

Tutorial – Visual Studio 2019

Lernziel:

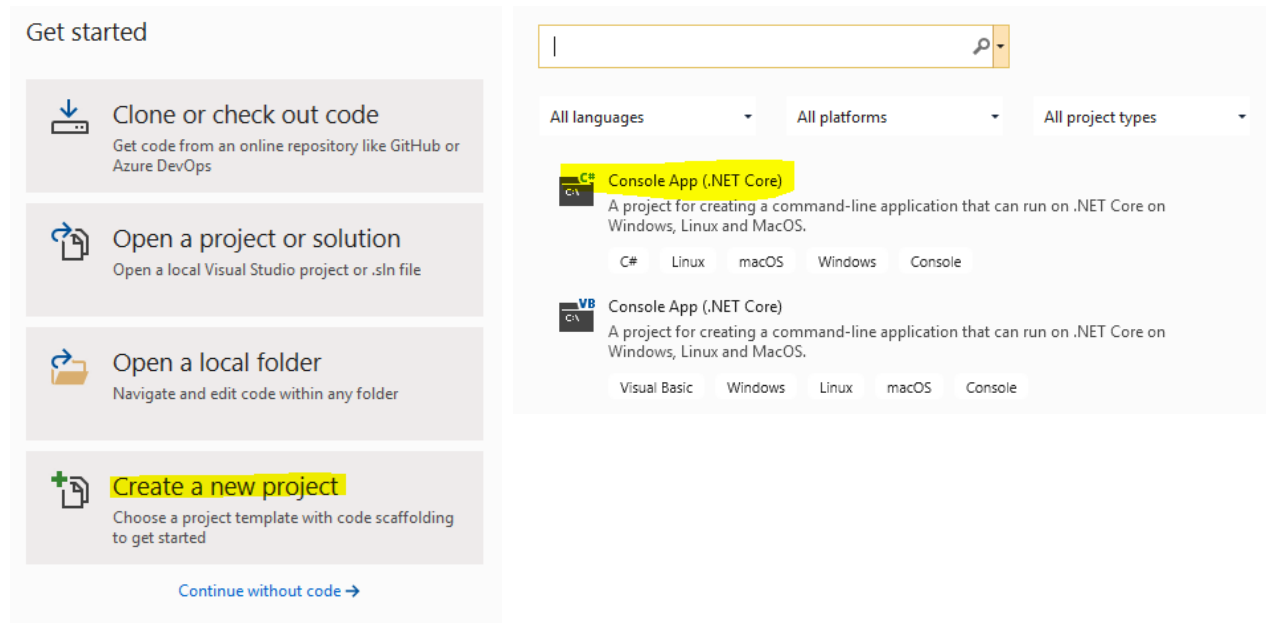
- Sie kennen einige wichtige Visual Studio Projektstrukturierungskonzepte und können mit VS2019 ein einfaches C#-Programm kompilieren und ausführen.

Voraussetzung: Sie haben mindestens VS2019 Community installiert

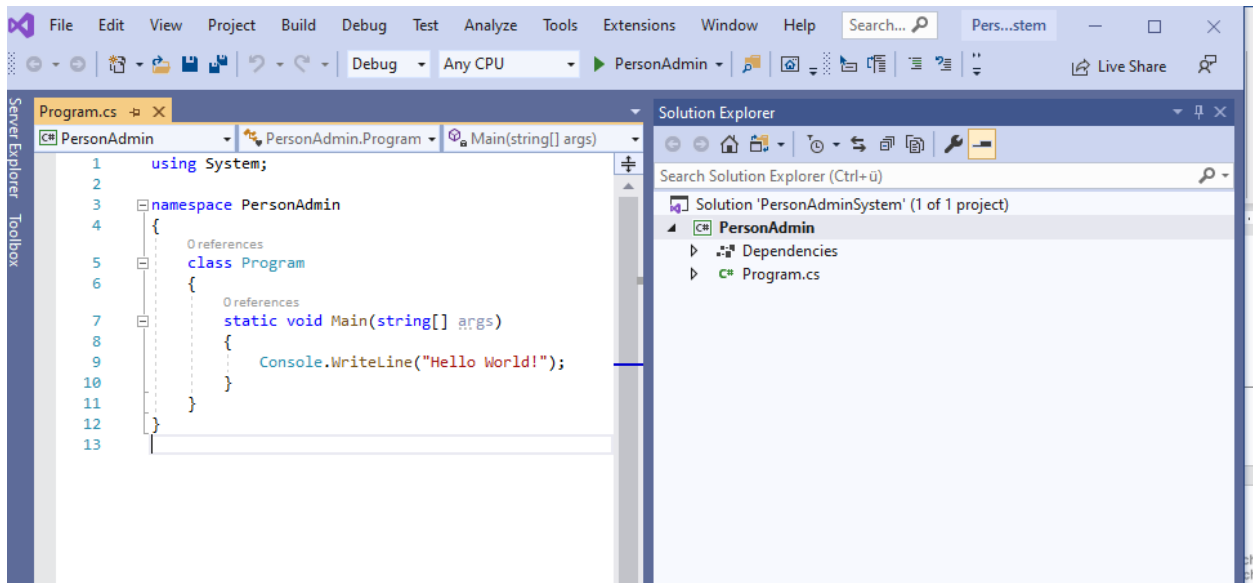
Das folgende Tutorial dient dazu, Sie mit einigen wichtigen Konzepten von Visual Studio vertraut zu machen.

