Window Presentation Foundation

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- http://www.miwilhelm.de
- Raum 2.202
- Tel. 03943 / 659 338

·Inhalt

- **Einführung WPF**
- Layouts
- C# Sprache
- Dialog-Elemente, Menüs
- 2D- / 3D-Grafik, Audio, Video, Animation
- Routet Events, Dependency Properties, Command
- Textdarstellung (Flow-FixedDocuments)
- Datenbanken
- Navigation / Browser
- Eigene Komponenten

Überblick

- **C#** ist das Äquivalent zu Java, aber auch eine Weiterentwicklung der Sprache C/C++, keine Pointer, (".", "::" und "->") nun .
- Plattformunabhängig, eher im Sinne MS Desktop, PDA, Web

- **.net**
- Common language runtime (CLR), mit Garbage Collection
- Common language subset (CLS), gemeinsame Sprachbasis. Es kann jede Sprache verwendet werden. Standardvokabular (Menge von Befehlen)
- Microsoft intermediate language (MSIL), Zwischensprache
- Just in Time Compiler (JIT)
- Windows Forms (Swing)
- ASP.net (Web Services)
- ADO.net (Datenbanken)
- XML, SOAP, UDDI (Kommunikation zw. den Komponenten

Visual Studio, C# und .net

- Entwickler: Andreas Hejlsberg
- Heilsberg hatte Turbo Pascal und Delphi mit entwickelt
- Absprung nach Redmond
- Entwicklung des Programmpaket .net / C#
- Philosophie weitgehend identisch
  - Drag & Drop der GUI
  - Propertyfenster mit Property-Methoden kein normales get/set
  - Antwort auf Java, Plattform unabhängig
  - Sprache C#
  - Aktuelles Framework 4,5
  - Common Language Runtime (CLR)
  - Unterschied: Übersetzung, kein Intepreter,
  - Übersetzen VOR oder während der Ausführung

Visual Studio, C# und .net

- Weitere Eigenschaften
  - Unterstützt viele Programmiersprachen (C#, VB, Delphi)
  - C#, VB, Delphi, C++
  - Umwandlung in einem Zwischencode
  - Sehr viele GUI-Elemente, bis zu Listview / Grid
  - Datenbank-Anbindung
  - Web-Server
  - Delegates statt Funktionspointer
  - Operator Überladen, nicht in Java
  - Eigenschaften (readX, writeX, get/set)
  - Alles Objekte, auch int und double
  - Anweisungen weitgehend identisch zu Java / C++
  - keine Header-Dateien
  - statt implement verwendet man using
  - Mehr Datentypen, z. B. unsigned int
  - Arrays mittels Blockstruktur, anders als in Java

Visual Studio, C# und .net

- Datentypen in C#

- byte	vorzeichenlos, 0 bis 255
- sbyte	vorzeichenbehaftet, -128 bis +127
- short	vorzeichenbehaftet, -32768 bis +32767
- ushort	vorzeichenlos, 0 bis 65535
- int	vorzeichenbehaftet, -2.147.483.648 bis + 2.147.483.648
- uint	vorzeichenlos, 0 bis 4.294,967,295
- long	vorzeichenbehaftet -2^{63} bis $2^{63}-1$
- ulong	vorzeichenlos 0 bis bis $2^{64}-1$
- single	32 Bit Gleitkommazahl, 7 Stellen
- double	64 Bit Gleitkommazahl, 16 Stellen
- bool	boolscher Wert
- char	Zeichen
- decimal	96 Bit Dezimalwert
- string	readonly Zeichenfolge, neu erzeugen oder StringBuilder

