DateTime és TimeSpan használata

Répásné Babucs Hajnalka

Dátum tárolás

- A dátum és időpont tárolása egyetlen számként történik. Ez a szám long típusú, vagyis 64 bites, előjeles szám.
- Ha ez a szám nulla, akkor az 1. év január 1-jét 0 óra 0 perc 0 másodpercét jelenti. (DateTime.MinValue)
- E szám eggyel történő növelése az időpontban 100 nanomásodpercet jelent.
- Ez a szám maximum a 9999. év december 31-ét 23 óra 59 perc 59 másodpercet képes eltárolni. (DateTime.MaxValue)

Aktuális dátum 1.

- Aktuális dátum érték: DateTime.Now
- Aktuális dátum értéke számként: DateTime.Now.Ticks
- Aktuális év: DateTime.Now.Year
- Aktuális hónap: DateTime.Now.Month
- Aktuális nap: DateTime.Now.Day
- Aktuális óra: DateTime.Now.Hour
- Aktuális perc: DateTime.Now.Minute
- Aktuális másodperc: DateTime.Now.Second
- Aktuális ezredmásodperc: DateTime.Now.Millisecond

Aktuális dátum 2.

- Aktuális dátum: DateTime.Now.Date vagy DateTime.Today
  - Csak annyiban változik így, hogy 0 óra 0 perc, 0 másodpercet jelenít meg!
  - Ha tényleg csak a dátum kell, string formázást használunk.
- Aktuális időpont a napon belül: DateTime.Now.TimeOfDay

Aktuális dátum 3.

- Aktuális nap neve angolul: DateTime.Now.DayOfWeek
- Az aktuális nap hányadik napja az évnek: DateTime.Now.DayOfYear
- Egy adott hónap hány napos:
  - DateTime.DaysInMonth(2019, 2);

Dátum és idő megváltoztatása 1.

- A DateTime típusú adatokkal különféle számítási műveleteket is végezhetünk.
- A függvények Add szóval kezdődnek. Ezek segítségével egy adott értéket adhatunk hozzá egy DateTime típushoz.
- Például az AddDays segítségével napokat adhatunk hozzá, vagy vehetünk el egy dátumból, mivel a függvény negatív értéket is elfogad.
  - DateTime.Now.AddDays(-1)

Dátum és idő megváltoztatása 2.

- AddYears – évek változtatása
- AddMonths – hónapok változtatása
- AddDays – napok változtatása
- AddHours – órák változtatása
- AddMinutes – percek változtatása

Dátum és idő megváltoztatása 3.

- AddSeconds – másodpercek változtatása
- AddMilliseconds – ezredmásodperc változtatása
- AddTicks – dátum, idő változtatása 100 nanomásodperc egységenként

DateTime változó létrehozása

- Két leggyakoribb paraméterezése a konstruktor esetén:
- DateTime datum1 = new DateTime(év, hónap, nap);
- DateTime datum2 = new DateTime(év, hónap, nap, óra, perc, másodperc);

Dátum ellenőrzés 1.

- Szöveg dátummá alakítása:
  - DateTime.Parse
  - DateTime.TryParse
  - DateTime.TryParseExact
    - Speciális dátum formátum ellenőrzés

Dátum ellenőrzés 2.

```
string input = "2019-02-06";
DateTime dateTime;
if (DateTime.TryParse(input, out dateTime))
{
    Console.WriteLine(dateTime);
}
else Console.WriteLine("Nem jól adta meg
    a dátumot!");
```

Dátum ellenőrzés 3.

```
using System.Globalization;


string dateString = "Mon 16 Jun 8:30 AM
    2008";
string format = "ddd dd MMM h:mm tt
    yyyy";
DateTime dateTime3;
if (DateTime.TryParseExact(dateString,
    format, CultureInfo.InvariantCulture,
    DateTimeStyles.None, out dateTime3))
{  Console.WriteLine(dateTime3);}
else { Console.WriteLine("Not a date");}
```

Dátum formázás 1.

- DateTime ma = DateTime.Now;
- ma.ToString()
  - 2019. 02. 06. 9:30:06
- ma.ToString("yyyy. MMMM dd.")
  - 2019. február 06.
- ma.ToString("d")
  - Rövid dátum (2019. 02. 06.)
- ma.ToString("D")
  - Hosszú dátum (2019. február 6., szerda)

Dátum formázás 2.

- ma.ToString("F")
  - Teljes dátum, idő (2019. február 6., szerda 6:30:12)
- ma.ToString("g")
  - Általános dátum, idő (2019. 02. 06. 6:30)
- ma.ToString("t")
  - Rövid idő (6:30)
- ma.ToString("T")
  - Hosszú idő (6:30:12)

Dátum formázás 3.

- ma.ToLongDateString()
  - 2019. február 6., szerda
- ma.ToShortDateString()
  - 2019. 02. 06.
- ma.ToLongTimeString()
  - 9:30:06
- ma.ToShortTimeString()
  - 9:30

Dátumok összehasonlítása

- CompareTo() függvény segítségével

```
DateTime d1 = new DateTime(2017, 9, 01);
DateTime d2 = new DateTime(2019, 6, 15);
int res = d1.CompareTo(d2);
if (res < 0) Console.WriteLine("Első dátum a korábbi");
else if (res > 0) Console.WriteLine("A második dátum a korábbi ");
else Console.WriteLine("A két dátum megegyezik ");
```

TimeSpan osztály

- Az idő tárolására alkalmas
- Valós számként tárolja az időpontot
- Leggyakoribb időpont beállítás konstruktor hívással:
  - TimeSpan ido = new TimeSpan(óra, perc, másodperc);

Időpont beállítások 1.

- TimeSpan.FromDays(1)
  - Napok számát állítja be
  - 1.00:00:00
- TimeSpan.FromHours(1)
  - Az órák számát állítja be
  - 01:00:00
- TimeSpan.FromMinutes(1)
  - A percek számát állítja be
  - 00:01:00

Időpont beállítások 2.

- TimeSpan.FromSeconds(1)
  - A másodpercek számát állítja be
  - 00:00:01
- TimeSpan.FromSeconds(0.5)
  - 00:00:00.5000000
- TimeSpan.FromMilliseconds(1)
  - Az ezredmásodpercek számát állítja be
  - 00:00:00.0010000

Tulajdonságok 1.

- ido.Days
- ido.Hours
- ido.Minutes
- ido.Seconds
- ido.Milliseconds

- Az adott idő értékéből a napokat / órákat / perceket / másodperceket / ezredmásodperceket adja vissza int-ként

Tulajdonságok 2.

- ido.TotalDays
- ido.TotalHours
- ido.TotalMinutes
- ido.TotalSeconds
- ido.TotalMilliseconds

- Visszaadja double-ként, hogy az adott időpont összesen hány nap / óra / perc / másodperc / ezredmásodperc

Számítási műveletek

- ido.Add(ido2)
  - Két időpont összeadása. Visszatérési érték típusa TimeSpan
- ido.Subtract(ido2)
  - Két időpont különbsége. Visszatérési érték típusa TimeSpan

Idő ellenőrzés 1.

- Szöveg idővé alakítása:
  - TimeSpan.Parse
  - TimeSpan.TryParse
  - TimeSpan.TryParseExact
    - Speciális idő formátum ellenőrzés

Idő ellenőrzés 2.

```
string input = "9:30:06";
TimeSpan timeSpan;
if (TimeSpan.TryParse(input, out timeSpan))
{
    Console.WriteLine(timeSpan);
}
else Console.WriteLine("Nem jól adta meg
    az időpontot!");
```