

METODOS Y SOBRECARGAS

1. **Métodos:** Un método es una función definida dentro de una clase que realiza una acción sobre los datos del objeto. Los métodos representan el comportamiento del objeto.

Ejemplo:

Clase: persona

Atributos: nonmbre

Acción:

Objetos:

```
class Persona:
    def __init__(self,nombre):
        self.nombre = nombre

    def saludar(self):
        print(f"Hola,soy {self.nombre}")

persona1 = Persona("Carlos")
persona2 = Persona("Maria")
persona1.saludar()
persona2.saludar()
```

2. **Parámetros en método:** Los métodos pueden recibir parámetros para trabajar con datos externos. El primer parámetro de todo método es de instancia es “self”, que representa al objeto mismo.

Ejemplo:

Clase: calculadora

Atributos: numero1, numero2

Acción:

Objetos:

```
class Calculadora:

    def __init__(self, numero1, numero2):

        self.__numero1 = numero1

        self.__numero2 = numero2


    def suma(self):

        return self.__numero1 + self.__numero2


    def resta(self):

        return self.__numero1 - self.__numero2


    def multiplicacion(self):

        return self.__numero1 * self.__numero2


    def division(self):

        if self.__numero2 != 0:

            return self.__numero1 / self.__numero2

        else:

            return "Error: División por cero"
```

3. **Retorno de métodos:** Un método puede devolver un valor usando la instrucción return.

Esto permite usar el resultado del método en otras partes del programa.

Ejemplo:

Clase: rectangulo

Atributos: base, altura

Acción:

Objetos:

```
class Rectangulo:

    def __init__(self,base,altura):

        self.base = base

        self.altura = altura


    def calcular_area(self):

        return self.base*self.altura

rectangulo = Rectangulo(10,5)

area = rectangulo.calcular_area()

print("el area del rectangulo es:",area)
```

4. **Sobre carga de métodos:** Consiste en definir múltiples de acciones de un método con el mismo nombre, pero diferentes parámetros.

Clase: operaciones

Atributos: a,b,c=None

Acción:

Objetos:

```
class Operaciones:

    def Sumar(self, a, b, c=None):

        if c is not None:

            return a + b + c

        else:

            return a + b


operaciones = Operaciones()

resultado1 = operaciones.Sumar(1, 3)

print("Suma con dos parámetros:", resultado1)

resultado2 = operaciones.Sumar(1, 3, 4)

print("Suma con tres parámetros:", resultado2)
```

Ejemplo 5:

Clase:circulo

Atributos:radio

Acción:

Objetos:

```
import math

class Circulo:

    def __init__(self,radio):

        self.radio = radio


    def calcular_area(self):

        area = math.pi * self.radio**2

        return area


    def calcular_perimetro(self):

        perimetro = 2 * math.pi* self.radio

        return perimetro


    def mostrar_informacion(self):

        print(f"Radio del circulo es:{self.radio}")

        print(f"El area es:{self.calcular_area():.2f}")

        print(f"El perimetro es:{self.calcular_perimetro():.2f}")


circulo = Circulo(7)

circulo.mostrar_informacion()
```