

تمارین بیشتر

جون هرکی دوست دارید اول خودتوت بزنی بعد برید سراغ جوابا (خ)

یه چیزی هم هست بحث بی معنی بودن سوال هاست مثلاً همین سوال یک به نظرتون ناقص و بی معنی میاد و این درسته چرا ؟ چون که ما مباحث ارث بری و پیشرفته شعی گرای میثل تعریف کردن یک متغیر که تمام ابجکت ها روی آن تاثیر بزارن رو نخوندیم پس حواستون باشه

1- سیستم کرایه یک نمایشگاه ماشین رو پیاده سازی کنید که :

نام فروشگاه و لوکیشن برای همه ماشین ها یکی باشه

موقع ثبت ماشین : قیمت کرایه , اسم ماشین , مقدار کارکرد به کیلومتر , تعداد روز در دسترس بودن , سال تولید , و ... به آن ماشین اختصاص داده شود
کاربران باید توان کرایه , تحویل , گرفتن اطلاعات همه ماشین ها و یا یک ماشین خواص , و ... رو داشته باشن

انتظار میره با پرینت شدن ابجکت ها کل اطلاعات ماشین ها برگرده

اسم تمام ماشین های کرایه شده و اسم ماشین های کرایه شده در حال حاضر باید در یک لیست ذخیره و آیتم های لیست (نه خود لیست) چاپ بشن در آخر

توی همین سوال میتونستم یک کلاس مشتری هم بسازی و بگیرم اگر مشتری پول داشت ماشین رو کرایه کنه ولی چون کلاس مشتری و ماشین باید به من متصل بشن (از ارث بری نیاز هست استفاده بشه) نگفتم

2-سیستم رزرو یک هتل رو پیاده سازی کنید که :

نام هتل و قیمت پایه اتاق ها و تعداد اتاق ها مشترک باشن بین تمام رزرو ها

موقع ثبت رزور باید اول یک پیش رزور انجام دهید : نام کلاینت و نام خانوادگی کلاینت و کد ملی کلاینت و تلفن کلاینت و تعداد روز رزور و تعداد اتاقی که میخان رو بگیرین

کاربر بعد از ثبت نام میتواند با دادن مقدار پول که چک شود در مقایسه با قیمت پایه اتاق رزرو کند و تعداد اتاق های رزور نهایی شده را افزایش دهد

کاربر میتواند اطلاعات اتاقش را بخونه

تمام اتاق های رزور شده رو در آخر چاپ کنید

انتظار میره با رشته شدن ابجکت ها فقط نام صاحب اتاق برگرده

3-سیستم انبار یک فروشگاه رو بنویسید :

اول تعدادی محصول با گرفتن اسم قیمت تعداد و تاریخ انقضا درست کنید

در آخر هم لیست تمام محصولات وارد شده رو برگردونید و تعداد محصولات رو

```

rented = []

Codiumate: Options | Test this class
class CarShop :
    shop_name = 'borqeii'
    loc = 'qom'

Codiumate: Options | Test this method
def __init__(self , i_price , i_name , i_used , i_avaible , i_created ):
    print("welcome to registration...")
    self.name = i_name
    self.price = i_price
    self.used = i_used
    self.avaible = i_avaible
    self.created = i_created

    print(f"{self.name} car is now registered")

Codiumate: Options | Test this method
def rent(self):
    print("renting now ...")
    rented.append(self.name)
    print(f"{self.name} is rented for you now ")

Codiumate: Options | Test this method
def withdraw(self):
    print("withrawing your car ...")
    rented.remove(self.name)
    print(f"{self.name} is now withrawed from your wallet")

Codiumate: Options | Test this method
def info(self):
    print(f"getting informations about {self.name} car")
    print(f'its name is {self.name} and its price is {self.price} its used for {self.used} and its creation date is {self.created}')

def __str__(self):
    return (f'its name is {self.name} and its price is {self.price} its used for {self.used} and its creation date is {self.created}')
```

```

bmw = CarShop(1000 , 'BMW' , 1000 , '12-22' , '2022')
benz = CarShop(5000 , 'Meybach' , 2000 , '23-5' , '2024')

bmw.rent()
bmw.info()
bmw.withraw()

benz.rent()
benz.info()

for i in rented :
    print(f'{i} is still rented')
```

```

welcome to registration...
BMW car is now registered
welcome to registration...
Meybach car is now registered
renting now ...
BMW is rented for you now
getting informations about BMW car
its name is BMW and its price is 1000 its used for 1000 and its creation date is 2022
withrawing your car ...
BMW is now withrawed from your wallet
renting now ...
Meybach is rented for you now
getting informations about Meybach car
its name is Meybach and its price is 5000 its used for 2000 and its creation date is 2024
Meybach is still rented
```

```
rented = {}
rented_count = 0
```

