

Bài 8.7: Cấu trúc DateTime

- ✓ Mục đích và các thuộc tính
- ✓ Các phương thức thường dùng
- ✓ Các chuẩn định dạng ngày giờ
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Mục đích sử dụng và các thuộc tính

- ✓ Cấu trúc readonly DateTime dùng để mô tả thông tin về ngày tháng và thời gian trong ngày.
- ✓ Các trường read only:
 - ✓ **MaxValue**: đại diện cho giá trị tối đa mà đối tượng DateTime có thể chứa. Tương đương 23:59:59.9999999 ngày 31/12/9999.
 - ✓ **MinValue**: đại diện cho giá trị tối thiểu mà đối tượng có thể chứa. Tương đương 00:00:00.0000000 ngày 1/1/1.
 - ✓ **UnixEpoch**: giá trị của hằng số này tương đương 00:00:00.0000000 ngày 1/1/1970. Hằng số này định nghĩa thời điểm tại đó thời gian Unix tương đương 0.
- ✓ Các thuộc tính:
 - ✓ **Date**: lấy ngày của đối tượng hiện thời. Giờ được thiết lập về 00:00:00.
 - ✓ **Day**: lấy ra ngày của tháng.
 - ✓ **DayOfWeek**: lấy ra ngày của tuần thể hiện trong đối tượng hiện thời.
 - ✓ **DayOfYear**: lấy ngày của năm hiện tại trong đối tượng DateTime.

Mục đích sử dụng và các thuộc tính

✓ Các thuộc tính:

- ✓ Hour: lấy giờ của ngày được biểu diễn trong đối tượng.
- ✓ Kind: lấy giá trị chỉ ra liệu thời gian được biểu diễn trong đối tượng hiện tại là dựa trên thời gian địa phương hay theo giờ UTC hay một chuẩn khác.
- ✓ Milisecond: lấy số mili giây trong đối tượng hiện thời.
- ✓ Minute: lấy số phút.
- ✓ Month: lấy tháng.
- ✓ Now: lấy đối tượng DateTime thể hiện thời gian tại thời điểm hiện tại của máy tính theo giờ địa phương.
- ✓ Second: lấy số giây thể hiện bởi đối tượng hiện thời.
- ✓ Ticks: lấy số tic tắc.
- ✓ TimeOfDay: trả về giờ của ngày.
- ✓ Today: lấy ngày hiện tại.
- ✓ UtcNow: lấy đối tượng DateTime
- ✓ Year: lấy năm trong đối tượng hiện tại. Biểu diễn theo múi giờ UTC.

Các constructor

- ✓ `DateTime(long ticks)`: tạo đối tượng DateTime với số tic tắc cho trước.
- ✓ `DateTime(long ticks, DateTimeKind kind)`: tạo đối tượng của DateTime với số tic tắc cho trước theo thời gian địa phương hoặc UTC.
- ✓ `DateTime(int year, int month, int day)`: tạo đối tượng của DateTime với ngày, tháng, năm cho trước.
- ✓ `DateTime(int year, int month, int day, Calendar calendar)`: tạo đối tượng của DateTime với ngày tháng năm trong lịch cho trước.
- ✓ `DateTime(int year, int month, int day, int hour, int minute, int second)`: tạo đối tượng của DateTime với ngày, tháng, năm, giờ, phút, giây cho trước.
- ✓ `DateTime(int year, int month, int day, int hour, int minute, int second, DateTimeKind kind)`: tạo đối tượng của DateTime với ngày, tháng, năm, giờ, phút, giây và kiểu ngày giờ cho trước.

Các constructor

- ✓ `DateTime(int year, int month, int day, int hour, int minute, int second, int milisec)`: tạo đối tượng của `DateTime` với ngày, tháng, năm, giờ, phút, giây và mili giây cho trước.
- ✓ `DateTime(int year, int month, int day, int hour, int minute, int second, int milisec, DateTimeKind kind)`: tạo đối tượng của `DateTime` với ngày, tháng, năm, giờ, phút, giây và mili giây với kiểu ngày giờ cho trước.

