REPORT

LAB 번호 : HW3

과목 및 분반 : 자바프로그래밍 2분반

제출일 : 2025.04.17

학번 : 32203919 (컴퓨터공학과)

이름 : 장천명

DKU 단국대학교 DANKOOK UNIVERSITY

구현 방식 설명

먼저, 프로그램을 만들기 위해서 역할별로, 클래스를 6개로 설정했다.

1. **HW3**
2. **AirQualityData**
3. **AirQualityGrade**
4. **AirQualityParser**
5. **AirQuality**
6. **AirQualityFinder**

HW3 클래스

Main 함수가 있는 클래스이며, 데이터 처리 흐름 제어를 하며, 기능을 실행하며, 사용자에게 결과를 보여주는 인터페이스 역할을 한다.

AirQualityData 클래스

먼저, AirQualityData는 데이터 구조를 설계했으며, 서울시 25개 구의 대기질 데이터를 상수로 저장시켰다. 각 데이터는 (날짜, 측정소, PM25, PM10, O3, NO2, CO, SO2)의 형식이다.
AirQualityData.getAllData() 메소드는 모든 데이터를 2차원 배열로 반환한다.

AirQualityParser 클래스

원시 데이터(AirQualityData에 있는 2차원 배열)를 AirQuality 객체로 변환한다. 그리고 각 측정소의 데이터를 객체 형태로 관리한다.

AirQuality 클래스

개별 측정소의 대기질 데이터를 표현한다. 날짜, 측정소명, 각 대기질 지표를 속성으로 가진다.
printGrade() 메소드로 대기질 등급을 표현한다.

AirQualityFinder 클래스

대기질 데이터 검색 기능을 제공하며, find(), findByPm25(), findByPm10()을 통해 조건 검색을 할 수 있는 메소드를 구현했다.

AirQualityGrade 클래스

Your code를 위해서 넣은 클래스이다. 대기질 등급을 정의했으며, 각 대기질 지표별 등급 판정 메소드를 제공한다.

주요 코드 설명

```
public enum AirQualityData {
    AVERAGE("2025-04-10 08:00", "평균", "39", "54", "0.0176", "0.0320", "0.57", "0.0029"),
    GANGNAM("2025-04-10 08:00", "강남구", "44", "59", "0.0177", "0.0356", "0.50", "0.0024"),
    // ... 다른 구들의 데이터

    public static String[][] getAllData() {
        AirQualityData[] values = values();
        String[][] result = new String[values.length][];
        for (int i = 0; i < values.length; i++) {
            result[i] = values[i].getData();
        }
        return result;
    }
}
```

AirQualityData 클래스의 주요 코드로, enum으로 대기질 데이터를 상수로 저장했다. 그리고 getAllData() 코드로 모든 데이터를 2차원 배열로 반환하는 메소드가 있다. 즉, enum의 모든 상수값을 배열로 반환시켜준다.

```
public class AirQuality {
    private LocalDateTime dateTime;
    private String station;
    private double pm25;
    private double pm10;
    private double o3;
    private double no2;
    private double co;
    private double so2;

    public static void printGrades(AirQuality aq) {
        System.out.println(aq.getStation() + ":");
        System.out.println("PM2.5: " + AirQualityGrade.fromPm25(aq.getPm25()) +
                           " (" + aq.getPm25() + ")");
        // ... 다른 지표들의 등급 출력
    }
}
```

