08 – Windows Forms Applications

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Agenda

- Einführung in Windows Forms
- Windows Forms Application mit Visual Studio
- Windows Forms Controls
- Eigene User Controls

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Einführung in Windows Forms

- Windows Forms Klassen sind im Namensraum `System.Windows.Forms`

```
using System;
using System.Windows.Forms;
```

- Jede Windows Forms Klasse muss von `System.Windows.Forms.Form` erben

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Einführung in Windows Forms

- Jede Windows Forms Applikation startet mit der Methode `Main()` der Klasse `Program`
- Es können auch Kommandozeilen-Argumente übergeben werden

```
static class Program
{
    [STAThread]
    static void Main(string[] args)
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}
```

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Einführung in Windows Forms

- Windows Forms Applikationen sind ereignisgesteuert (event-driven)
- Sie warten auf Ereignisse (Windows-Messages) in der Methode `Application.Run(…)`

```
static class Program
{
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}
```

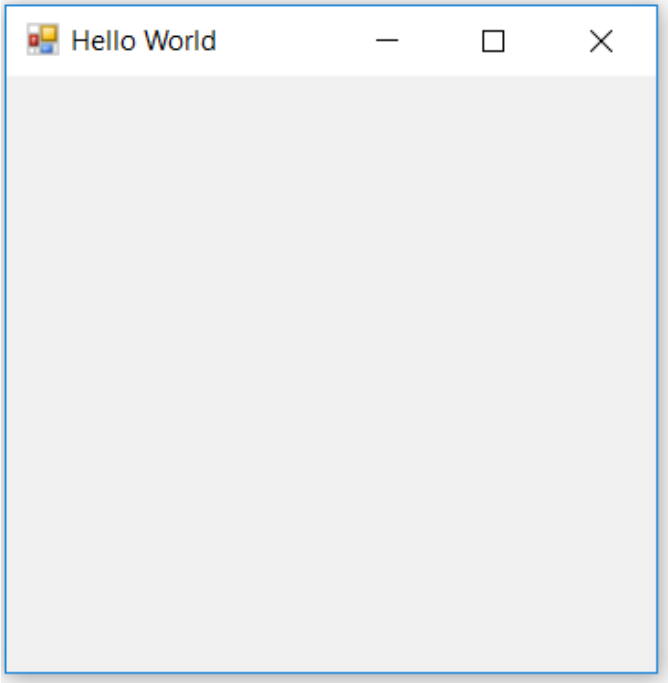
Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Einführung in Windows Forms

- “Hello World” Windows Forms Applikation:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        this.Text = "Hello World";
    }
}
```

Aussehen wird über Eigenschaften gesteuert



Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

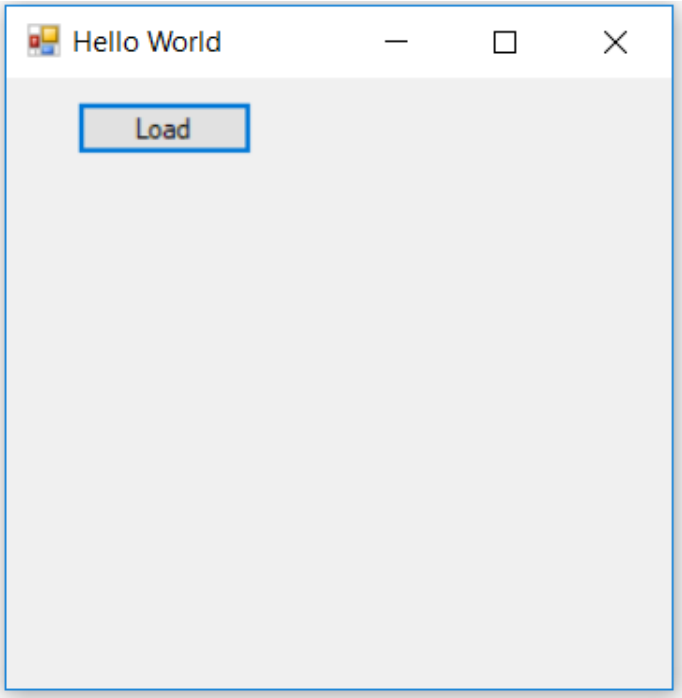
Einführung in Windows Forms

- Elemente wie z.B. ein Knopf werden instanziiert und zu den Controls der Form hinzugefügt

```
public partial class Form1 : Form
{
    Button btnLoad;
    public Form1()
    {
        InitializeComponent();
        this.Text = "Hello World";

        btnLoad = new Button();
        btnLoad.Text = "&Load";
        btnLoad.Left = 30;
        btnLoad.Top = 10;
        this.Controls.Add(btnLoad);
    }
}
```

Knopf deklarieren

Knopf instanziieren und zu den Form.Controls hinzufügen



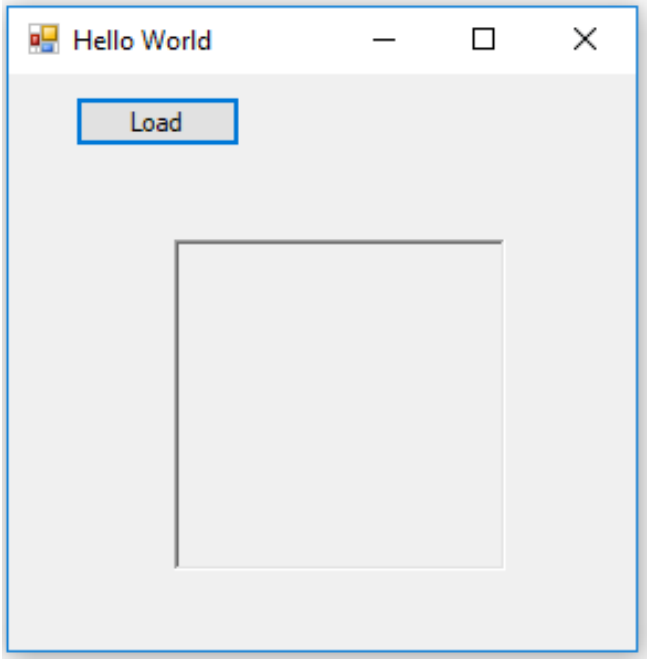
Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Einführung in Windows Forms

Controls hinzufügen: PictureBox

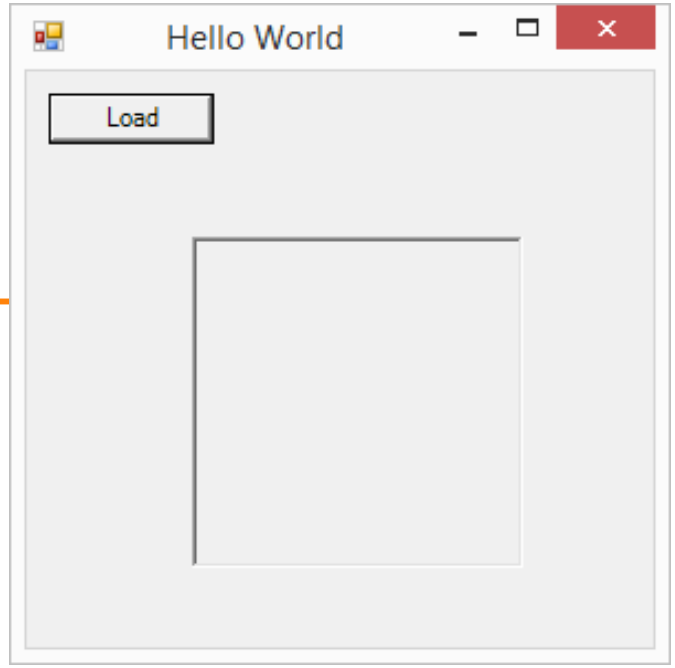
```
public partial class Form1 : Form
{
    Button btnLoad;
    PictureBox pbxShowImage;
    public Form1()
    {
        InitializeComponent();
        this.Text = "Hello World";

        initialize button

        #region initialize picturebox
        pbxShowImage = new PictureBox();
        pbxShowImage.BorderStyle = BorderStyle.Fixed3D;
        pbxShowImage.Width = Width / 2;
        pbxShowImage.Height = Height / 2;
        pbxShowImage.Left = (Width - pbxShowImage.Width) / 2;
        pbxShowImage.Top = (Height - pbxShowImage.Height) / 2;
        this.Controls.Add(pbxShowImage);
        #endregion
    }
}
```

PictureBox deklarieren



PictureBox instanziieren und zu den Form.Controls hinzufügen

Einführung in Windows Forms



Click-Event-handler zum Knopf hinzufügen

```
public Form1()
{
    InitializeComponent();
    this.Text = "Hello World";

    #region initialize button
    btnLoad = new Button();
    btnLoad.Text = "&Load";
    btnLoad.Left = 30;
    btnLoad.Top = 10;
    btnLoad.Click += new EventHandler(btnLoad_Click);
    this.Controls.Add(btnLoad);
    #endregion

    initialize picturebox
}

void btnLoad_Click(object sender, EventArgs e)
{

}
```

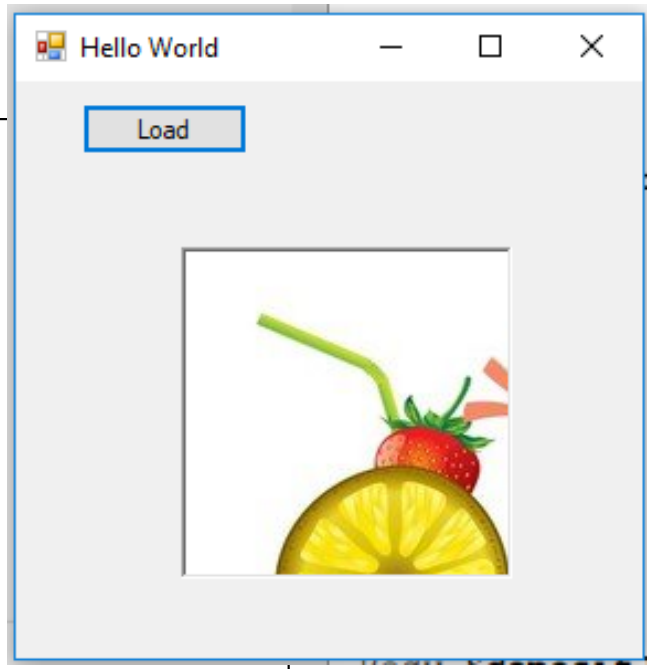
Click Event Handler hinzufügen

Click Event Handler

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Einführung in Windows Forms

Event-handler für den Knopf ausprogrammieren




```
public partial class Form1 : Form
{
    Button btnLoad;
    PictureBox pbxShowImage;
    public Form1()...

    void BtnLoad_Click(object sender, EventArgs e)
    {
        OpenFileDialog dialog = new OpenFileDialog();
        dialog.Title = "Open Image";
        dialog.Filter = "jpg files (*.jpg)|*.jpg|All files (*.*)|*.*";

