



ASIGNATURA:

- **PROGRAMACION ORIENTADA A OBJETOS**

NOMBRE DE LA TAREA IV:

- **CÓDIGO DE LA ESTRUCTURA DE HERENCIA DE CLASES.**

NOMBRE DEL CATEDRATICO:

- **MAE. ARNOL RAFAEL GUTIERREZ**

NOMBRE Y NÚMERO DE CUENTA:

- **JOSUÉ ALEJANDRO SANDOVAL GALINDO
(201630010142)**

FECHA DE ENTREGA:

- **27/02/2025**

INTRODUCCIÓN

A continuación, Explicare como este programa es un ejemplo de herencia en Java. Crea diferentes formas geométricas (círculo, cuadrado, triángulo y línea) y calcula sus áreas. También dibuja las formas, pero en este caso, solo muestra un mensaje en la consola porque no estamos usando los gráficos todavía.

ESTRUCTURA DEL PROGRAMA

Superclase Formas

- Esta es la clase principal de la que heredan las demás.
- Tiene dos atributos:
 - Color: El color de la forma.
 - Métodos:
 - Dibujar(): Muestra un mensaje genérico ("Dibujando una forma genérica").
 - Calculararea(): Muestra un mensaje genérico ("Calculando área de una forma genérica").

```
public class Formas {  
  
    private String color;  
  
    public Formas(String color) {  
  
        this.color = color;  
  
    }  
  
    public void establecerColor(String color) {  
  
        this.color = color;  
  
    }  
  
    public String obtenerColor() {  
  
        return color;  
  
    }  
  
    public void dibujar() {  
  
        System.out.println("Dibujando una forma genérica");  
  
    }  
  
    public void calcularArea() {  
  
        System.out.println("Calculando área de una forma genérica");  
  
    }  
  
}
```

```
}
```

Subclases (Circulo, Cuadrado, Triangulo, Linea)

- Cada una de estas clases hereda de la superclase Formas.
- Sobrescriben los métodos dibujar() y calculararea() para hacer algo específico para cada forma.

Clase Circulo

- Tiene un atributo adicional: radio.
- Sobrescribe los métodos:
 - Dibujar(): Muestra "Dibujando un círculo".
 - Calculararea(): Calcula el área del círculo usando la fórmula $\pi * \text{radio}$

```
public class Circulo extends Formas {  
  
    private double radio;  
  
    public Circulo(String color, double radio) {  
  
        super(color);  
  
        this.radio = radio;  
  
    }  
  
    @Override  
  
    public void dibujar() {  
  
        System.out.println("Dibujando un círculo");  
  
    }  
  
    @Override  
  
    public void calcularArea() {  
  
        double area = Math.PI * radio * radio;  
  
        System.out.println("El área del círculo es: " + area);  
  
    }  
  
}
```

Clase Cuadrado

- Tiene un atributo adicional: lado.
- Sobrescribe los métodos:
 - Dibujar(): Muestra "Dibujando un cuadrado".
 - Calculararea(): Calcula el área del cuadrado usando la fórmula $\text{lado} * \text{lado}$.

```
public class Cuadrado extends Formas {  
  
    private double lado;  
  
    public Cuadrado(String color, double lado) {  
  
        super(color);  
  
        this.lado = lado;  
  
    }  
  
    @Override  
  
    public void dibujar() {  
  
        System.out.println("Dibujando un cuadrado");  
  
    }  
  
    @Override  
  
    public void calcularArea() {  
  
        double area = lado * lado;  
  
        System.out.println("El área del cuadrado es: " + area);  
  
    }  
}
```

Clase Triangulo

- Tiene dos atributos adicionales: base y altura.
- Sobrescribe los métodos:
 - Dibujar(): Muestra "Dibujando un triángulo".
 - Calculararea(): Calcula el área del triángulo usando la fórmula $(base * altura) / 2$.

```
public class Triangulo extends Formas {  
  
    private double base;  
  
    private double altura;  
  
    public Triangulo(String color, double base, double altura) {  
  
        super(color);  
  
        this.base = base;  
  
        this.altura = altura;  
  
    }  
  
    @Override  
  
    public void dibujar() {  
  
        System.out.println("Dibujando un triángulo");  
  
    }  
  
    @Override  
  
    public void calcularArea() {  
  
        double area = (base * altura) / 2;  
  
        System.out.println("El área del triángulo es: " + area);  
  
    }  
  
}
```

Clase Linea

- La clase Linea representa una línea en un plano.
- Una línea es una forma geométrica que tiene una longitud, pero no tiene área en este caso.

```
public class Linea extends Formas {  
  
    private double longitud;  
  
    public Linea(String color, double longitud) {  
  
        super(color);  
  
        this.longitud = longitud;  
  
    }  
  
    @Override  
  
    public void dibujar() {  
  
        System.out.println("Dibujando una línea");  
  
    }  
  
    @Override  
  
    public void calcularArea() {  
  
        System.out.println("Una línea no tiene área");  
  
    }  
  
}
```

Estructura de la clase Main

- El método main es el punto de entrada del programa en Java.
- Aquí se ejecuta todo el código del programa.

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Pedir datos para el círculo
```

```
        System.out.print("Ingrese el color del círculo: ");
```

```
        String colorCirculo = scanner.nextLine();
```

```
        System.out.print("Ingrese el radio del círculo: ");
```

```
        double radio = scanner.nextDouble();
```

```
        Circulo circulo = new Circulo(colorCirculo, radio);
```

```
        // Pedir datos para el cuadrado
```

```
        scanner.nextLine(); // Limpiar el buffer
```

```
        System.out.print("Ingrese el color del cuadrado: ");
```

```
        String colorCuadrado = scanner.nextLine();
```

```
        System.out.print("Ingrese el lado del cuadrado: ");
```

```
        double lado = scanner.nextDouble();
```

```
        Cuadrado cuadrado = new Cuadrado(colorCuadrado, lado);
```

```
        // Pedir datos para el triángulo
```

```
        scanner.nextLine(); // Limpiar el buffer
```

```
        System.out.print("Ingrese el color del triángulo: ");
```

```
        String colorTriangulo = scanner.nextLine();
```



```
System.out.print("Ingrese la base del triángulo: ");

double base = scanner.nextDouble();

System.out.print("Ingrese la altura del triángulo: ");

double altura = scanner.nextDouble();

Triangulo triangulo = new Triangulo(colorTriangulo, base, altura);

// Pedir datos para la línea

scanner.nextLine(); // Limpiar el buffer

System.out.print("Ingrese el color de la línea: ");

String colorLinea = scanner.nextLine();

System.out.print("Ingrese la longitud de la línea: ");

double longitud = scanner.nextDouble();

Linea linea = new Linea(colorLinea, longitud);

// Mostrar resultados

System.out.println("\n--- Resultados ---");

circulo.dibujar();

circulo.calcularArea();

cuadrado.dibujar();

cuadrado.calcularArea();

triangulo.dibujar();

triangulo.calcularArea();

linea.dibujar();

linea.calcularArea();

scanner.close();
```

}

}