

Java Programming II

Lab3

514770-1

Fall 2020

9/29/2020

Kyoung Shin Park
Computer Engineering
Dankook University

Lab3

- Practice to write a program that displays various weather index using **Observer pattern**.
 - The WeatherDataSubject (concrete subject) takes List<WeatherData> and then updates WeatherData approximately every 1 second in a separate thread.
 - 4 concrete observers (CurrentConditionsDisplay, HeatIndexDisplay, WindChillTemperatureIndexDisplay, DecompositionIndexDisplay) can be added or deleted at any time.
 - The WeatherData updates are notified to all registered observers.

Lab3

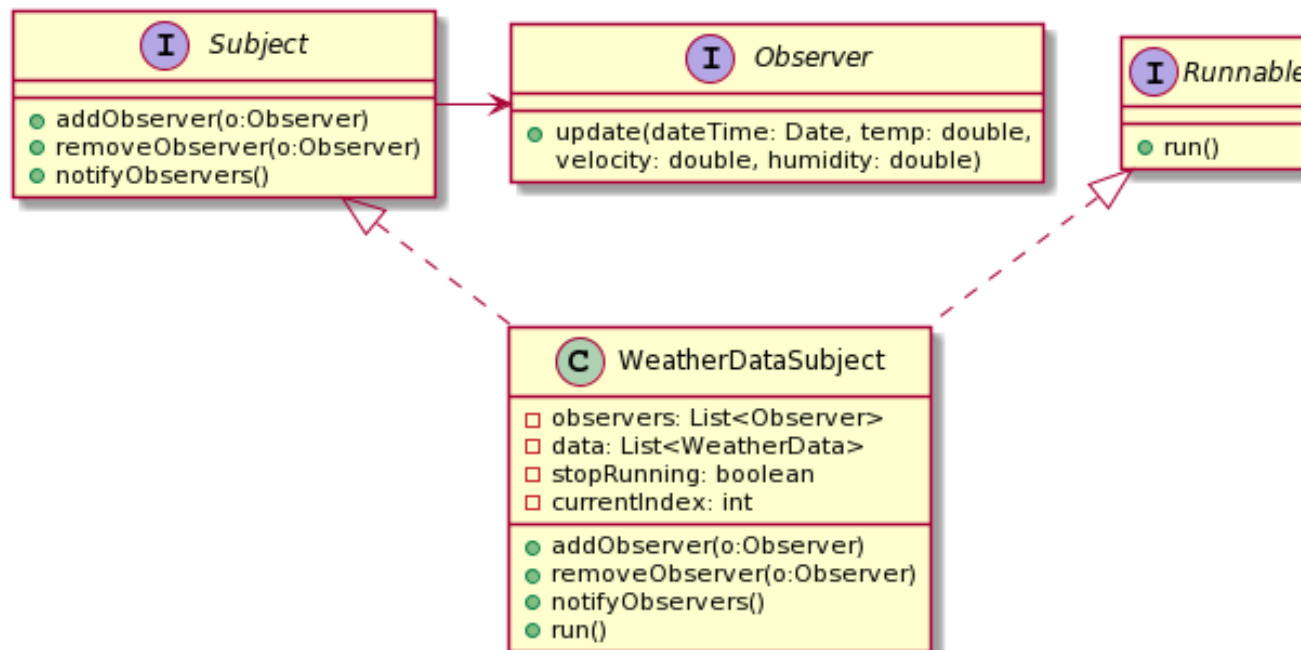
- ❑ **WeatherData** contains **dateTime**, **temperature**, **windVelocity**, **relativeHumidity**.
- ❑ **WeatherDataImporter** **loadCSV()** reads a csv file and load WeatherData into the ArrayList.

