

REPORT



LAB 번호 : LAB4

과목 및 분반 : 자바프로그래밍 2분반

제출일 : 2025.05.20

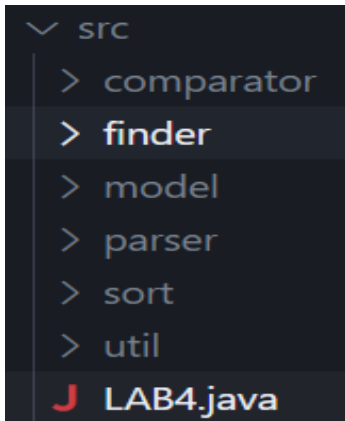
학번 : 32203919 (컴퓨터공학과)

이름 : 장천명



1. 전체 흐름 및 프로젝트 구성

1. String[][] 형식의 데이터를 객체 배열로 변환(Parser를 통해 이루어짐)
2. 기준에 따른 객체 필터링(Finder를 통해 이루어짐)
3. Comparator을 이용한 정렬(Sorter를 통해 이루어짐)
4. 테스트(main이 포함된 LAB4를 통해 이루어짐)



2. 주요 클래스 및 역할 상세 분석

2.1 model 패키지: 도메인 클래스

NaturalHazard.java (추상 클래스)

- 역할: 화산/지진 등 다양한 재해 객체의 공통 기반 제공 (추상화)

Volcano.java / Earthquake.java

- Volcano는 NaturalHazard를 상속받아 name, country, elevation, type, vei(폭발지수)를 추가
- Earthquake는 depth, magnitude(규모)를 추가
- 각 클래스는 오버라이딩된 toString()으로 세부정보 출력

2.2 parser 패키지: 데이터 파싱

VolcanoParser, EarthquakeParser

- NOAA 데이터는 String[][] 배열 형태 → 각 행을 Volcano, Earthquake 객체로 파싱
 - 결과 배열을 반환 (Volcano[], Earthquake[])
-

2.3 finder 패키지: 인터페이스 기반 필터링 전략

- 다양한 기준(국가, 고도, 유형, VEI, 규모, 깊이, 연도 등)을 클래스화
-

2.4 comparator 패키지: 정렬 기준 정의

- `Arrays.sort(arr, comparator)`에 넘길 비교 기준 클래스 구현
 - 총 18개 구현체로 속성별 다양한 정렬 지원 (예: 연도, 국가, 위도 등)
-

2.5 sort 패키지: 정렬 호출

VolcanoSorter, EarthquakeSorter

- index 번호를 통해 정렬 기준을 선택하도록 구현
-

2.6 LAB4.java: 실행 클래스

- 데이터 파싱 테스트
 - 필터링 및 정렬 테스트
 - 결과 출력
-

2.7 util 패키지: 출력 유틸(Your Code)

- 결과 출력할 때, 가독성을 올리도록 구현
-

3. 클래스 설명

NaturalHazard.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package model;

public abstract class NaturalHazard {
    private int year;
    private int month;
    private int day;
    private String location;
    private double latitude;
    private double longitude;

    public NaturalHazard(int year, int month, int day, String location, double latitude, double longitude) {
        this.year = year;
        this.month = month;
        this.day = day;
        this.location = location;
        this.latitude = latitude;
        this.longitude = longitude;
    }
}
```

NaturalHazard는 Earthquake와 Volcano의 기본이 되는 추상 클래스이다.

Earthquake와 Volcano는 공통적으로 year, month, day, location, latitude, longitude를 가진다.

+ getter/setter, +toString

Earthquake.java 코드

```
public class Earthquake extends NaturalHazard {
    private int depth;
    private double magnitude;

    public Earthquake(int year, int month, int day, String location,
        double latitude, double longitude, int depth, double magnitude) {
        super(year, month, day, location, latitude, longitude);
        this.depth = depth;
        this.magnitude = magnitude;
    }
}
```

NaturalHazard를 상속받는 지진 정보 클래스이다. 추가 속성으로는 depth와 magnitude가 있다.

Volcano.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package model;

public class Volcano extends NaturalHazard {
    private String name;
    private String country;
    private int elevation;
    private String type;
    private int vei; // Volcanic Explosive Index

    public Volcano(int year, int month, int day, String name, String location, String country,
        double latitude, double longitude, int elevation, String type, int vei) {
        super(year, month, day, location, latitude, longitude);
        this.name = name;
        this.country = country;
        this.elevation = elevation;
        this.type = type;
        this vei = vei;
    }
}
```

NaturalHazard를 상속받는 화산 정보 클래스이다. 추가 속성으로는 elevation, type, vei가 있다.

