**BÀI ÔN SỐ 1**

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

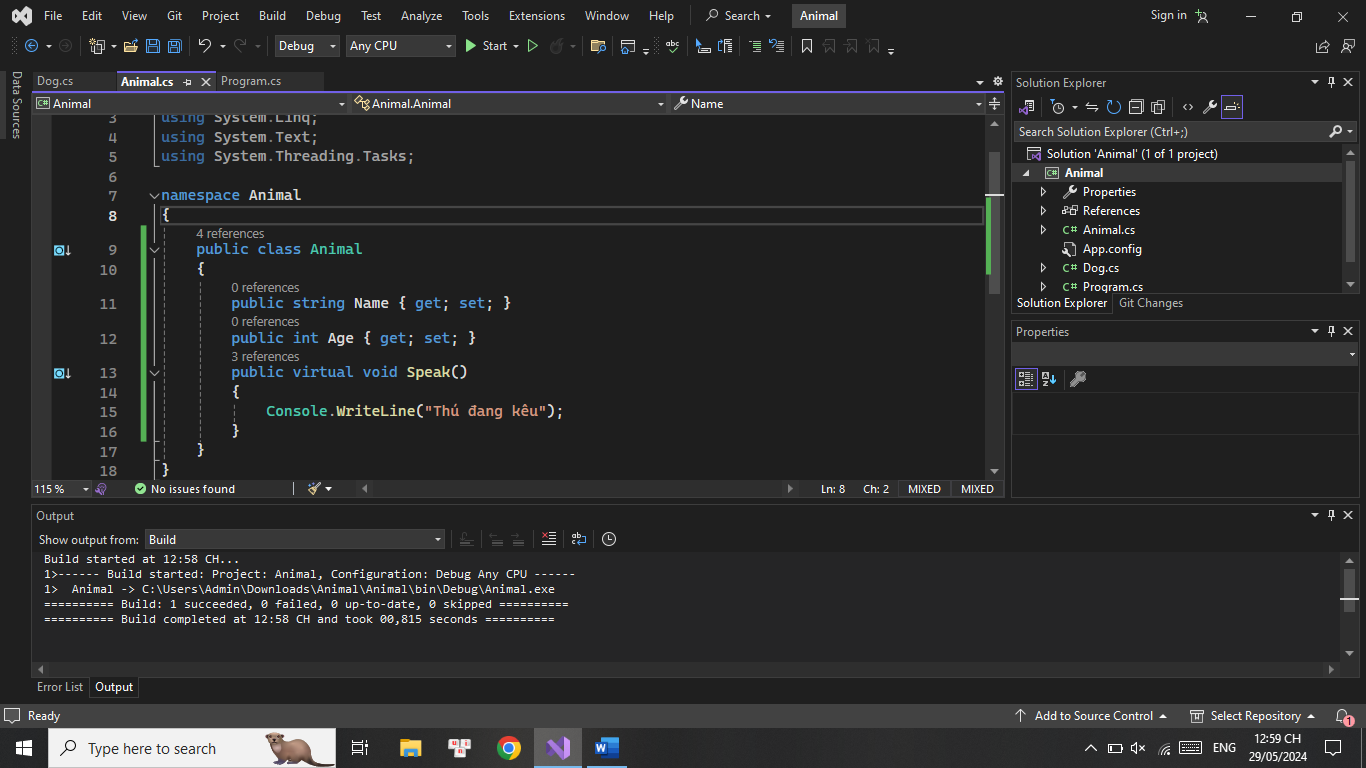
* Tính kế thừa cho phép tạo ra một lớp mới (lớp con) từ một lớp đã tồn tại (lớp cha). Lớp con sẽ thừa hưởng các thuộc tính và phương thức (trừ các thành phần private) của lớp cha, đồng thời có thể bổ sung thêm các thành phần riêng của mình.
* Mục đích:

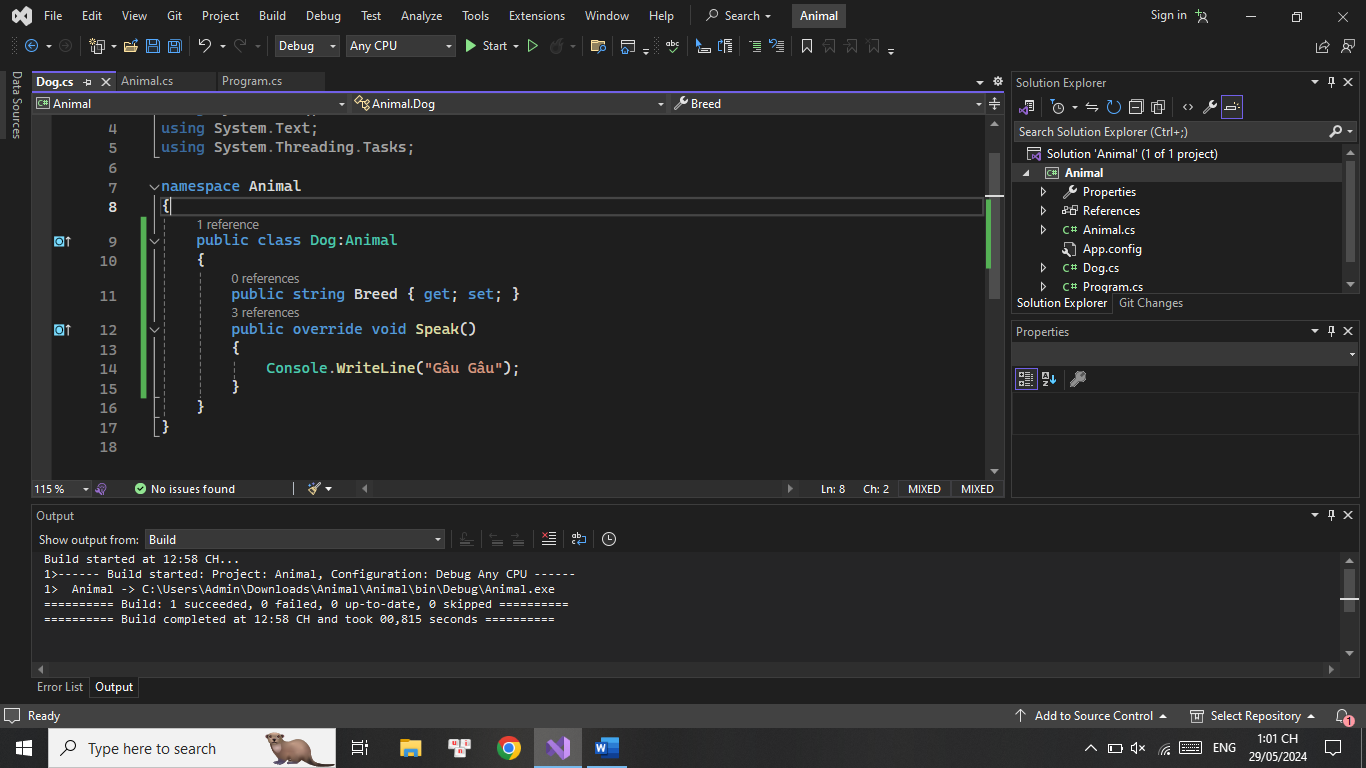
+ Tái sử dụng mã: giúp giảm thiểu việc viết mã lặp lại, tiết kiệm thời gian và công sức.

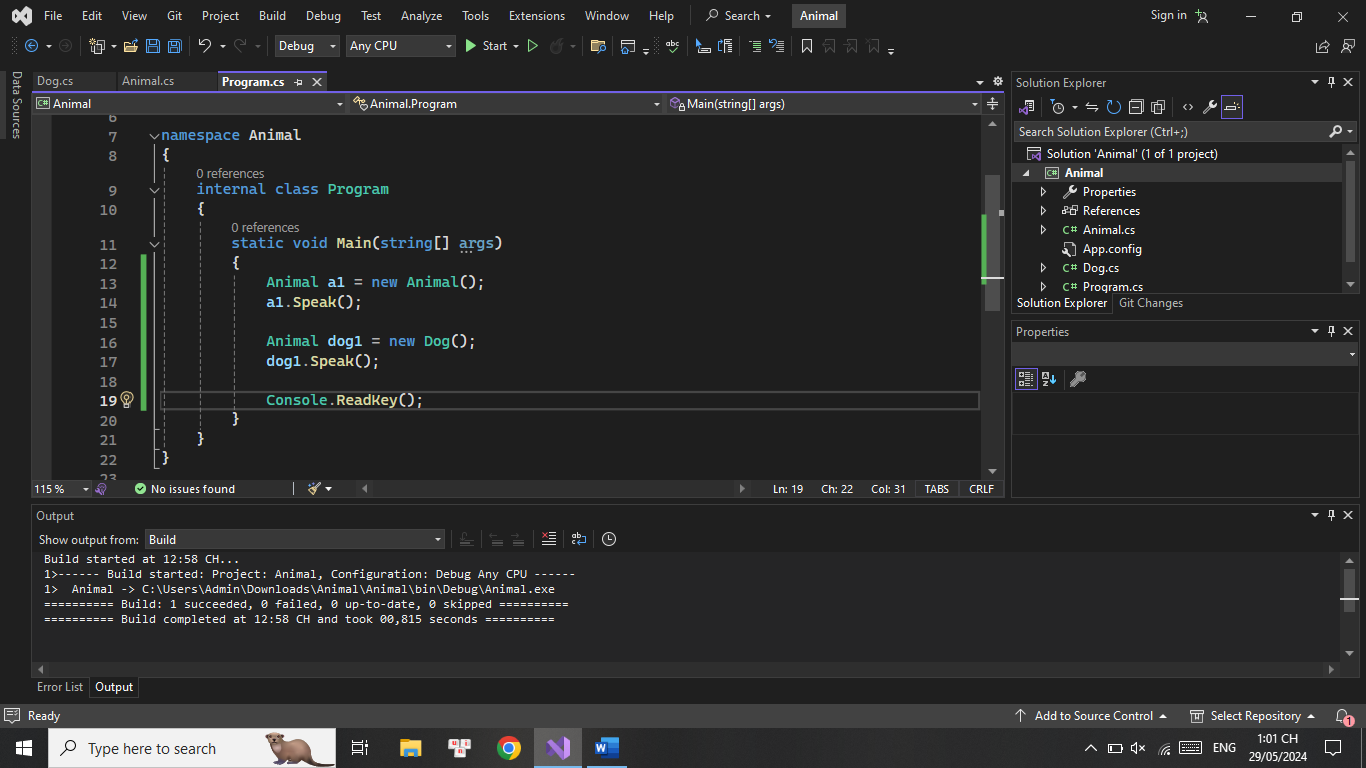
+ Tăng tính mở rộng: Cho phép dễ dàng mở rộng hệ thống bằng cách tạo ra các lớp con mới từ các lớp cha đã có.

