전략 패턴과 커맨드 패턴

1. 전략 패턴
   1. 전략 패턴이란 ? : 객체들이 할 수 있는 행위 각각에 대해 전략 클래스를 생성하고, 유사한 행위들을 캡슐화하는 인터페이스를 정의하여, 객체의 행위를 동적으로 바꾸고 싶은 경우 행위를 수정하는 것이 아닌 전략을 바꿔주는 방식으로 행위를 유연하게 확장하는 방법이다.
   2. 사용하는 이유 : 보통의 인터페이스 방식을 사용하면 기능을 변경을 원할 때 해당 객체의 구현 부분을 수정해주면 된다. 하지만 이런 수정방식은 SOLID(객체지향설계)의 원칙 중 OCP(Open-Close Principle)에 위배된다. OCP에 의하면 기존 객체의 메서드를 수정하지 않으면서 행위가 수정되어야 한다. 이런 방식을 사용하면 메서드를 일일이 수정해야 하고, 같은 메서드를 여러 클래스에서 똑같이 정의하고 있으므로 메서드의 중복이 발생하고 있다. OCP 위배와 메서드의 중복문제를 해결하기 위해 사용한다고 생각하면 된다.
   3. 전략 패턴 구현 : 예) 운송수단의 움직이는 방식
      1. 전략 생성 : 움직이는 두 방식에 대해 Strategy 클래스를 생성하도록 한다. (RailStrategy, LoadStrategy) 두 클래스는 move 메서드를 구현하며 해당 기능에 대해 구현한다. 그리고 두 전략 클래스를 캡슐화하기 위해 MovaleStrategy 인터페이스를 생성한다. 캡슐화하는 이유는 운송수단에 대한 전략뿐만 아니라, 다른 전략들의 추가를 고려한 설계이다.
      2. 운송 수단 클래스 정의 : 운송 수단은 move 메서드를 통해 움직인다. 하지만 이동 방식은 메서드로 구현하지 않고 어떻게 움직일 것인지에 대한 전략을 설정하고, 그 전략의 움직임 방식을 사용하여 움직이도록 한다. 전략을 설정하는 메서드인 setMovableStrategy()가 존재한다.
      3. 객체를 사용하는 Client 구현 : 각 객체를 생성한 이후, 운송 수단이 어떤 방식으로 움직이는지 설정하기 위해 setMovableStrategy 메서드 호출한다. 이때 버스라는 객체가 선로를 따라 움직이도록 바꾸고 싶다면 setMovableStrategy(new RailStrategy())와 같이 사용하면 버스로 선로를 따라 움직이는 방식으로 전략을 수정하여 사용하게 될 것이다.
2. 커맨드 패턴
   1. 커맨드 패턴이란? : 객체의 행위를 클래스로 만들어 캡슐화하는 패턴이다. 어떤 객체에서 다른 객체의 메서드를 실행하려면 그 객체를 참조하고 있어야하는 의존성이 발생한다. 그러나 커맨드 패턴을 적용하면 의존성 문제를 해결할 수 있다. 또한 기능이 수정되거나 변경이 일어날 때 A 클래스 코드를 수정없이 기능에 대한 클래스를 정의하면 되므로 시스템의 확장성이 유연해진다.
   2. 커맨드 패턴을 사용하는 이유 : 예로 구글홈을 생각해보자. 구글홈 내에서 히터와 램프의 기능을 조작하기 위해서는 해당 객체들의 참조가 필요할 것이고 구글홈의 기능이 늘어날수록 객체 프로퍼티는 더욱 늘어날 것이고, 상호작용 메서드에서의 분기도 점점 늘어날 것이다. 이는 OCP에 위배된다.
   3. 커맨드 패턴 적용 : 먼저 구글홈이 할 수 있는 기능들을 클래스로 만들어 각 기능들을 캡슐화한다(HeaterOnCommand, LampOnCommand) 그리고 구글홈 클래스의 상호작용 메서드에서 각 기능을 직접 호출하지 않고, 캡슐화한 Command 인터페이스의 메서드를 호출하도록 한다.
3. 전략패턴과 커맨드패턴의 차이
   1. 전략패턴은 ‘어떻게’ 를 우선시한다!

하고자 하는 것은 이미 정해져 있고, 방법을 어떻게 할지에 대한 유연성을 고려하며 구현한다. 인터페이스의 메서드에 직접적으로 의존을 하게 되어서, 해당 메소드의 매개변수들에 강하게 영향을 받는다.

* 1. 커맨드패턴은 ‘무엇’ 을 초점에 둔다!

어떻게 할지에 대한 방법은 외부에서 정의하며 주입하며, 그것을 실행하는 것이 중요하기 때문이다. 무엇을 할지를 선택하면 해당 행위를 하기 위한 필요한 변수까지 같이 캡슐화 하여 제공받기 때문에 유연하며, 영향을 받는 객체도 같이 제공을 받기 때문에 행위에 따른 영향을해당 객체도 조건에 따라 다르게 설정할 수 있다.