

Hướng dẫn lập trình OOP trong Python Phần III – MultilInheritance

Faculty: Nguyễn Hùng Cường

Python cho phép hỗ trợ đa thừa kế trong Python, nghĩa là một class trong Python có thể kế thừa nhiều lớp cha, đồng thời thừa hưởng nhiều thành phần của các lớp cha này.

Bước 1: Định nghĩa lớp cha

Mở Python IDE, tạo mới một Project. Sau đó hãy định nghĩa các lớp cha, ở đây ta đã định nghĩa 2 lớp cha là TruongThuySon và TaTon, mỗi lớp ta định nghĩa một method bên trong nó. Các method này sẽ được kế thừa bởi lớp con trong bước sau.

```
class TruongThuySon:
    def nhatduongchi(self):
        print("Nhat Duong Chi !")

class TaTon:
    def camnathu(self):
        print("Cam Na Thu !")
```

Bước 2: Định nghĩa lớp con

Sau khi đã định nghĩa lớp cha, ta sẽ định nghĩa lớp con. Để khai báo lớp con kế thừa từ các lớp cha, trong phần tên của lớp con, ta sẽ liệt kê tên của các lớp cha mà nó có kế thừa (mỗi một tên lớp cha sẽ được phân tách bởi dấu phẩy).

Trong bài này, ta đã định nghĩa một lớp con là TruongVoKy có kế thừa 2 lớp cha đã được định nghĩa ở bước trên. Trong lớp con này, ta đã định nghĩa một method có tên là chitiet(), trong method này ta đã gọi 2 method đã được thừa kế từ 2 lớp cha nói trên.

```
class TruongVoKy(TruongThuySon, TaTon):
    def chitiet(self):
        self.camnathu()
        self.nhatduongchi()
```

Bước 3: Sử dụng các lớp trên, thực thi và xem kết quả

Sau khi đã định nghĩa xong các lớp, ta sẽ khởi tạo đối tượng của lớp con, rồi gọi method chitiet() để xem kết quả.

```
obj = TruongVoKy()  
obj.chitiet();
```

Sau khi thực thi chương trình, ta có thể thấy method chitiet() đã hiển thị thông tin về 2 method được thừa kế từ 2 lớp cha đúng như mong muốn.

```
Cam Na Thu !  
Nhat Duong Chi !
```