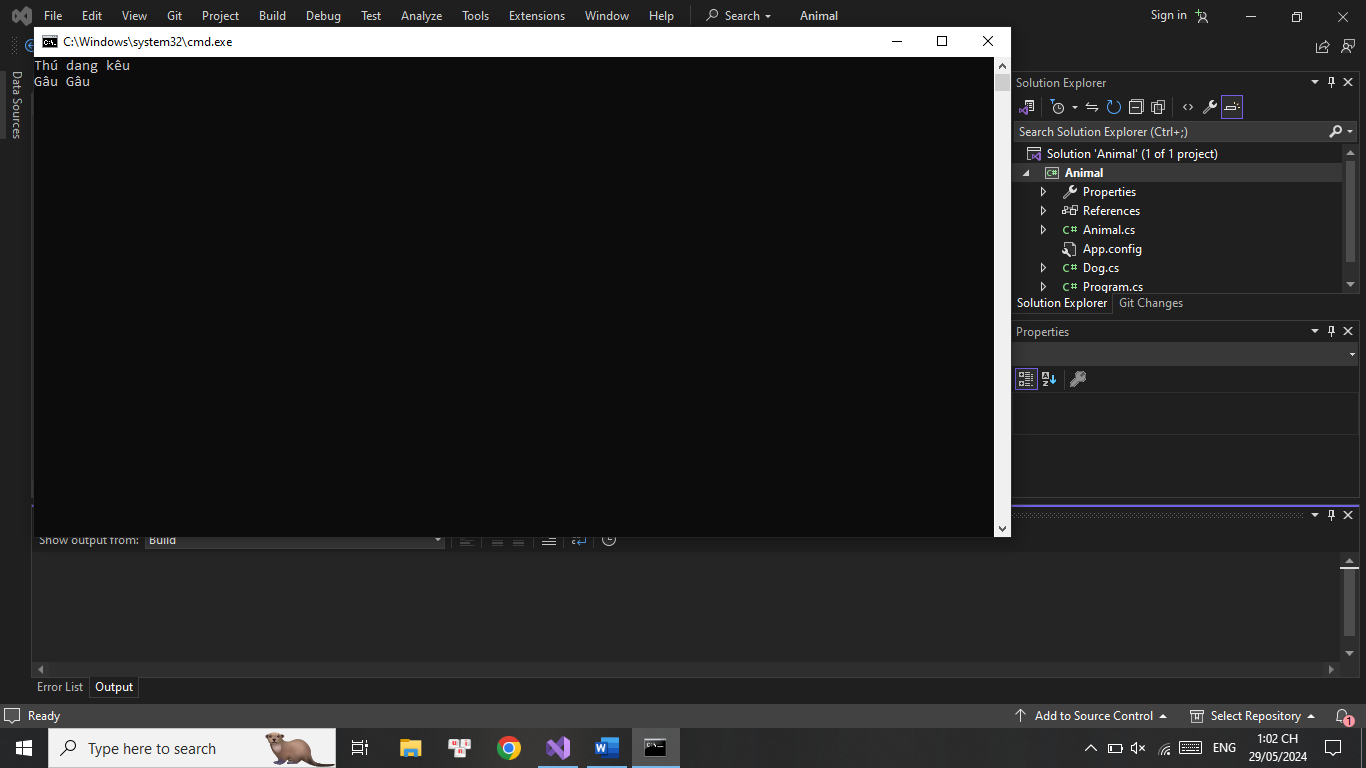
+ Tăng tính bảo trì: Giúp việc bảo trì hệ thống dễ dàng hơn do mã được tổ chức logic và hợp lý.

* Ví dụ minh họa:

