REPORT

자바 프로그래밍2 1분반 LAB8

제출일 2023.11.27
소속 소프트웨어학과
학번 32201817
이름 박정운

1. AbstractFinder<T>

```java
public abstract class AbstractFinder<T> {
    abstract String getUserInput(); //조건을 입력
    abstract Boolean predicate(T element); //조건 일치 여부 판단
    //Template method
    List<T> find(List<T> elements){
        List<T> find = new ArrayList<>();
        for(T e : elements){
            //객체가 조건에 맞으면
            if(predicate(e)){
                //List에 추가
                find.add(e);
            }
        }
        //List return
        return find;
    };
}
```

Template 메소드를 구현하는 AbstractFinder 클래스이다. 조건을 입력하는 메소드와 조건에 부합하는지를 판단하는 메소드는 abstract로 하위 클래스에서 정의하도록 하고, find 메소드를 이용하여 template를 구성한다. find를 통해 조건에 부합하는 원소들을 골라 List를 반환한다.

2. NameFinder (+ NumberFinder, SymbolFinder, WeightFinder, PhaseFinder, GroupFinder, PeriodFinder, TypeFinder)

```java
public class NameFinder extends AbstractFinder<PeriodicElement>{
    private Scanner sc = new Scanner(System.in);
    private String input = null;
    @Override
    String getUserInput() {
        //Name 입력
        System.out.print(s:"[Name Finder]: Enter the name : ");
        input = sc.nextLine();
        return input;
    }
    @Override
    Boolean predicate(PeriodicElement element) {
        if(input == null){
            input = getUserInput();
        }
        //Name 일치 여부 판단
        if(input.equals(element.getName())){
            return true;
        }
        return false;
    }
}
```

NameFinder를 비롯하여 PeriodicElement를 찾는 Finder concrete class는 다음과 같은 형태를 띈다. getUserInput을 통해 각 Finder에 맞는 조건을 사용자로부터 입력받고, Predicate를 통해 매개변수로 받은 원소의 값이 입력값과 동일한지 여부를 판단해 반환한다.

3. MainTest - 1

```
public class MainTest {
    public static void Main(){
        //csv로부터 PeriodicElement List 추출
        List<PeriodicElement> list = PeriodicElementImporter.loadCSV(filename:"PeriodicElements.csv");
        //list.forEach(System.out::println);
        // find PE by number, name, symbol, weight, period, group, phase, type
        System.out.println(x:"\n** PeriodicElement **");
        //Finders 정리
        List<AbstractFinder<PeriodicElement>> finders = Arrays.asList(
            new NumberFinder(), new NameFinder(), new SymbolFinder(), new WeightFinder(),
            new PeriodFinder(), new GroupFinder(), new PhaseFinder(), new TypeFinder());
        for (var finder : finders) {
            //input 받고 조건 일치하는 원소 List 출력
            String input = finder.getUserInput();
            System.out.println("You entered: " + input);
            List<PeriodicElement> found = finder.find(list);
            found.forEach(System.out::println);
        }
        System.out.println(x:"\n\n");
```

PeriodicElement의 Finder를 실행하는 코드이다. Csv 파일로부터 List<PeriodicElement>를 추출하고, Finder들을 리스트로 묶은 후 각 finder마다 입력값을 받고 조건에 맞는 List 를 출력해준다.

4. ChemicalCompoundJSONImporter

```
public class ChemicalCompoundJSONImporter {
    public static List<ChemicalCompound> importFile(String filePath) {
        System.out.println("file import: "+filePath);
        try (FileReader reader = new FileReader(filePath)) {
            Gson gson = new Gson();
            // TypeToken을 사용하여 List<ChemicalCompound>로
            List<ChemicalCompound> chemicalCompounds = gson.fromJson(
                reader, new TypeToken<List<ChemicalCompound>>(){}.getType());
            // 각 ChemicalCompound 객체에서 Map<PeriodicElement, Integer>로 변환하여 출력
            for (ChemicalCompound chemicalCompound : chemicalCompounds) {
                Map<PeriodicElement, Integer> compoundMap = chemicalCompound.getCompoundMap();
            }
            return chemicalCompounds;
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

JSON 파일로부터 ChemicalCompound 객체를 추출하는 코드이다. GSON을 이용하여 chemicalCompound List를 추출한 후, 각 ChemicalCompound 객체에서 Map<PeriodicElement, Integer>을 추출한다.

5. ChemicalCompound

```java
public class ChemicalCompound {
    private String name;
    private String symbol;
    private List<Compound> compounds; //구성원소 List
    private Phase phase;

    /* constructor */
    public ChemicalCompound(String name, String symbol,
         List<Compound> compounds, Phase phase) {
            this.name = name;
            this.symbol = symbol;
            this.compounds=compounds;
            this.phase = phase;
    }

    // calculate molecular weight from atomic weight * count
    public double getWeight() {
        double weight = 0;
        for(Compound e : compounds){
            //원소 무게 * 개수
            weight += e.getPeriodicElement().getMass()*e.getCount();
        }
        return weight;
    }
```

ChemicalCompound는 name, symbol, compounds(구성 원소들), Phase로 이루어진다. getWeight()은 구성 원소들의 개수와 무게를 고려하여 총 무게를 계산한다.

