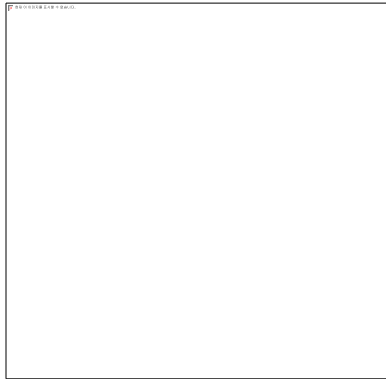


고급객체지향프로그래밍 실습과제 #04

조용주
ycho@smu.ac.kr

실습과제 #04

- ▣ 여기서는 일종의 공공 정보 디스플레이를 구현한다
- ▣ 단 여기서는 사용자가 원하는 정보 디스플레이의 조합이 가능하도록 만든다
 - 정보 디스플레이 조합은 displays.txt에 원하는 정보들을 순서대로 나열해서 지정한다(중복 지정 가능)



displays.txt 파일 내용

실습과제 #04

- 프로그램이 실행되면 displays.txt 파일을 읽어서 내용에 따라 적절한 디스플레이를 화면에 배치한다
- Hud Display는 기본적인 디스플레이에 해당되며 사용자가 추가할 수 있는 디스플레이는 speed, weather, time 이 있다
- 본 과제에서는 문제를 단순화시키기 위해 해당 디스플레이의 내용은 time은 해당 프로그램이 실행되는 시간, weather와 speed 디스플레이는 다음 슬라이드에서 보인 내용만을 보이는 것으로 한다
 - 현재 시각 정보는 java.time.LocalDateTime.now()함수를 이용해서 구할 수 있음

```
System.out.println(java.time.LocalDateTime.  
now().toString());
```

실습과제 #04

- 주어진 MP04 코드를 이해하고, 프로그램을 완성시킬 것
 - 클래스 다이어그램을 확인하고 없는 클래스들을 구현하거나 기존 클래스 코드를 수정할 것
 - Main.java는 수정하지 말 것
- MP04V1.jar 파일을 실행시켜서 프로그램이 어떻게 실행되어야 하는지 확인 가능
 - 실행 방법

```
java -jar MP04V1.jar
```
 - displays.txt 파일이 jar파일과 같은 폴더에 있어야 함

실습과제 #04 코드 실행 화면

Hud 디스플레이

Time 디스플레이

Weather 디스플레이

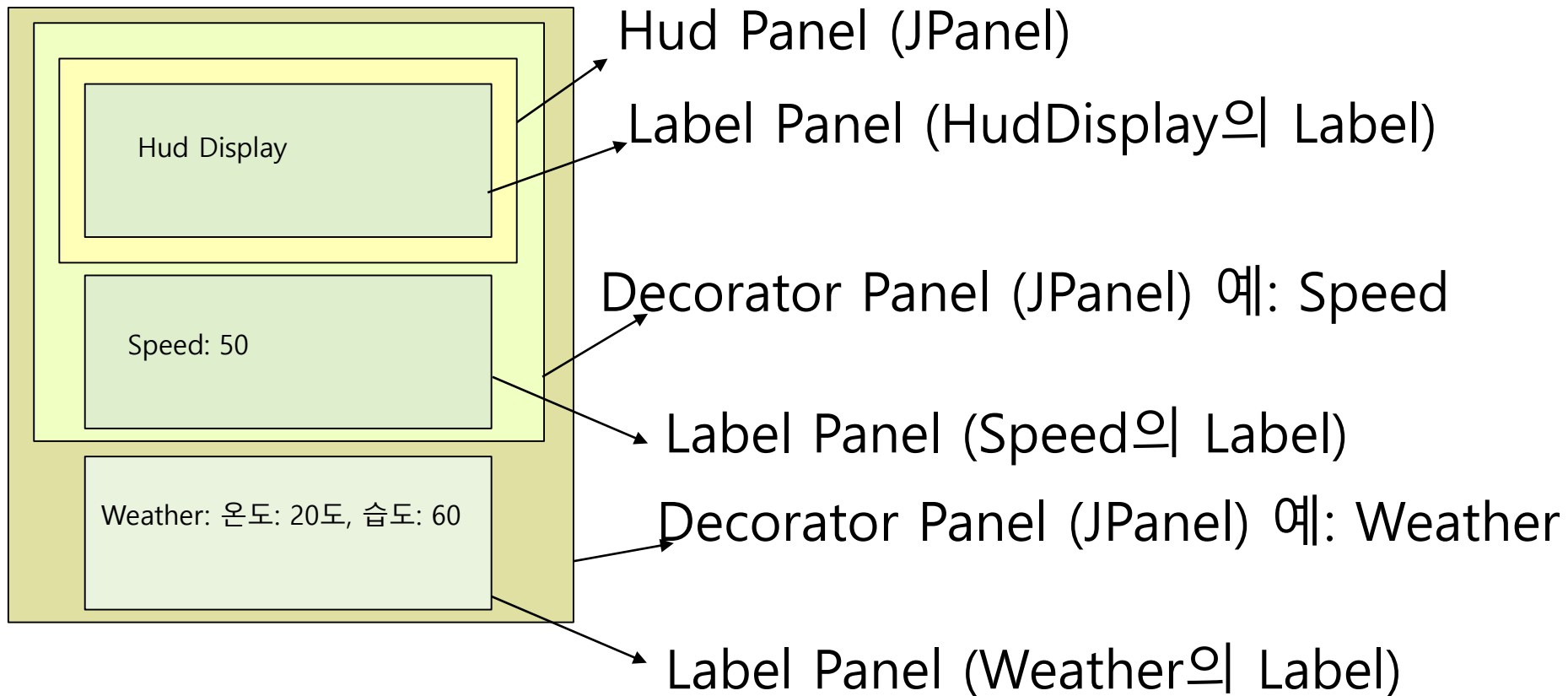
Speedometer 디스플레이

실습과제 #04 클래스 다이어그램



실습과제 #04 데코레이터 GUI 구조

- 기본 베이스 디스플레이인 HudDisplay 객체를 Speedometer 객체로 포장한(wrap) 후에 Weather 객체로 다시 한번 장식(decorate)한 경우를 보임



실습과제 #04 구현 내용

▣ LoadHudDisplays 클래스 구현

■ LoadDisplayInterface를 구현하는 클래스

▣ LoadHudDisplays.java

- 파일로부터 자료를 읽어서 각 줄(line)의 디스플레이 이름을 요소로 취하는 문자열 리스트를 구성
- 생성자는 디스플레이 구성 요소 문자열을 담은 파일의 이름을 입력으로 전달 받음
- LoadDisplayInterface.load() 함수는 입력과 출력 부분을 바꾸지 않음

실습과제 #04 구현 내용

- ▣ 데코레이터 디스플레이 클래스
 - SpeedometerDisplay.java
 - TimeDisplay.java
 - WeatherDisplay.java
 - LoadHudDisplays.java

실습과제 #04 구현 내용

- ▣ displays.txt파일의 내용과 데코레이터 클래스의 상관관계

문자열	해당 클래스
speed	SpeedometerDisplay
time	TimeDisplay
weather	WeatherDisplay