TRƯỜNG ĐẠI HỌC VĂN LANG

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI THU HOẠCH   
Lập Trình Python Nâng Cao**

NGÀNH: Công Nghệ Thông Tin

***Đề tài:***

**Tìm Hiểu Về OOP**

**MSSV: 2274802010310**

**Họ và tên: Nguyễn công huy**

**GVHD: Huỳnh Thái Học**

TP. Hồ Chí Minh – năm 2024

1. ***Giới thiệu về OOP***

**Lập trình hướng đối tượng (OOP)** là một kỹ thuật lập trình cho phép [**lập trình viên**](https://topdev.vn/blog/lap-trinh-vien/) tạo ra các đối tượng trong code trừu tượng hóa các đối tượng.

OOP sử dụng khái niệm về các đối tượng và các lớp. Một lớp học có thể được coi là một "bản thiết kế" cho các đối tượng. Chúng có thể có các thuộc tính riêng (đặc điểm mà chúng sở hữu) và phương thức (hành động chúng thực hiện).

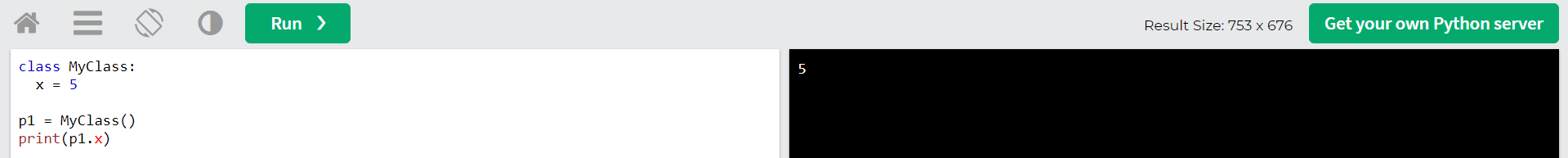
***Đối tượng****là những sự vật, sự việc mà nó có những tính chất, đặc tính, hành động giống nhau và ta gom góp lại thành đối tượng giống trong thực tế cuộc sống. Khi lập trình OOP, chúng ta sẽ định nghĩa các lớp (class) để gom (mô hình) các đối tượng thực tế.*

**Đối tượng**

Một đối tượng bao gồm 2 thông tin: **thuộc tính** và **phương thức**.

* **Thuộc tính**chính là những thông tin, đặc điểm của đối tượng. Ví dụ: con người có các đặc tính như mắt, mũi, tay, chân…
* **Phương thức**là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: một người sẽ có thể thực hiện hành động nói, đi, ăn, uống, . . .

1. ***Lập trình OOP trong python***



Đây là cách khai báo class trong python  
câu lệnh này có nghĩa là khai báo lớp MyClass và có thuộc tính x có giá trị bằng 5 sau đó gián p1 với MyClass và in tra thuộc tính x

Ảnh có chứa văn bản, phần mềm, ảnh chụp màn hình, Phần mềm đa phương tiện

Mô tả được tạo tự động

Đoạn code tạo ra một lớp Person với phương thức khởi tạo \_\_init\_\_ để gán giá trị cho các thuộc tính name và age. Khi tạo đối tượng p1 với tên "John" và tuổi 36, bạn có thể truy cập và in ra các thuộc tính đó

1. ***Các nguyên lí cơ bản của OOP***

**Tính đóng gói (Encapsulation)** hay còn gọi là **hiding information** giúp gom nhóm lại các thuộc tính (properties), phương thức (methods) và nhiều thành phần khác thành một đối tượng hay một đơn vị.

Tính đóng gói được triển khai bằng cách sử dụng **access modifier**: public, private, protected, internal.

* **public**: Có thể truy cập từ bất cứ đâu
* **private**: Chỉ có thể truy cập ở bên trong class
* **protected**: Chỉ có thể truy cập ở bên trong class và các class kế thừa từ class đó
* **internal**: Giống như public nhưng chỉ hạn chế trong 1 **assembly**.

class BankAccount:

def \_\_init\_\_(self, name, account\_number, create\_date):

self.name = name

self.account\_number = account\_number

self.create\_date = create\_date

def get\_account\_name(self):

return self.name

Tính kế thừa cho phép tạo ra 1 class con từ 1 class có sẵn và mở rộng class đó. Các class con có thể kế thừa lại các thuộc tính (properties) và phương thức (methods) của class cha, có thể không cần định nghĩa lại các phương thức hoặc tái định nghĩa (override) hoặc thêm các phương thức sử dụng riêng ở lớp con. Tính chất này giúp tái sử dụng, tận dụng mã nguồn có sẵn.

Ảnh có chứa văn bản, Phông chữ, hàng, biểu đồ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Tính đa hình cho phép một hành động có thể được thực hiện bằng nhiều cách khác nhau.

Có 2 cách vận dụng tính đa hình:

* Method **overloading**(đa hình khi biên dịch(**compile time**)): Trong 1 lớp (class) các phương thức (methods) có cùng tên nhưng kiểu trả về và tham số truyền vào khác nhau (số lượng, kiểu)

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

* Method **overriding**(đa hình ở thời điểm thực thi(**runtime**)): Các phương thức được thực hiện ở các lớp con kế thừa từ lớp cha (base class). Nội dung thực hiện bên trong mỗi lớp khác nhau tùy vào logic nghiệp vụ. Chỉ khi nào runtime ta mới biết được đối tượng sẽ sử dụng phương thức nào.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

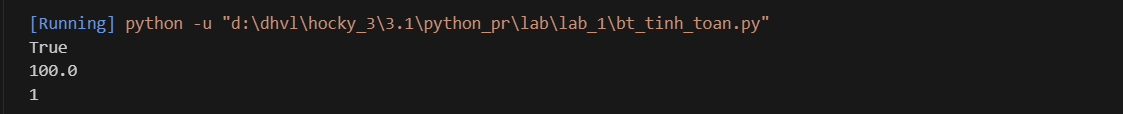
Mô tả được tạo tự động

**Tính trừu tượng (Abstraction)**

* So với 3 tính chất trên thì tính trừu tượng để giải thích hơi khó hiểu 1 tí bởi vì rất trừu tượng. ^^
* Tính trừu tượng cho phép tổng quát hóa một đối tượng. Nghĩa là ẩn đi những thông tin chi tiết bên trong, chỉ thể hiện ra những thông tin bên ngoài. Và nhìn vào thông tin bên ngoài đó ta có thể hiểu được đối tượng đó làm gì.
* Tính chất này được thể hiện qua việc sử dụng **interface** hoặc **abstract class** .

**Ảnh có chứa ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện, văn bản

Mô tả được tạo tự động**

****

1. ***Nhận xét về OOP***

**Các ưu điểm của lập trình hướng đối tượng**

* Dựa trên nguyên lý kế thừa, trong quá trình mô tả các lớp có thể loại bỏ những chương trình bị lặp, dư. Và có thể mở rộng khả năng sử dụng các lớp mà không cần thực hiện lại. Tối ưu và tái sử dụng code hiệu quả.
* Đảm bảo rút ngắn thời gian xây dựng hệ thống và tăng năng suất thực hiện.
* Sự xuất hiện của 2 khái niệm mới là **lớp**và **đối tượng** chính là **đặc trưng của phương pháp lập trình hướng đối tượng**. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng cấu trúc để lại. Ngoài ra 2 khái niệm này đã giúp biểu diễn tốt hơn thế giới thực trên máy tính.