VolcanicExplosivityIndex.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package model;

public enum VolcanicExplosivityIndex {
    VEI_0(value:0),
    VEI_1(value:1),
    VEI_2(value:2),
    VEI_3(value:3),
    VEI_4(value:4),
    VEI_5(value:5),
    VEI_6(value:6),
    VEI_7(value:7),
    VEI_8(value:8);
    private final int value;
    VolcanicExplosivityIndex(int value) {
        this.value = value;
    }
    public int getValue() {
        return value;
    }
    public static VolcanicExplosivityIndex fromInt(int value) {
        for (VolcanicExplosivityIndex vei : values()) {
            if (vei.value == value) {
                return vei;
            }
        }
        throw new IllegalArgumentException("Invalid VEI value: " + value);
    }
}
```

화산 폭발 지수 열거형으로, 각 등급은 value 필드로 실제 값을 저장한다.

VolcanoParser.java

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package parser;

import model.Volcano;

public class VolcanoParser {
    public static Volcano[] parse(String[][] volcanoData) {
        Volcano[] volcanoes = new Volcano[volcanoData.length];

        for (int i = 0; i < volcanoData.length; i++) {
            String[] data = volcanoData[i];

            // 데이터 파싱
            int year = Integer.parseInt(data[0]);
            int month = Integer.parseInt(data[1]);
            int day = data[2].isEmpty() ? 0 : Integer.parseInt(data[2]); // 빈 문자열 처리
            String name = data[3];
            String region = data[4]; // 지역 정보 추가
            String country = data[5];
            double latitude = Double.parseDouble(data[6]);
            double longitude = Double.parseDouble(data[7]);
            int elevation = Integer.parseInt(data[8]);
            String type = data[9];
            int vei = Integer.parseInt(data[10]);

            // Volcano 객체 생성
            volcanoes[i] = new Volcano(year, month, day, name, region, country,
                latitude, longitude, elevation, type, vei);
        }

        return volcanoes;
    }
}
```

먼저, 2차원 문자열 배열로 된 화산 데이터를 Volcano 객체 배열로 변환 시킨다.

그리고 각 행의 데이터를 파싱하여 Volcano 객체 생성을 한다.

날짜, 이름, 위치, 국가, 위도, 경도, 고도, 유형, VEI 정보를 처리한다.

VolcanicExplosivityIndex 열거형을 사용하여 VEI 처리를 한다.(VolcanicExplosivityIndex.java)

EarthquakeParser.java

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package parser;

import model.Earthquake;

public class EarthquakeParser {
    public static Earthquake[] parse(String[][] earthquakeData) {
        Earthquake[] earthquakes = new Earthquake[earthquakeData.length];

        for (int i = 0; i < earthquakeData.length; i++) {
            String[] data = earthquakeData[i];

            // 데이터 파싱
            int year = Integer.parseInt(data[0]);
            int month = Integer.parseInt(data[1]);
            int day = data[2].isEmpty() ? 0 : Integer.parseInt(data[2]); // 빈 문자열 처리
            String location = data[3];
            double latitude = Double.parseDouble(data[4]);
            double longitude = Double.parseDouble(data[5]);
            int depth = Integer.parseInt(data[6]);
            double magnitude = Double.parseDouble(data[7]);

            // Earthquake 객체 생성
            earthquakes[i] = new Earthquake(year, month, day, location,
                                             |         |         |         |         |         |         |
                                             latitude, longitude, depth, magnitude);
        }

        return earthquakes;
    }
}
```

2차원 문자열 배열로 된 지진 데이터를 Earthquake 객체 배열로 변환한다.

각 행의 데이터를 파싱하여 Earthquake 객체 생성한다.

날짜, 이름, 위치, 위도, 경도, 깊이, 진도 정보를 처리한다.

INaturalHazard.java, IEarthquakeFinder.java, IVolcanoFinder.java

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package finder;

import model.NaturalHazard;

public interface INaturalHazardFinder {
    NaturalHazard[] find(NaturalHazard[] hazards);
}
```

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명  
package finder;
```

```
import model.Earthquake;
```

```
public interface IEarthquakeFinder {  
    |   Earthquake[] find(Earthquake[] earthquakes);  
}
```

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명  
package finder;
```

```
import model.Volcano;
```

```
public interface IVolcanoFinder {  
    |   Volcano[] find(Volcano[] volcanoes);  
}
```

INaturalFinder는 기본 인터페이스 정의이다.

IEarthquakeFinder, IVolcanoFinder는 화산과 지진에 대한 특화된 인터페이스 정의이다.

NaturalHazardElevationFinder.java , NaturalHazardYearFinder.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package finder;

import model.NaturalHazard;
import java.util.ArrayList;

public class NaturalHazardElevationFinder implements INaturalHazardFinder {
    private double minLat;
    private double maxLat;
    private double minLon;
    private double maxLon;

    public NaturalHazardElevationFinder(double minLat, double maxLat, double minLon, double maxLon) {
        this.minLat = minLat;
        this.maxLat = maxLat;
        this.minLon = minLon;
        this.maxLon = maxLon;
    }

