

REPORT

자바 프로그래밍2 1분반 LAB6

제출일	2023.11.12
소속	소프트웨어학과
학번	32201817
이름	박정운

1. IFinderCommand

```
public interface IFinderCommand {  
    //실행  
    List<PeriodicElement> execute(List<PeriodicElement> list);  
    //실행 취소  
    List<PeriodicElement> undo();  
}
```

Command 클래스의 기본형이다. execute(실행)과 undo(실행취소)를 나타내는 메소드로 구성되어 있다.

2. FinderCommand

```
public abstract class FinderCommand implements IFinderCommand{  
    //이전 상태의 Elements  
    protected List<PeriodicElement> prevElements;  
  
    @Override  
    public List<PeriodicElement> undo(){  
        //이전 상태를 return  
        return prevElements;  
    }  
}
```

FinderCommand를 상속받은 추상 클래스인 FinderCommand이다. 이전 상태를 저장하는 prevElements List를 가지고 있으며, 이 리스트를 리턴하는 undo()가 구현되어 있다.

3. FinderCommandDatabase

```
public class FinderCommandDatabase {  
    //command class들을 이름별로 저장  
    private Map<String, IFinderCommand> commands = new HashMap<>();  
  
    //command class 추가  
    public void addCommand(String name, IFinderCommand command){  
        commands.put(name, command);  
    }  
  
    //command return  
    public IFinderCommand getCommand(String name){  
        return commands.get(name);  
    }  
}
```

Command를 Map 형태로 입력받아 저장한 후 string 값을 기준으로 객체를 리턴해주는 데이터베이스 객체이다. Command를 입력받는 addCommand와 리턴하는 getCommand로 구성된다.

4. FinderInvoker

```
import template.PeriodicElement;

public class FinderInvoker {
    //IFinderCommand
    private IFinderCommand command;
    //undo를 위한 stack
    private Stack<IFinderCommand> stack = new Stack<>();

    //Command 지정 -> stack에 추가
    public void setCommand(IFinderCommand command){
        this.command = command;
        stack.push(command);
    }

    //command 실행
    public List<PeriodicElement> execute (List<PeriodicElement> list){
        return command.execute(list);
    }

    //이전 상태로 되돌리기 -> stack에서 command pop 후 command.undo();
    public List<PeriodicElement> undo(){
        if(stack.empty()) return null;
        stack.pop();
        return command.undo();
    }
}
```

Command에서 추상화된 execute와 undo를 실행시켜주는 Invoker 클래스이다. 실행 과정들을 저장해줄 Stack이 있고, 실행시킬 Command를 지정하는 IFinderCommand가 있다. setCommand를 통해 객체를 입력받고 스택에 저장한 후, execute를 통해 실행시킨다. 추가적으로 client가 되돌리기를 원할 경우 스택이 비어있는지 아닌지를 판단한 후 비어있지 않다면 stack에서 객체를 pop 한 후 command의 undo를 실행시킨다.

5. UserInput

```
public class UserInput {
    //Scanner
    Scanner scanner = new Scanner(System.in);
    //String 받아오기
    public String getString(){
        String input = scanner.nextLine();
        return input;
    }
}
```

입력 값을 받아오는 UserInput 클래스이다. Scanner 객체를 지정한 후 nextLine()을 통해 String 값을 받고 리턴하는 getString 메소드가 있다.

```

//Int 받아오기
public int getInt(){
    int input = scanner.nextInt();
    return input;
}

//Exit 여부 판단하기
public boolean getExitKey(){
    System.out.println(x:"exit : 0 / continue : 1");
    int bool = scanner.nextInt();
    //버퍼에 남은 값 제거
    scanner.nextLine();
    if(bool == 1) return true;
    else return false;
}
}

```

추가적으로, int값을 받아와 리턴하는 getInt와 Exit 여부를 판단하는 getExitKey()가 있다. 0이 입력되면 반복을 멈추겠다는 의미이고, 1이 입력되면 반복을 계속하겠다는 의미로 해석한다.