        if (dialog.ShowDialog() == DialogResult.OK)
        {
            pbxShowImage.Image = new Bitmap(dialog.OpenFile());
        }
        dialog.Dispose();
    }
}
```

Übung 8.1

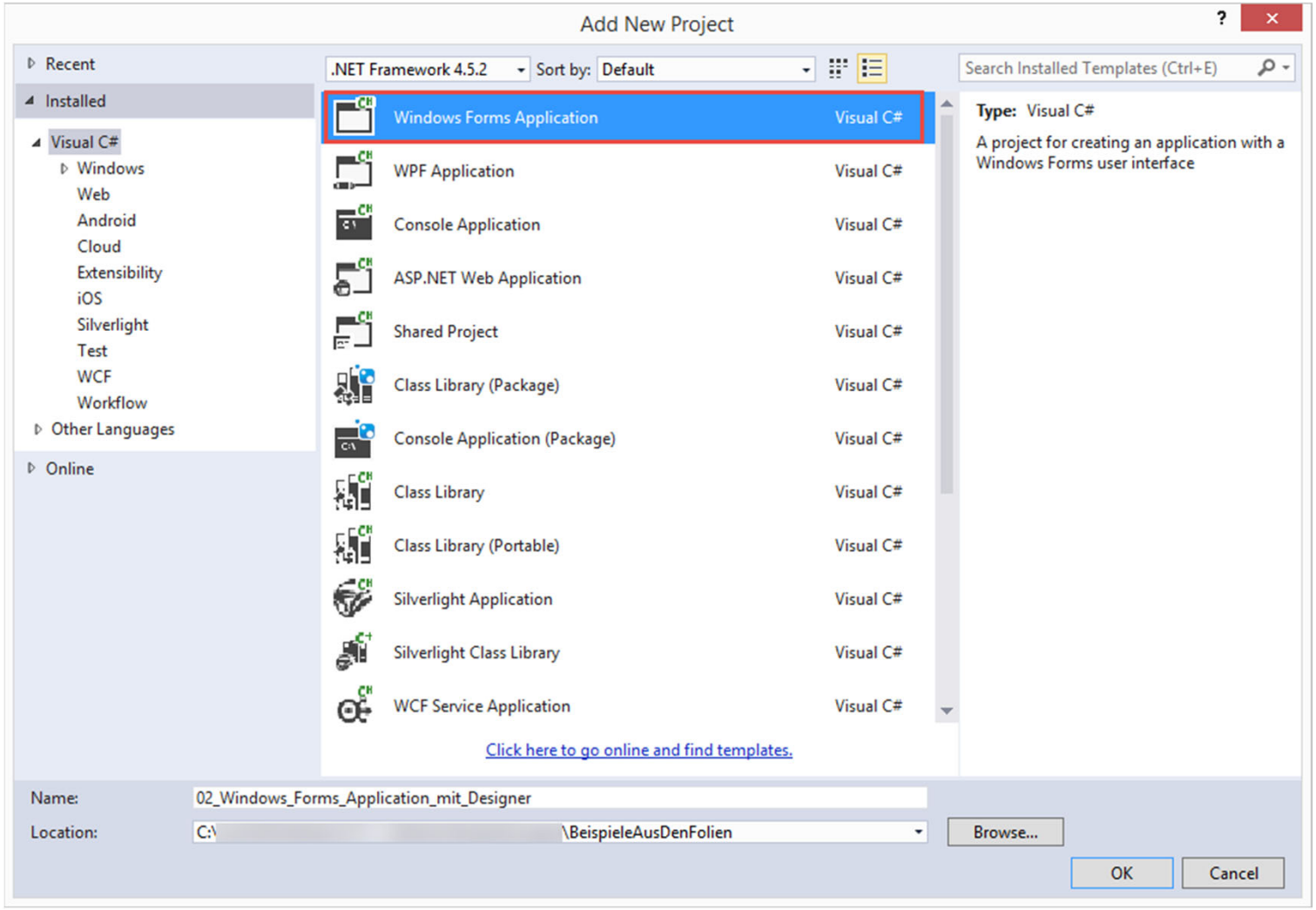
Windows Forms Applikation ohne Designer

- Schreibe eine Windows Forms Applikation, um ein Bild anzuzeigen ohne den Designer von Visual Studio zu benutzen.

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

WinForms mit Visual Studio

- Windows Forms Application Projekt



WinForms mit Visual Studio

- Windows Forms Application Projekteigenschaften

02_Windows_Forms_...tion_mit_Designer* | Form1.cs [Design]

Application
Build
Build Events
Debug
Resources
Services
Settings
Reference Paths
Signing
Security
Publish
Code Analysis

Configuration: N/A Platform: N/A

Assembly name: 02_Windows_Forms_Application_mit_Designer
Default namespace: _02_Windows_Forms_Application_mit_Designer
Target framework: .NET Framework 4.5.2
Output type: Windows Application
Startup object: (Not set)
Assembly Information...

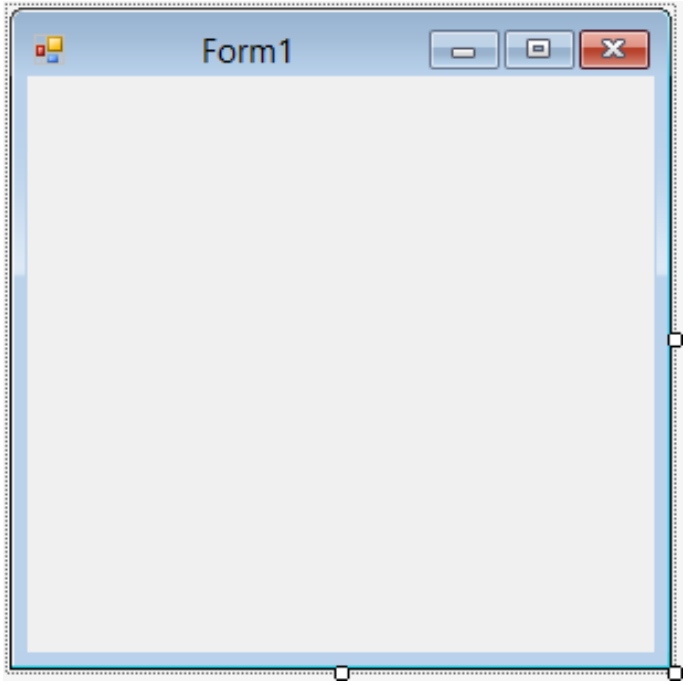
Resources
Specify how application resources will be managed:
Icon and manifest
A manifest determines specific settings for an application. To embed a custom manifest, first add it to your project and then select it from the list below.
Icon: (Default Icon)
Manifest: Embed manifest with default settings
Resource file:

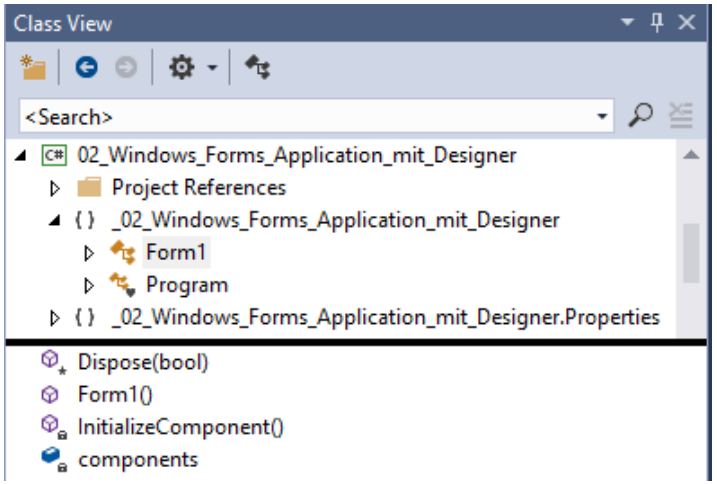
Output type: Windows Application

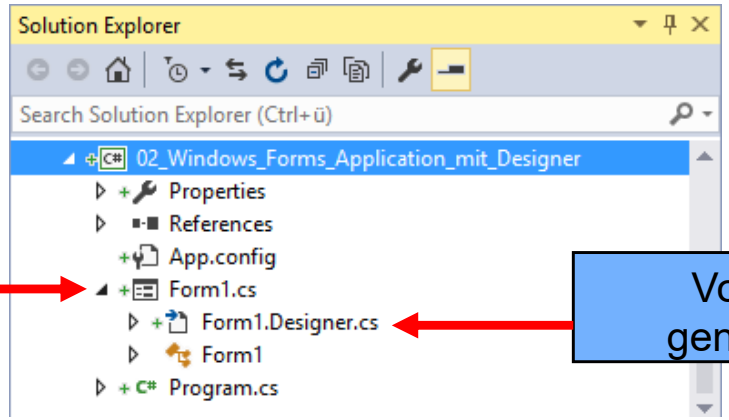
Bildungszentrum Uster Höhere Berufsbildung Uster HBU

Generierter Code (1/5)

Form Designer







Klasse Form

Dispose()

Konstruktor()

Initialize Component()

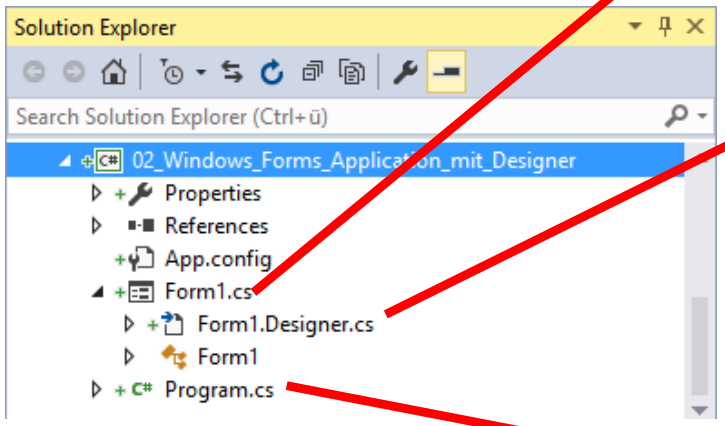
Main()

Components

Eigener Code

Vom Designer generierter Code

Generierter Code (2/5)



Form1.cs
- Class Form1 (partial)
- Form1()

Form1.Designer.cs
- Class Form1 (partial)
- components
- Dispose()
- InitializeComponent()

Program.cs
- Class Program
- Main()

Generierter Code (3/5)

- Hinzugefügte Namenräume:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
```

- Partielle Klasse – 1. Teil:

```
namespace WindowsApplication1
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }
  }
}
```

Bildungszentrum Uster Höhere Berufsbildung Uster HBU

Generierter Code (4/5)

- Partielle Klasse – 2. Teil

```
partial class Form1
{
  private System.ComponentModel.IContainer components = null;

  protected override void Dispose(bool disposing)
  {
    if (disposing && (components != null))
    {
      components.Dispose();
    }
    base.Dispose(disposing);
  }

...
```

Dispose() räumt das Form und alle Controls aus dem Speicher weg

Generierter Code (5/5)

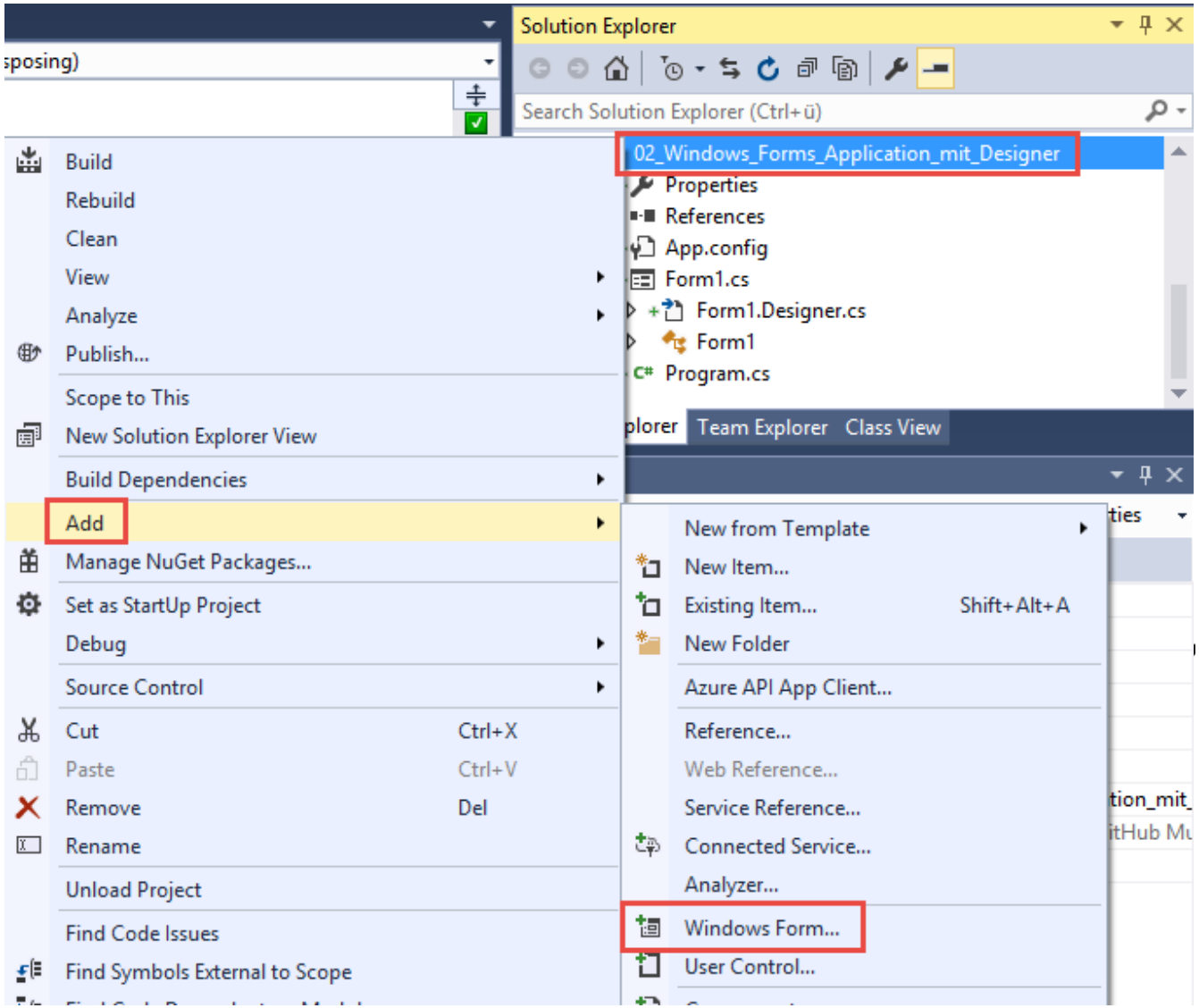
```
  #region Windows Form Designer generated code

  private void InitializeComponent()
  {
    this.SuspendLayout();
    //
    // Form1
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(292, 273);
    this.Name = "Form1";
    this.Text = "Form1";
    this.ResumeLayout(false);
  }

  #endregion
}
```

