

13. Python OOP

Thế nào là một lớp?

Một lớp là một nhóm logic của dữ liệu và phương thức. Nó cho phép tự do tạo các cấu trúc dữ liệu chứa nội dung tùy ý và qua đó khiến việc truy cập dễ dàng hơn.

Ví dụ: Khi nhân viên ngân hàng muốn tìm hiểu chi tiết khách hàng trực tuyến sẽ chuyển đến **lớp khách hàng**, trong đó tất cả các thuộc tính của khách hàng như chi tiết giao dịch, chi tiết rút tiền và tiền gửi, nợ tồn đọng, v.v. sẽ được liệt kê ra.

Trong bài này, chúng ta sẽ tìm hiểu:

- Cách định nghĩa lớp trong Python
- Cách hoạt động của kế thừa
- Hàm khởi tạo trong Python

Cách định nghĩa lớp Python

Để định nghĩa lớp, bạn cần chú ý những điểm sau:

Bước 1) Trong Python, các lớp được xác định bởi từ khóa **"Class"**

```
class myClass():
```

Bước 2) Bên trong các lớp, bạn có thể định nghĩa hàm hoặc phương thức thuộc lớp đó

```
def method1 (self):  
    print "Guru99"  
def method2 (self,someString):  
    print "Software Testing:" + someString
```

- Ở đây chúng ta đã định nghĩa method1 sẽ in ra "Guru99."
- Một phương thức khác mà chúng ta đã xác định là method2 sẽ in ra "Software Testing" + someString. Trong đó someString là biến được cung cấp ở lời gọi hàm.

Bước 3) Mọi thứ trong một lớp đều được thực thi, giống như mã trong hàm, vòng lặp, câu lệnh if, v.v. Những lệnh không được thực thi đều không nằm trong lớp

```
2 class myClass():  
3     def method1(self):  
4         print "Guru99"  
5  
6     def method2(self,someString):  
7         print "Software Testing:" + someString  
8
```

LƯU Ý: Về việc sử dụng "self" trong Python

- Đối số self dùng để chỉ tới chính bản thân đối tượng. Đó là lý do từ "self – bản thân" được sử dụng. Vì vậy trong phương thức này, self sẽ trở tới thực thể của đối tượng mà ta đang

thao tác trên đó.

- Self là tên được quy ước bởi Python để chỉ ra tham số đầu tiên của các phương thức trong lớp. Nó là một phần cú pháp trong Python được sử dụng để truy cập vào thành viên của đối tượng.

Bước 4) Để tạo một đối tượng của lớp

```
c = myClass()
```

Bước 5) Để gọi một phương thức trong một lớp

```
c.method1 ()  
c.method2 ("Testing is fun")
```

