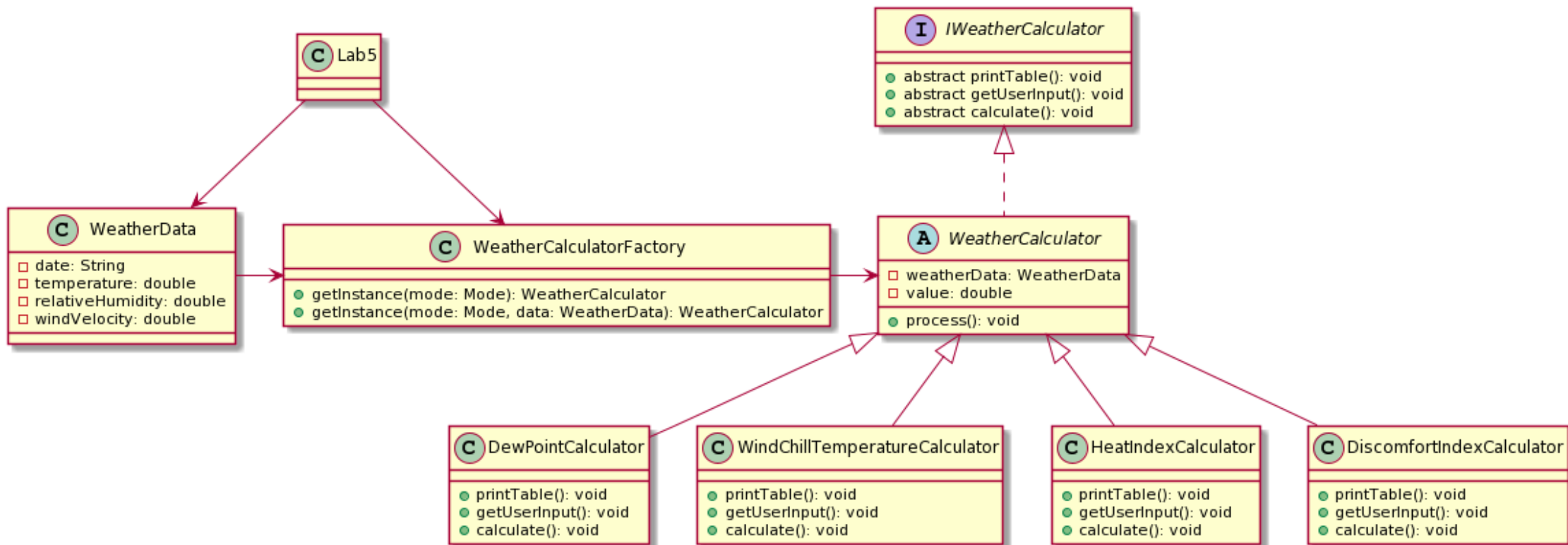


2021학년도 1학기
JAVA 프로그래밍 I

514760
2021년 봄학기
5/19/2021
박경신

과제 Lab5 (Interface, Collection, Package)



과제 Lab5 (Interface, Collection, Package)

- Lab5에서는 dto, dao, controller 패키지로 나눈다.
 - dto 패키지
 - Mode 열거형
 - WeatherData 클래스
 - IWeatherCalculator 인터페이스
 - WeatherCalculator 추상클래스, DewPointCalculator, WindChillTemperatureCalculator, HeatIndexCalculator, DiscomfortIndexCalcualtor, YourWeatherCalculatorClass 클래스
 - WeatherCalculatorFactory 클래스
 - dao 패키지
 - WeatherCalculatorListManager 클래스 – ArrayList<WeatherCalculator> 사용
 - WeatherCalculatorSetManager 클래스 – HashSet<WeatherCalculator> 사용
 - controller 패키지
 - Lab5 & UserInput 클래스

과제 Lab5 (Interface, Collection, Package)

- WeatherCalculatorListManager, WeatherCalculatorSetManager 클래스
 - WeatherCalculatorListManager 클래스

```
public class WeatherCalculatorListManager {  
    private List<WeatherCalculator> calculators = null;  
    public WeatherCalculatorListManager() {  
        this.calculators = new ArrayList<> ();  
    }  
}
```

