

REPORT

LAB8

자바프로그래밍2

제출일	2023. 11.27
소속	컴퓨터공학과
학번	32183520
이름	이 주성

목표

1. CSV, JSON 파일 원하는 객체로 역직렬화
2. 템플릿 메서드 패턴을 이용해 원소, 화합물을 검색하는 프로그램
3. strategy 패턴 이용했던 것을 템플릿 메서드 패턴으로 구현하자

CSV, JSON을 PeriodicElement, ChemicalCompound 객체 리스트로 역직렬화

FileImporter

- 파일을 가져오고 내보내는 메서드 보유

```
public interface FileImporter<E> {  
    // 파일 가져오기  
    List<E> importFile(String filepath);  
  
    // 파일 내보내기  
    void exportFile(String filepath, List<E> list);  
}
```

PeriodicElementCSVImporter

- FileImporter 구현
- CSV 파일을 읽어 PeriodicElement 객체 리스트로 역직렬화

```
// 지정한 파일을 불러와 파싱한 후 PeriodicElement 리스트로 반환하는 메소드  
public List<PeriodicElement> importFile(String filename) {  
    List<PeriodicElement> list = new ArrayList<PeriodicElement>();  
    String line = "";  
  
    // load data  
    try (BufferedReader br = new BufferedReader(new FileReader(filename))) {  
  
        String delimiter = ",";
```

```

        while ((line = br.readLine()) != null) {
            // #이 포함된 line은 출력만 해주고 continue
            if (line.contains("#")) {
                continue;
            }

            // use comma as separator
            String[] items = line.split(delimiter);

            // String[] items -> PeriodicElement
            PeriodicElement pe = parse(items);

            // 리스트로 추가
            list.add(pe);
        }

        // 예외처리
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    System.out.println();

    return list;
}

```

ChemicalCompoundJSONImporter

- FileImporter 구현
- JSON 파일을 읽어 ChemicalCompound 객체 리스트로 역직렬화
- ChemicalCompound는 그 안에 PeriodicElement와 count를 가진 CompoundElementInfo 리스트 보유하고 있으므로 그에 따른 역직렬화 로직도 필요

```

@Override
public List<ChemicalCompound> importFile(String filepath) {
    // 반환할 Element 객체 리스트
    List<ChemicalCompound> list = new ArrayList<>();
}

```

```

String line = "";

try (BufferedReader br = new BufferedReader(new FileReader(filepath))) {
    StringBuilder jsonString = new StringBuilder();

    while ((line = br.readLine()) != null) {
        jsonString.append(line);
    }

    Gson gson = new GsonBuilder()
        .registerTypeAdapter(ChemicalCompound.class, new
ChemicalCompoundDeserializer())
        .registerTypeAdapter(CompoundElementInfo.class, new
CompoundElementInfoDeserializer())
        .create();

    //JSON 문자열을 List<ChemicalCompound>으로 역직렬화.
    Type listType = new TypeToken<List<ChemicalCompound>>() {}.getType();
    List<ChemicalCompound> compounds = gson.fromJson(jsonString.toString(),
listType);

    list.addAll(compounds);

} catch (IOException e) {
    e.printStackTrace();
}

return list;
}

```

Deserializer

1. ChemicalCompoundDeserializer

- ChemicalCompound로 역직렬화 하는 로직

```

@Override
public ChemicalCompound deserialize(JsonElement json, Type typeOfT,
JsonDeserializationContext context) throws JsonParseException {
    JsonObject jsonObject = json.getAsJsonObject();

    String name = jsonObject.get("name").getAsString();
    String symbol = jsonObject.get("symbol").getAsString();
    Phase phase = Phase.names(jsonObject.get("phase").getAsString());
}

```

```

    JSONArray compoundsArray = jsonObject.getAsJSONArray("compounds");
    List<CompoundElementInfo> compounds = context.deserialize(compoundsArray, new
    TypeToken<List<CompoundElementInfo>>() {}.getType());

    // ChemicalCompound 객체를 생성하고 반환
    return new ChemicalCompound(name, symbol, compounds, phase);
}

```

2. CompoundElementInfoDeserializer

- ChemicalCompound의 CompoundElementInfo로 역직렬화 하는 로직

```

@Override
public ChemicalCompound deserialize(JsonElement json, Type typeOfT,
    JsonDeserializationContext context) throws JsonParseException {
    JsonObject jsonObject = json.getAsJsonObject();

    String name = jsonObject.get("name").getString();
    String symbol = jsonObject.get("symbol").getString();
    Phase phase = Phase.names(jsonObject.get("phase").getString());

    JSONArray compoundsArray = jsonObject.getAsJSONArray("compounds");
    List<CompoundElementInfo> compounds = context.deserialize(compoundsArray, new
    TypeToken<List<CompoundElementInfo>>() {}.getType());

    // ChemicalCompound 객체를 생성하고 반환
    return new ChemicalCompound(name, symbol, compounds, phase);
}

```

템플릿 메서드 패턴 적용

템플릿 메서드 패턴이란?

