



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



Unidad de académica: Análisis y diseño de algoritmos

Actividad: “POO métodos privados”

Equipo:

- Elizalde Hernández Alan
- Reyes Ruíz Yoselyn Estefany
- Solares Velasco Arturo Misael
- Solís Lugo Mayra
- Toral Hernández Leonardo Javier

Grupo: 3CV1

Profesor: García Floriano Andrés

Fecha:

30 de abril 2024

# Reporte de Práctica: Clases de Figuras Geométricas en Python y Java

En esta práctica, creamos clases para representar diferentes figuras geométricas tanto en Python como en Java. Luego, creamos instancias de estas clases y calculamos áreas, perímetros y colores de las figuras.

## Clases en Python

En Python, creamos las siguientes clases para representar figuras geométricas:

1. **Hexágono Regular (HexagonoRegular)**: Esta clase modela un hexágono regular y proporciona métodos para calcular su área, perímetro y obtener su color.

**Código de la clase (HexagonoRegular):**

```
3 class HexagonoRegular:
4     def __init__(self, lado, color):
5         self.__lado = lado
6         self.__color = color
7
8     def __area(self):
9         return (3 * math.sqrt(3) * self.__lado ** 2) / 2
10
11    def __perimetro(self):
12        return 6 * self.__lado
13
14    def getArea(self):
15        return self.__area()
16
17    def getPerimetro(self):
18        return self.__perimetro()
19
20    def getColor(self):
21        return self.__color
```

2. **Rombo (Rombo)**: Representa un rombo con sus diagonales y color asociado. Ofrece métodos para calcular área, perímetro y obtener el color del rombo.

**Código de la clase (Rombo)**

```

23 class Rombo:
24     def __init__(self, diagonal_mayor, diagonal_menor, color):
25         self.__diagonal_mayor = diagonal_mayor
26         self.__diagonal_menor = diagonal_menor
27         self.__color = color
28
29     def __area(self):
30         return (self.__diagonal_mayor * self.__diagonal_menor) / 2
31
32     def __perimetro(self):
33         return 4 * math.sqrt((self.__diagonal_mayor / 2) ** 2 + (self.__diagonal_menor / 2) ** 2)
34
35     def getArea(self):
36         return self.__area()
37
38     def getPerimetro(self):
39         return self.__perimetro()
40
41     def getColor(self):
42         return self.__color

```

3. **Trapezio (Trapezio):** Modela un trapezio con sus bases, altura y color. Tiene métodos para calcular área, perímetro y obtener el color del trapezio.

#### Código de la clase (Trapezio)

```

class Trapezio:
    def __init__(self, base_mayor, base_menor, altura, color):
        self.__base_mayor = base_mayor
        self.__base_menor = base_menor
        self.__altura = altura
        self.__color = color

    def __area(self):
        return ((self.__base_mayor + self.__base_menor) * self.__altura) / 2

    def __perimetro(self):
        lado_oblicuo = math.sqrt(((self.__base_mayor - self.__base_menor) / 2) ** 2 + self.__altura ** 2)
        return self.__base_mayor + self.__base_menor + 2 * lado_oblicuo

    def getArea(self):
        return self.__area()

    def getPerimetro(self):
        return self.__perimetro()

    def getColor(self):
        return self.__color

```

## Clases en Java

En Java, definimos las siguientes clases para figuras geométricas:

1. **Círculo (Circulo):** Esta clase representa un círculo con radio y color asociado. Ofrece métodos para calcular su área, perímetro y obtener el color del círculo.

## Código de la clase (Circulo)

```
Practica 6 > C:\Circulo.java
1 public class Circulo {
2     private String color;
3     private double radio;
4
5     public Circulo(String color, double radio) {
6         this.color = color;
7         this.radio = radio;
8     }
9
10    public String getColor() {
11        return color;
12    }
13
14    public double getArea() {
15        return area();
16    }
17
18    public double getPerimetro() {
19        return perimetro();
20    }
21
22    private double area() {
23        return Math.PI * radio * radio;
24    }
25
26    private double perimetro() {
27        return 2 * Math.PI * radio;
28    }
29 }
```

2. **Cuadrado (Cuadrado):** Modela un cuadrado con su longitud de lado y color. Proporciona métodos para calcular área, perímetro y obtener el color del cuadrado.

### Código de la clase (Cuadrado)

```
Practica 6 > Cuadrado.java
1  public class Cuadrado{
2      private String color;
3      private double lado;
4
5      public Cuadrado(String color, double lado) {
6          this.color = color;
7          this.lado = lado;
8      }
9
10     public String getColor() {
11         return color;
12     }
13
14     public double getArea() {
15         return area();
16     }
17
18     public double getPerimetro() {
19         return perimetro();
20     }
21
22     private double area() {
23         return lado * lado;
24     }
25
26     private double perimetro() {
27         return 4 * lado;
28     }
29 }
```

3. **Rectángulo (Rectangulo):** Representa un rectángulo con sus dimensiones de base y altura, así como su color. Tiene métodos para calcular área, perímetro y obtener el color del rectángulo.

### Código de la clase (Rectangulo)

```

Practica 6 > Rectangulo.java
1  public class Rectangulo {
2      private String color;
3      private double base;
4      private double altura;
5
6      public Rectangulo(String color, double base, double altura) {
7          this.color = color;
8          this.base = base;
9          this.altura = altura;
10     }
11
12     public String getColor() {
13         return color;
14     }
15
16     public double getArea() {
17         return area();
18     }
19
20     public double getPerimetro() {
21         return perimetro();
22     }
23
24     private double area() {
25         return base * altura;
26     }
27
28     private double perimetro() {
29         return 2 * (base + altura);
30     }
31 }

```

4. **Triángulo (Triangulo)**: Esta clase modela un triángulo con sus lados y color asociado. Ofrece métodos para calcular su área, perímetro y obtener el color del triángulo.

**Código de la clase (Triangulo)**

```

Triangulo.java x
Practica 6 > Triangulo.java
1  public class Triangulo {
2      private String color;
3      private double lado1;
4      private double lado2;
5      private double lado3;
6
7      public Triangulo(String color, double lado1, double lado2, double lado3) {
8          this.color = color;
9          this.lado1 = lado1;
10         this.lado2 = lado2;
11         this.lado3 = lado3;
12     }
13
14     public String getColor() {
15         return color;
16     }
17
18     public double getArea() {
19         return area();
20     }
21
22     public double getPerimetro() {
23         return perimetro();
24     }
25
26     private double area() {
27         double s = (lado1 + lado2 + lado3) / 2;
28         return Math.sqrt(s * (s - lado1) * (s - lado2) * (s - lado3));
29     }
30
31     private double perimetro() {
32         return lado1 + lado2 + lado3;
33     }
34 }

```

## Uso en el Programa Principal

En el programa principal, instanciamos objetos de cada clase de figura geométrica tanto en Python como en Java. Luego, utilizamos los métodos proporcionados por cada clase para obtener información sobre las figuras, como el área, el perímetro y el color.

Ambas implementaciones demuestran el uso de la programación orientada a objetos para modelar figuras geométricas y realizar cálculos relacionados con ellas de manera modular y eficiente. Además, muestran cómo los principios de encapsulación, abstracción, métodos y atributos se aplican en la práctica para crear un diseño de software limpio y mantenible.

## Resultados de la ejecución de los métodos de cada clase en Python

```
Resultados de las figuras geométricas como objetos

Área del hexágono: 64.9519052838329
Perímetro del hexágono: 30
Color del hexágono: azul

Área del rombo: 12.0
Perímetro del rombo: 14.422205101855956
Color del rombo: verde

Área del trapecio: 26.0
Perímetro del trapecio: 21.544003745317532
Color del trapecio: rojo
```

## Resultados de la ejecución de los métodos de cada clase en Java

```
penjdk-amd64/bin/java -cp /home/arturo/.config/Code/User/workspaces/redhat.java/jdt_ws/paradigmas-escom-2024_dd138db8/bin main
Area del circulo: 706.8583470577034
Perimetro del circulo: 94.24777960769379
Color del circulo: 255, 255, 255

Area del rectangulo: 3600.0
Perimetro del rectangulo: 260.0
Color del rectangulo: Verde

Area del cuadrado: 1369.0
Perimetro del cuadrado: 148.0
Color del cuadrado: Rojo

Area del triangulo: 6.0
Perimetro del triangulo: 12.0
Color del triangulo: Azul
```

En esta práctica, creamos clases para representar diferentes figuras geométricas tanto en Python como en Java. Luego, creamos instancias de estas clases y calculamos áreas, perímetros y colores de las figuras.

## Conclusiones

La implementación en Python utiliza clases con métodos privados y públicos para encapsular la funcionalidad y los atributos de las figuras geométricas.



La implementación en Java también sigue los principios de encapsulación y abstracción, pero utiliza métodos públicos para acceder a los cálculos de área y perímetro.

Ambas implementaciones demuestran la utilidad de la programación orientada a objetos para modelar figuras geométricas de manera modular y reutilizable.

El código en Python y Java produce resultados consistentes y precisos para los cálculos de área y perímetro de las figuras geométricas.

Este proyecto ilustra cómo los conceptos de OOP, como la encapsulación, la abstracción, los métodos y los atributos, se aplican en la práctica para crear un diseño de software limpio y mantenible.