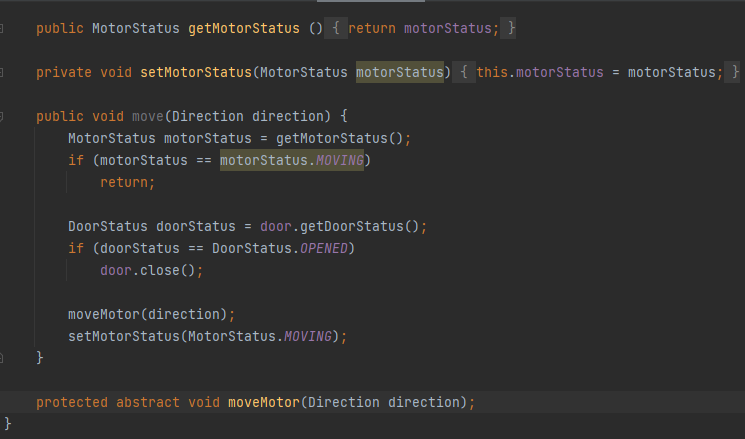
소프트웨어 분석 설계

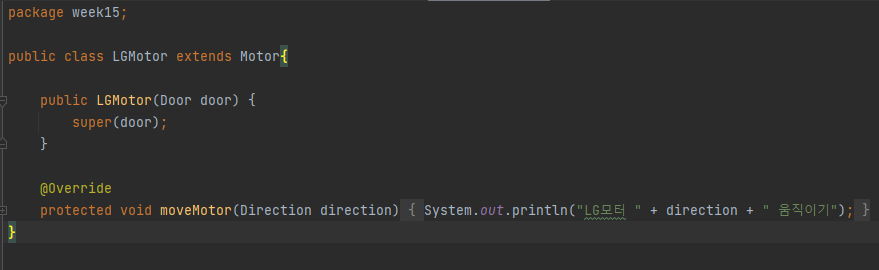
15주차

20170677 오융택

1. 코드 11-4 구현하기
   1. Motor.java



* 1. LGMotor.java



* 1. HyundaiMotor.java

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 1. Direction, DoorStatus, MotorStatus.java

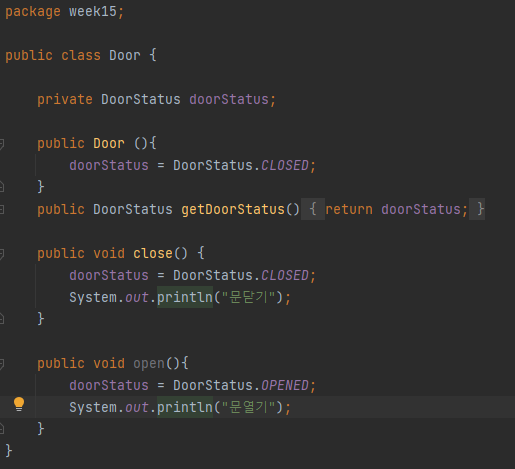
텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

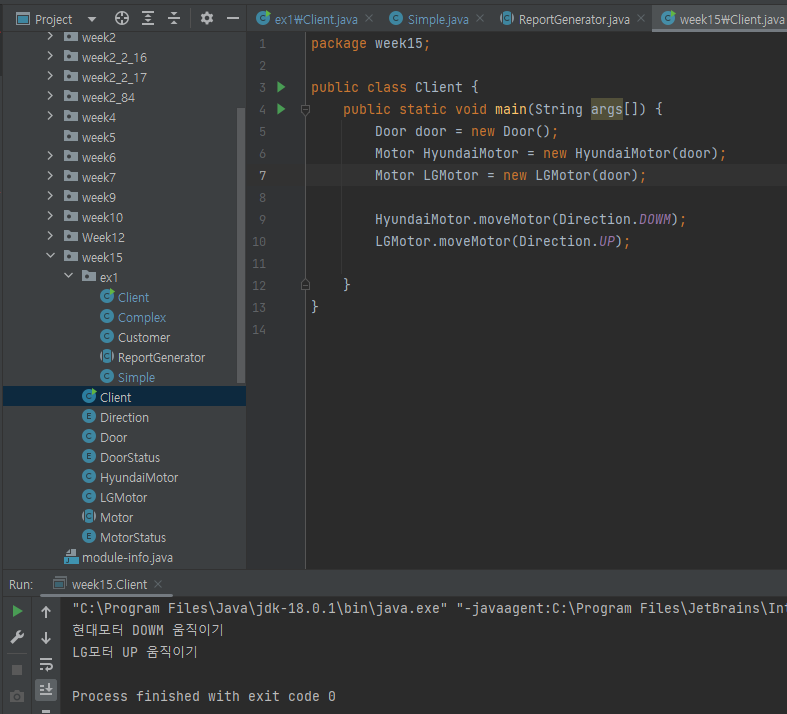
자동 생성된 설명텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

