

```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Main {
5     public static void main(String[] args) {
6         List<Figura> figuras = new ArrayList<>();
7
8         figuras.add(new Circulo(5.0, "Rojo"));
9         figuras.add(new Circulo(2.5, "Azul"));
10
11         figuras.add(new Rectangulo(4.0, 3.0, "Verde"));
12         figuras.add(new Rectangulo(10.0, 2.0, "Amarillo"));
13
14         figuras.add(new Triangulo(3.0, 4.0, 5.0, "Negro"));
15         figuras.add(new Triangulo(6.0, 7.0, 8.0, "Blanco"));
16
17         for (Figura f : figuras) {
18             f.mostrarInformacion();
19         }
20     }
21 }
```

Run Main x

```
"C:\Users\Santiago Lozano\jdk\openjdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=65310" -Dfile.encoding=UTF-8
=== CIRCULO ===
Color: Rojo
Radio: 5.0
Área: 78,5398
Perímetro (circunferencia): 31,4159
=== CIRCULO ===
Color: Azul
```

Este programa en Java gestiona diferentes figuras geométricas usando **POO**, específicamente **herencia y polimorfismo**.

- Usa una lista (ArrayList) para almacenar figuras de distintos tipos: **círculos**, **rectángulos** y **triángulos**, todas heredando de la clase abstracta Figura.
- Se crean varios objetos con sus atributos (como radio, color, base, altura, lados, etc.) y se agregan a la lista.
- Mediante un **bucle for-each**, se recorre la lista y, gracias al polimorfismo, cada figura ejecuta su propio método `mostrarInformacion()`.
- Cada clase calcula y muestra su **área** y **perímetro** según sus fórmulas.

En resumen: el código permite manejar distintos tipos de figuras en una sola estructura, mostrando su información sin necesidad de saber de qué tipo es cada figura. Esto lo hace flexible y fácil de extender.

The screenshot shows an IDE window with the file `Triangulo.java` open. The code defines a `Triangulo` class with three private double attributes (`lado1`, `lado2`, `lado3`) and one private String attribute (`color`). It includes a constructor that initializes these attributes and a `calcularArea()` method that uses Heron's formula to calculate the area. The `Run` tab at the bottom shows the execution output, which includes the command used to run the program and the resulting information for two triangles: one with sides 3.0, 4.0, 5.0 and color negro, and another with sides 5.0, 5.0, 5.0 and color blanco.

```
// Clase Triángulo
public class Triangulo {
    private double lado1;
    private double lado2;
    private double lado3;
    private String color;

    // Constructor
    public Triangulo(double lado1, double lado2, double lado3, String color) {
        this.lado1 = lado1;
        this.lado2 = lado2;
        this.lado3 = lado3;
        this.color = color;
    }

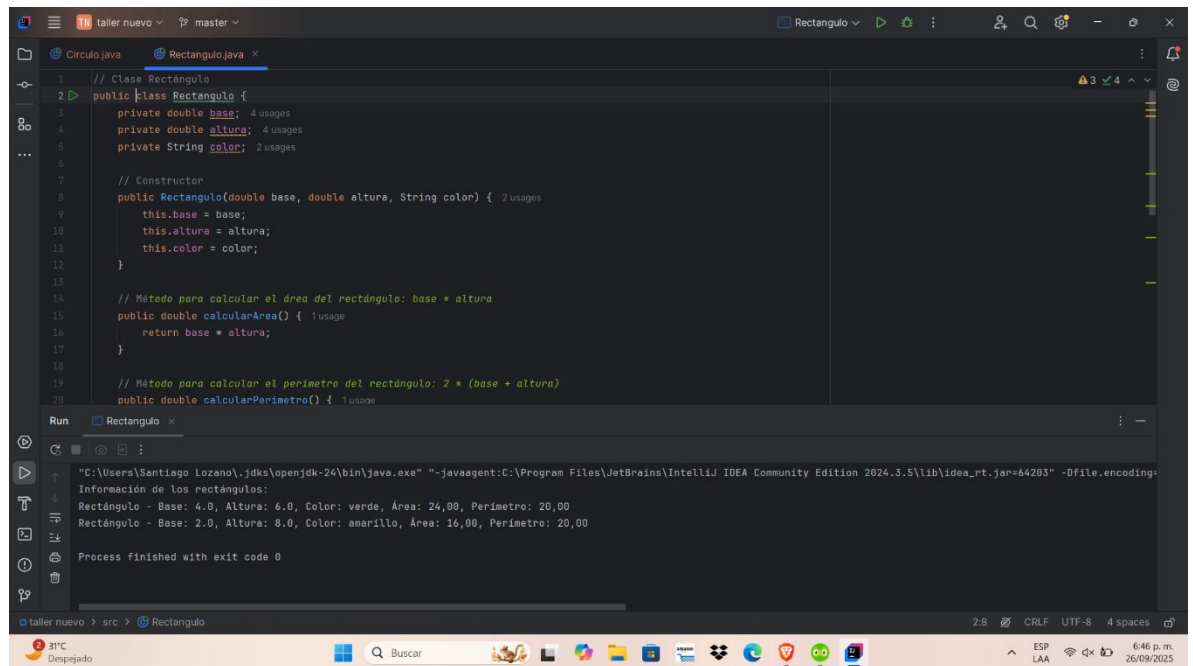
    // Método para calcular el área del triángulo usando la fórmula de Herón
    public double calcularArea() {
        double s = (lado1 + lado2 + lado3) / 2.0; // Semiperímetro
        return Math.sqrt(s * (s - lado1) * (s - lado2) * (s - lado3));
    }
}
```

