

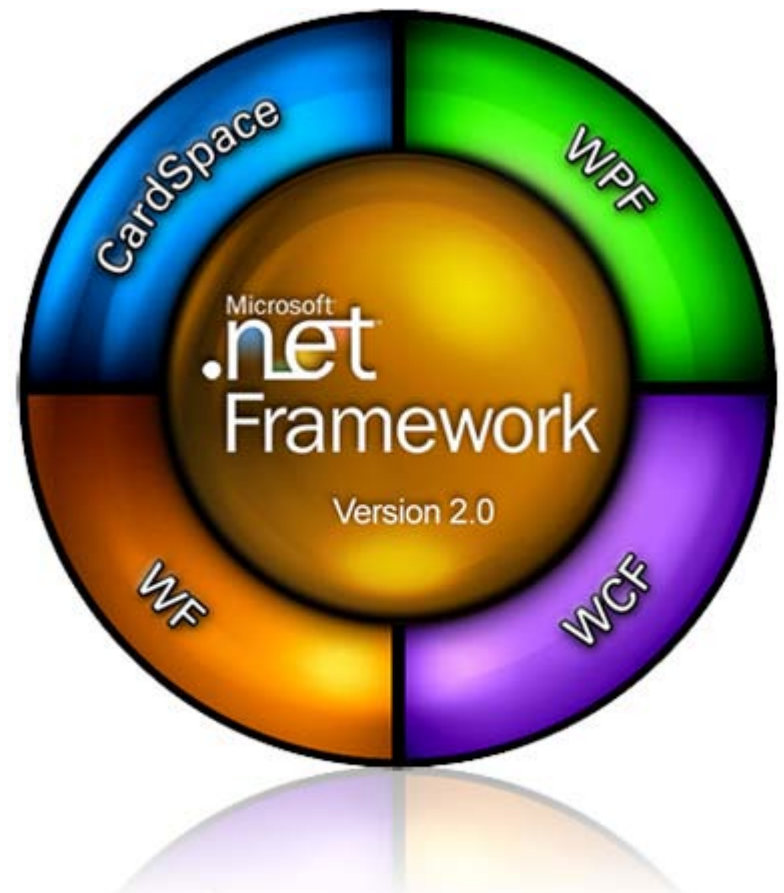
Das Wichtigste in Kürze

PROGRAMMIEREN II

.NET - Softwareplattform

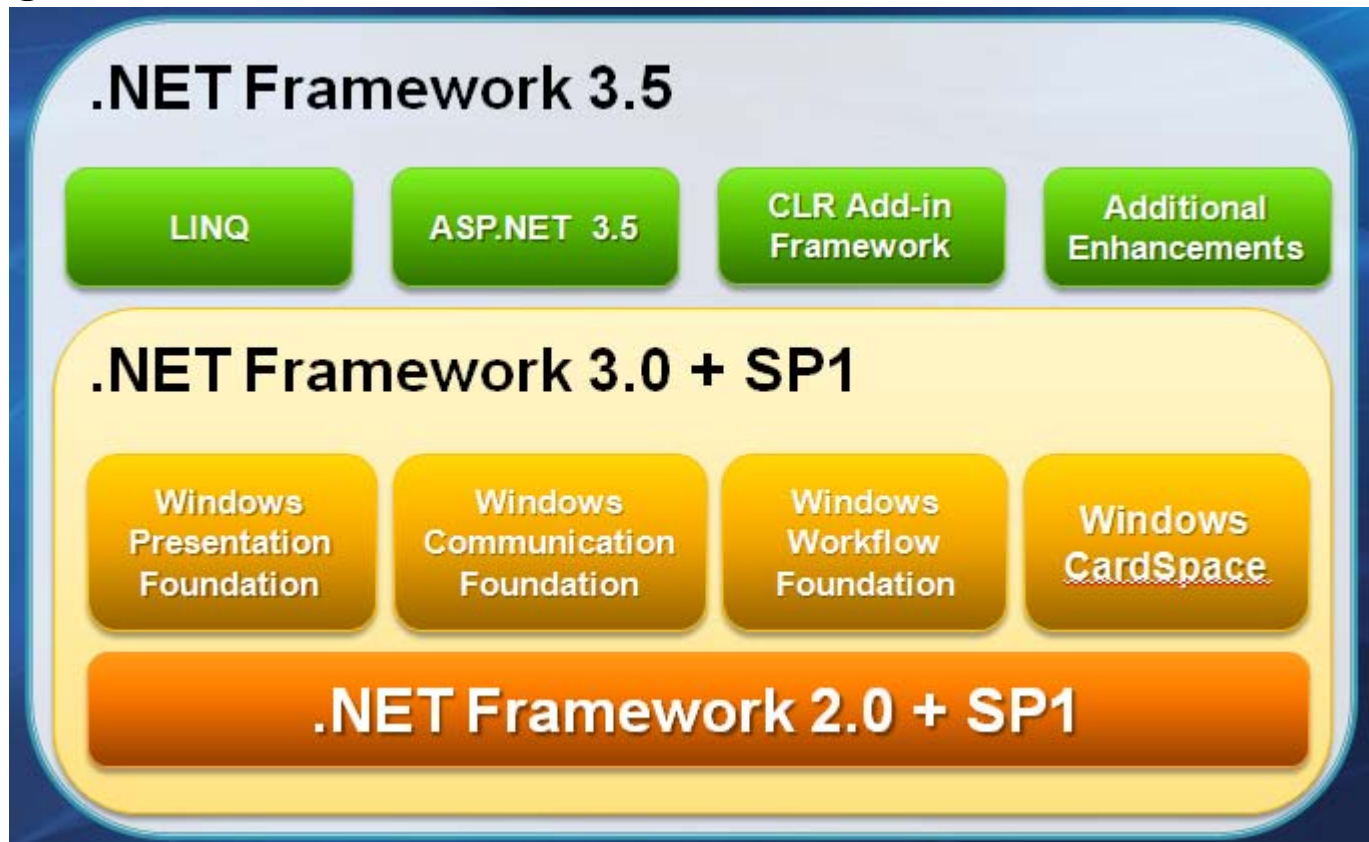
Aktuelle Version: 3.5 SP1, November 2009

Zweck: eine universale Softwareplattform, welche den Entwicklungsprozess von Programmen für verschiedene Zielgeräte erleichtert und beschleunigt.



.NET - Abstraktion

Das Framework bietet den Entwicklern abstrahierte, gesammelte Bereiche, mit welchen sich effizienter und – natürlich – objektorientierte Programmieren lässt.



.NET - Eigenschaften

Man könnte .NET als Betriebssystem unabhängig bezeichnen. Die Unterstützung für den Linux Interpreter Mono ist allerdings gering.

.NET kann in 3 Sprachen programmiert werden. C#, ASP.NET, J# - Zumal dies alle Microsoft-Sprachen sind, kann nicht von der vielgepriesenen Sprachunabhängigkeit gesprochen werden.

Die Speicherverwaltung in .NET funktioniert wie bei Java mit einem Garbage Collector, der überflüssige Speicherbelegung entfernt.

.NET ist objektorientiert.

