CƠ BẢN LỚP VÀ ĐỐI TƯỢNG TRONG PYTHON

# Chi tiết nội dung

## III.1. Xây dựng lớp và đối tượng

So với các ngôn ngữ lập trình khác, cơ chế lớp của Python thêm lớp với tối thiểu câu lệnh mới và ngữ nghĩa mới. Cơ chế xây dựng lớp đối tượng của Python là sự trộn lẫn cơ chế trong ngôn ngữ C++ và Modula-3. Các lớp của Python hỗ trợ tất cả các chuẩn đối tượng của lập trình hướng đối tượng, bao gồm:

* Cơ chế kế thừa cho phép nhiều lớp cha.
* Lớp trích dẫn có thể chồng nạp lên bất cứ các phương thức của lớp cha (lớp cơ sở).
* Một phương thức có thể triệu gọi phương thức trong lớp cha có cùng tên.

Các đối tượng có thể chứa nhiều loại dữ liệu khác nhau. Và cũng như việc sử dụng module, việc sử dụng các lớp sẽ làm cho Python năng động hơn: các đối tượng được tạo ra lúc thực thi (runtime) và có khả năng hiệu chỉnh sau khi được tạo thành.

Bảng so sánh lớp và đối tượng của Python với các ngôn ngữ khác:

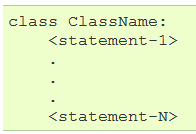
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ngôn ngữ** | | ***C++*** | ***Modula-3*** | ***Smaltalk*** |
| **Tiêu chí** | ***Giống*** | Mặc định các lớp, kể cả dữ liệu trong lớp là public (toàn cục). Trừ khi khai báo là private thì lớp sẽ là private hoặc khai báo biên là cục bộ (local) thì biến mới là cục bộ.  Hầu hết các phép toán có sẵn đều có cú pháp đặc biệt, có thể tái định nghĩa các lớp instances. | Hàm/phương thức được khai báo với tham số đầu tiên tường minh thể hiện đối tượng, sẽ được cung cấp ẩn khi gọi | Các lớp (class) chính là các đối tượng (object). Điều này sẽ hỗ trợ ngữ nghĩa tốt khi import hoặc đổi tên. |
| ***Khác*** | Các kiểu có sẵn có thể được sử dụng như các lớp cơ sở để phát triển tiếp nữa | |  |

*Lưu ý: Một số khái niệm cơ bản của lớp*

* *Namespace:định vị họ tên của đối tượng. Namespace sẽ quy định sự không liên quan giữa hai biến trong hai hàm hoặc hai lớp. Ví dụ: trong xử lý chuỗi có hàm tính độ dài chuỗi là len(“chuoi”), trong xử lý danh sách list có hàm đếm số phần tử trong danh sách len([“danh”, “sach”, “phan”,”tu”]). Hai hàm này tuy giống tên nhưng hoàn toàn khác nhau vì mỗi hàm sẽ thuộc 1 namespace khác nhau.*
* *Scope:phạm vi truy cập. Uu tiên truy xuất đến scope gần nhất trong đoạn chương trình. Ví dụ: nếu không mô tả tường minh mà hai biến có cùng tên nhưng phạm vi truy cập khác nhau thì biến có phạm vi truy cập gần hơn sẽ được sử dụng.*

***Bổ sung lý thuyêt:* Cú pháp định nghĩa lớp:**

Dạng đơn giản nhất về định nghĩa lớp như sau:

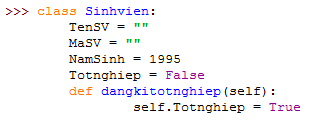


Các nguyên tắc cơ bản khi sử dụng **lớp (class)** trong Python:

* Giống như các ***module*** (được định nghĩa bằng từ khóa *def*), các **lớp** được định nghĩa bằng từ khóa *class* phải được gọi 1 lần trước khi khai báo sử dụng đối tượng.
* Khi lớp được thiết lập, các biến trong lớp sẽ thuộc namespace của lớp (mới được tạo).
* Một lớp được tạo có thể sử dụng theo 2 cách: cách các tham chiếu thuộc tính (attribute references) và cách thực thể hiện thực (instantiation).

**Các ví dụ thực tập:**

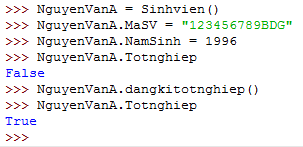
* Ví dụ 1 [Học về cách tạo lớp đơn giản và tạo instance của lớp]: Tạo 1 lớp Sinhvien có các thuộc tính: Tên sinh viên, mã số sinh viên, năm sinh sinh viên, ban đầu giá trị totnghiep là False, nghĩa là sinh viên chưa tốt nghiệp. Và một phương thức mang tên dangkitotnghiep. Một khi gọi đến phương thức này thì tức thì sinh viên đó được gán giá trị tốt nghiệp là True.



