Daueraufträge:

1. Die C# Schlüsselwörter auswendig lernen  
   <https://docs.microsoft.com/de-de/dotnet/csharp/language-reference/>
   1. Modifizierer
   2. Datentypen
2. Die C# Operatoren auswendig lernen  
   <https://docs.microsoft.com/de-de/dotnet/csharp/language-reference/operators/>
3. [Programmierhandbuch](https://docs.microsoft.com/de-de/dotnet/csharp/programming-guide/) studieren

Meisinger Klemens  
WIFI OÖ GmbH

3883 C# Teil 2 – Komplexe Anwendungen mit Datenbank

Teil von 3881 -> C# Komplettausbildung

* Bisher: 3880 Einführung in die objektorientierte Programmierung (C# und .Net)
* 3882 C# Teil 1 – Komponenten orientierte Anwendungen
* 3884 C# Projekt, 2er Gruppen, und Prüfung

Termine

|  |  |
| --- | --- |
| 1. Di, 12.02.2019 2. Do, [14.02.2019](#Feb14) 3. Di, [26.02.2019](#Feb26) 4. Do, [28.02.2019](#Feb28) 5. Di, [05.03.2019](#Mar05) 6. Do, [07.03.2019](#Mar07) 7. Di, [12.03.2019](#Mar12) 8. Do, [14.03.2019](#Mar14) 9. Di, [19.03.2019](#Mar19) | 1. Do, [21.03.2019](#Mar21) 2. Di, [26.03.2019](#Mar26) 3. Do, [28.03.2019](#Mar28) 4. Di, [02.04.2019](#Apr02) 5. Do, [04.04.2019](#Apr04) 6. Di, [09.04.2019](#Apr09) 7. Do, [11.04.2019](#Apr11) 8. Do, [25.04.2019](#Apr25) 9. Di, [30.04.2019](#Apr30) |

Zeiten:

17:30 – 18:45

19:00 – 20:15

20:30 – 21:45

Teilnehmerinnen und Teilnehmer

BAUER Angelika

~~BITZINGER Sarah~~

DUSANIC Marco

GEBETSROITHER Thomas

GRABNER Dominik

HEIDENBERGER Wolfram Rudolf Franz

KRAGL Markus

LEITNER Niklas Matthias

NIEDERMAYR Wolfgang

PANHOLZER Andreas

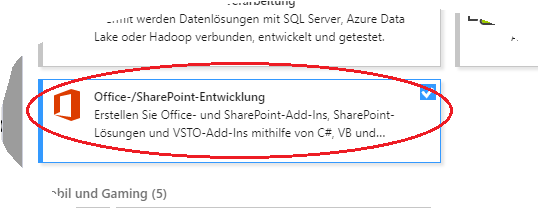
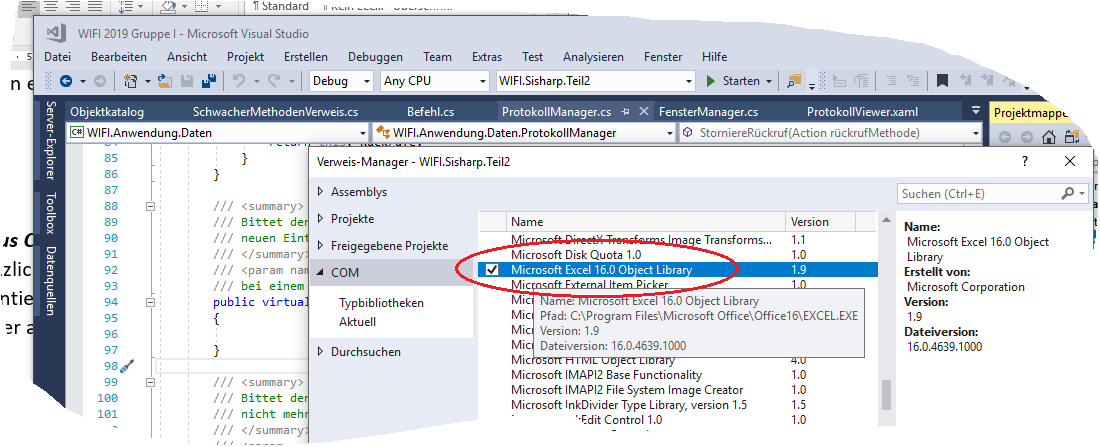
PIRKLBAUER Hermann

Frage:

* ***Wofür steht CLR?***
  + Für Common Language Runtime -> .Net Framework
* ***Wie unterscheidet WPF in Xaml die Steuerelemente?***
  + Überhaupt nicht
  + In Xaml müssen Objekte …  
      
    … nicht benannt werden
  + Sie werden über die Datenbindung gesteuert
  + In Xaml müssen Objekte nur dann benannt werden,  
    wenn sie in einer Datenbindung oder im Code behind benutzt werden.

SCHLAGER Jürgen

SCHWARZ Dietmar

* ***Wie kann ich aus C# Microsoft Office Excel steuern?***
  + Grundsätzlich können mit C# ein COM-AddIn für Excel implementiert werden.
  + Im Installer aber   
    
  + Excel „direkt fernsteuern“
  + Verweis auf  
    

STIEGELBAUER Selina

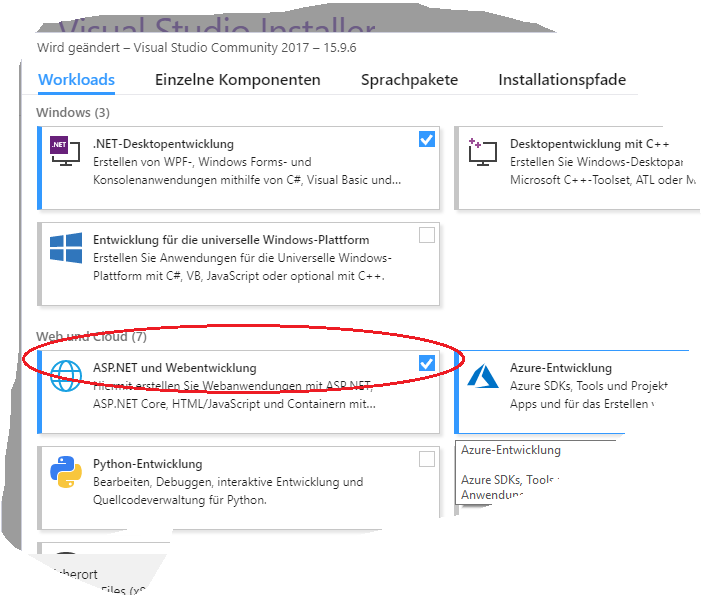
Frage:

* ***Bei einer mehrsprachigen Anwendungen werden in der System.Windows.MessageBox die Schaltflächen immer in der Sprache des Betriebssystem angezeigt. Was tun?***
  + Hier muss ein eigenes Formular für die MessageBox mit übersetzten Schaltflächen erstellt werden

SPECKHARD Christoph

ZENKOVIC Damir

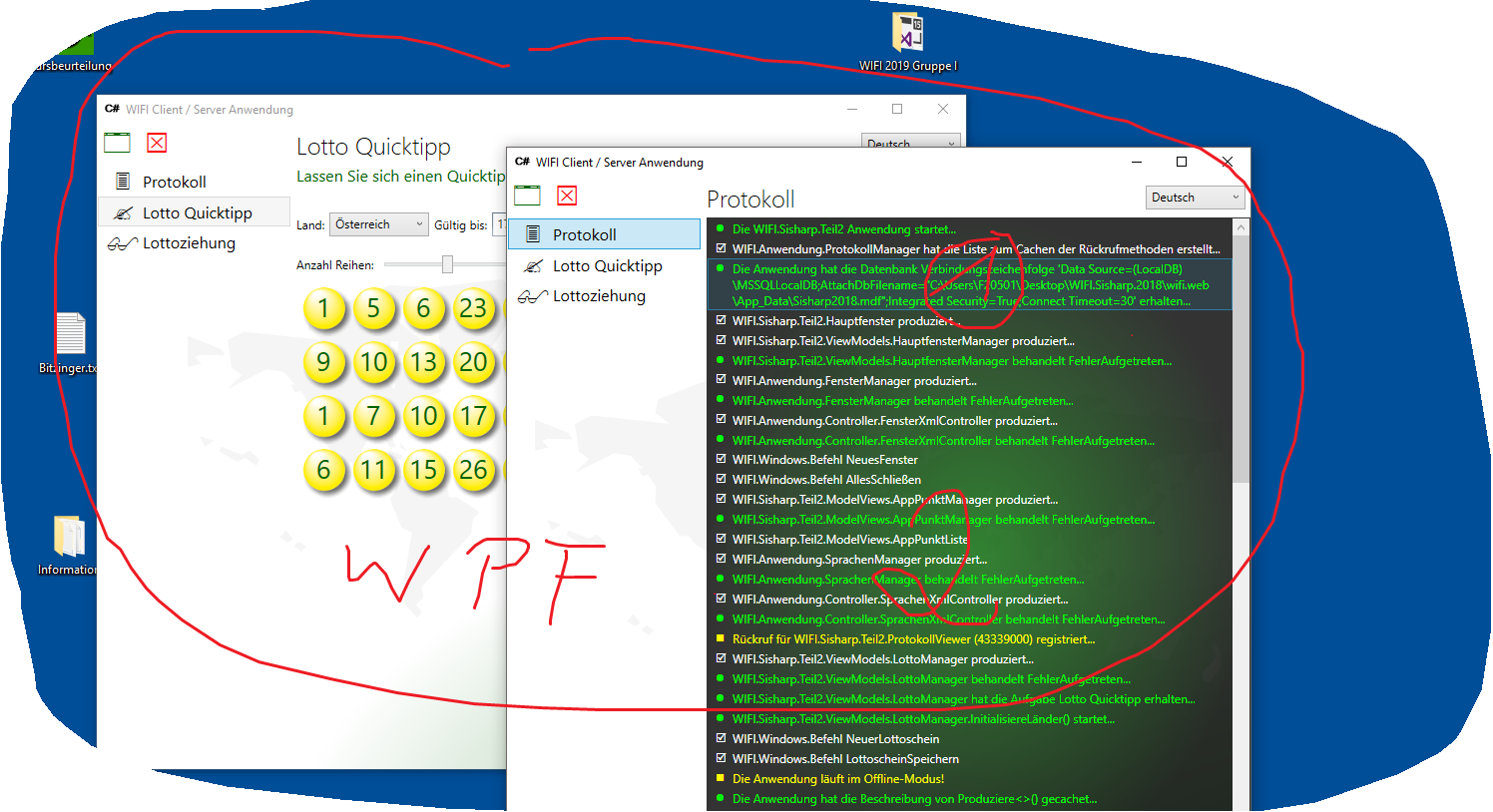
Zu den Inhaltsschwerpunkten

1. Firmenframework um die  
   Möglichkeiten, eine Datenbank zu  
   benutzen, erweitern
2. Damit man die Kontrolle behält,  
   eine Möglichkeit zur Protokollierung schaffen
3. Datenbank
4. Weil immer öfter notwendig, den  
   Zugriff auf die Datenbank hinter  
   einem SOAP Webdienst verstecken  
   (dazu:   
     
   muss die ASP.Net und Webentwicklung im Installer aktiviert sein)  
     
   Eine echte Client / Server Anwendung
5. Das Entwurfsmuster MVVM einhalten  
   (Model – View – ViewModel,   
   eine Erweiterung vom MVC (Model-View-Controller))
6. Die Benutzeroberfläche muss auf  
   allen neuen Geräten „scharf“ sein,   
   deshalb die neue Oberflächen-Beschreibung   
   WPF (Windows Presentation Foundation)

Im Lernen und Teil 1 Projekt wurde miteinander analysiert. Das müssen Sie im Projekt Training in 2er Gruppen wieder machen.

Hier wurde vom Trainer analysiert:

Ergebnis der Oberfläche:



(notwendige) Komponenten

|  |  |
| --- | --- |
| Assembly | Beschreibung |
| WIFI.Anwendung | Firmenframework für Anwendungen ohne Datenbank Zugriff |
| WIFI.Anwendung.Daten | Erweiterung von WIFI.Anwendung um die Möglichkeiten, Datenbanken anzusprechen |
| WIFI.Sisharp.Teil2 | Die Oberfläche der Teil 2 Anwendung. Eine WPF Anwendung.  Diese DARF **WIFI.Anwendung.Teil2.Daten** nicht verweisen. Teil2.Daten enthält „fiktiven geheimen“ Code. Standard Betriebsmodus muss „online“ sein. Administratoren sollen aber auch in den „offline“ Modus wechseln können und die Teil2.Daten Assembly dynamisch nachladen können. |
| WIFI.Sisharp.Teil2.Daten | In dieser Assembly befinden sich die Klassen, die die Demodatenbank des Trainings verwalten.  (Lottozahlen verwalten) |
| wifi.web | Ein Webdienst, der die Schnitte zu WIFI.Sisharp.Teil2.Daten bereitstellt. Dieser ist auf einem Internet Information Server (IIS) gehostet und nutzt die Teil2.Daten Assembly. |

Ein Problem in der .Net Entwicklung:

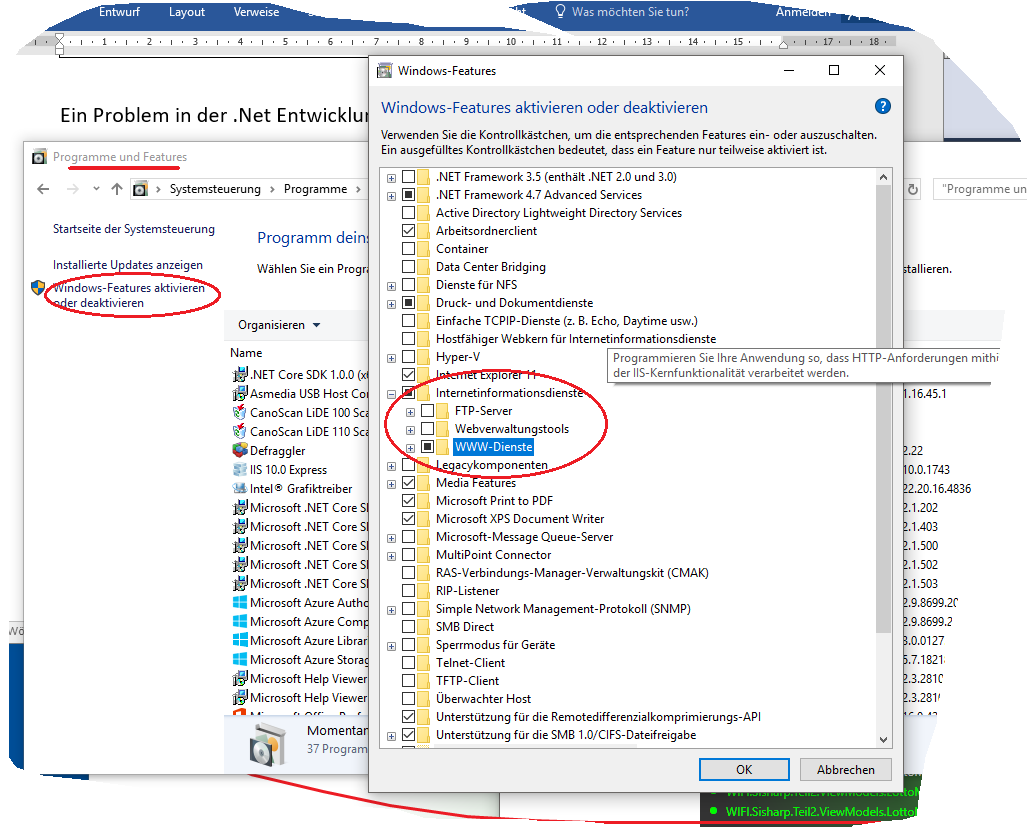
* Heute wird beim „Compilieren“ kein Maschinencode mehr produziert
* Heute wird MIL (Microsoft Intermediate Language, Assembler-Code) produziert und JIT („Just-In-Time“) compiliert
* Alle können hergehen und eine Assembly „decompilieren“ und den Assembler Code lesen
* Wenn jemand auf keinen Fall ihre Assembly in die Finger bekommen soll, dann darf diese nicht am Benutzercomputer installiert sein

Problem im Training:

* Webdienst hosten

Visual Studio hat einen IIS Express integriert, d.h. die Anwendung läuft, wenn Visual Studio läuft.

Windows Professional hat außerdem einen IIS integriert.



Falls jetzt jemand zu Hause aber nur Windows Home hat, gibt’s keinen IIS.

Deshalb:

* WIFI.Sisharp.Teil1 muss WIFI.Sisharp.Teil2.Daten auch ohne Verweis dynamisch nutzen können

Hinweis:

<https://docs.microsoft.com/en-us/visualstudio/ide/dotfuscator/?view=vs-2017>

* Hier ist eine Möglichkeit beschrieben, in eine Assembly so Unordnung reinzubringen, dass diese beim Decompilieren keinen Sinn ergibt, aber funktioniert.

Und wo sind jetzt diese „Decompilierungswerkzeuge“?

* Werden von .Net mitgliedert

<https://docs.microsoft.com/de-de/dotnet/framework/tools/>

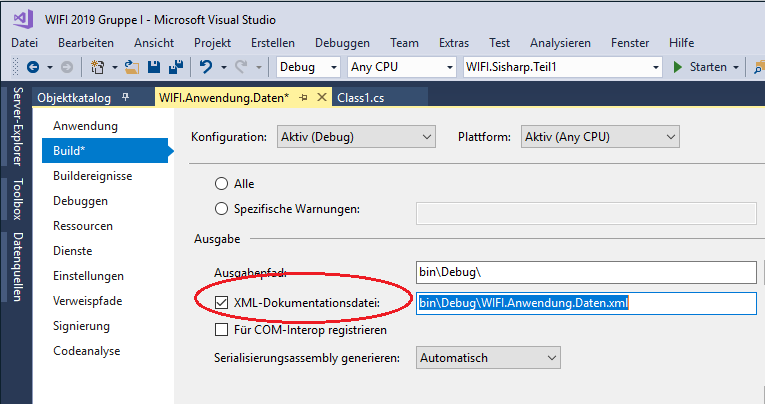
Zur ersten Assembly:

* WIFI.Anwendung.Daten

Warum wird nicht WIFI.Anwendung, das Firmenframework erweitert?

* Weil, wenn für mehrere Kunden Anpassen notwendig sind,  
  kann nicht ständig das Kern-Framework geändert werden
* Irgendwann ist das so ein Saustall, dass sich niemand mehr auskennt

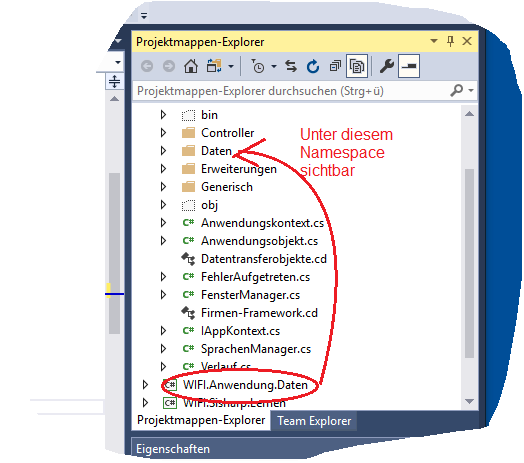
Nach dem Anlegen des Projekts …



… in den Eigenschaften „XML-Dokumentationsdatei“ aktivieren

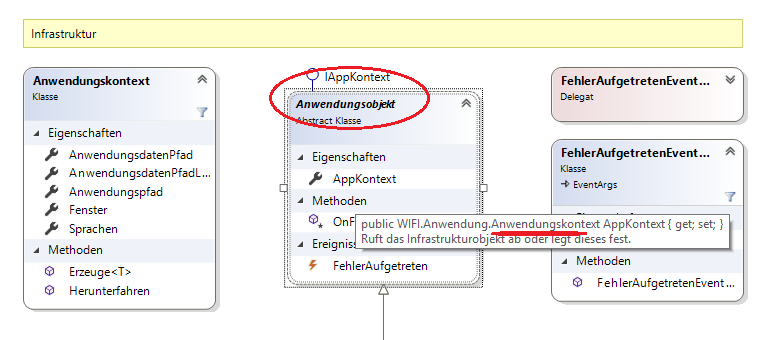
Außerdem die Assembly-Information sauber einstellen.

Hinweis:

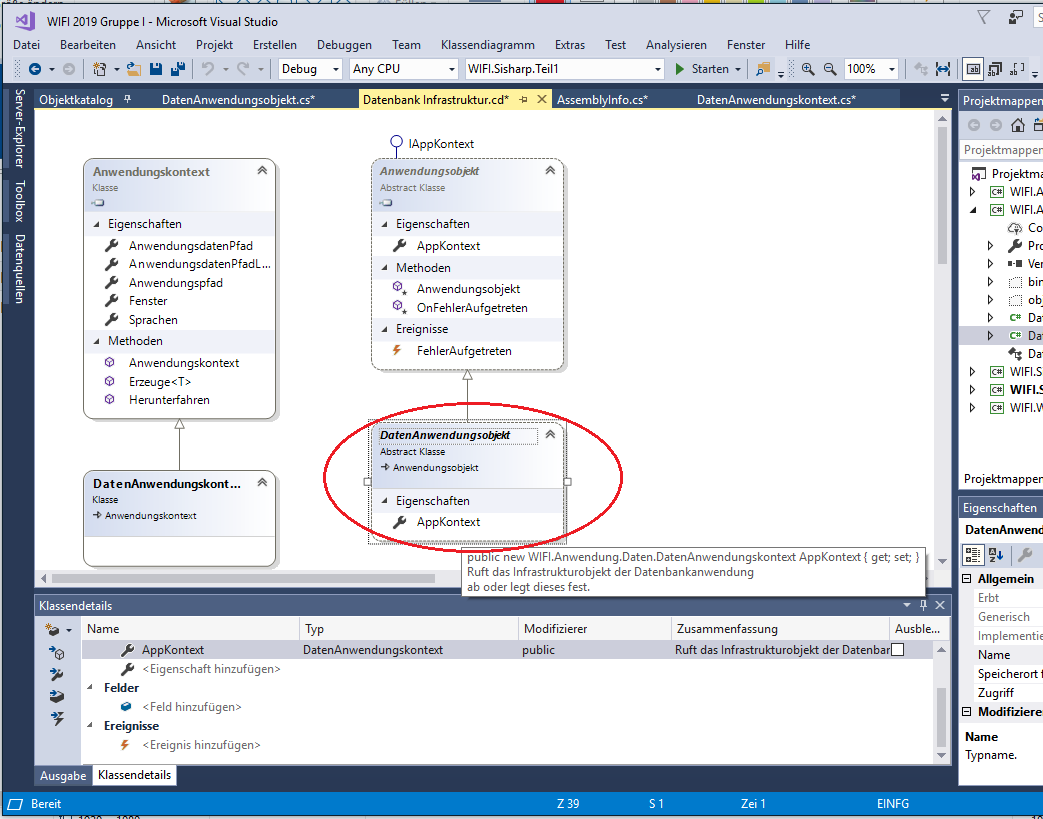


* Die Klassen der neuen Framework-Erweiterung „WIFI.Anwendung.Daten“ sind im „normalen“ herkömmlichen Daten-Namespace des Frameworks

Problem:



* Das Framework Anwendungsobjekt stellt AppKontext bereit
* Aber mit dem „falschen“ Datentyp
* Wir benötigen in Zukunft DatenAnwendungskontext
* Damit wir nicht ständig casten müssen,  
  verbergen wir das Casten in einer eigenen, neuen Basisklasse



Wir wissen, bei jedem Projekt, ist das erste Objekt, die Infrastruktur.

Wo beginnt eine Anwendung zu laufen?

