

[문제 1] 다음 조건을 만족하는 프로그램을 Spring Java Config 방식으로 작성하시오

1. 요구사항

- ① MemberRepository, DiscountPolicy, OrderService 인터페이스를 정의하고 각각의 구현체를 만든다.
- ② AppConfig 클래스를 생성하고, 자바 기반 설정으로 Bean 을 등록한다.
- ③ MainApp 클래스에서는 AnnotationConfigApplicationContext 를 이용해 설정 파일을 로딩하고, OrderService 빈을 호출한다.
- ④ 생성자 주입 방식으로 DI 를 구성한다

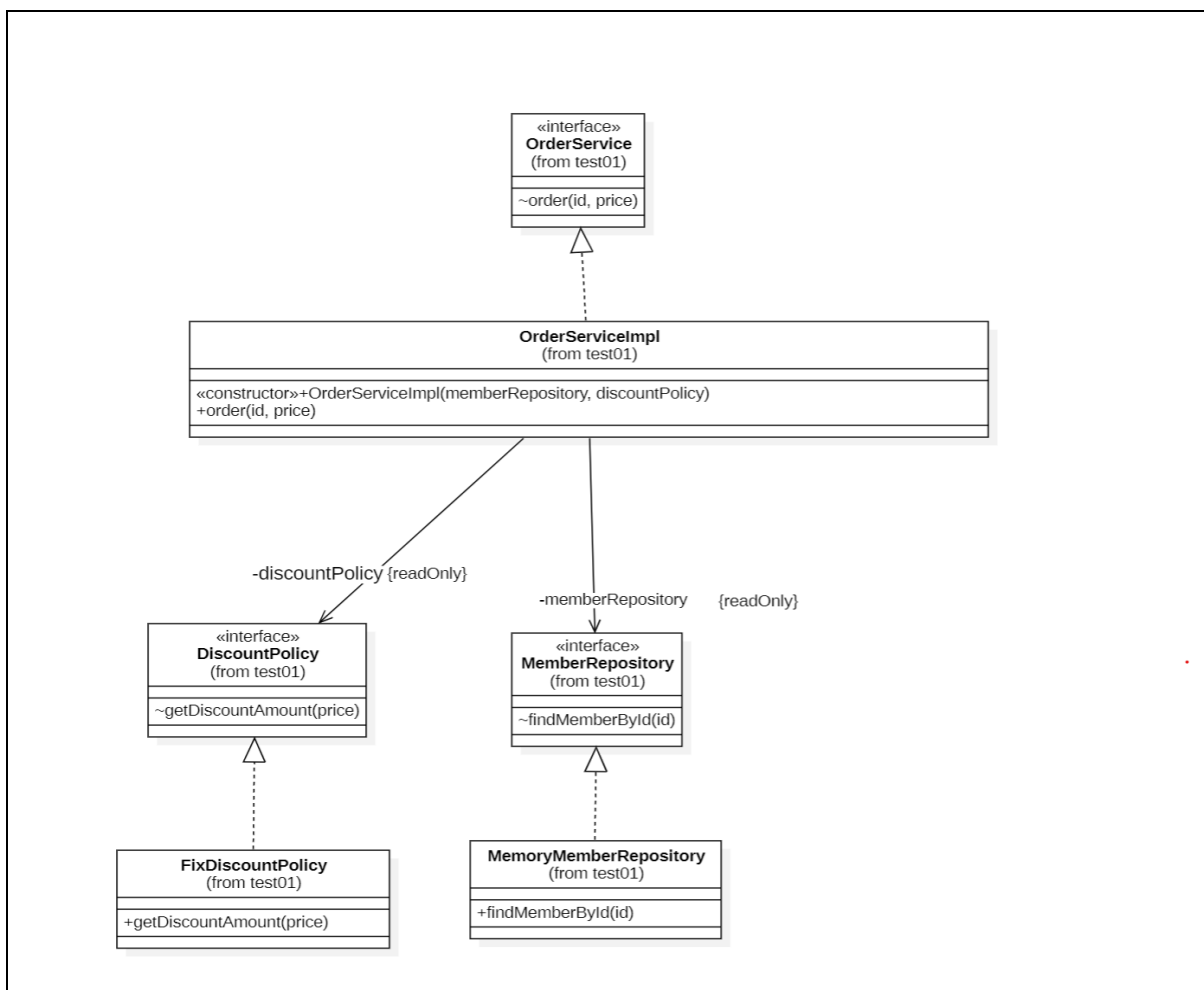
2. 클래스/메서드 요약

| 클래스명 | 메서드 | 설명 |
|------------------------|--|---|
| MemberRepository | + findMemberById(int): String | 회원 ID 로 회원 이름 조회 (interface) |
| MemoryMemberRepository | + findMemberById(int): String | 메모리 기반 회원 저장소 구현체 |
| DiscountPolicy | + getDiscountAmount(int): int | 주문 금액에 대한 할인 금액 반환 (interface) |
| FixDiscountPolicy | + getDiscountAmount(int): int | 고정 금액 할인 제공 |
| OrderService | + order(int id, int price): String | 회원 ID 와 가격을 받아 할인 처리 후 주문 정보 문자열 리턴 |
| OrderServiceImpl | + order(int id, int price): String | 생성자 주입 방식으로 주입받은 의존 객체를 통해 로직 수행 |
| AppConfig | + memberRepository(): MemberRepository | Bean 등록 메서드 |
| | + discountPolicy(): DiscountPolicy | Bean 등록 메서드 |
| | + orderService(): OrderService | Bean 등록 메서드 |
| MainApp | + main(String[] args): void | AnnotationConfigApplicationContext 로 설정 파일 실행 |

3. 실행 결과

회원: 홍길동
주문 금액: 10000 원
할인 금액: 1000 원
최종 결제 금액: 9000 원

4. Class Diagram



[문제 2] 조건에 따른 출력구문을 작성 하시오

1. 요구 사항

- ① [문제 1]의 AppConfig 에서 등록된 DiscountPolicy 를 FixDiscountPolicy 에서 RateDiscountPolicy 로 **교체**한다.
- ② OrderServiceImpl 클래스에는 **변경 사항 없이**, AppConfig 설정만으로 구현체를 유연하게 교체할 수 있어야 한다.

2. 클래스/메서드 요약

| 클래스명 | 메서드 | 설명 |
|--------------------|------------------------------------|--------------------------------|
| DiscountPolicy | + getDiscountAmount(int): int | 주문 금액에 대한 할인 금액 반환 (interface) |
| FixDiscountPolicy | + getDiscountAmount(int): int | 고정 금액 1000 원 할인 구현 |
| RateDiscountPolicy | + getDiscountAmount(int): int | 10% 비율 할인 구현 |
| OrderServiceImpl | + order(int id, int price): String | 할인 정책에 따라 결제 금액 반환 |
| AppConfig | + discountPolicy(): DiscountPolicy | 구현체 교체만으로 전체 흐름 유지 가능 |

3. 실행 결과

회원: 홍길동
주문 금액: 10000 원
할인 금액: 1000 원
최종 결제 금액: 9000 원

[문제 3] 조건에 따른 출력구문을 작성 하시오

1. 요구 사항

- ① MemberRepository, DiscountPolicy, OrderService 인터페이스와 구현체를 정의한다.
- ② 자바 코드에는 @Configuration, @Bean 을 사용하지 않고, Groovy 기반 Bean 설정 파일 (application.groovy)을 사용하여 DI 를 구성한다.
- ③ GenericGroovyApplicationContext 를 통해 컨텍스트를 로딩하고 주문 서비스를 실행한다.

2. 클래스/메서드 요약

| 클래스명 | 메서드 서명 | 설명 |
|------------------------|--|-------------------------------|
| MemberRepository | findMemberById(int id): String | 회원 ID 를 받아 회원 이름을 반환하는 인터페이스 |
| MemoryMemberRepository | findMemberById(int id): String | 메모리에 저장된 회원 정보를 반환하는 구현체 |
| DiscountPolicy | getDiscountAmount(int price): int | 가격을 받아 할인 금액을 반환하는 인터페이스 |
| FixDiscountPolicy | getDiscountAmount(int price): int | 고정 할인 금액(예: 1000 원)을 반환하는 구현체 |
| OrderService | order(int id, int price): String | 주문을 처리하고 결과 메시지를 반환하는 인터페이스 |
| OrderServiceImpl | OrderServiceImpl(MemberRepository, DiscountPolicy) | 생성자 주입을 통해 의존성 설정 |
| | order(int id, int price): String | 회원 ID 와 가격을 받아 주문 로직 처리 |

3. 실행 결과

| |
|------------------|
| 회원: 홍길동 |
| 주문 금액: 10000 원 |
| 할인 금액: 1000 원 |
| 최종 결제 금액: 9000 원 |

4. Groovy Bean 설정 (application.groovy)

```
beans {  
    memberRepository(com.example.MemoryMemberRepository)  
    discountPolicy(com.example.FixDiscountPolicy)  
    orderService(com.example.OrderServiceImpl, memberRepository, discountPolicy)  
}
```

5. Main 클래스

```
public class MainApp {  
    public static void main(String[] args) {  
        ApplicationContext context = new GenericGroovyApplicationContext  
("classpath:application.groovy");  
        OrderService service = context.getBean("orderService", OrderService.class);  
        System.out.println(service.order(1, 10000));  
    }  
}
```