    @Override
    public NaturalHazard[] find(NaturalHazard[] hazards) {
        ArrayList<NaturalHazard> result = new ArrayList<>();

        for (NaturalHazard hazard : hazards) {
            double latitude = hazard.getLatitude();
            double longitude = hazard.getLongitude();

            if (latitude >= minLat && latitude <= maxLat &&
                longitude >= minLon && longitude <= maxLon) {
                result.add(hazard);
            }
        }

        return result.toArray(new NaturalHazard[0]);
    }
}
```

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package finder;

import model.NaturalHazard;
import java.util.ArrayList;

public class NaturalHazardYearFinder implements INaturalHazardFinder {
    private int min;
    private int max;

    public NaturalHazardYearFinder(int min, int max) {
        this.min = min;
        this.max = max;
    }

    @Override
    public NaturalHazard[] find(NaturalHazard[] hazards) {
        ArrayList<NaturalHazard> result = new ArrayList<>();

        for (NaturalHazard hazard : hazards) {
            int year = hazard.getYear();
            if (year >= min && year <= max) {
                result.add(hazard);
            }
        }

        return result.toArray(new NaturalHazard[0]);
    }
}
```

공통 Finder 클래스들은 NaturalHazardYearFinder.java, NaturalHazardElevationFinder.java이며, INaturalHazardFinder 인터페이스를 구현했다.

특정 연도, 위도, 경도 범위 내의 자연재해를 찾는 Finder이다.

min과 max로 검색할 연도, 위도, 경도 범위를 지정한다.

ArrayList를 사용하여 결과를 동적으로 저장하고, getYear(), getLatitude(), getLongitude를 통해 각 자연재해의 연도, 위도, 경도를 확인이 가능하다.

VolcanoCountryFinder, VolcanoElevationFinder, VolcanoTypeFinder, VolcanoVeiFinder 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package finder;

import model.Volcano;
import java.util.ArrayList;

public class VolcanoCountryFinder implements IVolcanoFinder {
    private String country;

    public VolcanoCountryFinder(String country) {
        this.country = country;
    }

    @Override
    public Volcano[] find(Volcano[] volcanoes) {
        ArrayList<Volcano> result = new ArrayList<>();

        for (Volcano volcano : volcanoes) {
            if (volcano.getCountry().equals(country)) {
                result.add(volcano);
            }
        }

        return result.toArray(new Volcano[0]);
    }
}
```

// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영

```
package finder;
```

```
import model.Volcano;
```

```
import java.util.ArrayList;
```

```
public class VolcanoElevationFinder implements IVolcanoFinder {  
    private int min;  
    private int max;
```

```
    public VolcanoElevationFinder(int min, int max) {  
        this.min = min;  
        this.max = max;  
    }
```

```
    @Override
```

```
    public Volcano[] find(Volcano[] volcanoes) {  
        ArrayList<Volcano> result = new ArrayList<>();  
  
        for (Volcano volcano : volcanoes) {  
            int elevation = volcano.getElevation();  
            if (elevation >= min && elevation <= max) {  
                result.add(volcano);  
            }  
        }  
  
        return result.toArray(new Volcano[0]);  
    }  
}
```

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package finder;

import model.Volcano;
import java.util.ArrayList;

public class VolcanoTypeFinder implements IVolcanoFinder {
    private String type;

    public VolcanoTypeFinder(String type) {
        this.type = type;
    }

    @Override
    public Volcano[] find(Volcano[] volcanoes) {
        ArrayList<Volcano> result = new ArrayList<>();

        for (Volcano volcano : volcanoes) {
            if (volcano.getType().equals(type)) {
                result.add(volcano);
            }
        }

        return result.toArray(new Volcano[0]);
    }
}
```

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package finder;

import model.Volcano;
import model.VolcanicExplosivityIndex;
import java.util.ArrayList;

public class VolcanoVeiFinder implements IVolcanoFinder {
    private VolcanicExplosivityIndex vei;

    public VolcanoVeiFinder(VolcanicExplosivityIndex vei) {
        this.vei = vei;
    }

    @Override
    public Volcano[] find(Volcano[] volcanoes) {
        ArrayList<Volcano> result = new ArrayList<>();

        for (Volcano volcano : volcanoes) {
            if (volcano.getVei() == vei.getValue()) {
                result.add(volcano);
            }
        }

        return result.toArray(new Volcano[0]);
    }
}
```

이 네 개의 클래스는 IVolcanoFinder 인터페이스를 구현했으며, ArrayList를 사용한 동적 결과를 저장한다. 각각 country, elevation, type, vei 조건을 구현하였다. 조건을 만족하는 객체는 결과 리스트에 추가된다.

EarthquakeDepthFinder, EarthquakeMagnitudeFinder

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package finder;

import model.Earthquake;
import java.util.ArrayList;

public class EarthquakeDepthFinder implements IEarthquakeFinder {
    private int min;
    private int max;

    public EarthquakeDepthFinder(int min, int max) {
        this.min = min;
        this.max = max;
    }

    @Override
    public Earthquake[] find(Earthquake[] earthquakes) {
        ArrayList<Earthquake> result = new ArrayList<>();

        for (Earthquake earthquake : earthquakes) {
            int depth = earthquake.getDepth();
            if (depth >= min && depth <= max) {
                result.add(earthquake);
            }
        }

        return result.toArray(new Earthquake[0]);
    }
}
```

