**소프트웨어 설계 패턴 (실습 4)**

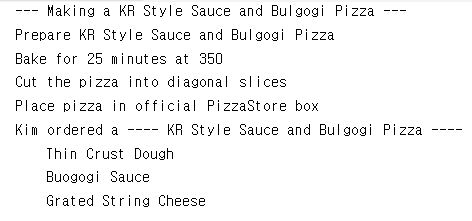
**1494047 김건우**

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

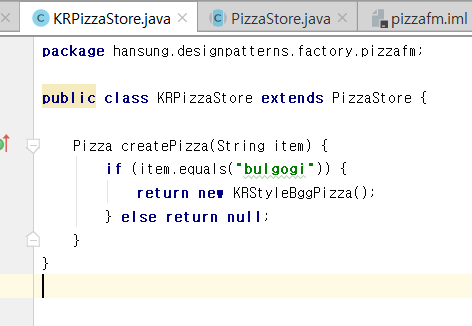
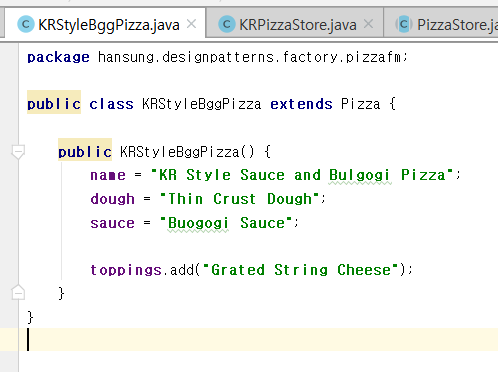
1. 피자 프렌차이즈 사업이 날로 확대되어서 드디어 한국에도 분점을 내게 되었습니다. 한국 분점의 피자는 한국스타일로 여러분이 원하는대로 정하면 됩니다. 가령 김치피자나 불고기 피자가 판매될 수도 있겠죠.

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

.



KR 스타일 불고기 피자 판매 실행

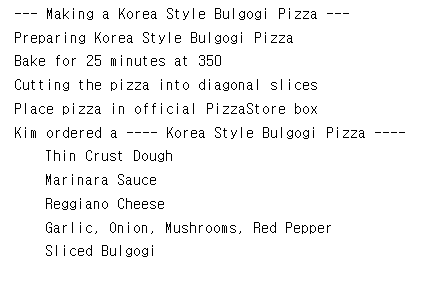
 

KR 피자 스토어 불고기 피자

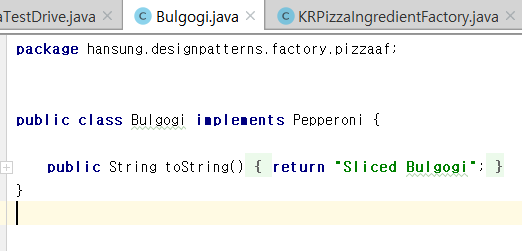
팩토리 메소드 패턴을 이용하여 분점과 메뉴를 추가하는 일은 간단했다.

별다른 코드의 수정 없이 분점과 메뉴를 복사 붙여넣기 하여 내가 원하는 이름으로 바꿔 주면 되는 아주 간단한 작업이었다. 수정되어야 하는 코드는 실제로 객체를 만들게 해주는 PizzatestDrive 뿐이었다.

