

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE  
SEDE SANTO DOMINGO**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS**

**CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**PERIODO** : Mayo 2023 – Septiembre 2023

**ASIGNATURA** : Programación Orientada a Objetos

**TEMA** : Herencia cálculo de áreas y perímetros.

**NOMBRES** : Molina Luna Miguel Angel

**NIVEL-PARALELO** : Segundo Semestre - A

**DOCENTE** : Ing. Javier Cevallos.

**FECHA DE ENTREGA** : 17/05/2023

**SANTO DOMINGO - ECUADOR**

**2023**

## Índice

<b>1. Introducción:</b>	3
<b>2. Sistemas de Objetivos</b>	3
Objetivo General:	3
Objetivos Específicos:	3
<b>3. Desarrollo:</b>	3
<b>4. Conclusiones:</b>	12
<b>5. Recomendaciones:</b>	12
<b>6. Bibliografía/ Referencias:</b>	13
<b>7. Anexos</b>	14

## **1. Introducción:**

Moya-2014 describe a la herencia como uno de los 4 pilares de la programación orientada a objetos (POO) junto con la Abstracción, Encapsulación y Polimorfismo. La herencia es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente, permite compartir automáticamente métodos y datos entre clases, subclases y objetos. El presente informe desarrolla un programa en lenguaje java en el cual calculamos áreas y perímetros de 4 figuras geométricas y en donde se puso en práctica la herramienta de herencia entre clases. El informe describirá la estructura del programa y proporcionará ejemplos de su funcionamiento.

## **2. Sistemas de Objetivos**

### **Objetivo General:**

Presentar un programa en Java que permita calcular áreas y perímetros de figuras geométricas utilizando conceptos de herencia y constructores.

### **Objetivos Específicos:**

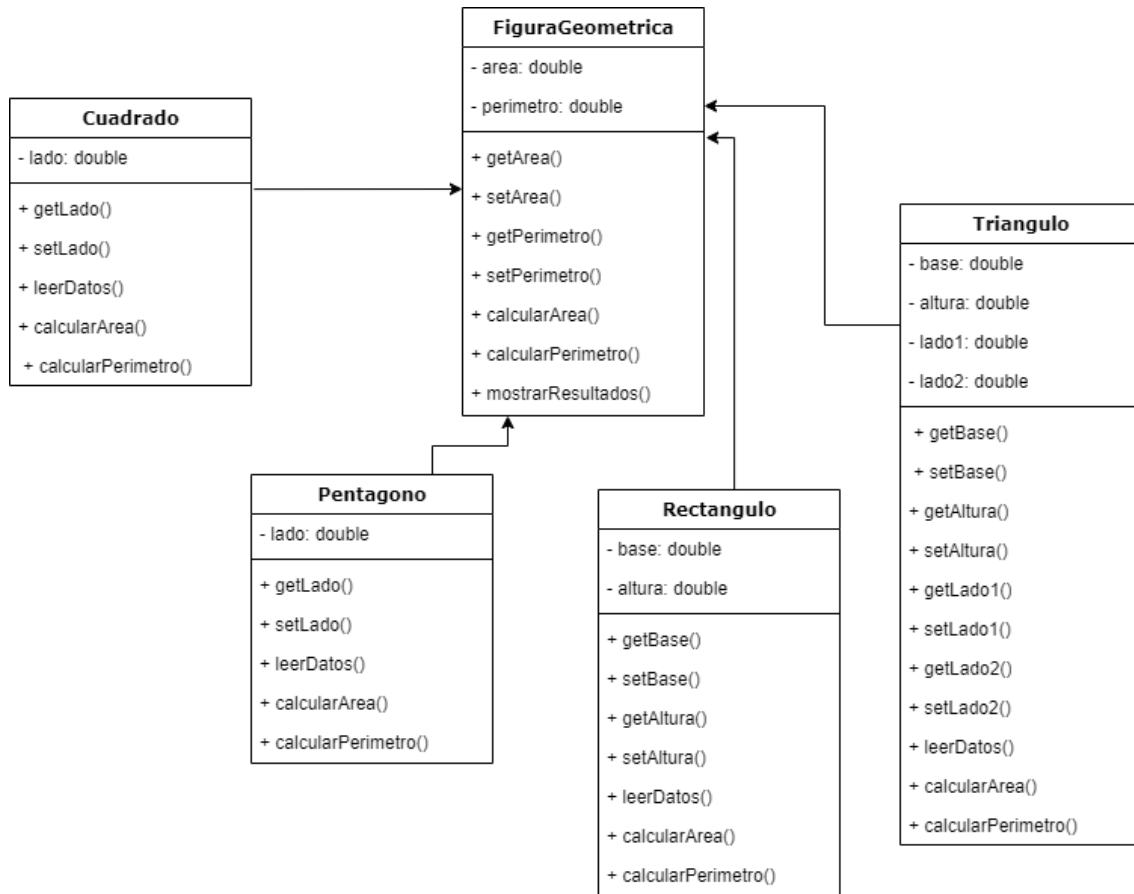
- a. Presentar la implementación de las clases Cuadrado, Pentágono, Rectángulo y Triangulo, destacando su relación con la clase base FiguraGeometrica.
- b. explicar cómo se utilizan los conceptos de herencia y constructores dentro del proyecto de figuras geométricas.