- 객체들이 동일한 알고리즘을 공유하지만 알고리즘의 특정 단계를 서로 다르게 구현해야 할 때 사용하는 패턴
- 알고리즘의 골격을 기본 클래스에서 정의한 후 특정 단계를 서브 클래스에 위임

- 알고리즘의 구조를 변경하지 않고도 특정 단계를 재정의 가능
- 객체간 필요한 부분만 유연하게 변경 가능

AbstractFinder<T> 추상 클래스

- 템플릿 메서드 `public List<T> find(List<T> elements);`
 - `final`로 불변 메서드로 만들기
 - `getUserInput()`으로 조건 입력받은 후 원소 검색 수행
 - 구체적인 검색 방법은 서브 클래스의 `predicate`에 의해 결정

```
public abstract class AbstractFinder<T> {
    // 조회한 원소 리스트 저장할 변수
    private final List<T> foundList = new ArrayList<>();

    // 템플릿 메서드 -> 알고리즘 뼈대로 final로 변하지 않도록 선언
    public final List<T> find(List<T> elements){
        // 사용자 입력을 받은 후 저장
        Object input = getUserInput();

        // 주어진 리스트를 돌면서 체크
        for (T e : elements) {
            // 사용자에게 입력받은 조건과 순환하고 있는 원소를 비교
            if (predicate(e, input)) {
                // 맞다면 리스트에 추가
                foundList.add(e);
            }
        }

        // 일치하는 리스트 반환
        return foundList;
    }

    // Finder 마다 입력받는 타입이 다르므로 제네릭으로 설정
    protected abstract <R> R getUserInput();
    protected abstract boolean predicate(T e, Object input);
}
```

XxxFinder 원소 검색 클래스

- AbstractFinder<T> 상속
- getUserInput(): 어떤 원소를 찾을 것인지(조건) 사용자 입력 받음
- predicate(): 조건에 따른 boolean 값

```
public class NumberFinder extends AbstractFinder<PeriodicElement> {

    // 사용자 입력(조건) 받기
    @Override
    protected Integer getUserInput() {
        int input;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("\n>>>> Start Find Element");
        System.out.print("조회하고 싶은 원소의 Number를 입력하세요(1 ~ 118): ");

        // 1~118 사이의 숫자 입력받을때까지
        while (true) {
            try {
                input = Integer.parseInt(br.readLine());

                if (input >= 1 && input <= 118) {
                    break;
                } else {
                    System.out.print("1과 118 사이의 값을 입력해주세요: ");
                }
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        }

        return input;
    }

    // 입력받은 조건과 일치하는지 확인
    @Override
    protected boolean predicate(PeriodicElement e, Object input) {
        return e.getNumber() == (int) input;
    }
}
```

나머지 Finder들도 동일한 형식

ChemicalCompoundXxxFinder 화합물 검색 클래스

- Name, Symbol, Weight, Phase, Element를 기준으로 화합물 검색
 - CO₂, H₂O 등 찾기
- AbstractFinder<T> 상속
- 마찬가지로 getUserInput(), predicate()로 조건 검색
 - getUserInput: 조건 입력받기
 - predicate: 입력받은 조건에 따른 boolean 값
- SymbolFinder를 이용해 ChemicalCompoundElementFinder 구현
 - O가 들어있는 모든 화합물 찾기

ChemicalCompoundNameFinder

```
public class ChemicalCompoundNameFinder extends AbstractFinder<ChemicalCompound>
{
    // 사용자 입력(조건) 받기
    @Override
    protected String getUserInput() {
        String input;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("\n>>>> Start Find CC Element");
        System.out.print("조회하고 싶은 화합물의 Name을 입력하세요(Water, Carbon
Dioxide ...): ");

        while (true) {
            try {
                input = br.readLine();

                if (input != null) {
                    break;
                } else {
                    System.out.print("다시 입력해주세요: ");
                }
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        }

        return input;
    }
}
```



```

// 입력받은 조건과 일치하는지 확인
@Override
protected boolean predicate(ChemicalCompound c, Object input) {
    return c.getName().equals((String) input);
}
}