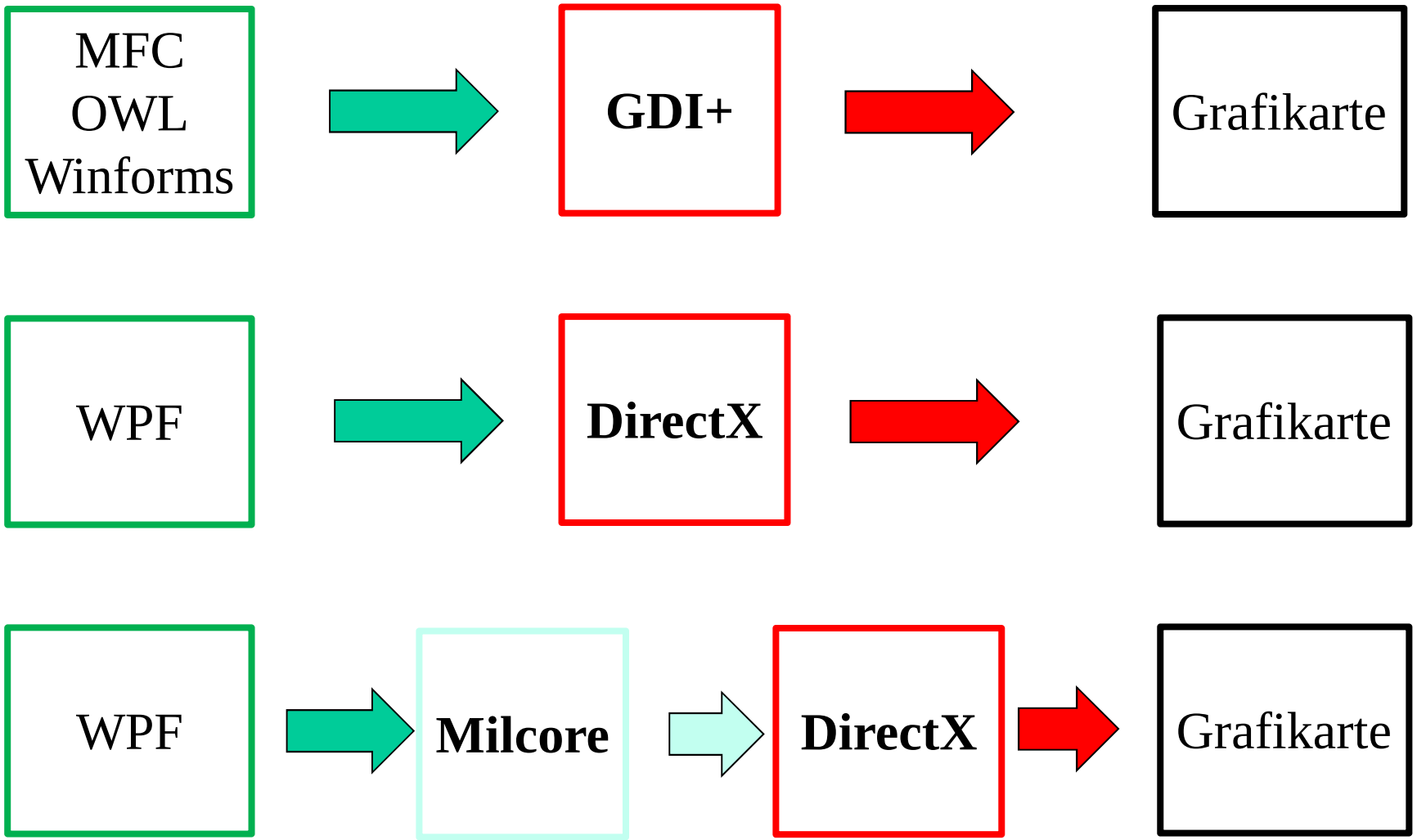
Windows-Programmierung

- Windows 1,0 bis Windows 8,0
  - ca. 4000 API-Funktionen
  - C-Funktionen
  - 150 Zeilen „Hello World“, pures API, 1985
  - Wrapper-Klassen: MFC, Borlands „Object Window Library“, WinForms
  - „Hello World“ nun „eine“ Zeile
  - Die GUI-Elemente verwenden den „Immediate Mode“
  - Jedes Zeichnen hat eine unmittelbare Auswirkung auf die Darstellung
  - Das Betriebssystem, Windows, benachrichtigt die GUI-Elemente, wenn sie neu gezeichnet werden sollen
  - Jedes Element hat eine „onPaint“-Methode, siehe auch Java
  - Jedes Element kann nur in seinem Bereich zeichnen
  - Win 7: XP-Modus à la onPaint
  - Win 7: Normal-Modus: Transparente Darstellung, Kenntnis über andere UI