Buch Seite 34

<http://openbook.galileocomputing.de/csharp/kap02.htm#Xxx746240>

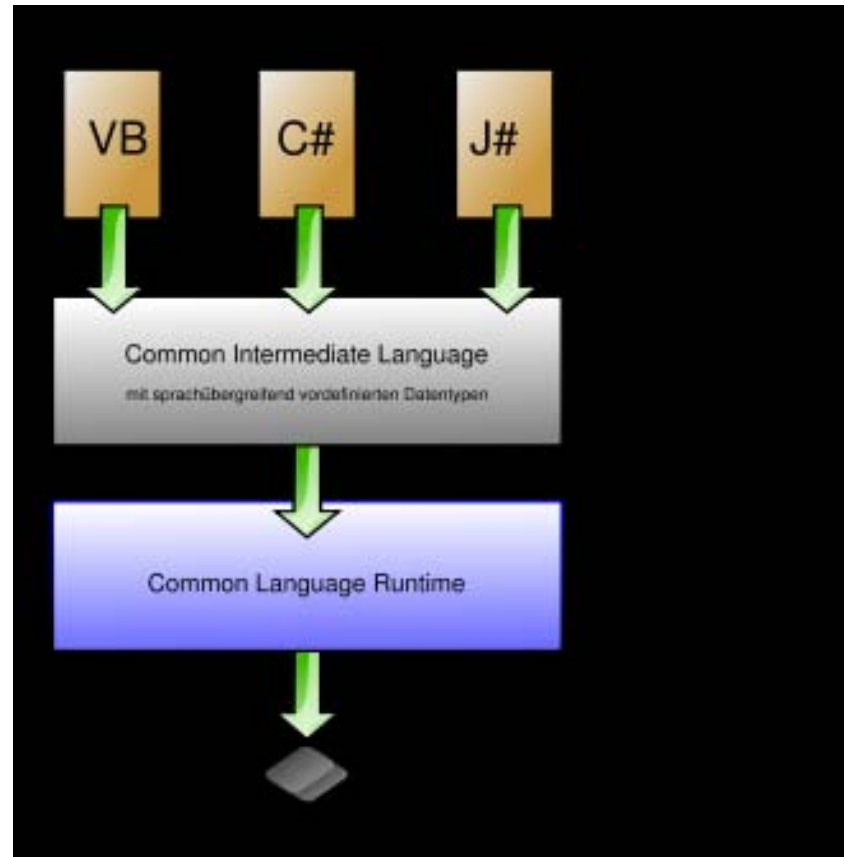
.NET - Architektur

Die Architektur teilt sich zum einen in die zur Verfügung stehenden Sprachen, welche durch die CIL zusammengefasst werden. Das CLR kompiliert danach das Präkompilat, und der Plattformunabhängige JITer übersetzt es in Maschinen Code.

Buch S. 36

[Galileo Openbook](#)

Frage: Liegen die kompilierten EXE Dateien im Maschinen- oder Bytecode vor.



.NET - Architekturvergleich

Begriff	.NET	Java
Sprachenvereiniger	CIL / CLS	Nur Java
Compiler	CLR	JDK (javac.exe)
Laufzeitumgebung	CLR	JRE
Pre-Compilat	MSIL	Bytecode
Übersetzter, Interpreter	JITer	JRE (java.exe)
(vor)Kompilierter Code	Assembly (EXE, DLL)	JAR

C# (cSharp) – Die Sprache

[.NET Poster](#)

Der Syntax von C# ist abgeleitet von C, C++ und Java. Trägt in sich aber sprachliche Erweiterungen, die helfen die .NET-Architektur vollumfänglich zu nutzen. C# ist auch objektorientiert.

C# (cSharp) – Vergleich

Wichtige Unterschiede zur Syntax

Merkmal	C#	Java
Bibliotheksreferenz	using	import
Paketierung	namespace	package
Sprachkonvention	Methoden und Eigenschaften werden mit Grossbuchstaben begonnen. E.g. Main-Methode	Methoden werden mit Kleinbuchstaben begonnen.

Visual Studio – Die IDE

Visual Studio ist Microsoft eigene Entwicklungsumgebung und vereinfacht die Erstellung von .NET Software wesentlich.

Funktionen

- Syntaxprüfung
- Kompilieren
- Dokumentationsunterstützung
- Vereinfachung von Datenanbindungen – LINQ
- GUI Programmierung
- Unterstützung für Softwareverteilung
- Projekt Management
- Einfach objektorientierte Modellierung mit UML-Klassendiagrammen

Persistenzen:: XML mit C#

- Einlesen eines XML Documents
- Auswählen mit `SelectSingleNode(String Pfad)`:
- <http://msdn.microsoft.com/de-de/library/fb63z0tw.aspx>

- Suche mit Hilfe von XML Pfaden (XPath):
- <http://de.wikipedia.org/wiki/XPath>
- <http://www.w3.org/TR/xpath>

Fehlerbehandlung

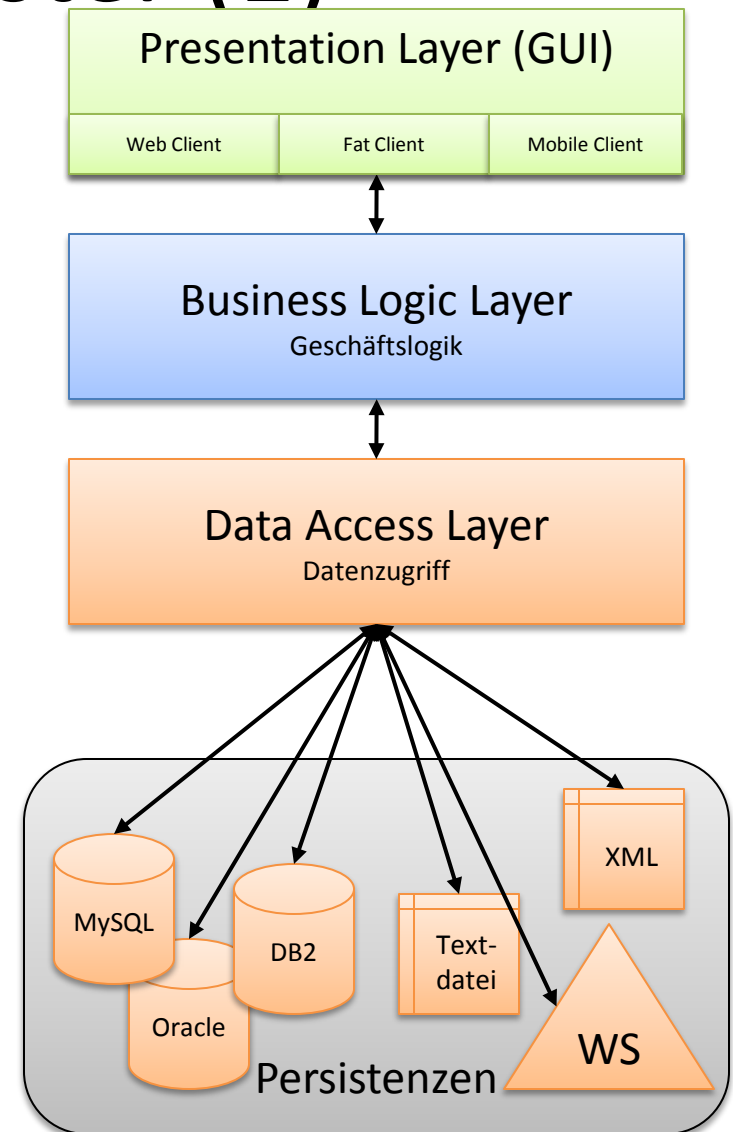
- Try/catch benutzen wir, um Laufzeitfehler – und nicht Kompilierfehler – programmatisch abzufangen.
- Im Try-Block steht der Code, welchen wir „versuchen“ auszuführen.
- Im Catch-Block bestimmen wir die Ausnahme (Exception) auf welche wir reagieren möchten und zusätzlich wie wir reagieren möchten.

```
try{
    //code
} catch(Exception ex){
    //falls ein Fehler auftaucht, führe den Code in diese Rumpf aus.
    Console.Out.WriteLine(ex.ErrorMessage);
}
```

Weitere Beispiele: <http://support.microsoft.com/kb/308345/de>
Visual Studio 2008 c#, Galileo Computing: Seite 448

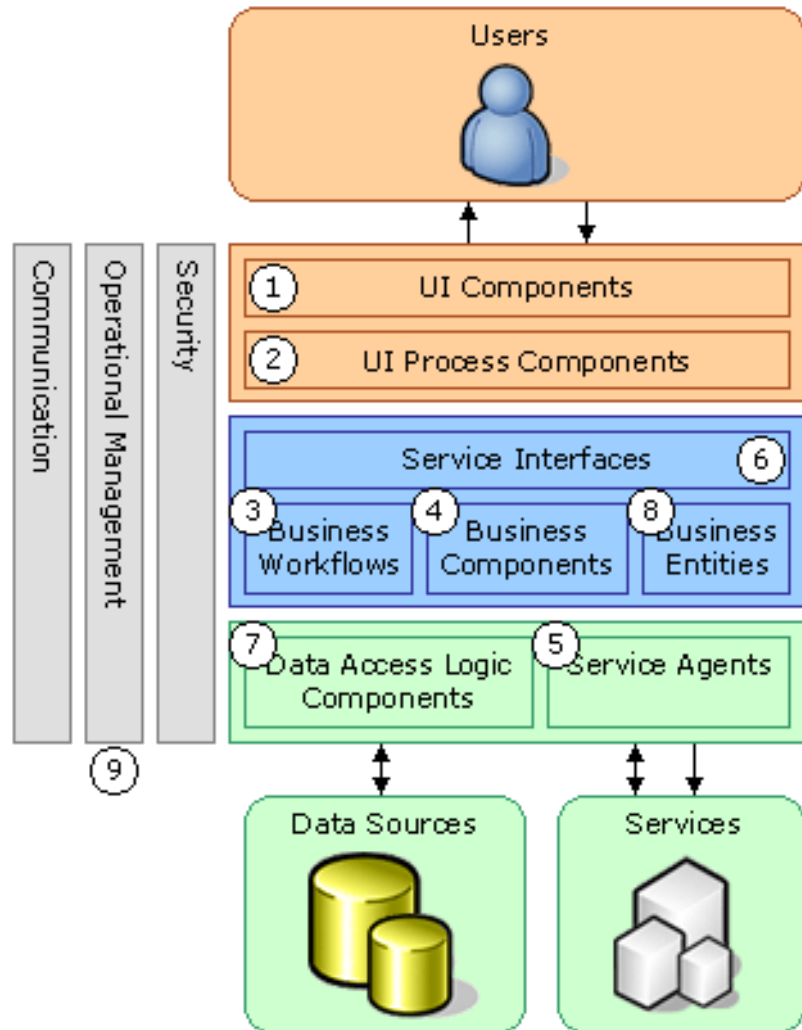
N-Tier Architekturmuster (1)