## **3. Desarrollo:**

Teniendo la siguiente definición de constructor: es un tipo específico de método que siempre tiene el mismo nombre que la clase y se utiliza para construir objetos de esa clase. No tiene tipo de dato específico de retorno, ni siquiera void. Esto se debe a que el tipo específico que debe devolver un constructor de clase es el propio tipo de la clase.

Para el desarrollo de la práctica se establecen las clases FiguraGeometrica, Cuadrado, Pentagono, Rectangulo y Triangulo, donde las figuras comparten los atributos de área y perímetro.

Con el siguiente diagrama de clases se demuestra la relación de herencia:



### Código en java de la clase padre (FiguraGeometrica):

```

package areaspractica;
class FiguraGeometrica {
    protected double area;
    protected double perimetro;

    public FiguraGeometrica() {
        area = 0.0;
        perimetro = 0.0;
    }

    public double getArea() {

```

```

        return area;
    }

    public void setArea(double area) {
        this.area = area;
    }

    public double getPerimetro() {
        return perimetro;
    }

    public void setPerimetro(double perimetro) {
        this.perimetro = perimetro;
    }

    public void calcularArea() {
        // Implementar cálculo del área segun la figura
    }

    public void calcularPerimetro() {
        // Implementa cálculo del perímetro segun la figura
    }

    public void mostrarResultados() {
        System.out.println("Area: " + area);
        System.out.println("Perimetro: " + perimetro);
    }
}

```

### **Código en java de la clase hija Cuadrado:**

```

package areaspractica;
import java.util.Scanner;
class Cuadrado extends FiguraGeometrica {
    private double lado;

    public Cuadrado() {
        lado = 0.0;
    }

    public void setLado(double lado) {
        this.lado = lado;
    }

    public double getLado() {
        return lado;
    }

    public void leerDatos() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Ingresa el valor del lado: ");
    }
}

```

```

        lado = sc.nextDouble();
    }

    @Override
    public void calcularArea() {
        setArea(lado * lado);
    }

    @Override
    public void calcularPerimetro() {
        setPerimetro(4 * lado);
    }
}

```

### **Código en java de la clase hija Pantagono:**

```

package areaspractica;
import java.util.Scanner;
class Pantagono extends FiguraGeometrica {
    private double lado;

    public Pantagono() {
        lado = 0.0;
    }

    public void setLado(double lado) {
        this.lado = lado;
    }

    public double getLado() {
        return lado;
    }

    public void leerDatos() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Ingrese el valor del lado: ");
        lado = sc.nextDouble();
    }

    @Override
    public void calcularArea() {
        setArea((5 * lado * lado) / (4 * Math.tan(Math.PI / 5)));
    }

    @Override
    public void calcularPerimetro() {
        setPerimetro(5 * lado);
    }
}

```

### **Codigo en java de la clase hija Rectangulo:**

```
package areaspractica;

import java.util.Scanner;

class Rectangulo extends FiguraGeometrica {
    private double base;
    private double altura;

    public Rectangulo() {
        base = 0.0;
        altura = 0.0;
    }

    public void setBase(double base) {
        this.base = base;
    }

    public double getBase() {
        return base;
    }

    public void setAltura(double altura) {
        this.altura = altura;
    }

    public double getAltura() {
        return altura;
    }

    public void leerDatos() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Ingrese el valor de la base: ");
        base = sc.nextDouble();
        System.out.println("Ingrese el valor de la altura: ");
        altura = sc.nextDouble();
    }

    @Override
    public void calcularArea() {
        setArea(base * altura);
    }

    @Override
    public void calcularPerimetro() {
        setPerimetro(2 * (base + altura));
    }
}
```

