소프트웨어 분석 설계

14주차

20170677 오융택

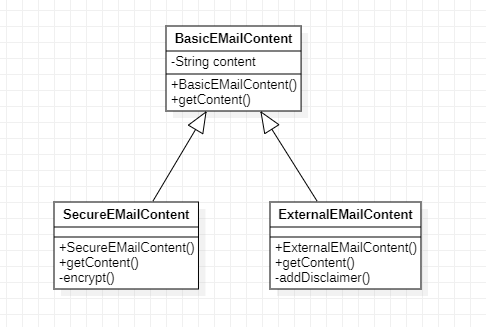
1. 연습문제 1번
   1. 각 클래스의 기능 설명하기

BasicEMailContent : 기본적으로 문자열 Content를 반환하는 클래스

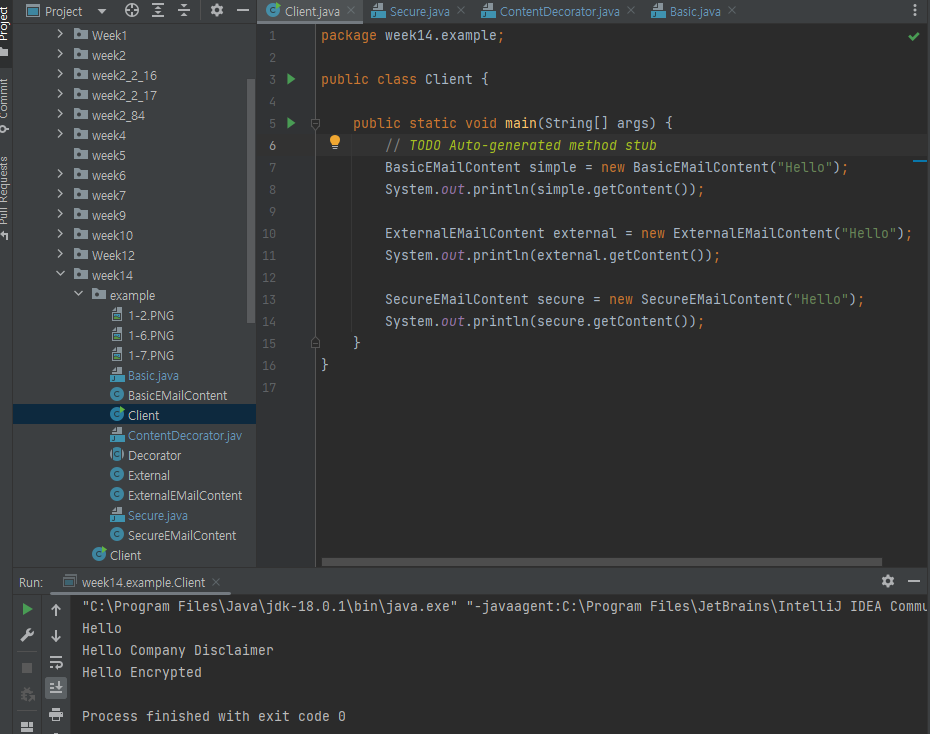
ExternalEMailContent : BasicEMailContent을 상속받아 추가적으로 Disclaimer라는 문자열을 추가하는 addDisclaimer 라는 메소드를 확장하는 자식 클래스

SecureEMailcontent : BasicEMailContent을 상속받아 암호화했다는 문자열을 출력하는 encrypt 메소드를 추가적으로 가지고 있는 자식 클래스

* 1. 클래스 다이어그램 작성하기



* 1. Client코드 실행 결과 작성하기



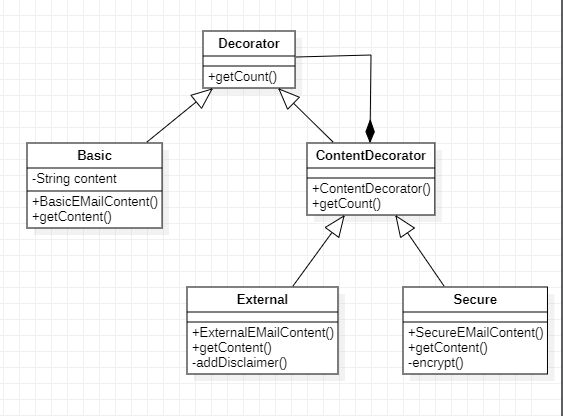
* 1. ~~OCP위반 설명하기~~

~~XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX~~

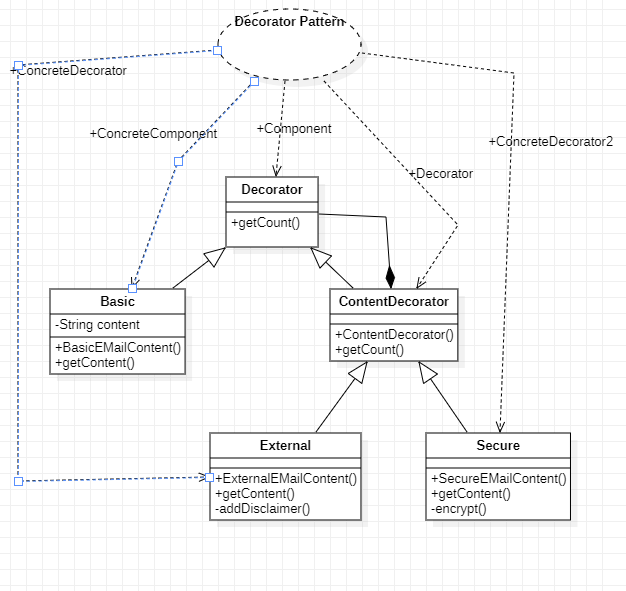
* 1. 데커레이터 패턴을 활용해 취약점 개선하기

데커레이터 패턴을 활용하면 필요한 기능을 추가적으로 조합하여 합체로봇 만들듯이 인스턴스를 찍어낼 수 있다. 이렇게 하면 새로 생긴 기능(Ex 번역하기)와 같은 기능을 OCP위배 없이 추가할 수 있다.

* 1. 개선된 설계를 클래스 다이어그램으로 표현하고 간단히 설명하기

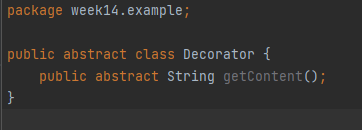


* 1. Meta 다이어그램으로 표현하기



* 1. 코드 작성하기

Decorator.java



Basic.java

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

ContentDecorator.java

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

External.java

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Secure.java

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

* 1. Client 클래스 재작성하고 Decorator패턴 추가하기

Transcribe.java

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Client.java

