Curso: Programación Orientada a Objetos

Tema:

Metodo especial __str__

Método __str__

Podemos hacer que se ejecute un método definido por nosotros cuando pasamos un objeto a la función print o cuando llamamos a la función str (convertir a string)

¿Qué sucede cuando llamamos a la función print y le pasamos como parámetro un objeto?

```
class Persona:
        def __init__(self, nom, ape):
                self.nombre=nom
                self.apellido=ape

persona1=Persona(“Jose”, “Rodriguez”)
print(persona1)
```

Método __str__

Python nos permite redefinir el método que se debe ejecutar. Esto se hace definiendo en la clase el método especial __str__
En el ejemplo anterior si queremos que se muestre el nombre y apellidos separados por coma, cuando llamemos a la función print el código que debemos implementar es el siguiente :

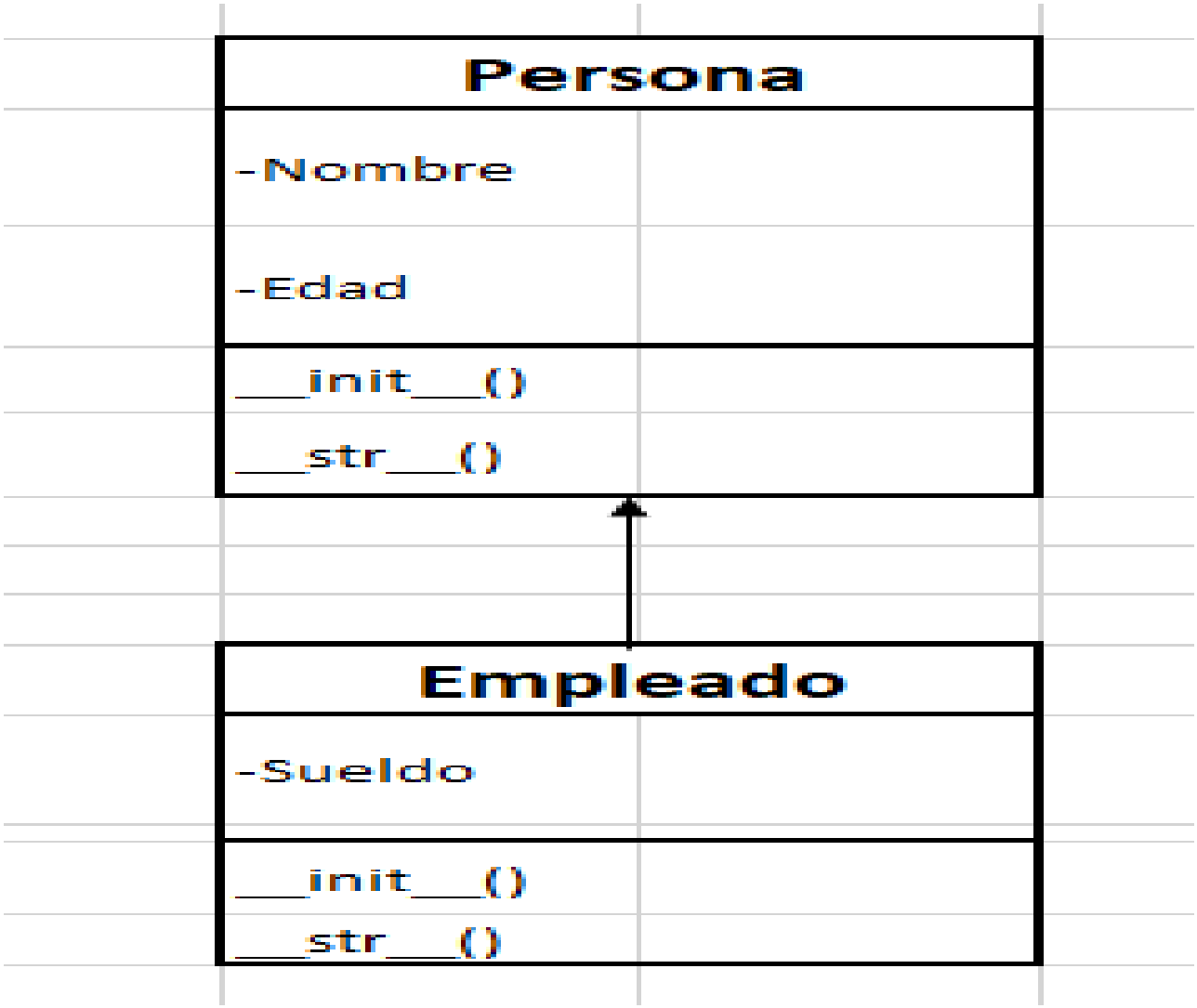
```
class Persona:
        def __init__(self, nom, ape):
                self.nombre=nom
                self.apellido=ape

        def __str__(self)
                return f'{self.nombre} , {self.apellido}'

persona1=Persona("Jose", "Rodriguez")
print(persona1)
```

