

- Prompt: “Viết ví dụ Python minh họa nguyên lý kế thừa (Inheritance) trong OOP với lớp Animal và Dog. Giải thích từng dòng code.”
- Quan sát xem AI có giải thích đúng từng bước không.
- Ghi nhận đoạn code kèm phần giải thích

Kết quả :

Có code với giải thích từng phần một.

Viết ví dụ Python minh họa nguyên lý kế thừa (Inheritance) trong OOP với lớp Animal và Dog. Giải thích từng dòng code.

Dưới đây là ví dụ Python minh họa nguyên lý kế thừa (Inheritance) với lớp `Animal` (cha) và lớp `Dog` (con), kèm giải thích chi tiết từng dòng để bạn dễ hiểu và dễ học OOP.

🌱 Ví dụ code Python

python

Copy code

```
# Lớp cha (Base class)
class Animal:
    def __init__(self, name):
        self.name = name

    def speak(self):
        print("Animal makes a sound")

# Lớp con (Derived class) kế thừa từ Animal
class Dog(Animal):
    def __init__(self, name, breed):
        super().__init__(name)
        self.breed = breed

    def speak(self):
        print(f"{self.name} says: Woof!")

# Tạo đối tượng từ lớp Dog
dog1 = Dog("Buddy", "Golden Retriever")

# Gọi các phương thức
dog1.speak()
```

■ Giải thích từng phần – từng dòng

◆ Dòng 1–2: Khai báo lớp `Animal`

python

Copy code

```
class Animal:
```

- Khai báo lớp `Animal`
- Đây là lớp cha (base class)

◆ Dòng 3–4: Hàm khởi tạo `__init__`

python

Copy code

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

- `__init__` là hàm khởi tạo
- `self.name` là thuộc tính của đối tượng
- Mỗi `Animal` đều có tên

◆ Dòng 6–7: Phương thức `speak()`

python

Copy code

```
def speak(self):  
    print("Animal makes a sound")
```


- Phương thức dùng để mô tả hành vi chung
- Lớp con có thể kế thừa hoặc ghi đè



Lớp Dog kế thừa Animal

◆ Dòng 11: Kế thừa

python


 Copy code

```
class Dog(Animal):
```

- Dog kế thừa từ Animal
- Dog có toàn bộ thuộc tính + phương thức của Animal

◆ Dòng 12–15: Hàm khởi tạo của Dog

python


 Copy code

```
def __init__(self, name, breed):  
    super().__init__(name)  
    self.breed = breed
```

- `super().__init__(name)` :
 - Gọi hàm khởi tạo của lớp cha
 - Giúp Dog có thuộc tính `name`
- `self.breed` :
 - Thuộc tính riêng của Dog

◆ Dòng 17–18: Ghi đè phương thức (Method Overriding)

python

 Copy code


```
def speak(self):  
    print(f"{self.name} says: Woof!")
```

- Dog ghi đè phương thức `speak()`
- Thể hiện đa hình (Polymorphism)



Tạo đối tượng và sử dụng


python

 Copy code

```
dog1 = Dog("Buddy", "Golden Retriever")
```

- Tạo đối tượng `dog1` từ lớp `Dog`
- `name = "Buddy"`
- `breed = "Golden Retriever"`

python

 Copy code

```
dog1.speak()
```

- Gọi phương thức `speak()`
- Do Dog ghi đè → in ra **Woof!**