

20. POO – Constructores y encapsulado de variables

1. Crea un programa que conste de una clase llamada **Alumno**. La clase tendrá dos propiedades, el **nombre** y la **nota** del alumno. Crearemos tres métodos para esta clase:
 - Un método que **inicialice** las propiedades del objeto. Este método debe recibir como parámetros el nombre y la nota del objeto.
 - Un método que **imprima** por pantalla los valores de las propiedades
 - Un método que muestre si el alumno ha **aprobado** o no.

Crea un objeto, asígnale un nombre y una nota.

A continuación, ejecuta el método que muestra sus datos y el método que nos informa sobre si ha aprobado o suspendido.

A la hora de crear un objeto de esta clase, debes incluir como parámetros (entre paréntesis) los valores de las propiedades del mismo en la llamada.

2. Haz una copia del módulo del ejercicio anterior y añade un segundo objeto. En este caso las propiedades del segundo alumno se pedirán a través de teclado. Muestra por pantalla la información de los dos alumnos e indica si han aprobado o han suspendido.

```
Nombre del segundo alumno: Luisa
Nota del segundo alumno: 6.7

- Primer alumno
-----
Nombre: José
Nota: 7
El alumno ha aprobado

- Segundo alumno
-----
Nombre: Luisa
Nota: 6.7
El alumno ha aprobado
```

3. Modifica el programa anterior de tal manera que una vez creado el segundo objeto alumno el programa muestre el nombre de la persona que tenga mayor nota.

```
Nombre del segundo alumno: Marta
Nota del segundo alumno: 8.3
Marta ha obtenido una nota más alta que José.
```

4. Elabora un programa que tenga una clase a la que llamaremos **Persona**. La clase tendrá como propiedades el **nombre** y la **edad** de una persona. Diseña los siguientes métodos:
 - Método para **inicializar** las propiedades del objeto.
 - Método para **mostrar** los **datos** del objeto.
 - Método que muestre por pantalla si la persona es **mayor de edad** o no.

Añade el código necesario para crear un objeto de la clase **Persona**, el programa ha de solicitar sus propiedades por pantalla y a continuación ejecutar los métodos que muestran las propiedades del objeto y que indique si el alumno es o no es mayor de edad.

```
Nombre de la persona: Luisa
```

```
Edad de Luisa: 17

Luisa
17
Luisa todavía no es mayor de edad
```

En el ejercicio anterior el método que analiza si la persona es o no es mayor de edad, imprime una cadena de texto. Muchas veces nos va a interesar no que el método escriba algo, si no que devuelva un valor a través de un return. Vamos a aplicar esta variación en el siguiente ejercicio.

5. Haz una copia del modulo creado en el ejercicio anterior y modifícalo de tal forma que el método que estudia si la persona es mayor de edad o no lo es, devuelva un valor True si la persona es mayor de edad y False si no lo es. El método no debe escribir nada por pantalla.

Crea un nuevo objeto de la clase **Persona**. Las propiedades de este objeto se definirán a través del teclado.

El programa preguntará a continuación al usuario si quiere jugar a un determinado juego. Si el usuario contesta que no, el programa terminará. Si el usuario contesta que Si el programa debe dar la bienvenida y permitirle, o no, jugar en función de que el usuario sea, o no sea, mayor de edad. Para ello debes utilizar el método que determina si la persona es mayor de edad o si no lo es.

```
Nombre de la persona: Marta
Edad de Marta: 31
¿Quieres jugar una partida de 'Python: The Game' (S/N)?a
¿Quieres jugar una partida de 'Python: The Game' (S/N)?S
Bienvenido
```

```
Nombre de la persona: Luisa
Edad de Luisa: 17
¿Quieres jugar una partida de 'Python: The Game' (S/N)?S
No tienes edad suficiente para jugar a 'Python: The Game'.
```

6. Crea un nuevo módulo y en él define la clase **Triangulo**. La clase tendrá cuatro propiedades:

- **lados**: Número de lados.
- **lado1**: Longitud de uno de los lados
- **lado2**: Longitud de otro de los lados
- **lado3**: Longitud del tercer lado

Una de estas propiedades ha de estar encapsulada ya que no puede tomar otro valor más que el definido en el constructor. ¿Cuál de ellas es? Tenlo en cuenta para al definir el constructor.

Define ahora los siguientes **métodos**:

- Método que **muestra** por pantalla las cuatro **propiedades** del objeto.
- Método que **muestre** por pantalla la longitud del **lado más largo**.
- Método que **muestra** por pantalla el **tipo de triángulo** del que se trata (equilátero, isósceles o escaleno).

Crea un objeto triángulo definiendo la longitud de sus valores a través de ordenes input.

A continuación, ejecuta los métodos que muestran por pantalla la longitud del lado más largo y el tipo de triángulo con el que estamos trabajando. Por último, muestra por pantalla el valor de la propiedad “lados” del objeto.

```
Longitud del primer lado:5
Longitud del segundo lado:6
Longitud del tercer lado:7
Número de lados: 3
Lado 1: 5.0
Lado 2: 6.0
Lado 3: 7.0
El lado más largo mide:7.0
Triángulo escaleno
```

7. Vamos a comprobar que sentido tiene haber encapsulado la propiedad “lados” en la clase anterior. Partiendo del código anterior, añade al final del mismo una línea de código que asigne un valor cualquiera a propiedad lado3.

```
miTriangulo.lado3=33333333
```

Vuelve a mostrar ahora las propiedades del objeto, ¿qué ha sucedido?

```
Longitud del primer lado:5
Longitud del segundo lado:6
Longitud del tercer lado:7
Número de lados: 3
Lado 1: 5.0
Lado 2: 6.0
Lado 3: 7.0
El lado más largo mide:7.0
Triángulo escaleno
Número de lados: 3
Lado 1: 5.0
Lado 2: 6.0
Lado 3: 33333333
```

Como puedes ver, hemos modificado el valor de la propiedad lado3 desde fuera del código de la clase. Intenta modificar ahora de la misma forma el valor de la propiedad número de lados y muestra el resultado por pantalla otra vez ¿Ocurre lo mismo?

8. Una regla básica de geometría establece que: Un triángulo no se podrá construir si la suma de la longitud de dos cualquiera de sus lados es menor o igual que la longitud del tercero. Modifica el método de construcción de los triángulos en los ejercicios anteriores para que en caso de que los lados no cumplan la condición anterior no se cree el objeto, sino que se genere un mensaje de error. Al final y como en los ejercicios anteriores se ha de mostrar por pantalla la información del objeto ejecutando los métodos correspondientes.

```
Longitud del primer lado:4
Longitud del segundo lado:5
Longitud del tercer lado:6
Número de lados: 3
Lado 1: 4.0
Lado 2: 5.0
Lado 3: 6.0
El lado más largo mide:6.0
Triángulo escaleno
```

```
Longitud del primer lado:4
Longitud del segundo lado:5
Longitud del tercer lado:9
No es posible construir el objeto
Traceback (most recent call last):
  File "D:\Google Drive\eclipse-workspace\17. P00 Constructores y encapsulado de variables\src\constructores_08.py", line 48, in <module>
    trian.propiedades()
  File "D:\Google Drive\eclipse-workspace\17. P00 Constructores y encapsulado de variables\src\constructores_08.py", line 14, in propiedades
    print("Número de lados:", self.__lados)
AttributeError: 'Triangulo' object has no attribute '_Triangulo__lados'
```

Como las últimas líneas de código intentan trabajar con un objeto que no se ha creado de forma correcta (no se cumplía la condición geométrica), se muestra un error.

Gestiona esta excepción de forma adecuada de tal manera que no se genere el error.

```
Longitud del primer lado:5
Longitud del segundo lado:4
Longitud del tercer lado:10
No es posible construir el objeto
```

9. Parámetros con valor por defecto.

Partiendo de la clase Persona creada en los ejercicios anteriores, modifica el constructor de la clase para que el parámetro edad tenga un valor por defecto de 18.

Crea dos objetos:

- Uno indicando nombre y edad.
- Otro indicando únicamente el nombre.

A continuación, muestra por pantalla las propiedades de ambos objetos e indica si son mayores de edad utilizando el método correspondiente.

10. Paso de argumentos por nombre

Partiendo del ejercicio anterior, crea un nuevo módulo en el que los objetos se creen especificando los argumentos por nombre.

Ejemplo:

```
p1 = Persona(nombre="Laura", edad=30)
```

```
p2 = Persona(edad=16, nombre="Miguel")
```

Comprueba que el orden de los argumentos ya no importa y las propiedades se asignan correctamente.

El programa deberá crear al menos dos objetos utilizando argumentos por nombre, mostrar sus propiedades e indicar si son mayores de edad.

11. Desafío final – Diseño flexible de constructores

Crea un nuevo módulo con una clase llamada **CuentaBancaria**. La clase tendrá las siguientes propiedades:

- **titular**
- **saldo**
- **activa** (valor booleano)

El constructor deberá cumplir las siguientes condiciones:

- El saldo tendrá un valor por defecto de 0.
- La cuenta estará activa por defecto (True).

Define los siguientes métodos:

- Método que muestre por pantalla los datos de la cuenta.
- Método ingresar(cantidad) que aumente el saldo.
- Método retirar(cantidad) que disminuya el saldo solo si hay dinero suficiente.
- Método desactivar() que cambie el estado de la cuenta a False.

Crea al menos tres objetos distintos utilizando:

- Parámetros posicionales.
- Parámetros por nombre.
- Parámetros combinados (uno posicional y otro por nombre).

Encapsula la propiedad activa.

Realiza varias operaciones con los métodos definidos y muestra el resultado final de cada cuenta.