

```
import java.util.*;
```

```
public class App {  
    static Scanner scanner = new Scanner(System.in);  
    static Scanner scannerBool = new Scanner(System.in);  
    static Scanner scannerStr = new Scanner(System.in);  
    public static void main(String[] args) throws Exception {  
        System.out.print("Enter side 1: ");  
        double sideA = scanner.nextDouble();  
        System.out.print("Enter side 2: ");  
        double sideB = scanner.nextDouble();  
        System.out.print("Enter side 3: ");  
        double sideC = scanner.nextDouble();  
        System.out.print("Enter color: ");  
        String color = scannerStr.nextLine();  
        System.out.print("Enter filled: ");  
        boolean filled = scanner.nextBoolean();  
        System.out.println();  
        GeometricObject triangle = new Triangle(sideA,sideB,sideC,color,filled);  
        System.out.println(triangle);  
    }  
}
```

```
import java.util.Date;
```

```
abstract class GeometricObject{  
    private String color;  
    private boolean filled;  
    private Date dateCreated;  
  
    protected GeometricObject(){  
        this.color = "Black";  
        this.filled = false;  
        this.dateCreated = new Date();  
    }  
  
    protected GeometricObject(String color,boolean filled){  
        this.color = color;  
        this.filled = filled;  
        this.dateCreated = new Date();  
    }  
}
```

```

//setter

public void setColor(String color){
    this.color = color;
}

public void setFilled(){
    this.filled = filled;
}

//getter

public String getColor(){
    return color;
}

public boolean isFilled(){
    return filled;
}

public Date getDate(){
    return dateCreated;
}

//abstract method
abstract public double getArea();
abstract public double getPerimeter();

@Override
public String toString(){
    return "Date Created: "+dateCreated+"\nColor: "+color+"\nFilled: "+filled;
}
}

```

```

class Triangle extends GeometricObject{
    private double sideA,sideB,sideC;
    private double perimeter,area;

    public Triangle(){
        this.sideA = 1;
        this.sideB = 1;
    }
}

```

```

        this.sideC = 1;
    }

    public Triangle(double sideA,double sideB,double sideC,String color,boolean filled){
        super(color,filled);
        this.sideA = sideA;
        this.sideB = sideB;
        this.sideC = sideC;
    }

    @Override
    public double getArea(){
        double s = (sideA + sideB + sideC) / 2;
        return Math.sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));
    }

    @Override
    public double getPerimeter(){
        return sideA + sideB + sideC;
    }

    public String toString(){
        return super.toString() + "Triangle: side1 = " + sideA +
            " side2 = " + sideB + " side3 = " + sideC + "\nPerimeter: " +
            String.format("%.2f",getPerimeter())+
            "\nArea: "+String.format("%.2f",getArea());
    }
}

```