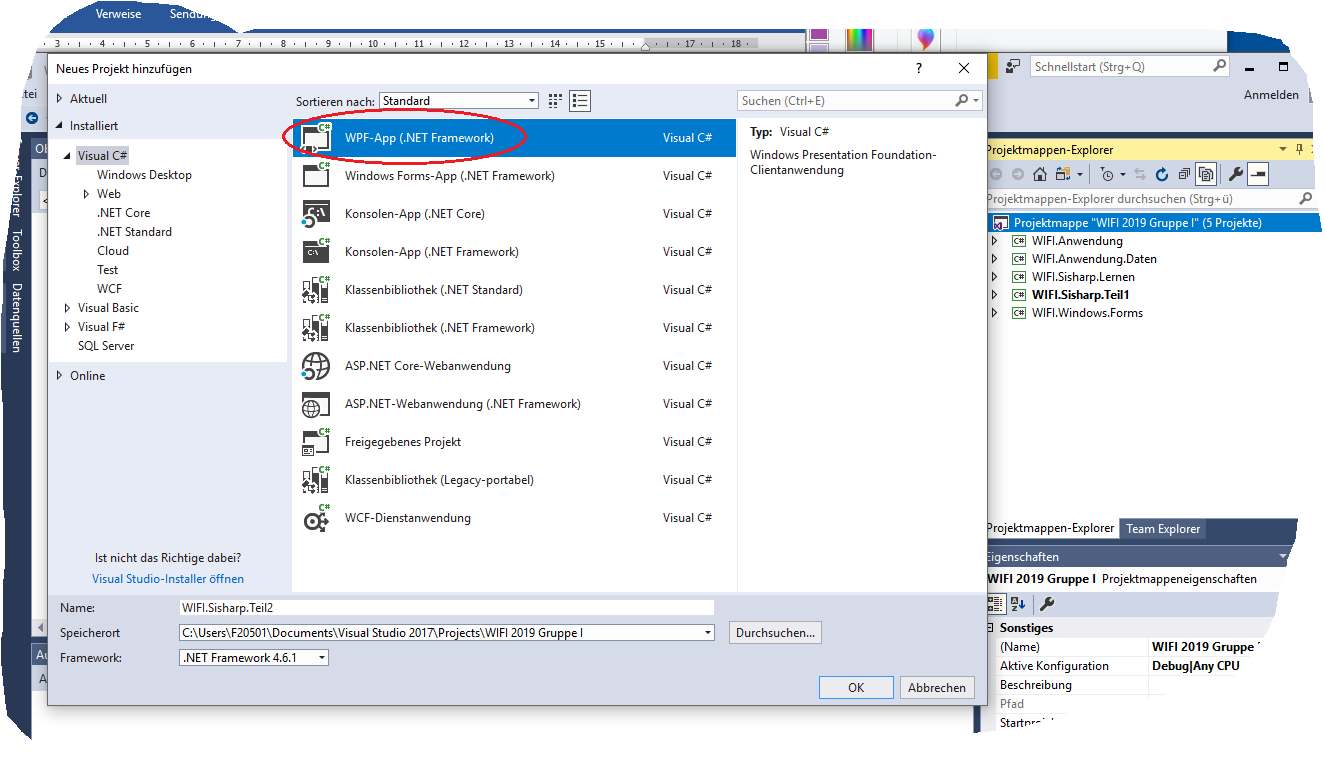
* Im Main()

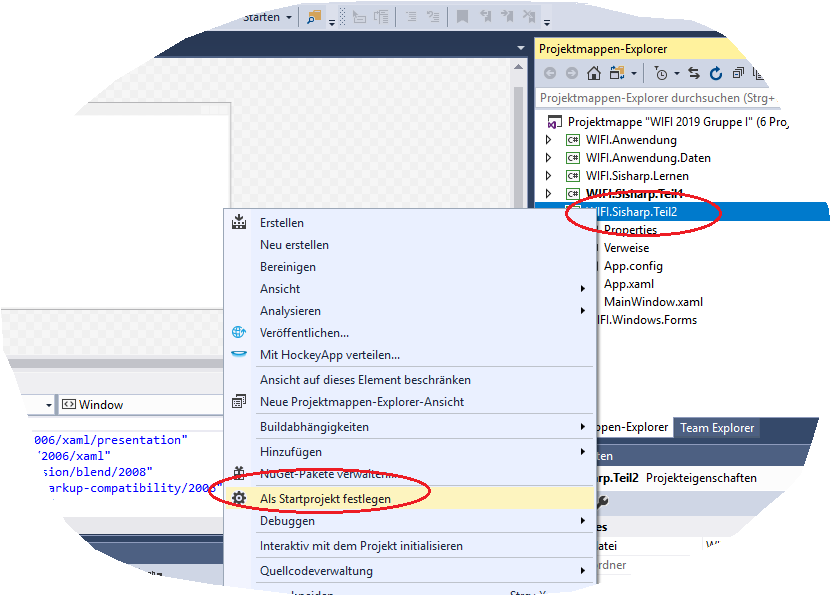
Jetzt benötigen wir die nächste Assembly, die WIFI.Sisharp.Teil2

Im 1. Teil: System.Windows.Forms

Dieses Mal:

* Windows Presentation Foundation,  
  eine WPF Anwendung



* Und zum Startprojekt…  
  

… machen.

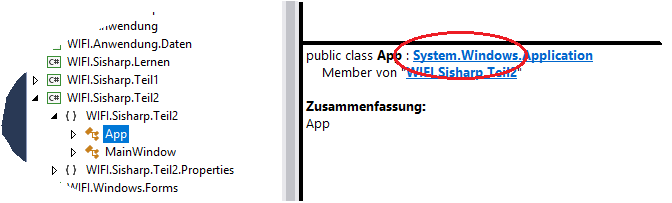
Hauptunterschied zwischen

* Forms  
  und
* WPF

Windows Forms nutzt die grafische Darstellung   
vom alten GDI (Graphical Device Interface)

WPF benutzt die Grafikhardware (DirectX) direkt.

Änderungen im Namespace:



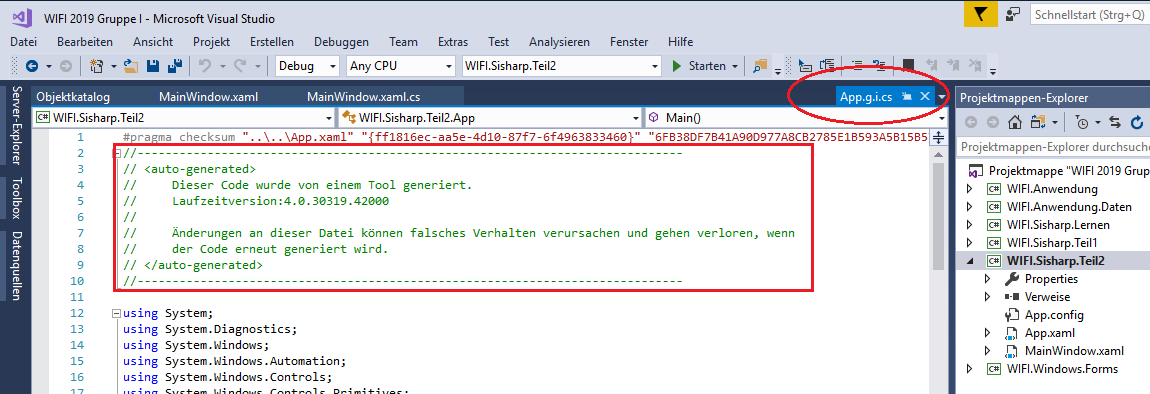
* WPF hat nur mehr „System.Windows“
* Forms hat ja „System.Windows.Forms“ benutzt

Unsere erste WPF Anwendung läuft.

Wir benötigen das Main…

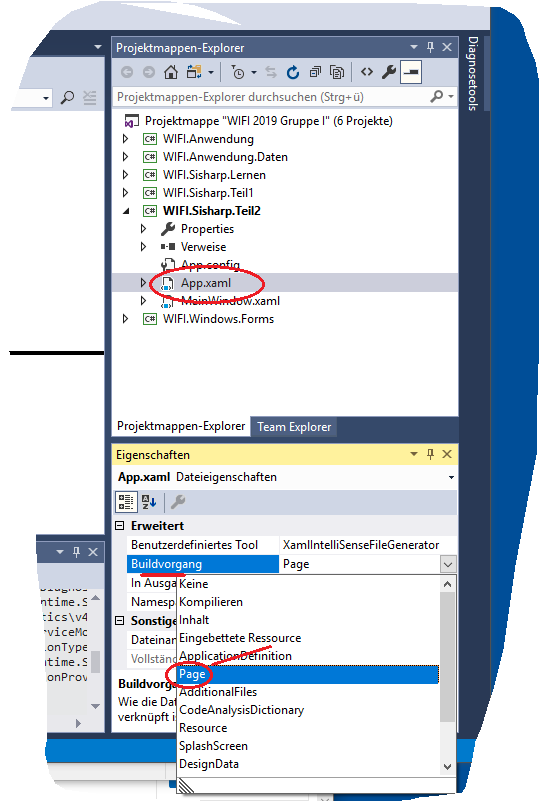
* Wo ist’s?

In einer „komischen Datei“ …



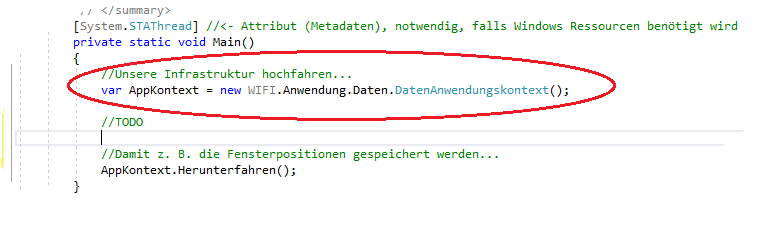
… vom Assistenten generiert, wo im Kommentar steht, dass man das nicht ändern darf.

Hilfe, wir wollen unser Main() zurück…

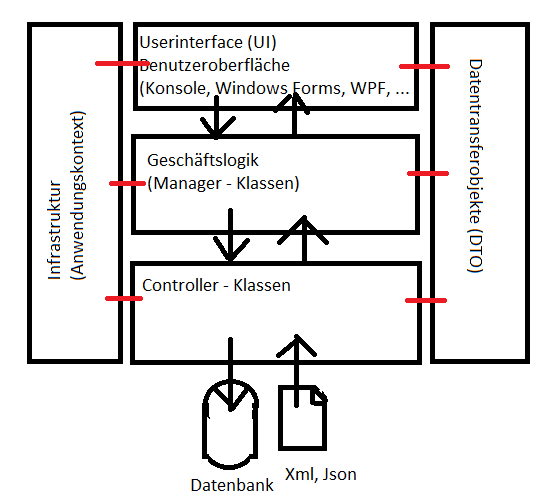


* Dazu bei „App.xaml“ Datei in den Eigenschaften den Buildvorgang von „ApplicationDefinition“ auf „Page“ umstellen

Für das Main() den Code Behind der App.xaml Datei öffnen…



Erinnerung an das Anwendungsdesign:



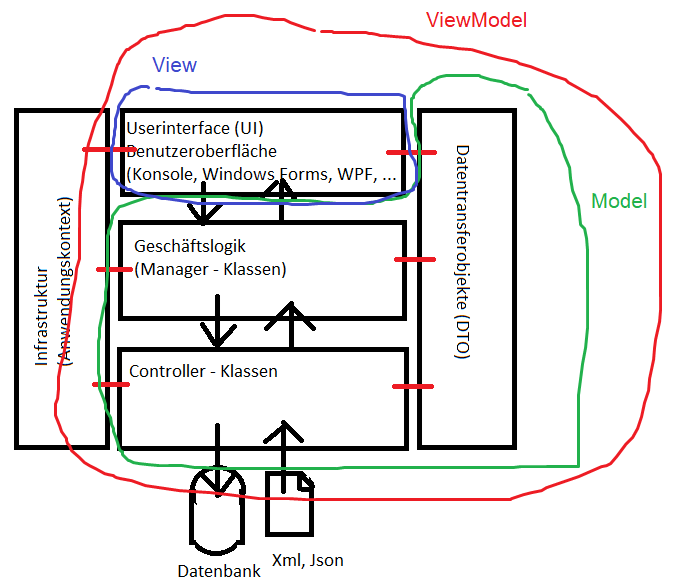
* Die Infrastruktur, der Anwendungskontext steht

Jetzt kommt aber eine gravierende Änderung zur Windows Forms:

* In der Windows Form wurde bei einem Formular im „Code-Behind“ programmiert.
* Problem:  
  Der „Code-Behind“ kann aber z. B. nicht automatisiert getestet werden
* Wunsch:  
    
  Die Beschreibung der Oberfläche und der dazu notwendige Code für die Steuerung MUSS getrennt werden.
* Dazu:  
    
  Ein Programmiermuster (Pattern)  
    
  Model – View – ViewModel (MVVM)

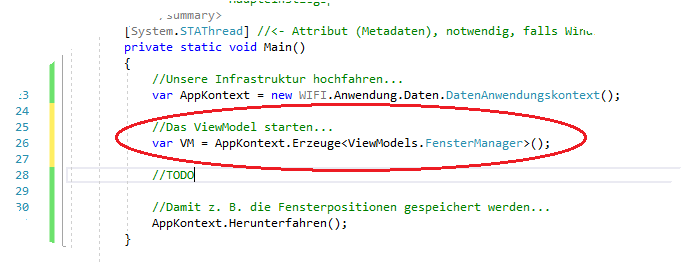
Aufteilung:

* Model  
  Hier sind alle Klassen, die die Daten und deren Verwaltung beschreiben
* View  
  Das sind alle Oberflächen OHNE Code
* ViewModel  
  Hier sind die Klassen, die die Oberfläche (die Views) steuern



Die Abbildung zeigt, alles was mit den Daten zu tun hat, wird als „Model“ bezeichnet. Alles was die Benutzer sehen, als „View“. Das Model und die Views werden von „ViewModel“ Klassen (Objekten) kontrolliert und gesteuert.

Deshalb …



… starten wir nach dem Hochfahren der Infrastruktur unser ViewModel

14.2.2019

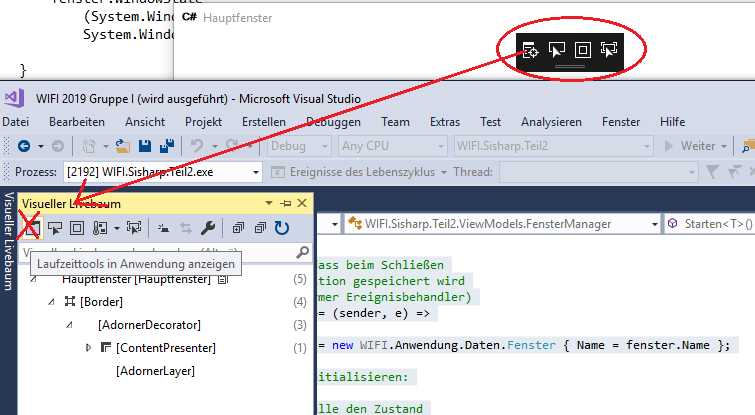
* Erster Teil:  
    
  Das Fenster soll sich die Position merken und   
  diese wiederherstellen

Bei der Windows Forms haben wir ein Basisfenster implementiert.

Wegen des MVVM Konzepts muss diesmal die Logik in   
unser ViewModel FensterManager

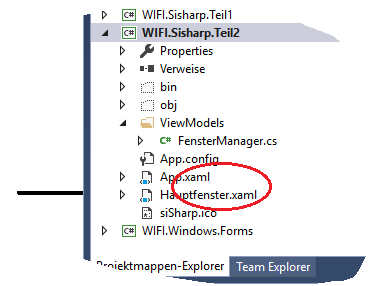
* InitialisiereFenster(…) Methode

Lästig:



* Das WPF Tool Fenster
* Kann über das „Livebaum“-Panel deaktiviert werden

Wofür stehen diese …

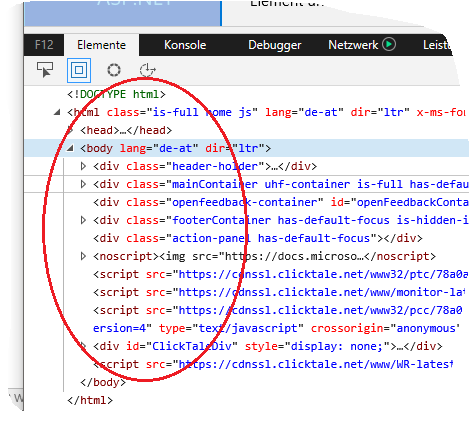


… neuen „xaml“ Erweiterungen?

Es wurde lange philosophiert, wie eine Oberfläche beschrieben werden kann. Man war sich nur einig, dass die Windows Forms obsolet ist.

Was tun?

* An das Internet erinnern



* Internetseiten sind mit HTML beschrieben

Wäre doch lustig, wenn „normale“ Anwendung auch damit beschrieben werden können.

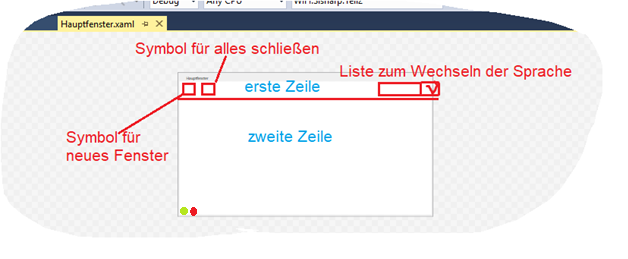
* Dabei aber ein neues Problem  
    
  HTML ist ziemlich eingeschränkt
* Die normalen Desktopanwendungen können mehr
* Irgendwie müssen wir „Klassen“ starten können
* Wir benötigen eine „freiere“ Beschreibung  
    
  Idee: XML
* Das muss jetzt „aufgeblasen“ werden, damit   
  Klassen gestartet werden:

Extensible Application Markup Language (XAML)

Verzeichnis der möglichen Klassen, die in XAML serienmäßig gestartet werden können:

<https://docs.microsoft.com/de-de/dotnet/framework/wpf/controls/index>

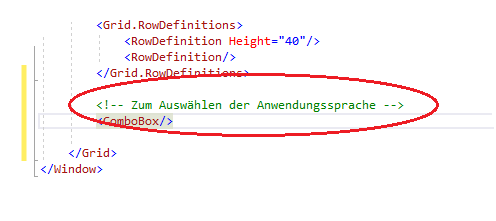
Erste Gehversuche in Xaml



Im Hauptfenster zwei Zeilen

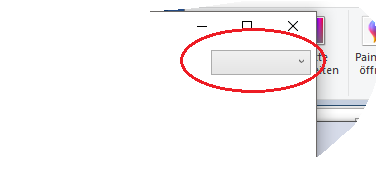
In den oberen Zeile links 2 Symbole und rechts eine Liste   
zum Auswählen der Sprache

Wichtig:



* Im XAML Kommentare einfügen!

Und wie geht’s weiter?



* Jetzt haben wir eine Liste. Sieht fast nach Teil 1
* Im Teil 1 wurde jetzt aber im Code behind gearbeitet
* Das ist pfui

Wir erstellen eine MVVM Anwendung. Hier soll auf den Code behind verzichtet werden.

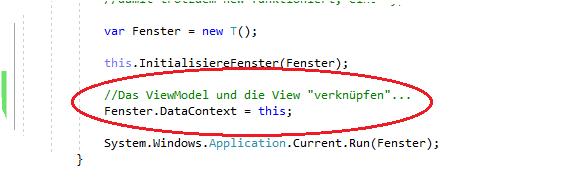
Wo ist die Liste der Sprachen aktuell?

* In unserem ViewModel FensterManager in der Infrastruktureigenschaft „AppKontext“

Wie kann das Fenster („Die View“) und das ViewModel verknüpft werden?

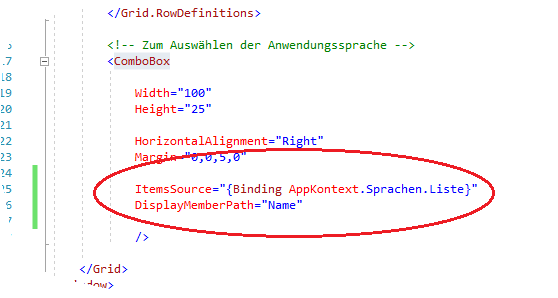
Dazu kennt WPF eine mächtige Datenbindung. Die Eigenschaft, wo diese Datenbindung das ViewModel sucht, heißt…

* DataContext



Die Abbildung zeigt, wie DataContext auf die aktuelle Instanz vom FensterManager eingestellt wird.

Zum Binden an das Objekt im DataContext …



… geschweifte Klammer und „Binding“ verwenden. Anschließend die zu benutzende Eigenschaft.

Hinweis:

* Es können NUR ÖFFENTLICHE EIGENSCHAFTEN gebunden werden!

Dokumentation:

<https://docs.microsoft.com/de-de/dotnet/framework/wpf/data/data-binding-overview>

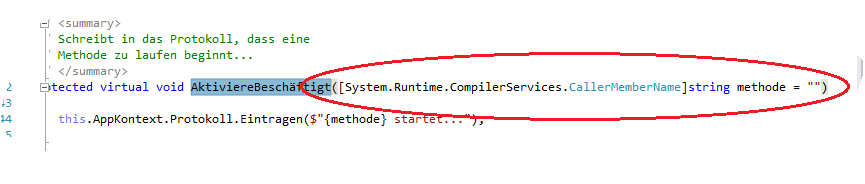
Damit wir die Kontrolle über unsere Anwendung nicht verlieren:

* Dem Anwendungskontext ein Protokoll beibringen

Nachdem die ersten Züge vom Protokoll arbeiten…

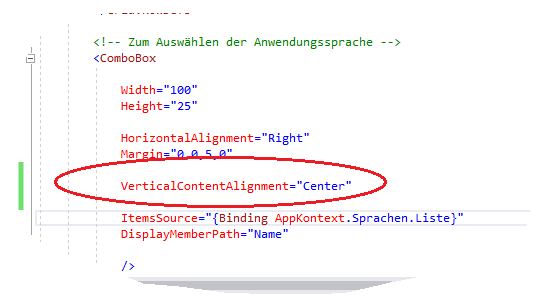
* Wie beim Lernen soll jede Methode  
  im Protokoll den Hinweis „xyz startet…“  
  und „xyz beendet.“ Hinterlassen

Lösung:



* Optionaler Parameter und ein Attribut aus den CompilerServices

26.2.2019

* Damit in der Liste der Sprachen…  
  

… der Inhalt vertikal zentriert ist.

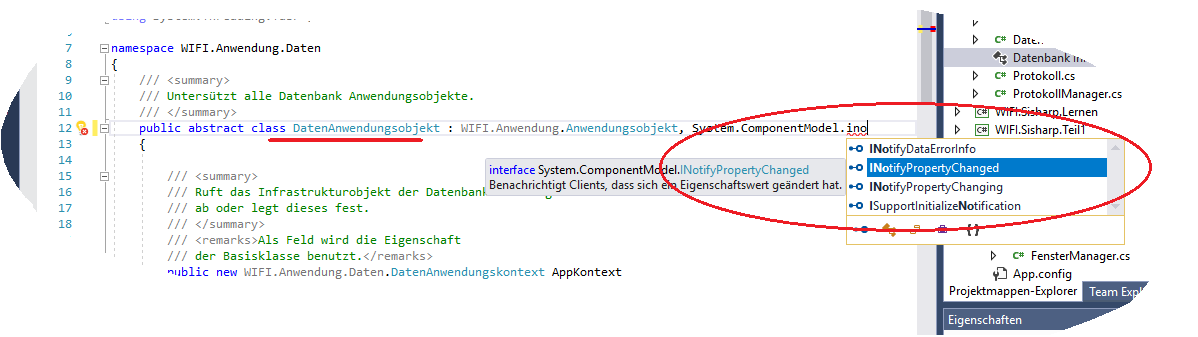
* Außerdem soll die aktuelle Sprache ausgewählt sein
  + Dazu muss das ViewModel „FensterManager“ eine  
    Eigenschaft bekommen

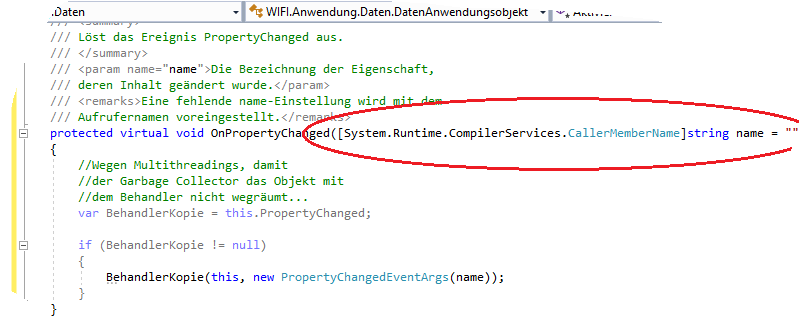
Hinweis:

* In WPF können **nur öffentliche Eigenschaften** gebunden werden,  
  keine Methoden



* Damit Änderungen der Datenbindung mitgeteilt werden,  
  das Ereignis PropertyChanged auslösen
* Damit das Ereignis nicht ständig implementiert werden muss,  
  ab damit in die Basisklasse



* Beim Ereignisauslöser …  
    
  … wird wieder das neue Attribut seit .Net 4.5 zur Voreinstellung benutzt.
* Anschließend kann …  
    
  … die neue Eigenschaft an die Liste gebunden werden.

