



C# Programmierung

Eine Einführung in das .NET Framework

Tag 3

•••

OOP erweitern und WinForms

OOP Kenntnisse ausweiten und Anwendung davon bestaunen –
Einstieg in die WinForms Entwicklung

Möglichkeit für Fragen

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

Der Bauplan

- Möglichkeit eine Klasse so zu erstellen, dass für die Tochterklassen einige **Notwendigkeiten** entstehen
- Das **abstract** Stichworts für Klassen (keine Instanz mög.)
- **abstract** für Methoden, Eigensch. (muss impl. werden)
- Bauen eines **Interfaces** mit dem Wort *interface*
- **Statische** { Klassen, Konstruktoren und Methoden }

Beispiel

01 – Bauplan

Modifizierer

- Nochmal: Wichtigkeit der Modifizierer
- Warum sollte man **nicht** immer *public* verwenden?
- Was ist der Unterschied zwischen *internal* und *private* bzw. *internal protected* und *protected*
- Verwendung eines privaten Konstruktors – inwiefern kann dies sinnvoll sein?

Beispiel

02 - Modifizierer

Konstruktor

- Zugriff auf Basisklassenkonstruktoren mit **base**
- Zugriff auf *weitere Konstruktoren* der Klasse mit **this**
- **Destruktor** in C# vorhanden, aber (eigentlich) unnützlich – sollte auch nicht genutzt werden! (wieso? Stichwort: *Garbage Collector*)
- **Konstruktorablauf** – wie läuft das alles ab? Wer ruft wen wann auf?

Jimmy

- Wer den Konstruktor nicht sofort versteht hat noch nichts verpasst
- Der Konstruktor ist eine Methode, der Bedingungen für die Erstellung eines Objektes definieren kann
- Somit kann „Jimmy“ ein Objekt erstellen
- Konstruktor sagt ihm was er dafür benötigt



Beispiel

03 - Konstruktor

Einstieg in die Forms...

- Zuvor: *Neue Schlüsselwörter* für Klassen
- Klasse über **sealed** vor Vererbung schützen (bringt Performance und schützt vor Fehlern)
- Klasse mit **partial** als unvollständig kennzeichnen (bringt Übersicht – Teamcharakter!)
- Anwendung im .NET Framework, z.B. ist *Application* versiegelt; Designer erstellt *Form* als partielle Klasse

...mit dem Designer

- Die neuen Funktionen des Visual Studios im Überblick
- Was ist die **ToolBox**? Hier finden wir Steuerelemente!
- Wie kann ich **Steuerelemente** mit meinem Code verbinden? (Wie geht das mit Ereignissen?)
- Welche Vorteile bieten mir die Forms?
- Das wichtigste zu den Forms zusammengefasst!

Beispiel

04 – Forms Einstieg

Abschließendes Beispiel

- Bauen einer Klasse, welche von *einer* abstrakten Klasse und *mehreren* Interfaces erbt
- Einige Prototypen Funktionen entwickeln und implementieren – einige Funktionen überschreiben
- **Jeden** Konstruktor mit einer `MessageBox.Show()` Mitteilung versehen
- Instanz dieser Klasse im Form erstellen