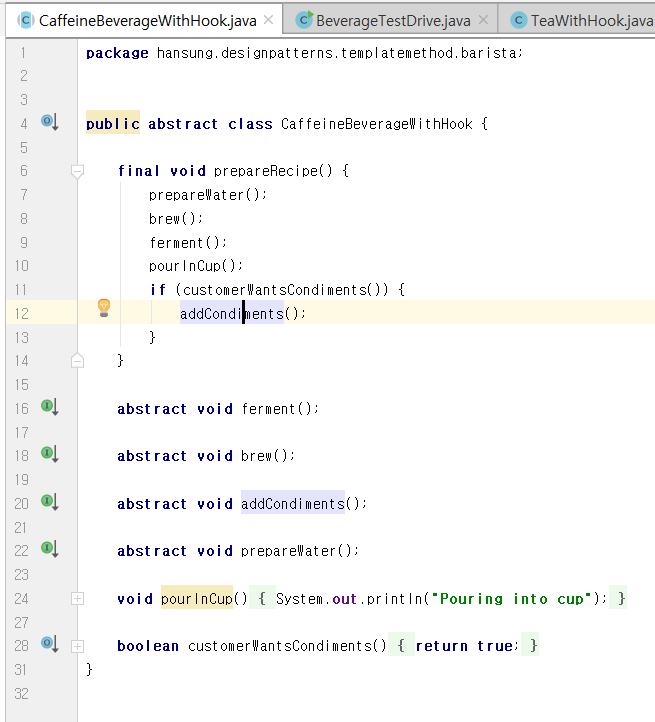
**소프트웨어 설계 패턴 (실습 9)**

**1494047 김건우**

1. Coffee, Tea 뿐만 아니라 위의 더치 커피를 만들기 위해서 CaffeineBeverageWithHook 클래스의 템플릿 메소드를 수정하고, 필요하면 훅 메소드를 추가하시오. 코드에 대한 설명도 포함해야 합니다.



클래스에 숙성을 위한 메소드Ferment를 추가 하고 찬물을 준비 하기 위해 boilwater메소드를 prepareWater메소드로 바꾸어 주었다. 둘 다 추상메소드로 구현하여 실질적으로 구현은 각자 클래스 안에서 해주는 것으로 되어있다.

그 후 fermant를 prepareRecipe 에 추가해 주었다.

1. 수정된 CaffeineBeverageWithHook 클래스를 바탕으로, Coffee, Tea 클래스를 수정하고, DutchCoffee 클래스를 추가하시오. 코드에 대한 설명도 포함해야 합니다.

더치커피 티(커피)

더치커피와 티가 구분되는 확실한 부분은 ferment메소드의 구현 여부이다.

실제로 구현되는 클래스는 더치커피 클래스뿐이다. 나머지 커피와 티는 숙성이 필요 하지 않으니 추상 메소드로 두고 구현하지 않았다. 나머지는 그대로 두고 메소드 내용만 더치커피에 맞게끔 수정해 주었다

1. 위의 실습 내용을 바탕으로 템플릿 패턴의 장단점을 토의해 봅시다.

템플릿 메소드 패턴은 코드의 중복을 줄일 수 있고 쉽게 추가적인 객체를 추가 해 줄 수 있다. 다만 객체와는 별개로 메소드 부분에서는 기존의 패턴과 다른 메소드를 추가 해 주어야 한다면 이전의 코드를 모두 수정해 주어야 하는 불상사가 생길 수 있다. 이번 실습을 예로 들어서 더치를 추가 해 줄 때 숙성의 fermnet같은 경우 더치에서만 필요로 하는 부분이었기 때문에 메소드를 추가 해 주었을 때 다른 클래스 에서도 메소드를 생성해 주어야 한다는 불편함이 있었다. 기존의 코드를 그대로 복사 해서 사용 할 수 있다는 장점이 있기는 했지만 뿌리부터 변경되는 점이 있다면 메소드를 추가하고 그 메소드가 모두 기존의 클래스에 영향을 끼친다면 OCP원칙에 위배되는 것이 아닌가 하는 생각이 들었다.