Hier werden die Komponenten initialisiert

WinForms mit Visual Studio

Ein Form hinzufügen

- Rechtsklick auf Projekt
- Hinzufügen
- Windows Form hinzufügen



WinForms mit Visual Studio

Form-Eigenschaften und -Ereignisse

Properties

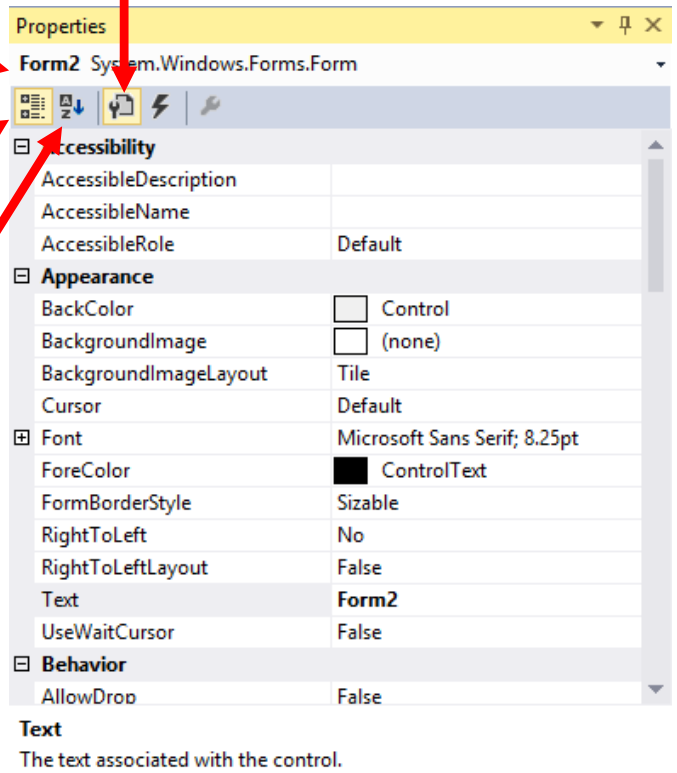
Form Name

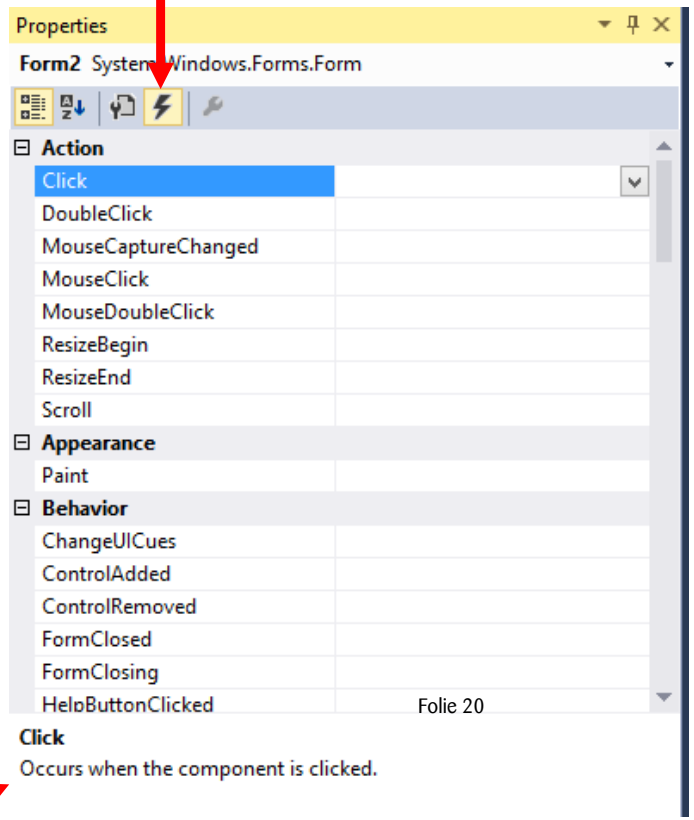
Kategorien

Sortieren

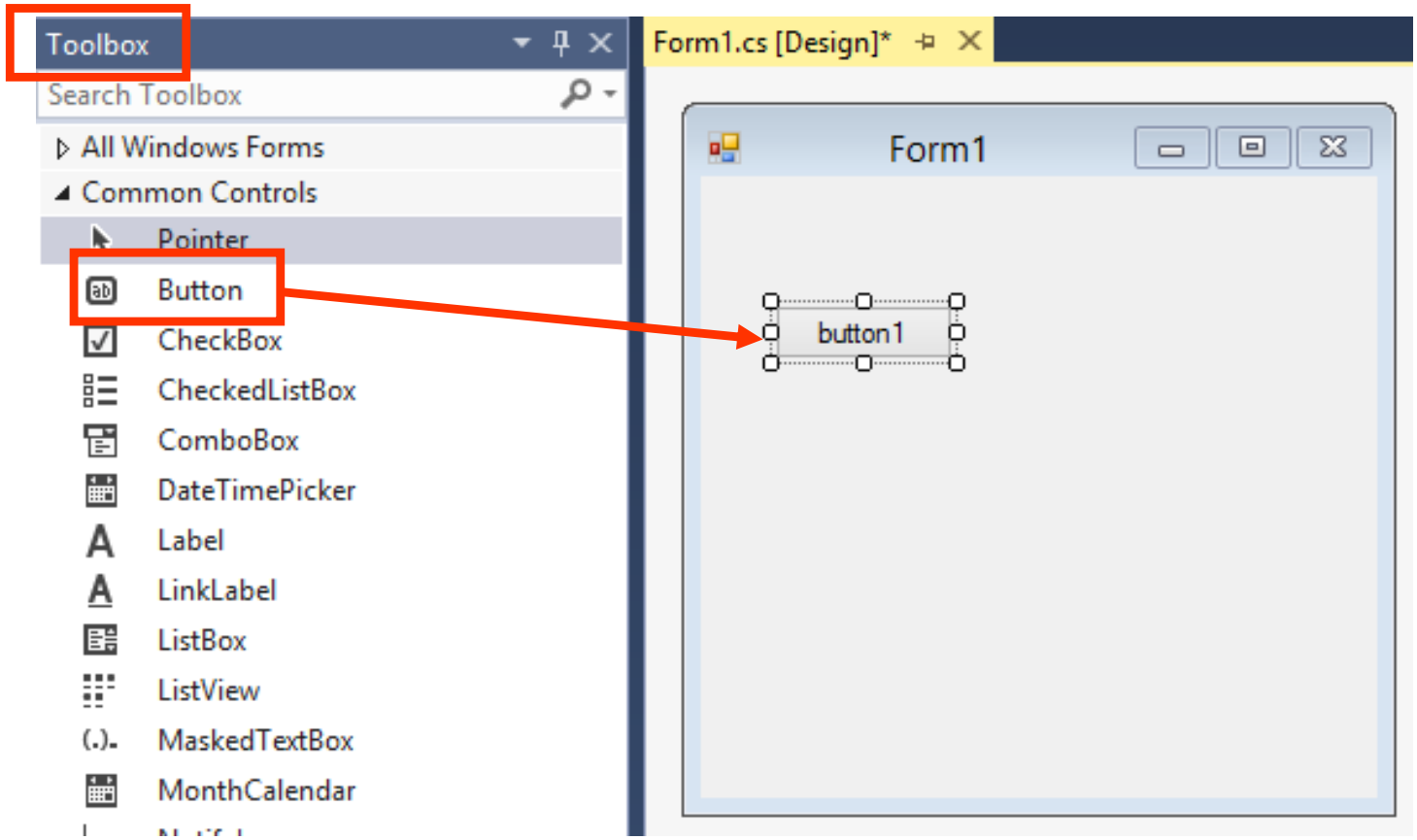
Events

Beschreibung





Controls hinzufügen



Übung 8.2

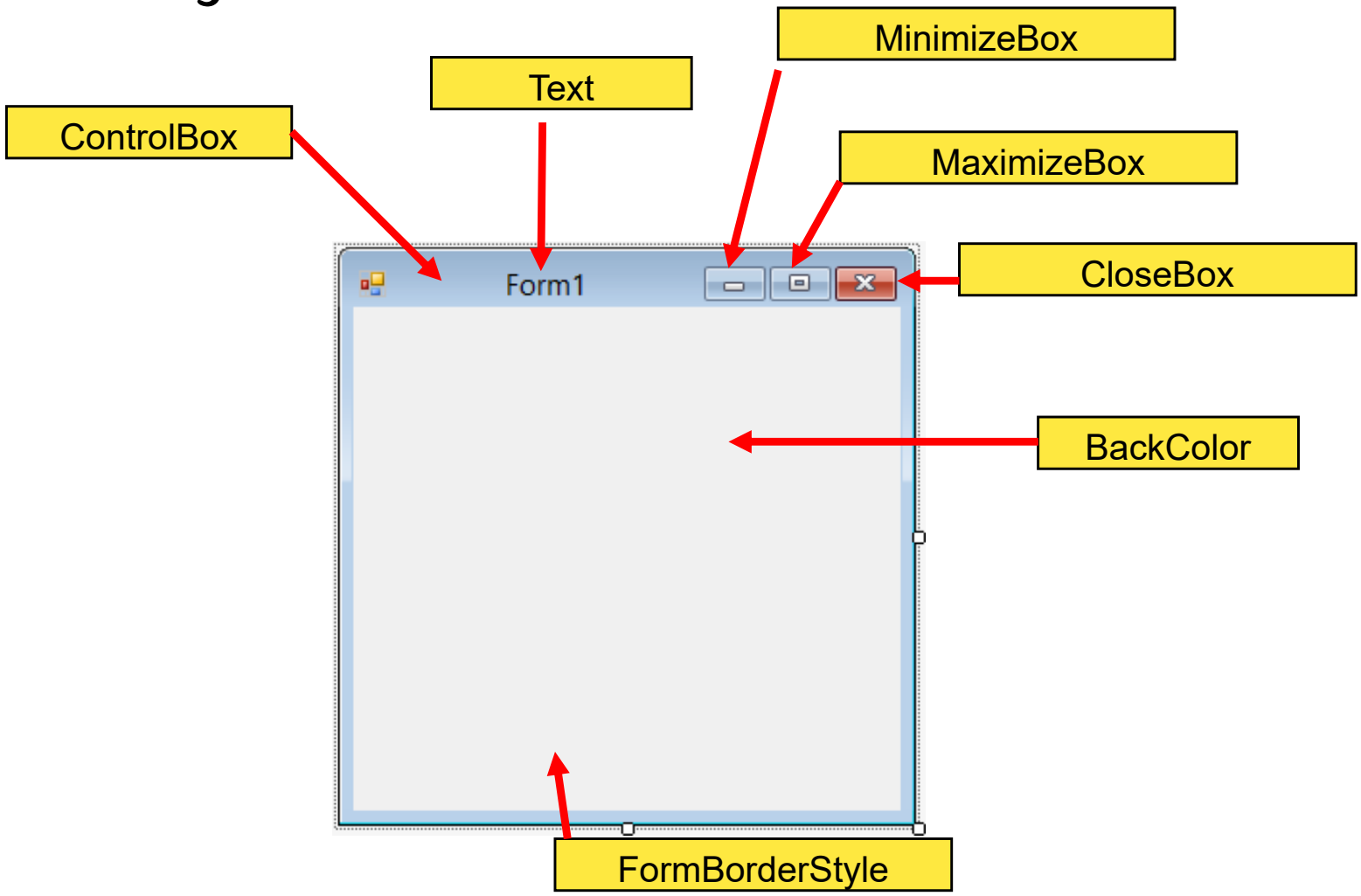
Erste Windows Forms Application mit Visual Studio Form Designer

- Erstelle die gleiche Windows Forms Application von Übung 8.1 mit Hilfe von Visual Studio und dem Form Designer.

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

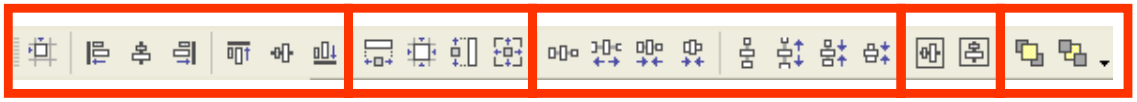
Windows Forms Controls
Forms

Formulareigenschaften

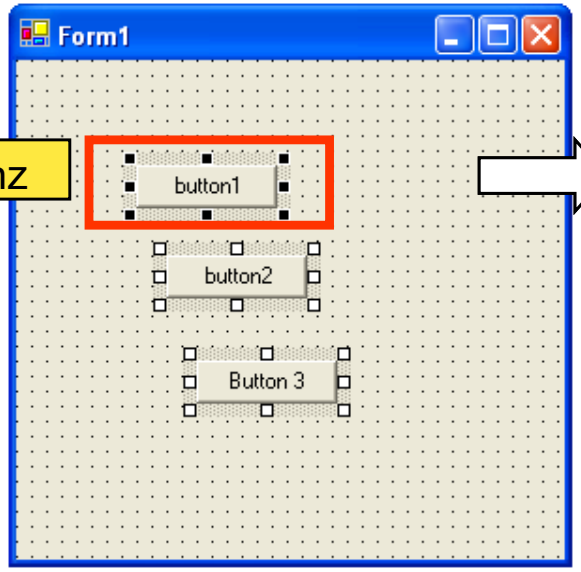


Controls ausrichten

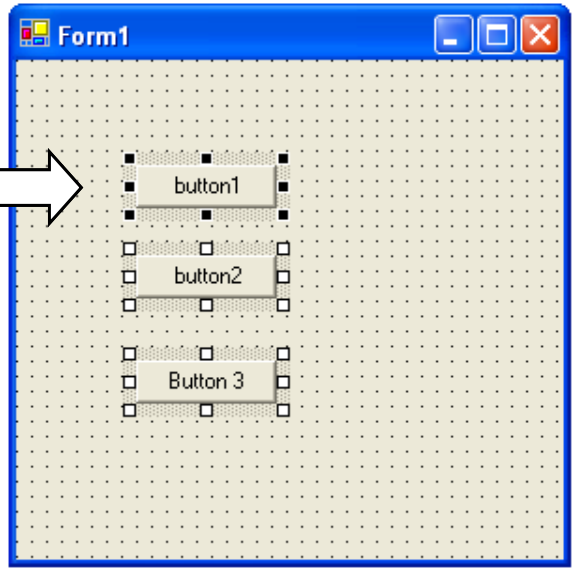
Layout Toolbar



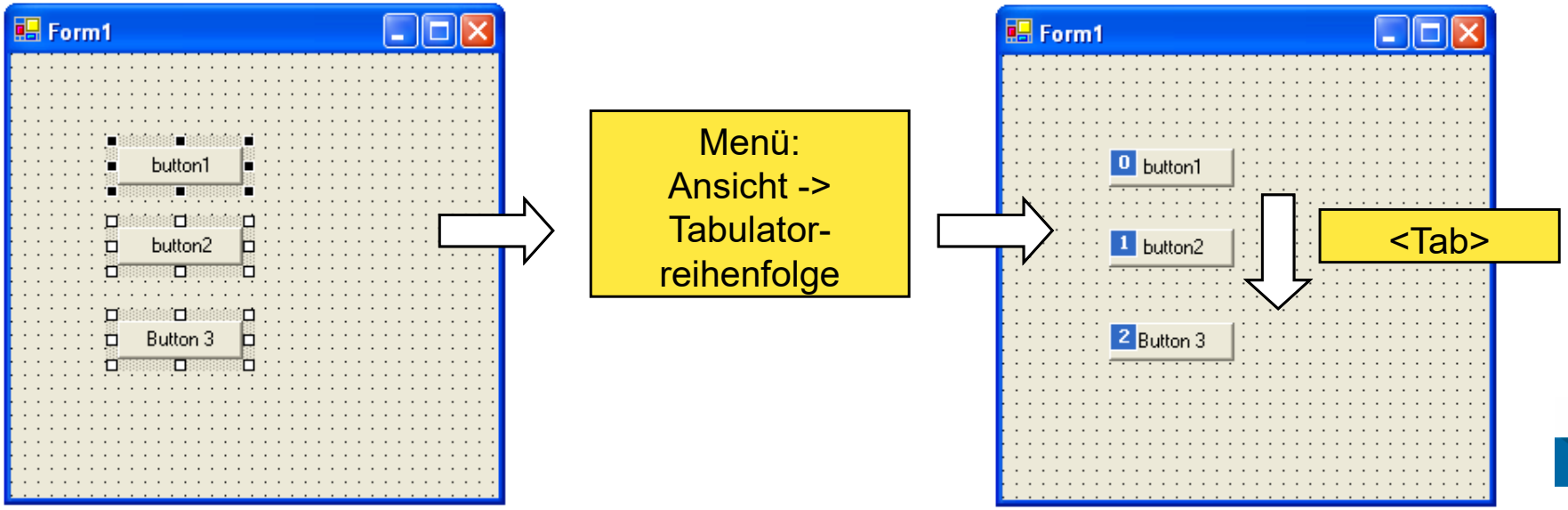
Referenz



Ausrichten links



Controls Tabulatorreihenfolge



Einige Formulareigenschaften

Eigenschaft:	Beschreibung:
AutoScroll	Scrollbars einfügen
BackGroundImage	Grafik für den Hintergrund
ContextMenu	Kontextmenü des Formulars
Controls	Collection aller enthaltenen Controls
DesktopLocation	Formularposition
Icon	Icon des Formulars
Menu	Menü des Formulars
Modal	Modal angezeigt?
Size	Formulargrösse
Text	Text in Titelleiste
Visible	Formular sichtbar?

