

Bài 2.10: Tìm hiểu string

- ✓ Khái niệm và đặc điểm
- ✓ Khởi tạo đối tượng string
- ✓ Một số hằng số và phương thức
- ✓ Ví dụ minh họa

Khái niệm và đặc điểm

- ✓ Một string là một tập các kí tự sử dụng để biểu diễn một nội dung văn bản nào đó.
- ✓ Về bản chất, cấu thành của một đối tượng string là các đối tượng kí tự đơn.
- ✓ Giá trị của string là nội dung chứa trong string đó.
- ✓ String biểu diễn các kí tự 16 bit Unicode.
- ✓ Bản thân string là bất biến, tức các giá trị trong string sau khi tạo ra là không thể thay đổi.
- ✓ Kích thước tối đa của string có thể lên tới 2GB, khoảng 1 tỉ kí tự.

Khởi tạo đối tượng string

Ta có thể khởi tạo đối tượng string bằng nhiều cách:

- ✓ Gán chuỗi kí tự cho biến string.
- ✓ Gọi hàm khởi tạo lớp String.
- ✓ Sử dụng phép nối chuỗi để tạo chuỗi mới.
- ✓ Truy cập một thuộc tính hoặc gọi tới phương thức trả về một string.
- ✓ Gọi tới phương thức định dạng để chuyển giá trị của đối tượng sang string.

```
string message = "Hello World!"; // cách phổ biến
string fullName = new string("Tran Van Nam".ToCharArray()); // gọi hàm khởi tạo
string address = "Cau Giay" + "-" + "Ha Noi"; // sử dụng phép nối chuỗi
Console.WriteLine(message);
Console.WriteLine(fullName);
Console.WriteLine(address);
```

Một số thông tin về string

- ✓ Để lấy độ dài tính bằng số kí tự có trong string ta gọi `obj.Length`.
- ✓ Để lấy string rỗng `""` ta có thể gọi `string.Empty`

Một số phương thức thường dùng

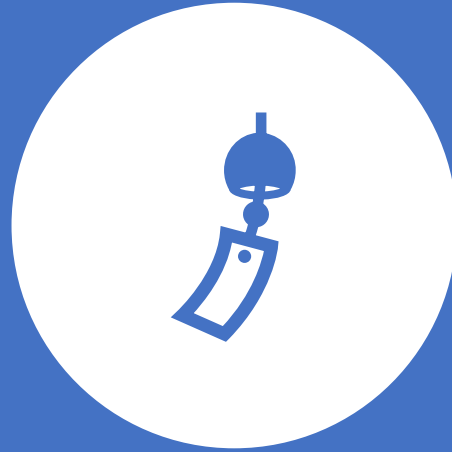
Phương thức	Mô tả
Clone()	Trả về tham chiếu tới đối tượng hiện thời của string.
Compare(str1, str2)	So sánh và trả về tương quan giữa 2 string str1, str2. Kết quả là số nguyên âm nếu str1 < str2, 0 nếu str1 == str2 và dương nếu str1 đứng sau str2 theo thứ tự sắp xếp a-z.
Compare(str1, str2, ignoreCase)	So sánh 2 chuỗi. Nếu ignoreCase là true thì hai chuỗi str1, str2 so sánh không phân biệt chữ hoa, thường. Trả về tương tự phương thức Compare với 2 tham số ở trên.
CompareTo(other)	So sánh string hiện tại với other và trả về kết quả tương tự phương thức Compare 2 tham số ở trên.
Concat(params)	Nối các tham số lại thành 1 string. Có thể nối 2 hay nhiều string, các phần tử mảng, tập hợp, các object...
Contains(key)	Trả về true nếu tồn tại kí tự/xâu kí tự key trong string hiện tại.
Contains(key, compare)	Trả về kết luận key có tồn tại trong string hiện tại không theo xác định của compare.
Copy(str)	Sao chép và trả về string mới với giá trị tương tự của str.
EndWith(key)	Xác định xem liệu string hiện tại có kết thúc với kí tự hoặc chuỗi kí tự trong key hay không.
Equals(str1, str2) Equal(object) Equals(str)	Xác định xem hai string str1, str2 có tương đương hay không. Xác định xem string hiện tại và object/str có tương đương nhau về giá trị không.
GetHashCode()	Trả về mã băm của string hiện tại.

Một số phương thức thường dùng

IndexOf(key) IndexOf(key, startPos)	Trả về vị trí xuất hiện đầu tiên của key trong string hiện tại bắt đầu từ vị trí startPos. Nếu không tồn tại key, trả về -1.
IndexOfAny(char[] anyOf) IndexOfAny(char[] anyOf, startPos)	Trả về vị trí xuất hiện đầu tiên của bất kì kí tự nào trong mảng kí tự anyOf trong string hiện thời. Bắt đầu tìm kiếm từ vị trí startPos. Nếu không tìm thấy, trả về -1.
Insert(startPos, value)	Chèn value vào vị trí startPos của string hiện tại và trả về string mới sau khi chèn.
IsNullOrEmpty(value)	Chỉ ra liệu value là null hay string rỗng "".
IsNullOrWhiteSpace(value)	Xác định xem value là null hay chỉ chứa kí tự khoảng trắng.
LastIndexOf(key) LastIndexOf(key, startPos)	Trả về vị trí xuất hiện cuối cùng của key trong string hiện thời. Bắt đầu từ vị trí startPos nếu cung cấp. Không tồn tại thì trả về -1.
LastIndexOfAny(char[] anyOf) LastIndexOfAny(char[] anyOf, startPos)	Trả về vị trí xuất hiện cuối cùng của bất kì kí tự nào trong mảng kí tự anyOf trong string hiện thời. Bắt đầu tìm kiếm từ vị trí startPos. Nếu không tìm thấy, trả về -1.
PadLeft(totalWidth)	Trả về string căn lề phải bằng cách thêm các kí tự khoảng trắng vào phía bên trái để đủ totalWidth kí tự.
PadRight(totalWidth)	Trả về string căn lề trái bằng cách thêm các kí tự khoảng trắng vào phía bên phải của nó cho đủ totalWidth kí tự.
Remove(startPos) Remove(startPos, count)	Trả về string mới bằng cách xóa bỏ các kí tự còn lại hoặc xóa count kí tự từ vị trí startPos.
Replace(old, new)	Trả về một string sau khi thay thế tất cả các kí tự hoặc string old bởi kí tự hoặc string trong tham số new.
Split(separator)	Tách chuỗi kí tự hiện thời tại các vị trí được chỉ định trong separator và trả về mảng các string sau khi tách.

Một số phương thức thường dùng

StartWith(key)	Xác định xem string hiện tại có bắt đầu bởi giá trị cho trong key không.
Substring(startPos) Substring(startPos, size)	Trả về string trích ra từ string hiện tại bắt đầu từ vị trí startPos tới hết string hoặc tới khi đủ size kí tự.
ToCharArray(startPos, length) ToCharArray()	Chuyển length kí tự từ vị trí startPos hoặc tất cả kí tự từ string hiện thời thành mảng char.
ToLower()	Trả về một string sau khi đã viết thường tất cả các kí tự trong string hiện tại.
ToUpper()	Trả về một string sao chép từ string hiện tại sau khi viết hoa tất cả kí tự trong đó.
Trim(char[] chars) Trim(key) Trim()	Xóa bỏ các kí tự cho trước trong key hoặc chars ở đầu và cuối string và trả về string sau khi xóa. Nếu không cung cấp tham số thì hàm xóa bỏ các kí tự khoảng trắng.
TrimEnd(char[] chars) TrimEnd(key) TrimEnd()	Xóa bỏ các kí tự cho trước trong key hoặc chars ở cuối string và trả về string sau khi xóa. Nếu không cung cấp tham số thì hàm xóa bỏ các kí tự khoảng trắng.
TrimStart(char[] chars) TrimStart(key) TrimStart()	Xóa bỏ các kí tự cho trước trong key hoặc chars ở đầu string và trả về string sau khi xóa. Nếu không cung cấp tham số thì hàm xóa bỏ các kí tự khoảng trắng.



Nội dung tiếp theo

Các cấu trúc điều khiển