

C# Advanced Meerdere vensters & WPF vervolg

Koen Bloemen



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be





2. Werken met meerdere vensters

6.1 TabControl

6.2 StatusBar

6.3 PasswordBox



2. Werken met meerdere vensters

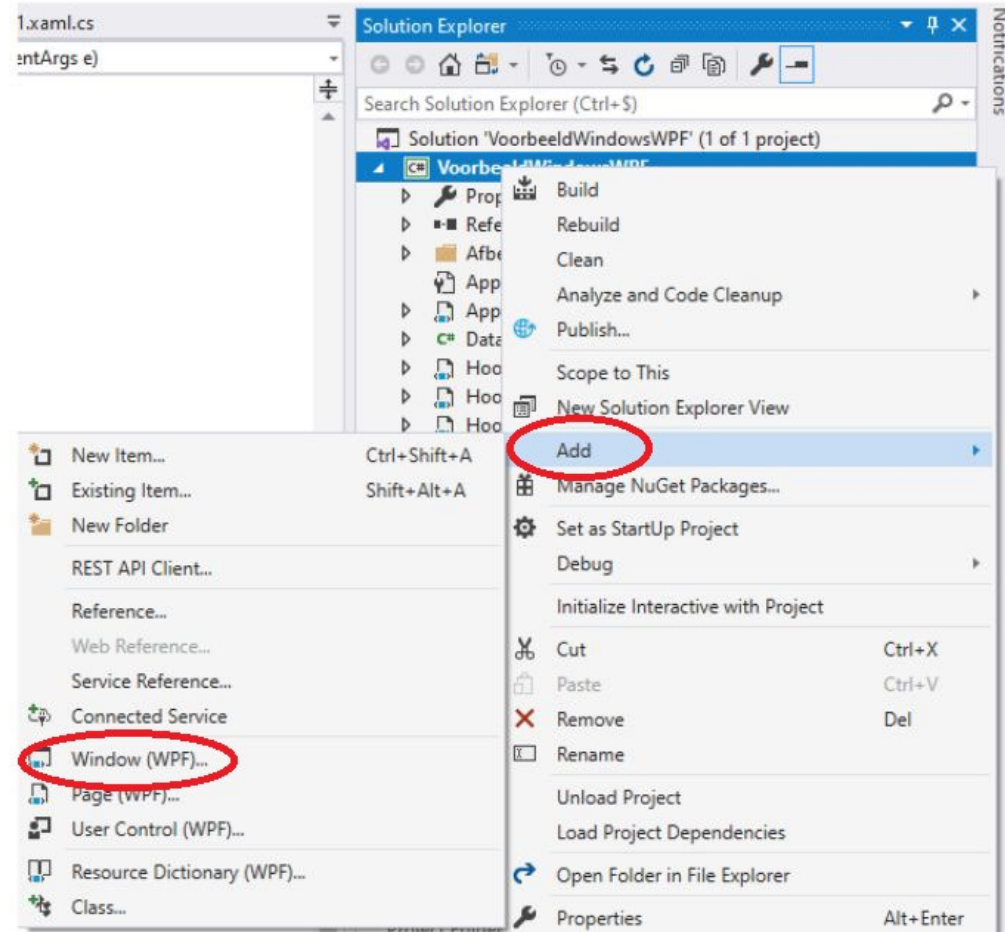
6.1 TabControl

6.2 StatusBar

6.3 PasswordBox

Nieuw window aanmaken

- Rechterklik op je project (niet de Solution!)
- Add > Window (WPF)...



Vershil modeless/modaal Window

Stel ik heb 2 vensters met elk 2 knoppen op

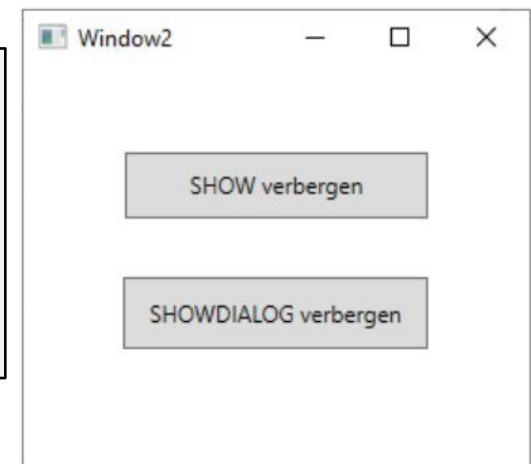
- MainWindow.xaml

```
<Grid>
<Button x:Name="BtnShow" Content="Show" HorizontalAlignment="Left"
    Margin="53,48,0,0" VerticalAlignment="Top" Width="75" Click="BtnShow_Click"/>
<Button x:Name="BtnShowDialog" Content="ShowDialog" HorizontalAlignment="Left"
    Margin="53,71,0,0" VerticalAlignment="Top" Width="75"
    Click="BtnShowDialog_Click"/>
</Grid>
```



- Window2.xaml

```
<Grid>
<Button x:Name="BtnShowHide" Content="SHOW verbergen" HorizontalAlignment="Left"
    Margin="30,25,0,0" VerticalAlignment="Top" Width="168"
    Click="BtnShowHide_Click"/>
<Button x:Name="BtnShowDialogHide" Content="SHOWDIALOG verbergen"
    HorizontalAlignment="Left" Margin="30,48,0,0" VerticalAlignment="Top"
    Width="168" Click="BtnShowDialogHide_Click"/>
</Grid>
```



Venster openen

- MainWindow.xaml.cs

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }

    private void BtnShow_Click(object sender, RoutedEventArgs e)
    {
        Window2 w2 = new Window2();
        w2.Show();
    }
}
```


Vershil modeless/modaal Window

- MainWindow.xaml.cs

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }
    // Modeless venster tonen: je kan altijd wisselen tussen vensters
    private void BtnShow_Click(object sender, RoutedEventArgs e)
    {
        Window2 w2 = new Window2();
        w2.Show();
    }
    // Modaal venster tonen (dialogvenster)
    // Je moet nieuwe venster eerst sluiten voordat je naar het vorige venster terug kan.
    private void BtnShowDialog_Click(object sender, RoutedEventArgs e)
    {
        Window2 w2 = new Window2();
        w2.ShowDialog();
    }
}
```

Vershil modeless/modaal Window

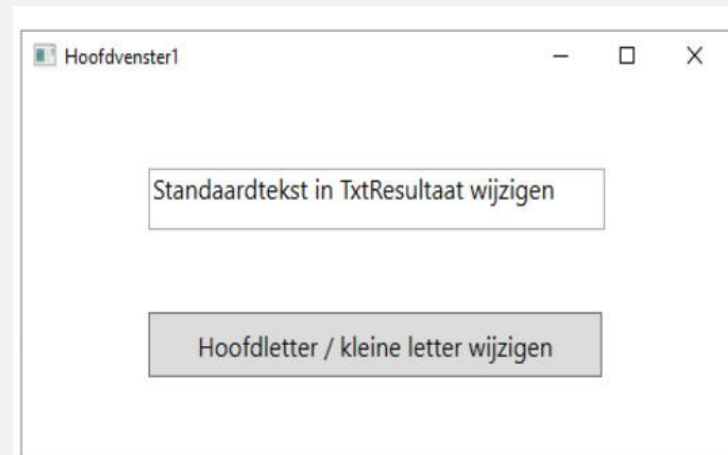
- Window2.xaml.cs

```
public partial class Window2 : Window
{
    public Window2()
    {
        InitializeComponent();
    }
    private void BtnShowHide_Click(object sender, RoutedEventArgs e)
    {
        // Verberg huidig venster
        // Terug tonen via Show();
        Hide();
        // Sluit huidig venster
        Close();
    }
    private void BtnShowDialogHide_Click(object sender, RoutedEventArgs e)
    {
        Hide();
        Close();
    }
}
```


Data uitwisselen tussen verschillende Windows

Manier 1: Doorgeven via constructor

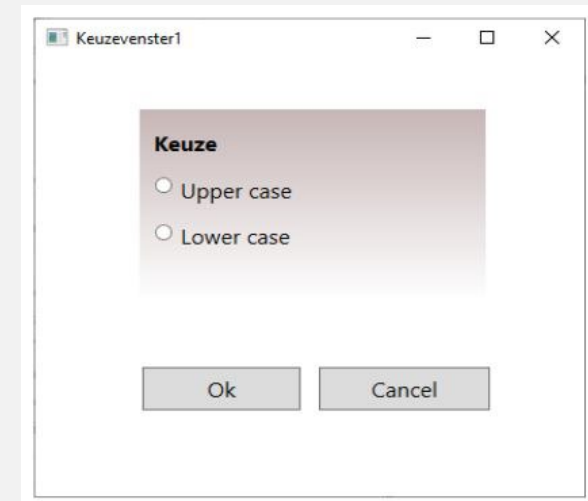
```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }
    private void BtnWijzigen_Click(object sender, RoutedEventArgs e)
    {
        KeuzeWindow keuzeWindow = new KeuzeWindow(TxtResultaat); // geef TextBox van MainWindow door aan KeuzeWindow
        // Modaal venster oproepen
        keuzeWindow.ShowDialog();
        // Opvangen op welke knop bij het KeuzeWindow geklikt is.
        if (keuzeWindow.DialogResult == true)
        {
            MessageBox.Show("Knop Ok gedrukt.");
        }
        else
        {
            MessageBox.Show("Knop Cancel/Sluiten gedrukt.");
        }
    }
}
```



Data uitwisselen tussen verschillende Windows

Manier 1: Doorgeven via constructor

```
public partial class KeuzeWindow : Window
{
    private TextBox tb;
    public KeuzeWindow(TextBox txt)
    {
        InitializeComponent();
        tb = txt; // De 2 objecten worden aan elkaar gekoppeld.
    }
    private void BtnOk_Click(object sender, RoutedEventArgs e)
    {
        if (RadUpper.IsChecked == true)
            tb.Text = tb.Text.ToUpper();
        else
            tb.Text = tb.Text.ToLower();
        DialogResult = true; // Doorgeven op OK geklikt.
    }
    private void BtnCancel_Click(object sender, RoutedEventArgs e)
    {
        DialogResult = false; // Doorgeven op CANCEL geklikt.
    }
}
```

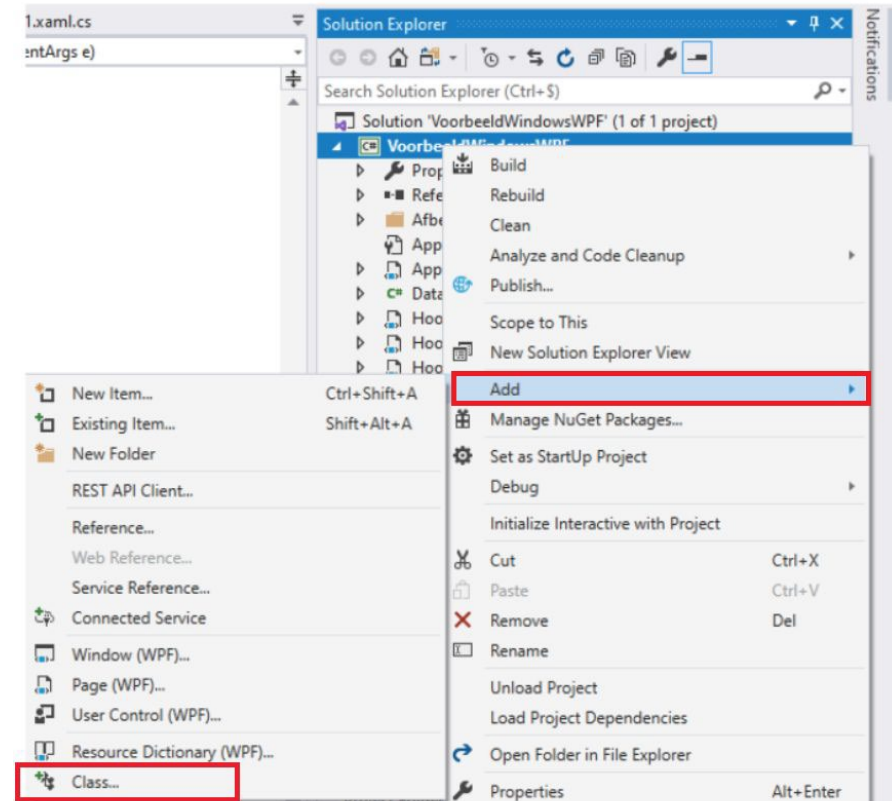


Data uitwisselen tussen verschillende Windows

Manier 2: Opslaan als static variabele in static class en daarna terug uitlezen

- Rechterklik op je project (niet de Solution!)
- Add > Class...
- Maak een nieuwe class aan (bijvoorbeeld Data.cs)

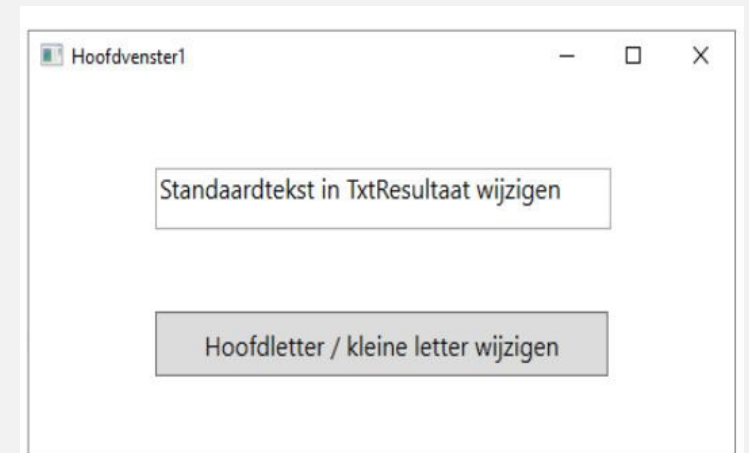
```
public static class Data
{
    public static string Tekst = string.Empty;
}
```



Data uitwisselen tussen verschillende Windows

Manier 2: Opslaan als static variabele in static class en daarna terug uitlezen

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
        //Inhoud van TxtResultaat toekennen aan klasse Data.
        Data.Tekst = TxtResultaat.Text;
    }
    private void BtnWijzigen_Click(object sender, RoutedEventArgs e)
    {
        KeuzeWindow keuzeWindow = new KeuzeWindow();
        keuzeWindow.Show();
    }
    private void Window_Activated(object sender, EventArgs e)
    {
        //MainWindow opnieuw actief => terug kopiëren
        TxtResultaat.Text = Data.Tekst;
    }
}
```



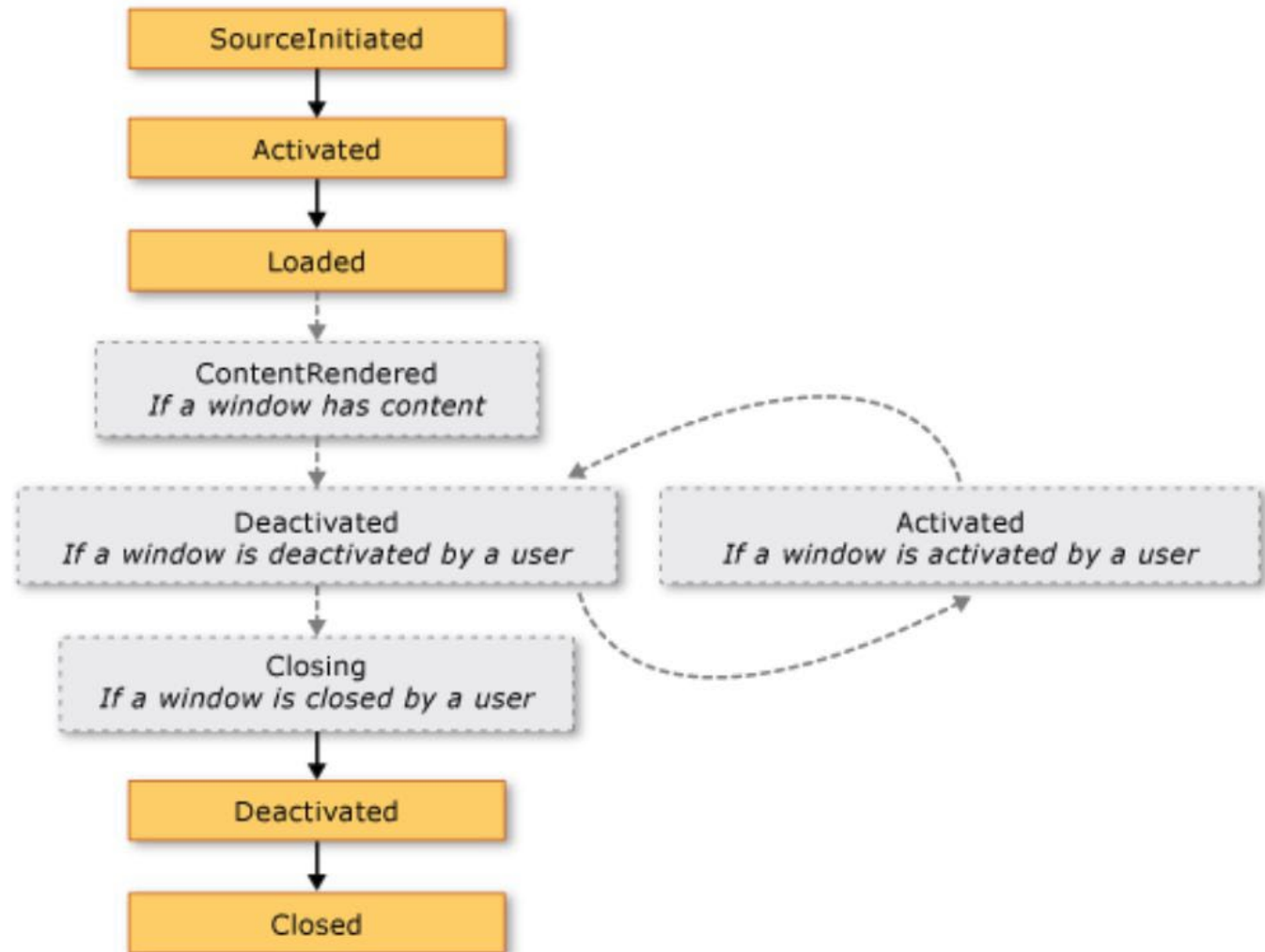
Data uitwisselen tussen verschillende Windows

Manier 2: Opslaan als static variabele in static class en daarna terug uitlezen

```
public partial class KeuzeWindow : Window
{
    public KeuzeWindow()
    {
        InitializeComponent();
    }
    private void BtnOk_Click(object sender, RoutedEventArgs e)
    {
        if (RadUpper.IsChecked == true)
            Data.Tekst = Data.Tekst.ToUpper(); // Data.Tekst veranderen
        else
            Data.Tekst = Data.Tekst.ToLower(); // Data.Tekst veranderen
            // KeuzeWindow verbergen zodat MainWindow terug actief wordt (Window_Activated).
            // Daarna Data.Tekst in de MainWindow weer uitlezen
            Hide();
    }
    private void BtnCancel_Click(object sender, RoutedEventArgs e)
    {
        Close();
    }
}
```

Levensduur van een Window

Van boven naar beneden



Levensduur van een Window

Volgorde bij het openen van Window

- Window_Loaded event

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Window wordt geladen.");
}
```

- InitializeComponent() die in de constructor staat.
 - Doorloopt de XAML van het Window.

```
public MainWindow()
{
    InitializeComponent(); // Window opbouwen uit XAML
    // hieronder kan je private/public variabelen initialiseren
}
```

Levensduur van een Window

Volgorde bij het sluiten van Window

- Window_Closing event
 - Window gaat sluiten

```
private void Window_Closing(object sender, System.ComponentModel.CancelEventArgs e)
{
    MessageBox.Show("Window gaat sluiten.");
}
```

- Window_Closed event
 - Window is gesloten

```
private void Window_Closed(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Window is gesloten.");
}
```

Levensduur van een Window

- `Close()`
 - Huidig venster sluiten => triggered een `Closing()` event
- `Application.Current.Shutdown()`
 - Sluit de hele applicatie => triggered een `Closing()` event
- `Environment.Exit(0)`
 - Sluit de hele applicatie => triggered **GEEN** `Closing()` event

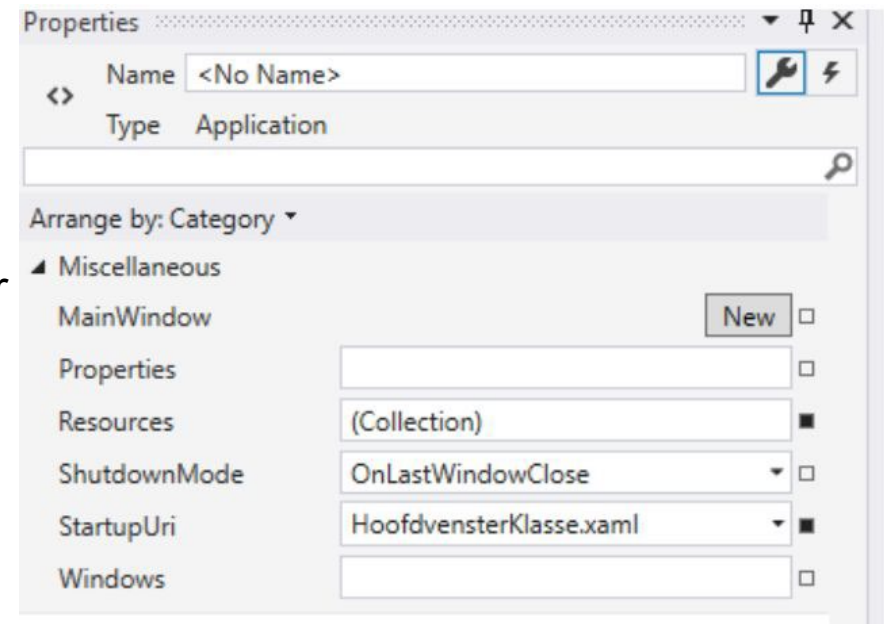
Levensduur van een Window

Kiezen welk venster wordt geopend bij het starten van de applicatie

- Dubbelklik op App.xaml
 - In de StartupUri zie je welk venster geopend wordt

```
StartupUri="MainWindow.xaml"
```

- Kijk naar de Properties
 - StartupUri toont welk venster geopend wordt
 - ShutdownMode bepaalt wanneer de applicatie volledig sluit
 - OnLastWindowClose: sluiten als laatste venster gesloten wordt
 - OnMainWindowClose: sluiten als hoofdvenster gesloten wordt
 - OnExplicitShutdown: enkel sluiten bij Close() of Application.Current.Shutdown()





2. Werken met meerdere vensters

6.1 TabControl

6.2 StatusBar

6.3 PasswordBox

TabControl

- Tabbladen (zoals in webbrowsers).
- Handig om veel info op 1 Window te krijgen en te verdelen.
- Wisselen tussen tabbladen: Ctrl+Tab
- Properties:
 - Items: bevat de tabbladen
 - TabStripPlacement: waar komen tabknoppen (Left, Right, Top, Bottom).
- Methods:
 - Focus zetten op 1ste tabblad: `tabMijnTabblad.SelectedIndex = 0; // index -1: geen tabblad`
- XAML van TabControl:
 - TabControl bestaat uit TabItems (tabbladen) met een Header (titel).
 - In elk TabItem kan je een Grid, DockPanel, StackPanel,... zetten.
 - HorizontalAlignment="Stretch" en VerticalAlignment="Stretch"
 - Tabblad zowel horizontaal als verticaal uitrekken bij resizen van Window

TabControl

```
<TabControl HorizontalAlignment="Stretch" VerticalAlignment="Stretch">
  <TabItem Header="TabItem1">
    <Grid Background="#FFE5E5E5">
      <TextBlock Text="Dit is het eerste tabblad. Alle controls worden binnen het Grid
        ontworpen." />
    </Grid>
  </TabItem>
  <TabItem Header="TabItem2">
    <Grid Background="#FFE5E5E5">
      <TextBlock Text="Dit is het tweede tabblad." />
    </Grid>
  </TabItem>
</TabControl>
```



TabControl

- Binnen de Header kan je ook Images of andere controls zetten
- Gebruik hiervoor een StackPanel

```
<TabControl HorizontalAlignment="Stretch" VerticalAlignment="Stretch">
  <TabItem>
    <TabItem.Header>
      <StackPanel Orientation="Horizontal">
        <Image Source="Afbeeldingen/briefcase.png" />
        <TextBlock Text=" Tabblad 1"/>
      </StackPanel>
    </TabItem.Header>
    <Grid Background="#FFE5E5">
      <TextBlock Text="Dit is het eerste tabblad. Alle controls worden binnen het Grid
      ontworpen."/>
    </Grid>
  </TabItem>
  <!-- ANALOOG VOOR HET TWEDE TABITEM -->
</TabControl>
```





2. Werken met meerdere vensters

6.1 TabControl

6.2 StatusBar

6.3 PasswordBox

TabControl

- StatusBar = statusbalk \Rightarrow status weergeven van toepassing
- Kan je onderverdelen in verschillende controls
- Staat meestal onderaan Window

```
<StatusBar DockPanel.Dock="Bottom" Height="30" VerticalAlignment="Top">  
  <TextBlock Text=" Ready" FontStyle="Italic" FontSize="16"/>  
  <Image Source="Afbeeldingen/CSharpLogo.png" Margin="10,0,0,0"/>  
  <Image Source="Afbeeldingen/FileFromWeb_6281.png" Margin="5,0,0,0"/>  
  <TextBlock x:Name="TextBlockDatumTijd" Text="Datum en tijd" Margin="400,0,0,0"/>  
</StatusBar>
```

Ready C# 

vrijdag 24 januari 2020 11:42:51



2. Werken met meerdere vensters

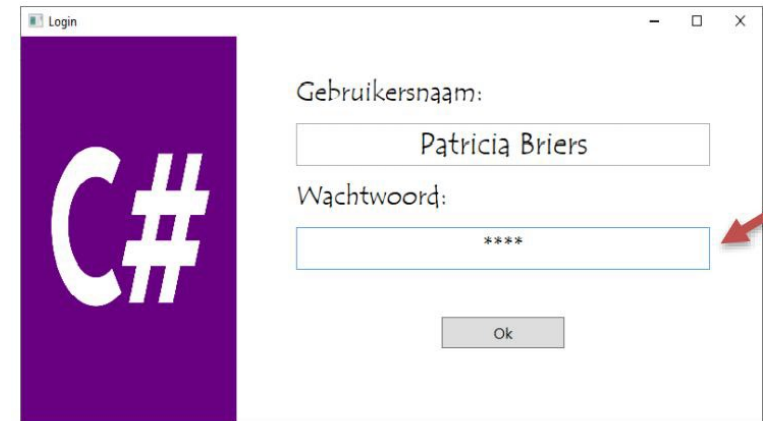
6.1 TabControl

6.2 StatusBar

6.3 PasswordBox

PasswordBox

- PasswordBox = TextBox waar asterisken (*) verschijnen als je tekst ingeeft.
- Voor ingeven van paswoorden.
- Properties:
 - PasswordChar: karakter dat verschijnt (vaak *)
 - Password: de eigenlijke ingevoerde tekst
 - MaxLength: maximale toegestane lengte



```
<PasswordBox x:Name="PwdBoxLogin" MaxLength="4" PasswordChar="*" />
```