Họ và Tên: Nguyễn Thanh Kiên.

MSSV: 22110092.

Bài tập:

Xây dựng lớp Customer (Khách hàng) gồm các thuộc tính CustomerID (ví dụ dạng C001), CustomerName, CustomerAddress, CustomerPhone, CustomerType. Trong đó CustomerType có các loại sau (TrungThanh, TiemNang, CanQuanTam, KhachHangKhac). Sau đó xây dựng lớp Company với các thuộc tính (ListOfCustomers (danh sách các khách hàng của công ty), CompanyName (Tên công ty))

* Xây dựng các phương thức thiết lập cho hai lớp Customer và Company 👍
* Xây dựng các phương thức get và set cho từng thộc tính 👍
* Xây dựng các hàm CustomerInfo và CompanyInfo để xuất thông tin của khách hàng và của công ty 👍
* Viết hàm SearchCustomer với tham số truyền vào có thể là Tên Khách Hàng hoặc là Index (dùng generic) 👍
* Viết một hàm mở rộng ConvertToString cho lớp Customer để có thể dùng Console.WriteLine xuất thông tin khách hàng (dùng Extensions) 👍
* Xây dựng một từ điển để định nghĩa các khái niệm loại khách hàng và xuất thông tin đó kèm theo thông tin khách hàng (dùng Dictionary) 👍
* Tạo một sự kiện để cập nhật số lượng khách hàng khi thêm hoặc xóa khách hàng đó khỏi công ty 👍

Làm thêm: Không.

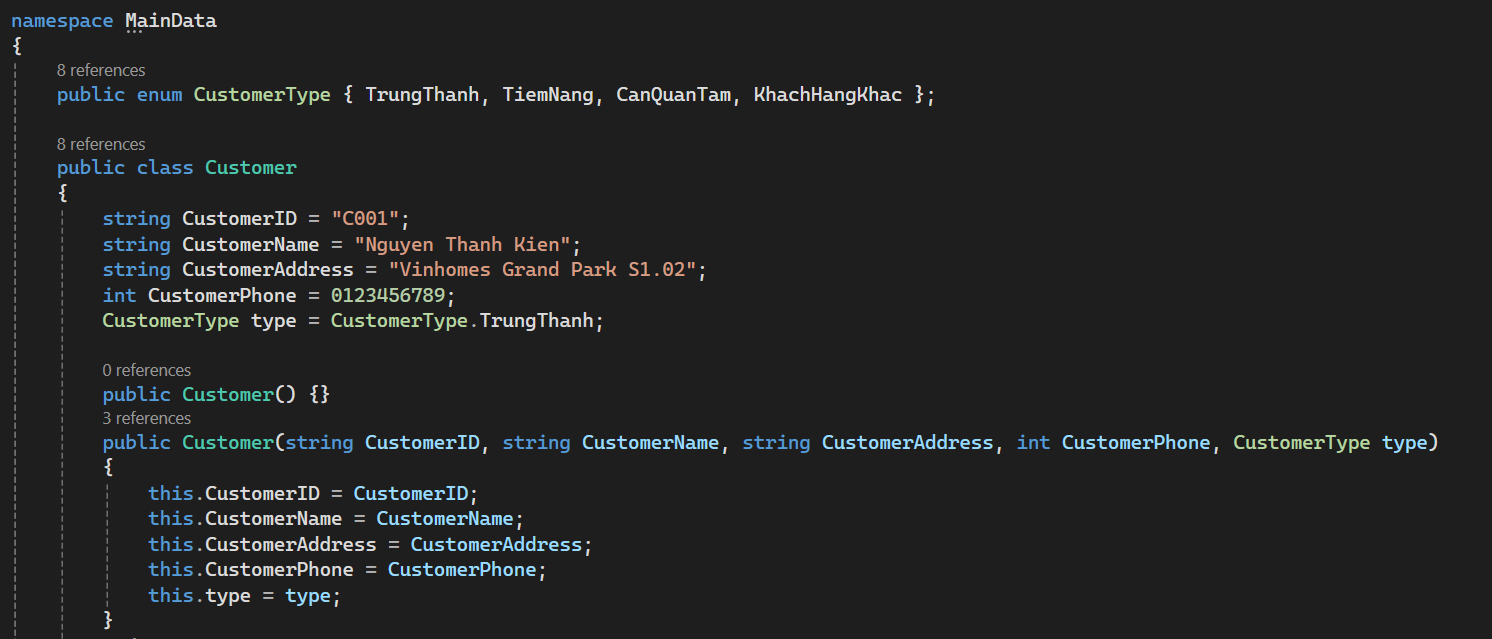
Bài làm:

1. Xây dựng các phương thức thiết lập cho hai lớp Customer và Company. Lớp Customer (Khách hàng) gồm các thuộc tính CustomerID (ví dụ dạng C001), CustomerName, CustomerAddress, CustomerPhone, CustomerType. Trong đó CustomerType có các loại sau (TrungThanh, TiemNang, CanQuanTam, KhachHangKhac). Sau đó xây dựng lớp Company với các thuộc tính (ListOfCustomers (danh sách các khách hàng của công ty), CompanyName (Tên công ty))

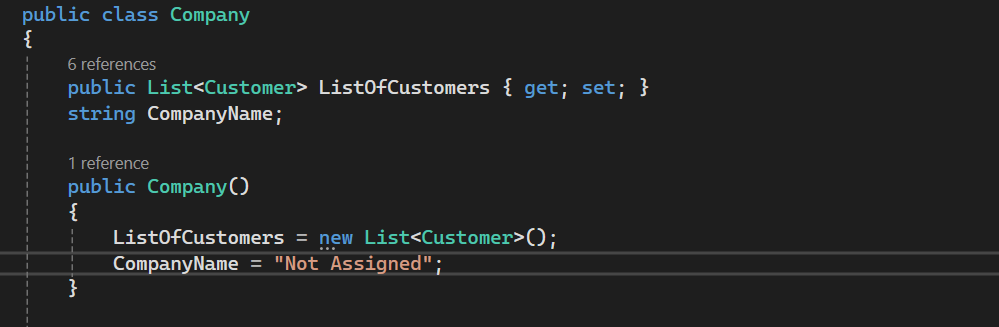
+ Đầu tiên, em khai báo các thư viện System. Sau đó, để xây dượng phương thức thiết lập cho hai lớp, em tạo một namespace MainData, trong đó có chứa hai lớp Customer và Company.

+ Ở lớp Customer có yêu cầu thuộc tính CustomerType gồm nhiều loại, vì vậy, em sử dụng kiểu dữ liệu Enum, là kiểu dữ liệu dùng để gán một cách thoải mái nhất, nhất là các yêu cầu về loại trong 1 thuộc tính. Lập trình .Net thì chúng ta dùng Enum nhiều mới chuẩn.

+ Em khai báo lớp Customer với 3 phương thức thiết lập dạng chuỗi, 1 phương thức thiết lập dạng số và 1 phương thức còn lại thiết lập dưới dạng Enum.



