

Práctico 5 -POO

Ejercicio 1

Crear una clase **Guerrero** que sea reciba como atributos:

- Nombre del guerrero.
- Vida del guerrero.
- Ataque del guerrero.
- Arma.

Como métodos deberá tener:

- `__str__`.
- Atacar.
- Mejorar arma. (Deberá mejorar la categoría del arma)

El atributo **Arma** será otra **clase** que tendrá los siguientes atributos:

- Tipo de arma.
- Ataque del arma.
- Categoría del arma.

y la clase Arma tendrá los siguientes métodos:

- `__str__`.
- Mejorar categoria. (Deberá aumentar el daño del arma)

Una vez creadas ambas clases, deberá crear dos guerreros, uno será “John Snow” y el otro será un “caminante blanco” y deberá hacer que “John Snow” ataque al caminante blanco.

Ejercicio 2

Crear la siguiente jerarquía de clases para representar diferentes figuras geométricas utilizando herencia, clases abstractas y polimorfismo.

1. **Clase Abstracta:** Define una clase abstracta llamada **FiguraGeometrica** que contenga los siguientes métodos abstractos:
 - `area(self)`: devuelve el área de la figura.
 - `perimetro(self)`: devuelve el perímetro de la figura.

2. Clases Derivadas:

- Clase **Circulo**: Deriva de FiguraGeometrica e implementa los métodos area y perimetro. El constructor debe recibir el radio del círculo.
 - Clase **Rectangulo**: Deriva de FiguraGeometrica e implementa los métodos area y perimetro. El constructor debe recibir la base y la altura del rectángulo.
 - Clase **Triangulo**: Deriva de FiguraGeometrica e implementa los métodos area y perimetro. El constructor debe recibir los tres lados del triángulo.
3. Polimorfismo: Crea una lista de diversas figuras geométricas (por ejemplo, uno de cada tipo: Circulo, Rectangulo, Triangulo) y escribe un bucle que recorra la lista llamando a los métodos area y perimetro de cada figura, mostrando los resultados.

Ejercicio 3

Extiende el programa anterior que implementa figuras geométricas con los siguientes requisitos:

Clase **FiguraBidimensional**: Crea una nueva clase llamada FiguraBidimensional que Esta clase debe contener un método clonar(self) que devuelve una copia de la instancia (puedes usar copy para realizar la clonación).

Clase **FiguraTridimensional**: Crea una clase diferente llamada FiguraTridimensional, que debe incluir un método calcular_volumen(self) que deberá ser implementado por las clases derivadas que representen figuras tridimensionales.

Crear la clase Derivada **Cubo**: Crea una nueva clase llamada Cubo que derive de FiguraGeometrica FiguraBidimensional, FiguraTridimensional. Esta clase debe tener un constructor que acepte el lado del cubo.

Modificar la clase Circulo para que herede ahora también de FiguraBidimensional.