```java
import java.awt.Color;

public  class MyPolygon extends Shape
{
    private int N;
    private int radius; //rad. of circle
    private int height = radius *2;
    private int width = radius *2;

    public MyPolygon() {
        super();
    }

    public MyPolygon(int height, int width) {
        super();
        this.height = height;
        this.width = width;
    }


    public void setWidth(int width) {
        this.width = width;
    }

    public void setHeight(int width) {
        this.width = width;
    }
    public int getWidth() {return this.width;}
    public int getHeight() {return this.height;}

    //getters
    public int getRadius()
    {return radius;}

    //setters
    public void setRadius(int radius)
    {this.radius = radius;}


    //returns the area of Polygon
    @Override
```

```
47      public double getArea()
48      {
49          double area = (1/4) * N * ((getSide() * getSide())/
    (Math.tan(Math.PI/N)));
50          return area;
51      }
52
53      @Override
54      //returns the Perimeter of Polygon
55      public double getPerimeter()
56      {return N * getSide();}
57
58
59      //returns the interior angle (in degrees) of the Polygon
60      public double getAngle()
61      {
62          double angleInterior = (N-2) * 180/N;
63          return angleInterior;
64      }
65
66
67
68      public int getSide()
69      {
70          int sideLength = (int) (2 * getRadius() * Math.PI/N);
71          return sideLength;
72      }
73
74
75      public String toString()
76      {
77          String result = "( side length = " + getSide() +
78                  ", interior angle = " + getAngle()  + ", perimeter = " +
79                  getPerimeter() + ", area = " + getArea() + ")";
80          return result;
81      }
82
83
84      @Override
85      public int[] getBoundingBox()
86      {
```

```
87          int x1 = getX();
88          int y1 = getY();
89          int x2 = x1 + (radius *2);
90          int y2 = y1 + (radius *2);
91          int[] box = {x1,x2,y1,y2};
92
93          return box;
94      }
95
96
97      @Override
98      public void draw(Graphics g)
99      {
100         Polygon p = new Polygon();
101         for (int i = 0; i < N; i++)
102         {
103             //p.addPoint(x, y);
104             p.addPoint((int) (getX() + getRadius() * Math.cos(i * 2 *
    Math.PI / N)),
105                     (int) (getY()- getRadius() * Math.sin(i * 2 *
    Math.PI / N)));
106         }
107
108         g.setColor(getColor());
109         g.fillPolygon(p);
110     }
111
112 }
113
```