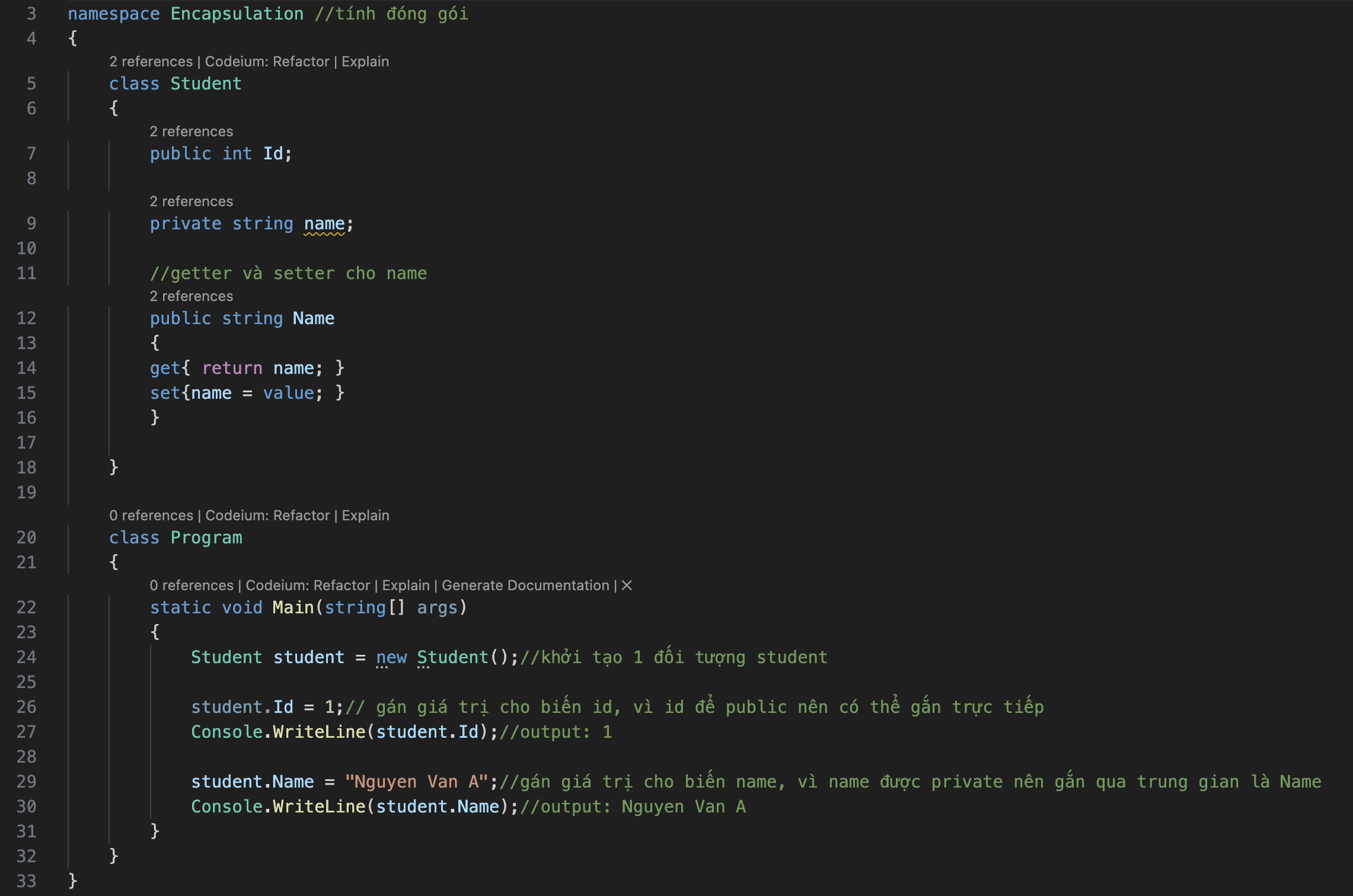
**Tính đóng gói**

Tính đóng gói (encapsulation) "đóng gói" thuộc tính và phương thức của đối tượng (hoặc lớp) thông qua việc giới hạn quyền truy cập (hoặc thay đổi) giá trị của thuộc tính hoặc quyền gọi phương thức.

Nói cách khác tính đóng gói cho phép kiểm soát quyền truy cập (và thay đổi) giá trị của thuộc tính hoặc quyền gọi phương thức của đối tượng (hoặc lớp) và đối tượng (hoặc lớp) con.

Một lớp được đóng gói đầy đủ có các hàm getter và setter được sử dụng để đọc và ghi dữ liệu. Lớp này không cho phép truy cập dữ liệu trực tiếp.



