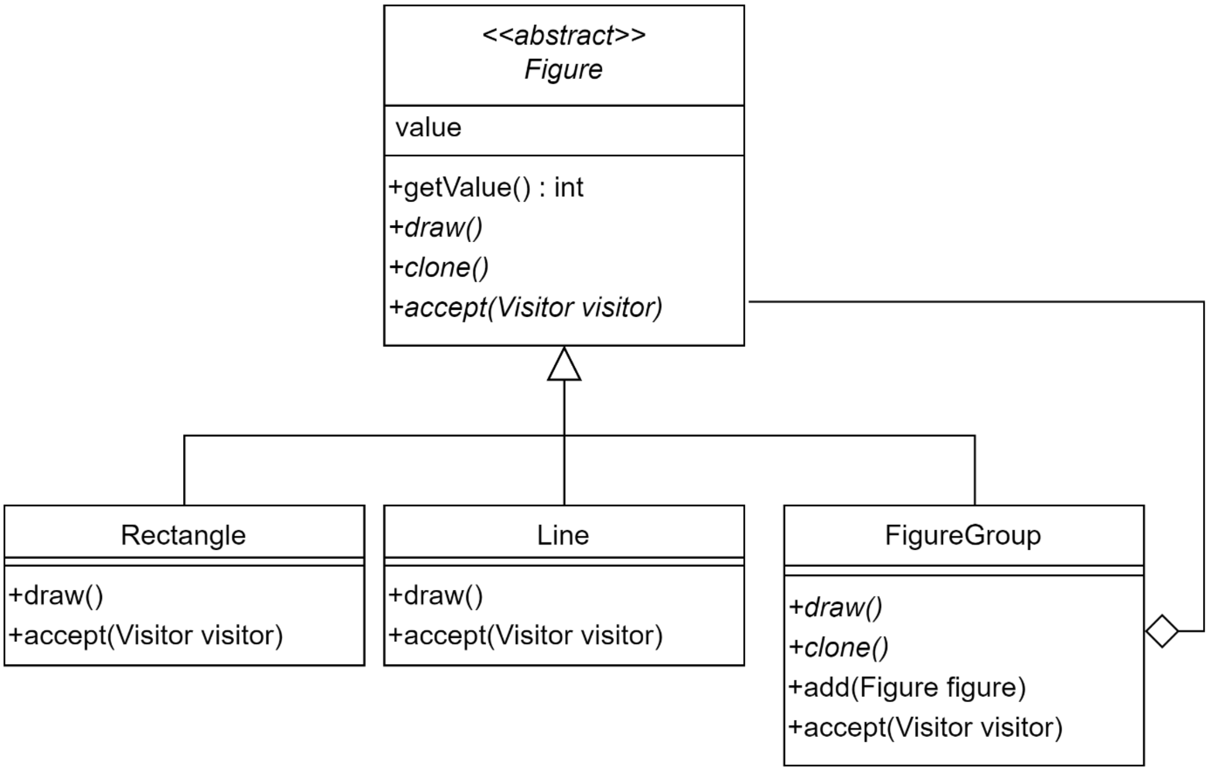
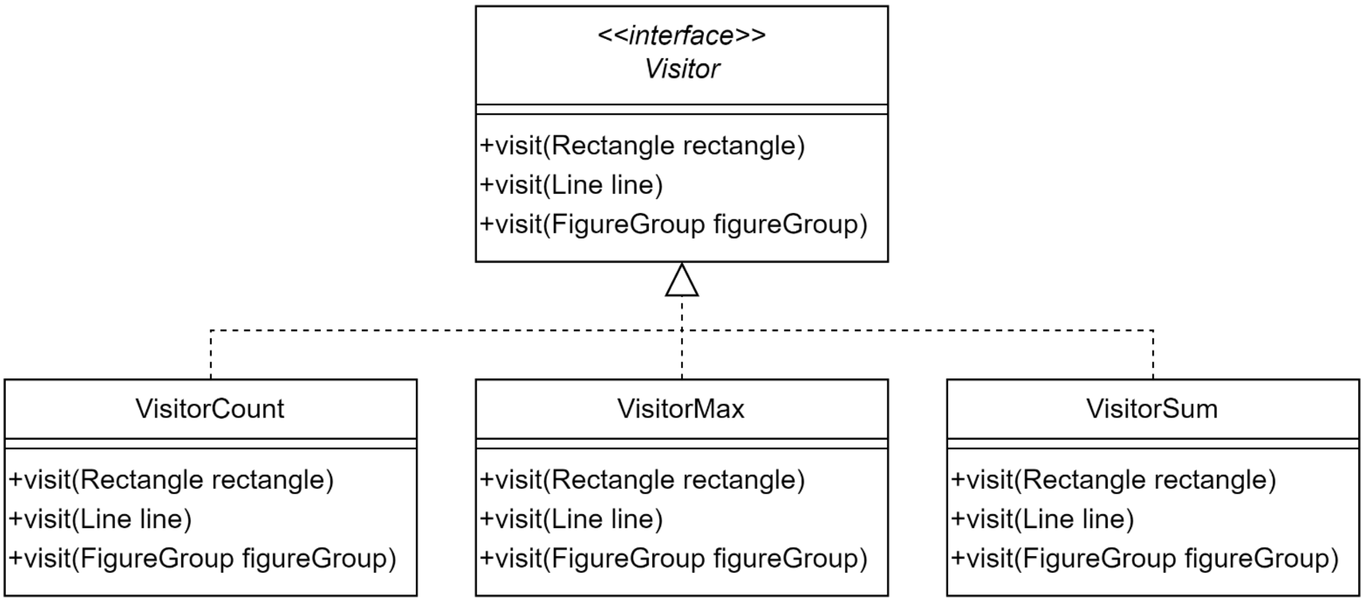
1) 개요


