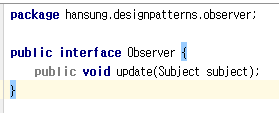
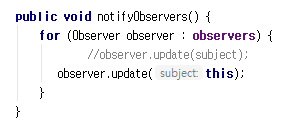
소프트웨어 설계패턴 과제(실습2)

1494067 이찬희

1번 push방식 -> pull 방식으로 수정

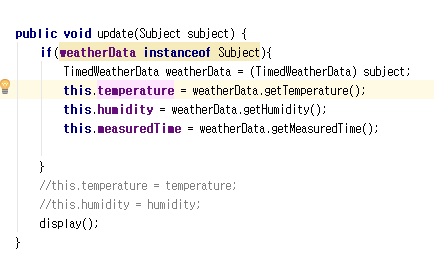


->Observer 인터페이스에서 subject 인터페이스 변수를 생성

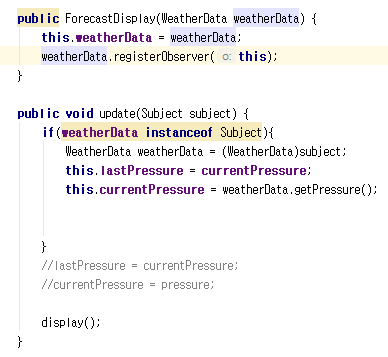


->WeatherData 클래스에서 nofifyObservers()를 다음과 같이 수정

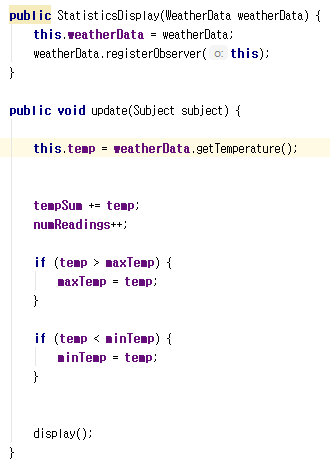
다음으로 온도, 습도, 압력 클래스의 수정



->CurrentConditionsDisplay 클래스의 update 메소드 수정



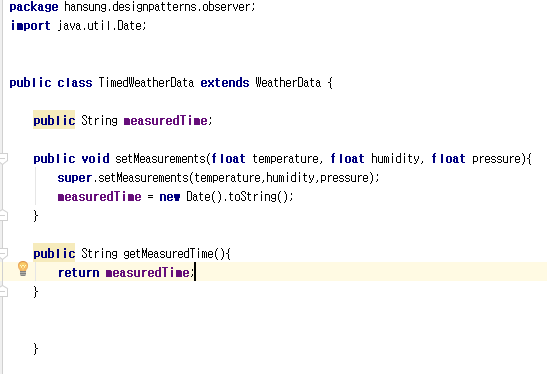
->ForcastDisplay 클래스의 update 메소드 수정



->StatisticsDisplay 클래스의 update 메소드 수정

(결과 화면은 TIme까지 표시한 채로 찍어놔서 맨 밑에 첨부하겠습니다)

2번

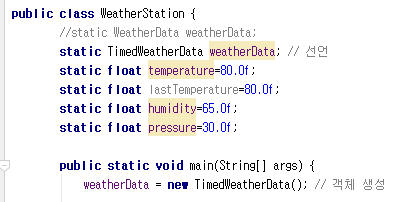


->WeatherData 클래스를 상속받는 TimedWeatherData 클래스를 생성하고

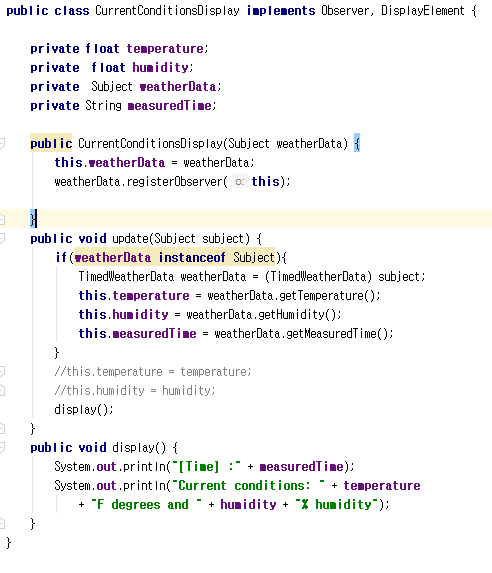
measuredTime 변수에 현재 시간 저장 이후 CurrentConditionDisplay 클래스에서

시간을 불러오기 위해 새 메소드 getMeasuredTime 생성

setMeasurements 메소드를 오버라이딩 항ㅆ음

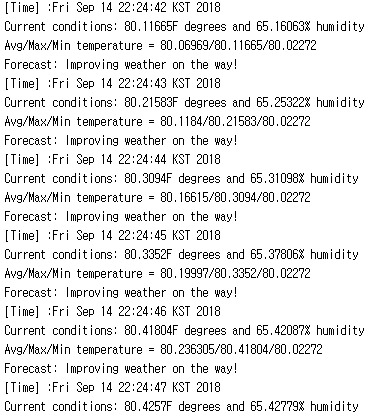


->WeatherStation 클래스 수정



* CurrentConditionsDisplay 클래스 수정

최종 결과 사진



3번 push와 pull 장단점

Push 장점 : 변경사항을 바로 알 수 있다.

단점 : 데이터가 너무 크게 되면 클래스가 통지를 했을 시 전송되는 시간이 느림

넘겨질 정보가 늘어나면 인터페이스를 수정해 줘야 함

Pull 장점 : 데이터가 큰 경우에는 push보다 효율적임

정보가 늘어나도 필요로 하는 부분만 수정해도 됌

단점 : 통지를 받으면 옵저버가 다시 클래스에서 데이터를 가져와야 함