



VS2022 - C#13 - .NET 9.X

Console UI Helper Classes

Handleiding

Versie: 1-jan-2025

Johan Vandaele

INHOUDSOPGAVE

1	INLEIDING	3
1.1	Doelstelling	3
1.2	Vereiste voorkennis	3
1.3	Nodige software	3
1.4	Source Code	4
2	METHODES VOOR INPUT EN OUTPUT	5
2.1	De ConsoleHelper Class	5
2.1.1	Inleiding	5
2.1.2	NuGet packages	5
2.1.3	Enumerations	5
2.1.4	Settings	5
2.2	Menu Helper Class	6
2.2.1	Inleiding	6
2.2.2	De menu	6
2.2.3	Voorbeeld van een menu	8
2.2.4	Methods	10
3	DEMO PROJECT	13
4	MENU VOOR DE VOORBEELDEN	14
5	WINDOWS TERMINAL – WINDOWS CONSOLE HOST	17
5.1	Inleiding	17
5.2	Via Windows Settings	17
5.3	Via Visual Studio	17
5.4	Via Windows Terminal	18
6	COLOFON	20

1 Inleiding

1.1 Doelstelling

Je leert in dit document werken met de helper classes voor de **Console** User Interface (**Console App (.NET Core)**). In deze classes vind je een aantal methodes terug die het je eenvoudiger maken om console programma's te schrijven.

Deze methodes omvatten volgende onderdelen:

- Input
- Output
- Menu

Het is evenwel niet de bedoeling om de interne werking van deze methodes in te studeren. Je moet deze methodes wel kunnen gebruiken in toepassingen waarin de Console als User Interface wordt gebruikt.

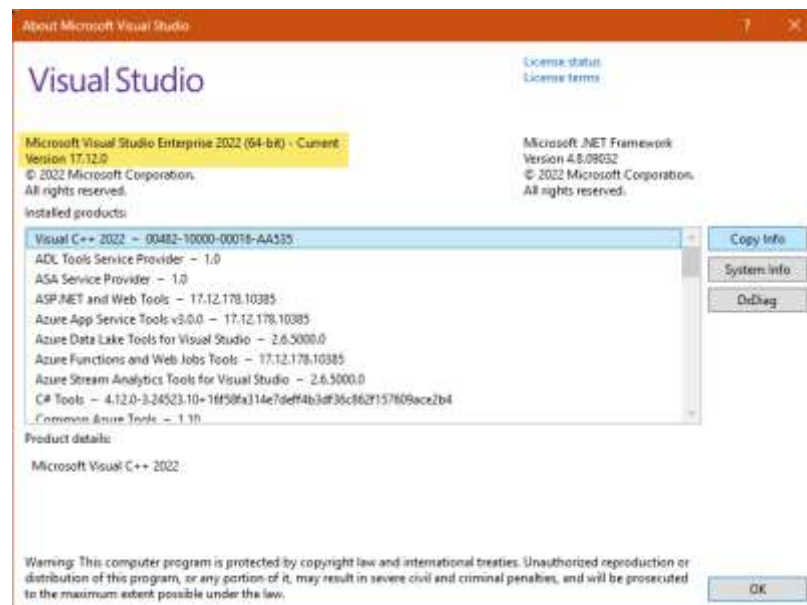
1.2 Vereiste voorkennis

- C# PF

1.3 Nodige software

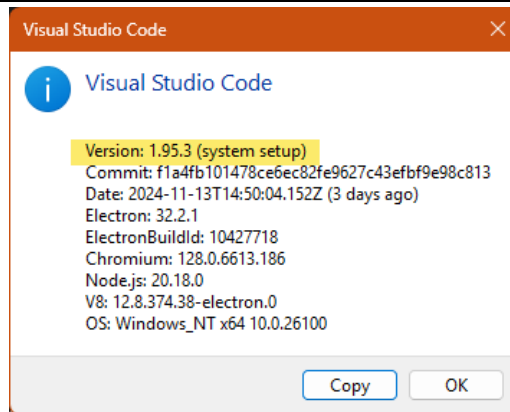
- Visual Studio 2022 versie 17.12.0+

<https://visualstudio.microsoft.com/vs/>



- Visual Studio Code versie 1.95.3+ (optioneel)

<https://www.visualstudio.com/downloads/>



- .Net9 versie 9.0.100+

<https://dotnet.microsoft.com/en-us/download/dotnet/9.0>

Je kan de geïnstalleerde versie van **.NET** opvragen.

Open de opdrachtprompt window en tik in:

```
dotnet --version
```

 A screenshot of a Windows Command Prompt window. The title bar says "Command Prompt". The text inside shows the command prompt output for the 'dotnet' command:


```
Microsoft Windows [Version 10.0.26100.2314]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks          Display the installed SDKs.
  --list-runtimes      Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\User>dotnet --version
9.0.100

C:\Users\User>
```

- Zet alle solutions die je maakt in de theorie, de oefeningen en de test in **GitHub**. Maak daarvoor een **private** repository aan en geef de instructeur toegang daarop.

<https://github.com/>

1.4 Source Code

<https://github.com/JohanVandaele/ConsoleHelper>

2 Methodes voor Input en Output

2.1 De ConsoleHelper Class

2.1.1 Inleiding

Om het Console project te vereenvoudigen en overzichtelijk te houden gaan we gebruik maken van een Input/output helper class. Deze helper class is een partial class van `program.cs`. Hierdoor hoef je zelf geen input meer te valideren of output te genereren en kunnen we ons uitsluitend concentreren op het essentiële gedeelte van de cursus. De interne werking van deze class maakt geen deel uit van de opleiding. Je hoeft ze enkel te kunnen toepassen in de voorbeelden, oefeningen en testen. Je kan van de methodes uit deze class gebruik maken in de console toepassingen die je maakt.

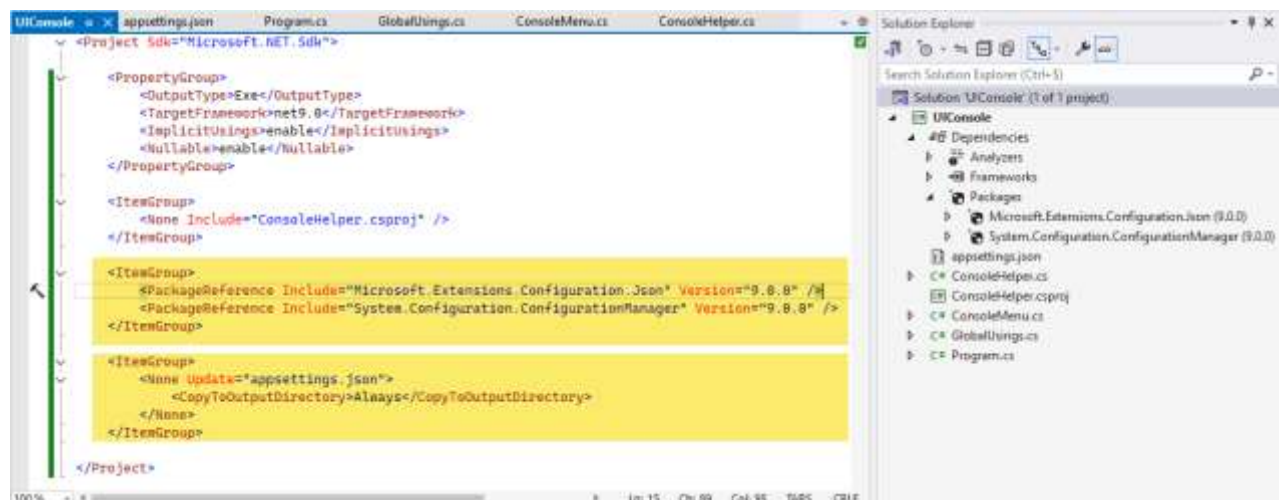
Je copieert de classes `ConsoleHelper.cs`, `ConsoleMenu.cs` en `appsettings.json` in het UI project vanuit de startbestanden die bij deze handleiding horen.

2.1.2 NuGet packages

Installeer volgende NuGet packages:

- Microsoft.Extensions.Configuration.Json
- System.Configuration.ConfigurationManager

Wijzig het Project-bestand `UIConsole.csproj`:



Deze packages laten toe dat je een aantal instellingen kunt terugvinden in het bestand `appSettings.json`.

2.1.3 Enumerations

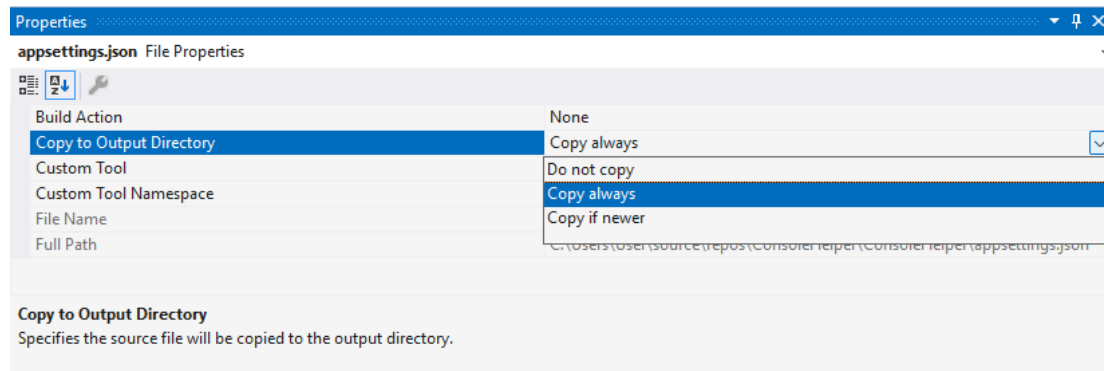
- OptionMode
 - Optional : input van een waarde is niet verplicht (kan empty of null zijn)
 - Mandatory : input van een waarde is verplicht
- SelectionMode
 - Single : je kan één object kiezen uit een lijst
 - Multiple : je kan meerdere objecten kiezen uit een lijst
 - None : je kan geen objecten kiezen uit een lijst. Toont enkel de objecten

2.1.4 Settings

In `appSettings.json` kan je een aantal instellingen doen die de uitvoering van deze Helper classes kunnen beïnvloeden.

Indien de settings niet worden teruggevonden in `appSettings.json`, worden default settings toegepast.

Zorg ervoor dat deze settings altijd mee worden gecopiëerd naar de map waar de toepassing wordt uitgevoerd.



Project ConsoleHelper : appsettings.json	
1	{
2	"ConsoleSettings": {
3	"AlternateRows": "True",
4	"Beep": "True",
5	"ListPageSize": "19",
	"ShowWindow": "3" // HIDE = 0, MAXIMIZE = 3, MINIMIZE = 6, RESTORE = 9
	}
	}

- (1) De naam van deze setting is **ConsoleSettings**.
- (2) **AlternateRows = True** : Toont bij een lijst de even en oneven rijen in een andere kleur. Een andere waarde toont alle rijen in dezelfde kleur.
- (3) **Beep = true** : Laat een pieptoon horen wanneer er een input wordt gevraagd. Een andere waarde laat geen piep horen.
- (4) **ListPageSize = aantal** : Bij een lange lijst kan je paginering toepassen. Hier geef je in uit hoeveel lijnen een pagina bestaat. Na elke pagina heb je de keuze om naar de volgende pagina te gaan, onmiddellijk verder gaan naar het einde van de lijst of het tonen van de lijst stopzetten. Is de waarde "0" dan wordt geen paginering toegepast.
- (5) **ShowWindow = 0/3/6/9** : De manier waarop het Console window initiëel wordt geopend.

De settings **AlternateRows**, **Beep** en **ListPageSize** kan je ook in je programma aanpassen.

2.2 Menu Helper Class

2.2.1 Inleiding

Om het Console project eenvoudig en overzichtelijk te houden gaan we gebruik maken van een I/O-helper class (`ConsoleHelper.cs`) en een menu helper class (`ConsoleMenu.cs`). Deze classes zijn een partial classes van `Program.cs`. Hierdoor hoeft je zelf geen menu of I/O operaties meer te programmeren en kunnen we ons uitsluitend richten op de essentie van de cursus.

Je kan deze helper classes vanuit de startbestanden overnemen in de root van het Console project.

We plaatsen vervolgens de volledige menu in een object en maken gebruik van methodes om de menu te manipuleren. De interne werking van de menu maakt geen deel uit van de opleiding. Je hoeft ze enkel te kunnen toepassen.

2.2.2 De menu

De menu bestaat uit 3 soorten items die overgeërfd worden van `MenuItem`:

- Submenu
- MenuActions
- MenuLijn

De menuitems worden getoond in de volgorde van voorkomen in de lijst.

MenuItem

Property	Type	Omschrijving
Id	int	"Unieke" id van een menuitem. Via deze id kan je een menuitem activeren/deactiveren of zichtbaar/onzichtbaar maken.
Label	string	Label van het menuitem; Leeg indien Hoofdmenu
SubMenuTitel	string	Indien submenu: de titel van de submenu
IsActive	Enum: MenuItemActive	Mogelijke waarden: Enabled, Disabled

SubMenu : MenuItem

Property	Type	Omschrijving
MenuItems	List<MenuItem>	Lijst van de onderliggende menuitems van deze submenu
Richting	Enum: Direction	Mogelijke waarden: Horizontal, Vertical

MenuAction : MenuItem

Property	Type	Omschrijving
MenuAction00	Action	Uit te voeren methode
MenuAction01	Action<object>	Niet in gebruik
Par01	object	Niet in gebruik

MenuLijn : MenuItem

Property	Type	Omschrijving
geen		Tekent een scheidingslijn tussen menuitems

Methodes

SetActive(SubMenu *sm*, List<int> *menuIds*, MenuItemIsActive *isActive*)

Zet een lijst van menuitems in een submenu op **enabled** of **disabled**.

SetVisible(SubMenu *sm*, List<int> *menuIds*, MenuItemVisible *visible*)

Zet een lijst van menuitems in een op **visible** of **hidden**.

SetLabel(SubMenu *sm*, List<int> *menuIds*, string *label*)

Wijzigt de label tekst van een menu item.

ResetMenu(SubMenu *subMenu*)

Plaats de volledige menu terug in zijn oorspronkelijke vorm.

ToonMenu(string *appTitel*, SubMenu *menuItem*)

Start de menu van de toepassing.

PrintMenu(string *appTitel*, SubMenu *menuItem*)

2.2.3 Voorbeeld van een menu

Als voorbeeld voeg je volgende code toe in `program.cs`:

UIConsole - Program.cs

```
namespace UIConsole;

public partial class Program
{
    // -----
    // Static variables - Application
    // -----
    public static string MenuGegevens => $"";

    // ----
    // Menu
    // ----
    public static SubMenu menu = new SubMenu
    (
        01, null, "Hoofdmenu", MenuItemActive.Enabled, MenuItemVisible.Visible, new List<MenuItem>
        {
            new SubMenu(02,"<A>ccount", "AccountMenu", MenuItemActive.Enabled
            , MenuItemVisible.Visible, new List<MenuItem>
            {
                new MenuItemAction (03,"<I>nloggen", "Inloggen", MenuItemActive.Enabled, MenuItemVisible.Visible, Inloggen),
                new MenuItemAction (04,"<U>itloggen", "Uitloggen", MenuItemActive.Disabled, MenuItemVisible.Visible, Uitloggen),
                new MenuItemAction (05,"<W>ijzig paswoord", "Wijzig paswoord", MenuItemActive.Disabled,
                MenuItemVisible.Hidden, AanpassenPaswoord),
                new MenuItemAction (06,"Zet klant <A>ctief", "Activeren klant", MenuItemActive.Disabled
                , MenuItemVisible.Hidden, ActiveerUser),
                new MenuItemLijn (),
                new MenuItemAction (08,"<T>oon accountgegevens", "Gegevens klant", MenuItemActive.Disabled
                , MenuItemVisible.Hidden, ToonAccountGegevens),
                new MenuItemAction (09,"<R>egistreren", "Nieuwe klant", MenuItemActive.Enabled
                , MenuItemVisible.Visible, Registeren),
            },
            new MenuItemAction(10,"<B>estellen", "Bestel een voorstelling", MenuItemActive.Disabled
            , MenuItemVisible.Hidden, KiesGenre),
            new SubMenu(11,"<R>eservaties", "ReservatieMenu", MenuItemActive.Disabled
            , MenuItemVisible.Hidden, new List<MenuItem>
            {
                new MenuItemAction (12,"<T>oon reservaties", "Lijst van reservaties", MenuItemActive.Enabled
                , MenuItemVisible.Hidden, ToonReservaties),
                new MenuItemAction (13,"<V>erwijder reservaties", "Verwijder gereserveerde voorstellingen"
                , MenuItemActive.Enabled, MenuItemVisible.Hidden, VerwijderBevestigdeReservaties),
            },
            new SubMenu(14,"<W>inkelmandje", "WinkelmandjeMenu", MenuItemActive.Disabled
            , MenuItemVisible.Hidden, new List<MenuItem>
            {
                new MenuItemAction(15,"<T>onen", "Overzicht winkelmandje", MenuItemActive.Enabled
                , MenuItemVisible.Hidden, ToonMandje),
                new MenuItemAction(16,"<V>erwijderen", "Verwijder", MenuItemActive.Disabled
                , MenuItemVisible.Hidden, VerwijderBestelling),
                new MenuItemAction(17,"<A>frekenen", "Reken af", MenuItemActive.Disabled
                , MenuItemVisible.Hidden, Afrekenen),
            },
        ),
    );

    // ----
    // Main
    // ----
    private static void Main(string[] args)
    {
        //PrintMenu("Het Cultuurhuis", menu); // Toon menustructuur
        ToonMenu("Het Cultuurhuis", menu); // Start het programma
    }

    // -----
    // ----- S E C U R I T Y -----
    // -----

    // -----
    // Inloggen
    // -----
    public static void Inloggen()
    {
        throw new NotImplementedException();
    }

    // -----
    // Uitloggen
    // -----
    public static void Uitloggen()
    {

```



```

4      throw new NotImplementedException();
4    }

4    // -----
4    // AanpassenPaswoord
4    // -----
4    public static void AanpassenPaswoord()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // Activeer User
4    // -----
4    public static void ActiveerUser()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // ToonAccountGegevens
4    // -----
4    static public void ToonAccountGegevens()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // Registreren
4    // -----
4    public static void Registreren()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // ----- B E S T E L -----
4    // -----

4    // -----
4    // Kies Genre
4    // -----
4    static public void KiesGenre()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // Toon Reservaties
4    // -----
4    public static void ToonReservaties()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // Verwijder Bevestigde reservatie
4    // -----
4    public static void VerwijderBevestigdeReservaties()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // Toon mandje
4    // -----
4    public static void ToonMandje()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // VerwijderBestelling
4    // -----
4    public static void VerwijderBestelling()
4    {
4        throw new NotImplementedException();
4    }

4    // -----
4    // Afrekenen
4    // -----
4    static public void Afrekenen()
4    {
4        throw new NotImplementedException();
4    }
4 }

```

Voorbeeld van `appsettings.json` bij gebruik van specifieke settings voor de toepassing.

ConsoleHelper – appsettings.json

```
{
  "ConsoleSettings": {
    "AlternateRows": "True",
    "Beep": "True",
    "ListPageSize": "19",
    "ShowWindow": "3" // HIDE = 0, MAXIMIZE = 3, MINIMIZE = 6, RESTORE = 9
  }
}
```

Voorbeeld van de menu (met `PrintMenu()`):



47

- (1) De variabele `MenuGegevens`. Deze tekst verschijnt automatisch bovenaan de menu.
- (2) De menu structuur met de naam van de gekozen naam `menu`. Je kan in één toepassing meerdere menus aanmaken elk met hun eigen naam. Een menu item kan verwijzen naar een bepaalde methode (4).
- (3) De `main` methode waarin je de menu oproept.
- (4) De methodes naar waar de menu verwijst.
- (5) Optioneel kan je ook configuratie settings inlezen vanuit `appsettings.json`.

2.2.4 Methods

Je kan nu ook van de volgende methodes gebruiken in je consoleprogramma:

1. `void DrukToets()`
 - Returns : void
2. `string LeesString(string label, int maxLength, OptionMode optionMode)`
 - label : tekst voor input
 - minLength : minimum lengte van de ingave
 - maxLength : maximum lengte van de ingave
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : De ingegeven waarde als string
3. `string LeesPaswoord(string label, int minLength, int maxLength, OptionMode optionMode)`

- label : tekst voor input
 - minLength : minimum lengte van de ingave
 - maxLength : maximum lengte van de ingave
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : De ingegeven waarde als string
4. **DateTime?** LeesDatum(**string** label, **DateTime** MinDate, **DateTime** MaxDate, **OptionMode** optionMode)
- label : tekst voor input
 - MinDate : de vroegste datum voor ingave
 - MaxDate : de uiterste datum voor ingave
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : De ingegeven waarde als datetime
5. **int?** LeesInt(**string** label, **int** Min, **int** Max, **OptionMode** optionMode)
float? LeesFloat(**string** label, **float** Min, **float** Max, **OptionMode** optionMode)
decimal? LeesDecimal(**string** label, **decimal** Min, **decimal** Max, **OptionMode** optionMode)
T? LeesGetal<**T**>(**string** label, **T** min, **T** max, **OptionMode** optionMode = **OptionMode.Optional**) **where** **T** : **INumber**<**T**>
- label : tekst voor input
 - Min : de minimum waarde van de ingave
 - Max : de maximum ingave van de ingave
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : De ingegeven waarde
6. **Bool?** LeesBool(**string** label, **OptionMode** optionMode)
- label : tekst voor input
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : false of true, naargelang de ingave (Y=true, N=false)
7. **List<object>** LeesLijst(**string** titel, **IEnumerable<object>** I, **List<string>** DisplayValues, **SelectionMode** selectionMode, **OptionMode** optionMode)
- title : tekst voor input
 - objecten : een lijst van objecten waaruit men een keuze kan maken
 - displayValues : de waarde die op de console wordt getoond voor elk van de objecten
 - selectionMode : Keuze uit volgende mogelijkheden:
 - Single (keuze van één object),
 - Multiple (keuze van meerdere objecten),
 - None (geen keuze, enkel om de lijst van de objecten te tonen)
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : Een lijst van de gekozen objecten
8. **object** LeesKeuzeUitLijst(**string** label, **List<object>** keuzeLijst, **OptionMode** optionMode)
- label : tekst voor input
 - keuzeLijst : keuzemogelijkheden
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : de gemaakte keuze
9. **string** LeesTelefoonnummer (**string** label, **OptionMode** optionMode)
- label : tekst voor input
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : telefoonnummer
10. **string** LeesEmailAdres(**string** label, **OptionMode** optionMode)
- label : tekst voor input
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : email adres
11. **string** LeesWebsiteUrl(**string** label, **OptionMode** optionMode)
- label : tekst voor input
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : URL van een website
12. **string** LeesRegex(**string** label, **Regex** regex, **OptionMode** optionMode)
- label : tekst voor input
 - regex : het formaat van de ingave
 - optionMode : Optional (geen verplichte ingave), Mandatory (verplichte ingave)
 - Returns : de ingave

13. `void ToonFoutBoodschap ToonFoutBoodschap(string tekst)`
`void ToonInfoBoodschap (string tekst)`
- tekst : tekst van de foutboodschap

3 Demo project

Om het gebruik van de **ConsoleHelper** te verduidelijken maken we een demo programma waarin we alle methodes gebruiken.

Voeg aan de solution een nieuw console project toe met de naam **UIConsole**. Copieer daarin de **ConsoleHelper.cs** de **ConsoleMenu.cs** classes, het bestand **appSettings.json** en de **Program.cs** class die je vindt bij de startbestanden onder de map **UIConsole**. Je maakt van dat project het startup project.

De source code van al deze classes kan je vinden op:

<https://github.com/JohanVandaele/ConsoleHelper>

Je kan hier zelf ook nog testen of voorbeelden aan toevoegen.

Je kan nu de werking van **UIConsole** bekijken.

Wijzig je in de **Main()** method de oproepen **ToonMenu()** naar **PrintMenu()** dan zie je volgende structuur van het menu:

Project UIConsole : Program.cs

```
static void Main(string[] args)
{
    MenuGegevens = $"Demo toepassing voor het testen van de ConsoleHelper methods";

    // - - -
    // Menu
    // - - -
    PrintMenu("UI Demo", menu); // Toon menustructuur
    //ToonMenu("UI Demo", menu); // Start het programma
}
```



4 Menu voor de voorbeelden

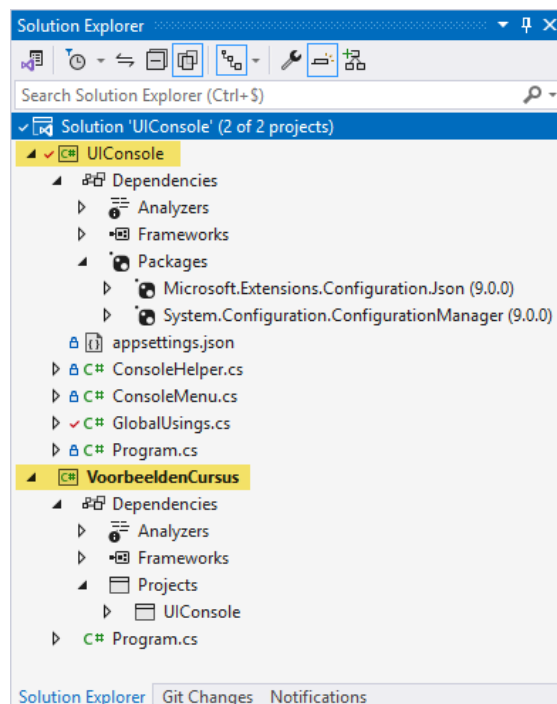
Om de voorbeelden en oefeningen in de cursus toegankelijk en overzichtelijk te maken, gaan we gebruik maken van een menustructuur die geleidelijk aan opgebouwd wordt naargelang de theorie vordert. Voor elk voorbeeld voegen we een menu-item toe.

In de menu kan je een getal ingeven om het desbetreffende voorbeeld uit te voeren of een 'X' om het programma te stoppen.

Deze code wordt in de voorbeelden verder uitgewerkt:

- Voeg voor elk voorbeeld een menuitem toe.
- Voeg voor elk voorbeeld een methode toe met de code van het voorbeeld uit de cursus.

Maak van het Console **VoorbeeldenCursus** project het startup project.



Program.cs in het project **VoorbeeldenCursus** project:

VoorbeeldenCursus - Program.cs	
	<pre> using System.Reflection; namespace UIConsole; public partial class Program { private static readonly List<string> Menu = // Spread operator [1 "Menu Item 0", "Menu Item 1", "Menu Item 2", // "Menu Item n", / // // Plaats hier de verschillende menu items //]; private static string Titel(string t) => \$"{Ansi.CURSORHOME}{Ansi.EraseScreen}{new string('=', t.Length)}\n{t}\n\nnew string('=', t.Length)}\n"; private static void Main(string[] args) { StartConsole(); 2 Console.Title = "EFCoreStart"; var keuze = string.Empty; </pre>

```

while (keuze != "X")
{
    //ResetConsole(); // Problem when using Terminal !!!
    Console.Write($"{ConsBGC}{ConsFGC}{Ansi.EraseScreen}{Ansi.CURSORHOME}");
    Console.WriteLine($"{Ansi.UnderLineOn}MENU{Ansi.UnderLineOf}");

    var item = 0;
    foreach (var m in Menu) Console.WriteLine($"{item++}. {m}");

    keuze = LeesString("Keuze ('X' om te stoppen):", 1, 2, OptionMode.Mandatory)!.ToUpper();

    if (keuze != "X")
    {
        Seperator = true;

        // Reflection
        try
        {
            //ResetConsole(); // Problem when using Terminal !!!
            //Console.Write($"{ConsBGC}{ConsFGC}{Ansi.EraseScreen}{Ansi.CURSORHOME}");

            if (keuze.All(char.IsNumber) && int.Parse(keuze) < Menu.Count)
            Console.WriteLine(Titel($"{"00"[..(-keuze.Length + 2)] + keuze} " + ". " + Menu.ElementAt(int.Parse(keuze))));

            //typeof(Program).InvokeMember($"{Item{"00".Substring(0, -keuze.Length + 2)} + keuze}"
            typeof(Program).InvokeMember
            (
                $"{Item{"00"[..(-keuze.Length + 2)] + keuze}"
                , BindingFlags.InvokeMethod | BindingFlags.Static | BindingFlags.NonPublic
                , null
                , null
                , new object[] { new object[] { Titel($"{"00"[..(-keuze.Length + 2)] + keuze} " + ". " +
Menu.ElementAt(int.Parse(keuze)), 123 } }
                );
            }
            catch (Exception)
            {
                ToonFoutBoodschap("Ongeldige keuze");
            }
        }

        // ===
        // End
        // ===
        if (keuze == "X") break;

        DrukToets();
    }
}

// -----

private static bool Seperator = false;

private static void SeperateLog()
{
    if (!Seperator) return;

    Thread.Sleep(2000); // Wait to Finish logging threat
    if (Seperator) Console.WriteLine("\n- - - - -\n");
    Seperator = false;
}

// -----

private static void ProcessException(Exception e)
{
    Console.WriteLine($"{Ansi.UnderLineOn}Exception{Ansi.UnderLineOf}\n\n{e.Message}{(e.InnerException
!= null ? "\n\n" + e.InnerException.Message : "")}");
    Thread.Sleep(2000); // Wait to Finish logging threat
}

// -----
// Menu-items
// -----
// 00. Menu Item 0
private static void Item00(object[] args)
{
    try
    {
        ToonTekst("Kiekeboe", Ansi.FDMAGENTA);
        ToonTekst("Kiekeboe");
        ToonInfoBoodschap("InfoTekst");
        ToonFoutBoodschap("FoutTekst");
    }
    catch (Exception e)
    {
        ProcessException(e);
    }
}

```

```
// -----  
3 // 01. Menu Item 1  
private static void Item01(object[] args)  
{  
    try  
    {  
        ToonInfoBoodschap("Hello World");  
    }  
    catch (Exception e)  
    {  
        ProcessException(e);  
    }  
}  
  
// -----  
3 // 02. Menu Item 2  
private static void Item02(object[] args)  
{  
    try  
    {  
        ToonInfoBoodschap("Hello Johan");  
    }  
    catch (Exception e)  
    {  
        ProcessException(e);  
    }  
}  
}
```

1. Lijst van menu items
2. Aangepaste titel
3. Methodes die opgeroepen worden voor elk menu item

5 Windows Terminal – Windows Console Host

5.1 Inleiding

Voor het uitvoeren van een console toepassing kan je kiezen tussen 2 apps: **Terminal** of **Console Host**.

Deze keuze kan je maken op 3 manieren:

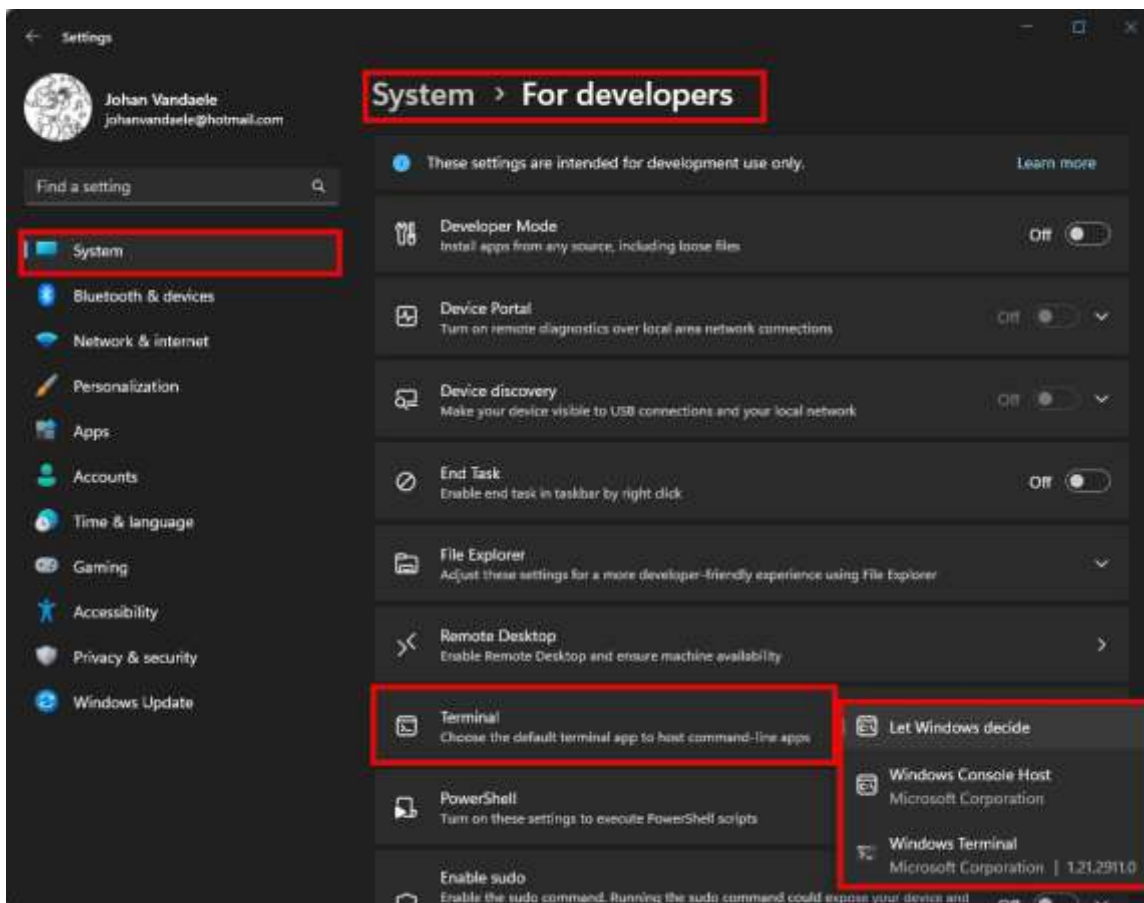
- Via **Windows Settings**
- Via **Visual Studio**
- Via **Windows Terminal**

5.2 Via Windows Settings

Ga naar Windows Settings:

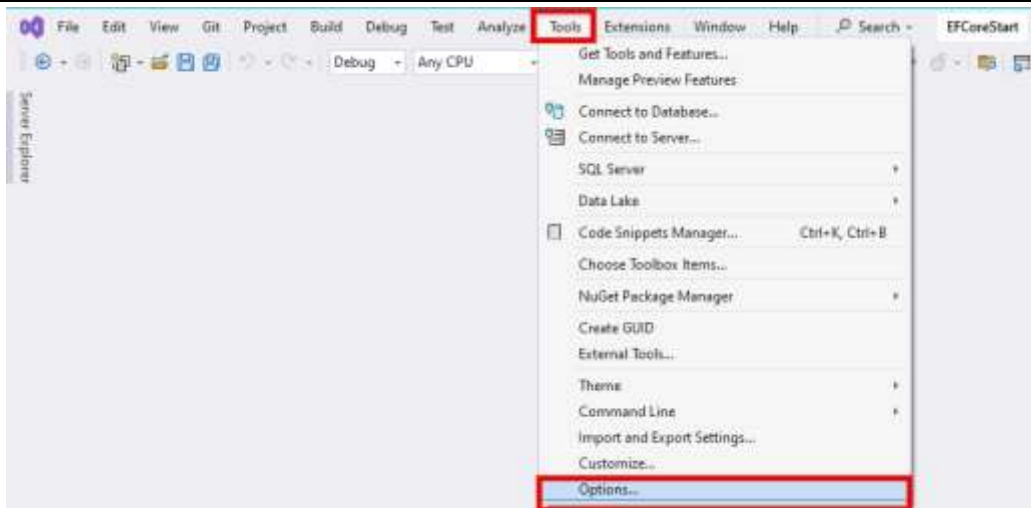
- Druk **Windows + I**
- Tik **Settings** in de windows menu

Kies **System** > **For developers** > Maak bij het item **Terminal** je keuze.



