

Java2D – Drawing Circles – One Trace Use Full Trace (No Slicing)

Supporting Applications : [TerpPaint]

```
import java.awt.image.ColorModel;
import java.awt.image.IndexColorModel;
import java.awt.Graphics2D;
import java.awt.Window;
import java.awt.Frame;
import java.awt.Color;
import java.awt.Insets;
import java.awt.Point;
import java.awt.BasicStroke;
import java.awt.geom.Ellipse2D$Double;
import java.awt.print.PageFormat;
import java.awt.geom.AffineTransform;
import java.awt.GridBagLayout;
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.Dialog;
import java.awt.Rectangle;
import java.awt.Dimension;
import java.awt.GraphicsConfiguration;
import java.awt.GridBagConstraints;
import java.util.Hashtable;
import java.awt.event.ActionListener;
import java.awt.image.ImageObserver;
import java.awt.event.WindowAdapter;
import java.awt.image.BufferedImage;
import java.awt.print.PrinterJob;
import java.awt.event.MouseEvent;
import java.awt.Toolkit;
import java.awt.Graphics;
import java.awt.Container;
import java.awt.image.WritableRaster;

public class AppctionListener
implements ActionListener {
}

public class AppWindowAdapter
extends WindowAdapter {
}

public class SomeClass {
    public void someMethod() {
        Container container1 = new Container();
        PrinterJob.getPrinterJob();
        Rectangle rectangle = new Rectangle();|(Rectangle)|(int,int,int,int)|(int,int)|(Point,Dimension)|(Dimension)|(Point); // REPEATED!
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension2 = toolkit.getScreenSize();
        Frame frame = new Frame(GraphicsConfiguration)|(String)|(String,GraphicsConfiguration)|();
        AppctionListener appctionListener = new AppctionListener(frame);
        Container container = frame.getParent();
        Point point = new Point(Point)|(int,int)|();
        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[])|(double,double,double,double)|(double[])|()|(double,double,double,double,int);
        affineTransform.setToScale(double,double);
        GridBagConstraints gridBagConstraints = new GridBagConstraints(int,int,int,int,double,double,int,int,Insets,int,int)|();
        Component.getLocation():(Point)|(Point):(Point);
        double app_double1 = Point2D.getX(); // REPEATED!
        double app_double = Point2D.getY(); // REPEATED!
        AppWindowAdapter appWindowAdapter = new AppWindowAdapter();
        BorderLayout borderLayout = new BorderLayout(int,int)|();
        Dimension dimension = new Dimension(Dimension)|(int,int)|();
        Component.setPreferredSize(dimension);
        Component.setSize(int,int):()|(Dimension):();
        Component.getBounds(Rectangle):(Rectangle)|():(Rectangle);
        Component.setLocation(Point):()|(int,int):();
        Dialog.setTitle(String);
        Component.setVisible(boolean);
        Component.addMouseListener(MouseListener);
        Component.addMouseMotionListener(MouseMotionListener);
        Window.addWindowListener(appWindowAdapter);
        Window.pack();
        BufferedImage bufferedImage = new BufferedImage(int,int,int)|(int,int,int,IndexColorModel)|(ColorModel,WritableRaster,boolean,Hashtable); // REPEATED!
        Insets insets = new Insets(int,int,int,int);
        GridBagLayout gridBagLayout = new GridBagLayout();
        int app_int = MouseEvent.getX(); // REPEATED!
        int app_int1 = MouseEvent.getY(); // REPEATED!
        Ellipse2D$Double ellipse2D$Double = new Ellipse2D$Double(app_int && app_int1); // REPEATED!
        Dimension dimension1 = Component.getPreferredSize();
        String string = dimension1.toString();
        Container.setLayout(gridBagLayout && borderLayout);
        Component.add(gridBagConstraints);
        app_void4.paintAll(Graphics); // REPEATED!
        Component.repaint():()|(int,int,int,int):()|(long,int,int,int,int):()|(long):(); // REPEATED!
        PageFormat pageFormat = new PageFormat();
        /* Cyclic Statements */
        graphics2D.draw(rectangle && ellipse2D$Double); // REPEATED!
        int app_int5 = bufferedImage.getHeight(); // REPEATED!
        graphics2D.drawImage(bufferedImage && affineTransform); // REPEATED!
        int app_int3 = bufferedImage.getWidth(ImageObserver); // REPEATED!
        graphics2D.fillRect(int,int,int,int);
        int app_int4 = bufferedImage.getHeight(ImageObserver); // REPEATED!
        BufferedImage bufferedImage1 = bufferedImage.getSubimage(int,int,int,int);
        Graphics2D graphics2D = bufferedImage.createGraphics(); // REPEATED!
        bufferedImage1.flush(); // REPEATED!
        int app_int2 = bufferedImage.getWidth(); // REPEATED!
        graphics2D.setColor(Color); // REPEATED!
        graphics2D.setStroke(basicStroke); // REPEATED!
    }
}
```