<<abstract>>
Figure
value
+getValue() : int
+draw()
+clone()
+accept(Visitor visitor)
Rectangle
+draw()
+accept(Visitor visitor)
Line
+draw()
+accept(Visitor visitor)
FigureGroup
+draw()
+clone()
+add(Figure figure)
+accept(Visitor visitor)


<<interface>>
Visitor
+visit(Rectangle rectangle)
+visit(Line line)
+visit(FigureGroup figureGroup)
VisitorCount
+visit(Rectangle rectangle)
+visit(Line line)
+visit(FigureGroup figureGroup)
VisitorMax
+visit(Rectangle rectangle)
+visit(Line line)
+visit(FigureGroup figureGroup)
VisitorSum
+visit(Rectangle rectangle)
+visit(Line line)
+visit(FigureGroup figureGroup)

2) Figure.java

```java
package visitor.e2;

public abstract class Figure implements Cloneable {
    int value;

    public Figure(int value) {
        this.value = value;
    }

    public int getValue() {
        return value;
    }

    public abstract void draw(int indent);

    @Override
    public Figure clone() throws CloneNotSupportedException {
        return (Figure)super.clone();
    }

    public abstract void accept(Visitor visitor);
}
```

3) Rectangle.java

```java
package visitor.e2;

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 accept(Visitor visitor) {
        visitor.visit(this);
    }
}
```

4) Line.java

```java
package visitor.e2;

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 accept(Visitor visitor) {
        visitor.visit(this);
    }

}
```

5) FigureGroup.java

```java
package visitor.e2;

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(₩n", padding);
        for (Figure figure : figures)
            figure.draw(indent + 1);
        System.out.printf("%s)₩n", 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 void accept(Visitor visitor) {
        visitor.visit(this);
        for (Figure f : figures)
            f.accept(visitor);
    }

}
```

6) Visitor.java

```java
package visitor.e2;

public interface Visitor {
    void visit(Rectangle rectangle);
    void visit(Line line);
    void visit(FigureGroup group);
}
```