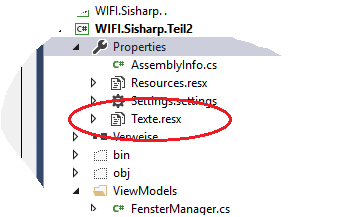
Wichtig:

{Binding} sucht die Eigenschaft im aktuellen DataContext – Objekt.

Bei uns eingestellt vom ViewModels.FensterManager

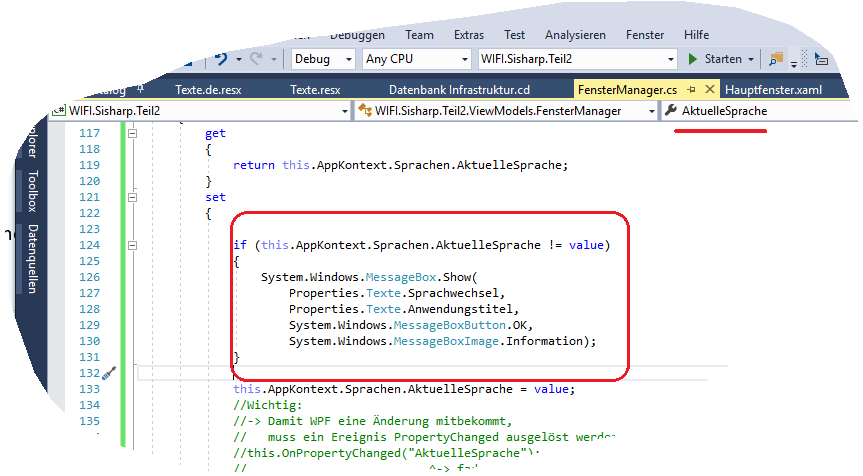
Weiter geht’s mit den mehrsprachigen Texten für die Anwendung…

Dazu …



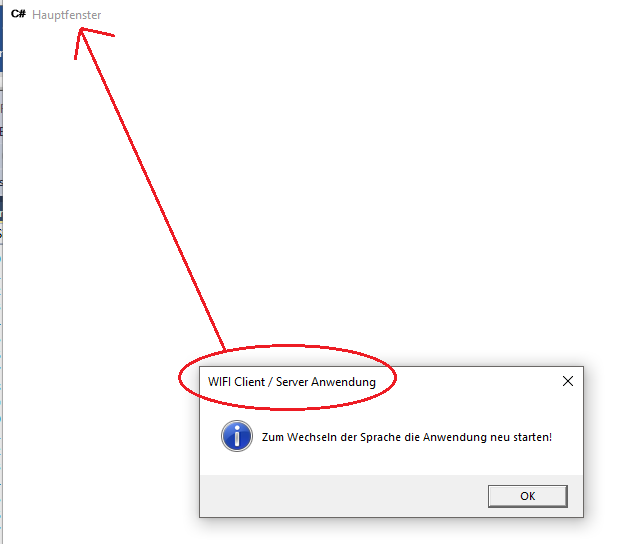
… muss nicht unbedingt „Resources.resx“ benutzt werden. Der besseren Strukturierung wegen können beliebige Ressource Dateien hinzugefügt werden.

Auf diese Ressourcen …



… kann wie gewohnt über den Properties-Namespace zugegriffen werden.

Nächster Wunsch:



* Wir hätten gerne die neue Ressource Anwendungstitel an den Fenstertitel gebunden

VORSICHT FALLE!!!!

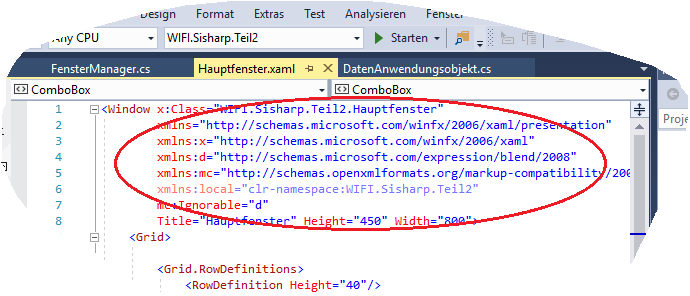
* Gebunden werden können nur öffentliche Eigenschaften



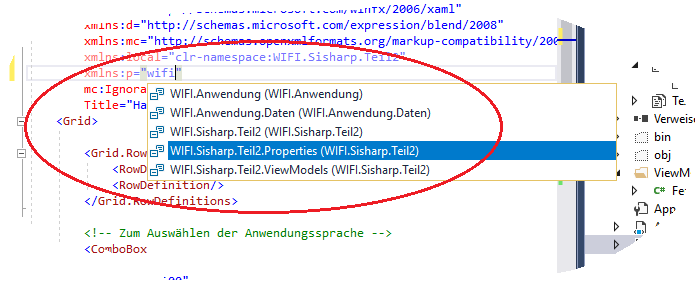
* Der Zugriffsmodifizierer der Ressourcedateien ist aber „Internal“
* Auf „Public“ ändern

Neues Problem:

* Wie findet Xaml die möglichen Klassen?

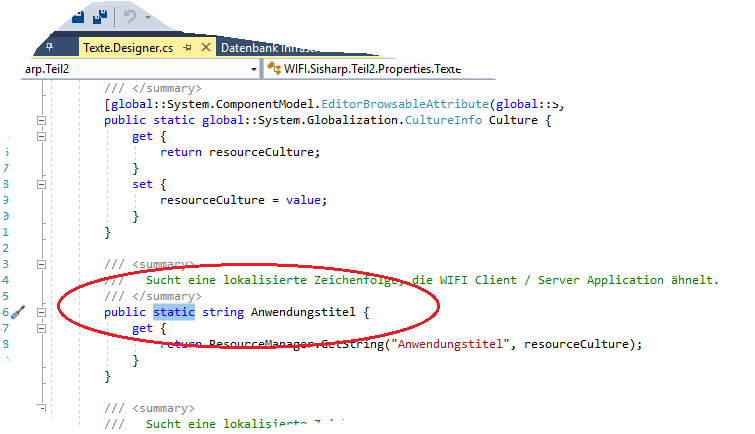


* Über Xml-Namespaces Attribute
* Hier ist aber noch kein Xml-Namespace für unsere Properties, wo die Texte-Ressourcen liegen

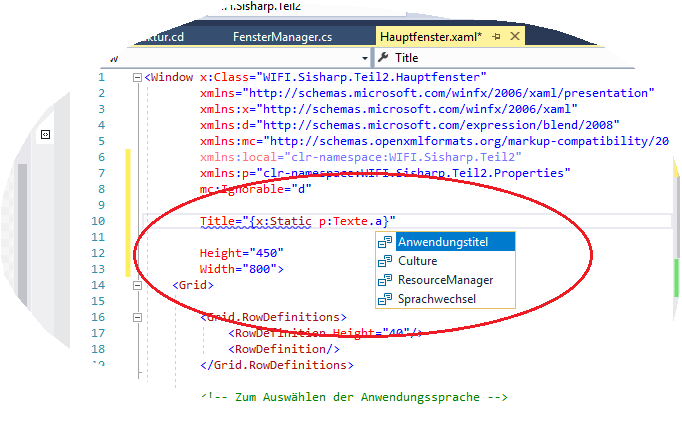


* Deshalb einen neuen Namespace mit Alias „p“ wie Properties auf WIFI.Sisharp.Teil2.Properties zeigen lassen

Zum Binden an Texte.Anwendungstitel …



… muss der Datenbindung mitgeteilt werden, dass es sich um eine statische Bindung handelt.

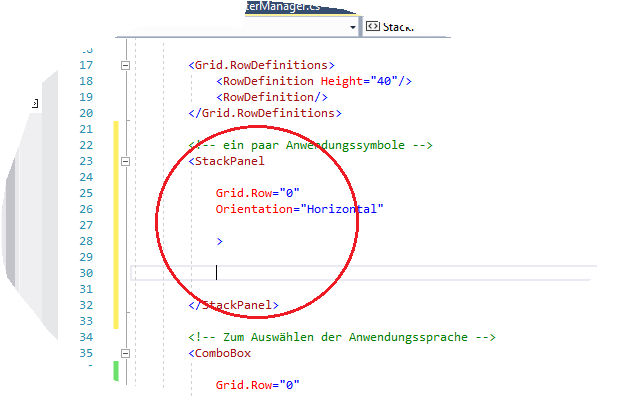


* Dazu aus dem x – Namespace „Static“ und aus dem p – Namespace Texte.Anwendungstitel benutzen

Wichtig:

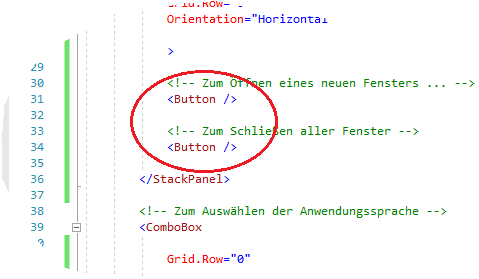
* Wird die statische Datenbindung innerhalb derselben Assembly benutzt, die Assembly vorher neu kompilieren.

Zurück zum Oberflächen Design:



* Das mächtigste Werkzeug zum Positionieren: Das Grid
* Wenn’s schnell gehen soll:  
  Ein StackPanel
* Die Standardausrichtung des Inhalts ist „untereinander“  
  Wir wollen „nebeneinander“ (Orientation = "Horizontal")

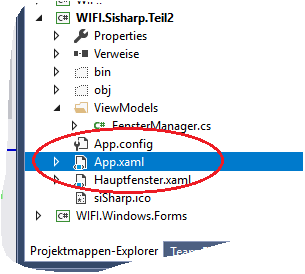
Keine Lust, …



… bei mehreren Objekten dieselben Attribute einzustellen.

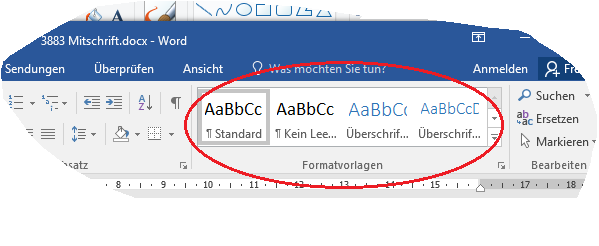
* Was tun?

Uns fragen, was Microsoft mit „App.xaml“ vorhat



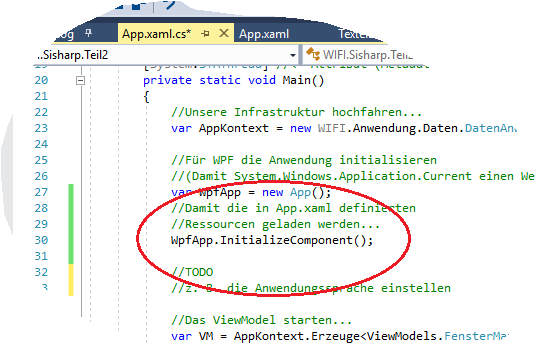
* Die App.xaml ist dazu gedacht,  
  Einstellungen, Ressourcen, zu hinterlegen,  
  die man überall in der Anwendung braucht

Sowie Word, …



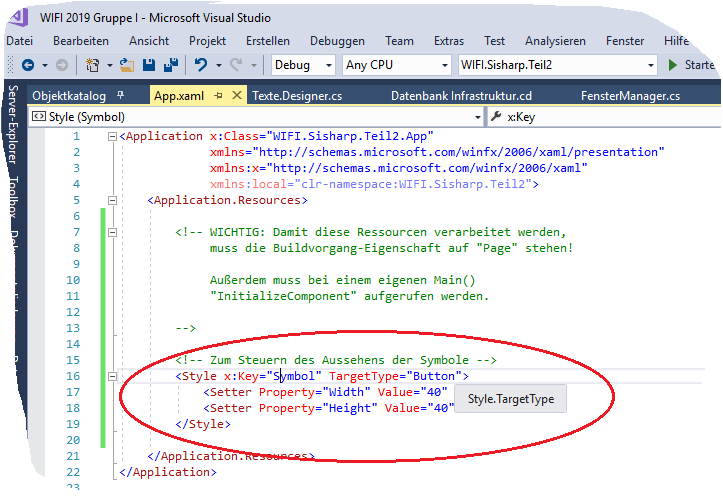
… mit den Formatvorlagen („Stile“) können WPF Vorlagen definiert werden,  
die überall benutzt werden können, wenn sie in App.xaml definiert sind

Damit diese Ressourcen geladen werden, …



… bei einem eigenen Main „InitializeComponent“ aufrufen!

In einem Stil …



… kann mit dem Setter eine Eigenschaft eingestellt werden.

Die Frage, …

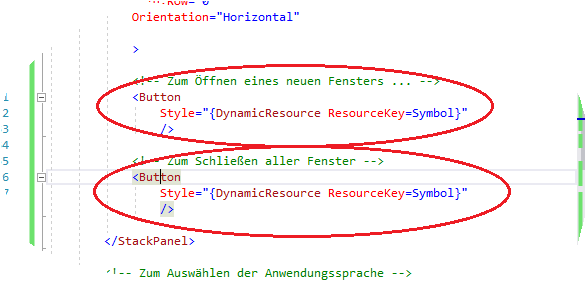
* Wie wird an so einen Stil gebunden?

Bisher:

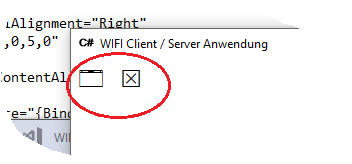
* {x:Static …} an eine „normale“ Ressource
* {Binding …} an den DataContext

Jetzt das Dritte:

* An die Ressourcen in App.xaml wird „dynamisch“ gebunden



Neues Problem:

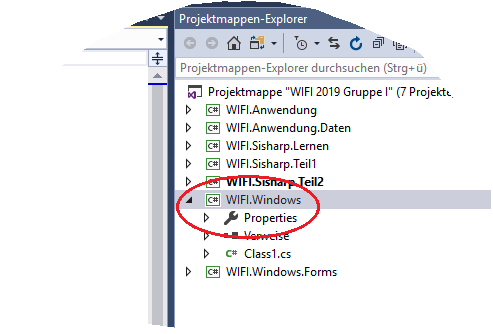


* Unsere Symbole müssen „arbeiten“, d.h. wir benötigen Methoden
* In Windows Forms hätten wir den Code behind genommen
* Hier aber MVVM Entwurfsmuster, d.h.  
  die Methoden gehören in ein ViewModel, in unserem Fall den FensterManager und müssen gebunden wurden
* Gebunden können aber nur öffentliche Eigenschaften   
  werden, keine Methoden

Lösung:

* Wir benötigen ein Objekt, das wir als Eigenschaft präsentieren  
  und in dem eine Methode versteckt ist
* Dazu kennt WPF ein Interface ICommand

Damit das in Zukunft nicht wieder erstellt werden muss, …



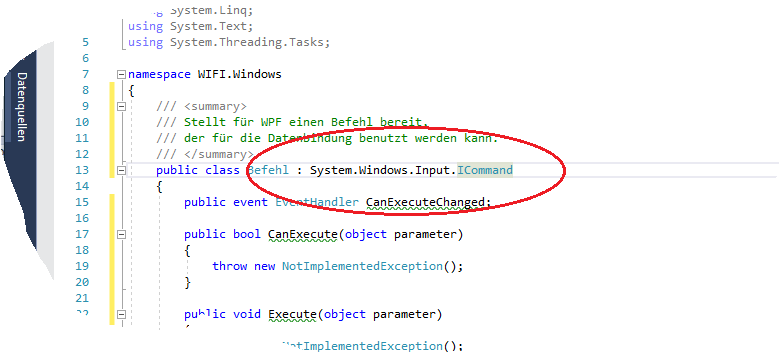
… eine neue Assembly.

Natürlich …



… wieder sofort die AssemblyInfo ausfüllen und in den Eigenschaften unter „Build“ die Xml-Dokumentationsdatei aktivieren.

Für die WPF Datenbindung …



… ist ICommand zu implementieren.

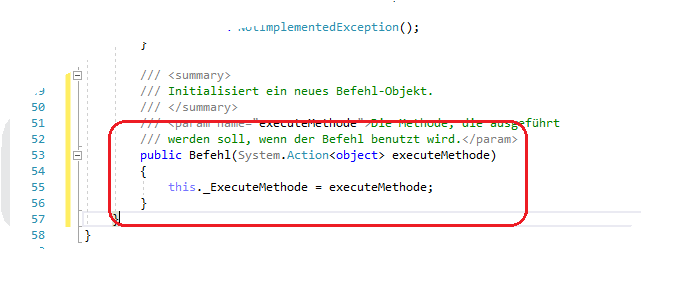
Die Methode Execute des Interfaces muss unsere Methode kapseln.

* Zum Merken einer Speicheradresse einer Methode ist  
  ein Delegate notwendig.
* Wir benötigen einen Delegate, der etwas  
  ohne Rückgabe arbeitet und einen Parameter vom Typ object hat
* Im .Net, in der Kernbibliothek bei System.Action<T>()

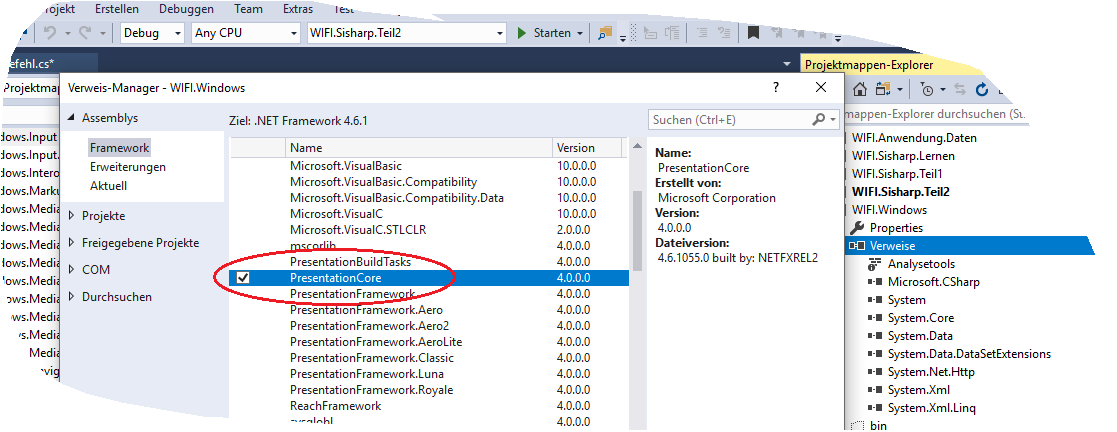


* Wie bekommen wir in den internen privaten Delegate  
  einen Wert?

Das kann nur über den Konstruktor laufen…



Zum Verknüpfen des Ereignisses vom Befehl …

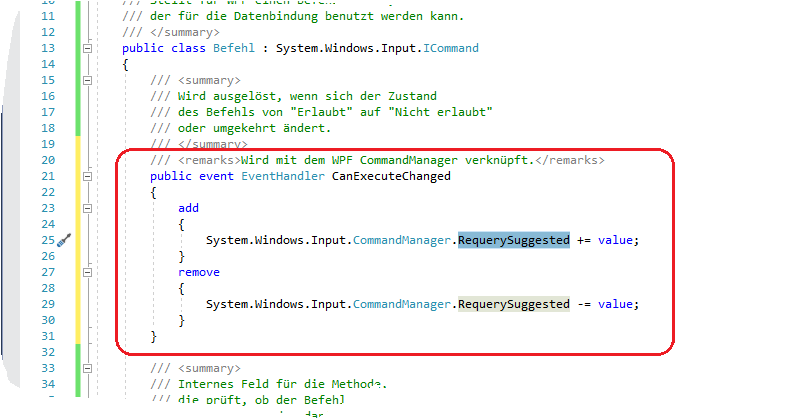


… mit dem WPF CommandManager einen Verweis auf PresentationCore hinzufügen.

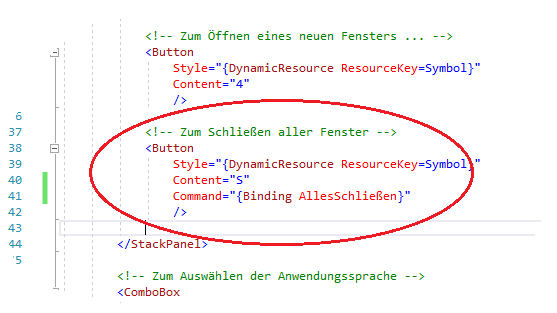
Die Frage ist, …

* Wie wird ein Ereignis an ein anderes gebunden?

<https://docs.microsoft.com/de-de/dotnet/csharp/programming-guide/events/how-to-implement-custom-event-accessors>



Die Objekte vom Typ WIFI.Windows.Befehl …

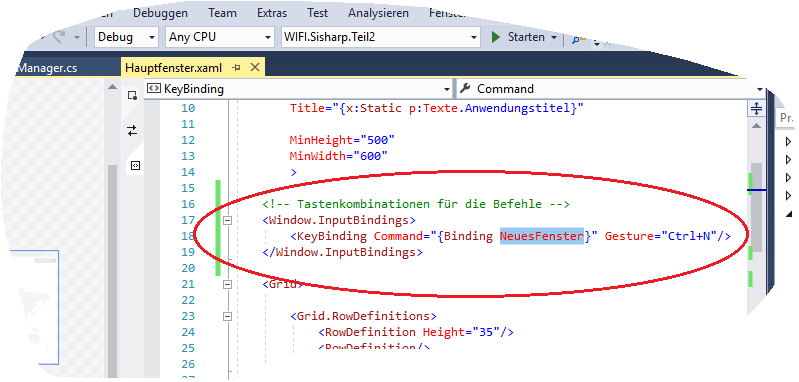


… können anschließen für Datenbindung benutzt werden.

28.2.2019

* Einen Tool-Tipp für die Symbole NeuesFenster und AlleFensterSchließen
* Im ViewModel FensterManager ein neuer Befehl „NeuesFenster“

Wie wird ein Befehl mit einer Tastenkombination verknüpft?



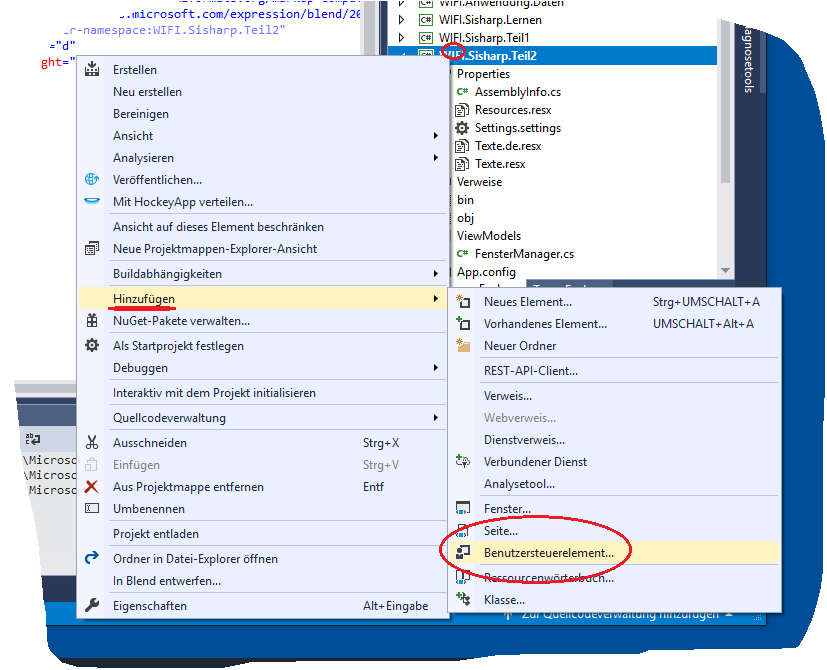
* Über die InputBindings Auflistung des Fenster.

