

20. Herencia – Ejercicio videojuego

1. Se desea crear un videojuego con los 3 tipos de personajes, perros, gatos y ratones. Todos comparten 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 no a través del input sino a través del método `__init__` del constructor.

```
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. Partiremos del ejercicio anterior pero antes de nada elimina todo el código que no pertenezca a la definición de la clase Animal.

Cada uno de estos tipos de personajes quedará definido por una nueva clase. Todas las nuevas clases de personajes heredarán las propiedades fuerza y vida de la superclase Animal. Cada una de las nuevas clases tendrá unas propiedades particulares:

- Clase Perro:
 - Propiedad: size (número 3)

- Clase Gato:
 - Propiedad: hambre (número 2)
- Clase Raton:
 - Propiedad: color (“blanco”)

Creando un objeto de la clase Perro y muestra sus propiedades por pantalla sus propiedades utilizando el método propiedades.

Ten en cuenta que vas a utilizar el constructor de la superclase Animal para crear el objeto Perro, por lo tanto, deberás pasar como parámetro el valor de la propiedad 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, aunque funciona correctamente, no tiene mucho sentido ya que no estamos mostrando la propiedad tamaño del objeto de clase 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 (puede sonar un poco extraño, se explica como hacerlo en el punto “Sobrescribir un método heredado” de tus apuntes).

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

5. Añade un objeto tipo Gato y otro tipo Raton. Muestra sus propiedades 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. Vamos a complicar un poco las cosas modificando como creamos los objetos tipo Perro. Hasta ahora para crear un objeto de la clase Perro utilizamos una orden del tipo miperro = Perro(valor) donde valor es un número que indica el valor de la fuerza del objeto creado. Sin embargo, el objeto Perro tiene otra propiedad, size, a la que siempre le asignamos un valor 3.

Modifica el código del ejercicio anterior para que al crear un objeto Perro, utilicemos dos parámetros, el primero será la fuerza y el segundo el tamaño (size). El truco es utilizar un constructor para los objetos tipo Perro que a través de la función super() llame al constructor de los objetos tipo Animal. Intenta resolver esta situación.

Observa que también puedes utilizar la función `super()` para simplificar el método propiedades de la subclase Perro

```
Introduce el valor de 'fuerza' del objeto a crear:5
Introduce el valor del 'tamaño' del objeto a crear:1
Propiedades del Perro:
Vida: 10
Fuerza: 5.0
Tamaño: 1.0
```

7. Toma el código del ejercicio anterior e intenta imprimir por pantalla el nombre del objeto que has creado:

```
fuerza=float(input("Introduce el valor de 'fuerza' del objeto a crear:"))
size=float(input("Introduce el valor del 'tamaño' del objeto a crear:"))
perro=Perros(fuerza,size)
perro.propiedades()
print(perro)
```

Mostrará como resultado:

```
Introduce el valor de 'fuerza' del objeto a crear:5
Introduce el valor del 'tamaño' del objeto a crear:2
Propiedades del Perro:
Vida: 10
Fuerza: 5.0
Tamaño: 2.0
<__main__.Perros object at 0x000001C1A54D7580>
```

Generando un mensaje referido al objeto, pero no una información relevante.

Python permite crear un método especial de clase con el nombre `__str__` al que podemos añadir el código que nos interese. Lo que hace especial a este método, es que se ejecutará cuando invoquemos el nombre el objeto al que hace referencia.

8. Modifica el ejercicio anterior para que cuando escribamos la orden `print(perro)` se muestre un mensaje en el que se indique el valor de la vida, fuerza y tamaño del objeto en una única línea.

```
fuerza=float(input("Introduce el valor de 'fuerza' del objeto a crear:"))
size=float(input("Introduce el valor del 'tamaño' del objeto a crear:"))
perro=Perros(fuerza,size)
perro.propiedades()
print(perro)
```

Muestre:

```
Introduce el valor de 'fuerza' del objeto a crear:4
Introduce el valor del 'tamaño' del objeto a crear:3
Propiedades del Perro:
Vida: 10
Fuerza: 4.0
Tamaño: 3.0
Este perro tiene 4.0 puntos de fuerza, 10 puntos de vida y un tamaño de 3.0
```