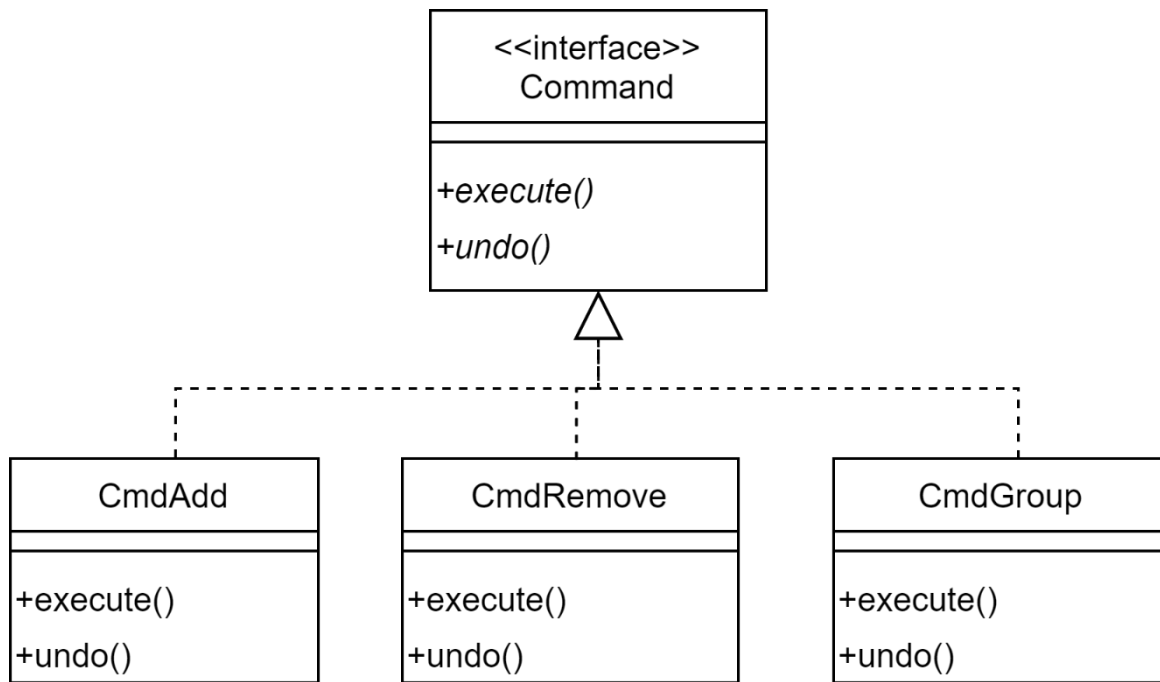


1) 개요



2) Figure.java

```
1 package command.e2;
2
3 public abstract class Figure implements Cloneable {
4
5     public abstract void draw(int indent);
6
7     @Override
8     public Figure clone() throws CloneNotSupportedException {
9         return (Figure)super.clone();
10    }
11 }
```

3) Rectangle.java

```
1 package command.e2;
2
3 public class Rectangle extends Figure {
4     String label;
5
6     public Rectangle(String label) {
7         this.label = label;
8     }
9
10    @Override
11    public void draw(int indent) {
12        String padding = " ".repeat(indent);
13        System.out.printf("%sRectangle(%s)\n", padding, label);
14    }
15 }
```

4) Line.java

```
1 package command.e2;
2
3 public class Line extends Figure {
4     String label;
5
6     public Line(String label) {
7         this.label = label;
8     }
9
10    @Override
11    public void draw(int indent) {
12        String padding = " ".repeat(indent);
13        System.out.printf("%sLine(%s)\n", padding, label);
14    }
15 }
```

5) FigureGroup.java

```
1 package composite.g2;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class FigureGroup extends Figure {
7     private List<Figure> figures = new ArrayList<Figure>();
8
9     @Override
10    public Figure clone() throws CloneNotSupportedException {
11        FigureGroup group = new FigureGroup();
12        for (Figure figure : figures)
13            group.add(figure.clone());
14        return group;
15    }
16
17    @Override
18    public void draw(int indent) {
19        String padding = " ".repeat(indent);
20        System.out.printf("%sGroup\n", padding);
21        for (Figure figure : figures)
22            figure.draw(indent + 1);
23        System.out.printf("%s)\n", padding);
24    }
25
26    public void add(Figure f) {
27        figures.add(f);
28    }
29
30    public int getCount() { return figures.size(); }
31    public Figure get(int index) { return figures.get(index); }
32 }
```