7) VisitorCount.java

```java
package visitor.e2;

public class VisitorCount implements Visitor {
    int lineCount = 0;
    int rectangleCount = 0;
    int groupCount = 0;


    @Override
    public void visit(Rectangle rectangle) {
        ++rectangleCount;
    }

    @Override
    public void visit(Line line) {
        ++lineCount;
    }

    @Override
    public void visit(FigureGroup group) {
        ++groupCount;
    }

    @Override
    public String toString() {
        return String.format("line=%d rectangle=%d group=%d₩n",
                lineCount, rectangleCount, groupCount);
    }
}
```

8) VisitorMax.java

```java
package visitor.e2;

public class VisitorMax implements Visitor {
     int max = Integer.MIN_VALUE;

    @Override
    public void visit(Rectangle rectangle) {
        max = Math.max(max, rectangle.getValue());
    }

    @Override
    public void visit(Line line) {
        max = Math.max(max, line.getValue());
    }

    @Override
    public void visit(FigureGroup group) {
        max = Math.max(max, group.getValue());
    }

    @Override
    public String toString() {
        return String.format("max=%d₩n", max);
    }

}
```

9) VisitorSum.java

```java
package visitor.e2;

public class VisitorSum implements Visitor {
     int sum = 0;

    @Override
    public void visit(Rectangle rectangle) {
        sum += rectangle.getValue();
    }

    @Override
    public void visit(Line line) {
        sum += line.getValue();
    }

    @Override
    public void visit(FigureGroup group) {
        sum += group.getValue();
    }

    @Override
    public String toString() {
        return String.format("sum=%d₩n", sum);
    }
}
```

10) Example2.java

```java
package visitor.e2;

import java.util.Scanner;

public class Example2 {

    static FigureGroup root = new FigureGroup(0);

    static void drawFigures() {
        System.out.println();
        for (int i = 0; i < root.getCount(); ++i) {
            System.out.printf("%d: ", i);
            root.get(i).draw(0);
        }
        System.out.println();
    }

    static void execute(String cmd) {
        try {
            Visitor visitor = null;
            String[] a = cmd.split(" +");
            switch (a[0].toLowerCase()) {
            case "rectangle": root.add(new Rectangle(Integer.valueOf(a[1]))); break;
            case "line": root.add(new Line(Integer.valueOf(a[1]))); break;
            case "remove":
                int index = Integer.valueOf(a[1]);
                root.remove(index);
                break;
            case "duplicate":
                Figure figure = root.get(Integer.valueOf(a[1]));
                root.add(figure.clone());
                break;
            case "group":
                FigureGroup group = new FigureGroup(0);
                for (int i = 1; i < a.length; ++i)
                    group.add(root.get(Integer.valueOf(a[i])));
                root.add(group);
                for (int i = 0; i < group.getCount(); ++i)
                    root.remove(group.get(i));
                break;
            case "max":
                root.accept(visitor = new VisitorMax());
                System.out.println(visitor);
                break;
            case "sum":
                root.accept(visitor = new VisitorSum());
                System.out.println(visitor);
                break;
            case "count":
                root.accept(visitor = new VisitorCount());
                System.out.println(visitor);
                break;
            case "quit": System.exit(0); break;
            }
        } catch (Exception e) {
        }
    }

    static void prompt() {
        System.out.printf("  사각형 : rectangle 정수₩n");
        System.out.printf("  선     : line 정수₩n");
        System.out.printf("  삭제   : remove 번호₩n");
        System.out.printf("  복제   : duplicate 번호₩n");
        System.out.printf("  그룹   : group 번호1, 번호2,...₩n");
        System.out.printf("  최대값 : max₩n");
        System.out.printf("  합계   : sum₩n");
        System.out.printf("  도형 수: count₩n");
        System.out.printf("  종료   : quit₩n");
        System.out.printf("  ? ");
    }

    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {
            while (true) {
                prompt();
                String cmd = scanner.nextLine();
                execute(cmd);
                drawFigures();
            }
        }
    }
}
```

최소값 찾기 등 새 작업을 구현할 때 수정해야 하는 부분은?