- Alle umfangreichen Applikationen, die Änderungen unterworfen sind, werden nach einem N-Tier Architekturmuster (AM) entworfen.
- Das AM bietet den Vorteil der Wiederverwendbarkeit z. Bsp. der Geschäftslogikelemente, für andere Schichten.
- Weiter, eine klare Trennung von Verantwortung für den Datenzugriff, die Geschäftslogik und graphischen Benutzeroberflächen.
- Die Trennung schafft zudem Übersicht.
- Die 3 Schichten können in gewissen Szenarien auch zusammengelegt oder in noch mehr Schichten aufgeteilt werden.



N-Tier Architekturmuster (2)

- Der detaillierte Graph der 3-Schichten Architektur kann noch weiter aufgeteilt und mit Sicherheits- und Kommunikationsrichtlinien belegt werden.
- Die Aufteilung bleibt aber gleich.



Schnittstellen / Interfaces für das Contracting

- Als Contracting verstehen wir den implementierten Vertrag zwischen zwei oder mehreren Entwicklern oder Entwicklergruppen.
- In C# wie auch in Java werden Schnittstellen mit dem Schlüsselwort “interface” definiert.
- In Java kann jeweils nur eine Schnittstelle implementiert werden, in c# mehrere.
 - Java Syntax: “implements IMeinInterface”
 - C# Syntax: “: IMeinInterface”
- Weiterführende Literatur:
 - <http://openbook.galileocomputing.de/csharp/kap10.htm#t26>

Dokumentation

- Projektdokumentation beginnen wir mit 3 Slashes (///)
 - Gespeichert im Ordner /Visuals
Studio/Projekte/[Projekt]/bin/debug/[Projekt].xml
 - Kommentieren von Klassen und Methoden
 - Nutzung verschiedener Tags zur strukturierten Beschreibung:
<http://msdn.microsoft.com/en-us/library/5ast78ax%28VS.71%29.aspx>
 - Zweck: Überblick über die Applikation (oder Komponente / Teile der Applikation)
 - Zielpublikum: Architekten und Entwickler
- Lokale Dokumentation beginnen wir mit zwei Slashes (//)
 - Kommentieren innerhalb von Methoden oder Klassenvariablen
 - Zweck: Transparenz schaffen und Erklärung der Details.
 - Zielpublikum: Entwickler

Verteilung - Weitergabeprojekte

- Zweck: Verteilung von Applikationen an verschiedene Benutzer
- Weitergabe Projekte
 - Setup Projekt
 - Web-Setup Projekt
 - Click-One
- Kriterien für die richtige Wahl?
 - Netzverbindung?
 - Mitgelieferte, nicht ins EXE eingeschlossene Dateien?
 - Systemänderungen und Einträge nötig? Registry? Dateitypen-Verbindung? Icons platzieren?
- http://openbook.galileocomputing.de/visual_csharp/visual_csharp_31_001.htm