import java.util.Scanner;

public class Exercise13\_01 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter three sides of the triangle: ");

double side1 = input.nextDouble();

double side2 = input.nextDouble();

double side3 = input.nextDouble();

System.out.print("Enter a color: ");

String color = input.next();

System.out.print("Is the triangle filled (true / false)? ");

boolean filled = input.nextBoolean();

Triangle triangle = new Triangle(side1, side2, side3, color, filled);

System.out.println(triangle);

}

}

abstract class GeometricObject {

private String color = "white"; // Corrected typo here

private boolean filled;

private java.util.Date dateCreated;

protected GeometricObject() {

dateCreated = new java.util.Date();

}

protected GeometricObject(String color, boolean filled) {

dateCreated = new java.util.Date();

this.color = color;

this.filled = filled;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public boolean isFilled() {

return filled;

}

public void setFilled(boolean filled) {

this.filled = filled;

}

public java.util.Date getDateCreated() {

return dateCreated;

}

@Override

public String toString() {

return "created on " + dateCreated + "\ncolor: " + color +

" and filled: " + filled;

}

public abstract double getArea();

public abstract double getPerimeter();

}

class Triangle extends GeometricObject {

private double side1;

private double side2;

private double side3;

public Triangle() {

}

public Triangle(double side1, double side2, double side3) {

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

public Triangle(double side1, double side2, double side3,

String color, boolean filled) {

super(color, filled); // Call the superclass constructor first

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

public double getSide1() {

return side1;

}

public void setSide1(double side1) {

this.side1 = side1;

}

public double getSide2() {

return side2;

}

public void setSide2(double side2) {

this.side2 = side2;

}

public double getSide3() {

return side3;

}

public void setSide3(double side3) {

this.side3 = side3;

}

@Override

public double getArea() {

double s = (side1 + side2 + side3) / 2;

return Math.sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

}

@Override

public double getPerimeter() {

return side1 + side2 + side3;

}

@Override

public String toString() {

return super.toString() + "\nArea: " + getArea() +

"\nPerimeter: " + getPerimeter();

}

}