/\*\*

\* RectangleViewer.java

\* Hunter Damron

\* Honor Code: Cheat I did not

\* Purpose: Prints the rectangles formed by PrintRectangle.java

\* with a JFrame

\*/

import javax.swing.JFrame;

import javax.swing.JComponent;

public class RectangleViewer {

public static void main(String[] args) {

JFrame frame = new JFrame();

frame.setSize(600, 400);

frame.setTitle("3 Random Rectangles");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JComponent component = new PrintRectangle();

frame.add(component);

frame.setVisible(true);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* RectangleIntersectionAgain.java

\* Hunter Damron

\* Purpose: Generates three random rectangles in a 600 x 400 plane with width of 50 to 200 and height of 50 to 100;

\* Generates rectangles which represent the intersections of rectangles including one for the intersection of all three;

\* Prepares rectangles for printing in JFrame component through RectancleViewer.java

\* Honor Statement: On my honor, I have neither given nor received any unauthorized help on this assignment

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.awt.Rectangle; //import for Rectangle class

import java.util.Random; //import for Rectangle class

import javax.swing.JComponent;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Color;

public class PrintRectangle extends JComponent {

public void paintComponent(Graphics g) {

Graphics2D g2 = (Graphics2D) g;

Random generator = new Random(); //Random object to generate random numbers

int topx; //Top left corner of Rectangles (reused between Rectangles)

int topy; //Top right corner of Rectangles (reused between Rectangles)

int width; //Width of Rectangles (reused between Rectangles)

int height; //Height of Rectangles (reused between Rectangles)

boolean out = false; //Boolean to test if anything is outputted to determine if a no intersection is appropriate

Rectangle r1, r2, r3, r12, r23, r31, rall; //All rectangles and intersections

width = generator.nextInt(151) + 50; //Generates width between 50 and 200, inclusive

height = generator.nextInt(51) + 50; //Generates height between 50 and 100, inclusive

topx = generator.nextInt(601 - width); //Generates topx so entire Rectangle is between 0 and 600, inclusive

topy = generator.nextInt(401 - height); //Generates topy so entire Rectangle is between 0 and 600, inclusive

r1 = new Rectangle(topx, topy, width, height); //Generates first Rectangle r1

width = generator.nextInt(151) + 50; /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

height = generator.nextInt(51) + 50; /\* Same as above but for \*/

topx = generator.nextInt(601 - width); /\* Rectangle r2 \*/

topy = generator.nextInt(401 - height); /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

r2 = new Rectangle(topx, topy, width, height); //Generates second Rectangle r2

width = generator.nextInt(151) + 50; /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

height = generator.nextInt(51) + 50; /\* Same as above but for \*/

topx = generator.nextInt(601 - width); /\* Rectangle r2 \*/

topy = generator.nextInt(401 - height); /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

r3 = new Rectangle(topx, topy, width, height); //Generates third Rectangle r3

// System.out.println("// r1: " + r1 + "\n// r2: " + r2 + "\n// r3: " + r3); //Prints the three Rectangles

g2.setColor(Color.RED);

g2.draw(r1);

g2.setColor(Color.GREEN);

g2.draw(r2);

g2.setColor(Color.BLUE);

g2.draw(r3);

if(r1.intersects(r2)) { //Continues to create intersection of r1 and r2 if any

r12 = r1.intersection(r2); //Makes Rectangle r12 into intersection of r1 and r2

// System.out.println("Rectangle r1 intersects r2 at " + r12); //Prints intersection

g2.setColor(Color.YELLOW);

g2.draw(r12);

g2.fill(r12);

out = true; //Sets out to true because there is an intersection

}

if(r2.intersects(r3)) { //Continues to create intersection of r2 and r3 if any

r23 = r2.intersection(r3); //Makes Rectangle r23 into intersection of r2 and r3

// System.out.println("Rectangle r2 intersects r3 at " + r23); //Prints intersection

g2.setColor(Color.CYAN);

g2.draw(r23);

g2.fill(r23);

out = true; //Sets out to true because there is an intersection

}

if(r3.intersects(r1)) { //Continues to create intersection of r3 and r1

r31 = r3.intersection(r1); //Makes Rectangle r31 into intersection of r3 and r1

// System.out.println("Rectangle r3 intersects r1 at " + r31); //Prints intersection

g2.setColor(Color.MAGENTA);

g2.draw(r31);

g2.fill(r31);

out = true; //Sets out to true because there is an intersection

if(r31.intersects(r2)) { //Continues to create intersection of r31 and r2 which is the intersection of all three

rall = r31.intersection(r2); //Makes Rectangle rall into intersection of all r31 and r2

// System.out.println("All Rectangles intersect at " + rall); //Prints intersection

g2.setColor(Color.BLACK);

g2.fill(rall);

g2.draw(rall);

}

}

if(!out) { //Continues to print no intersection if out has not been set to true (meaning no intersection is present)

// System.out.println("No intersections were detected"); //Prints no intersection

}

}

}