6) Command.java

```
1 package command.e2;
2
3 interface Command {
4     void execute();
5     void undo();
6 }
```

7) CmdAdd.java

```
1 package command.e2;
2
3 import java.util.List;
4
5 class CmdAdd implements Command {
6     List<Figure> figures; // 전체 도형 목록
7     Figure figure; // add 할 도형
8     int index; // add 한 위치
9
10    CmdAdd(List<Figure> figures, Figure figure) {
11        this.figures = figures;
12        this.figure = figure;
13    }
14
15    @Override
16    public void execute() {
17        figures.add(figure); // 도형을 목록에 add 한다
18        index = figures.size() - 1; // add 한 위치 기록
19    }
20
21    @Override
22    public void undo() {
23        figures.remove(index);
24    }
25 };
```

8) CmdRemove.java

```
1 package command.e2;
2
3 import java.util.List;
4
5 class CmdRemove implements Command {
6     List<Figure> figures; // 전체 도형 목록
7     int index;           // remove 할 위치
8     Figure figure;       // remove 된 도형
9
10    CmdRemove(List<Figure> figures, int index) {
11        this.index = index;
12        this.figures = figures;
13    }
14
15    @Override
16    public void execute() {
17        figure = figures.remove(index);
18    }
19
20    @Override
21    public void undo() {
22        figures.add(index, figure);
23    }
24 };
```

9) CmdDuplicate.java

```
1 package command.e2;
2
3 import java.util.List;
4
5 class CmdDuplicate implements Command {
6     List<Figure> figures; // 전체 도형 목록
7     int index1, index2; // 복제할 도형의 index, 복제된 도형의 index
8
9     CmdDuplicate(List<Figure> figures, int index) {
10        this.figures = figures;
11        this.index1 = index;
12    }
13
14    @Override
15    public void execute() {
16        try {
17            Figure figure = figures.get(index1);
18            figures.add(figure.clone()); // 복제
19            index2 = figures.size() - 1; // 복제된 도형의 위치 기록
20        } catch (CloneNotSupportedException e) {
21            e.printStackTrace();
22        }
23    }
24
25    @Override
26    public void undo() {
27        figures.remove(index2);
28    }
29 }
```

10) CmdGroup.java

```
1 package command.e2;
2
3 import java.util.Collections;
4 import java.util.List;
5
6 public class CmdGroup implements Command {
7     List<Figure> figures; // 전체 도형 목록
8     List<Integer> indexes; // 그룹으로 묶을 도형 목록
9     FigureGroup group;    // 생성된 그룹
10    int index;             // 생성된 그룹의 위치
11
12    public CmdGroup(List<Figure> figures, List<Integer> indexes) {
13        this.figures = figures;
14        this.indexes = indexes;
15    }
16
17    @Override
18    public void execute() {
19        Collections.sort(indexes); // undo 구현을 위해 정렬함
20        group = new FigureGroup();
21        for (int index : indexes)
22            group.add(figures.get(index));
23        figures.add(group);
24        index = figures.size() - 1; // 생성된 그룹위 위치 기록
25        for (int i = 0; i < group.getCount(); ++i)
26            figures.remove(group.get(i));
27    }
28
29    @Override
30    public void undo() {
31        for (int i = 0; i < indexes.size(); ++i)
32            figures.add(indexes.get(i), group.get(i));
33        figures.remove(index);
34    }
35 }
```