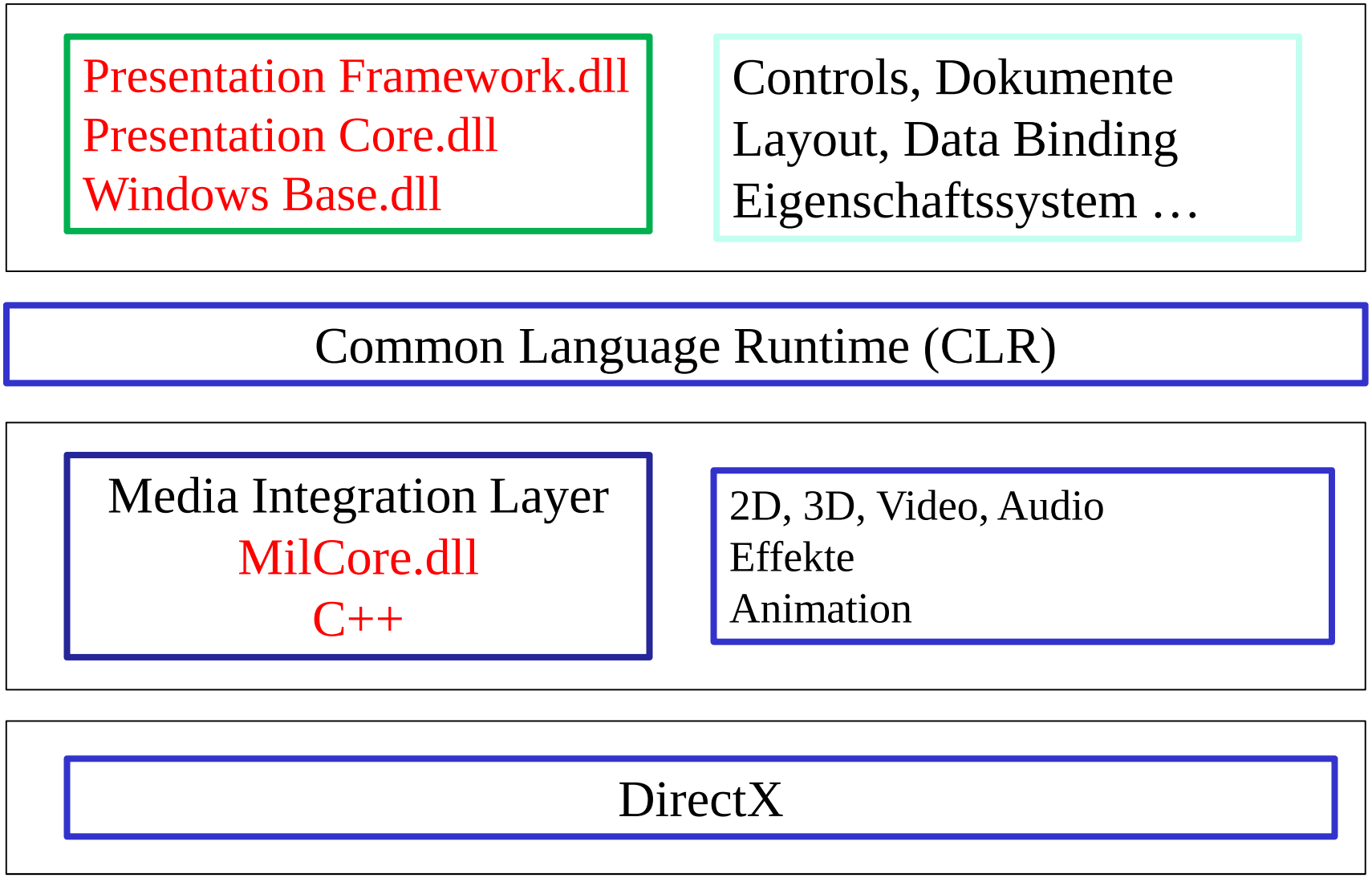
Windows-Programmierung

- Windows Presentation Foundation
  - **Radikaler Umbruch**, kein GDI, kein GDI+
  - Verwendet über die „Milcore.dll“ **DirectX** bzw. die Grafikkarte
  - WPF wurde in C# entwickelt, Milcore in C++
  - „Alle“ GUI-API-Funktionen werden umgangen
  - Trennung Code und Layout (C# und XAML)
  - Zeichnet sich selber
  - Button kann Elemente enthalten
  - viele Layout-Panels
  - Styles
  - Flexible Trigger (Routed-Events)
  - mächtige DataBinding
  - 2D- und 3D-Grafiken, Animationen, Audio und Video
  - Text und Dokumente (FlowDocument, Fixed-Document)
  - Browsertechnik, neue Navigation
  - Vectorbasierend

Windows-Programmierung



WPF-Aufbau


Presentation Framework.dll
Presentation Core.dll
Windows Base.dll
Controls, Dokumente
Layout, Data Binding
Eigenschaftssystem …
Common Language Runtime (CLR)
Media Integration Layer
MilCore.dll
C++
2D, 3D, Video, Audio
Effekte
Animation
DirectX

WindowsBase

- WindowsBase.dll
  - Enthält die Basislogik für Windows-Anwendungen, alles außer GUI
  - Ist in C# geschrieben
  - Bietet die grundlegenden Methoden (main, Event-Loop)
  - Bietet die Technik „Dependency Properties“
  - Bietet die Technik „Routed Events“
  - PresentationCore benutzt windowsbase.dll
  - PresentationFramework benutzt windowsbase.dll

Presentation Core

- PresentationCore.dll
  - Enthält die Verbindung „Visual Tree“ auf .net und „Composition Tree“ für milcore.dll
  - „Visual Tree“ beeinhaltet alle GUI-Elemente des Fensters
  - Verwendet die Klasse „Visual“
  - Visual ist die Basisklasse aller WPF-UI-Elementen
  - Der „Composition Tree“ ist die Analogie in Richtung „DirectX“
  - Der „Composition Tree“ besteht aus mehr Elementen

Presentation Framework

- PresentationFramework.dll
  - Enthält alle „WPF-GUI-Elemente“
  - Controls, Dokumente, Layout-Panels, Benutzerführung, Animation
  - Audio, Bildern und Video-Klassen
  - Ist **die** „Kern-DLL“ für „Windows Presentation Foundation“
  - Bietet also ein UI-Framework

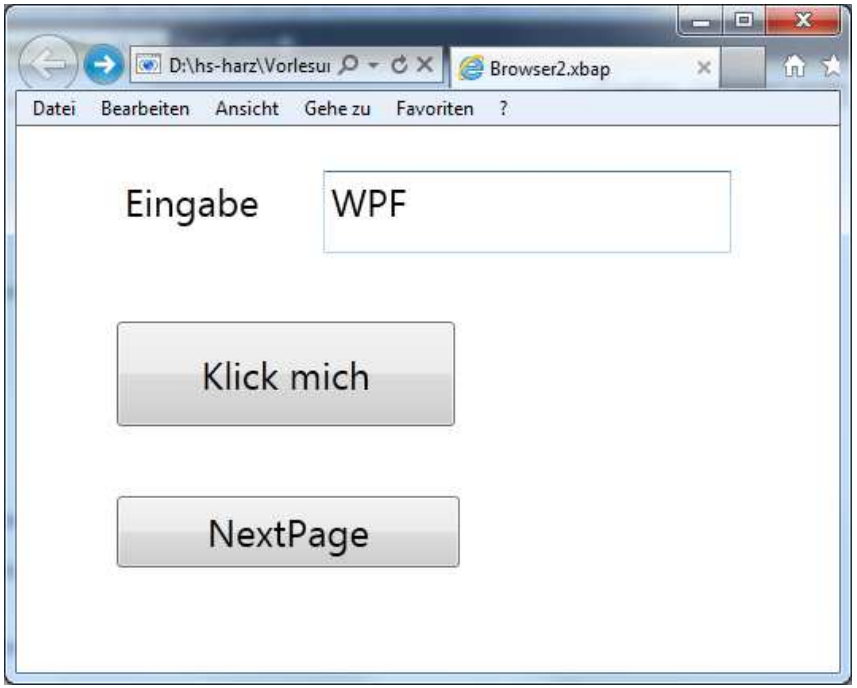
MilCore (Media Integration Layer)

- Milcore.dll
  - Kapselt den Zugriff auf DirectX
  - Alle Darstellungen der WPF wurden hier programmiert
  - Entwickelt in native Code (C++)
  - Vektorbasierende Darstellung

Eigenschaften von WPF

- Zeichnet mit Hilfe von DirectX, volle Ausnutzung der Hardware
- Komplett neue Bibliothek, kein GDI
- Keine onPaint-Methode, die Elemente zeichnen sich selber
- Vektor-, statt Rasterbasierende Elemente
- 2D- und 3D-Elemente
- Zwischenspeicher für Zeichnungsdaten
- Animationen, Audio, Video, z. B.: Video auf einem Würfel
- Neue Layout-Technik
- Klass. Desktop- oder Browserprogrammierung
- Neue GUI-Element-Techniken
  - ListBox mit Schalter, Text, CheckBox
  - Neue Möglichkeiten mit Flow- und Fixed-Documents
- Trennung Code und Layout (XAML)
- Einbau von Styles à la CSS und Templates
- Routed Events, Dependency Properties, Commands

Navigationsprogramm mit WPF



Nur mit IExplorer lauffähig
Mozilla PlugIn deaktiviert

```
this.NavigationService.Navigate(new
Uri("SecondPage.xaml", UriKind.Relative));
```

GUI-Elemente

- **Date Controls:**
  - Calendar
  - DatePicker
- **Textcontrols:**
  - DocumentViewer — Fließ- und festen Text
  - InkCanvas
  - TextBlock — Label mit WordWrap
    TextTrimming (Word, Char)
    Kann Bilder enthalten
  - TextBox(Base)
  - PasswortBox
  - RichtextBox

GUI-Elemente: ItemsControls

- Button
- RepeatButton System.Windows.Controls.Primitives;
  - Kein PreviewMouseLeftButtonDown
- ToggleButton System.Windows.Controls.Primitives;
- CheckBox
- ComboBox
- DataGrid
- Label
- ListBox
- ListView
- Menu
- RadioButton
- StatusBar
- TreeView
- WebBrowser

GUI-Elemente

- **RangeControls:**
  - ProgressBar
  - Slider
  - ScrollBar — separater ScrollBar (Vertik, Horiz.)
  - ScrollViewer — Flächenelement mit Scrollbars

GUI-Elemente

- **ContentControls:**
  - ContentControl à la div in HTML
  - Expander aus HTML, AJAX
  - Frame mit Navigation à Browser
  - GroupBox
  - Separator Splitter, à la Explorer
  - SinglePageViewer à la TabControl mit Zoom
  - TabControl

GUI-Elemente

- **Sonstiges:**
  - HyperLink
  - Image
  - Shape — Ellipse, Line, Path, Polygon, Polyline, Rectangle (meist in Canvas)
  - Decorator — Rand etc.
  - Adorner — Drag&Drop, Eingabevalidierung
  - ToolTip

  - WindowsFormsHost — Ein Element, mit dem Sie ein Windows Forms-Steuerelement auf einer WPF-Seite hosten können.

Layout-Elemente (kein Anchor mehr vorhanden)

- **Elemente:**

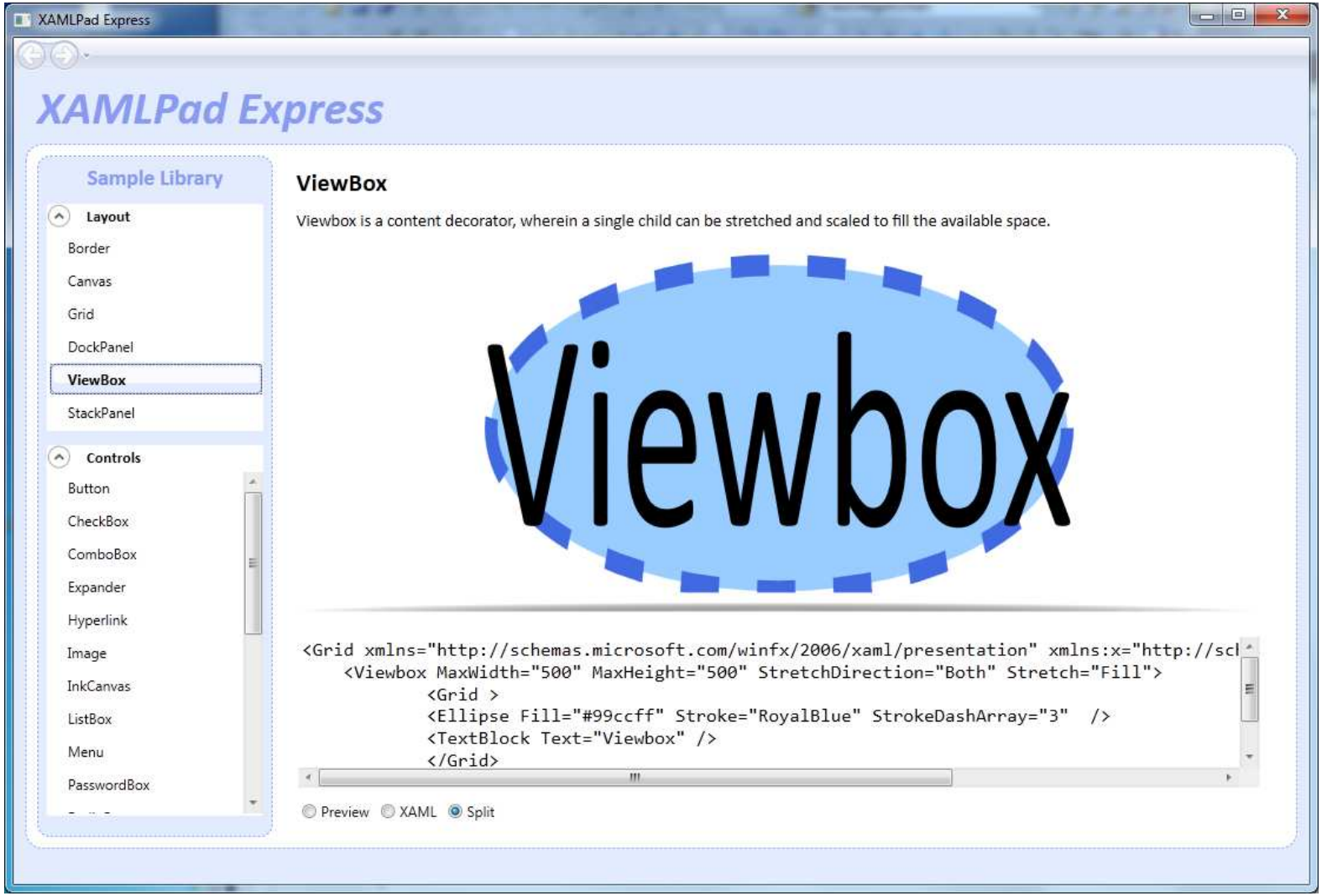
Canvas	Position mittels x/y, setLayout(null);
DockPanel	andocken an Elemente, letzte Fill
Grid	GridBagLayout in Java
UniformGrid	Grid in Java
StackPanel	Orientation="Horizontal"
	Orientation= "**Vertical**"
ViewBox	Ein Element, aber mit Zoom
WrapPanel	FlowLayout
VirtualizingStackPanel	Scrolling sichtbarer Elemente (ListBox)
DataGridCellsPanel	Layout für DataGrid-Element

Layout-Elemente

Hilfselemente:

- Border — Rand
- GridSplitter — Ändern der Breite einer Zelle

XAMLPad Express: Link auf meiner Homepage



Doppelklick: SdkXamlBrowser.csproj

WPF und XAML

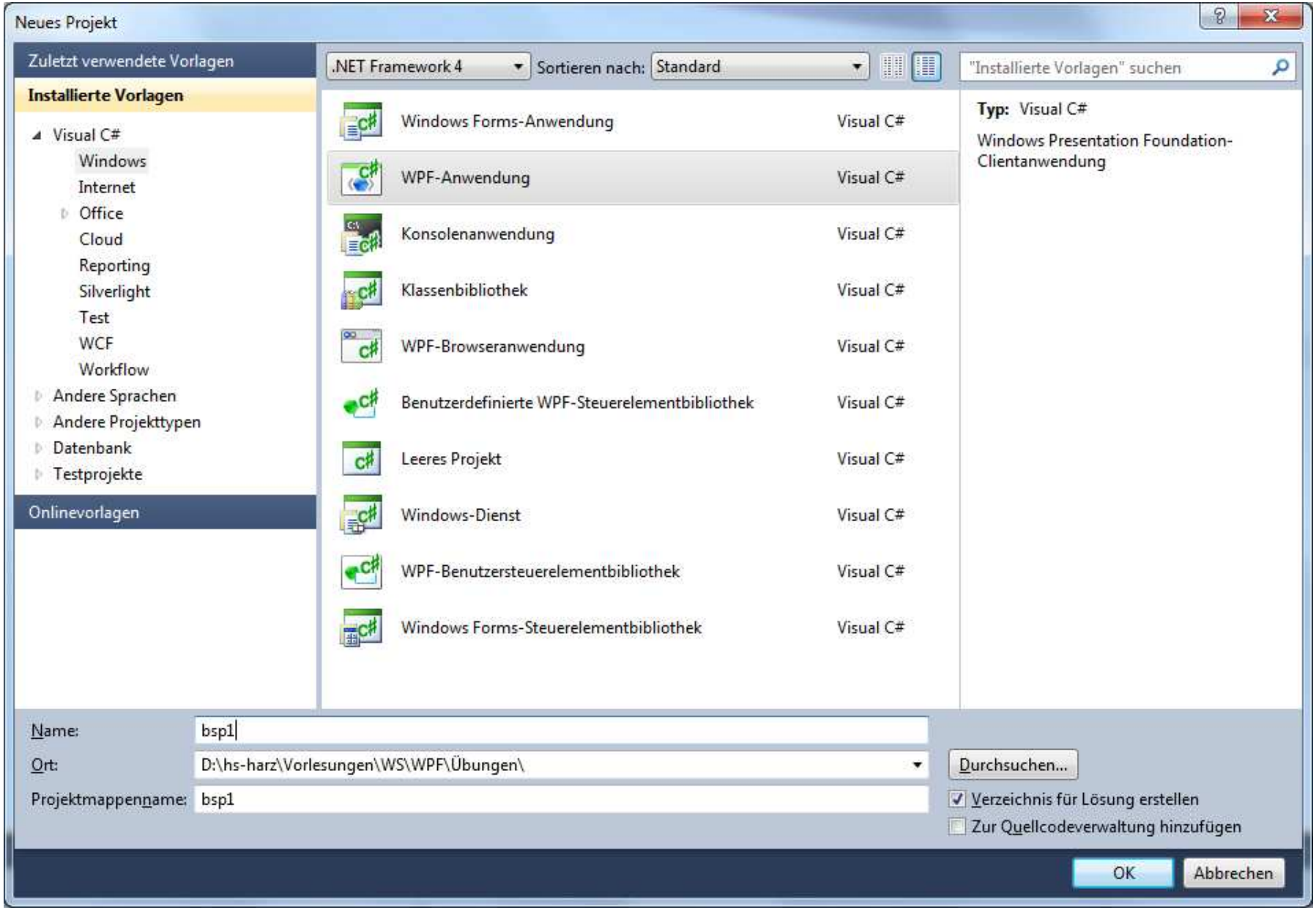
- **Winforms:**
  - Zwei partial Klassen: form1.cs, form.Designer.cs

