

스프링핵심원리-예제만들기

2021년 8월 3일 화요일 오후 4:10

비즈니스 요구사항과 설계

회원

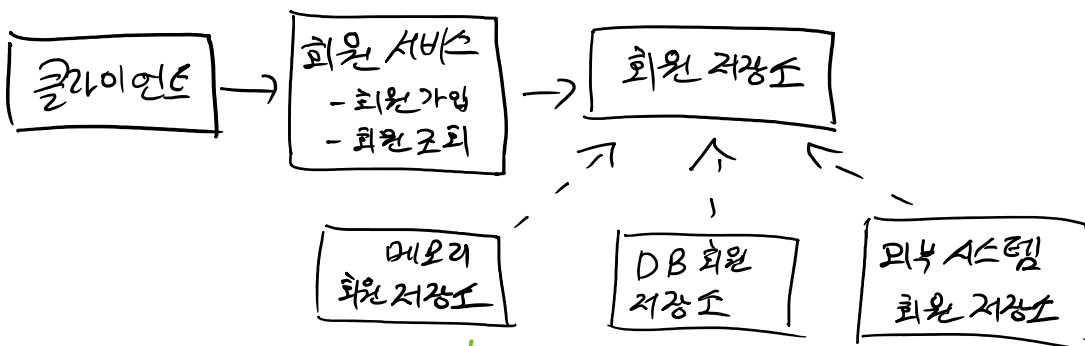
- 회원 가입, 조회
- 일반과 vip 등급
- 회원 데이터는 자체 db, 외부 시스템 연동 (미확)

주문과 할인 정책

- 회원은 상품주문
- 회원 등급에 따라 할인 정책을 적용
- vip는 천원 할인해주는 고정 금액할인 (변경가능해야)
- 할인 정책은 변경가능성 높음, 할인을 아예 안해버릴수도?

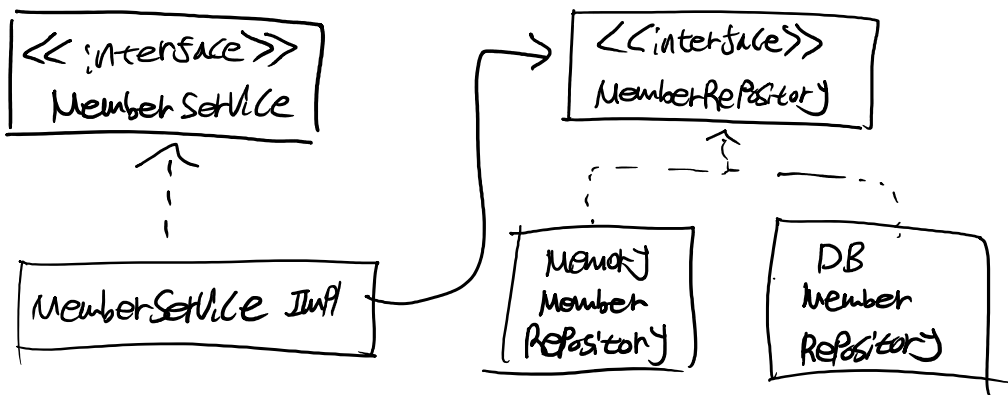
스프링없는 순수한 자바로만 개발해보는 버전으로 진행할거임

회원 도메인 설계

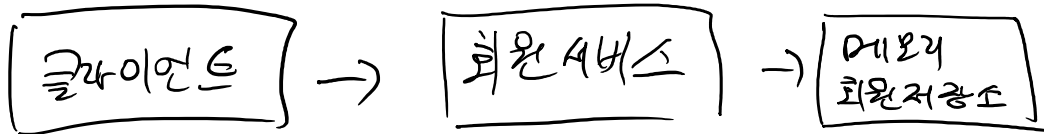


회원 클래스 다이어그램

개념용, 나중에 확정되면 갈아끼울 계획



회원 객체 다이어그램



회원서비스 : MemberServiceImpl

문제점:

다른 저장소로 변경할 때 OCP가 지켜지는가?

DIP를 지키고 있는가?

의존관계가 인터페이스뿐만이 아니라 구현체에도 의존 (DIP 위반)

```

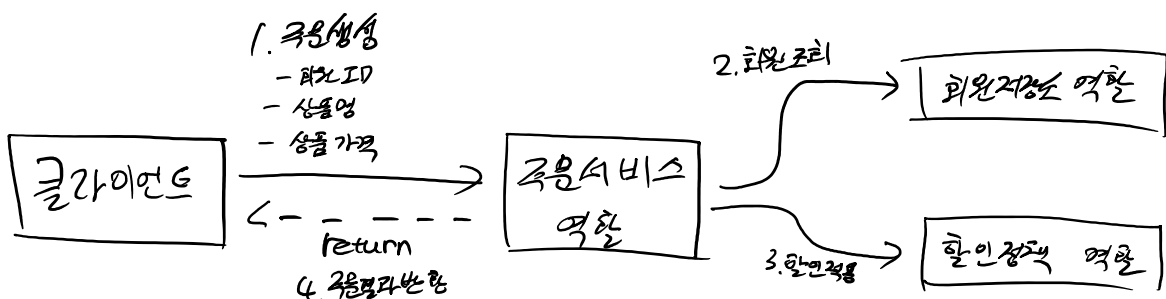
public class MemberServiceImpl implements MemberService {
    private final MemberRepository memberRepository = new MemoryMemberRepository();
  }
  
```

Handwritten annotations: A green arrow points from MemberRepository to MemberService, labeled "<<interface>>". Another green arrow points from new MemoryMemberRepository() to MemberRepository, labeled "구현체".

