

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Coyla Idme Leonel

Alumno:

Ticona Miramira Roberto Angel

Introducción a la Programación Orientada a Objetos

» DESCRIPCIÓN

La Programación Orientada a Objetos (POO) es un paradigma que organiza el software en torno a objetos, los cuales combinan datos (atributos) y comportamientos (métodos) en una misma entidad. Estos objetos se basan en clases, que funcionan como molde o plantillas que definen sus características y acciones.

» EJERCICIO 1

Definir lo siguiente.

- Clase: Coche.
- Atributos(Propiedades): Marca, Modelo, Color.
- Comportamiento(Acción): Arranca, Acelera, Frena.
- Objeto: Toyota, Corolla, Blanco

Código

```
1 class Coche:
2     def __init__(self, Marca, Modelo, Color):
3         self.Marca = Marca
4         self.Modelo = Modelo
5         self.Color = Color
6
7     def Arranca(self):
8         print(f"El coche {self.Marca} {self.Modelo} {self.Color} encendió")
9
10    def Acelera(self):
11        print(f"El coche {self.Marca} {self.Modelo} {self.Color} aceleró")
12
13    def Frena(self):
14        print(f"El coche {self.Marca} {self.Modelo} {self.Color} frenó")
15
16 mi_coche = Coche("Toyota", "Corolla", "Blanco")
17
18 mi_coche.Arranca()
19 mi_coche.Acelera()
20 mi_coche.Frena()
```

Ejecución

```
1 El coche Toyota Corolla Blanco encendió
2 El coche Toyota Corolla Blanco aceleró
3 El coche Toyota Corolla Blanco frenó
```

» EJERCICIO 2

Calcular la hipotenusa utilizando el teorema de Pitágoras.

- Clase: Pitágoras.
- Atributos: Cateto a, Cateto b.
- Comportamiento: Hipotenusa.
- Objeto: Triángulo(4,3)

Código

```
1 class Pitagoras:
2     def __init__(self, catetoa, catetob):
3         self.catetoa = catetoa
4         self.catetob = catetob
5
6     def Hipotenusa(self):
7         return (self.catetoa ** 2 + self.catetob ** 2)**(1/2)
8
9 triangulo = Pitagoras(4,3)
10
11 print("La hipotenusa es", triangulo.Hipotenusa())
```

Ejecución

```
1 La hipotenusa es 5.0
```

» EJERCICIO 3

Calcular el área y el perímetro de un rectángulo.

- Clase: Rectángulo.
- Atributos: Base, Altura.
- Comportamiento: Área, Perímetro.
- Objeto: Operación(5,6)

Código

```
1 class Rectangulo:
2     def __init__(self, base, altura):
3         self.base = base
4         self.altura = altura
5
6     def area(self):
7         return self.base * self.altura
8
9     def perimetro(self):
10         return 2 * (self.base + self.altura)
11
12 operacion = Rectangulo(5, 6)
13
14 print("Área: ", operacion.area())
15 print("Perímetro: ", operacion.perimetro())
```

Ejecución

```
1 Área:  30
2 Perímetro:  22
```

» EJERCICIO 4

Calcular la cantidad de ladrillos en un metro cuadrado.

Clase: Ladrillo.

Atributos: Longitud, Altura, Ancho.

Comportamiento: Cantidad de ladrillos en $1m^2$.

Objeto: Corregido(0.24, 0.09, 0.13)

Código

```
1 class Ladrillos:
2     def __init__(self, longitud, altura, ancho):
3         self.longitud = longitud
4         self.altura = altura
5         self.ancho = ancho
6
7     def calcular(self):
8         return (1 / ((self.longitud + 0.015) * (self.altura + 0.015)))
9
10 #Solicitar datos al usuario
11
12 longitud = float(input("Ingrese la longitud del ladrillo: "))
13 altura = float(input("Ingrese la altura del ladrillo: "))
14 ancho = float(input("Ingrese el ancho del ladrillo: "))
15
16 #Objeto
17
18 cantidad = Ladrillos(longitud, altura, ancho)
19
20 print("La cantidad de ladrillos en 1 metro^2 es: ", cantidad.calcular(), " sin
    ↳ desperdicios")
21 print("La cantidad de ladrillos corregido en 1 metro^2 es: ",
    ↳ cantidad.calcular()*1.05, " con desperdicios")
22 print("La cantidad de ladrillos en 8.05 metros^2 es: ",
    ↳ cantidad.calcular()*1.05*8.05)
```

Ejecución

```
1 Ingrese la longitud del ladrillo: 0.24
2 Ingrese la altura del ladrillo: 0.09
3 Ingrese el ancho del ladrillo: 0.13
4 La cantidad de ladrillos en 1 metro^2 es:  37.34827264239029  sin desperdicios
5 La cantidad de ladrillos corregido en 1 metro^2 es:  39.21568627450981  con
    ↳ desperdicios
6 La cantidad de ladrillos en 8.05 metros^2 es:  315.686274509804
```