Codiumate: Options | Test this class

```
class Hotel:
```

```
    hotel_name = 'persian'
    price = 2000
    room_count = 100
```

Codiumate: Options | Test this method

```
def __init__(self, c_name, c_lastname, c_id, c_phone, c_days, c_rooms):
```

```
    print("hello to our hotel ...")
```

```
    self.name = c_name
```

```
    self.lastname = c_lastname
```

```
    self.id = c_id
```

```
    self.phone = c_phone
```

```
    self.days = c_days
```

```
    self.rooms = c_rooms
```

```
    print(f"{self.name} with {self.lastname} with {self.id} with {self.phone} is registered for {self.days} days for {self.rooms} rooms")
```

```
    for i in range(1, self.rooms):
```

```
        rented.update({f'room{i}': {'name': self.name, 'lastname': self.lastname, 'phone': self.phone}})
```

Codiumate: Options | Test this method

```
def rent(self, wallet):
```

```
    global rented_count
```

```
    if self.room_count < rented_count :
```

```
        print("hotel rooms are filled , please come back later ...")
```

```
    else :
```

```
        if wallet > self.price:
```

```
            rented_count += 1
```

```
            print("The room is now rented finally for you, enjoy your trip! \n")
```

```
            print('your rooms are : \n')
```

```
            for i in range(1 , self.rooms):
```

```
                print( rented.get(f'room{i}'))
```

Codiumate: Options | Test this method

```
def show_info(self):
```

```
    print("getting your rooms info ...")
```

```
    print(f"{self.name} with {self.lastname} with {self.id} with {self.phone} is registered for {self.days} days for {self.rooms} rooms")
```

```
def __str__(self):
```

```
    return self.name
```

```
c1 = Hotel('erfan', 'safarali', '370270', '09016492293', 20, 3)
c1.rent(1000000)
```

```
c2 = Hotel('mamadi', 'gholi', '3722220', '09373260257', 40, 6)
c2.show_info()
```

```
print(str(c2))
```

```
hello to our hotel ...
```

```
erfan with safarali with 370270 with 09016492293 is registered for 20 days for 3 rooms
```

```
The room is now rented finally for you, enjoy your trip!
```

```
your rooms are :
```

```
{'name': 'erfan', 'lastname': 'safarali', 'phone': '09016492293'}
```

```
{'name': 'erfan', 'lastname': 'safarali', 'phone': '09016492293'}
```

```
hello to our hotel ...
```

```
mamadi with gholi with 3722220 with 09373260257 is registered for 40 days for 6 rooms
```

```
getting your rooms info ...
```

```
mamadi with gholi with 3722220 with 09373260257 is registered for 40 days for 6 rooms
```

```
mamadi
```

```

test.py > ...
Codumate: Options | Test this class
1 class Shop:
2     p_list = []
3     p_count = 0
4
5     Codumate: Options | Test this method
6     def init(self, p_name, p_price, p_count, p_expire_date):
7         self.product_name = p_name
8         self.product_price = p_price
9         self.product_count = p_count
10        self.product_expiration_date = p_expire_date
11
12        print(f'{self.product_name} with {self.product_price} with {self.product_count} with {self.product_expiration_date} is registered')
13
14        Shop.p_list.append(self.product_name)
15        Shop.p_count += 1
16
17    p1 = Shop()
18    p1.init('pofak', 2000, 100, '2024')
19    p2 = Shop()
20    p2.init('chips', 5000, 200, '2024')
21
22    print(Shop.p_list)
23    print(Shop.p_count)
24

```

```

pofak with 2000 with 100 with 2024 is registered
chips with 5000 with 200 with 2024 is registered
['pofak', 'chips']
2

```

در این مثال اخرشو دیدید نترسید . میشه که ابجکت رو ساخت ولی بعدش با متود `__init__` نیاد داندن اینیت رو صدا میزنه مقدار داد