

# **Komponenten & Frameworks**

Das .NET Framework

WS18/19

# Einführung

## Inhaltsverzeichnis (Teil 1)

0. Einordnung in Software-Entwicklungsprozess

1. Komponenten

1.1 Definition

1.2 Komponententechnologien

2. Frameworks

2.1 Einführung

2.2 Exemplarische Frameworks

3. Das .NET Framework

3.1 Übersicht

3.2 Hello World - von der Codierung(C#) zur Ausführung

3.3 Verteilte Anwendungen in .NET (Übersicht)

## Einführung

### Inhaltsverzeichnis (Teil 2)

#### 4. Kurzeinführung in C#

##### 4.1 Grundsätzlicher Aufbau

##### 4.2 Typsystem

##### 4.3 Vererbung

##### 4.4 Delegaten

##### 4.5 Reflection

##### 4.6 Attribute

##### 4.7 Serialisierung

#### 5. Winforms, WPF

#### [6. .NET Remoting]

#### 7. ASP.NET

## Einführung

### Literaturverzeichnis

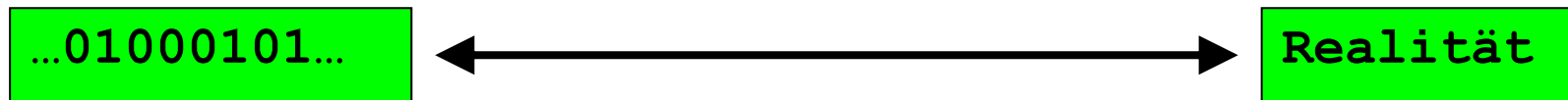
- [Buschmann] Frank Buschmann et. Al.: Pattern orientierte Software Architektur, Addison-Wesley, 2002
- [Eberhart] Eberhart/Fischer: Web Services, Hanser-Verlag  
ISBN 3-446-22530-7
- [Eberling] Eberling/Lessner: Enterprise JavaBeans3  
Hanser 2007, ISBN 3-446-41085-6
- [Freeman] E. Freeman: Head First Design Patterns,  
O'Reilly, ISBN 0-596-00712-4
- [Gamma] Erich Gamma, Entwurfsmuster, 1996,  
Addison Wesley, ISBN 3-8273-1862-9
- [Liberty] Jesse Liberty, Programmieren mit C#, O'Reilly  
ISBN 3-89721-415-6
- [MacDonald] Matthew MacDonald, Distributed Applications,  
Microsoft Press, 2003, ISBN 0-7356-1933-6

## Einführung

- [Mössenböck] Hanspeter Mössenbeck et. al: Die .NET Technologie, dpunkt , ISBN 3-89864-421-9
- [Richter] Microsoft .NET Framework Programing in C# Microsoft-Press, ISBN 3-86063-984-6
- [Rupp] Heiko W. Rupp, EJB 3 für Umsteiger, dpunkt, 2007, ISBN 978-3-89864-429-7
- [Szyperski] Clemens Szyperski: Component Software - Beyond Object-Oriented Programming, Addison-Wesley, 2. Edition, 2002, ISBN 0-201-74572-0
- [Tanenbaum] Tanenbaum, van Stehen: Verteilte Systeme Pearson Verlag ISBN 3-8273-7057-4

## Einführung

### 0. Einordnung in Software-Entwicklungsprozess



Ein Programm besteht aus einer Sequenz aus 0 und 1

⇒ Programm prinzipiell erstellbar durch Definition dieser Sequenz

Aber: fehleranfällig, unüberschaubar und praktisch nicht durchführbar

Lösungsansatz: Entwicklung geeigneter Programmiersprachen

## Einführung

### Programmiersprachen

- Assembler (Maschinensprache)
- Basic, Cobol, C, ... Lisp, Prolog
- Natural, ...
- Objektorientierte Sprachen (C++, Java, C#)

Ziel: Realität soll leichter abbildbar sein

⇒ Programmiersprachen nähern sich der Realität



Im Software-Engineering nähert man sich von der Realität ausgehend der Implementierung in Programmiersprachen.

## Einführung

### Verkleinern der Lücke

- Massnahmen im Softwareengineering
  - Werkzeuge zur Entwicklung geeigneter Realitätsmodelle
  - automatische Codegenerierung
  - Modell Driven Architecture (MDA)
  
- Massnahmen auf der Implementierungsseite
  - Weiterentwicklung von Programmiersprachen
  - Domain Specific Language
  - Bereitstellung eines erweiterbaren Rahmen
    - > „Framework“
  - Schlagwort: Entwicklung einer Anwendung durch Modellieren ohne zu programmieren
  - Bereitstellung leistungsfähiger Bausteine
    - > „Komponenten“



## Einführung

### Allgemeiner Trend

Die Entwicklungsumgebungen bewegen sich weg von dem Computer hin zur Realität.

Zielrichtung:

**Modellieren statt programmieren!**