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

## Đoạn mã Python:

class Animal:  
 def \_\_init\_\_(self, name):  
 # Hàm khởi tạo: lưu tên động vật  
 self.name = name  
  
 def speak(self):  
 # Phương thức chung cho mọi động vật  
 return f"{self.name} makes a sound."  
  
  
class Dog(Animal):  
 def \_\_init\_\_(self, name, breed):  
 # Gọi \_\_init\_\_ của lớp cha để khởi tạo thuộc tính chung  
 super().\_\_init\_\_(name)  
 # Thuộc tính riêng của Dog  
 self.breed = breed  
  
 def speak(self):  
 # Ghi đè (override) phương thức speak của lớp cha  
 return f"{self.name} the {self.breed} barks."  
  
  
# Sử dụng lớp  
my\_pet = Dog("Rex", "Beagle") # Tạo đối tượng Dog  
print(my\_pet.speak()) # Gọi phương thức speak (sẽ dùng phương thức của Dog)  
print(isinstance(my\_pet, Animal)) # Kiểm tra tính kế thừa: True nếu my\_pet là Animal

## Giải thích từng dòng:

class Animal: - Định nghĩa một lớp tên là Animal (lớp cha/ cơ sở).

def \_\_init\_\_(self, name): - Định nghĩa hàm khởi tạo nhận tham số name để khởi tạo đối tượng.

self.name = name - Lưu tham số name vào thuộc tính instance self.name.

def speak(self): - Định nghĩa phương thức speak cho lớp Animal; có thể được dùng hoặc ghi đè ở lớp con.

return f"{self.name} makes a sound." - Phương thức speak trả về chuỗi mô tả âm thanh (mô phỏng hành vi chung).

class Dog(Animal): - Định nghĩa lớp Dog kế thừa từ lớp Animal (ký hiệu bằng (Animal)).

def \_\_init\_\_(self, name, breed): - Hàm khởi tạo của Dog nhận thêm tham số breed (loài/giống chó).

super().\_\_init\_\_(name) - Gọi hàm khởi tạo của lớp cha để khởi tạo thuộc tính chung (name).

self.breed = breed - Lưu tham số breed vào thuộc tính instance self.breed (thuộc tính riêng của Dog).

def speak(self): - Định nghĩa lại (override) phương thức speak để Dog có hành vi cụ thể hơn.

return f"{self.name} the {self.breed} barks." - Phương thức speak của Dog trả về chuỗi mô tả tiếng sủa cùng tên và giống chó.

my\_pet = Dog("Rex", "Beagle") - Tạo một đối tượng Dog tên là Rex thuộc giống Beagle.

print(my\_pet.speak()) - Gọi phương thức speak() trên my\_pet; do Dog đã ghi đè speak, nên phương thức của Dog được gọi.

print(isinstance(my\_pet, Animal)) - Kiểm tra my\_pet có phải là một thể hiện (instance) của lớp Animal không — trả về True do Dog kế thừa Animal.

## Đánh giá tính đúng đắn của phần giải thích:

Tổng quan: Phần giải thích ở trên là chính xác và đầy đủ cho ví dụ cơ bản về kế thừa trong OOP:  
- Giải thích chỉ rõ lớp cha (Animal) và lớp con (Dog) cùng với cách gọi super() để tái sử dụng khởi tạo của lớp cha.  
- Nêu rõ khái niệm ghi đè phương thức (override) khi Dog định nghĩa lại phương thức speak().  
- Thể hiện cách kiểm tra quan hệ kế thừa bằng isinstance().  
  
Gợi ý nâng cao (không bắt buộc):  
- Có thể nhắc thêm về việc sử dụng super() trong các môi trường đa kế thừa (multiple inheritance).  
- Nếu muốn, có thể minh họa gọi phương thức của lớp cha từ lớp con (ví dụ: super().speak()).