### **Código en java de la clase hija Triangulo:**

```
package areaspractica;

import java.util.Scanner;

class Triangulo extends FiguraGeometrica {
    private double base;
    private double altura;
    private double lado1;
    private double lado2;

    public Triangulo() {
        base = 0.0;
        altura = 0.0;
        lado1 = 0.0;
        lado2 = 0.0;
    }

    public void setBase(double base) {
        this.base = base;
    }

    public double getBase() {
        return base;
    }

    public void setAltura(double altura) {
        this.altura = altura;
    }

    public double getAltura() {
        return altura;
    }

    public void setLado1(double lado1) {
        this.lado1 = lado1;
    }

    public double getLado1() {
        return lado1;
    }

    public void setLado2(double lado2) {
        this.lado2 = lado2;
    }

    public double getLado2() {
        return lado2;
    }
}
```



```

public void leerDatos() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Ingrese el valor de la base (lado): ");
    base = sc.nextDouble();
    System.out.println("Ingrese el valor de la altura: ");
    altura = sc.nextDouble();
    System.out.println("Ingrese el valor del lado 1: ");
    lado1 = sc.nextDouble();
    System.out.println("Ingrese el valor del lado 2: ");
    lado2 = sc.nextDouble();
}

@Override
public void calcularArea() {
    setArea((base * altura) / 2);
}

@Override
public void calcularPerimetro() {
    setPerimetro(base + lado1 + lado2);
}
}

```

**Código en java de la clase principal que realiza el menú y los llamados a clases y generación de objetos adecuados:**

```

package areaspractica;

import java.util.Scanner;

public class AreasPractica {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int opcion=0;
        while (opcion!=5){
            // Mostrar el menú de opciones
            System.out.println("1. Cuadrado");
            System.out.println("2. Pentagono");
            System.out.println("3. Rectangulo");
            System.out.println("4. Triangulo");
            System.out.println("5. Salir");
            System.out.println("Seleccione una opcion:");

            opcion = sc.nextInt();

            // Realizar acciones según la opción seleccionada
            switch (opcion) {
                case 1:

```

```

        Cuadrado cuadrado = new Cuadrado();
        cuadrado.leerDatos();
        cuadrado.calcularArea();
        cuadrado.calcularPerimetro();
        cuadrado.mostrarResultados();
        break;
    case 2:
        Pentagono pentagono = new Pentagono();
        pentagono.leerDatos();
        pentagono.calcularArea();
        pentagono.calcularPerimetro();
        pentagono.mostrarResultados();
        break;
    case 3:
        Rectangulo rectangulo = new Rectangulo();
        rectangulo.leerDatos();
        rectangulo.calcularArea();
        rectangulo.calcularPerimetro();
        rectangulo.mostrarResultados();
        break;
    case 4:
        Triangulo triangulo = new Triangulo();
        triangulo.leerDatos();
        triangulo.calcularArea();
        triangulo.calcularPerimetro();
        triangulo.mostrarResultados();
        break;
    case 5:
        System.out.println("Ha salido del programa...");
        break;
    default:
        System.out.println("Opcion invalida");
        break;
    }
}
sc.close();
}
}

```

### Prueba de que el código funciona adecuadamente:

```
compile:
run:
1. Cuadrado
2. Pentagono
3. Rectangulo
4. Triangulo
5. Salir
Seleccione una opcion:
1
Ingrese el valor del lado:
5
Area: 25.0
Perimetro: 20.0
```

Captura 1. Comprobación opción cuadrado.

```
1. Cuadrado
2. Pentagono
3. Rectangulo
4. Triangulo
5. Salir
Seleccione una opcion:
2
Ingrese el valor del lado:
8
Area: 110.11055363769388
Perimetro: 40.0
```

Captura 2. Comprobación opción Pentagono.

```
1. Cuadrado
2. Pentagono
3. Rectangulo
4. Triangulo
5. Salir
Seleccione una opcion:
3
Ingrese el valor de la base:
6
Ingrese el valor de la altura:
7,5
Area: 45.0
Perimetro: 27.0
1. Cuadrado
2. Pentagono
3. Rectangulo
4. Triangulo
5. Salir
Seleccione una opcion:
|
```

Captura 3. Comprobación opción Rectangulo.

```

-----
1. Cuadrado
2. Pentagono
3. Rectangulo
4. Triangulo
5. Salir
Seleccione una opcion:
4
Ingrese el valor de la base (lado):
4
Ingrese el valor de la altura:
3
Ingrese el valor del lado 1:
1
Ingrese el valor del lado 2:
2
Area: 6.0
Perimetro: 7.0

```

Captura 3. Comprobación opción Triangulo.

