

Re: Draw a scaled arrow

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-05/msg01866.html>

- *From:* RichT <someone@xxxxxxxxxxxxxx>
 - *Date:* Tue, 20 May 2008 21:50:17 +0100
-

Yes. From math class, matrix multiplication is *_not_* commutative:-) In particular, the normal concatenation of operations in AffineTransform applies the transformations in reverse order. See concatenate() and preConcatenate().

In the example below, each arrow is rotated about its tip, then scaled, then translated to its destination. Suppose you wanted to rotate each arrow about its center. The arrow is four units wide, so translate(-2, 0) before rotating. Experiment with adding the translate() in different places to see the effect.

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Polygon;
import java.awt.Shape;
import java.awt.geom.AffineTransform;
import javax.swing.JFrame;

/**
 * Test AffineTransform
 * @author John B. Matthews
 */
public class Arrows extends JFrame {

    private static Shape arrow = initPoly();
    private static AffineTransform at = new AffineTransform();

    public static void main(String args[]) {
        JFrame frame = new Arrows();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 400);
        frame.setVisible(true);
    }

    public void paint(Graphics g) {
        Graphics2D g2D = (Graphics2D) g;
        g2D.setPaint(Color.BLACK);
        g2D.drawLine(0, 200, 400, 200);
    }
}
```

Re: Draw a scaled arrow

```
g2D.drawLine(200, 0, 200, 400);
g2D.setPaint(Color.BLUE);
drawShape(g2D, 100, 100, Math.PI / 2);
drawShape(g2D, 300, 100, Math.PI);
drawShape(g2D, 100, 300, 0);
drawShape(g2D, 300, 300, -Math.PI / 2);
}

/**
 * Draw a rotated, scaled and translated arrow.
 * Note that the normal order of concatenation
 * applies the last transform first.
 * @see AffineTransform#concatenate()
 */
private void drawShape(Graphics2D g2D, int x, int y, double theta) {
    at.setToIdentity();
    at.translate(x, y);
    at.scale(30, 30);
    at.rotate(theta);
    Shape shape = at.createTransformedShape(arrow);
    g2D.fill(shape);
}

/** Create a west pointing arrow with the tip at the origin. */
private static Polygon initPoly()
{
    Polygon poly = new Polygon();
    poly.addPoint( 0, 0
```