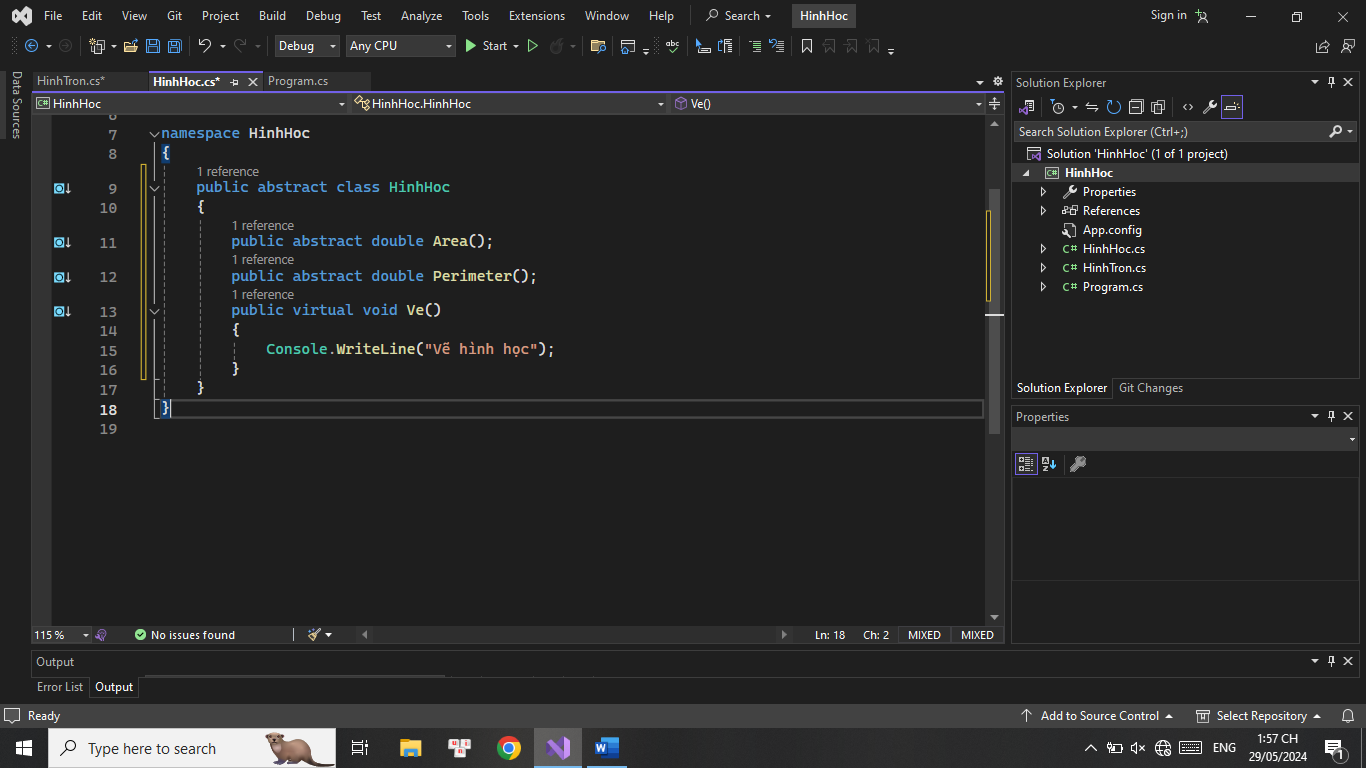


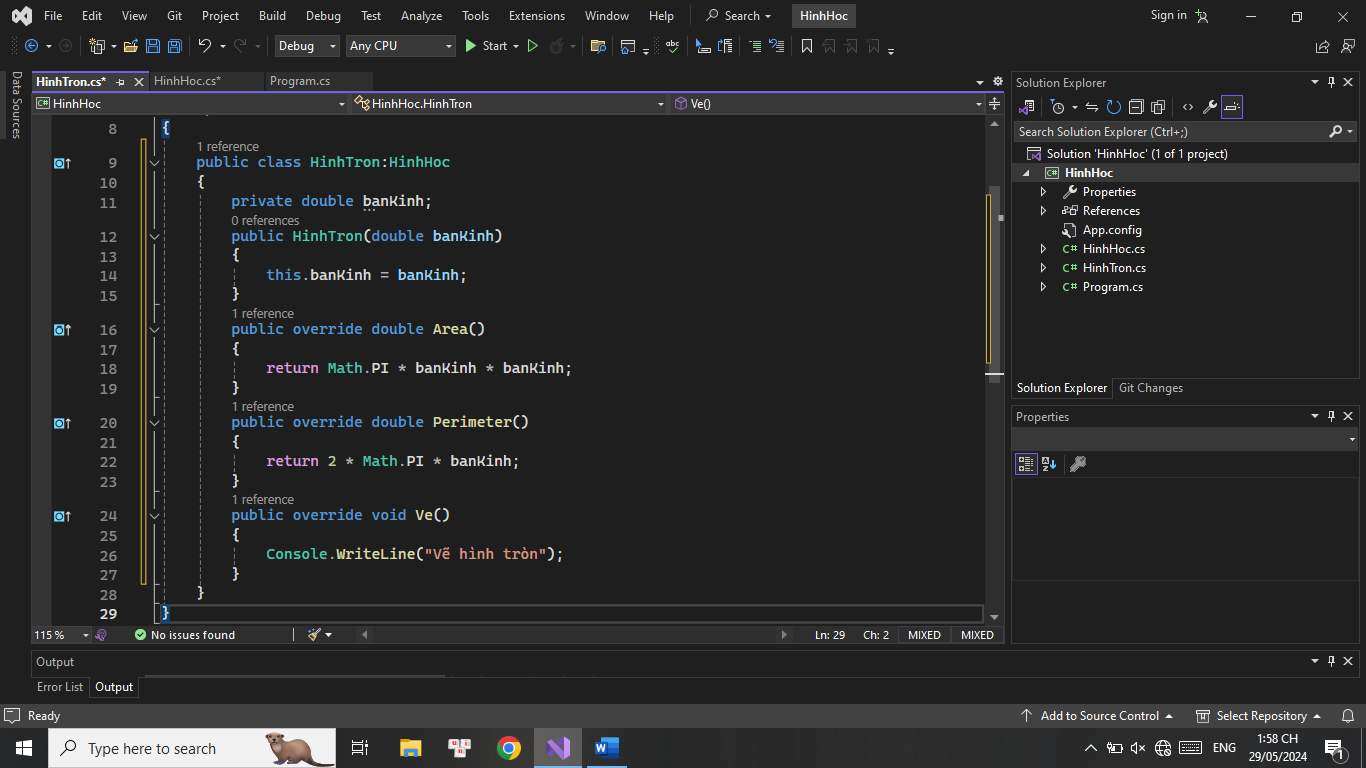


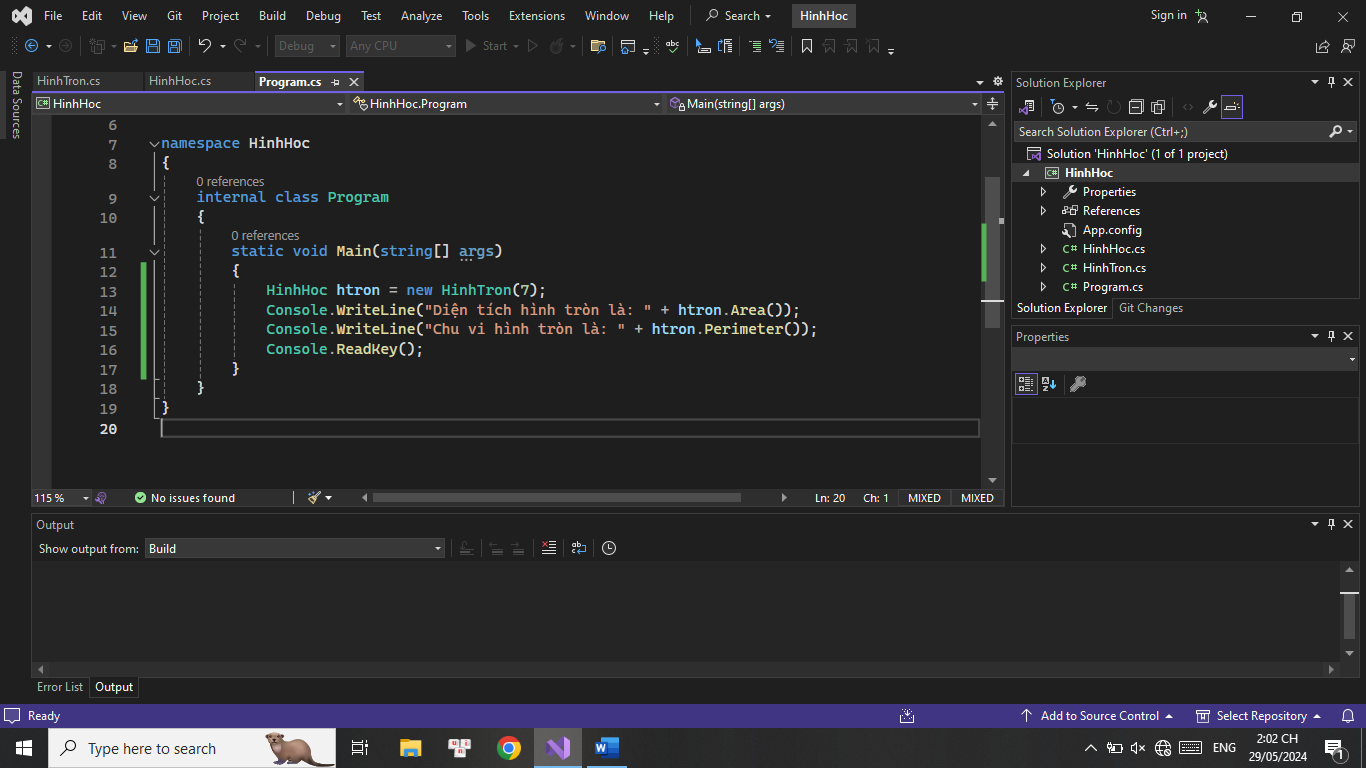


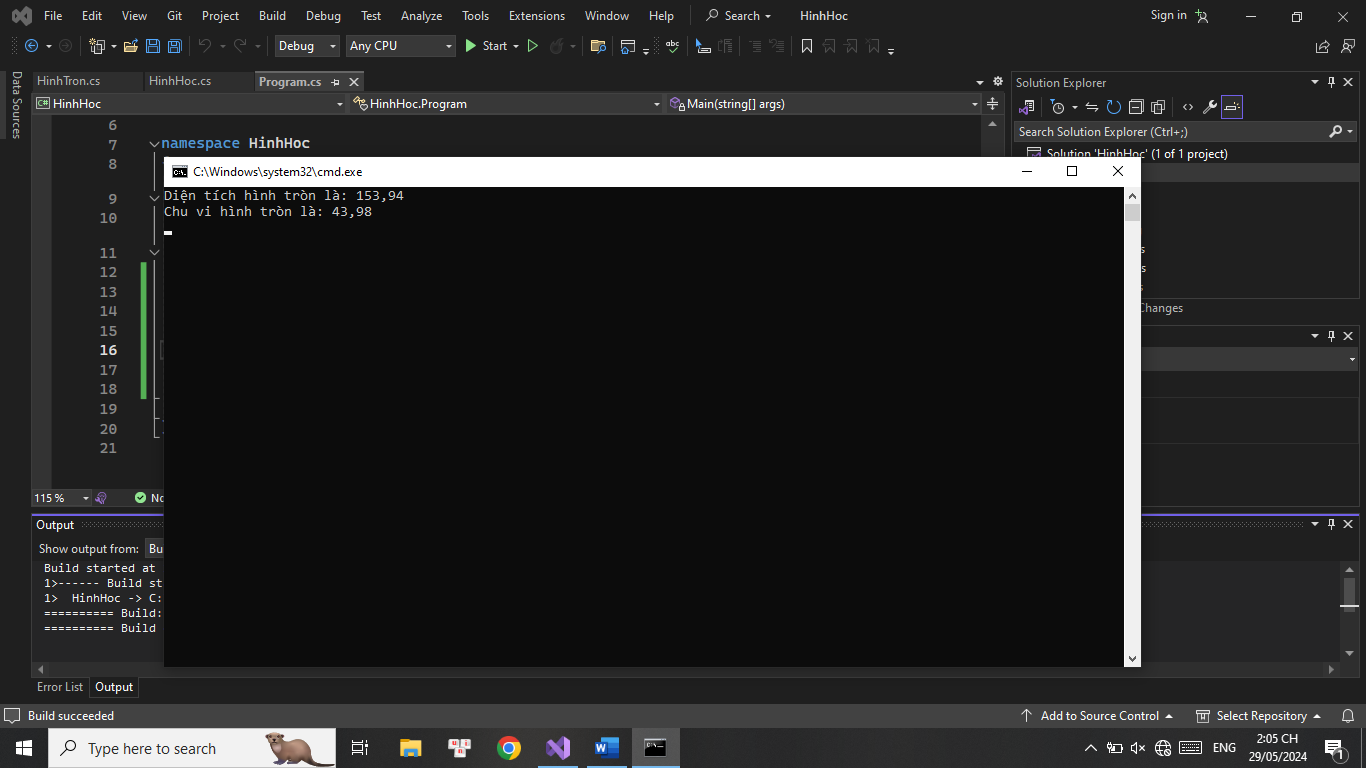


1. **Giải thích sự khác biệt giữa các loại kế thừa: đơn kế thừa, đa kế thừa, và kế thừa đa cấp. Cho ví dụ minh họa:**

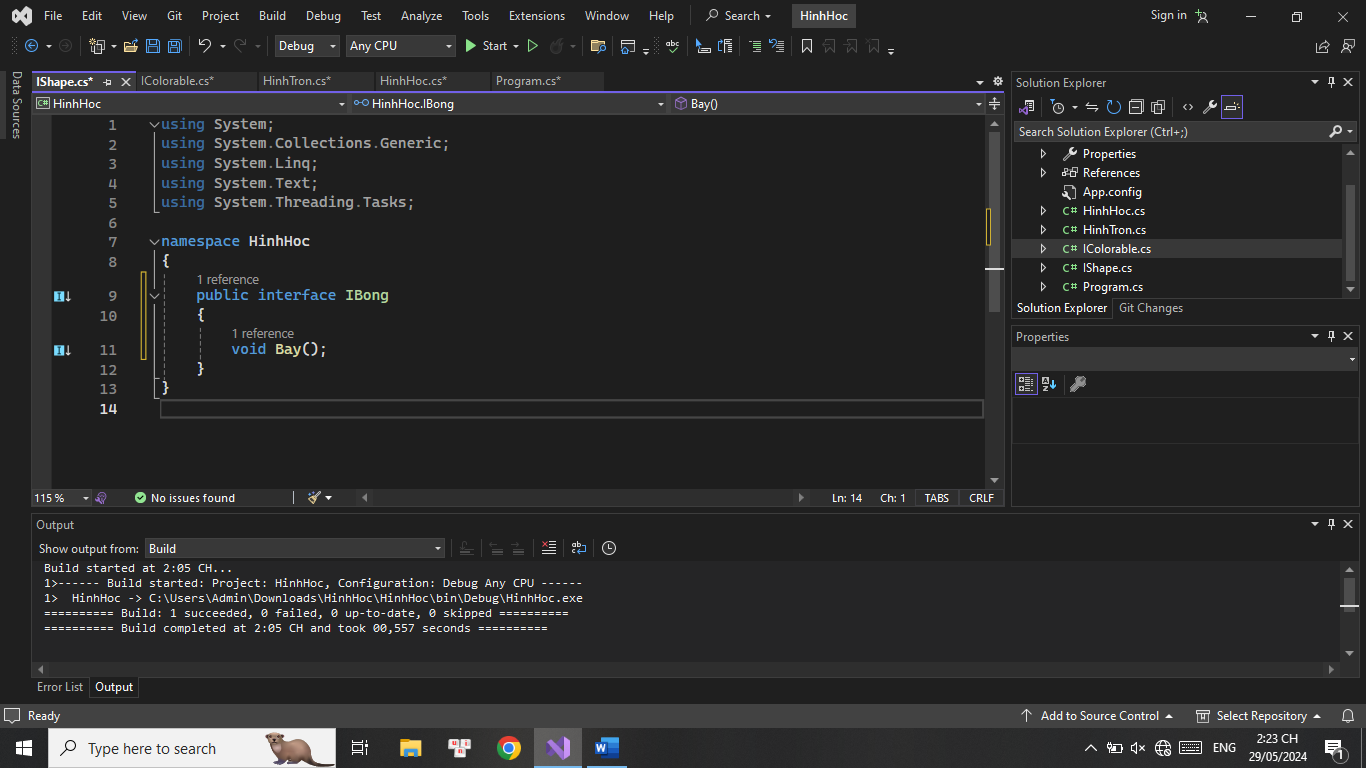
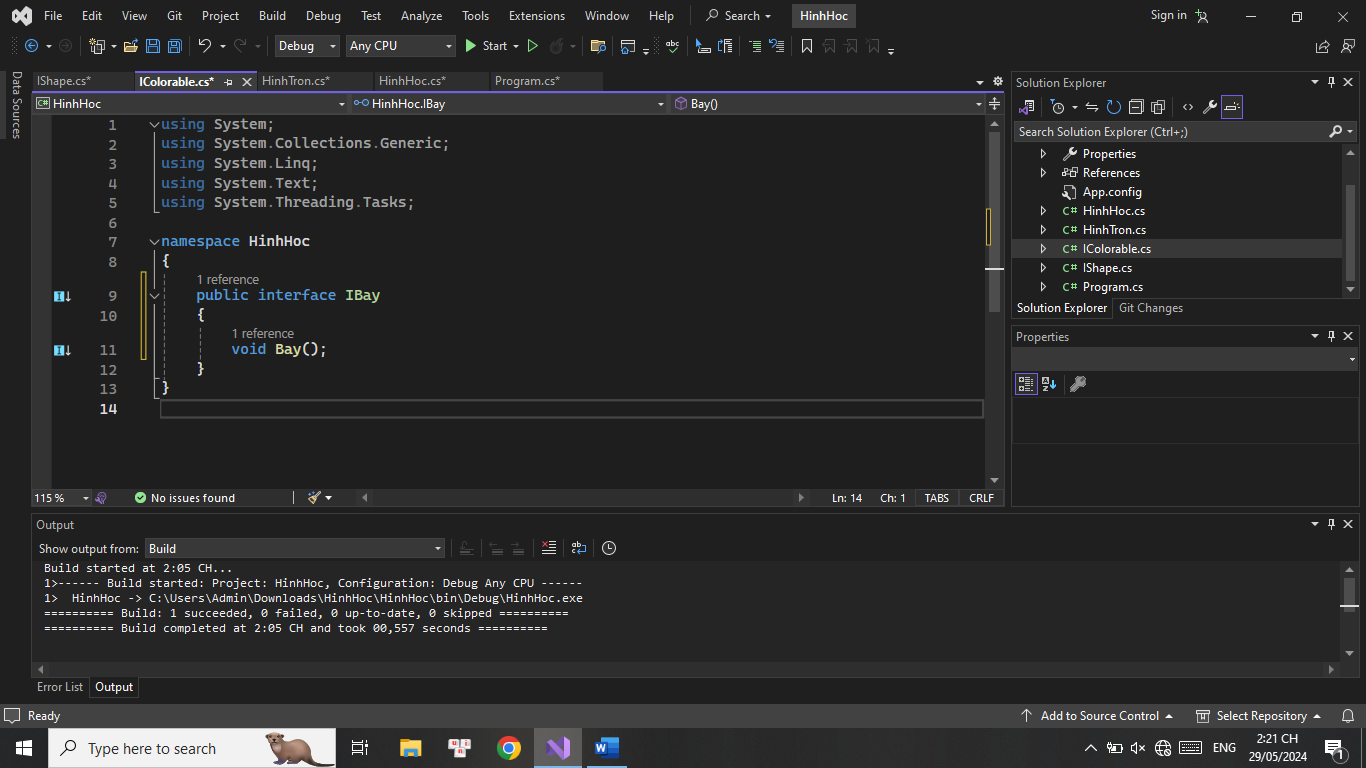
* Đơn kế thừa: là một lớp chỉ được kế thừa từ đúng một lớp khác (lớp con chỉ có duy nhất một lớp cha), là loại kế thừa phổ biến nhất giúp tổ chức mã rõ ràng và dễ dàng quản lý.

