```java
package gobalkrishnan_v_18_06_1995.dimension3;

import gobalkrishnan_v_18_06_1995.color.gkiColor;


public class gki3Point {
public double x,y,z,rx,ry,rz,sx,sy,sz;

public void x(double x){this.x=x;}
public void y(double y){this.y=y;}
public void z(double z){this.z=z;}
public double x(){return x;}
public double y(){return y;}
public double z(){return z;}

public void set(double x,double y,double z){
    x(x);
    y(y);
    z(z);
}
public void set(gki3Point p){
    x(p.x);
    y(p.y);
    z(p.z);
}
public void set(int i,double gki){
    switch(i){
    case 0:x(gki);break;
    case 1:y(gki);break;
    case 2:z(gki);break;
    }
}

public double get(int i){
    switch(i){
    case 0:return x;
    case 1:return y;
    case 2:return z;
    default: return new Double(null);
    }
}

public gki3Point(){}
public gki3Point(double x,double y,double z){
    set(x,y,z);
}
public gki3Point(gki3Point gki){
    set(gki);
}

public void scale(double s){
    x*=s;
    y*=s;
    z*=s;
    sx=s;
    sy=s;
    sz=s;
}
```

```
public gki3Point mul(double s){
  gki3Point p=new gki3Point();
  p.set(this);
  p.scale(s);
  return p;
}

public void scaleX(double x){
    this.x*=x;
    sx=x;
}
public void scaleY(double y){this.y*=y;
   sy=y;
}
public void scaleZ(double z){this.z*=z;
   sz=z;
}
public void scale(double sx,double sy,double sz){
    x*=sx;
    y*=sy;
    z*=sz;
    this.sx=sx;
    this.sy=sy;
    this.sz=sz;
}
public void scale(gki3Point p){
    x*=p.x;
    y*=p.y;
    z*=p.z;
    this.sx=p.x;
    this.sy=p.y;
    this.sz=p.z;
}



public double magnitude(){
    double val=x*x + y*y +z*z;
    double mag=Math.sqrt(val);
    return mag;
}
public void normalize(){
    double len=magnitude();
    if(len==0){return;}
    scale(1.0d/len);
}
public gki3Point times(double d){
    gki3Point p=new gki3Point(this);
    p.scale(d);
    return p;
}
public void add(gki3Point a,gki3Point b){
    x=a.x+b.x;
    y=a.y+b.y;
    z=a.z+b.z;
}
public void add(gki3Point b){
    add(this,b);
}
```

```
public gki3Point plus(gki3Point b){
    gki3Point plus=new gki3Point(this);
    plus.add(b);
    return plus;
}

public void sub(gki3Point a,gki3Point b){
    x=a.x-b.x;
    y=a.y-b.y;
    z=a.z-b.z;
}
public void sub(gki3Point b){
    sub(this,b);
}
public gki3Point minus(gki3Point b){
    gki3Point p=new gki3Point(this);
    p.sub(b);
    return p;
}

public double dot(gki3Point p){
    return x*p.x+y*p.y+z*p.z;
}
public double dotAngle(gki3Point 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(gki3Point c,gki3Point p){
    gki3Point u=new gki3Point(x-c.x,y-c.y,z-c.z);
    gki3Point v=new gki3Point(p.x-c.x,p.y-c.y,p.z-c.z);
    return (u.x*v.x)+(u.y*v.y)+(u.z*v.z);
}

public double dotAngle(gki3Point c,gki3Point p){
    gki3Point u=new gki3Point(x-c.x,y-c.y,z-c.z);
    gki3Point v=new gki3Point(p.x-c.x,p.y-c.y,p.z-c.z);
    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 gki3Point cross(gki3Point p){
    gki3Point cross=new  gki3Point();
    cross.x((y*p.z)-(z*p.y));
    cross.y((z*p.x)-(x*p.z));
    cross.z((x*p.y)-(y*p.x));

    return cross;
}
public gki3Point cross(gki3Point c,gki3Point p){
    gki3Point u=new gki3Point(x-c.x,y-c.y,z-c.z);
    gki3Point v=new gki3Point(p.x-c.x,p.y-c.y,p.z-c.z);
```

```
    gki3Point cross=new  gki3Point();
    cross.x((u.y*v.z)-(u.z*v.y));
    cross.y((u.z*v.x)-(u.x*v.z));
    cross.z((u.x*v.y)-(u.y*v.x));
    return cross;


}
double angle;
public void rotateX(gki3Point c,double a){
    angle=a;
    gki3Point p=new gki3Point(x,y-c.y,z-c.z);
    double ag=Math.toRadians(angle);
    double cos=Math.cos(ag);
    double sin=Math.sin(ag);
    double y_=(p.y*cos)-(p.z*sin);
    double z_=(p.y*sin)+(p.z*cos);
    y=y_+c.y;
    z=z_+c.z;
}
public void rotateY(gki3Point c,double a){
    angle=a;
    gki3Point p=new gki3Point(x-c.x,y-c.z,z-c.z);
    double ag=Math.toRadians(angle);
    double cos=Math.cos(ag);
    double sin=Math.sin(ag);
    double x_=(p.z*sin)+(p.x*cos);
    double z_=(p.z*cos)-(p.x*sin);
    double y_=(p.y);
    x=x_+c.x;
    z=z_+c.z;
    y=y_+c.y;


}

public void rotateZ(gki3Point c,double a){
    angle=a;
    gki3Point p=new gki3Point(x-c.x,y-c.y,z);
    double ag=Math.toRadians(angle);
    double cos=Math.cos(ag);
    double sin=Math.sin(ag);
    double x_=(p.x*cos)-(p.y*sin);
    double y_=(p.x*sin)+(p.y*cos);
    x=x_+c.x;
    y=y_+c.y;
}

public String toString(){
    return "("+x+","+y+","+z+")";
}
public String get(){
    return x+","+y+","+z;
}

public gkiColor color;
public void color(gkiColor col) {
    // TODO Auto-generated method stub
```

```
    this.color=col;
}

}
```