

# REPORT

자바프로그래밍 2 1분반

## Lab 2.

제출일: 2023.09.25

이름: 이 주 성

학번: 32183520

## 주어진 **csv** 파일 분석

```
#AtomicNumber,Element,Symbol,AtomicMass,Period,Group,Phase,Type
1,Hydrogen,H,1.007,1,1,gas,Nonmetal
2,Helium,He,4.002,1,18,gas,Noble Gas
```

- **Periodic Element**의 **8**가지의 값이 **" , "**를 기준으로 구분되어 있는 **csv** 파일

## **PeriodicElement** 클래스

- **8**가지 필드 (**number, name, symbol, mass, period, group, phase, type**)
- 모든 필드의 값을 파라미터로 가진 생성자
- **getter & setter**
- **toString**

## **PeriodicElementImporter**

- 생성자
- **parse()**
  - **" , "**로 구분된 **8**가지 값들을 가진 배열(**items**)를 받아 **PeriodicElement** 객체로 생성하는 메서드
- **loadCSV()**
  - **csv** 파일을 열어서 한줄씩 읽어온다.
  - **"#"**가 포함된 줄은 출력만 해주고 넘어간다.
  - **" , "**를 기준으로 **split**해서 문자열 배열에 담은 후 **pars()** 메소드의 파라미터로 넘긴다.
- **saveCSV()**
  - 지정된 경로에 **PeriodicElement** 리스트를 파일로 **write**한다.

## PeriodicElementFinder 클래스

- **Strategy Pattern**이 적용된 **IPeriodicElementFinder**를 필드로 가지고 있어 런타임에 필요한 **xxxFinder**를 **set**해서 사용
- **set**한 **xxxFinder**의 **find()** 메서드로 검색

## IPeriodicElementFinder 인터페이스

```
interface IPeriodicElementFinder {  
    8 implementations  
    void find(List<PeriodicElement> periodicElements);  
}
```

- 원소 기호를 검색하기 위해 만든 인터페이스
- 어떤 속성으로 검색할지 **Strategy Pattern**으로 런타임에 결정하기 위해 인터페이스로 만듦
- 이 인터페이스를 구현한 **xxxFinder**가 오버라이딩한 **find()** 메소드로 검색

## 8가지 Finder

- 8가지의 다른 기준으로 **IPeriodicElementFinder**를 구현한 **Finder** 클래스

## NumberFinder

- **1~118** 사이의 입력 값 제한
  - **getIntegerBetween()**
  - 범위 넘어가면 다시 입력 받음

- **for**문으로 모든 원소 돌면서 입력값으로 받은 숫자와 같은 **number**를 가진 **Periodic Element** 검색
- 검색 후 **display()** 출력 - 추가한 코드
  - **IPeriodicElementDisplay**

```
public interface IPeriodicElementDisplay {
    2 usages  1 implementation
    void display(PeriodicElement periodicElement);
}
```

- **PeriodicElementDisplay**

```
public class PeriodicElementDisplay implements IPeriodicElementDisplay {
    2 usages
    @Override
    public void display(PeriodicElement periodicElement) {
        if (periodicElement != null) {
            System.out.println(periodicElement);
            System.out.println("\n... End of Search ...\n");
        }
    }
}
```

## NameFinder

- **String input**
  - **getString()**
    - 아무값도 입력하지 않았을 경우 다시 입력 받음
- 대문자, 소문자 구분없이 검색하기 위해 전부 소문자로 변환 후 비교 - 추가한 코드

```
for (PeriodicElement pe : periodicElements) {
    // 원소이름도 추출해서 소문자로 변환
    if (pe.getName().toLowerCase().equals(name)) {
        IPeriodicElementDisplay ped = new PeriodicElementDisplay();
    }
}
```

## SymbolFinder

- **NameFinder**와 거의 동일

## MassFinder

- **double** 타입을 입력받는 부분을 제외하고 **NumberFinder**와 동일
- 너무 정확한 질량으로 검색하기엔 불편하니 오차 범위 **0.1**을 둬 - 추가한 코드

```
for (PeriodicElement pe : periodicElements) {  
    if (Math.abs(pe.getMass() - mass) < 0.1) { // 오차범위 0.1 적용  
        IPeriodicElementDisplay ped = new PeriodicElementDisplay();
```

## PeriodFinder

- **1~7** 사이의 값만 검색 가능
  - **getIntegerBetween()**
- 여러개의 값이 있을 수 있으므로 **List**에 담아 출력
  - **displayList()** - 추가한 코드

```
public interface IPeriodicElementDisplay {  
    4 usages 1 implementation  
    void display(PeriodicElement periodicElement);  
    1 usage 1 implementation  
    void displayList(List<PeriodicElement> periodicElementList);  
}
```

```
@Override  
public void displayList(List<PeriodicElement> periodicElementList) {  
    for (PeriodicElement pe : periodicElementList) {  
        System.out.print(pe.getName()+" ");  
    }  
    System.out.println();  
}
```

## GroupFinder

- **1~18** 입력 가능 **null**일 경우도 있음

- **getNullOrIntegerBetween**
- **null**일 경우 **0**을 반환받고 따로 처리

```
// null인 경우
if (input == 0) {
    for (PeriodicElement pe : periodicElements) {
        if (pe.getGroup() ≥ 1 && pe.getGroup() ≤ 18) {
            continue;
        } else {
            list.add(pe);
        }
    }
}
```

- 나머지는 **PeriodFinder**와 동일

## PhaseFinder

- 타입을 제외하고 **PeriodFinder**와 동일

## TypeFinder

- **null** 입력이 가능한 것을 제외하고 **PhaseFinder**와 동일
  - **getStringNullable**

## MainTest 실행

- 검색 방법을 번호로 선택 - 추가한 코드
  - **0**을 선택하면 검색 종료

검색 방법:

1. Number
2. Name
3. Symbol
4. Mass
5. Period
6. Group
7. Phase
8. Type
0. 종료

검색 방법을 번호로 선택해주세요 (0을 입력하면 종료):

검색 방법을 번호로 선택해주세요 (0을 입력하면 종료): 4

... Start searching by mass ...

Please enter [mass] of PeriodicElement: 10

not found

done..

검색 방법을 번호로 선택해주세요 (0을 입력하면 종료): 6

... Start searching by group ...

Please enter [group] of PeriodicElement: 2

Beryllium

Magnesium

Calcium

Strontium

Barium

Radium

검색 방법을 번호로 선택해주세요 (0을 입력하면 종료): 0

검색을 종료합니다.