

Bài 8.6: Cấu trúc TimeOnly

- ✓ Mục đích và các thuộc tính
- ✓ Các phương thức thường dùng
- ✓ Ví dụ minh họa

Mục đích sử dụng và các thuộc tính

- ✓ Cấu trúc readonly TimeOnly dùng để mô tả thông tin về thời gian trong ngày gồm giờ, phút, giây, micro giây.
- ✓ Giá trị hợp lệ trong khoảng 00:00:00 đến 23:59:59.999999 dương lịch sau công nguyên.
- ✓ DateOnly và TimeOnly được hỗ trợ từ phiên bản .NET 6.
- ✓ Các thuộc tính:
 - ✓ **Hour**: lấy ra giá trị giờ của đối tượng hiện thời.
 - ✓ **Minute**: lấy số phút của đối tượng hiện tại.
 - ✓ **Second**: lấy số giây của đối tượng hiện thời.
 - ✓ **Milisecond**: lấy số mili giây trong đối tượng hiện thời.
 - ✓ **MaxValue**, **MinValue**: lấy thời gian tối đa/tối thiểu có thể tạo từ struct TimeOnly.
 - ✓ **Ticks**: lấy số tic tắc từ đối tượng hiện thời.

Các constructor

- ✓ `TimeOnly(long ticks)`: tạo đối tượng `TimeOnly` với số tick cho trước.
- ✓ `TimeOnly(int hour, int minute)`: tạo đối tượng của `TimeOnly` với giờ phút cho trước.
- ✓ `TimeOnly(int hour, int minute, int second)`: tạo đối tượng của `TimeOnly` với giờ, phút, giây cho trước.
- ✓ `TimeOnly(int hour, int minute, int second, int milisec)`: tạo đối tượng của `TimeOnly` với giờ, phút, giây, mili giây cho trước.

Các phương thức khác

Phương thức và mô tả
<p>AddHours(double value); thêm value giờ vào số giờ trong đối tượng hiện thời. Trả về đối tượng TimeOnly mới chứa thông tin thời gian của đối tượng hiện tại và số giờ thêm vào.</p> <p>AddHours(double value, out int wrappedDays); trả về đối tượng TimeOnly mới có dữ liệu thời gian của đối tượng hiện thời cộng thêm số giờ được cung cấp trong tham số. Nếu số giờ thêm vào tạo ra số ngày vượt quá thì số ngày này sẽ được cung cấp trong tham số thứ hai.</p>
<p>Add(TimeSpan value); trả về một đối tượng TimeOnly trong đó đã thêm giá trị của khoảng thời gian vào giá trị của đối tượng hiện thời.</p> <p>Add(TimeSpan value, out int wrappedDays); trả về đối tượng TimeOnly đã thêm giá trị của khoảng thời gian vào giá trị của đối tượng hiện thời. Nếu quá trình thêm vào tạo ra số ngày vượt quá trong tham số thứ hai.</p>
<p>AddMinutes(double value); thêm value phút vào đối tượng hiện thời. Trả về một đối tượng TimeOnly mới sau khi bổ sung value phút vào nó.</p>
<p>CompareTo(TimeOnly other); CompareTo(Object other); So sánh giá trị của other với đối tượng hiện thời và trả về số nguyên thể hiện sự khác biệt về thời gian(sớm hơn, muộn hơn hay bằng với) so với tham số nhận được.</p>
<p>Equals(TimeOnly other); Equals(Object other); Trả về true nếu đối tượng hiện thời tương đương với other. Trả về false nếu ngược lại.</p>
<p>FromDateTime(DateTime dateTime); trả về đối tượng TimeOnly từ phần thời gian giờ phút giây của đối tượng dateTime.</p>

Các phương thức khác

FromTimeSpan(TimeSpan timeSpan); trả về đối tượng của TimeOnly từ khoảng thời gian cung cấp trong tham số, tính từ nửa đêm.
GetHashCode(); trả về mã băm của đối tượng hiện thời.
IsBetween(TimeOnly start, TimeOnly end); kiểm tra xem đối tượng hiện thời có nằm trong khoảng thời gian cho trước không. Nên kiểm tra các giá trị start, end có tương đương nhau không trước khi gọi phương thức này.
Parse(string str); trả về đối tượng TimeOnly tương ứng string đã cung cấp. Ngoại lệ nếu str là null. Parse(string s, IFormatProvider? provider, DateTimeStyles style); chuyển đổi string s sang TimeOnly sử dụng provider và style cho trước. Ngoại lệ nếu s null và s không chứa kí tự hợp lệ thể hiện ngày tháng năm.
ToShortTimeString(); trả về string giờ phút giây rút gọn của đối tượng hiện thời.
ToLongTimeString(); trả về string đầy đủ về giờ phút giây đại diện cho đối tượng hiện thời.
ToString(); chuyển đổi đối tượng hiện thời thành string đại diện tương ứng ở dạng ngắn gọn. ToString(IFormatProvider? provider); tương tự trên nhưng sử dụng provider. ToString(string? format); chuyển đổi đối tượng TimeOnly hiện tại sang string đại diện tương ứng sử dụng định dạng được quy ước trong tham số.
ToTimeSpan(); chuyển đổi đối tượng TimeOnly sang đối tượng TimeSpan.
TryFormat(Span<char> destination, out int charsWritten, ReadOnlySpan<char> format=default, IFormatProvider? provider=default); thử định dạng giá trị của đối tượng TimeOnly hiện tại sang khoảng thời gian của các giá trị kí tự.
TryParse(string? s, out TimeOnly result); chuyển string s sang đối tượng TimeOnly result tương ứng và trả về một giá trị bool chỉ ra rằng liệu việc chuyển đổi có thành công hay không.

Các toán tử so sánh

- ✓ Phép `==` so sánh tự tương đương giữa hai đối tượng.
- ✓ Phép `>` xác định xem đối tượng thứ nhất có muộn hơn đối tượng thứ hai không.
- ✓ Phép `>=` xác định xem liệu một đối tượng `TimeOnly` có tương đương hay muộn hơn đối tượng `TimeOnly` ở toán hạng bên phải không.
- ✓ Phép `<` xác định xem đối tượng ở toán hạng trái có sớm hơn đối tượng ở toán hạng bên phải không.
- ✓ Phép `<=` xác định xem đối tượng ở toán hạng trái có sớm hơn hoặc tương đương đối tượng ở toán hạng bên phải không.
- ✓ Phép `!=` xác định sự không tương đương của hai đối tượng `TimeOnly`.

Ví dụ

```
// tạo đối tượng TimeOnly và hiển thị chuỗi kí tự đại diện tương ứng
Console.OutputEncoding = Encoding.UTF8;
TimeOnly time = new TimeOnly(15, 34, 56, 999);
Console.WriteLine("==> Thông tin của đối tượng vừa tạo: ");
Console.WriteLine($"{time.Hour}:{time.Minute}:{time.Second}");
Console.WriteLine($"Giờ: {time.Hour}");
Console.WriteLine($"Phút: {time.Minute}");
Console.WriteLine($"Giây: {time.Second}");
Console.WriteLine($"Mili giây: {time.Millisecond}");

// hiển thị string đại diện cho thời gian
Console.WriteLine($"Thông tin thời gian dạng ngắn: {time.ToShortTimeString()}");
Console.WriteLine($"Thông tin thời gian dạng dài: {time.ToLongTimeString()}");

// tạo đối tượng thời gian mới từ đối tượng thời gian có sẵn
var otherTime = time.AddMinutes(150);
Console.WriteLine("==> Thông tin thời gian sau khi thêm 150 phút: ");
Console.WriteLine($"{otherTime.Hour}:{otherTime.Minute}:{otherTime.Second}");
Console.WriteLine($"Giờ: {otherTime.Hour}");
Console.WriteLine($"Phút: {otherTime.Minute}");
Console.WriteLine($"Giây: {otherTime.Second}");
Console.WriteLine($"Mili giây: {otherTime.Millisecond}");

// so sánh thời gian
Console.WriteLine($"{time} == {otherTime} ? {time == otherTime}");
Console.WriteLine($"{time} != {otherTime} ? {time != otherTime}");
Console.WriteLine($"{time} > {otherTime} ? {time > otherTime}");
```

Ví dụ

```
==> Thông tin của đối tượng vừa tạo:  
15:34:56  
Giờ: 15  
Phút: 34  
Giây: 56  
Mili giây: 999  
Thông tin thời gian dạng ngắn: 15:34  
Thông tin thời gian dạng dài: 15:34:56  
==> Thông tin thời gian sau khi thêm 150 phút:  
18:4:56  
Giờ: 18  
Phút: 4  
Giây: 56  
Mili giây: 999  
15:34 == 18:04 ? False  
15:34 != 18:04 ? True  
15:34 > 18:04 ? False
```




Nội dung tiếp theo

Tìm hiểu struct DateTime