

Ejercicio 4: Figuras geométricas con herencia y validación (25 puntos)

Modela un sistema de figuras geométricas utilizando herencia y cálculo de propiedades.

Requisitos:

1. Clase base `Figura` con métodos abstractos:
 - `calcular_area()`
 - `calcular_perimetro()`
2. Subclases:
 - `Rectangulo(base, altura)`
 - `Circulo(radio)`
 - `Triangulo(lado1, lado2, lado3)` que verifique si los lados forman un triángulo válido.
3. Implementa los métodos correspondientes para cada figura.
4. Crea una lista con al menos 2 figuras de cada tipo. Muestra su área y perímetro.

```
from abc import ABC, abstractmethod
import math
```

```
class Figura(ABC):
    @abstractmethod
    def calcular_area(self):
        pass

    @abstractmethod
    def calcular_perimetro(self):
        pass

class Rectangulo(Figura):
    def __init__(self, base, altura):
        self.base = base
        self.altura = altura

    def calcular_area(self):
        return self.base * self.altura

    def calcular_perimetro(self):
        return 2 * (self.base + self.altura)
```

```

class Circulo(Figura):
    def __init__(self, radio):
        self.radio = radio

    def calcular_area(self):
        return math.pi * self.radio ** 2

    def calcular_perimetro(self):
        return 2 * math.pi * self.radio

class Triangulo(Figura):
    def __init__(self, lado1, lado2, lado3):
        if (lado1 + lado2 > lado3 and
            lado1 + lado3 > lado2 and
            lado2 + lado3 > lado1):
            self.lado1 = lado1
            self.lado2 = lado2
            self.lado3 = lado3
        else:
            raise ValueError("Lados no forman un triángulo válido")

    def calcular_area(self):
        s = (self.lado1 + self.lado2 + self.lado3) / 2
        return math.sqrt(s * (s - self.lado1) * (s - self.lado2) * (s - self.lado3))

    def calcular_perimetro(self):
        return self.lado1 + self.lado2 + self.lado3

figuras = [
    Rectangulo(4, 5),
    Rectangulo(2, 3),
    Circulo(3),
    Circulo(5),
    Triangulo(3, 4, 5),
    Triangulo(6, 7, 8)
]

for figura in figuras:
    print(f'{type(figura).__name__}: Área = {figura.calcular_area():.2f}, Perímetro = {figura.calcular_perimetro():.2f}')

```