

Grundstruktur einer WPF-Anwendung

Inhalt

| | |
|--------------------------------------|---|
| Erstellung einer WPF-Anwendung | 1 |
| Verhalten der Oberfläche | 2 |
| XAML-Datei | 3 |
| code-behind-Datei | 3 |

Anlegen eines WPF-Projekts

Es wird ein Projekt vom Typ 'WPF-Anwendung' (.NET Core) erstellt. Nach Auswahl des Ordners und Eingabe eines Projektnamens wird die Oberfläche vom Visual Studio entsprechend gestaltet.

- In der Mitte oben befindet sich das interaktive Design-Fenster.
- Am linken Rand liegt die Toolbox zur Auswahl der Steuerelemente, die sich im Designer aufziehen und positionieren lassen. Die Toolbox wird durch den Menüpunkt 'Ansicht/Toolbox' angezeigt.
- Am rechten Rand befindet sich das Eigenschaftensfenster zur Auswahl der Eigenschaften einzelner Steuerelemente. Das Eigenschaftens-Fenster wird durch den Menüpunkt 'Ansicht/Eigenschaftensfenster' oder die Taste F4 angezeigt.
- In der Mitte unten befindet sich das Fenster für den XAML-Code. Der XAML-Code beschreibt vollständig das Aussehen des gestalteten Fensters.

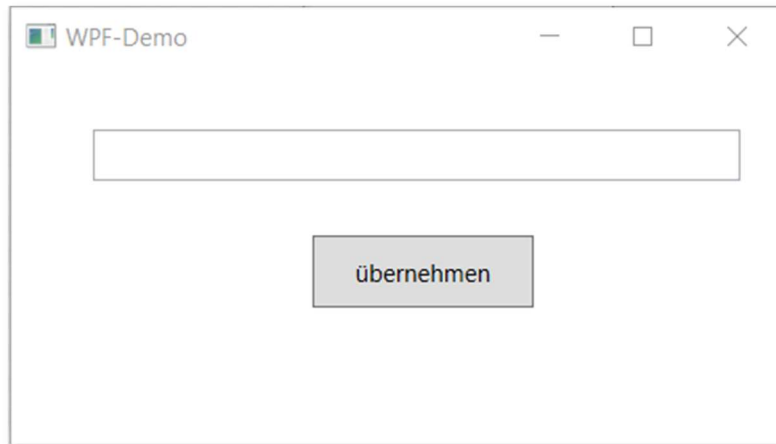
Erstellung einer WPF-Oberfläche

Die verschiedenen Steuerelemente werden in der Toolbox ausgewählt und in der Oberfläche aufgezogen und positioniert. Ausgewählte Steuerelemente können im Eigenschaftensfenster konfiguriert werden. Alternativ können im XAML-Code Beschriftungen verändert (Attribute 'Text' bzw. 'Content') oder den Steuerelementen Namen gegeben werden (Attribut 'Name'). Dabei darf die XML-Grundstruktur nicht verändert werden.

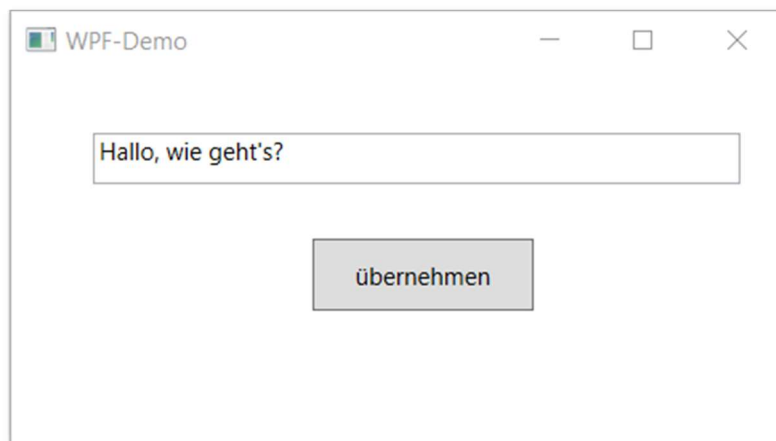
Ein Doppelklick auf ein Steuerelement, welches aktiv auf Ereignisse reagieren soll, legt in der code-behind-Datei einen Methodenrahmen an. Diese Methode wird beim Programmaufbau aufgerufen, wenn z.B. ein Button angeklickt wird. In die Methode wird der Quellcode zur Ereignisverarbeitung eingefügt.

Verhalten der Oberfläche

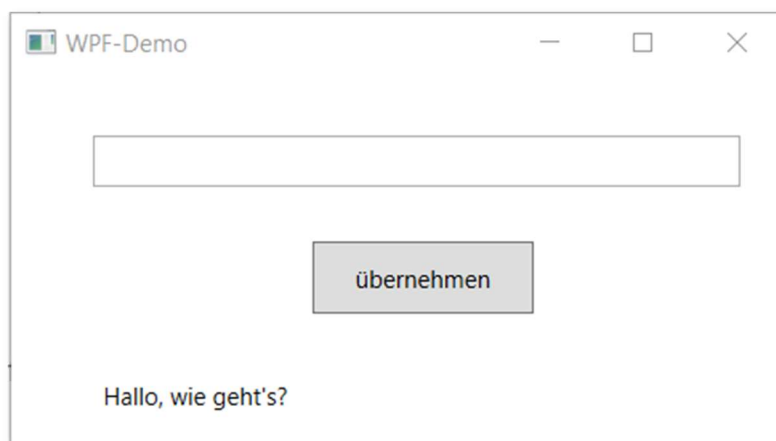
Die Oberfläche enthält eine TextBox zur Eingabe eines Textes, einen Button und ein Label zur Anzeige des Textes.



Der Benutzer kann in das Eingabefeld (TextBox) einen Text eingeben.



Durch Klick auf 'übernehmen' wird der eingegebene Text in das Beschriftungsfeld (Label) verschoben.



XAML-Datei

Das Aussehen der grafischen Oberfläche wird in der Datei 'MainWindow.xaml' beschrieben (XAML: eXtensible Application Markup Language).

Das Element <Window> entspricht dem gesamten Fenster inklusive Rahmen. Die Attribute 'Title', 'Height' und 'Width' erlauben das Festlegen des Titels, der Höhe und der Breite des Fensters.

Das Element <Grid> legt fest, nach welchen Regeln die Steuerelemente im Inneren des Fensters positioniert werden können. Die Steuerelemente können über das Attribut 'Margin' absolut positioniert werden.

Die Elemente <TextBox>, <Button> und <Label> entsprechen den Steuerelementen. Damit sie in der code-behind-Datei angesprochen werden können, erhalten sie einen Namen mit dem Attribut 'x:Name'. In der code-behind-Datei gibt es dann ein Attribut mit demselben Namen und dem Typ des Steuerelements. Das Attribut 'Click' enthält den Namen der Ereignismethode für das Klick-Ereignis.

MainWindow.xaml

```
<Window x:Class="WPFDemo.MainWindow"
  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="WPF-Demo" Height="225.2" Width="400.8">
  <Grid>
    <TextBox x:Name="txtEingabe" HorizontalAlignment="Left" Height="26"
      Margin="41,31,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
      Width="323" />
    <Button x:Name="btnÜbernehmen" Content="übernehmen" HorizontalAlignment="Left"
      Height="36" Margin="150,84,0,0" VerticalAlignment="Top" Width="111"
      Click="BtnÜbernehmen_Click"/>
    <Label x:Name="lblAnzeige" HorizontalAlignment="Left" Height="29"
      Margin="41,148,0,0" VerticalAlignment="Top" Width="323"/>
  </Grid>
</Window>
```

code-behind-Datei

Die Datei 'MainWindow.xaml.cs' enthält den Quellcode, der ausgeführt wird, wenn in der grafischen Oberfläche ein Ereignis ausgelöst wird, z.B. durch eine Benutzergeste.

MainWindow.xaml.cs

```
using System;
using System.Windows;

namespace WPFDemo
{
    //
    // Die Fensterklasse 'MainWindow' ist von der Systemklasse 'Window' abgeleitet.
    // Es handelt sich um eine partielle Klasse ('partial'),
    // weil intern aus der XAML-Datei ein zweiter Teil der Klasse generiert wird,
    // der die Referenzen der Steuerelemente als Attribute enthält.
    // 'MainWindow' enthält den 'Code-Behind' zur grafischen Oberfläche und
    // verarbeitet die empfangenen Benutzergesten als Ereignisse.
    //
    public partial class MainWindow : Window
    {
        //
        // Der Konstruktor der Klasse 'MainWindow' sorgt für die Interpretation
        // der XAML-Datei und für die Darstellung des Fensters.
        // 'InitializeComponent()' muss ausgeführt worden sein, bevor auf die
        // Steuerelement-Objekte zugegriffen werden kann.
        //
        public MainWindow()
        {
            InitializeComponent();
        }

        //
        // Die Methode 'BtnÜbernehmen_Click' wird automatisch aufgerufen,
        // wenn der Benutzer auf den Button klickt.
        // Der Name dieser Ereignismethode entspricht der Voreinstellung
        // und setzt sich aus dem Namen des Steuerelements ('BtnÜbernehmen')
        // und der Art des Ereignisses ('Click') zusammen.
        //
        private void BtnÜbernehmen_Click(object sender, RoutedEventArgs e)
        {
            //
            // Die Referenz 'txtEingabe' verweist auf ein Steuerelement vom Typ
            // TextBox (= Eingabefeld). Für die Benennung der Steuerelemente
            // wird hier die ungarische Notation verwendet.
            // Die Klasse 'TextBox' hat eine les- und schreibbare Property 'Text'
            // für den Zugriff auf den eingegebenen bzw. angezeigten Text
            // (Typ 'string').
            //
            string eingabe = txtEingabe.Text; // Den eingegebenen Text abrufen.
            //
            // Die Referenz 'lblAnzeige' verweist auf ein Objekt der Klasse 'Label'.
            // Die Klasse 'Label' hat eine les- und schreibbare Property 'Content'.
            //
            lblAnzeige.Content = eingabe; // Den eingegebenen Text im Label anzeigen.
            txtEingabe.Text = ""; // Die TextBox zurücksetzen.
        }
    }
}
```