

Taller python programación orientada a objetos

Santiago Barrientos Medina

Universidad Nacional de Colombia

Programación orientada a objetos

Jaime Alberto Guzmán Luna

Grupo 2

Facultad de Minas

Ingeniería de sistemas e informática

SOLUCION

A). el peso del carro 1 es una tonelada porque en los valores por defecto de los parámetros del init el del peso es uno, el carro uno al ser creado no entrega valores , deja esto vacio, asi que al no tenerlos , se le asignan los que tiene el init por defecto que en este caso es una tonelada

B).el motor del carro 3 es igual a nulo o None, esto por la razón de que al ser creado el carro 3, este no entrego un valor para la variable motor del init , asi que se le asigna la que por defecto tiene el parámetro motor en el init , que es None

C). la marca del carro 0 es igual a “Ford” , esto por el hecho de que el carro 0 al ser creado no entrega valores asi que se le asignan los que hay por defecto en el init y en el caso especifico de la marca es “Ford

D). la línea 35 imprime “Sander Stepway”, pero por que?. Porque si nos fijamos bien, se crean todos y cada uno de los carros, y luego se le entrega una lista de ellos a un método que busca cual es el mas pesado, el método hace un recorrido por elemento y va verificando el peso con una variable de referencia que tomo del carro en posición 0, si el carro en el que esta la iteración es mas pesado que la variable de referencia, pues se cambia la referencia por el mas pesado y se cambia el peso también, en la lista que se le entrega el mas pesado es el sander y es exactamente lo que devuelve este método : “Sander Stepway”

E). lo que imprime ahora el código con el cambio que le hicimos es “Tracker” , esto porque como lo explicamos anteriormente, el método itera en la lista y lo que hace el cambio que hace que la impresión sea diferente , es el hecho de que ahora el valor por defecto del parámetro peso en el init es 4 toneladas, de los 3 carros creados el único que recibe el peso por defecto es el carro numero 1 y como los otros dos carros pasan valores menores , este es el mas pesado

F). para que el inicializador de carro pueda recibir indefinido numero de parámetros, podemos usar la notación *args de esta manera:

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

esto hace que podamos seguir pasando valores y estos se almacenen en args, podemos seguir hasta donde deseemos y ellos estarán en args

también se me ocurre la manera **kwargs que funciona de manera parecida a lo anterior pero con claves y con valores que se pasarían así clave:valor