**Đề ôn tập 1:Tính đóng gói và kế thừa**

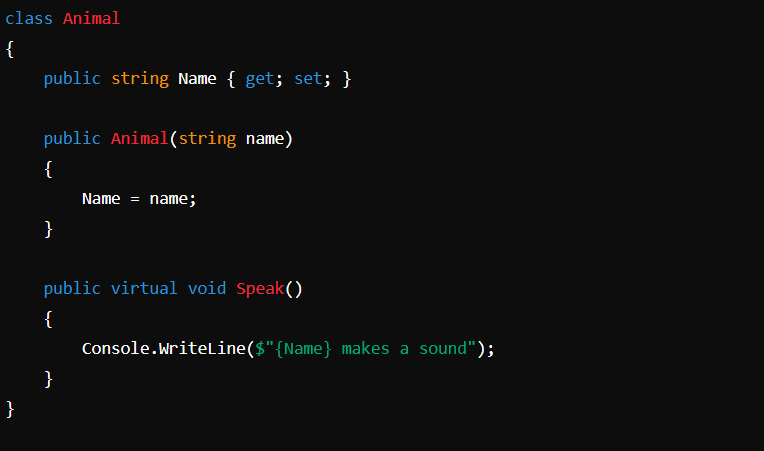
Yêu cầu trả lời trực tiếp kết quả vào file word và nộp

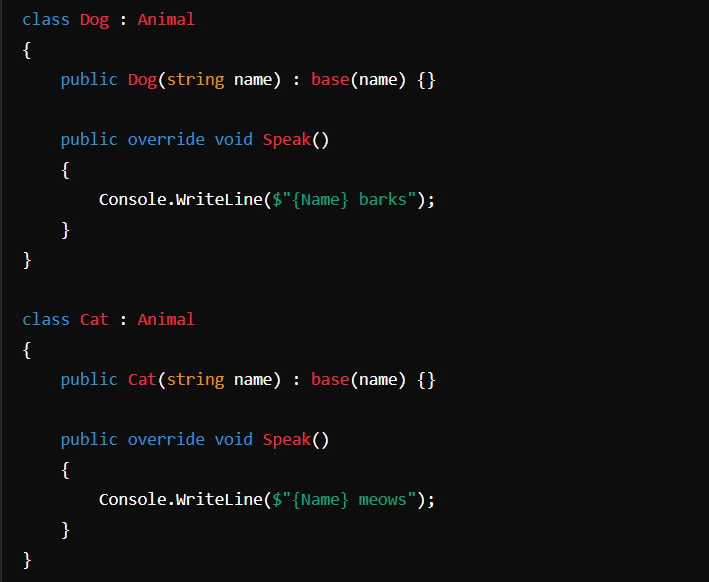
1. Định nghĩa tính kế thừa trong lập trình hướng đối tượng và cho biết mục đích của nó. Cho ví dụ minh họa

Tính kế thừa là một khái niệm trong lập trình hướng đối tượng (OOP), cho phép một lớp (class) kế thừa các thuộc tính (properties) và phương thức (methods) từ một lớp khác. Lớp kế thừa được gọi là lớp con (subclass) hoặc lớp dẫn xuất (derived class), trong khi lớp bị kế thừa được gọi là lớp cha (superclass) hoặc lớp cơ sở (base class).

**Mục đích:**

* **Tái sử dụng mã (Code Reusability):** Giúp tái sử dụng mã nguồn, giảm thiểu việc viết lại mã và tăng tính tái sử dụng.
* **Mở rộng tính năng (Extend Functionality):** Cho phép mở rộng và tùy biến các tính năng của lớp cơ sở mà không làm ảnh hưởng đến lớp cơ sở.
* **Duy trì và bảo trì (Maintenance):** Giúp quản lý, bảo trì và cập nhật mã nguồn dễ dàng hơn, vì các thay đổi trong lớp cơ sở sẽ tự động áp dụng cho tất cả các lớp con.
* **Đa hình (Polymorphism):** Cho phép một phương thức có thể có nhiều dạng khác nhau, cung cấp khả năng linh hoạt và dễ mở rộng của chương trình.



