1 python程序设计#1作业

班级：307

学号：2021211138

姓名：陈朴炎

1.1 作业题目

每人独立设计并实现一个小型python程序（功能不限），代码需要涉及：class类、对象实例化、继承（分别定义父类和子类）、对象方法（self参数）、类方法（@classmethod）、静态方法（@staticmethod）、对象属性、类属性、多态。

1.2 作业内容

```
class Driveable:    # 启动及停止
    def start_engine(self):
        pass

    def stop_engine(self):
        pass

class Entertainment:    # 娱乐功能
    def play_music(self):
        pass

    def operate_air_conditioner(self):
        pass

class Vehicle:      # 父类 交通工具
    def __init__(self, name, speed):
        self.name = name
        self.speed = speed

    def description(self):
        return f"This is a {self.name} and its maximum speed is {self.speed}
km/h."

    @classmethod
    def create_vehicle(cls, name, speed):
        return cls(name, speed)

    @staticmethod
    def vehicle_type():
        return "Generic Vehicle"

class Car(Vehicle, Driveable, Entertainment):
    # 实现Car的启动和停止
    def start_engine(self):
```

```
        return "Starting the car's engine."

    def stop_engine(self):
        return "Stopping the car's engine."

    # 让Car可以播放音乐
    def play_music(self):
        return "Playing music in the car."
    # 让Car可以开关空调
    def operate_air_conditioner(self):
        return "Operating the air conditioner in the car."

    def description(self):
        return f"This is a {self.name} car and its maximum speed is {self.speed}
km/h."

    # 返回交通工具类型
    @staticmethod
    def vehicle_type():
        return "Car"

class Bicycle(Vehicle, Driveable):
    def start_engine(self):
        return "Pedaling the bicycle to start."

    def stop_engine(self):
        return "Stopping pedaling to stop the bicycle."

    def description(self):
        return f"This is a {self.name} bicycle and its maximum speed is
{self.speed} km/h."

    @staticmethod
    def vehicle_type():
        return "Bicycle"

if __name__ == '__main__':
    car = Car.create_vehicle("Wenjie", 180)
    bicycle = Bicycle.create_vehicle("Mountain Bike", 30)

    vehicles = [car, bicycle]

    for vehicle in vehicles:
        print(f"{vehicle.description()} - Type: {vehicle.vehicle_type()}")
        if isinstance(vehicle, Driveable):
            print(f"Drive: {vehicle.start_engine()} / {vehicle.stop_engine()}")
        if isinstance(vehicle, Entertainment):
            print(f"Entertainment: {vehicle.play_music()} /
{vehicle.operate_air_conditioner()}")
```

```
This is a Wenjie car and its maximum speed is 180 km/h. - Type: Car
Drive: Starting the car's engine. / Stopping the car's engine.
Entertainment: Playing music in the car. / Operating the air conditioner in the
car.
This is a Mountain Bike bicycle and its maximum speed is 30 km/h. - Type: Bicycle
Drive: Pedaling the bicycle to start. / Stopping pedaling to stop the bicycle.
```

1.3 代码说明

Driveable 类和 Entertainment 类：

这两个类本质是两个接口，用于判断该交通工具是否可以驾驶和提供娱乐功能。 Driveable 类里的两个方法：start_engine 和 stop_engine，表示交通工具的启动和停止。 Entertainment 类里的两个方法：play_music 和 operate_air_conditioner，表示交通工具提供的音乐播放和开关空调操作功能。

Vehicle 类：

是交通工具父类，之后的Car类和Bicycle都要继承这个类。 init 方法可以初始化交通工具的名称和最大速度。 description 方法返回交通工具的描述信息。 create_vehicle 用于创建交通工具对象。 vehicle_type 是一个静态方法，用于返回交通工具的类型信息。

Car 类：

Car 是 Vehicle 类的子类，它继承了 Driveable 和 Entertainment 两个类，表示Car可以启动、停止，也可以开关空调、播放音乐。 start_engine 和 stop_engine 方法实现了启动和停止汽车引擎的具体功能。 play_music 和 operate_air_conditioner 方法提供了在汽车上的音乐播放和空调操作功能。 description 和 vehicle_type 方法分别重写了父类的方法，使得Car可以描述它自己的类型信息。

Bicycle 类：

Bicycle 也是 Vehicle 类的子类，继承了 Driveable 类，但是它不能打开空调，也不能播放音乐。 start_engine 和 stop_engine 两个方法显示了，自行车需要通过踏板驱动。 description 和 vehicle_type 方法分别重写了父类的方法，以提供自行车特定的描述和类型信息。

主函数部分：

在主函数部分，我创建了一个car类和bicycle类的对象实例，根据它们是否支持Driveable、Entertainment类，进行函数调用及输出