Windows Forms Controls
Forms

Formular Ereignisse

Ereignis:	…tritt ein wenn:
– Activate	Formular wird aktiviert
– FormClosing	Formular soll geschlossen werden
– FormClosed	Formular ist geschlossen
– Deactivate	Ein anderes Formular wird aktiviert
– Load	Formular wird geladen
– Paint	Formular muss neu gezeichnet werden
– Resize	Grösse des Formulars wird verändert
– HelpRequested	Hilfe angefordert (F1)
– Layout	Controls müssen neu positioniert werden
– LocationChanged	Position des Formulars ändert
– MdiChildActivated	MDI-Child Fenster wird aktiviert

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Windows Forms Controls
Ereignisse

Mausereignisse

Ereignis:	…tritt ein wenn:
Click	Klick aufs Objekt
DoubleClick	Doppelklick aufs Objekt
MouseDown	Eine Maustaste niedergedrückt
MouseUp	Eine Maustaste losgelassen
MouseMove	Maus wird bewegt
MouseEnter	Maus in das Control hineinbewegt
MouseLeave	Maus aus dem Control hinausbewegt

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Windows Forms Controls
Ereignisse

Mausereignisse verarbeiten

```
public class MyForm : Form
{ ...

public MyForm()
  { ...
      // Event Handler hinzufügen
      btnLoad.Click += new System.EventHandler(this.OnLoadClick);
    ...
  }

  // Event Handler ausprogrammieren
  private void OnLoadClick(object sender, System.EventArgs e)
  {
    Button button = (sender as Button);
    ...
  }
}
```

Windows Forms Controls
Ereignisse

Tastaturereignisse

Ereignis:	…tritt ein wenn:
■ KeyPress	Eine Taste gedrückt
■ KeyDown	Eine Taste nach unten bewegt
■ KeyUp	Eine Taste losgelassen

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Windows Forms Controls
Ereignisse

Tastaturereignisse verarbeiten 

```
public class MyForm : Form
{ ...

  public MyForm()
  { ...
    // KeyPreview einschalten
    this.KeyPreview = true;
    // Event Handler hinzufügen
    this.KeyDown += new System.Windows.Forms.KeyEventHandler(this.KeyHandler);
  ...
  }

  // Event Handler ausprogrammieren
  private void KeyHandler(object sender, System.Windows.Forms.KeyEventArgs e)
  {
    ...
  }
}
```

Windows Forms Controls
Ereignisse

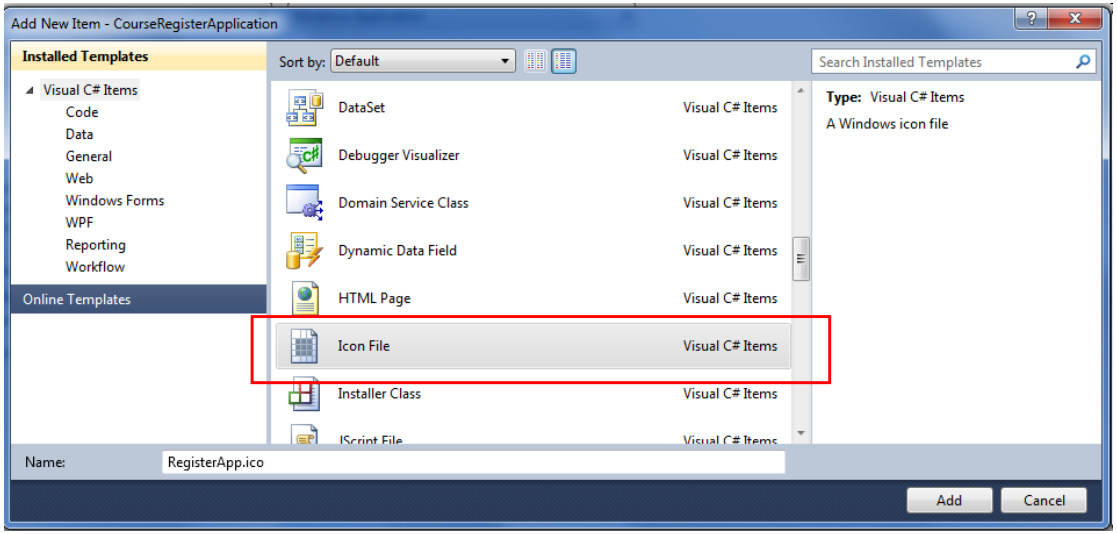
Einige weitere Ereignisse

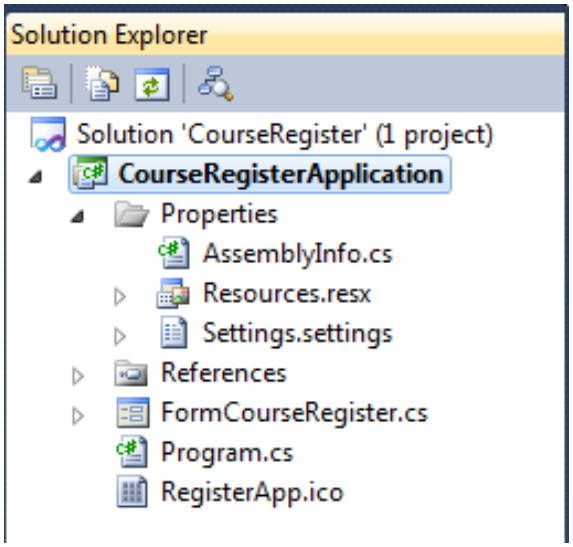
Ereignis:	…tritt ein wenn:
– Change	Inhalt der Komponente geändert
– Enter	Komponente erhält Fokus
– DragDrop	Objekt über Komponente abgelegt
– DragOver	Objekt über Komponente gezogen
– HelpRequested	Hilfe angefordert (F1)
– Leave	Komponente verliert den Fokus
– Paint	Steuerelement wird gezeichnet
– Resize	Grösse der Komponente wird verändert
– Validate	Inhalt von Steuerelement wird überprüft

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Application Icon setzen (1/3)

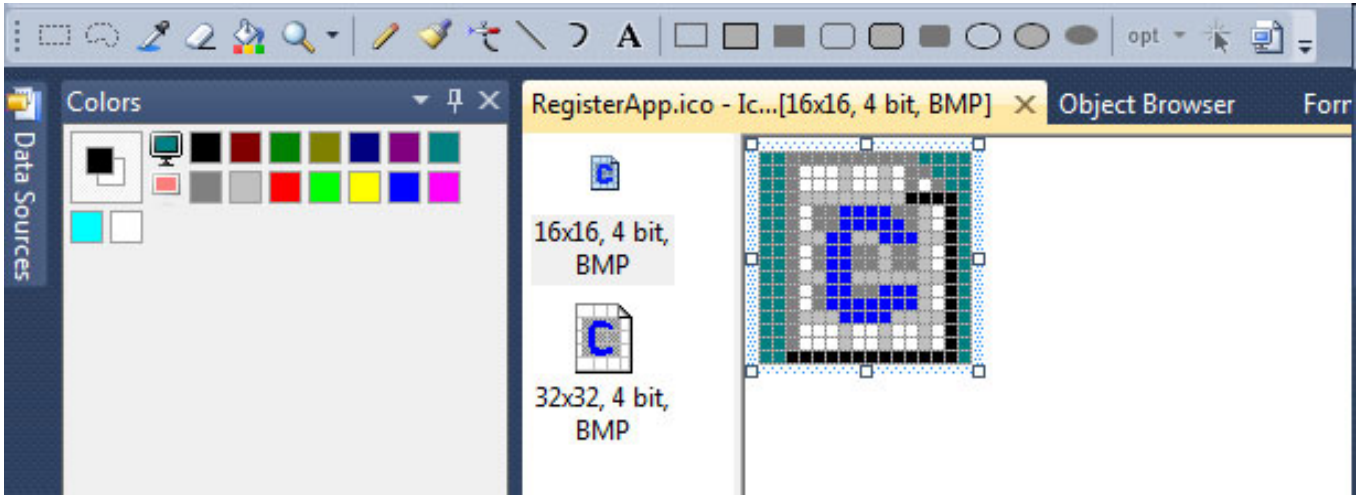
- Icon-Datei hinzufügen





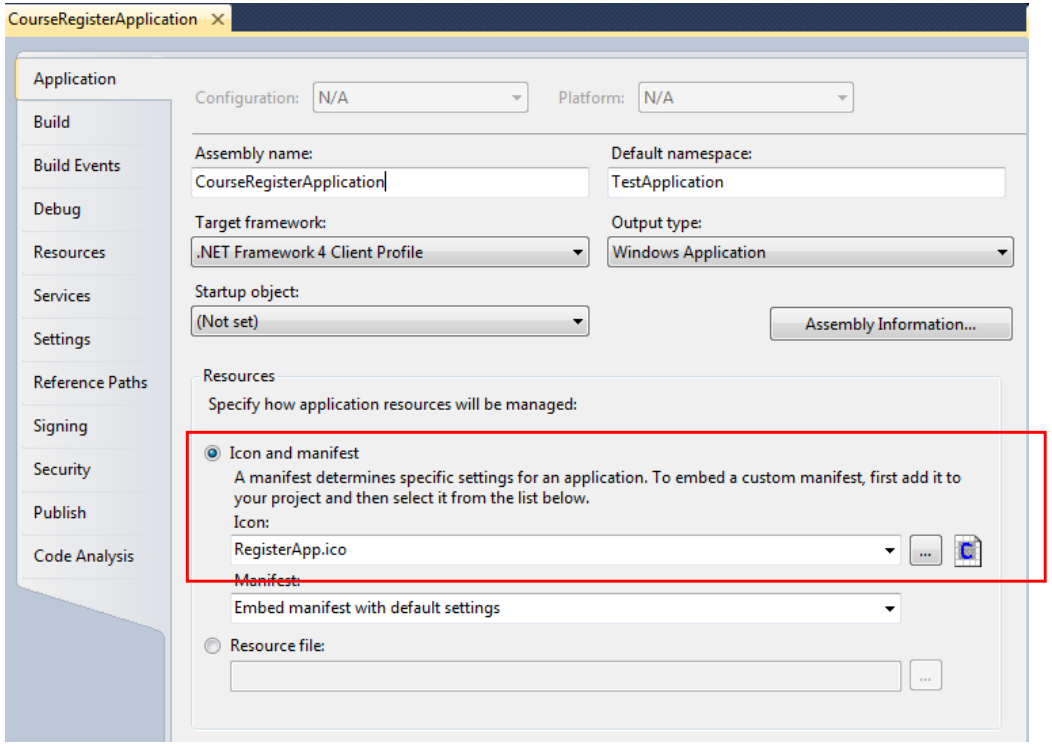
Application Icon setzen (2/3)

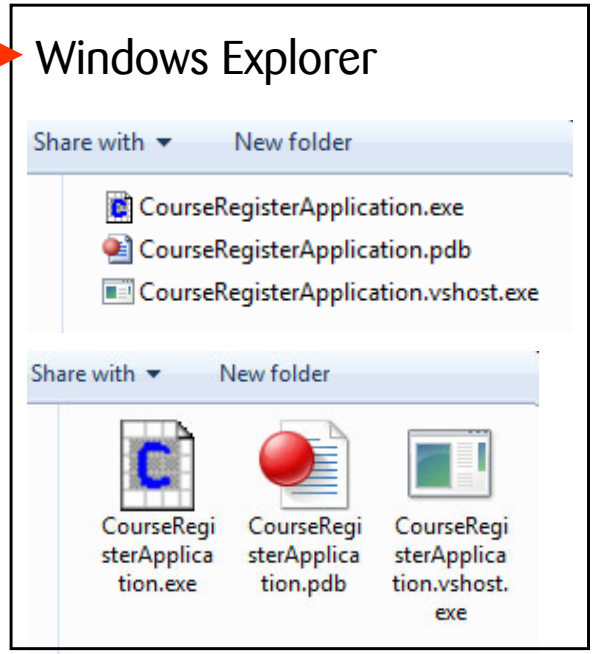
- Icon-Editor



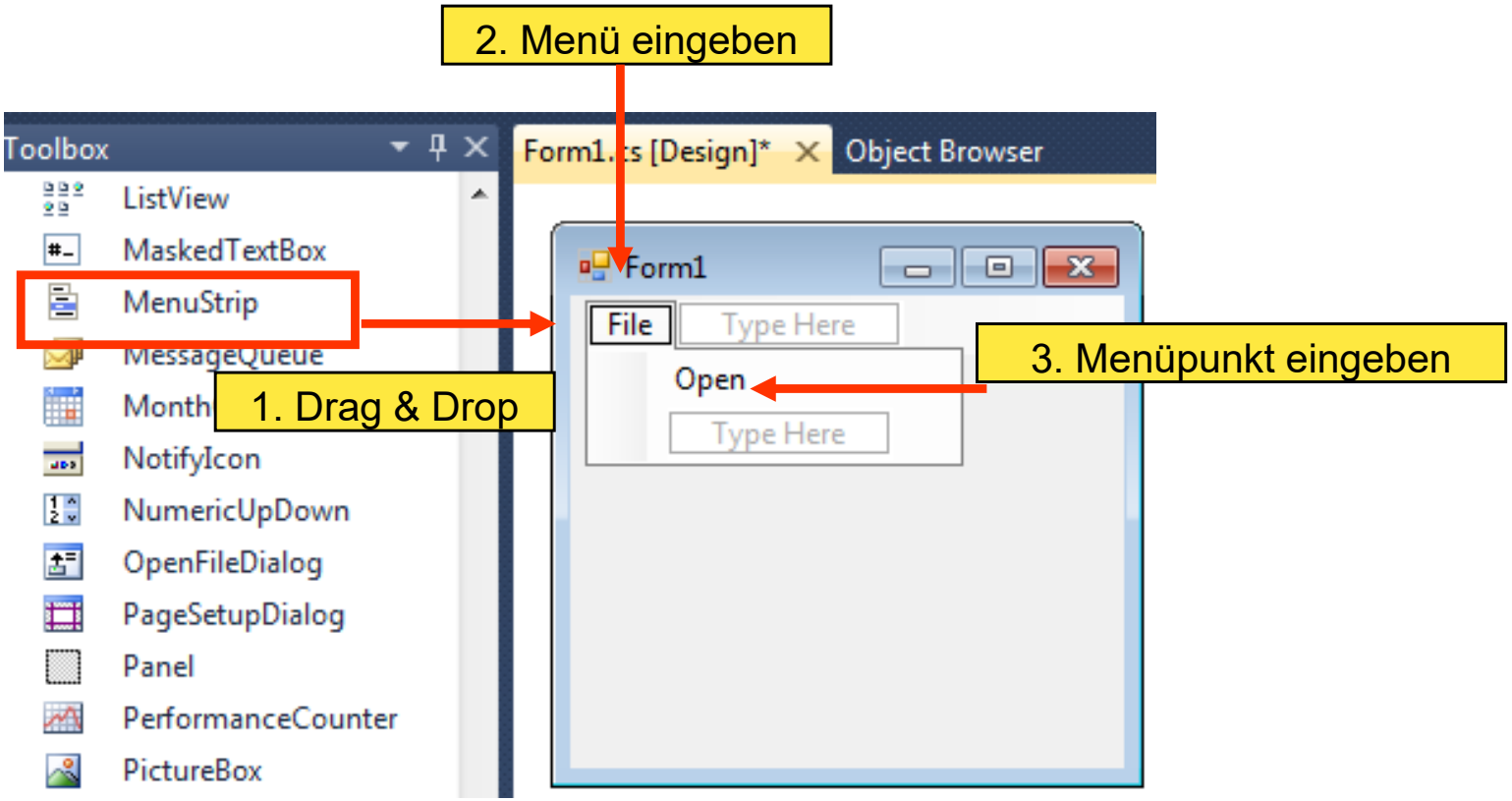
Application Icon setzen (3/3)