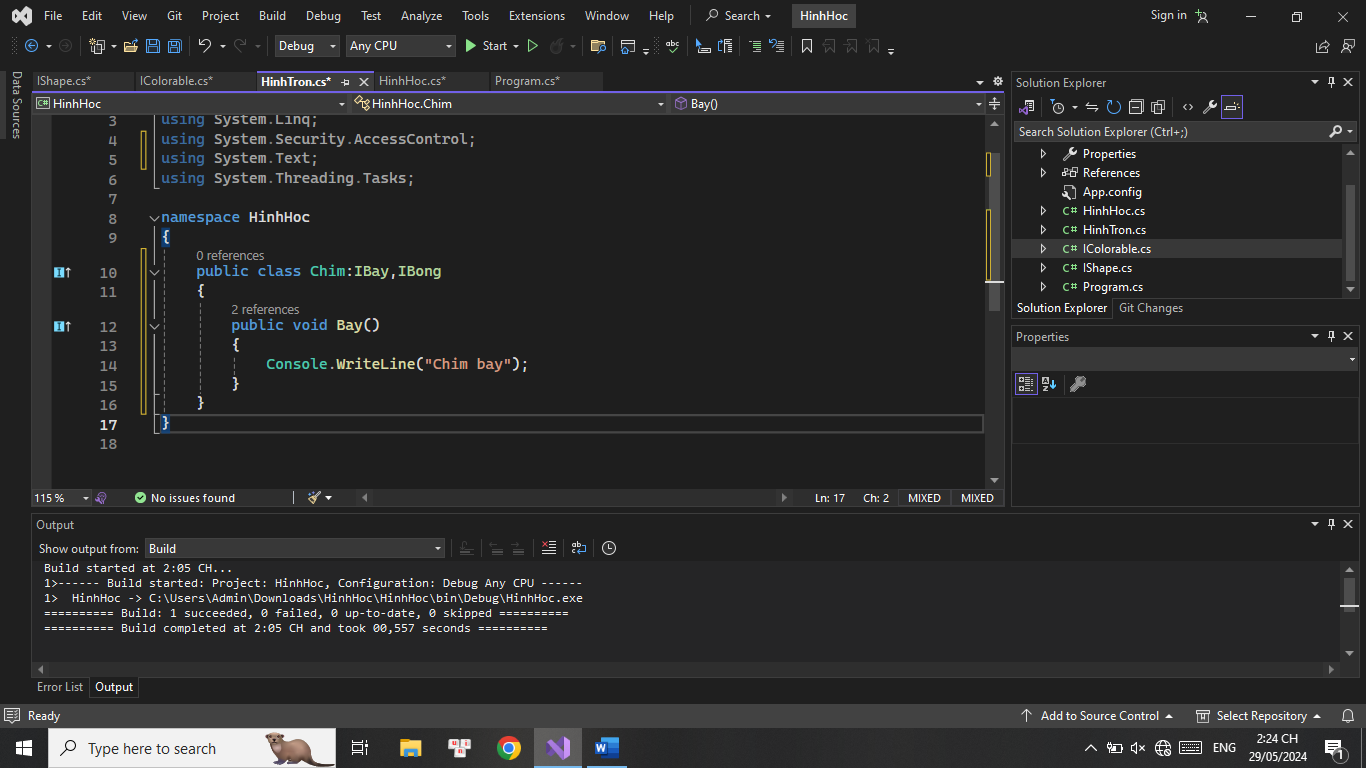




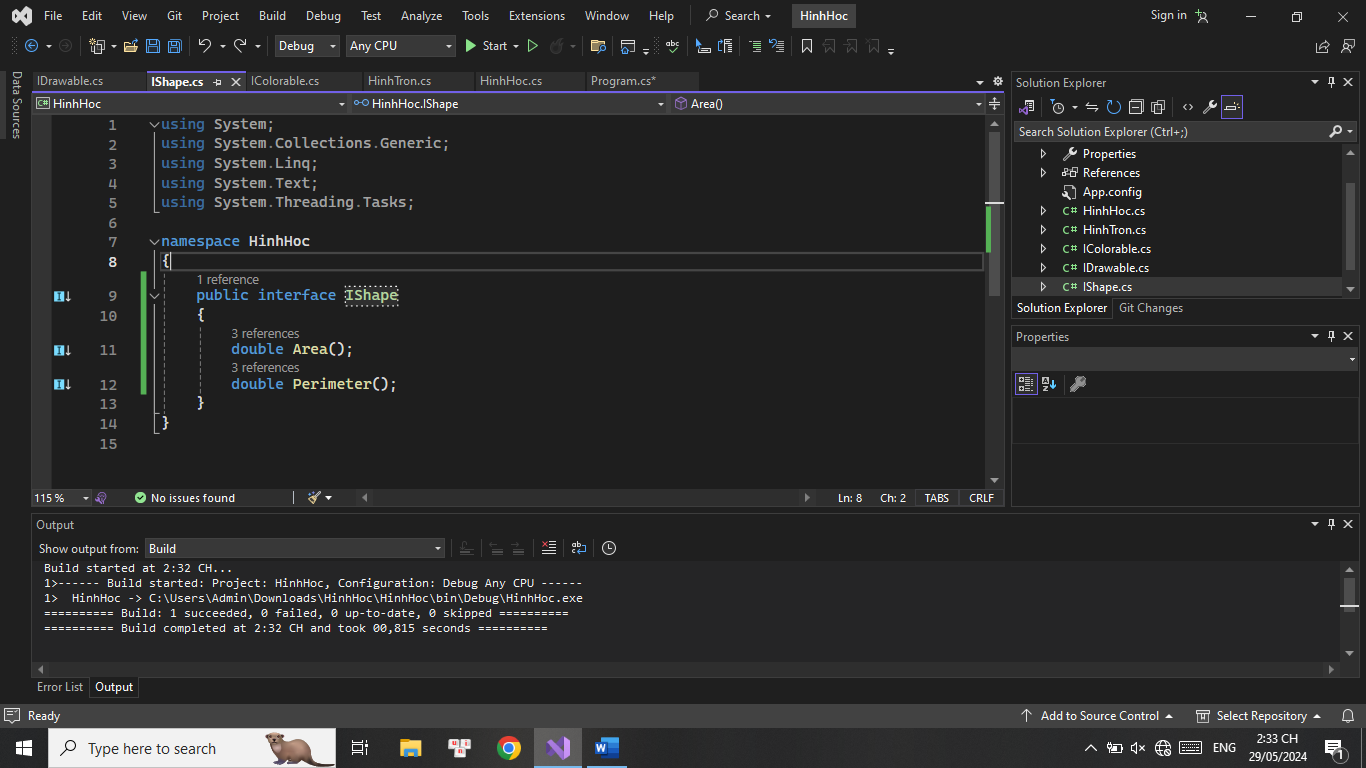


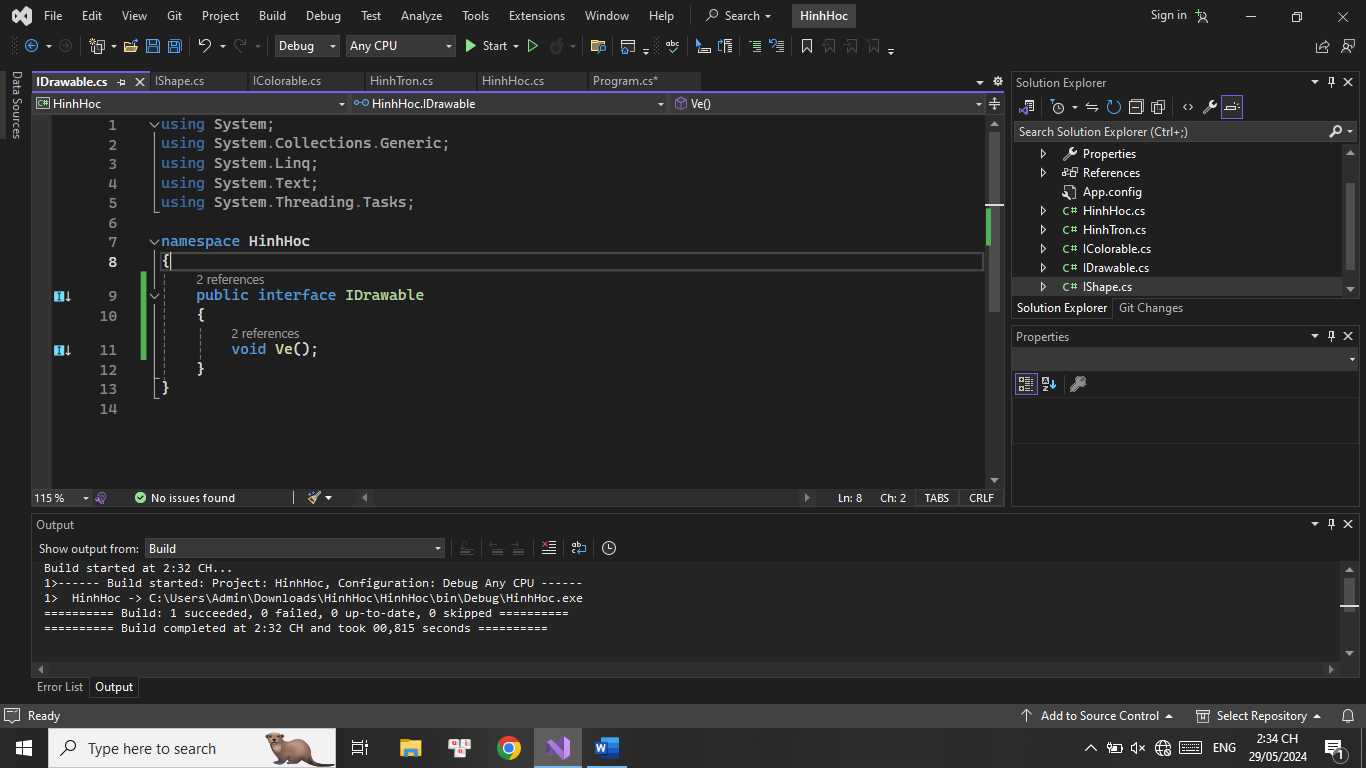
* Đa kế thừa: là một lớp con kế thừa từ nhiều lớp cha, ít được sử dụng vì có thể dẫn đến sự phức tạp và mâu thuẫn trong mã.

