# **REPORT**

자바프로그래밍2 1분반 Lab2

제출일 2023.09.21

**소속** 소프트웨어학과

**학번** 32201817

이름 박정운

### 1. IPeriodicElementFinder 인터페이스

IPeriodicElementFinder 인터페이스는 find()라는 abstract method를 가짐으로써 '검색' 이라는 행위를 유연하게 확장할 수 있도록 한다.

```
import java.util.List;
import template.PeriodicElement;

public interface IPeriodicElementFinder {
    //abstract method - find()
    void find(List<PeriodicElement> list);
}
```

#### 2. PeriodicElementFinder 클래스

PeriodicElementFinder 클래스는 객체를 주입 받음으로써 '검색'이라는 행위를 동적으로 변경할 수 있도록 한다. IPeriodicElementFinder를 주입 받아 find() 메소드를 실행시킨다.

```
import java.util.List;
import template.PeriodicElement;

public class PeriodicElementFinder {
    //field
    private IPeriodicElementFinder finder = null;

    //constructor
    public PeriodicElementFinder(IPeriodicElementFinder finder) {
        this.finder = finder;
    }

    //method1 - setFinder()
    public void setFinder(IPeriodicElementFinder finder){
        this.finder = finder;
    }

    //method2 - find()
    public void find(List<PeriodicElement> list) {
        finder.find(list);
    }
}
```

# 3. UserInput 클래스

UserInput 클래스는 User의 Input 값을 랜덤하게 반환하는 메소드들의 집합이다.

```
/* UserInput : User의 Input 값을 Random하게 반환하는 Method Group */
public class UserInput {

//[NumberFinder, GroupFinder, PeriodFinder 사용]
//min 과 max 사이에 있는 Random int를 반환
public static int getIntegerBetween(int min, int max){

if(min >= max) {

//{Exception} min은 max보다 클 수 없다.

throw new IllegalArgumentException(

s:"max value cannot be less than min value.");

}

// util.Random을 통해 min과 max 사이 값 반환
return new Random().nextInt((max-min)-1)+min;
}
```

[1] getIntegerBetween(int min, int max)는 min과 max 사이의 값을 랜덤하게 반환하는 메소드이다. min이 max보다 큰 경우 예외가 발생하고, 입력이 옳은 경우엔 Random 라이브러리를 통해 min과 max 사이에 있는 값을 반환한다.

내가 추가한 메소드[2] getPeriodicElementName(List<PeriodicElement> list)는 원소의 Name Input 값을 랜덤하게 반환하는 메소드이다. [1]에서 정의한 메소드를 이용해 1~118 사이 랜덤한 값을 받아오고, 이 index에 해당하는 element의 name을 반환한다. Name이 null인 경우엔 예외가 발생한다.

<mark>내가 추가한 메소드</mark> [3] getPeriodicElementSymbol(List<PeriodicElement> list)는 [2]와 동일하게 동작하여 랜덤한 Symbol 값을 반환한다.

# 4-1. GroupFinder 클래스

```
public class GroupFinder implements IPeriodicElementFinder {
   @Override
    public void find(List<PeriodicElement> list) {
       System.out.println(x:"*Group finder");
       System.out.println(x:"Please enter [group] of PeriodicElement: ");
       //UserInput 객체를 통해 1~18 사이 값을 random하게 저장
       int group = UserInput.getIntegerBetween(min:1, max:18);
       System.out.println("group: "+ group);
       //한 group에 해당되는 element가 다수일 수 있으므로, List 형태로 값을 검색
       List<PeriodicElement> groupElement = new ArrayList<>();
       for(PeriodicElement element : list){
           //List 객체의 element.group 값과 Input 값이 동일한지 확인
           if(element.getGroup()==group){
               groupElement.add(element);
       System.out.println("["+group+"] PeriodicElement Information: ");
       for(PeriodicElement element : groupElement){
           //결과 List에 있는 element의 Information을 순차적으로 출력
           System.out.println(element.toString());
       System.out.println();
```

GroupFinder 클래스는 IPeriodicElementFinder를 상속받아 group 정보를 기반으로 element를 검색하는 기능을 수행한다. UserInput의 getIntegerBetween()를 통해 input 값을 받아온다. 이 때 element는 1~18 group까지 있기 때문에 이 사이의 값을 저장한다. 또한, 한 그룹에 여러 개의 원소가 존재하기 때문에 for문과 List를 이용하여 검색을 수행하고 결과 값을 출력한다.