- Projekteigenschaften

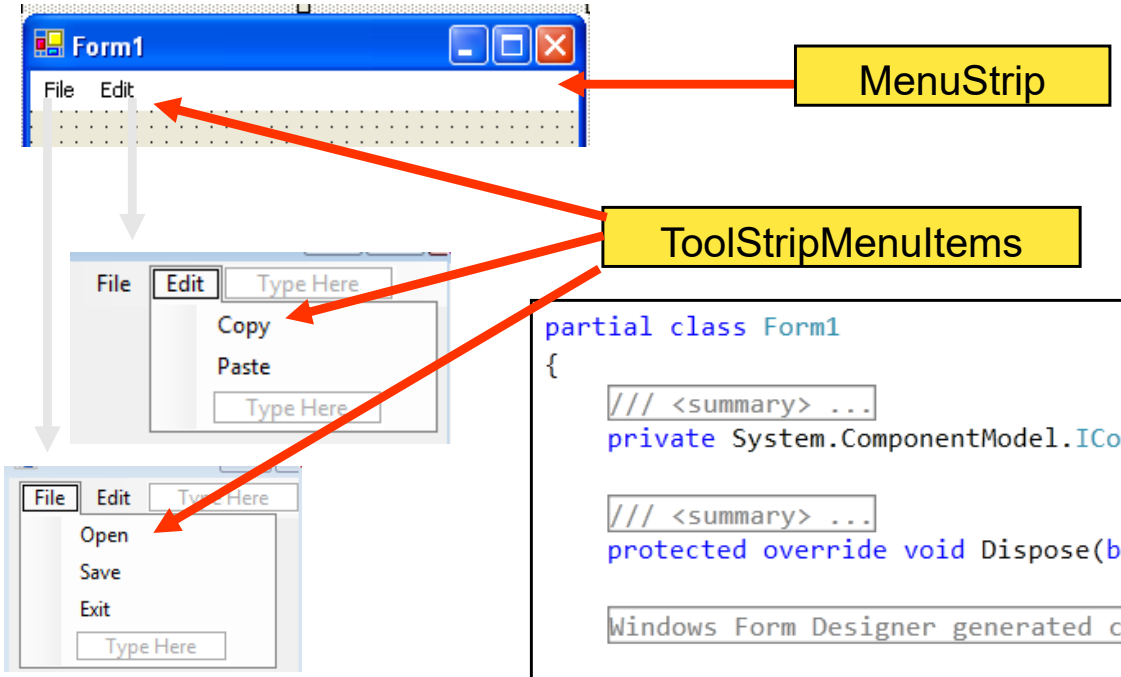




MenuStrip (1/5)



MenuStrip (2/5)




```
partial class Form1
{
    /// <summary> ...
    private System.ComponentModel.IContainer components = null;

    /// <summary> ...
    protected override void Dispose(bool disposing)...

    Windows Form Designer generated code

    private System.Windows.Forms.MenuStrip menuStrip1;
    private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem openToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem saveToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem editToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem copyToolStripMenuItem;
    private System.Windows.Forms.ToolStripMenuItem pasteToolStripMenuItem;
}
```

MenuStrip (3/5)

- MenuStrip und ToolStripMenuItem instanziieren

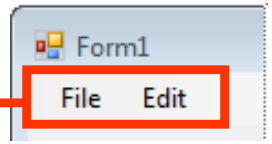
```
private void InitializeComponent()
{
    this.menuStrip1 = new System.Windows.Forms.MenuStrip();
    this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.openToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.editToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.saveToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.copyToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.pasteToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.menuStrip1.SuspendLayout();
    this.SuspendLayout();
```

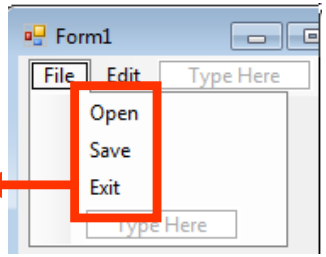
Unterbricht vorübergehend die Layoutlogik für das Steuerelement

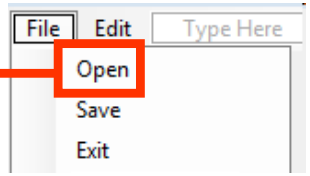
Windows Forms Controls
Menus

MenuStrip (4/5)

```
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.fileToolStripMenuItem,
this.editToolStripMenuItem});
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(224, 24);
this.menuStrip1.TabIndex = 0;
this.menuStrip1.Text = "menuStrip1";
//
// fileToolStripMenuItem
//
this.fileToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.openToolStripMenuItem,
this.saveToolStripMenuItem,
this.exitToolStripMenuItem});
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
this.fileToolStripMenuItem.Text = "File";
//
// openToolStripMenuItem
//
this.openToolStripMenuItem.Name = "openToolStripMenuItem";
this.openToolStripMenuItem.Size = new System.Drawing.Size(152, 22);
this.openToolStripMenuItem.Text = "Open";
```







MenuStrip (5/5)

- EventHandler hinzufügen

Form1.Designer.cs

```
//
// openToolStripMenuItem
//
this.openToolStripMenuItem.Name = "openToolStripMenuItem";
this.openToolStripMenuItem.Size = new System.Drawing.Size(152, 22);
this.openToolStripMenuItem.Text = "Open";
this.openToolStripMenuItem.Click += new System.EventHandler(this.openToolStripMenuItem_Click);
//
```

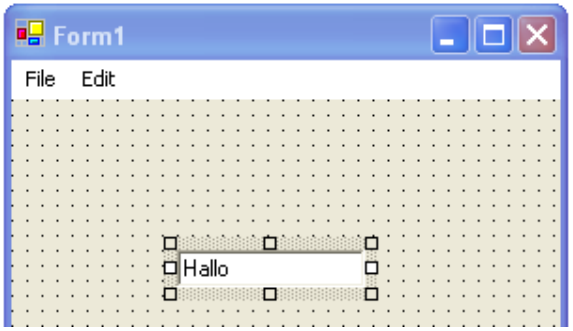
- EventHandler

Form1.cs

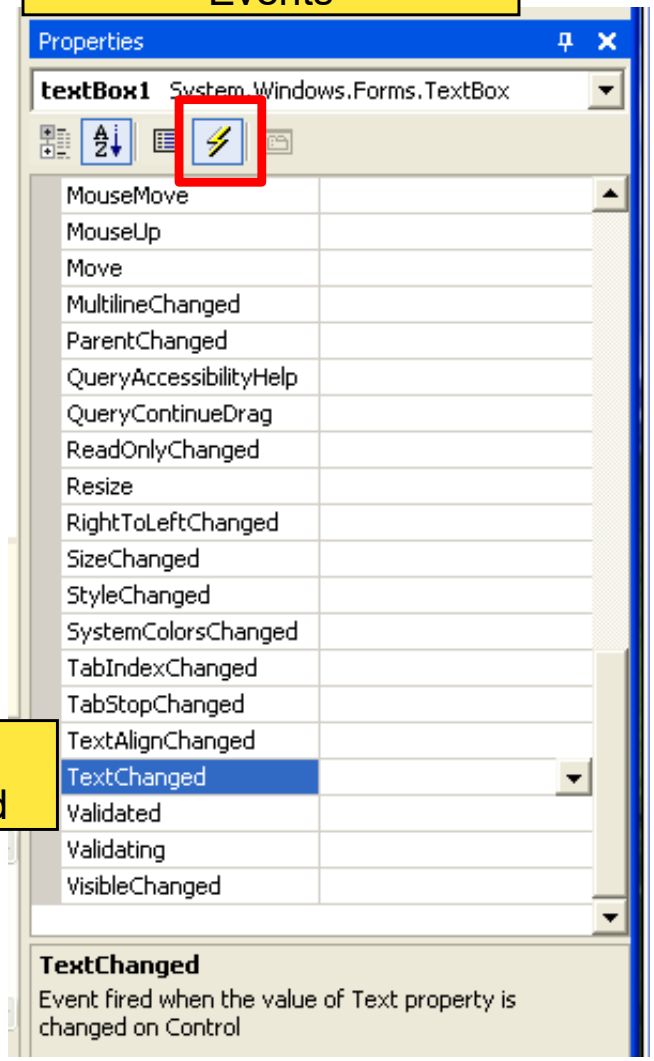
```
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{

}
```

Windows Forms Controls

TextBox - Control

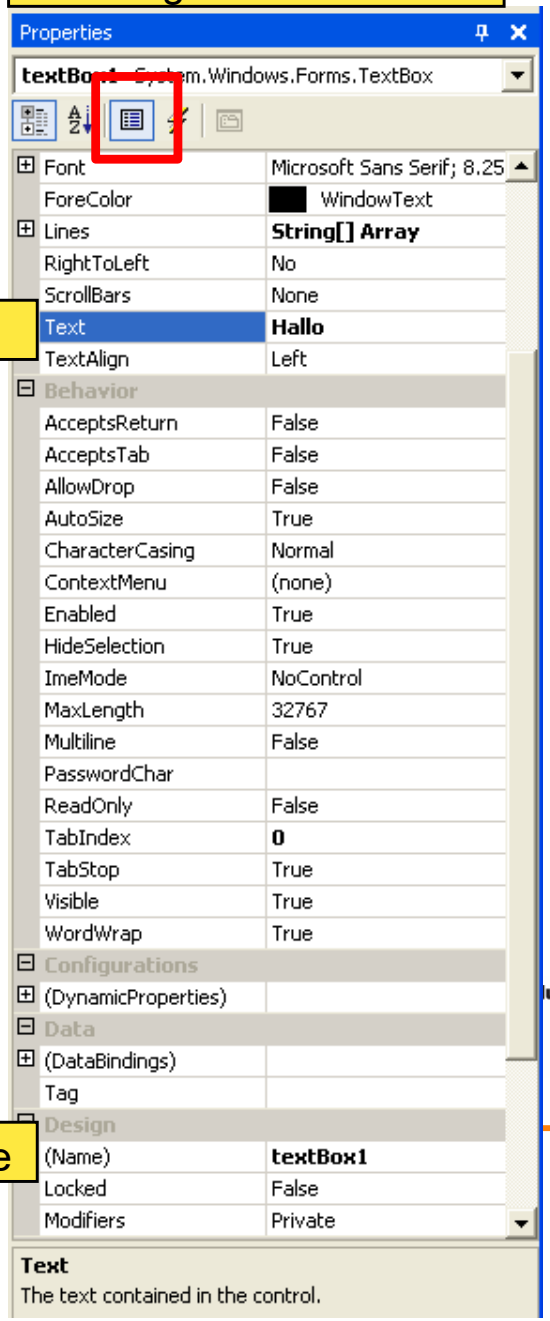


Events



Event: TextChanged

Eigenschaften

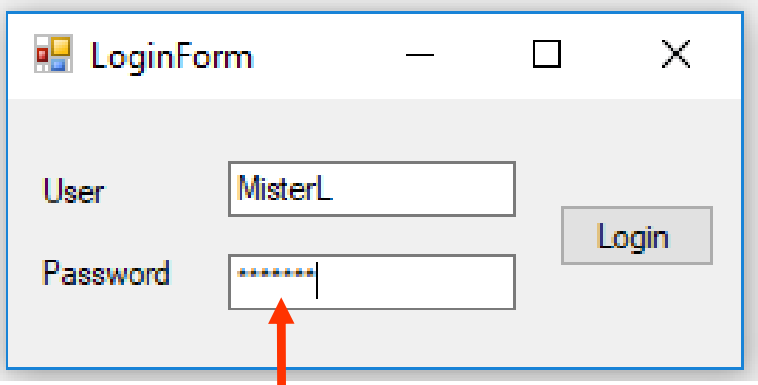


Text

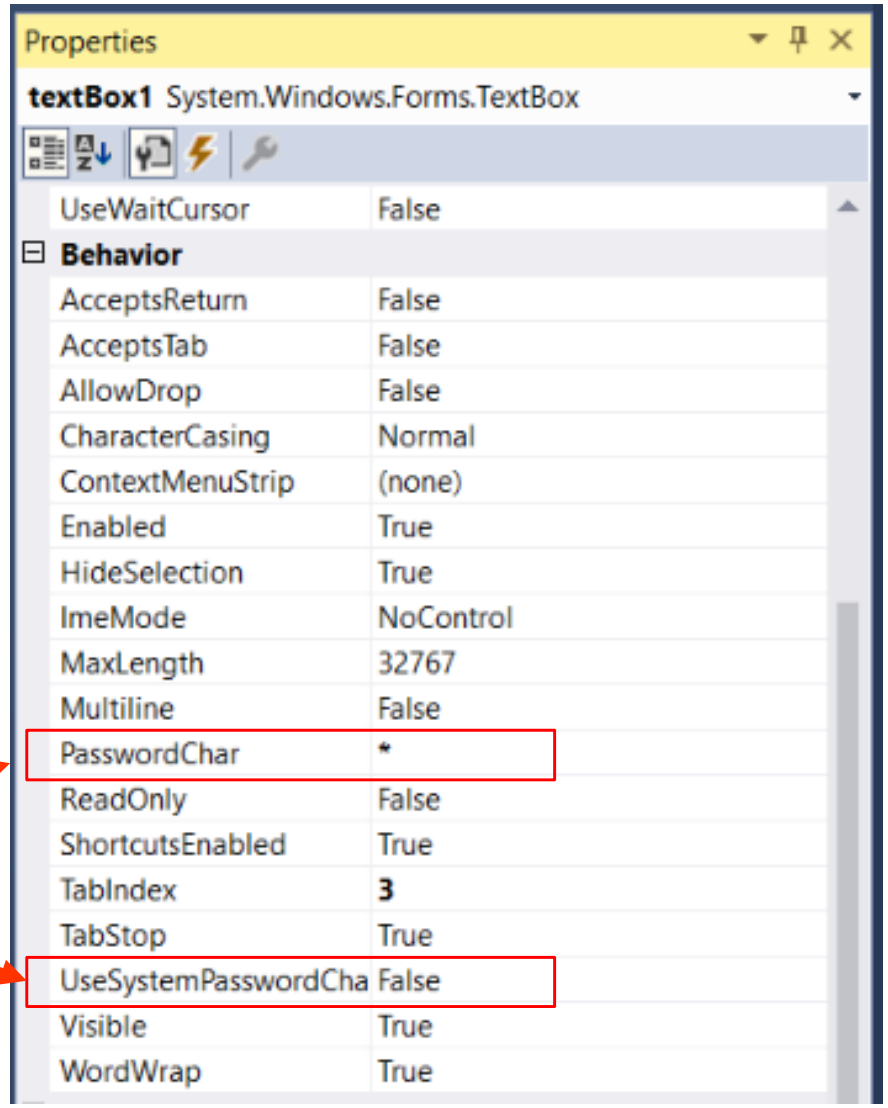
Name

Windows Forms Controls
Password

TextBox – für Passworteingabe

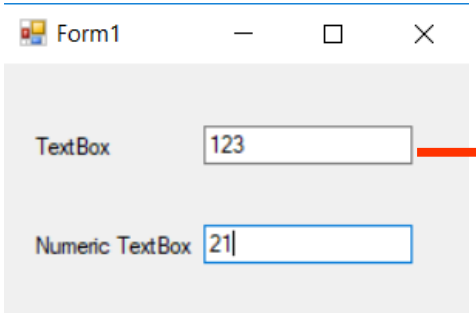


Eigenschaften
für Passwort



Windows Forms Controls
Numerische TextBox

Variante 1 – KeyPressHandler für TextBox



Event: KeyPress

```
public partial class Form1 : Form
{
    1 reference
    public Form1()...

    1 reference
    private void NumericTextBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        // Überprüfen on die gedrückte Taste BACKSPACE '(char)8' oder Numerisch ist.
        if (e.KeyChar >= '0' && e.KeyChar <= '9' || e.KeyChar == (char)8)
        {
            e.Handled = false; // Die Eingabe nicht verwerfen
        }
        else
        {
            e.Handled = true; // Die Eingabe verwerfen
        }
    }
}
```

