Java20-Lab3

32191826 박주희

<Observer>

한 객체의 상태가 바뀌면 그 객체에 의존하는 다른 객체들에게 연락이 가고 자동으로 내용이 갱신되는 방식 🡪 one-to-many 의존성

Observable 클래스: 등록된 observer 관리

새로운 데이터가 들어오면 등록된 Observer에게 데이터 전달

<Thread>

하나의 프로세스 내부에서 독립적으로 실행되는 하나의 작업 단위를 말하며, 세부적으로는 운영체제에 의해 관리되는 하나의 작업 혹은 태스크를 의미한다.

직접 상속 받아서 thread를 생성할수도 있고 Runnable 인터페이스를 구현해서 생성할수도 있음

<Interface Observer>

lecture4-exercise와는 다르게 JDK에서 지원하는 Observable과 Observer API를 활용하지 않고 직접 구현

<Class Display>

currentconditions, DecompositionIndex, HeatIndex, WindChillTemperatureIndex를 각각 display하는 클래스

Observer,DisplayElement를 상속받기 때문에 display(),update()를 모두 override

update에서는 각자에 기준에 맞게 ENUM을 index에 저장하고 display에서는 이를 출력

<Class WeatherData,WeatherDataSubject>

날씨의 정보를 가지고 있는 class WeatherData

WeatherDataSubject 에서 List<WeatherData> data 정보를 받아서 WeatherData를 매 1초마다 새로 갱신(thread 이용)

**try** {

observers.add(o);

Thread.*sleep*((**long**)(Math.*random*() \* 1000));

}**catch**(Exception e) {}

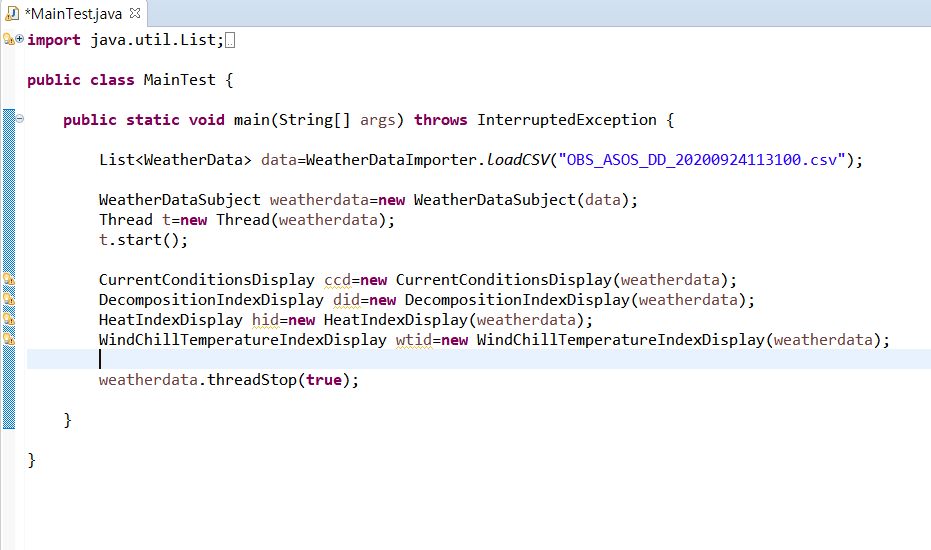
random 시간으로 observers에 observer o를 추가

<Class WeatherDataImporter>

loadCSV메소드에서 filename을 받아서 해당 file을 읽고 날씨 정보를 List<WeatherData> weatherdatas에 저장

BufferedReader와 parse를 이용

<Class MainTest>



파일에서 정보를 읽어서 WeatherData형으로 List에 저장

Thread 생성 후 start

Disply class들 생성 및 add **this**.weatherData.addObserver(**this**);

Thread를 stop 시키기 위해서 WeatherDataSubject의 bool 함수 stopRunning 이용