

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Coyla Idme Leonel

Alumno:

Ticona Miramira Roberto Angel

Encapsulamiento

» DESCRIPCIÓN

En POO, el encapsulamiento es un principio que consiste en ocultar los detalles internos del funcionamiento de un objeto y restringir el acceso directo a algunas de sus componentes, exponiendo solo lo necesario a través de interfaces públicas (métodos)

» EJERCICIO 1

Desarrollar un objeto Persona, utilizando atributos y métodos privados.

- Clase: Persona.
- Atributo: nombre, edad.
- Acción: get_edad, get_nombre, set_edad, set_nombre.
- Objetos: persona = Factorial("María", 35)

Código

```
1 class Persona:
2     def __init__(self, nombre, edad):
3         self.__nombre = nombre # atributo, con __ se llama atributo privado
4         self.__edad = edad # atributo, con __ se llama atributo privado
5
6     def get_edad(self):
7         return self.__edad
8
9     def get_nombre(self):
10        return self.__nombre
11
12    def set_edad(self, nueva_edad):
13        if nueva_edad > 0:
14            self.__edad = nueva_edad
15        else:
16            print("Edad no valida")
17
18    def set_nombre(self, nuevo_nombre):
19        self.__nombre = nuevo_nombre
20        return self.__nombre
21
22 persona = Persona("Ana", 30)
23 print(persona.get_nombre())
24 print(persona.get_edad())
25 print(persona.get_nombre(), persona.get_edad())
26
27 persona.set_edad(35)
28 persona.set_nombre("Maria")
29
```

```
30 print(persona.get_edad())
31 print(persona.get_nombre())
```

Ejecución

```
1 Ana
2 30
3 Ana 30
4 35
5 María
```

» EJERCICIO 2

Hallar el área y el perímetro de un círculo.

- Clase: Circulo.
- Atributo: radio.
- Acción: calcular_area, calcular_perímetro.
- Objeto: circulo = Circulo(3)

Código

```
1 import math
2 class Circulo:
3     def __init__(self, radio):
4         self.__radio = radio
5
6     def get_radio(self):
7         return self.__radio
8
9     def set_radio(self, nueva_radio):
10        if nueva_radio > 0:
11            self.__radio = nueva_radio
12        else:
13            print("Radio no valido")
14
15    def calcular_area(self):
16        return math.pi * self.__radio ** 2
17
18    def calcular_perimetro(self):
19        return 2 * math.pi * self.__radio
20
21 circulo = Circulo(3)
22 print("Área del círculo: ", round(circulo.calcular_area(),2))
23 print("Perímetro del círculo: ", round(circulo.calcular_perimetro(),2))
```

Ejecución

```
1 Área del círculo:  28.27
2 Perímetro del círculo:  18.85
```

» EJERCICIO 3

Hallar el área y el perímetro de un rectángulo.

- Clase: Rectangulo.
- Atributo: base, altura.
- Acción: calcular_area, calcular_perímetro.

- Objeto: rectangulo = Rectangulo(4,5)

Código

```
1 class Rectangulo:
2     def __init__(self, base, altura):
3         self.__base = base
4         self.__altura = altura
5
6     def get_base(self):
7         return self.__base
8
9     def get_altura(self):
10        return self.__altura
11
12    def set_base(self, nueva_base):
13        if nueva_base > 0:
14            self.__base = nueva_base
15        else:
16            print("Base invalida")
17
18    def set_altura(self, nueva_altura):
19        if nueva_altura > 0:
20            self.__altura = nueva_altura
21        else:
22            print("Altura invalida")
23
24    def calcular_area(self):
25        return self.__altura * self.__base
26
27    def calcular_perimetro(self):
28        return (self.__altura + self.__base) * 2
29
30 rectangulo = Rectangulo(4,5)
31 print("Área del rectángulo ", rectangulo.calcular_area())
32 print("Perímetro del rectángulo ", rectangulo.calcular_perimetro())
```

Ejecución

```
1 Área del rectángulo  20
2 Perímetro del rectángulo  18
```

» EJERCICIO 4

Crear una calculadora que dado dos números, calcule su suma, resta, producto y división.

- Clase: Calculadora.
- Atributo: num1, num2.
- Acción: suma(), resta(), producto(), cociente().
- Objeto: calculadora = calculadora(2,5)

Código

```
1 class Calculadora:
2     def __init__(self, num1, num2):
3         self.__num1 = num1
4         self.__num2 = num2
5
6     def get_num1(self):
7         return self.__num1
8
```

```
9     def get_num2(self):
10         return self.__num2
11
12     def set_num1(self, nuevo_num1):
13         if nuevo_num1 > 0:
14             self.__num1 = nuevo_num1
15         else:
16             print("Número invalido")
17
18     def set_num2(self, nuevo_num2):
19         if nuevo_num2 > 0:
20             self.__num2 = nuevo_num2
21         else:
22             print("Número invalido")
23
24     def suma(self):
25         return self.__num1 + self.__num2
26
27     def resta(self):
28         return self.__num1 - self.__num2
29
30     def producto(self):
31         return self.__num1 * self.__num2
32
33     def cociente(self):
34         return self.__num1 / self.__num2
35
36 def main():
37     calculadora = Calculadora(2,5)
38     print("Suma: ", calculadora.suma())
39     print("Resta: ", calculadora.resta())
40     print("Producto: ", calculadora.producto())
41     print("División: ", calculadora.cociente())
42 if __name__=="__main__":
43     main()
```

Ejecución

```
1 Suma: 7
2 Resta: -3
3 Producto: 10
4 División: 0.4
```