

CSC148 Worksheet 6 Solution

Hyungmo Gu

April 19, 2020

Question 1

- The two classes already defined are: Vehicle and SuperDuperManager

The additional classes required to create for this exercise are: Car, Helicopter and MagicCarpet

Car, Helicopter and MagicCarpet are child classes of Vehicle.

Question 2

- a. The following are attributes possessed by all vehicles

- type
- initial_position
- moves_to
- move_diagonally
- fuel_usage

Correct Solution:

The following are attributes possessed by all vehicles

- position
- fuel

- b. No. Referencing the following code in *worksheet_6_starter_code.py*,

```

1  class Vehicle:
2      ...
3
4      def __init__(self, initial_fuel: int,
5                    initial_position: Tuple[int, int]) -> None:
6          ...
7
8          self.fuel = initial_fuel
9          self.position = initial_position
10

```

we can come up with the following examples.

- Vehicle(100, (10,20))
- Vehicle(50, (5, 10))

Here, we can see the two vehicles have different value of fuel and initial position.

- c. *fuel_needed* not implemented because each child classes have different fuel consumption rate, and the method is to be defined by the child classes by overriding it.
- d. The following methods must be defined in each of its subclasses

- Car
 - fuel_needed
 - move
- Helicopter
 - fuel_needed
 - move
- MagicCarpet
 - __init__
 - move

Correct Solution:

- Car
 - __init__
 - * Necessary because the parameter *position* must be set as optional
 - * Necessary because *self.position* must default to (0,0) if the argument of *position* not given.
 - fuel_needed
 - * Necessary because vehicle uses fuel

- * Necessary because needs to define the fuel cost based on it not being able to moving diagonally.
- Helicopter
 - `__init__`
 - * Necessary because the parameter *position* must be set as optional
 - * Necessary because *self.position* must default to (3,5) if the argument of position not given.
 - `fuel_needed`
 - * Necessary because vehicle uses fuel
 - * Necessary because needs to define the fuel cost based on it being able to move diagonally.
- MagicCarpet
 - `__init__`
 - * Necessary to set the parameters *initial_fuel*, *initial_position* as optional
 - * Necessary to randomize the value of *self.position*.
 - `move`
 - * Necessary to set the parameters *new_x* and *new_y* as optional.
 - * Necessary to randomize the value of new position.

Question 3

```

1      """
2      Initializing SuperDuperManager:
3      >>> s = SuperDuperManager()
4      >>> s._vehicles
5      {}
6
7      Adding Vehicles:
8      >>> s.add_vehicle('Car', '1', 100)
9      >>> s._vehicles['1'].__class__.__name__
10     'Car'
11     >>> s.add_vehicle('Helicopter', '1', 100)
12     >>> s._vehicles['1'].__class__.__name__
13     'Car'
14
15     >>> s.add_vehicle('Helicopter', '2', 100)
16     >>> s._vehicles['2'].__class__.__name__
17     'Helicopter'
18
19     >>> s.add_vehicle('UnreliableMagicCarpet', '3', 100)
20     >>> s._vehicles['3'].__class__.__name__
21     'UnreliableMagicCarpet'

```

```

22
23 Moving Vehicle:
24 >>> s._vehicles['1'].position
25 (0,0)
26 >>> s.move_vehicle('1', 1, 1)
27 >>> s._vehicles['1'].position
28 (1,1)
29
30 >>> s._vehicles['2'].position
31 (3,5)
32 >>> s.move_vehicle('2', 1, 1)
33 >>> s._vehicles['2'].position
34 (4,6)
35
36 >>> s._vehicles['3'].position
37 (4,8)
38 >>> s._vehicles['3'].position
39 (12,4)
40 >>> s.move_vehicle('3', 1, 1)
41 >>> s._vehicles['3'].position
42 (100,100)
43
44 Get Vehicle Position:
45 >>> s.get_vehicle_position('1')
46 (1,1)
47
48 >>> s.get_vehicle_position('2')
49 (4,6)
50
51 >>> s.get_vehicle_position('3')
52 (50,200)
53
54 Get Vehicle Fuel:
55 >>> s.get_vehicle_fuel('1')
56 98
57
58 >>> s.get_vehicle_fuel('2')
59 99
60
61 >>> s.get_vehicle_fuel('2')
62 100
63 """

```

Question 4

a. The instance attribute *id_* is used to keep track of vehicles.

The type of the instance attribute is string.

b. The vehicles are initialized in class *SuperDuperManager*'s *add_vehicle* method.

Question 5

Question 6

Question 7

Question 8