

Shape.java

```
1 import java.awt.Canvas;
6
7 public abstract class Shape extends Canvas implements
  ShapePositionInterface{
8     private int x;
9     private int y;
10    private Color color;
11
12
13
14    public Shape() {
15        super();
16    }
17
18    public Shape(int x, int y) {
19        super();
20        this.x = x;
21        this.y = y;
22    }
23
24    //constructor
25    public Shape(int x, int y, Color color) {
26        super();
27        this.x = x;
28        this.y = y;
29        this.color = color;
30    }
31
32
33
34    //getters
35    public int getX() {
36        return x;
37    }
38    public int getY() {
39        return y;
40    }
41    public Color getColor() {
42        return color;
43    }
44
```

Shape.java

```
45
46 //setters
47 public void setX(int x) {
48     this.x = x;
49 }
50 public void setY(int y) {
51     this.y = y;
52 }
53 public void setColor(Color color) {
54     this.color = color;
55 }
56
57
58 @Override
59 public int[] getPoint()
60 {
61     int[] point = {this.x, this.y};
62     return point;
63 }
64
65 @Override
66 public void moveTo(int dx, int dy)
67 {
68     this.x = x + dx;
69     this.y = y + dx;
70 }
71
72 @Override
73 public double distanceTo(int x, int y) {
74     double distance = Math.sqrt(Math.pow(x - this.x, 2) + Math.pow(y
- this.y, 2));
75     return distance;
76 }
77
78 @Override
79 public boolean doOverlap(Shape s1, Shape s2)
80 {
81     int[] s1Box = this.getBoundingBox();
82     int[] s2Box = this.getBoundingBox();
83
84     boolean xOverlap = false;
```

Shape.java

```
85     boolean yOverlap = false;
86
87     if(s1Box[0] >= s2Box[0] && s1Box[0] <= s2Box[1])
88     {
89         xOverlap = true;
90     }else if(s1Box[1] >= s2Box[0] && s1Box[1] <= s2Box[1])
91     {
92         xOverlap = true;
93     }else if(s1Box[0] >= s2Box[0] && s1Box[1] <= s2Box[1])
94     {
95         xOverlap = true;
96     }else if(s1Box[1] >= s2Box[0] && s1Box[0] <= s2Box[1])
97     {
98         xOverlap = true;
99     }
100
101
102     if(s1Box[2] >= s2Box[2] && s1Box[2] <= s2Box[3])
103     {
104         yOverlap = true;
105     }else if(s1Box[3] >= s2Box[2] && s1Box[3] <= s2Box[3])
106     {
107         yOverlap = true;
108     }else if(s1Box[2] >= s2Box[2] && s1Box[2] <= s2Box[3])
109     {
110         yOverlap = true;
111     }else if(s1Box[3] >= s2Box[2] && s1Box[3] <= s2Box[3])
112     {
113         yOverlap = true;
114     }
115
116     if(xOverlap && yOverlap)
117     {
118         return true;
119     }else
120     {
121         return false;
122     }
123
124
125 }
```

Shape.java

```
126
127     public abstract void setWidth(int width);
128     public abstract void setHeight(int width);
129     public abstract int getWidth();
130     public abstract int getHeight();
131     //abstract methods
132     @Override
133     public abstract int[] getBoundingBox();
134     public abstract void draw(Graphics g);
135
136
137 }
138
```