6. Concrete Command - Name, Number, Period, Group, Phase, Weight, Symbol

```

public class GroupFinderCommand extends FinderCommand{
    private String group;
    private UserInput userInput = new UserInput();
    //command 실행 (캡슐화 구현)
    @Override
    public List<PeriodicElement> execute(List<PeriodicElement> list) {
        prevElements = list; //이전 상태 저장
        List<PeriodicElement> foundElements = new ArrayList<>();

        if (group == null) {
            //group 값 입력
            System.out.print(s:"Please enter [group] of PeriodicElement[1-18]:");
            group = userInput.getString();
        }
        //String을 ',' 로 잘라 저장
        String [] splitArray = group.split(regex:",");
        for(PeriodicElement e : list){
            for(String group : splitArray){
                //splitArray를 돌려 값이 같은 원소 추출
                if(e.getGroup() == Integer.parseInt(group)){
                    foundElements.add(e);
                }
            }
        }
        //추출한 원소들 리턴
        return foundElements;
    }
}

```

FinderCommand를 상속받은 Concrete 객체는 다음과 같이 구성된다. 모든 구현체가 위 GroupFinderCommand와 유사한 형태를 띈다. 구현체에는 execute가 오버라이드 되어 구현된다. 일단 입력받은 인자를 통해 prevElements를 초기화한다. 그리고 찾을 멤버 값을 입력받는다. 여러 개의 값을 동시에 입력받을 수 있도록 String으로 입력받고 ','를 기준으로 분리해 배열에 저장한다. 그리고 인자로 받은 리스트를 돌려 group 값이 같은 원소를 찾아 foundElements에 추가한다. 그리고 추출한 원소들을 리턴한다.

7. MainTest

```
public static void Run() {
    //csv에서 값 load
    List<PeriodicElement> list= PeriodicElementImporter.loadCSV(filename:"PeriodicElements.csv");
    UserInput userInput = new UserInput();

    // incremental finder database에 추가
    FinderCommandDatabase database = new FinderCommandDatabase();
    database.addCommand(name:"number", new NumberFinderCommand());
    database.addCommand(name:"group", new GroupFinderCommand());
    database.addCommand(name:"phase", new PhaseFinderCommand());
    database.addCommand(name:"name", new NameFinderCommand());
    database.addCommand(name:"symbol", new SymbolFinderCommand());
    database.addCommand(name:"period", new PeriodFinderCommand());
    database.addCommand(name:"weight", new WeightFinderCommand());

    //foundlist 초기화
    List<PeriodicElement> foundList= deepcopy(list);
    String commandName = null; IFinderCommand command = null;

    // invoker class
    FinderInvoker invoker = new FinderInvoker();
```

MainTest에서는 일단, csv파일을 통해 원소값을 로드해온다. 그리고 데이터베이스 객체를 생성한 후 그 데이터베이스 객체에 {이름, Command 객체}를 저장한다. 차례로 foundList, invoker, commandName, IFinderCommand를 초기화해준다.

```
do {
    //어떤 command를 이용할 것인지 입력 받기
    System.out.print("Please enter command [e.g. number|name|symbol|weight|period|group|phase]"
        +"or [undo]: ");
    commandName = userInput.getString();
    command = database.getCommand(commandName); //database에서 해당 command 객체 리턴
    // undo가 입력되면 foundList= invoker.undo()
    if(commandName.equals(anObject:"undo")) {
        foundList = invoker.undo();
        // undo시 empty stack이라면 foundList= null, 그러면 다시 deepcopy(list)
        if (foundList == null) foundList = deepcopy(list);
        //undo한 결과를 출력
        foundList.forEach(System.out::println);
    }else if(command != null){
        //command를 invoker에 세팅 후 실행 (출력)
        invoker.setCommand(command);
        foundList=invoker.execute(foundList);
        foundList.forEach(System.out::println);
    }
    //반복 여부 확인
} while(userInput.getExitKey());
```

그 후 do-while 문을 돌게 된다. 어떤 명령을 이용할 것인지를 입력받고 데이터베이스에서 해당 command 객체를 불러온다. Command 객체가 아닌 undo 가 입력될 경우 해당 스택에 이전 상태가 존재하는 지 확인하고 존재한다면 이전 명령의 값들을 출력한다. 명령이 입력될 경우 invoker에 command를 추가하고 실행한 후 값들을 출력해준다. 한 사이클이 끝나면 다시 반복할 것인지 여부를 확인하고 반복을 원할 경우 계속한다.

