

Java Programming II

Lab3

514770-1

Fall 2023

9/26/2023

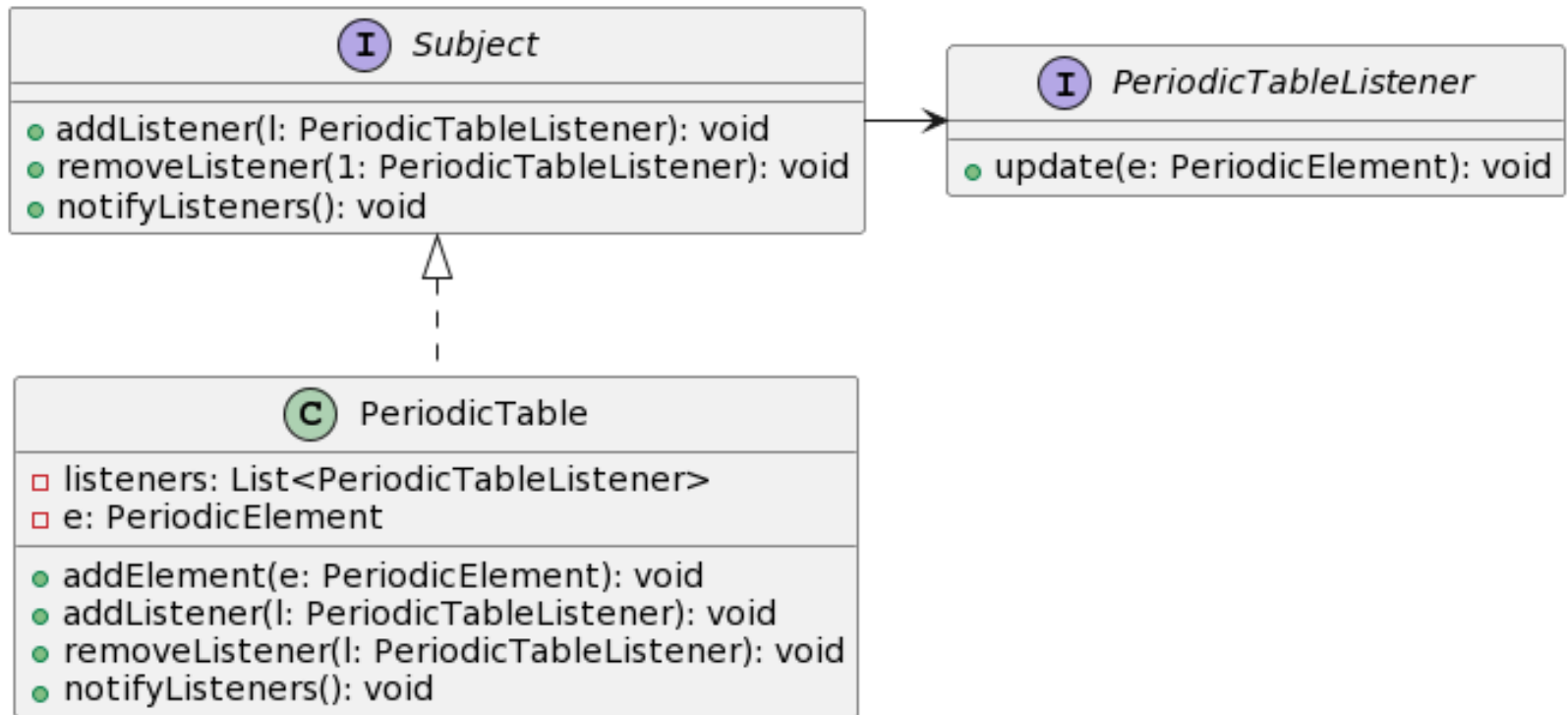
Kyoung Shin Park
Computer Engineering
Dankook University

Lab3

- Practice to write a program that shows **Periodic Element** using **Observer pattern**.
 - The class **PeriodicElement** contains **number, symbol, name, weight, period, group, phase, and type**.
 - Phase enum contains gas, liq, solid, artificial
 - You can get notifications when certain Periodic Element's number, name, symbol, period, group, phase are subscribed to.
 - The concrete Subject (PeriodicTable) updates (addElement) using random Thread.sleep.
 - The 6 concrete Observers (NumberListener, NameListener, SymbolListener, PeriodListener, GroupListener, PhaseListener) can be added or deleted at any time using random Thread.sleep.
 - When there is new added Periodic Element in PeriodicTable, it is notified to all subscribed listeners.