#### 4. Conclusiones:

La utilización de herencia ha facilitado la reutilización de código al establecer una relación jerárquica entre las clases. Esto ha evitado la duplicación de atributos y métodos comunes, permitiendo una estructura más organizada y mantenible.

Los constructores han sido utilizados para inicializar los objetos de las clases, permitiendo la asignación de valores iniciales a los atributos de manera más eficiente. Esto ha simplificado el proceso de creación de objetos y ha asegurado la consistencia de los datos.

#### 5. Recomendaciones:

Ampliar el programa agregando más clases de figuras geométricas, como círculos, elipses o trapecios. Esto permitirá calcular áreas y perímetros de una mayor variedad de figuras.

Explorar la posibilidad de añadir funcionalidades adicionales al programa, como calcular el perímetro diagonal de un rectángulo o determinar si un triángulo es equilátero, isósceles o escaleno. Esto agregará valor y versatilidad al programa.

## **6. Bibliografía/ Referencias:**

Caules, C. Á. (2021, agosto 17). Java Constructor y buenas prácticas. Arquitectura Java. <https://www.arquitecturajava.com/java-constructor-y-buenas-practicas/>

Tutorial de Java - constructor. (s/f). Dis.um.es. Recuperado el 18 de junio de 2023, de <http://dis.um.es/~bmoros/Tutorial/parte5/cap5-6.html>

Caules, C. Á. (2021, agosto 17). Java Constructor y buenas prácticas. Arquitectura Java. <https://www.arquitecturajava.com/java-constructor-y-buenas-practicas/>

## 7. Anexos (OPCIONAL)

### Clase FiguraGeometrica:

```
1 package areaspractica;
2 class FiguraGeometrica {
3     protected double area;
4     protected double perimetro;
5
6     public FiguraGeometrica() {
7         area = 0.0;
8         perimetro = 0.0;
9     }
10
11     public double getArea() {
12         return area;
13     }
14
15     public void setArea(double area) {
16         this.area = area;
17     }
18
19     public double getPerimetro() {
20         return perimetro;
21     }
22
23     public void setPerimetro(double perimetro) {
24         this.perimetro = perimetro;
25     }
26
27     public void calcularArea() {
28         // Implementar cálculo del área según la figura
29     }
30
31     public void calcularPerimetro() {
32         // Implementa cálculo del perímetro según la figura
33     }
34
35     public void mostrarResultados() {
36         System.out.println("Área: " + area);
37         System.out.println("Perímetro: " + perimetro);
38     }
39 }
```

### Clase Cuadrado:

```
1 package areaspractica;
2 import java.util.Scanner;
3 class Cuadrado extends FiguraGeometrica {
4     private double lado;
5
6     public Cuadrado() {
7         lado = 0.0;
8     }
9
10    public void setLado(double lado) {
11        this.lado = lado;
12    }
13
14    public double getLado() {
15        return lado;
16    }
17
18    public void leerDatos() {
19        Scanner sc = new Scanner(System.in);
20        System.out.println("Ingrese el valor del lado: ");
21        lado = sc.nextDouble();
22    }
23
24    @Override
25    public void calcularArea() {
26        setArea(lado * lado);
27    }
28
29    @Override
30    public void calcularPerimetro() {
31        setPerimetro(4 * lado);
32    }
33 }
```

## Clase Pentagono:

```
1 package areaspractica;
2 import java.util.Scanner;
3 class Pentagono extends FiguraGeometrica {
4     private double lado;
5
6     public Pentagono() {
7         lado = 0.0;
8     }
9
10    public void setLado(double lado) {
11        this.lado = lado;
12    }
13
14    public double getLado() {
15        return lado;
16    }
17
18    public void leerDatos() {
19        Scanner sc = new Scanner(System.in);
20        System.out.println("Ingrese el valor del lado: ");
21        lado = sc.nextDouble();
22    }
23
24    @Override
25    public void calcularArea() {
26        setArea((5 * lado * lado) / (4 * Math.tan(Math.PI / 5)));
27    }
28
29    @Override
30    public void calcularPerimetro() {
31        setPerimetro(5 * lado);
32    }
33 }
```