Các phương thức thường dùng

Phương thức và mô tả
Add(TimeSpan value); tạo mới và trả về một đối tượng DateTime trong đó đã thêm một khoảng thời gian cho trước trong value vào đối tượng hiện thời.
AddDays(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số ngày cho trước trong value vào đối tượng hiện thời.
AddHours(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số giờ cho trước trong value vào đối tượng hiện thời.
AddMilliseconds(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số mili giây cho trước trong value vào đối tượng hiện thời.
AddMinutes(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số phút cho trước trong value vào đối tượng hiện thời.
AddSeconds(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số giây cho trước trong value vào đối tượng hiện thời.
AddMonths(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số tháng cho trước trong value vào đối tượng hiện thời.
AddYears(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số năm cho trước trong value vào đối tượng hiện thời.
AddTicks(double values); tạo mới và trả về một đối tượng DateTime trong đó đã thêm số tic tắc cho trước trong value vào đối tượng hiện thời.
Compare(DateTime d1, DateTime d2); so sánh 2 đối tượng d1 và d2 và trả về giá trị số nguyên thể hiện mối tương quan sớm hơn(âm), bằng(0) hay muộn hơn(dương) giữa chúng.

Các phương thức thường dùng

CompareTo(DateTime other); so sánh xem đối tượng hiện thời và other sớm hơn, muộn hơn hay tương đương nhau. Trả về giá trị số nguyên tương tự phương thức trên.

DaysInMonth(int year, int month); trả về số ngày của tháng và năm cho trước trong tham số.

Equals(DateTime other);

Equals(Object other);

Equals(DateTime d1, DateTime d2);

Trả về giá trị true/false xét xem liệu đối tượng hiện thời có tương đương với other về ngày giờ hay không. Hoặc d1 có tương đồng ngày giờ với d2 không.

FromFileTime(long fileTime); chuyển file thời gian trong window sang thời gian địa phương tương ứng. Tham số cho dưới dạng số tíc tắc.

GetDateTimeFormat(); chuyển giá trị của đối tượng sang chuỗi kí tự dạng chuẩn ngày giờ.

GetDateTimeFormat(char format); như trên nhưng chỉ định rõ ràng chuẩn ngày giờ cần chuyển đổi.

Các kí tự chỉ định có thể là d, D, f, F, g, G, m, M, o, O...

GetDateTimeFormat(IFormatProvider? provider); chuyển giá trị của đối tượng hiện tại sang string theo chuẩn provider cho trước.

GetDateTimeFormat(char format, IFormatProvider? provider); tương tự trên nhưng chỉ rõ chuẩn ngày giờ cần chuyển đổi.

GetHashCode(); lấy mã băm.

GetTypeCode(); trả về kiểu TypeCode của đối tượng DateTime hiện thời.

IsDaylightSavingTime(); xác định xem liệu đối tượng hiện tại có nằm trong phạm vi của thời gian ánh sáng ban ngày của múi giờ hiện tại không.

Các phương thức thường dùng

<code>IsLeapYear(int year);</code> kiểm tra xem một năm cho trước có phải năm nhuận không.
<code>Parse(string value);</code> <code>Parse(string value, IFormatProvider? provider);</code> Chuyển đổi chuỗi kí tự chứa thời gian, ngày giờ sang đối tượng <code>DateTime</code> tương ứng.
<code>SpecifyKind(DateTime value, DateTimeKind kind);</code> tạo một đối tượng <code>DateTime</code> có cùng số tic tắc với đối tượng hiện thời nhưng theo kiểu ngày giờ cho trong tham số thứ hai.
<code>Subtract(DateTime value);</code> Trả về đối tượng <code>TimeSpan</code> sau khi trừ đối tượng hiện thời cho tham số.
<code>ToLocalTime();</code> chuyển giá trị của <code>DateTime</code> hiện tại sang đối tượng thời gian cục bộ.
<code>ToLongDateString();</code> chuyển đổi dữ liệu trong đối tượng hiện thời sang chuỗi kí tự ngày tháng đầy đủ. <code>ToLongTimeString();</code> chuyển dữ liệu trong đối tượng hiện thời sang thời gian giờ phút đầy đủ.
<code>ToShortDateString();</code> <code>ToShortTimeString();</code> Tương tự trên nhưng cho ra string rút gọn của ngày, giờ.
<code>ToString();</code> trả về chuỗi kí tự đại diện cho đối tượng <code>DateTime</code> hiện tại. <code>ToString(string? format);</code> chuyển đổi giá trị của đối tượng <code>DateTime</code> hiện tại sang chuỗi kí tự cho trước sử dụng định dạng chuyển đổi cho trong tham số. Trả về chuỗi kí tự sau khi chuyển đổi. <code>ToString(IFormatProvider? provider);</code> trả về string đại diện cho đối tượng hiện thời theo định dạng của nền văn hóa bản địa được chỉ định trong tham số.
<code>ToUniversalTime();</code> chuyển đổi giá trị của đối tượng hiện thời sang đối tượng <code>DateTime</code> mới theo giờ UTC.
<code>ParseExact(string str, string format, IFormatProvider? provider);</code> chuyển đổi string chứa thông tin ngày giờ, thời gian trong <code>str</code> sang đối tượng <code>DateTime</code> theo đúng định dạng cho trước trong tham số <code>format</code> với nền văn hóa cho trước trong <code>provider</code> .

Các chuẩn định dạng ngày giờ

- ✓ Chuẩn định dạng ngày giờ sử dụng 1 kí tự chữ cái đơn để định nghĩa chuỗi kí tự đại diện của đối tượng ngày giờ.
- ✓ Bất kì định dạng ngày giờ nào chứa 1 hoặc nhiều kí tự bao gồm cả khoảng trắng được biên dịch như định dạng ngày giờ tùy biến cá nhân hóa.
- ✓ Ta có thể sử dụng định dạng trên theo 2 cách:
 - ✓ Để định nghĩa chuỗi kí tự là kết quả từ hành động định dạng.
 - ✓ Hoặc để định nghĩa biểu hiện của chuỗi kí tự đại diện cho ngày giờ có thể chuyển sang đối tượng DateTime hoặc DateTimeOffset qua thao tác chuyển đổi kiểu(Parsing).

Bảng các chuẩn định dạng ngày giờ

Kí tự định dạng	Mô tả và ví dụ
"d"	Định dạng ngày rút gọn. Ví dụ: 15/03/2025.
"D"	Định dạng ngày đầy đủ. Ví dụ: Sunday, Mar 20, 2025.
"f"	Định dạng ngày giờ đầy đủ dạng rút gọn: Sunday, Mar 20, 2025 15:34.
"F"	Định dạng ngày giờ đầy đủ dạng dài: Sunday, Mar 20, 2025 15:34:54.
"g"	Ngày giờ thông thường dạng ngắn: 15/06/2025 13:26.
"G"	Ngày giờ thông thường dạng dài: 15/06/2025 13:26:47.
"M", "m"	Định dạng tháng và ngày của tháng: June 28.
"O", "o"	Định dạng ngày giờ khứ hồi: 2025-08-24T01:02:03.0000000-07:00.
"R", "r"	Định dạng RFC1123: Sun, 20 Mar, 2025 20:45:30 GMT.
"S"	Ngày giờ dạng ngắn: 2022-04-20T13:14:57.
"t"	Giờ dạng rút gọn: 13:15.
"T"	Giờ dạng dài: 13:15:24.
"u"	Định dạng ngày giờ universal dạng rút gọn: 2022-04-20 13:15:36Z.
"U"	Định dạng ngày giờ universal dạng dài: Sunday, April 20, 2022 8:45:32 AM.
"Y", "y"	Định dạng năm và tháng: June 2025.
"d"	Ngày của tháng dạng 1-31. Ví dụ 1, 15, 30.
"dd"	Ngày của tháng dạng 2 số từ 01 đến 31. Ví dụ 01, 06, 25