- WeatherCalculatorMapManager 클래스

```
public class WeatherCalculatorSetManager {  
    private Set<WeatherCalculator> calculators = null;  
    public WeatherCalculatorSetManager() {  
        this.calculators = new HashSet<> ();  
    }  
}
```

과제 Lab5 (Interface, Collection, Package)

□ WeatherCalculatorListManager 클래스

- `public void add(WeatherCalculator c) // 리스트에 새로 추가`
- `public void remove(WeatherCalculator c) // 리스트에서 해당 calculator 삭제`
- `public void set(int index, WeatherCalculator c) // 리스트에서 index에 calculator로 교체`
- `Public WeatherCalculator get(int index) // 리스트에서 index의 calculator 반환`
- `public int indexOf(WeatherCalculator c) // 리스트에서 해당 calculator의 index 반환`
- `public int size() { } // 리스트의 크기 반환`
- `public void print() { } // 전체 리스트 프린트`
- `public List<WeatherCalculator> select(Predicate<? super WeatherCalculator> predicate) // 리스트에서 해당 predicate 조건에 맞는 모든 calculator 리스트 반환`
- `public static WeatherCalculator getRandom(List<WeatherCalculator> list) // 인자로 넘겨준 리스트에서 랜덤하게 하나 선택해서 반환`

과제 Lab5 (Interface, Collection, Package)

- WeatherCalculatorSetManager 클래스
 - `public void add(WeatherCalculator c) // 세트에 새로 추가`
 - `public void remove(WeatherCalculator c) // 세트에서 해당 c 삭제`
 - `public int size() // 세트의 크기 반환`
 - `public void print() // 세트 프린트`
 - `public Set<WeatherCalculator> select(Predicate<? super WeatherCalculator> predicate) // 리스트에서 해당 predicate 조건에 맞는 모든 calculator 세트 반환`
 - `public static WeatherCalculator getRandom(Set<WeatherCalculator> set) // 인자로 보 내준 셋에서 랜덤하게 하나 선택해서 반환`

과제 Lab5 (Interface, Collection, Package)

- WeatherData 와 WeatherCalculatorFactory 클래스를 작성한다
 - WeatherData 클래스
 - String dateTime
 - double temperature
 - double relativeHumidity
 - double windVelocity
 - 생성자, Getter/Setter, toString()
 - WeatherCalculatorFactory 클래스
 - **public static WeatherCalculator getInstance(Mode mode);** // Mode에 따른 DewPointCalculator, WindChillTemperatureCalculator, HeatIndexCalculator, DiscomfortIndexCalculator, YourClass 객체 반환 메소드
 - **public static WeatherCalculator getInstance(Mode mode, WeatherData data);** // Mode에 따른 DewPointCalculator(data), WindChillTemperatureCalculator(data), HeatIndexCalculator(data), DiscomfortIndexCalculator(data), YourClass(data) 객체 반환 메소드 추가

과제 Lab5 (Interface, Collection, Package)

- 인터페이스 IWeatherCalculator로 구현하는 방식
 - IWeatherCalculator 인터페이스 추가
 - `void printTable();` // 추상메소드
 - `void getUserInput();` // 추상메소드
 - `void calculate();` // 추상메소드
 - WeatherCalculator 추상클래스는 IWeatherCalculator 인터페이스를 상속
 - `protected WeatherData weatherData;` // 날씨데이터 사용
 - `protected double value;`
 - `protected WeatherCalculator(WeatherData weatherData) {`
 `this.weatherData = weatherData;`
 `calculate();` // 동적바인딩으로 자식클래스의 calculate() 호출
 `}`
 - `public final void process() { 내부 구현 같음 }` // 내부코드에서는 동적바인딩으로 자식클래스의 오버라이딩된 메소드 호출
 - `public static double fromFahrenheitToCelsius(double F) { 내부 구현 같음 }`
 - `public static double fromCelsiusToFahrenheit(double C) { 내부 구현 같음 }`

