**Tìm hiểu Kế thừa và Đa hình trong C#**

Trong C#, kế thừa (inheritance) và đa hình (polymorphism) là hai khái niệm quan trọng trong lập trình hướng đối tượng. Dưới đây là giải thích chi tiết và ví dụ về cả hai khái niệm:

**1. Kế thừa (Inheritance)**

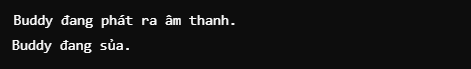
Kế thừa là khả năng một lớp (class) có thể kế thừa các thuộc tính và phương thức từ một lớp khác. Lớp kế thừa (derived class) có thể sử dụng lại hoặc mở rộng các đặc điểm của lớp cha (base class).

Công dụng của Kế thừa:

* Tái sử dụng mã nguồn: Các lớp con không cần phải viết lại mã nguồn của lớp cha.
* Giảm thiểu trùng lặp: Các phương thức và thuộc tính chung có thể được định nghĩa trong lớp cha và được chia sẻ cho các lớp con.

Ví dụ về Kế thừa trong C#

Output:

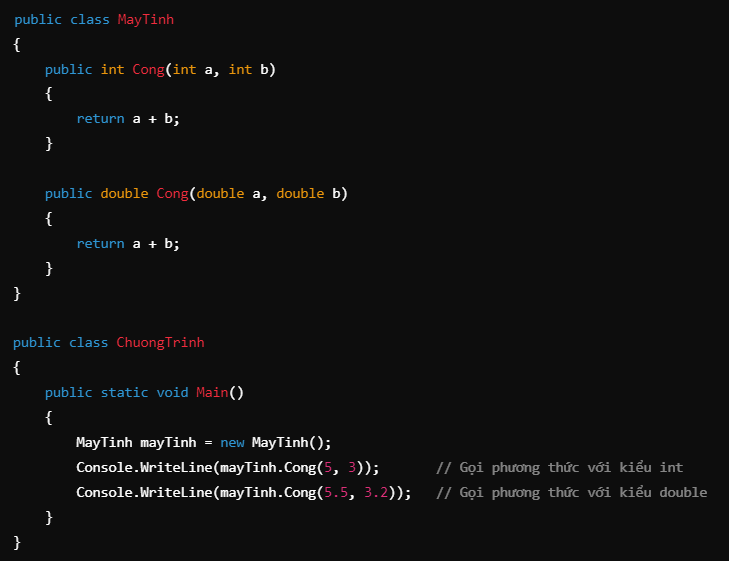


**2. Đa hình (Polymorphism)**

Đa hình cho phép đối tượng có thể được xử lý theo nhiều dạng khác nhau. Trong C#, có hai loại đa hình chính:

* Đa hình tĩnh (Compile-time Polymorphism): Thường thông qua method overloading (nạp chồng phương thức). Các phương thức có cùng tên nhưng khác nhau về kiểu hoặc số lượng tham số.
* Đa hình động (Run-time Polymorphism): Được hiện thực qua method overriding (ghi đè phương thức), cho phép lớp con thay thế phương thức của lớp cha.

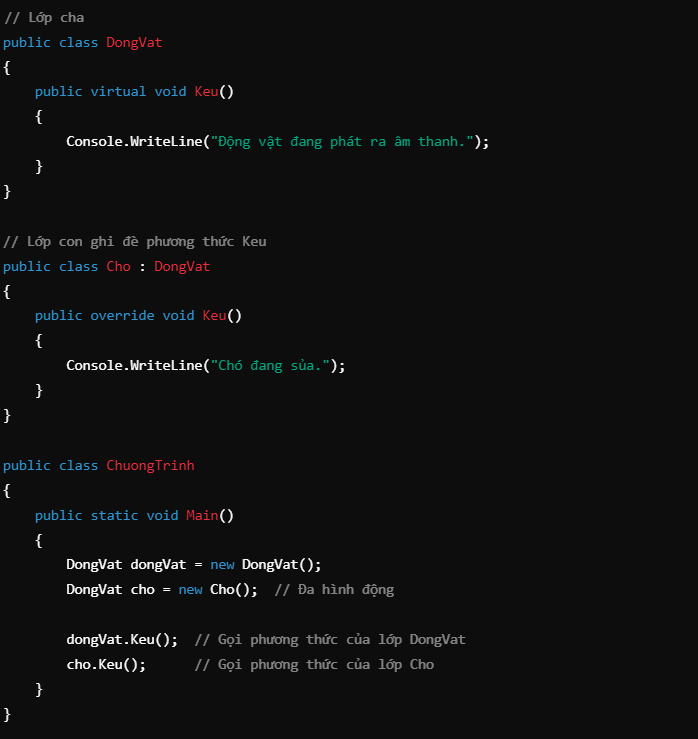
Ví dụ về Đa hình tĩnh (Compile-time Polymorphism) - Method Overloading



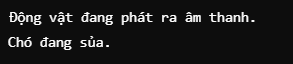
Output:



Ví dụ về Đa hình động (Run-time Polymorphism) - Method Overriding



Output:



**3. So sánh và Ứng dụng**

So sánh giữa Kế thừa và Đa hình:

*Điểm giống nhau:*

* Cả hai đều là nguyên tắc của lập trình hướng đối tượng.
* Cả hai đều giúp tái sử dụng mã và làm mã nguồn dễ quản lý hơn.

*Điểm khác nhau:*

* Kế thừa liên quan đến mối quan hệ giữa lớp cha và lớp con. Lớp con kế thừa đặc tính của lớp cha.
* Đa hình liên quan đến khả năng đối tượng có thể được sử dụng dưới nhiều hình thức khác nhau, giúp mã linh hoạt hơn.

*Ứng dụng thực tế*:

* Kế thừa: Sử dụng khi bạn có các lớp có đặc điểm chung và muốn tái sử dụng mã. Ví dụ: Lớp Vehicle (Phương tiện) có thể là lớp cha của các lớp Car (Ô tô) và Bike (Xe đạp).
* Đa hình: Dùng để tạo ra mã linh hoạt. Ví dụ: Hệ thống tính lương trong đó các loại nhân viên khác nhau (full-time, part-time) có cách tính lương khác nhau, nhưng vẫn được gọi chung thông qua một phương thức tính lương.

Ví dụ về Kế thừa và Đa hình kết hợp trong thực tế

Output:

