



스프링 핵심 원리

발표자: 이청수

강의 내용 요약

- 스프링이란 무엇인가?
- 스프링의 핵심은 무엇인가?
- 좋은 객체 지향 프로그래밍이란 무엇인가?
- 스프링과 객체 지향의 접점은 무엇인가?

스프링이란 무엇인가?

EJB(겨울)를 넘어 새로운 시작(봄)이라는 뜻
“자바 생태계에서 가장 대중적인 어플리케이션 개발 프레임워크”



스프링이란 무엇인가?



스프링 프레임워크

- **핵심 기술:** 스프링 DI 컨테이너, AOP, 이벤트, 기타
- **웹 기술:** 스프링 MVC, 스프링 WebFlux
- **데이터 접근 기술:** 트랜잭션, JDBC, ORM 지원, XML 지원
- **기술 통합:** 캐시, 이메일, 원격접근, 스케줄링
- **테스트:** 스프링 기반 테스트 지원
- **언어:** 코틀린, 그루비
- 최근에는 스프링 부트를 통해서 스프링 프레임워크의 기술들을 편리하게 사용

스프링이란 무엇인가?

스프링 부트



스프링을 편리하게 사용할 수 있도록 지원하는
스프링 프레임워크의 서브 프로젝트

단독으로 실행 가능한 스프링 어플리케이션을 쉽게 생성
(웹서버 내장)

스프링의 핵심이란 무엇인가?

스프링의 핵심은 자바라는 객체지향 언어의 장점을 살려서,
좋은 객체지향 어플리케이션을 개발할 수 있게 도와주는 것!

객체 지향 언어의 장점

유지/보수가 쉽도록 프로그램을 유연하게 만드는 것

- **악덕 기획자:** 서비스 오픈 직전에 할인 정책을 지금처럼 고정 금액 할인이 아니라 좀 더 합리적인 주문 금액 당 할인하는 정률% 할인으로 변경하고 싶어요. 예를 들어서 기존 정책은 VIP가 10000원을 주문하든 20000원을 주문하든 항상 1000원을 할인했는데, 이번에 새로 나온 정책은 10%로 지정해두면 고객이 10000원 주문시 1000원을 할인해주고, 20000원 주문시에 2000원을 할인해주는 거예요!
- **순진 개발자:** 제가 처음부터 고정 금액 할인은 아니라고 했잖아요.
- **악덕 기획자:** 애자일 소프트웨어 개발 선언 몰라요? "계획을 따르기보다 변화에 대응하기를"

실제 현업에서는 스펙이 바뀌는 경우가 흔함😱

객체 지향 프로그래밍의 특징

추상화

복잡한 자료, 모듈, 시스템 등으로부터 핵심적인 개념 또는 기능을 간추려 내는 것

캡슐화

객체의 속성(data fields)과 행위(메서드, methods)를 하나로 묶고, 실제 구현 내용 일부를 외부에 감추어 은닉함

상속

객체들 간의 관계를 구축하는 방법

다형성

객체 지향 특징 - 다형성

객체를 **역할**과 **구현**으로 구분하는 것

객체 지향 특징 - 다형성 (예시)



편의점 알바

역할: 물건 창고에 넣기, 음료수 채우기, 손님 응대하기 등등

역할만 충실하게 수행하면, 구현은 알바 마음대로



편의점 알바A

구현: 정확히 오후 9시에 박스를 창고에 옮긴다. 10분뒤에 음료수를 채운다 등등



편의점 알바B

구현: 퇴근하기 5분전에 박스를 창고에 옮기고 음료수를 채운다 등등

물론 이렇게 하다간 짤릴수도 있음

객체 지향 특징 - 다형성

객체를 **역할**과 **구현**으로 구분하는 것

- > 알바를 컴퓨터 부품 같아 끼우듯이 유연하게 변경할 수 있음👍
- > 다만, 현실이랑 다르게 역할은 한번 정하면 변경이 어려움
- > 역할(인터페이스)을 처음부터 잘 설계해야 한다는 한계점

