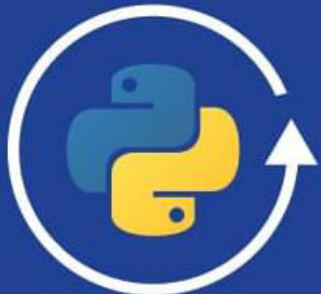
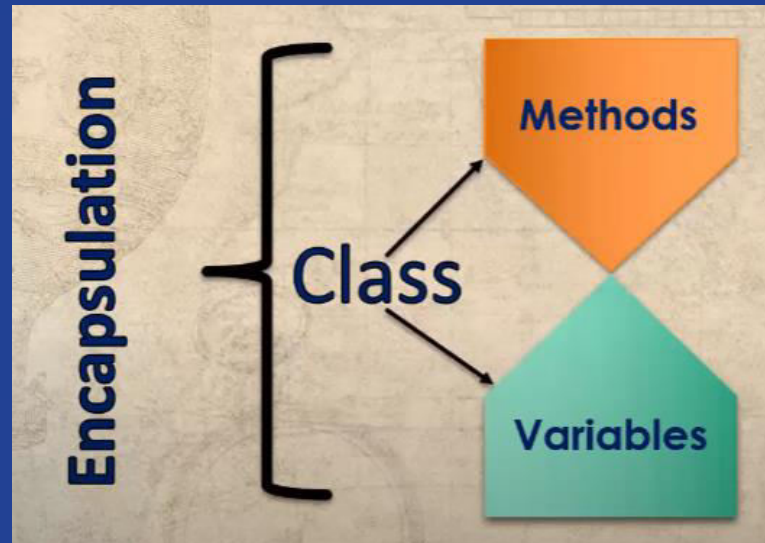


“Tự Học Lập Trình Python “

Bài 42.4: Lập trình Hướng đối tượng Part 4

→ Kế thừa – Bao đóng – Đa hình ←



9. Tính kế thừa (*Inheritance*):

Ngoài thuộc tính chung, chắc chắn khi bán hàng, các đối tượng khác nhau sẽ có các thuộc tính riêng khác nhau

#VD Phone có 4G, 5G

Chúng ta có thể gán thuộc tính 4G, 5G vào mặt hàng bán phẩm =>
Tạo lớp con, ngoài thuộc tính chung, còn thêm thuộc tính khác nữa, mà chỉ nó có

```
# khởi tạo class con
class Phone(Item): # tức là Phone là con Item
    def __init__(self, name: str, price: float, quantity=0, loai="4G"):
        # sử dụng hàm super để có quyền truy cập vào tất cả
        # các thuộc tính mà chúng ta kế thừa từ đó
        super().__init__(
            name, price, quantity
        )
        self.loai = loai

phone1 = Phone("samsung Note20", 150, 5, "5G")
print(phone1.name)
print(phone1.loai)
```

samsung Note20
5G

10. Tính bao đóng (*encapsulation*): Trong class , sẽ luôn có các thuộc tính mà ta muốn cố định nó. Điều này ngăn chặn dữ liệu bị sửa đổi trực tiếp.

Ví dụ:

```
#Encapsulation python
class Bank:
    def __init__(self, hoten, cmt):
        self.hoten = hoten #public
        self.cmt = cmt #public

class Nhanvien(Bank):
    def __init__(self, hoten, cmt):
        super().__init__(hoten, cmt)

        self.__luong = 5000 #sử dụng 2 dấu __ gán private o cho phép truy cập từ bên ngoài
        # code để hr department ( phòng nhân sự ) có thể truy cập, can thiệp vào lương

    def get_luong(self):
        return self.__luong

    def set_luong(self, value):
        self.__luong = value

nv1= Nhanvien("Mai", "5554122")
print(nv1.hoten, nv1.cmt)
#print(nv1.__luong)
#print(dir(nv1))

nv1.set_luong(9999999) # set lương mới cho nv
print(nv1.get_luong())
```

11 . Tính đa hình (*Polymorphism*):

```
class sinhvien_nuocngoai:
    def __init__(self, hoten, ma_sinh_vien, que_quan):
        self.hoten = hoten
        self.ma_sinh_vien = ma_sinh_vien
        self.que_quan = que_quan

    def chao(self):
        print("Hello")
```

```
class sinhvien_viet:
    def __init__(self, hoten, ma_sinh_vien, que_quan):
        self.hoten = hoten
        self.ma_sinh_vien = ma_sinh_vien
        self.que_quan = que_quan

    def chao(self):
        print("Xin chào")
```

=> Có thể xây dựng hàm chung cho cả 2 class trên không ???

```
def hi(sinhvien_nuocngoai): sinhvien_nuocngoai.chao()
```