***Coding Like the Wind | Drehbuch***

***Intro***

* Wesentliches Ziel der Session: Produktivitätstips für den Alltag zeigen (nicht ausschließlich VS 2013 präsentieren!)

***Navigation (Samples based on tc All.sln)***

Call Hierarchy vs. Find all references:

**Screen setup:**

* Unten: Call Hierarchie
* Hauptbereich: Edit

**Szenario: Auswirkungen einer Methoden-Änderung**

Ich möchte etwas an der Implementierung von DataContext.InitializeEnvironment ändern.  
Auf welche public Methoden hätte das indirekt Auswirkungen?

**Demo:**

1. Gehe zu InitializeEnvironment  
   **Bookmark Code Navigation - Call Hierarchy – InitializeEnvironment**
2. Collapse to definition (**Ctrl+M, O**) für besseren Überblick
3. Funktion ist private -> welche public-Methoden rufen sie auf?
4. Für diesen Fall suboptimale Funktion "Find all references (**Ctrl+K, R**)" zeigen  
   (Nachteil: kein Hinweis auf Funktionsname)
5. "Call Hierarchy (**Ctrl+K, T**)" aktivieren
6. Mit "Calls to..." alle Aufrufer suchen
7. Über mehrere Stufen nachvollziehen, bis public Aufrufer gefunden werden
8. "Code Definition Window" einblenden (**Ctrl+W, D**)
9. Wir wollen floating: Ganze untere Leiste abdocken -> Fehler -> Ctrl+Doppelklick auf Fenster -> ...
10. Prozess wiederholen und Vorteil des Code Def Windows erklären
11. Code Def Window mit Referenzen zeigen

**Szenario: Ableitungen finden**

Wer implementiert aller die abstrakte Basisklasse PersistedProperty?

**Demo:**

1. Gehe zu PersistedProperty
2. **Bookmark Code Navigation - Goto Definition - PersistedProperty**
3. Collapse to definition (**Ctrl+M, O**) für besseren Überblick
4. Zeigen, dass Call Hierarchy hier nicht funktioniert
5. Mit "Find all references (**Ctrl+K, R**)" alle Implementierungen finden
6. Neues Feature im Solution Explorer -> dafür müssen wir aber zunächst das aktuelle File im Solution Explorer finden!
7. Ctrl+Q, **@opt track** (ev. stillschweigend an dieser Stelle tippen): Track active item einschalten
8. Aufklappen bis auf Klassenebene, rechte Maustaste -> **Derived Types**

**Apropos Quick Launch: Haben Sie soeben gesehen, wie ich eine Option gefunden habe?**

* **@opt** track, @opt line num
* **@menu** explorer
* **@mru** (most recently used)

<http://msdn.microsoft.com/en-us/library/hh417697.aspx>

**Szenario: Suche im Sourcecode**

Suche die Klasse, die binäre Properties im tc Modell implementiert.

**Demo:**

1. Find in files (**Ctrl+Shift+F**) mit "Binary"
   1. F8 zum Springen zum nächsten Element zeigen
2. **Find in files mit RegEx** "bin.\*prop.\*"
   1. Expression Builder verwenden!
3. Navigate to (**Ctrl+,**) mit "**bin prop**"
4. Navigate to (Ctrl+,) mit "**binary Property**"  
   (Nur files weil case insensitiv)
5. Navigate to (Ctrl+,) mit "**BinaryProperty**"
6. Navigate to (Ctrl+,) mit "**BP**" (Pascal Casing)
7. **TQWHR** -> TestQueriesWithHierarchicalRelation
   1. zeigen, dass Akronyme auch bei IntelliSense funktioniert (new TQWHR..)
8. **DSDS** ->... ;-)

**Szenario: Definition finden**

Implementierung von DataContext.SelectSingle(string) verstehen.

**Demo:**

1. Suche **SelectSingle** mit **Navigate to**.
2. Mit **F12** den Callstack nach unten steigen.
3. **F12** auch auf eine referenzierte **.NET Funktion** zeigen (zB **Enumerable.Any** -> zeigt Extension-Klasse Enumerable an)
4. Mit **Alt+F12 Peek into Definition** zeigen
5. Mit "Navigate Backward" (**Ctrl+Minus**) wieder zurück navigieren
   1. Navigationsbuttons in der Toolbar zeigen
6. Anhand von DbClient.Select kann "**Reference Highlighting**" gezeigt werden
7. Navigation mit **Ctrl+Shift+Up/Down** zeigen
8. Anhand von DbClient.Select kann auch **Outlining** gezeigt werden
9. Collapse to definitions (**Ctrl+M, O**)
10. Toggle outlining (**Ctrl+M, M**)
11. Ad hoc block (**Ctrl+M, H**)
12. #region