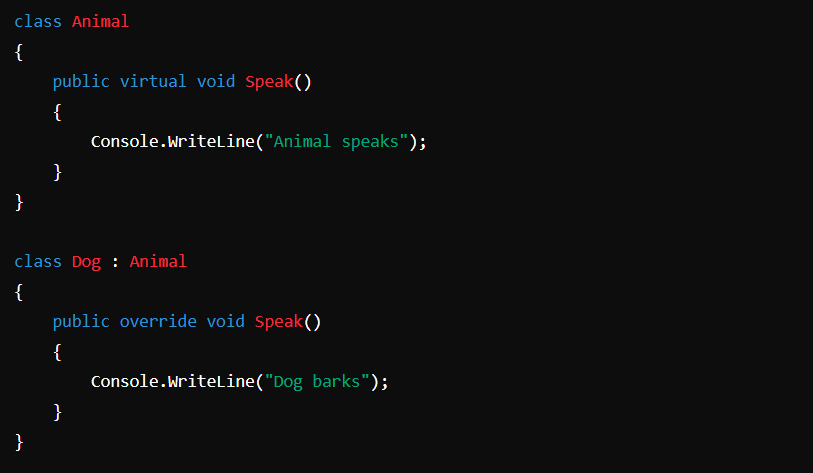


1. Giải thích sự khác biệt giữa các loại kế thừa: đơn kế thừa, đa kế thừa, và kế thừa đa cấp. Cho ví dụ minh họa.

Trong lập trình hướng đối tượng, kế thừa có thể được phân loại thành ba loại chính: đơn kế thừa, đa kế thừa, và kế thừa đa cấp.

#### Đơn kế thừa (Single Inheritance)

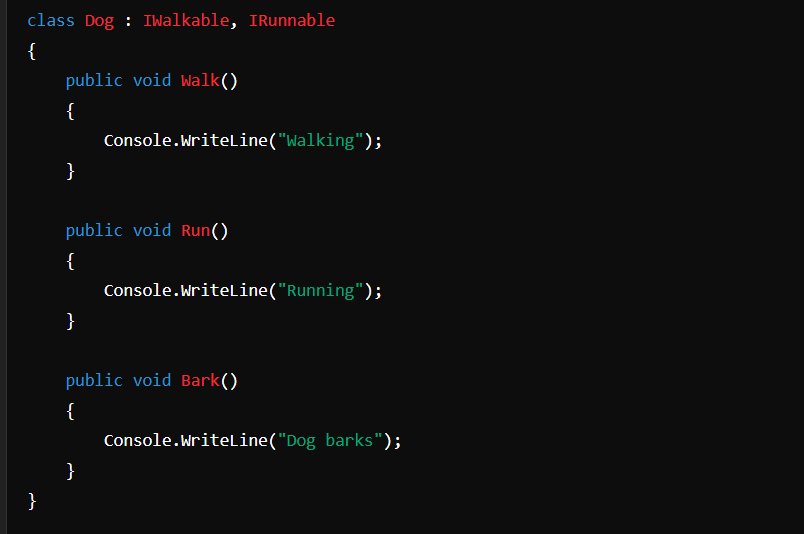
Đơn kế thừa xảy ra khi một lớp con chỉ kế thừa từ một lớp cha duy nhất.



#### Đa kế thừa (Multiple Inheritance)

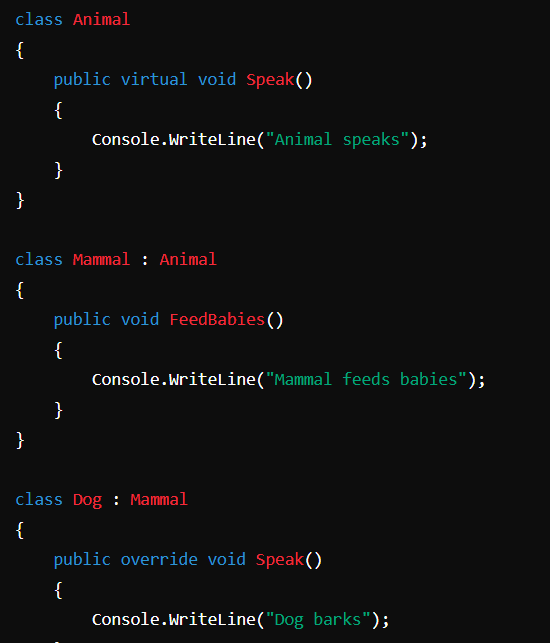
C# không hỗ trợ đa kế thừa trực tiếp cho các lớp, nhưng nó hỗ trợ đa kế thừa thông qua các interface.