- **WPF:**
  - Eine Klasse: form1.cs
  - Eine XAML-Datei form.xaml (optional)
  - XAML als deklarative Programmiersprache
  - Definieren, *was* gemacht wird. Nicht *wie*
- XAML
  - Alle Elemente werden in einem Baum dargestellt (DOM)
  - à la XHTML
  - Mit Attributen

XAML: 1. Beispiel, Visual Tree

```
<Window x:Class="bsp1.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title=„1. Beispiel"  Height="224"  Width="525"  Margin="1,25,1,1" >

    <StackPanel Height="175" HorizontalAlignment="Left" Margin="10,10,0,0"
            Name="stackPanel1" VerticalAlignment="Top" Width="465">

      <Label Content="Label" Height="31" Name="label1" Width="400" />
      <TextBox Height="24" Name="textBox1" Width="400" />
      <Button Content="Button" Height="37" Name="button1" Width="400"
                              Margin="0,50,0,0" Click="button1_Click" />
    </StackPanel>

</Window>
```

Margin: Left, Top, Right, Bottom


Neues Projekt
Zuletzt verwendete Vorlagen
Installierte Vorlagen
Visual C#
Windows
Internet
Office
Cloud
Reporting
Silverlight
Test
WCF
Workflow
Andere Sprachen
Andere Projekttypen
Datenbank
Testprojekte
Onlinevorlagen
.NET Framework 4
Sortieren nach:
Standard
"Installierte Vorlagen" suchen
Windows Forms-Anwendung
Visual C#
WPF-Anwendung
Visual C#
Konsolenanwendung
Visual C#
Klassenbibliothek
Visual C#
WPF-Browseranwendung
Visual C#
Benutzerdefinierte WPF-Steuerelementbibliothek
Visual C#
Leeres Projekt
Visual C#
Windows-Dienst
Visual C#
WPF-Benutzersteuerelementbibliothek
Visual C#
Windows Forms-Steuerelementbibliothek
Visual C#
Typ: Visual C#
Windows Presentation Foundation-Clientanwendung
Name:
bsp1
Ort:
D:\hs-harz\Vorlesungen\WS\WPF\Übungen\
Durchsuchen...
Projektmappenname:
bsp1
Verzeichnis für Lösung erstellen
Zur Quellcodeverwaltung hinzufügen
OK
Abbrechen

