```
import javax.swing.*; import java.awt.*; import java.awt.event.*; public class Program extends JFrame implements MouseListener{ Graphics g; int x1,y1,x2,y2; double xarr[]=new double[3]; double yarr[]=new double[3]; public Program(){ setSize(600,600); setVisible(true); setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); setTitle("Test Program"); setLocationRelativeTo(null); g=getGraphics(); addMouseListener(this); } public void DrawObject(){ DDA((int)xarr[0], (int)yarr[0], (int)xarr[1], (int)yarr[1]); DDA((int)xarr[1], (int)yarr[1], (int)xarr[2], (int)yarr[2]); DDA((int)xarr[2], (int)yarr[2], (int)xarr[0], (int)yarr[0]); } public void mouseEntered(MouseEvent m){} public void mouseClicked(MouseEvent m){ x1=m.getX(); y1=m.getY(); xarr[0]=x1; yarr[0]=y1; xarr[1]=x1; yarr[1]=y1-100; xarr[2]=x1+30; yarr[2]=y1-30; g.setColor(Color.BLUE); DrawObject(); Translate(-x1,-y1); ReflectX(); Translate(x1,y1); DrawObject(); } public void mousePressed(MouseEvent m){ } public void mouseReleased(MouseEvent m){ } public void mouseExited(MouseEvent m){} public void DDA(int x1,int y1,int x2,int y2){ double step,x,y,xinc,yinc; int dx,dy; dx=x2-x1; dy=y2-y1; if(Math.abs(dx)>Math.abs(dy)){ step=Math.abs(dx); }else{ step=Math.abs(dy); } x=x1; y=y1; g.drawLine((int)Math.round(x), (int)Math.round(y), (int)Math.round(x), (int)Math.round(y)); xinc=dx/step; yinc=dy/step; for(int i=1; i<=step; i++){ x=x+xinc; y=y+yinc; g.drawLine((int)Math.round(x), (int)Math.round(y), (int)Math.round(x), (int)Math.round(y)); } } public void Translate(int tx,int ty){ for(int i=0; i<3; i++){ xarr[i]=xarr[i]+tx; yarr[i]=yarr[i]+ty; } } public void Scaling(double sx, double sy){ for(int i=0; i<3; i++){ xarr[i]=xarr[i]*sx; yarr[i]=yarr[i]*sy; } } public void Rotation(double angle){ double teta=Math.toRadians(angle); double xx; for(int i=0; i<3; i++){ xx = xarr[i]*Math.cos(teta) - yarr[i]*Math.sin(teta); yarr[i] = xarr[i]*Math.sin(teta) + yarr[i]*Math.cos(teta); xarr[i] = xx; } } public void ReflectX(){ for(int i=0; i<3; i++){ xarr[i]=xarr[i]*1; yarr[i]=yarr[i]*-1; } } public void ReflectY(){ for(int i=0; i<3; i++){ xarr[i]=xarr[i]*-1; yarr[i]=yarr[i]*1; } } public void ReflectPerpendicular(){ for(int i=0; i<3; i++){ xarr[i]=-xarr[i]; yarr[i]=-yarr[i]; } } public static void main(String[] args){ new Program(); } }
```