11) Example2.java

```
1 package command.e2;
2
3 import java.util.ArrayList;
4 import java.util.LinkedList;
5 import java.util.List;
6 import java.util.Scanner;
7
8 public class Example2 {
9
10     static List<Figure> figures = new ArrayList<Figure>();
11     static LinkedList<Command> undoStack = new LinkedList<Command>();
12
13     static void drawFigures() {
14         System.out.println();
15         for (int i = 0; i < figures.size(); ++i) {
16             System.out.printf("%d: ", i);
17             figures.get(i).draw(0);
18         }
19         System.out.println();
20     }
21
22     static void execute(String cmd) {
23         try {
24             int index;
25             Command command = null;
26             String[] a = cmd.split("[, ]+");
27             switch (a[0].toLowerCase()) {
28                 case "rectangle":
29                     command = new CmdAdd(figures, new Rectangle(a[1]));
30                     break;
31                 case "line":
32                     command = new CmdAdd(figures, new Line(a[1]));
33                     break;
34                 case "remove":
35                     index = Integer.valueOf(a[1]);
36                     command = new CmdRemove(figures, index);
37                     break;
38                 case "move":
39                     index = Integer.valueOf(a[1]);
40                     command = new CmdMove(figures.get(index), 2, 3);
41                     break;
42                 case "duplicate":
43                     index = Integer.valueOf(a[1]);
44                     command = new CmdDuplicate(figures, index);
45                     break;
46                 case "group":
47                     List<Integer> indexes = new ArrayList<>();
48                     for (int i = 1; i < a.length; ++i)
49                         indexes.add(Integer.valueOf(a[i]));
50                     command = new CmdGroup(figures, indexes);
51                     break;
52                 case "undo":
53                     command = undoStack.pop();
54                     command.undo();
55                     command = null;
56                     break;
57                 case "quit":
58                     System.exit(0);
59                     break;
60             }
61             if (command != null) {
62                 command.execute();
63                 undoStack.push(command); // undo 하기 위해 기록함
64             }
65         } catch (Exception e) {
66         }
67     }
68 }
```

```

69     static void prompt() {
70         System.out.printf(" 사각형 : rectangle 레이블\n");
71         System.out.printf(" 선      : line 레이블\n");
72         System.out.printf(" 삭제   : remove 번호\n");
73         System.out.printf(" 이동   : move 번호\n");
74         System.out.printf(" 복제   : duplicate 번호\n");
75         System.out.printf(" 그룹   : group 번호1, 번호2,...\n");
76         System.out.printf(" Undo   : undo\n");
77         System.out.printf(" 종료   : quit\n");
78         System.out.printf(" ? ");
79     }
80
81     public static void main(String[] args) {
82         try (Scanner scanner = new Scanner(System.in)) {
83             while (true) {
84                 prompt();
85                 String cmd = scanner.nextLine();
86                 execute(cmd);
87                 drawFigures();
88             }
89         }
90     }
91 }

```

실행 사례

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo
 종료 : quit
 ? rectangle A

0: Rectangle(A)

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo
 종료 : quit
 ? line B

0: Rectangle(A)

1: Line(B)

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo
 종료 : quit
 ? duplicate 0

0: Rectangle(A)

1: Line(B)

2: Rectangle(A)

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo

종료 : quit
? group 0, 1

```
0: Rectangle(A)
1: Group(
  Rectangle(A)
  Line(B)
)
```

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? duplicate 1

```
0: Rectangle(A)
1: Group(
  Rectangle(A)
  Line(B)
)
2: Group(
  Rectangle(A)
  Line(B)
)
```

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? group 0, 1

```
0: Group(
  Rectangle(A)
  Line(B)
)
1: Group(
  Rectangle(A)
  Group(
    Rectangle(A)
    Line(B)
  )
)
```

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? duplicate 1

```
0: Group(
  Rectangle(A)
  Line(B)
)
1: Group(
  Rectangle(A)
  Group(
    Rectangle(A)
    Line(B)
  )
)
```

```

2: Group(
  Rectangle(A)
  Group(
    Rectangle(A)
    Line(B)
  )
)

```

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo
 종료 : quit
 ? undo

```

0: Group(
  Rectangle(A)
  Line(B)
)
1: Group(
  Rectangle(A)
  Group(
    Rectangle(A)
    Line(B)
  )
)

```

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo
 종료 : quit
 ? undo

```

0: Rectangle(A)
1: Group(
  Rectangle(A)
  Line(B)
)
2: Group(
  Rectangle(A)
  Line(B)
)

```

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo
 종료 : quit
 ? undo

```

0: Rectangle(A)
1: Group(
  Rectangle(A)
  Line(B)
)

```

사각형 : rectangle 레이블
 선 : line 레이블
 삭제 : remove 번호
 복제 : duplicate 번호
 그룹 : group 번호1, 번호2,...
 Undo : undo

종료 : quit
? undo

0: Rectangle(A)
1: Line(B)
2: Rectangle(A)

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? undo

0: Rectangle(A)
1: Line(B)

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? undo

0: Rectangle(A)

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? undo

사각형 : rectangle 레이블
선 : line 레이블
삭제 : remove 번호
복제 : duplicate 번호
그룹 : group 번호1, 번호2,...
Undo : undo
종료 : quit
? quit