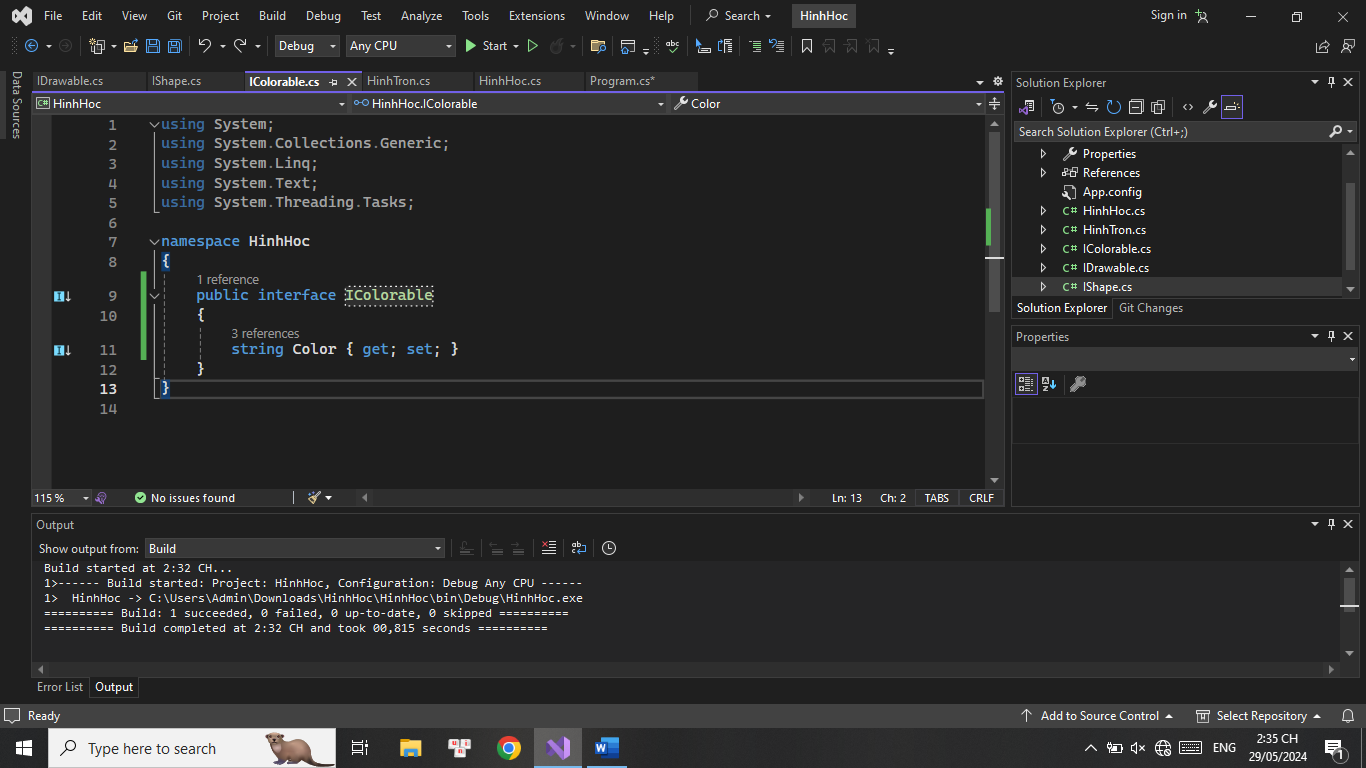


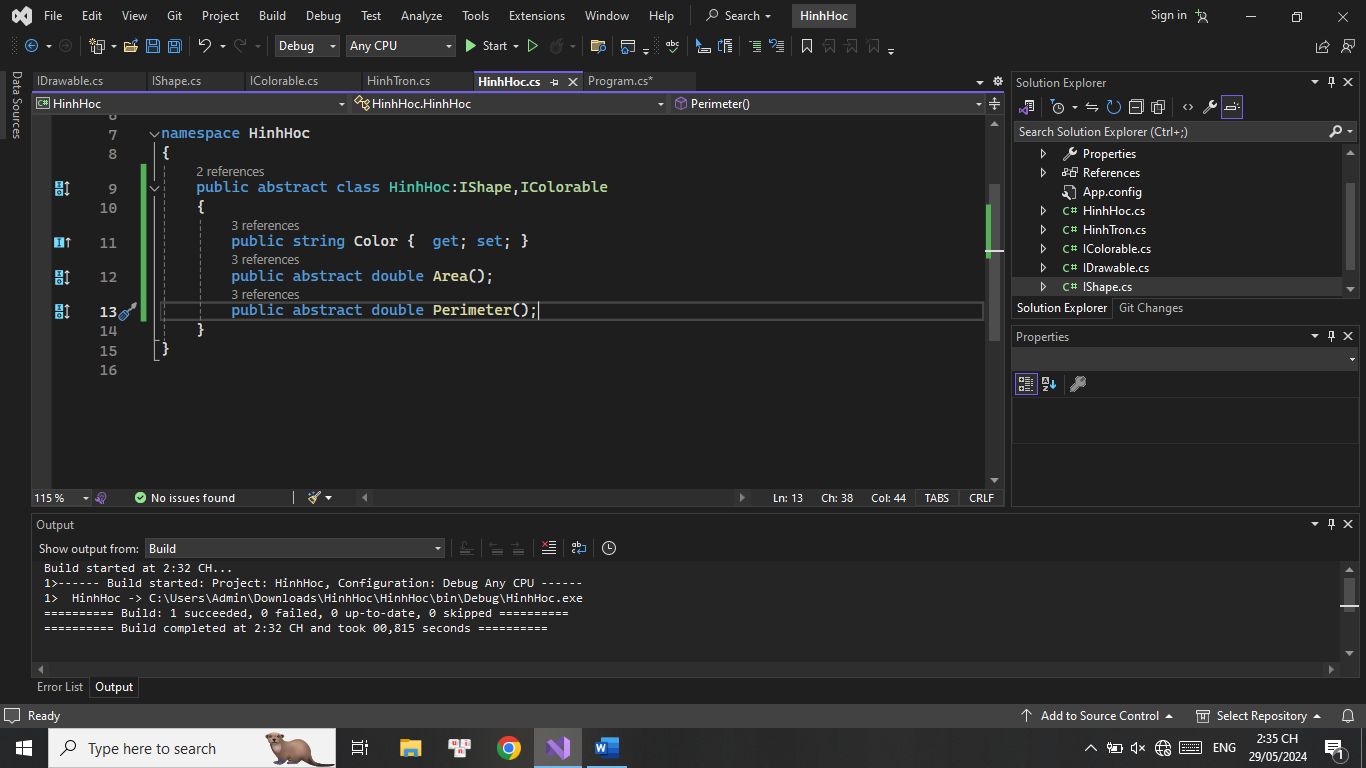


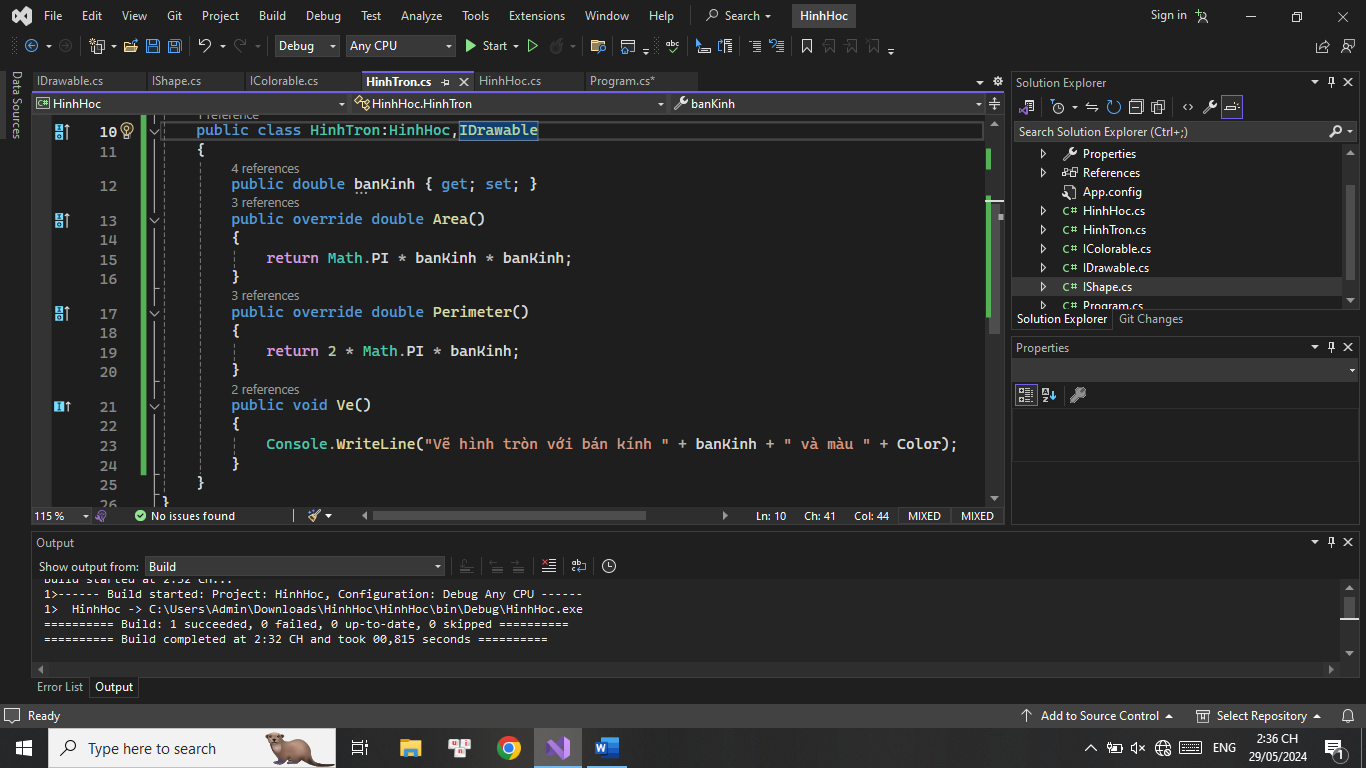
* Kế thừa đa cấp: là một lớp con kế thừa từ một lớp cha, lớp cha đó lạ kế thừa từ một lớp cha khác, tạo ra mối quan hệ phân cấp giữa các lớp.

