Java Programming II
Lab8

514770-1
Fall 2023
11/14/2023

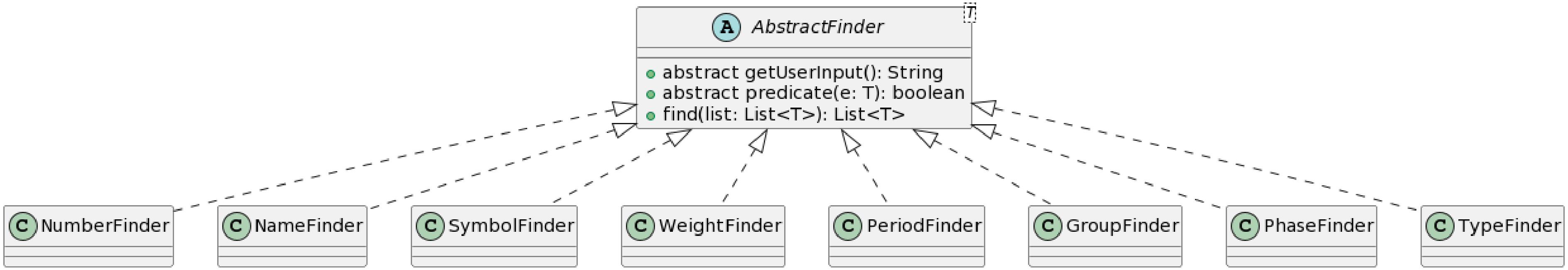
Kyoung Shin Park
Computer Engineering
Dankook University

Lab8

- Practice to write a program that **search** for **Element** by **various methods** using **Template Method pattern.**
  - **AbstractFinder<T>** generic class uses different predicate in **find()** method**.**
  - **NumberFinder, NameFinder, SymbolFinder, WeightFinder, PeriodFinder, GroupFinder, PhaseFinder, TypeFinder** class provides each **getUserInput** and **predicate.**
  - **ChemicalCompoundNameFinder, ChemicalCompoundSymbolFinder, ChemicalCompoundWeightFinder, ChemicalCompoundPhaseFinder, ChemicalCompoundElementFinder** class provides each **getUserInput** and **predicate.**
    - **ChemicalCompoundElementFinder** class uses SymbolFinder / NameFinder class.

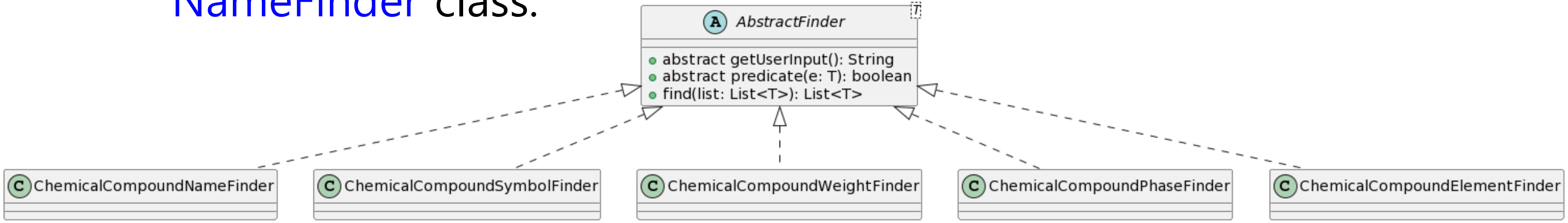
Lab8

- **AbstractFinder** class use the Template Method Pattern
  - public List<T> find(List<T> elements); // not changed
  - public abstract String getUserInput(); // changeable
  - public abstract Boolean predicate(T element); // changeable
- **NumberFinder, NameFinder, SymbolFinder, WeightFinder, PeriodFinder, GroupFinder, PhaseFinder, TypeFinder extends AbstractFinder<PeriodicElement>**



Lab8

- **AbstractFinder** class use the Template Method Pattern
- **ChemicalCompoundNameFinder, ChemicalCompoundSymbolFinder, ChemicalCompoundWeightFinder, ChemicalCompoundPhaseFinder, ChemicalCompoundElementFinder extends AbstractFinder<ChemicalCompound>**
  - ChemicalCompoundElementFinder class uses SymbolFinder / NameFinder class.


A AbstractFinder
T
abstract getUserInput(): String
abstract predicate(e: T): boolean
find(list: List<T>): List<T>
C ChemicalCompoundNameFinder
C ChemicalCompoundSymbolFinder
C ChemicalCompoundWeightFinder
C ChemicalCompoundPhaseFinder
C ChemicalCompoundElementFinder

Lab8

- ChemicalCompound class

```
public class ChemicalCompound {
    private String name;
    private String symbol;
    private Map<PeriodicElement, Integer> compounds;
    private Phase phase;
    public ChemicalCompound(String name, String symbol,
Map<PeriodicElement, Integer> compounds, Phase phase) {
        // 중간생략
    }
    // calculate molecular weight from atomic weight * count
    public double getWeight() {
    // 중간생략    }
}
```

Lab8

- **ChemicalCompounds.json**
  - **ChemicalCompoundJSONImporter** implements FileImporter<ChemicalCompound>
    - List<ChemicalCompound> importFile(String filepath)
    - void exportFile(String filepath, List<ChemicalCompound> list)
  - **ChemicalCompoundSerializer** implements JsonSerializer<ChemicalCompound>
  - **ChemicalCompoundDeserializer** implements JsonDeserializer<ChemicalCompound>
- **PeriodicElements.csv & PeriodicElements.json**
  - **PeriodicElementJSONImporter & PeriodicElementCSVImporter** implements FileImporter<PeriodicElement>
  - **PeriodicElementSerializer** implements JsonSerializer<PeriodicElement>
  - **PeriodicElementDeserializer** implements JsonDeserializer<PeriodicElement>

Lab8

- MainTest class

```
List<PeriodicElement> list =
PeriodicElementCSVImporter.importFile("PeriodicElements.csv");
list.forEach(System.out::println);
// find PE by number, name, symbol, weight, period, group,
phase, type
List<AbstractFinder<PeriodicElement>> finders = // 중간생략..
for (var finder : finders) {
    String input = finder.getUserInput();
    System.out.println("You entered: " + input);
    List<PeriodicElement> found = finder.find(list);
    found.forEach(System.out::println);
}
```

Lab8

- MainTest class

```
List<ChemicalCompound> list2 =
ChemicalCompoundJSONImporter.importFile("ChemicalCompounds.json");
list2.forEach(System.out::println);
// find CC by name, symbol, weight, phase, element (using symbol)
List<AbstractFinder<ChemicalCompound>> finders2 = // 중간생략..
for (var finder2 : finders2) {
    String input2 = finder2.getUserInput();
    System.out.println("You entered: " + input2);
    List<ChemicalCompound> found2 = finder2.find(list2);
    found2.forEach(System.out::println);
}
```

Submit to e-learning

- Add your code (e.g., additional method, class, routine, etc) in the Lab8 assignment.
- Submit the Lab8 assignment (JAVA23-2-Lab8-YourID-YourName.zip including the report) to e-learning.