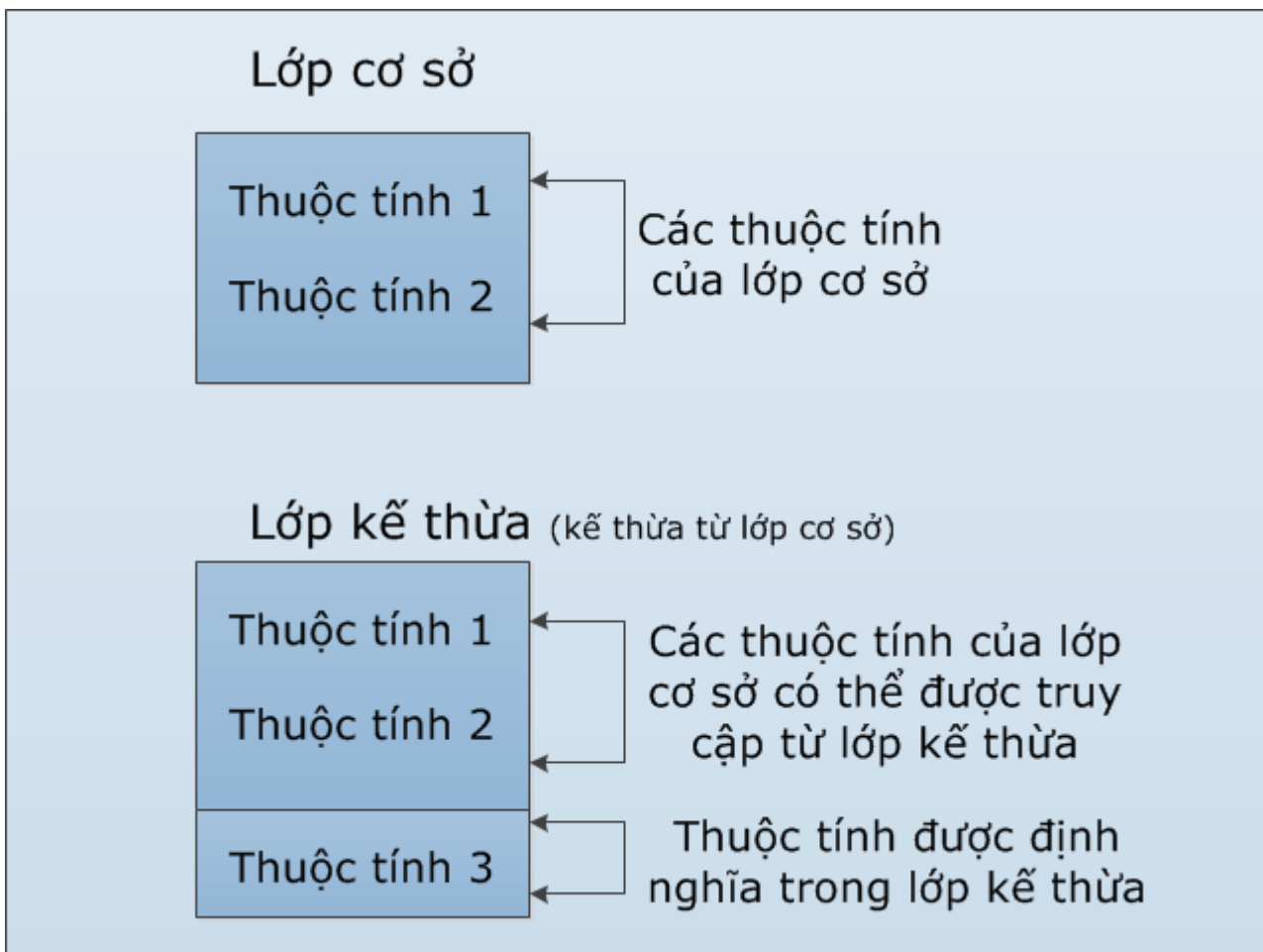
- Lưu ý rằng khi chúng ta gọi method1 hoặc method2, chúng ta không phải cung cấp từ khóa self. Điều này được tự động thực hiện khi chạy Python.
- Khi Python chạy, giá trị "self" sẽ tự động được truyền vào phương thức của lớp dù bạn có chủ động truyền nó vào hay không.
- Bạn chỉ cần quan tâm tới các đối số khác self.

Bước 6) Đây là đoạn mã hoàn chỉnh

```
# Example file for working with classes  
class myClass():  
    def method1(self):  
        print("Guru99")  
  
    def method2(self,someString):  
        print("Software Testing:" + someString)  
  
def main():  
    # exercise the class methods  
    c = myClass ()  
    c.method1()  
    c.method2(" Testing is fun")  
  
if __name__ == "__main__":  
    main()
```

Cách thức hoạt động của kế thừa

Kế thừa là một tính năng được sử dụng trong lập trình hướng đối tượng; nó liên quan tới việc khai báo một lớp mới mà không cần thay đổi hoặc thay đổi rất ít so với lớp hiện có. Lớp mới được gọi là lớp **kế thừa** và lớp mà nó kế thừa được gọi là lớp **cha sở**. Python hỗ trợ kế thừa; nó cũng hỗ trợ đa kế thừa. Một lớp kế thừa các thuộc tính và phương thức hành vi từ một lớp khác gọi là lớp con hoặc lớp thừa kế.



Cú pháp kế thừa trong Python

```
class DerivedClass(BaseClass):
    body_of_derived_class
```

Bước 1) Chạy đoạn mã sau

```
# Example file for working with classes
class myClass():
    def method1(self):
        print("Guru99")

class childClass(myClass):
    #def method1(self):
        #myClass.method1(self);
        #print ("childClass Method1")

    def method2(self):
        print("childClass method2")

def main():
    # exercise the class methods
    c2 = childClass()
    c2.method1()
    #c2.method2()

if __name__ == "__main__":
    main()
```

Lưu ý rằng trong childClass, method1 không được định nghĩa, nhưng được kế thừa từ lớp cha myClass. Đầu ra vẫn là "Guru99".

Bước 2) Xóa dấu chú thích # ở dòng 8 & 10 sau đó chạy đoạn mã.

Bây giờ, method1 được định nghĩa trong childClass và kết quả "childClass Method1" được hiển thị chính xác.

Bước 3) Xóa dấu chú thích # ở dòng 9 sau đó chạy đoạn mã.

Bạn có thể gọi phương thức của lớp cha bằng cách sử dụng cú pháp:

```
ParentClassName.MethodName(self)
```

Trong trường hợp này, chúng ta gọi: myClass.method1(self) và giá trị Guru99 được in ra như mong muốn.

Bước 4) Xóa dấu chú thích # ở dòng 19 và chạy đoạn mã.

Phương thức 2 của lớp con được gọi và "childClass method2" được in ra như mong muốn.

Hàm khởi tạo trong Python

Hàm khởi tạo là một phương thức thuộc lớp giúp khởi tạo đối tượng với giá trị cho trước.

Nó bắt đầu bằng hai dấu gạch dưới (_). Nó là phương thức __init __()

Trong ví dụ dưới đây, chúng ta gán tên của người dùng bằng cách sử dụng hàm tạo.

```
class User:
    name = ""

    def __init__(self, name):
        self.name = name

    def sayHello(self):
        print("Welcome to Guru99, " + self.name)

User1 = User("Alex")
User1.sayHello()
```

Đầu ra sẽ là:

Welcome to Guru99, Alex

Ví dụ sử dụng Python 2

Các đoạn mã nguồn ở trên sử dụng Python3, nếu muốn sử dụng Python2, hãy xem đoạn mã dưới đây:

```
# How to define Python classes
# Example file for working with classes
class myClass():
    def method1(self):
        print "Guru99"

    def method2(self,someString):
        print "Software Testing:" + someString

def main():
```

```

# exercise the class methods
c = myClass ()
c.method1()
c.method2(" Testing is fun")

if __name__ == "__main__":
    main()

#How Inheritance works
# Example file for working with classes
class myClass():
    def method1(self):
        print "Guru99"

class childClass(myClass):
    #def method1(self):
        #myClass.method1(self);
        #print "childClass Method1"

    def method2(self):
        print "childClass method2"

def main():
    # exercise the class methods
    c2 = childClass()
    c2.method1()
    #c2.method2()

if __name__ == "__main__":
    main()

```

Tổng kết

"Lớp" là một nhóm logic các phương thức và dữ liệu. Lớp trong Python cung cấp tất cả các tính năng tiêu chuẩn của lập trình hướng đối tượng.

- Cơ chế kế thừa lớp
- Một lớp kế thừa có thể ghi đè bất kỳ phương thức nào thuộc lớp cơ sở
- Một phương thức có thể gọi phương thức của lớp cơ sở có cùng tên
- Lớp trong Python được định nghĩa bởi chính từ khóa "**class**"
- Bên trong các lớp, bạn có thể định nghĩa các hàm hoặc phương thức và chúng là một phần thuộc lớp đó.
- Các câu lệnh trong lớp cần được thụt lề, giống như mã nguồn trong hàm, vòng lặp, câu lệnh if, v.v.
- Đối số self dùng để chỉ tới chính bản thân đối tượng. Self là tên được quy ước bởi Python để chỉ ra tham số đầu tiên của các phương thức trong lớp.
- Khi chạy Python, giá trị "self" sẽ được truyền vào tự động khi bạn gọi một phương thức từ đối tượng, dù cho bạn có chủ động cung cấp nó hay không.
- Trong Python, một lớp có thể kế thừa các thuộc tính và phương thức hành vi từ một lớp khác và nó được gọi là lớp con hoặc lớp thừa kế.