



# C#

Die neue Sprache für Microsoft .NET

H.Mössenböck

# Inhalt



1. Überblick
2. Symbole
3. Typen
4. Ausdrücke
5. Deklarationen
6. Anweisungen
7. Klassen und Structs
8. Vererbung
9. Interfaces
10. Delegates
11. Ausnahmen
12. Namespaces und Assemblies
13. Attribute
14. Native Calls
15. Threads
16. XML-Kommentare
17. Base Class Library
18. Generics

## Literatur:

- Mössenböck: **Softwareentwicklung mit C#**, dpunkt, 2003
- Beer, Birngruber, Mössenböck, Wöß: **Die .NET-Technologie**, dpunkt, 2002
- B.Albahari, P.Drayton, B.Merrill: **C# Essentials**. O'Reilly, 2001
- Microsoft: **C# Language Specification**. Microsoft Press, 2001
- S.Robinson et al: **Professional C#**, Wrox Press, 2001
- Referenzinformation und Tutorials auf .Net-SDK-CD



# *Überblick*

# Merkmale von C#



## Sehr ähnlich zu Java

70% Java, 10% C++, 5% Visual Basic, 15% neu

### Wie in Java

- Objektorientierung (einf. Vererbung)
- Interfaces
- Exceptions
- Threads
- Namespaces (wie Packages)
- Strenge Typprüfung
- Garbage Collection
- Reflection
- Dyn. Laden von Code
- Spezielle String-Unterstützung
- ...

### Wie in C++

- (Operator) Overloading
- Zeigerarithmetik in Unsafe Code
- Einige syntaktische Details

# *Neue Features in C#*



## Wirklich neu

- Referenzparameter
- Objekte am Stack (Structs)
- Blockmatrizen
- Enumerationen
- Uniformes Typsystem
- goto
- Attribute
- Systemnahes Programmieren
- Versionierung

## "Syntactic Sugar"

- Komponentenunterstützung
  - Properties
  - Events
- Delegates
- Indexers
- Operator Overloading
- foreach-Iterator
- Boxing/Unboxing
- ...

# Einfachstes C#-Programm



## File Hello.cs

```
using System;

class Hello {

    static void Main() {
        Console.WriteLine("Hello World");
    }

}
```

- benutzt Namespace *System*
- Hauptmethode muß immer *Main* heißen
- Gibt auf Konsole aus
- Dateiname und Klassenname müssen *nicht* übereinstimmen.

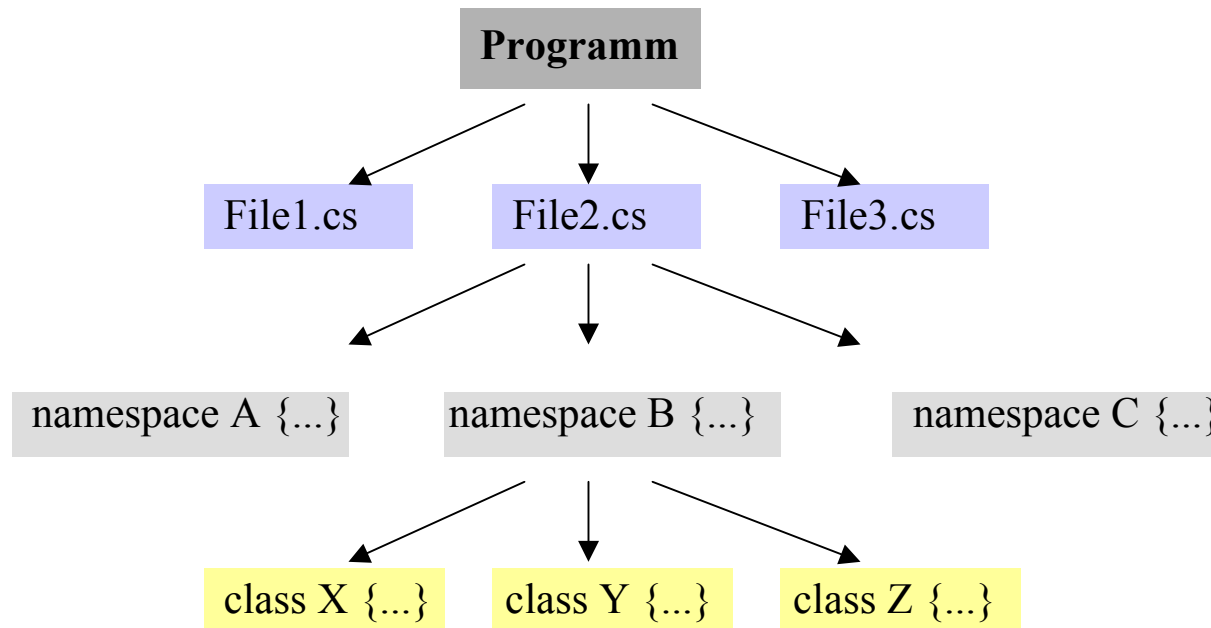
## Übersetzen (im Command-Prompt)

```
csc Hello.cs
```

## Ausführen

```
Hello
```

# Gliederung von C#-Programmen



- Wenn kein Namespace angegeben => Namenloser Standardnamespace
- Namespace kann auch Structs, Interfaces, Delegates, Enums enthalten
- Namespace kann in verschiedenen Files "wiedereröffnet" werden
- Einfachster Fall: 1 File, 1 Klasse

# Programm aus 2 Dateien



## Counter.cs

```
class Counter {  
    int val = 0;  
    public void Add (int x) { val = val + x; }  
    public int Val () { return val; }  
}
```

## Prog.cs

```
using System;  
  
class Prog {  
  
    static void Main() {  
        Counter c = new Counter();  
        c.Add(3); c.Add(5);  
        Console.WriteLine("val = " + c.Val());  
    }  
}
```

## Übersetzen

```
csc /target:exe Counter.cs Prog.cs
```

## Ausführen

```
Prog
```

## Arbeiten mit DLLs

```
csc /t:library Counter.cs  
=> erzeugt Counter.dll
```

```
csc /r:Counter.dll Prog.cs  
=> erzeugt Prog.exe
```