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

* AI giải thích đúng từng bước

**Đoạn code:**

# Định nghĩa lớp cha (superclass)

class Animal:

def \_\_init\_\_(self, name):

self.name = name

def speak(self):

return "Some sound"

# Định nghĩa lớp con (subclass) kế thừa từ Animal

class Dog(Animal):

def \_\_init\_\_(self, name, breed):

# Gọi lại constructor của lớp cha bằng super()

super().\_\_init\_\_(name)

self.breed = breed

# Ghi đè (override) phương thức speak của lớp cha

def speak(self):

return "Woof!"

# Tạo đối tượng Dog

my\_dog = Dog("Buddy", "Golden Retriever")

# Gọi các phương thức

print(my\_dog.name) # In ra tên

print(my\_dog.breed) # In ra giống loài

print(my\_dog.speak()) # In ra tiếng sủa

**Phần giải thích:**

1. Khai báo một lớp cha tên Animal.

class Animal:

1. Hàm khởi tạo (constructor). Khi tạo một đối tượng Animal, thuộc tính name sẽ được gán.

def \_\_init\_\_(self, name):

self.name = name

1. Phương thức mặc định speak, cho tất cả động vật (chưa cụ thể).

def speak(self):

return "Some sound"

1. Khai báo lớp con **Dog**, kế thừa từ lớp cha **Animal**.

class Dog(Animal):

1. Định nghĩa constructor cho Dog:

* super().\_\_init\_\_(name) gọi lại constructor của lớp cha (Animal) để gán name
* self.breed = breed thêm thuộc tính riêng cho Dog là giống loài.

def \_\_init\_\_(self, name, breed):

super().\_\_init\_\_(name)

self.breed = breed

1. Ghi đè (override) phương thức speak của Animal. Với Dog, speak trả về "Woof!".

def speak(self):

return "Woof!"

1. Tạo một đối tượng Dog có tên **Buddy**, giống **Golden Retriever**.

my\_dog = Dog("Buddy", "Golden Retriever")

1. In ra:

* my\_dog.name → **Buddy**
* my\_dog.breed → **Golden Retriever**
* my\_dog.speak() → **Woof!**

print(my\_dog.name)

print(my\_dog.breed)

print(my\_dog.speak())