주문과 할인 도메인 설계

주문과 할인 정책

- 회원은 상품 주문
- 회원 등급에 따라 할인 정책 적용 가능
- 할인 정책은 모든 vip는 1000원 고정금액 할인 (변경가능)
- 할인 정책 변경 가능성 높음, 아예 안할 수도 + 결정이 오픈직전까지 미정인 상황



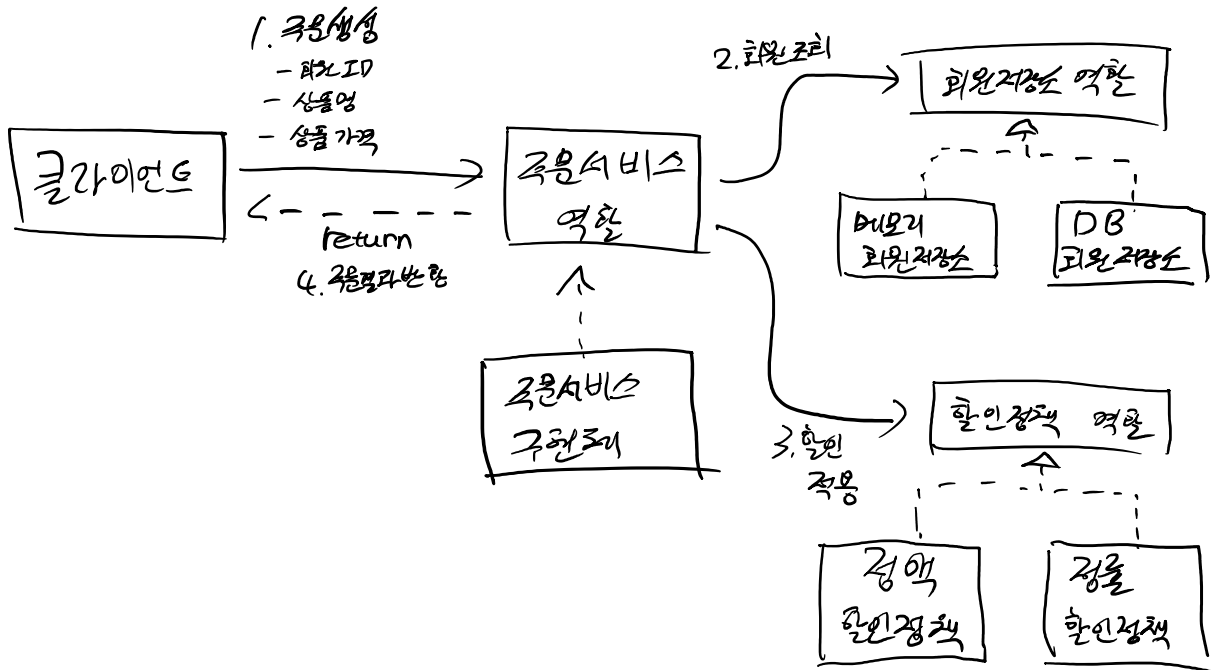
1. 주문 생성 : 클라이언트가 주문서비스에 주문생성 요청

2. 회원 조회 : 할인을 위해서는 회원 등급이 필요하다 그래서 주문 서비스는 회원 저장소에

서 회원 조회

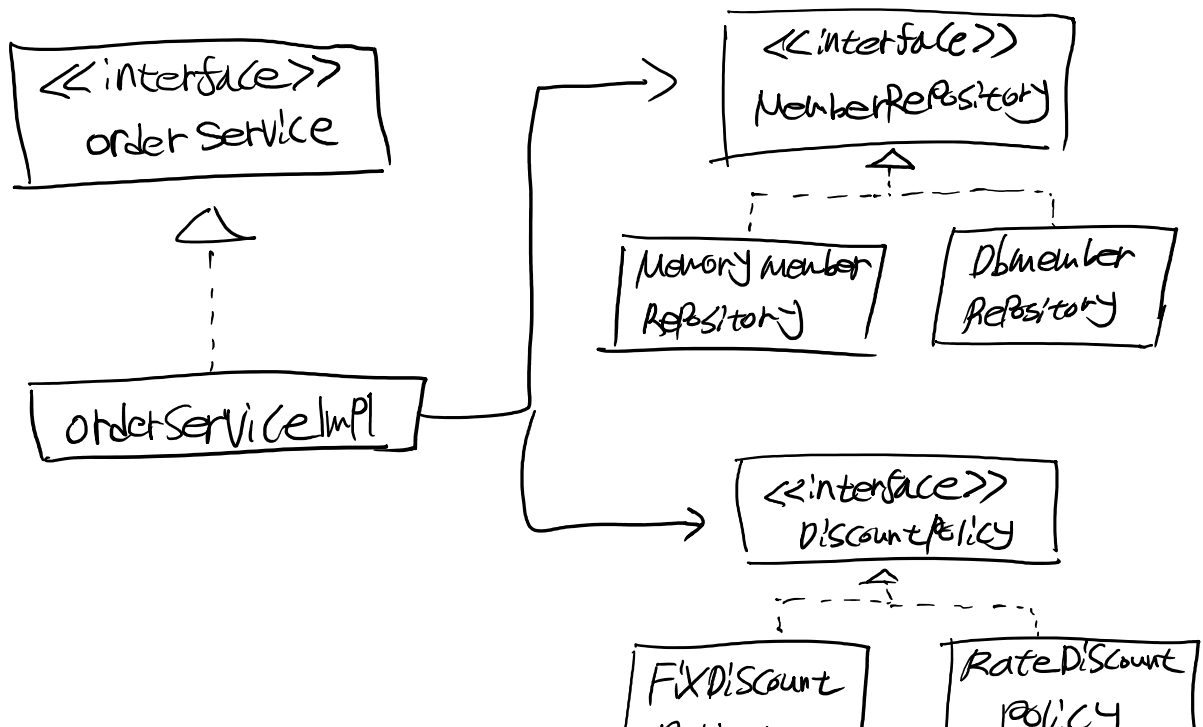
3. 할인 적용 : 주문 서비스는 회원 등급에 따른 할인 여부를 할인 정책에 위임
