

ESTRUCTURAS SELECTIVAS

En el contexto de P.O.O. con Python, las estructuras selectivas son aquellas que permiten tomar decisiones durante la ejecución del programa, es decir, ejecutar diferentes bloques de código según se cumpla o no una condición.

Aunque las estructuras selectivas no son exclusivas de la programación P.O.O.(también se usan en programación estructurada), en un programa P.O.O. se usan dentro de elementos o funciones que pertenecen a clases para controlar el flujo de ejecución.

Definición: Una estructura selectiva en la P.O.O. en Python es una construcción de control que permite ejecutar diferentes bloques de código dentro de métodos de clases, dependiendo de si se cumple una o varias condiciones.

1. Estructura selectiva simple: Se ejecuta un bloque de código solo si se cumple la condición, en caso contrario, no hace nada.

Ejemplo:

```
if (x>0)
    print("X" es positiva)
```

2. Estructura selectiva doble: Permite dos ramas: una si la condición es verdadera, otra si es falsa.

Ejemplo: if (x % 2==0)

```
Print("X es un número par")
```

Else:

```
Print("X es un número impar")
```

3. Estructura selectiva múltiple:

Gestiona más de dos alternativas mutuamente exclusivas. Solo se ejecuta la primera condición verdadera.

Ejemplo: if(" X >= 90")

```
Print("x aprobado con distinción")
```

```
Elif (x>= 90)
```

```
Print("x aprobado")
```

Else:

```
Print("x reprobado")
```

Ejemplos de códigos:

Ejemplo N°1:

Clase: Pitágoras

Atributos: catetoa, catetob

Acción: calcular_hipotenusa()

Objeto: operación

```

class Pitagoras:

    def __init__(self, catetoa, catetob):
        self.catetoa = catetoa
        self.catetob = catetob

    def calcular_hipotenusa(self):
        return (self.catetoa**2 + self.catetob**2)**0.5

catetoa = float(input("Ingrese el valor del cateto a: "))
catetob = float(input("Ingrese el valor del cateto b: "))

operacion = Pitagoras(catetoa, catetob)
print("La hipotenusa es", operacion.calcular_hipotenusa())

```

Ejem N°2:

Clase: rectangulo

Atributo: base, altura

Acción: calcular_area(),
calcular_perimetro()

Objeto: operacion =
Rectangulo(5, 6)

```

class Rectangulo:

    def __init__(self, base, altura):
        self.base = base
        self.altura = altura

    def calcular_area(self):
        return self.base * self.altura

    def calcular_perimetro(self):
        return 2 * (self.base + self.altura)

operacion = Rectangulo(5, 6)

print("El Área es:", operacion.calcular_area())
print("El Perímetro es:", operacion.calcular_perimetro())

```

Ejemplo N°3:

Clase: ladrillo

Atributos: longitud, altura

Acción: calcular_cantidad de ladrillo()

Objeto: miladrillo = Ladrillo(longitud, altura)

```
class Ladrillo:

    def __init__(self, longitud, altura):
        self.longitud = longitud
        self.altura = altura

    def calcular_cantidad_ladrillos(self, area_m2):

        area_ladrillo_con_mortero = (self.longitud + 0.015) * (self.altura + 0.015)
        cantidad = area_m2 / area_ladrillo_con_mortero
        return cantidad

longitud = float(input("Introduce la longitud del ladrillo (en metros): "))
altura = float(input("Introduce la altura del ladrillo (en metros): "))
area_pared = float(input("Introduce el área de la pared (en metros cuadrados): "))

mi_ladrillo = Ladrillo(longitud, altura)

cantidad_ladrillos = mi_ladrillo.calcular_cantidad_ladrillos(area_pared)
cantidad_con_desperdicio = cantidad_ladrillos * 1.05

print(f"\nSe necesitan aproximadamente {round(cantidad_ladrillos)} ladrillos sin considerar desperdicio.")
print(f"Con un 5% de desperdicio: {round(cantidad_con_desperdicio)} ladrillos.")
```

Ejemplo N°4:

Clase:coche

Atributos: marca, modelo, color

Acción: mostrar_info(), arrancar()

Objeto:mi coche=coche()

```
class coche:  
    def __init__(self,marca,modelo,color):  
        self.marca = marca  
        self.modelo = modelo  
        self.color = color  
  
    def mostrar_info(self):  
        print(f"Coche: {self.marca} {self.modelo} {self.color}")  
  
    def arrancar(self):  
        print(f"El {self.marca} {self.modelo} ha arrancado ")  
  
mi_coche = coche("Toyota","Corolla","blanco")  
  
mi_coche.mostrar_info()  
mi_coche.arrancar()
```

Ejemplo N°5:

Clase:numero

Atributos:valor

Acción:clasificar()

Objeto:tipo = num clasificar()

```
class Numero:  
    def __init__(self,valor):  
        self.valor = valor  
  
    def clasificar(self):  
        if self.valor == 0:  
            return "Nulo"  
        elif self.valor%2==0:  
            return "Par"  
        else:  
            return "Impar"  
ejemplo = [Numero(0), Numero(2), Numero(5)]  
  
for num in ejemplo:  
    tipo = num.clasificar()  
    print(f"El Numero {num.valor} es {tipo}")
```

Ejemplo N°6:

Clase: persona

Atributos: nombre,edad

Acción: clasificar()

Objeto: persona=Persona(nombre,edad)

```
class Persona:  
    def __init__(self, Nombre, Edad):  
        self.Nombre = Nombre  
        self.Edad = Edad  
  
    def clasificar(self):  
        if self.Edad >= 18:  
            print(f"{self.Nombre} es mayor de edad.")  
        else:  
            print(f"{self.Nombre} es menor de edad.")  
  
    nombre = input("Introduce el nombre: ")  
    edad = int(input("Introduce la edad: "))  
  
    persona = Persona(nombre, edad)  
  
    persona.clasificar()
```

Ejemplo N°7:

Clase: empleado

Atributos:nombre, cargo, salario

Acción: aplicarAumento

Objeto: emp = Empleado(nombre, cargo, salario)

```
class Empleado:  
    def __init__(self, nombre, cargo, salario):  
        self.nombre = nombre  
        self.cargo = cargo  
        self.salario = salario  
  
    def aplicarAumento(self):  
        if self.cargo == "Gerente":  
            porcentaje = 0.10  
        elif self.cargo == "Supervisor":  
            porcentaje = 0.07  
        elif self.cargo == "Operario":  
            porcentaje = 0.05  
        else:
```

```
porcentaje = 0.0

nuevoSalario = self.salario * (1 + porcentaje)
return nuevoSalario

n = int(input("¿Cuántos empleados quieres ingresar? "))

empleados = []

for i in range(n):
    print(f"\nEmpleado {i+1}:")
    nombre = input("Nombre: ")
    cargo = input("Cargo (Gerente, Supervisor, Operario): ")
    salario = float(input("Salario: "))

    emp = Empleado(nombre, cargo, salario)
    empleados.append(emp)

print("\nSalarios con aumento aplicado:")

for emp in empleados:
    nuevo = emp.aplicarAumento()
    print(f"{emp.nombre} ({emp.cargo}): salario nuevo = {nuevo:.2f}")
```