package RotateText;  
  
import java.awt.\*;  
import java.awt.geom.\*;  
import java.awt.font.\*;  
import javax.swing.\*;  
  
public class RotateText extends JPanel{  
 static int *angdeg*=0;  
 static int *init\_fontSize* = 100;  
  
 @Override  
 // importing paint method to use Graphics library  
 public void paint(Graphics g){  
 // creating 2D graphic object  
 Graphics2D g2d = (Graphics2D) g;  
 //  
 g2d.setRenderingHint(RenderingHints.*KEY\_ANTIALIASING*, RenderingHints.*VALUE\_ANTIALIAS\_ON*); //  
 //setting background  
 g2d.setColor(Color.*YELLOW*); //to remove trail of painting  
 g2d.fillRect(0,0,getWidth(),getHeight());  
  
 // creating font  
 Font font = new Font("times new roman",Font.*ROMAN\_BASELINE*,*init\_fontSize*);  
 g2d.setFont(font); //setting font of surfaceh  
 FontRenderContext frc = g2d.getFontRenderContext();  
 TextLayout layout = new TextLayout("O", font, frc);  
  
 //getting width & height of the text  
 double sw = layout.getBounds().getWidth();  
 double sh = layout.getBounds().getHeight();  
  
 //getting original transform instance  
 AffineTransform saveTransform=g2d.getTransform();  
 g2d.setColor(Color.*green*);  
 Rectangle rect = this.getBounds();  
  
 /\*creating instance set the translation to the mid of the component\*/  
 AffineTransform affineTransform = new AffineTransform();  
 affineTransform.setToTranslation((rect.width)/1.5,(rect.height)/1.5);  
  
 //rotate with the anchor point as the mid of the text  
 affineTransform.rotate(Math.*toRadians*(*angdeg*), 0, 0);  
 g2d.setTransform(affineTransform);  
 g2d.drawString("O",(int)-sw/2,(int)sh/2);  
  
 //restoring original transform  
 g2d.setTransform(saveTransform);  
 }  
  
 public static void main(String[] args) throws Exception{  
 JFrame frame = new JFrame("Rotated text");  
 RotateText rt=new RotateText();  
 frame.add(rt);  
 frame.setSize(500, 500);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setLocationRelativeTo(null);  
 frame.setVisible(true);  
  
 while(true){  
 Thread.*sleep*(10); //sleeping then increasing angle by 5  
 *angdeg*=(*angdeg*>=360)?0:*angdeg*+ 5; //  
 *init\_fontSize* = *init\_fontSize* + 1;  
 rt.repaint(); //repainting the surface  
 }  
 }  
}