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
package finder;

import model.Earthquake;
import java.util.ArrayList;

public class EarthquakeMagnitudeFinder implements IEarthquakeFinder {
    private double min;
    private double max;

    public EarthquakeMagnitudeFinder(double min, double max) {
        this.min = min;
        this.max = max;
    }

    @Override
    public Earthquake[] find(Earthquake[] earthquakes) {
        ArrayList<Earthquake> result = new ArrayList<>();

        for (Earthquake earthquake : earthquakes) {
            double magnitude = earthquake.getMagnitude();
            if (magnitude >= min && magnitude <= max) {
                result.add(earthquake);
            }
        }

        return result.toArray(new Earthquake[0]);
    }
}
```

이 두 개의 클래스는 IEarthquakeFinder 인터페이스를 구현했으며, 마찬가지로 ArrayList를 사용하여 결과를 수집한다. 조건을 만족하는 객체는 결과 리스트에 추가된다.

EarthquakeSorter.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package sort;

import java.util.Arrays;
import comparator.*;
import model.Earthquake;

public class EarthquakeSorter {
    public static void sort(Earthquake[] earthquakes, int index) {
        switch (index) {
            case 0:
                Arrays.sort(earthquakes, new NaturalHazardYearComparator());
                break;
            case 1:
                Arrays.sort(earthquakes, new EarthquakeMonthComparator());
                break;
            case 2:
                Arrays.sort(earthquakes, new EarthquakeDayComparator());
                break;
            case 3:
                Arrays.sort(earthquakes, new EarthquakeLocationComparator());
                break;
            case 4:
                Arrays.sort(earthquakes, new EarthquakeLatitudeComparator());
                break;
            case 5:
                Arrays.sort(earthquakes, new EarthquakeLongitudeComparator());
                break;
            case 6:
                Arrays.sort(earthquakes, new EarthquakeDepthComparator());
                break;
            case 7:
                Arrays.sort(earthquakes, new EarthquakeMagnitudeComparator());
                break;
            default:
                throw new IllegalArgumentException("Invalid sort index: " + index);
        }
    }
}
```

지진 데이터를 정렬하는 클래스이다. 정적 메소드 sort를 제공하며, 두 개의 매개변수를 받는다. 첫 번째, 매개변수는 Earthquake[] earthquake로 정렬할 지진 배열이다. 두 번째는 int index로 정렬 기준을 선택하는 인덱스이다.

VolcanoSorter.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package sort;
import java.util.Arrays;
import comparator.*;
import model.Volcano;
public class VolcanoSorter {
    public static void sort(Volcano[] volcanoes, int index) {
        switch (index) {
            case 0:
                Arrays.sort(volcanoes, new NaturalHazardYearComparator());
                break;
            case 1:
                Arrays.sort(volcanoes, new VolcanoMonthComparator());
                break;
            case 2:
                Arrays.sort(volcanoes, new VolcanoDayComparator());
                break;
            case 3:
                Arrays.sort(volcanoes, new VolcanoNameComparator());
                break;
            case 4:
                Arrays.sort(volcanoes, new VolcanoLocationComparator());
                break;
            case 5:
                Arrays.sort(volcanoes, new VolcanoCountryComparator());
                break;
            case 6:
                Arrays.sort(volcanoes, new VolcanoLatitudeComparator());
                break;
            case 7:
                Arrays.sort(volcanoes, new VolcanoLongitudeComparator());
                break;
            case 8:
                Arrays.sort(volcanoes, new VolcanoElevationComparator());
                break;
        }
    }
}
```

화산 데이터를 정렬하는 클래스이다. 마찬가지로 정적 메소드 sort를 제공하며, 두 개의 매개변수를 받는다. 하나는 정렬할 화산 배열인 Volcano[] volcanoes이고, 두 번째는 정렬 기준을 선택하는 인덱스이다.

EarthquakeSorter와 VolcanoSorter는 공통적으로 Strategy Pattern을 사용하며, 자바의 Array.sort() 메소드를 사용하여 정렬을 수행한다.

NaturalHazardYearComparator.java 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천명
package comparator;

import java.util.Comparator;
import model.NaturalHazard;

public class NaturalHazardYearComparator implements Comparator<NaturalHazard> {
    @Override
    public int compare(NaturalHazard h1, NaturalHazard h2) {
        return Integer.compare(h1.getYear(), h2.getYear());
    }
}
```

Compare 메소드는 두 개의 NaturalHazard 객체를 매개변수로 받는다. 그리고 Integer.compare()를 사용해 두 객체의 연도를 비교한다. h1의 연도가 h2의 연도보다 작을 때 음수이고, 두 연도가 같으면 0이고, h1의 연도가 h2의 연도보다 클 때 양수이다.



나머지 Comparator 클래스들도 이와 유사하다. compare(매개변수1, 매개변수2)로 매개변수를 받고, Integer.compare()로 비교하는 방식으로 구현되어 있다.