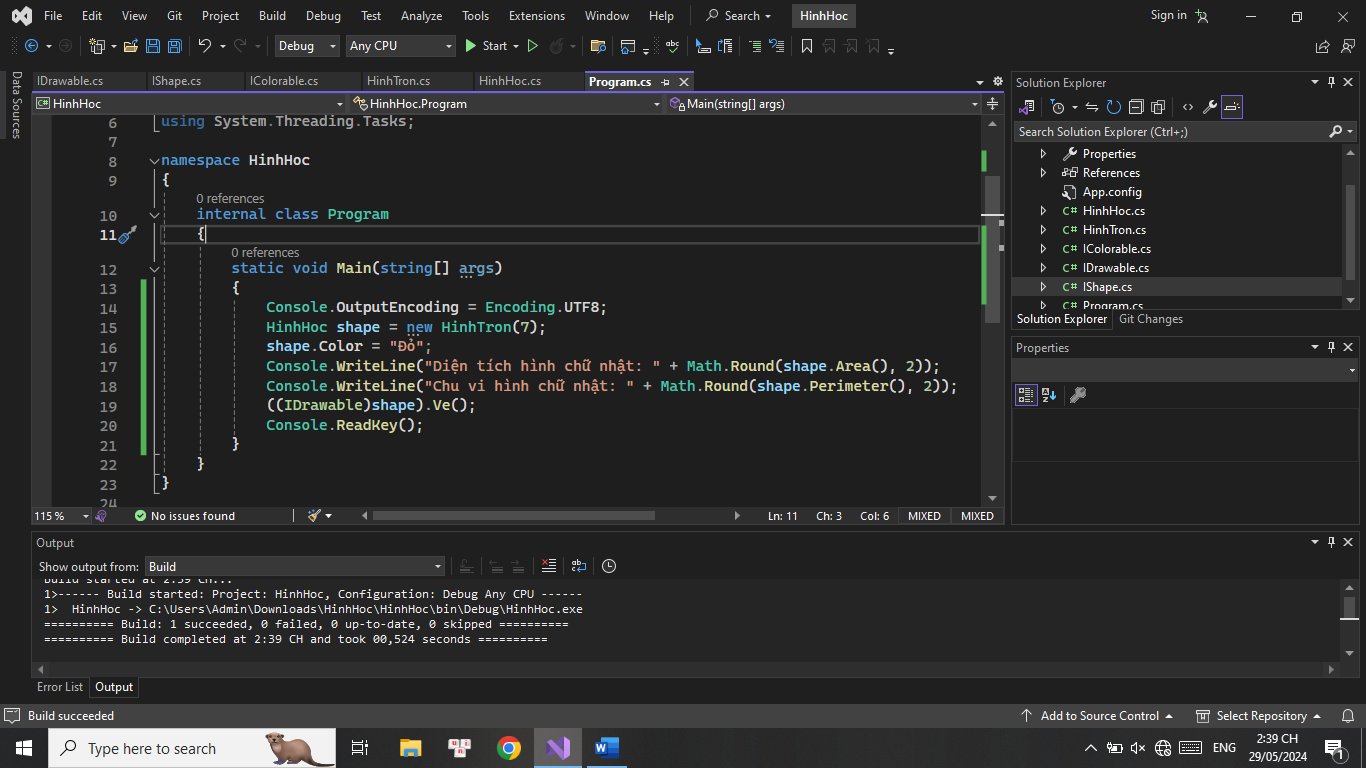


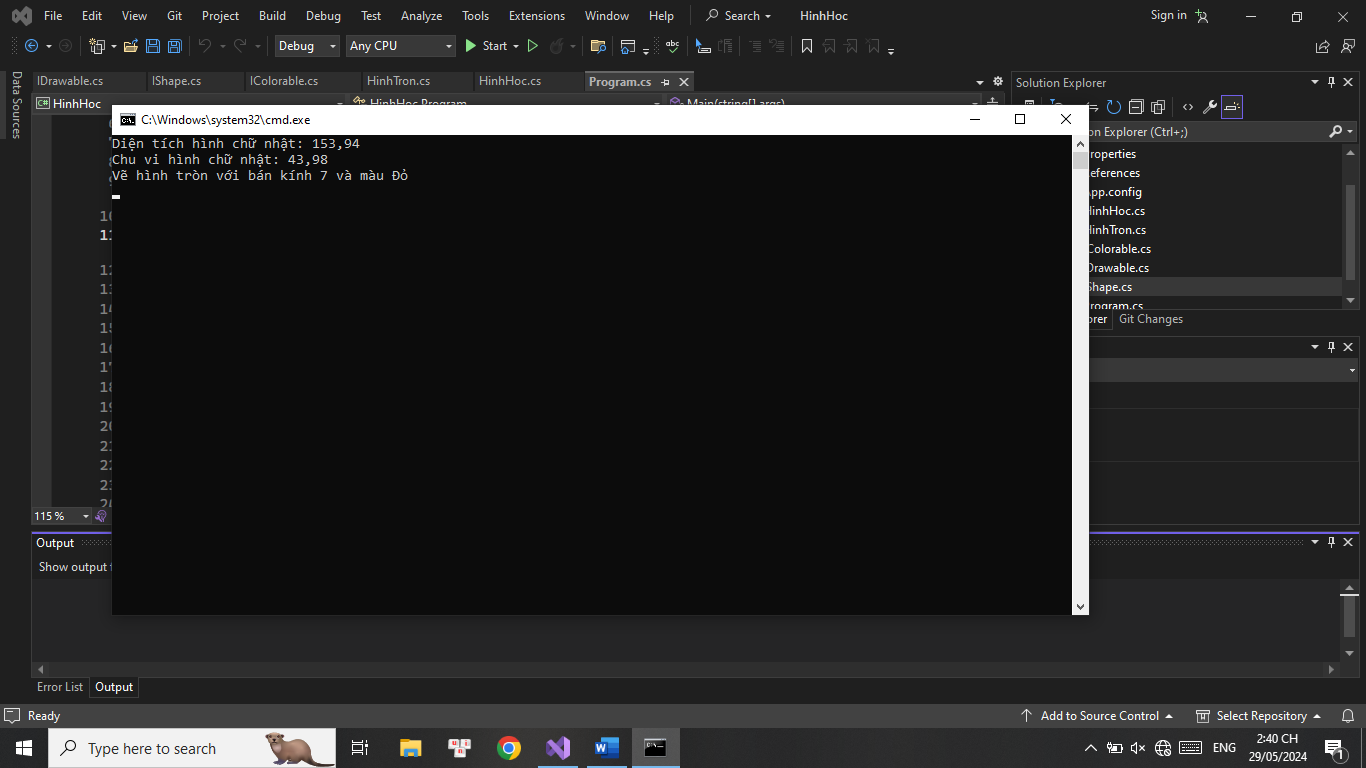






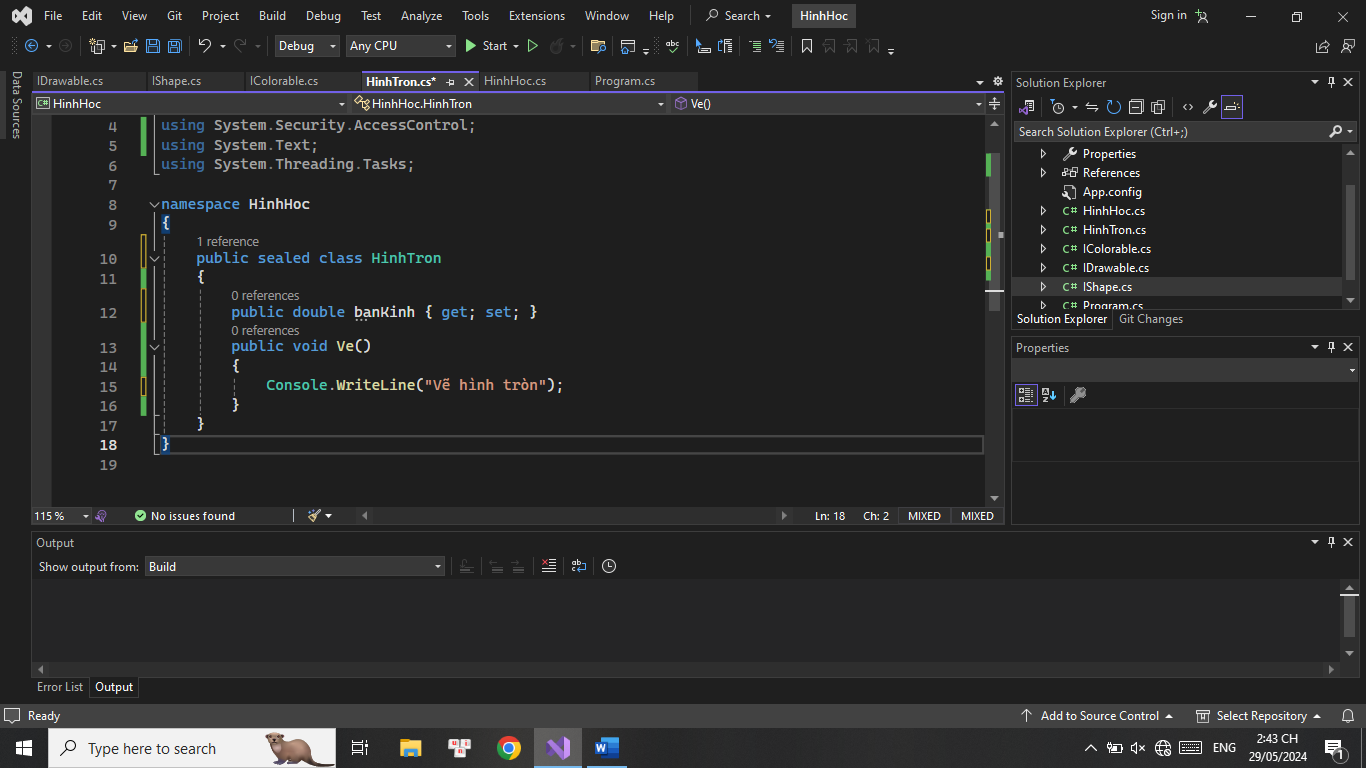






1. **Làm thế nào để ngăn chặn một lớp bị kế thừa? Cho ví dụ minh họa.**

* Sử dụng từ khóa ***“Sealed”*** để ngăn chặn một lớp bị kế thừa vì theo ý nghĩa của nó, khi sử dụng từ khóa ***“Sealed”*** cho lớp thì lớp đó không được kế thừa, còn khi sử dụng cho phương thức thì phương thức đó không được phép overrride.
* Ví dụ minh họa:



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.**

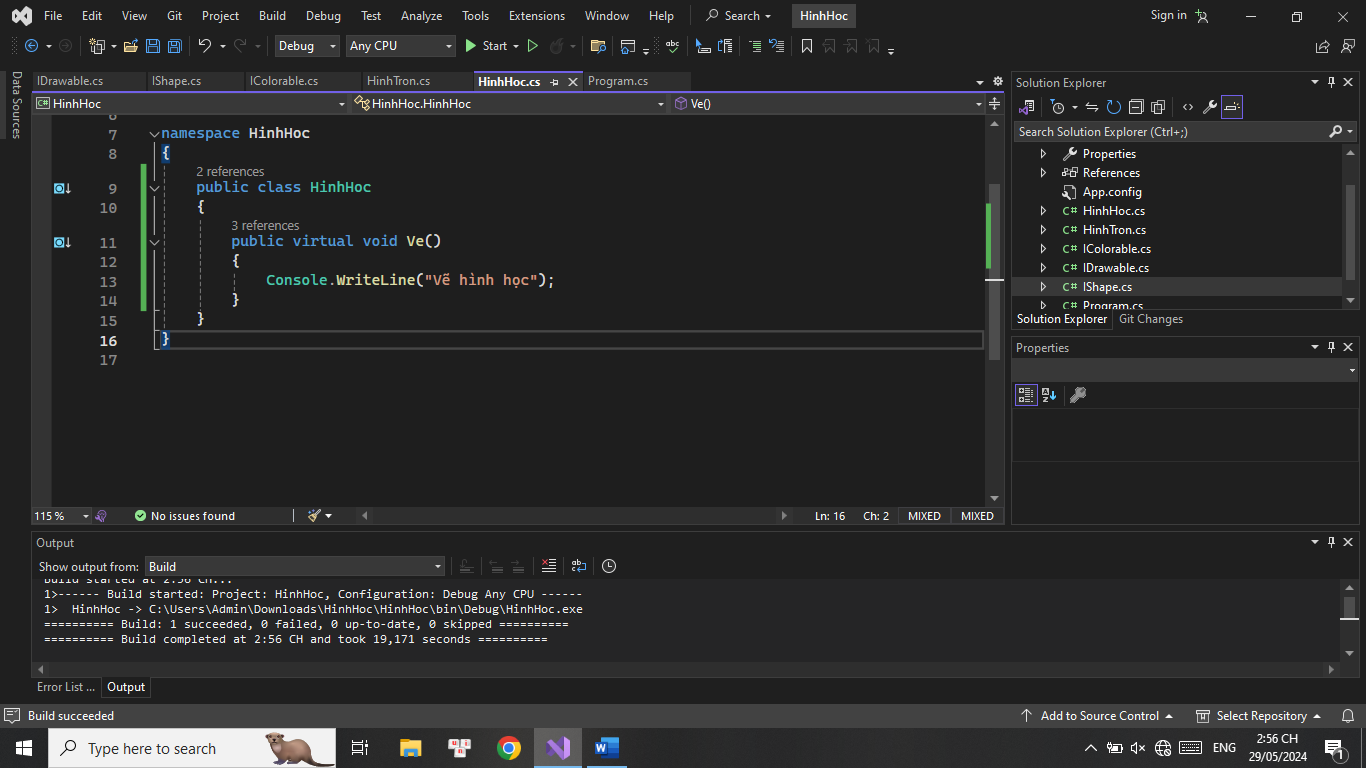
* Từ khóa ***“base”*** được sử dụng để truy cập các thuộc tính và phương thức của lớp cơ sở từ lớp con.
* Mục đích sử dụng:

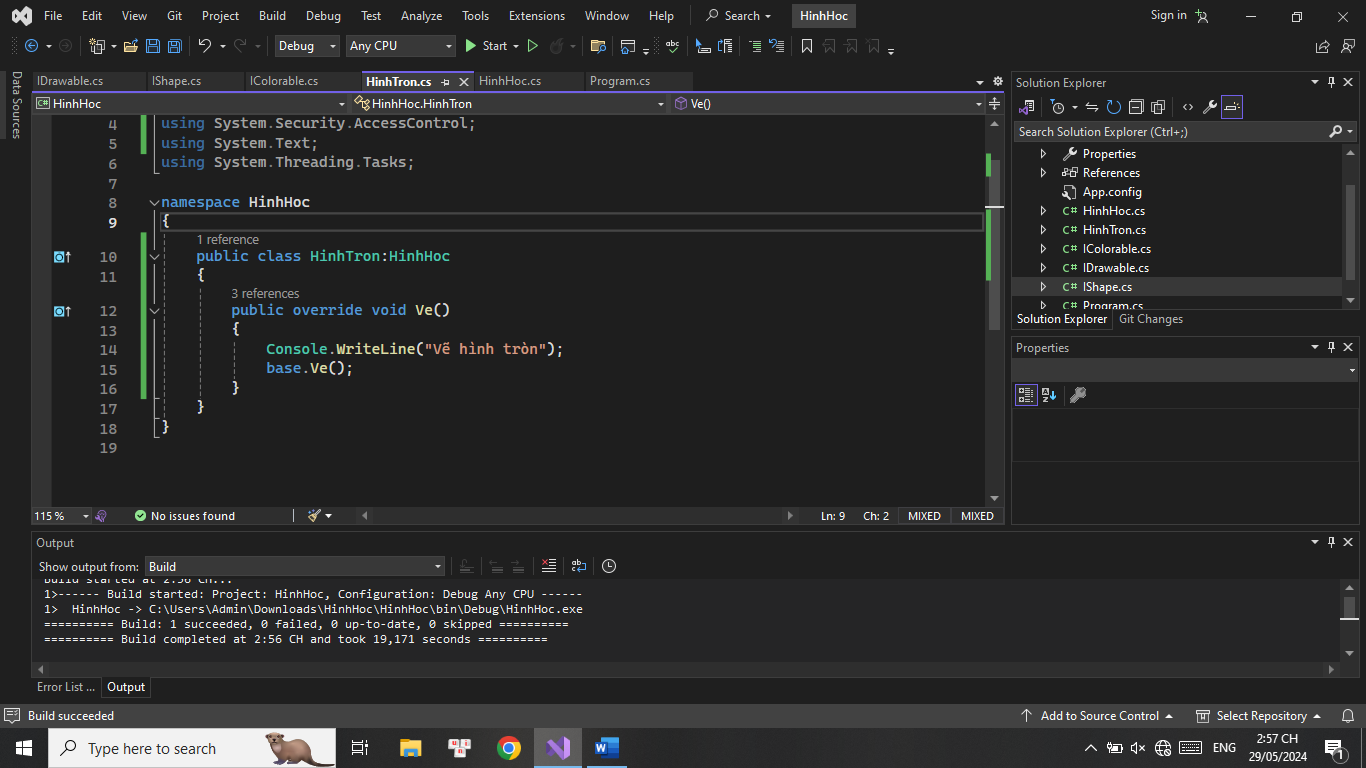
+ Gọi phương thức của lớp cơ sở: khi một lớp con override một phương thức của lớp cơ sở, có thể sử dụng ***“base”*** để gọi phiên bản gốc (base) của phương thức đó từ lớp con.

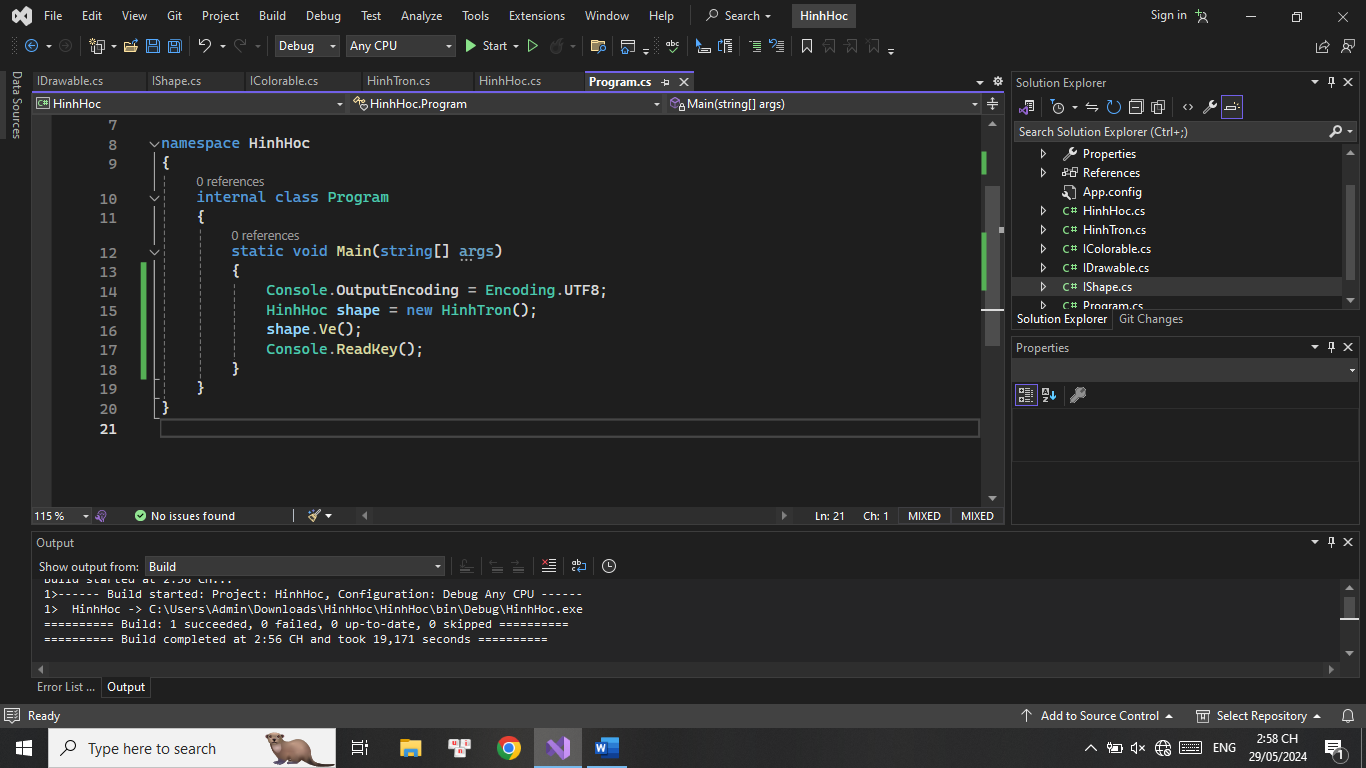
+ Truy cập thuộc tính của lớp cơ sở: ***“base”*** cho phép truy cập trực tiếp vào các thuộc tính của lớp cơ sở từ lớp con.

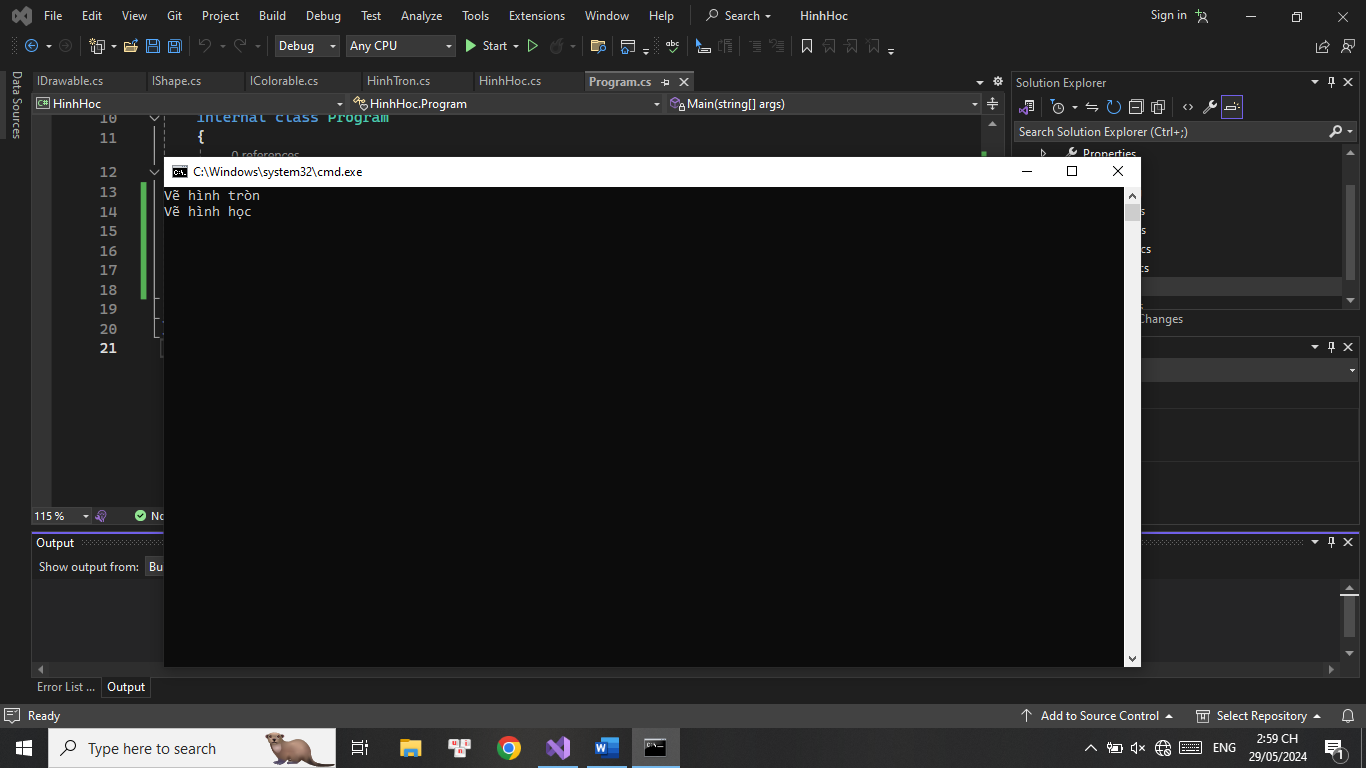
+ Gọi hàm khởi tao của lớp cơ sở: khi tạo một đối tượng của lớp con, có thể sử dụng ***“base”*** để gọi hàm khởi tạo của lớp cơ sở.