I. Anlegen eines Projektes

- 1) Starten Sie VS2019
- 2) Legen Sie ein neues „Konsolen“ Projekt. Rufen Sie "Create a new project" auf und wählen Sie den Project Type Console App (.NET Core)“ für C# aus. Wählen Sie next und geben Sie als Projektnamen „PersonAdmin“ ein. Geben Sie als Solution Name „PersonAdminSystem“ ein.



- 3) Schliessen Sie das Anlegen des neuen Projektes mit Create ab.
- 4) Sie sehen nun im Solution Explorer (falls dieser nicht sichtbar ist, mittels „View/Solution Explorer“ oder Ctrl+Alt+L öffnen) die neue Solution und das Projekt.



Hinweis: Sie haben nun zwei wichtige Konzepte zur Organisation von Softwareprojekten kennengelernt. Ein „Project“ ist eine Sammlung von Files, die zu einer ausführbaren Datei (Assembly als dll, exe) kompiliert wird. Eine „Solution“ ist eine Sammlung von mehreren Projekten, die zusammen eine Software Applikation ausmachen (typischerweise eine exe-Datei und mehrere DLLs).

II. Kompilieren und Ausführen

VS2019 legt automatisch eine Klasse Program mit einer Main-Methode an.

Hinweis: Beachten Sie, dass Methodennamen in C# mit einem Grossbuchstaben beginnen.

- 5) Sie können nun Ihr Programm kompilieren, z.B. mittels „Build/Rebuild“ oder rechte Maus auf die Solution und «Build Solution», oder mit Ctrl+Shift+B.

Hinweis: Im Gegensatz zu Eclipse, kompiliert VisualStudio das Programm nicht automatisch im Hintergrund

- 6) Mittels „Debug/Start Debugging“ (oder F5) können Sie das Programm ausführen. Sie werden nur kurz ein Konsolen-Fenster aufblitzen sehen, da das Programm noch nichts macht.

III. Der erste Code

- 7) Benennen Sie zunächst die Klasse Program in PersonAdminApplication um. Führen Sie die Umbenennung durch die Refactoring Option "Rename" durch. Diese benennt auch direkt das File der Klasse um. Kompilieren Sie danach das Programm wieder.
- 8) Ersetzen Sie Ihre Main()-Methode durch folgende Implementierung:

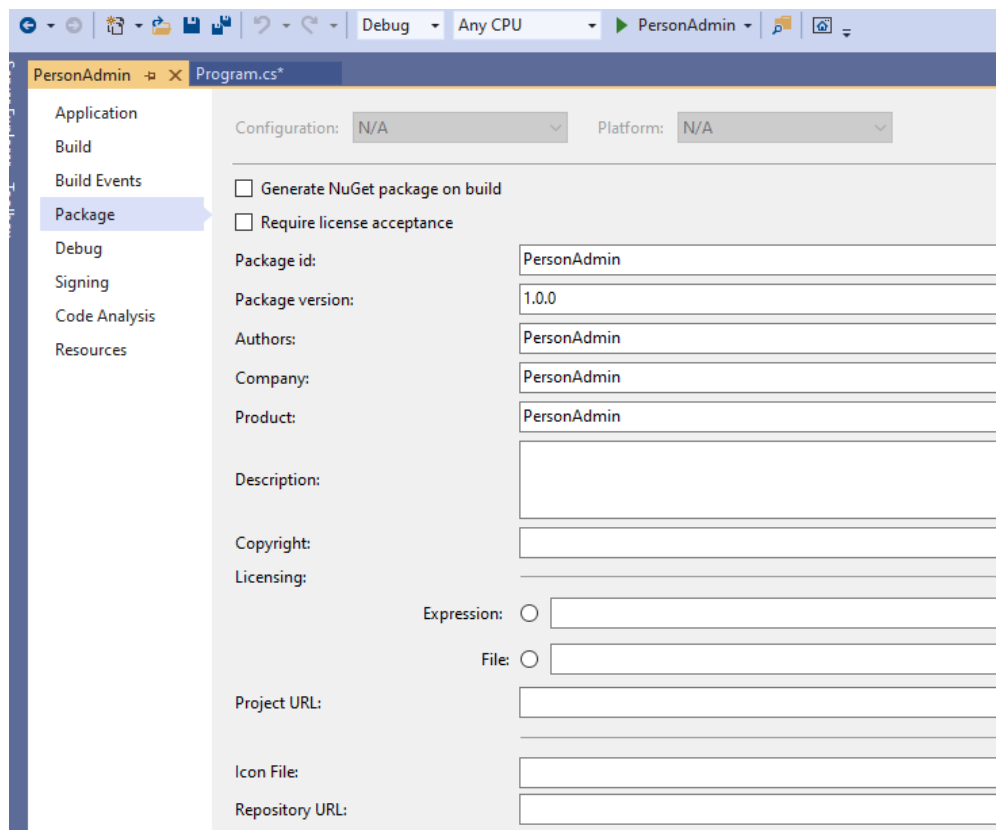
```
static void Main(string[] args)
{
    Console.WriteLine($"My first C# Program:
                        {Assembly.GetExecutingAssembly().GetName().Version}");
    Console.WriteLine("Press any key to quit");
    Console.ReadKey();
}
```

Sie müssen dazu zu Beginn des Files folgende Zeile hinzufügen.

```
using System.Reflection;
```

Hinweise:

- Wenn das `using`-Statement fehlt, wird `Assembly` mit roten Wellenlinien unterlegt. Mit `CTRL+."` oder rechte Maus/Glühbirne können Sie das Quickfix-Menü aufrufen. Über dieses Menü können Sie die `using`-Zeile von Visual Studio automatisch einfügen lassen.
- Das Programm liest die bei den Projekt-Properties unter "Package" voreingestellte Versionsnummer und schreibt sie auf die Konsole.



- Das "\$" beim Aufruf der `WriteLine`-Methode ist das Konzept der String-Interpolation in C# für das direkte Einbinden von Code in die String-Erzeugung.
- Entfernen Sie eventuell vorhandene, aber nicht benötigte Namespace (usings)

9) Führen Sie nun das Programm erneut aus. Was wird Ihnen angezeigt?

IV. Hinzufügen einer neuen Bibliothek und eigene Klassen

Sie sollen nun etwas Applikationslogik zu dem Programm dazu implementieren. Das wird jedoch nicht im Konsolenprogramm selbst, sondern in einer separaten Bibliothek vorgenommen.

- 10) Legen Sie ein neues Projekt `PersonAdminLib` vom Typ „Class Library (.NET Core)“ an. Dies machen Sie nun mittels Rechtsklick auf Ihre Solution im Solution Explorer und „Add/New Project“. Geben Sie unter «Name» den Namen «PersonAdminLib» ein. Das neue Projekt wird der bestehenden Solution hinzugefügt. Schliessen Sie mit OK ab. Sie sehen nun in Ihrer Solution zwei Projekte.