#### Kế thừa đa cấp (Multilevel Inheritance)

Kế thừa đa cấp xảy ra khi một lớp con kế thừa từ một lớp mà lớp đó cũng kế thừa từ một lớp khác.



1. Làm thế nào để ngăn chặn một lớp bị kế thừa? Cho ví dụ minh họa.

Trong C#, để ngăn chặn một lớp bị kế thừa, bạn sử dụng từ khóa sealed. Khi một lớp được đánh dấu là sealed, lớp đó không thể bị kế thừa bởi bất kỳ lớp nào khác.



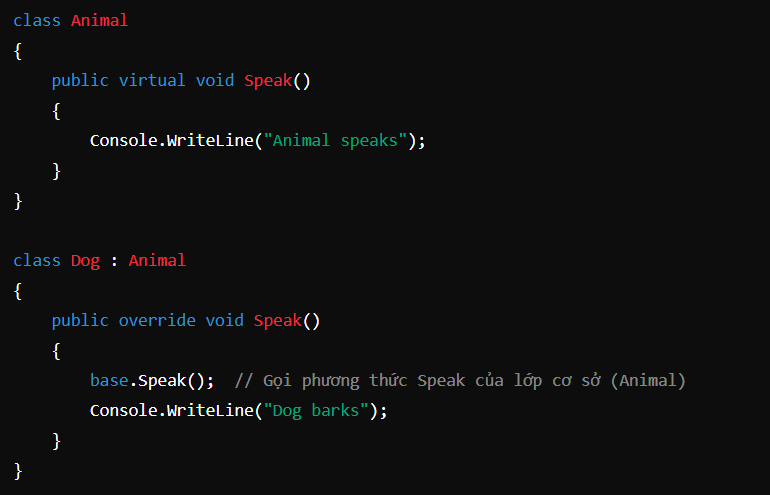


### Lý do sử dụng lớp sealed

* **Bảo mật:** Ngăn chặn việc mở rộng lớp trong các thư viện mà bạn không muốn cho phép mở rộng, giúp bảo vệ các phương thức và dữ liệu nội bộ của lớp.
* **Hiệu suất:** Các lớp sealed có thể tối ưu hóa hiệu suất vì trình biên dịch biết rằng không có lớp nào kế thừa từ nó và có thể thực hiện một số tối ưu hóa nhất định.

1. Giải thích từ khóa base trong C# và cung cấp ví dụ sử dụng base để gọi phương thức của lớp cơ sở. Cho ví dụ minh họa.

Từ khóa base trong C# được sử dụng để truy cập các thành viên của lớp cơ sở từ bên trong lớp dẫn xuất. Nó thường được sử dụng để gọi các phương thức, thuộc tính hoặc các chỉ mục của lớp cơ sở đã bị ghi đè trong lớp dẫn xuất. Ngoài ra, base cũng được sử dụng để gọi constructor của lớp cơ sở từ constructor của lớp dẫn xuất.



**Thực hành 1:**

A screenshot of a computer

Description automatically generated

Từ lược đồ trên, tạo các lớp và bổ sung Phương thức tạo lập, toString

Trong lớp DanhSachNhanVien thực hiện các yêu cầu sau:

1. Phương thức nhập thủ công
2. Phương thức đọc và xuất dữ liệu từ file cho collection
3. Từ các thuộc tính của các Lớp Nguoi, NhanVien, QuanLy xây dựng các phương thức tương ứng như tìm kiếm, sắp xếp, cập nhật, thêm, xóa, sửa. Ví dụ: với thuộc tính Phòng của QuanLy có thể thực hiện:
   1. Tìm tất cả quản lý thuộc phòng nào đó
   2. Xóa tất cả quản lý của phòng nào đó
   3. Sắp xếp nhân viên theo chiều tăng, giả, của phòng
   4. …..

**Thực hành 2:** thực hiện các yêu cầu tương tự Thực hành 1

A screenshot of a computer

Description automatically generated