



## 3-3 编程练习

自定义一个交通工具类(Vehicle)，并根据提示对该类进行进一步封装，使其拥有工具类型、速度、体积等属性值。通过自定义实例方法实现交通工具的前移、速度设置、获取当前速度、加速行驶、减速行驶、实例信息展示、实例类型判别等功能。

效果图如下：

```
我的所属类型为: SUV, 速度: 20km/h, 体积: (3.6, 1.9, 1.75)
我已向前移动了50米
我的时速为: 40km/h
我的速度由40km/h提升到了50 km/h
我的速度由50km/h下降到了35 km/h
类型匹配
```

### 任务

- 1、自定义一个交通工具类(Vehicle)
- 2、设置类属性trans\_type（默认值为SUV）和实例属性速度speed（int类型，单位为km/h）、体积size（tuple类型，单位为米。）
- 3、自定义方法 show\_info()，打印实例的所属类型和速度、体积的值；
- 4、自定义实例方法如下：
  - (1) 定义move()方法，实现打印"我已向前移动了50米"
  - (2) 定义set\_speed(new\_speed)方法，设置对应实例的速度为new\_speed km/h
  - (3) 定义get\_speed()方法，如果(2)中设置了速度值则打印出来，打印格式为"我的时速为: 设置的速度值 km/h"
  - (4) 定义speed\_up()方法，设置每次调用时实例的速度都增加10km/h，并打印"我的速度由xx km/提升到了xx km/h"
  - (5) 定义speed\_down()方法，设置每次调用时实例的速度都降低15km/h，并打印"我的速度由xx km/下降到了xx km/h"
- 5、自定义方法 transport\_identify()，判断实例的trans\_type属性是否为SUV。若是则打印"类型匹配"，反之则打印"类型不匹配"
- 6、初始化实例对象tool\_1，并根据上述效果图调用对应方法

### 任务提示

类的初始化方法中所传参数size是元组类型，直接传入实例的长，宽，高即可，如size=(10,10,10)

hello.py

```
1 class Vehicle(object):
2     # 自定义Vehicle类属性
3
4     # 自定义实例的初始化方法
5
6     # 自定义实例方法show_info，打印实例的速度和体
7
8     # 自定义实例方法move，打印"我已向前移动了50米"
9
10    # 自定义实例方法set_speed，设置对应的速度值
11
12    # 自定义实例方法get_speed，打印当前的速度值
13
14    # 自定义实例方法speed_up，实现对实例的加速
15
16    # 自定义实例方法speed_down，实现对实例的减速
17
18    # 自定义实例方法transport_identify，实现对实例
19
20 if __name__ == "__main__":
21     tool_1 = Vehicle(20, (3.6, 1.9, 1.75))
22
23     # 调用实例方法 打印实例的速度和体积
24
25     # 调用实例方法 实现实例的前移
26
27     tool_1.set_speed(40)
28     # 调用实例方法 打印当前速度
29
30     # 调用实例方法 对实例进行加速
31
32     # 调用实例方法 对实例进行减速
33
34     # 调用实例方法 判断当前实例的类型
35
```



重置代码

帮助

提交