

Benutzung von Microsoft Visual Studio

Hier am Beispiel der deutschen Version 2017

Bei 2022 aber ähnlich.

In der Vorlesung wird es eine Vorführung anhang von Visual
Studio 2022 geben.

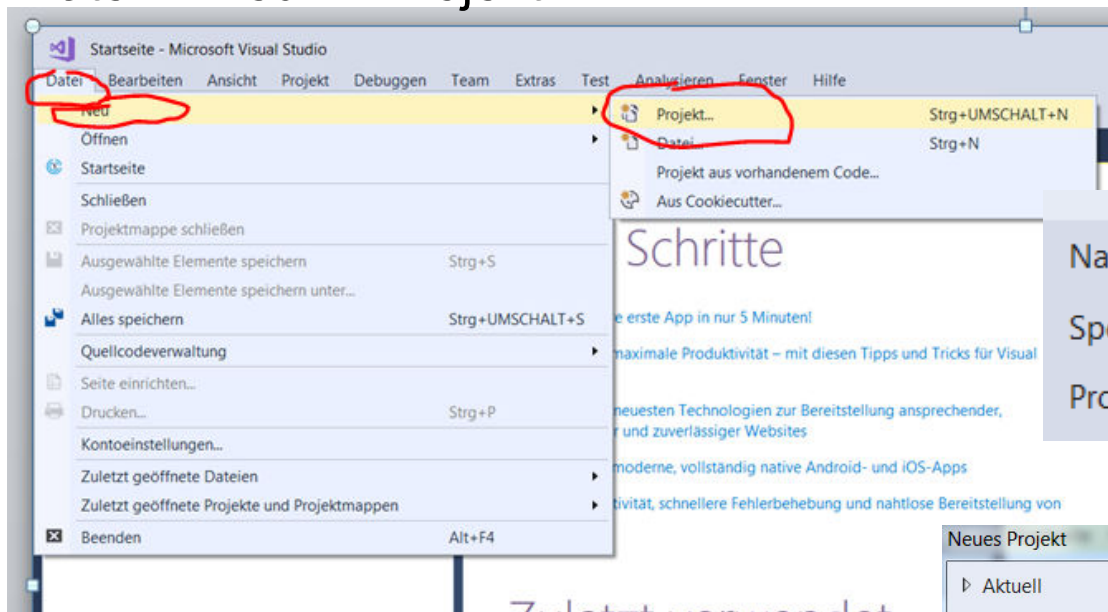
Siehe hierzu auch die eigene Datei „VisualStudio2022CommInstallation.pdf“

Erstellung einer Konsolenanwendung für Visual Studio (1)

Erstellung eines neuen Projektes (entspricht einer Makefile-Erstellung):

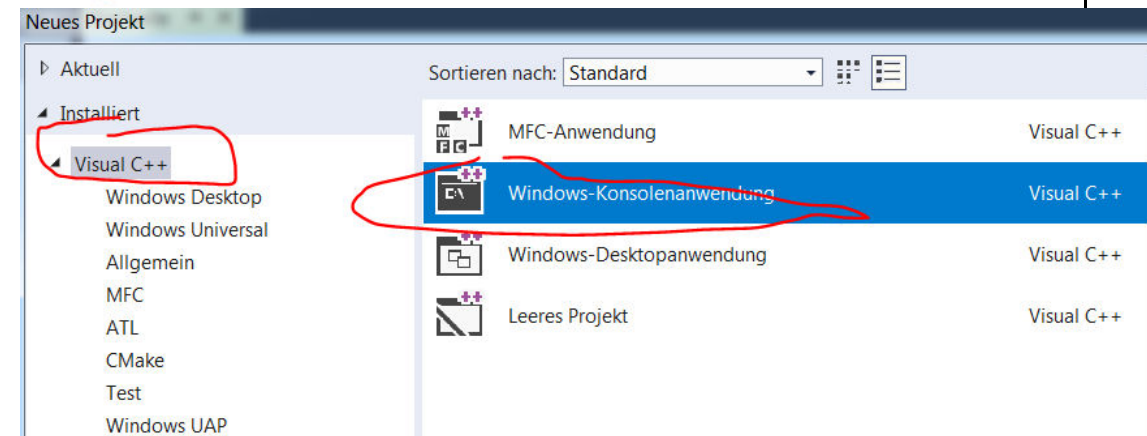
Datei → Neu → Projekt

Pfad durch Browsen bei Speicherort auswählen und **Name** eingeben.
OK anklicken.



Name:	CntKeyWords-VS2017
Speicherort	R:\VorlesungsUnterlagen\Folien\Vorles02\SW\
Projektmappenname:	CntKeyWords-VS2017

Bei Vorlagen „Visual C++ auswählen
Und dort „Win32-Konsolenanwendung“
auswählen, aber
nicht Win32-Projekt und auch nicht
Leeres Projekt!