AirQuality 클래스는 개별 측정소의 대기질 데이터를 표현하는 클래스로, 모든 대기질 지표를 속

성으로 가진다. 그리고 printGrade를 사용하여 각 대기질 지표의 등급을 출력하는 메소드이다. 여기서 AirQualityGrade를 사용해 등급을 판정한다.

```
public static AirQuality[] find(AirQuality[] data, double minPm25, double maxPm25, double minPm10, double maxPm10) {
    ArrayList<AirQuality> list = new ArrayList<>();
    for (AirQuality aq : data) {
        if (aq.getPm25() >= minPm25 && aq.getPm25() <= maxPm25 &&
            aq.getPm10() >= minPm10 && aq.getPm10() <= maxPm10) {
            list.add(aq);
        }
    }
    return list.toArray(new AirQuality[0]);
```

```
public static AirQuality[] findByPm10(AirQuality[] data, double min, double max) {
    ArrayList<AirQuality> list = new ArrayList<>();
    for (AirQuality aq : data) {
        if (aq.getPm10() >= min && aq.getPm10() <= max) {
            list.add(aq);
        }
    }
    return list.toArray(new AirQuality[0]);
}
```

```
public static AirQuality[] findByPm25(AirQuality[] data, double min, double max) {
    ArrayList<AirQuality> list = new ArrayList<>();
    for (AirQuality aq : data) {
        if (aq.getPm25() >= min && aq.getPm25() <= max) {
            list.add(aq);
        }
    }
    return list.toArray(new AirQuality[0]);
}
```

AirQualityFinder 클래스에서 각각 PM25, PM10를 검색 그리고 둘 다 동시에 만족하는 데이터를 검색 메소드이다. Find 메소드는 PM25와 PM10을 조건을 둘 다 충족하는 데이터를 검색하며, FindByPm10은 PM10 조건을 충족시키는 데이터를 찾으며, FindByPm25는 PM25 조건을 충족시키는 데이터를 찾는다.

```java
public enum AirQualityGrade {
    GOOD, MODERATE, UNHEALTHY;

    public static AirQualityGrade fromPm25(double value) {
        if (value <= 30) return GOOD;
        else if (value <= 80) return MODERATE;
        else return UNHEALTHY;
    }

    public static AirQualityGrade fromPm10(double value) {
        if (value <= 50) return GOOD;
        else if (value <= 100) return MODERATE;
        else return UNHEALTHY;
    }
    // ... 다른 지표들의 등급 판정 메소드
}
```

Your code를 위한 AirQualityGrade의 대기질 등급을 정의하는 enum이며, 각 대기질 지표별로 등급을 판정하는 메소드이다. 기준값에 따라 GOOD, MODERATE, UNHEALTHY 등급을 반환한다. 참고로 Your code는 지표들의 등급을 판정하는 것으로 하였다.

```java
public class AirQualityParser {
    public static AirQuality[] parse(String[][] data) {
        AirQuality[] result = new AirQuality[data.length];
        for (int i = 0; i < data.length; i++) {
            result[i] = new AirQuality(
                LocalDateTime.parse(data[i][0],
                    DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm")),
                data[i][1],
                Double.parseDouble(data[i][2]),
                Double.parseDouble(data[i][3]),
                Double.parseDouble(data[i][4]),
                Double.parseDouble(data[i][5]),
                Double.parseDouble(data[i][6]),
                Double.parseDouble(data[i][7])
            );
        }
        return result;
    }
}
```

AirQualityParser 클래스의 AirQuality 메소드는 문자열 배열을 AirQuality 객체로 변환한다. 날짜 형식 변환 및 숫자 데이터 파싱을 하며, 모든 데이터를 객체로 변환하여 반환한다.

실행 결과 화면

```
String[][] data = AirQualityData.getAllData();
```

main함수의 흐름을 따라가보면, 먼저 AirQualityData enum에서 모든 데이터들을 2차원 배열로 가져온다.

```
AirQuality[] parsed = AirQualityParser.parse(data);
```

2차원 배열 데이터들을 AirQuality 객체 배열로 바꾼다. 그러면 각 행의 데이터가 하나의 AirQuality 객체로 생성된다.

```
System.out.println(x:"== 전체 데이터 출력 ==");
for (AirQuality aq : parsed) {
    System.out.println(aq);
}
```