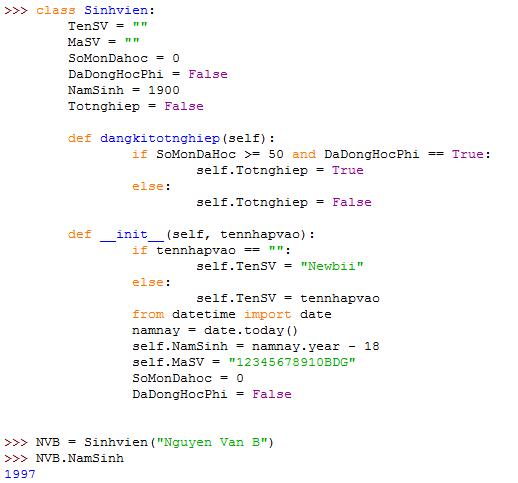
Khi được tạo ra, các biến Sinhvien.TenSV, Sinhvien.MaSV, Sinhvien.dangkitotnghiep là những tham chiếu thuộc tính (attribute references) và chúng ta có thể truy cập trực tiếp.

Tuy nhiên, để sử dụng dạng thực thể, ta phải khai báo 1 thực thể nào đó là Sinhvien. Ví dụ:

* Khai báo một thể hiện (instance) của lớp Sinhvien là NguyenVanA.
* Instance này có mã sinh viên là: ‘123456789BDG’
* Instance có năm sinh là 1996
* Và kiểm tra xem instance NguyenVanA đã tốt nghiệp chưa. Sau đó triệu gọi phương thức dangkitotnghiep() để sinh viên NguyenVanA được tốt nghiệp!



* Lưu ý: Một phương thức (method) là một hàm (function) của đối tượng. Ví dụ: để xử lý danh sách list, chúng ta có các phương thức: append (thêm vào cuối danh sách), insert (chèn thêm 1 phần tử vào đâu đó danh sách),… Tuy nhiên, chúng ta phải lưu ý khi sử dụng *NguyenVanA.dangkitotnghiep()* được gọi là ***function object*** (đối tượng hàm); còn khi gọi *Sinhvien().dangkitotnghiep()* ta gọi là ***method object*** (đối tượng phương thức). Ngoài ra, lưu ý hai cách gọi hàm sau là tương đương: *NguyenVanA.dangkitotnghiep() và Sinhvien().dangkitotnghiep(NguyenVanA)*
* Ví dụ 2 [***Hàm dựng***]:



* Mở rộng lớp Sinhvien vừa tạo theo hướng sau: Sinh viên nào vào trường học cũng có tên là ‘Newbii’ (nếu không nhập tên), năm sinh là năm hiện tại trừ 18, mã số mặc định là ‘12345678910BDG’, bổ sung số môn đã học, ban đầu số môn đã học là 0. Ngoài ra, bổ sung thêm một giá trị ghi nhận đã đóng đủ học phí rồi (True) hay chưa (False). Ban đầu, sinh viên vào trường chưa đóng học phí nên giá trị sẽ là False. Số môn đã học và điều kiện đóng đủ học phí là điều kiện để xét tốt nghiệp. Khi muốn tốt nghiệp sinh viên phải học nhiều hơn 50 môn và đã đóng đủ học phí (giá trị đóng đủ học phí là True). Tóm lại, hàm dựng là hàm thiết lập các giá trị ban đầu của một thực thể.

## III.2. Kế thừa và đa kế thừa

Lớp con và lớp cha: Tình huống thực tế: lớp Guitar là lớp con của lớp Nhạc Cụ; lớp Người, Chó, Mèo là các lớp con của lớp Động vật; lớp Động vật là lớp con của lớp Sinh vật

Trong Python, một lớp có thể kế thừa bằng việc khai báo các lớp cha trong khi khai báo lớp:

Ví dụ:

class DongVat(SinhVat)

<các lệnh trong lớp DongVat>

Khi đó, lớp con sẽ kế thừa những đặc tính của lớp cha mà không cần xây dựng lại.

## III.5. Bài tập lập trình

Xây dựng lớp đối tượng xe cộ với các yêu cầu:

* Có các thuộc tính: Chủ nhân, bảng số xe, tình trạng xe, danh sách tọa độ X, danh sách tọa độ Y, danh sách thời gian.
* Thực hiện nhập dữ liệu
* Viết phương thức: Tìm khoảng cách một xe và xe khác xe (có cùng thời gian)
* Xây dựng phương thức kiểm tra xem: trong thời gian cụ thể nào đó có xe nào ở gần khu vực một xe nào đó 200 mét không? (biết tọa độ X,Y tính theo mét).
* Xây dựng lớp xe hơi kế thừa lớp xe cộ và có thêm thông tin: màu, loại xe, năm sản xuất, khí thải của xe.
* Xây dựng các phương thức: tính xem giờ nào là tất cả các xe thải nhiều nhất; cho một xe và một thời điểm, hỏi có xe cùng loại trong vòng 200 mét không và cách nhau khoảng 5 phút.
* **Các đoạn code bên trên:**

class Sinhvien:

TenSV = ""

MaSV = ""

SoMonDahoc = 0

DaDongHocPhi = False

NamSinh = 1900

Totnghiep = False

def dangkitotnghiep(self):

if SoMonDaHoc >= 50 and DaDongHocPhi == True:

self.Totnghiep = True

else:

self.Totnghiep = False

def \_\_init\_\_(self, tennhapvao):

if tennhapvao == "":

self.TenSV = "Newbii"

else:

self.TenSV = tennhapvao

from datetime import date

namnay = date.today()

self.NamSinh = namnay.year - 18

self.MaSV = "12345678910BDG"

SoMonDahoc = 0

DaDongHocPhi = False