```java
//List -> Map<PeriodicElement, Integer> 로 변환
public Map<PeriodicElement, Integer> getCompoundMap() {
    Map<PeriodicElement, Integer> compoundMap = new HashMap<>();
    for (Compound compound : compounds) {
        compoundMap.put(compound.getPeriodicElement(), compound.getCount());
    }
    return compoundMap;
}
```

getCompoundMap은 List<Compound> 를 Map<PeriodicElement, Integer> 형태로 변환하고 Map을 반환한다.

6. ChemicalCompoundElementFinder(+ChemicalCompoundNameFinder, ChemicalCompoundSymbolFinder, ChemicalCompoundWeightFinder, ChemicalCompoundPhaseFinder)

```
public class ChemicalCompoundElementFinder extends AbstractFinder<ChemicalCompound>{
    private Scanner sc = new Scanner(System.in);
    private String input = null;
    @Override
    String getUserInput() {
        //Element name 입력
        System.out.print(s:"[Element Finder]: Enter the element : ");
        input = sc.nextLine();
        return input;
    }
    @Override
    Boolean predicate(ChemicalCompound element) {
        if(input == null){
            input = getUserInput();
        }
        //Map<PeriodicElement, int>에서 PeriodicElement만 추출
        Set<PeriodicElement> keySet = element.getCompoundMap().keySet();
        for(PeriodicElement key : keySet){
            //Set 중 Name이 일치하는 원소 있는지 확인
            if(input.equals(key.getName())){
                return true;
            }
        }
        return false;
    }
}
```

ChemicalCompoundElementFinder도 PeriodicFinder와 동일합니다. 입력 값으로 조건을 입력받고, Predicate 함수를 통해 해당 원소가 조건에 만족하는지를 판단하여 줍니다.

7. MainTest - 2

```
    System.out.println(x:"** ChemicalCompound **");
    //json으로부터 ChemicalCompound List 추출
    List<ChemicalCompound> list2 = ChemicalCompoundJSONImporter
        .importFile(filePath:"ChemicalCompounds.json");
    //list2.forEach(System.out::println);
    // find CC by name, symbol, weight, phase, element (using symbol)
    //finders 정리
    List<AbstractFinder<ChemicalCompound>> finders2 = Arrays.asList(
        new ChemicalCompoundNameFinder(), new ChemicalCompoundSymbolFinder(),
        new ChemicalCompoundWeightFinder(), new ChemicalCompoundPhaseFinder(),
        new ChemicalCompoundElementFinder());
    for (var finder2 : finders2) {
        //input 받고 조건 일치하는 원소 List 출력
        String input2 = finder2.getUserInput();
        System.out.println("You entered: " + input2);
        List<ChemicalCompound> found2 = finder2.find(list2);
        found2.forEach(System.out::println);
    }
}
```

ChemicalCompound를 JSONImporter를 통해 List 형태로 추출한 후, finder들을 List에 저장하고 input을 받고 조건이 일치하는 원소 List를 출력하는 것을 반복한다.

Result

```
file import: PeriodicElements.csv
line contains #: #AtomicNumber,Element,Symbol,AtomicMass,Period,Group,Phase,Type
load successfully