AirQuality 객체를 전부 출력한다.

```
== 전체 데이터 출력 ==
AirQuality{datetime=2025-04-10T08:00, station='평균', pm25=39.0, pm10=54.0, o3=0.0176, no2=0.032, co=0.57, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='강남구', pm25=44.0, pm10=59.0, o3=0.0177, no2=0.0356, co=0.5, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='강동구', pm25=41.0, pm10=56.0, o3=0.0116, no2=0.0387, co=0.77, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='강북구', pm25=31.0, pm10=40.0, o3=0.0282, no2=0.0204, co=0.49, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='강서구', pm25=41.0, pm10=61.0, o3=0.0095, no2=0.0422, co=0.56, so2=0.0039}
AirQuality{datetime=2025-04-10T08:00, station='관악구', pm25=32.0, pm10=42.0, o3=0.0141, no2=0.0407, co=0.62, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='광진구', pm25=38.0, pm10=52.0, o3=0.0165, no2=0.0361, co=0.59, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='구로구', pm25=50.0, pm10=57.0, o3=0.017, no2=0.0299, co=0.47, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='금천구', pm25=36.0, pm10=55.0, o3=0.0102, no2=0.0481, co=0.56, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='노원구', pm25=36.0, pm10=51.0, o3=0.0161, no2=0.0333, co=0.58, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='도봉구', pm25=34.0, pm10=50.0, o3=0.0272, no2=0.0225, co=0.61, so2=0.0033}
AirQuality{datetime=2025-04-10T08:00, station='동대문구', pm25=41.0, pm10=60.0, o3=0.0207, no2=0.0261, co=0.56, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='동작구', pm25=39.0, pm10=54.0, o3=0.0269, no2=0.0296, co=0.49, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='마포구', pm25=40.0, pm10=46.0, o3=0.0219, no2=0.0233, co=0.5, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='서대문구', pm25=42.0, pm10=52.0, o3=0.0162, no2=0.0255, co=0.74, so2=0.0041}
AirQuality{datetime=2025-04-10T08:00, station='서초구', pm25=47.0, pm10=68.0, o3=0.0143, no2=0.0363, co=0.5, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='성동구', pm25=42.0, pm10=56.0, o3=0.0143, no2=0.0354, co=0.52, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='성북구', pm25=33.0, pm10=49.0, o3=0.0239, no2=0.0235, co=0.66, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='송파구', pm25=32.0, pm10=49.0, o3=0.0139, no2=0.0413, co=0.51, so2=0.0038}
AirQuality{datetime=2025-04-10T08:00, station='양천구', pm25=43.0, pm10=62.0, o3=0.01, no2=0.0422, co=0.6, so2=0.0032}
AirQuality{datetime=2025-04-10T08:00, station='영등포구', pm25=42.0, pm10=51.0, o3=0.0149, no2=0.0342, co=0.55, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='용산구', pm25=41.0, pm10=56.0, o3=0.0155, no2=0.0323, co=0.56, so2=0.0034}
AirQuality{datetime=2025-04-10T08:00, station='은평구', pm25=38.0, pm10=53.0, o3=0.0108, no2=0.0272, co=0.68, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='종로구', pm25=43.0, pm10=59.0, o3=0.0246, no2=0.0243, co=0.56, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='중구', pm25=43.0, pm10=53.0, o3=0.028, no2=0.0225, co=0.56, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='중랑구', pm25=34.0, pm10=53.0, o3=0.0158, no2=0.0281, co=0.53, so2=0.0025}
```

모든 객체를 출력한 값이다.

```java
// PM2.5와 PM10 농도 범위에 따른 데이터 검색 및 출력
System.out.println(x:"\n== PM25 20~50, PM10 40~60 조건 검색 ==");
AirQuality[] filtered = AirQualityFinder.find(parsed, minPm25:20, maxPm25:50, minPm10:40, maxPm10:60);
for (AirQuality aq : filtered) {
    System.out.println(aq);
}

// PM2.5 농도만을 기준으로 검색 (20~50)
System.out.println(x:"\n== PM25 20~50 조건 검색 ==");
AirQuality[] filteredByPm25 = AirQualityFinder.findByPm25(parsed, min:20, max:50);
for (AirQuality aq : filteredByPm25) {
    System.out.println(aq);
}

// PM10 농도만을 기준으로 검색 (40~60)
System.out.println(x:"\n== PM10 40~60 조건 검색 ==");
AirQuality[] filteredByPm10 = AirQualityFinder.findByPm10(parsed, min:40, max:60);
for (AirQuality aq : filteredByPm10) {
    System.out.println(aq);
}
```

