

第15课 租借共享单车——项目实训

根据题目需要，一个单车类Bike，用来实例化单车对象，绑定单车的属性。一个平台管理类Manage，将查询车辆，共享车辆，租借车辆，归还车辆功能封装到该类中。所以我们需要用到初始化方法__init__，让实例被创建时自动获得这些属性。

欢迎使用共享单车租借系统

- 1.查询所有车辆
 - 2.共享车辆
 - 3.租借车辆
 - 4.归还车辆
 - 5.退出系统
- 请输入数字选择对应功能：

#####

同时我们需要实现上图中的 45个功能，那就得定义另一个类 Manage，该类含有五个方法（含主菜单）：

- ①、主菜单：def menu()
- ②、查询车辆：info_bike()
- ③、共享车辆：add_bike()
- ④、租借车辆：lease_bike()
- ⑤、归还车辆：revert_bike()

```
#单车类
class Bike:
    def __init__(self):
        #单车初始化

class Manage:
    #系统菜单
    def menu(self):

    #单车信息
    def info_bike(self):

    #共享单车
    def add_bike(self):

    #租借车辆
    def lease_bike(self):

    #归还车辆
    def revert_bike(self):
```

#####

而为什么要分为两个类呢，主要是我们 Bike 类是为了让程序每次实例时初始化它的属性值，Manage 类是为了实现我们的功能，二者区分开来，有利于代码的易读和整洁性

#####

好了，我们再看回代码，Bike 类不难理解，一个初始化函数 __init__ 包含了四个实例化属性，而另一个函数 __str__，这个函数也是实例化后就执行，并返回 return 后面所带的值

#####

```
class Bike:
    # 初始化方法 no代表车辆编号、age代表车辆年限、
    # state代表车辆状态，0代表待租借，1代表租借中
    def __init__(self, NO, age, state=0):
        self.NO = NO
        self.age = age
        self.state = state

    def __str__(self):
        if self.state == 0:
            status = '待租借'
        else:
            status = '租借中'
        return '车辆编号%d 已经运行%d年，车辆状态：%s' % (self.NO, self.age, status)
```

```
bike = Bike(1001, 2)
print(bike)
```

#####

接下来我们分析一下 Manage 类

#####

```
16 class Manage:
17     #定义一个列表，用来存储所有的车辆
18     bike_list = []
19
20     def __init__(self):
21         bikeA = Bike(1001,2)
22         bikeB = Bike(1002,2)
23         bikeC = Bike(1003,1)
24         self.bike_list.append(bikeA)
25         self.bike_list.append(bikeB)
26         self.bike_list.append(bikeC)
```

#####

这里也定义了一个初始化函数，其中 bikeA、bikeB、bikeC 是实例化对象，实例化了类 Bike，即这三个是默认存在的。接下来的三个 append 语句的结果是 self.bike_list=[bikeA, bikeB, bikeC]，即实例化对象的属性.bike_list含有这三个值

#####

```

27 #系统菜单
28 def menu(self):
29     print("欢迎使用共享单车租借系统\n")
30     while True:
31         print('1.查询所有车辆\n 2.共享车辆\n 3.租借车辆\n 4.归还车辆\n 5.退出系统\n')
32         select = int(input('请输入所选功能对应得数字: '))
33         if select == 1:
34             #单车信息
35             self.info_bike()
36         elif select == 2:
37             #共享单车
38             self.add_bike()
39         elif select == 3:
40             #租借车辆
41             self.lease_bike()
42         elif select == 4:
43             #归还车辆
44             self.revert_bike()
45         elif select == 5:
46             #退出系统
47             print('期待您的下次使用! 祝您生活愉快! ')
48             break

```

#####

接下来的 menu 方法，主要是提供了我们对于功能的选择，根据我们的选择跳转执行相对应的方法

#####

```

50 #单车信息
51 def info_bike(self):
52     for bike in self.bike_list:
53         print(bike)

```

#####

info_bike 方法中，for bike in self.bike_list，根据我们之前学的知识，这一步是为了遍历上方说的 self.bike_list 列表，而因为实例化 Bike类后会执行 __str__ 函数，即会有返回值，所以可以显示目前所有的车辆信息

#####

```

54 #共享单车
55 def add_bike(self):
56     new_NO = int(input('请输入车辆编码'))
57     new_age = int(input('请输入车辆使用年限'))
58
59     new_bike = Bike(new_NO,new_age)
60     self.bike_list.append(new_bike)
61     print('车辆共享成功! ')

```

#####

add_bike方法中，代码比较简单，是让我们手动输入后，把这些值传入后实例化类 Bike，并在 self.bike_list 中添加

#####

```

77     def select_bike(self,NO):
78         #遍历整个自行车列表
79         for bike in self.bike_list:
80             #如果存在输入编号与车辆列表中的编号一致
81             if bike.NO == NO:
82                 #返回该车辆信息
83                 return bike
84

```

#####

select_bike方法，我们依旧是遍历我们的 self.bike_list列表，相信同学们没有忘记 self.bike_list=[bikeA, bikeB, bikeC]，而 bikeA, bikeB, bikeC 是我们的实例化对象，所以 bike.NO 实际上就是我们的实例化属性 NO，当实例化属性NO 等于我们外部传入的值NO 时，我们就返回当前匹配的实例化对象。

#####

```

62     #租借车辆
63     def lease_bike(self):
64         lease_NO = int(input("请输入租借的车辆编号"))
65         res = self.select_bike(lease_NO)
66         if res != None:
67             if res.state ==1:
68                 #车辆租借中
69                 print("你来晚了，车被租走了")
70             else:
71                 #车辆待租借
72                 print("租借成功，欢迎您使用绿色出行")
73                 res.state =1
74         else:
75             print("该车辆不存在")

```

#####

lease_bike方法，手动输入后，调用我们上面说到的 select_bike 方法，如果返回值不为空，则继续下一步，再判断我们的 state 属性是否为 1，是则无法租借，否则可以租借车辆，并把当前车辆的 state 值改为 1（当该车辆未归还则无法租借）

#####

```

85         #归还车辆
86     def revert_bike(self):
87         #输入归还车辆的编号
88         revert_NO = int(input("请输入归还的车辆编号"))
89         #进行查询
90         res = self.select_bike(revert_NO)
91         #车辆存在
92         if res != None:
93             #租借中，还车成功
94             if res.state == 1:
95                 #还车成功
96                 print("还车成功，期待下次使用")
97                 res.state = 0
98             #未租借，等待租借
99             else:
100                 #车辆等待租借
101                 print("车辆整备完成，等待租借")
102         #车辆不存在
103         else:
104             print("该车辆不存在,想必您是输错了")
105     def __init__(self):

```

#####

最后一个方法 revert_bike，依旧是传入select_bike方法进行判断，根据返回值判断该车辆是否之前已经存在在这个系统中，有则继续判断 state 值，不为 0 则归还成功

#####

最后我们实例化类 Manage，再调用它的 menu 方法执行开始程序的运行