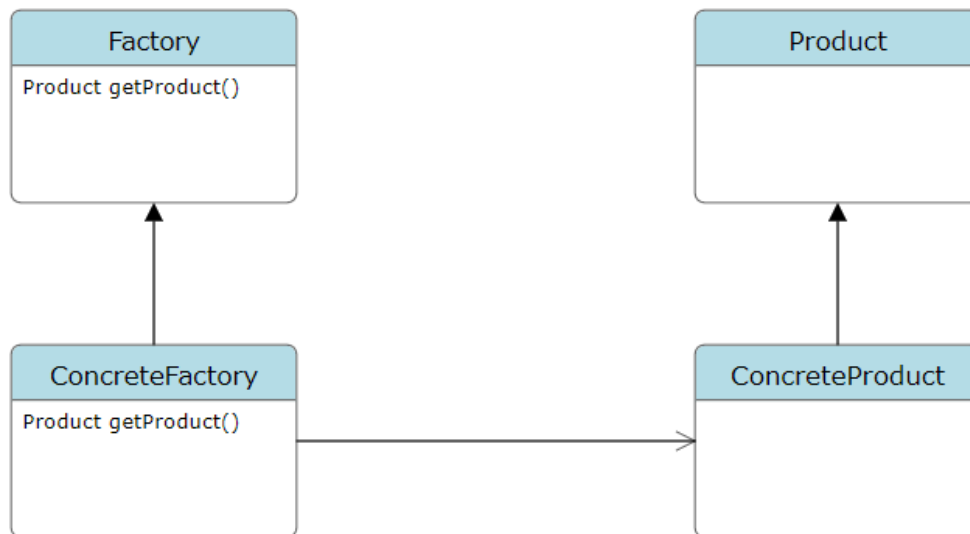
# Decorator Pattern

→ The decorators attach additional responsibility to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.



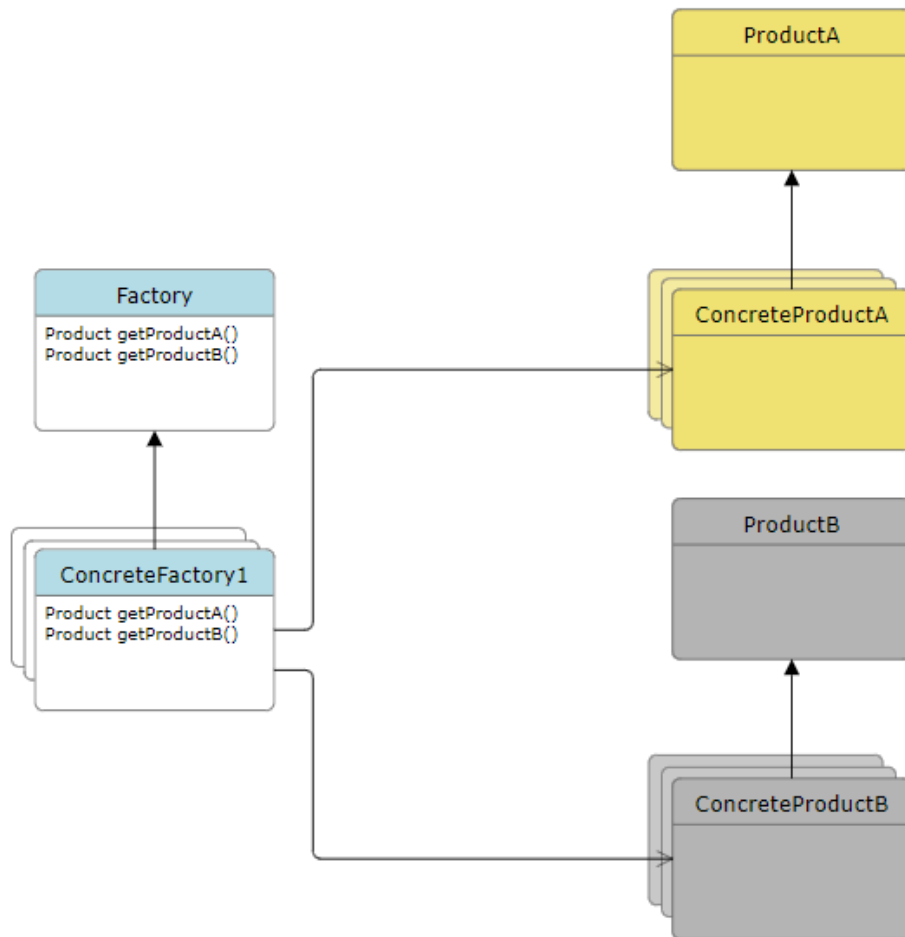
## Simple Factory Pattern

→ The factory method pattern defines an interface for creating the objects but lets the subclasses decide which class to instantiate.



# Abstract Factory Pattern

→ Provide an interface for creating a family of related or dependent objects without specifying their concrete classes.



# Singleton Pattern

→ The singleton pattern ensures that a class has only one instance and provides a global point of access to it.