PM 조건에 따라 검색하도록 했다. 첫 번째는 PM10, PM25가 중복되는 조건으로, 두번째는 PM25 조건으로, 세번째는 PM10조건으로 검색하도록 했으며 각각 결과를 출력한다.

```
== PM25 20~50, PM10 40~60 조건 검색 ==
AirQuality{datetime=2025-04-10T08:00, station='평균', pm25=39.0, pm10=54.0, o3=0.0176, no2=0.032, co=0.57, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='강남구', pm25=44.0, pm10=59.0, o3=0.0177, no2=0.0356, co=0.5, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='강동구', pm25=41.0, pm10=56.0, o3=0.0116, no2=0.0387, co=0.77, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='강북구', pm25=31.0, pm10=40.0, o3=0.0282, no2=0.0204, co=0.49, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='관악구', pm25=32.0, pm10=42.0, o3=0.0141, no2=0.0407, co=0.62, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='광진구', pm25=38.0, pm10=52.0, o3=0.0165, no2=0.0361, co=0.59, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='구로구', pm25=50.0, pm10=57.0, o3=0.017, no2=0.0299, co=0.47, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='금천구', pm25=36.0, pm10=55.0, o3=0.0102, no2=0.0481, co=0.56, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='노원구', pm25=36.0, pm10=51.0, o3=0.0161, no2=0.0333, co=0.58, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='도봉구', pm25=34.0, pm10=50.0, o3=0.0272, no2=0.0225, co=0.61, so2=0.0033}
AirQuality{datetime=2025-04-10T08:00, station='동대문구', pm25=41.0, pm10=60.0, o3=0.0207, no2=0.0261, co=0.56, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='동작구', pm25=39.0, pm10=54.0, o3=0.0269, no2=0.0296, co=0.49, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='마포구', pm25=40.0, pm10=46.0, o3=0.0219, no2=0.0233, co=0.5, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='서대문구', pm25=42.0, pm10=52.0, o3=0.0162, no2=0.0255, co=0.74, so2=0.0041}
AirQuality{datetime=2025-04-10T08:00, station='성동구', pm25=42.0, pm10=56.0, o3=0.0143, no2=0.0354, co=0.52, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='성북구', pm25=33.0, pm10=49.0, o3=0.0239, no2=0.0235, co=0.66, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='송파구', pm25=32.0, pm10=49.0, o3=0.0139, no2=0.0413, co=0.51, so2=0.0038}
AirQuality{datetime=2025-04-10T08:00, station='영등포구', pm25=42.0, pm10=51.0, o3=0.0149, no2=0.0342, co=0.55, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='용산구', pm25=41.0, pm10=56.0, o3=0.0155, no2=0.0323, co=0.56, so2=0.0034}
AirQuality{datetime=2025-04-10T08:00, station='은평구', pm25=38.0, pm10=53.0, o3=0.0108, no2=0.0272, co=0.68, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='종로구', pm25=43.0, pm10=59.0, o3=0.0246, no2=0.0243, co=0.56, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='중구', pm25=43.0, pm10=53.0, o3=0.028, no2=0.0225, co=0.56, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='중랑구', pm25=34.0, pm10=53.0, o3=0.0158, no2=0.0281, co=0.53, so2=0.0025}
```

```
== PM25 20~50 조건 검색 ==
AirQuality{datetime=2025-04-10T08:00, station='평균', pm25=39.0, pm10=54.0, o3=0.0176, no2=0.032, co=0.57, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='강남구', pm25=44.0, pm10=59.0, o3=0.0177, no2=0.0356, co=0.5, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='강동구', pm25=41.0, pm10=56.0, o3=0.0116, no2=0.0387, co=0.77, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='강북구', pm25=31.0, pm10=40.0, o3=0.0282, no2=0.0204, co=0.49, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='강서구', pm25=41.0, pm10=61.0, o3=0.0095, no2=0.0422, co=0.56, so2=0.0039}
AirQuality{datetime=2025-04-10T08:00, station='관악구', pm25=32.0, pm10=42.0, o3=0.0141, no2=0.0407, co=0.62, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='광진구', pm25=38.0, pm10=52.0, o3=0.0165, no2=0.0361, co=0.59, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='구로구', pm25=50.0, pm10=57.0, o3=0.017, no2=0.0299, co=0.47, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='금천구', pm25=36.0, pm10=55.0, o3=0.0102, no2=0.0481, co=0.56, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='노원구', pm25=36.0, pm10=51.0, o3=0.0161, no2=0.0333, co=0.58, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='도봉구', pm25=34.0, pm10=50.0, o3=0.0272, no2=0.0225, co=0.61, so2=0.0033}
AirQuality{datetime=2025-04-10T08:00, station='동대문구', pm25=41.0, pm10=60.0, o3=0.0207, no2=0.0261, co=0.56, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='동작구', pm25=39.0, pm10=54.0, o3=0.0269, no2=0.0296, co=0.49, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='마포구', pm25=40.0, pm10=46.0, o3=0.0219, no2=0.0233, co=0.5, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='서대문구', pm25=42.0, pm10=52.0, o3=0.0162, no2=0.0255, co=0.74, so2=0.0041}