- **악덕 기획자:** 서비스 오픈 직전에 할인 정책을 지금처럼 고정 금액 할인이 아니라 좀 더 합리적인 주문 금액 당 할인하는 정률% 할인으로 변경하고 싶어요. 예를 들어서 기존 정책은 VIP가 10000원을 주문하든 20000원을 주문하든 항상 1000원을 할인했는데, 이번에 새로 나온 정책은 10%로 지정해두면 고객이 10000원 주문시 1000원을 할인해주고, 20000원 주문시에 2000원을 할인해주는 거예요!
- **순진 개발자:** 제가 처음부터 고정 금액 할인은 아니라고 했잖아요.
- **악덕 기획자:** 애자일 소프트웨어 개발 선언 몰라요? "계획을 따르기보다 변화에 대응하기를"

역할: 정해진 조건(VIP)에 따라 할인을 적용하기

구현: 무조건 1000원을 할인해주는 정액 할인

구현: 무조건 10%를 할인해주는 정률 할인

```

public interface DiscountPolicy {

    /**
     * @return 할인 대상 금액
     */
    int discount(Member member, int price);
}

```

역할: 정해진 조건(VIP)에 따라 할인을 적용하기

```

public class FixDiscountPolicy implements DiscountPolicy {

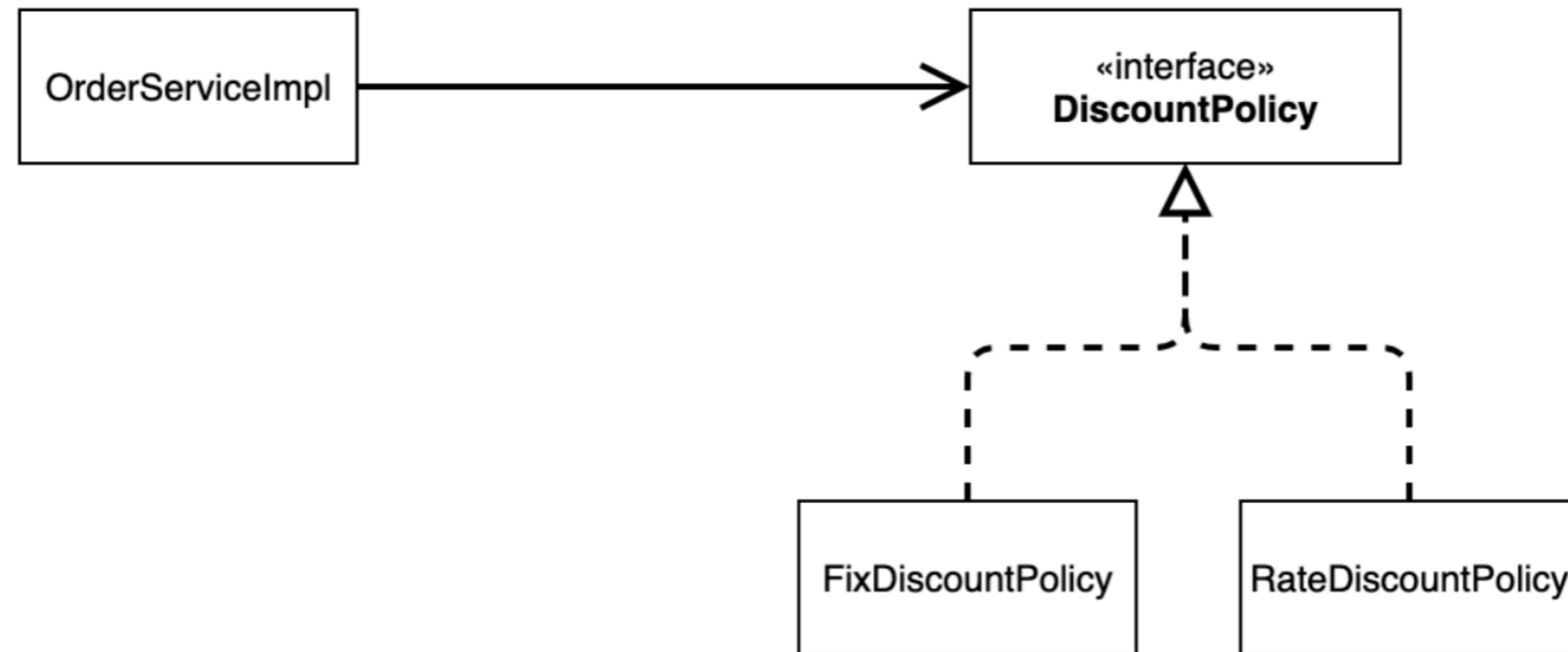
    private int discountFixAmount = 1000; //1000원 할인

    @Override
    public int discount(Member member, int price) {
        if (member.getGrade() == Grade.VIP) {
            return discountFixAmount;
        } else {
            return 0;
        }
    }
}

