

Laboratorio Herencia

Crear una aplicación de **herencia** usando las clases Empleado (superclase) y Vendedor (subclase).

Crema una carpeta c6 y un nuevo archivo empleados.py

```
empleados.py > ...
1  # Clase base o superclase
2  class Empleado:
3      def __init__(self, nombre, salario):
4          self.nombre = nombre
5          self.salario = salario
6
7      def mostrar_datos(self):
8          print(f"Nombre: {self.nombre}")
9          print(f"Salario: {self.salario} €")
10
11
12  # Clase derivada o subclase
13  class Vendedor(Empleado):
14      def __init__(self, nombre, salario, comision):
15          super().__init__(nombre, salario) # Llamada al constructor de Empleado
16          self.comision = comision
17
18      def mostrar_datos(self):
19          super().mostrar_datos() # Llama al método de la superclase
20          print(f"Comisión: {self.comision} €")
21
22
23  # Ejemplo de uso
24  if __name__ == "__main__":
25      emp1 = Empleado("Juan", 2500)
26      vend1 = Vendedor("Ana", 2000, 500)
27
28      print("=== EMPLEADO ===")
29      emp1.mostrar_datos()
30
31      print("\n=== VENDEDOR ===")
32      vend1.mostrar_datos()
```

Resultado

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LAB-B_Docente\Downloads\c6> & C:/Users/LAB-B_Docente/AppData/Local/Programs/Python/Python39-64/Python.exe c:/Users/LAB-B_Docente/Downloads/c6/empleados.py
=== EMPLEADO ===
Nombre: Juan
Salario: 2500 €

=== VENDEDOR ===
Nombre: Ana
Salario: 2000 €
Comisión: 500 €
PS C:\Users\LAB-B_Docente\Downloads\c6>
```

- Empleado es la **superclase o clase padre**.
- El **constructor __init__** recibe nombre y salario como parámetros y los guarda como atributos del objeto.
- El método `mostrar_datos()` muestra la información básica del empleado.
- `Vendedor(Empleado)` indica que Vendedor hereda de Empleado.
- Usa `super().__init__(nombre, salario)` para reutilizar el constructor de la clase padre.
- Agrega un nuevo atributo: `comision`.
- Redefine el método `mostrar_datos()`:
 - Primero llama a `mostrar_datos()` de la clase base (`super().mostrar_datos()`).
 - Luego muestra la comisión del vendedor.
- Se crean dos objetos:
 - `emp1` de tipo `Empleado`.
 - `vend1` de tipo `Vendedor`.
- Se imprime la información usando el método `mostrar_datos()`.

Caso 2: Clases Persona y Cliente:

```

persona.py > ...
1  class Persona:
2      def __init__(self, nombre, edad):
3          self.nombre = nombre
4          self.edad = edad
5
6      def mostrar_datos(self):
7          print(f"Nombre: {self.nombre}")
8          print(f"Edad: {self.edad} años")
9
10
11  class Cliente(Persona):
12      def __init__(self, nombre, edad, numero_cliente):
13          super().__init__(nombre, edad)
14          self.numero_cliente = numero_cliente
15
16      def mostrar_datos(self):
17          super().mostrar_datos()
18          print(f"Número de cliente: {self.numero_cliente}")
19
20
21  if __name__ == "__main__":
22      persona1 = Persona("Luis", 35)
23      cliente1 = Cliente("María", 28, "C1023")
24
25      print("=== DATOS DE PERSONA ===")
26      persona1.mostrar_datos()
27
28      print("\n=== DATOS DE CLIENTE ===")
29      cliente1.mostrar_datos()

```

Resultado

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=== DATOS DE PERSONA ===
Nombre: Luis
Edad: 35 años

=== DATOS DE CLIENTE ===
Nombre: María
Edad: 28 años
Número de cliente: C1023
PS C:\Users\LAB-B_Docente\Downloads\c6>

```