PrintUtil.java (Your Code) 코드

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장천영
/* Your Code
| 출력할 때 출력 형식을 지정하는 클래스로 보기 좋게 표시되도록 함
*/
package util;

public class PrintUtil {
    public static void printSeparator() {
        System.out.println(x:"-----");
    }

    public static void printTitle(String title) {
        printSeparator();
        System.out.println(title);
        printSeparator();
    }

    public static void printCount(String type, int count) {
        System.out.printf(format:"총 %s 개수: %d개\n", type, count);
    }
}

```

main에서 구현 한 것들을 테스트하는데, 보기 좋게 만들기 위해 구현한 클래스이다.

printSeparator()를 구현해 구분선을 만들어서 , printTitle()을 구현해 제목을 붙여준다. 그리고 printCount()를 통해 지진과 화산 타입의 개수를 간략하게 알 수 있다.

LAB4.java(main)

```
// LAB4, 자바 프로그래밍, 2분반, 25.05.20, 32203919 장현영
import model.*;
import parser.*;
import finder.*;
import sort.*;
import util.PrintUtil;
import java.util.Arrays;

public class LAB4 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        // 데이터 생성
        String[][] volcanoData = {
            {"2020", "1", "12", "Taal", "Luzon-Philippines", "Philippines", "14.011", "120.998", "311", "Stratovolcano", "4"},
            {"2021", "3", "5", "Sangay", "Ecuador", "Ecuador", "-2.005", "-78.341", "5286", "Stratovolcano", "2"},
            {"2021", "3", "", "Pacaya", "Guatemala", "Guatemala", "14.382", "-90.601", "2569", "Complex volcano", "1"},
            {"2021", "4", "9", "Soufriere St. Vincent", "W Indies", "St. Vincent & the Grenadines", "13.330", "-61.180", "1220", "Stratovolcano", "4"},
            {"2021", "5", "19", "Stromboli", "Italy", "Italy", "38.789", "15.213", "924", "Stratovolcano", "2"},
            {"2021", "5", "22", "Nyiragongo", "Africa-C", "Congo, DRC", "-1.520", "29.250", "3470", "Stratovolcano", "2"},
            {"2021", "9", "19", "La Palma", "Canary Is", "Spain", "28.570", "-17.830", "2426", "Stratovolcano", "3"},
            {"2021", "12", "4", "Semeru", "Java", "Indonesia", "-8.108", "112.922", "3657", "Stratovolcano", "4"},
            {"2022", "1", "15", "Hunga Tonga-Hunga Ha'apai", "Tonga-SW Pacific", "Tonga", "-20.553", "-175.384", "114", "Submarine volcano", "5"},
            {"2022", "6", "22", "Popocatepetl", "Mexico", "Mexico", "19.023", "-98.622", "5393", "Stratovolcano", "2"},
            {"2022", "10", "9", "Stromboli", "Italy", "Italy", "38.789", "15.213", "924", "Stratovolcano", "2"},
            {"2022", "12", "4", "Stromboli", "Italy", "Italy", "38.789", "15.213", "924", "Stratovolcano", "2"},
            {"2022", "12", "4", "Semeru", "Java", "Indonesia", "-8.108", "112.922", "3657", "Stratovolcano", "4"},
            {"2023", "7", "7", "Bagana", "Bougainville-SW Paci", "Papua New Guinea", "-6.137", "155.196", "1855", "Lava cone", "2"},
            {"2024", "5", "11", "Marapi", "Sumatra", "Indonesia", "-0.380", "100.474", "2885", "Complex volcano", "2"},
            {"2024", "6", "3", "Kanlaon", "Philippines-C", "Philippines", "10.410", "123.130", "2422", "Stratovolcano", "3"},
            {"2024", "8", "17", "Sheveluch", "Kamchatka", "Russia", "56.653", "161.360", "3283", "Stratovolcano", "4"}
        };

        String[][] earthquakeData = {
            {"2020", "1", "28", "CUBA: GRANMA; CAYMAN IS; JAMAICA", "19.440", "-78.755", "10", "7.7"},
            {"2020", "3", "25", "RUSSIA: KURIL ISLANDS", "48.986", "157.693", "57", "7.5"},
            {"2020", "6", "18", "KERMADEC ISLANDS: S OF, RAOUL", "-33.294", "-177.838", "10", "7.4"},
            {"2020", "6", "23", "MEXICO: OAXACA", "16.029", "-95.901", "26", "7.4"},
            {"2020", "7", "17", "PAPUA NEW GUINEA: MOROBE, OROA", "-7.843", "147.766", "80", "7"},
            {"2020", "7", "22", "ALASKA", "55.030", "-158.522", "28", "7.8"},
            {"2020", "10", "19", "ALASKA", "54.608", "-159.655", "33", "7.6"},
            {"2020", "10", "30", "GREECE: SAMOS; TURKEY: IZMIR", "37.918", "26.790", "21", "7"},
            {"2021", "2", "10", "NEW CALEDONIA: LOYALTY ISLANDS", "-23.054", "171.601", "10", "7.7"},
            {"2021", "2", "13", "JAPAN: OFF FUKUSHIMA", "37.745", "141.749", "49", "7.1"},
            {"2021", "3", "4", "KERMADEC ISLANDS: S OF, RAOUL", "-29.613", "-177.843", "55", "7.4"},
            {"2021", "3", "4", "KERMADEC ISLANDS: SSE OF RAOUL ISLAND", "-29.740", "-177.267", "19", "8.1"},
            {"2021", "3", "20", "JAPAN: HONSHU ISLAND", "38.457", "141.648", "43", "7"}
        };
    }
}
```

화산과 지진 데이터를 2차원 문자열 배열로 정의한다.

