

19. POO – Creando clases y objetos sencillos

1. Crea un modulo .py y define en él una clase, a la que llamarás **Libro**. Esta clase debe definir las características de los libros de tu biblioteca personal a través de dos propiedades:
 - a. **propietario**: El valor de esta propiedad ha de ser tu nombre completo.
 - b. **read**: Esta propiedad (estado) tomará inicialmente el valor lógico False e indicará si un libro ha sido ya leído o todavía no.

Crea ahora un objeto tipo **libro** y muestra por pantalla el valor de sus dos propiedades.

```
Andrés Sánchez  
False
```

2. Crea un nuevo módulo, copia del anterior, y añade a la clase **Libro** dos métodos:
 - a. El primer método **informará** de si un libro ha sido ya leído o todavía no. Para ello ha de mostrar uno de los dos mensajes siguientes:
 - a. Si la propiedad read tiene el valor False: “Todavía no has leído este libro”.
 - b. Si la propiedad read tiene el valor True: “Ya has leído este libro”.
 - b. El segundo método **realizará la acción** de cambiar el valor de la propiedad read de False a True.

A continuación, el programa realizará las siguientes operaciones:

- Crea un objeto de la clase **Libro**.
- Muestra por pantalla el valor de la propiedad propietario y del estado read al iniciar la ejecución de programa.
- Ejecuta el método que informa sobre si hemos leído el libro o no.
- Ejecuta el método que cambia el valor de la propiedad read a True
- Muestra de nuevo las propiedades el objeto que acabas de crear.
- Ejecuta el método que informa sobre si hemos leído el libro o no.

```
Propiedades iniciales  
-----  
Propietario: Andrés Sánchez  
Estado propiedad read: False  
Todavía no has leído este libro  
  
Propiedades tras lectura  
-----  
Propietario: Andrés Sánchez  
Estado propiedad read: True  
Ya has leído este libro
```

3. Crea un nuevo módulo. En el programa define una clase, a la que llamarás **Triangulo**. La clase debe de tener cuatro propiedades:
 - **tipo**: tomará el valor “Triángulo”
 - **lados**: asígnale el valor 3
 - **base**: asígnale el valor 0
 - **altura**: asígnale el valor 0

Crea un objeto tipo “triángulo” y muestra por pantalla el valor de sus cuatro propiedades.

```
Propiedades objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 0
Longitud de la altura: 0
```

4. Partiendo de una copia del programa anterior, crea un segundo objeto triángulo y muestra las propiedades de los dos objetos por pantalla. Enriquece un poco el formato de las órdenes print para mostrar de manera más clara la información.

```
Propiedades primer objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 0
Longitud de la altura: 0

Propiedades segundo objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 0
Longitud de la altura: 0
```

5. Crea una copia del ejercicio anterior y realiza en ella las siguientes modificaciones:.

Una vez que se han mostrado los datos de los dos triángulos modifica los valores de la base y la altura del primer triángulo (no en la clase triángulo, debes cambiarlos en el objeto). Los nuevos valores para el primer objeto serán:

- base= 5
- altura=4

Muestra de nuevo las propiedades de los dos objetos.

```
Propiedades primer objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 0
Longitud de la altura: 0

Propiedades segundo objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 0
Longitud de la altura: 0

***** DATOS ACTUALIZADOS *****
Propiedades primer objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 5
Longitud de la altura: 4

Propiedades segundo objeto clase Triangulo
-----
```

```
-----  
Tipo: Triángulo  
Número de lados: 3  
Longitud de la base: 0  
Longitud de la altura: 0
```

6. Crea un nuevo programa. Define en él de nuevo la clase **Triangulo** con las mismas propiedades que en los ejercicios anteriores:

- **tipo:** tomará el valor “Triángulo”
- **lados:** asígnale el valor 3
- **base:** asígnale el valor 0
- **altura:** asígnale el valor 0

Crea un objeto de la clase **Triangulo** y asígnale el valor 5 a la longitud de la base y 4 a la longitud de la altura.

Crea un segundo objeto triángulo. El programa debe preguntar al usuario por la longitud de la base y de la altura de este segundo triángulo. Establece estos valores como estado de las propiedades base y altura del segundo triángulo.