**Tính đa hình**

Tính đa hình trong C# nghĩa là có nhiều hình thái. Trong lập trình hướng đối tượng, tính đa hình thường được diễn đạt như là "một Interface, nhiều hàm".

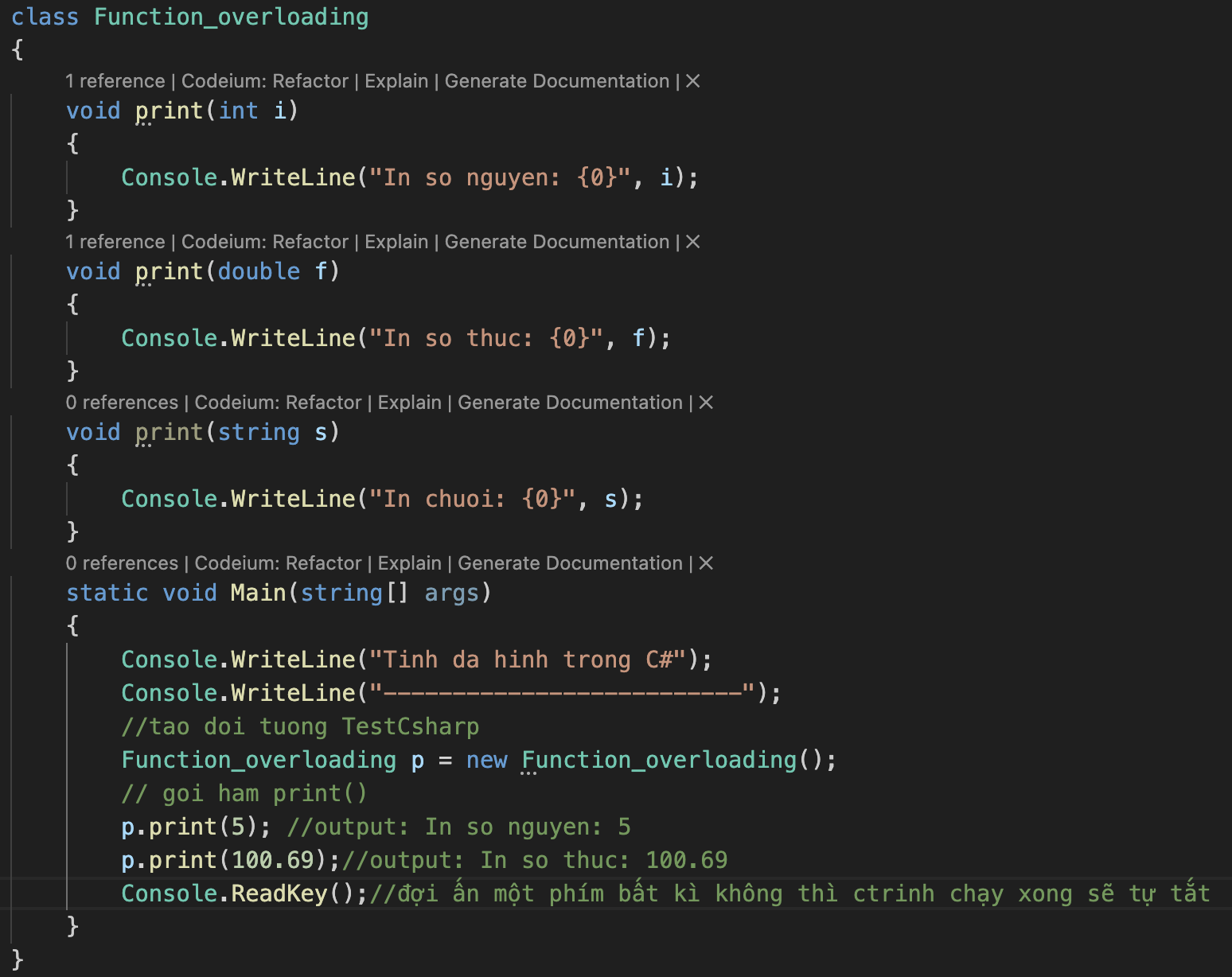
Tính đa hình trong C# có thể là **static** hoặc **dynamic**. Trong đó, kiểu **đa hình static** có thể được gọi là đa hình tĩnh và kiểu **đa hình dynamic** có thể được gọi là đa hình động.

Đa hình static trong C#

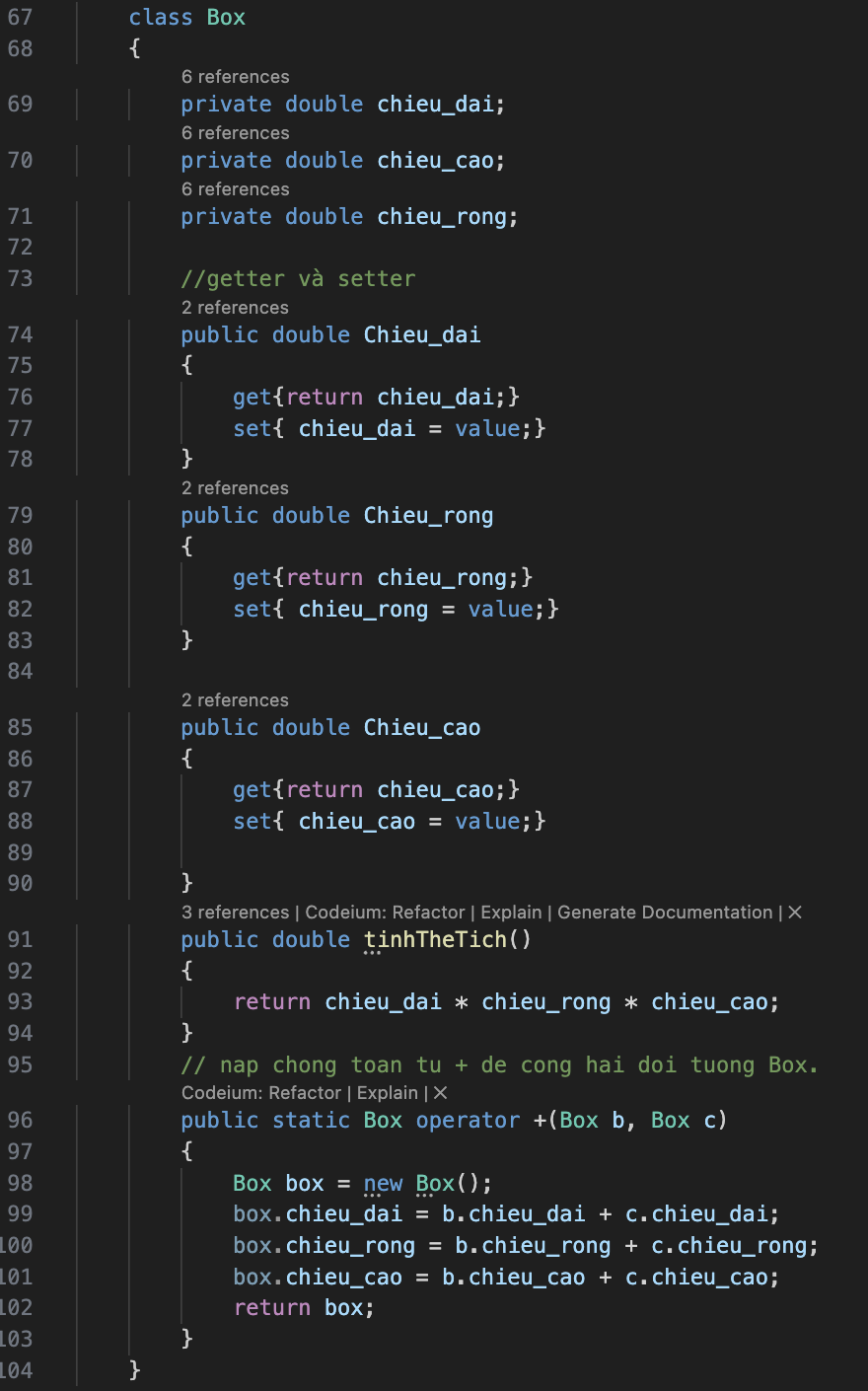
Trong đa hình tĩnh, phần phản hồi tới một hàm được xác định tại compile time(thời gian biên dịch). Trong khi đó với đa hình động, nó được quyết định tại runtime.

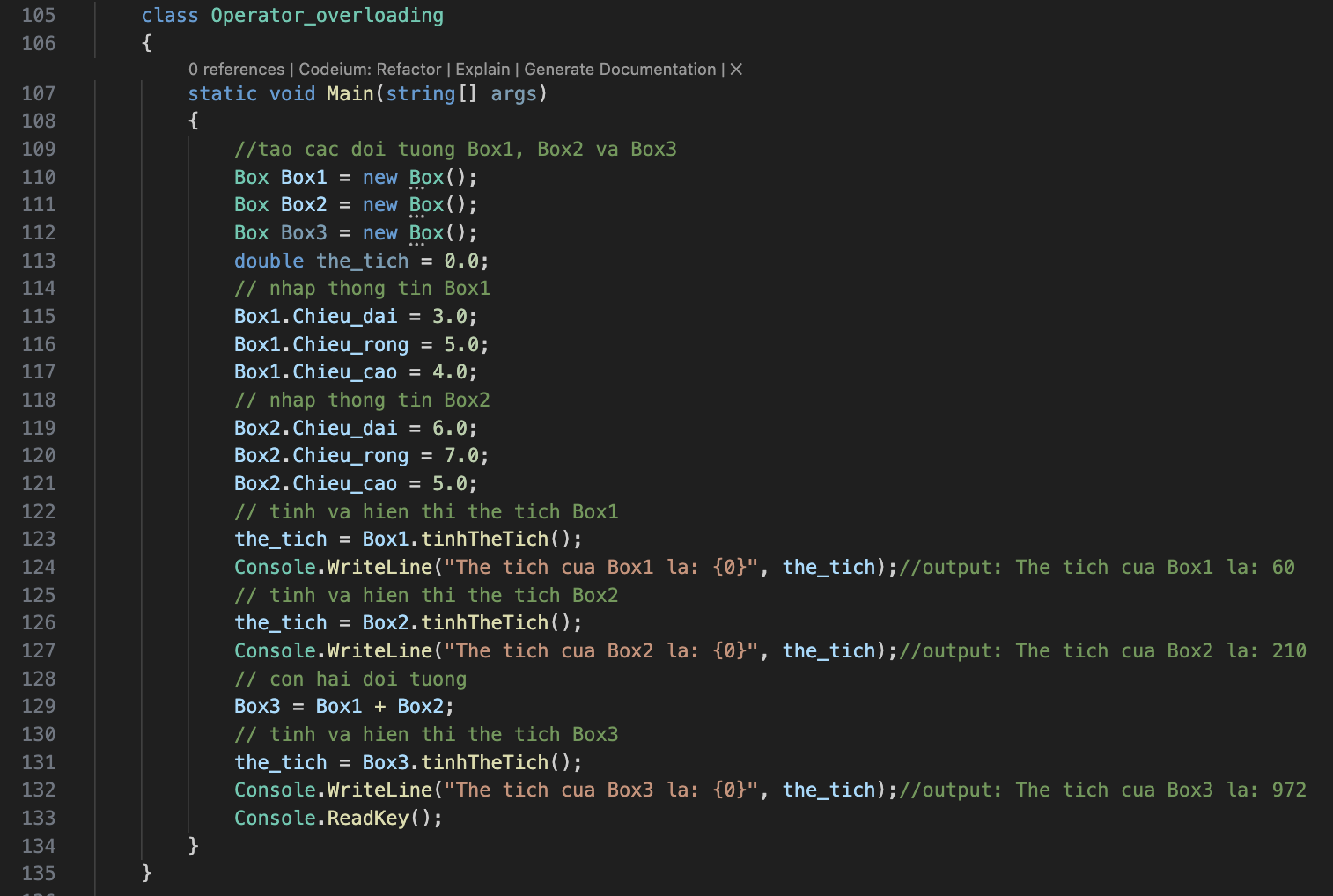
Kỹ thuật liên kết một hàm với một đối tượng trong thời gian biên dịch được gọi là Early Binding. Nó cũng được gọi là Static Binding. C# cung cấp hai kỹ thuật để triển khai đa hình tĩnh. Chúng là:

* Nạp chồng hàm (Function overloading)



* Nạp chồng toán tử (Operator overloading)

