C# Essentials

Veel Gebruikte Klasses

Sander De Puydt

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Korte inhoud

- Math
- String
- Datetime
- TimeSpan
- DispatcherTimer
- Random
- Samenvatting

Math

- Math klasse wordt gebruikt om getalbewerkingen te vereenvoudigen
- System.Math

```
static void Main(string[] args)
{
    // constante getallen
    Console.WriteLine("De omtrek van de cirkel met straal 3 is");
    Console.WriteLine(2*Math.PI*3);
    Console.WriteLine($"Indien je het getal e nodig hebt,"+
    " gebruik je Math.E {Math.E}");
}
```

Math

- Verschillende methodes om getallen af te ronden
  - Wat is het verschil tussen Round, Ceiling en Floor?

```
// 15 / 4 zonder komma getal?
double deling = 15.0 / 4;
Math.Floor(deling);
Math.Round(deling);
Math.Ceiling(deling);
```

Math

- Verschillende methodes om getallen af te ronden
  - Wat is het verschil tussen Round, Ceiling en Floor?

```
// 15 / 4 zonder komma getal?
double deling = 15.0 / 4;    // 3.75
Math.Floor(deling);          // 3
Math.Round(deling);          // 4
Math.Ceiling(deling);        // 4
```

Math

- Minimum, maximum en absolute waarde

```
int getal = -42;
int absoluteWaardeGetal = Math.Abs(getal);
Console.WriteLine(Math.Max(getal, absoluteWaardeGetal));
Console.WriteLine(Math.Min(getal, absoluteWaardeGetal));

// Wat is de verwachte output?
```

Math

- Machten, logaritmes, en vierkantswortels
  - Math.Pow(), Math.Log(), Math.Sqrt()

```
Math.Pow(2, 3); // Pow(grondtal, exponent)
Math.Log(8, 2);
// Log(getalWaarvanLogaritmeWordtGenomen, grondtal)
Math.Sqrt(225); // Sqrt() = vierkantswortel
```

Math

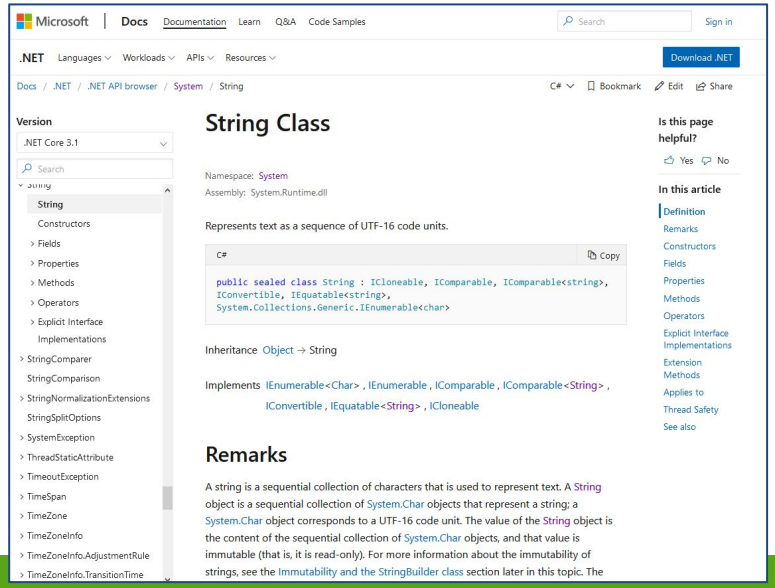
- Machten, logaritmes, en vierkantswortels
  - Navigeer met ctrl-klik op de methode

```
668         //
669         // Summary:
670         //     Returns the logarithm of a specified number in a specified base.
671         //
672         // Parameters:
673         //   a:
674         //     The number whose logarithm is to be found.
675         //
676         //   newBase:
677         //     The base of the logarithm.
678         //
679         // Returns:
680         //     One of the values in the following table. (+Infinity denotes System.Double.PositiveInfinity,
681         //     -Infinity denotes System.Double.NegativeInfinity, and NaN denotes System.Double.NaN.)
682         //     anewBase Return value a> 0 (0 <newBase< 1) -or-(newBase> 1) lognewBase(a) a<
683         //     0 (any value) NaN (any value) newBase< 0 NaN a != 1 newBase = 0 NaN a != 1 newBase
684         //     = +Infinity NaN a = NaN (any value) NaN (any value) newBase = NaN NaN (any value)
685         //     newBase = 1 NaN a = 0 0 <newBase< 1 +Infinity a = 0 newBase> 1 -Infinity a =
686         //     +Infinity 0 <newBase< 1 -Infinity a = +Infinity newBase> 1 +Infinity a = 1 newBase
687         //     = 0 0 a = 1 newBase = +Infinity 0
688         public static double Log(double a, double newBase);
```

String

- Er bestaan een heleboel functies om strings te manipuleren
  - Meer uitleg over een methode nodig?
    - Ctrl + klik in VisualStudio code
    - https://docs.microsoft.com/

```
//
// Summary:
//     Concatenates two specified instances of System.String.
//
// Parameters:
//   str0:
//     The first string to concatenate.
//
//   str1:
//     The second string to concatenate.
//
// Returns:
//     The concatenation of str0 and str1.
public static String Concat(String? str0, String? str1);
```



String

- String methodes

Compare()	vergelijkt de inhoud van strings (A<B = -1 , A=B = 0, A>B = 1)
Concat()	voegt strings samen
Equals()	vergelijken van de waarden in strings
Insert()	voegt een string in een andere string in
IndexOf()	geeft positie waar string begint
Join()	voegt een array naar een string met het opgegeven separator
IsNullOrEmpty	test of een string Null is of leeg ("")
Remove()	verwijdert een gedeelte van een sting
Replace()	vervangt een letterteken door een ander teken
PadLeft()/PadRight()	vult links/rechts met opgegeven karakters aan

String

- String methodes

Split()	splitst een string in een array van strings
StartsWith()/EndsWith	geeft true of false als een string begint/eindigt met opgegeven string
Substring()	leest een substring uit een string
ToLower()	zet een string om in kleine letters,
ToUpper()	zet een string om in hoofdletters
Trim()	verwijdert een bepaald teken links en recht van een string
TrimEnd()	verwijdert een bepaalde teken aan einde van string
TrimStart()	verwijdert een bepaalde teken aan begin van string

Datetime

- Hoewel we tijd zouden kunnen opslaan in string waardes, zou hier veel extra rekenwerk bijkomen.
  - Oplossing: Data type dat tijd en tijdspanne voorsteld
    - DateTime
    - TimeSpan

```
TimeSpan timeSpan = new TimeSpan(1,0,0);
Console.WriteLine($"Eén uur in TimeSpan klasse = {timeSpan}");

DateTime day = new DateTime(2020,3,18);
Console.WriteLine($"18 maart in 2020 [{day.ToString("yyyy-MM-dd")}]");
```

Datetime

- Aantal nuttige properties:

DateTime.Today	11/9/2020
DateTime.Now	11/9/2020 8:30:00 AM
datum.Date	11/9/2020 12:00:00 AM
datum.Day	9
datum.DayOfWeek	Monday
datum.DayOfYear	314
datum.Hour	8
datum.Minute	30
datum.Month	11
datum.TimeOfDay	8:30:00.0000000
datum.Year	2020

Datetime

- Aantal nuttige methodes:

datum.AddDays(36)	voegt dagen toe of trekt dagen af
datum.AddMonths(12)	voegt maanden toe of trekt maanden af
datum.AddYears(2)	voegt jaren toe of trekt jaren af
datum.Subtract(datum)	geeft verschil in dagen, uren en minuten (geen maanden, jaren)
datum.Subtract(datum).Days	geeft verschil in dagen tussen de opgegeven datums
DateTime.Parse (string)	zet string om naar datum
datum.ToLongDateString()	geeft lange datum: dddd mmmm yyyy
datum.ToShortDateString()	geeft korte datumnotatie: dd/mm/yyyy
datum.ToLongTimeString()	geeft lange tijdsnotatie:hh:mm:ss
datum.ToShortTimeString()	geeft korte tijdsnotatie:hh:mm

DispatcherTimer

- Wanneer er een opdracht herhaaldelijk op verschillende momenten uitgevoerd moet worden
  - DispatcherTimer wordt gebruikt om taken op vaste intervallen uit te voeren
    - Zet actie om te verwerken
    - Zet interval
    - Start dispatcher

DispatcherTimer

- DispatcherTimer code voorbeeld

```
public MainWindow()
{
    InitializeComponent();
    DispatcherTimer dispatcher = new DispatcherTimer();

    // Installeren van timer dmv de klasse aan te spreken.
    DispatcherTimer wekker = new DispatcherTimer();
    // Timer laten aflopen om de seconde.
    wekker.Tick += new EventHandler(DispatcherTimer_Tick);
    wekker.Interval = new TimeSpan(0, 0, 1);
    //uren, minuten, seconden
    // Timer laten starten
    wekker.Start();
    // TIJD instellen.
    DateTime tijd = DateTime.Now;
    LblTijd.Content = $"{tijd.ToLongDateString()}{tijd.ToLongTimeString()}";
}
```

DispatcherTimer

- DispatcherTimer code voorbeeld
  - Vergeet niet een label met de naam LblTijd te maken

```
private void DispatcherTimer_Tick(object sender, EventArgs e)
{
     LblTijd.Content =
          $"{DateTime.Now.ToLongDateString()}
               {DateTime.Now.ToLongTimeString()}";
}
```

Random

- De Random klasse wordt gebruikt wanneer er een willekeurig getal nodig is.
  - Dit kan nuttig zijn wanneer je:
    - Wilt dat het programma een onvoorspelbare actie neemt.
    - Een willekeurig getal wil genereren.
    - Een kans wil simuleren.

Random

- Voorbeeldcode Random klasse

```
static void Main(string[] args)
{
    // rand voorziet het genereren van willekeurige getallen
    Random rand = new Random();
    // je kan een seed meegeven aan het random object
    Random rand2 = new Random(2);
    // met Next() kan je een getal genereren
    Console.WriteLine(rand.Next());
    Console.WriteLine(rand.Next(2));

    // NextDouble() geeft een waarde uit [0, 1[
    Console.WriteLine(rand.NextDouble());
}
```

Samenvatting

- Werken met bestaande klasses maakt het programmeren eenvoudiger. We kunnen reeds ontwikkelde code hergebruiken.
  - Getal bewerkignen: Math
  - String bewerkingen: String
  - Tijd gerelateerde programmas: Datetime, Timespan, DispatcherTimer
  - willekeurige getallen: Random
- Gebruik maken van documentatie voor verduidelijking.