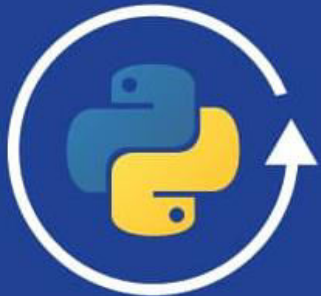


“Tự Học Lập Trình Python “

Bài 42.3: Lập trình Hướng đối tượng Part 2

→ Method ←

Object-Oriented Programming



6. Chữa bài tập oop 1:

Tạo 1 class tên là `sinhvien`, với các thuộc tính : `hoten`, `ma_sinh_vien`, `que_quan`

Nhập vào 3 object là `sv1`, `sv2`, `sv3`

Xuất ra tên `sv1` và quê `sv1`

Xuất ra quê của cả 3 `sv1`, `sv2`, `sv3`

```
class sinhvien:
    def __init__(self, hoten, ma_sinh_vien, que_quan):
        self.hoten = hoten
        self.ma_sinh_vien = ma_sinh_vien
        self.que_quan = que_quan
sv1 = sinhvien("Hứa Thị Lèo", 123, "An Giang")
sv2 = sinhvien("Bành Thị Nỗi", 124, "Long An")
sv3 = sinhvien("Bành Thị Nỗi", 125, "Hà Nội")
print(f"sinh viên 1 có tên là {sv1.hoten}, quê ở {sv1.que_quan}")
print(f"Quê quán của sv1, sv2, sv3 lần lượt là: {sv1.que_quan}, {sv2.que_quan}, {sv3.que_quan}")
```



7. Method :

Phương thức là các hàm được định nghĩa bên trong một class. Các phương thức được sử dụng để thực hiện các công việc cụ thể

```
class Item():
    def __init__(self, name, price: float, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity
```

```
# khởi tạo phương thức
def Tong_Gia(self):
    return self.price * self.quantity
```

```
item1 = Item("Phone",1000,2)
item2 = Item("Laptop",3000,2)
```

```
print(item1.Tong_Gia())
```



7. Method :

staticmethod () : phương thức tĩnh

Là một phương thức tiện ích chung thực hiện một tác vụ riêng lẻ.

Bên trong phương thức này không nhận bất kỳ tham số nào như self hay cls

#Đề ý màu của biến trong ()

```
#khởi tạo phương thức tĩnh staticmethod ()
```

```
@staticmethod
```

```
def Check_gia(gia):
```

```
    if gia <=500:
```

```
        return "Cheap, đặt ở tầng 1 "
```

```
    else:
```

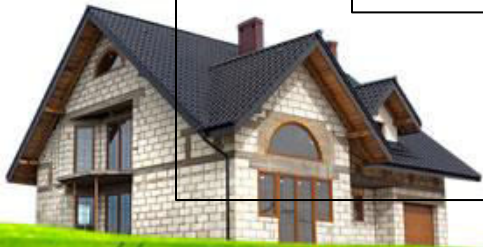
```
        return "Expensive, đặt ở tầng 2 "
```



8. Gán kiểu dữ liệu và kiểm tra điều kiện thuộc tính :

```
class Item():  
    def __init__(self, name, price, quantity):  
        self.name = name  
        self.price = price  
        self.quantity = quantity  
  
    # khởi tạo phương thức  
    def Tong_Gia(self):  
        return self.price * self.quantity  
  
item1 = Item("Phone", "1000", 2)  
item2 = Item("Laptop", 3000, 2)  
print(item1.Tong_Gia())
```

10001000



8. Gán kiểu dữ liệu và kiểm tra điều kiện thuộc tính :

```
class Item():
    def __init__(self, name:str, price:float, quantity=0):
        self.name = name
        self.price = price
        self.quantity = quantity

    # kiểm tra điều kiện thuộc tính:
    assert price >=0 ,f"{price} không hợp lệ, giá phải đảm
    bảo >0 "
    assert quantity >=0 ,f"{quantity} không hợp lệ, Quantity
    phải đảm bảo >0 "
    # khởi tạo phương thức
    def Tong_Gia(self):
        return self.price * self.quantity

item1 = Item("Phone",1000,2)
item2 = Item("Laptop",3000,2)
print(item1.Tong_Gia())
```

2000