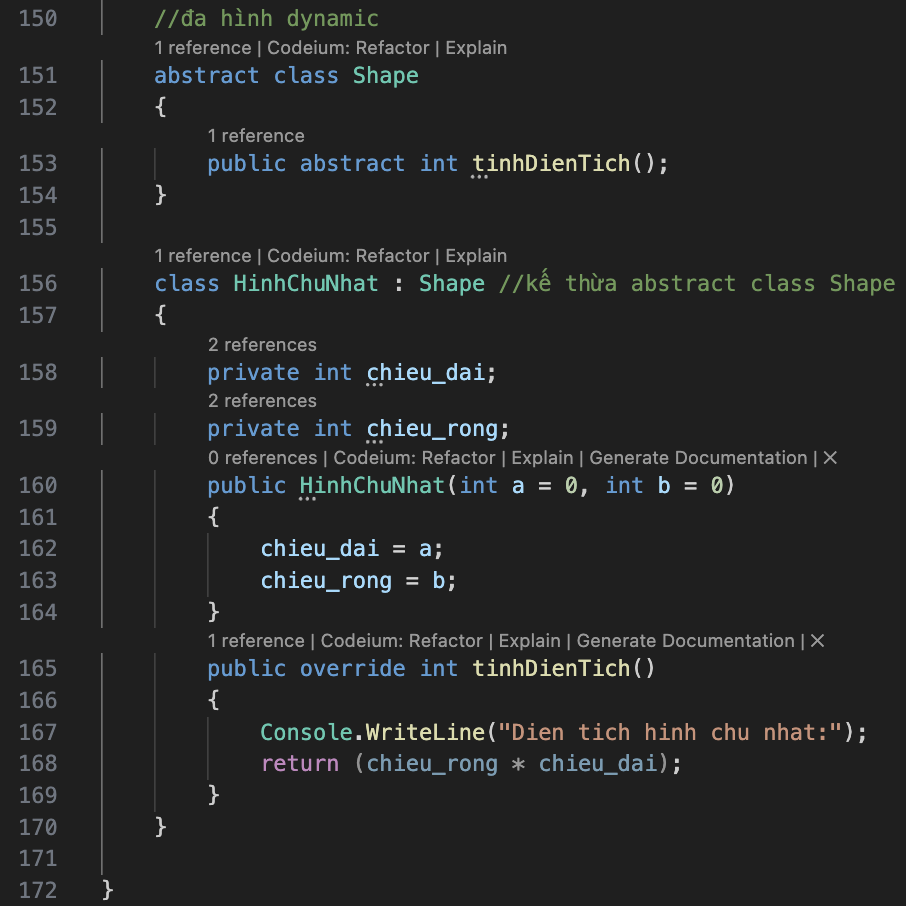




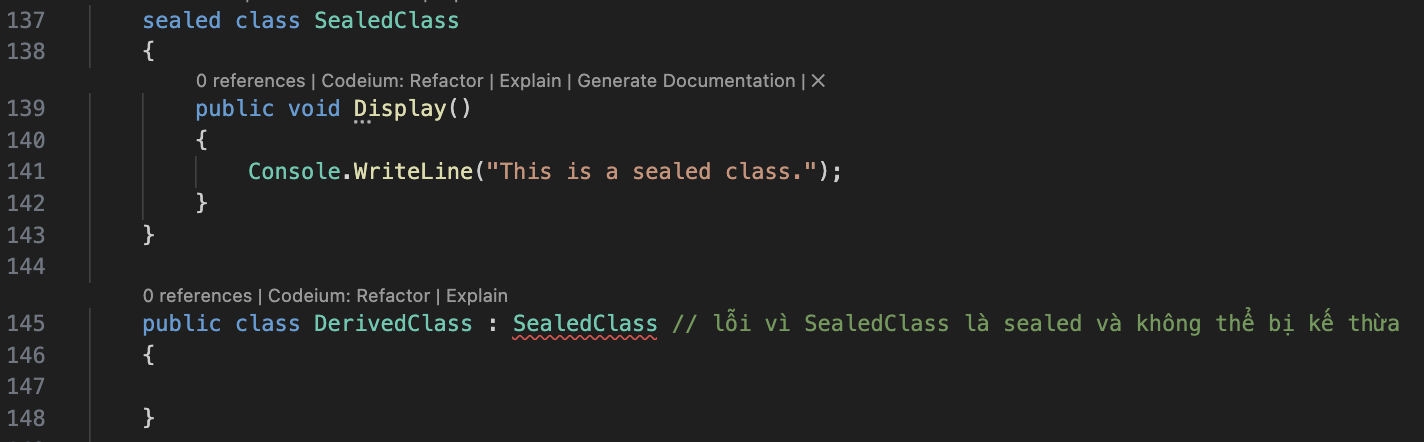
## Đa hình dynamic trong C#

C# cho phép bạn tạo các lớp abstract (trừu tượng) mà được sử dụng để cung cấp trình triển khai cục bộ lớp của một Interface. Trình triển khai (Implementation) được hoàn thành khi một lớp kế thừa kế thừa từ nó. Dưới đây là một số qui tắc về các lớp abstract trong C#:

* Bạn không thể tạo một Instance (sự thể hiện) của một lớp abstract.
* Bạn không thể khai báo một phương thức abstract ở bên ngoài một lớp abstract.
* Khi một lớp được khai báo là **sealed**, nó không thể được kế thừa, các lớp abstract không thể được khai báo là sealed.

****

từ khóa sealed được sử dụng để ngăn chặn việc một lớp bị kế thừa. Khi một lớp được khai báo là sealed, không có lớp nào có thể kế thừa từ nó.

****