



UNIFRANZ

PROGRAMACION DE SISTEMA EMBEBIDOS

DEFENSA Hito 2

INTEGRANTE:

JOSE YHILMAR VILLCA MAYTA



PARTE TEORICA

- Que es un Sistema embebido
- Mencione 5 ejemplos
- Menciona diferencias o similitudes entre un SO, Sistema movil y un Sistema embebido
- A que se refieren los terminos MCU y MPU
- Cuales son los pilares de POO
- Mencione los componentes en que se basan la POO
- Defina: Multiplataforma, Multiparadigma, Multiproposito y Lenguaje interpretado



A que se refiere con encapsulacion?

```
class Jugador:
    nombreCompleto = ''
    apellidos = ''
    ci = ''
    edad = ''

    def __init__(self, nombres, apellidos, ci, edad):
        self.nombreCompleto = nombres
        self.apellidos = apellidos
        self.ci = ci
        self.edad = edad

    def getNombreCompleto(self):
        return self.nombreCompleto

    def setNombreCompleto(self, nombres):
        self.nombreCompleto = nombres
```

A que se refiere con Herencia?

Clase Padre

```
class Vehicle:
    color = None
    wheels = None

    def __init__(self, color, wheels):
        self.color = color
        self.wheels = wheels

    def __str__(self):
        return f'Color: {self.color} \nLlantas: {self.wheels}'
```

Clase hijo

```
from vehiculo import Vehicle
class Bicycle(Vehicle):
    saddles = None
    chainDrive = None

    def __init__(self, color, wheels, saddles, chainDrive):
        super(Bicycle, self).__init__(color, wheels)
        self.saddles = saddles
        self.chainDrive = chainDrive

    def __str__(self):
        return Vehicle.__str__(self)+f'\nAsiento: {self.saddles} ' \
            f'\nTipo de cadena: {self.chainDrive}'
```

- Que es una clase
- Que es un objeto
- Que es una instancia



PARTE PRACTICA

Llevar el siguiente código JAVA a Python.

```
class Main {  
  
    public static void main(String[] args) {  
  
        System.out.println("Enter two numbers");  
        int first = 10;  
        int second = 20;  
  
        System.out.println(first + " " + second);  
  
        // add two numbers  
        int sum = first + second;  
        System.out.println("The sum is: " + sum);  
    }  
}
```

```
print("Ingrese dos numeros")  
first = 10  
second = 20  
print('N: ' + str(first) + '\nN: ' + str(second))  
  
sum = first + second  
print(f'La suma de {first} + {second} = {sum}')
```

```
Ingrese dos numeros  
N: 10  
N: 20  
La suma de 10 + 20 = 30  
  
Process finished with exit code 0
```


Crear el código JAVA y Python para el siguiente análisis.



Propiedad

name
email
gender
nationality

Comportamiento

Write book
Write a movie
Change nationality
Change email

```
class Persona:
    name = ''
    email = ''
    gender = ''
    nationality = ''

    def __init__(self, name, email, gender, nationality):
        self.name = name
        self.email = email
        self.gender = gender
        self.nationality = nationality

    def __str__(self):
        return f'Nombre: {self.name} \nEmail: {self.email} \nGenero: {self.gender} \nNacionalidad: {self.nationality}'

    def writeBook(self):...

    def writeAMovie(self):...

    def changeGender(self, newGender):...

    def changeEmail(self, newEmail):...
```

```
class Persona{
    private String name;
    private String email;
    private String gender;
    private String nationality;

    public Persona(String name, String email, String gender, String nationality)
    {
        this.name = name;
        this.email = email;
        this.gender = gender;
        this.nationality = nationality;
    }

    void writeBook(){
        System.out.println("Escribes un libro");
    }
    void writeMovie(){
        System.out.println("Escribes una pelicula");
    }
    public void changeNationality(String nationality) {
        this.nationality = nationality;
    }
    public void changeGender(String gender) {
        this.gender = gender;
    }
    void imprimir(){
        System.out.println("Nombre:" + this.name + "\nEmail: " + this.email + "\nGenero: " + this.gender + "\nNacionalidad: " + this.nationality);
    }
}
```

Crear un programa Python que genere los primeros N números de la serie fibonacci.

El programa tiene que leer un valor por consola.

- Ejem: N = 8

Para el valor leído anteriormente, la salida debería ser:

- 0, 1, 1, 2, 3, 5, 8, 13,

```
ingrese un numero: 8
Serie de fibonacci
Serie: 0, 1, 1, 2, 3, 5, 8, 13,
```

```
class Series and Strings:
    pass

    def fibonacci(self, n):
        a = 0
        b = 1
        c = 1
        result = ''

        for i in range(1, n, 1):
            result += str(c) + ', '

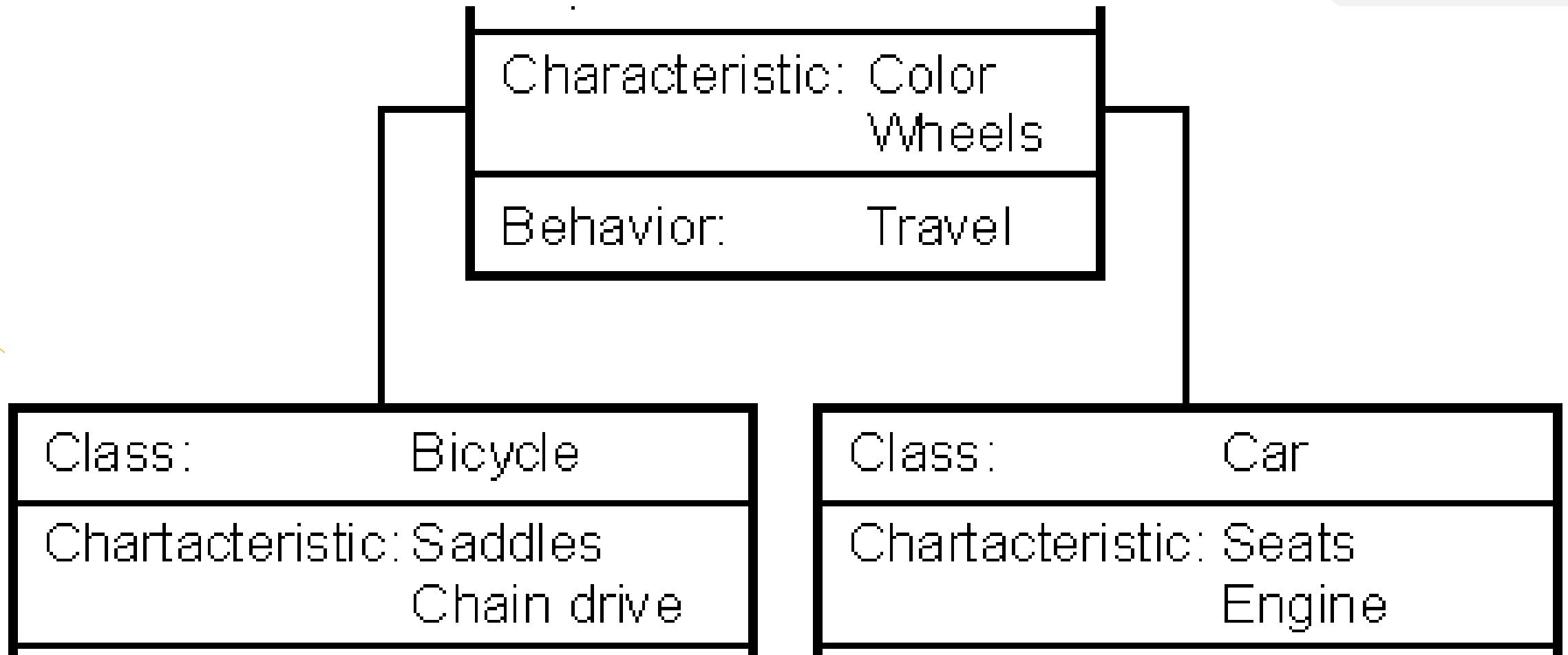
            c = a + b
            a = b
            b = c
        print("Serie: 0,", result)

s1 = Series and Strings()

n = int(input('ingrese un numero: '))
print('Serie de fibonacci')
s1.fibonacci(n)
```

POO - Crear las clases necesarias para resolver el siguiente planteamiento.

FR



Clase padre

Vehicle

```
class Vehicle:
    color = None
    wheels = None

    def __init__(self, color, wheels):
        self.color = color
        self.wheels = wheels

    def __str__(self):
        return f'Color: {self.color} \nLlantas: {self.wheels}'

    def travel(self):
        print('Puede viajar')
```

Resultados

Clase auto

```
Color: Rojo
Llantas: 4 llantas
Asientos: 4 asientas
Motor: electrico
te mueves
Comienzas a acelerar
```

Clase bicicleta

```
Color: Gris
Llantas: 2 llantas
Asiento: Montura de cuero
Tipo de cadena: Mecanica
te mueves
Comienzas a acelerar
```

Clase hijo

Bicycle

```
from vehiculo import Vehicle
class Bicycle(Vehicle):
    saddles = None
    chainDrive = None

    def __init__(self, color, wheels, saddles, chainDrive):
        super(Bicycle, self).__init__(color, wheels)
        self.saddles = saddles
        self.chainDrive = chainDrive

    def __str__(self):
        return Vehicle.__str__(self)+f'\nAsiento: {self.saddles} ' \
            f'\nTipo de cadena: {self.chainDrive}'

    def start(self):
        print('te mueves')

    def accelerate(self):
        print('Comienzas a acelerar')

c1 = Bicycle('Gris', '2 llantas', 'Montura de cuero', 'Mecanica')
print(c1)
```

Car

```
from vehiculo import Vehicle
class Car(Vehicle):
    seats = None
    engine = None

    def __init__(self, color, wheels, seats, engine):
        super(Car, self).__init__(color, wheels)
        self.seats = seats
        self.engine = engine

    def __str__(self):
        return Vehicle.__str__(self)+f'\nAsientos: {self.seats} ' \
            f'\nMotor: {self.engine}'

    def start(self):
        print('te mueves')

    def accelerate(self):
        print('Comienzas a acelerar')

c1 = Car('Rojo', '4 llantas', '4 asientos', 'electrico')
print(c1)
```



Thank You.