** PeriodicElement **
[Number Finder]: Enter the number : 5
You entered: 5
PeriodicElement [number=5, name=Boron, symbol=B, mass=10.811, period=2, group=13, phase=solid, type=Metalloid]
[Name Finder]: Enter the name : Nitrogen
You entered: Nitrogen
PeriodicElement [number=7, name=Nitrogen, symbol=N, mass=14.007, period=2, group=15, phase=gas, type=Nonmetal]
[Symbol Finder]: Enter the symbol : B
You entered: B
PeriodicElement [number=5, name=Boron, symbol=B, mass=10.811, period=2, group=13, phase=solid, type=Metalloid]
[Weight Finder]: Enter the weight(ex.50~100): 10~20
You entered: 10~20
PeriodicElement [number=5, name=Boron, symbol=B, mass=10.811, period=2, group=13, phase=solid, type=Metalloid]
PeriodicElement [number=6, name=Carbon, symbol=C, mass=12.011, period=2, group=14, phase=solid, type=Nonmetal]
PeriodicElement [number=7, name=Nitrogen, symbol=N, mass=14.007, period=2, group=15, phase=gas, type=Nonmetal]
PeriodicElement [number=8, name=Oxygen, symbol=O, mass=15.999, period=2, group=16, phase=gas, type=Nonmetal]
PeriodicElement [number=9, name=Fluorine, symbol=F, mass=18.998, period=2, group=17, phase=gas, type=Halogen]
[Period Finder]: Enter the period(1~7): 2
You entered: 2
PeriodicElement [number=3, name=Lithium, symbol=Li, mass=6.941, period=2, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=4, name=Beryllium, symbol=Be, mass=9.012, period=2, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=5, name=Boron, symbol=B, mass=10.811, period=2, group=13, phase=solid, type=Metalloid]
PeriodicElement [number=6, name=Carbon, symbol=C, mass=12.011, period=2, group=14, phase=solid, type=Nonmetal]
PeriodicElement [number=7, name=Nitrogen, symbol=N, mass=14.007, period=2, group=15, phase=gas, type=Nonmetal]
PeriodicElement [number=8, name=Oxygen, symbol=O, mass=15.999, period=2, group=16, phase=gas, type=Nonmetal]
PeriodicElement [number=9, name=Fluorine, symbol=F, mass=18.998, period=2, group=17, phase=gas, type=Halogen]
PeriodicElement [number=10, name=Neon, symbol=Ne, mass=20.18, period=2, group=18, phase=gas, type=Noble Gas]
[Group Finder]: Enter the group(1~18): 2
You entered: 2
PeriodicElement [number=4, name=Beryllium, symbol=Be, mass=9.012, period=2, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=12, name=Magnesium, symbol=Mg, mass=24.305, period=3, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=20, name=Calcium, symbol=Ca, mass=40.078, period=4, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=38, name=Strontium, symbol=Sr, mass=87.62, period=5, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=56, name=Barium, symbol=Ba, mass=137.327, period=6, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=88, name=Radium, symbol=Ra, mass=226.0, period=7, group=2, phase=solid, type=Actinide]
```

```
You entered: liq
PeriodicElement [number=35, name=Bromine, symbol=Br, mass=79.904, period=4, group=17, phase=liq, type=Halogen]
PeriodicElement [number=80, name=Mercury, symbol=Hg, mass=200.59, period=6, group=12, phase=liq, type=Transition Metal]
[Type Finder]: Enter the type: Halogen
You entered: Halogen
PeriodicElement [number=9, name=Fluorine, symbol=F, mass=18.998, period=2, group=17, phase=gas, type=Halogen]
PeriodicElement [number=17, name=Chlorine, symbol=Cl, mass=35.453, period=3, group=17, phase=gas, type=Halogen]
PeriodicElement [number=35, name=Bromine, symbol=Br, mass=79.904, period=4, group=17, phase=liq, type=Halogen]
PeriodicElement [number=53, name=Iodine, symbol=I, mass=126.904, period=5, group=17, phase=solid, type=Halogen]

