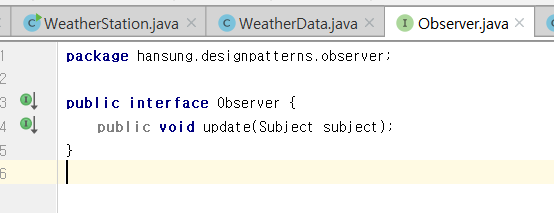
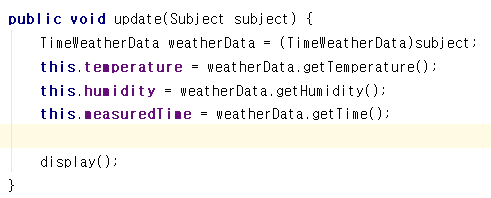
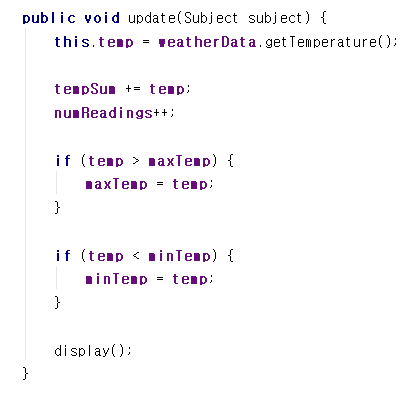
**소프트웨어 설계 패턴 (실습 2)**

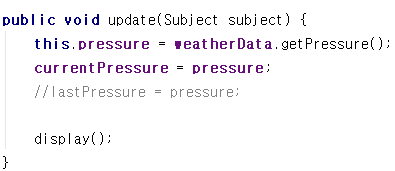
**1494047 김건우**

1. notifyObservers() 메소드에서 호출하는 observer의 update() 메소드는 푸시 방식으로 값을 전달하는데, 이를 풀 방식으로 동작하도록 수정하여라.

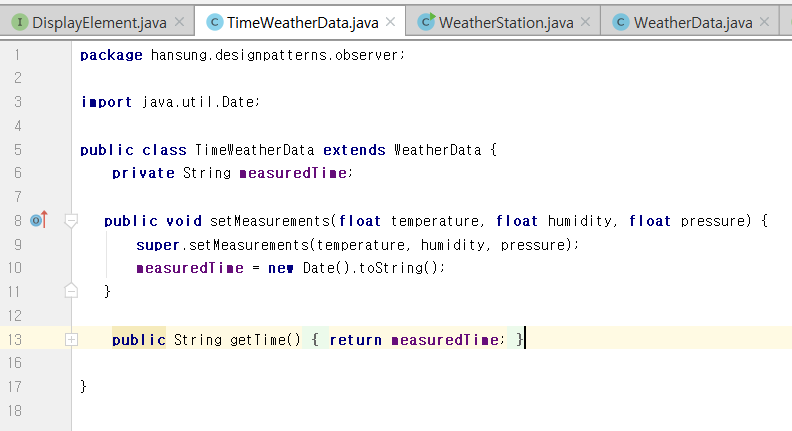
 - Observer.java

 -CurrentConditionsDisplay

-StatisticsDisplay

 - ForecastDisplay

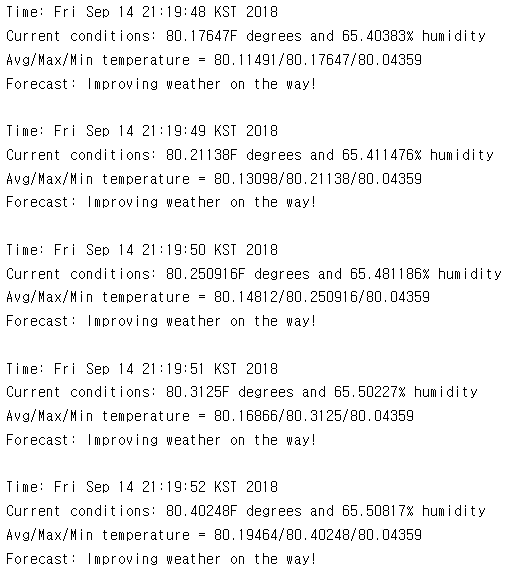
1. WeatherData의 서브클래스인 TimedWeatherData를 정의하고, 상태변화가 측정된 시간 (measuredTime)을 String 타입의 속성으로 추가하라. 또한, setMeasurements() 메소드가 호출될 때마다, 현재 시간을 java.util.Date 클래스를 이용하여 얻어와서 (힌트, new Date().toString()) measuredTime 속성의 값으로 설정하도록 setMeasurement() 메소드를 오버라이딩 하여라.

- TimeWeatherData

내부에 setMeasurements가 오버라이딩 되어 있다.

1. CurrentConditionsDisplay 클래스를 수정하여 측정된 시간이 출력될 수 있도록 하여라. (참고, WeatherStation 클래스의 main함수내에서 WeatherData 객체 대신에 TimedWeatherData객체를 생성하고 이를 각 Display 객체에 초기화 해야 한다.)

예 “[Time: Sat Sep 03 02:38:15 KST 2016] Current conditions: 80.070755F degrees and 65.02126% humidity”

 최종 결과 모습

1. 위의 결과를 바탕으로 푸시 방식과 풀 방식의 장단점에 대해서 논의하시오.

|  |  |  |
| --- | --- | --- |
|  | Pull 방식 | Push 방식 |
| 장점 | 정보 변경 통지를 비교적 먼저 받은 후에 결과를 확인할 수 있다. | 변경사항이 통지와 같이 오기 때문에 바로바로 정보를 확인 할 수 있다. |
| 단점 | 변경사항과 통지가 따로 오기 때문에 정보를 받는데 시간이 걸릴 수 있다. | 만약 Data가 기가 단위로 커질 경우 결과는 물론이고 통지 자체가 늦어진다. |