```
package gobalkrishnan_v_18_06_1995.dimension2;

import gobalkrishnan_v_18_06_1995.color.gkiColor;


public class gki2Point {
public double x,y,angle;
public gkiColor color;

public void color(gkiColor c){color=c;}
public void x(double x){this.x=x;}
public void y(double y){this.y=y;}
public double x(){return x;}
public double y(){return y;}
public void set(double x,double y){
    x(x);
    y(y);
}
public void set(gki2Point p){
    x(p.x);
    y(p.y);
}
public void set(gki2Point p,gkiColor c){
    x(p.x);
    y(p.y);
   color(c);
}
public void set(int i,double gki){
```

```
    switch(i){
    case 0:x(gki);break;
    case 1:y(gki);break;
    }
}
public double get(int i){
    switch(i){
    case 0:return x;
    case 1:return y;
    default : return new Double(null);
    }
}

public gki2Point(){}
public gki2Point(double x,double y){
    set(x,y);
}
public gki2Point(gki2Point p){
    set(p);
}

public gki2Point(gki2Point p,
        gkiColor c) {
    set(p, c);
    // TODO Auto-generated constructor stub
}
public double magnitude(){
     double val= x*x + y*y;
```

```
     double magnitude=Math.sqrt(val);
     return magnitude;
}

public double dot(gki2Point p){
    return x*p.x+y*p.y;
}
public double dotAngle(gki2Point p){
    double a=magnitude();
    double b=p.magnitude();
    double dot=dot(p)/(double)(a*b);
    double inv=Math.acos(dot);
    return Math.toDegrees(inv);
}
public double dot(gki2Point c,gki2Point p){
    gki2Point u=new gki2Point(x-c.x,y-c.y);
    gki2Point v=new gki2Point(p.x-c.x,p.y-c.y);
    return (u.x*v.x)+(u.y*v.y);
}

public double dotAngle(gki2Point c,gki2Point p){
    gki2Point u=new gki2Point(x-c.x,y-c.y);
    gki2Point v=new gki2Point(p.x-c.x,p.y-c.y);
    double a=u.magnitude();
    double b=v.magnitude();
    double dot=(u.dot(v))/(double)(a*b);
    double inv=Math.acos(dot);
    return Math.toDegrees(inv);
```

```
}

public void add(gki2Point a,gki2Point b){
    x=a.x+b.x;
    y=a.y+b.y;
}
public void add(gki2Point b){
    add(this,b);
}
public gki2Point plus(gki2Point b){
    gki2Point p=new gki2Point(this);
    p.add(b);
    return p;
}
public void sub(gki2Point a,gki2Point b){
    x=a.x-b.x;
    y=a.y-b.y;
}
public void sub(gki2Point b){
    sub(this,b);
}
public gki2Point minus(gki2Point b){
    gki2Point p=new gki2Point(this);
    p.sub(b);
    return p;
}
public void scale(double g){
```

```
    x*=g;
    y*=g;
}
public void scaleX(double g){x*=g;}
public void scaleY(double g){y*=g;}
public void translateX(double x){this.x+=x;}
public void translateY(double y){this.y+=y;}
public void translate(double x,double y){
    this.x+=x;
    this.y+=y;
}
public void rotate(gki2Point c,double angle){
    gki2Point p=new gki2Point(x-c.x,y-c.y);

    this.angle=angle;
    double rad=Math.toRadians(angle);
    double cos=Math.cos(rad);
    double sin=Math.sin(rad);
    double x_=(p.x*cos)-(p.y*sin);
    double y_=(p.x*sin)+(p.y*cos);
    x=x_+c.x;
    y=y_+c.y;


    System.out.println((int)x+":"+(int)y);

}
```

```
public void shearX(double shx){
    double x_=this.x+ shx*this.y;
    double y_=this.y;
 x=x_;
 y=y_;
}
public void shearY(double shy){
    double x_=x;
    double y_=x*shy+y;
 x=x_;
 y=y_;
}


public gki2Point times(double g){
    gki2Point p=new gki2Point(this);
    p.scale(g);
    return p;
}

public void normalize(){
    double m=magnitude();
    if(m==0){return;}
    scale(1.0d/m);
}


public double distance(gki2Point a,gki2Point b){
```

```
    double x= a.x-b.x;
    double y= a.y-b.y;
    return Math.sqrt(x*x+y*y);
}

public gki2Point addScaled(gki2Point a,double g,gki2Point b){
    gki2Point o=new gki2Point();
    o.x=a.x+(g*b.x);
    o.y=a.y+(g*b.y);
    return o;
}

public double distance(gki2Point b){
    return distance(this, b);
}

public gki2Point addScaled(double g,gki2Point b){
    return addScaled(this,g, b);
}
public String toString(){
    return "("+x+","+y+")";
}
}
```