POO

CLASE

PascalCase
Estructura de programación que permite definir un conjunto de métodos y atributos que describen a un objeto.

ATRIBUTO O PROPS

Es el equivalente a definir una variable dentro de una clase

METODO DE CLASE

Al pertenecer a una clase, siempre va a a recibir como parámetro self.
A través de la palabra reservada self, se accede a métodos y propiedades de la clase

INSTANCIAS

Es un objeto único que a priori ya fue modelado con lo que constituye a una clase. Al crear esa instancia ya le dimos valores.

POLIMORFISMO

Significa que objetos de diferentes clases pueden ser accedidos utilizando la misma interfaz, mostrando un comportamiento distinto

ABSTACCÓN

Consiste en captar las características y funcionalidades que un objeto desempeña y estos son representados en clases por medio de atributos y métodos de dicha clase.

ENCAPSULAMIENTO

Es un mecanismo de protección. Protege los datos de un objeto contra su modificación por quien no tenga derecho a acceder a ellos.

HERENCIA

Es posible crear una clase a partir de otra clase padre. Se puede usar la herencia cuando una clase hija puede aprovechar toda o parte de la funcionalidad de la clase padre.

```python
# ENCAPSULAMIENTO
class Persona():
    def __init__(self, nombre, apellido, dni):
        self.__nombre = nombre #ATRIBUTOS PRIVADOS
        self.__apellido = apellido
        self.__dni = dni

    def getApellido(self):
        return self.__apellido

    def getNombre(self):
        return self.__nombre

    def getDni(self):
        return self.__dni


p1 = Persona("Emilce", "Palma", 44105560)
print("Apellido: ", p1.getApellido())
```

```python
#  HERENCIA
class Alumno(Persona):
    def __init__(self, nombre, apellido, dni, legajo):
        super().__init__(nombre, apellido, dni)
        self.__legajo = legajo

    def getLegajo(self):
        return self.__legajo
```

```python
# POLIMORFISMO
class Auto():
    def acelerar(self):
        print("Me transporto en 4 ruedas")

class Moto():
    def acelerar(self):
        print("Me transporto en 2 ruedas")

class Camion():
    def acelerar(self):
        print("Me transporto en 18 ruedas")


def acelerarVehiculo(vehiculo):
    vehiculo.acelerar()
```

```python
class Vehiculo():
    def __init__(self):
        self.vehiculos = [] #COMPOSICION

    def agregarVehiculo(self, v):
        self.vehiculos.append(v)

    def acelerar(self):
        for vehiculo in self.vehiculos:
            vehiculo.acelerar() #OTRA FORMA DE POLIMORFISMO
```