

Métodos Operadores

Alumno: Puma Huanca Anthony Rusbel

Docente: Ing. Coyla Idme Leonel

Lenguajes de programación II – FINESI

Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Métodos Operadores

En Python se tienen los siguientes métodos especiales (también llamados *métodos mágicos* o *dunder methods*) que permiten sobrecargar operadores:

`__str__` → Permite mostrar (imprimir) un objeto de forma legible.

`__eq__` → Permite comparar dos objetos con el operador `==`.

`__gt__` → Permite verificar si un objeto es mayor que otro (`>`).

`__add__` → Permite sumar dos objetos con el operador `+`.

`__sub__` → Permite restar dos objetos con el operador `-`.

Ejemplo 1

Crear el objeto `Persona` y usar los métodos operadores.

Clase: `Persona`.

Atributos: Nombre, Edad.

Acciones: `__str__`, `__eq__`, `__gt__`, `__add__`.

Objetos: `persona1 = Persona("Carlos", 30)`, `persona2 = Persona("Ana", 25)`, `persona3 = Persona("Lucia", 30)`.

```
1 class Persona:
2     def __init__(self, nombre, edad):
3         self.nombre = nombre
4         self.edad = edad
5
6     def __str__(self):
7         return f"{self.nombre}, {self.edad} años"
8
9     def __eq__(self, otra):
10        return self.edad == otra.edad
11
```

```

12     def __gt__(self, otra):
13         return self.edad > otra.edad
14
15     def __add__(self, otra):
16         return self.edad + otra.edad
17
18 persona1 = Persona("Carlos", 30)
19 persona2 = Persona("Ana", 25)
20 persona3 = Persona("Lucia", 30)
21
22 print(persona1)
23 print(persona2)
24 print(persona3)
25
26 print(persona1 == persona2)
27 print(persona1 == persona3)
28
29 print(persona1 > persona2)
30
31 print("Suma de edades:", persona1 + persona2)
32 print("Suma de edades:", persona1 + persona3)

```

Listing 1: Código Ejemplo 1

Ejecución:

```

Carlos, 30 años
Ana, 25 años
Lucia, 30 años
False
True
True
Suma de edades: 55
Suma de edades: 60

```

Ejemplo 2

Crear el objeto Producto y usar los métodos operadores.

Clase: Producto.

Atributos: Nombre, Precio, Stock.

Acciones: __str__, __eq__, __add__.

Objetos: prod1 = Producto("arroz", 3.50, 20), prod2 = Producto("arroz", 3.50, 15), prod3 = Producto("azúcar", 4.00, 10).

```

1 class Producto:
2     def __init__(self, nombre, precio, stock):
3         self.nombre = nombre
4         self.precio = precio

```

```

5         self.stock = stock
6
7     def __str__(self):
8         return f"{self.nombre} - S/.{self.precio:.2f} stock: {self.
9             stock}"
10
11    def __eq__(self, otra):
12        return self.nombre == otra.nombre
13
14    def __add__(self, otra):
15        return self.precio + otra.precio
16
17 prod1 = Producto("Arroz", 3.50, 20)
18 prod2 = Producto("Arroz", 3.50, 15)
19 prod3 = Producto("Az car", 4.00, 10)
20
21 print(prod1)
22 print(prod2)
23
24 print(prod1 == prod2)
25 print(prod1 == prod3)
26
27 print("Suma de precios:", prod1 + prod2)

```

Listing 2: Código Ejemplo 2

Ejecución:

```

Arroz - S/.3.50 stock: 20
Arroz - S/.3.50 stock: 15
Azúcar - S/.4.00 stock: 10
True
False
Suma de precios: 7.0

```

Ejemplo 3

Crear el objeto CuentaBancaria que permite ingresar el nombre y el saldo.

Clase: CuentaBancaria.

Atributos: Titular, Saldo.

Acciones: mostrar(), __sub__.

Objeto: cuenta1 = CuentaBancaria(nombre, dinero).

```

1 class CuentaBancaria:
2     def __init__(self, titular, saldo):
3         self.titular = titular
4         self.saldo = saldo

```

```

5
6     def mostrar(self):
7         print(f"Titular: {self.titular} - Saldo: ${self.saldo:.2f}")
8
9     def __sub__(self, cantidad):
10        if isinstance(cantidad, (int, float)):
11            if cantidad <= self.saldo:
12                return CuentaBancaria(self.titular, self.saldo -
13                                         cantidad)
14            else:
15                print("Fondos insuficientes")
16                return self
17        else:
18            print("Operador no valido")
19            return self
20
21 nombre = input("Ingrese su nombre: ")
22 dinero = float(input("Ingrese su saldo: "))
23 cuenta1 = CuentaBancaria(nombre, dinero)
24
25 cuenta2 = cuenta1 - 250
26 cuenta2.mostrar()
27
28 cuenta3 = cuenta2 - 700
29 cuenta3.mostrar()

```

Listing 3: Código Ejemplo 3

Ejecución:

```

Ingrese su nombre: Luis
Ingrese su saldo: 1000
Titular: Luis - Saldo: $750.00
Titular: Luis - Saldo: $50.00

```