```

// 데이터 파싱
Volcano[] volcanoes = VolcanoParser.parse(volcanoData);
Earthquake[] earthquakes = EarthquakeParser.parse(earthquakeData);

PrintUtil.printTitle(title:"Finder 테스트");
PrintUtil.printCount(type:"화산", volcanoes.length);
PrintUtil.printCount(type:"지진", earthquakes.length);

// Volcano Finder 테스트
PrintUtil.printTitle(title:"1. Volcano Country Finder 테스트");
VolcanoCountryFinder countryFinder = new VolcanoCountryFinder(country:"Italy");
Volcano[] italyVolcanoes = countryFinder.find(volcanoes);
System.out.println(x:"이탈리아 화산:");
for (Volcano v : italyVolcanoes) {
    System.out.println(v);
}

PrintUtil.printTitle(title:"2. Volcano Elevation Finder 테스트");
VolcanoElevationFinder elevationFinder = new VolcanoElevationFinder(min:3000, max:6000);
Volcano[] highVolcanoes = elevationFinder.find(volcanoes);
System.out.println(x:"고도 3000-6000m 화산:");
for (Volcano v : highVolcanoes) {
    System.out.println(v);
}

PrintUtil.printTitle(title:"3. Volcano Type Finder 테스트");
VolcanoTypeFinder typeFinder = new VolcanoTypeFinder(type:"Stratovolcano");
Volcano[] stratovolcanoes = typeFinder.find(volcanoes);
System.out.println(x:"성층화산:");
for (Volcano v : stratovolcanoes) {
    System.out.println(v);
}

PrintUtil.printTitle(title:"4. Volcano VEI Finder 테스트");
VolcanoVeiFinder veiFinder = new VolcanoVeiFinder(VolcanicExplosivityIndex.VEI_4);
Volcano[] vei4Volcanoes = veiFinder.find(volcanoes);
System.out.println(x:"VEI 4 화산:");
for (Volcano v : vei4Volcanoes) {
    System.out.println(v);
}

```

문자열 데이터를 Volcano와 Earthquake 객체 배열로 변환한다. 그리고 화산 검색 테스트를 한다.

테스트 내용은 국가별 검색 (이탈리아 화산), 고도별 검색 (3000-6000m), 유형별 검색 (성층화산), VEI별 검색 (VEI 4) 이다.

지진 검색 테스트 내용은 깊이별 검색 (10-30km), 규모별 검색 (8.0-9.0) 이다.

```

// Earthquake Finder 테스트
PrintUtil.printTitle(title:"5. Earthquake Depth Finder 테스트");
EarthquakeDepthFinder depthFinder = new EarthquakeDepthFinder(min:10, max:30);
Earthquake[] shallowEarthquakes = depthFinder.find(earthquakes);
System.out.println(x:"깊이 10-30km 지진:");
for (Earthquake e : shallowEarthquakes) {
    System.out.println(e);
}

PrintUtil.printTitle(title:"6. Earthquake Magnitude Finder 테스트");
EarthquakeMagnitudeFinder magnitudeFinder = new EarthquakeMagnitudeFinder(min:8.0, max:9.0);
Earthquake[] strongEarthquakes = magnitudeFinder.find(earthquakes);
System.out.println(x:"진도 8.0-9.0 지진:");
for (Earthquake e : strongEarthquakes) {
    System.out.println(e);
}

PrintUtil.printTitle(title:"Sorter 테스트");

// Volcano Sorter 테스트
PrintUtil.printTitle(title:"1. Volcano Sorter 테스트");
System.out.println(x:"연도 기준 정렬:");
VolcanoSorter.sort(volcanoes, index:0);
for (Volcano v : volcanoes) {
    System.out.println(v);
}

PrintUtil.printTitle(title:"VEI 기준 정렬");
VolcanoSorter.sort(volcanoes, index:10);
for (Volcano v : volcanoes) {
    System.out.println(v);
}

// Earthquake Sorter 테스트
PrintUtil.printTitle(title:"2. Earthquake Sorter 테스트");
System.out.println(x:"연도 기준 정렬:");
EarthquakeSorter.sort(earthquakes, index:0);
for (Earthquake e : earthquakes) {
    System.out.println(e);
}

PrintUtil.printTitle(title:"진도 기준 정렬");
EarthquakeSorter.sort(earthquakes, index:7);
for (Earthquake e : earthquakes) {
    System.out.println(e);
}

```

화산 정렬 테스트는 연도 기준 정렬과 VEI 기준 정렬이다.

지진 정렬 테스트는 연도 기준 정렬과 진도 기준 정렬이다.

4. 실행 결과창

