



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
- B ¿Cuál es el motor del carro3? Argumente su respuesta
- C ¿Cuál es la marca del carro0 (En caso de que este no estuviera comentado en la línea 30)? Justifique lo que sucede
- 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):
```

- E ¿Qué imprime la línea 35 después del cambio? Argumente su respuesta
- F ¿Cómo modificar el inicializador de Carro, para que pueda recibir indefinido número de parámetros?

Solución

- a Ha carro1 se le asigna el peso 1 al ser el peso establecido por defecto en el constructor.
- b Carro3 no se le paso un valor por lo que toma el valor del constructor en este caso seria "Ford"
- c Al carro() no se le paso ningun valor por lo que su marca seria Ford como en el caso anterior por el constructor
- d Imprime el atributo de referencia del carro2 ya que esta linea llama el metodo carro_mas_pesado y se pasan como argumentos los 3 objetos y este metodo devuelve el que tenga el peso mayor, su return devuelve el atributo referencia
- e Imprimirá tracker que es el atributo de referencia de carro1 porque este tendra el mayor peso de todos
- f Dos de las formas para realizar esta modificación en Python seria agregar en el inicializador ya sea la palabra args o la palabra kwargs las cuales cada parámetro adicional a los que pose el constructor los guarda en el caso de args en una tupla y en el caso de kwargs en un diccionario con nombre (clave-valor) en el caso de kwargs también necesitara un for para hacer de forma correcta su función de la siguiente manera

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