KeyPress EventHandler

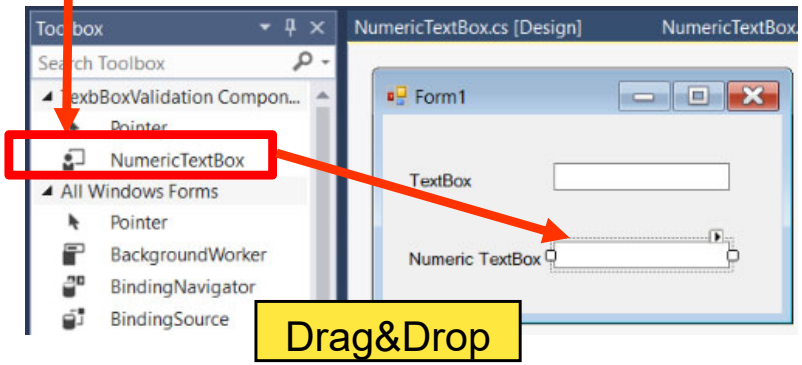
Windows Forms Controls
Numerische TextBox

Variante 2 – Klasse TextBox erweitern

Neue Klasse
NumericTextBox : TextBox

```
class NumericTextBox : TextBox
{
    0 references
    protected override void OnKeyPress(KeyPressEventArgs e)
    {
        base.OnKeyPress(e); // OnKeyPress() der Basisklasse aufrufen

        // Überprüfen on die gedrückte Taste BACKSPACE '(char)8' oder Numerisch ist.
        if (e.KeyChar != (char)8 && !char.IsNumber(e.KeyChar))
        {
            // Das Event als abgehandelt markieren
            e.Handled = true;
        }
    }
}
```

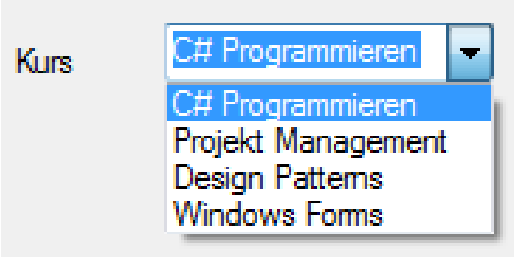
Override Method:
OnKeyPress()



Drag&Drop

Windows Forms Controls

ComboBox – Control



- Initialisieren

```
private void InitializeComboboxes()
{
    cbxCourse.Items.AddRange(new string[] { "C# Programmieren", "Projekt Management", "Design Patterns", "Windows Forms" });
    cbxCourse.SelectedIndex = 0;
}
```

- EventHandler zu SelectedIndexChanged

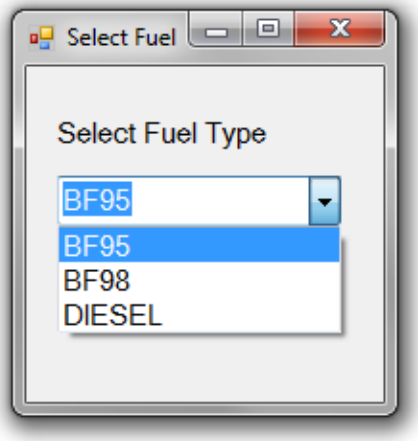
```
private void cbxCourse_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show("Selected Item = " + cbxCourse.SelectedItem);
}
```

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

ComboBox – Enum als Datenquelle (1/2)

```
public enum FuelTypes { BF95, BF98, DIESEL }
```

Enum als Datenquelle

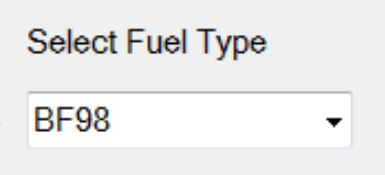


- Initialisieren

```
public partial class SelectFuelForm : Form
{
    public SelectFuelForm()
    {
        InitializeComponent();
    }

    private void SelectFuelForm_Load(object sender, EventArgs e)
    {
        cbxSelectFuelType.DataSource = Enum.GetValues(typeof(FuelTypes));
    }
}
```

- Selektieren mit Code

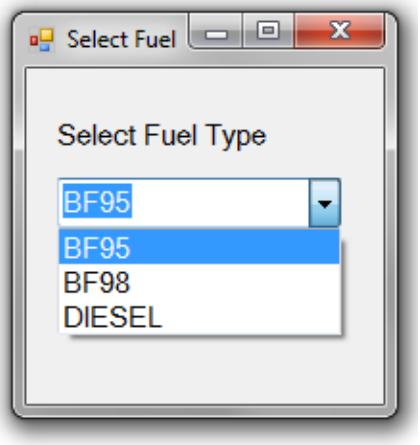
```
cbxSelectFuelType.SelectedIndex = (int) FuelTypes.BF98;
```



ComboBox – Enum als Datenquelle (2/2)

```
public enum FuelTypes { BF95, BF98, DIESEL }
```

Enum als Datenquelle



- Auswahl in `switch` auswerten

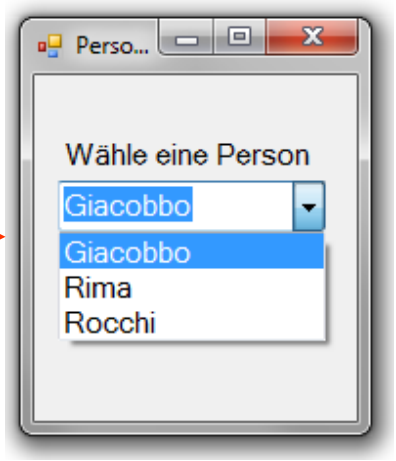
```
private void cbxSelectFuelType_SelectedIndexChanged(object sender, EventArgs e)
{
    switch ((FuelTypes)cbxSelectFuelType.SelectedItem)
    {
        case FuelTypes.BF95:
            break;
        case FuelTypes.BF98:
            break;
        case FuelTypes.DIESEL:
            break;
        default:
            break;
    }
}
```

Windows Forms Controls

ComboBox – Liste von Objekten als Datenquelle

```
class Person
{
    public int PersonalID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string MiddleName { get; set; }
}
```

Personenliste als Datenquelle

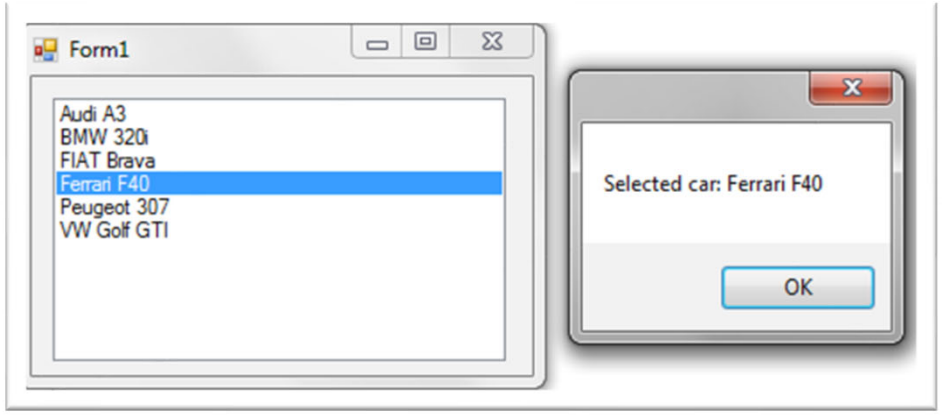


- Initialisieren

```
private List<Person> personList = new List<Person>();
private BindingSource cbxBindingSource = new BindingSource();

private void SelectPersonForm_Load(object sender, EventArgs e)
{
    personList.Add(new Person() { PersonalID = 2, FirstName = "Viktor", LastName = "Giacobbo" });
    personList.Add(new Person() { PersonalID = 32, FirstName = "Marco", LastName = "Rima" });
    personList.Add(new Person() { PersonalID = 87, FirstName = "Massimo", LastName = "Rocchi" });

    cbxBindingSource.DataSource = personList;

    cbxPersonList.DataSource = cbxBindingSource;
    cbxPersonList.DisplayMember = "LastName";
    cbxPersonList.ValueMember = "PersonalID";
}
```

Windows Forms Controls

ListBox – Control



- Initialisieren

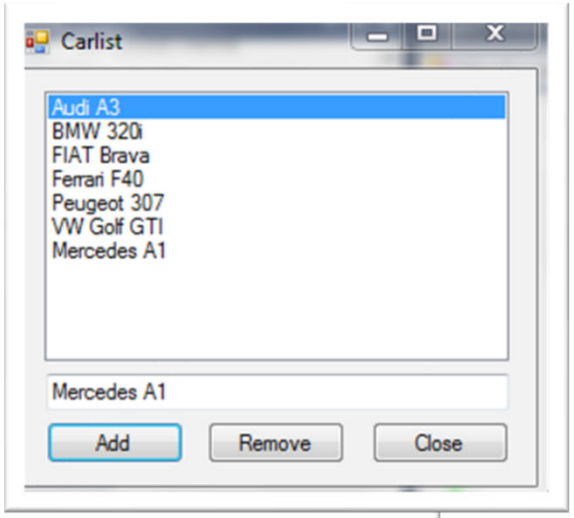
```
public Form1()
{
    InitializeComponent();
    InitCarList();
}

public void InitCarList()
{
    List<string> cars = new List<string>();
    cars.AddRange(new string[] {"Audi A3", "BMW 320i", "FIAT Brava", "Ferrari F40", "Peugeot 307", "VW Golf GTI" });
    listBoxCars.DataSource = cars;
}
```

- EventHandler zu SelectedIndexChanged

```
private void listBoxCars_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show("Selected car: " + listBoxCars.SelectedItem);
}
```

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

ListBox – Control-Beispiel mit Add und Remove (1/2)



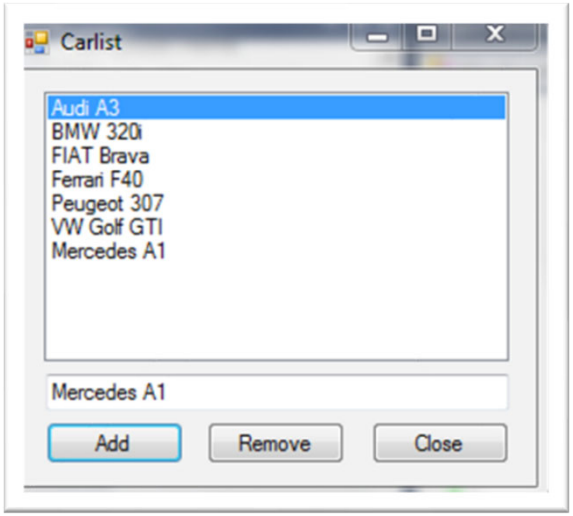
- Initialisieren

```
List<string> cars = new List<string>();

public Form1()
{
    InitializeComponent();
    InitCarList();
}

public void InitCarList()
{
    cars.AddRange(new string[] {"Audi A3", "BMW 320i", "FIAT Brava", "Ferrari F40", "Peugeot 307", "VW Golf GTI" });
    UpdateList();
}
```

- Liste aktualisieren

```
private void UpdateList()
{
    listBoxCars.DataSource = null;
    listBoxCars.DataSource = cars;
}
```

ListBox – Control-Beispiel mit Add und Remove (2/2)



- List-Items hinzufügen

```
private void buttonAdd_Click(object sender, EventArgs e)
{
    cars.Add(textBoxEdit.Text);
    UpdateList();
}
```

- List-Items löschen

```
private void buttonRemove_Click(object sender, EventArgs e)
{
    if ((listBoxCars.SelectedIndex >= 0) && (listBoxCars.SelectedIndex < cars.Count))
    cars.RemoveAt(listBoxCars.SelectedIndex);
    UpdateList();
}
```

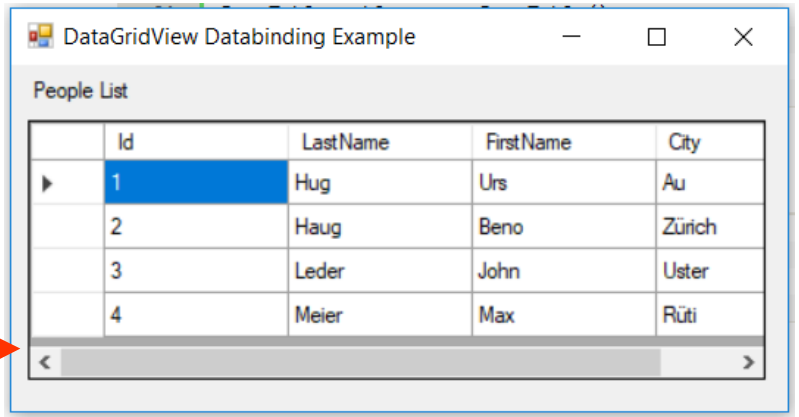
- Exit

```
private void buttonClose_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

DataGridView – Databinding an List<T>

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        List<Person> people = PersonRepository.GetPeopleList();
        PersonDataGridView.DataSource = people;
    }
}
```

Databinding




```
public static class PersonRepository
{
    public static List<Person> GetPeopleList()
    {
        List<Person> peopleList = new List<Person>()
        {
            new Person() {Id = 1, FirstName = "Urs", LastName = "Hug", City = "Au"},
            new Person() {Id = 2, FirstName = "Beno", LastName = "Haug", City = "Zürich"},
            new Person() {Id = 3, FirstName = "John", LastName = "Leder", City = "Uster"},
            new Person() {Id = 4, FirstName = "Max", LastName = "Meier", City = "Rüti"},

        };

        return peopleList;
    }
}
```

Data Source: List<Person>

Data Class: Person

```
public class Person
{
    public int Id { get; set; }
    public string LastName { get; set; }
    public string FirstName { get; set; }
    public string City { get; set; }
}
```