Método __str__

El método __str__ también se ejecuta si llamamos a la función str y pasamos como parámetro un objeto que tiene definido dicho método:

```
class Persona:
        def __init__(self, nom, ape):
                self.nombre=nom
                self.apellido=ape

        def __str__(self)
                return f'{self.nombre} , {self.apellido}'

persona1=Persona("Jose" "Rodriguez")
persona2=Persona("Ana" "Martinez")
print(persona1 , persona2)
```

Método __str__

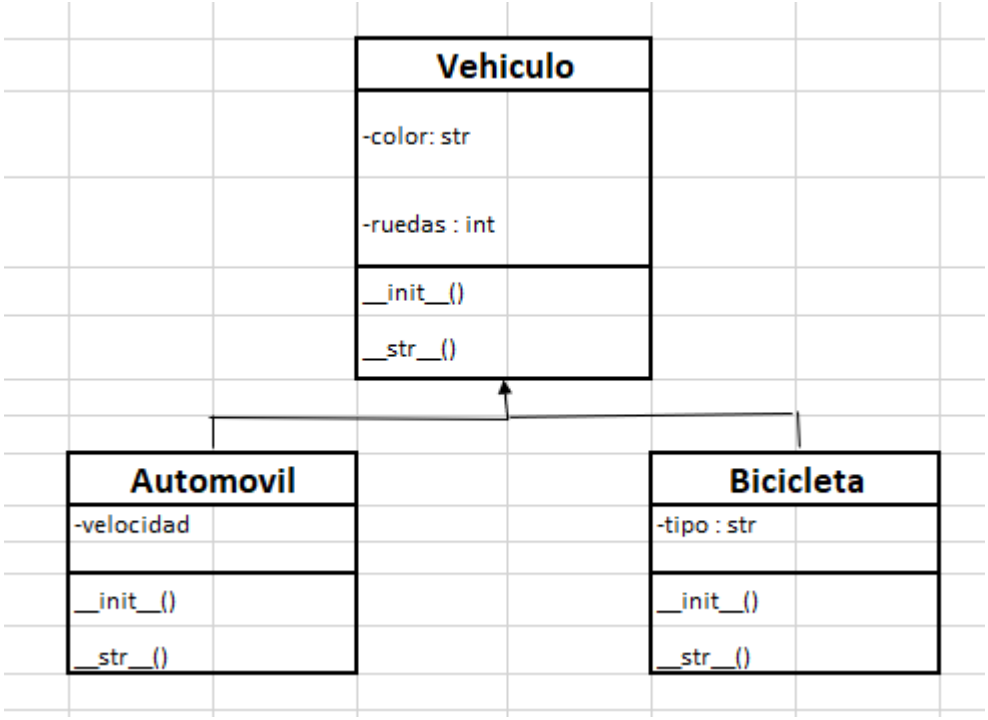


```
class Persona:
    def __init__(self, nombre, edad):
        self.nombre = nombre
        self.edad = edad

    def __str__(self):
        return f'Persona[Nombre: {self.nombre}, Edad: {self.edad}]'

class Empleado(Persona):
    def __init__(self, nombre, edad, sueldo):
        super().__init__(nombre, edad)
        self.sueldo = sueldo

    def __str__(self):
        return f'Empleado[Sueldo: {self.sueldo}] {super().__str__()} ‘

persona1 = Persona('Juan', 28)
print(persona1)

empleado1 = Empleado('Karla', 30, 5000)
print(empleado1)
```

Método __str__

Problemas

1.




```
Creamos un objeto de la clase Vehiculo
color: Rojo
ruedas: 4
Creamos un objeto de la clase hija Coche
color: Azul
ruedas: 4
velocidad: 150
Creamos un objeto de la clase hija Bicicleta
color: Blanca
ruedas: 2
tipo: Urbano
```

2. Definir una clase llamada Punto con dos atributos x e y.
   Crearle el método especial __str__ para retornar un string con el formato (x,y).

3. Declarar una clase llamada Familia. Definir como atributos el nombre del padre, madre y una lista con los nombres de los hijos. Definir el método especial __str__ que retorne un string con el nombre del padre, la madre y de todos sus hijos.

4. Desarrollar un programa que implemente una clase llamada Jugador.
Definir como atributo su nombre y puntaje. Codificar el método especial __str__ que retorne el nombre y si es principiante (menos de 1000 puntos) o experto (1000 o mas puntos).