8. Result

```
/Users/jeongwoon/Documents/univ/3-2/javaProgramming/JAVA23-2-Lab6-32201817-박 정 운 /HW6/bin App
file import: PeriodicElements.csv
line contains #: #AtomicNumber,Element,Symbol,AtomicMass,Period,Group,Phase,Type
load successfully
Please enter command [e.g. number|name|symbol|weight|period|group|phase]or [undo]: group
Please enter [group] of PeriodicElement[1-18]:1,2,3,4
PeriodicElement [number=1, name=Hydrogen, symbol=H, mass=1.007, period=1, group=1, phase=gas, type=Nonmetal]
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=11, name=Sodium, symbol=Na, mass=22.99, period=3, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=12, name=Magnesium, symbol=Mg, mass=24.305, period=3, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=19, name=Potassium, symbol=K, mass=39.098, period=4, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=20, name=Calcium, symbol=Ca, mass=40.078, period=4, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=21, name=Scandium, symbol=Sc, mass=44.956, period=4, group=3, phase=solid, type=Transition Metal]
PeriodicElement [number=22, name=Titanium, symbol=Ti, mass=47.867, period=4, group=4, phase=solid, type=Transition Metal]
PeriodicElement [number=37, name=Rubidium, symbol=Rb, mass=85.468, period=5, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=38, name=Strontium, symbol=Sr, mass=87.62, period=5, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=39, name=Yttrium, symbol=Y, mass=88.906, period=5, group=3, phase=solid, type=Transition Metal]
PeriodicElement [number=40, name=Zirconium, symbol=Zr, mass=91.224, period=5, group=4, phase=solid, type=Transition Metal]
PeriodicElement [number=55, name=Cesium, symbol=Cs, mass=132.905, period=6, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=56, name=Barium, symbol=Ba, mass=137.327, period=6, group=2, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=57, name=Lanthanum, symbol=La, mass=138.905, period=6, group=3, phase=solid, type=Lanthanide]
PeriodicElement [number=72, name=Hafnium, symbol=Hf, mass=178.49, period=6, group=4, phase=solid, type=Transition Metal]
PeriodicElement [number=87, name=Francium, symbol=Fr, mass=223.0, period=7, group=1, phase=solid, type=Alkaline Earth Metal]
PeriodicElement [number=88, name=Radium, symbol=Ra, mass=226.0, period=7, group=2, phase=solid, type=Actinide]
PeriodicElement [number=89, name=Actinium, symbol=Ac, mass=227.0, period=7, group=3, phase=solid, type=Actinide]
PeriodicElement [number=104, name=Rutherfordium, symbol=Rf, mass=261.0, period=7, group=4, phase=artificial, type=Transactinide]
exit : 0 / continue : 1
1
Please enter command [e.g. number|name|symbol|weight|period|group|phase]or [undo]: name
Please enter [name] of PeriodicElement:Barium,Calcium,Radium,Sodium
PeriodicElement [number=11, name=Sodium, symbol=Na, mass=22.99, period=3, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=20, name=Calcium, symbol=Ca, mass=40.078, period=4, 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]
exit : 0 / continue : 1
1
Please enter command [e.g. number|name|symbol|weight|period|group|phase]or [undo]: number
Please enter [number] of PeriodicElement[1-118]:11,20
PeriodicElement [number=11, name=Sodium, symbol=Na, mass=22.99, period=3, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=20, name=Calcium, symbol=Ca, mass=40.078, period=4, group=2, phase=solid, type=Alkaline Earth Metal]
exit : 0 / continue : 1
1
Please enter command [e.g. number|name|symbol|weight|period|group|phase]or [undo]: undo
PeriodicElement [number=11, name=Sodium, symbol=Na, mass=22.99, period=3, group=1, phase=solid, type=Alkali Metal]
PeriodicElement [number=20, name=Calcium, symbol=Ca, mass=40.078, period=4, 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]
exit : 0 / continue : 1
```