Nächste Baustelle(n)



* Damit das Hauptfenster.xaml kontrollierbar bleibt,  
  ein eigenes Steuerelement für den Aufgabenbereich

Dazu …



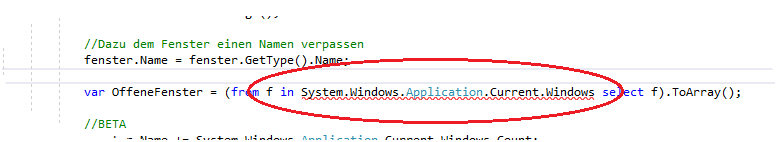
… ein neues Benutzersteuerelement hinzufügen.

Wie kommen wir zu den Daten, den Aufgaben?

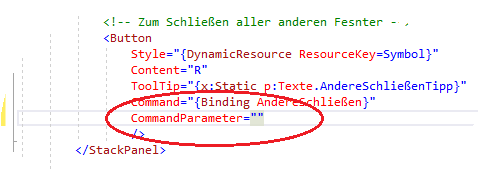
5. März 2019

Zum Finden der ersten freien Fensternummer:

Idee:

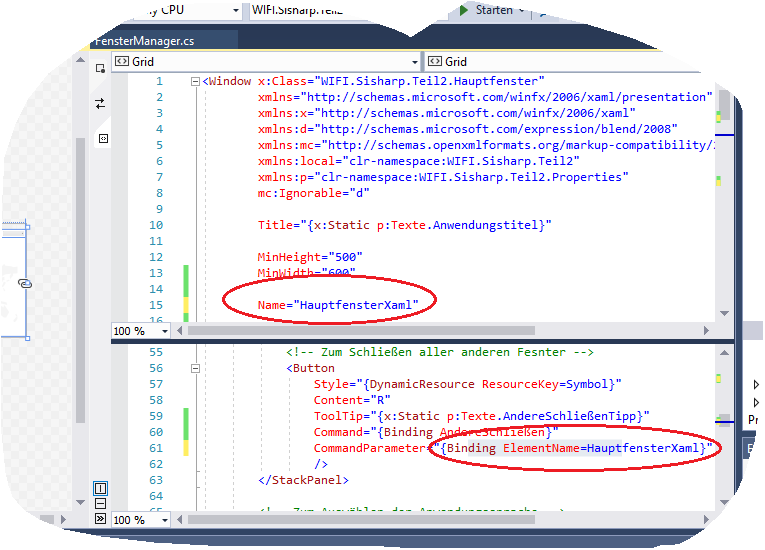
* LINQ – Sprachintegrierte Abfrage zum Finden  
    
  Geht ned ☹  
  
* Anderes Idee:
  + Eigene Liste
  + Nur die Namen der Fenster einfüllen
  + In einer Schleife solange die Fensternummer  
    erhöhen, solange die Nummer vorhanden ist

Zum Schließen aller anderen Fenster …



… wollen wir das aktuelle Fenster an die Methode AndereSchließen übergeben

Damit bei der Datenbindung auf ein anderen Xaml Element gebunden werden kann, muss dieses benannt werden.



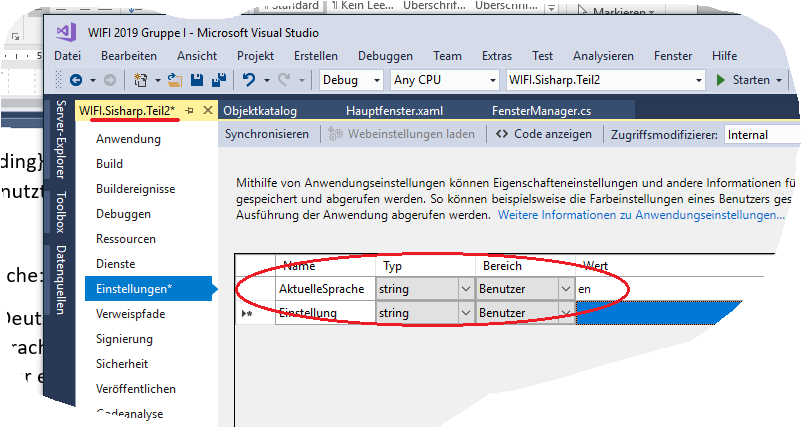
Im Rahmen von {Binding} kann anschließend das ElementName – Attribut zur internen Bindung benutzt werden.

Zur Anwendungssprache:

* Aktuell ja nur Deutsch
* Jetzt soll die Sprache benutzt werden,  
  die die Anwender einstellen

Dazu müssen wir uns die eingestellte Sprache merken…

* In der Anwendungskonfiguration

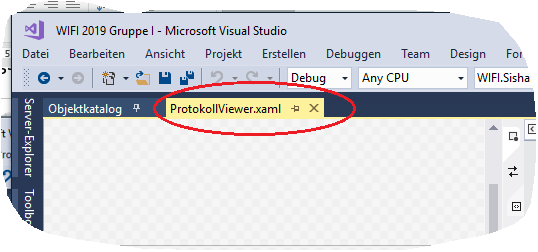


Nächste Baustelle:

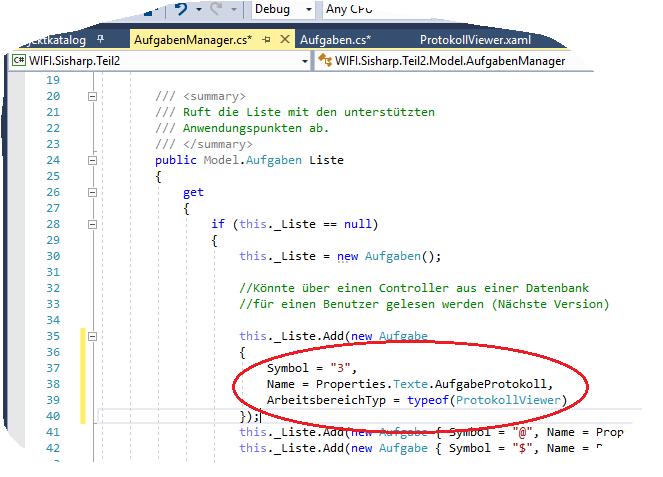
* Das Anwendungsprotokoll visualisieren

Erster Schritt:

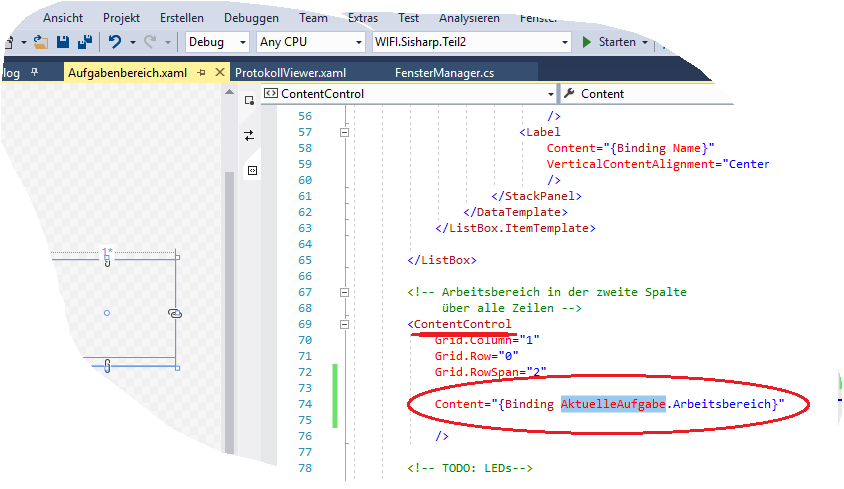
* Ein neues ProtokollViewer.xaml Steuerelement



* Das Model.Aufgabe Objekt muss den Typ des Viewers kennen

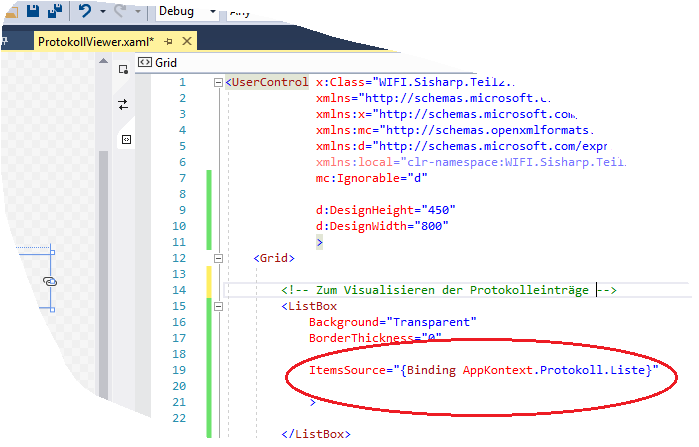


Der neue Arbeitsbereich muss …



… an das ContentControl gebunden werden.

Im neuen ProtokollViewer kann …



… die Liste an das Protokoll unserer Infrastruktur gebunden werden. Funktioniert, weil das ViewModel FensterManager im DataKontext die Infrastruktur geerbt hat.

Neben der Datenbindung ein weiterer Pluspunkt von WPF:

<https://docs.microsoft.com/de-de/dotnet/framework/wpf/graphics-multimedia/wpf-graphics-rendering-overview>

* Der Grafik Bereich
* Wird direkt auf der Grafikkarte gerechnet  
  und nicht mehr, wie Windows Forms über das Betriebssystem GDI+

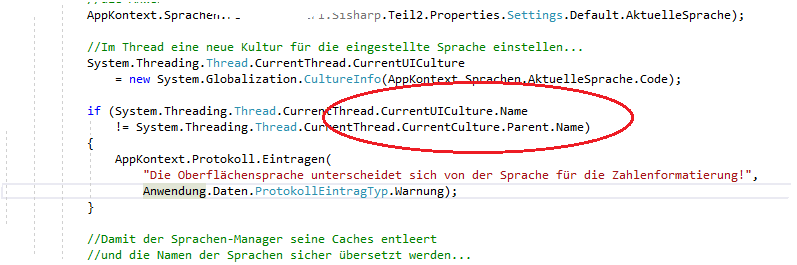
Im Protokoll soll automatisch ein Eintrag vom Typ NeueInstanz, wann die Infrastruktur ein neues Anwendungsobjekt erzeugt.

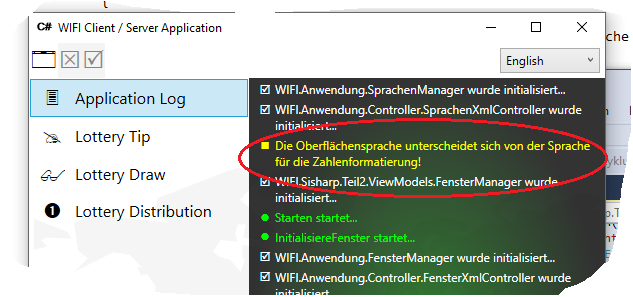
7. März 2019

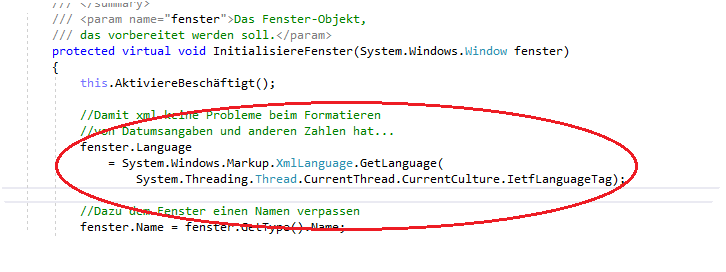
- Im ProtokollViewer einen DataTrigger für Warnungen in gelb.

Damit wir zu einer Warnung kommen:

* Einen Eintrag im Protokoll erstellen,  
  wenn sich die Oberflächensprache von der  
  Sprache, die für Zahlenformatierung benutzt wird,  
  unterscheidet



* Übrigens:  
    
    
  xml hat Probleme mit der Zahlenformatierung. Probleme mit der Sprache. Microsoft hat mit Xaml die Sprachmöglichkeiten erweitert. Und damit das funktioniert, muss dem xml die Sprache mitgeteilt werden.



* Beim Öffnen eines Fensters soll  
  eine Aufgabe standardmäßig ausgewählt sein
* Über ein Kontextmenü soll eine Aufgabe  
  von den Benutzern als Standard festgelegt werden können.
* Den Index der Aufgabe merken  
  wir uns in der Anwendungskonfiguration



Nächster Wunsch:

* Der ProtokollViewer soll bei einem neuen  
  Eintrag zum Ende blättern

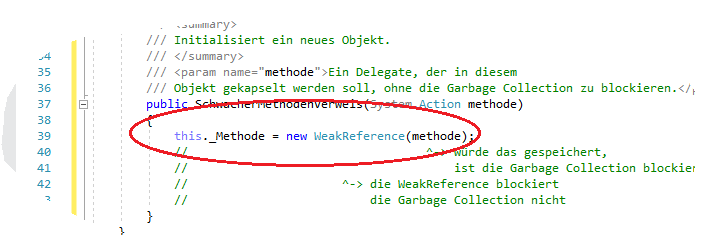
Problem:

* Methoden so sammeln, dass der Besitzer von der Garbage Collection entfernt werden kann

Lösung:

* System.WeakRefernce

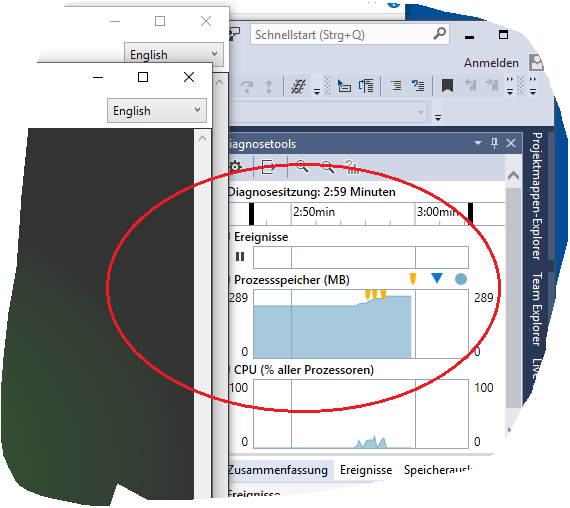
12. März 2019



Wie kommen wir zum Rückruf?

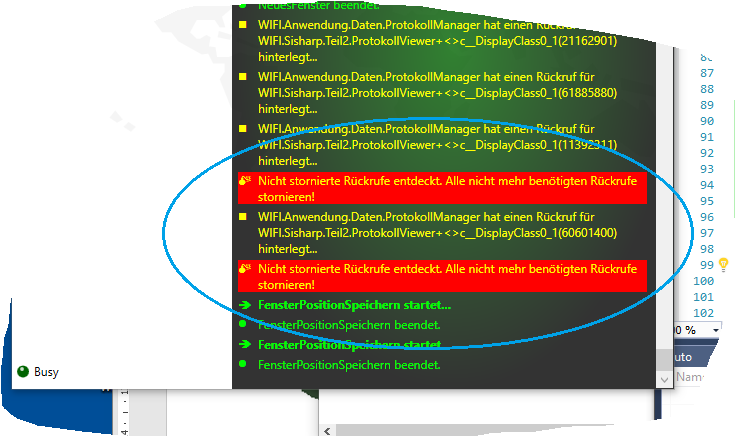
* Über das ViewModel extrem aufwändig
* Im Konstruktor vom ProtokollViewer  
  (Code behind)

Extrem wichtig:



* Den Speicherverlauf einer Anwendung im Auge behalten
* Aufpassen, dass keine „Speicherlöcher“

Nur WeakReference – Objekte …



… ermöglichen der Garbage Collection Objekte zu entfernen, obwohl die Speicheradresse woanders noch benutzt ist.

14. März 2019

Problem

„WPK/L/U/V“

Ich benötige alles vor dem ersten Schrägstrich.

Das „WPK/…“ ist ein Text, als System.String

Und muss mit diesem gearbeitet werden, hofft man, dass die String-Klasse Möglichkeiten bereitstellt.

Kürzeste Lösung:

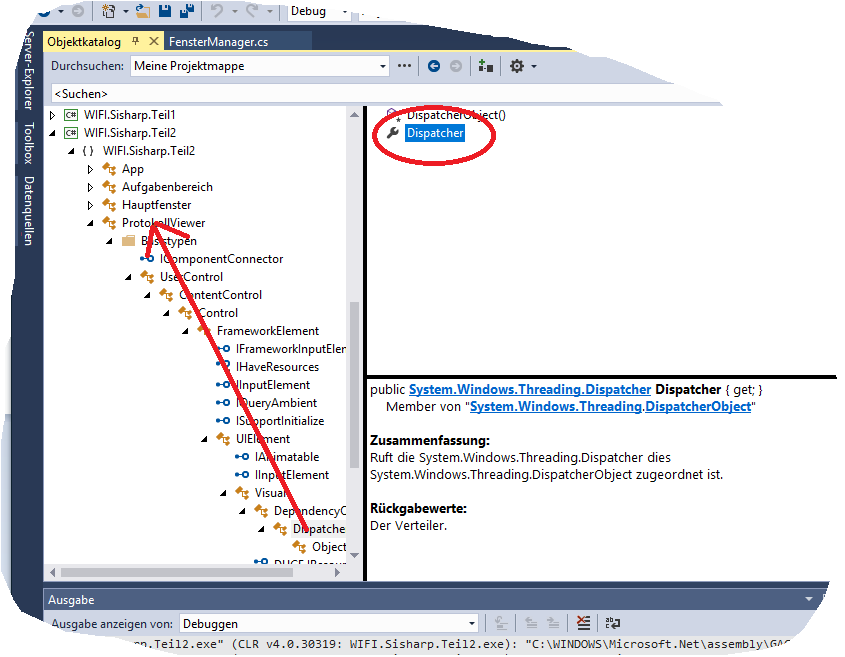
"WPK/L/U/".Split('/')[0]

Damit ein Anwendungsteil einen anderen nicht blockiert, …

* Mit Threads arbeiten

Auf der einen Seite sind Threads praktisch.

**Auf der anderen Seite die Hölle.**

* Sobald ein Thread läuft, kann dieser nur abgebrochen werden
* KEIN THREAD DARF IN DIE DATEN EINES ANDEREN THREADS GREIFEN!
* In WPF ist Multithreading von der ersten Minute an mitgedacht  
    
  Alles ist eine Erweiterung vom DispatcherObject. Dieses stellt die Dispatcher-Eigenschaft bereit, mit dem das Multithreading in WPF realisiert ist.

Microsoft hat ein (neues) Programmmuster zum  
Arbeiten mit Threads:

* Das Aufgabenbasierte asynchrone Muster  
  (TAP, task oriented asynchronous pattern)
* Das löst das   
  Ereignisorientiert Muster ab

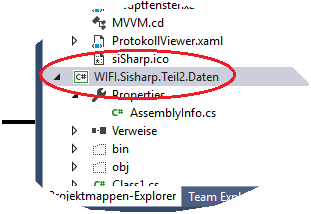
<https://docs.microsoft.com/de-de/dotnet/standard/asynchronous-programming-patterns/>

Zum restlichen Aufgabenbereich:

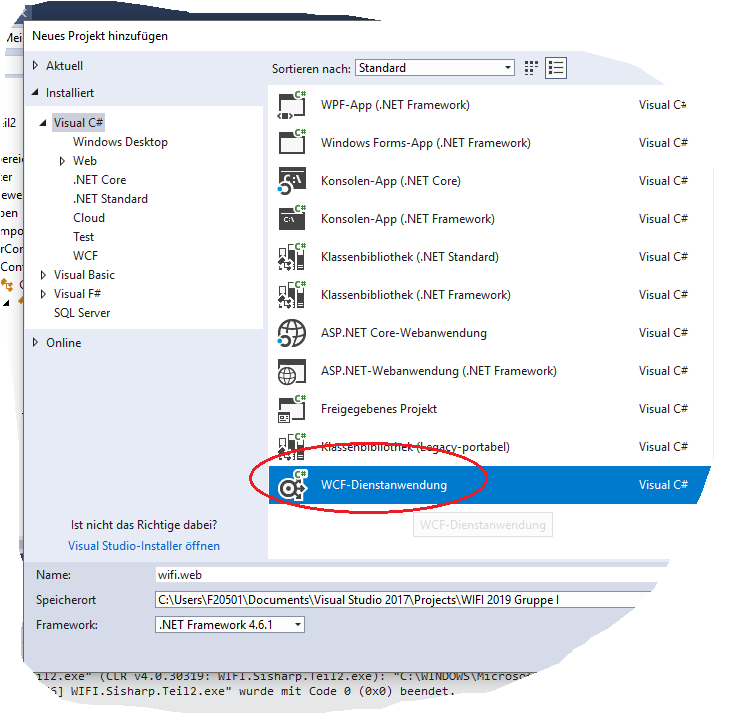
* Lottoziehungen und Lottoquicktipps in  
  einer Datenbank verwalten

Neue Assembly:

WIFI.Sisharp.Teil2.Daten

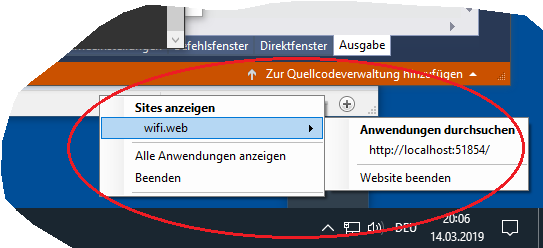


* Problem:  
    
  Diese Assembly enthält (Fiktiv) Algorithmen,  
  die eine Kunde auf keinen Fall zu Gesicht   
  bekommen soll.
* Deshalb dürfen wir im WIFI.Sisharp.Teil2  
  KEINEN VERWEIS auf diese Assembly machen
* Wir wollen die Logik von WIFI.Sisharp.Teil2.Daten  
  über einen WebDienst veröffentlichen



Voi cool:

* Wird im Visual Studio eine Web Anwendung entwickelt,  
  kann diese über den IIS – Express (wird vom Studio verwaltet)  
  sofort getestet werden



Problem:

* Damit in Zukunft unsere Anwendung läuft,  
  muss immer vorher das Studio und das Projekt gestartet werden
* Uninteressant

Lösung:

* Wenn der Dienst fertig ist, diesen  
  auf einen „echten“ Internet Information Server (IIS)  
  portieren
* Falls mit Windows Home gearbeitet wird,  
  ist dieser IIS nicht verfügbar

Deshalb:

* WIFI.Sisharp.Teils2 so implementieren,  
  dass zwischen ONLINE und OFFLINE Version  
  umgeschaltet werden kann
  + **Online**  
    Benutze Webdienst
  + **Offline**  
    Lade WIFI.Sisharp.Teil2.Daten  
    dynamisch (ohne Verweis)
* Dazu eine Konfigurationseinstellung

Neues Problem:

* Unser Webdienst am IIS muss die  
  Infrastruktur, den Anwendungskontext, erzeugen
* Falls irgendwo Fehler auftreten (am häufigsten Probleme mit Zugriffsrechten) werden diese automatisch im Anwendungsprotokoll  
  registriert
* Aber für uns „unsichtbar“

Lösung

* Der Protokoll-Manager muss die Möglichkeit haben,  
  das Protokoll in eine Textdatei zu schreiben

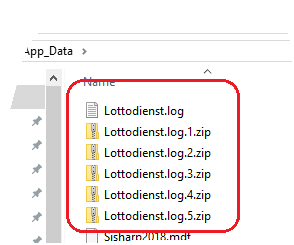
AUFPASSEN BEIM ERSTELLEN VON PROTOKOLL:

* Ned ewig und drei Tage Protokolldateien schreiben
* Irgendwann ist die Server Platte voll

Deshalb:

* Eine Methode Zusammenräumen(int maxGenerationen)

Das Ergebnis vom Zusammenräumen könnte so …



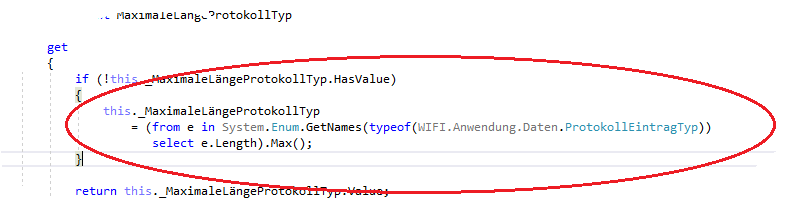
… aussehen.

Zu etwas Anderen:

* Die WIFI.Sisharp.Teil2 Anwendung darf keinen Verweis  
  auf WIFI.Sisharp.Teil2.Daten besitzen
* Das ist lästig  
  Weil beim Verweis kopiert das Studio die Assembly  
  automatisch in das Verzeichnis, wo die Assembly benötigt wird
* Was kann man tun,  
  damit wir nicht manuell die Datei  
  kopieren müssen?