```
-----
Finder 테스트
-----
총 화산 개수: 17개
총 지진 개수: 17개
-----
1. Volcano Country Finder 테스트
-----
이탈리아 화산:
Date: 2021-05-19, Location: Italy, Coordinates: (38.789, 15.213), Name: Stromboli, Country: Italy, Elevation: 924m, Type: Stratovolcano, VEI: 2
Date: 2022-10-09, Location: Italy, Coordinates: (38.789, 15.213), Name: Stromboli, Country: Italy, Elevation: 924m, Type: Stratovolcano, VEI: 2
Date: 2022-12-04, Location: Italy, Coordinates: (38.789, 15.213), Name: Stromboli, Country: Italy, Elevation: 924m, Type: Stratovolcano, VEI: 2
-----
2. Volcano Elevation Finder 테스트
-----
고도 3000-6000m 화산:
Date: 2021-03-05, Location: Ecuador, Coordinates: (-2.005, -78.341), Name: Sangay, Country: Ecuador, Elevation: 5286m, Type: Stratovolcano, VEI: 2
Date: 2021-05-22, Location: Africa-C, Coordinates: (-1.520, 29.250), Name: Nyiragongo, Country: Congo, DRC, Elevation: 3470m, Type: Stratovolcano, VEI: 2
Date: 2021-12-04, Location: Java, Coordinates: (-8.108, 112.922), Name: Semeru, Country: Indonesia, Elevation: 3657m, Type: Stratovolcano, VEI: 4
Date: 2022-06-22, Location: Mexico, Coordinates: (19.023, -98.622), Name: Popocatepetl, Country: Mexico, Elevation: 5393m, Type: Stratovolcano, VEI: 2
Date: 2022-12-04, Location: Java, Coordinates: (-8.108, 112.922), Name: Semeru, Country: Indonesia, Elevation: 3657m, Type: Stratovolcano, VEI: 4
Date: 2024-08-17, Location: Kamchatka, Coordinates: (56.653, 161.360), Name: Sheveluch, Country: Russia, Elevation: 3283m, Type: Stratovolcano, VEI: 4
-----
3. Volcano Type Finder 테스트
-----
성층화산:
Date: 2020-01-12, Location: Luzon-Philippines, Coordinates: (14.011, 120.998), Name: Taal, Country: Philippines, Elevation: 311m, Type: Stratovolcano, VEI: 4
Date: 2021-03-05, Location: Ecuador, Coordinates: (-2.005, -78.341), Name: Sangay, Country: Ecuador, Elevation: 5286m, Type: Stratovolcano, VEI: 2
Date: 2021-04-09, Location: W Indies, Coordinates: (13.330, -61.180), Name: Soufriere St. Vincent, Country: St. Vincent & the Grenadines, Elevation: 1220m, Type: Stratovolcano, VEI: 4
Date: 2021-05-19, Location: Italy, Coordinates: (38.789, 15.213), Name: Stromboli, Country: Italy, Elevation: 924m, Type: Stratovolcano, VEI: 2
Date: 2021-05-22, Location: Africa-C, Coordinates: (-1.520, 29.250), Name: Nyiragongo, Country: Congo, DRC, Elevation: 3470m, Type: Stratovolcano, VEI: 2
Date: 2021-09-19, Location: Canary Is, Coordinates: (28.570, -17.830), Name: La Palma, Country: Spain, Elevation: 2426m, Type: Stratovolcano, VEI: 3
Date: 2021-12-04, Location: Java, Coordinates: (-8.108, 112.922), Name: Semeru, Country: Indonesia, Elevation: 3657m, Type: Stratovolcano, VEI: 4
Date: 2022-06-22, Location: Mexico, Coordinates: (19.023, -98.622), Name: Popocatepetl, Country: Mexico, Elevation: 5393m, Type: Stratovolcano, VEI: 2
Date: 2022-10-09, Location: Italy, Coordinates: (38.789, 15.213), Name: Stromboli, Country: Italy, Elevation: 924m, Type: Stratovolcano, VEI: 2
Date: 2022-12-04, Location: Italy, Coordinates: (38.789, 15.213), Name: Stromboli, Country: Italy, Elevation: 924m, Type: Stratovolcano, VEI: 2
Date: 2022-12-04, Location: Java, Coordinates: (-8.108, 112.922), Name: Semeru, Country: Indonesia, Elevation: 3657m, Type: Stratovolcano, VEI: 4
Date: 2024-06-03, Location: Philippines-C, Coordinates: (10.410, 123.130), Name: Kanlaon, Country: Philippines, Elevation: 2422m, Type: Stratovolcano, VEI: 3
-----
Date: 2024-08-17, Location: Kamchatka, Coordinates: (56.653, 161.360), Name: Sheveluch, Country: Russia, Elevation: 3283m, Type: Stratovolcano, VEI: 4
