1) GroupSumResult.java

```
1  package visitor.e1a;
2
3  public class GroupSumResult {
4      public int lineSum = 0;
5      public int rectangleSum = 0;
6      public int groupSum = 0;
7
8      @Override
9      public String toString() {
10         return String.format("line=%d rectangle=%d group=%dWn",
11                 lineSum, rectangleSum, groupSum);
12     }
13 }
```

2) Figure.java

```
1  package visitor.e1a;
2
3  public abstract class Figure implements Cloneable {
4      int value;
5
6      public Figure(int value) {
7          this.value = value;
8      }
9
10     public abstract void draw(int indent);
11
12     @Override
13     public Figure clone() throws CloneNotSupportedException {
14         return (Figure)super.clone();
15     }
16
17     public int max() {
18         return value;
19     }
20
21     public int sum() {
22         return value;
23     }
24
25     public abstract void count(CountResult result);
26     public abstract void groupSum(GroupSumResult result);
27 }
```

3) Rectangle.java

```java
package visitor.e1a;

public class Rectangle extends Figure {

    public Rectangle(int value) {
        super(value);
    }

    @Override
    public void draw(int indent) {
        String padding = "  ".repeat(indent);
        System.out.printf("%sRectangle(%d)₩n", padding, value);
    }

    @Override
    public void count(CountResult result) {
        result.rectangleCount++;
    }

    @Override
    public void groupSum(GroupSumResult count) {
        count.rectangleSum += this.value;
    }
}
```

4) Line.java

```java
package visitor.e1a;

public class Line extends Figure {
    String label;

    public Line(int value) {
        super(value);
    }

    @Override
    public void draw(int indent) {
        String padding = "  ".repeat(indent);
        System.out.printf("%sLine(%d)₩n", padding, value);
    }

    @Override
    public void count(CountResult count) {
        count.lineCount++;
    }

    @Override
    public void groupSum(GroupSumResult count) {
        count.lineSum += this.value;
    }

}
```

5) FigureGroup.java

```java
package visitor.e1a;

import java.util.ArrayList;
import java.util.List;

public class FigureGroup extends Figure {

    public FigureGroup(int value) {
        super(value);
    }

    private List<Figure> figures = new ArrayList<Figure>();

    @Override
    public Figure clone() throws CloneNotSupportedException {
        FigureGroup group = new FigureGroup(value);
        for (Figure figure : figures)
            group.add(figure.clone());
        return group;
    }

    @Override
    public void draw(int indent) {
        String padding = "  ".repeat(indent);
        System.out.printf("%sGroup(Wn", padding);
        for (Figure figure : figures)
            figure.draw(indent + 1);
        System.out.printf("%s)Wn", padding);
    }

    public void add(Figure f) {
        figures.add(f);
    }

    public int getCount() {
        return figures.size();
    }

    public Figure get(int index) {
        return figures.get(index);
    }

    public void remove(int index) {
        figures.remove(index);
    }

    public void remove(Figure figure) {
        figures.remove(figure);
    }

    @Override
    public int max() {
        int result = Integer.MIN_VALUE;
        for (Figure figure : figures)
            result = Math.max(result, figure.max());
        return result;
    }

    @Override
    public int sum() {
        int result = 0;
        for (Figure figure : figures)
            result += figure.sum();
        return result;
    }

    @Override
    public void count(CountResult result) {
```

```
69          result.groupCount++;
70          for (Figure figure : figures)
71              figure.count(result);
72      }
73
74      @Override
75      public void groupSum(GroupSumResult result) {
76          result.groupSum += this.value;
77          for (Figure figure : figures)
78              figure.groupSum(result);
79      }
80  }
```

6) Example1a.java

```
1  package visitor.e1a;
2
3  import java.util.Scanner;
4
5  public class Example1a {
6
7      static FigureGroup root = new FigureGroup(0);
8
9      static void drawFigures() {
10         System.out.println();
11         for (int i = 0; i < root.getCount(); ++i) {
12             System.out.printf("%d: ", i);
13             root.get(i).draw(0);
14         }
15         System.out.println();
16     }
17
18     static void execute(String cmd) {
19         try {
20             CountResult countResult;
21             GroupSumResult groupSumResult;
22             String[] a = cmd.split("[, ]+");
23             switch (a[0].toLowerCase()) {
24             case "rectangle": root.add(new Rectangle(Integer.valueOf(a[1]))); break;
25             case "line": root.add(new Line(Integer.valueOf(a[1]))); break;
26             case "remove":
27                 int index = Integer.valueOf(a[1]);
28                 root.remove(index);
29                 break;
30             case "duplicate":
31                 Figure figure = root.get(Integer.valueOf(a[1]));
32                 root.add(figure.clone());
33                 break;
34             case "group":
35                 FigureGroup group = new FigureGroup(0);
36                 for (int i = 1; i < a.length; ++i)
37                     group.add(root.get(Integer.valueOf(a[i])));
38                 root.add(group);
39                 for (int i = 0; i < group.getCount(); ++i)
40                     root.remove(group.get(i));
41                 break;
42             case "max":
43                 System.out.printf("max = %dWn", root.max());
44                 break;
45             case "sum":
46                 System.out.printf("sum = %dWn", root.sum());
47                 break;
48             case "count":
49                 countResult = new CountResult();
50                 root.count(countResult);
51                 System.out.println(countResult);
52                 break;
53             case "groupsum":
54                 groupSumResult = new GroupSumResult();
55                 root.groupSum(groupSumResult);
56                 System.out.println(groupSumResult);
57                 break;
58             case "quit": System.exit(0); break;
59             default:
60                 System.out.println(a[0]);
61             }
62         } catch (Exception e) {
63             e.printStackTrace();
64         }
65     }
66
67     static void prompt() {
68         System.out.printf("  사각형   : rectangle 정수Wn");
```

```
69          System.out.printf("  선      : line 정수Wn");
70          System.out.printf("  삭제    : remove 번호Wn");
71          System.out.printf("  복제    : duplicate 번호Wn");
72          System.out.printf("  그룹    : group 번호1, 번호2,...Wn");
73          System.out.printf("  최대값  : maxWn");
74          System.out.printf("  합계    : sumWn");
75          System.out.printf("  도형 수 : countWn");
76          System.out.printf("  그룹합계: groupsumWn");
77          System.out.printf("  종료    : quitWn");
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  }
```

도형에 대한 작업을 추가할 때 마다, 도형 클래스들을 수정해야 한다.