

14. 일관성 있는 협력

0) 서론

잘 설계된 어플리케이션

이해하기 쉽고 수정이 용이하며, 재사용 가능한 협력의 모임

객체지향 설계의 목표

책임을 수행하는 객체들의 협력을 기반으로 결합도가 낮고 재사용 가능한 코드 구조를 만드는 것

객체지향 패러다임의 장점

설계를 재사용할 수 있다

설계를 재사용하기 위해 객체들의 협력 방식을 일관적으로 만들어야 한다

일관성 있는 협력 패턴을 사용 하면 코드가 이해하기 쉽고 직관적이며 수정에 유연해진다

1) 핸드폰 과금 시스템 변경하기

기본 정책 구현

FixedFeePolicy

TimeOfDayDiscountPolicy

4 개의 List 로 조건을 구현

DayOfWeekDiscountPolicy

클래스로 조건을 구현

DurationDiscountPolicy

FixedFeePolicy 를 상속해 재사용으로 조건을 구현

기본 정책 구현이라는 유사한 문제를 해결하지만, 정책을 구현하는 방식이 달라 설계에 일관성이 없다

코드를 이해하기 어렵고 구현을 추가할 때마다 코드의 일관성은 더 어렵는다

객체지향에서 기능 구현하는 방법은 객체의 협력을 만드는 것

협력을 일관성 있게 만들어야 유지보수가 쉬운 시스템을 만든다

2) 설계에 일관성 부여하기