**소프트웨어 설계 패턴 (실습 4)**

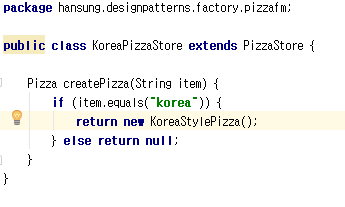
**1494067 이찬희**

이번 실습에서는 예제코드로 주어진pizzafm(팩토리 패턴 예)과 pizzaf(추상 팩토리 예)를 바탕으로 진행합니다.

1. 피자 프렌차이즈 사업이 날로 확대되어서 드디어 한국에도 분점을 내게 되었습니다. 한국 분점의 피자는 한국스타일로 여러분이 원하는대로 정하면 됩니다. 가령 김치피자나 불고기 피자가 판매될 수도 있겠죠.

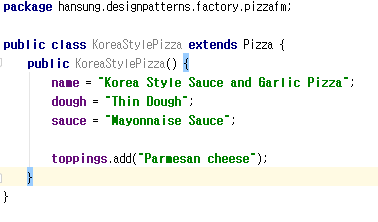
pizzafm의 코드를 바탕으로 한국 분점을 새로이 내기 위해 어떤 클래스가 새로이 추가되었고, 어떤 코드가 수정되었나요.

첫번째로 PizzaStore 클래스를 상속받는 KoreaPizzaStore 클래스 생성

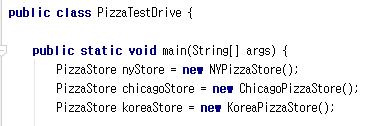


두번째로 재료가 포함된 Pizza 클래스를 상속받는 KoreaStylePizza

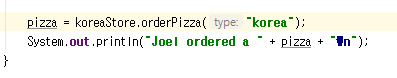
클래스 생성



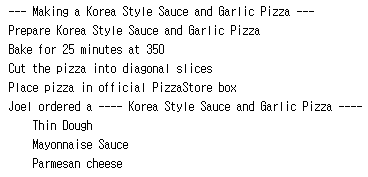
세번째로 PizzaTestDrive 클래스에 KoreaPizzaStore 클래스 객체인 koreaStore를 추가하고



출력해주는 코드를 추가

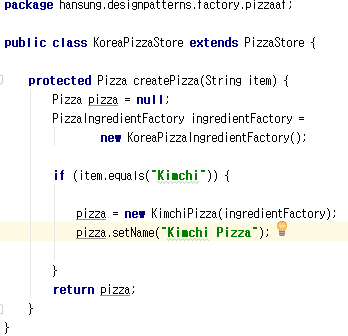


결과화면

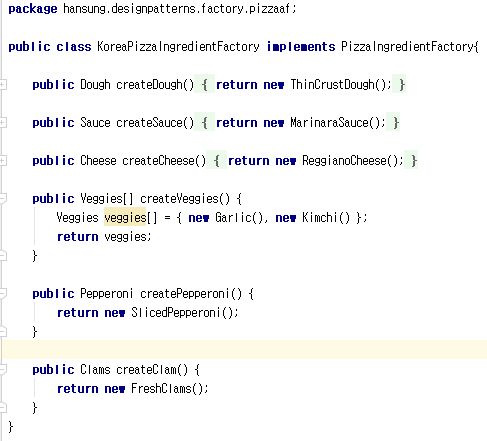


1. 이번에는 pizzaf의 코드를 바탕으로 1번 문제를 적용해보세요.

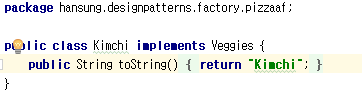
첫번째로 PizzaStore 클래스를 상속받는 KoreaPizzaStore 클래스 생성



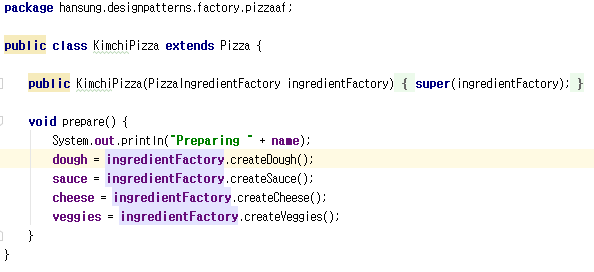
두번째로 PizzaIngredientFactory 인터페이스를 구현한 KoreaIngredientFactory 클래스 생성



세번쨰로 Veggies 인터페이스를 구현한 김치 클래스 생성

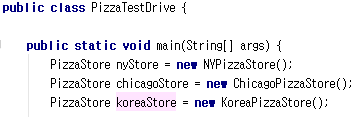


네번쨰로 Pizza 클래스를 상속받는 KimchiPizza 클래스 생성

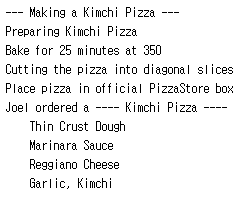


마지막으로 PizzaTestDirve에 KoreaPizzaStore 객체인 koreastore 생성하고

출력 코드 추가







1. pizzafm코드를 바탕으로 했을 때와 pizzaf 코드를 바탕으로 수정했을 때와 어떤 차이가 있는지, 장단점을 비교 토의하세요.

Pizzaaf 코드(추상화) :

만약 공동된 부분이 많은 피자가 새로 만들어진다면

이를 만들기가 쉽다는 장점이 있다.

하지만 새롭게 추가될 피자가 있다면

첫번쨰로 PizaaIngredientFactory 인터페이스에 영향을 주게 된다.

새로운 메소드를 추가해 줘야 한다. 새로운 메소드가 추가되면

위의 인터페이스를 구현하고 있는 시카고,NY피자팩토리 클래스에서도 메소드를 재정의해줘야 하기 때문에 코드의 수정이 많아진다.

Pizzaafm코드(팩토리예제):

재료가 많이 다른 피자라도 손쉽게 구현할 수 있는 장점이 있다.

하지만 새로운 피자마자 도우, 소스, 치즈, 채소 등을 하나하나 추가해 줘야 해서 손이 많이 간다는 단점이 있다.