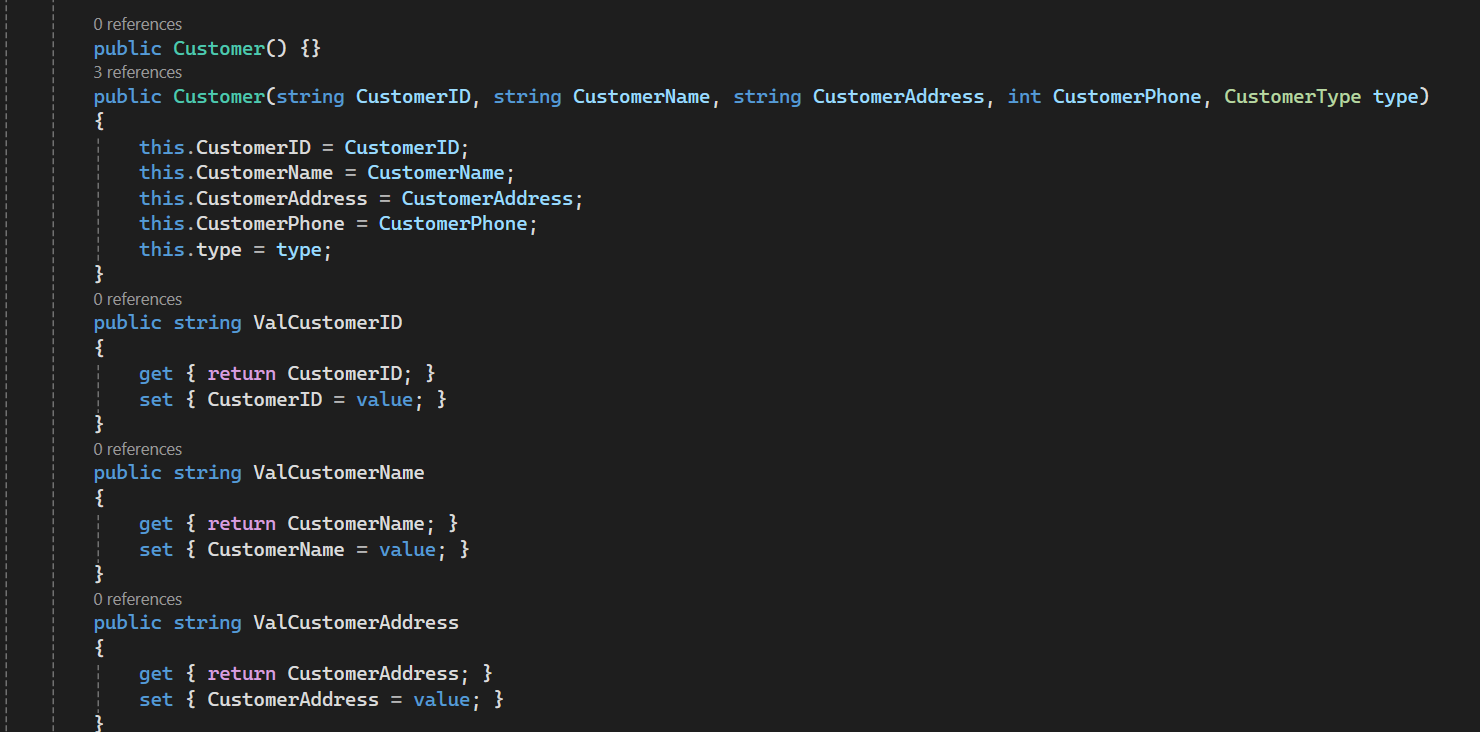
+ Ở lớp Company, em khai báo với 2 phương thức thiết lập, 1 là sử dụng kiểu dữ liệu List<> cho ListOfCustomers và phương thức thiết lập còn lại là dạng chuỗi cho tên công ty.

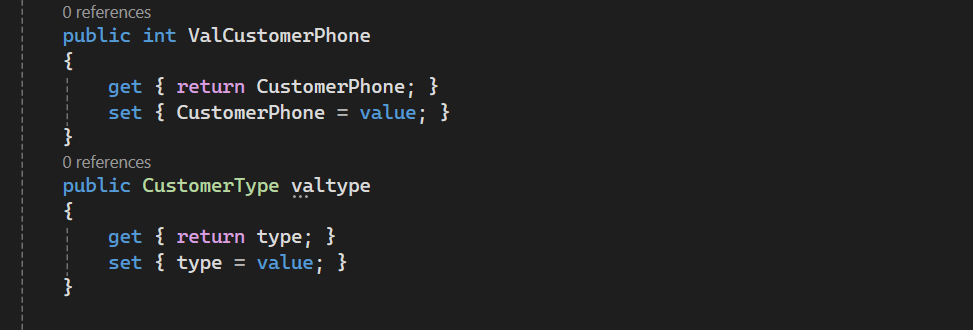


1. Xây dựng các phương thức get và set cho từng thộc tính

Em xây dựng phương thức get và set cho từng thuộc tính như sau.

+ Ở lớp Customer:





+ Ở lớp Company:

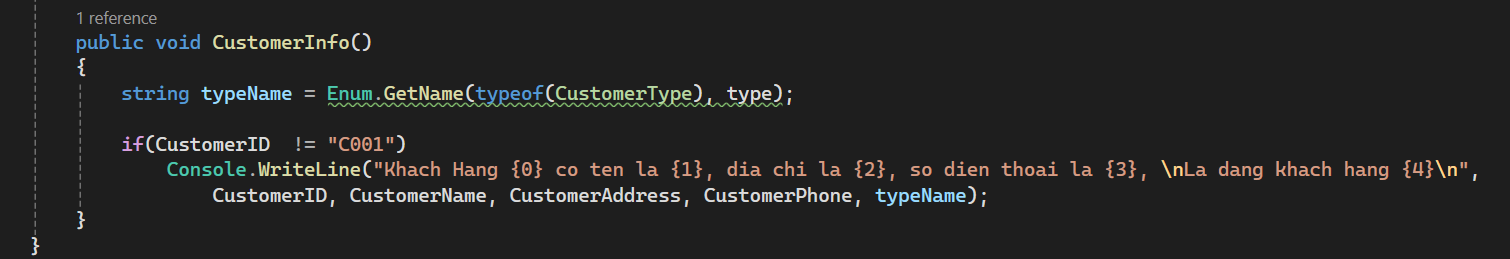


Chú ý rằng, vì ListOfCustomer đã có sẵn thuộc tính tự động, vì vậy ta sẽ không cần khai báo thêm riêng get và set cho List đó.

1. Xây dựng các hàm CustomerInfo và CompanyInfo để xuất thông tin của khách hàng và của công ty

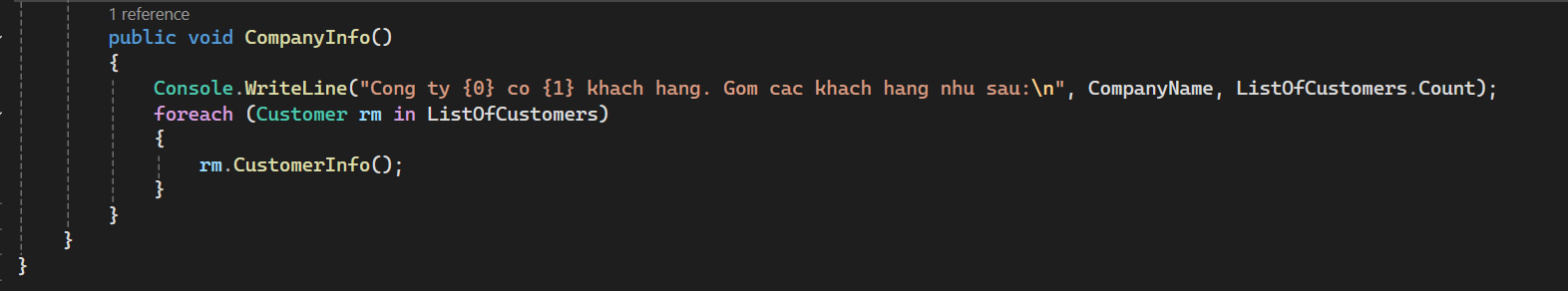
Để xuất thông tin khách hàng và công ty ra màn hình Console, em xây dựng thêm 2 hàm Info trong từng class tương ứng.

+ Ở lớp Customer, em xây dựng hàm CustomerInfo:

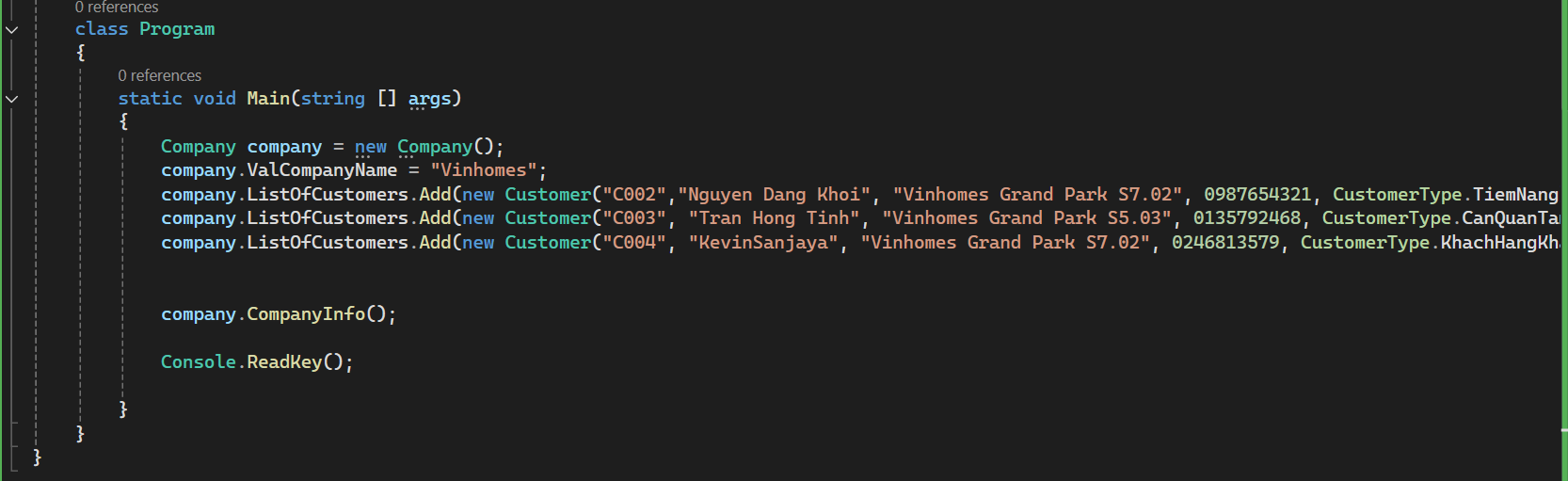


Chú ý rằng, em không xuất ra thông tin khách hàng tương ứng với CustomerID C001.

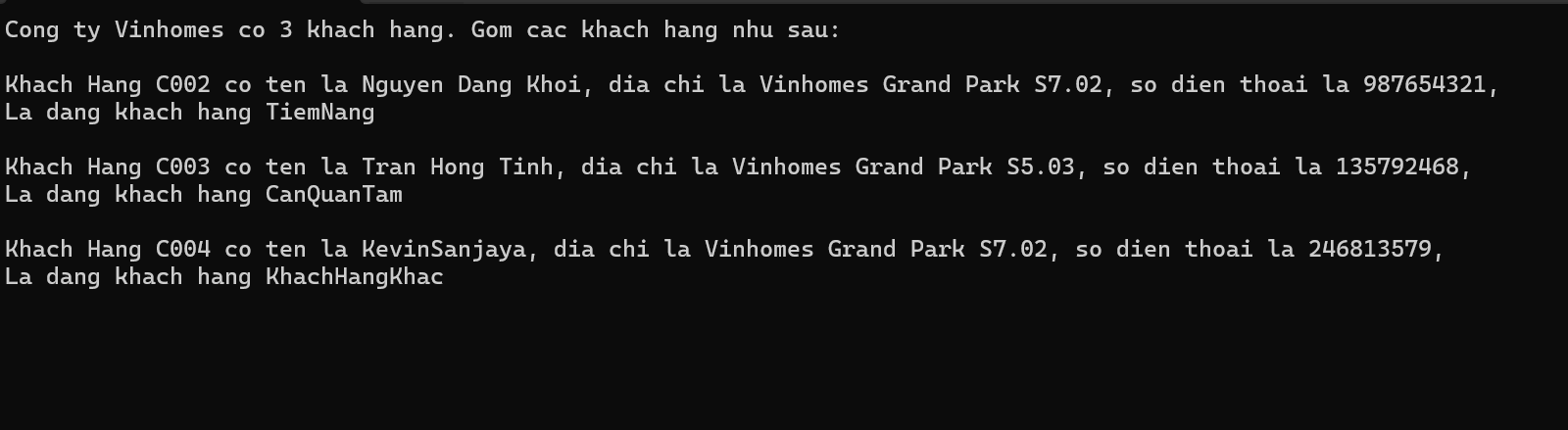
+ Ở lớp Company, em xây dựng hàm CompanyInfo:



+ Sau đó, em tạo một class Program trong namespace CompanyManagement, class này chứa hàm main dùng để chứa thông tin 3 customer mà em tạo sẵn với các thông tin tương ứng có sẵn để xuất ra màn hình.

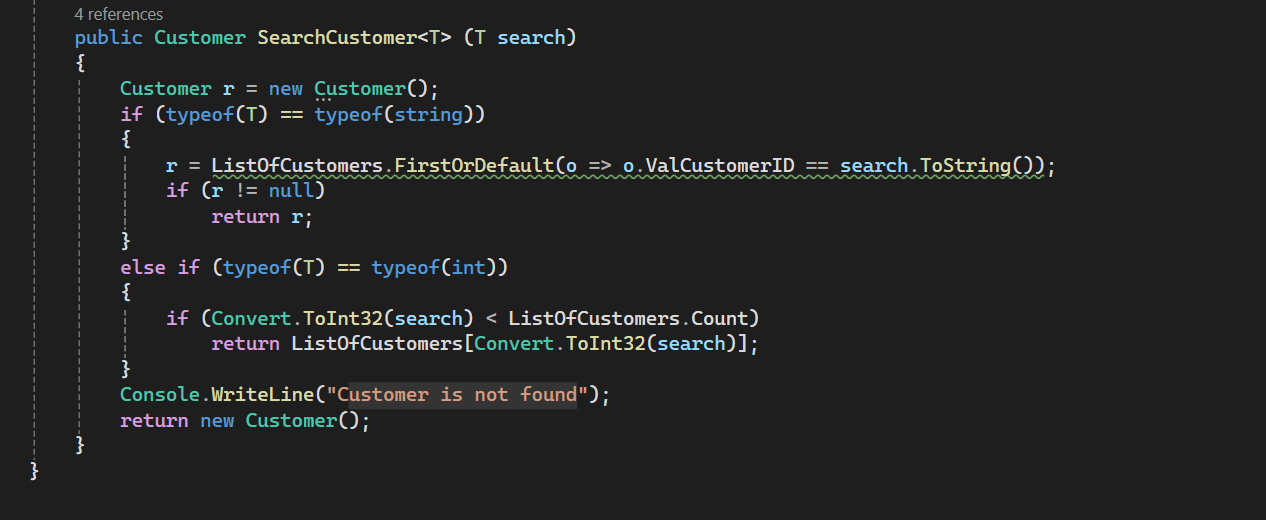


**Màn hình Console:**

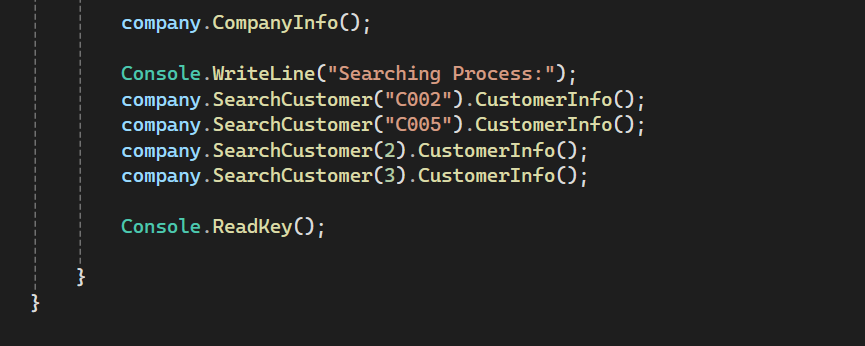


1. Viết hàm SearchCustomer với tham số truyền vào có thể là Tên Khách Hàng hoặc là Index (dùng generic).

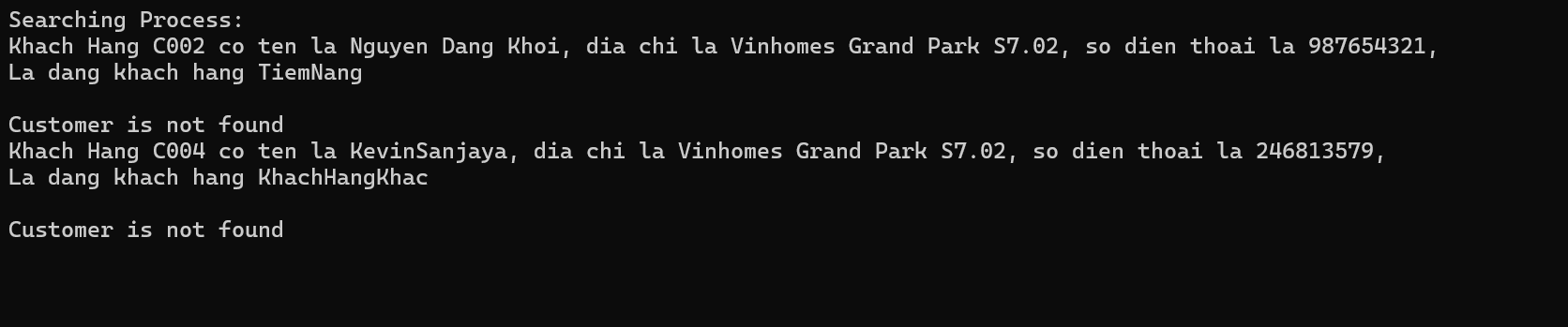
+ Trong lớp Company, em khai báo hàm SearchCustomer với sử dụng generic tham số truyền vào. Tức là em tạo kiểu dữ liệu generic, kiểu dữ liệu tổng thể. Lớp này sẽ được xác định kiểu dữ liệu gì khi mình truyền vào. Nếu T là kiểu string thì sẽ tìm kiếm dưới dạng string, tìm kiếm xong sẽ trả về đối tượng r. Hàm tìm kiếm sẽ lục tìm từng customer và kiểm tra customerID bằng đúng ID mình search và trả về cho chúng ta. Nếu T là kiểu số, em sẽ convert nó dưới dạng 32 bit và so sánh với số lượng customer. Còn trong trường hợp tìm kiếm không chữ hay không số thì trả về Customer is not found.



+ Em xây dựng hàm Main với các test case search như sau:



**Màn hình Console:**

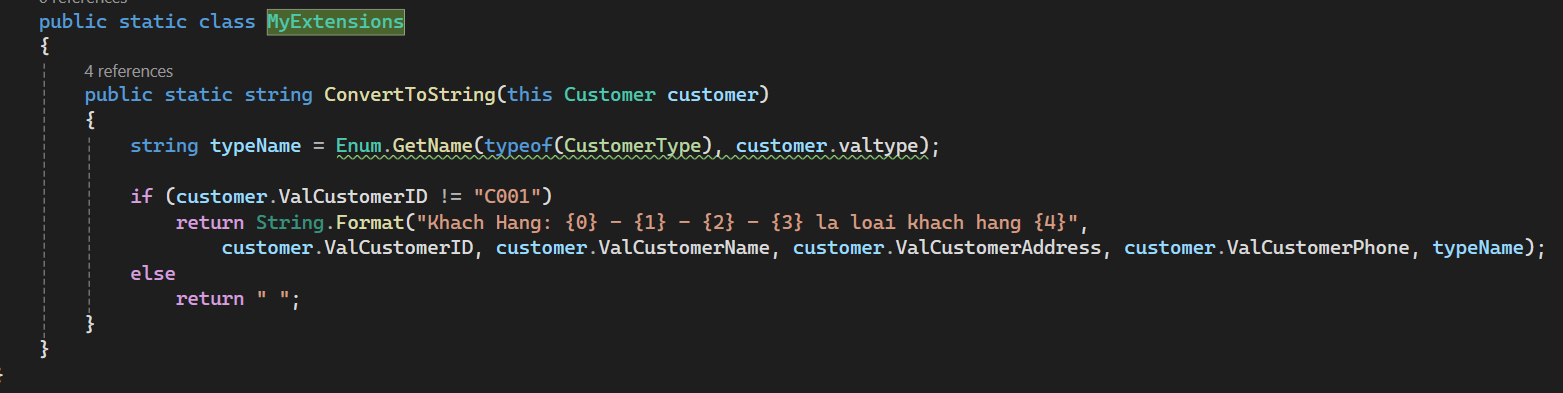


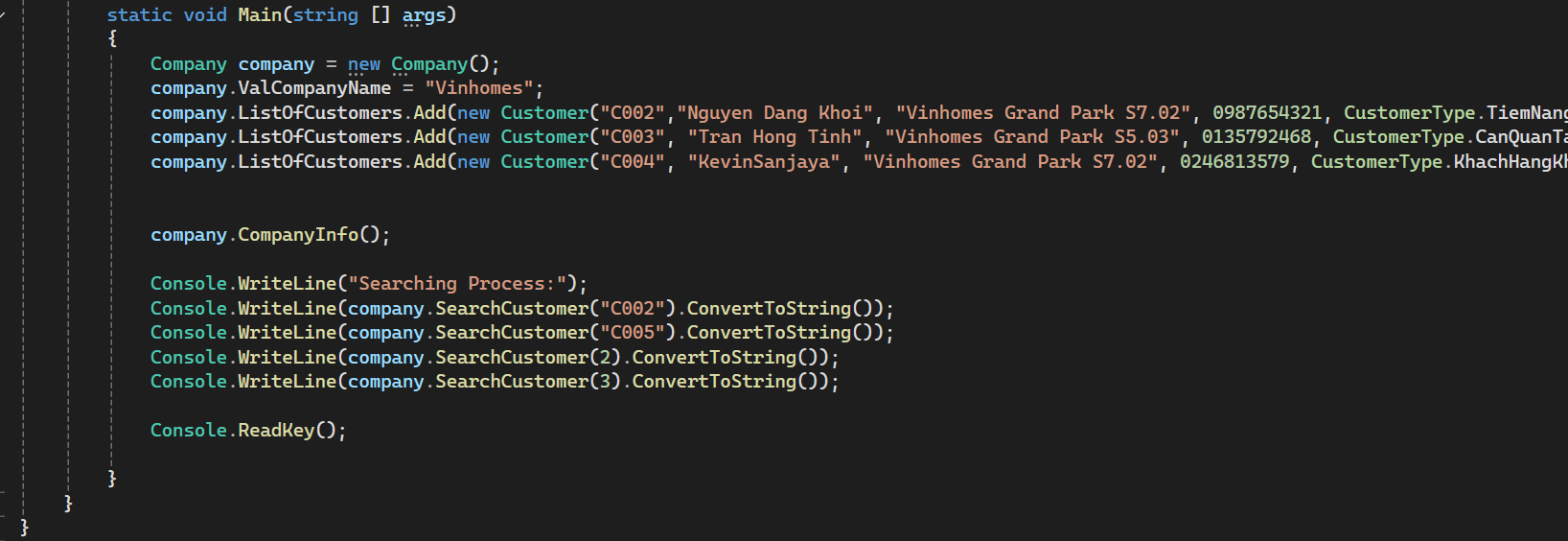
Ở test case “C005” không cho ra kết quả vì C005 không tồn tại.

Ở test case dạng số, khi gọi SearchCustomer(3), em đang tìm phòng với chỉ số 3. Tuy nhiên, **chỉ số** ở đây bắt đầu từ 0, không phải từ 1. Danh sách Customers của em hiện có 3 người: C002 - > index = 0, C003 -> index = 1, C004 -> index =2. Vì vậy, với index = 3, sẽ không tìm thấy customer tương ứng vì vượt quá customer.count với chỉ số tối đa là 2.

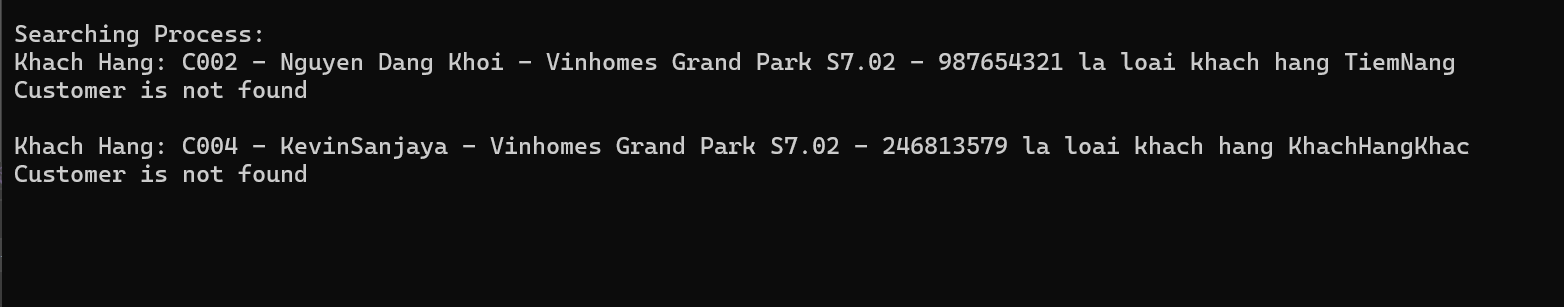
1. Viết một hàm mở rộng ConvertToString cho lớp Customer để có thể dùng Console.WriteLine xuất thông tin khách hàng (dùng Extensions)

+ Trong namespace MainData, em tạo lớp MyExtensions để tạo thêm một hàm mở rộng hàm ConvertToString cho lớp Customer để xuất thông tin khách hàng. Mở rộng phương thức là mở rộng phương thức có sẵn, lớp MyExtensions này sẽ giúp em viết lạm hàm ConverToString dưới kiểu của mình. Mở rộng sẽ giúp em thay thế hoặc là tạo thêm, vì em đã mở rộng cho Customers rồi nên sẽ thực hiện phương thức này sau khi mở rộng



+ Trong hàm Main, em vẫn giữ các test case tương ứng. 

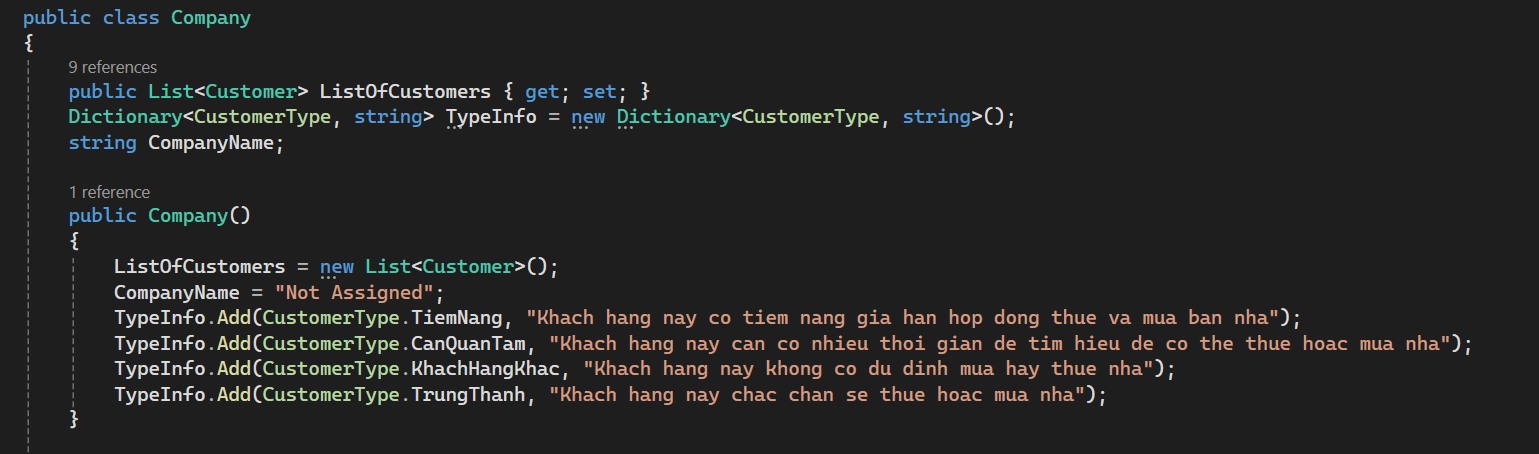
Màn hình Console:



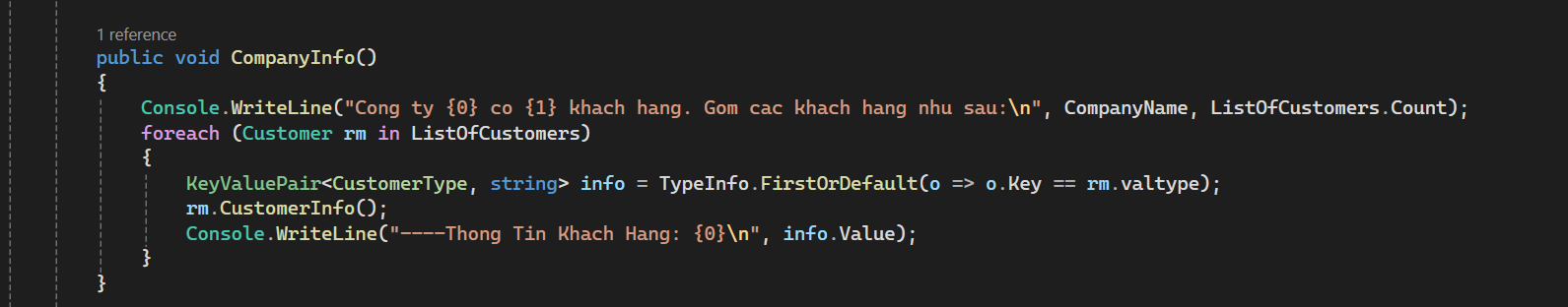
Xuất hiện Customer is not found em đã giải thích ở trên tương tự với test case này.

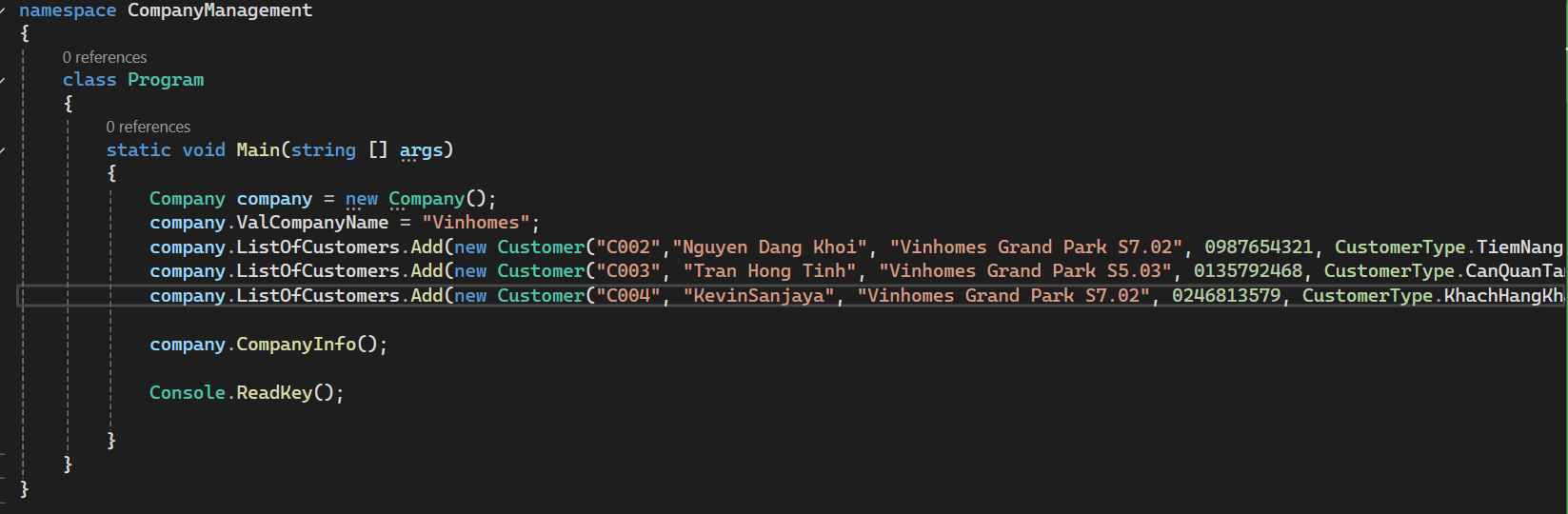
1. Xây dựng một từ điển để định nghĩa các khái niệm loại khách hàng và xuất thông tin đó kèm theo thông tin khách hàng (dùng Dictionary)

+ Trong lớp Company, em sử dụng Dictionary là từ điển gồm chìa khóa và giá trị, để ánh xạ giữa thông tin khách hàng với thông tin chi tiết thêm. Hơn nữa, em dùng Lambda để tìm kiếm thông tin dựa trên khách hàng trong hàm truy xuất thông tin công ty. Khi mà tìm kiến, thì o (biểu thức lambda) sẽ lần lượt là đối tượng của TypeInfo và nó sẽ kiểm tra khi bằng cái em muốn hay không, nếu bằng thì sẽ trả về o.

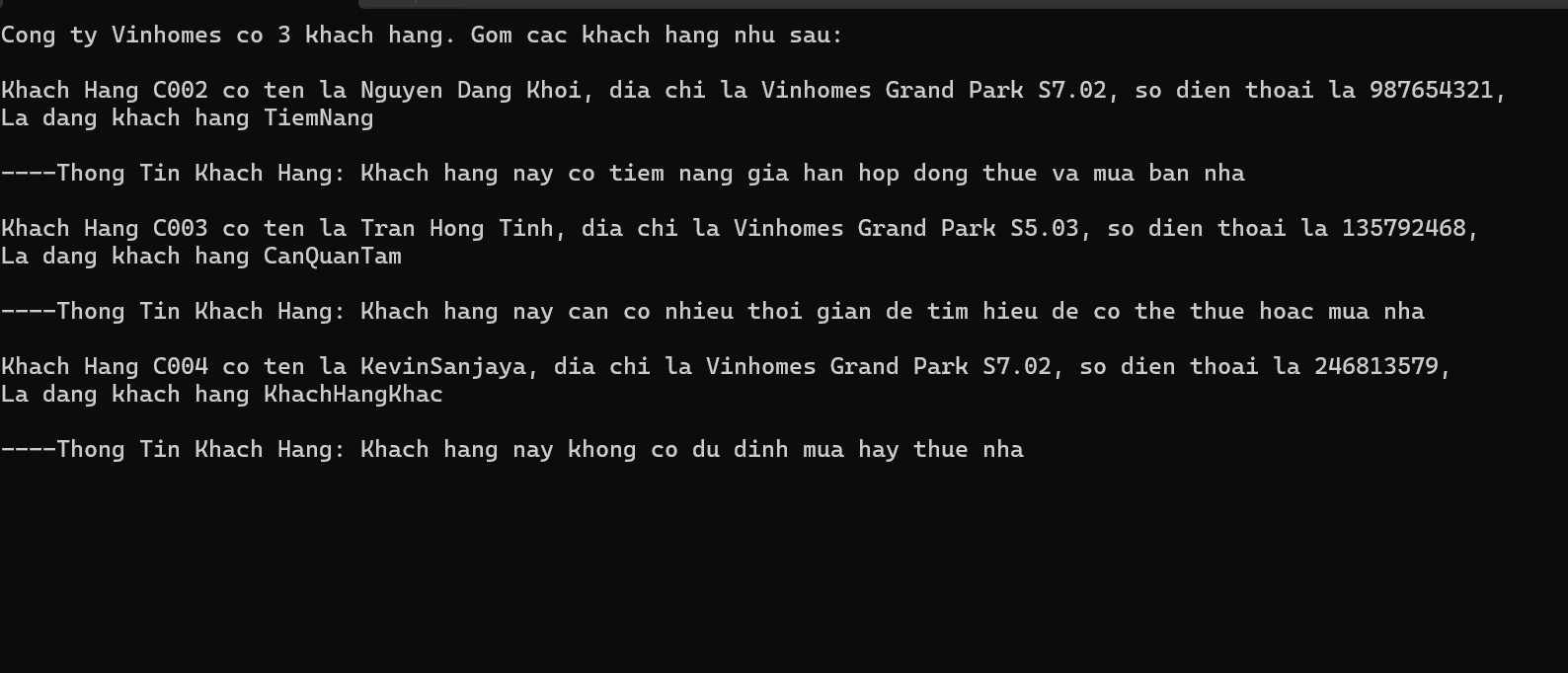


Em tiếp tục thêm KeyValuePair vào CompanyInfo

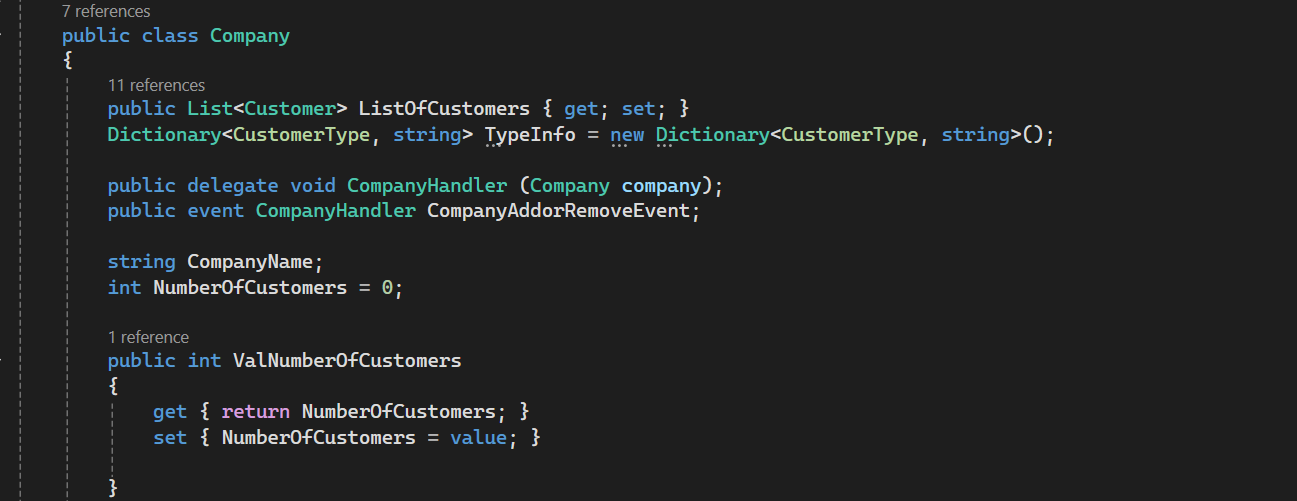


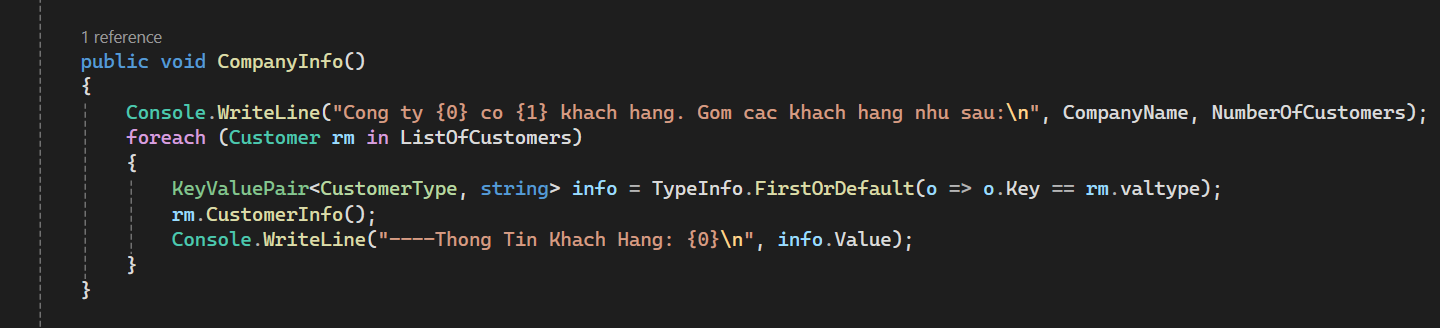
+ Em thay đổi, xóa searching ở hàm main: 

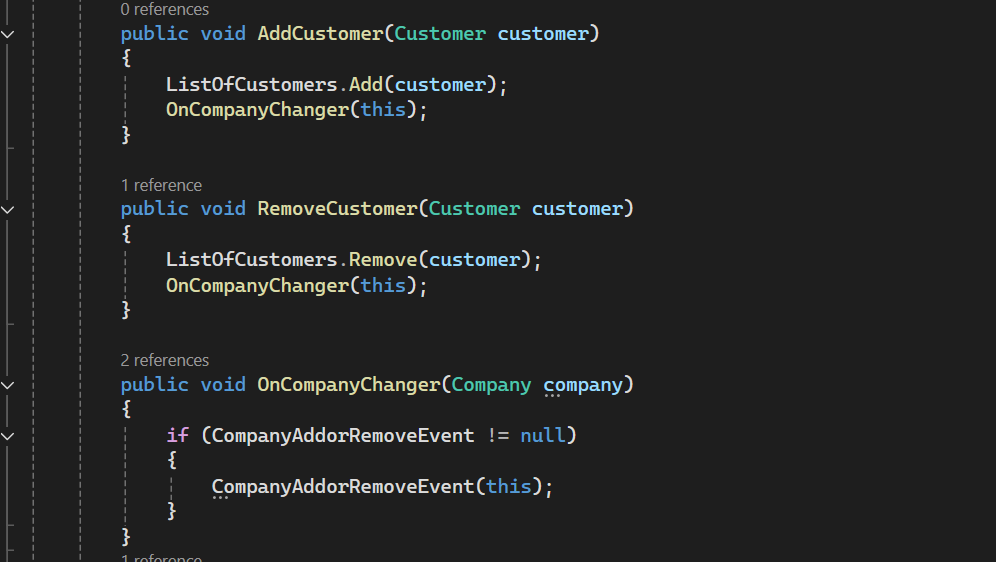
**Màn hình Console:**



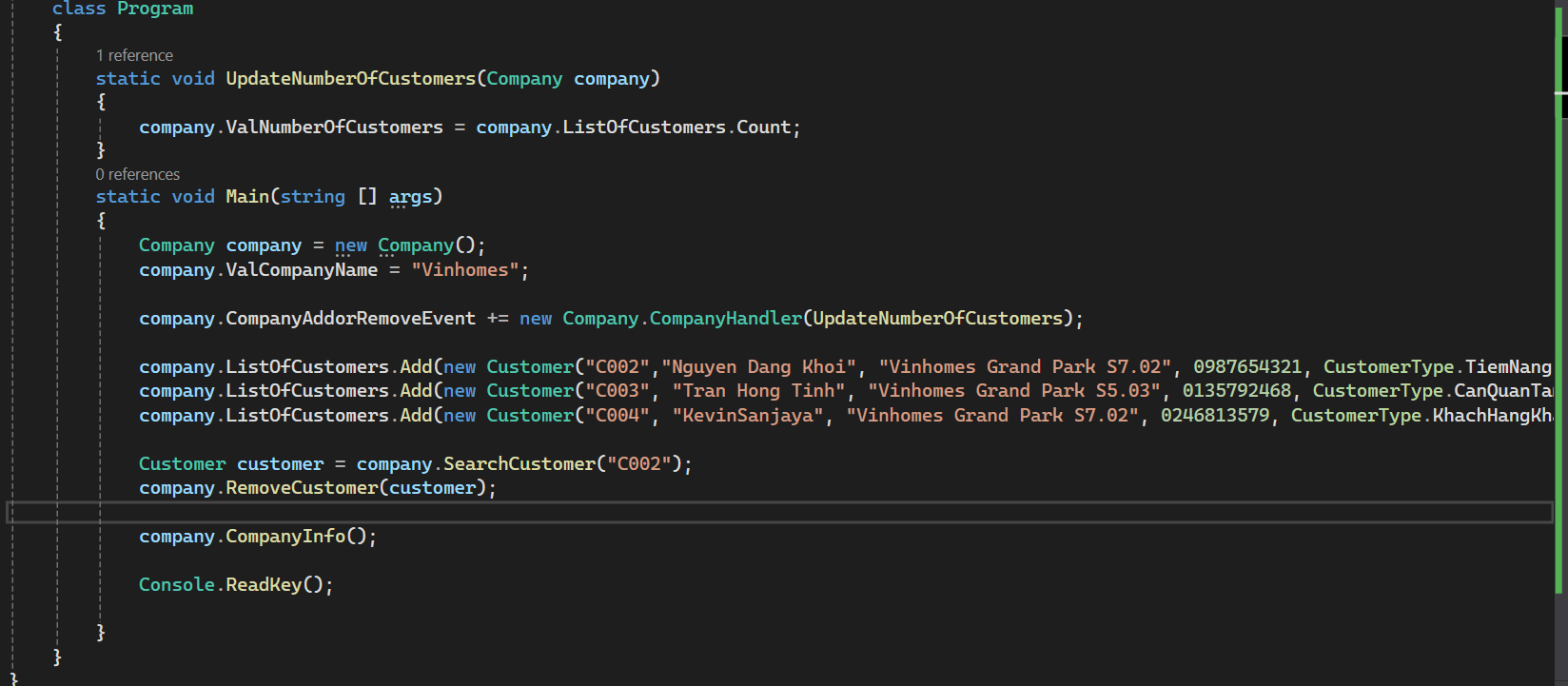
1. Tạo một sự kiện để cập nhật số lượng khách hàng khi thêm hoặc xóa khách hàng đó khỏi công ty.

+ Trong lớp hotel, em khai báo delegate CompanyHandler và Event CompanyAddorRemoveEvent. Với CompanyHandler là con trỏ hàm đến những hàm có tham số truyền vào là Company cần cập nhật. Em tiếp tục khai báo thuộc tính NumberOfCustomer với hai phương thức get set và cho giá trị ban đầu = 0, và cuối cùng em thêm 2 thao tác xóa và thêm customer ra khỏi company. Em cập nhật lại hàm companyinfo để xuất thông tin. 

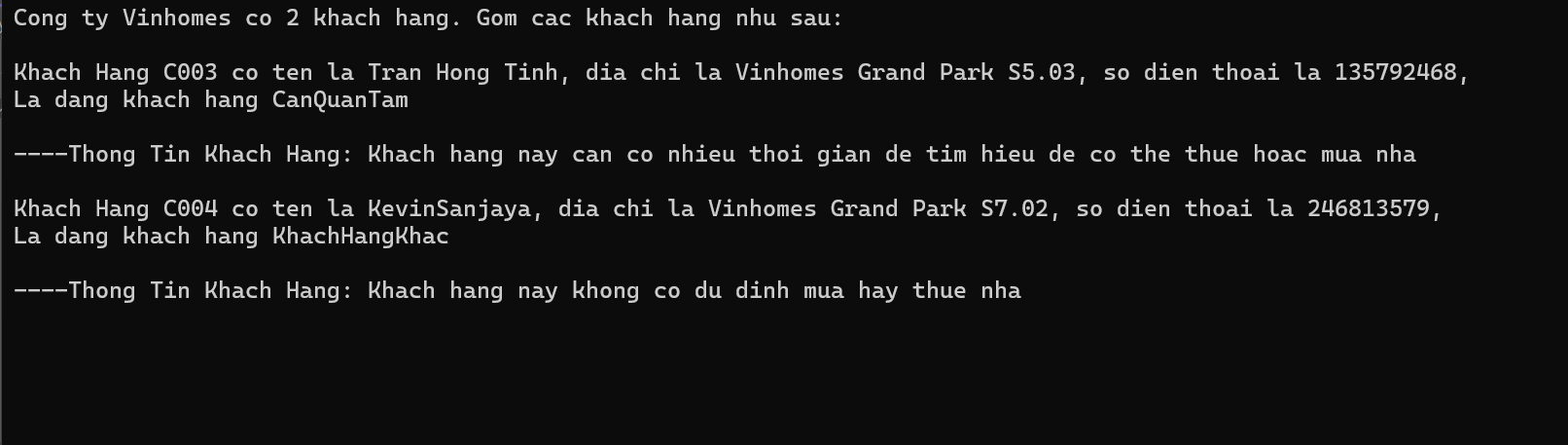




+ Em cập nhật lại hàm Main để xóa thông tin ID C002 và thêm hàm UpdateNumberOfCustomers



**Màn hình Console:**

****

Event là liên quan đến sự kiện. Một khi event diễn ra, CompanyHandler sẽ liên kết với 1 hàm nào đó có tham số truyền vào là company và trả về là void. Khi thêm customer vô, sự kiện OnCompanyChanger phải thực thi, nó sẽ gọi sự kiện mình đăng ký là CompanyAddorRemoveEvent, sự kiện thực thi sẽ yêu cầu con trỏ hàm CompanyHandler thực hiện 1 hàm nào đó. CompanyHandler sẽ thực hiện hàm UpdateNumberOfCustomers .