Comer, Nicholas G.

Report

This report details some of the quirks in the code provided. First, The X and Y coordinates refer to the center of the shape, therefore all the shapes in the output have the same X and Y coordinates. In order to find the bounding box and draw the shapes, I found the upper right corner by subtracting half the width from the x-coordinate. From there it is trivial to find the rest of the corners. The shape class is abstract, meaning that not all the functions implemented from the interfaces have bodies. These abstract functions will be overwritten in subclasses such as polygon and rectangle. The center of the shapes is the center of the canvas. The center of the canvas is always half the width and half the height. I created the output picture by using JFrame.

Application.java

1 import java.awt.Canvas;

2 import java.awt.Color;

3 import java.awt.Graphics;

4 import java.lang.Object;

5 import java.awt.\*;

6 import javax.swing.JFrame;

7

8 import javax.swing.JFrame;

9

10 public class Application extends Canvas {

11

12public static void main(String[] args)

13{

14JFrame frame = new JFrame("Shapes");

15Canvas canvas = new Application();

16canvas.setSize(1080,720);

17frame.add(canvas);

18frame.pack();

19frame.setVisible(true);

20}

21

22public void paint(Graphics g)

23{

24Rectangle R1 = new Rectangle(540,360,800,400,Color.yellow);

25Oval O1 = new Oval(540,360,800,400,Color.green);

26Rectangle R2 = new Rectangle(540,360,525,300,Color.magenta);

27Oval O2= new Oval(540,360,525,300,Color.blue);

28Rectangle R3 = new Rectangle(540,360,345,225,Color.orange);

29Oval O3 = new Oval(540,360,345,225,Color.gray);

30

31R1.draw(g);

32O1.draw(g);

33R2.draw(g);

34O2.draw(g);

35R3.draw(g);

36O3.draw(g);

37}

38

39 }

40