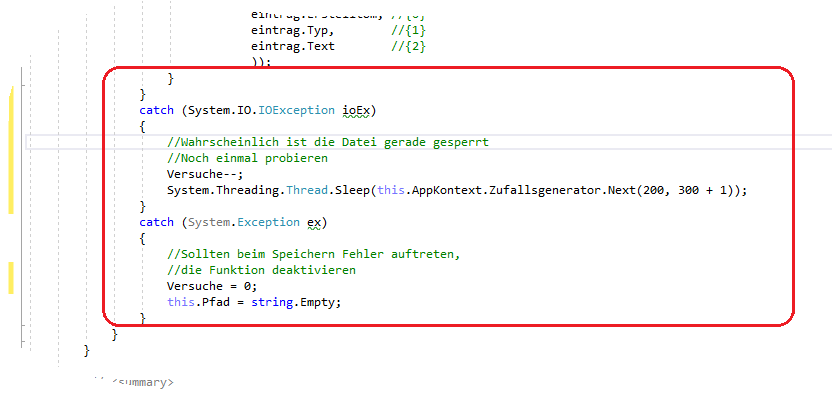
19. März 2019

* Language Integrated Query (LINQ)



* Das Universalwerkzeug für viele  
  Aufgabenstellungen im Bereich  
  der Daten, die bereits im Speicher sind

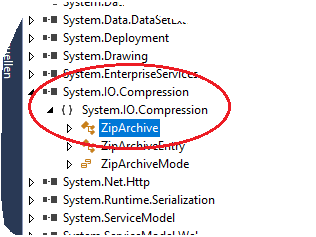
Nachtrag zu try-catch



Der catch-Bereich kann mit einer Fallentscheidung verglichen werden. Alle Fehler-Klassen sind eine Erweiterung von System.Exception. In seltenen Fällen kann notwendig sein, unterschiedlich auf Fehler zu reagieren.

Damit das funktioniert, muss System.Exception aber der letzte Zweig sein!

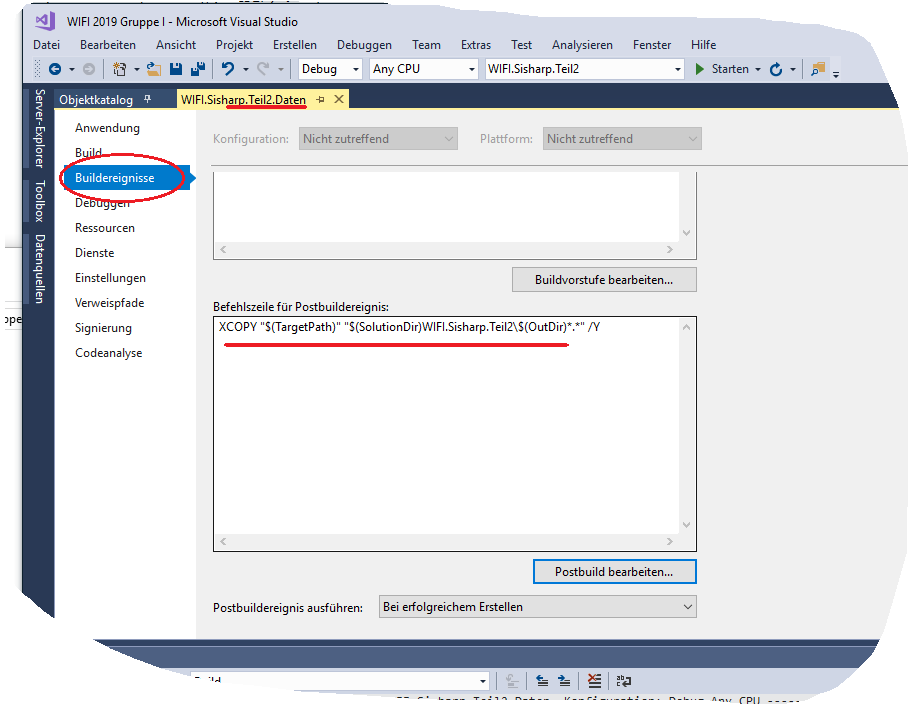
Das Zippen ist in .Net mittlerweile enthalten…



* Dazu einen Verweis auf „System.IO.Compression“

Hinweis: System.IO.Compression.FileSystem würde nur eine Datei zippen

Super – Visual Studio kann nach erfolgreichem Kompilieren …



… gewünschte Konsolenbefehle aus dem Betriebssystem ausführen.

In unserem Fall soll die Assembly „WIFI.Sisharp.Teil2.Daten.dll“ in das Hauptprojekt kopiert werden, weil wir keinen Verweis erstellen, damit später, falls der Webdienst benutzt, diese Datei nicht beim Kunden liegen muss.

Zum Webdienst…

* Den vorgeschlagenen Dienstnamen ändern

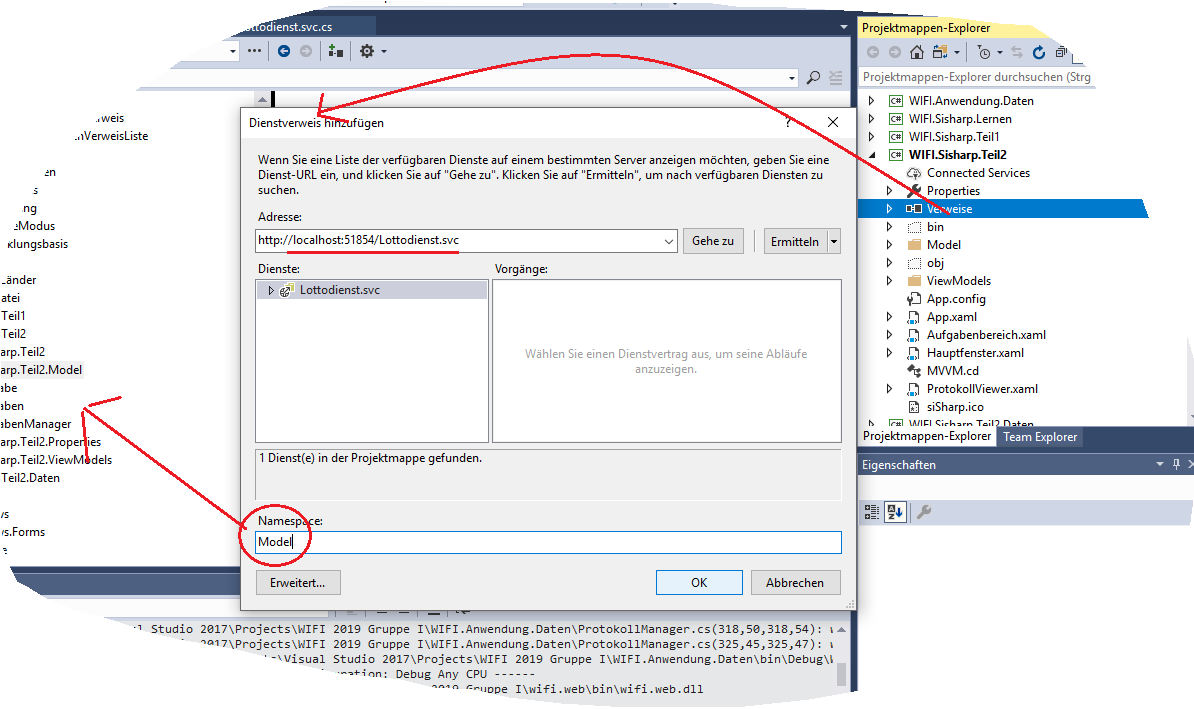
Wie kann eine Assembly eine andere benutzen?

* Mit einem Verweis

Wie kann eine Assembly einen Webdienst benutzen?

* Durch einen Dienstverweis

Visual Studio interpretiert die WSDL Xml Information und erstellt lokale Klassen



* Dabei einen Namespace angeben, unter dem die   
  Klassen aufgerufen werden sollen
* Weil wir eine M-V-VM Anwendung bauen,  
  nutzen wir für alle Klassen, die mit Daten zu tun haben,  
  den Model Namespace

Hauptfrage:

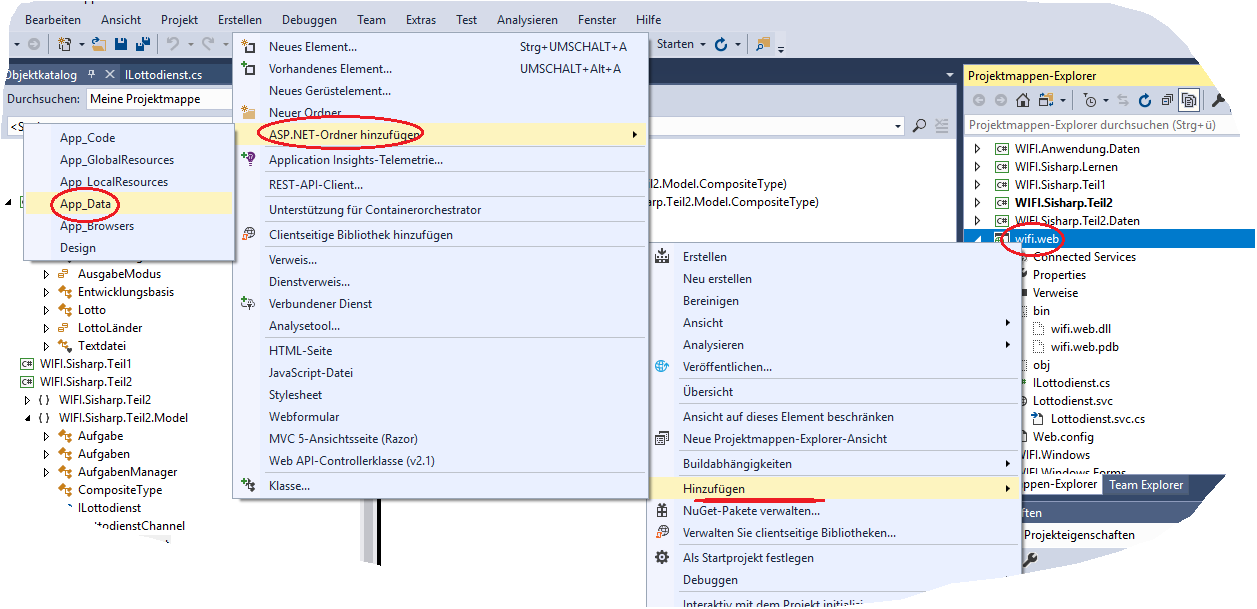
* Wo speichern wir die Daten?

Im Echtbetrieb später

* In einer Datenbank
* Datenbanken werden von   
  einem relationalen Datenbankmanagementsystem (RDBMS)  
  verwaltet, z. b.
  + Oracle (z. B. gerne von SAP)
  + Postgresql (freie, kostenlose Datenbank)
  + Microsoft Sql Server
* Eine Datenbank-Datei muss am Server „angehängt“ sein
* Das ist für uns NICHT SUPER

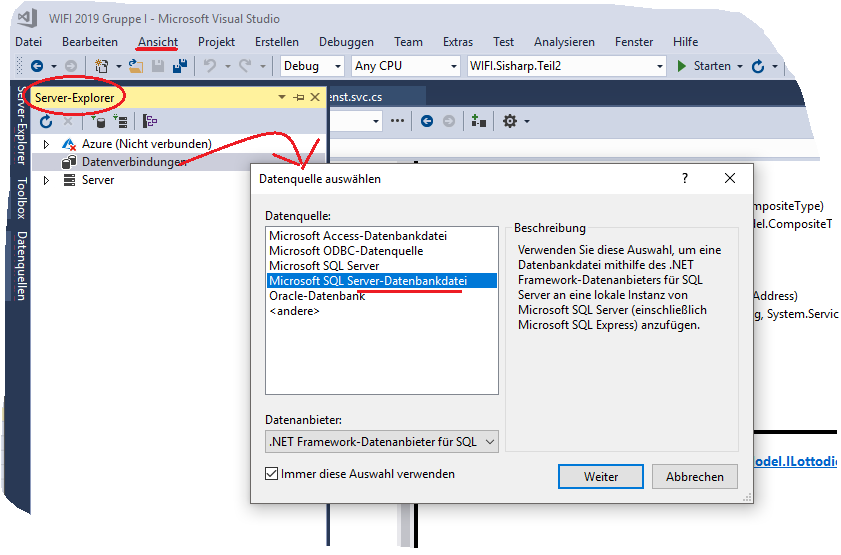
Lösung im Training:

* Der Microsoft Sql Server unterstützt (seit 2008)  
  Datenbankdateien, die nach Bedarf „dynamisch“ angehängt werden
* Das ist für uns super
* Wo legen wir die Datei hin?
  + In eine Spezialordner, den  
    der ASPX Webdienst nutzt



* Eine ASPX Anwendung, so wie unser Webdienst

nutzt für Daten den APP\_DATA Ordner



* Zum Erstellen der Datenbankdatei aus dem Studio Server-Explorer eine Datenbankverbindung für „Microsoft SQL Server-Datenbankdatei“ erstellen.
* Benutzter Datenbankname: CS2019.mdf

MDF -> „Microsoft Database File“

21. März 2019

Frage vom Herrn Schwarz zum Auswendiglernen:

* Erinnerung an die Einführung
  + Drei Problemfelder
    - Programmieren  
      Nachdenken, einen Algorithmus finden
      * Sequenz
      * Verzweigung
      * Schleifen
    - Die Algorithmen muss man tippen  
      Eine Programmiersprache, C#  
      Auswendig: Schüsselwörter
    - Für viele Aufgabenbereiche  
      gibt’s fertige Lösungen: die Klassen  
      im Rahmen der OOP  
      nur zum Verstehen
      * Erfahrung, zeitaufwändig
      * Grob weiß man die Namespaces
        + WIFI.Anwendung  
          Firmenframework
        + WIFI.Anwendung.Daten  
          Framework Erweiterung für Anwendungen  
          mit Datenbanken, z. B. mit der   
          Möglichkeit, ein Anwendungsprotokoll  
          zu führen
        + …

Zur Datenbank:

* Wir haben eine SQL Server Datenbankdatei   
  „Microsoft Database File (MDF)
* Diese wird aber nicht fix angehängt,  
  sondern dynamisch (seit SQL Server 2008)

Wo befindet sich der SQL Server?

* Der wurde auf der Maschine installiert,  
  als im Installer vom Studio „WEB Design“ aktiviert wurde
* Wo sieht man diesen?
  + Es handelt es sich um   
    eine Dienstanwendung, die ohne  
    angemeldetem Benutzer läuft
  + „Dienste“ vom Betriebssystem
* Wie findet eine eigene Anwendung die   
  Datenbank?
* Mit dem ConnectionString  
    
  Die Hölle auf Erden

Eine tolle Beschreibung aller gängigen und nicht gängigen Datenbank Verbindungszeichenfolgen…

[www.connectionstrings.com](http://www.connectionstrings.com)

Damit wir die Datenbankverbindung für unsere Anwendungen   
erleichtern..

* Eine Basisklasse DatenbankObjekt

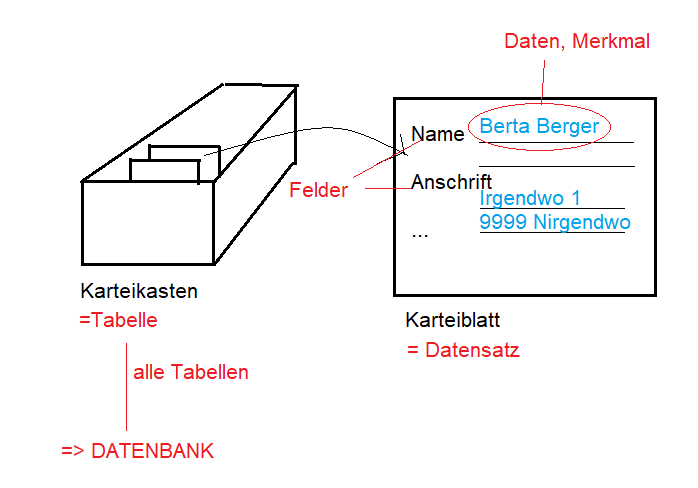
Wo soll das DatenbankObjekt benutzt werden?

* Im WIFI.Sisharp.Teil2.Daten Assembly

Was soll diese Assembly können?

* Es soll für ein Land Lotto Quicktipps berechnen können
* Diese Quicktipps sollen gespeichert werden können
* Außerdem sollen die Lottozahlen einer Ziehung  
  abgerufen werden können

Crash – Kurs Datenbanken



Grundbegriffe

* Eine Datenbank ist eine Sammlung von Tabellen
  + In der Theorie brauchte eine Firma  
    nur eine Datenbank
  + Warum?
    - Oberster Grundsatz  
      „**Dieselbe Information darf nur EINMAL gespeichert sein**“
    - Damit die Richtigkeit der Daten gewährleistet  
      werden kann („Datenkonsistenz“)
    - Sollte dieselbe Information öfter gespeichert sein,  
      spricht man von „Redundanz“
  + Die Anwendung, die eine Datenbank verwaltet,  
    muss in der Lage sein, Tabellen („Relation“) zu verknüpfen  
    (Deshalb ist Excel keine Datenbank)  
      
    „Relationale Datenbankmanagementsysteme“  
    (RDBMS)
* In einer Tabelle befinden sich Datensätze
* Im Datensatz befinden sich Felder
  + Mind. ein Feld oder eine Feldgruppe  
    darf nur eindeutige Merkmale enthalten
  + Diese Felder, die eindeutig sind, werden  
    als Schlüsselfelder bezeichnet
  + Über diese Schlüsselfelder werden Tabellen  
    miteinander verknüpft
  + Die anderen Felder heißen „Nichtschlüsselfelder“
  + Sollte ein Feld in der aktuellen Tabelle kein  
    Schlüssel, aber in der Tabelle, sein, spricht  
    von „Fremdschlüsselfeld“

Beispiele für RDBMS Anwendungen?

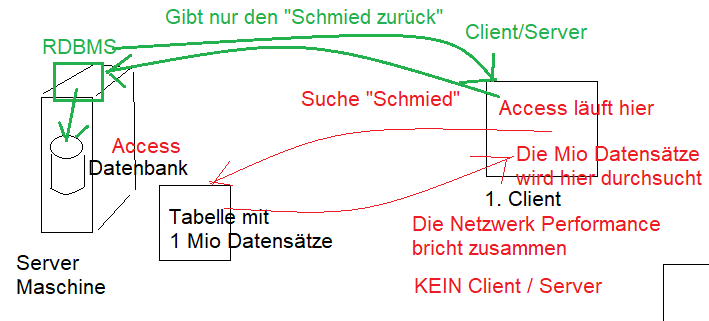
Echte Client / Serveranwendungen

* Oracle, kostenpflichtig1
* Postgresql („Post-Gres-ql“)  
  Kostenlos
* MySql, kostenlos, aber von Oracle gekauft
* Microsoft SQL Server, kostenpflichtig1

1 Bei beiden Firmen gibt es eine „Express-Version“, die  
kostenlos ist. Einschränkung: es kann nur ein Prozessor benutzt werden,  
Arbeitsspeicher wird nicht voll genutzt und die Größe der Datenbankdatei ist beschränkt

Wo ist Access?

* Das ist auch ein RDBMS
* Aber für einen Einzelplatz
* Keine Client / Serveranwendung



Access kann hauptsächlich als „Frontend“ eingesetzt mit verknüpften   
SQL Server Datenbank-Tabellen

Wie kommt man zu den Tabellen?

1. Formularanalyse

d.h. alle Zettel, Karteiblätter, … sammeln, für  
die eine Anwendung erstellt werden soll

Immer ausgefüllte …



… und die schlimmsten Fälle analysieren und die Feldnamen auflisten

z. B.

- Name

- Anschrift

- Handy

2. Normalisieren

-> Damit die Felder in Tabellen übergeführt werden können,  
müssen „Vorschriften“ eingehalten werden, die   
so genannten „Normalformen“

Die Informatik kennt ca. 15 solcher Vorschriften

In der Realität zumindest die ersten vier Normalformen einhalten!!!!

Erste Normalform (1. NF)

* Alle Felder müssen atomar sein und dürfen nur einmal vorkommen.  
  Nicht atomare Felder in die Bestandteile zerlegen und  
  mehrfachvorkommende Felder mit dem aktuellen Schlüssel  
  als Teilschlüssel in eine eigene Tabelle bringen.

z. B.

- Name

- Anschrift

- Handy

Was ist hier Schlüssel?

* Nicht einmal alle Felder miteinander

Was tun?

* Ein Surrogat definieren,  
  einen künstlichen Schlüssel

z. B. Kundennummer

Ist Name atomar?

* Nein, weil „Berta Berger“ ist Vor- und Zuname, …

Anschrift?

* Nein, weil Straße, PLZ, Ort, …

Handy? -> Das ist kein Feld, das ist eine Kommunikationseinrichtung, davon gibt’s mehrere, z. B. Diensthandy, Privathandy, …

* Kommt mehrfach vor, laut 1. NF nicht erlaubt
* Neue Tabelle „Kommunikation“  
    
  Inhalt
  + Der aktuelle Schlüssel als Teilschlüssel  
    Kundennummer
  + Zweiter Teilschlüssel  
    Laufende Nummer
  + Typ (Handy, E-Mail, …)
  + Wert

Zwischenstand

|  |  |  |  |
| --- | --- | --- | --- |
| **Kunden** |  | **Kommunikation** | |
| Nr | - | Kunde | Fremdschlüssel |
| Zuname |  | LfdNr |  |
| Vorname |  | Typ |  |
| TitelVorne |  | Wert |  |
| TitelHinten |  |  |  |
| … |  |  |  |
| Straße |  |  |  |
| PLZ |  |  |  |
| Ort |  |  |  |

Zweite Normalform (2. NF)

Eine Tabelle ist in 2. NF, wenn sie in 1. NF ist und alle Nichtschlüsselfelder vom Gesamtschlüssel abhängen. Hängt ein Nichtschlüsselfeld nur von einem Teilschlüssel ab, muss mit diesen eine eigene Tabelle erstellt werden.

Beispiel:

* Wir finden eine Lieferdetails-Tabelle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lieferschein** | **Artikel** | Name | Menge | Preis |
| **1000** | **4711** | Irgendwas | 10 | 9,9 |
| **1000** | **4712** | Nochetwas | 1 | 10 |
| **1001** | **4712** | Nochetwas | 5 | 9,5 |
| **…** |  |  |  |  |

1. NF? Ja
2. NF? NEIN
   1. Beim Preis passt’s
   2. Bei der Menge passt’s auch
   3. ABER  
      Das Nichtschlüssel Name (Artikelname) ist  
      nur vom Artikel (der Nr) bestimmt und nicht  
      vom Lieferschein, also falsch

|  |  |  |  |
| --- | --- | --- | --- |
| **Lieferschein** | **Artikel** | Menge | Preis |
| **1000** | **4711** | 10 | 9,9 |
| **1000** | **4712** | 1 | 10 |
| **1001** | **4712** | 5 | 9,5 |
| **…** |  |  |  |

Artikel

|  |  |
| --- | --- |
| **Artikel** | Name |
| **4711** | Irgendwas |
| **4712** | Nochetwas |
|  |  |

Dritte Normalform (3. NF)

Eine Tabelle ist in 3. NF, wenn sie in 2. NF ist und kein Nichtschlüsselfeld ein anderes Nichtschlüsselfeld bestimmt. Sollte ein Nichtschlüsselfeld ein anderes Nichtschlüsselfeld bestimmen, ist mit diesen eine eigene Tabelle zu erstellen.

Beispiel:

* Wir finden eine neuere Lieferscheindetails-Tabelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lieferschein** | **LfdNr** | Artikel | Name | Menge | Preis |
| **1000** | **1** | 4711 | Irgendwas | 10 | 9,9 |
| **1000** | **2** | 4712 | Nochetwas | 1 | 10 |
| **1001** | **1** | 4712 | Nochetwas | 5 | 9,5 |
| **…** |  |  |  |  |  |

1. und 2. NF ist erfüllt

Problem mit den Nichtschlüsselfeldern Artikel und Name, weil das Nichtschlüsselfeld Artikel bestimmt das andere Nichtschlüsselfeld Name. Verboten. Eigene Tabelle

Artikel

|  |  |
| --- | --- |
| **Artikel** | Name |
| **4711** | Irgendwas |
| **4712** | Nochetwas |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lieferschein** | **LfdNr** | Artikel | Menge | Preis |
| **1000** | **1** | 4711 | 10 | 9,9 |
| **1000** | **2** | 4712 | 1 | 10 |
| **1001** | **1** | 4712 | 5 | 9,5 |
| **…** |  |  |  |  |

Fehlt noch die vierte Normalform…

Vierte Normalform („Boyce Codd“)

Sollte man in einer Spalte, in einem Feld, Merkmale als „Schlüssel“ verwenden, für diese Merkmale eine eigene Tabelle mit einem Surrogat erstellen.