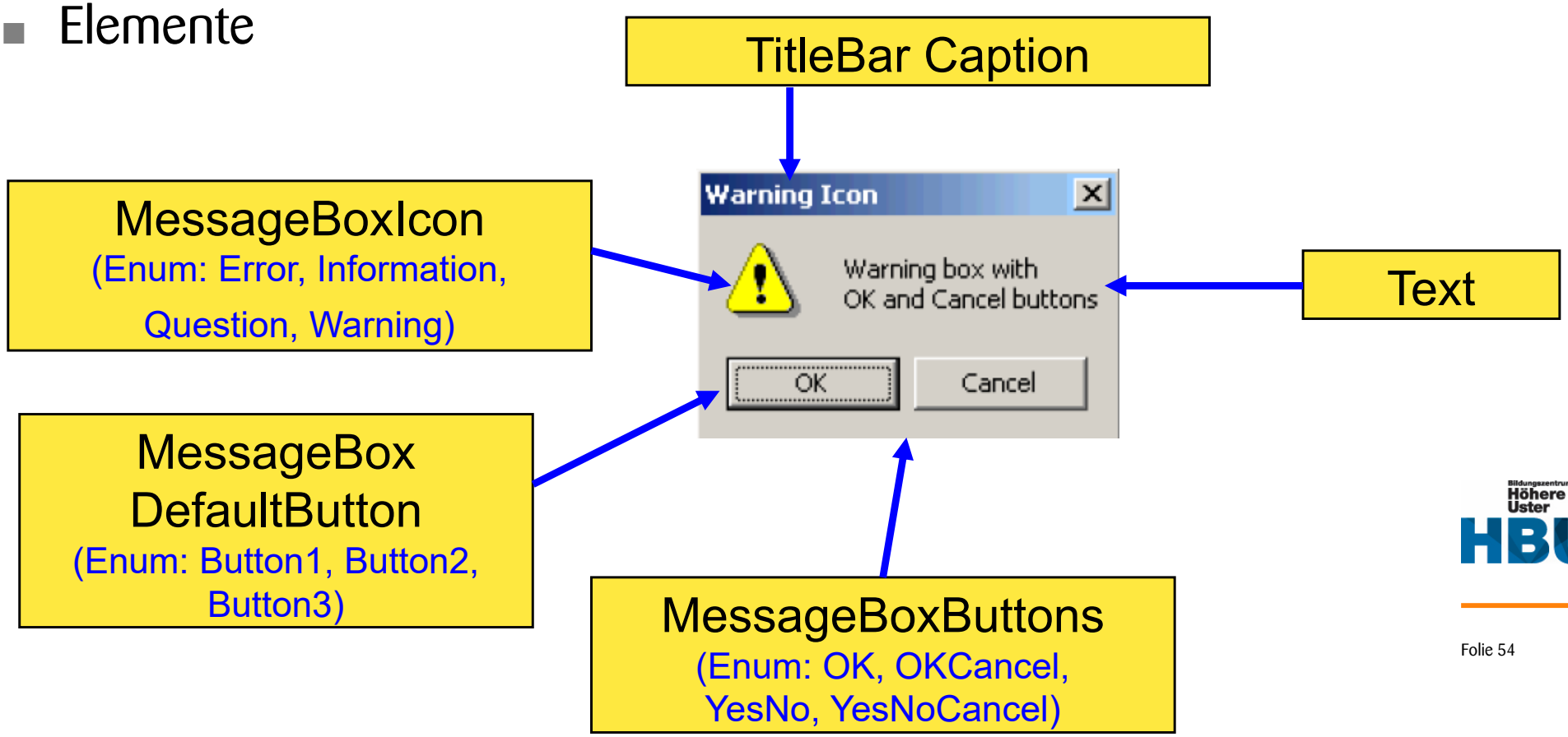
MessageBox (1/4)

- Eine `MessageBox` wird benutzt, um einfache Meldungen anzuzeigen
- Der Benutzer muss diese quittieren oder eine Auswahl treffen
- Die Applikation wartet auf die Benutzteraktion (modal)
- Kann nicht instanziiert werden

```
// Anzeigen einer MessageBox mit:

MessageBox.Show(...)
```

MessageBox (2/4)

- Elemente



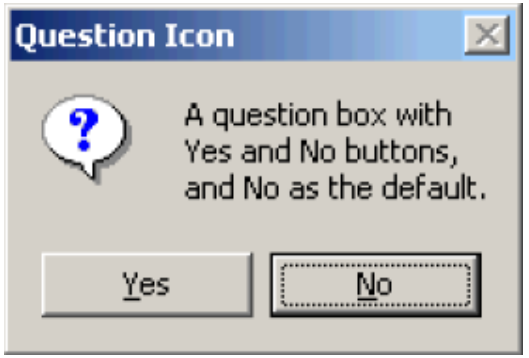
MessageBox (3/4)

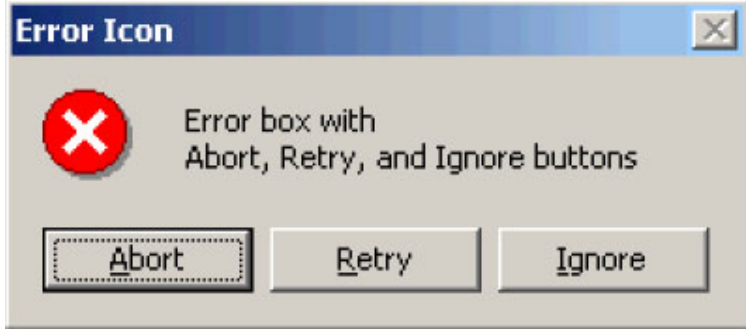
- Einige Varianten der Methode `Show()`

```
public static DialogResult Show( string text );

public static DialogResult Show( IWin32Window owner, string text );

public static DialogResult Show( string text, string caption );

public static DialogResult Show( IWin32Window owner,
                                 string text,
                                 string caption,
                                 MessageBoxButtons buttons );

public static DialogResult Show( IWin32Window owner,
                                 string text,
                                 string caption,
                                 MessageBoxButtons buttons,
                                 MessageBoxIcon icon,
                                 MessageBoxDefaultButton defaultButton,
                                 MessageBoxOptions options );
```

MessageBox (4/4)

- Beispiele von `MessageBox`-Typen





Windows Forms Controls
Dialoge

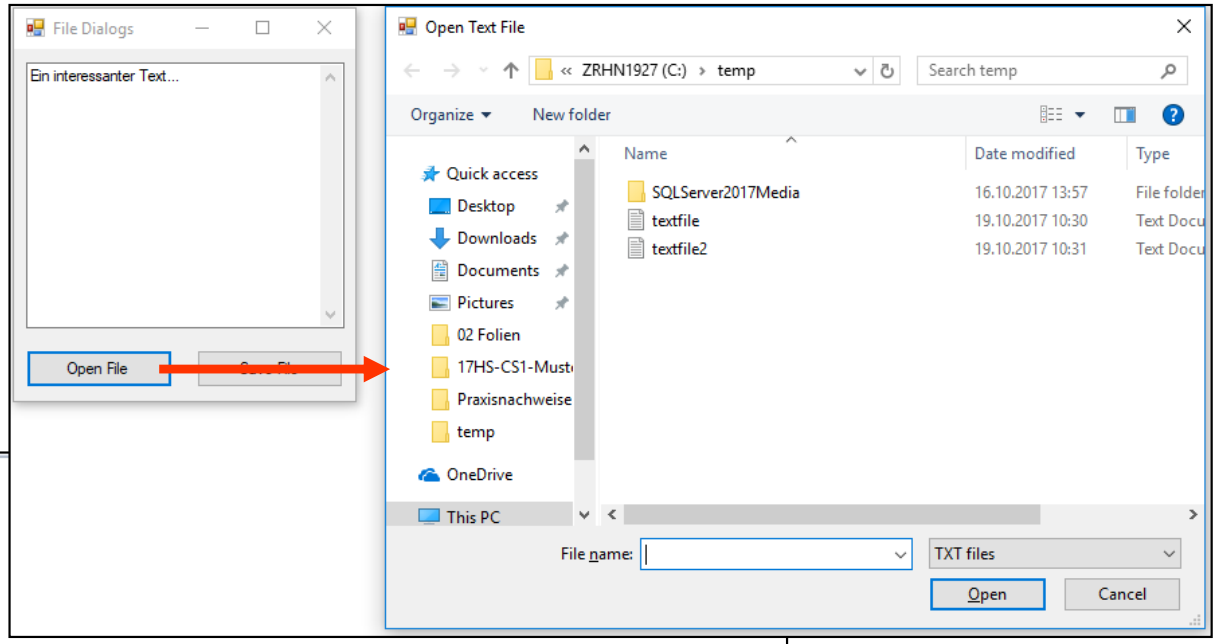
Die `DialogResult`-Werte:

- Abort → “Abort” Button gedrückt
- Cancel → “Cancel” Button gedrückt
- Ignore → “Ignore” Button gedrückt
- No → “No” Button gedrückt
- None → Kein Result-Wert, d.h. der Dialog ist noch offen
- OK → “OK” Button gedrückt
- Retry → “Retry” Button gedrückt
- Yes → “Yes” Button gedrückt

Namensraum: `System.Windows.Forms`

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Windows Forms Controls Dialoge

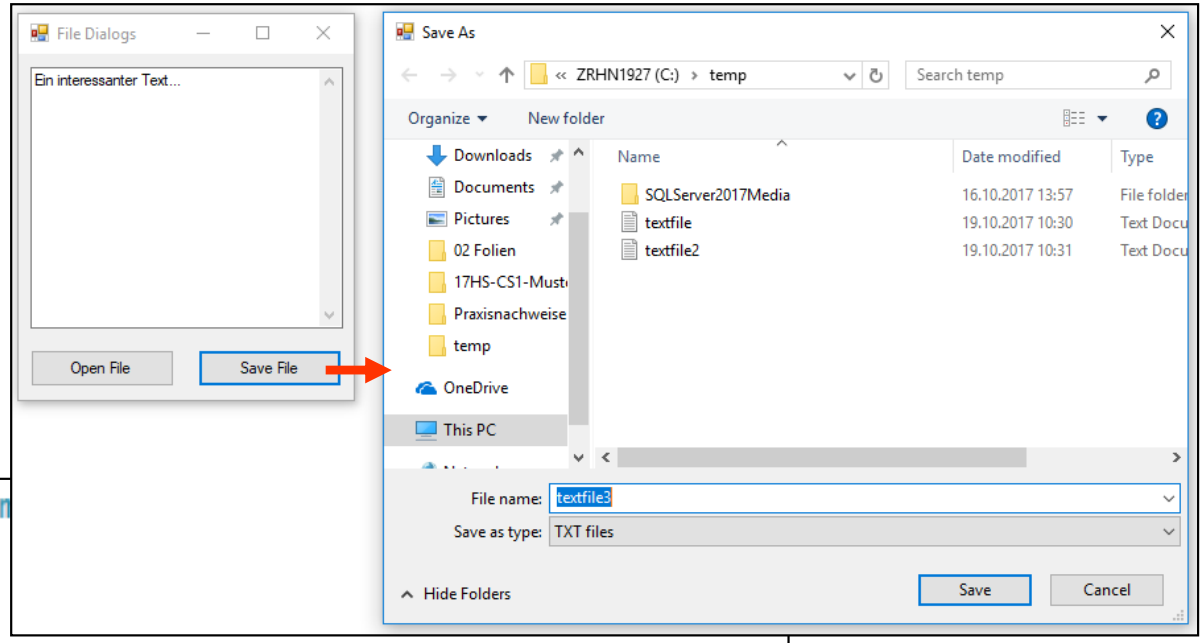


OpenFileDialog

```
public partial class FileDialogsExampleForm : Form
{
    string directory = @"C:\";

    public FileDialogsExampleForm()...

    private void openFileButton_Click(object sender, EventArgs e)
    {
        OpenFileDialog openFileDialog = new OpenFileDialog();
        openFileDialog.Title = "Open Text File";
        openFileDialog.Filter = "TXT files|*.txt";
        openFileDialog.InitialDirectory = directory;
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            try
            {
                if ((openFileDialog.OpenFile()) != null)
                {
                    string fileContent = File.ReadAllText(openFileDialog.FileName);
                    fileContentTextBox.Text = fileContent;
                    directory = Path.GetDirectoryName(openFileDialog.FileName);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Fehler: Das File konnte nicht gelesen werden. Exception: " + ex.Message);
            }
        }
        openFileDialog.Dispose();
    }
```

Windows Forms Controls Dialoge

SaveFileDialog

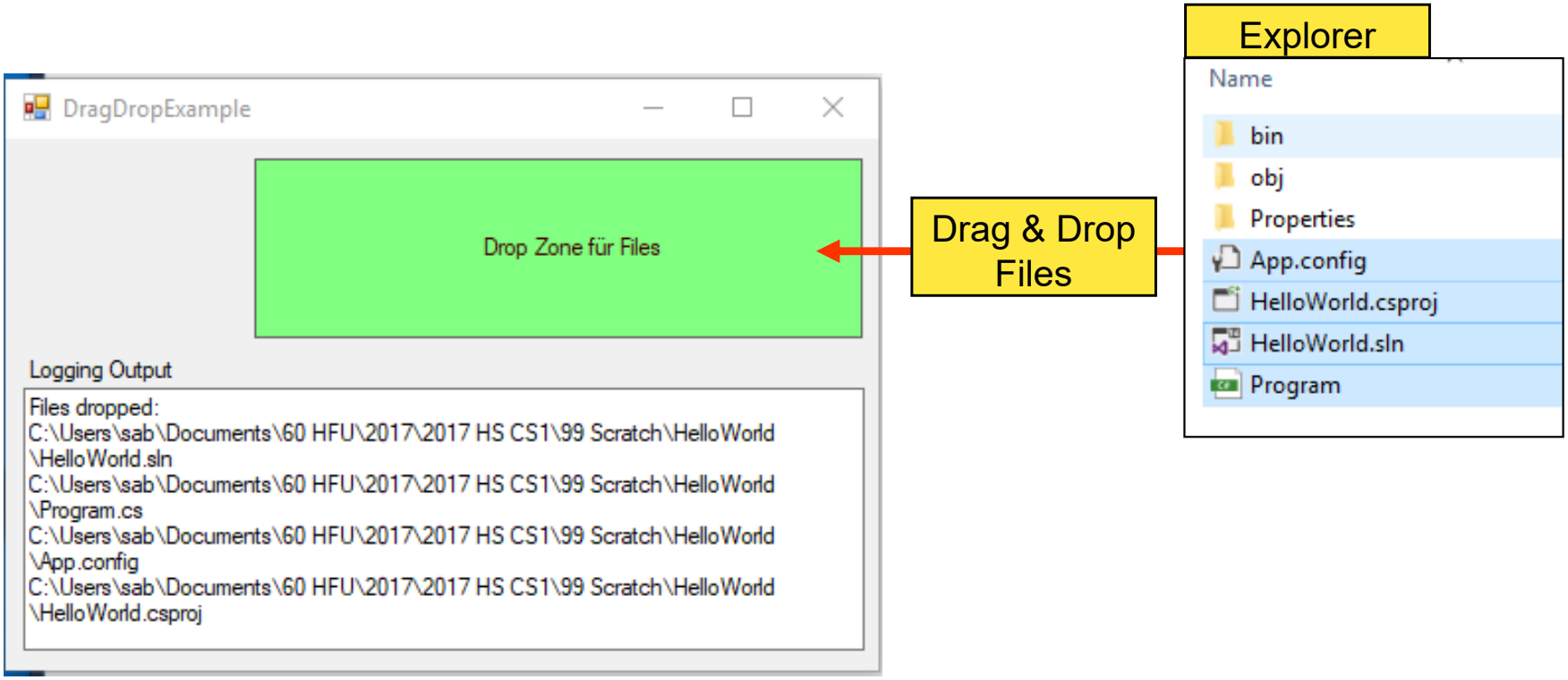



```
public partial class FileDialogsExampleForm
{
    string directory = @"C:\";

    public FileDialogsExampleForm()...

    private void openFileButton_Click(object sender, EventArgs e)...

    private void saveFileButton_Click(object sender, EventArgs e)
    {
        SaveFileDialog savefileDialog = new SaveFileDialog();
        savefileDialog.Filter = "TXT files|*.txt";
        savefileDialog.InitialDirectory = directory;

        if (savefileDialog.ShowDialog() == DialogResult.OK)
        {
            File.WriteAllText(savefileDialog.FileName, fileContentTextBox.Text);
            directory = Path.GetDirectoryName(savefileDialog.FileName);
        }
        savefileDialog.Dispose();
    }
}
```

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Windows Forms Controls
Drag & Drop

Beispiel: Drag & Drop Files auf eine Zone (Panel)

