



C#, ick liebe dir

Ich laufe nicht jedem neuen Hype hinterher.
Ich bin der Meinung: Nicht jede neue Funktion
macht glücklicher.

Früher hatte ich mal eine andere Einstellung. Da musste ich die neueste Software sofort installieren und damit herumspielen. Das hat sich inzwischen geändert. Mag es am Alter oder der Erfahrung liegen, aber lieber auf bewährte Tools setzen, als Unmengen an Zeit in Neues, Unbekanntes zu versenken.

Diese Einstellung hat sich in letzter Zeit wesentlich verschlimmert.

Umso reizender ist es dann, wenn man tatsächlich über etwas stolpert, das einem ein Lächeln ins Gesicht zaubert. So geschehen bei drei neuen C#-Features, die ich inzwischen intensiv nutze. Da sind die Namespaces, die für die gesamte Datei gelten. Statt eine zusätzliche Klammerebene aufzumachen, schreibt man:

```
namespace MeinNamespace;
```

Die Klammern entfallen und der Namespace gilt ab dort bis zum Ende der Datei. Das zweite Feature ist der `new()`-Operator. Eine neue Instanz lässt sich damit so schreiben:

```
MeineKlasse meineKlasse = new();
```

Das ist elegant, komplett aussagekräftig und spart ein bisschen Code, der ja nicht nur geschrieben, sondern auch gelesen werden müsste. Schließlich finde ich cool, dass man die Eigenschaften einer Klasse in geschweiften Klammern nach dem Konstruktor setzen und die runden Klammern weglassen kann.

```
MeineKlasse meineKlasse = MeineKlasse{ Name = "HalloWelt" };
```

Alles in allem spart das nur wenig Tipparbeit. Doch empfinde ich den Code leichtgewichtiger, weil Unnötiges oder Doppeltes gar nicht dasteht.

Wenn die lieben Sprachentwickler weiter in diese Richtung gehen, bin ich gerne bereit, neue Sprachfeatures dazuzulernen. Reduzieren von unnötigem Code hin zu einem Minimum an nötiger Syntax ist klasse – es muss ja nicht gleich in die Sprache Whitespace [1] münden.

Viel Spaß mit der dotnetpro wünscht Ihnen

Tilman Börner
Chefredakteur dotnetpro

[1] [https://de.wikipedia.org/wiki/Whitespace_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Whitespace_(Programmiersprache))



Jürgen Gutsch

blickt in die Zukunft von
ASP.NET Core der Version 7 (S. 8)



Daniel Basler

löst mit einem Codebot mathe-
matische Funktionen (S. 10)



Michael Wresche

stellt Result Objects als Alterna-
tive zu Exceptions vor (S. 48)