

JAVA 프로그래밍 I

514760
2026년 봄학기
5/6/2026
박경신

Weather Data

```
public static String[][] data = {  
    // === [1-15] 극단적 기온 (Extreme Temperature: 폭염 및 한파) ===  
    {"Astana", "Kazakhstan", "51.17", "71.43", "2026-01-28", "Snow", "-26", "-16"},  
    {"Ottawa, Ontario", "Canada", "45.3225", "-75.669167", "2026-01-28", "Sunny Periods", "-21", "-11"},  
    {"South Bend, Indiana", "United States of America", "41.7", "-86.32", "2026-01-28", "Snow", "-19", "-10"},  
    {"Canberra", "Australia", "-35.308889", "149.200278", "2026-01-28", "Partly Cloudy", "15", "42"},  
    {"Riyadh", "Saudi Arabia", "24.72", "46.73", "2026-01-28", "Fine", "27", "39"},  
    {"Bamako", "Mali", "12.533333", "-7.95", "2026-01-28", "Sunny", "17", "35"},  
    {"Seoul", "Republic of Korea", "37.57", "126.97", "2026-01-28", "Clear", "-10", "-2"},  
    {"Ulaanbaatar", "Mongolia", "47.916667", "106.916667", "2026-01-28", "Clear", "-28", "-15"},  
    {"Kuwait City", "Kuwait", "29.38", "47.98", "2026-01-28", "Fine", "9", "23"},  
    {"Winnipeg, Manitoba", "Canada", "49.91", "-97.23", "2026-01-28", "Snow", "-22", "-14"},  
    {"Phoenix, Arizona", "United States of America", "33.45", "-112.07", "2026-01-28", "Fine", "10", "36"},  
    {"Tallinn", "Estonia", "59.398056", "24.602778", "2026-01-28", "Snow", "-11", "-7"},  
    {"Minneapolis, Minnesota", "United States of America", "44.88", "-93.22", "2026-01-28", "Mostly Cloudy", "-18", "-9"},  
    {"Baghdad", "Iraq", "33.33", "44.43", "2026-01-28", "Sunny", "8", "21"},  
    {"Stockholm", "Sweden", "59.33", "18.05", "2026-01-28", "Sunny Intervals", "-6", "-2"},  
    ...  
};
```

과제 Lab4 (Inheritance & Interface)

- **Weather 클래스** (Lab3)
- **WeatherIcon 열거형** (Lab3)
- **WeatherParser 클래스** (Lab3)
 - `public static Weather[] parse(String[][] data)`
- **WeatherFinder 클래스** (Lab3)
 - `public static Weather[] find(Weather[] data, WeatherIcon weatherIcon)`
 - 등등...

과제 Lab4 (Inheritance & Interface)

□ Searchable 인터페이스

- Weather[] search(Weather[] weathers); // 조건에 맞는 결과 반환
- String getSearchCriteria(); // 검색 조건 설명 문자열

□ Visualizable 인터페이스

- String getColorCode(); // 콘솔용 ANSI 컬러 코드 반환
- String getIconSymbol(); // 날씨 아이콘 (이모지 등)
- int getSeverityLevel(); // 1 ~ 10 사이의 위험도

□ Reportable 인터페이스

- WeatherReport generateReport(); // 상세 보고서 객체 반환

□ WeatherReport 클래스

- String title;
- String details
- String date;

과제 Lab4 (Inheritance & Interface)

□ 추상클래스 WeatherAlert implements Searchable, Reportable, Visualizable

- final Weather weather
- String alertLevel // LOW/MEDIUM/HIGH
- 생성자, getter, setter
- abstract String getAlertMessage();
- abstract boolean isTriggered();
- String getSummary() { .. } // isTriggered()이면 alertLevel, cityName, countryName, getAlertMessage() 출력; 아니면, alertLevel, cityName, 경보 없음 출력

과제 Lab4 (Inheritance & Interface)

□ ExtremeWeatherAlert extends WeatherAlert 구현

```
private static final WeatherIcon[] EXTREME_ICONS = {  
    WeatherIcon.THUNDERSTORMS,  
    WeatherIcon.DUST,  
    WeatherIcon.FOG  
};
```

```
private static boolean isExtreme(Weather weather) { .. 구현 } 극한 기상 경보 도시인지 판별  
public static String generateSummaryReport(Weather[] weathers) { .. 구현 } 극한 기상 경보별  
도시 출력  
그외 모든 추상 메소드 구현
```

과제 Lab4 (Inheritance & Interface)

□ PrecipitationAlert extends WeatherAlert 구현

```
private static final WeatherIcon[] PRECIPITATION_ICONS = {  
    WeatherIcon.RAIN,  
    WeatherIcon.SNOW,  
    WeatherIcon.SHOWERS,  
    WeatherIcon.ISOLATED_SHOWER  
};
```

```
private static boolean isPrecipitation(Weather weather) { .. 구현 } 강수 경보 도시인지 판별  
public static String generateSummaryReport(Weather[] weathers) { .. 구현 } 강수 기상 경보별  
도시 출력  
그외 모든 추상 메소드 구현
```

과제 Lab4 (Inheritance & Interface)

□ TemperatureAlert extends WeatherAlert 구현

```
private static final int HEAT_THRESHOLD = 35; // 폭염 기준 최고 기온 (°C)
private static final int COLD_THRESHOLD = -10; // 한파 기준 최저 기온 (°C)
private static boolean isHeat(Weather weather) { .. 구현 } // >35 인지
private static boolean isCold(Weather weather) { .. 구현 } // <-10 인지
public static String generateSummaryReport(Weather[] weathers) { .. 구현 } 폭염별/한파별 도시 정보 출력
```

과제 제출

- Lab4 메인에서는 data를 parse해서 Weather 객체의 배열로 만든 후,
 - Lab3 도시 날씨 검색 테스트를 하고,
 - Rome, Jakarta, Ottawa, 등 5~7개 여러 도시에 대한 ExtremeWeatherAlert, PrecipitationAlert, TemperatureAlert 날씨 경보 테스트를 수행한다.
 - 또한 각 클래스의 정적 요약 보고서인 generateSummaryReport를 호출하여 전체적인 날씨 경보를 확인한다.
- Lab4와 보고서 전체를 묶어서 e-learning에 과제 제출 (due by 5/12)
 - 본인이 원하는 코드 추가 구현 및 테스트
 - 보고서에 전체 코드 분석 및 구현 내용 자세히 설명