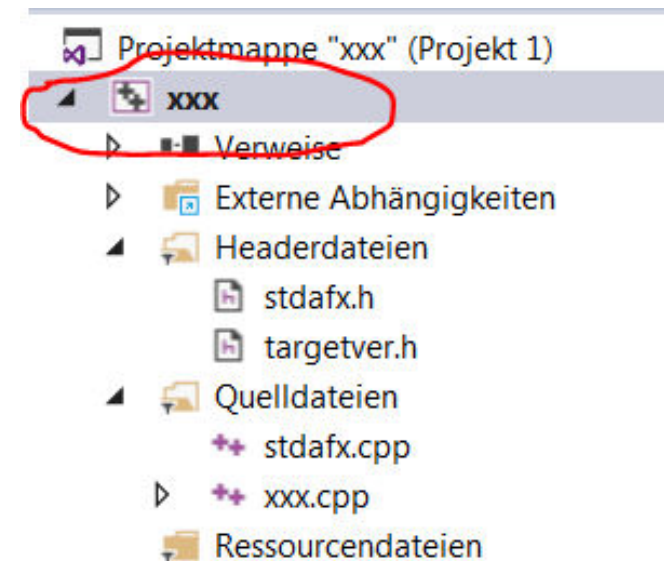
Erstellung einer Konsolenanwendung für Visual Studio 2017 (2)

Die Erstellung einer Konsolen-Anwendung setzt nun vorkompilierte Header-Dateien voraus, eigentlich eine nützliche Sache, verwirrt aber zu Beginn nur.

Deshalb sind noch weitere Einstellung erforderlich.

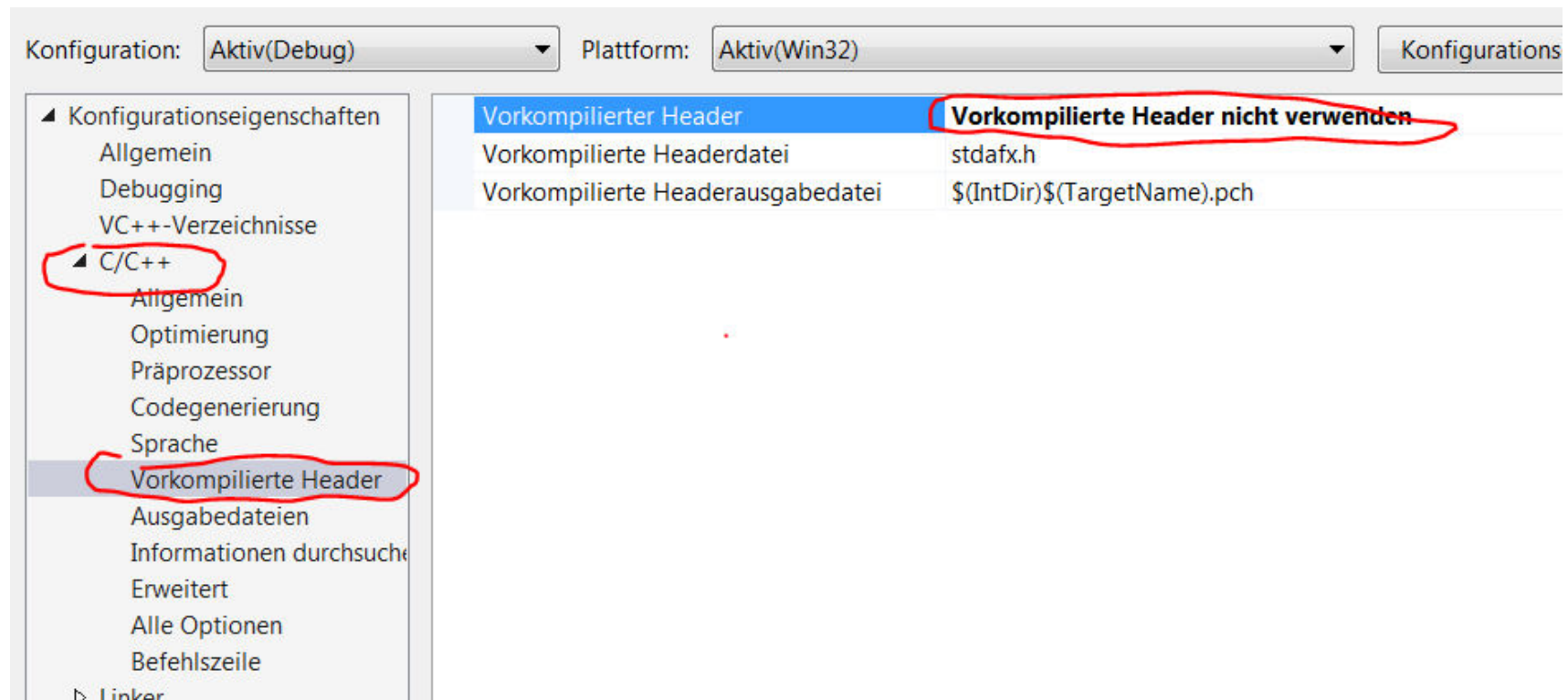
Sie müssen einstellen, dass Sie keine vorkompilierten Header-Dateien verwenden wollen:

Restklick auf den Projektnamen (hier „xxx“) und Einstellungen unten auswählen



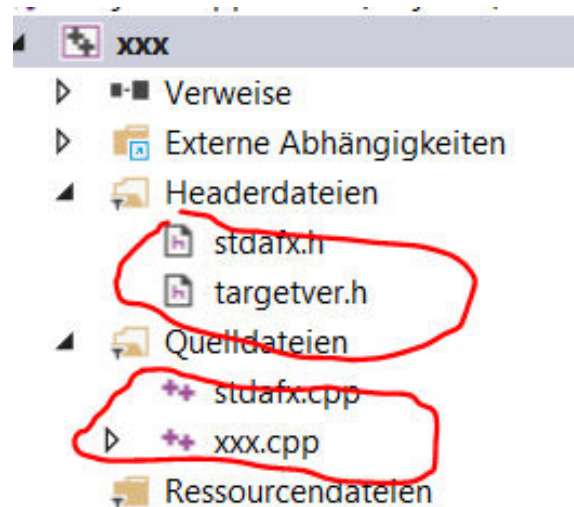
Erstellung einer Konsolenanwendung für Visual Studio 2017 (3)

Wählen Sie nun den Reiter C/C++ aus und hier Vorkompilierte Header und stellen dort ein, dass Sie vorkompilierte Header nicht verwenden wollen.



Erstellung einer Konsolenanwendung für Visual Studio 2017 (4)

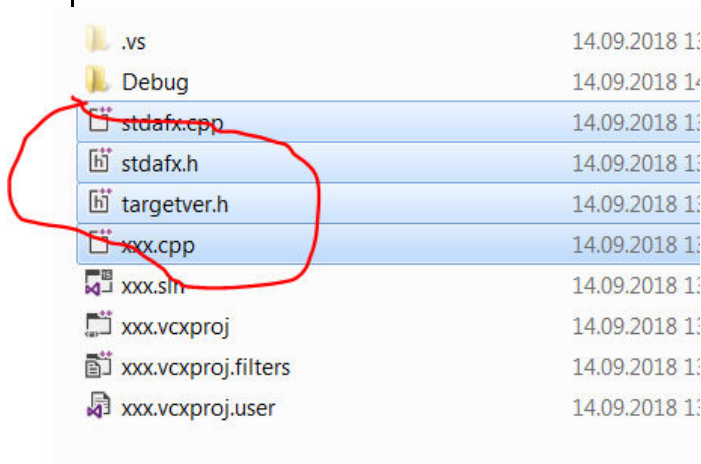
Klicken Sie die vier automatisch erzeugten Dateien an und klicken jeweils auf die Entf-Taste bestätigen Sie das Entfernen aus dem Projekt



Erstellung einer Konsolenanwendung für Visual Studio 2017 (5)

Entfernen Sie diese Dateien nun auch noch physikalisch von der Festplatte. Löschen Sie nur diese vier Code-Dateien. Die anderen Dateien werden weiterhin benötigt

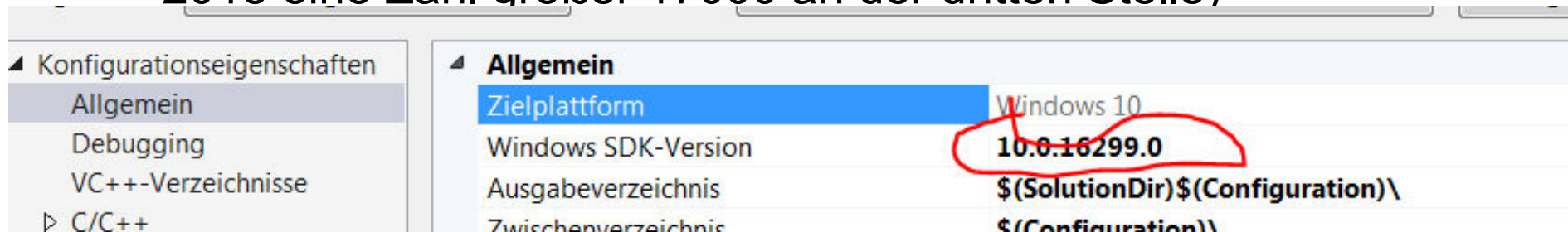
Sie müssen nun noch einstellen, dass Sie keine vorkompilierten Header-Dateien verwenden wollen:



Komische Fehlermeldungen

Sofern Visual Studio nicht erklärbare Fehlermeldungen, wie Probleme beim Include von Systemdateien wie z.B. windows.h angibt, kann es an der fehlerhaften Einstellung der SDK-Version liegen.

Das tritt auf, wenn Sie Projektdaten von anderen Leuten (wie z.B. von mir) übernehmen, die eine andere Visual Studio Version verwenden. Gehen Sie dann auf Einstellungen → Konfigurationseigenschaften → Allgemein → Windows SDK-Version und versuchen dort eine höhere Version anzugeben (aktuell Okt 2018 eine Zahl größer 17000 an der dritten Stelle)



