

C# Essentials


Eventprocedures en parameters

Lector: Tom Quareme

Wat voorafging...

- Programmeren en probleemoplossend denken
- Visual Studio (2019): C# en WPF
- Datatypes, variabelen en constanten
- Conversies
- Operatoren
- Controlestructuren:
 - Selectie
 - Iteratie
- Hulpklassen: System.Math, System.String, System.Text.StringBuilder,...
- Methods en parameters
 - Functieprocedures
 - Voidprocedures
 - ???

Eventprocedures en parameters

- Handelen een **event** (gebeurtenis) af.
- Observer pattern
 - Event: vb/ KeyDown(), KeyUp(), Click(), MouseEnter(), MouseDown(), MouseUp(),...
 - Publisher/sender: (vb. BtnKlik) 
 - Subscriber(s): (vb. MainWindow)
 - Subscriben (+=), unsubscriben (-=)
 - Krijgt notificatie.
 - Bevat een **eventprocedure** die wordt uitgevoerd na de notificatie.

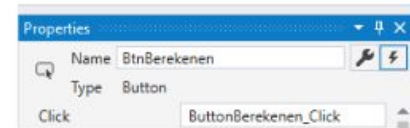


Even herhalen

- **Event aanmaken**

Je kan de gebeurtenis (vb. Click) op 3 verschillende manieren aan je object koppelen:

- Dubbel klik op button en je krijgt het standaardevent. Vb Click bij Button, TextChanged bij TextBox,...
- Selecteer in het Property-venster de juiste gebeurtenis.
- Typ Click in de XAML-tag van het object (XAML venster) en kies vervolgens op <New Event Handler>



Eventprocedures en parameters

- **Event signatuur**

```
private void MyButton_Click(object sender, EventArgs e)
{
    Button myButton = sender as Button;
    myButton.Content = "Clicked!";
}
```

MyButton_Click(object sender)

MyButton_Click()

- **sender**

- Heeft als type **object**.
- Referentie naar object of control die het event oproept (Button, CheckBox).
- Kan je gebruiken om eigenschappen van de control te tonen of aan te passen.

- **e**

- Is de event data
- Click event: **EventArgs** of **RoutedEventArgs** als type.
- Key event: **KeyEventArgs** als type.

Live (probeer ook zelf uit):
Click(),
MouseEnter(),
MouseLeave()

Eventprocedures en parameters

```
private void MyTextBox_KeyDown(object sender, KeyEventArgs e)
{
    if (e.Key == Key.Return)
    {
        ResultLabel.Content = "Enter";
    }
    else if (e.Key == Key.T)
    {
        ResultLabel.Content = "t of T";
    }
    else if (e.Key >= Key.NumPad0 && e.Key <= Key.NumPad9)
    {
        ResultLabel.Content = "Numeriek";
    }
    else if (Keyboard.IsKeyDown(Key.LeftShift) || Keyboard.IsKeyDown(Key.RightShift))
    {
        ResultLabel.Content = "Shift";
    }
    else
    {
        e.Handled = true;
    }
}
```

Eventprocedures en parameters

```
private void MyTextBox_KeyDown(object sender, KeyEventArgs e)
{
    if ((e.Key == Key.R) && (e.KeyboardDevice.Modifiers == ModifierKeys.Control))
    {
        ResultLabel.Content = "Ctrl+R";
    }
    else if (e.Key == Key.K && e.KeyboardDevice.Modifiers == ModifierKeys.None)
    {
        ResultLabel.Content = "Kleine k";
    }
    else if (e.Key == Key.K && e.KeyboardDevice.Modifiers == ModifierKeys.Shift)
    {
        ResultLabel.Content = "Grote K";
    }
    else if ((e.Key >= Key.D0 && e.Key <= Key.D9)
        && e.KeyboardDevice.Modifiers == (ModifierKeys.Alt | ModifierKeys.Control))
    {
        ResultLabel.Content = "Alt Gr + cijfer";
    }
    else
    {
        e.Handled = true;
    }
}
```


Eventprocedures en parameters

```
private void MyTextBox_KeyUp(object sender, KeyEventArgs e)
{
    MyTextBox.Text = "";
    if (e.Key == Key.T)
    {
        ResultLabel.Content = "t of T gelost";
    }
}
```

Live (probeer ook zelf uit):
KeyDown(),
KeyUp()

Volgende lessen...

- WPF besturingselementen
 - Window, Label, TextBox, Button, Image, CheckBox, RadioButton,...
 - Panelen: Grid, Canvas, StackPanel, WrapPanel, DockPanel,...
- Array (1D, 2D)
- System.Array functies (Sort, BinarySearch, Reverse, Copy,...)
- System.Collections.Generic:
 - List<T>
 - Dictionary<K, V>
- Exceptions: fouten afhandelen