

Computer Science Advanced

BÀI 1. GIỚI THIỆU VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

1. Lập Trình Hướng Đối Tượng

Lập trình hướng đối tượng (*Object-Oriented Programming / OOP*) là một mẫu hình lập trình dựa trên khái niệm về **đối tượng** (*object*). Một đối tượng bao gồm các **thuộc tính** (*attributes*) và **phương thức** (*methods*).

Ví dụ: Chú chó trong hình bên là một đối tượng.

| Thuộc tính thể hiện những đặc điểm của chú chó | Phương thức thể hiện các hành động chú chó có thể thực hiện |
|---|--|
| Giống: <i>Shiba Inu</i> | Ăn Ngủ |
| Kích cỡ: <i>65cm</i> | Ngồi Chạy |
| Tuổi: <i>3 tuổi</i> | |
| Màu lông: <i>Vàng</i> | |



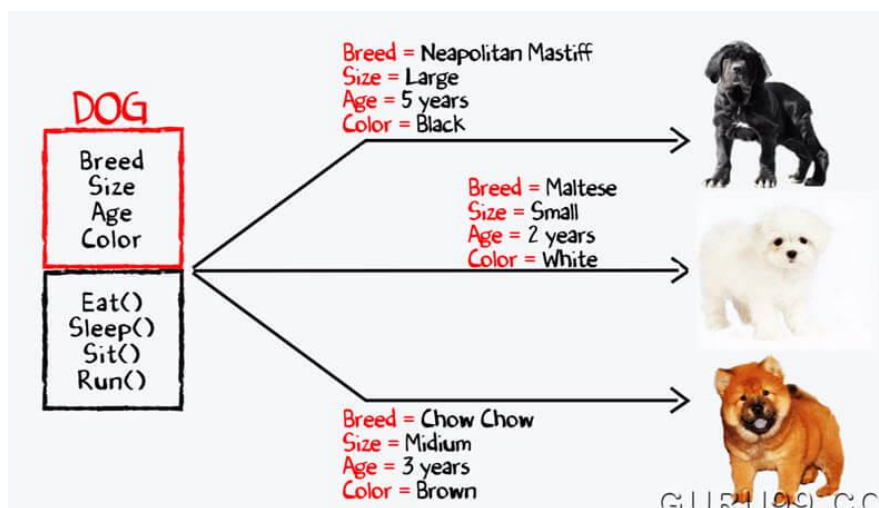
Ảnh: [Daily Paws](#)

LỚP VÀ ĐỐI TƯỢNG

Lớp (*class*) là cách ta thể hiện chung các đối tượng. Những đối tượng có đặc tính tương tự nhau được tập hợp thành một lớp.

Ngược lại, từ một lớp, ta có thể tạo ra các **thực thể** riêng biệt, gọi là **đối tượng**.

Ví dụ: Trong hình bên, *Dog* là một lớp đại diện cho các chú chó nói chung. Ngược lại, mỗi chú chó là một **đối tượng** được thực thể hóa từ lớp *Dog*.



Ảnh: [Gurugoo](#)

Cả lớp và đối tượng đều có các thuộc tính và phương thức:

| | Lớp | Đối tượng |
|--------------------|--|---|
| Thuộc tính | <p>Thể hiện đặc điểm của lớp hoặc đặc điểm chung của tất cả các đối tượng trong lớp.</p> <p><i>Ví dụ:</i> Mọi chú chó đều có bốn chân.</p> | <p>Thể hiện đặc trưng riêng của một đối tượng cụ thể.</p> <p><i>Ví dụ:</i> Chú chó đầu tiên trong hình trên có lông màu đen.</p> |
| Phương thức | <p>Thể hiện hành động trừu tượng của lớp.</p> <p><i>Ví dụ:</i> Lớp Dog có thể tự miêu tả về chính nó (là vật nuôi trong nhà có mũi thính).</p> | <p>Thể hiện cách một đối tượng cụ thể thực hiện hành động.</p> <p><i>Ví dụ:</i> Một chú chó chạy bằng cách di chuyển xen kẽ hai chân trước và hai chân sau.</p> |

2. Lập Trình Hướng Đối Tượng với Python

ĐỊNH NGHĨA LỚP

Đoạn code bên hiện thực lớp **Rectangle** đại diện cho hình chữ nhật. Lớp này định nghĩa:

- Thuộc tính của lớp: *kiểu* (**kind**) của lớp.
- Phương thức của lớp: *miêu tả* (**describe**) về lớp.
- Thuộc tính của đối tượng: *chiều dài* (**self.height**) và *chiều rộng* (**self.width**).
- Phương thức của đối tượng: phương thức *tính chu vi* (**calculate_perimeter**) và *tính diện tích* (**calculate_area**).
- Các thuộc tính của đối tượng nhận giá trị trong *phương thức khởi tạo* (**__init__**). Phương thức này được gọi khi ta khởi tạo một đối tượng từ lớp.

```
# class definition
class Rectangle:

    kind = "geometric shape"           # class attribute

    def __init__(self, height, width): # constructor
        self.height = height          # instance attribute
        self.width = width            # instance attribute

    @classmethod
    def describe(cls):                 # class method
        print(f'I am a {cls.kind}.')
        print(f'My class name is {cls.__name__}.')

    def calculate_perimeter(self):      # instance method
        return (self.height + self.width)*2

    def calculate_area(self):           # instance method
        return self.height * self.width
```

Chú ý:

- Theo nguyên tắc chung, ta viết hoa chữ cái đầu tiên trong mỗi từ của tên lớp, ví dụ MyClass.
- Tham số đầu tiên trong phương thức lớp đại diện cho lớp được gọi, thường được đặt tên là **cls**.
- Tham số đầu tiên trong phương thức đối tượng đại diện cho đối tượng được gọi, thường được đặt là **self**.

SỬ DỤNG LỚP ĐÃ ĐỊNH NGHĨA

| | Code | Output |
|-------------------------------|---|--|
| Truy vấn thuộc tính lớp | <code>print(Rectangle.kind)</code> | geometric shape |
| Thay đổi thuộc tính lớp | <code>Rectangle.kind = "shape"</code> <code>print(Rectangle.kind)</code> | shape |
| Gọi phương thức lớp | <code>Rectangle.describe()</code> | I am a shape. My class name is Rectangle. |
| Khởi tạo đối tượng | <code>rect = Rectangle(3, 2)</code> | |
| Truy vấn thuộc tính đối tượng | <code>print(rect.height)</code> | 3 |
| Thay đổi thuộc tính đối tượng | <code>rect.height = 5</code> <code>print(rect.height)</code> | 5 |
| Gọi phương thức đối tượng | <code>print(rect.calculate_area())</code> | 10 |