RESPUESTAS

1. **¿Cuál es el peso del carro1? Argumente su respuesta**

Como al carro1 no se le definió un peso al ser creado, a este atributo se le asigna por defecto 1, como esta estipulado en el constructor (\_\_init\_\_)

1. **¿Cuál es el motor del carro3? Argumente su respuesta**

Para el carro3, este tendrá de motor un None, debido a que al momento de crearlo no se le asigno un motor, por tanto toma el valor estipulado en el \_\_init\_\_ que es None

1. **¿Cuál es la marca del carro0 (En caso de que este no estuviera comentado en la línea 30)? Justifique lo que sucede**

El objeto con referencia carro0 no se crearía, debido a que se necesita pasar el valor de referencia, para que se cree correctamente, por ende el programa nos arroja error si se intenta ejecutar. Por otro lado si se llegara a pasar el valor del atributo “referencia”, al objeto se le definiría pro defecto al atributo marca = “Ford”

1. **¿Qué imprime la línea 35? Justifique**

La línea 35 imprimiria “Sandero Stepway” ya que al usar el método estático carro\_mas\_pesado(), se busca el objeto de tipo Carro con el valor mayor en el atributo \_peso, retornando la referencia de este.

**- Modifique el inicializador de la clase de la siguiente manera:**

**Gráfico

Descripción generada automáticamente**

1. **¿Qué imprime la línea 35 después del cambio? Argumente su respuesta**Luego de la modificación, el código imprimiría “Tracker” debido a que el método estático carro\_mas\_pesado(), busca el objeto de tipo Carro con el mayor valor en el atributo \_peso, y en este caso como a carro1 no se le asigno valor en \_peso, toma el de por defecto que es 4, siendo mayor al del carro2 que es de 3 el cual fue el que se imprimió antes del cambio. Siendo así que la función en este caso retorna “Tracker” que es el atributo \_referencia de carro1.
2. **¿Cómo modificar el inicializador de Carro, para que pueda recibir indefinido número de parámetros?**

Para que pueda recibir indefinido numero de parámetros, el inicializador de Carro debe estar compuesto por \*args (para recibir cualquier tipo de parámetros) y \*\*kwargs (Para recibir parámetros tipo clave = valor, que se guardan como diccionarios), los nombres args y kwargs son por convención, pero se puede poner cualquier otro nombre si y solo si se coloca \*(en caso de args) o \*\*(en caso de kwargs) antes del nombre.