Muestra las propiedades de los dos triángulos por pantalla.

```
---- Datos segundo triángulo ----  
Introduce la longitud de la base: 5.4  
Introduce la longitud de la altura:3.27  
Propiedades del primer triángulo  
-----  
Tipo: Triángulo  
Número de lados: 3  
Longitud de la base: 5  
Longitud de la altura: 4  
  
Propiedades del segundo triángulo  
-----  
Tipo: Triángulo  
Número de lados: 3  
Longitud de la base: 5.4  
Longitud de la altura: 3.27
```

7. Partiendo de una copia del archivo del ejercicio anterior, añade un procedimiento a la clase **Triangulo**. Este nuevo procedimiento ha de devolver el valor la superficie del triángulo.

Añade el código necesario para que se muestre la superficie de los dos triángulos. Este valor se tiene que obtener llamando al procedimiento creado en la primera parte del ejercicio.

```
---- Datos segundo triángulo ----  
Introduce la longitud de la base: 5.4  
Introduce la longitud de la altura:3.27  
Propiedades del primer triángulo  
-----  
Tipo: Triángulo  
Número de lados: 3  
Longitud de la base: 5  
Longitud de la altura: 4  
Superficie triángulo: 10.0
```

```
Propiedades del segundo triángulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 5.4
Longitud de la altura: 3.27
Superficie triángulo: 8.829
```

8. Crea una copia del módulo anterior y realiza las modificaciones necesarias para conseguir que el programa nos informe de cual de los dos triángulos tiene una superficie mayor:

```
---- Datos segundo triángulo ----
Introduce la longitud de la base: 5
Introduce la longitud de la altura:8
Propiedades del primer triángulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 5
Longitud de la altura: 4
Superficie triángulo: 10.0

Propiedades del segundo triángulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 5.0
Longitud de la altura: 8.0
Superficie triángulo: 20.0
El segundo triángulo tiene una superficie mayor que el primero
```

```
---- Datos segundo triángulo ----
Introduce la longitud de la base: 4
Introduce la longitud de la altura:5
Propiedades del primer triángulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 5
Longitud de la altura: 4
Superficie triángulo: 10.0

Propiedades del segundo triángulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 4.0
Longitud de la altura: 5.0
Superficie triángulo: 10.0
Los dos triángulos tienen la misma superficie
Propiedades segundo objeto clase Triangulo
-----
Tipo: Triángulo
Número de lados: 3
Longitud de la base: 0
Longitud de la altura: 0
```

9. **Clase Videojuego** (métodos y parámetros)

Crea un nuevo módulo en el que definiremos una nueva clase con el nombre **Videojuego**. Asigna a la clase las siguientes **propiedades**:

- **titulo:** "Sin título"
- **nivel:** 1
- **puntuacion:** 0
- **completado:** False

Añade a la clase los siguientes métodos básicos:

- **subir_nivel():** incrementa nivel en 1
- **completar_juego():** cambia completado a True
- **propiedades():** Muestra por pantalla el valor de las propiedades del objeto de la clase

Crea un objeto de la clase Videojuego y muestra sus propiedades antes y después de ejecutar los métodos.

10. Métodos con parámetros

Añade a la clase definida en el ejercicio anterior los siguientes métodos:

- **sumar_puntos(puntos):** añade puntos a puntuacion
- **cambiar_titulo(nuevo_titulo):** cambia el título

A modo de ejemplo de uso, crea un objeto de la clase Videojuego y aplica los métodos necesarios para incrementar su puntuación en 150 puntos, subir su nivel en una unidad y modificar su título por la expresión 'Aventura Espacial'.

Por último muestra las propiedades actualizadas del objeto.

11. Interacción entre objetos

Crea un segundo videojuego, modifica sus propiedades con métodos y luego:

- Muestra ambos videojuegos
- Indica cuál tiene mayor puntuación
- Si hay empate, indícalo por pantalla

12. Método con parámetro booleano (True/False)

Añade a la clase Videojuego una nueva propiedad:

- **favorito:** False

Añade el siguiente método a la clase Videojuego:

```
def marcar_favorito(self, favorito):  
    self.favorito = favorito
```

Utilizando el método anterior, establece los siguientes valores para la propiedad favorito de cada objeto:

- juego1.favorito = False
- juego2.marcar_favorito(True)

Muestra por pantalla si cada juego es favorito o no.

13. Método que recibe otro objeto

Añade a la clase Videojuego un nuevo método:

```
def comparar_puntuacion(self, otro_videojuego):  
    # comparar self.puntuacion con otro_videojuego.puntuacion
```

Uso:

juego1.comparar_puntuacion(juego2)

Muestra cuál tiene mayor puntuación o si hay empate.

14. Reiniciar objeto

Crea un método reiniciar_juego() que:

- Ponga nivel = 1
- Ponga puntuacion = 0
- Ponga completado = False

Ejecuta el método y muestra el resultado para comprobar que funciona.