AirQuality{datetime=2025-04-10T08:00, station='서초구', pm25=47.0, pm10=68.0, o3=0.0143, no2=0.0363, co=0.5, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='성동구', pm25=42.0, pm10=56.0, o3=0.0143, no2=0.0354, co=0.52, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='성북구', pm25=33.0, pm10=49.0, o3=0.0239, no2=0.0235, co=0.66, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='송파구', pm25=32.0, pm10=49.0, o3=0.0139, no2=0.0413, co=0.51, so2=0.0038}
AirQuality{datetime=2025-04-10T08:00, station='양천구', pm25=43.0, pm10=62.0, o3=0.01, no2=0.0422, co=0.6, so2=0.0032}
AirQuality{datetime=2025-04-10T08:00, station='영등포구', pm25=42.0, pm10=51.0, o3=0.0149, no2=0.0342, co=0.55, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='용산구', pm25=41.0, pm10=56.0, o3=0.0155, no2=0.0323, co=0.56, so2=0.0034}
AirQuality{datetime=2025-04-10T08:00, station='은평구', pm25=38.0, pm10=53.0, o3=0.0108, no2=0.0272, co=0.68, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='종로구', pm25=43.0, pm10=59.0, o3=0.0246, no2=0.0243, co=0.56, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='중구', pm25=43.0, pm10=53.0, o3=0.028, no2=0.0225, co=0.56, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='중랑구', pm25=34.0, pm10=53.0, o3=0.0158, no2=0.0281, co=0.53, so2=0.0025}
```

```
== PM10 40~60 조건 검색 ==
AirQuality{datetime=2025-04-10T08:00, station='평균', pm25=39.0, pm10=54.0, o3=0.0176, no2=0.032, co=0.57, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='강남구', pm25=44.0, pm10=59.0, o3=0.0177, no2=0.0356, co=0.5, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='강동구', pm25=41.0, pm10=56.0, o3=0.0116, no2=0.0387, co=0.77, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='강북구', pm25=31.0, pm10=40.0, o3=0.0282, no2=0.0204, co=0.49, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='관악구', pm25=32.0, pm10=42.0, o3=0.0141, no2=0.0407, co=0.62, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='광진구', pm25=38.0, pm10=52.0, o3=0.0165, no2=0.0361, co=0.59, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='구로구', pm25=50.0, pm10=57.0, o3=0.017, no2=0.0299, co=0.47, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='금천구', pm25=36.0, pm10=55.0, o3=0.0102, no2=0.0481, co=0.56, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='노원구', pm25=36.0, pm10=51.0, o3=0.0161, no2=0.0333, co=0.58, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='도봉구', pm25=34.0, pm10=50.0, o3=0.0272, no2=0.0225, co=0.61, so2=0.0033}
AirQuality{datetime=2025-04-10T08:00, station='동대문구', pm25=41.0, pm10=60.0, o3=0.0207, no2=0.0261, co=0.56, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='동작구', pm25=39.0, pm10=54.0, o3=0.0269, no2=0.0296, co=0.49, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='마포구', pm25=40.0, pm10=46.0, o3=0.0219, no2=0.0233, co=0.5, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='서대문구', pm25=42.0, pm10=52.0, o3=0.0162, no2=0.0255, co=0.74, so2=0.0041}
AirQuality{datetime=2025-04-10T08:00, station='성동구', pm25=42.0, pm10=56.0, o3=0.0143, no2=0.0354, co=0.52, so2=0.0025}
AirQuality{datetime=2025-04-10T08:00, station='성북구', pm25=33.0, pm10=49.0, o3=0.0239, no2=0.0235, co=0.66, so2=0.0031}
AirQuality{datetime=2025-04-10T08:00, station='송파구', pm25=32.0, pm10=49.0, o3=0.0139, no2=0.0413, co=0.51, so2=0.0038}
AirQuality{datetime=2025-04-10T08:00, station='영등포구', pm25=42.0, pm10=51.0, o3=0.0149, no2=0.0342, co=0.55, so2=0.0026}
AirQuality{datetime=2025-04-10T08:00, station='용산구', pm25=41.0, pm10=56.0, o3=0.0155, no2=0.0323, co=0.56, so2=0.0034}
AirQuality{datetime=2025-04-10T08:00, station='은평구', pm25=38.0, pm10=53.0, o3=0.0108, no2=0.0272, co=0.68, so2=0.0028}
AirQuality{datetime=2025-04-10T08:00, station='종로구', pm25=43.0, pm10=59.0, o3=0.0246, no2=0.0243, co=0.56, so2=0.0029}
AirQuality{datetime=2025-04-10T08:00, station='중구', pm25=43.0, pm10=53.0, o3=0.028, no2=0.0225, co=0.56, so2=0.0024}
AirQuality{datetime=2025-04-10T08:00, station='중랑구', pm25=34.0, pm10=53.0, o3=0.0158, no2=0.0281, co=0.53, so2=0.0025}
```

```java
// 각 측정소의 대기질 등급 출력
System.out.println(x:"\n== 각 측정소 등급 출력 ==");
for (AirQuality aq : parsed) {
    AirQuality.printGrades(aq);
}
```