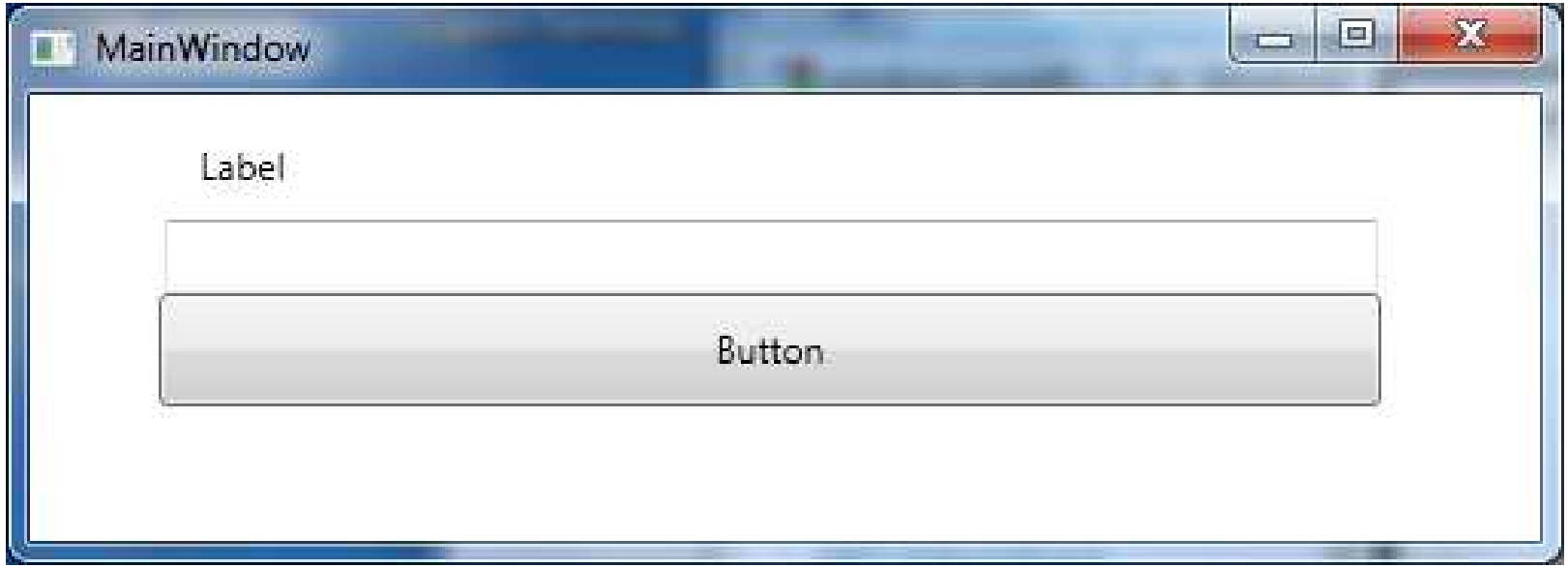
XAML: 1. Beispiel, Visual Tree

Leerer Rumpf:

```
<Window x:Class="bsp1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="MainWindow" Height="350" Width="525">

   <Grid>

   </Grid>
</Window>
```

XAML:

- Beispiel Eingabe eines Textes
- Ausgabe des Textes



XAML und Attribute

- **<Typname Attribut="Value" />:**
  - Ersetzt durch
  - <Typname>
    - <Typname.Attribut>
      - Value
    - </Typname.Attribut>
  - </Typname>

XAML: Attribute mit Typkonvertierung

- **<Grid Margin="10,20,30,40" />**

  - Ersetzt durch

  - <Grid>
    <Grid.Margin>
    <Thickness Left="10" Top="20" Right="30" Bottom="40"/>
    </Grid.Margin>
  - </Grid>

    - **Vereinfachung muss man in eigene Klassen einbauen**
    - **Get und set-Methoden werden weitergeleitet**

XAML: Markup Extension

- **Anbindung an eine Datenbank**

– Die Attribute erhalten eine Referenz auf ein beliebiges Objekt
– <TextBlock Text="{Binding Path=Name}" … />

- **Beispiel**

```
<DataGridTextColumn
    x:Name="iDColumn" Binding="{Binding Path=ID}"
    Header="ID"  Width="*" IsReadOnly="True" />

<DataGridTextColumn
    x:Name="nameColumn" Binding="{Binding Path=Lastname}"
    Header="Lastname" Width="3*" />
```

XAML: Markup Extension

C# Sprache: **Literatur und Links**

Softwareentwicklung mit C#
Hanspeter Mössenböck
dpunkt.Verlag
ISBN 3-89864-406-5

Visual C+ 2005
Günter Born, Benjamin Born
Entwickler.press
ISBN 978-3-939084-40-2

- http://www.guidetocsharp.de
- http://msdn.microsoft.com/de-de/library/kx37x362.aspx

WPF: Literatur

WPF und XAML
Rainer Stropek, Karin Huber
entwickler.press
ISBN 978-3-939084-60-0

Windows Presentation Foundation
Thomas Claudius Huber
Galileo Computing
ISBN 978-3-8362-1538-1

Windows Presentation Foundation
Adam Nathan
SAMS-Verlag
ISBN 0-672-32891-7