Lab3

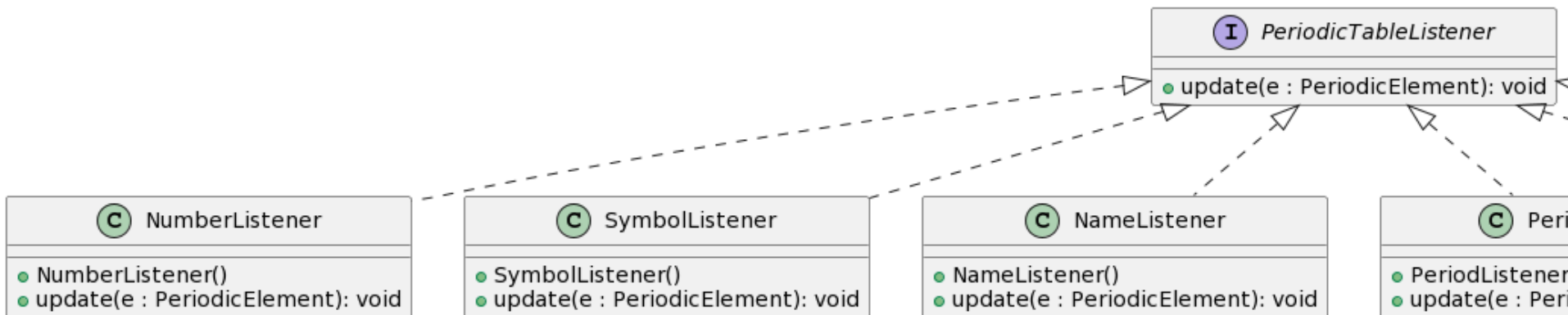
- **PeriodicTable** updates(add) PeriodicElement.
 - **notifyListeners()** updates PeriodicElement to all subscribed listeners.



Lab3

□ Observers

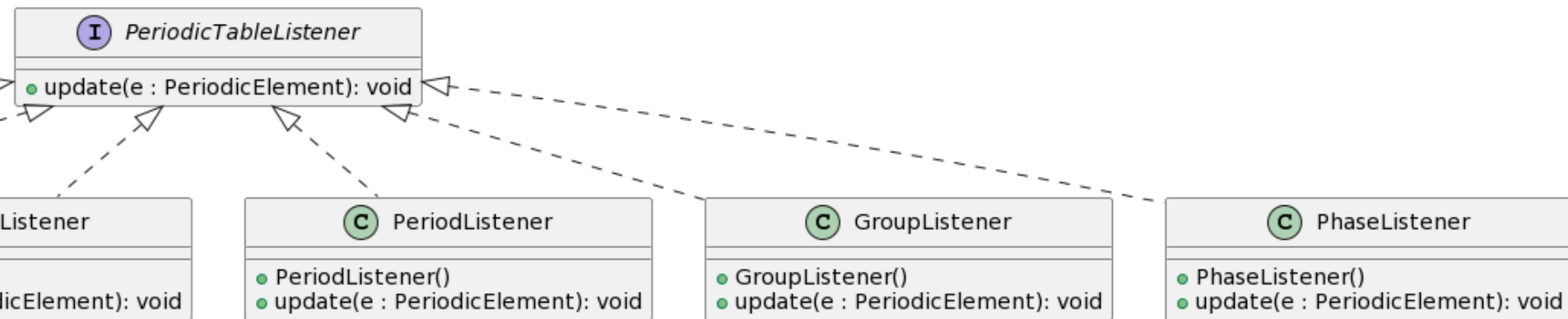
- PeriodicTableByNumberListener - displays PE if it's matched with UserInput list of numbers (e.g. 1,10,20,30,80).
- PeriodicTableBySymbolListener - displays PE if it's matched with UserInput list of symbols (e.g. H,O,C,S).
- PeriodicTableByNameListener - displays PE if it's matched with UserInput list of names (e.g. Gold,Silver).



Lab3

□ Observers

- PeriodicTableByPeriodListener - displays PE if it's matched with UserInput period (e.g. 1~7).
- PeriodicTableByGroupListener - displays PE if it's matched with UserInput group (e.g. 1~18).
- PeriodicTableByPhaseListener - displays PE if it's matched with UserInput phase (e.g. gas/liq/solid/artificial).



Lab3

```
public class GroupListener implements
PeriodicTableListener {
    int group = 0;
    public GroupListener(int group) {
        this.group = group;
    }
    @Override
    public void update(PeriodicElement pe){
        // if pe.getGroup() == this.group, then print pe
    }
}
```

Lab3

- ❑ MainTest class tests add new observers and delete observers at random time to see if the registered observers can display its data using the data updates.
 - First, initiate `PeriodicTable`
 - Then, `add observers` (`NumberListener`, `NameListener`, `SymbolListener`, `PeriodListener`, `GroupListener`, `PhaseListener`)
 - For each `PeriodicElement` in the list, `add PeriodicElement in PeriodicTable`, and `Sleep randomly` using `Thread.sleep(1000) // sleep 1 second`
 - Then, `delete some observers`.
 - Then, run for each `PeriodicElement` in the list, `add PeriodicElement in PeriodicTable`, and `Sleep randomly` using `Thread.sleep(1000) // sleep 1 second`
 - Then, `stop thread`.

Lab3

```
public class MainTest {
    public static void main(String[] args) {
        List<PeriodicElement> list = ... // 생략
        PeriodicTable t = new PeriodicTable();
        System.out.print("GroupListener: Please enter
[group#] of PeriodicTable to be updated: [1~18]");
        int group = UserInput.getIntegerBetween(1, 18);
        t.addListener(new GroupListener(group));
        for (PeriodicElement pe : list) {
            Thread.sleep(1000);
            t.addElement(pe);
        }
        // removeListener ...
    }
}
```


Submit to e-learning

- Add your code (e.g., additional method, class, routine, etc) in the Lab3 assignment.
- Submit the Lab3 assignment (**Java23-2-HW2-YourID-YourName.zip** including the report) to e-learning **due by 10/2**.