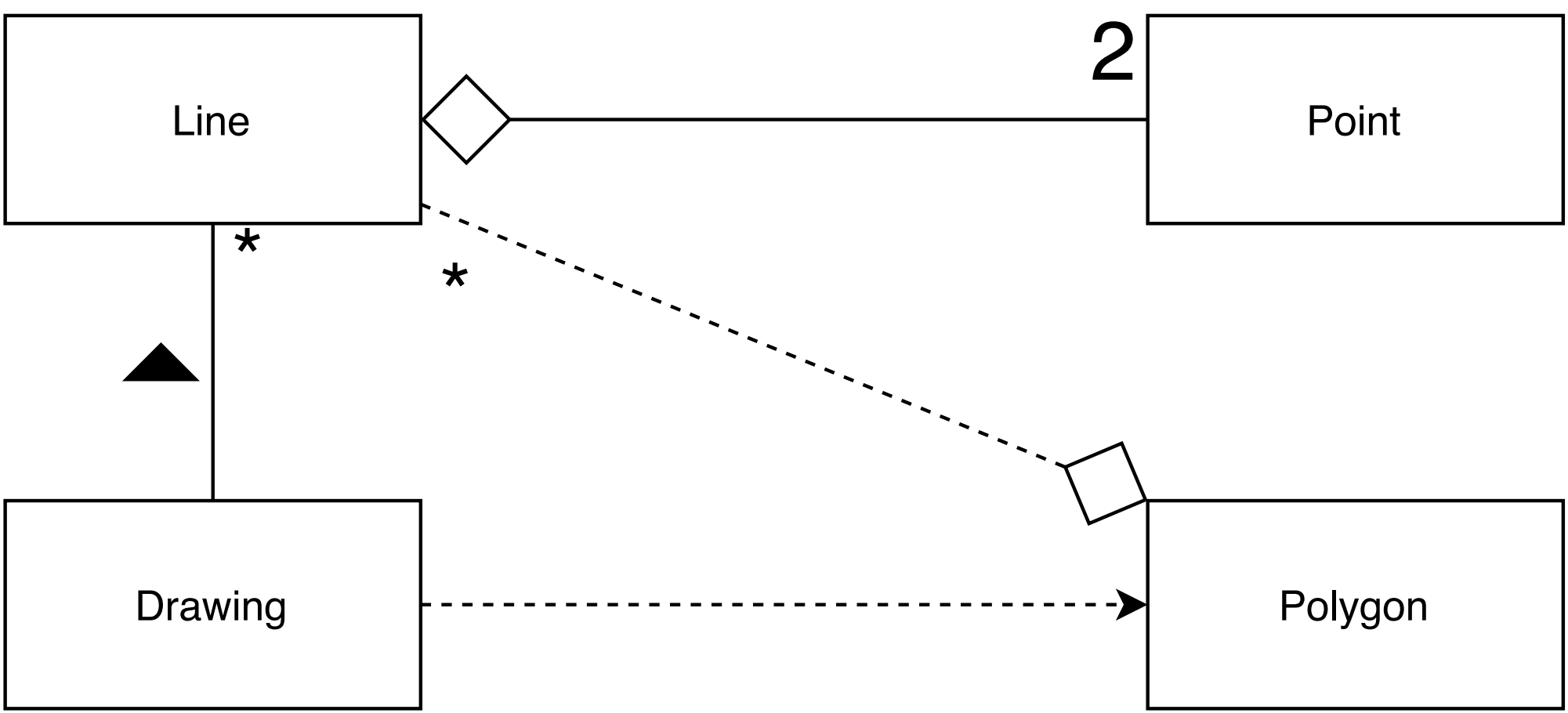
Line
2
Point
*
*
Drawing
Polygon

```
The lines in polygon 1 are:
        Line 1: starts at (20, 30) and ends at (50, 100)
        Line 2: starts at (50, 100) and ends at (105, 30)
        Line 3: starts at (105, 30) and ends at (20, 30)
The perimeter of the polygon 1 is: 250.18
The lines in polygon 2 are:
        Line 4: starts at (120, 130) and ends at (150, 200)
        Line 5: starts at (150, 200) and ends at (200, 130)
        Line 6: starts at (200, 130) and ends at (120, 130)
The perimeter of the polygon 2 is: 242.18
The lines in polygon 3 are:
        Line 7: starts at (320, 330) and ends at (250, 400)
        Line 8: starts at (250, 400) and ends at (400, 330)
        Line 9: starts at (400, 330) and ends at (320, 330)
The perimeter of the polygon 3 is: 344.52
```

```java
// Point.java

class Point {
    private int x, y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    static public double distance(Point a, Point b){
        double diffx = a.x - b.x;
        double diffy = a.y - b.y;
        return Math.sqrt(diffx * diffx + diffy * diffy);
    }

    public String toString(){
        // Intended Format:
        // (20, 30)

        String s = "(" + x + ", " + y + ")";
        return  s;
    }
}

```

```java
// Line.java

class Line {

    Point start, end;
    private static int classID = 0;
    private int objID;

    /*
     * Default constructor, makes a line from two points, where each point is the end of the line.