Beispiel:

|  |  |
| --- | --- |
| **Kommunikation** | |
| Kunde | Fremdschlüssel |
| LfdNr |  |
| Typ |  |
| Wert |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Kunde** | **LfdNr** | Typ | Wert |
| **1** | **1** | Telefon privat | +43 0000 1234 |
| **1** | **2** | Telefon firma | +43 0000 12345 |
| **1** | **3** | Handy | +43 9999 43210 |
| **2** | **1** | Handy | +43 9999 43211 |
| **2** | **2** | E-Mail | name@firma.dom |
| **…** | **…** | … | … |

1. NF -> Ja
2. NF -> Ja
3. NF -> Ja

Und trotzdem gibt’s ein Problem:

* Der Begriff „Handy“ ist negativ behaftet
* Auftrag: „Handy“ durch „Mobil“ zu ersetzen
* Wenn alles richtig ist, dürften wir das nur EINMAL MACHEN MÜSSEN

Lösung

* Eigene Tabelle „Kommunikationstypen“

|  |  |
| --- | --- |
| Typ | Name |
| 1 | Telefon privat |
| 2 | Telefon firma |
| 3 | ~~Handy~~ Mobil |
| 4 | E-Mail |
| … |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Kunde** | **LfdNr** | Typ | Wert |
| **1** | **1** | 1 | +43 0000 1234 |
| **1** | **2** | 2 | +43 0000 12345 |
| **1** | **3** | 3 | +43 9999 43210 |
| **2** | **1** | 3 | +43 9999 43211 |
| **2** | **2** | 4 | name@firma.dom |
| **…** | **…** | … | … |

Falls Fehler in der Normalisierung sind,   
fallen einen diese tausende Male später auf den Kopf!

HIER FEHLEN BEZIEHUNGEN

Wie heißt die genormte Abfragesprache, um mit dem   
Inhalt von Datenbanken zu reden?

* Seit 1970
* Standard von IBM
* „Structured Query Language“ (SQL)

Das Besondere von SQL

* Deklarative Sprache
* Keine funktionale, wie z. B. C#  
  In C# muss man tippen, wie man etwas will
* In SQL muss man tippen,  
  was man will  
  SQL kümmert sich darum, wie das gemacht  
  wird, was sie wollen

Das hat Microsoft lange nicht können

Wer hat das perfekt gekonnt?

* Fox Pro

Erfinder vom „Rushmore Algorithmus“

* Von Microsoft gekauft

Woraus setzt SQL zusammen?

* Dem Datendefinitionsteil  
  „Data Definition Language“ (DDL)
* Dem Datenmanipulationsteil  
  „Data Manipulation Language“ (DML)

DDL

1. Erstellen von Objekten  
   CREATE … z. B. CREATE TABLE
2. Ändern von Objekten  
   ALTER … z. b. ALTER TABLE
3. Löschen von Objekten  
   DROP … z. B. DROP TALBE
4. …

Übrigens:

* SQL ist nicht case sensitiv, d.h. Groß-/Kleinschreibung ist wurscht
* Üblicherweise werden diese aber groß getippt

DML

1. Holen von Daten  
   SELECT -> ungefährlich
2. Einfügen von Daten  
   INSERT -> wurscht
3. Ändern von Daten  
   UPDATE -> nie ohne WHERE zum Einschränken der Daten
4. Löschen von Daten  
   DELETE -> nie ohne WHERE zum Einschränken der Daten

UPDATE und DELETE sind tödlich.

Weil jedes der erwähnten Schlüsselwörter hat zahlreiche Detailschlüsselwörter, z. B. WHERE zum Einschränken von Daten

SQL ist genormt vom ANSI Institut

Jeder Hersteller hat seinen eigenen Dialekt.

Der Microsoft Dialekt heißt …

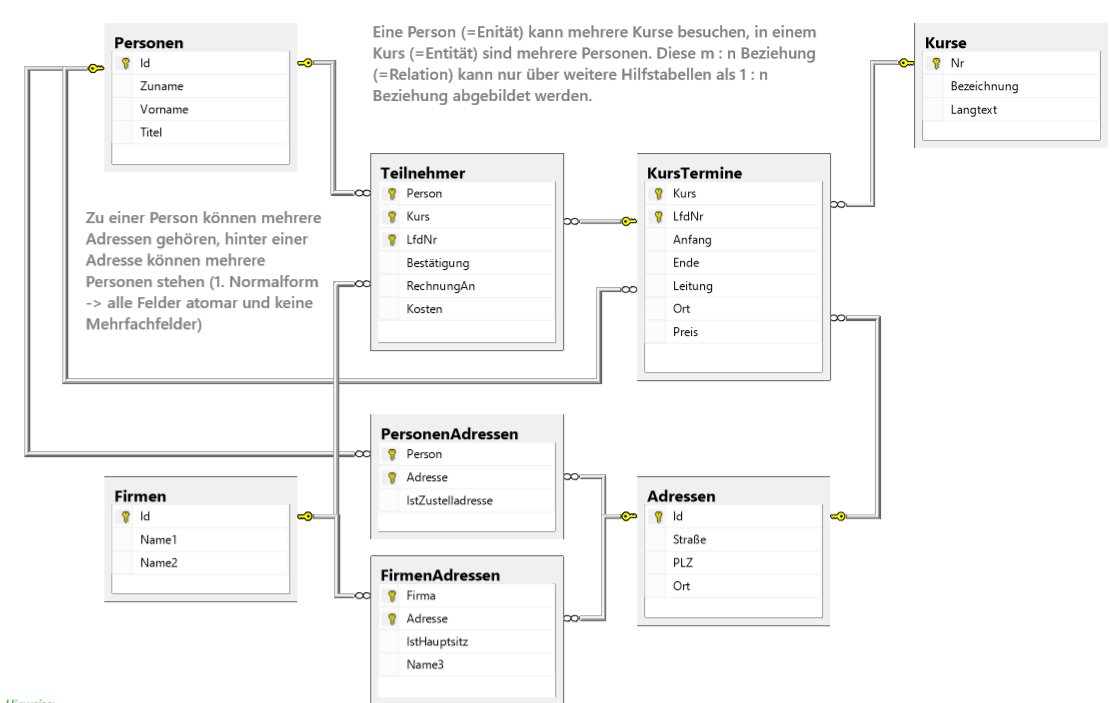
* Transact SQL oder kurz T-SQL

Die Dokumentation:

* <https://docs.microsoft.com/de-de/sql/t-sql/language-reference?view=sql-server-2017>

26. März 2019

Zur Hausübung mit dem Datenmodell…



Auszugsweise könnte das so wie abgebildet laufen…

* IST BEI WEITEM NICHT VOLLSTÄNDIG!!!

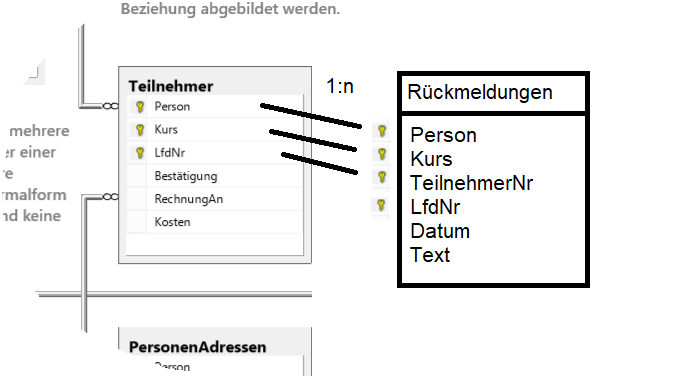
Frage:

* Wann ist’s besser anstelle eines zusammengesetzten Schlüssels einen künstlichen Schlüssel (Surrogat) zu erstellen?

z. B. die Teilnehmer-Tabelle in der Abbildung hat einen Schlüssel aus drei Teilschlüssel (Person, Kurs und die laufende Nummer)

Momentan tragbar, weil nicht alle drei Teilschlüssel in einer weiteren Detailtabelle benötigt werden.

Würden z. B. auch die Rückmeldungen von einer teilnehmenden Person gesammelt…



Die Abbildung zeigt, dass von der Teilnehmer-Tabelle zur Detailtabelle Rückmeldungen eine Beziehung aus drei Teilschlüssel benötigt würde.

In diesem Fall wäre ein Surrogat für die Teilnehmer Tabelle überlegenswert

Noch nicht erwähnt, aber bei der Hausübung vorkommen…

> Das Ergebnis ist ein  
  
Entity-Relationship (ER) Diagramm

„Entitäten“ stehen untereinander in Beziehung:

Folgendes wird unterschieden:

**1 zu 1 Beziehung**  
Falls in einer Tabelle für jeden Schlüssel genau ein Datensatz in einer zweiten Tabelle ist. In diesem Fall die Felder in EINE Tabelle. D.h. 1 : 1 Beziehung wird man (kaum) finden.

*Das Einzige, was in einer relationalen Datenbank abgebildet werden kann…*

**1 zu n Beziehung**

D.h. für einen Datensatz der linken Tabelle kann in einer zweiten Tabelle, kein Datensatz, einer oder beliebig viele Detailsätze enthalten sein.

*In der Realität am häufigsten…*

**m zu n Beziehung**

Kann von keiner relationalen Datenbank direkt abgebildet werden. Hier sind Hilfstabellen notwendig, über die die m : n Beziehung als 1 : n realisiert wird,   
z. B. ein Person (Entität) kann mehrere Trainings (Entität) besuchen, also 1 : n. Umgekehrt sind in einem Training mehrere Personen, also wieder 1 : n. In Summe als stehen die Entitäten „Personen“ und „Trainings“ in einer m : n Beziehung. Mind. eine weitere Zuordnungstabelle, z. B. PersonBesuchtTraining ist notwendig.

Zur Datenbankprogrammierung:

* Diese läuft über System.Data  
  und untergeordnete Namespaces,  
  z. B. System.Data.SqlClient  
  falls mit Sql Server gearbeitet wird
* Für andere Datenbanken, z. B.  
  Oracle, … muss vom Hersteller  
  die entsprechende Assembly downgeloaded  
  und verwiesen werden

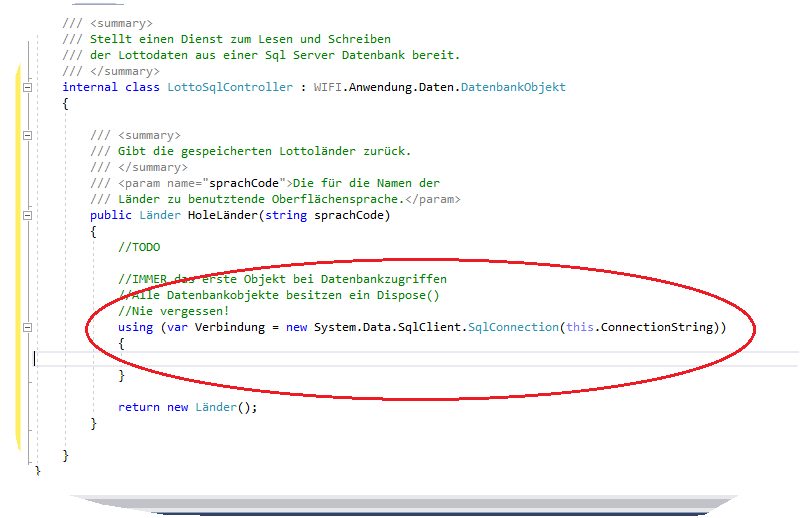
Zum Erstellen von Verbindungszeichenfolgen für den  
Sql Server gibt es System.Data.SqlClient.SqlConnectionStringBuilder

-

Zu unserer Datenbankanwendung:

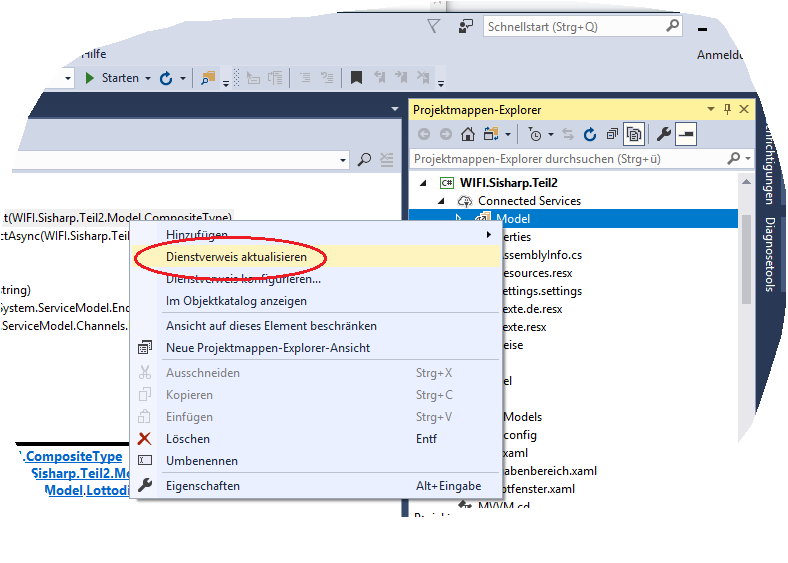
* Wir wollen Lotto Daten für mehrere

Länder verwalten



Die Abbildung zeigt, wie mit dem ConnectionString ein ConnectionObjekt gebaut wird…

Hinweis:

* Immer wenn sich der Inhalt von einem Web-Dienst  
  geändert hat, …  
    
    
  … den Dienstverweis aktualisieren.

Zum Webdienst (eine ASPX Anwendung)

* Die läuft auf „einem anderen“ Rechner (Realität)
* Wird für jeden Benutzer neu initialisiert  
  (Standard)
* Es gibt aber auch die Möglichkeit,  
  die gesamte Anwendung zu cachen

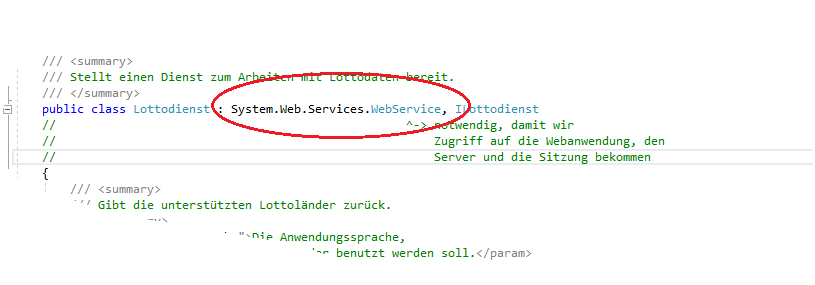
Aktuell erweitert aber unser …



… Dienst System.Objekt.

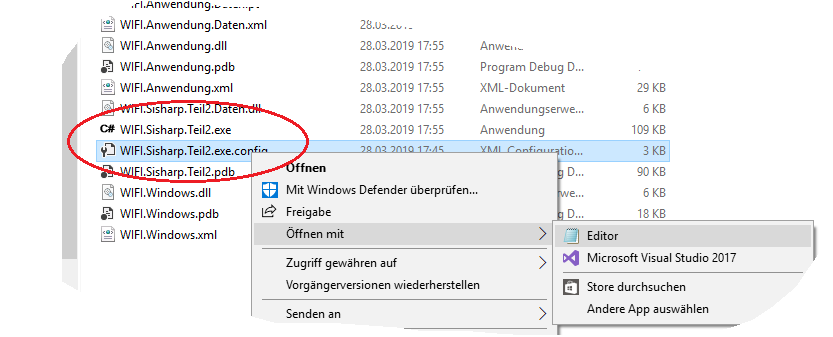
Hier haben keinen Zugriff auf die ASPX Anwendung im Hintergrund.

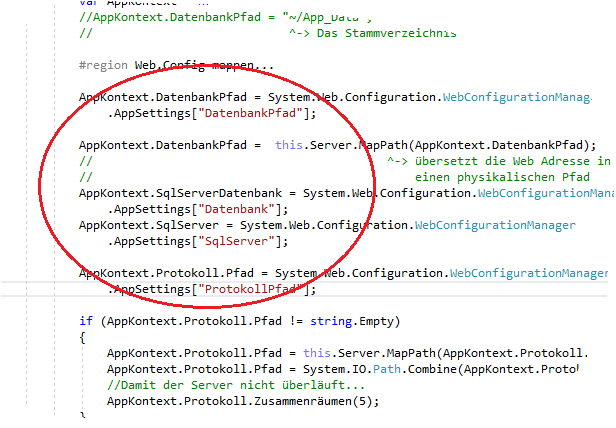
* Wir benötigten eine andere Basis



28. März 2019

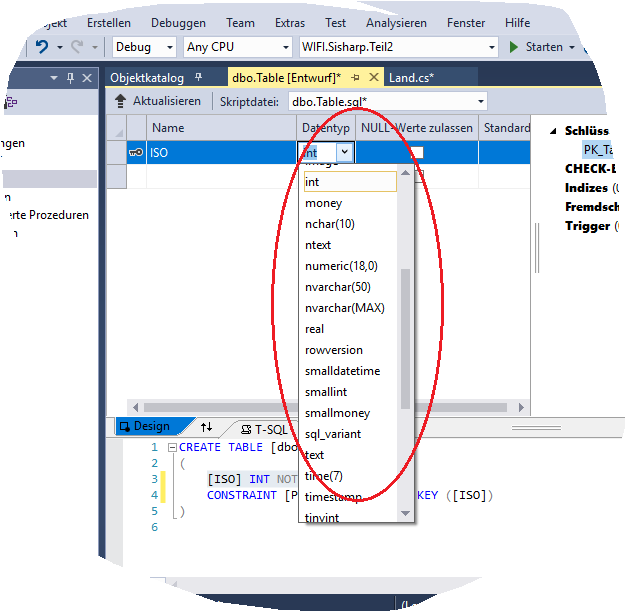
Wir wissen…

* „normale“ .Net Anwendungen werden  
  über die Anwendungskonfiguration gesteuert  
    
  
* ASPX – Anwendungen werden über die  
    
  web.config  
    
  gesteuert.



Mit System.Web.Configuration.WebConfigurationManager können die Einstellungen übernommen werden.

Tabellen erstellen



Für jedes Feld wird ein Datentyp benötigt.

Die Datenbank-Datentypen entsprechen nicht den .Net Datentypen

Die Liste der Datentypen für den Sql Server …

<https://docs.microsoft.com/de-de/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017>

z. B. Zeichenfolgen

.Net System.String, UNICODE (2 Byte Zeichensatz)

Im SQL oft…

Char -> 1 Byte Zeichensatz, ANSI, NICHT MEHR BENUTZEN!

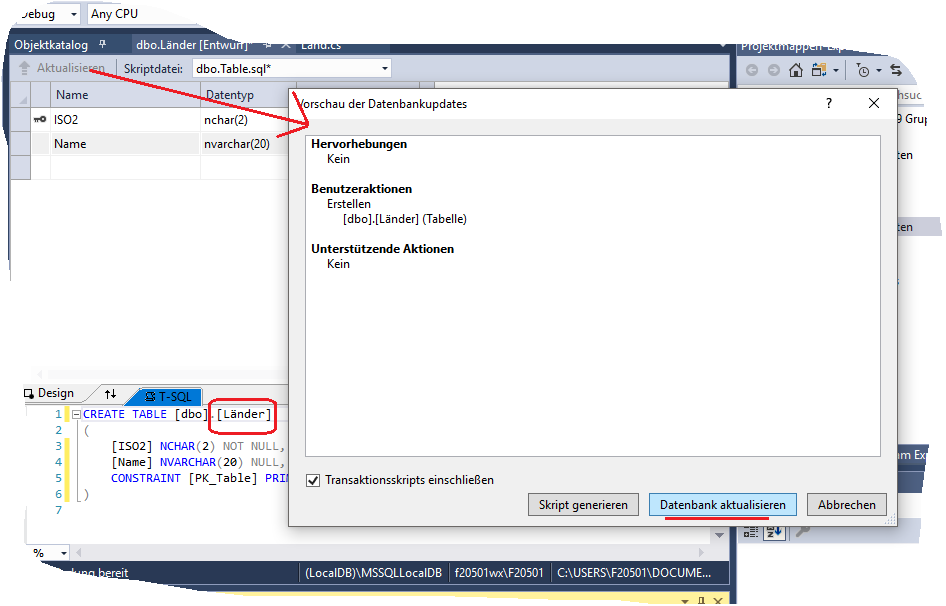
Unicode („National“, kurz N)

NCHAR -> 2 Byte Zeichensatz, UNICODE

* Problem: z. B. NCHAR(10) (max. 10 Zeichen, Speicherbedarf 20 Byte)  
    
  Wird hier „WIFI“ gespeichert, 4 Zeichen, bekommt man  
  beim Abrufen „WIFI······“  
    
  FIXE Länge, d.h. es werden beim Holen (SELECT) immer alle Zeichen geliefert, die nicht benutzen mit Leerzeichen

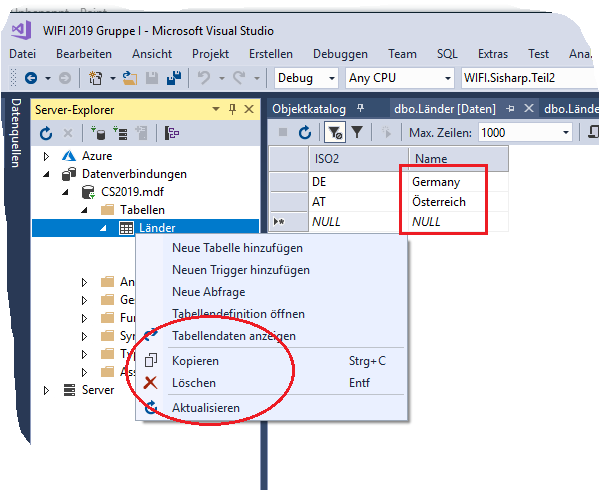
Lösung

* Variable Länge  
    
  NVARCHAR(10)  
    
  Wird hier „WIFI“ gespeichert, wird auch nur „WIFI“ geholt.   
  Max. 10 Zeichen



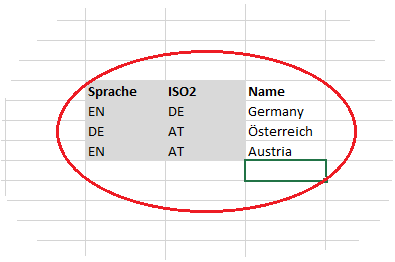
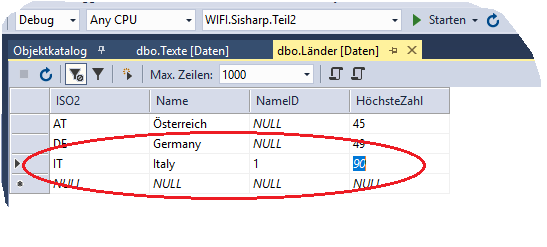
Die Abbildung zeigt, wie im DDL die Tabelle benannt und über „Aktualisieren“ auf die Datenbank abgesetzt wird.

Werden anschließend die Tabellen aktualisiert …



… ist die Tabelle sichtbar und können über „Tabellendaten anzeigen“ Informationen getippt werden.

