```java
package gobalkrishnan_v_18_06_1995.dimension3;

import gobalkrishnan_v_18_06_1995.dimension2.gki2Point;



public class gkiPerspective {
public double aspectratio,far,near,top,bottom,left,right,angle;

public void angle(double a){angle=a;}
public void aspectRatio(double ar){aspectratio=ar;}
public void far(double f){far=f;}
public void near(double n){near=n;}
public void top(double t){top=t;}
public void bottom(double b){bottom=b;}
public void left(double l){left=l;}
public void right(double r){right=r;}

gki3Point in,out;
gkiMatrix m=new gkiMatrix();
public void perspective(double a,double ar,double n,double f,double b,double t,double
l,double r) {

    angle=a;
    aspectratio=ar;
    near=n;
    far=f;
    right=r;
    left=l;
    top=t;
    bottom=b;



    double scale= Math.tan(Math.toRadians(a/2d))*near;



    right=aspectratio*scale;
    left=-right;
    top =scale;
    bottom=-top;

    p_frustum();
}

public void p_frustum() {


    double[][] m=this.m.m4;
    m[0][0]=2*near/(double)(right-left);
    m[0][1]=0;
    m[0][2]=0;
    m[0][3]=0;

    m[1][0]=0;
    m[1][1]=2*near/(double)(top-bottom);
    m[1][2]=0;
    m[1][3]=0;
```

```
        m[2][0]=(right+left)/(double)(right-left);
        m[2][1]=(top+bottom)/(double)(top-bottom);
        m[2][2]=-(far+near)/(double)(far-near);
        m[2][3]=-1;

        m[3][0]=0;
        m[3][1]=0;
        m[3][2]=-2*far*near/(double)(far-near);
        m[3][3]=0;

    }

    public gki3Point multpoint(gki3Point p) {
        gki3Point o=new gki3Point();
        o.x=p.x*m.m4[0][0]+p.y*m.m4[1][0]+p.z*m.m4[2][0]+ m.m4[3][0];
        o.y=p.x*m.m4[0][1]+p.y*m.m4[1][1]+p.z*m.m4[2][1]+ m.m4[3][1];
        o.z=p.x*m.m4[0][2]+p.y*m.m4[1][2]+p.z*m.m4[2][2]+ m.m4[3][2];
        double w=p.x*m.m4[0][3]+p.y*m.m4[1][3]+p.z*m.m4[2][3]+ m.m4[3][3];


        return o;
    }
    double x,y;
    public gki2Point screen2d(gki3Point p){
        gki3Point ps=multpoint(p);
        //System.out.println(ps);
    //  if(ps.z!=0 && Double.isFinite(ps.z)){
        x=ps.x/(double)ps.z;
        y=ps.y/(double)ps.z;
        //}
        return new gki2Point(x, y);

    }
    }
```