DoubleSpongeBob_3_Scaling_Translation =>
        DoubleSpongeBob_4_Scaling_Translation_Synch

1. Add DrawingListener interface

[DrawingListener.java]

```
public interface DrawingListener {

        void originChanged(int w_newOriginX, int w_newOriginY);
}
```

2. Add list of listeners and addDrawingListener method to DrawingComponent

```
private ArrayList<DrawingListener> listeners;

public void addDrawingListener(DrawingListener listener) {
        listeners.add(listener);
}
```

3. Add notifyOriginChanged method to DrawingComponent and call
        it from mouseDragged

```
private void notifyOriginChanged(int w_newOriginX, int w_newOriginY)
{
        for (DrawingListener listener : listeners) {
                listener.originChanged(w_newOriginX, w_newOriginY);
        }
}

@Override
public void mouseDragged(MouseEvent e) {
        if (dragging) {
                int d_X = e.getX();
                int d_Y = e.getY();

                AffineTransform transform = new AffineTransform();
                transform.scale(scale, scale);
                transform.translate(-w_dragStartOriginX,
-w_dragStartOriginY);

                Point2D d_Pt = new Point2D.Double(d_X, d_Y);
                Point2D w_Pt = new Point2D.Double();
                try
                {
                        transform.inverseTransform(d_Pt, w_Pt);
                }
                catch (NoninvertibleTransformException ex) {
                        return;
                }
                int w_X = (int)w_Pt.getX();
                int w_Y = (int)w_Pt.getY();

                int w_deltaX = w_X - w_dragStartX;
                int w_deltaY = w_Y - w_dragStartY;

                w_originX = w_dragStartOriginX - w_deltaX;
                w_originY = w_dragStartOriginY - w_deltaY;

*****           notifyOriginChanged(w_originX, w_originY);

                repaint();
        }
}
```

4. Implement two listeners on the Drawing class (one for each window),
   add addDrawingListener method to DrawingFrame, and
   add a listener to each frame in Drawing

[Drawing]

```
private DrawingListener drawingListener1 = new DrawingListener() {

        @Override
        public void originChanged(int w_newOriginX, int w_newOriginY)
{
                frame2.setOrigin(w_newOriginX, w_newOriginY);
        }
};

private DrawingListener drawingListener2 = new DrawingListener() {

        @Override
        public void originChanged(int w_newOriginX, int w_newOriginY)
{
                frame1.setOrigin(w_newOriginX, w_newOriginY);
        }
};
```

[Drawing.run]

```
frame1.addDrawingListener(drawingListener1);
frame2.addDrawingListener(drawingListener2);
```

[DrawingFrame]

```
public void addDrawingListener(DrawingListener listener) {
        component.addDrawingListener(listener);
}
```

5. Add setOrigin methods to DrawingComponent and DrawingFrame

[DrawingComponent]

```
public void setOrigin(int w_newOriginX, int w_newOriginY) {
        w_originX = w_newOriginX;
        w_originY = w_newOriginY;
        this.repaint();
}
```

[DrawingFrame]

```
public void setOrigin(int w_newOriginX, int w_newOriginY) {
        component.setOrigin(w_newOriginX, w_newOriginY);
}
```