

23. Herencia básica y sobrescritura de métodos

1. Se desea crear un videojuego con los 3 tipos de personajes: perros, gatos y ratones. Todos van a compartir 2 propiedades que serán la vida (número de 1 a 10) y la fuerza (número de 1 a 5).

Crea una clase a la que llamaremos **Animal**, la clase ha de tener las siguientes características:

- **Propiedades:**
 - **vida:** Todos los objetos tendrán una vida inicial de 10 puntos.
 - **fuerza:** Esta propiedad quedará definida a través de un parámetro cuando se cree un objeto de la clase **Animal**.
- **Métodos:**
 - **propiedades():** Debe mostrar por pantalla un texto en el que se indique el valor de las propiedades vida y fuerza de un objeto de la clase.

Crea un objeto de la clase **Animal**, asígnale un valor 5 a su propiedad fuerza y muestra sus propiedades por pantalla utilizando el método propiedades.

```
Propiedades del Animal:  
Vida: 10  
Fuerza: 5
```

2. La propiedad fuerza no podrá tomar en ningún caso un valor mayor de 5. Modifica el programa anterior de tal forma que el valor de la propiedad fuerza se solicite a través de un input.

A continuación, modifica el constructor de los objetos **Animal** de tal forma que el objeto solamente se cree cuando el valor introducido para la propiedad fuerza sea un número menor o igual que cinco. Este control se tiene que realizar a través del método `__init__` del constructor. Comprueba que salta un error si le das un valor mayor

```
Introduce el valor de 'fuerza' del objeto a crear:4  
Propiedades del Animal:  
Vida: 10  
Fuerza: 4.0
```

```
Introduce el valor de 'fuerza' del objeto a crear:6  
Traceback (most recent call last):  
  File "D:\Google Drive\eclipse-workspace\20._  
Herencia\src\herencia_02.py", line 13, in <module>  
    perro.propiedades()  
  File "D:\Google Drive\eclipse-workspace\20._  
Herencia\src\herencia_02.py", line 9, in propiedades  
    print("Vida:",self.vida,"\nFuerza:",self.fuerza)  
AttributeError: 'Animal' object has no attribute 'vida'
```

3. Vamos a crear ahora tres tipos de personajes para nuestro juego. Elimina todo el código de prueba del ejercicio anterior dejando solo la definición de la clase **Animal**.

Cada uno de estos tipos de personajes quedará definido por una nueva clase. Cada una de las nuevas clases heredarán las propiedades de la superclase **Animal**. Además, cada una

de las nuevas subclases tendrá una propiedad particular que definiremos directamente en su código específico:

- Clase Perros:
 - Propiedad: size (número 3)
- Clase Gatos:
 - Propiedad: hambre (número 2)
- Clase Ratones:
 - Propiedad: color (“blanco”)

Crea un objeto de la clase Perros y muestra sus propiedades por pantalla sus propiedades utilizando el método propiedades.

Ten en cuenta que, como **Perros** hereda de **Animal**, usarás el constructor de la superclase, y por lo tanto, deberás pasar como parámetro el valor de la fuerza.

```
Introduce el valor de 'fuerza' del objeto a crear:5
Propiedades del Animal:
Vida: 10
Fuerza: 5.0
```

4. Observa que el programa anterior funciona correctamente, pero no tiene mucho sentido ya que no muestra la propiedad tamaño (size) exclusiva de la subclase **Perros**.

Crea dentro de la clase **Perros** un método que sobrescriba el método propiedades de la superclase y muestre las propiedades del objeto de la clase perros.

OJO: Para no repetir código, tu nuevo método debe llamar obligatoriamente al método de la superclase utilizando la función **super().propiedades()**, y justo después imprimir únicamente la nueva propiedad tamaño.

```
Introduce el valor de 'fuerza' del objeto a crear:5
Propiedades del Perro:
Vida: 10
Fuerza: 5.0
Tamaño: 3
```

5. Añade ahora un objeto tipo **Gatos** y otro tipo **Ratones**. Sobrescribe también sus métodos **propiedades()** (utilizando **super()**) para que muestren la información completa. Muestra todos por pantalla.

```
Introduce el valor de 'fuerza' del primer objeto a crear:4
Propiedades del Perro:
Vida: 10
Fuerza: 4.0
Tamaño: 3
Introduce el valor de 'fuerza' del segundo objeto a crear:3
Propiedades del Gato:
Vida: 10
Fuerza: 3.0
Hambre: 2
Introduce el valor de 'fuerza' del tercer objeto a crear:1
Propiedades del Ratón:
Vida: 10
Fuerza: 1.0
Color: blanco
```

6. Las subclases no solo modifican lo que heredan, ¡también pueden tener comportamientos completamente nuevos!

Añade un método llamado `ladrar()` exclusivo de la clase `Perros` que imprima por pantalla: "¡Guau, guau!".

Llama a este método desde tu objeto `perro` en el programa principal. A continuación, intenta que el objeto `gato` llame al método `ladrar()`. ¿Qué ocurre y por qué? (Borra el intento del `gato` una vez compruebes el error para que el programa siga funcionando).