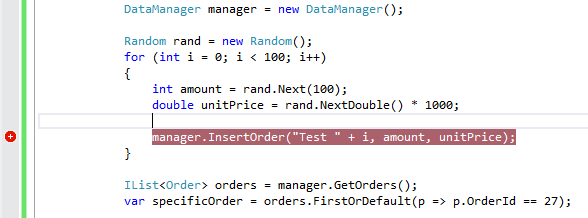
***Allgemeine Visual Studio 2012 Features***

1. Neues Visual Studio starten
   1. Startpage zeigen
2. Neues **Testprojekt** **„CodingLikeTheWind.Test“** erstellen
   1. New Project-Dialog
3. Neue **ClassLibrary „CodingLikeTheWind.Data“** erstellen
4. Referenz auf ClassLibrary im Testprojekt hinzufügen
   1. **AddReference-Dialog**
5. Code tippen (oder einfügen)
   1. Automatische Formatierung (**Ctrl+E, F**)
   2. Block Selection (**Alt+Click&Drag**, zB Datentyp in mehreren Zeilen gleichzeitig austauschen)
   3. Zoom (**Ctrl+Mausrad**)
6. ToDo / Bookmarks / Tasklist Shortcuts
   1. // TODO -> erscheint in Tasklist (**Ctrl+W, T**) unter „Comments“
   2. Bookmark hinzufügen (**Ctrl+B, T**) -> erscheint unter „Bookmarks“ (**Ctrl+W, B**)
   3. Tasklist Shortcut hinzufügen (**Ctrl+E, T**) -> erscheint in Tasklist (**Ctrl+W,T**) unter „Shortcuts“

***Code generation, Intellisense, Snippets***

1. Test-Driven: Neue Testmethode erstellen (Snippet: testm)
2. Klasse anlegen
   1. DataManager manager = new DataManager()
   2. **Strg+.** mit Generate New Type
   3. Klasse in ClassLibrary erstellen lassen
3. Methode von DataManager wie unter (2) erstellen
4. DataManager-Klasse aus vorgefertigtem Sourcecode befüllen -> Referenzen auf SQL CE nicht vorhanden
   1. **AddReference**-Dialog
   2. **Strg+.**
5. Test Explorer zeigen
   1. Test *ConvertXamlToObjectTree* im time cockpit Projekt laufen lassen
6. Intellisense-Features zeigen
   1. Modes (wechseln mit **Ctrl+Alt+Space**)
   2. Member anzeigen (**Ctrl+J**)
   3. Parameter-Informationen anzeigen (**Ctrl+Shift-Space** – Achtung: Cursor muss innerhalb der Funktions-Klammern stehen)
   4. Quick-Info anzeigen (**Ctrl+K,I** – Cursor muss auf Methodenname stehen)
7. Snippets
   1. Standard-Snippets (if, try/catch, for)
   2. Snippet UI (**Ctrl+K, X**)
   3. Snippet Manager (**Ctrl+K, Ctrl+B**)
   4. Surround with (**Ctrl+K, S**)
8. Item templates
   1. Create empty class library
   2. Add a class implementing *INotifyPropertyChanged*
   3. Add some references (e.g. *WindowsBase, PresentationFramework, PresentationCore*)
   4. Use *File/Export Template…* to generate ItemTemplate ZIP File
      1. Talk about deployment folders
      2. Use item template
   5. Use *File/Export Template…* to generate ProjectTemplate ZIP File
   6. Create new project with *Extensibility/Item Template*
      1. Talk about project setup and placeholders
      2. Build project and show ZIP file
   7. Create new VSIX project and add item template from previous step
      1. Build project and sho VSIX file
      2. Install VSIX file
      3. Use item template
      4. Show template in *Tools/Extensions and Upgrades*
      5. Uninstall VSIX file
9. T4
   1. Open existing solution *CodeGenerationT4*
      1. Talk about design time code generation
      2. Talk about runtime code generation

***Debugging***

1. Conditional Breakpoints
   1. zB Breakpoint bei zufälligem Einfügen nur dann feuern, wenn große Mengen bestellt wurden (amount > 70)  
      
2. ~~Intellitrace: Beispiel ist mit SQL CE-Datenbank~~
3. Immediate Window (**Ctrl+D,I**)
   1. Funktioniert sowohl zur Debugzeit (Kontext des aktuellen Breakpoints) oder zur Designzeit (hier müssen aber zB Namespaces angegeben werden)
4. Data Tips
5. Object ID’s
   1. zB bei den letzten zwei Zeilen (IList<Order> ….) vom oberen Screenshot zeigen
6. DLR-Support (kurzes Beispiel coden)