

Estructuras selectivas

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

Estructuras selectivas

Descripción

En el contexto de programación orientada a objetos 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 POO, también se usan en programación estructurada; en un programa orientado a objetos se usan dentro de métodos o funciones que pertenecen a clases para controlar el flujo de ejecución.

Ejercicio 1

Identificar si un número es nulo, par o impar.

Clase: `Numero`.

Atributo: `Valor`.

Acción: Clasificar el número.

Objetos: `Numero(0)`, `Numero(2)`, `Numero(5)`.

```
1 class Numero:
2     def __init__(self, valor):
3         self.valor = valor
4
5     def clasificar(self):
6         if self.valor == 0:
7             return "Nulo"
8         elif self.valor % 2 == 0:
9             return "Par"
10        else:
11            return "Impar"
12
13 ejemplos = [Numero(0), Numero(2), Numero(5)]
```

```

14
15 for num in ejemplos:
16     tipo = num.clasificar()
17     print(f"El número {num.valor} es {tipo}")

```

Listing 1: Código Ejercicio 1

Ejecución:

El número 0 es Nulo
 El número 2 es Par
 El número 5 es Impar

Ejercicio 2

Determinar si una persona es mayor de edad.

Clase: Persona.

Atributos: Nombre, Edad.

Comportamiento: Es_mayor_de_edad().

Objeto: Persona("Maria", 25).

```

1 class Persona:
2     def __init__(self, Nombre, Edad):
3         self.Nombre = Nombre
4         self.Edad = Edad
5
6     def Es_mayor_de_edad(self):
7         if self.Edad > 18:
8             return "Es mayor de edad"
9         else:
10            return "No es mayor de edad"
11
12 ejemplo = Persona("Maria", 25)
13 res = ejemplo.Es_mayor_de_edad()
14
15 print(ejemplo.Nombre, res)

```

Listing 2: Código Ejercicio 2

Ejecución:

Maria Es mayor de edad

Ejercicio 3

Determinar el aumento de sueldo de un trabajador dependiendo del cargo.

Clase: Empleado.

Atributos: Nombre, Cargo, Salario.

Acción: AplicarAumento().

Reglas:

Gerente: 10 %

Supervisor: 7 %

Operario: 5 %

Objetos:
Empleado("Carlos", "Gerente", 2000)

Empleado("Maria", "Supervisor", 2000)

Empleado("Ana", "Interna", 800)

Empleado("Roberto", "Operario", 1600)

```
1 class Empleado:
2     def __init__(self, nombre, cargo, salario):
3         self.nombre = nombre
4         self.cargo = cargo
5         self.salario = salario
6
7     def AplicarAumento(self):
8         if self.cargo == "Gerente":
9             porcentaje = 0.10
10        elif self.cargo == "Supervisor":
11            porcentaje = 0.07
12        elif self.cargo == "Operario":
13            porcentaje = 0.05
14        else:
15            porcentaje = 0.0
16
17        nuevoSalario = self.salario * (1 + porcentaje)
18        return nuevoSalario
19
20 empleado1 = Empleado("Carlos", "Gerente", 2000)
21 empleado2 = Empleado("Maria", "Supervisor", 2000)
22 empleado3 = Empleado("Ana", "Interna", 800)
23 empleado4 = Empleado("Roberto", "Operario", 1600)
24
25 for emp in (empleado1, empleado2, empleado3, empleado4):
26     nuevo = emp.AplicarAumento()
27     print(f"{emp.nombre} {emp.cargo}: salario nuevo {nuevo:.2f}")
```

Listing 3: Código Ejercicio 3

Ejecución:

Carlos Gerente: salario nuevo 2200.00

Maria Supervisor: salario nuevo 2140.00

Ana Interna: salario nuevo 800.00

Jose Operario: salario nuevo 1680.00