

Java Programming II

Lab6

514770-1

Fall 2023

10/31/2023

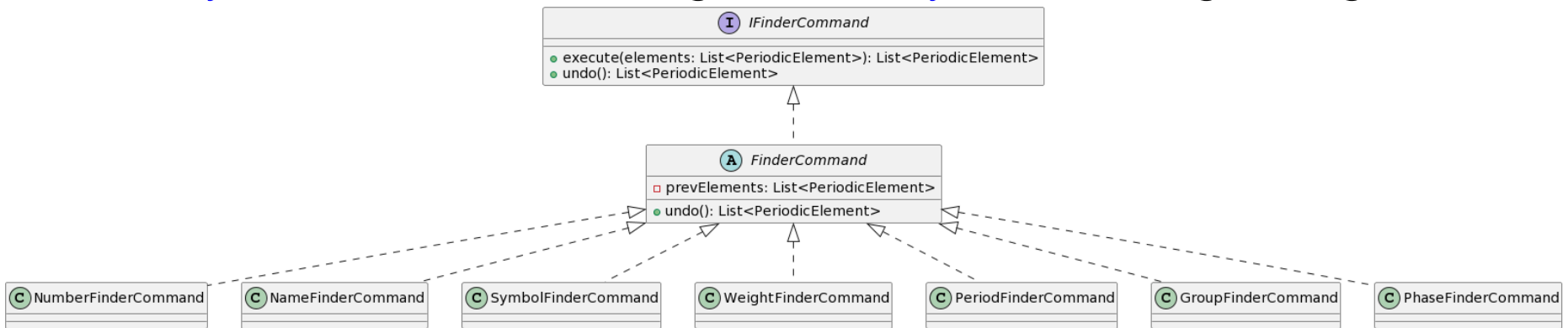
Kyoung Shin Park
Computer Engineering
Dankook University

Lab6

- Practice to write an **incremental find program** of Periodic Elements using **Command pattern**.
 - FinderCommandDatabase has a map between string name and actual FinderCommand class.
 - IFinderCommand interface has execute and undo.
 - FinderCommand abstract class implements undo.
 - NumberFinderCommand, NameFinderCommand, etc implements execute.
 - FinderInvoker is the invoker class.
 - Stack<IFinderCommand> stack (for undo operation).
 - void setCommand(IFinderCommand command)
 - List<PeriodicElement> execute(List<PeriodicElement> elements)
 - List<PeriodicElement> undo()

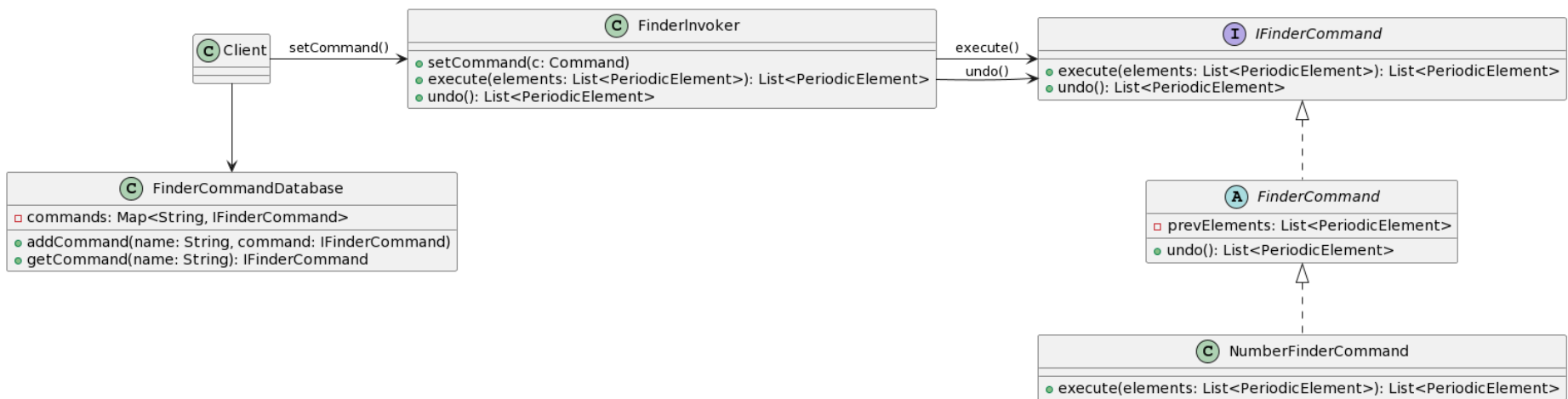
Lab6

- ❑ NumberFinderCommand, NameFinderCommand, SymbolFinderCommand, WeightFinderCommand, PeriodFinderCommand, GroupFinderCommand, PhaseFinderCommand executes or undo the find operation.
 - WeightFinderCommand gets double range user input, e.g. 50.0~100.0
 - NumberFinderCommand, NameFinderCommand, SymbolFinderCommand get the array of int, String, String



Lab6

- ❑ The MainTest client use **FinderInvoker** to set command and then, execute the command.
- ❑ You can also call the command undo.



Lab6

```
public class PhaseFinderCommand extends FinderCommand {
    private Phase phase;

    @Override
    public List<PeriodicElement> execute(List<PeriodicElement>
elements){
        prevElements = elements;
        if (phase == null) {
            System.out.print("Please enter [phase] of PeriodicElement
[e.g. gas, liq, solid, artificial]: ");
            phase = Phase.valueOf(UserInput.getString());
        }
        // your code to find PEs by phase

        return foundElements;
    }
}
```

Lab6

```
public class MainTest {
    public static List<PeriodicElement> deepcopy(List<PeriodicElement> list) {
        List<PeriodicElement> copy = new ArrayList<>();
        for (PeriodicElement e : list) {
            copy.add(e);
        }
        return copy;
    }
    public static void main(String[] args) {
        List<PeriodicElement> list =
            PeriodicElementImporter.loadCSV("PeriodicElements.csv");

        // incremental finder
        FinderCommandDatabase database = new FinderCommandDatabase();
        database.addCommand("number", new NumberFinderCommand());
        ... // 중간 생략
        // create foundList
        List<PeriodicElement> foundList = deepcopy(list);
    }
}
```

Lab6

```
// invoker class
FinderInvoker invoker = new FinderInvoker();
do {
    System.out.print("Please enter command [e.g.
number|name|symbol|weight|period|group|phase] or [undo]: ");
    String commandname = UserInput.getString();
    // command가 입력되지 않으면, 다시 입력
    // undo가 입력되면 foundList = invoker.undo()
    // undo시 empty stack이라면 foundList = null, 그러면 다시 deepcopy(list)
    // number|name|... 등 Finder 명령이 입력되면
    invoker.setCommand(database.getCommand(commandname));
    foundList = invoker.execute(foundList);
    foundList.forEach(System.out::println);
} while(!UserInput.getExitKey());
```

Submit to e-learning

- Add your code (e.g., additional method, class, routine, etc) in the Lab6 assignment.
- Submit the Lab6 assignment (JAVA23-2-Lab6-ID-name.zip including the report) to e-learning.