Bảng các chuẩn định dạng ngày giờ

"ddd"	Tên thứ của tuần dạng rút gọn. Ví dụ Mon, Sun.
"dddd"	Tên ngày của tuần đầy đủ. Ví dụ Monday.
"f" đến "ffffff"	K chữ số đầu tiên trong phần thập phân của giây. Mỗi kí tự f tương ứng 1 chữ số.
"F" đến "FFFFFF"	K chữ số đầu tiên khác 0 trong phần thập phân của giây. Nếu k chữ số đều bằng 0 thì không hiển thị.
"g", "gg"	Phần thời đại. Ví dụ sau công nguyên A.D
"h"	Giờ dạng 12 tiếng. Ví dụ 1, 10.
"hh"	Giờ dạng 12 tiếng với 2 chữ số từ 01 đến 12. Ví dụ 03, 11.
"H"	Giờ dạng 24 tiếng từ 1 đến 23. Ví dụ 1, 23.
"HH"	Giờ dạng 24 tiếng từ 01 đến 23. Ví dụ 01, 21.
"K"	Thông tin múi giờ. Ví dụ Z, +07:00
"m"	Phút từ 0 đến 59. Ví dụ 0, 26.
"mm"	Phút từ 00 đến 59. Ví dụ 01, 27.
"M"	Tháng 1-12. Ví dụ 1, 3.
"MM"	Tháng từ 01-12. Ví dụ 03, 09.
"MMM"	Tên tháng 3 chữ cái. Ví dụ Jan, Sep.
"MMMM"	Tên tháng đầy đủ. Ví dụ January, September.
"s"	Giây từ 0-59. Ví dụ: 1, 58.
"ss"	Giây từ 00-59. Ví dụ: 01, 25.
"t"	Kí tự đầu của AM/PM. Ví dụ A, P.
"tt"	Kí hiệu AM hoặc PM.

Bảng các chuẩn định dạng ngày giờ

"y"	Năm. Từ 0 đến 99: 1, 6.
"yy"	Năm 2 chữ số từ 01 đến 99: 01, 99.
"yyy"	Năm 3 chữ số từ 001 đến 999.
"yyyy"	Năm có 4 chữ số từ 0001 đến 9999.
"yyyyy"	Năm có 5 chữ số: 02025.
"z"	Phần giờ của múi giờ: +7.
"zz"	Phần giờ 2 chữ số của múi giờ: +07.
"zzz"	Phần giờ và phút của múi giờ.
":"	Kí tự phân tách thời gian: 15:27:36.
"/"	Kí tự phân tách ngày tháng: 22/05/2025.
"string" 'string'	Chuỗi kí tự phân tách các thành phần của ngày, giờ.
%	Định nghĩa các kí tự theo sau nó là định dạng theo yêu cầu của người dùng.
\	Kí tự thường sử dụng đi kèm tại đầu các kí tự đặc biệt.
Bất kì kí tự nào khác	Không xác định. Văng ngoại lệ FormatException.

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 `DateTime` có tương đương hay muộn hơn đối tượng `DateTime` 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 `DateTime`.

Ví dụ

```
var dateFormat = "dd/MM/yyyy"; // định dạng ngày tháng
var timeFormat = "HH:mm:ss"; // định dạng giờ phút
var dateTimeFormat = "MMMM dd, yyyy HH:mm:ss";
var dateString = "27/04/2009";
var dateAndTime = "January 16, 2025 15:34:26";

DateTime now = DateTime.Now; // lấy ngày giờ tại thời điểm chương trình chạy
// hiển thị thời gian hiện tại:
Console.WriteLine($"Giờ hiện tại: {now.ToString(timeFormat)}");
Console.WriteLine($"Ngày hiện tại: {now.ToString(dateFormat)}");
Console.WriteLine($"Ngày giờ hiện tại: {now.ToString(dateTimeFormat)}");

// tạo đối tượng date time từ string với định dạng cho trước
DateTime birthDate = DateTime.ParseExact(dateString, dateFormat, null);
Console.WriteLine("Sinh nhật của tôi: " + birthDate.ToString(dateFormat));

var otherDateTime = DateTime.ParseExact(dateAndTime, dateTimeFormat, null);
Console.WriteLine("Ngày giờ đầy đủ: " + otherDateTime.ToLongDateString());
Console.WriteLine("Ngày giờ tự định dạng: " + otherDateTime.ToString(dateTimeFormat));
Console.WriteLine("Giờ tự định dạng: " + otherDateTime.ToString(timeFormat));
```

C:\WINDOWS\system32\cmd.exe

```
Giờ hiện tại: 10:48:54
Ngày hiện tại: 28/04/2022
Ngày giờ hiện tại: April 28, 2022 10:48:54
Sinh nhật của tôi: 27/04/2009
Ngày giờ đầy đủ: Thursday, January 16, 2025
Ngày giờ tự định dạng: January 16, 2025 15:34:26
Giờ tự định dạng: 15:34:26
```



Nội dung tiếp theo

Tìm hiểu thao tác với file