```

구현: 무조건 1000원을 할인해주는 정액 할인

객체 지향 특징 - 다형성



객체를 **역할**과 **구현**으로 구분하는 것

-> 컴퓨터 부품 같아 끼우듯이 유연하게 변경할 수 있음👍

스프링의 핵심이란 무엇인가?

스프링의 핵심은 자바라는 객체지향 언어의 강력한 장점을 살려서,

좋은 객체지향 어플리케이션을 개발할 수 있게 도와주는 것!

(유지/보수가 쉽도록 프로그램을 **유연하게** 만들도록 도와줌)

좋은 객체 지향 프로그래밍이란

SRP: 단일 책임 원칙(single responsibility principle)

OCP: 개방 폐쇄 원칙(Open/closed principle)

(확장은 열려있고 변경은 닫혀있음, 다형성 활용)

LSP: 리스코프 치환 원칙(Liskov substitution principle)

(구현체는 역할을 반드시 준수)

ISP: 인터페이스 분리 원칙(Interface segregation principle)

(가능한 세밀하게 역할 분리)

DIP: 의존관계 역전 원칙(Dependency inversion principle)

(구현이 아닌, 추상화에 의존)

스프링과 객체 지향의 접점

OCP: 개방 폐쇄 원칙(Open/closed principle)

확장에는 열려있으나 변경에는 닫혀있어야 한다.

DIP: 의존관계 역전 원칙(Dependency inversion principle)

구체화에 의존하지 말고, 추상화에 의존하라

필요한 의존 관계를 직접 만드는 경우,

좋은 객체 지향 설계 원칙을 위반할 가능성 존재😓

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

```
public class MemberService {  
    // private MemberRepository memberRepository = new MemoryMemberRepository();  
    private MemberRepository memberRepository = new JdbcMemberRepository();  
}
```

변경에 닫혀있지 않음
+ 추상화와 구체화에 모두 의존함

스프링과 객체 지향의 접점

좋은 객체 지향 설계 원칙을 지키기 위해서는
의존 관계를 주입해주는 Config 클래스가 필요함

-> 할일이 많음

-> 스프링 프레임워크에서 대신 만들어 제공

(DI, DI 컨테이너 등)

그 외

스프링 컨테이너를 생성하고, 빈으로 등록하는 방법, 테스트 하는 방법,
스프링 빈을 조회하는 방법, 빈 설정 파일 등록하는 방법 등등

단축키 꿀팁

커맨드 + 쉬프트 + 엔터 (중간에서 줄바꿈)

커맨드 + 옵션 + v (자동 생성)

커맨드 + e (최근 파일 조회)

psvm (main 메서드 자동 생성)

soutv (system.out.println 자동 생성)

커맨드 + n (generate)

커맨드 + d (그자리에 복붙)

Thank you!