



## PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

### Preguntas de análisis

Antes de la línea 34:

- A. ¿Cuál es el peso del carro1? Argumente su respuesta  
R//: El carro1 no tiene un peso definido al momento de crearse, por lo tanto, su peso será el valor por defecto que asigna el constructor, que es 1.
- B. ¿Cuál es el motor del carro3? Argumente su respuesta  
R//: El carro3 tampoco tiene un motor asignado explícitamente, por lo que su valor será None, ya que así lo define el constructor.
- C. ¿Cuál es la marca del carro0 (En caso de que este no estuviera comentado en la línea 30)? Justifique lo que sucede  
R//: Si no se le asigna una marca al carro0, su valor será "Ford", porque el constructor lo establece como valor predeterminado.
- D. ¿Qué imprime la línea 35? Justifique
- Modifique el inicializador de la clase de la siguiente manera:

#### **Línea original**

```
- def __init__(self, referencia, peso = 1, marca = "Ford", motor = None):
```

#### **Línea nueva**

```
- def __init__(self, referencia, peso = 4, marca = "Ford", motor = None):
```

R//: La línea 35 ejecuta el método `carro_mas_pesado`, que compara los pesos de los tres objetos carro. Devuelve el atributo de referencia del objeto que tiene el mayor peso. En este caso, imprime el atributo de referencia de carro2.

- E. ¿Qué imprime la línea 35 después del cambio? Argumente su respuesta  
R//: Después de modificar el inicializador, imprimirá "tracker", que corresponde al atributo de referencia del carro1. Esto ocurre porque, tras el cambio, su peso es el más alto entre los objetos.

- F. ¿Cómo modificar el inicializador de Carro, para que pueda recibir indefinido número de parámetros?

R//: Para permitir que el inicializador acepte un número indeterminado de parámetros, se puede usar `*args` o `**kwargs`.

- Con `*args`, los valores adicionales se almacenan en una tupla.
- Con `**kwargs`, los valores adicionales se almacenan como clave-valor en un diccionario. Para asignar estos valores a atributos específicos, se usa un ciclo for:

```
for key, value in kwargs.items():  
    setattr(self, key, value)
```