

## Aufgabe 26

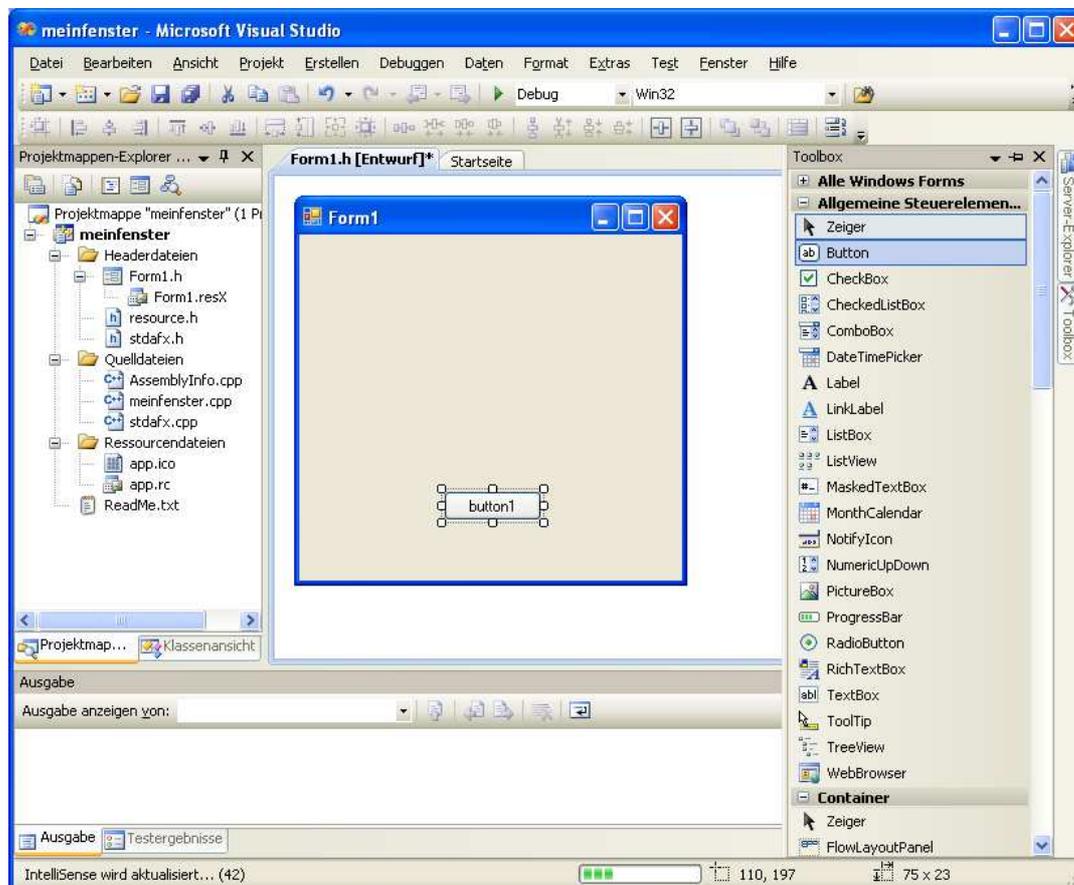
Bisher haben wir Programme für die Konsole entwickelt. Diese Aufgabe ist eine kleine Demonstration, um Ihnen einen ersten Einblick in die Windows-Programmierung zu geben.

Öffnen Sie **Visual Studio 2008** und wählen Sie unter **Datei**→**Neu**→**Projekt** im Dialogfenster erst “**CLR**” (.NET Common Language Runtime) und dann rechts “**Windows Forms-Anwendung**” aus. Nennen Sie Ihr Projekt “**meinfenster**”, klicken Sie auf **OK**.



In der Mitte sehe Sie anschließend den Entwurf Ihres Fensters. Bewegen Sie den Mauszeiger über den Reiter **Toolbox** am rechten Bildschirmrand, so dass die Toolbox ausklappt (wenn der Reiter nicht angezeigt wird, wählen Sie “**Ansicht**→**Toolbox**”).

Klicken Sie in der Toolbox auf “**Button**” und ziehen sie den Button in das Fenster “**Form1**” hinein. Sie können den Button im Fenster frei verschieben oder seine Größe ändern.



Klicken Sie anschließend doppelt auf den Button. Es wird der Quellcode von “**Form1.h**” angezeigt und der Cursor springt in die Funktion

```
System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
}
```

Diese Funktion wird aufgerufen, wenn Sie später in Ihrem Fenster auf den Button klicken. Fügen Sie zwischen die öffnende Klammer “{” und die schließende Klammer “}” des Funktionsrumpfes die Zeile

```
MessageBox::Show("Hallo Welt!", "Achtung!!",
    MessageBoxButtons::OK,
    MessageBoxIcon::Exclamation);
```

ein, d.h. Sie erhalten

```
System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    MessageBox::Show("Hallo Welt!", "Achtung!!",
        MessageBoxButtons::OK,
        MessageBoxIcon::Exclamation);
}
```

Bedeutung: Beim Aufruf von `MessageBox::Show()` wird eine Dialogbox auf dem Bildschirm angezeigt. Klicken Sie links im Projektmappen-Explorer auf “**Form1.h**”, um das Fenster mit dem Button wieder anzuzeigen.

Übersetzen und starten Sie Ihr Programm durch Klicken auf .

## Aufgabe 27

Betrachten Sie wieder das Fenster “Form1” mit dem Button “button1”. Klicken Sie mit der rechten Maustaste auf den Button und wählen Sie “Eigenschaften” aus. Es werden die Eigenschaften des Buttons angezeigt. Ändern Sie die Eigenschaft “Text” von “button1” auf “Klick mich!”. Starten Sie Ihr Programm erneut.

## Aufgabe 28

Gehen Sie mit dem Cursor ganz nach oben in der Datei `Form1.h` und ergänzen Sie in den ersten Zeilen der Datei

```
#include <cmath>
#include <string>
#include <sstream>
```

```
using namespace std;
using namespace System;
```

Die alternierende harmonische Reihe

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} \dots$$

konvergiert gegen  $\ln(2) \approx 0.69314718055995$ .

Fügen Sie nun Programmzeilen in die Funktion `button1_Click()` ein, die die Reihe bis zum 1000sten Summanden berechnet und das Ergebnis in der Variablen  $d$  abspeichern.

Mit den folgenden Zeilen können Sie das Ergebnis  $d$  in einer `MessageBox` ausgeben. Dabei wird das `sout`-Objekt analog zum bekannten `cout`-Objekt verwendet. Leider geht es in diesem Fall nicht auf weniger komplizierte Weise. Ignorieren Sie die sehr unelegante Syntax.

```
stringstream sout;
sout.precision(14);
sout<<"Ergebnis: "<<d<<" ist fast "<<log(2.);

String^ Str=gcnew String(sout.str().c_str());

MessageBox::Show(Str, "Achtung!!",
    MessageBoxButtons::OK,
    MessageBoxIcon::Exclamation);
```

Wieviele Summanden brauchen Sie, um  $\ln(2)$  bis auf 10 Stellen genau auszurechnen. Handelt es sich um ein gutes Verfahren? Vergleichen Sie mit Aufgabe 16.