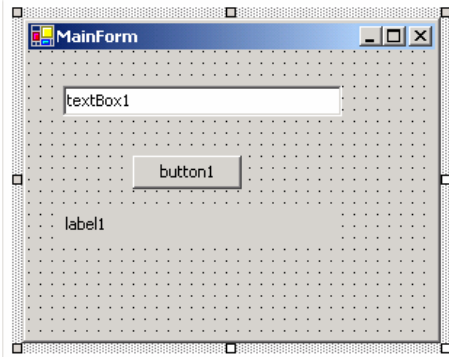


Zuweisungen

Beispiel: Ausgabe eines Textes in einem Label

Erstelle ein neues Programm. Wähle den grafischen Formulardesigner aus. Ziehe einen Button, ein Label und eine TextBox auf das Formular und ordne sie sinnvoll an, z. B.:



Ändere den Text von button1 in „zeige“ und erzeuge wieder eine Ereignisbehandlung. Trage folgendes ein:

```
label1.Text = textBox1.Text;
```

Damit wird der Eigenschaft Text von label1 der Wert der Eigenschaft Text des Objektes textBox1 zugewiesen.

Das Gleichheitszeichen in der Programmierung mit C# hat nicht die aus dem Alltag (bzw. aus der Mathematik) gewohnte Bedeutung. Das Gleichheitszeichen hat hier eine ähnliche Bedeutung wie bei Zahlenfolgen. Die Eigenschaft auf der linken Seite erhält den Wert des Ausdrucks auf der rechten Seite.

Man spricht in der Informatik von einer **Zuweisung**. Bedeutend bei einer Zuweisung ist, dass immer zunächst der Ausdruck auf der rechten Seite berechnet wird und erst dann das Ergebnis auf die linke Seite übertragen wird.

Teste das Programm.

Hinweis: Normalerweise sollte man vor einer TextBox ein Label setzen, in dem steht, was man in der TextBox eingeben soll. Die TextBox kann einen sinnvollen Anfangswert enthalten.

Zum Beispiel steht dann im Label (bzw. deren Eigenschaft Text) „auszugebender Text“ und die TextBox bleibt leer. Bitte nicht den Hinweis, was eingegeben werden soll, in die TextBox schreiben, auch wenn das manche anderen Anwendungen tun.

Aufgabe 2:

Wie kann ich dafür sorgen, dass im Label der Text nur in Großbuchstaben erscheint?

Tip: Tippe hinter `textBox1.Text` einen Punkt und untersuche die angezeigten Methoden und Eigenschaften.

Variablen

Variablen sind Behälter, in denen man verschiedene Werte speichern kann. Im Gegensatz zur Mathematik muss man in C# sofort angeben, welche Werte man überhaupt in der Variable ablegen will.

Wenn man ganze Zahlen in einer Variablen speichern will, hat die Variable den **Datentyp** `int`. Dezimalzahlen, genauer gesagt Fließkommazahlen, werden in `double` Variablen gespeichert. Zeichenketten werden im Datentyp `String` gespeichert.

Jede Variable hat außerdem einen Namen, der aus folgenden Elementen bestehen kann:

- den *Buchstaben* des Alphabets (d.h. a,b,c,...). C# unterscheidet hierbei zwischen Groß- und Kleinschreibung.
- dem *Unterstrich* `"_"`
- den *Ziffern* 0, 1, 2, . . . ,9

Variablen dürfen außerdem nicht so heißen, wie Befehle in C# und sie dürfen nicht mit einer Ziffer beginnen.

Beispiel zum Arbeiten mit Variablen:

Erstelle ein neues Programm. Ziehe einen Button und zwei Labels auf das Formular. Erzeuge eine Ereignisbehandlung für den Button und fülle sie mit folgendem Quelltext:

```
int x;  
x=3+4;  
label1.Text = x.ToString();  
label2.Text = "x = "+x.ToString();
```

Aufgabe 3:

Teste das Programm. Was wird angezeigt?

Das Programm enthält viele neue Elemente:

- Durch `int x;` wird eine Variable mit dem Namen x deklariert und damit Speicher reserviert.
- `x=3+4;` ist eine Zuweisung. In der Variablen x wird der Wert des Ausdrucks 3+4 (d. h. 7) gespeichert.
- Jede Eigenschaft hat einen Datentyp. Die Eigenschaft Text ist eine Zeichenkette (ein String). Deshalb muss man die Zahl in eine Zeichenkette umwandeln, um sie im Label ausgeben. Dies geschieht mit Hilfe der Methode `ToString()` von x.
- `"x = "+x.ToString();` demonstriert, wie man Text und eine Variable gemeinsam ausgeben kann. Der Text zwischen den Anführungszeichen wird wieder unverändert ausgegeben.

Aufgabe 4:

Ändere das Programm, so dass die Variable x andere Werte enthält. Verändere auch die Ausgabe.

Aufgabe 5:

Welche der folgenden Bezeichner sind in C# erlaubt? Warum sind die anderen nicht erlaubt?

```
Hallo_Welt, 10kleineSchokoküsse, _H_A_L_L_O_ , Das_war's,  
hallo123, hallo_123, Hallo Welt, class
```

Aufgabe 6:

Schreibe ein C#-Programm, das die Zahlen 12 und 9 addiert, subtrahiert, multipliziert und dividiert. Dabei sollen die beiden Zahlen und das jeweilige Ergebnis in einer passenden Variablen gespeichert und diese jeweils mit Hilfe von Labels ausgegeben werden.