

Práce s časem

Úvod

V této kapitole si ukážeme, jak používat třídy `DateTime` a `TimeSpan` k ukládání a počítání s časem.

Třída `DateTime`

Na práci s časem budeme používat třídu `DateTime`. Do její instance můžeme uložit daný bod v časové linii. Tudíž datum i čas (proto `Date Time`). Náš objekt typu `DateTime` založíme jednoduše:

Deklarace a inicializace

```
// Založení DateTime bez parametrů
DateTime dt = new DateTime();
```

Když při založení nepoužijeme žádné parametry, tak se nám nastaví hodnoty na původní. Což je datum `1.1.0001 0:00:00`. Třída však obsahuje ještě několik dalších parametrických konstruktorů, které nám umožní určit náš vlastní čas:

Konstruktory třídy `DateTime`

```
// Založení se zadáním roku / měsíce / dne (3. ledna 2021)
DateTime dt = new DateTime(2021, 1, 3);

// Založení se zadáním roku / měsíce / dne a hodiny / minuty / sekundy (2. prosince 2021, 11:32)
DateTime dt2 = new DateTime(2021, 12, 2, 11, 32, 0);
```

Toto není výčet všech možných konstruktorů, ale tyto jsou nejvyužívanější.

Počítače pracují s jednotkou času, které říkáme ticks. Každá desetitisícina milisekundy, se dá vyjádřit jedním `tickem` (tzn. `1ms = 10000 ticks`). S touto jednotkou můžeme instanci třídy `DateTime` také založit, ale tento zápis se moc nepoužívá, jelikož z hlavy není tak jednoduché převést ticky na datum.

Konstruktory třídy `DateTime`

```
// Založení DateTimeu s datem 19 /12 / 1539 09:13:53
```

```
DateTime dt = new DateTime(485649547898999489);
```

Posun v čase

Pokud pracujeme s časem, nejspíš nebudeme chtít mít uložené jedny stálé hodnoty. Teď si tedy ukážeme, jak manipulovat s jednotlivými jednotkami času. Microsoft si pro nás připravil několik užitečných metod:

Posun času (1)

```
dt.AddYears(1);  
dt.AddMonths(2);  
dt.AddDays(5);  
dt.AddHours(4);  
dt.AddMinutes(1);  
dt.AddSeconds(99);  
dt.AddMilliseconds(700);  
dt.AddTicks(9999);
```

Zadání samozřejmě lze i různě variovat:

Posun času (2)

```
dt.AddYears(-1);  
dt.AddMonths(2);  
dt.AddDays(5.99);  
dt.AddHours(-4.156);  
dt.AddMinutes(1.19156);  
dt.AddSeconds(-2.11);  
dt.AddMilliseconds(0.555);  
dt.AddTicks(-4894498);
```

Pro každou jednotku existuje vlastní metoda `Add[jednotka]()`. Pomocí nich můžeme přidávat či odebírat jednotlivé jednotky času a posouvat tím datum, které jsme na začátku určili. Zadané hodnoty nemusí být pouze kladné. Můžeme metodě dát i minusová čísla a tím posunem čas zpět (ukázka 2).

Dny, hodiny, minuty, sekundy a milisekundy se dají zadávat i pomocí desetinného čísla. Roky a měsíce ne, jelikož nejsou pokaždé stejně dlouhé.

Ticky se musí zadávat v celých číslech, jelikož jsou nejmenší možná jednotka.

Získání aktuálního času

Dost často se při tvorbě programu, který nějakým způsobem používá `DateTime`, setkáme s tím, že budeme chtít získat čas, který je právě v tu danou dobu. Pro tyto účely nám slouží statická vlastnost `Now`.

Momentální čas

```
DateTime casTed = DateTime.Now;
```

Toto nám do nové proměnné `casTed` uloží datum a čas, kdy se tato operace provedla. Do teď jsme si ukázali jen manipulaci s třídou `DateTime`, ale nyní si představíme pár způsobů, jak z jejích hodnot získat lehce čitelný výstup. Jedná se o metodu `ToString()` a její speciální varianty.

Výpis času

```
// Dlouhý výpis data (úterý 3. září 2019)
Console.WriteLine( dt.ToLongDateString() );

// Dlouhý výpis času (13:03:16)
Console.WriteLine( dt.ToLongTimeString() );

// Krátký výpis data (3.09.2019)
Console.WriteLine( dt.ToShortDateString() );

// Krátký výpis času (13:03)
Console.WriteLine( dt.ToShortTimeString() );

// Krátký výpis data i času (3.09.2019 13:03:16)
Console.WriteLine( dt.ToString() );
```

Samozřejmě bychom mohli vzít ručně hodnoty z instance `DateTime` a nějak je poskládat, ale tyto formátovací metody nám práci velmi zjednoduší.

Třída TimeSpan

Další důležitou třídou je `TimeSpan`. Ta nám narozdíl od `DateTime` neudává přesný bod v čase, ale časový rozsah. Můžeme ho vytvořit podobně jako `DateTime` pomocí konstruktora, nebo ho můžeme získat díky odečtení dvou instancí třídy `DateTime`.

`TimeSpan` neumí ukládat roky a měsíce. Pouze dny a menší jednotky, jelikož roky i měsíce nemusí být vždy stejně dlouhé.

Inicializace třídy `TimeSpan`

```
// Časový úsek dlouhý 5 hodin, 2 minuty a 1 sekundu
TimeSpan ts = new TimeSpan(5, 2, 1);

// Získání TimeSpanu pomocí rozdílu dvou DateTimeů
TimeSpan rozdil = dt2 - dt1;

// Přičtení TimeSpanu k DateTimeu pomocí metody Add()
dt1.Add(ts);

// Přičtení TimeSpanu k DateTimeu pomocí operátoru
dt1 += ts;
```

Jak vidíte, se třídou `TimeSpan` se dá pracovat velmi jednoduše. Instance třídy `TimeSpan` mají rovněž přístupné vlastnosti jako je `Days`, `Hours`, `Minutes` a `Seconds`, které nám umožní získat jednotlivé časové úseky z intervalu.

Ukázky užití

Chceme si ohřát jídlo v mikrovlnné troubě a poté budeme chtít vědět, jaký bude čas, až se nám jídlo ohřeje. Nejdříve tedy zjistíme momentální čas pomocí `DateTime.Now` a poté k němu přičteme `TimeSpan`, který si předtím vytvoříme

Ukázka 1

```
// Momentální čas (začátek)
DateTime zacatek = DateTime.Now;

// Vytvoření TimeSpanu o délce 5 minut a 30 sekund (doba ohřívání)
TimeSpan casOhrivani = new TimeSpan(0, 5, 30);

// Čas, kdy bude jídlo připraveno ke konzumaci
DateTime konecnyCas = zacatek + casOhrivani;

// Výpis konečného času
Console.WriteLine(konecnyCas.ToShortTimeString());
```

V dalším příkladu budeme chtít změřit čas auta v závodě. Na začátku budeme čekat na stisk klávesy, po kterém se nám uloží startovní čas. Po druhém stisku klávesy pak zjistíme, jak dlouho auto trvalo projet celou trať:

Ukázka 2

```
// Start závodu
Console.WriteLine("Auto čeká na odstartování");
Console.ReadKey();
DateTime start = DateTime.Now;

Console.WriteLine("Auto vyrazilo!");

Console.ReadKey();
DateTime konec = DateTime.Now;

// Rozdíl mezi koncem a začátkem závodu je čas jeho průběhu
TimeSpan casZavodu = konec - start;
Console.WriteLine("Auto dorazilo do cíle v čase:" + casZavodu.ToString());
Console.ReadKey();
```

Shrnutí

V této kapitole jsme se naučili používat třídy `DateTime` a `TimeSpan`. Nesmíme zapomenout, že třída `DateTime` ukládá přesný čas a `TimeSpan` jen časový úsek.

Revision #2

Created 2025-05-21 10:32:24 UTC by Magdalena Dobešová

Updated 2025-05-21 10:42:29 UTC by Magdalena Dobešová