과제 Lab5 (Interface, Collection, Package)

- DewPointCalculator, DiscomfortIndexCalculator, HeatIndexCalculator, WindChillTemperatureCalculator, YourClass 클래스는 WeatherCalculator 추상클래스를 상속받아 weatherData를 사용하여 추상메소드 내부 구현
 - 각 클래스의 멤버필드는 weatherData 객체 사용

```
public WindChillTemperatureCalculator(WeatherData weatherData) {    super(weatherData);    }  
@Override  
public void calculate() {    this.value = calculate(weatherData.getTemperature(),  
weatherData.getWindVelocity());    }  
@Override  
public void printTable() { WCT 테이블 출력 }  
@Override  
public void getUserInput() { weatherData 사용자 입력 }  
@Override  
public String toString() {  
return "WindChillTemperatureCalculator [temperature=" + weatherData.getTemperature() + "  
windVelocity=" + weatherData.getWindVelocity()+ ", value=" + value + ", index=" +  
WindChillTemperatureIndex.getIndex(value) + "];"  
}
```

과제 Lab5 (Interface, Collection, Package)

- equals와 hashCode 오버라이딩하여 같은 데이터임을 명시함 – HashSet을 위해

@Override

```
public boolean equals(Object other) { // Object.equals overriding
    if (this == other) return true;
    if (other instanceof WindChillTemperatureCalculator) {
        WindChillTemperatureCalculator that = (WindChillTemperatureCalculator) other;
        return this.weatherData.getTemperature() == that.weatherData.getTemperature() &&
            this.weatherData.getWindVelocity() == that.weatherData.getWindVelocity() &&
            this.getValue() == that.getValue();
    }
    return false;
}
```

@Override

```
public int hashCode() { // Object.hashCode overriding
    return Objects.hash(weatherData.getTemperature(), weatherData.getWindVelocity(), getValue());
}
```

과제 제출

- Lab5 클래스를 테스트하고 더 많은 테스트를 해본다. (Lab5.java 제공)

```
static WeatherData[] weatherData = {  
    new WeatherData("2019-01-01", 30.38, 46, 4.0265), new WeatherData("2019-02-01", 33.8, 47, 4.0265),  
    new WeatherData("2019-03-01", 44.78, 51, 4.6976), new WeatherData("2019-04-01", 53.78, 51, 4.2502),  
    new WeatherData("2019-05-01", 66.92, 47, 4.6976), new WeatherData("2019-06-01", 72.5, 61, 3.8028),  
    new WeatherData("2019-07-01", 78.62, 69, 4.0265), new WeatherData("2019-08-01", 80.96, 69, 3.5791),  
    new WeatherData("2019-09-01", 72.68, 65, 4.9213), new WeatherData("2019-10-01", 61.52, 62, 4.6976),  
    new WeatherData("2019-11-01", 45.68, 56, 4.9213), new WeatherData("2019-12-01", 34.52, 58, 4.6976),  
    new WeatherData("2020-01-01", 34.88, 56, 4.6976), new WeatherData("2020-02-01", 36.5, 58, 5.1450),  
    new WeatherData("2020-03-01", 45.86, 46, 5.5924), new WeatherData("2020-04-01", 51.98, 50, 6.7108),  
    new WeatherData("2020-05-01", 64.4, 67, 5.3687), new WeatherData("2020-06-01", 75.02, 68, 5.1450),  
    new WeatherData("2020-07-01", 75.38, 77, 5.3687), new WeatherData("2020-08-01", 79.7, 85, 5.1450),  
    new WeatherData("2020-09-01", 70.52, 71, 5.5924), new WeatherData("2020-10-01", 57.74, 61, 4.6976),  
    new WeatherData("2020-11-01", 46.4, 64, 4.9213), new WeatherData("2020-12-01", 31.46, 58, 4.9213)  
};
```

- Lab5과 보고서 전체를 묶어서 e-learning에 과제 제출
 - Lab4와 Lab5의 방식을 비교 분석한다.
 - 배열과 ArrayList와 HashSet의 차이점을 비교 분석한다.