-----
4. Volcano VEI Finder 테스트
-----
VEI 4 화산:
Date: 2020-01-12, Location: Luzon-Philippines, Coordinates: (14.011, 120.998), Name: Taal, Country: Philippines, Elevation: 311m, Type: Stratovolcano, VEI: 4
Date: 2021-04-09, Location: W Indies, Coordinates: (13.330, -61.180), Name: Soufriere St. Vincent, Country: St. Vincent & the Grenadines, Elevation: 1220m, Type: Stratovolcano, VEI: 4
Date: 2021-12-04, Location: Java, Coordinates: (-8.108, 112.922), Name: Semeru, Country: Indonesia, Elevation: 3657m, Type: Stratovolcano, VEI: 4
Date: 2022-12-04, Location: Java, Coordinates: (-8.108, 112.922), Name: Semeru, Country: Indonesia, Elevation: 3657m, Type: Stratovolcano, VEI: 4
Date: 2024-08-17, Location: Kamchatka, Coordinates: (56.653, 161.360), Name: Sheveluch, Country: Russia, Elevation: 3283m, Type: Stratovolcano, VEI: 4
-----
5. Earthquake Depth Finder 테스트
-----
깊이 10-30km 지진:
Date: 2020-01-28, Location: CUBA: GRAMMA; CAYMAN IS; JAMAICA, Coordinates: (19.440, -78.755), Depth: 10km, Magnitude: 7.7
Date: 2020-06-18, Location: KERMADEC ISLANDS: S OF, RAOUL, Coordinates: (-33.204, -177.838), Depth: 10km, Magnitude: 7.4
Date: 2020-06-23, Location: MEXICO: OAXACA, Coordinates: (16.029, -95.901), Depth: 26km, Magnitude: 7.4
Date: 2020-07-22, Location: ALASKA, Coordinates: (55.030, -158.522), Depth: 28km, Magnitude: 7.8
Date: 2020-10-30, Location: GREECE: SAMOS; TURKEY: IZMIR, Coordinates: (37.918, 26.790), Depth: 21km, Magnitude: 7.0
Date: 2021-02-10, Location: NEW CALEDONIA: LOYALTY ISLANDS, Coordinates: (-23.054, 171.601), Depth: 10km, Magnitude: 7.7
Date: 2021-03-04, Location: KERMADEC ISLANDS: SSE OF RAOUL ISLAND, Coordinates: (-29.740, -177.267), Depth: 19km, Magnitude: 8.1
Date: 2021-05-21, Location: CHINA: QINGHAI PROVINCE, Coordinates: (34.598, 98.251), Depth: 10km, Magnitude: 7.3
-----
6. Earthquake Magnitude Finder 테스트
-----
진도 8.0-9.0 지진:
Date: 2021-03-04, Location: KERMADEC ISLANDS: SSE OF RAOUL ISLAND, Coordinates: (-29.740, -177.267), Depth: 19km, Magnitude: 8.1
Date: 2021-07-29, Location: ALASKA PENINSULA, Coordinates: (55.325, -157.841), Depth: 32km, Magnitude: 8.2
-----
Sorter 테스트
-----
-----
1. Volcano Sorter 테스트
-----
연도 기준 정렬:
Date: 2020-01-12, Location: Luzon-Philippines, Coordinates: (14.011, 120.998), Name: Taal, Country: Philippines, Elevation: 311m, Type: Stratovolcano, VEI: 4
Date: 2021-03-05, Location: Ecuador, Coordinates: (-2.005, -78.341), Name: Sangay, Country: Ecuador, Elevation: 5286m, Type: Stratovolcano, VEI: 2
Date: 2021-03-00, Location: Guatemala, Coordinates: (14.382, -90.601), Name: Pacaya, Country: Guatemala, Elevation: 2569m, Type: Complex volcano, VEI: 1
Date: 2021-04-09, Location: W Indies, Coordinates: (13.330, -61.180), Name: Soufriere St. Vincent, Country: St. Vincent & the Grenadines, Elevation: 1220m, Type: Stratovolcano, VEI: 4
-----
Date: 2020-10-19, Location: ALASKA, Coordinates: (54.608, -159.655), Depth: 33km, Magnitude: 7.6
Date: 2020-01-28, Location: CUBA: GRAMMA; CAYMAN IS; JAMAICA, Coordinates: (19.440, -78.755), Depth: 10km, Magnitude: 7.7
Date: 2021-02-10, Location: NEW CALEDONIA: LOYALTY ISLANDS, Coordinates: (-23.054, 171.601), Depth: 10km, Magnitude: 7.7
Date: 2020-07-22, Location: ALASKA, Coordinates: (55.030, -158.522), Depth: 28km, Magnitude: 7.8
Date: 2021-03-04, Location: KERMADEC ISLANDS: SSE OF RAOUL ISLAND, Coordinates: (-29.740, -177.267), Depth: 19km, Magnitude: 8.1
Date: 2021-07-29, Location: ALASKA PENINSULA, Coordinates: (55.325, -157.841), Depth: 32km, Magnitude: 8.2
```