

```
package grafica;
```

```
import java.applet.*;  
import java.awt.geom.*;  
import java.awt.*;
```

```
public class Vizual2D {
```

```
    float XFm, XFM, YFm, YFM;
```

```
    int XPm, XPM, YPm, YPM;
```

```
    boolean tip_tran, decup;
```

```
    float sx, sy, tx, ty;
```

```
public Vizual2D()
```

```
{  
    super();  
    XPm = 0; YPm = 0;  
    XPM = 0; YPM = 0;  
    XFm = XFM = YFm = YFM = 0;  
    tip_tran = false; decup = false;  
    sx = sy = tx = ty = 0;  
    calcTran();  
}
```

```
public Vizual2D(float xf1, float yf1, float xf2, float yf2, int xp1, int yp1, int  
xp2, int yp2)
```

```
{  
    super();  
    XPm = xp1; YPm = yp1; XPM = xp2; YPM = yp2;  
    XFm = xf1; XFM = xf2; YFm = yf1; YFM = yf2;  
    tip_tran = false; decup = false;  
    calcTran();  
}
```

```
public void fereastra(float x1, float y1, float x2, float y2)
```

```
{  
    XFm = x1; XFM = x2; YFm = y1; YFM = y2;  
    calcTran();  
}
```

```
public void decupare(Graphics g, boolean dec)
```

```
{  
    if (dec)  
        g.setClip(XPm, YPm, XPM - XPm, YPM - YPm);  
    else  
        g.setClip(null);  
    decup = dec;  
}
```

```
public AffineTransform matrice()
```

```
{  
    AffineTransform t = new AffineTransform(sx, 0.f, 0.f, -sy, tx, -ty + YPm + YPM);  
    return t;  
}
```

```
public void poarta(int x1, int y1, int x2, int y2)
```

```
{  
    XPm = x1; XPM = x2; YPm = y1; YPM = y2;  
    calcTran();  
    if (decup)  
        g.setClip(x1, y1, x2 - x1, y2 - y1);  
}
```

```
public void cadruFereastra(Graphics g)
```

```
{  
    g.drawRect(xDisp(XFm), yDisp(YFm),  
               xDisp(XFM) - xDisp(XFm), yDisp(YFm) - yDisp(YFM));  
}
```

```
public void cadruPoarta(Graphics g)
```

```
{  
    g.drawRect(XPm, YPm, XPM - XPm, YPM - YPm);  
}
```

```
private void calcTran()
```

```
{  
    if (XFM > XFm && YFM > YFm)  
    {  
        sx = (XPM - XPm) / (XFM - XFm);  
        sy = (YPM - YPm) / (YFM - YFm);  
        if (tip_tran) //scalare uniforma  
            sx = sy = (sx < sy) ? sx : sy;  
        tx = XPm - sx * XFm + (XPM - XPm - sx * (XFM - XFm)) / 2;  
    }
```

```
    ty = YPm - sy * YFm + (YPM - YPm - sy * (YFM - YFm)) / 2;
}
else
    sx = sy = tx = ty = 0;
}
```

**public void tipTran(boolean tip)**

```
{
    if (tip != tip_tran)
    {
        tip_tran = tip;
        calcTran();
    }
}
```

**public int xDisp(float xf)**

```
{
    return (int) (xf * sx + tx);
}
```

**public float xLog(int xp)**

```
{
    return (xp - tx) / sx;
}
```

**public int yDisp(float yf)**

```
{
    return (int) (YPM + YPm - (yf * sy + ty));
}
```

**public float yLog(int yp)**

```
{
    return (YPM + YPm - yp - ty) / sy;
}
}
```