

OOP

Lập trình hướng đối tượng (Object-Oriented Programming - OOP) là một phương pháp lập trình dựa trên khái niệm **đối tượng** (object), mô phỏng thế giới thực bằng cách tổ chức dữ liệu và hành vi thành các **lớp** (class) và **đối tượng** (object).

- Cơ bản về OOP trong Python

Cơ bản về OOP trong Python

I. Khái niệm chính

Thuật ngữ	Giải thích
<code>Class</code> (Lớp)	Khuôn mẫu để tạo ra các đối tượng.
<code>Object</code> (Đối tượng)	Một thể hiện cụ thể của lớp.
<code>Attribute</code> (Thuộc tính)	Biến đại diện cho đặc điểm của đối tượng.
<code>Method</code> (Phương thức)	Hàm định nghĩa hành vi của đối tượng.
<code>Constructor</code> (<code>__init__</code>)	Hàm khởi tạo được gọi khi tạo đối tượng mới.
<code>Inheritance</code> (Kế thừa)	Lớp con kế thừa thuộc tính và phương thức từ lớp cha.
<code>Encapsulation</code> (Đóng gói)	Giấu thông tin nội bộ, chỉ cho phép truy cập thông qua các phương thức.
<code>Polymorphism</code> (Đa hình)	Đối tượng có thể có nhiều hình thức khác nhau khi dùng phương thức giống tên.

Ví dụ cơ bản

```
class Animal:
    def __init__(self, name):
        self.name = name

    def speak(self):
        return f"{self.name} makes a sound."

# Kế thừa
class Dog(Animal):
    def speak(self):
        return f"{self.name} says Woof!"

dog = Dog("Rex")
```

```
print(dog.speak()) # Rex says Woof!
```

Khung học lập trình hướng đối tượng trong Python

Giai đoạn 1: Nắm chắc các khái niệm cơ bản

☐

Lớp và đối tượng (`class`, `object`)

☐

`__init__()` và thuộc tính của đối tượng (`self`)

☐

Phương thức (methods)

☐

Các loại thuộc tính (instance vs class attributes)

Giai đoạn 2: Hiểu về tính năng nâng cao

☐

Kế thừa (`Inheritance`)

☐

Ghi đè phương thức (`Method overriding`)

☐

Đa hình (`Polymorphism`)

☐

Đóng gói (`Encapsulation`)

☐

Thuộc tính riêng tư (`__private`, `__protected`)

Giai đoạn 3: Thực hành & ứng dụng

☐

Quản lý học sinh, giáo viên, lớp học

☐

Quản lý sản phẩm, đơn hàng trong cửa hàng

☐

Mô phỏng trò chơi đơn giản với đối tượng như nhân vật, quái vật...

☐

Thiết kế hệ thống quản lý thư viện hoặc nhà sách

Giai đoạn 4: Áp dụng OOP vào dự án thực tế

☐

☐ Áp dụng OOP trong lập trình với **Flask, Django**

☐

☐ Tổ chức mô hình **MVC** sử dụng OOP

☐

☐ Kết hợp OOP với các khái niệm **Design Patterns**

Mục tiêu cuối cùng

- Có thể tư duy và phân tích vấn đề theo hướng đối tượng
- Thiết kế chương trình rõ ràng, mở rộng dễ dàng
- Viết mã sạch, dễ bảo trì, dễ tái sử dụng