#### 4-2. NameFinder 클래스

```
public class NameFinder implements IPeriodicElementFinder{
   @Override
   public void find(List<PeriodicElement> list) {
       System.out.println(x:"*Name finder");
       System.out.println(x:"Please enter [name] of PeriodicElement: ")
       //UserInput 객체를 통해 원소의 name을 random하게 저장
       String name = UserInput.getPeriodicElementName(list);
       System.out.println("name: "+ name);
       PeriodicElement resultElement = null;
       //List 객체의 element.name 값과 Input 값이 동일한지 확인
       for(PeriodicElement element : list){
           if(element.getName().equals(name)){
               resultElement = element;
           }
       //{Exception} Input에 해당되는 element가 없는 경우
       if(resultElement == null) throw new IllegalArgumentException(
           s:"No element exists that matches the element name.");
       //검색한 element의 Information을 출력
       System.out.println("["+name+"] PeriodicElement Information: ");
       System.out.println(resultElement.toString()+"\n");
```

NameFinder 클래스도 4-1과 동일하게 동작한다. 하지만 name은 한 element가 가지고 있는 unique한 값이기 때문에 한 name에 한 element가 매칭된다는 특징이 있다. 따라서 Input값과 동일한 하나의 element를 검색하고 정보를 출력한다.

#### 4-3. NumberFinder 클래스

```
public class NumberFinder implements IPeriodicElementFinder{

@Override
public void find(List<PeriodicElement> list) {

System.out.println(x:"*Number finder");
System.out.println(x:"Please enter [number] of PeriodicElement: ");

//UserInput 객체를 통해 1~118 사이 값을 random 하게 저장
int number = UserInput.getIntegerBetween(min:1, max:118);
System.out.println("number: "+ number);

//index 값을 이용해 element 검색
PeriodicElement element = list.get(number-1);

//검색한 element의 information 출력
System.out.println("["+number+"] PeriodicElement Information: ");
System.out.println(element.toString()+"\n");
}
```

NumberFinder 클래스도 위 두 클래스와 동일하게 동작한다. 이 때 Number는 원소의 id와 같다. 따라서 List의 index 값과 매칭되기 때문에 index값을 통해 검색을 수행한다.

## 4-4. PeriodFinder 클래스

```
public class PeriodFinder implements IPeriodicElementFinder{

@Override
public void find(List<PeriodicElement> list) {
    System.out.println(x:"*Period finder");
    System.out.println(x:"Please enter [period] of PeriodicElement: ");

    //UserInput 객체를 통해 1~7 사이 값을 random하게 저장
    int period = UserInput.getIntegerBetween(min:1, max:7);
    System.out.println("period: "+ period);
```

```
//UserInput 객체를 통해 1~7 사이 값을 random하게 저장
int period = UserInput.getIntegerBetween(min:1, max:7);
System.out.println("period: "+ period);
//한 period에 해당되는 element가 다수일 수 있으므로, List 형태로 값을 검색
List<PeriodicElement> periodElement = new ArrayList<>();
for(PeriodicElement element : list){
   //period가 input 값을 넘어갈 경우 반복문을 벗어남
   if(element.getPeriod() == period +1) break;
   //period가 input 값과 동일한지 확인
   if(element.getPeriod()==period){
       periodElement.add(element);
System.out.println("["+period+"] PeriodicElement Information: ");
for(PeriodicElement element : periodElement){
   //결과 List에 있는 element의 Information을 순차적으로 출력
   System.out.println(element.toString());
System.out.println();
```

PeriodFinder 클래스는 GroupFinder와 거의 동일하게 동작한다. 한 가지 특징을 언급하자면, List에는 주기율표가 순차적으로 저장되어 있기 때문에 period 값도 순차적으로 증가한다. 따라서 검색 시 period가 input을 벗어나면 검색을 중단하 고 반복문을 벗어난다.

# 4-5. SymbolFinder 클래스

```
public class SymbolFinder implements IPeriodicElementFinder{
    @Override
    public void find(List<PeriodicElement> list) {
        System.out.println(x:"*Symbol finder");
        System.out.println(x:"Please enter [symbol] of PeriodicElement: ");

        //UserInput 객체를 통해 원소의 symbol을 random하게 저장
        String symbol = UserInput.getPeriodicElementSymbol(list);
        System.out.println("symbol: "+ symbol);
```

```
PeriodicElement resultElement = null;

//List 객체의 element.symbol 값과 Input 값이 동일한지 확인

for(PeriodicElement element : list){

    if(element.getSymbol().equals(symbol)){

        resultElement = element;
    }

}

//{Exception} Input에 해당되는 element가 없는 경우

if(resultElement == null) throw new IllegalArgumentException(
    s:"No element exists that matches the element symbol.");

//검색한 element의 Information을 출력

System.out.println("["+symbol+"] PeriodicElement Information: ");

System.out.println(resultElement.toString()+"\n");

}
```

SymbolFinder 클래스도 NameFinder와 동일하게 동작한다.