** ChemicalCompound **
file import: ChemicalCompounds.json
[Name Finder]: Enter the name : Methane
You entered: Methane
ChemicalCompound{name='Methane', symbol='CH4', compounds=[Compound{count='4', name='Hydrogen'}, Compound{count='1', name='Carbon'}], phase=gas}
[Symbol Finder]: Enter the symbol : C2H5OH
You entered: C2H5OH
ChemicalCompound{name='Ethanol', symbol='C2H5OH', compounds=[Compound{count='6', name='Hydrogen'}, Compound{count='2', name='Carbon'}, Compound{count='1',
 name='Oxygen'}], phase=liq}
[Weight Finder]: Enter the weight(ex.50~100): 10~20
You entered: 10~20
[Phase Finder]: Enter the phase(gas,liq,solid,artificial): liq
You entered: liq
ChemicalCompound{name='Water', symbol='H2O', compounds=[Compound{count='1', name='Oxygen'}, Compound{count='2', name='Hydrogen'}], phase=liq}
ChemicalCompound{name='Ethanol', symbol='C2H5OH', compounds=[Compound{count='6', name='Hydrogen'}, Compound{count='2', name='Carbon'}, Compound{count='1',
 name='Oxygen'}], phase=liq}
ChemicalCompound{name='Sulfuric Acid', symbol='H2SO4', compounds=[Compound{count='2', name='Hydrogen'}, Compound{count='4', name='Oxygen'}, Compound{count
='1', name='Sulfur'}], phase=liq}
ChemicalCompound{name='Acetic Acid', symbol='CH3COOH', compounds=[Compound{count='4', name='Hydrogen'}, Compound{count='2', name='Carbon'}, Compound{count
='2', name='Oxygen'}], phase=liq}
ChemicalCompound{name='Carbon Tetrachloride', symbol='CCl4', compounds=[Compound{count='1', name='Carbon'}, Compound{count='4', name='Chlorine'}], phase=l
iq}
ChemicalCompound{name='Hydrogen Peroxide', symbol='H2O2', compounds=[Compound{count='2', name='Oxygen'}, Compound{count='2', name='Hydrogen'}], phase=liq}
ChemicalCompound{name='Carbon Disulfide', symbol='CS2', compounds=[Compound{count='2', name='Sulfur'}, Compound{count='1', name='Carbon'}], phase=liq}
ChemicalCompound{name='Methanol', symbol='CH3OH', compounds=[Compound{count='4', name='Hydrogen'}, Compound{count='1', name='Carbon'}, Compound{count='1',
 name='Oxygen'}], phase=liq}
ChemicalCompound{name='Benzene', symbol='C6H6', compounds=[Compound{count='6', name='Hydrogen'}, Compound{count='6', name='Carbon'}], phase=liq}
You entered: Oxygen
ChemicalCompound{name='Water', symbol='H2O', compounds=[Compound{count='1', name='Oxygen'}, Compound{count='2', name='Hydrogen'}], phase=liq}
ChemicalCompound{name='Carbon Dioxide', symbol='CO2', compounds=[Compound{count='2', name='Oxygen'}, Compound{count='1', name='Carbon'}], phase=gas}
ChemicalCompound{name='Glucose', symbol='C6H12O6', compounds=[Compound{count='12', name='Hydrogen'}, Compound{count='6', name='Carbon'}, Compound{count='6
', name='Oxygen'}], phase=solid}
ChemicalCompound{name='Ethanol', symbol='C2H5OH', compounds=[Compound{count='6', name='Hydrogen'}, Compound{count='2', name='Carbon'}, Compound{count='1',
 name='Oxygen'}], phase=liq}
ChemicalCompound{name='Sulfuric Acid', symbol='H2SO4', compounds=[Compound{count='2', name='Hydrogen'}, Compound{count='4', name='Oxygen'}, Compound{count
='1', name='Sulfur'}], phase=liq}
ChemicalCompound{name='Acetic Acid', symbol='CH3COOH', compounds=[Compound{count='4', name='Hydrogen'}, Compound{count='2', name='Carbon'}, Compound{count
='2', name='Oxygen'}], phase=liq}
ChemicalCompound{name='Ammonium Nitrate', symbol='NH4NO3', compounds=[Compound{count='2', name='Nitrogen'}, Compound{count='3', name='Oxygen'}, Compound{c
ount='4', name='Hydrogen'}], phase=solid}
ChemicalCompound{name='Hydrogen Peroxide', symbol='H2O2', compounds=[Compound{count='2', name='Oxygen'}, Compound{count='2', name='Hydrogen'}], phase=liq}
ChemicalCompound{name='Carbon Monoxide', symbol='CO', compounds=[Compound{count='1', name='Oxygen'}, Compound{count='1', name='Carbon'}], phase=gas}
ChemicalCompound{name='Sodium Hydroxide', symbol='NaOH', compounds=[Compound{count='1', name='Oxygen'}, Compound{count='1', name='Hydrogen'}, Compound{cou
nt='1', name='Sodium'}], phase=solid}
ChemicalCompound{name='Sodium Carbonate', symbol='Na2CO3', compounds=[Compound{count='3', name='Oxygen'}, Compound{count='1', name='Carbon'}, Compound{cou
nt='2', name='Sodium'}], phase=solid}
ChemicalCompound{name='Methanol', symbol='CH3OH', compounds=[Compound{count='4', name='Hydrogen'}, Compound{count='1', name='Carbon'}, Compound{count='1',
 name='Oxygen'}], phase=liq}
ChemicalCompound{name='Sulfur Dioxide', symbol='SO2', compounds=[Compound{count='1', name='Sulfur'}, Compound{count='2', name='Oxygen'}], phase=gas}
ChemicalCompound{name='Calcium Carbonate', symbol='CaCO3', compounds=[Compound{count='3', name='Oxygen'}, Compound{count='1', name='Carbon'}, Compound{cou
nt='1', name='Calcium'}], phase=solid}
jeongwoon@jeongwoonui-MacBookPro HW8 %
```