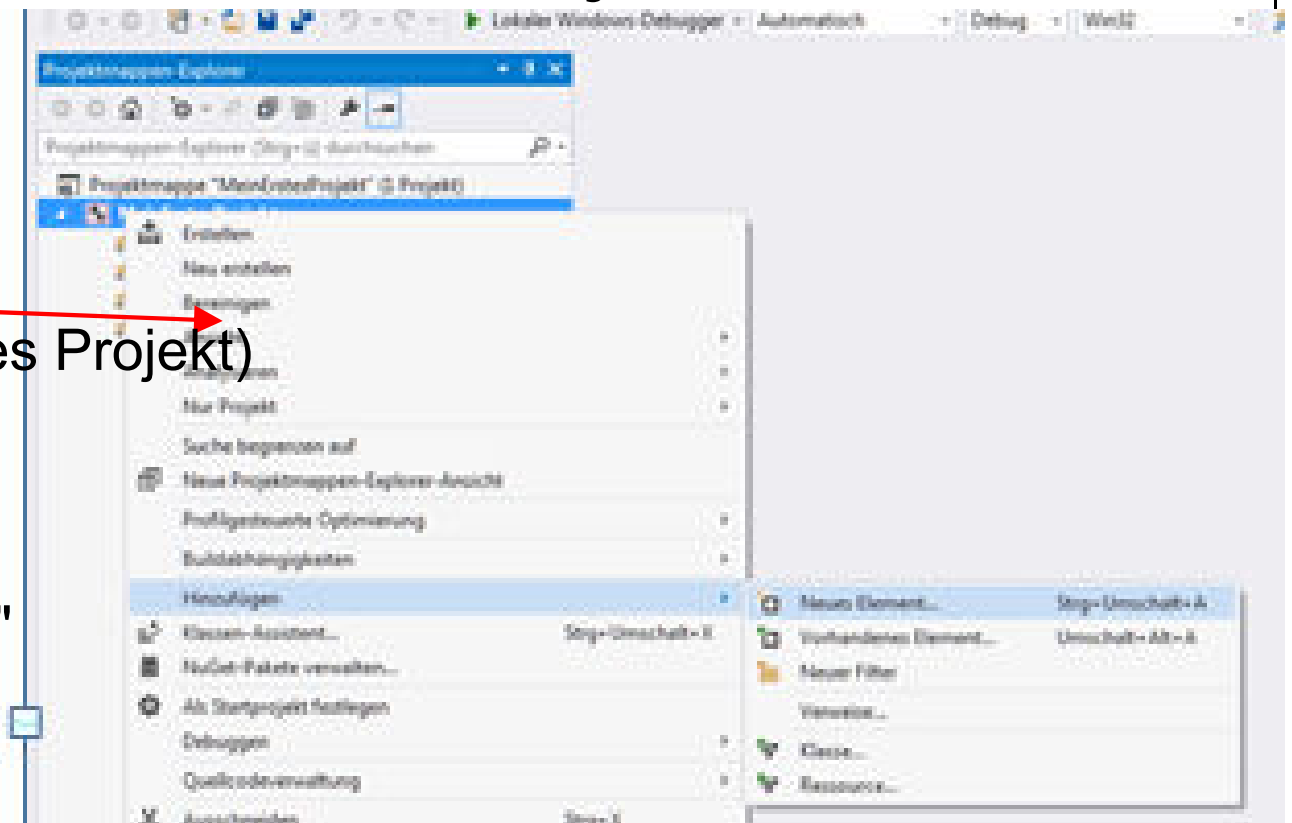
Achtung: Diese Versionsnummer muss Ihnen angeboten werden.



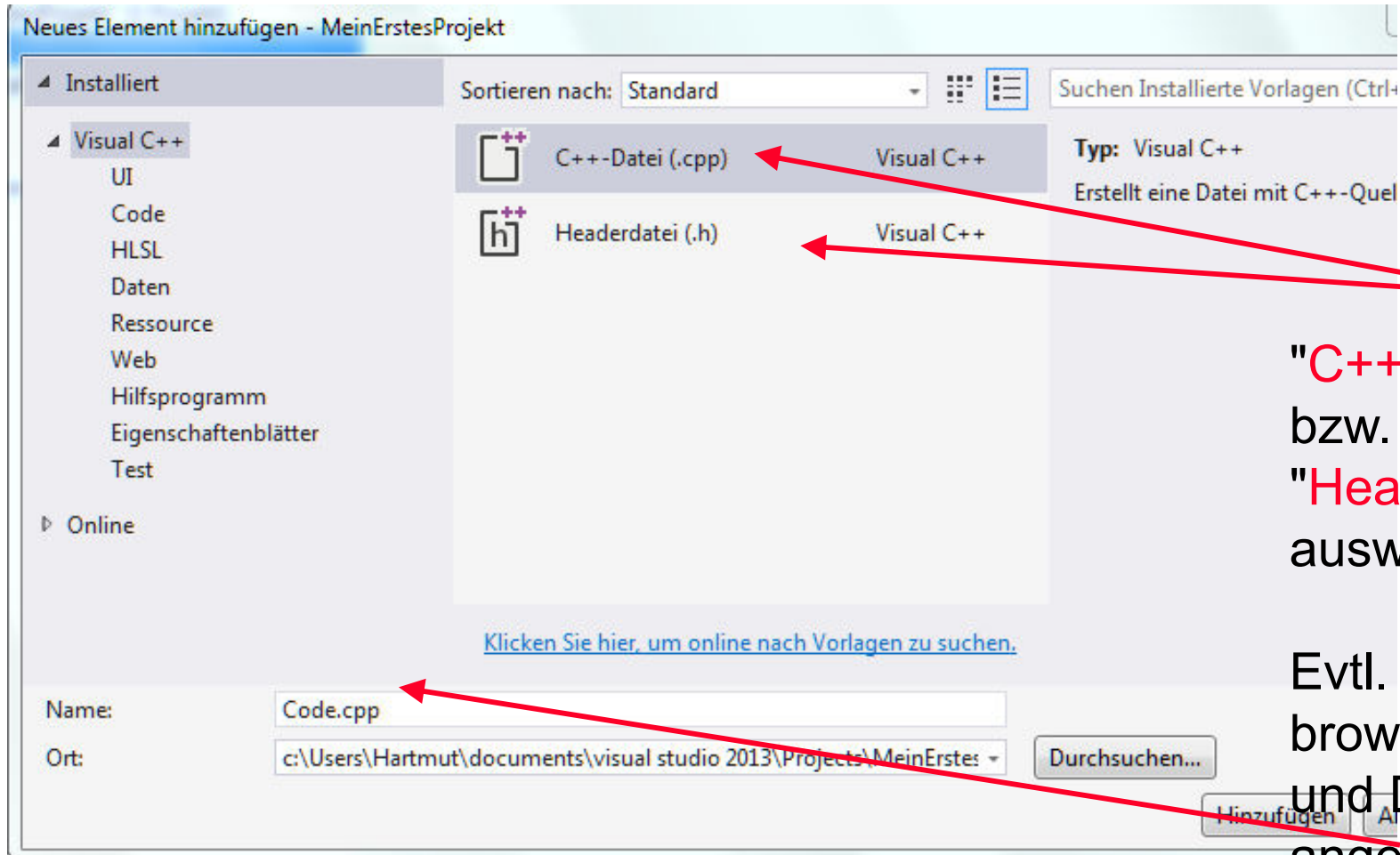
Hinzufügen von neuen Dateien zum Projekt

Projektname (hier MeinErstes Projekt)
rechts anklicken

"Hinzufügen" →
"Neues Element hinzufügen"
auswählen



Hinzufügen von neuen Dateien zum Projekt (2)



"C++-Datei (.cpp)"
bzw.
"Headerdatei (.h)"
auswählen

Evtl. zum Verzeichnis
browsen
und Dateiname
angeben

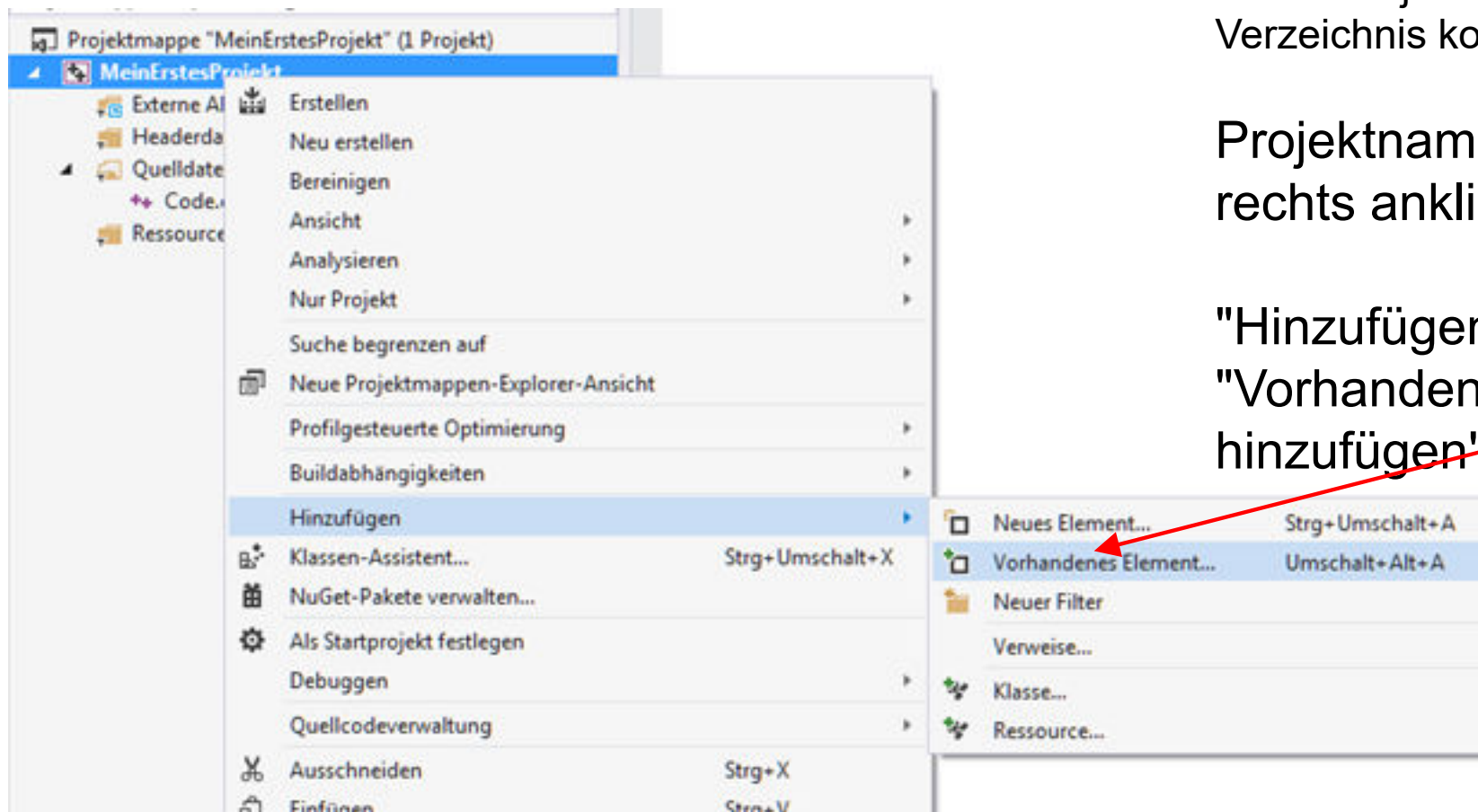
Hinzufügen klicken.