* Ví dụ minh họa gọi phương thức của lớp cơ sở:

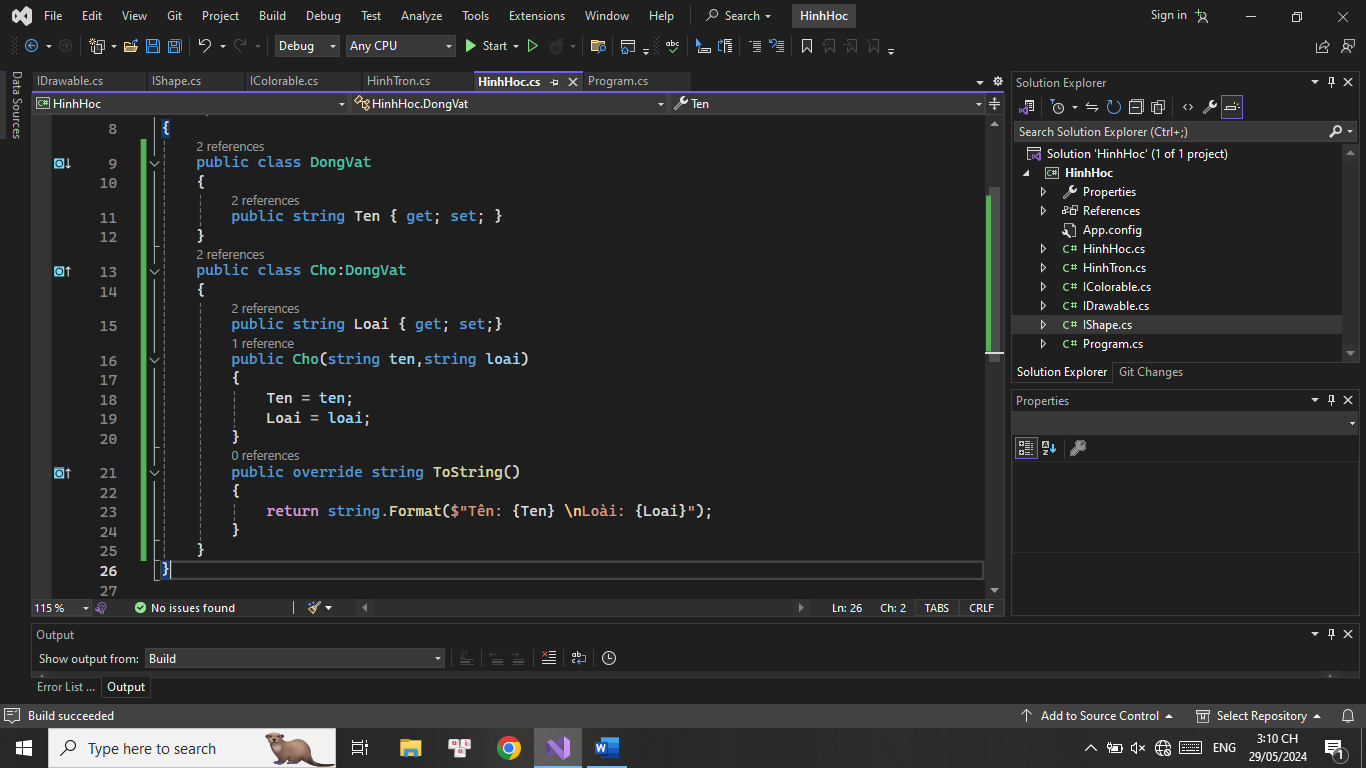


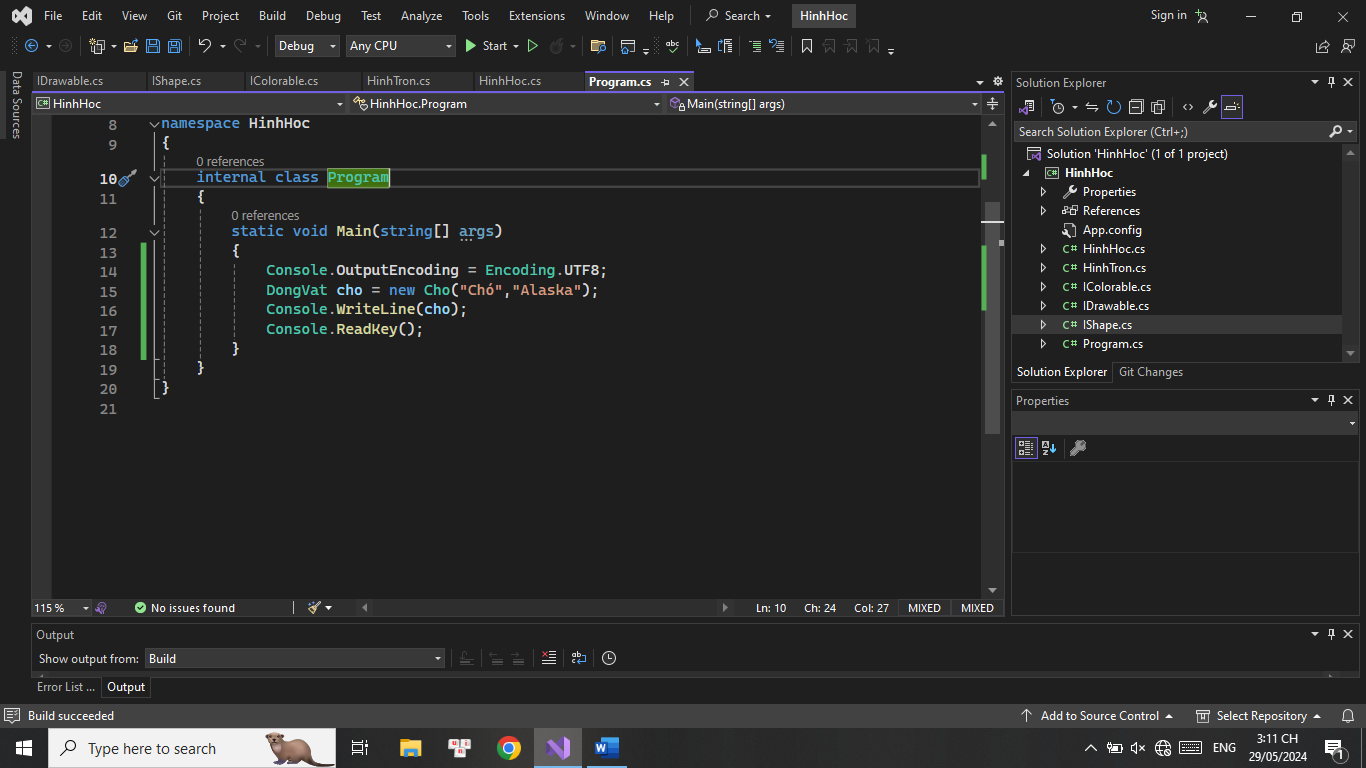


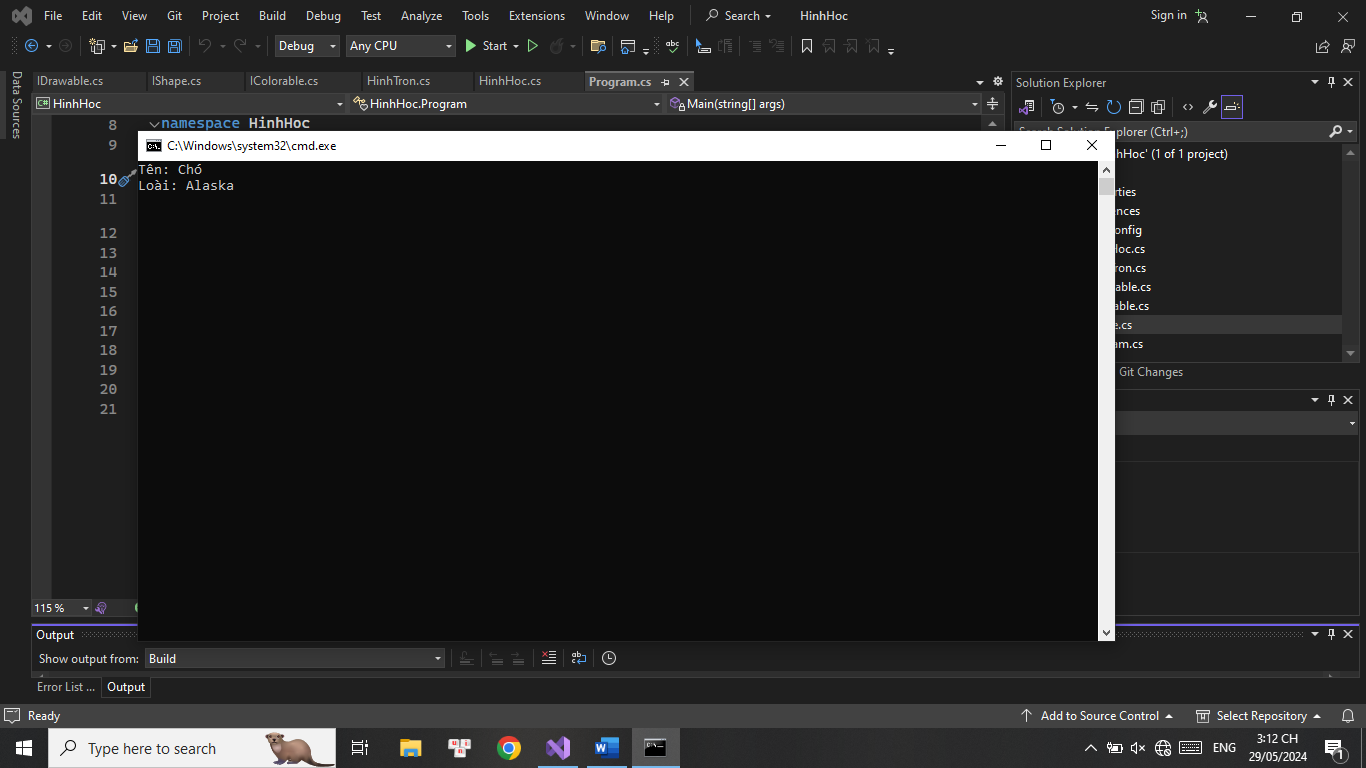




* Ví dụ minh họa truy cập thuộc tính của lớp cơ sở:







* Ví dụ minh họa gọi hàm khởi tạo của lớp cơ sở:

