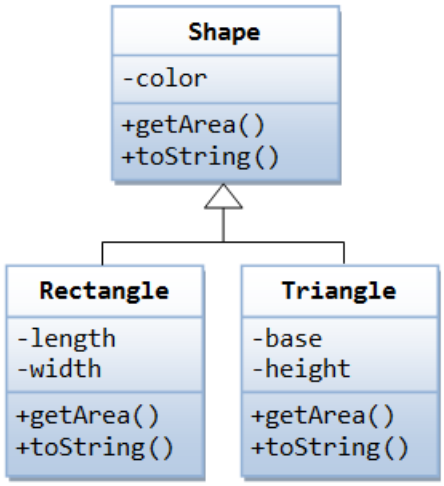


Shape.java

```
// Define superclass Shape
public class Shape {
  // Private member variable
  private String color;
    // Constructor
  public Shape (String color) {
    this.color = color;
  }
   @Override
  public String toString() {
    return "Shape of color=\"" + color + "\"";
  }
    // All shapes must has a method called getArea()
  public double getArea() {
    System.err.println("Shape unknown! Cannot compute area!");
    return 0;   // Need a return to compile the program
  }
}
```

Rectangle.java

```
// Define Rectangle, subclass of Shape
public class Rectangle extends Shape {
  // Private member variables
  private int length;
  private int width;

  // Constructor
  public Rectangle(String color, int length, int width) {
    super(color);
    this.length = length;
    this.width = width;
  }
    @Override
  public String toString() {
    return "Rectangle of length=" + length + " and width=" + width + ", subclass of " + super.toString();
  }
    @Override
  public double getArea() {
    return length*width;
  }
}
```

Triangle.java

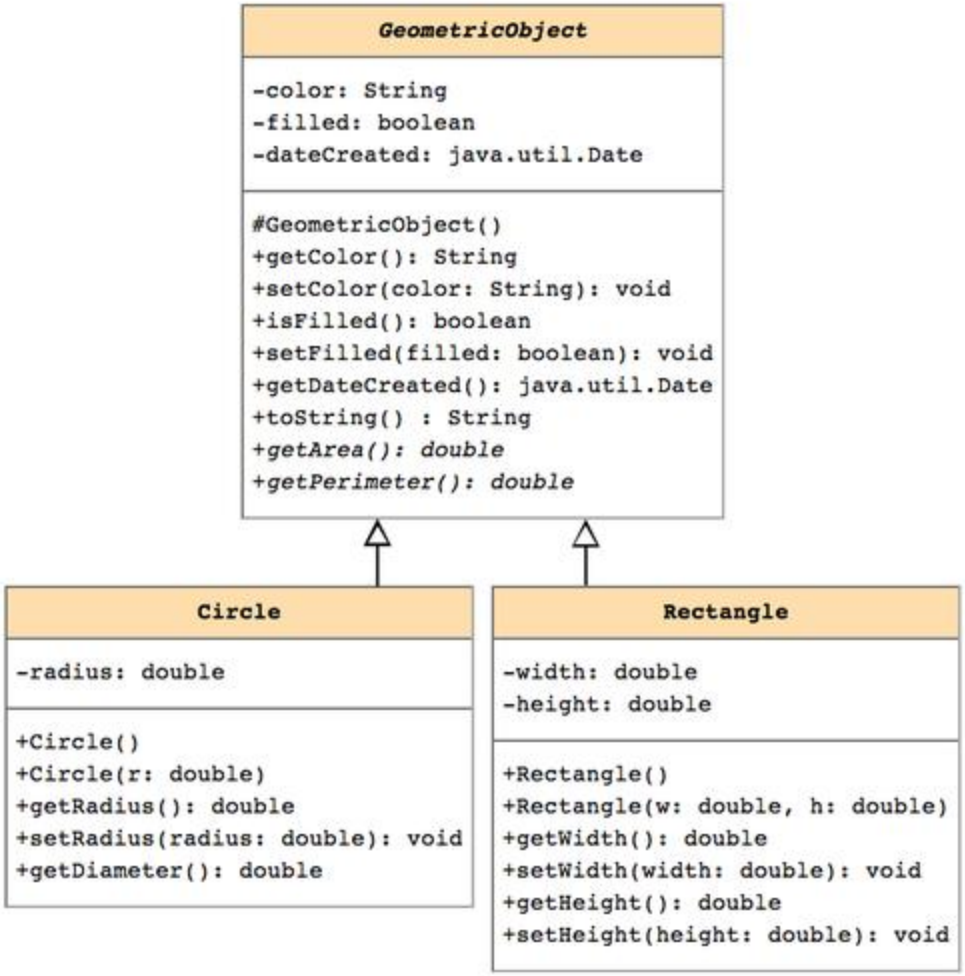
```
// Define Triangle, subclass of Shape
public class Triangle extends Shape {
  // Private member variables
  private int base;
```

```
  private int height;

  // Constructor
  public Triangle(String color, int base, int height) {
    super(color);
    this.base = base;
    this.height = height;
  }
    @Override
  public String toString() {
    return "Triangle of base=" + base + " and height=" + height + ", subclass of " + super.toString();
  }
    @Override
  public double getArea() {
    return 0.5*base*height;
  }
}
```

TestShape.java

In our application, we could create references of Shape, and assigned them instances of subclasses, as follows:

```
// A test driver program for Shape and its subclasses
public class TestShape {
  public static void main(String[] args) {
    Shape s1 = new Rectangle("red", 4, 5);
    System.out.println(s1);
    System.out.println("Area is " + s1.getArea());

    Shape s2 = new Triangle("blue", 4, 5);
    System.out.println(s2);
    System.out.println("Area is " + s2.getArea());
  }
}
```



Sample Class Diagram