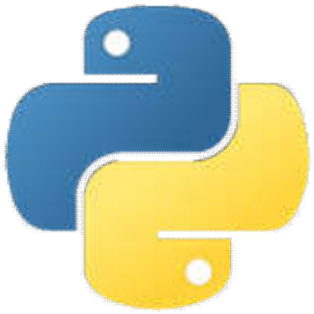

Hướng đối tượng trong Python

AUTHOR: LÊ TRUNG THỰC - EMAIL: THUCLT12A@GMAIL.COM

Nội dung

1. Hướng đối tượng (Object Oriented)
2. Class
3. Attribute & Method
4. Constructor
5. Bài tập thực hành

Hướng đối tượng

- Hướng đối tượng trong **Python** biểu hiện ở việc sử dụng class
- Class là một nguyên mẫu (prototype) để tạo ra các đối tượng (object hay instance)
- Các khái niệm cơ bản sau đây:
  - Trừu tượng (Abstraction)
  - Đa hình (Polymorphism)
  - Đóng gói (Encapsulation)
  - Kế thừa (Inheritance)

- **Abstraction - Tính trừu tượng**: thường được xác định trong khái niệm gọi là lớp trừu tượng (Abstraction class).
- **Abstraction class** - là một class mà bên trong nó chứa một hoặc nhiều phương thức trừu tượng. **Phương thức trừu tượng** ở đây là một phương thức mà chúng ta chỉ được phép khai báo và thực thi nó.

- **Encapsulation – Tính đóng gói**: không cho phép người sử dụng các đối tượng thay đổi trạng thái nội tại của một đối tượng. Chỉ có các phương thức nội tại của đối tượng cho phép thay đổi trạng thái của đối tượng đó.
- Ví dụ: một lớp *Network*, có phương thức *connection()* dùng để kết nối đến các máy tính trong mạng LAN, thì bạn chỉ quan tâm rằng phương thức này làm công việc kết nối mạng chứ không quan tâm đến việc nó kết nối như thế nào.

- **Inheritance - Tính kế thừa**: Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa.
- **Polymorphism - Tính đa hình**: Thể hiện thông qua việc gửi các thông điệp (message). Việc gửi các thông điệp này có thể so sánh như việc gọi các hàm bên trong của một đối tượng. Các phương thức dùng trả lời cho một thông điệp sẽ tùy theo đối tượng mà thông điệp đó được gửi tới sẽ có phản ứng khác nhau.

Lớp

- Class – Là một nguyên mẫu dùng để định nghĩa đối tượng.
- Cú pháp khởi tạo một class:

```
class ClassName:
  'Mô tả ngắn về class (Không bắt buộc)'
  # Code ...
```

Thuộc tính & Phương thức

- **Attribute – Thuộc tính** là biến nằm trong một lớp. Thuộc tính mô tả các đặc tính của một đối tượng.
- **Class variable** – Tương đương với khái niệm trường tĩnh (Static Field) như **Java. Biến của lớp (gần như khái niệm biến toàn cục)** có thể được truy cập thông qua tên lớp hoặc thông qua đối tượng từ một lớp khác trong cùng project.
- **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.

Phương thức khởi tạo

- **Constructor** – là một phương thức đặc biệt của class (trong python mặc định là **__init__**)
- ***self*** **-** Tham số đầu tiên của constructor (Một từ khóa trỏ chính đến class đó)
- Constructor được sử dụng để tạo ra một đối tượng
- Constructor gán giá trị từ tham số vào thuộc tính của đối tượng.
- Một class chỉ có một phương thức khởi tạo
- Nếu class không được định nghĩa constructor, Python mặc định hiểu constructor là **__init__(self)**

```
# Class to define Rectangle
class Rectangle :
    'This is Rectangle class'
    # Contructor
    def __init__(self, width, height):

        self.width= width
        self.height = height

    def getWidth(self):
        return self.width

    def getHeight(self):
        return self.height

    # Method to get area
    def getArea(self):
        return self.width * self.height
```

```
from rectangle import Rectangle

# Create object
rect = Rectangle(10,7)

print ("width = ", rect .width)
print ("height = ", rect .height)
print ("getWidth() = ", rect .getWidth())
print ("getArea() = ", rect .getArea())
```

Tham số mặc định trong constructor

- Python cho phép tham số của phương thức có giá trị mặc định.
- Chú ý: *Tất cả các tham số bắt buộc (required parameters) phải đặt trước tất cả các tham số có giá trị mặc định.*

```
class Person :

    # default value: age & gender.
    def __init__ (self, name, age = 1, gender = "Male" ):

        self.name = name
        self.age = age
        self.gender= gender


    def showInfo(self):

        print ("Name: ", self.name)
        print ("Age: ", self.age)
        print ("Gender: ", self.gender)
```

```
from person import Person

# Create Person object.
aimee = Person("Aimee", 21, "Female")
aimee.showInfo()

print (" ---------- ")

# age, gender default.
alice = Person( "Alice" )
alice.showInfo()

print (" ---------- ")
# gender default.
tran = Person("Tran", 37)
tran.showInfo()
```

Bài tập thực hành

Bài 1:

- a) Tạo các class **học viên** với các thuộc tính tương ứng như sau:
  - Học viên (Họ tên, ngày sinh, email, điện thoại, địa chỉ, đơn vị công tác)
- b) Tạo phương thức showInfo trả về đầy đủ thông tin của cá nhân. Sau đó gọi phương thức này để in ra thông tin cá nhân của mình.
- c) Tạo phương thức getInfo với các tham số mặc định truyền vào: *Địa chỉ = 'Hà Nội', Đơn vị công tác = 'ITPlus'.* Sau đó gọi hàm getInfo và in ra thông tin trong 2 trường hợp:
  - TH1: Truyền vào các thông tin cá nhân nhưng không truyền vào giá trị Địa chỉ và Đơn vị công tác.
  - TH2: Truyền vào đầy đủ các thông tin cá nhân.

THANKS FOR ATTENDING!