4. 주문 결과 반환 : 주문 서비스는 할인 결과를 포함한 주문 결과를 반환한다

ps: 주문 데이터 DB에 저장하는 과정 생략, 주문 아이템화 생략



역할과 구현 분리

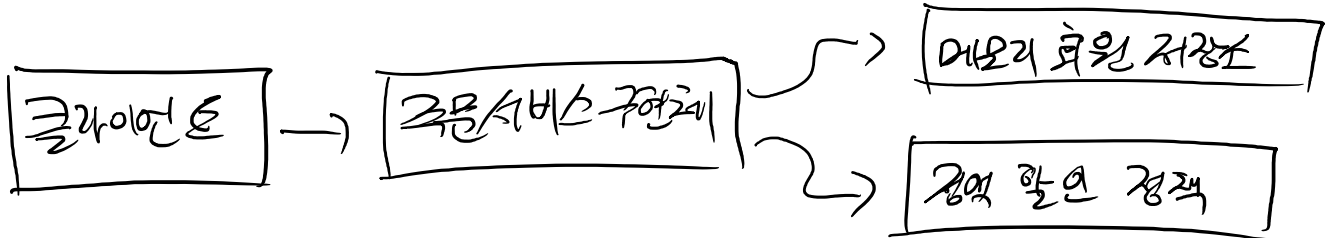
주문 도메인 클래스 다이어그램



FixDiscount
Policy

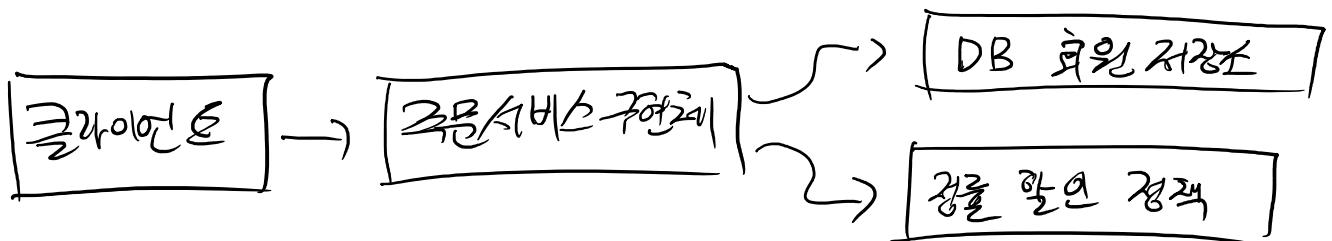
RateDiscount
Policy

주문 도메인 객체 다이어그램1 (실제로 객체 만들어서 동적으로 이뤄지는)



회원정보 메모리에서 조회, 정액 할인 정책 지원해도 주문서비스 변경X
역활의 협력관계 그대로 사용 가능

주문 도메인 객체 다이어그램2



회원정보 DB에서 조회, 정액 할인 정책 지원해도 주문서비스 변경X
역활의 협력관계 그대로 사용 가능