```
package gobalkrishnan_v_18_06_1995.color;

public class gkiRGB_to_HSL {
public int red,green,blue;
public double r_,g_,b_,hue,saturation,light;

private double c_255_to_1(int i){
    return i/255d;
}
private int c_1_to_255(double i){
    return (int) (i*255);
}
private double min(double a,double b,double c){
    if((a<b)&&(a<c)){
        return a;
    }else if(b<c){
        return b;
    }else{
        return c;
    }
}
private double max(double a,double b,double c){
    if((a>b)&&(a>c)){
        return a;
    }else if(b>c){
        return b;
    }else{
        return c;
    }
}

double max,min,delta;
public void setRGB(int r,int g,int b){
  setRed(r);
  setGreen(g);
  setBlue(b);
}
public void setRed(int red){
    this.red=red;
    process();
}
public void setGreen(int green){
    this.green=green;
    process();
}
public void setBlue(int blue){
    this.blue=blue;
```

```
    process();
}

private void process(){
    initialize();
    hue();
    light();
    saturation();
}

private void initialize(){
    r_=c_255_to_1(red);
    g_=c_255_to_1(green);
    b_=c_255_to_1(blue);
    max=max(r_,g_,b_);
    min=min(r_,g_,b_);
    delta=max-min;
}

private void hue(){
    if(delta==0){
        hue=0;
    }
    if(max==r_){
        hue=60*(((g_-b_)/(double)delta)%6);
    }
    if(max==g_){
        hue=60*(((b_-r_)/(double)delta)+2);
    }
    if(max==b_){
        hue=60*(((r_-g_)/(double)delta)+4);
    }

}
private void light(){
    light=(max+min)/2d;
}

private void saturation(){
    if(delta==0){
        saturation=0;
    }
    if((delta<0 || delta >0)){
        double mag=2*light-1;
        double val=Math.sqrt(mag*mag);
        saturation= delta/(double)(1-val);
    }
```

```
}

@Override
public String toString() {
    return "gkiRGB_to_HSL [hue=" + hue + ", saturation=" + saturation
            + ", light=" + light + "]";
}



}
```