## Clase Rectangulo:

```
1  package areaspractica;
2  import java.util.Scanner;
3  class Rectangulo extends FiguraGeometrica {
4      private double base;
5      private double altura;
6      public Rectangulo() {
7          base = 0.0;
8          altura = 0.0;
9      }
10     public void setBase(double base) {
11         this.base = base;
12     }
13     public double getBase() {
14         return base;
15     }
16     public void setAltura(double altura) {
17         this.altura = altura;
18     }
19     public double getAltura() {
20         return altura;
21     }
22     public void leerDatos() {
23         Scanner sc = new Scanner(System.in);
24         System.out.println("Ingrese el valor de la base: ");
25         base = sc.nextDouble();
26         System.out.println("Ingrese el valor de la altura: ");
27         altura = sc.nextDouble();
28     }
29     @Override
30     public void calcularArea() {
31         setArea(base * altura);
32     }
33     @Override
34     public void calcularPerimetro() {
35         setPerimetro(2 * (base + altura));
36     }
37 }
```

### Clase Triangulo:

```
1 package areaspractica;
2 import java.util.Scanner;
3 class Triangulo extends FiguraGeometrica {
4     private double base;
5     private double altura;
6     private double lado1;
7     private double lado2;
8     public Triangulo() {
9         base = 0.0;
10        altura = 0.0;
11        lado1 = 0.0;
12        lado2 = 0.0;
13    }
14    public void setBase(double base) {
15        this.base = base;
16    }
17    public double getBase() {
18        return base;
19    }
20    public void setAltura(double altura) {
21        this.altura = altura;
22    }
23    public double getAltura() {
24        return altura;
25    }
26    public void setLado1(double lado1) {
27        this.lado1 = lado1;
28    }
29    public double getLado1() {
```

```

30         return lado1;
31     }
32     public void setLado2(double lado2) {
33         this.lado2 = lado2;
34     }
35     public double getLado2() {
36         return lado2;
37     }
38     public void leerDatos() {
39         Scanner sc = new Scanner(System.in);
40         System.out.println("Ingrese el valor de la base (lado): ");
41         base = sc.nextDouble();
42         System.out.println("Ingrese el valor de la altura: ");
43         altura = sc.nextDouble();
44         System.out.println("Ingrese el valor del lado 1: ");
45         lado1 = sc.nextDouble();
46         System.out.println("Ingrese el valor del lado 2: ");
47         lado2 = sc.nextDouble();
48     }
49     @Override
50     public void calcularArea() {
51         setArea((base * altura) / 2);
52     }
53     @Override
54     public void calcularPerimetro() {
55         setPerimetro(base + lado1 + lado2);
56     }

```

### Clase Main o principal:

```
1 package areaspractica;
2 import java.util.Scanner;
3 public class AreasPractica {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int opcion=0;
7         while (opcion!=5){
8             // Mostrar el menú de opciones
9             System.out.println("1. Cuadrado");
10            System.out.println("2. Pentagono");
11            System.out.println("3. Rectangulo");
12            System.out.println("4. Triangulo");
13            System.out.println("5. Salir");
14            System.out.println("Seleccione una opcion:");
15            opcion = sc.nextInt();
16            // Realizar acciones según la opción seleccionada
17            switch (opcion) {
18                case 1:
19                    Cuadrado cuadrado = new Cuadrado();
20                    cuadrado.leerDatos();
21                    cuadrado.calcularArea();
22                    cuadrado.calcularPerimetro();
23                    cuadrado.mostrarResultados();
24                    break;
25                case 2:
26                    Pentagono pentagono = new Pentagono();
27                    pentagono.leerDatos();
28                    pentagono.calcularArea();
29                    pentagono.calcularPerimetro();
```

```

30         pentagono.mostrarResultados();
31         break;
32     case 3:
33         Rectangulo rectangulo = new Rectangulo();
34         rectangulo.leerDatos();
35         rectangulo.calcularArea();
36         rectangulo.calcularPerimetro();
37         rectangulo.mostrarResultados();
38         break;
39     case 4:
40         Triangulo triangulo = new Triangulo();
41         triangulo.leerDatos();
42         triangulo.calcularArea();
43         triangulo.calcularPerimetro();
44         triangulo.mostrarResultados();
45         break;
46     case 5:
47         System.out.println(x: "Ha salido del programa...");
48         break;
49     default:
50         System.out.println(x: "Opcion invalida");
51         break;
52     }
53 }
54 sc.close();
55 }
56 }

```