

## 第55章 DateTime構造体とTimeSpan構造体



DateTime構造体は、ある瞬間を表しますが、TimeSpan構造体は瞬間から瞬間までの間隔を表現します。

値を持ったDateTime構造体同士で差を求めることができます。結果はTimeSpan構造体となります。

たとえば、2006年12月10日をDateTime構造体に格納する方法はいくつかあります。

コンストラクタを利用して

```
dt = new DateTime(2006, 12, 10);
```

とするのが手っ取り早いかも知れません。あるいは、staticなParseメソッドを使って

```
dt = DateTime.Parse("2006/12/10");
```

と、する方法もあります。どちらも時刻を指定しなければ、0時0分0秒に設定されます。

DateTime構造体のコンストラクタには、多数のオーバーロードバージョンが存在します。この章では、次の2つを知っていれば充分でしょう。

```
public DateTime (
    int year,
    int month,
    int day
)
public DateTime (
    int year,
    int month,
    int day,
    int hour,
    int minute,
    int second
)
```

DateTime構造体のParseメソッドもオーバーロードバージョンがいくつかあります。

```
public static DateTime Parse (
    string s
)
```

sには、日付と時刻を格納した文字列を指定します。時刻のみを指定したときは本日の日付になります。

では、簡単なサンプルを見てみましょう。

```
// setdate01.cs
```

```
using System;
```

```
class setdate01
{
    public static void Main()
    {
        DateTime dt = new DateTime(1955, 5, 31);
        Console.WriteLine(dt.ToString("F"));

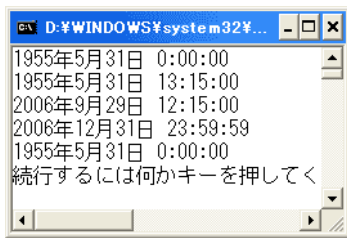
        dt = new DateTime(1955, 5, 31, 13, 15, 0);
        Console.WriteLine(dt.ToString("F"));

        dt = DateTime.Parse("12:15");
        Console.WriteLine(dt.ToString("F"));

        dt = DateTime.Parse("2006/12/31 23:59:59");
        Console.WriteLine(dt.ToString("F"));

        dt = DateTime.Parse("昭和30年5月31日");
        Console.WriteLine(dt.ToString("F"));
    }
}
```

実行結果は次のようになります。



日付の文字列に元号が含まれていてもよいことがわかります。

次にDateTime構造体どうしの差を求めてみましょう。この場合結果はTimeSpan構造体に格納されます。

```
// caldate01.cs

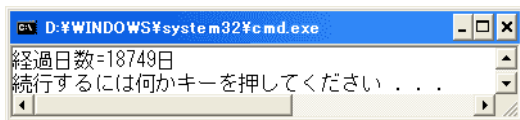
using System;

class caldate01
{
    public static void Main()
    {
        DateTime dt1 = new DateTime(1955, 5, 31);
        DateTime dt2 = DateTime.Now;

        TimeSpan ts = dt2 - dt1;

        Console.WriteLine("経過日数={0}日", ts.Days);
    }
}
```

実行結果は次のようになります。



すでに、上のサンプルプログラムで出てきていますが、TimeSpan構造体の 主なプロパティには、次のようなものがあります。

```
public int Days { get; }
public int Hours { get; }
public int Minutes { get; }
public int Seconds { get; }
public int Milliseconds { get; }
```

それぞれ、TimeSpan構造体が表す日数、時間、分、秒、ミリ秒を端数なしで取得します。

```
public double TotalDays { get; }
public double TotalHours { get; }
public double TotalMinutes { get; }
public double TotalSeconds { get; }
public double TotalMilliseconds { get; }
```

それぞれ、TimeSpan構造体が表す日数、時間、分、秒、ミリ秒の合計を端数を含めて取得します。

```
public long Ticks { get; }
```

このインスタンス内のタイマ刻みの数を取得します。タイマ刻みとは、100ナノ秒です。

TimeSpan構造体のコンストラクタには4つのオーバーロードバージョンがあります。

```
public TimeSpan (
    long ticks
)
public TimeSpan (
    int hours,
    int minutes,
    int seconds
)
public TimeSpan (
    int days,
    int hours,
    int minutes,
    int seconds
)
public TimeSpan (
    int days,
    int hours,
```

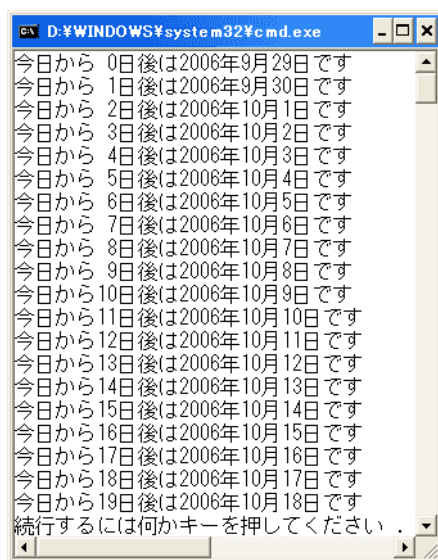
```
int minutes,  
int seconds,  
int milliseconds  
)
```

ticksにはタイマ刻み、hoursには時間、minutesには分数、secondsには秒、millisecondsにはミリ秒を指定します。

DateTime構造体にTimeSpan構造体の加算をすることにより、何日後とか何時間後の日時 を取得することができます。

```
// calcdatetime02.cs  
  
using System;  
  
class calcdatetime02  
{  
    public static void Main()  
    {  
        DateTime dt = DateTime.Now;  
        TimeSpan ts;  
  
        for (int i = 0; i < 20; i++)  
        {  
            ts = new TimeSpan(i, 0, 0, 0);  
            DateTime dtx = dt + ts;  
            Console.WriteLine("今日から{0, 2}日後は{1}です", i, dtx.ToString("D"));  
        }  
    }  
}
```

実行結果は次のようになります。



---

[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 30/Sep/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。