

소프트웨어 설계 패턴

IT융합공학부 윤세영

유튜브 주소: <https://youtu.be/hu5yQKG4fMc>

템플릿 메소드 패턴

컴포지트 패턴

스테이트 패턴

컴파운드 패턴

템플릿 메소드 패턴

특정 작업을 처리하는 일부분을 서브 클래스로 캡슐화
전체적인 구조를 바꾸지 않고, 특정 단계에서 수행하는 내용을 바꿈

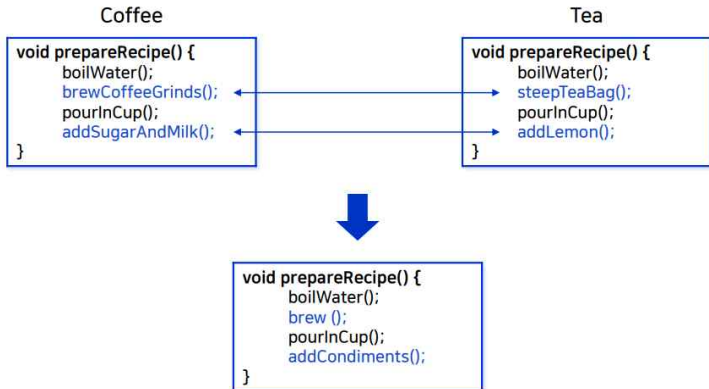
커피 만드는 법

1. 물을 끓인다.
2. 끓는 물에 커피를 우려낸다.
3. 커피를 컵에 따른다.
4. 설탕과 우유를 추가한다.

홍차 만드는 법

1. 물을 끓인다.
2. 끓는 물에 차를 우려낸다.
3. 차를 컵에 따른다.
4. 레몬을 추가한다.

템플릿 메소드 패턴



템플릿 메소드 패턴

장점

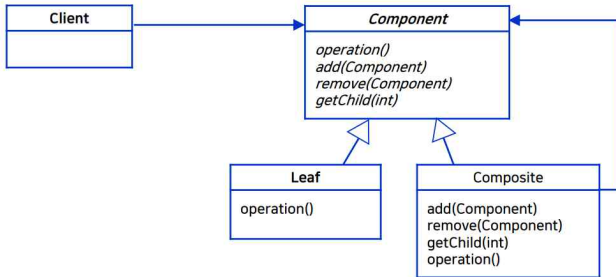
1. 알고리즘 코드 재사용 용이
2. 코드 중복 최소화
3. 핵심 로직 관리 용이

단점

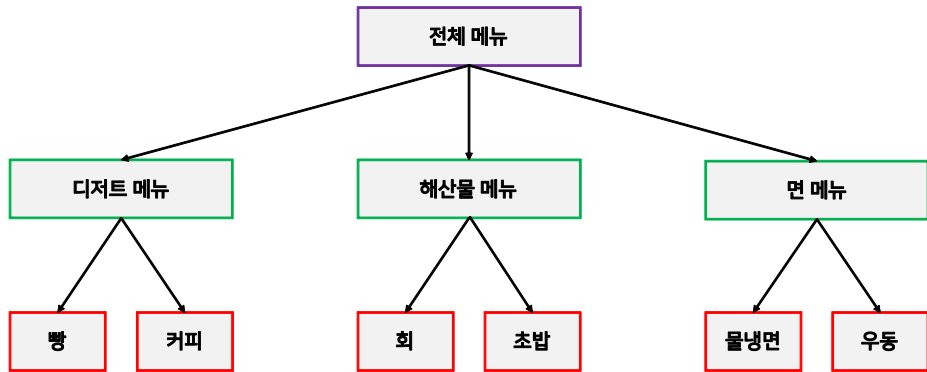
1. 추상 클래스의 증가로 클래스 관리 복잡
2. 상위 클래스 수정시 하위 클래스에도 영향

컴포지트 패턴

객체들을 트리 구조로 구성
부분과 전체를 나타내는 계층구조로 되어 있음



컴포지트 패턴



컴포지트 패턴

장점

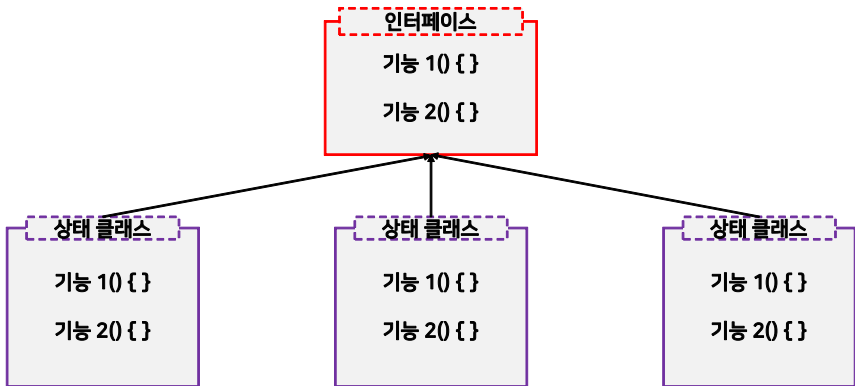
1. 복합 객체를 구성하는 개별적인 객체들을 동일한 방법으로 다룰 수 있음
2. 추가와 수정이 쉬움

단점

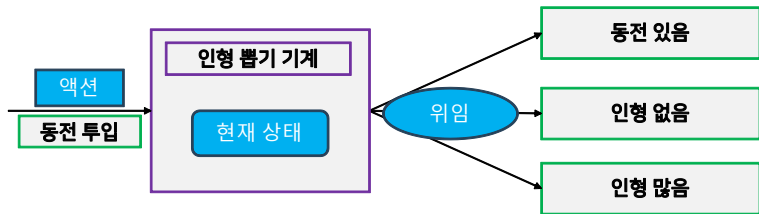
1. 객체가 단순할 경우 오히려 더 복잡하게 만들 수 있음
2. 개별 객체와 복합 객체가 서로 다른 인터페이스를 갖는 경우 사용하기 힘들

스태이트 패턴

객체가 특정 상태에 따라 행위를 달리하는 상황에서,
상태를 객체화 하여 상태가 직접 행동할 수 있도록 위임하는 패턴



스테이트 패턴



스테이트 패턴

장점

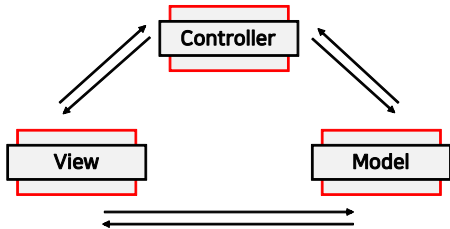
1. 코드 복잡도를 줄임
2. 단일 책임 원칙을 준수함
3. 개방 폐쇄 원칙을 준수함

단점

1. 상태 별로 클래스를 생성하기 때문에 관리 해야 할 클래스 수의 증가
2. 객체에 적용할 상태가 적을 경우 오히려 구조를 복잡하게 만들 수 있음

컴파운드 패턴

반복적으로 생길 수 있는 일반적인 문제를 해결하기 위한 용도로 두 개 이상의 패턴을 결합해서 사용하는 것
단순한 패턴의 결합이 모두 컴파운드 패턴은 아님



컴파운드 패턴(MVC)

장점

1. 모델, 뷰, 컨트롤러의 분업화
2. 변경이 쉬움

단점

1. 상태변화에 따른 처리 속도가 느릴 수 있음
2. 오버헤드가 발생할 수 있음

Q & A