

Trabajo final del hito_2 de estructura de datos

¿A que se refiere cuando se habla de POO?

POO (**Programación Orientada a Objetos**) es un paradigma de programación que se enfoca en la creación de objetos que interactúan entre sí para llevar a cabo una tarea determinada.

¿Cuáles son los 4 componentes que componen POO?

Abstracción: Es el proceso de identificar las características y comportamientos relevantes de un objeto del mundo real y representarlos en un modelo abstracto.

Encapsulamiento: Es el proceso de ocultar los detalles internos de un objeto y exponer sólo las interfaces públicas.

Herencia: Es un mecanismo que permite crear una nueva clase a partir de una clase existente, heredando sus atributos y métodos y añadiendo nuevos atributos y métodos.

Polimorfismo: Es la capacidad de un objeto de tomar diferentes formas o comportarse de diferentes maneras dependiendo del contexto.

¿Cuáles son los pilares de POO?.

- ▶ Abstracción
- ▶ Encapsulamiento
- ▶ Herencia
- ▶ Polimorfismo

¿Qué es Encapsulamiento y muestre un ejemplo?

es un concepto que hace referencia a la ocultación de los detalles internos de un objeto y a la exposición de sólo las interfaces públicas, es decir, los métodos y atributos que se desean hacer accesibles desde fuera del objeto.

```
class Persona:
    def __init__(self, nombre, edad):
        self.__nombre = nombre
        self.__edad = edad

    def get_nombre(self):
        return self.__nombre

    def set_nombre(self, nombre):
        self.__nombre = nombre

    def get_edad(self):
        return self.__edad

    def set_edad(self, edad):
        self.__edad = edad
```

¿Qué es Abstracción y muestre un ejemplo?

En programación orientada a objetos, la abstracción es el proceso de identificar las características y comportamientos relevantes de un objeto del mundo real y representarlos en un modelo abstracto.

```
class Vehiculo:
    def __init__(self, modelo, marca, color, año, velocidad_maxima):
        self.modelo = modelo
        self.marca = marca
        self.color = color
        self.año = año
        self.velocidad_maxima = velocidad_maxima

    def acelerar(self):
        print("El vehículo está acelerando...")

    def frenar(self):
        print("El vehículo está frenando...")

    def girar(self, direccion):
        print("El vehículo está girando hacia", direccion)
```

¿Que es Herencia y muestre un ejemplo?

la herencia es un mecanismo que permite crear una nueva clase a partir de una clase existente, heredando sus atributos y métodos y añadiendo nuevos atributos y métodos. La nueva clase creada se conoce como subclase o clase derivada, mientras que la clase original se conoce como superclase o clase base.

```
class Coche(Vehiculo):  
    def __init__(self, modelo, marca, color, año, velocidad_maxima, numero_puertas):  
        super().__init__(modelo, marca, color, año, velocidad_maxima)  
        self.numero_puertas = numero_puertas  
        self.capacidad_pasajeros = capacidad_pasajeros  
        self.tipo_combustible = tipo_combustible  
  
    def encender(self):  
        print("El coche está encendiendo...")  
  
    def apagar(self):  
        print("El coche está apagando...")  
  
    def cambiar_velocidad(self, velocidad):  
        print("El coche está cambiando de velocidad a", velocidad)
```

¿Qué es Polimorfismo y muestre un ejemplo?

el polimorfismo se refiere a la capacidad de objetos de diferentes clases de responder a un mismo mensaje o método de forma distinta. En otras palabras, objetos distintos pueden responder a una misma llamada de método con comportamientos diferentes.


```
class Vehiculo:
    def __init__(self, modelo, marca, color, año, velocidad_maxima):
        self.modelo = modelo
        self.marca = marca
        self.color = color
        self.año = año
        self.velocidad_maxima = velocidad_maxima

    def acelerar(self):
        pass

class Coche(Vehiculo):
    def acelerar(self):
        print("El coche está acelerando...")

class Moto(Vehiculo):
    def acelerar(self):
        print("La moto está acelerando...")

class Bicicleta(Vehiculo):
    def acelerar(self):
```

```
coche = Coche("Sedan", "Toyota", "Rojo", 2021, 200, 4, 5, "Gasolina")
moto = Moto("Ninja", "Kawasaki", "Verde", 2022, 300)
bicicleta = Bicicleta("Urbana", "Bianchi", "Negra", 2023, 25)

vehiculos = [coche, moto, bicicleta]

for vehiculo in vehiculos:
    vehiculo.acelerar()
```

¿Que es un ARRAY?

Un array es una estructura de datos que almacena un conjunto de elementos del mismo tipo en una secuencia ordenada, accesible mediante un índice o una clave. Los arrays son una herramienta muy común en la programación y se utilizan para almacenar y manipular datos de manera eficiente.

Cada elemento en un array se identifica por su índice, que es un valor entero que comienza en 0 y se incrementa en 1 para cada elemento adicional.

¿Qué son los paquetes en JAVA?

En Java, un paquete (package) es un mecanismo que permite organizar y estructurar clases relacionadas en una jerarquía de directorios. Los paquetes se utilizan para evitar conflictos de nombres y para facilitar la reutilización de código, ya que las clases de un paquete pueden ser importadas en otros programas para su uso.

¿Cómo se define una clase main en JAVA y muestra un ejemplo?

En java la clase principal main es la que contiene el método que es el punto de entrada de cualquier programa.

```
public class NombreDeLaClase {  
    public static void main(String[] args) {  
        // código del método main  
    }  
}
```