```

ChemicalCompoundWeightFinder

```

public class ChemicalCompoundWeightFinder extends
AbstractFinder<ChemicalCompound> {

    // 사용자 입력(조건) 받기
    @Override
    protected double[] getUserInput() {
        double[] input;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("\n>>>> Start Find CC Element");
        System.out.print("조회하고 싶은 화합물의 Weight 범위를 입력하세요(예시: 3~5) :
");

        while (true) {
            try {
                input = Arrays.stream(br.readLine().split("~"))
                    .mapToDouble(Double::parseDouble).toArray();

                if (input.length == 2) {
                    break;
                } else {
                    System.out.println("다시 입력해주세요 (예시: 3~5): ");
                }
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        }

        Arrays.sort(input);

        return input;
    }
}

```

```

// 입력받은 조건과 일치하는지 확인
@Override
protected boolean predicate(ChemicalCompound e, Object input) {
    double[] d = (double[]) input;
    return (e.getWeight() >= d[0] && e.getWeight() <= d[1]);
}
}

```

ChemicalCompoundElementFinder

- 입력한 원소를 하나라도 가지고 있는 화합물을 검색한다.

```

@Override
protected String getUserInput() {
    String input;

    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("\n>>> Start Find CC Element");
    System.out.println("해당 원소를 1개라도 포함하고 있는 화합물을 조회합니다.");
    System.out.print("조회하고 싶은 원소를 입력하세요(H,O, ...): ");

    while (true) {
        try {
            input = br.readLine();

            if (input != null) {
                break;
            } else {
                System.out.print("다시 입력해주세요: ");
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    return input;
}

// 입력받은 조건과 일치하는지 확인
@Override
protected boolean predicate(ChemicalCompound c, Object input) {

```

```

    for (CompoundElementInfo ce : c.getCompounds()) {
        if (ce.getElement().getSymbol().equals((String) input)) {
            return true;
        }
    }
    return false;
}

```

결과

```

>>>> Start Find Element
조화하고 싶은 원소의 Number를 입력하세요(1 ~ 118): 1
PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]

>>>> Start Find Element
조화하고 싶은 원소의 Name을 입력하세요: Carbon
PeriodicElement [number=6, symbol=C, name=Carbon, weight=12.011, period=2, group=14, phase=solid, type=Nonmetal]

>>>> Start Find Element
조화하고 싶은 원소의 Symbol을 입력하세요: O
PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.999, period=2, group=16, phase=gas, type=Nonmetal]

>>>> Start Find Element
조화하고 싶은 원소의 Weight 범위를 입력하세요(예시: 3~5) : 10~20
PeriodicElement [number=5, symbol=B, name=Boron, weight=10.811, period=2, group=13, phase=solid, type=Metallloid]
PeriodicElement [number=6, symbol=C, name=Carbon, weight=12.011, period=2, group=14, phase=solid, type=Nonmetal]
PeriodicElement [number=7, symbol=N, name=Nitrogen, weight=14.007, period=2, group=15, phase=gas, type=Nonmetal]
PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.999, period=2, group=16, phase=gas, type=Nonmetal]
PeriodicElement [number=9, symbol=F, name=Fluorine, weight=18.998, period=2, group=17, phase=gas, type=Halogen]

