1. **아키텍처 설계에서 클래스 수준의 설계시 자주 사용되는 아래의 디자인패턴 각각에 대해 적용 사례를 들어 자세히 설명하시오.**

1) 팩토리 메소드

팩토리 메소드는 생성패턴으로 객체 생성을 서브 클래스에서 처리하도록 분리하여 캡슐화한 패턴이다. 상위 클래스에서 인터페이스만 정의하고 실제 생성은 서브 클래스가 담당한다.

2) 추상 팩토리

추상팩토리는 생성패턴으로 구체적인 클래스에 의존하지 않고 인터페이스를 통해 서로 연관, 의존하는 객체들의 그룹으로 생성하여 추상적으로 표현한다, 연관된 서브클래스를 묶어 한 번에 교체하는 것이 가능하다.

3) 어댑터

어댑터는 구조패턴으로 호환성ㄹ이 없는 클래스들의 인터페이스를 다른 클래스가 이용할 수 있도록 변환해주는 패턴이다. 기존의 클래스를 이용하고 싶지만 인터페이스가 일치하지 않을 떄 이용한다.

4) 싱글톤

싱글톤은 생성패턴으로 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만 여러 프로세스가 동시에 참조할 수 는 없다. 클래스 내에서 인스턴스가 하나뿐임을 보장하며 불필요한 메모리 낭비를 최소화 한다.

5) 컴포지트

컴포지트는 구조패턴으로 여러 객체를 가진 복합 객체와 단일 객체를 구분 없이 다루고자 할 때 사용하는 패턴이다. 객체들을 트리 구조로 구성하여 디렉터리 안에 디렉터리가 읶ㅆ듯이 복합객체 안에 복합 객체가 포함되는 구조를 구현할 수 있다.

6) 옵서버

옵서버는 행위패턴으로 한 객체의 상태가 변화하면 객체에 상속되어 있는 다른 객체들에게 변화된 상태를 전달하는 패턴이다.

7) 반복자

반복자는 행위패턴으로 자료 구조와 같이 접근이 잦은 객체에 동일한 인터페이스를 사용하도록 하는 패턴이다.