printGrades 메소드를 이용하여 각 객체의 등급을 출력하도록 했다.

```java
public static void printGrades(AirQuality aq) {
    System.out.println("측정소: " + aq.getMeasurementStationName());
    System.out.println("측정시간: " + aq.getMeasurementDateTime());
    System.out.println("PM2.5 등급: " + AirQualityGrade.fromPm25(aq.getPm25()));
    System.out.println("PM10 등급: " + AirQualityGrade.fromPm10(aq.getPm10()));
    System.out.println("O3 등급: " + AirQualityGrade.fromO3(aq.getO3()));
    System.out.println("NO2 등급: " + AirQualityGrade.fromNo2(aq.getNo2()));
    System.out.println("CO 등급: " + AirQualityGrade.fromCo(aq.getCo()));
    System.out.println("SO2 등급: " + AirQualityGrade.fromSo2(aq.getSo2()));
    System.out.println(x:"-----------------------------------------");
}
```

이 함수가 호출되어서, 객체의 속성들에 따라 등급이 출력된다.

```
== 각 측정소 등급 출력 ==
측정소: 평균
측정시간: 2025-04-10T08:00
PM2.5 등급: MODERATE
PM10 등급: MODERATE
O3 등급: GOOD
NO2 등급: MODERATE
CO 등급: GOOD
SO2 등급: GOOD
-----------------------------------------
측정소: 강남구
측정시간: 2025-04-10T08:00
PM2.5 등급: MODERATE
PM10 등급: MODERATE
O3 등급: GOOD
NO2 등급: MODERATE
CO 등급: GOOD
SO2 등급: GOOD
-----------------------------------------
측정소: 강동구
측정시간: 2025-04-10T08:00
PM2.5 등급: MODERATE
PM10 등급: MODERATE
O3 등급: GOOD
NO2 등급: MODERATE
CO 등급: GOOD
SO2 등급: GOOD
-----------------------------------------
```