Hinzufügen von existierenden Dateien zum Projekt

Zunächst die
Datei per Explorer
in das Projekt-
Verzeichnis kopieren

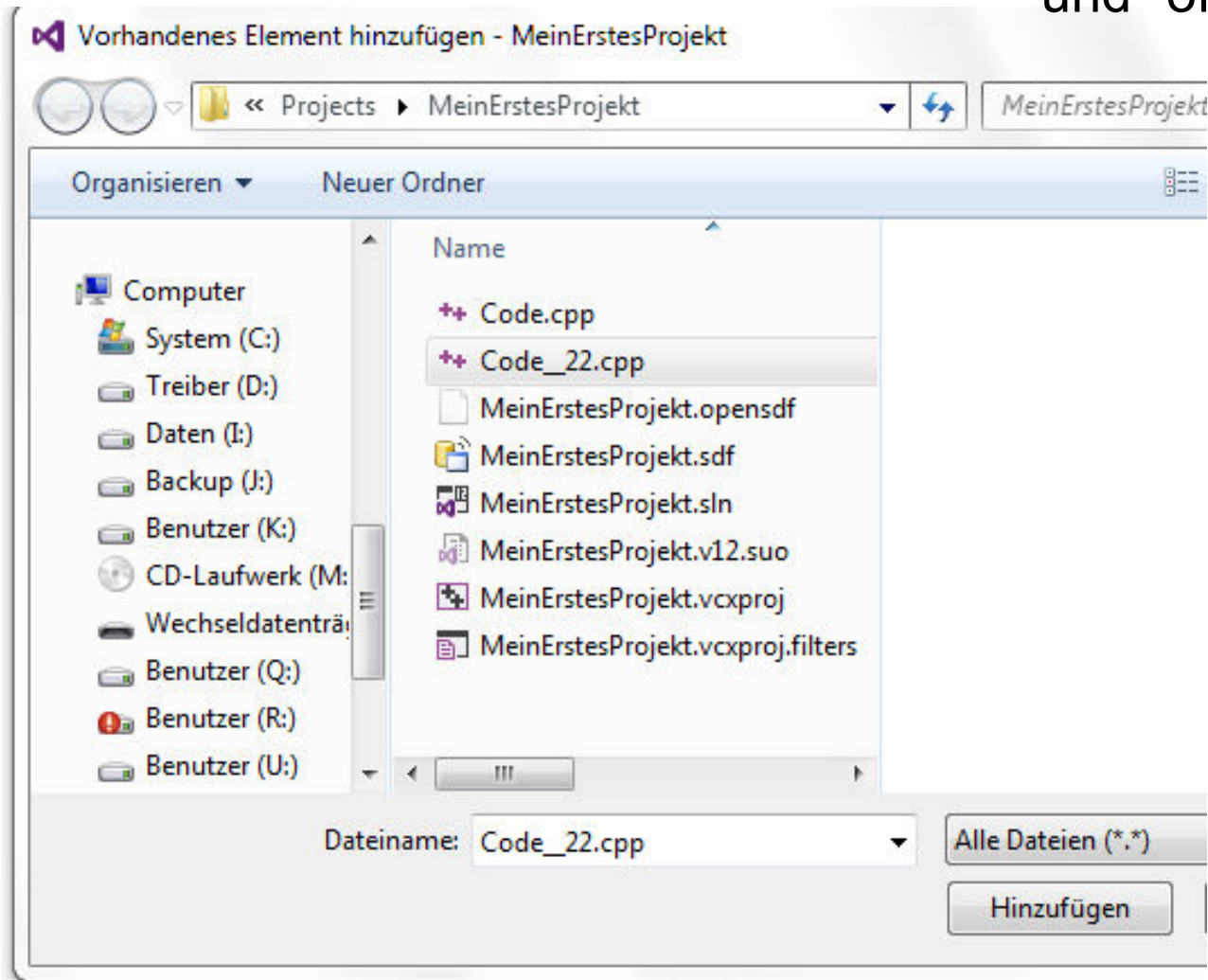
Projektname
rechts anklicken

"Hinzufügen" →
"Vorhandenes Element
hinzufügen" auswählen



Hinzufügen von existierenden Dateien zum Projekt (2)

Datei mit der Maus auswählen
und "öffnen" wählen



Hinzufügen von existierenden Dateien zum Projekt (3)

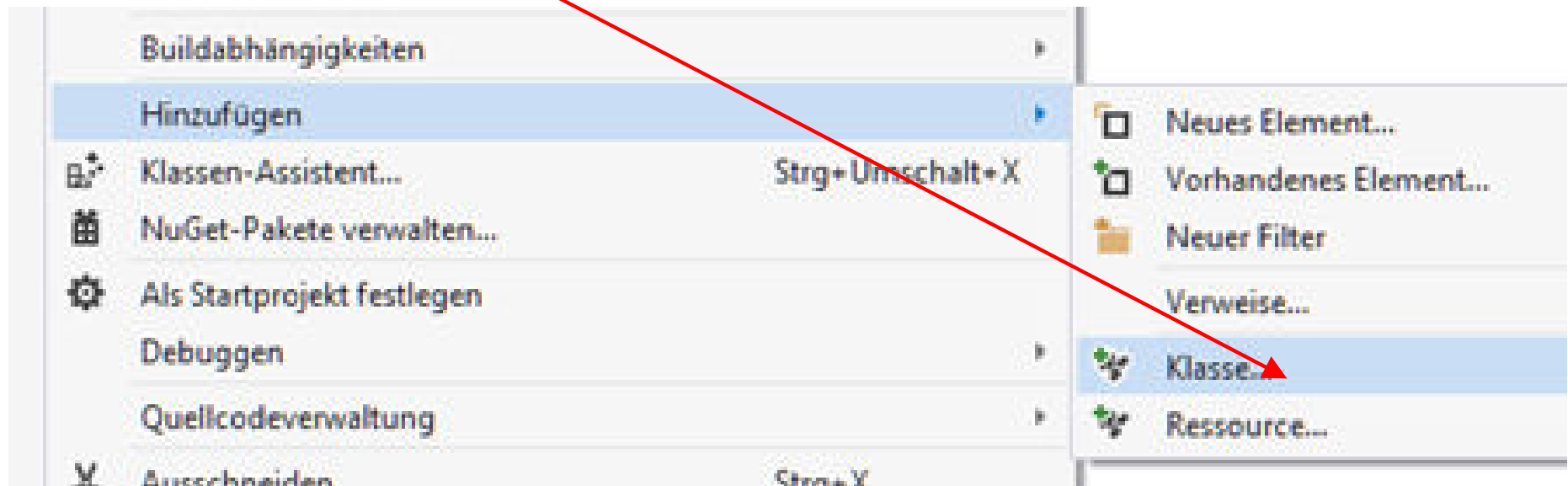
Man kann die Datei aus dem Windows-Explorer, die dem Projekt hinzugefügt werden sollen, aber auch einfach mit der Maus in das Projektfenster ziehen.

