



colourbox.de

## Anzeige von Datenlisten: Listbox und Combobox

### Inhalt

Nutzung von Combobox und Listbox als Auswahlelemente.....	2
Verhalten des Programms.....	3
Kommentierter Quellcode des Programms.....	4
Monat.cs .....	4
MainWindow.xaml.....	5
MainWindow.xaml.cs .....	6

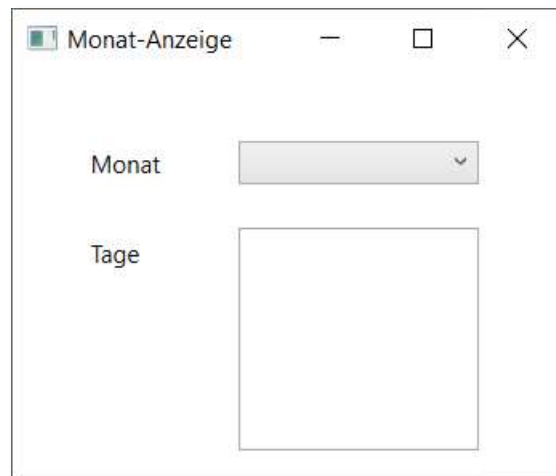
## Nutzung von Combobox und Listbox als Auswahlelemente

- Combobox und Listbox können als Auswahlelemente verwendet werden. Beide Elemente können mit beliebigen Datentypen gefüllt werden.
- Eine Combobox erlaubt eine platzsparende Auswahl von einem Element aus vielen. Zusätzlich kann eine Combobox für eine Texteingabe freigeschaltet werden.
- Eine Listbox erlaubt eine übersichtliche Auswahl von einem oder mehreren Elementen aus vielen.
- Combobox und Listbox können eine interne oder eine externe Datenspeicherung verwenden. Die interne Datenspeicherung geschieht mit der Property "Items", die externe mit "ItemsSource" und einem Array oder einer Liste. Die Verwendung der internen bzw. externen Datenquelle sollte nicht gemischt werden.
- Bei Objekt-Datentypen (z.B. eigenen Klassen) wird die ToString()-Methode zur Formatierung der Einträge aufgerufen. Wird die ToString()-Methode nicht zur Verfügung gestellt, wird eine Standardformatierung verwendet. Bei den Basisdatentypen wird die Standardformatierung verwendet.
- Bei Auswahl eines Eintrags in einer Combobox oder Listbox wird das Ereignis "SelectionChanged" ausgelöst.
- Der ausgewählte Eintrag kann mit der Property "SelectedItem" abgerufen werden. Man erhält den Eintrag als eine Object-Referenz. Diese muss gegebenenfalls in den korrekten Datentyp umgewandelt werden.

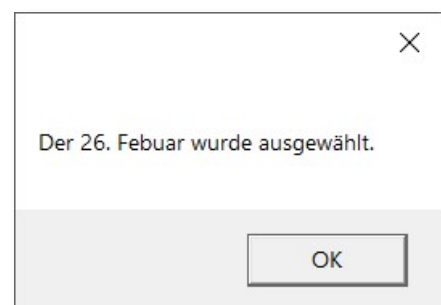
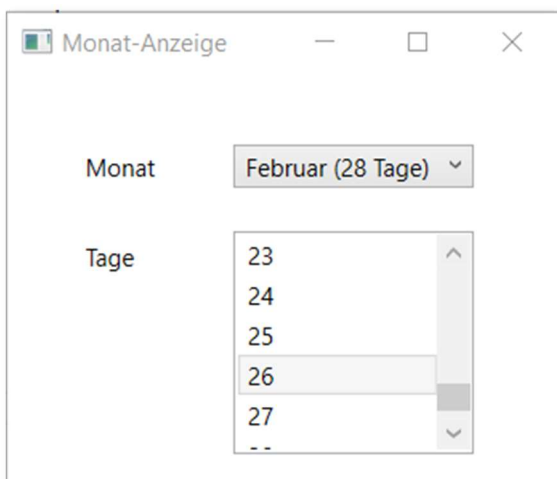
## Verhalten des Programms

Die Oberfläche enthält eine Combobox zur Auswahl eines Monats und eine Listbox zur Anzeige der Tage in dem Monat.

Beim Start des Programms ist die Combobox mit Monat-Objekten gefüllt, aber noch ist kein Monat ausgewählt. Die Listbox ist zunächst leer.



Der Benutzer wählt z.B. den Monat Februar aus und dann den Tag 26. Daraufhin bekommt er in einem Popup-Fenster das entsprechende Datum angezeigt.



## Kommentierter Quellcode des Programms

### Monat.cs

```
using System;

namespace MonatAnzeige
{
    class Monat
    {
        // Kurzschreibweise für Properties.
        // 1. Speicherplatz für Attribute wird angelegt.
        // 2. Der lesende Zugriff ist öffentlich,
        //    der schreibende Zugriff ist nur in dieser Klasse möglich.
        public String Name { get; private set; }
        public int Tage { get; private set; }

        // Konstruktor zur Initialisierung der Properties.
        public Monat(String name, int tage)
        {
            Name = name;
            Tage = tage;
        }

        // Standard-Formatierungsmethode für ein Monat-Objekt.
        // Wird automatisch aufgerufen
        // 1. bei Verwendung in Console.WriteLine(),
        // 2. innerhalb einer String-Verknüpfung,
        // 3. bei einer Anzeige in Combobox oder Listbox.
        public override string ToString()
        {
            return Name + " (" + Tage + " Tage)";
        }
    }
}
```

## MainWindow.xaml

```
<Window x:Class="MonatAnzeige.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="Monat-Anzeige" Height="242.458" Width="290"
        Loaded="Window_Loaded" Closed="Window_Closed">
    <Grid>
        <ComboBox x:Name="cbxMonat" HorizontalAlignment="Left"
                Margin="113,36,0,0" VerticalAlignment="Top"
                Width="120" SelectionChanged="cbxMonat_SelectionChanged"/>
        <Label Content="Monat" HorizontalAlignment="Left" Margin="34,34,0,0"
                VerticalAlignment="Top"/>
        <Label Content="Tage" HorizontalAlignment="Left" Height="26"
                Margin="34,79,0,0" VerticalAlignment="Top" Width="64"/>
        <ListBox x:Name="lbxTag" HorizontalAlignment="Left" Height="111"
                Margin="113,79,0,0" VerticalAlignment="Top" Width="120"
                SelectionChanged="LbxTag_SelectionChanged"/>
    </Grid>
</Window>
```

## MainWindow.xaml.cs

```
using System;
using System.Windows;
using System.Windows.Controls;

namespace MonatAnzeige
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            // Diese Methode wird automatisch aufgerufen,
            // wenn beim Programmstart die grafische Oberfläche
            // geladen wird.
            private void Window_Loaded(object sender, RoutedEventArgs e)
            {
                // Ein Array von Monat-Objekten wird angelegt und gefüllt.
                // Das Array hat die Länge 12.
                Monat[] monate =
                {
                    new Monat("Januar", 31),
                    new Monat("Februar", 28),
                    new Monat("März", 31),
                    new Monat("April", 30),
                    new Monat("Mai", 31),
                    new Monat("Juni", 30),
                    new Monat("Juli", 31),
                    new Monat("August", 31),
                    new Monat("September", 30),
                    new Monat("Oktober", 31),
                    new Monat("November", 30),
                    new Monat("Dezember", 31)
                };
                // In diesem Beispiel erhält die Combobox seine Daten
                // durch eine externe Datenquelle.
                cbxMonat.ItemsSource = monate;
            }
        }
    }
}
```

```
// Diese Methode wird automatisch aufgerufen, wenn der Benutzer
// in der Combobox einen Monat auswählt.
private void cbxMonat_SelectionChanged(object sender,
                                     SelectionChangedEventArgs e)
{
    // Die Property "SelectedItem" liefert eine Object-Referenz.
    // Diese wird in eine Monat-Referenz umgewandelt.
    // Damit kann z.B. auf die Anzahl der Tage zugegriffen werden.
    Monat monat = (Monat)cbxMonat.SelectedItem;

    // In diesem Beispiel wird der interne Datenspeicher
    // der Listbox verwendet.
    // Zunächst werden eventuell zuvor enthaltene Daten gelöscht.
    lbxTag.Items.Clear();
    // Dann werden neue Dateneinträge eingefügt (Datentyp 'int').
    for (int tag = 1; tag <= monat.Tage; tag++)
    {
        lbxTag.Items.Add(tag);
    }
}

// Diese Methode wird automatisch aufgerufen, wenn der Benutzer
// in der Listbox einen Tag auswählt.
// Generell wird die Methode aufgerufen, wenn der Auswahlzustand
// der Listbox sich ändert.
private void LbxTag_SelectionChanged(object sender,
                                     SelectionChangedEventArgs e)
{
    // Ist überhaupt ein Tag ausgewählt worden?
    if(lbxTag.SelectedItem != null)
    {
        Monat monat = (Monat)cbxMonat.SelectedItem;
        int tag = (int)lbxTag.SelectedItem;
        string ausgabe = "Der " + tag + ". " + monat.Name +
                        " wurde ausgewählt.";
        // Eine MessageBox ist ein Popup-Fenster
        // zur Benachrichtigung des Benutzers.
        MessageBox.Show(ausgabe);
    }
}
}
```