#### 5. MainTest

```
// Jeong woon Park (Lab2, Dankook University, Department of Software) 2023/09/21
public class MainTest {
   //IPeriodicElementFinder를 상속받은 하위 객체들의 모음
   private IPeriodicElementFinder[] finders = {
       new NumberFinder(), new NameFinder(), new GroupFinder(),
       new PeriodFinder(), new SymbolFinder()};
  public MainTest(){
       //csv 파일에서 데이터 load 후 출력
       List<PeriodicElement> list =
           PeriodicElementImporter.loadCSV(filename: "PeriodicElements.csv");
       System.out.println(x:"\nlist all");
       list.forEach(System.out::println);
       System.out.println(x:"\n");
       //finders의 find 메소드를 순차적으로 출력
       System.out.println(x:"----- find PeriodicElement -----\n");
       for(IPeriodicElementFinder finder: finders){
           PeriodicElementFinder f = new PeriodicElementFinder(finder);
           f.find(list);
```

교수님께서 주신 코드와 동일하다. csv를 읽어와 데이터를 list에 저장하고 find를 순차적으로 시행한다.

## **Result**

\* 단순 데이터 출력 결과는 첨부하지 않았다.

Please enter [symbol] of PeriodicElement:

PS C:\Users\박정운\OneDrive\바탕 화면\DKU 3-2\java programming2\HW2>

[Cl] PeriodicElement Information:

symbol: Cl

```
----- find PeriodicElement
 *Number finder
Please enter [number] of PeriodicElement:
 number: 37
[37] PeriodicElement Information:
PeriodicElement [number=37, name=Rubidium, symbol=Rb, mass=85.468, period=5, group=1, phase=solid, type=Alkali Metal]
 Please enter [name] of PeriodicElement:
 name: Sulfur
[Sulfur] PeriodicElement Information:
PeriodicElement [number=16, name=Sulfur, symbol=S, mass=32.065, period=3, group=16, phase=solid, type=Nonmetal]
Please enter [group] of PeriodicElement:
 group: 1
 [1] PeriodicElement Information:
[1] PeriodicElement Information:

PeriodicElement [number=1, name=Hydrogen, symbol=H, mass=1.007, period=1, group=1, phase=gas, type=Nonmetal]

PeriodicElement [number=3, name=Lithium, symbol=Li, mass=6.941, period=2, group=1, phase=solid, type=Alkali Metal]

PeriodicElement [number=11, name=Sodium, symbol=Na, mass=22.99, period=3, group=1, phase=solid, type=Alkali Metal]

PeriodicElement [number=19, name=Potassium, symbol=K, mass=39.098, period=4, group=1, phase=solid, type=Alkali Metal]

PeriodicElement [number=37, name=Rubidium, symbol=Rb, mass=85.468, period=5, group=1, phase=solid, type=Alkali Metal]

PeriodicElement [number=55, name=Cesium, symbol=Cs, mass=132.905, period=6, group=1, phase=solid, type=Alkali Metal]

PeriodicElement [number=87, name=Francium, symbol=Fr, mass=223.0, period=7, group=1, phase=solid, type=Alkaline Earth Metal]
 Please enter [period] of PeriodicElement:
   [3] PeriodicElement Information:
 PeriodicElement Information:

PeriodicElement [number=11, name=Sodium, symbol=Na, mass=22.99, period=3, group=1, phase=solid, type=Alkali Metal]

PeriodicElement [number=12, name=Magnesium, symbol=Mg, mass=24.305, period=3, group=2, phase=solid, type=Alkaline Earth Metal]

PeriodicElement [number=13, name=Aluminum, symbol=Al, mass=26.982, period=3, group=13, phase=solid, type=Metal]

PeriodicElement [number=14, name=Silicon, symbol=Si, mass=28.086, period=3, group=14, phase=solid, type=Metalloid]

PeriodicElement [number=15, name=Phosphorus, symbol=P, mass=30.974, period=3, group=15, phase=solid, type=Nonmetal]

PeriodicElement [number=16, name=Sulfur, symbol=S, mass=32.065, period=3, group=16, phase=solid, type=Nonmetal]

PeriodicElement [number=17, name=Chlorine, symbol=Cl, mass=35.453, period=3, group=17, phase=gas, type=Halogen]

PeriodicElement [number=18, name=Argon, symbol=Ar, mass=39.948, period=3, group=18, phase=gas, type=Noble Gas]
   *Symbol finder
```

PeriodicElement [number=17, name=Chlorine, symbol=Cl, mass=35.453, period=3, group=17, phase=gas, type=Halogen]