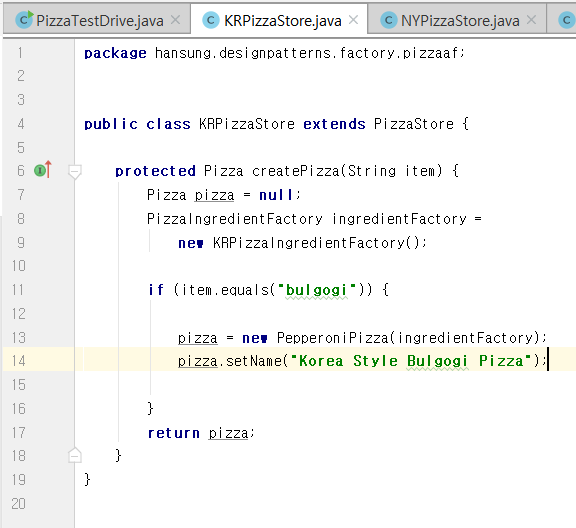
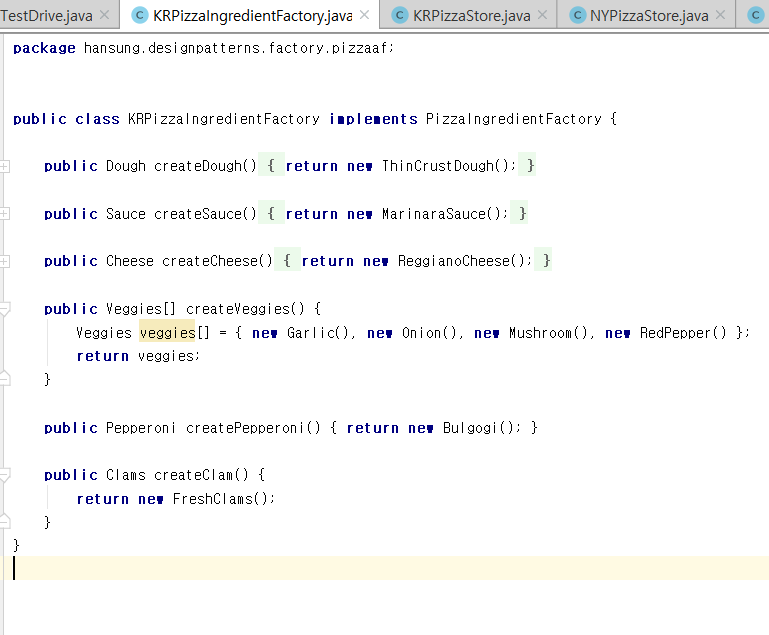
1. 이번에는 pizzaf의 코드를 바탕으로 1번 문제를 적용해보세요.

실행 결과 피자 생성



원재료 추가 (페퍼로니에 불고기)

스토어 추가 팩토리 추가

추상 팩토리 패턴을 이용하니 메뉴의 다양화와 선택한 메뉴에 따라 원재료들이 바로바로 바뀌는게 팩토리 메소드와는 다른 장점임을 알았다. 하지만 가장 대표적인 불편함으로 클래스 수가 너무 많아져 개발자 이외에는 한눈에 파악하기가 힘들어 보였다. 메뉴와 분점 추가에 대한 부분은 분점은 역시 복사 붙여 넣기로 해결이 가능했다. 메뉴는 원재료를 추가하고 조합에 맞게 피자를 생성해주면 되지만 기존 분점에 메뉴를 추가하려면 팩토리도 수정해 주어야 한다는 단점이 있었다.

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

과제를 하면서 내가 직접적으로 받은 느낌은 pizzafm의 경우 피자 자체를 클래스로 만들어 버려서 생성 자체가 깔끔했다. 스토어를 추가하고 메뉴를 추가해 주면 되는 간단한 작업이었다. 단점은 인터페이스를 이용하지 않아 디자인 원칙에 위배 된다는 점과 메뉴가 살짝 바뀌거나 조정되더라도 새로운 클래스를 만들어 주어야 한다는 점이다.

pizzaf같은 경우 재료를 클래스로 만들고 팩토리에서 조합을 하는 방식이기 때문에 메뉴 자체의 세세한 레시피 변경과 분점별로 메뉴를 좀 더 세분화 할 수 있다는 장점이 있었다. 하지만 처음 프로젝트를 열었을 때 pizzafm에 비해 한눈에 파악 하기가 힘든 점이 있었다. 인터페이스 위주로 코딩을 하기에는 더 좋은 부분이 있었다. 디자인 원칙적인 측면으로도 더 좋았다.

메뉴가 많은 피자가게면 추상 팩토리 패턴을 사용하는 것이 바람직하고 분점이 많은 피자가게라면 팩토리 메소드 패턴을 사용하는 것이 바람직 하다고 생각했다.