```

```
>>>> Start Find Element
조회하고 싶은 원소의 Period를 입력하세요(1 ~ 7): 1
PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]
PeriodicElement [number=2, symbol=He, name=Helium, weight=4.002, period=1, group=18, phase=gas, type=Noble Gas]

>>>> Start Find Element
조회하고 싶은 원소의 Group을 입력하세요(1 ~ 18 혹은 null일 경우 -1을 입력): 3
PeriodicElement [number=21, symbol=Sc, name=Scandium, weight=44.956, period=4, group=3, phase=solid, type=Transition Metal]
PeriodicElement [number=39, symbol=Y, name=Yttrium, weight=88.906, period=5, group=3, phase=solid, type=Transition Metal]
PeriodicElement [number=57, symbol=La, name=Lanthanum, weight=138.905, period=6, group=3, phase=solid, type=Lanthanide]
PeriodicElement [number=89, symbol=Ac, name=Actinium, weight=227.0, period=7, group=3, phase=solid, type=Actinide]

>>>> Start Find Element
조회하고 싶은 원소의 Phase를 입력하세요(gas, liq, solid, artificial): liq
PeriodicElement [number=35, symbol=Br, name=Bromine, weight=79.904, period=4, group=17, phase=liq, type=Halogen]
PeriodicElement [number=80, symbol=Hg, name=Mercury, weight=200.59, period=6, group=12, phase=liq, type=Transition Metal]

>>>> Start Find Element
조회하고 싶은 원소의 Type을 입력하세요: Halogen
PeriodicElement [number=9, symbol=F, name=Fluorine, weight=18.998, period=2, group=17, phase=gas, type=Halogen]
PeriodicElement [number=17, symbol=Cl, name=Chlorine, weight=35.453, period=3, group=17, phase=gas, type=Halogen]
PeriodicElement [number=35, symbol=Br, name=Bromine, weight=79.904, period=4, group=17, phase=liq, type=Halogen]
PeriodicElement [number=53, symbol=I, name=Iodine, weight=126.904, period=5, group=17, phase=solid, type=Halogen]
```

```
>>>> Start Find CC Element
조회하고 싶은 화합물의 Name을 입력하세요(Water, Carbon Dioxide ...): Water
ChemicalCompound{name='Water', symbol='H2O', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.999, period=2, group=16, phase=gas, type=Nonmetal]}]}

>>>> Start Find CC Element
조회하고 싶은 화합물의 Symbol을 입력하세요 (H2O, CO2 ...): CO2
ChemicalCompound{name='Carbon Dioxide', symbol='CO2', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.999, period=2, group=16, phase=gas, type=Nonmetal]}]}

>>>> Start Find CC Element
조회하고 싶은 화합물의 Weight 범위를 입력하세요(예시: 3~5) : 10~20
ChemicalCompound{name='Water', symbol='H2O', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.999, period=2, group=16, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Methane', symbol='CH4', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Ammonia', symbol='NH3', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}

>>>> Start Find CC Element
조회하고 싶은 원소의 Phase를 입력하세요(gas, liq, solid, artificial): solid
ChemicalCompound{name='Glucose', symbol='C6H12O6', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Ammonium Nitrate', symbol='NH4NO3', compounds=[CompoundElementInfo{element=PeriodicElement [number=7, symbol=N, name=Nitrogen, weight=14.007, period=2, group=15, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Sodium Chloride', symbol='NaCl', compounds=[CompoundElementInfo{element=PeriodicElement [number=11, symbol=Na, name=Sodium, weight=22.99, period=3, group=1, phase=solid, type=Transition Metal]}]}
ChemicalCompound{name='Sodium Hydroxide', symbol='NaOH', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.99, period=2, group=16, phase=solid, type=Nonmetal]}]}
ChemicalCompound{name='Sodium Carbonate', symbol='Na2CO3', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.99, period=2, group=16, phase=solid, type=Nonmetal]}]}
ChemicalCompound{name='Calcium Carbonate', symbol='CaCO3', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.99, period=2, group=16, phase=solid, type=Nonmetal]}]}]
```

```
>>>> Start Find CC Element
해당 원소를 1개라도 포함하고 있는 화합물을 조회합니다.
조회하고 싶은 원소를 입력하세요(H, O, ...): H
ChemicalCompound{name='Water', symbol='H2O', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.999, period=2, group=16, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Glucose', symbol='C6H12O6', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Methane', symbol='CH4', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Ethanol', symbol='C2H5OH', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Sulfuric Acid', symbol='H2SO4', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Ammonia', symbol='NH3', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Acetic Acid', symbol='CH3COOH', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Hydrochloric Acid', symbol='HCl', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Ammonium Nitrate', symbol='NH4NO3', compounds=[CompoundElementInfo{element=PeriodicElement [number=7, symbol=N, name=Nitrogen, weight=14.007, period=2, group=15, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Hydrogen Peroxide', symbol='H2O2', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.99, period=2, group=16, phase=solid, type=Nonmetal]}]}
ChemicalCompound{name='Propane', symbol='C3H8', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Ethylene', symbol='C2H4', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Sodium Hydroxide', symbol='NaOH', compounds=[CompoundElementInfo{element=PeriodicElement [number=8, symbol=O, name=Oxygen, weight=15.99, period=2, group=16, phase=solid, type=Nonmetal]}]}
ChemicalCompound{name='Methanol', symbol='CH3OH', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}
ChemicalCompound{name='Benzene', symbol='C6H6', compounds=[CompoundElementInfo{element=PeriodicElement [number=1, symbol=H, name=Hydrogen, weight=1.007, period=1, group=1, phase=gas, type=Nonmetal]}]}]
```