

WPF Programme mit mehreren Fenstern

Inhalt

Behandelte Themen	1
Im Visual Studio ein weiteres Fenster erzeugen	1
Benutzerführung im Beispielprogramm.....	2
Anzeige des Hauptfensters beim Start des Programms	2
Anzeige des modalen Eingabefensters nach Betätigen von ‚Eingabe‘	2
Eingabe eines Textes durch den Benutzer.....	2
Übernahme des Textes in das Hauptfenster nach Schließen des Eingabefensters.....	2
Beispielprogramm	3
App.xaml.....	3
Hauptfenster.xaml.....	3
Hauptfenster.xaml.cs	4
Eingabefenster.xaml	5
Eingabefenster.xaml.cs.....	5

Behandelte Themen

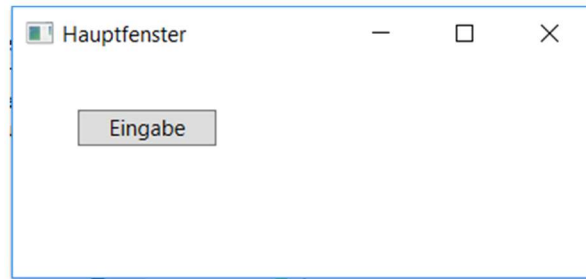
- Wie lege ich bei einem WPF-Programm mit mehreren Fenstern fest, mit welchem Fenster das Programm startet?
- Was ist ein modaler Dialog und wie erzeuge ich ihn?
- Wie greife ich auf die Bedienelemente in einem anderen Fenster zu?

Im Visual Studio ein weiteres Fenster erzeugen

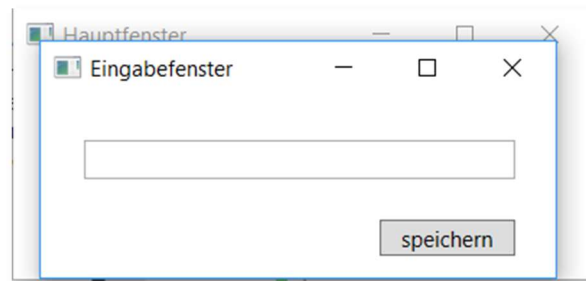
- Klick mit rechter Maustaste auf Projektnamen
- Im erscheinenden Menü auf ‚hinzufügen‘ klicken.
- 'Neues Element' auswählen.
- 'Fenster (WPF)' auswählen.

Benutzerführung im Beispielprogramm

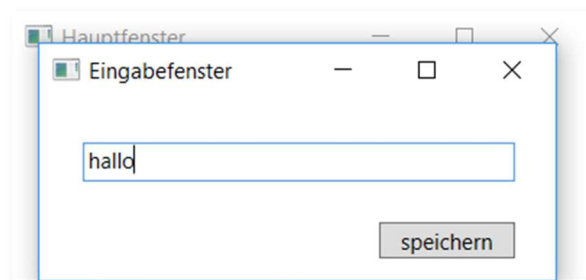
Anzeige des Hauptfensters beim Start des Programms



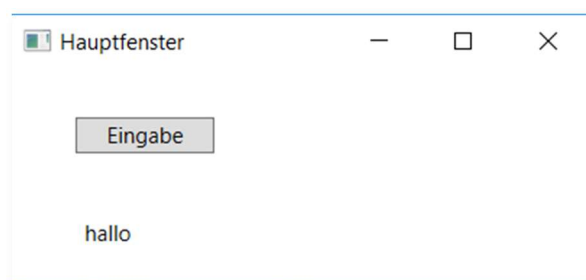
Anzeige des modalen Eingabefensters nach Betätigen von 'Eingabe'



Eingabe eines Textes durch den Benutzer



Übernahme des Textes in das Hauptfenster nach Schließen des Eingabefensters



Beispielprogramm

App.xaml

Mit dem Attribut ‚StartupUri‘ in der Datei ‚App.xaml‘ kann man den Namen der XAML-Datei für das Startfenster festlegen ('Hauptfenster.xaml').

```
<Application x:Class="ZweiFenster.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    StartupUri="Hauptfenster.xaml">
    <Application.Resources>

    </Application.Resources>
</Application>
```

Hauptfenster.xaml

```
<Window x:Class="ZweiFenster.Hauptfenster"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Title="Hauptfenster" Height="157.602" Width="335.672">
    <Grid>
        <Button x:Name="btnEingabe" Content="Eingabe" HorizontalAlignment="Left"
            Margin="36,27,0,0" VerticalAlignment="Top" Width="77"
            Click="btnEingabe_Click"/>
        <Label x:Name="lblAnzeige" Content="" HorizontalAlignment="Left"
            Height="25" Margin="36,78,0,0" VerticalAlignment="Top"
            Width="270"/>
    </Grid>
</Window>
```

Hauptfenster.xaml.cs

- Um das Eingabefenster anzuzeigen, erzeugt man zunächst ein Objekt von der entsprechenden Klasse.
- Aufruf der Methode 'ShowDialog()' zeigt das Eingabefenster an und zwar
 - **modal**, d.h. das Hauptfenster ist nicht mehr zugreifbar, und
 - **blockierend**, d.h. das Programm des Hauptfensters stoppt, bis das Eingabefenster geschlossen wird.
- Über das Objekt des Eingabefensters können die Steuerelemente in Form von Attributen direkt angesprochen werden.

```
using System.Windows;

namespace ZweiFenster
{
    public partial class Hauptfenster : Window
    {
        public Hauptfenster()
        {
            InitializeComponent();
        }

        private void btnEingabe_Click(object sender, RoutedEventArgs e)
        {
            // Objekt des Eingabefensters erzeugen.
            Eingabefenster eingabefenster = new Eingabefenster();
            // Eingabefenster als modalen Dialog aufrufen.
            eingabefenster.ShowDialog();

            // Der weitere Ablauf dieses Programms ist gestoppt,
            // solange der modale Dialog angezeigt wird.

            // Eingegebenen Text abrufen.
            // Die Bedienelemente des anderen Fensters sind öffentlich zugänglich.
            string eingabetext = eingabefenster.tbxEingabe.Text;
            lblAnzeige.Content = eingabetext;
        }
    }
}
```

Eingabefenster.xaml

```
<Window x:Class="ZweiFenster.Eingabefenster"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        mc:Ignorable="d"
        Title="Eingabefenster" Height="138.596" Width="300">
    <Grid>
        <TextBox x:Name="tbxEingabe" HorizontalAlignment="Left" Height="21"
                Margin="24,25,0,0" TextWrapping="Wrap" Text=""
                VerticalAlignment="Top" Width="239"/>
        <Button x:Name="btnSpeichern" Content="speichern"
                HorizontalAlignment="Left" Margin="188,69,0,0"
                VerticalAlignment="Top" Width="75" Click="btnSpeichern_Click"/>
    </Grid>
</Window>
```

Eingabefenster.xaml.cs

Bei Bedarf kann der Konstruktor des zweiten Fensters zusätzlich Parameter entgegennehmen. Mit der Methode 'Close()' wird das Eingabefenster geschlossen und der modale, blockierende Dialog beendet.

```
using System.Windows;

namespace ZweiFenster
{
    public partial class Eingabefenster : Window
    {
        public Eingabefenster()
        {
            InitializeComponent();
        }

        private void btnSpeichern_Click(object sender, RoutedEventArgs e)
        {
            // Dieses Fenster schließen.
            Close();
        }
    }
}
```