* 1. Door.java



* 1. Client.java



1. 연습문제
   1. 템플릿 메서드 패턴을 사용해 클래스 다이어그램으로 표현하고 설명하기

텍스트, 도표, 라인, 평행이(가) 표시된 사진

자동 생성된 설명

Customer : Report제너레이터가 항상 가지고 있어야 하는 인스턴스를 책임지는 클래스

ReportGenerator : Customer 리스트 인스턴스를 가지고 잇는 추상 클래스, generate메소드 안에서 renew()메소드를 호출하여 보고서의 출력 대상이 될 Customer인스턴스 리스트를 다시 연산한다.

이 중 renew()는 추상 메소드 이므로 자식 클래스에서 구현해야하는 책임을 준다.

Simple/Complex : ReportGenerator클래스를 상속받는 구체화 클래스, renew()메소드를 반드시 재정의해줘야 한다.

* 1. 탬플릿 패턴을 이용해 개선한 설계에서 각 구성 요소에 대응하는 클래스와 메서드를 구분하라.  
     Customer : 보고서 생성의 대상이 되는 데이터를 가지고 있는 인스턴스

ReportGenerator : Customer 인스턴스를 속성으로 가지고 보고서를 출력하는 추상클래스

Generate() : 보고서를 출력하는 public 메서드

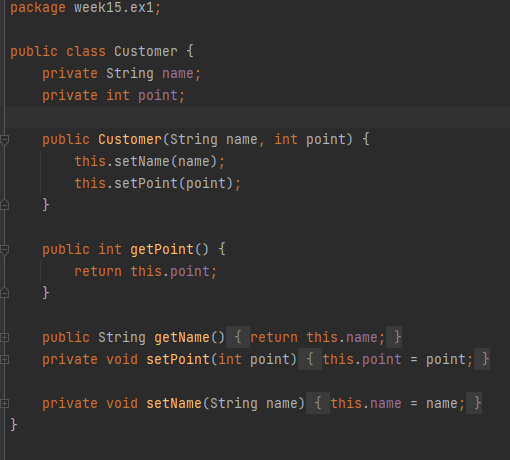
Renew() : 아직 구현되지 않은 protected List<Customer>추상 메서드

Simple / Complex : ReportGenerator을 상속받는 자식클래스

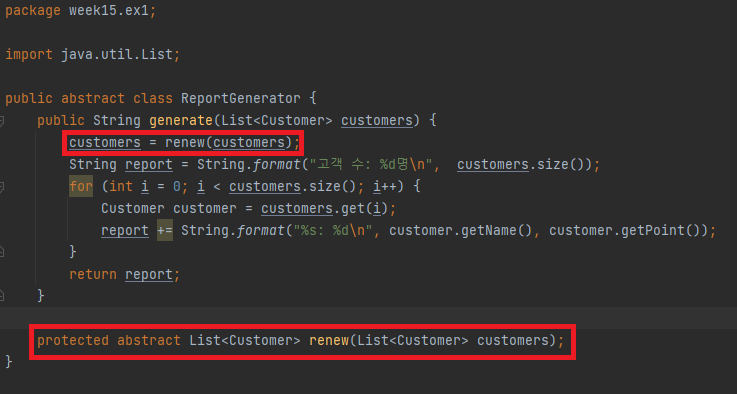
Renew() : Generate()메소드의 실행시 보고서 출력 되상이 될   
List<Customer> Customers 에서 조건에 맞는 Customer만 뽑아 새로운 Customers리스트를 리턴함

* 1. 코드 작성하기

Customer.java



ReportGenerator.java

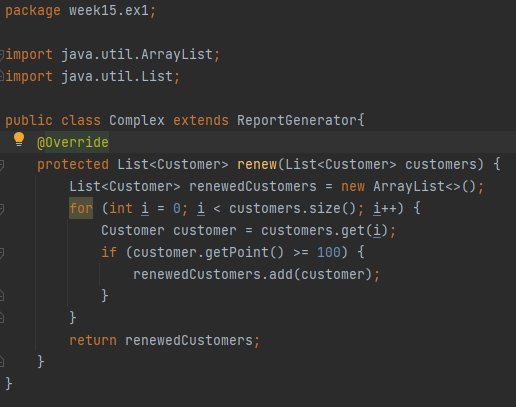


Simple.java

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Complex.java



* 1. Client클래스 재작성하고 실행 결과 기록하기

Client.java

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명