

C# Programmierung

Eine Einführung in das .NET Framework

Tag 5

Windows Forms

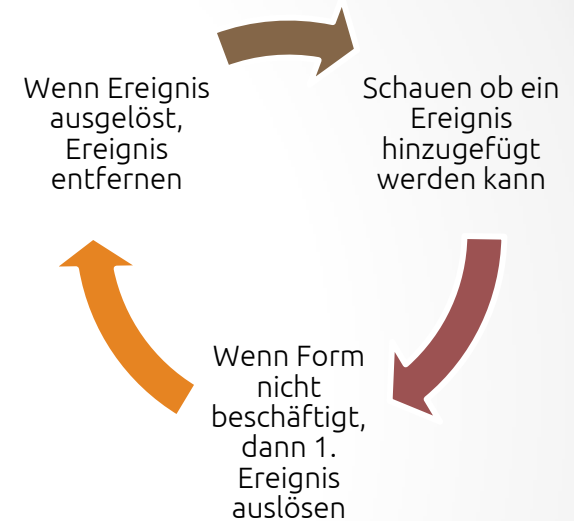
Fortgeschrittene Windows Forms Anwendungen mit eigenen Steuerelementen erstellen.

Möglichkeit für Fragen

- Kurze Wiederholung des Vortags
- Fragen zu den Aufgaben?
- Fragen zu der Vorlesung von gestern?
- Anmerkungen und Wünsche?

Die Application Klasse

- Blick in Program.cs
- Die Ereignisschleife
- Wichtig: `Application.DoEvents();`
- Starten der Schleife mit `Application.Run(Form);`
- Wichtige Informationen zur Laufzeitumgebung der Anwendung gewinnen (z.B. aktuelles Verzeichnis)



Beispiel

01 – Application Klasse

Die Environment Klasse

- Wichtige **Systemeigenschaften** zusammengefasst
- **Pfadangaben** zu den Ordnern des Benutzers bzw. der Programmeinstellungen
- **Hardware** (Netzwerk, Laufwerke, ...) Eigenschaften und
- Informationen zu **Software** (Windows Version, Pfadangaben, ...)

Beispiel

02 - Environment

Der Designer

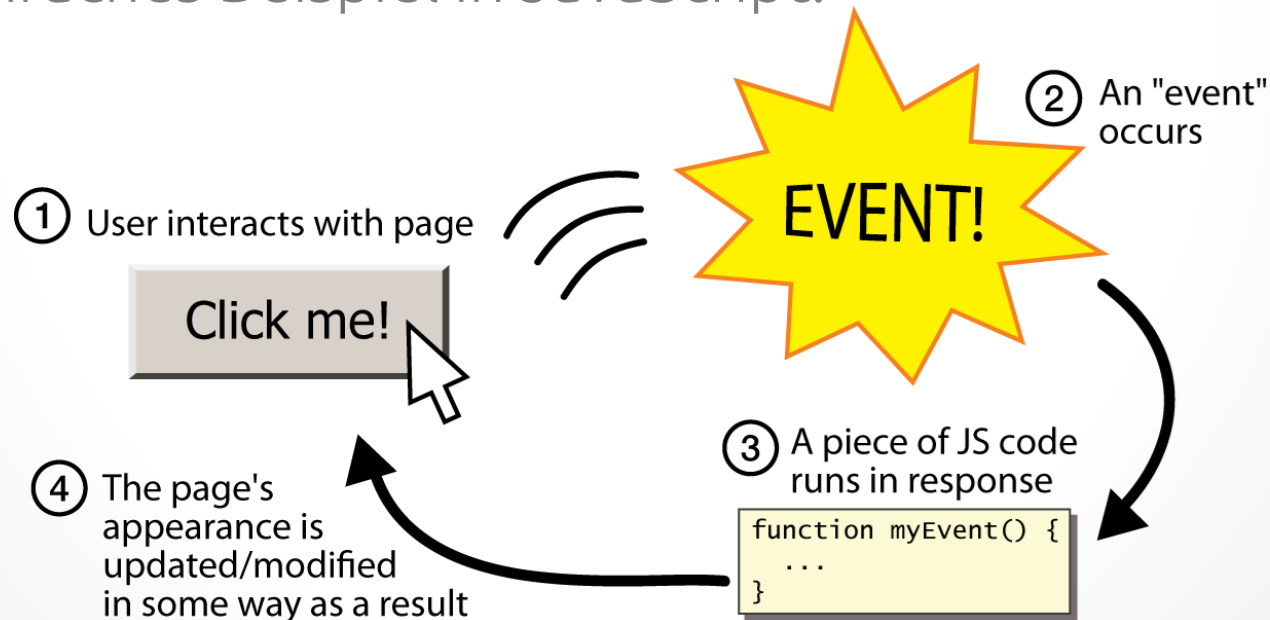
- Die **Eigenschaften** Box und die **Toolbox** und der **Projektmappen-Explorer**
- Das richtige **Layout: Anchor** und **Dock** verstehen
- Listen bearbeiten und *Ereignisse festlegen*
- Einige **Steuerelemente** in der Übersicht:
- *Radio, Checkbox, ListBox, TextBox, ComboBox* und *NumericUpDown*

Ereignisse

- Eigenschaften und Methoden von Controls und Forms allgemein
- Maus- und Tastaturereignisse festlegen
- Einige **Steuerelemente** per *Designer* hinzufügen
- Ein **Steuerelement** per *Hand* dem Formular hinzufügen
- Was wann bevorzugen? { Designer, Manuell }

Wie... Ereignisse?!

- Ereignisse verbinden (z.B.) die grafische Benutzeroberfläche mit unseren Methoden
- Ein einfaches Beispiel in JavaScript:



Die MessageBox

- Nützliche Klasse um kleine **Dialoge** zu erstellen oder Informationen an den Nutzer weiter zu geben
- Nutzerinteraktion möglich: *DialogResult*
- Die MessageBox im Detail – Überladungen
- Unterschied zwischen **Modalen** und **nicht Modalen** Dialogen [dies ist sehr wichtig!]

Beispiel

03 – Ereignisse, MessageBox

Eigene Dialoge

- Ein Form aus einem Form heraus aufrufen
- Unterschied zwischen *Show()* und *ShowDialog()*
- Das Form entsprechend anpassen (*FormBorderStyle*, *ShowInTaskbar*, ...) und auf Werte zugreifen
- Ein Form in einem Form aufrufen (MDI Forms erstellen)

Bitflags

- Wichtige Form-Methode: `SetStyle(style, bool);`
- Die Enumeration `style` ist eine sog. Bitflag Enumeration, d.h. ihre Werte sind nur 1-bit lang, z.B. 1, 2, 4, 8, 16, ... und können somit kombiniert werden (z.B.: $23 = 16+4+2+1$)
- Erklärung von Bitflags (Attribut: `[Flags]`) – Verkettung über `|` und Auslesen ob eine Flag gesetzt über `&`:

```
if((example & ExEnum.Test) == ExEnum.Test)
```

Beispiel

04 – Dialoge und Bitflags

Eigene Steuerelemente

- Analog zu eigenen Formularen (erben von *Form*) wird hier von *Control* geerbt
- Weitere Möglichkeiten: **UserControl**, **Component**
- Unterscheidung nur in Performance und Modifikationsmöglichkeiten – Beste Wahl ist meistens wenn man von *Control* oder Tochter davon erbt
- Bsp: Erstellen eines **BlinkLabels** wie im Skript

Beispiel

05 – Eigene Steuerelemente

Abschließendes Beispiel

- Eigenes Steuerelement, das Systeminformationen anzeigen soll
- Mögliche Informationen in **Bitflag**-Enumeration gruppieren (zum Aus- und Einlesen)
- Für Designer und IntelliSense Informationen hinzufügen