C WeatherData
<ul style="list-style-type: none">❑ dateTime❑ temperature❑ windVelocity❑ relativeHumidity
<ul style="list-style-type: none">● getDateTime()● getTemperature()● getWindVelocity()● getRelativeHumidity()● toString()

C WeatherDataImporter
<ul style="list-style-type: none">● loadCSV(filename: String): List<WeatherData>

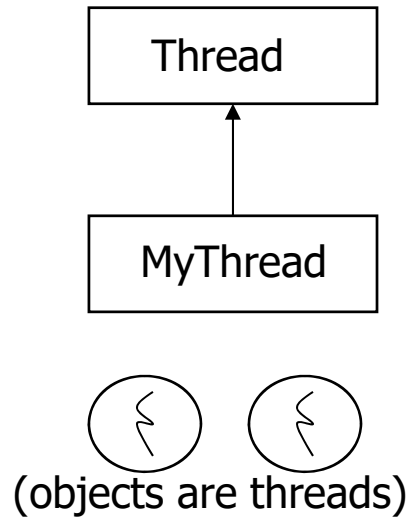
Lab3

- **WeatherDataSubject** is a separate thread that runs WeatherData updates randomly.
 - The constructor takes List<WeatherData>.
 - run() **updates currentIndex** and notifyObservers().
 - notifyObservers() **updates WeatherData** using currentIndex to **all registered observers**.

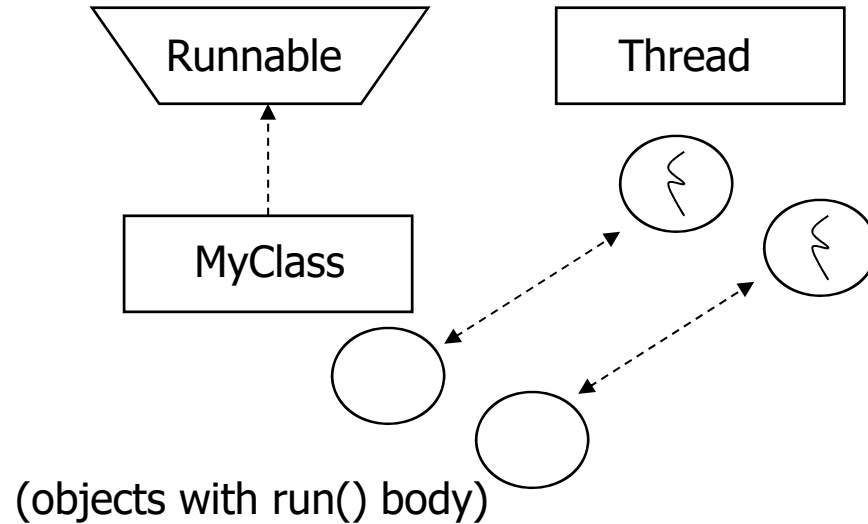


Java Threads

1. Create a class that **extends** the **Thread** class
2. Create a class that **implements** the **Runnable** interface



[a]



[b]

1. Extending the Thread Class

- ❑ Create a class by extending Thread class and override run() method:

```
class MyThread extends Thread
{
    public void run()
    {
        // thread body of execution
    }
}
```

- ❑ Create a thread:

```
MyThread thr1 = new MyThread();
```

- ❑ Start Execution of threads:

```
thr1.start();
```

- ❑ Create and Execute:

```
new MyThread().start();
```

1. Extending the Thread Class

```
class MyThread extends Thread {  
    public void run() {  
        System.out.println(" this thread is running ... ");  
    }  
}
```

```
class ThreadEx1 {  
    public static void main(String [] args ) {  
        MyThread t = new MyThread();  
        t.start();  
    }  
}
```

2. Threads by implementing Runnable interface

- ❑ Create a class that implements the interface Runnable and override run() method:

```
class MyThread implements Runnable
{
    ....
    public void run()
    {
        // thread body of execution
    }
}
```

- ❑ Creating Object:

```
MyThread myObject = new MyThread();
```

- ❑ Creating Thread Object:

```
Thread thr1 = new Thread( myObject );
```

- ❑ Start Execution:

```
thr1.start();
```


2. Threads by implementing Runnable interface

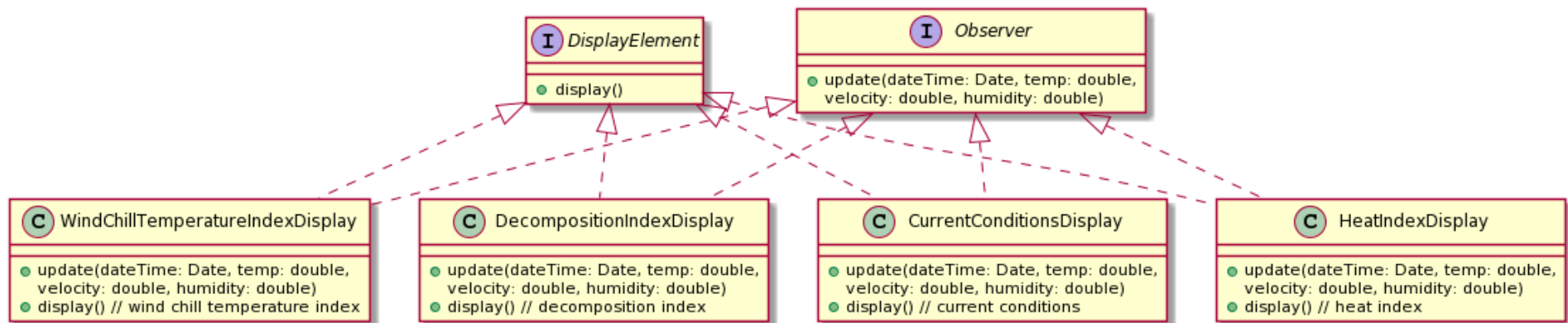
```
class MyThread implements Runnable {  
    public void run() {  
        System.out.println(" this thread is running ... ");  
    }  
}
```

```
class ThreadEx2 {  
    public static void main(String [] args ) {  
        Thread t = new Thread(new MyThread());  
        t.start();  
    }  
}
```

Lab3

□ Observers

- CurrentConditionsDisplay
- HeatIndexDisplay
- WindChillTemperatureIndexDisplay
- DecompositionIndexDisplay



Lab3

□ CurrentConditionsDisplay

- CurrentConditionsDisplay displays WeatherData.

□ HeatIndexDisplay

- http://www.kma.go.kr/HELP/basic/help_01_04.jsp
- enum HeatIndex { VERY_HIGH, HIGH, USUAL, LOW }
- double calculate(double F, double R)

**HI = -42.379 + (2.04901523*F) + (10.14333127*R) -
(0.22475541*F*R) - (0.00683770*F*F) - (0.05481717*R*R) +
(0.00122874*F*F*R) + (0.00085282*F*R*R) -
(0.00000199*F*F*R*R) [F: farenheit temperature, R:
relativeHumidity]**

Lab3

□ WindChillTemperatureIndexDisplay

- http://www.kma.go.kr/HELP/basic/help_01_07.jsp
- enum WindChillTemperatureIndex { DANGER, WARNING, CAUTION, AWARE }
- double calculate(double T, double V)

$$\text{WCTI} = 13.12 + 0.6215 * T - 11.37 * V^{0.16} + 0.3965 * V^{0.16} * T$$

T [T: temperature(celsius), V : wind velocity(km/h)]

