

Java2D – Drawing Circles – One Trace

Use Concept Trace Slicing

Supporting Applications : [TerpPaint]

```
import java.awt.image.IndexColorModel;
import java.awt.Graphics2D;
import java.awt.image.ColorModel;
import java.awt.GraphicsConfiguration;
import java.awt.LayoutManager;
import java.awt.Frame;
import java.awt.Color;
import java.awt.Point;
import java.util.Hashtable;
import java.awt.BasicStroke;
import java.awt.event.ActionListener;
import java.awt.image.ImageObserver;
import java.awt.image.BufferedImage;
import java.awt.geom.Ellipse2D$Double;
import java.awt.geom.Point2D;
import java.awt.geom.AffineTransform;
import java.awt.Component;
import java.awt.event.MouseEvent;
import java.awt.PopupMenu;
import java.awt.Graphics;
import java.awt.Container;
import java.awt.Rectangle;
import java.awt.Dimension;
import java.awt.image.WritableRaster;

public class AppctionListener
implements ActionListener {
}

public class SomeClass {

    public void someMethod() {
        Dimension dimension = new Dimension(Dimension)|| (int,int)|| ();
        Component.setPreferredSize(dimension);
        Component.addMouseListener(MouseListener);
        Component.addMouseMotionListener(MouseMotionListener);
        Rectangle rectangle = new Rectangle()|(Rectangle)|| (int,int,int,int)|| (int,int)|| (Point,Dimension)|| (Dimension)|| (Point); // REPEATED!
        BufferedImage bufferedImage = new BufferedImage(int,int,int)|| (int,int,int,IndexColorModel)|| (ColorModel,WritableRaster,boolean,Hashtable); // REPEATED!
        int app_int = MouseEvent.getX(); // REPEATED!
        int app_int1 = MouseEvent.getY(); // REPEATED!
        Ellipse2D$Double ellipse2D$Double = new Ellipse2D$Double(app_int && app_int1); // REPEATED!
        Frame frame = new Frame(GraphicsConfiguration)|| (String)|| (String,GraphicsConfiguration)|| ();
        Container container = frame.getParent();
        AppctionListener appctionListener = new AppctionListener(frame);
        Dimension dimension1 = Component.getPreferredSize();
        String string = dimension1.toString();
        Container.setLayout(LayoutManager);
        Component.add(PopupMenu);
        app_void9.paintAll(Graphics); // REPEATED!
        Component.repaint():()|| (int,int,int,int):()|| (long,int,int,int,int):()|| (long):(); // REPEATED!
        double app_double1 = Point2D.getX(); // REPEATED!
        double app_double = Point2D.getY(); // REPEATED!
        BasicStroke basicStroke = new BasicStroke(float,int,int)|| (float)|| ()|| (float,int,int,float)|| (float,int,int,float,float[],float); // REPEATED!
        AffineTransform affineTransform = new AffineTransform(AffineTransform)|| (float,float,float,float,float)|| (float[])||
        (double,double,double,double,double,double)|| (double[])|| ()|| (double,double,double,double,double,double,int);
        affineTransform.setToScale(double,double);
        /*
        Cyclic Statements
        graphics2D.draw(rectangle && ellipse2D$Double); // REPEATED!
        Graphics2D graphics2D = bufferedImage.createGraphics(); // REPEATED!
        int app_int5 = bufferedImage.getHeight(); // REPEATED!
        bufferedImage1.flush(); // REPEATED!
        graphics2D.drawImage(bufferedImage && affineTransform); // REPEATED!
        int app_int3 = bufferedImage.getWidth(ImageObserver); // REPEATED!
        int app_int2 = bufferedImage.getWidth(); // REPEATED!
        graphics2D.fillRect(int,int,int,int); // REPEATED!
        graphics2D.setColor(Color); // REPEATED!
        int app_int4 = bufferedImage.getHeight(ImageObserver); // REPEATED!
        BufferedImage bufferedImage1 = bufferedImage.getSubimage(int,int,int,int);
        graphics2D.setStroke(basicStroke); // REPEATED!
    }
}
```