

comum com o método *setY* (linhas 27-30) dentre os sete métodos restantes, então $Q = 10 + 1 = 11$, e $P = 7 + 6 = 13$.

```

01 public class Point implements Subject {
02
03     private HashSet observers;
04
05     private int x;
06
07     private int y;
08
09     private Color color;
10
11     public Point(int x, int y, Color color) {
12         this.x=x;
13         this.y=y;
14         this.color=color;
15         this.observers = new HashSet();
16     }
17
18     public int getX() { return x; }
19
20     public int getY() { return y; }
21
22     public void setX(int x) {
23         this.x=x;
24         notifyObservers();
25     }
26
27     public void setY(int y) {
28         this.y=y;
29         notifyObservers();
30     }
31
32     public Color getColor() { return color; }
33
34     public void setColor(Color color) {
35         this.color=color;
36         notifyObservers();
37     }
38
39     public void attach(Observer o) {
40         this.observers.add(o);
41     }
42
43     public void detach(Observer o) {
44         this.observers.remove(o);
45     }
46
47     public void notifyObservers() {
48         for (Iterator e = observers.iterator() ; e.hasNext() ;) {
49             ((Observer)e.next()).update(this);
50         }
51     }
52 }

```

Figura 11 – Código para exemplificar o uso da métrica de coesão LCOO

A seguir é a vez do método *setX* (linhas 22-25) que não manipula nenhum atributo comum aos seis métodos restantes, logo $Q = 11 + 0 = 11$, e $P = 13 + 6 = 19$. Da mesma forma, o método *setY* (linhas 27-30) não manipula nenhum atributo comum aos cinco métodos restantes, logo $Q = 11 + 0 = 11$, e $P = 19 + 5 = 24$. O método *getColor* (linha 32) usa o mesmo atributo que o método *setColor* (linhas