□ DecompositionIndexDisplay

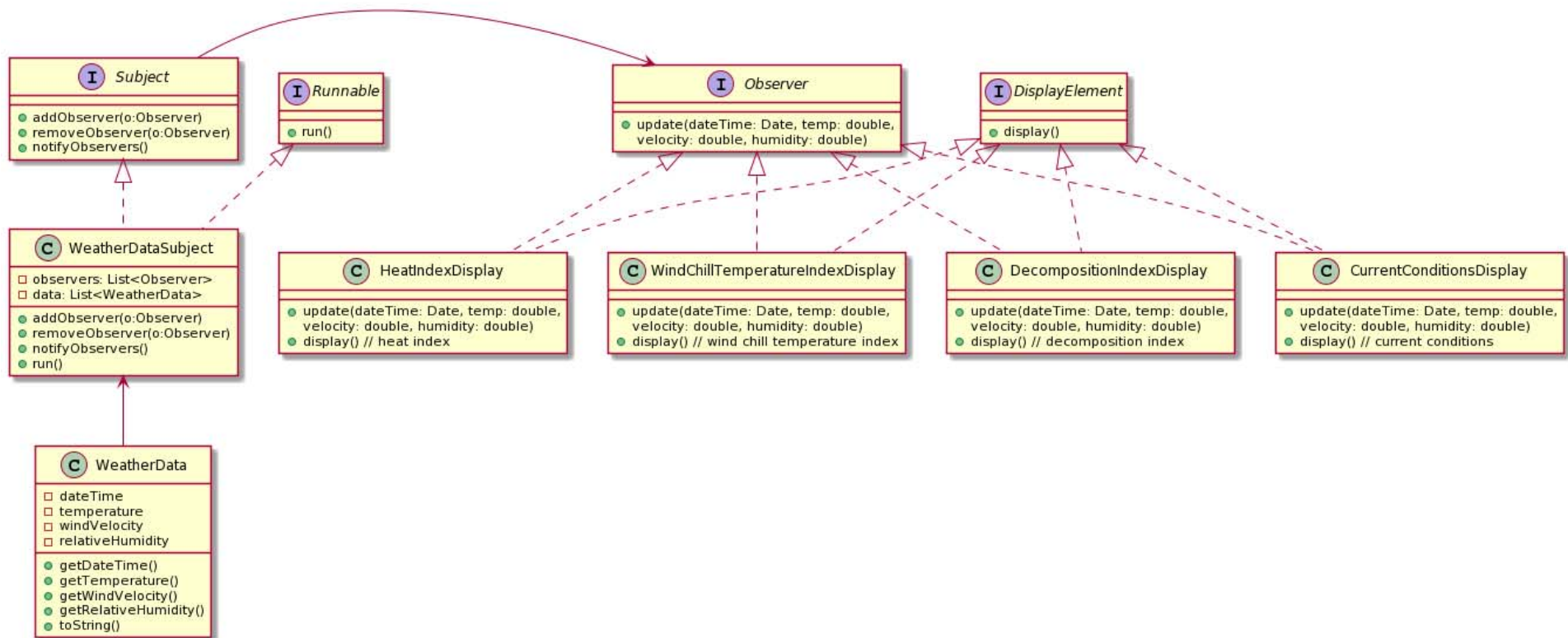
- http://www.kma.go.kr/HELP/basic/help_01_03_01.jsp
- enum DecompositionIndex { HIGH, NORMAL, LOW }
- double calculate(double RH, double T)

$$\text{DI} = ((\text{RH} - 65) / 14) * (1.054 * T)$$

[RH: relative humidity (%), T: temperature (celsius)]

Lab3

Class Diagram



Lab3

- Main class tests add new observers and delete observers at random time to see if the registered observers can display its data using the weather data updates.
 - First, `WeatherDataImporter load List<WeatherData>`.
 - Then, `instantiate WeatherDataSubject` and then `start the thread`.
 - `Sleep randomly` using `Thread.sleep(1000) // sleep 1 second`
 - Then, `add observers` (`CurrentConditionsDisplay`, `HeatIndexDisplay`, `WindChillTemperatureIndexDisplay`, `DecomposeIndexDisplay`) with short term sleep.
 - Then, `delete observers`.
 - Then, `stop thread`.

Submit to e-learning

- Add your code (e.g., additional method, class, routine, etc) in the Lab3 assignment.
- Submit the Lab3 assignment (JAVA20-2-Lab3-ID-name.zip including the report) to e-learning.