Beim Inhalt der Spalte „Name“ gibt’s mehrere Probleme…

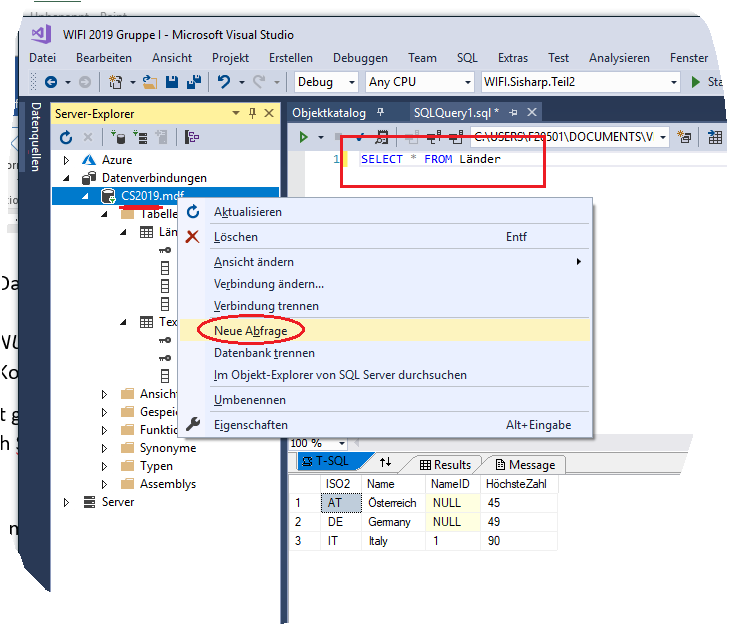
1. Die Bezeichnung „Name“ ist überall auch eine „Eigenschaft“  
     
   Deshalb motzt SQL später mit einer Warnung.  
   Bei Microsoft T-SQL müssen Bezeichnungen, die  
   eine Bedeutung haben können, in **eckige Klammer** geschrieben  
   werden, z. B. [Name]  
     
   In den Feldnamen keine Leer- und Sonderzeichen, also Bindestrich, …  
   In solchen müssen immer eckige Klammern benutzt werden.  
   Das nervt.
2. In einer Spalte unterschiedliche Sprachen.  
     
   Die Mehrsprachigkeit in der Datenbank   
   ist nicht leicht zum Umsetzen.  
     
   Wir haben keine Ressource-Dateien.  
     
   Wir könnten eine zusätzliche Spalte „Sprache“   
   erstellen und diese in den Schlüssel aufnehmen  
     
   KEINE GUTE IDEE!  
     
     
   Wird hier in deutscher Sprache gearbeitet,  
   fehlt der Eintrag für Deutschland.  
     
   Das andere Problem, in solchen Tabellen gibt’s viele Spalten.  
   Die meisten sind nummerisch, also Zahlen. Zahlen müssen nicht übersetzt werden.  
     
   Der Trainer bevorzugt folgende Technik:  
     
   Eine eigene Tabelle für „Texte“  
     
     
   - Übersetzungen stehen in einer eigenen Tabelle und sind mit einem Schlüssel verknüpft.  
     
   Die eigentliche Spalte „Name“ wird als Standard benutzt, wenn keine Übersetzung vorhanden ist.
3. Das schlimmste Wort im Datenbankbereich  
     
   *NULL*, d.h. ein Feld ist nicht initialisiert.  
   Kommt in einem SQL Ausdruck *NULL* vor, ist alles *NULL*

Im .Net gibt’s nur eine Klasse, mit der auf *NULL* geprüft werden kann,   
nämlich System.DBNull und die Eigenschaft Value.

Nachdem jetzt Tabellen mit Daten existieren, können mit …

* SELECT

… beginnen.



Innerhalb vom Studio auf der verbundenen Datenbank „Neue Abfrage“

Zum Holen: SELECT

Alle Felder: Stern \*  
(DARF NIE IN EINER PRODUKTIONSABFRAGE BENUTZT WERDEN!)

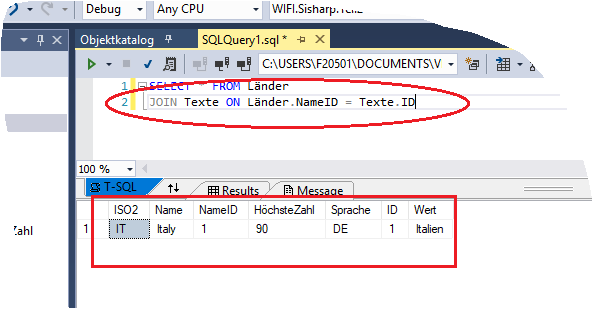
* Erster Fehler:  
  Zu viele Daten holen
* Immer nur die Felder, die benötigt werden

FROM: Definiert die Datenquelle

Im Rahmen vom „FROM“ können mit dem JOIN Zusatz Tabellen verknüpft werden. Dazu werden die Schlüssel bzw. Fremdschlüsselfelder benutzt

Standard

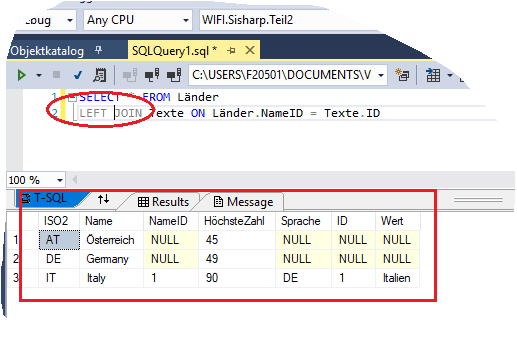
[INNER ]JOIN



* Im Ergebnis nur die Datensätze, wo der benutzte Schlüssel  
  in beiden Tabellen vorkommt

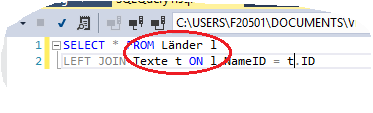
In den meisten Fällen möchte man aber alle Datensätze der einen Tabelle (die linke Tabelle) und nur dann etwas aus der andern (der rechten Tabelle), wenn ein Datensatz vorhanden

LEFT JOIN



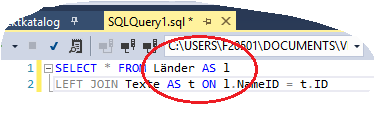
Tipp:

* Damit die Ausdrücke „Länder.NameID = Texte.ID“ kürzer werden,  
  mit Aliases arbeiten



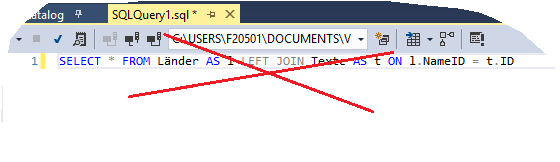
Aber nicht direkt nach einer Bezeichnung das Alias tippen.

* UNBEDINGT immer das optionale AS benutzen,  
  damit später Tippfehler, z. B. fehlende Beistriche  
  rascher erkannt werden



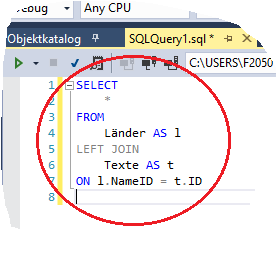
Das AS – Schlüsselwort ist fast überall erlaubt.

NIE …

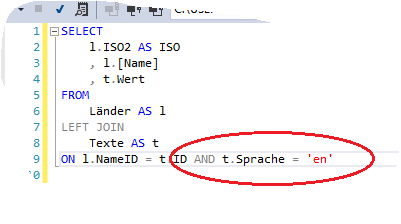


… SQL Anweisungen in einer Zeile (einer „Wurscht“) schreiben.

Immer mit Zeilenumbrüchen und Einrückungen sauber strukturieren



Im Rahmen vom ON beim JOIN können …



… mit **AND** bzw. **OR** Bedingungen verknüpft werden.

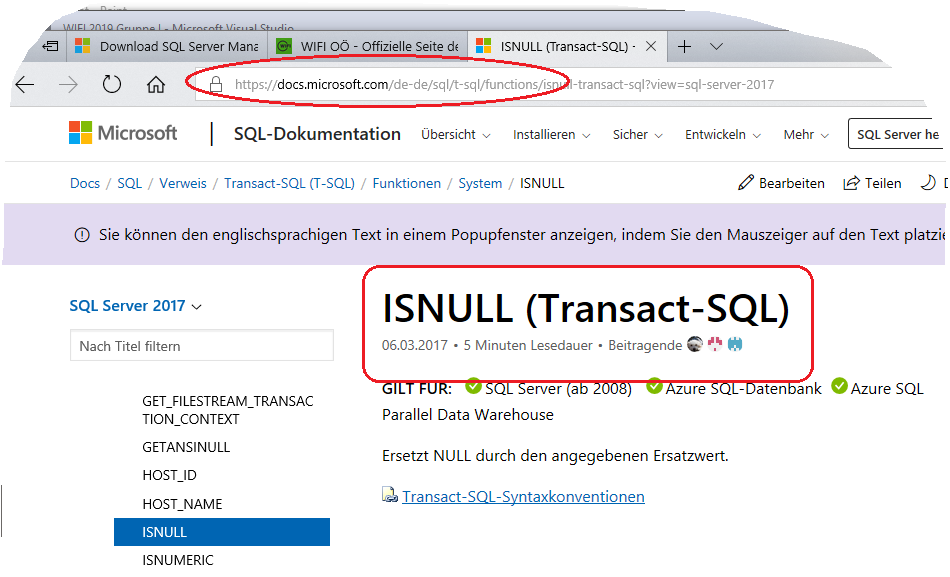
Nur **ein Gleich** beim Vergleich und **Text unter Hochkomma**.

* Microsoft Datenbanken sind standardmäßig  
  auf „**CI**“ (case insensitiv) und nicht „**CS**“ (case sensitiv) eingestellt.

(Kann nur im „großen“ Sql Server Managment Studio konfiguriert werden. Das ist übrigens kostenlos und kann auch ohne Sql Server installiert werden)

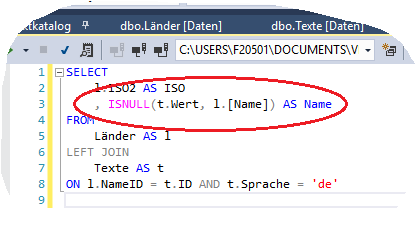
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>

Wie jedes RDBMS unterstützt auch der Sql Server und T-Sql eine Reihe von Funktionen…



…, z. B. ISNULL um das schlimmste Wort im Datenbankbereich   
in den Griff zu bekommen.

Folgende Abbildung zeigt, …



… wie bei fehlender Übersetzung die Originalbezeichnung zurückgegeben wird.

ISNULL entspricht der Excel (Tabellen-)Funktion WENNFEHLER

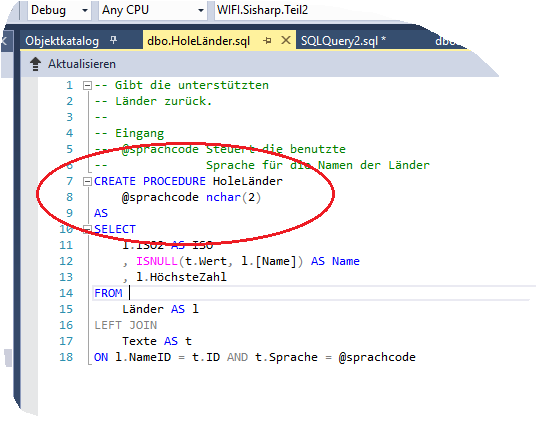
Wir haben keine Lust, diese Anweisung im C# neu zu tippen.

Wichtig:

* KEINE SQL AUSDRÜCKE IM CODE!
* Diese gehören in die Datenbank!
* Müssen jedes Mal neu interpretiert werden.

Wie kann eine Datenbank SQL Ausdrücke speichern?

1. Sicht (View, Abfrage)  
     
   Hier sind aber keine Parameter möglich
2. Gespeicherte Prozedur  
   In diesen können ganze SQL „Anwendungen“ versteckt werden  
   Diese erlaubten Parameter  
   Diese werden vom Sql Server nur einmal interpretiert  
   und gecachet.

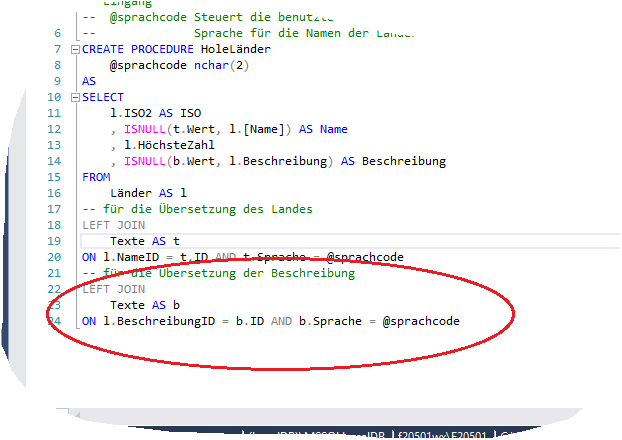


WICHTIG:

* Alle Parameter / Variablen in T-SQL   
  müssen mit einem @ Klammeraffen beginnen!
* UND – natürlich wird sauber kommentiert!

2. April 2019

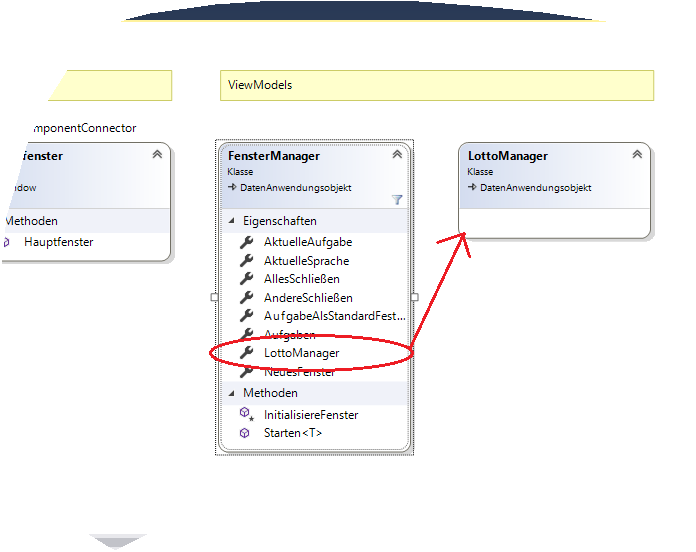
In der Tabelle Länder soll auch eine Bezeichnung für das Spiel hinterlegt werden, z. B. „Lotto 6 aus 45“ für Österreich (engl. „Lottery 6 out of 45“)



* Die Abbildung zeigt, dass in einer Abfrage dieselbe Tabelle öfter benutzt werden kann. In diesem Fall muss mit einem Alias gearbeitet werden.

Anschließend kann „HoleLänder“ im Controller fertiggemacht werden.

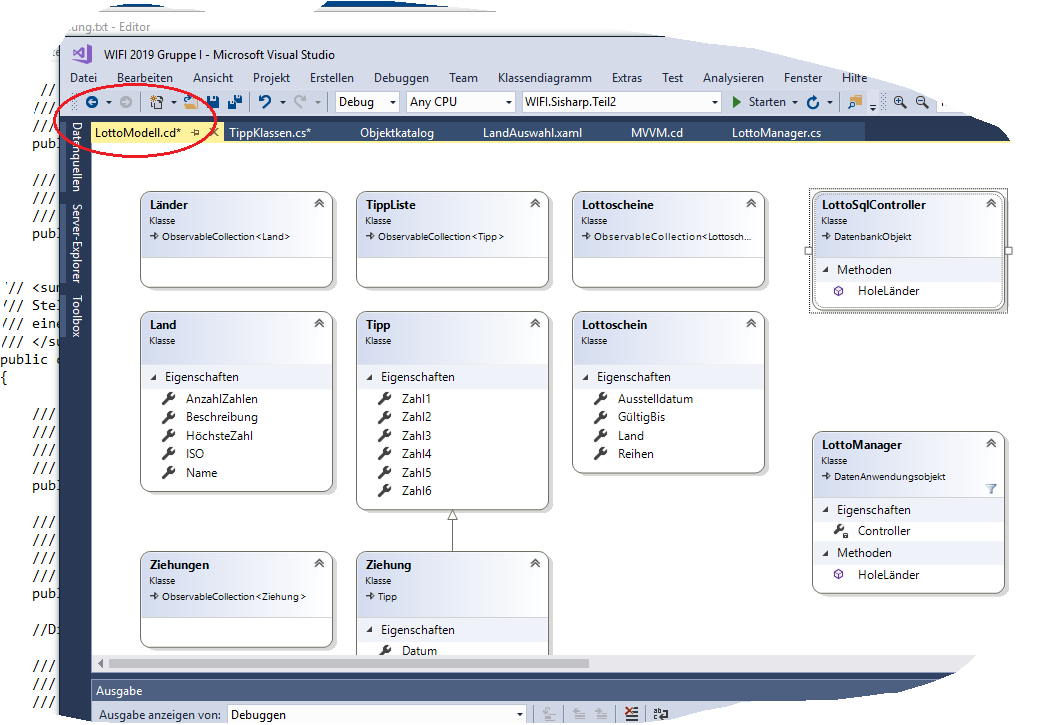
Zurück zur Oberfläche…



* Damit die FensterManager Klasse nicht zusätzlich aufgeblasen wird,  
  ein eigenes ViewModel zum Steuern der Oberfläche im Bereich QuicktippViewer, …

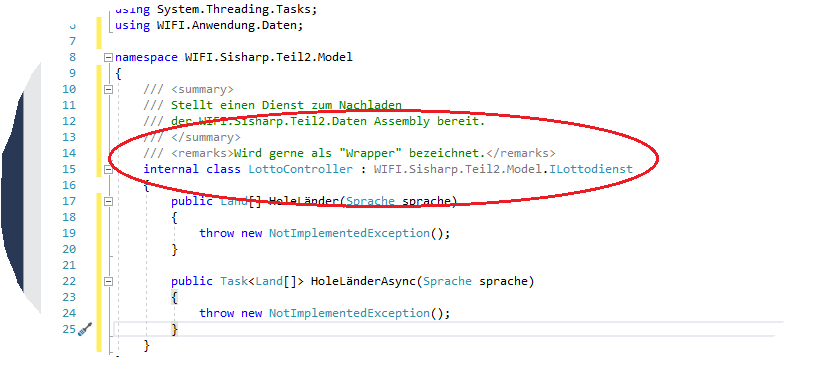
4. April 2019

Zum Aufwärmen …



… Klassen für das Lottomodell definieren.

Damit unsere Anwendung einmal online mit dem Webdienst und das andere Mal offline betrieben werden kann, …



… einen lokalen LottoController entwickeln, der all das kann, was der Webdienst anbietet.

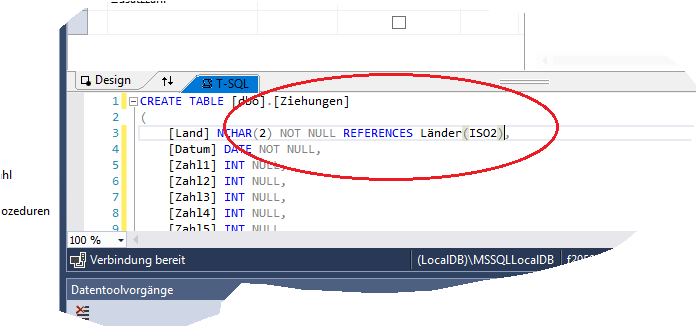
Damit das Holen der Länder …



… den Rest der Anwendung nicht blockiert, wird die asynchrone Methode aufgerufen. Weil diese den aktuellen Thread nicht blockiert, lauft die Anwendung weiter. Vom .Net wird registriert, wann die Methode fertig ist. Heute nicht mehr „Ereignisorientiert“ sondern „Aufgabenorientiert“ mit „async“ und „await“.

9. April 2019

Wichtig zur Hausübung:

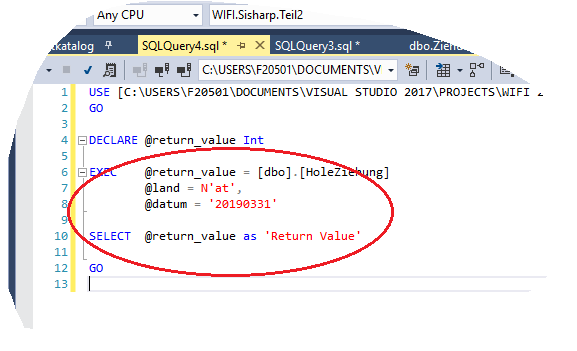


Beim Erstellen der Tabelle „Ziehungen“ darauf achten, die Tabelle „Länder“ und „Ziehungen“ über „ISO2“ und „Land“ in einer 1:n Beziehung stehen.

Solche Beziehungen können im SQL Server Management Studio grafisch erstellt werden. Hier im Visual Studio über das REFERENCES – Schlüsselwort.

Das Land in Ziehungen ist jetzt ein „Fremdschlüsselfeld“, von dem auch die Datenbank weiß.

Zum Testen einer gespeicherten Prozedur direkt …

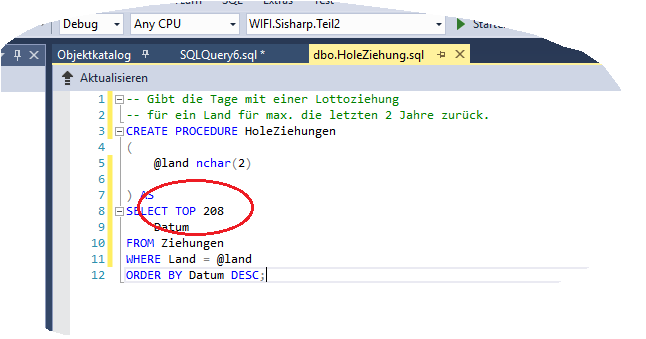


… auf der Datenbank, die EXECUTE Anweisung verwenden.

Hinweis:

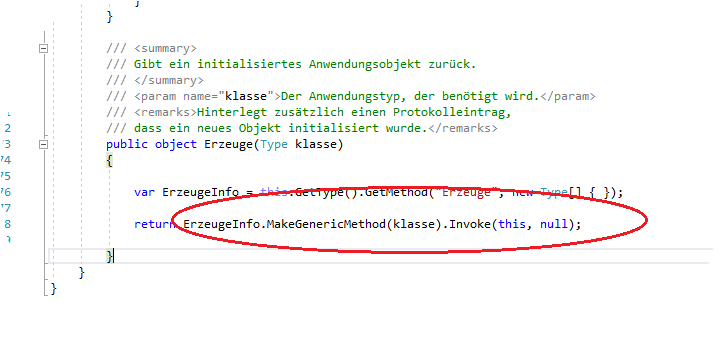
* In T-SQL können viele Anweisung mit  
  den ersten vier Zeichen abgekürzt eingegeben werden
* Datumsangaben am besten unter Hochkomma im  
  ANSI Format (JJJJMMTT) eingegeben

Hinweis zum SELECT

* Mit SELECT DISTINCT kann erreicht werden,  
  dass nur eindeutige Datensätze zurückgegeben werden
* MIT SELECT TOP n …  
    
  … kann die Anzahl der zurückgegebenen Datensätze eingeschränkt werden.

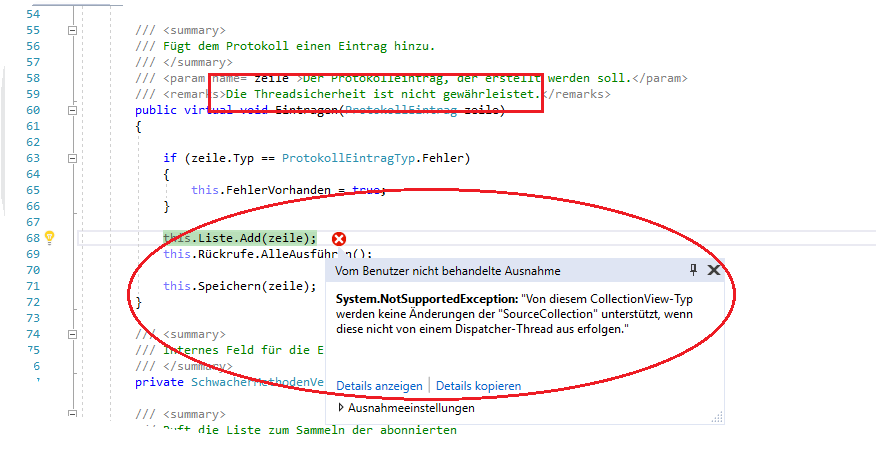
Manchmal praktisch zu wissen:

* Aus dem Code dynamisch eine generische Methode  
  aufrufen, wobei der Typ erst zur Laufzeit feststeht…



* „MakeGenericMethode“ in einer MethodInfo aus der Reflection

Neues Problem:



* So toll Multithreading ist, so lästig kann’s sein
* Der Protokollmanager stürzt ab und meint,  
  dass das Hinzufügen einer Zeile nicht zulässig ist,  
  weil wir aus einem anderen Thread kommen

Lösung:

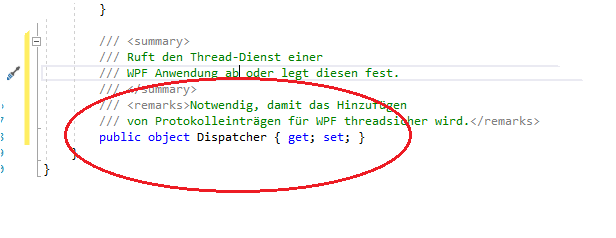
* In Windows Forms: Die Eigenschaft „InvokeRequired“ prüfen,  
  falls ja, in den aktuellen Thread durch Aufruf der Windows.Forms.Form.Invoke – Methode und einem Callback umschalten
* In WPF:  
    
  Den zentralen Thread-Verteiler nutzen,  
  den Dispatcher

Problem:

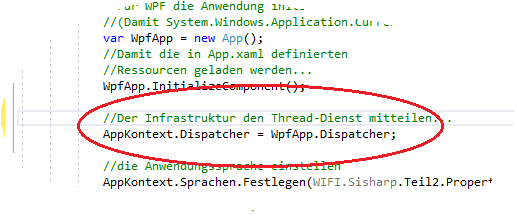
* Der ProtokollManager ist aber in einer Assembly, die  
  WPF nicht verweist, also hat keinen Dispatcher

Lösung:

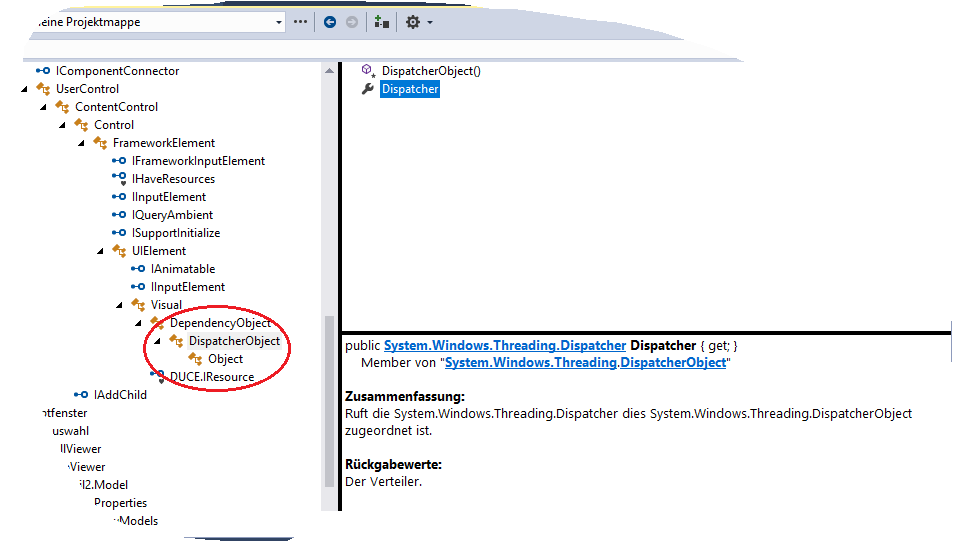
* Der Infrastruktur einen Dispatcher beibringen



Den Dispatcher möglichst bald einstellen, wenn die WPF Anwendung läuft, d.h. wir sind im Main…

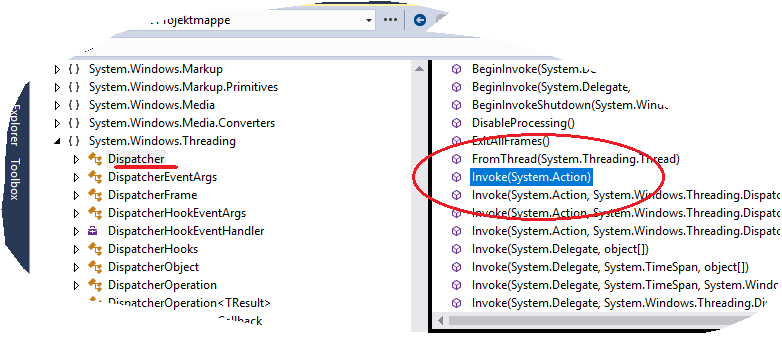


ALLES in WPF ist eine Erweiterung vom …



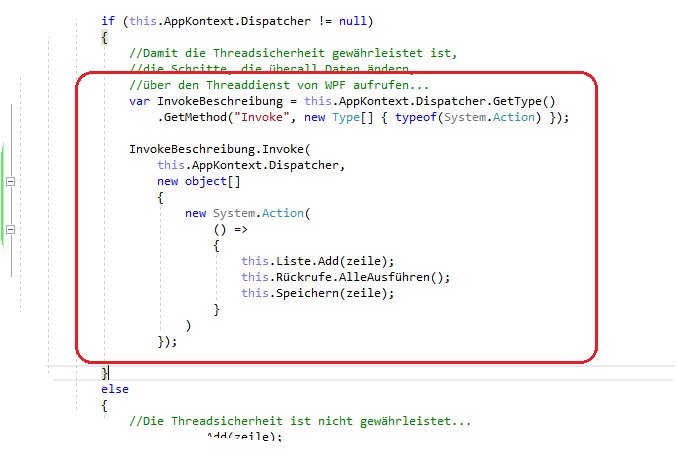
… DispatcherObject aus der Assembly WindowsBase.

Damit wir threadsicher werden, …



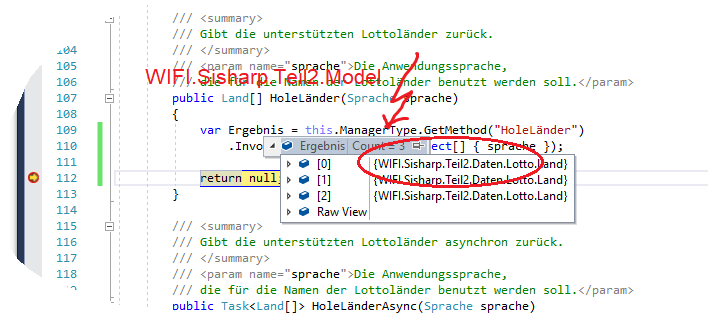
… muss der unsichere Code über die Invoke-Methode des Dispatchers ausgeführt werden.

Im Folgenden …

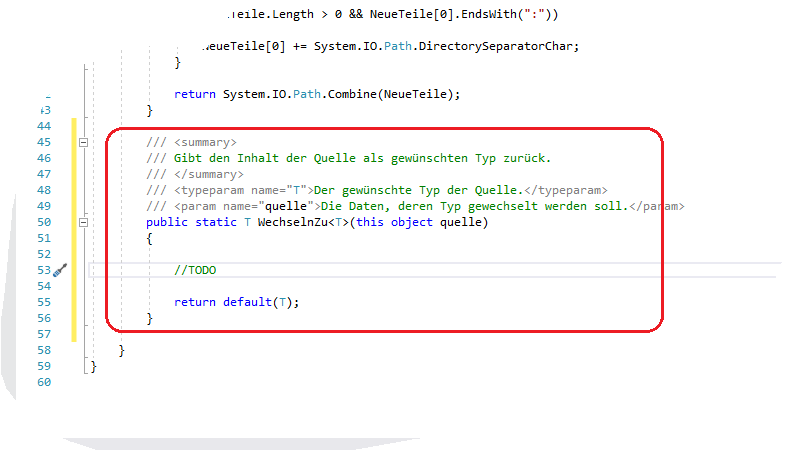


… ist gezeigt, wie die einfachste Invoke-Methode vom Dispatcher für eine Aufgabe, die threadsicher sein soll, als System.Action aufgerufen wird.

Neues Problem:



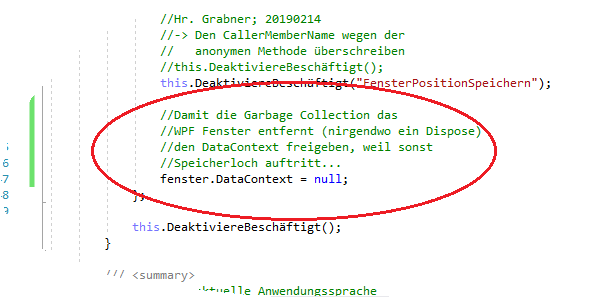
* Das Interface vom Webdienst schreibt …  
  WIFI.Sisharp.Teil2.Model.Land  
  … vor.
* Durch die Offline – Benutzung der  
  Assembly bekommen wir aber …  
  WIFI.Sisharp.Teil2.Daten.Lotto.Land
* Diese beiden sind nicht kompatibel,  
  obwohl dieselben Daten enthalten sein
* Wir müssen die Daten kopieren,  
  zu Model.Land wechseln
* Weil das ein häufiges Problem ist,  
  eine Erweiterungsmethode



* Dieses eigene WechselnZu ist komfortabler als  
  System.Convert.ChangeType

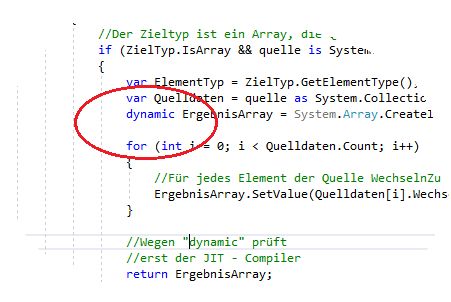
11. April 2019

Damit unser Speicherloch gestopft wird, …



… beim Schließend des WPF Fensters den DataContext freigeben.

Damit die Typprüfung von C#, vor allem in der generischen Programmierung und dem Arbeiten mit LINQ, entschärft werden kann…



* „dynamic“

25.4.2019

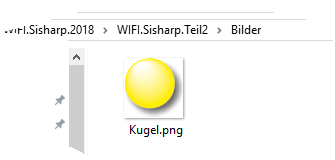
Mit Hilfe vom System.Configuration.ConfigurationManager …



… kann die Anwendungskonfiguration ausgelesen werden.

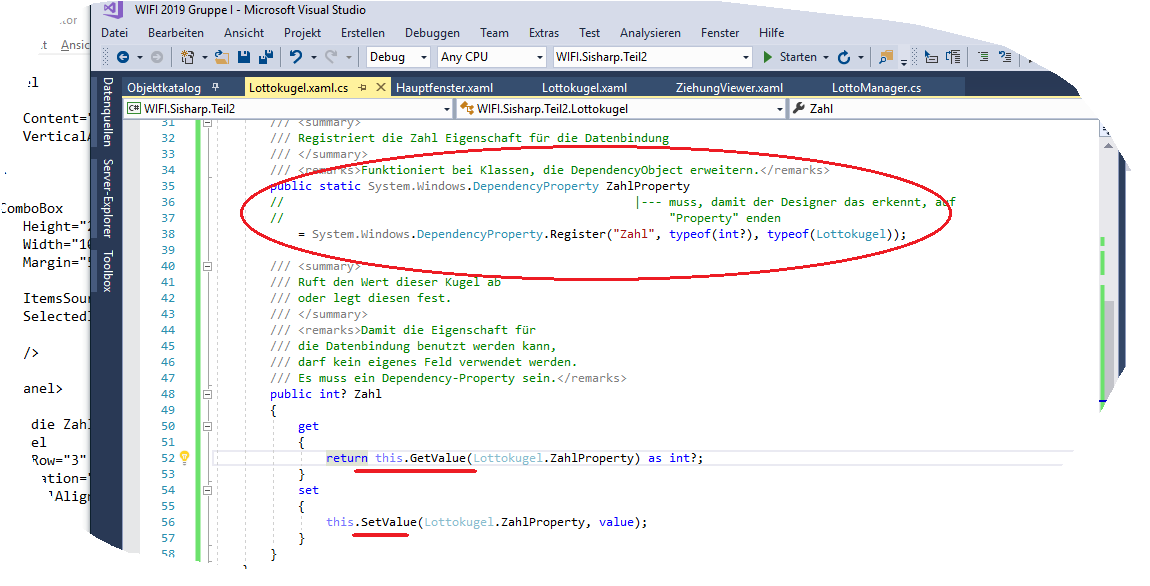
Verweis auf System.Configuration notwendig.

Neuer Bereich:

* Steuerelement für eine Lottokugel,  
    
  wo ein Bild im Hintergrund ist  
  und ein Wert, die Lottozahl, gebunden   
  werden kann
* Wie wird in WPF eine Eigenschaft für ein Steuerelement  
  implementiert, das für die Datenbindung benutzt werden kann?

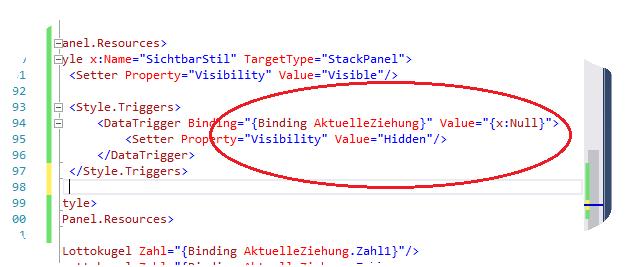
Lösung:

* DependencyProperty  
  (im Code Behind vom Steuerelement)



30.4.2019

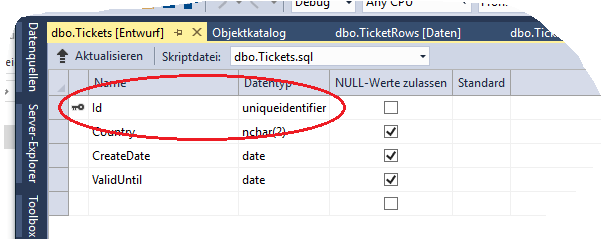
* Damit die Lottokugeln „unsichtbar“ werden, wenn  
  das Datum geändert wurde …



… in WPF mit DataTrigger für das StackPanel mit den Kugeln arbeiten.

* Damit auf „null“ in WPF geprüft werden kann, „{x:Null}“ benutzen.

Zusätzliche Hinweise zu Datenbanktabellen:



* Wir wissen, jede Tabelle benötigt einen Schlüssel

Welche Werte werden als Schlüssel herangezogen?

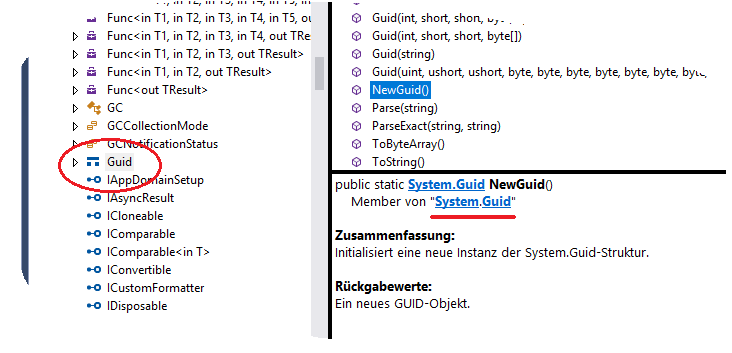
* Alt:  
    
  „Sequentielle Schlüssel“, d.h. „fortlaufende Nummern“

Nachteile:

* Bevor ein Datensatz gespeichert werden kann, muss die  
  „nächste Nummer“ ermittelt werden, d.h. es ist ein Lesezugriff notwendig noch ohne, dass etwas passiert
* Keine „dezentrale“ Erfassung möglich.  
  „Zusammenspielen“ von Daten ist lästig
* Ein einmal festgelegter Schlüsselwert soll sich nicht mehr ändern

Deshalb:

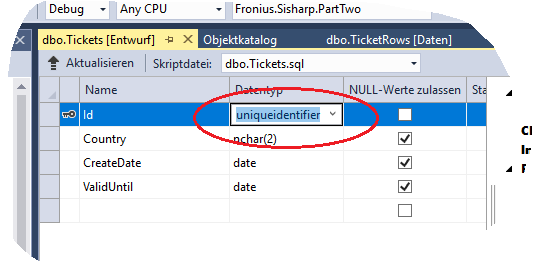
* NEU, nur Vorteile (für die Programmierung)
* Global Unique Identifier
  + Eine 128 Bit Zahl, hexadezimal dargestellt
  + Ein Teil davon ist Regionsbezogen
  + Ein Teil davon ist Zeitbezogen
  + Ein Teil davon ist zufällig
  + Laut Microsoft weltweit eindeutig beim Erstellen
* In .Net

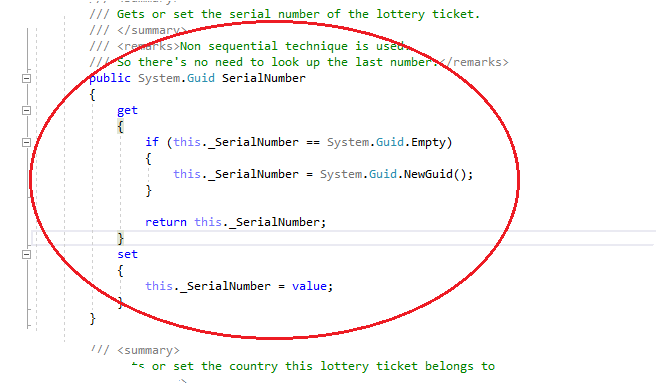


* System.Guid

Es gibt nur einen Nachteil (für die Datenbank-Administration)

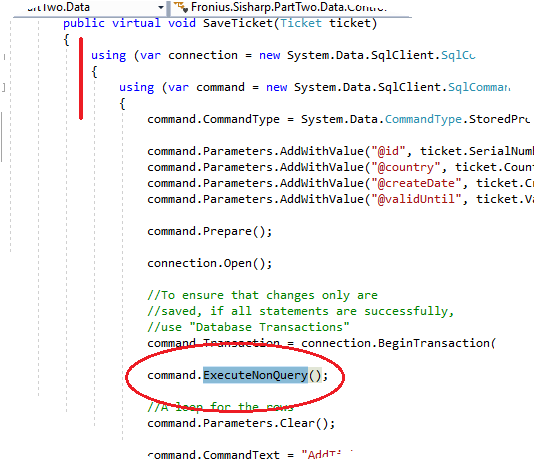
* Eine Datenbank speichert die Datensätze sortiert nach dem Schlüsselwert (Cluster)
* Andere Technik: Tabellen „ohne Schlüssel“ (darf’s im Grunde nicht geben, werden als „Heap“ (Haufen) bezeichnet) speichern die  
  Daten in der Reihenfolge, wie sie anfallen  
    
  Diese Tabellen nur für Protokolle benutzen, wo Daten „nie“ abgefragt werden
* Zum Nachteil:  
  Weil die Guid einen Zufallsanteil hat, kommen die Schlüssel „unsortiert“  
  und die Datenbank „fragmentiert“, d.h. die Datenbank-Administration muss „defragmentieren“



* Im C# Code kommt man …  
  

… einfach im Getter zu einem neuen eindeutigen Schlüssel.

Im C# Code für SQL INSERT, UPDATE und DELETE …

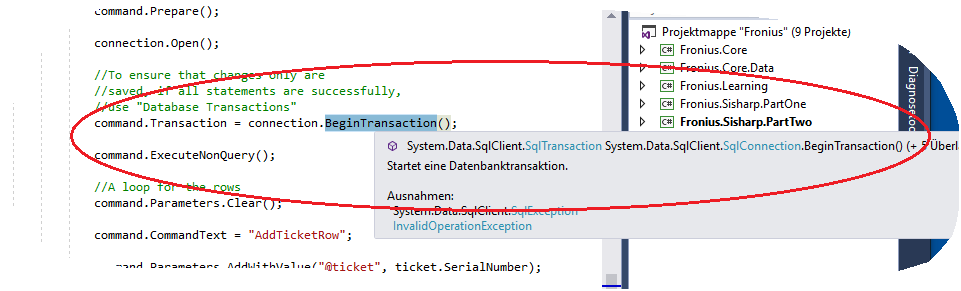


… sind nur die Verbindung und der Befehl notwendig. Kein Leser.

Zum Ausführen von SQL INSERT, UPDATE und DELETE nur Befehl.ExecuteNonQuery

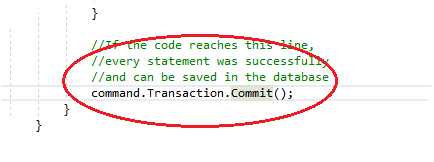
WICHTIG:

* Wenn eine Methode in der Datenbank mehrere (unterschiedliche) Schreibzugriffe hat, muss oft sichergestellt werden, dass nur alle Zugriffe als Gesamtheit „richtig“ sind. Falls ein Teil missling, ist alles ungültig
* Dazu müssen die Datenbank-Zugriffe „gecachet“ werden
* Das nennt man „Transaktionen“



Die Abbildung zeigt, wie über die Verbindung eine neue Transaktion gestartet und dem Befehl zugewiesen wird.

* Aufpassen:  
    
  Sobald eine Transaktion gestartet ist, ist der benutzte Teil in der Datenbank für andere Benutzer „gesperrt“.  
  Umsichtig arbeiten, Transaktionen sobald als möglich schließen!



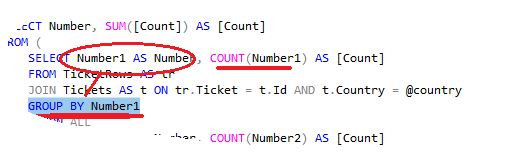
* Mit der Methode „Commit()“ wird eine Transaktion als erfolgreich gekennzeichnet und die Datenbank schreibt die Daten

Noch etwas zum SQL:

* SQL kann Berechnungen über mehrere Datensätze durchführen
* Dazu gibt’s die Aggregatfunktionen
* Z. B. SUM, COUNT, AVG, MIN, MAX, …

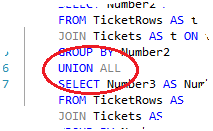
<https://docs.microsoft.com/de-de/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-2017>

Beispiel:



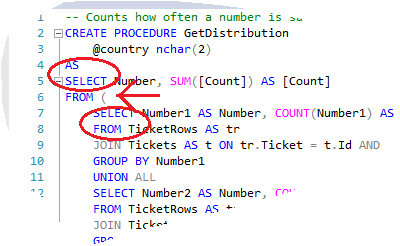
* Fasse alle Werte im Tabellenfeld „Number1“ zusammen (GROUP BY) und zähle, wie oft ein Wert vorkommt (COUNT)

Außerdem ist es möglich, das Ergebnis von mehreren SELECT Anweisungen zu einem Ergebnis …

b

… zusammenzufassen. Das SQL Schlüsselwort dazu lautet „UNION ALL“

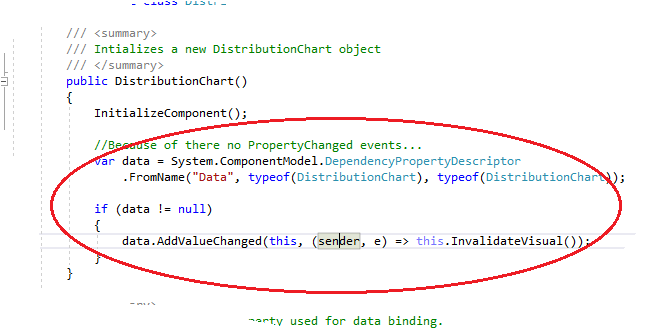
Daneben können innerhalb einer SQL Anweisung „Unterabfragen“ benutzt werden.



Das ist fast überall erlaubt. Von der Syntax her müssen Unterabfragen in runde Klammern!

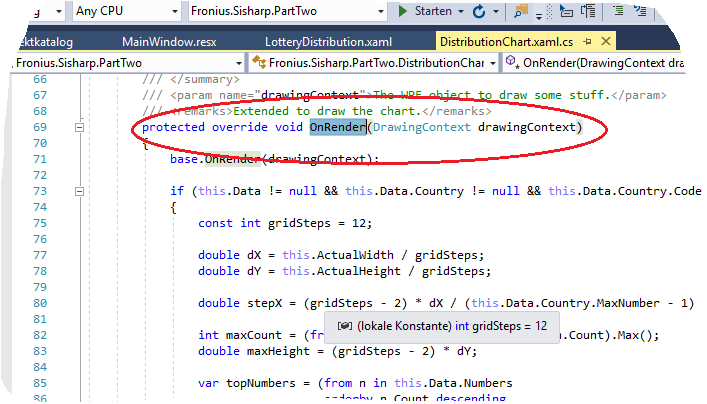
Zurück zu den DependencyProperties (z. B. LottoKugel.Zahl)

* Lästig  
    
  Der Setter löst kein „OnPropertyChanged“ aus



* Mit System.ComponentModel.DependencyPropertyDescriptor bekommt man aber eine AddValueChanged-Methode der man eine anonyme Methode mitgeben kann, die sagt, was passieren soll, wenn sich der Wert ändert. Mit „InvalidateVisual“ wird ein WPF Steuerelement neu gezeichnet (in der Forms die Refresh() Methode)

Zum Schluss:



Dem „OnPaint(System.Drawing.Graphics)“ entspricht in WPF „OnRender(DrawingContext)“