class Persona:

    def \_\_init\_\_(self, nombre, edad):

        self.\_nombre = nombre

        self.\_edad = edad

    @property

    def nombre(self):

        return self.\_nombre

    @nombre.setter

    def nombre(self, nombre):

        self.\_nombre = nombre

    @property

    def edad(self):

        return self.\_edad

    @edad.setter

    def edad(self, edad):

        self.\_edad = edad

    def \_\_str\_\_(self): # Override = Sobreescribir

        return f"Persona: [ Nombre: {self.\_nombre}, Edad: {self.\_edad}]"

class Empleado(Persona):  # Hereda de la clase Persona

    def \_\_init\_\_(self, nombre, edad, sueldo):

        super().\_\_init\_\_(nombre, edad)

        self.\_sueldo = sueldo

    @property

    def sueldo(self):

        return self.\_sueldo

    @sueldo.setter

    def sueldo(self, sueldo):

        self.\_sueldo = sueldo

empleado1 = Empleado("Ariel", 40, 75000)

print(empleado1.nombre)

print(empleado1.edad)

print(empleado1.sueldo)

# tarea: Encapsular los atributos y agregar los metodos getters and setters

# Crear otro objeto, pasar los datos para nomnbre, edad y sueldo

# Mostrar estos datos, luego modificar y mostrar nuevamente

empleado2 = Empleado("Liliana",38, 70000)

print(empleado2.nombre)

print(empleado2.edad)

print(empleado2.sueldo)

empleado2.nombre = "Natalia"

empleado2.edad = 35

empleado2.sueldo = 75000

print(empleado2.nombre)

print(empleado2.edad)

print(empleado2.sueldo)

# Asistencia NOVIEMBRE 2022 ANDRES WINCKLER

"""

    Definir una clase padre llamada Vehiculo y dos clases hijas llamadas

    Auto y bicicleta, las cuales heredan de la clase padre Vehiculo. La clase

    padre deber tener los siguientes atributos y metodos

    Vehiculo(clase padre)

    -Atributos (color, ruedas)

    -Metodos (\_\_init\_\_(color,rueda) y \_\_str\_\_())

    Auto(clase hija de vehiculo)

    -Atributos(velocidad (km/hr))

    -metodods(--init--(color, ruedas, velocidad) y \_\_str\_\_())

    -Bicicleta(clase hija de vehiculo)

    -Atributos(tipo(urbana/montaña/etc.)

    -Metodos(\_\_init\_\_(color, ruedas, tipo) y \_\_str\_\_())

    crear un objeto de cada clase

    """

class Vehiculo:

    def \_\_init\_\_(self, color, ruedas):

        self.color = color

        self.ruedas = ruedas

    def \_\_str\_\_(self):

        return "color: "+ self.color + " \nruedas: "+ str(self.ruedas)

class Auto(Vehiculo):

    def \_\_init\_\_(self, color, ruedas, velocidad):

        super().\_\_init\_\_(color, ruedas)

        self.velocidad = velocidad

    def \_\_str\_\_(self):

        return super().\_\_str\_\_()+ "\nVelocidad(km/h): "+str(self.velocidad)

class Bicicleta(Vehiculo):

    def \_\_init\_\_(self, color, ruedas, tipo):

        super().\_\_init\_\_(color, ruedas)

        self.tipo = tipo

    def \_\_str\_\_(self):

        return super().\_\_str\_\_()+"\nTipo: "+ self.tipo

# primer objeto clase padre ---> vehiculo

vehiculo = Vehiculo("rojo",4)

print("Objeto clase Vehiculo\n",vehiculo)

#Segundo objeto , el primero de la clase Auto

auto= Auto("Amarillo",6,140)

print("\nobjeto clase Auto\n\n", auto)

#Tercer objeto, primero de la clase Bicicleta

bicicleta = Bicicleta("Negra", 2,"Todo Terreno")

print("\nObjeto clase Bicicleta\n\n",bicicleta)