     */
    public Line(Point a, Point b) {
        start = a;
        end = b;
        objID = ++ classID;
    }


    public double distance(){
        return Point.distance(start, end);
    }

    public String toString()
    {
        // Intended Format:
        // Line 1: starts at (20, 30), and ends at (50, 100)

        String s = "Line " + objID + ": starts at " + start.toString() + " and ends at " + end.toString();
        return s;
    }
}

```

```java
// Polygon.java

import java.util.*;

class Polygon {
    private final LinkedHashSet <Line> polygon;
    private int objID;
    private static int classID;
    Iterator <Line> it;

    public Polygon(LinkedHashSet<Line> polygon) {
         this.polygon = new LinkedHashSet<Line>();
         for(Line l: polygon)
             this.polygon.add (l);
         objID = ++ classID;
         it = this.polygon.iterator();
    }

    public Iterator <Line> getLine() {
        it = polygon.iterator();
        return it;
    }

    public static int classID(){
        return classID;
    }


    public String toString() {
        // Intended Format:
        // The lines in polygon 1 are:
        //   Line 1: starts at (20, 30), and ends at (50, 100)
        //   Line x: etc.

        Iterator<Line> it = getLine();

        String s = "The lines in polygon " + objID + " are:";

        while(it.hasNext()) {
            s += "\n\t" + it.next().toString();

        }

        return s;
    }
}
```

```java
// Drawing.java

import java.util.Iterator;
import java.util.LinkedHashSet;

public class Drawing {
    LinkedHashSet<Line> lines;

    public void drawPolygon(LinkedHashSet <Line> lines){
        this.lines = lines;
        Polygon p = new Polygon(lines);
        System.out.println(p);
        System.out.printf("The perimeter of the polygon %d is: %.2f \n" , Polygon.classID(), perimeter(p));
    }

    private double perimeter(Polygon p){
        Iterator<Line> it = p.getLine();
        double perim = 0;
        while(it.hasNext()){
            perim += it.next().distance();
        }
        return perim;
    }

    public static void main(String[] args) {
        Drawing drawing = new Drawing();

        Point [] points = {
                           new Point(20,30), new Point (50, 100), new Point (105, 30),
                           new Point(120,130), new Point (150, 200), new Point (200, 130),
                           new Point(320,330), new Point (250, 400), new Point (400, 330)
                           };

        Line [] lines = {
                           new Line(points[0], points[1]),
                           new Line(points[1], points[2]),
                           new Line(points[2], points[0]),
                           new Line(points[3], points[4]),
                           new Line(points[4], points[5]),
                           new Line(points[5], points[3]),
                           new Line(points[6], points[7]),
                           new Line(points[7], points[8]),
                           new Line(points[8], points[6])
                           };

        LinkedHashSet<Line> poly1 = new LinkedHashSet<Line>();
        poly1.add(lines[0]);
        poly1.add(lines[1]);
        poly1.add(lines[2]);

        drawing.drawPolygon(poly1);

        LinkedHashSet<Line> poly2 = new LinkedHashSet<Line>();
        poly2.add(lines[3]);
        poly2.add(lines[4]);
        poly2.add(lines[5]);

        drawing.drawPolygon(poly2);
```

```
59
60          LinkedHashSet<Line> poly3 = new LinkedHashSet<Line>();
61          poly3.add(lines[6]);
62          poly3.add(lines[7]);
63          poly3.add(lines[8]);
64
65          drawing.drawPolygon(poly3);
66      }
67
68  }
69
```