Visual Studio ordnet sie dann schon automatisch dem richtigen Ordner (Headerdateien oder Quelldateien) zu.

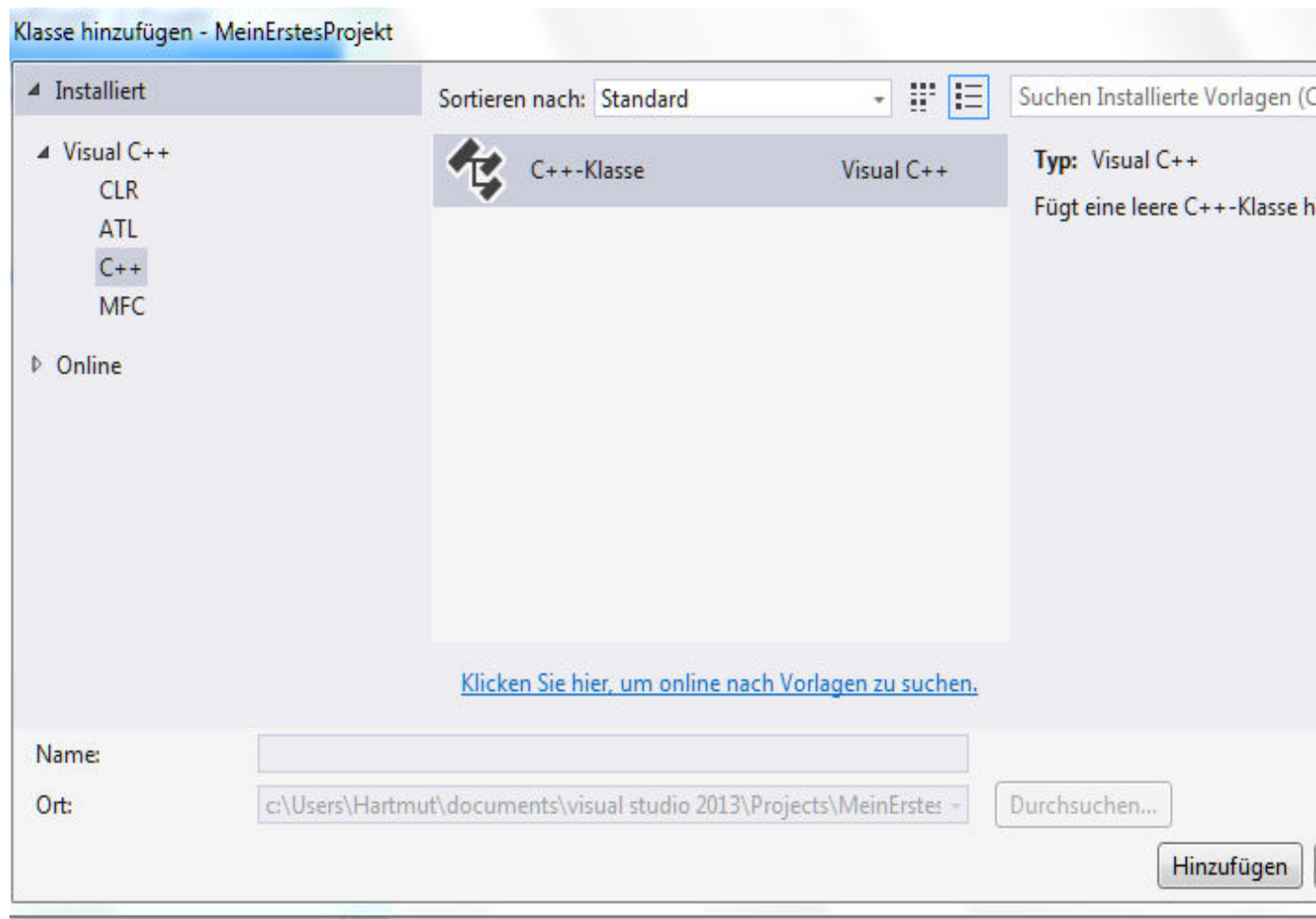
Hinzufügen von neuen Klassen zum Projekt

Projektname
rechts anklicken

"Hinzufügen" →
"Klasse" auswählen



Hinzufügen von neuen Klassen zum Projekt (2)



„C++“ anklicken
„C++-Klasse“
anklicken
und „Hinzufügen“
wählen

Hinzufügen von neuen Klassen zum Projekt (3)

Klassenname: Student

H-Datei: Student.h

CPP-Datei: Student.cpp

Basisklasse:

Zugriff: public

☐ Virtueller Destruktor

☐ Inline

☐ Verwaltet

Fertig stellen Abbrechen

Im Dialog "Klassenname" eingeben und ggf. Name der Header- und Source-Datei anpassen. **"Fertig stellen"** wählen

Debug-Möglichkeiten

Einzelschrittmodus

Taste F10

Hineinspringen in Funktion

Taste F11

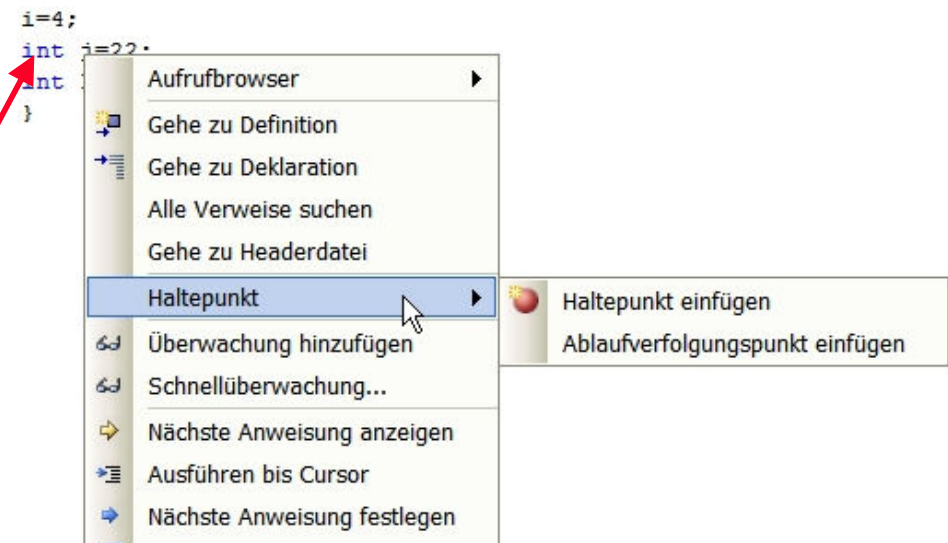
Ausführen bis zum Funktionsende

Tasten Shift F11

Haltepunkt an Cursorposition ein- oder ausschalten: Taste F9

Viele weitere Dinge gehen über den Menüpunkt „Debuggen“

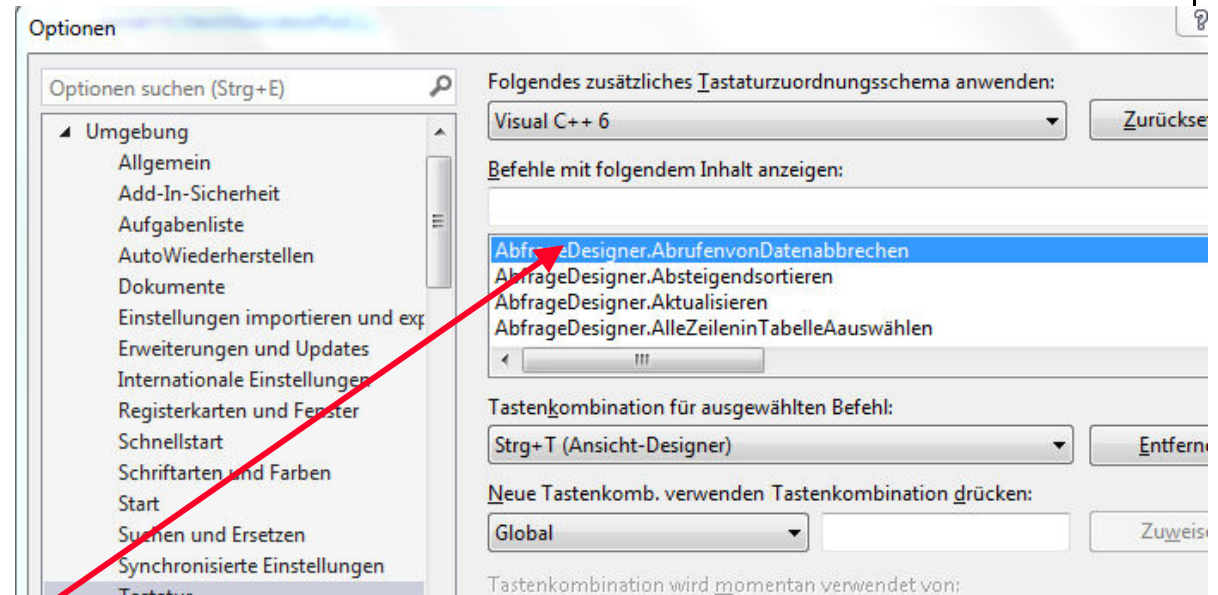
Anklicken einer Anweisung mit rechter Maus öffnet ebenfalls ein Menü zum z.B. Einfügen von Haltepunkten.



Damit die Tasten entsprechend belegt sind, muss noch Visual C++-Tastatur-Belegung eingestellt sein, Siehe nächste Folie

Die verschiedenen Programmierparadigmen von C++

Einstellen der Visual C++-Tastatur Belegung



Außerdem sollten alle Entwickler die gleiche Tastaturbelegung verwenden, sodass F7 z.B. immer den Build-Vorgang startet usw. Hierzu ist die C/C++-Tastatur einzustellen, und zwar unter „Umgebung → Tastatur“ ist „Visual C++ 6“ auszuwählen, siehe obige Abbildung: