

# Object Oriented Programming - Day 9

Ngày 20 tháng 6 năm 2024

Ngày thực hiện:	19/06/2024
Người thực hiện:	Đinh Thị Tâm
Nguồn:	AIO2024 - Day 9
Nguồn dữ liệu (nếu có):	<a href="#">Link Data Sources of Day 9</a>
Từ khóa:	Tuple - Cosine Similarity
Người tóm tắt:	Đinh Thị Tâm

## 1. Mô tả

**Lập trình hướng đối tượng (OOP)** là một phương pháp lập trình tập trung vào việc tạo ra các "đối tượng", là những đơn vị cơ bản đại diện cho các thực thể trong thế giới thực. Mỗi đối tượng có các thuộc tính (data) và phương thức (hành vi) riêng. OOP mang lại nhiều lợi ích cho việc lập trình, bao gồm:

- Tái sử dụng mã: OOP cho phép bạn viết mã có thể được sử dụng lại nhiều lần cho các mục đích khác nhau. Điều này giúp tiết kiệm thời gian và công sức, đồng thời giúp mã dễ dàng bảo trì hơn.
- Tính linh hoạt: OOP giúp bạn dễ dàng thay đổi và mở rộng mã của mình. Khi bạn cần thêm tính năng mới, bạn chỉ cần tạo ra các đối tượng mới hoặc sửa đổi các đối tượng hiện có.
- Tính bảo trì: OOP giúp mã của bạn dễ đọc và dễ hiểu hơn. Điều này giúp bạn dễ dàng tìm và sửa lỗi, đồng thời giúp những người khác dễ dàng hiểu mã của bạn.

## 2. Bài tập: Tạo lớp Staff gồm các thuộc tính như: name, age, address, salary, total\_time. Và các phương thức như: show\_info, calculate\_bonus. Biết tiền thưởng được tính bằng công thức dưới đây:

- Nếu tổng giờ làm của nhân viên  $\geq 200$  thì **thưởng** = lương \* 20
- Nếu tổng giờ làm của nhân viên  $\geq 100$  và  $< 200$  thì **thưởng** = lương \* 10
- Nếu tổng giờ làm của nhân viên  $< 100$  thì **thưởng** = lương

### (a) Code

```
1 # define class Staff
2 class Staff:
3     def __init__(self, name, age, address, salary, total_time):
4         self.__name = name
5         self.__age = age
6         self.__address = address
7         self.__salary = salary
8         self.__total_time = total_time
```

```
9
10 def show_info(self):
11     print(f'Name: {self.__name}, age: {self.__age}', end=' ')
12     print(f'address: {self.__address}, salary: {self.__salary}')
13     print(f'\t\toatal_time: {self.__total_time}, bonus: ', end=' ')
14     print(f'{self.calculate_bonus()}')
15
16 def calculate_bonus(self):
17     bonus = 0
18     if self.__total_time >= 200:
19         bonus = self.__salary*0.2
20     elif self.__total_time >= 100:
21         bonus = self.__salary*0.1
22     return bonus
23
24
25 # test
26 staff1 = Staff('An', 25, 'Q10_TPHCM', 30000000, 300)
27 staff2 = Staff('Binh', 28, 'Q2_TPHCM', 50000000, 199)
28 staff3 = Staff('Nga', 29, 'Q.3_TPHCM', 10000000, 99)
29 # output
30 staff1.show_info()
31 staff2.show_info()
32 staff3.show_info()
```

(b) Kết quả

```
Name: An, age: 25 address: Q10_TPHCM, salary: 30000000
        toatal_time: 300, bonus: 6000000.0
Name: Binh, age: 28 address: Q2_TPHCM, salary: 50000000
        toatal_time: 199, bonus: 5000000.0
Name: Nga, age: 29 address: Q.3_TPHCM, salary: 10000000
        toatal_time: 99, bonus: 0
```