



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 no se le definió un peso por ende su peso será de 1 al ser el peso establecido por defecto en el constructor o en este caso al ser Python el método que sirve de constructor.
- b) Carro3 no se le definió un motor cuando se creo por defecto por ende su motor será un Nonetype tal y como se definió en el constructor.
- c) Al carro() no se le definió la marca por ende al igual que en los casos anteriores su marca seria Ford como lo define el constructor o en este caso al ser Python el método que sirve de constructor.
- d) La línea 35 llama al método carro_mas_pesado y da como datos los tres objetos carros, este método compara sus atributos pesos y el que tenga el atributo de peso mas alto devuelve su atributo de referencia, por esto la línea 35 imprime el atributo de referencia del carro2
- e) Imprimirá "tracker" que es el atributo de referencia del objeto carro1, el cual tendría el atributo de peso más alto
- 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)
```