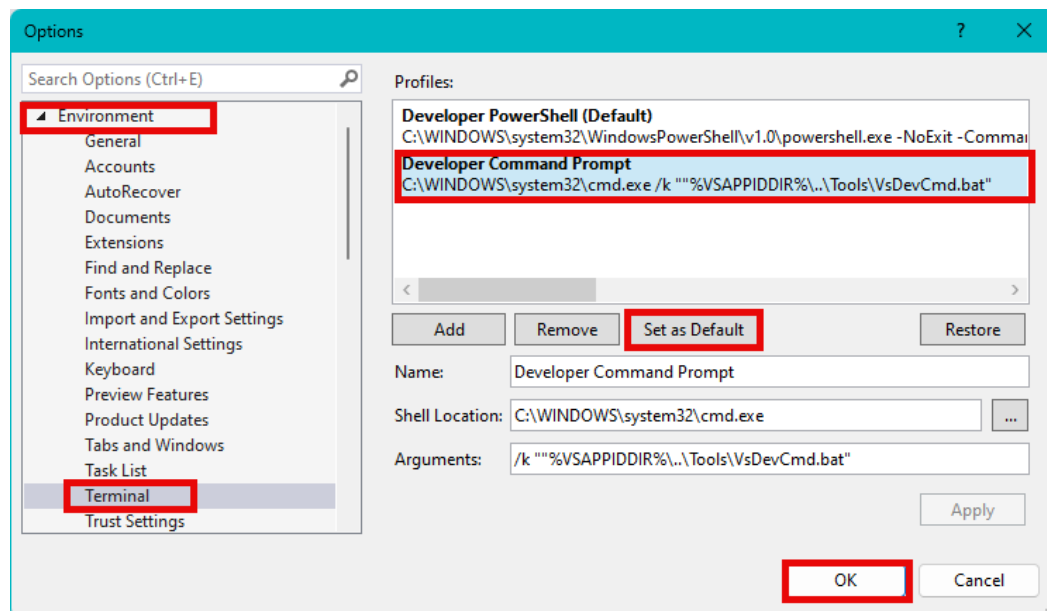
5.3 Via Visual Studio

Kies in Visual Studio: **Tools** > **Options...**



Maak je keuze in het **Options** Window:

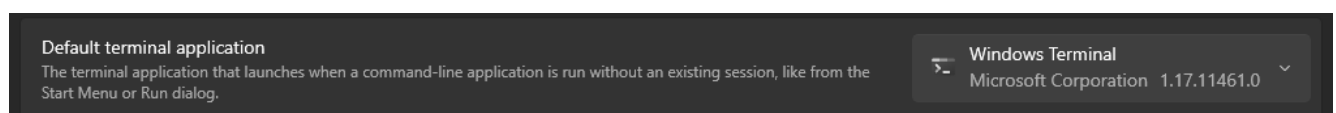
- Kies de rubriek **Environment** > **Terminal**
- Kies de gewenste Console App uit de lijst
- Klik op **Set as Default**
- Klik **OK**

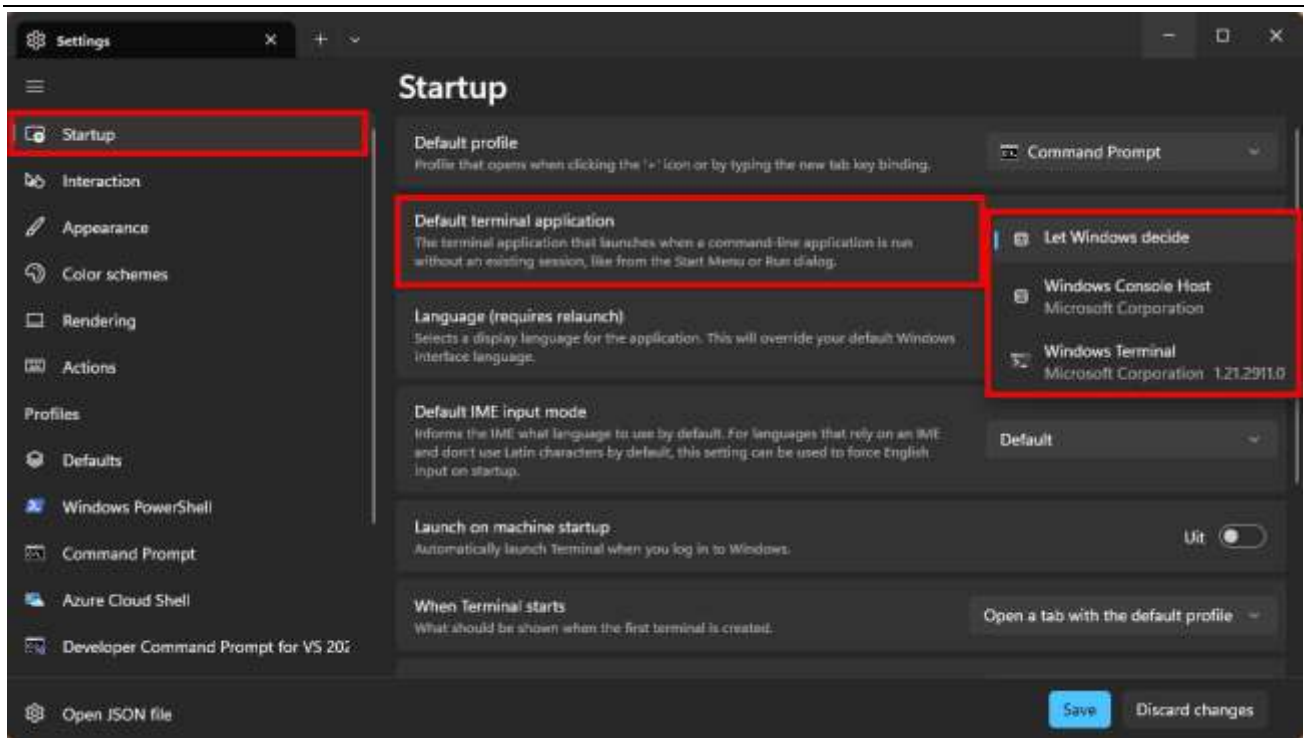


5.4 Via Windows Terminal

Start de **Windows terminal** app op.

Windows Terminal App > Settings > Startup > Default terminal application: Windows Terminal





6 COLOFON

Medewerkers: Johan Vandaele

Versie: 1-jan-2025
