

```

package gobalkrishnan_v_18_06_1995.frame;

import gobalkrishnan_v_18_06_1995.color.gkiColor;

public class gkiPixel {

    public BufferedImage im;
    public int[] data;
    public int[] temp;
    Dimension d=Toolkit.getDefaultToolkit().getScreenSize();
    int maxWidth=(int)d.getWidth();
    int maxHeight=(int)d.getHeight();
    int width,height,viewWidth,viewHeight;
    int id;
    String name,subName;

    public void maximumSize(){
        setSize(d);
    }
    public void normalSize(){
        width=(int) (d.getWidth()/1.3d);
        height=(int) (d.getHeight()/1.3d);
        setSize(width, height);
    }
    public void setSize(int w,int h){
        viewWidth=w;
        viewHeight=h;
        im=new BufferedImage(viewWidth, viewHeight, BufferedImage.TYPE_INT_ARGB);
        data=new int[viewWidth*viewHeight];
        temp=new int[viewWidth*viewHeight];
        initialize();
    }
    public void setSize(Dimension d){
        viewWidth=(int) d.getWidth();
        viewHeight=(int) d.getHeight();
        im=new BufferedImage(viewWidth, viewHeight, BufferedImage.TYPE_INT_ARGB);
        data=new int[viewWidth*viewHeight];
        temp=new int[viewWidth*viewHeight];
        initialize();
    }

    public gkiPixel(){
        maximumSize();
    }
    public void setARGB(int i,int j,gkiColor c){
        data[i+j*viewWidth]=1;
        im.setRGB(i, j, new Color(c.argb,true).getRGB());
    }
    public gkiColor getARGB(int i,int j){
        gkiColor c=new gkiColor(im.getRGB(i, j));
        return c;
    }
}

```

```

public void initialBound(){
    for(int i=0;i<viewWidth;i++){
        for(int j=0;j<viewHeight;j++){
            if(im.getRGB(i, j)!=0){
                temp[i+j*viewWidth]=1;
            }
        }
    }
}

public void color(gkiColor g){
    for(int i=0;i<viewWidth;i++){
        for(int j=0;j<viewHeight;j++){
            im.setRGB(i, j, new Color(g.argb,true).getRGB());
        }
    }
}

public void tempBound(){
    for(int i=0;i<viewWidth;i++){
        for(int j=0;j<viewHeight;j++){
            if(data[i+j*viewWidth]!=0){
                temp[i+j*viewWidth]= data[i+j*viewWidth];
            }
        }
    }
}

public void tempBound(int[] data){
    for(int i=0;i<viewWidth;i++){
        for(int j=0;j<viewHeight;j++){
            temp[i+j*viewWidth]= data[i+j*viewWidth];
        }
    }
}

public boolean contain(int x,int y){
    for(int i=0;i<viewWidth;i++){
        for(int j=0;j<viewHeight;j++){
            int s=temp[i+j*viewWidth];
            if(s!=0){
                if((x==i) && (y==j)){
                    return true;
                }
            }
        }
    }
}

```

```
        return false;
    }
    public void initialize(){
        for(int i=0;i<viewWidth;i++){
            for(int j=0;j<viewHeight;j++){
                im.getRGB(i, j);
                data[i+j*viewWidth]=0;
                temp[i+j*viewWidth]=0;
                im.setRGB(i, j, new Color(0,true).getRGB());
            }
        }
    }

    public BufferedImage getImage(){
        return im;
    }
}
```