Run Triangulo x

```
"C:\Users\Santiago Lozano\jdk8\openjdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=64259" -Dfile.encoding=UTF-8
Información de los triángulos:
Triángulo - Lados: 3.0, 4.0, 5.0, Color: negro, Área: 6,00, Perímetro: 12,00
Triángulo - Lados: 5.0, 5.0, 5.0, Color: blanco, Área: 10,83, Perímetro: 15,00
Process finished with exit code 0
```

La clase **Triangulo** representa triángulos guardando sus lados y color. Permite calcular su **área** usando la **fórmula de Herón** y mostrar datos como lados, color, área y perímetro.

En el ejemplo, se crean triángulos y se muestran sus características en consola, facilitando el manejo y cálculo de información de varias figuras en un programa



```
1 // Clase Rectángulo
2 public class Rectangulo {
3     private double base; 4 usages
4     private double altura; 4 usages
5     private String color; 2 usages
6
7     // Constructor
8     public Rectangulo(double base, double altura, String color) { 2 usages
9         this.base = base;
10        this.altura = altura;
11        this.color = color;
12    }
13
14    // Método para calcular el área del rectángulo: base * altura
15    public double calcularArea() { 1 usage
16        return base * altura;
17    }
18
19    // Método para calcular el perímetro del rectángulo: 2 * (base + altura)
20    public double calcularPerimetro() { 1 usage
```

Run Rectangulo

"C:\Users\Santiago Lozano\.jdk\openjdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=64203" -Dfile.encoding: UTF-8

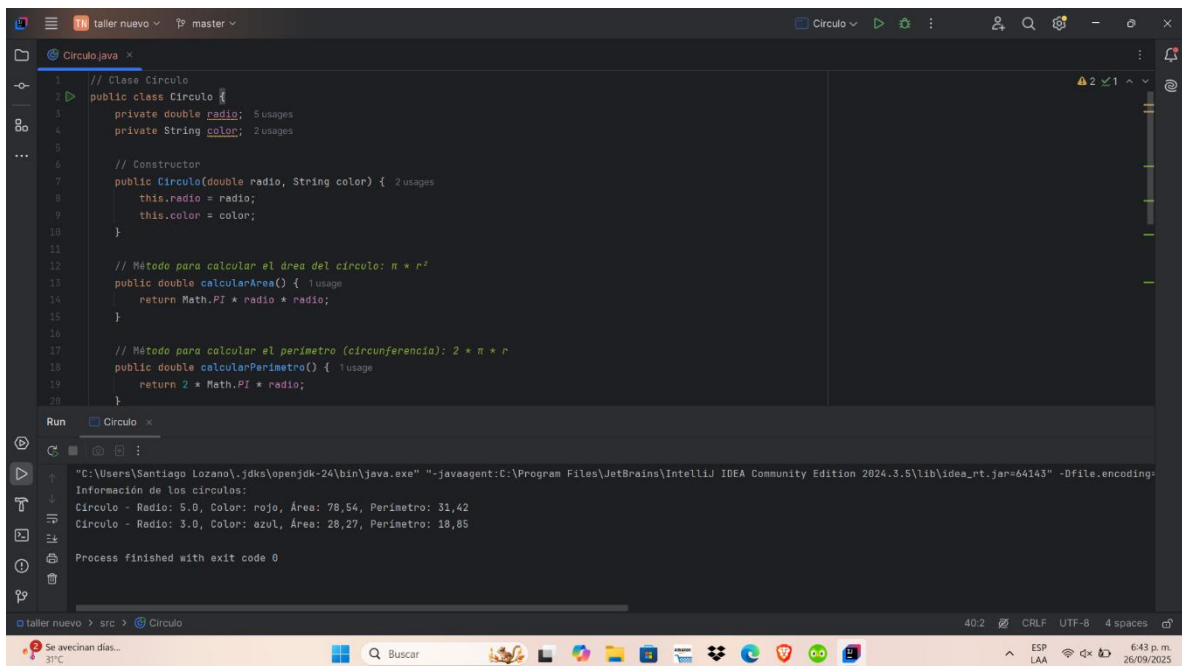
Información de los rectángulos:

Rectángulo - Base: 4.0, Altura: 6.0, Color: verde, Área: 24,00, Perímetro: 20,00

Rectángulo - Base: 2.0, Altura: 8.0, Color: amarillo, Área: 16,00, Perímetro: 20,00

Process finished with exit code 0

La clase **Rectangulo** permite crear rectángulos con base, altura y color. Incluye métodos para **calcular el área** ($\text{base} \times \text{altura}$) y **el perímetro** ($2 \times (\text{base} + \text{altura})$). Además, muestra la información de cada rectángulo, como sus dimensiones, color y resultados de los cálculos, facilitando su gestión dentro de un programa.



```
1 // Clase Circulo
2 public class Circulo {
3     private double radio; 5 usages
4     private String color; 2 usages
5
6     // Constructor
7     public Circulo(double radio, String color) { 2 usages
8         this.radio = radio;
9         this.color = color;
10    }
11
12    // Método para calcular el área del círculo:  $\pi \times r^2$ 
13    public double calcularArea() { 1 usage
14        return Math.PI * radio * radio;
15    }
16
17    // Método para calcular el perímetro (circunferencia):  $2 \times \pi \times r$ 
18    public double calcularPerimetro() { 1 usage
19        return 2 * Math.PI * radio;
20    }
21 }
```

Run Circulo

```
"C:\Users\Santiago Lozano\jdk\openjdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=64143" -Dfile.encoding=UTF-8
Información de los círculos:
Círculo - Radio: 5.0, Color: rojo, Área: 78.54, Perímetro: 31.42
Círculo - Radio: 3.0, Color: azul, Área: 28.27, Perímetro: 18.85
Process finished with exit code 0
```

La clase **Circulo** representa un círculo con dos atributos: **radio** y **color**.

- Tiene un **constructor** para inicializar estos valores.
- Incluye un método para **calcular el área** usando la fórmula $\pi \times r^2$.
- También un método para **calcular el perímetro** (circunferencia) con la fórmula $2 \times \pi \times r$.
- En la salida de consola, muestra la información de cada círculo: su radio, color, área y perímetro.

Ejemplo mostrado:

Círculo - Radio: 5.0, Color: rojo, Área: 78.54, Perímetro: 31.42

Círculo - Radio: 3.0, Color: azul, Área: 28.27, Perímetro: 18.85

En resumen: permite crear círculos, calcular sus medidas y mostrar sus datos de forma sencilla.