

```
public class Triangle extends GeometricObject{
    double side1;
    double side2;
    double side3;
    //constructor
    Triangle(){
    }
    Triangle(double side1,double side2,double side3){
        this.side1=side1;
        this.side2=side2;
        this.side3=side3;
        if(side1<0||side2<0||side3<0){
            System.out.println("Error: side length cannot be negative");
            System.exit(0);
        }
        if(!(side1+side2>side3 && side1+side3>side2 && side2+side3>side1)){
            System.out.println("Error: these 3 sides can't form triangle");
            System.exit(0);
        }
    }
    Triangle(double side1,double side2,double side3,String color,boolean filled){
        super(color, filled);
        this.side1=side1;
        this.side2=side2;
        this.side3=side3;
        if(side1<0||side2<0||side3<0){
            System.out.println("Error: side length cannot be negative");
            System.exit(0);
        }
        if(!(side1+side2>side3 && side1+side3>side2 && side2+side3>side1)){
            System.out.println("Error: these 3 sides can't form triangle");
            System.exit(0);
        }
    }
    //accessor
    public double getSide1(){return side1;}
    public double getSide2(){return side2;}
    public double getSide3(){return side3;}
    //mutator
    public void setSide1(double side1){this.side1=side1;}
    public void setSide2(double side2){this.side2=side2;}
    public void setSide3(double side3){this.side3=side3;}
    //method
    public double getArea(){
        double s=(side1+side2+side3)/2;
        return Math.sqrt(s*(s-side1)*(s-side2)*(s-side3));
    }
    public double getPerimeter(){
        return side1+side2+side3;
    }
    public String toString(){
        return "side1 = "+side1+" side2 = "+side2+" side3 = "+side3+"\n"+super.toString();
    }
}
```