- 11) VS2019 legt Ihnen automatisch eine neue Klasse Class1 im File Class1.cs an. Löschen Sie dieses File.
- 12) Fügen Sie dem neuen Projekt mittels Rechtsklick auf das Projekt und „Add/Class...“ die Klasse Person hinzu (als Typ „Class“ auswählen). Geben Sie in dem Eingabefeld den entsprechenden Filenamen im Dialog eingeben. Machen Sie die Klasse public.
- 13) Fügen Sie der Klasse die public Properties string Firstname und string Surname hinzu

```
public string Firstname { get; }
public string Surname { get; }
```

- 14) Implementieren Sie den folgenden Konstruktor:

```
public Person(string firstname, string surname)
{
    Firstname = firstname;
    Surname = surname;
}
```

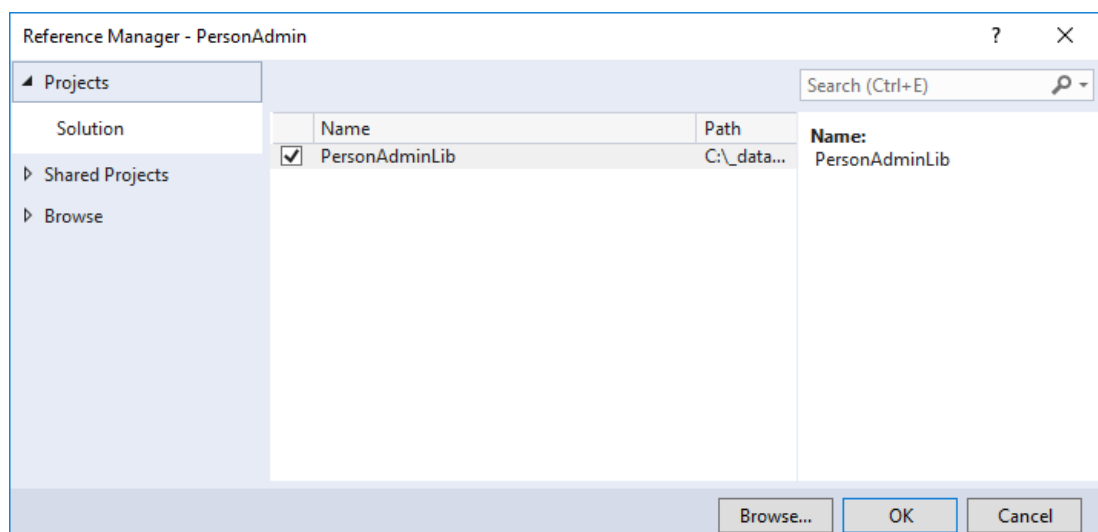
Kompilieren Sie die Bibliothek.

Hinweis: Die Klasse String existiert in C# in Gross- und Klein-Schreibweise. Es wird typischerweise die Klein-Schreibweise bevorzugt. Technisch gibt es aber keinen Unterschied zwischen den Varianten (siehe <http://stackoverflow.com/questions/7074/what-is-the-difference-between-string-and-string>).

V. Einbinden einer externen Bibliothek

Nun soll die neue Klasse in der Konsolen-Applikation verwendet werden. Dazu müssen Sie diese Klasse zunächst der Konsolenapplikation bekannt machen.

- 15) Fügen Sie nun die neue Bibliothek der Konsolen-Applikation als Referenz hinzu. Klicken Sie im Solution Explorer beim Projekt PersonAdmin mit Rechtsklick auf „Dependencies“ und dann „Add References...“ und wählen Sie dann unter „Projects/Solution“ die obige Bibliothek aus.



- 16) Legen Sie nun in Ihrer Main-Methode eine Instanz der Klasse Person an und geben Sie deren Inhalt mittels der Getter-Methoden auf der Konsole aus.

Hinweis: Sie können wieder mittels `using PersonAdminLib`; zunächst den Namespace importieren.

- 17) Führen Sie nun das Programm erneut aus. Was wird Ihnen angezeigt?

Gratulation – Sie haben Ihre erste .NET-Applikation geschrieben!

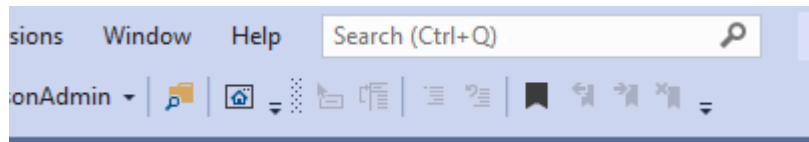
VI. Tipps und Tricks

Shortcut Keys

Sie arbeiten mit der Tastatur wesentlich effizienter als mit der Maus. Hier finden Sie die offizielle und vollständige Liste von Microsoft

<http://visualstudioshortcuts.com/2017/>

Search



Über Search können Sie eine Textsuche nach allen Funktionen im Visual Studio durchführen. Damit werden alle Stellen aufgelistet, wo ein Befehl mit dem eingetippten Wort vorkommt.