

.NET Programmierung für industriennahe Anwendungen

Wintersemester 2022

(c) 2022

Roland J. Graf / Gerald Lochner

Labor 04: C# Extension Methods, Unit Tests und Windows Forms GUIs

1.1. Themen

- Vertiefung der C#/.NET Programmierung
- Implementieren von automatischen Tests in Visual Studio ⇒ Unit Tests
- Optional: Erstellen einer einfachen Windows GUI mit WinForm

1.2. Projektaufgaben

Im nächsten Schritt gehen wir wieder einen Schritt tiefer in die allgemeine Softwareentwicklung unter .NET

1.2.1. Erstellung einer Assembly mit Extension Methods

- Erstellen Sie innerhalb Ihrer Solution eine neue Assembly, die dann mehrere Extension Methods aufnehmen soll. Wählen Sie hierzu ein beliebige Klasse aus, die Sie mit einigen Methoden erweitern möchten, ohne jedoch den Quelltext der zugrundeliegenden Klasse abzuändern. Mit Extension Methods werden vor allem Klassen erweitert, deren Quelltext beispielsweise nicht vorliegt oder deren Ableitung nicht erlaubt ist, da die Klassen mit dem Modifizierer *sealed* markiert wurden. ZB die Klasse 'String' wäre so eine Klasse; Sie können aber auch jede andere Klasse Ihrer Solution mit neuen Methoden erweitern). Legen Sie gemeinsam im Team den Namespace und die Klasse fest.
- Erstellen Sie parallel ein eigenes TestProjekt (MSTest Projekt) für die Unit Tests der Methoden Ihres neuen Assembly.
- Legen Sie nun fest, welche neue Extension Methode erstellt werden soll (Name, Parameter, Rückgabewert,...). Nun implementiert ein Teammitglied die Methode, während das andere Teammitglied zeitgleich die Unit Tests programmiert. Der eine Entwickler hat also die Aufgabe bzw. das Ziel, die Methode so stabil und unangreifbar wie möglich zu implementieren, während der andere das Ziel verfolgt, die Tests so kritisch wie möglich durchzuführen. Man verwendet hierbei also ganz bewusst auch unzuverlässige Parameter um zu überprüfen, wie die zu testende Methode damit umgeht. I.d.R. sollten passende Exceptions geworfen werden. Irgendwann ist die Methode fertig und der Test. Nun wird getestet. Eine Methode gilt dann als fertig implementiert,

wenn sie bei allen(!) Tests mit *PASSED* bestätigt wurde.

1.2.2. Erstellung von Unit Tests für Ihre Assemblies

- Erstellen Sie nun für jede Assembly Ihrer Solution ein eigenes TestProjekt (MSTest).
- Erstellen Sie für alle public und protected Methoden eigene Unit Tests. Teilen Sie sich auch hier wieder die Arbeit, wenn Sie mehrere Assemblies in Ihrer Solution haben. Optimalerweise testet immer jeweils jene Kolleg:in jene Klassen und Methoden, welche die Klassen/Methode *nicht* implmentiert hat. Die Klasse und Methode sollte ja ausreichend dokumentiert sein, um nun auch Tests erstellen zu können. Wenn nicht, dann wird das schnell auffallen.

Starten Sie Ihre Unit Tests über den Test Eplorer. Am Ende sollte bei einem vollen Durchlauf jede Methode getestet werden und alle Methoden sollten ein grünes *PASSED* Symbol anzeigen.

- Weiterführender Lesestoff
 - [Walkthrough: Create and run unit tests for managed code](#)
 - Foliensatz

Je nach Komplexität Ihrer Assemblies, werden Sie mit der obigen Aufgabe mehr oder weniger schnell fertig sein. Für jene Teams, die nun noch Zeit, Muse und Lust haben, gibts eine optionale Aufgabe:

1.2.3. Optional: Erstellung eines Modus Clients mit einer WinForm GUI

- Erstellen Sie in Ihrer Solution ein neues WinForm Projekt
- Designen Sie eine einfache Windows Forms GUI mit der Basisfunktionalität Ihres Modbus Clients
 - Verbindung mit dem Server (IP Adresse, Port, Connect, Disconnect,...)
 - Anzeige ausgewählter Modbus Register (Festlegung und Anzeige der anzuzeigenden Register, Pollen und Refresh bei Änderung der Registerwerte)
 - Setzen ausgewählter Register (Zuweisung neuer Werte)
 - Buttons für die Hauptfunktionen (z.B: Start und Stop der Maschine,...)

NOTE

Legen Sie den Fokus auf ein schönes, ergonomisches GUI. Gerade bei einer GUI gilt: Manchmal ist weniger mehr.

1.2.4. Allgemeine Anforderungen:

- Es gelten für die Implementierung die gleichen Regeln wie für Aufgabe 01 bis 03.

1.3. Projektabgaben

- Abgabebedingungen und Formate: siehe Moodle-Kurs