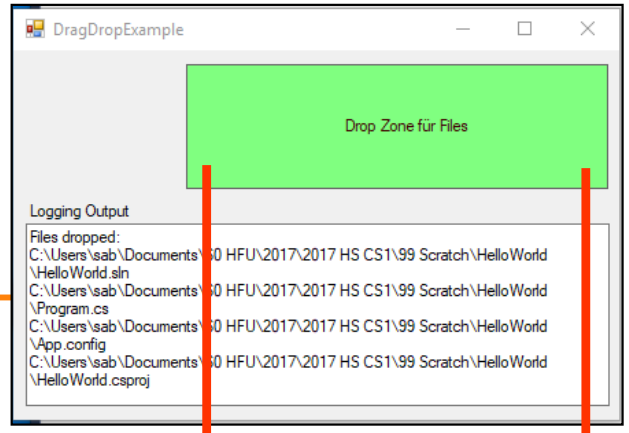


Windows Forms Controls
Drag & Drop

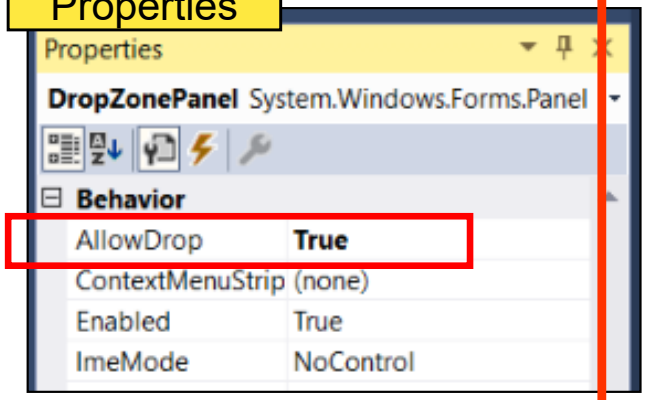
Beispiel: Drag & Drop Files auf eine Zone (Panel)

- EventHandler für DragEnter und DragDrop

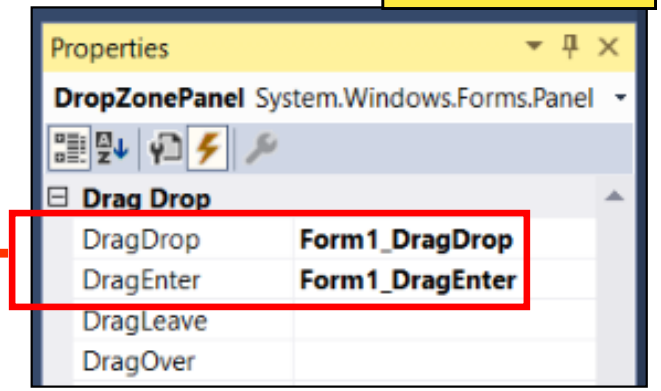
```
private void Form1_DragEnter(object sender, DragEventArgs e)
{
    Debug.Print("DragEnter");
    if (e.Data.GetDataPresent(DataFormats.FileDrop))
    {
        e.Effect = DragDropEffects.Copy;
    }
}

private void Form1_DragDrop(object sender, DragEventArgs e)
{
    string[] files = (string[])e.Data.GetData(DataFormats.FileDrop);
    StringBuilder sb = new StringBuilder();
    sb.AppendLine("Files dropped:");
    foreach (string file in files)
    {
        sb.AppendLine(file);
    }
    OutputTextBox.Text = sb.ToString();
}
```



Properties



Events



Windows Forms Controls
Eigene User Controls erstellen

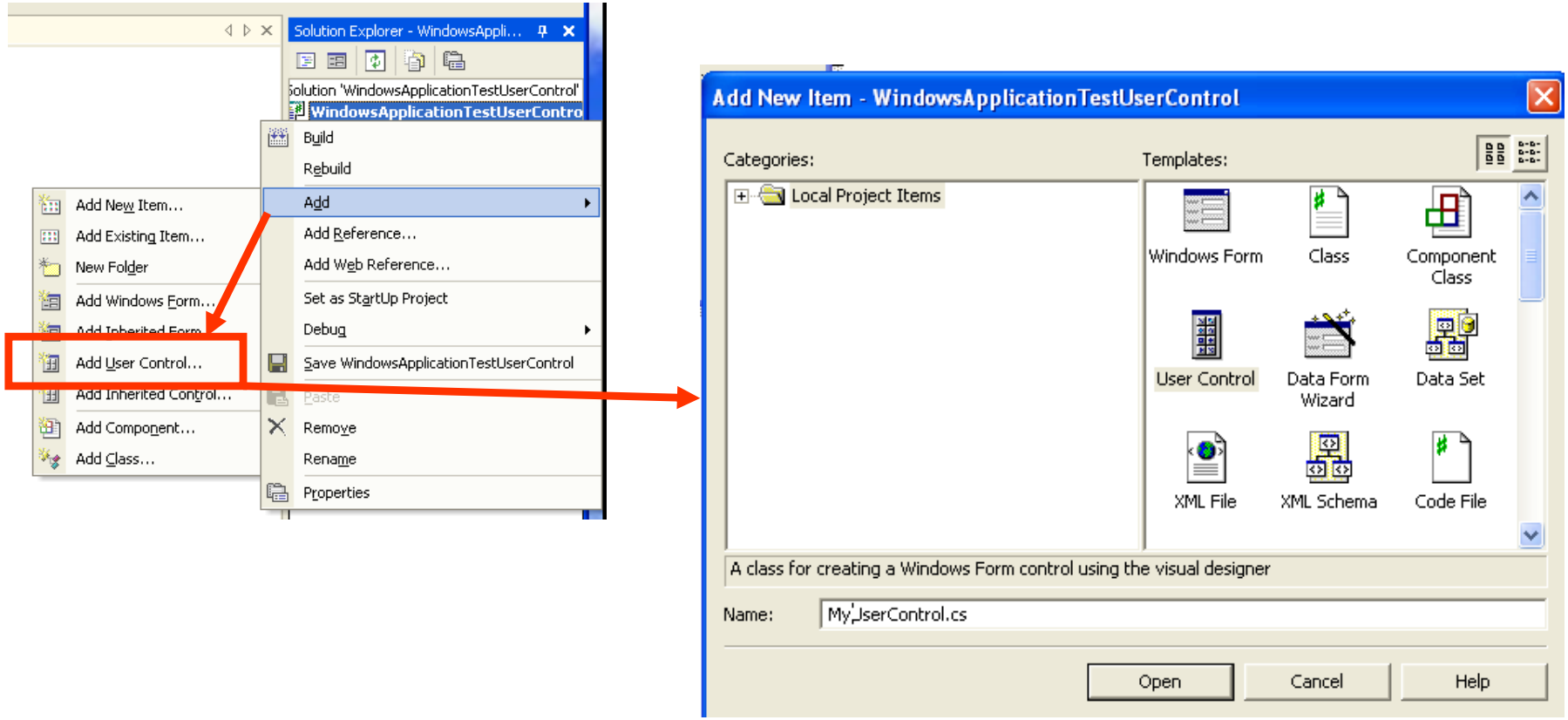
Was ist ein User Control?

- Ein User Control kann in Windows Forms Applikationen benutzt werden
- In der Toolbox unter „My User Controls“ ersichtlich
- Es besteht aus anderen WinForms-Controls
- Es wird abgeleitet von:
  `System.Windows.Forms.UserControl`
- Das User Control verhält sich wie andere WinForms-Controls

Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

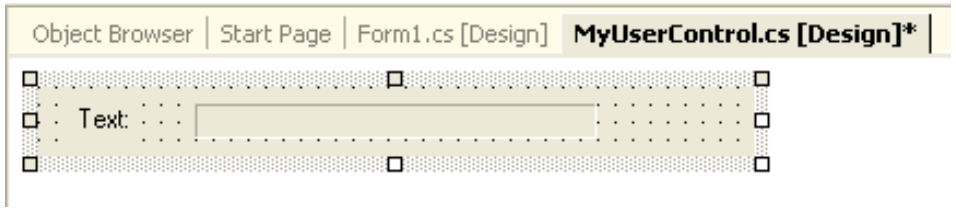
Windows Forms Controls
Eigene User Controls erstellen

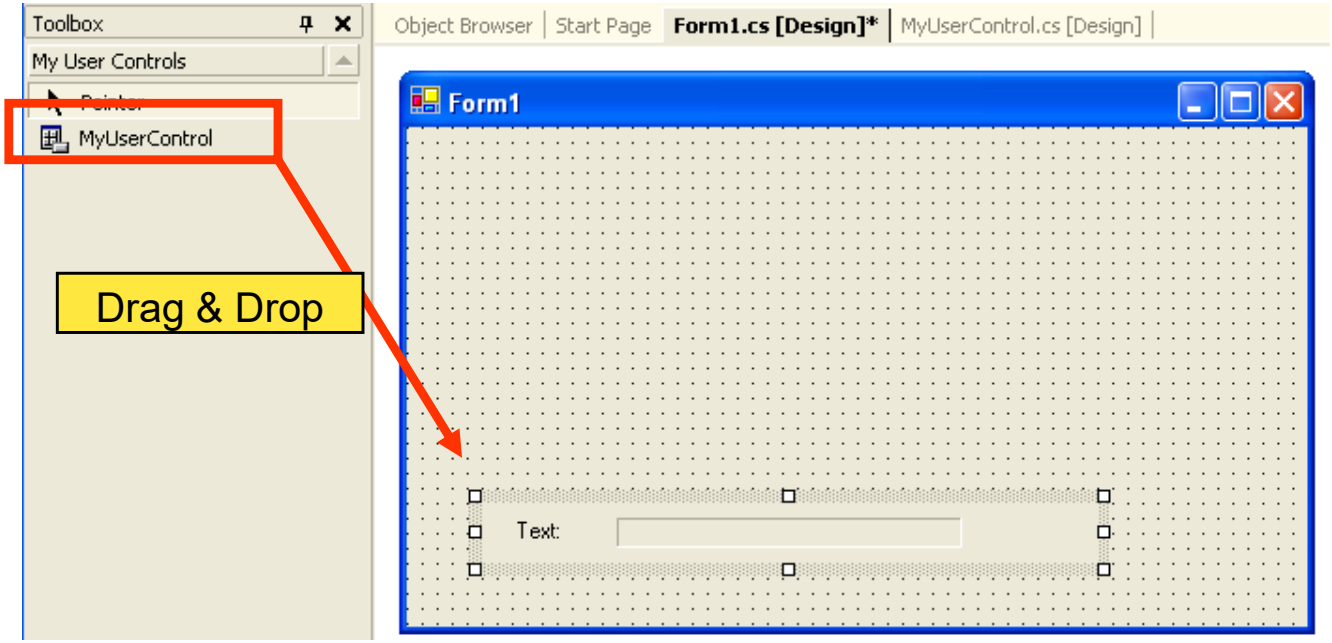
Ein neues User Control erstellen



Windows Forms Controls
Timer Control

Mein User Control erstellen:





Übung 8.3

User Control: Uhrzeit mit Fortschrittsbalken

- 1) Erstelle ein **User Control** mit einem **TextLabel** und einer **ProgressBar**.

  Das User Control soll in der Hauptanwendung benutzt werden, um die **Stunden, Minuten und Sekunden** der aktuellen Zeit anzuzeigen.

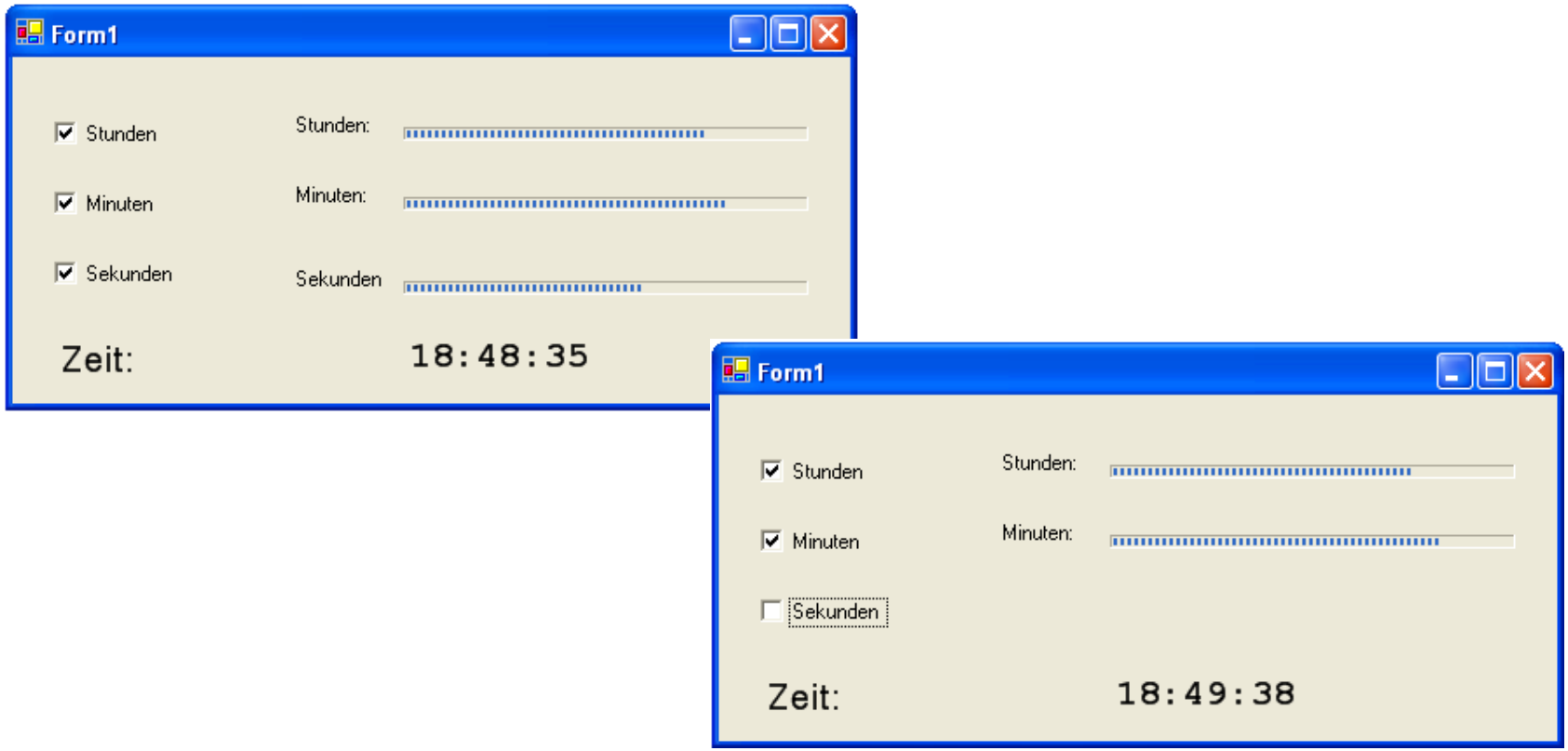
- 2) Benutze das **Timer Control**, um die Anzeige im Sekundentakt zu aktualisieren.

- 3) Die ProgressBars sollen mit den entsprechenden **CheckBoxen** einzeln aktiviert resp. deaktiviert werden können.

Übung 8.3

User Control: Uhrzeit mit ProgressBar

Musterlösung



Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU