

OrderServiceImpl 에서 오더 서비스와 관련된 로직만 해야되는 데 discountPolicy 구현체를 직접 선택하고 있음. 구체적인 것까지 serviceimpl이 선택하고 생성하고 있음. -. 배우가 여자주인공을 직접 데려와서 섭외하는 것까지 하는 것과 비슷함.

: 관심사를 분리할 필요가 있음. 배우(구현체)는 대본만 연습해서 상대 배우가 바뀌더라도 대본에만 맞춰서 공연을 수행할 수 있어야한다.

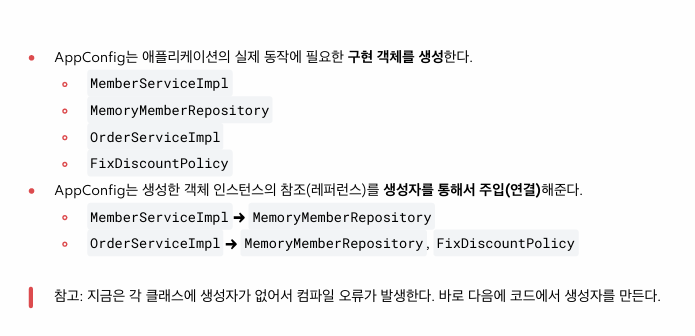
실제 실행되는 객체들은 본인의 역할만 수행하게 해주고, 어떤 구현체가 들어갈지 선택하는 것은 공연 기획자가 해줘야한다.

: 앱컨피그.



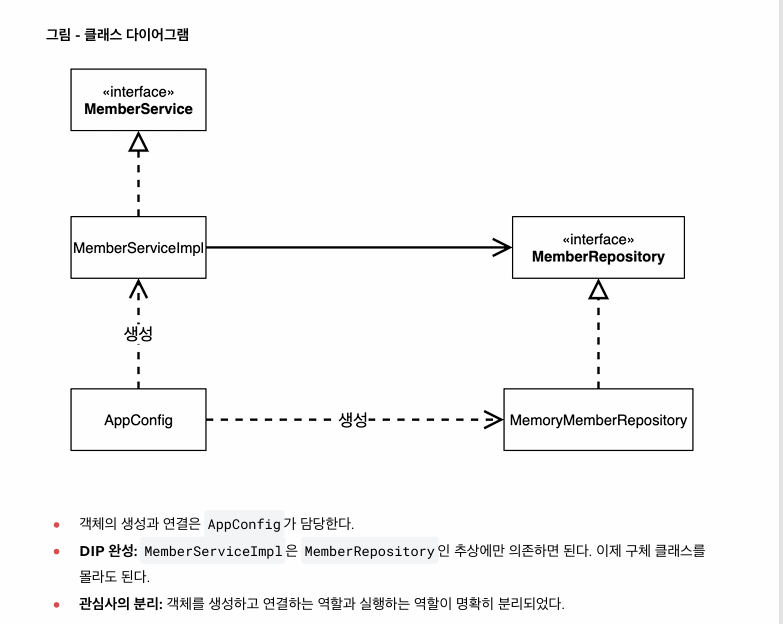


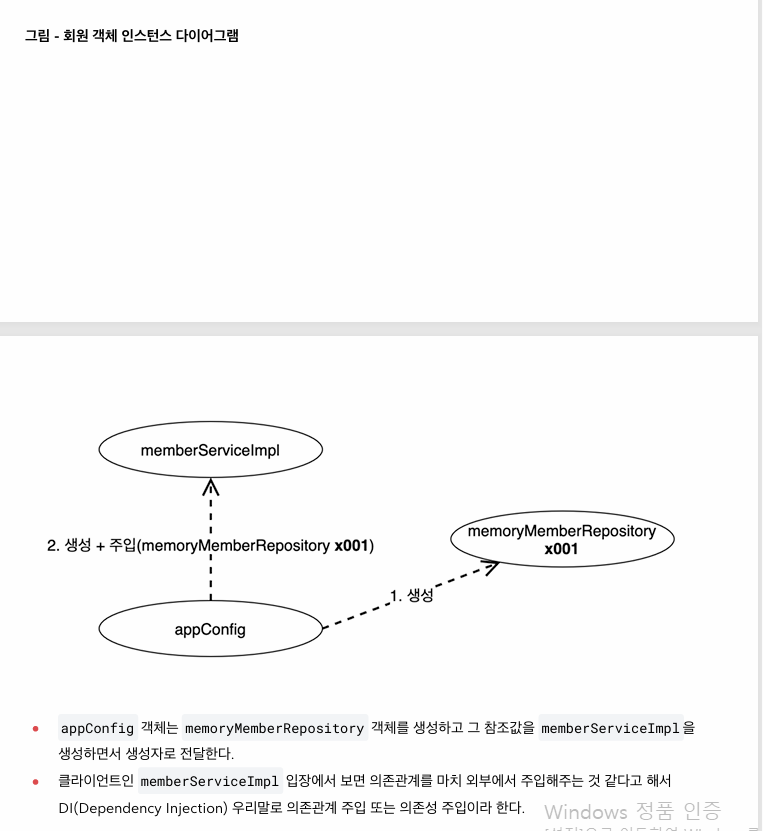
MemberServiceImpl에 memberRepository 생성을 위한 생성자만 만들고, AppConfig에서 MemberRepository 생성을 해준다. MemberServiceImpl은 구현체에 의존하지 않고, 인터페이스에만 의존하고 있다. 즉 DIP를 지키고 있음.



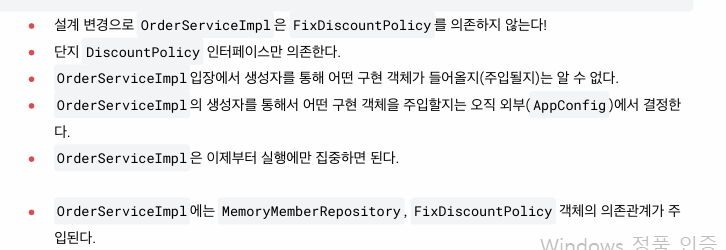


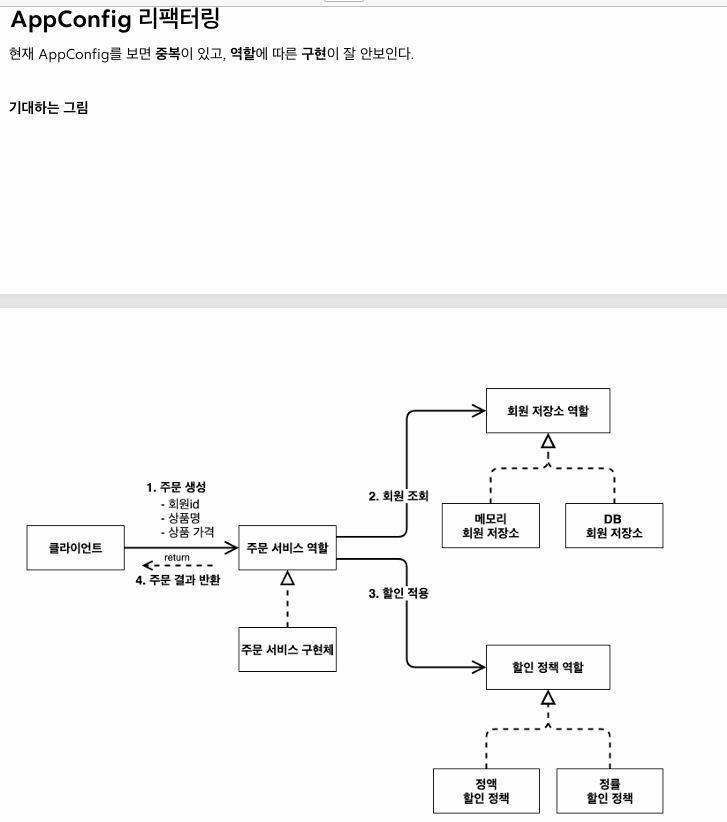




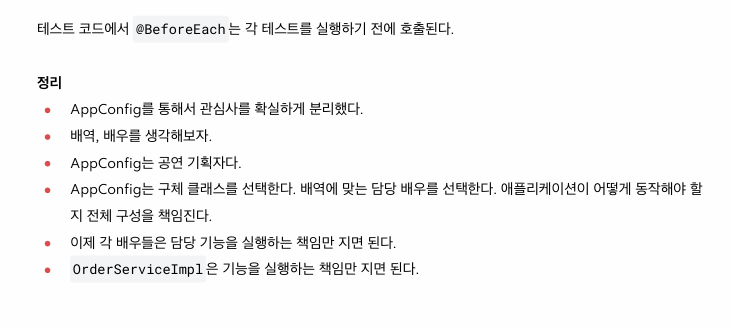




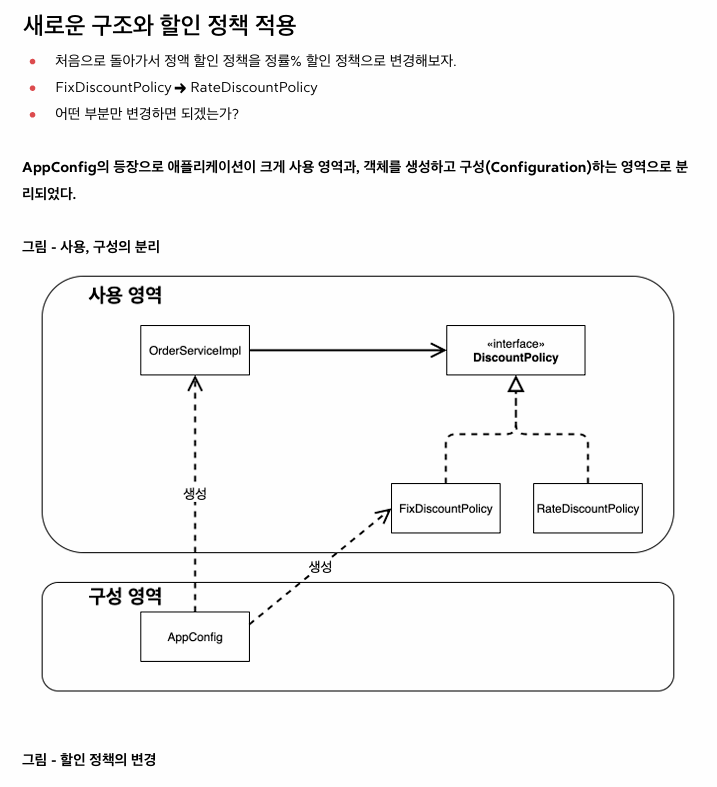


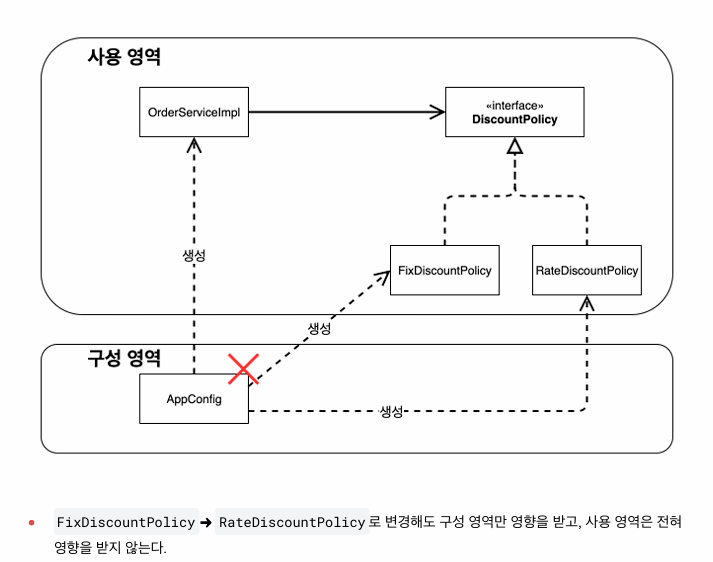


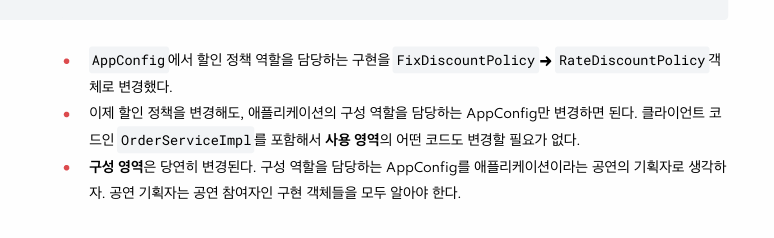


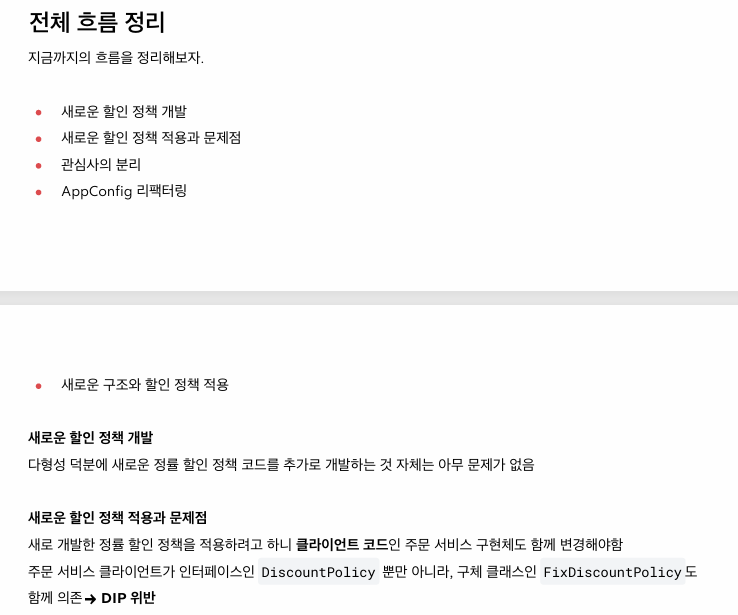


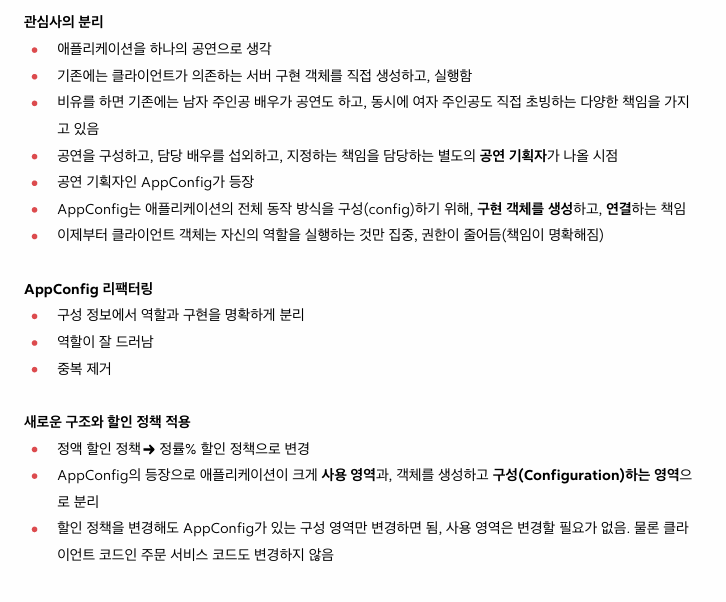


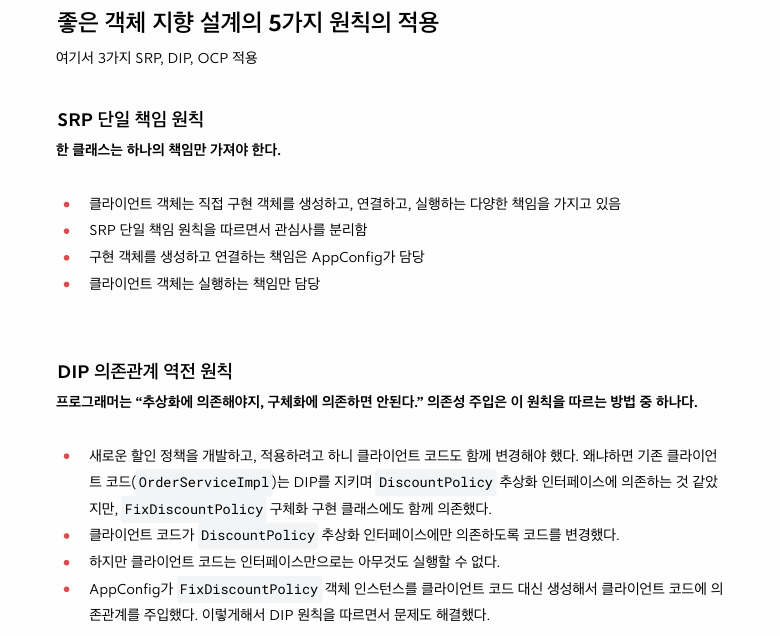












OrderServiceImpl 과 같은 객체는 직접 구현 객체를 생성하고 연결하고 실행하는 굉장히 다양한 책임을 가지고 있었음.(SRP 위반)

AppConfig로 객체 생성 민 의존 관계 주입 역할을 분리하면서 코드 변경의 유연성 확보 가능.

